TESIS DOCTORAL

Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado

F. Javier Martínez de Pisón Ascacibar

UNIVERSIDAD DE LA RIOJA

TESIS DOCTORAL

Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado

F. Javier Martínez de Pisón Ascacibar

Universidad de La Rioja
Servicio de Publicaciones
2003

Esta tesis doctoral, dirigida por el Doctor D. Joaquín Bienvenido Ordieres Meré, fue leida el 20 de Junio de 2003, y obtuvo la calificación de Sobresaliente cum Laude por Unanimidad.

Edita: Universidad de La Rioja
Servicio de Publicaciones

ISBN 84-688-2870-X

UNIVERSIDAD DE LA RIOJA
DEPARTAMENTO DE INGENIERÍA MECÁNICA

TESIS DOCTORAL

OPTIMIZACIÓN MEDIANTE TÉCNICAS DE MINERÍA DE DATOS DEL CICLO DE RECOCIDO DE UNA LÍNEA DE GALVANIZADO

AUTOR:
D. FRANCISCO JAVIER MARTÍNEZ DE PISÓN ASCACÍBAR

DIRECTOR:
DR. JOAQUÍN BIENVENIDO ORDIERES MERÉ

JUNIO 2.003

UNIVERSIDAD DE LA RIOJA
DEPARTAMENTO DE INGENIERÍA MECÁNICA

OPTIMIZACIÓN MEDIANTE TÉCNICAS DE MINERÍA DE DATOS DEL CICLO DE RECOCIDO DE UNA LÍNEA DE GALVANIZADO

MEMORIA PRESENTADA PARA LA OBTENCIÓN DEL GRADO DE DOCTOR EN INGENIERÍA INDUSTRIAL POR LA UNIVERSIDAD DE LA RIOJA

AUTOR:
D. FRANCISCO JAVIER MARTÍNEZ DE PISÓN ASCACÍBAR

DIRECTOR:
DR. JOAQUÍN BIENVENIDO ORDIERES MERÉ

LOGROÑO, JUNIO DE 2.003

A MI MADRE MARÍA DEL CARMEN
Y A MI ESPOSA ALPHA,
CON TODO MI CORAZÓN, CON TODO MI AMOR

AGRADECIMIENTOS

En primer lugar, me gustaría dar las gracias a Joaquín B. Ordieres Meré, por todo lo que me ha enseñado en estos años, no solo con respecto a las cuestiones técnicas, sino también por lo que considero más importante, el aspecto humano. En cierta forma, gran parte del mérito de este trabajo se lo debo a él, ya que siempre que ha sido necesario me ha abierto un hueco en su "apretada agenda", siempre me ha atendido y aconsejado con excelente sabiduría y prudencia, incluso cuando él tenía un montón de cosas que hacer y yo irrumpía de repente como si solo mis problemas fueran importantes...

También a Eliseo Vergara por toda su disposición y ayuda, que ha sido especialmente importante en las últimas etapas de esta tesis. Además de a mi hermano Eduardo, Fernando Alba, Manuel Castejón, Luis María López, Juan Martín Miruri, Javier Bretón y demás compañeros de trabajo y amigos que me han apoyado y animado en los peores momentos.

Por otro lado, desearía agradecer a todo el personal de Aceralia por todos los consejos, por sus correcciones y eficaz respuesta de todas las dudas que se me planteaban. Especialmente a Juan Antonio González Rodríguez, a Jonatan y demás personas con las que me he relacionado durante este trabajo.

Por último, y sobretodo, quiero agradecer y dedicar esta Tesis, a mi esposa Alpha y a mi madre María del Carmen.

A la primera porque ella ha "sufrido y vivido" cada uno de los pasos que he ido realizando, porque ella me ha apoyado y ayudado con todo su amor y comprensión, sobretodo cuando estaba más desesperado y desilusionado, porque se ha sacrificado mucho por mi, porque me ha enseñado mucho a ser paciente, a perseverar, a ser fuerte...

Y a mi madre, por todo su amor, por todos los esfuerzos y sacrificios que ha hecho por mi, por su lucha por sacarnos adelante, por mil millones de cosas que le debo... A ti, madre querida, te lo debo todo...

¡¡¡Os quiero mucho a las dos!!!

RESUMEN

La búsqueda constante por aumentar la calidad del producto fabricado y reducir los gastos ocasionados por fallos en el proceso de fabricación, son requisitos fundamentales en una planta industrial. Cada vez se buscan métodos y herramientas más eficientes que puedan servir de ayudar en estas tareas. Un ejemplo de ellas, es el *Data Mining*.

Las herramientas de *Data Mining* y estadística multivariante son útiles cuando se dispone de un volumen de históricos importante y de buena calidad. El análisis de los históricos con estas nuevas técnicas puede ayudar en múltiples facetas: control de calidad, identificación de sistemas, determinación de causas en fallos del proceso, detección de anomalías, prevención de fallos, modelización de sistemas, obtención de reglas y patrones de comportamiento, búsqueda de causas y relaciones entre variables, etc.

En esta tesis, se presenta una aplicación de la metodología CRISP-DM [CRI00], para la mejora, dentro de una línea de galvanizado en continuo de bobinas de acero, del tratamiento térmico de la lámina de acero antes de su paso por la inmersión del baño de zinc líquido.

El control y planificación de este proceso de recocido es clave para la mejora de las propiedades de la banda y del recubrimiento.

A lo largo de esta tesis, se muestran los pasos que han llevado a desarrollar:

- Una metodología que, mediante el uso de algoritmos genéticos y redes neuronales, permite la optimización de las curvas de consigna del horno y velocidades de la banda entre bobinas de diferentes dimensiones, reduciendo la diferencia de temperatura esperada de la banda y la real.
- Un clasificador de bobinas según la composición de los aceros que ha resultado ser una excelente herramienta para la predicción de roturas de banda o para detectar otro tipo de problemas debidos a bobinas con aceros anómalos.
- Un sensor-software que proyecta los puntos de operación del horno y que puede ayudar considerablemente en las tareas de visualización de la tendencia de los puntos de operación del horno.

ÍNDICE DE CONTENIDOS

CAPÍTULO 1

INTRODUCCIÓN

1.1 INTRODUCCIÓN

Hoy en día, uno de los objetivos del mundo industrial se orienta hacia la venta de productos de mayor valor añadido.

En el caso de las grandes empresas siderúrgicas, uno de estos productos son los aceros recubiertos mediante electrocincado y galvanizado[1], ya que éstos, debido a sus propiedades anticorrosión, están experimentando una creciente demanda en sectores tales como la automoción, la fabricación de electrodomésticos y la construcción.

Así, debido a esta búsqueda de mayor valor añadido y al aumento de las exigencias de los clientes, las empresas plantean una estrategia de mejora continua en cada una de las fases de que consta el proceso de galvanizado.

Una de estas fases corresponde con el tratamiento térmico que se realiza a la lámina de acero, antes de su paso por la inmersión del baño de zinc líquido. Ésta zona está compuesta por una serie de hornos que elevan la temperatura de la banda hasta una temperatura objetivo, manteniéndola durante un cierto tiempo, para pasar, posteriormente, a enfriarla siguiendo unas curvas previamente establecidas para cada tipo de acero.

El objetivo de este ciclo térmico es múltiple:

- Primero, se realiza un precalentamiento con una limpieza previa de aceites de laminación y otros contaminantes en un horno de baja temperatura (450-800ºC). La misión es limpiar la banda y calentarla hasta una temperatura de 450ºC, según espesores y ciclos térmicos. También se busca reducir la capa de óxido superficial a un espesor controlado. La limpieza se realiza por volatilización de los aceites de laminación, arrastre mecánico de partículas de suciedad y reducción del óxido

[1] El recubrimiento galvanizado es un método de protección contra la corrosión muy valorado consistente en la disposición de moléculas de metal protector sobre el material a proteger. Este metal protector acostumbra a ser el zinc por sus excelentes y muy conocidas propiedades ante la corrosión, y se suele acompañar en menor medida de aleantes para mejorar las propiedades de adherencia.

superficial al pasar la banda entre dos filas de mecheros cuya llama, reductora y abierta, calienta las zonas hasta una temperatura máxima de 1.260ºC.

- Después, se aumenta la temperatura de la banda hasta un valor superior a 780ºC para recristalizar el metal endurecido que sale de la laminación en frío y homogeneizar la estructura cristalina. También se trata de eliminar la capa superficial de óxido.
- Una vez alcanzada la temperatura buscada, se mantiene durante un tiempo para permitir un engrosamiento del grano.
- La banda, que alcanzó su máxima temperatura, se enfría de modo controlado y lento, para conseguir unas características mecánicas adecuadas buscando mejorar la textura del metal.
- Por último, se realiza un enfriamiento rápido para llevar a la banda hasta una temperatura más adecuada para realizar el recubrimiento, y para preparar el acero al tratamiento de envejecimiento congelando una cantidad máxima de carbono en sobresaturación.

Como se puede observar, el control de la temperatura de la banda en cada una de estas fases es fundamental si se quiere obtener unas propiedades adecuadas y un buen recubrimiento. Este control se realiza actualmente mediante la aplicación de un modelo matemático del horno basado en ecuaciones diferenciales.

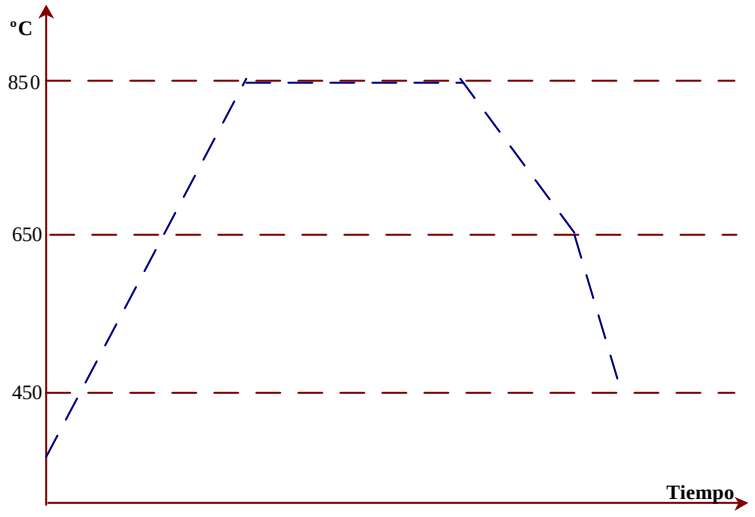


Figura 1. Ejemplo de un ciclo de recocido.

1.2 OBJETIVOS DE LA TESIS

En esta tesis se trata de mejorar, mediante el uso de herramientas de análisis multivariante y de Minería de Datos, la planificación y control del horno de una línea de galvanizado de bobinas de acero. Fundamentalmente, se busca:

- **Comprender el sistema a partir de los históricos del proceso.**
- **Explicar y predecir roturas de banda u otras paradas.**
- **Mejorar el control y la planificación de las curvas de consigna del horno para cada tipo de bobina, reduciendo el error entre la temperatura real de la banda y la esperada según el ciclo térmico que le haya sido asignada.**

Para ello, se plantean los siguientes objetivos generales (los cuales son ampliados y descritos con más profundidad en el capítulo 4):

- **Obtención de conocimiento oculto del proceso mediante técnicas de minería de datos e inteligencia artificial: causas de paradas o fallos, variables más importantes, relación entre variables, etc.**
- **Creación de nuevos modelos, basados en redes neuronales u otras técnicas de inteligencia artificial, para la optimización de las consignas de control del horno.**
- **Desarrollo de herramientas para la planificación y prevención de roturas de banda o paradas.**
- **Creación de técnicas de monitorización de los puntos de operación del horno y generación de alarmas.**

Esto permitirá mejorar considerablemente:

- **El conocimiento que se tiene del proceso de galvanizado en continuo**.
- **Las características mecánicas finales de la banda y la calidad final del recubrimiento ya que éstas dependen en gran medida de la planificación y control de las curvas de recocido a las que son sometidas cada una de las bobinas.**
- **La reducción de paradas imprevistas.**
- **La detección y prevención de problemas que puedan surgir.**
- **La seguridad del proceso.**
- **La forma de clasificar las bobinas.**
- **Etc.**

1.3 ESTRUCTURA DE LA TESIS

Esta tesis se desarrolla en nueve capítulos (además de un apartado de referencias), cuya estructura, tras este *primer capítulo* de introducción, es la siguiente:

- En el *capítulo segundo* se describe con detalle el proceso en que se divide la línea de galvanizado, haciendo especial hincapié en el sistema de control del horno y en el modelo teórico del mismo.
- En el *capítulo tercero* se ofrece una revisión del concepto del *Data Mining*, las metodologías y técnicas más comunes, organizadas de una nueva forma según las últimas tendencias y orientadas hacia el uso de la mejora de procesos industriales. Se describe brevemente, la metodología escogida: CRISP-DM [CRI00].
- En el *capítulo cuarto* se describen con profundidad los objetivos, resultados esperados y criterios de éxito a utilizar, tal y como especifica la primera fase de la metodología *CRISP-DM*.
- El *capítulo quinto* entra de lleno en las dos siguientes fases: el análisis y preparación de los datos. Aquí se describen los primeros resultados obtenidos en los análisis exploratorios de los datos, la preparación y preprocesado de los mismos, las preselección de variables y creación de otras nuevas, y las técnicas utilizadas para el desarrollo de una base de datos más consistente.
- En el *capítulo sexto* se realiza un estudio exhaustivo mediante técnicas de minería de datos y de análisis multivariante para: determinar la relación entre variables, búsqueda de conocimiento oculto, búsqueda de las causas que generan paradas o errores de control, análisis del comportamiento del sistema según el "modo de uso", etc. También se describe el proceso de creación y validación del clasificador de bobinas.
- En el *capítulo séptimo*, se entra en la fase de modelizado, donde se desarrolla un sensor-software para la proyección de los puntos de operación del horno y la generación de alarmas. Además, se describen los pasos que han consistido en la creación de una serie de modelos no lineales, mediante redes neuronales, para la predicción de las curvas de consigna óptimas de temperatura y velocidad de la banda; y para modelar el comportamiento de la misma. Se demuestra, cómo el uso de estos modelos y la optimización de las curvas de consigna mediante algoritmos genéticos, puede ayudar considerablemente a reducir los errores entre las temperaturas de la banda reales y esperadas.
- En el *capítulo octavo* se presentan los resultados de validación de los clasificadores y redes neuronales, con una nueva base de datos.
- Y por último, en el *capítulo noveno*, se desarrollan las conclusiones, la forma de implementar los resultados obtenidos y las líneas de futuro.

CAPÍTULO 2

DESCRIPCIÓN DEL PROCESO INDUSTRIAL

2.1 INTRODUCCIÓN

En este capítulo, se pretende realizar la descripción completa de la línea de galvanizado continuo por inmersión, haciéndose un mayor énfasis en el sistema de control del horno, por ser el objeto de este trabajo. Esta descripción, con modificaciones menores, es válida para la práctica totalidad de líneas de galvanizado continuo por inmersión instaladas en todo el mundo, pues todas ellas son prácticamente similares.

A grandes rasgos se puede describir de la siguiente manera: El primer paso en la línea es la formación de una banda continua a partir de las bobinas de acero procedentes del tren tándem. A continuación, la banda atraviesa una serie de hornos en los que recibe un tratamiento térmico, paso previo a su inmersión en el baño de zinc líquido. **Este tratamiento es esencial para la mejora de las propiedades de la banda.**

Éste tratamiento, consiste en elevar la temperatura de la banda hasta una temperatura objetivo, manteniéndola durante un cierto tiempo, para pasar a enfriarla posteriormente siguiendo unas curvas previamente establecidas para cada tipo de acero.

De este baño, la banda sale verticalmente pasando entre las cuchillas de aire que regulan el espesor del recubrimiento. El control del espesor de recubrimiento en las líneas de galvanizado en continuo se ha abordado con técnicas muy diferentes [VER99][JAC95][TAK92]. En la actualidad, el sistema empleado en la mayoría de las líneas de galvanizado en continuo es el conocido como *control por cuchillas de aire*. La banda continua de acero, tras sumergirse en el baño de zinc fundido, pasa entre dos toberas denominadas cuchillas por su especial forma- que soplan aire a presión contra el recubrimiento de zinc cuando aún es líquido. La posición de dichas cuchillas y la presión del aire proyectado contra la banda, entre otros factores, determina la cantidad de zinc que permanecerá sobre la banda. El resto de zinc escurrirá hacia el baño de metal líquido.

Por último, atraviesa una serie de pasos refrigerantes donde recibe un tratamiento químico de pasivizado para ser bobinada de nuevo.

A continuación, en este capítulo se describe el proceso de galvanizado completo y el modo de control del horno.

2.2 EVOLUCIÓN DEL PROCESO DE GALVANIZADO

Los métodos de aplicación del zinc al hierro y de superficies de acero se han ido modificando a lo largo de los años, cambiándose a su vez las instalaciones requeridas para su producción [JAG93][SAM93][BAU94][JON94][HAY93][EAN94]. Sin embargo, hasta 1936 los elementos básicos de los procesos de galvanizado, no experimentaron modificaciones fundamentales. El método empleado hasta entonces consistía en la limpieza por decapado de las superficies a galvanizar, el "*fluxing*" de la superficie para facilitar la humectabilidad con el zinc y el precalentamiento del material a galvanizar hasta la temperatura de recubrimiento.

Con el empleo de chapa laminada en frío como metal base para la chapa galvanizada, fueron añadiéndose al proceso otros pasos como la eliminación de aceites de laminación mediante limpieza alcalina y el recocido del acero antes del recubrimiento.

Siempre ha sido admitido por los expertos en galvanización, que **la preparación superficial del metal base, el control de la temperatura de recubrimiento y la composición del baño, son los factores que afectan en mayor medida a los productos galvanizados**, y cuando son controlados estos tres factores en sus valores óptimos, se puede conseguir un producto recubierto de alta calidad.

En 1936 se dio un paso importante en los procesos de galvanizado continuo con la instalación en *Butler* (Pennsylvania) de la primera línea basada en el proceso de recubrimiento SENDZIMIR. Este proceso consigue mejoras importantes en los tres puntos anteriormente citados, los cuales son controlados fácilmente. La incorporación del recocido a la propia línea, elimina un proceso intermedio al recibir las bobinas a galvanizar directamente del tren de laminación en frío.

Básicamente, el proceso consta de:

- Limpieza previa de aceites de laminación y otros contaminantes en un horno de baja temperatura (400ºC). En este horno se forma una capa de óxido superficial de espesor controlado.
- Un recocido posterior en un horno de atmósfera reductora (25% N_2-75% H_2) que elimina la capa superficial de óxido y coloca la banda a la temperatura de inmersión.
- Inmersión en el baño de zinc.

Posteriormente se incorporó en 1.951 el proceso NO OXIDANTE a las líneas SENDZIMIR, que consiste en realizar la limpieza de la banda en horno no oxidante a temperaturas próximas a 1.250ºC. Esto permitió reducir la longitud del horno de recocido y reducción al no ser ya necesario tanto tiempo de permanencia en el mismo.

2.2.1 PROBLEMAS DEL PROCESO DE GALVANIZADO EN CONTINUO

La calidad de la chapa galvanizada, es el punto de mira de gran número de mejoras introducidas a través del tiempo en las líneas continuas de galvanizado. Se han ido introduciendo nuevas tecnologías encaminadas a la mejora de los siguientes aspectos:

- Adherencia del recubrimiento.
- Aptitud de la deformación.
- Tratamiento de acabado.

Sin embargo, subsisten algunos problemas como la peor deformación de la chapa galvanizada respecto a la chapa laminada en frío [ROD00].

Sucedía que la chapa galvanizada con estrella normal –figura que forma el zinc al cristalizar sobre la banda de acero-, era difícil de pintar y por esta razón se han lanzado al mercado dos clases de chapa de acabado mate, es decir, sin estrella, aunque hoy día, pintar la chapa de acabado normal ya no es ningún problema.

También parece claro que la resistencia a la corrosión en circunstancias normales, viene determinada por el espesor del recubrimiento, supuesto este constante.

Si el recubrimiento es uniforme y del mismo espesor, la vida de la chapa galvanizada está determinada por circunstancias naturales. La siguiente da una idea de esto.

VIDA EN AÑOS DE PRODUCTOS GALVANIZADOS							
		TIPO DE ATMÓSFERA					
Espesor (mm)	Peso (g/m2)	Rural	Marina Tropical	Marina Templada	Submarina	Urbana	Industrial
9.14	610.31	50	40	35	30	25	15
5.84	381.45	35	30	25	20	17	9
4.57	305.16	25	20	15	12	10	7
2.79	183.09	10	8	7	5	4	3
1.68	112.91	7	6	5	4	3	2
1.12	76.29	5	4	3	3	2	1

Tabla 1. Vida en años de productos galvanizados (Fuente: ACERALIA Corporación Siderúrgica *[CSI85]*).

En esta tabla, [CSI85], el peso de recubrimiento indica el total de ambas caras de la chapa, como es habitual al referirse a productos galvanizados. Por otro lado, se observa claramente que la vida de la chapa es proporcional al peso o espesor de recubrimiento. Dado que la resistencia a la corrosión de la chapa viene dado por la zona con menor espesor de recubrimiento de zinc, la uniformidad del mismo es de vital importancia.

2.3 DESCRIPCIÓN DE LA LÍNEA

El material de partida son bobinas de acero procedentes de laminación. Las bobinas tienen ya el espesor requerido procedente del tren del laminado en frío. Para transformar las bobinas en una banda continua, éstas se debobinan, despuntando la cabeza y la cola de la bobina, y se sueldan a solape. Las impurezas arrastradas de la laminación se eliminan mediante el calentamiento en atmósfera no oxidante. Una vez limpia la banda, se somete a un ciclo de calentamiento y enfriamiento, conocido como recocido, para mejorar las características mecánicas de la banda.

A continuación, la banda se sumerge en un pote con zinc fundido quedando revestida de este metal. El control del revestimiento se realiza proyectando aire a presión sobre la banda recubierta.

Si durante el almacenamiento o transporte se produjeran condiciones suaves de oxidación se formarían sobre la superficie de la banda manchas blancas correspondientes a la presencia de óxido. Este fenómeno no supone una disminución de las características anticorrosivas del recubrimiento, pero si desluce el aspecto de la banda. Para prevenir su aparición, se somete a la banda a un tratamiento superficial de ácido crómico. Después de este proceso, la banda es aplanada obteniéndose el producto acabado bien en forma de bobinas o de chapas cortadas.

Las principales características una línea de galvanizado tipo son:

Material		**Banda de acero reducido en frío y banda temperizada**
Espesor	mín. máx.	0.3mm (0.012”) 4.2 mm (0.164”)
Ancho	mín. máx.	609.6 mm (24”) 1270 mm (50”)

Tabla 2. Características del Material de la Línea.

Bobinas de entrada	
Diámetro interior	508 mm (20”)
Diámetro exterior	1854 mm (73”)
Pesos	20385 Kg (45000 lb.)

Tabla 3. Características de las bobinas de entrada.

Bobinas de salida		
Diámetro interior		508 mm (20”)
Diámetro exterior		1854 mm (73”)
Pesos		20385 Kg (45000 lb.)+peso de recubrimiento
Espesor de recubrimiento	Min	90 gr/m^2
	Max	900 gr/m^2

Tabla 4. Características de las bobinas de salida.

Chapas de salida		
Longitud de la chapa	Min. Max.	965 mm (38") 6096 mm (240")
Peso del paquete		9060 Kg. (20000 lb.)
Peso del paquete de rechazos		6795 Kg. (15000 lb.)

Tabla 5. Características de las chapas de salida.

La línea puede considerarse dividida en tres secciones:

- **Sección I o de Entrada** que comprende las unidades de debobinado, enderezado, corte y soldadura.
- **Sección II o de Proceso** que comprende la unidad de tensión, horno de proceso, unidad de revestimiento, control de revestimiento, equipo de enfriamiento, unidad de tratamiento químico y unidades de aplanado.
- **Sección III o de Salida** que comprende el equipo de bobinado y de salida de bobinas, tijera, aplanadoras, inspección, apiladora y salida de chapas de rechazo, apilado y salida de chapas de 1ª calidad.

Velocidad de la línea	
Sección I (entrada)	5.5 a 114 m/min (18 a 375 f.p.m)
Sección II (proceso)	5.5 a 114 m/min (18 a 375 f.p.m)
Sección III (Rebobinado y corte)	5.5 a 114 m/min (18 a 375 f.p.m)
Sección III (salida de chapa)	15 a 137 m/min (50 a 450 f.p.m)
Velocidad de enhebrado	23 m/min (75 f.p.m)

Tabla 6. Velocidades de la línea en sus diferentes secciones.

2.3.1 SECCIÓN DE ENTRADA

La línea dispone de *skids* para carga de bobinas, que pueden almacenar varias bobinas del ancho máximo.

Las dos debobinadoras son del tipo mandril expandible y disponen de un carro desplazable, transversal y vertical, por medio de un sistema hidráulico. La banda se endereza a la salida de las debobinadora en dos aplanadoras. Una unidad doble de tiro, tira de la banda para su enhebrado.

La tijera doble nº1 de corte hacia atrás, operada neumáticamente, sirve para despuntar los extremos de las bobinas y eliminar la banda fuera del espesor requerido. A la entrada de la tijera se encuentra una unidad doble de rodillos de tiro.

A continuación, se dispone de una soldadora del tipo solape con punzón, que ejecuta la unión de los extremos de la bobina con la cabeza de la siguiente, para producir banda continua. El punzonado tiene por objeto facilitar la detección de la soldadura. A la entrada de la soldadura está localizada una unidad de rodillos de tiro.

2.3.2 SECCIÓN DEL PROCESO

La tensión de la banda se mantiene a través de los rodillos de tensión (brida) nº 1.

Al objeto de mantener constante la velocidad en esta sección, se dispone de un almacenamiento de banda, realizado por el carro de lazos nº 1. Dos rodillos guía permiten el correcto centrado de la banda. A continuación se dispone la unidad de tensión nº 2 y de una unidad para medir la tensión de la banda.

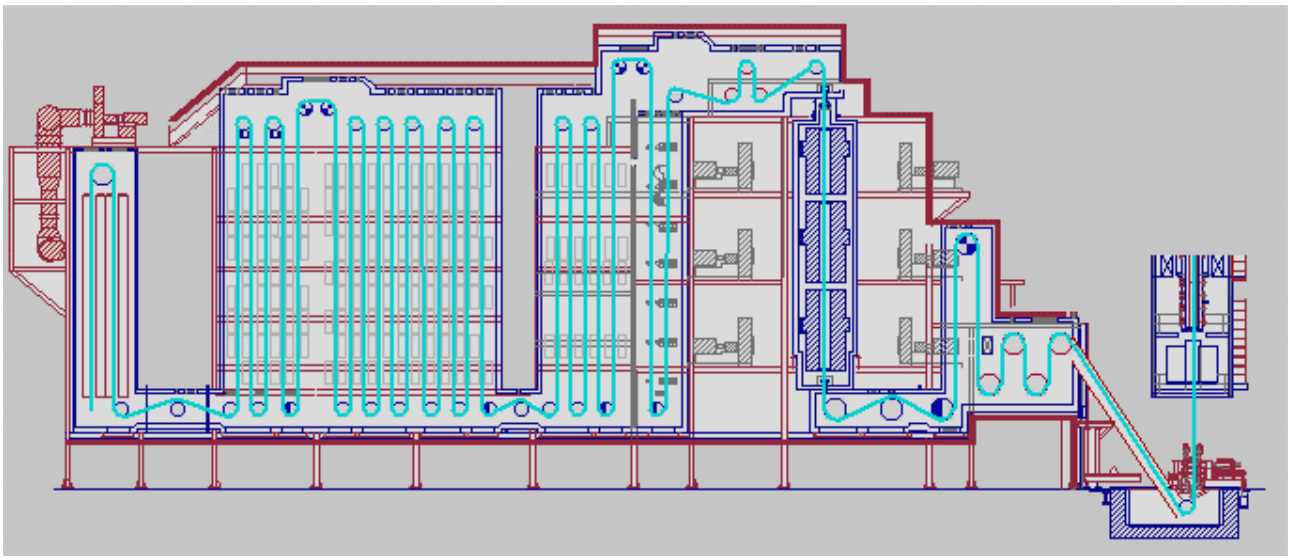


Figura 2. Sección del horno de galvanizado.

El horno de proceso puede someter a la banda a diferentes tratamientos térmicos: normalizado, recocido y *full-hard*, aunque es el segundo el más habitual en esta producción. Se puede dividir a su vez, en cinco zonas.

2.3.2.1 ZONA DE PRECALENTAMIENTO Y LIMPIEZA (F-1)

La misión de esta sección es limpiar la banda, que procede directamente del Tren Tándem, y calentarla hasta una temperatura de 450-800ºC, según espesores y ciclos térmicos.

La limpieza se realiza por volatilización de los aceites de laminación, arrastre mecánico de partículas de suciedad y reducción del óxido superficial, al pasar la banda entre dos filas de mecheros cuya llama, reductora y abierta, calienta las zonas hasta una temperatura máxima de 1.260ºC.

Esta sección tiene un total de 24 mecheros de gas de hornos de cok, de mezcla de tobera. De estos mecheros, 12 están situados sobre el paso de banda y los otros 12 en la parte inferior, colocados todos ellos al tresbolillo. A efectos de regulación y control, el conjunto se divide en cuatro zonas de 6 mecheros cada una.

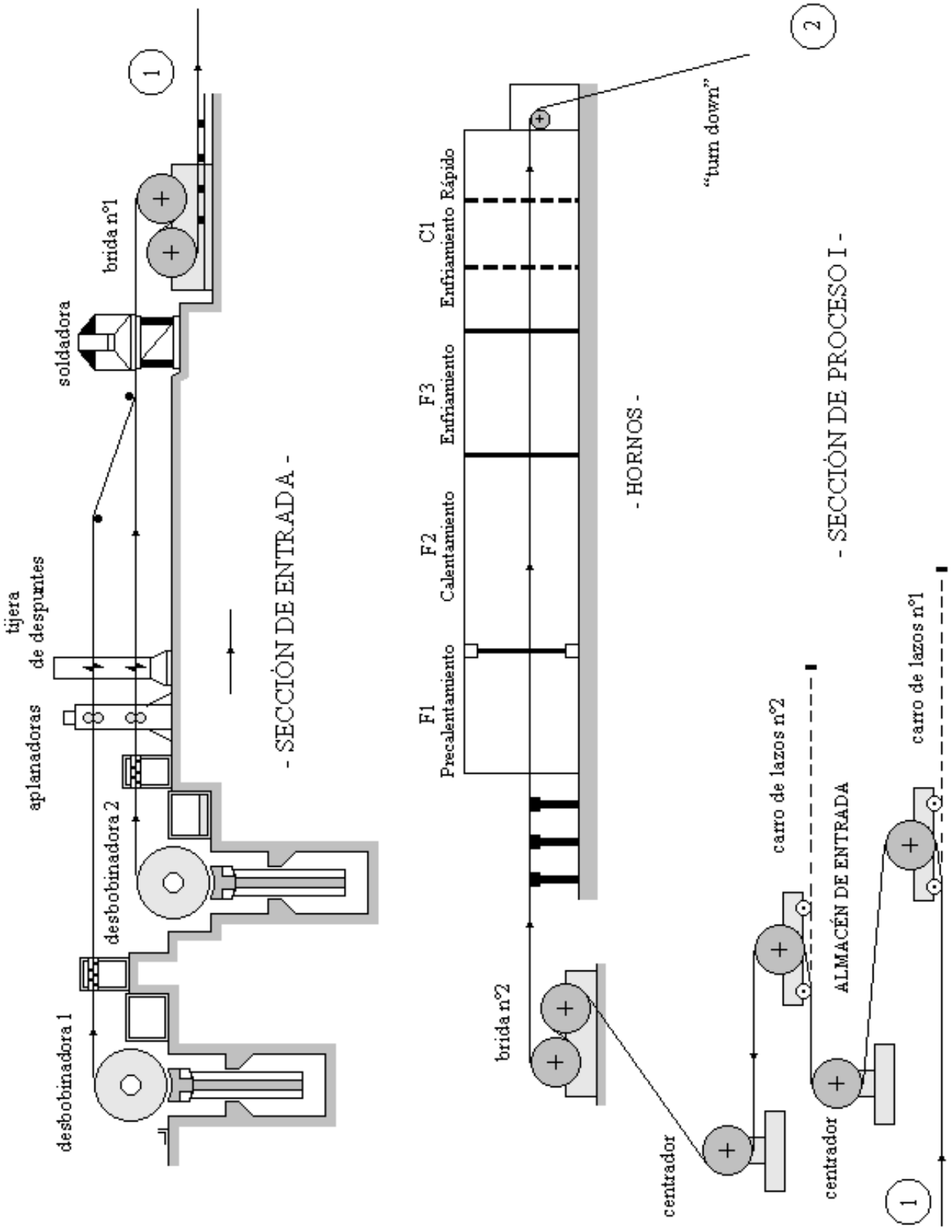


Figura 3. Sección de entrada de la línea de galvanizado (fuente: [VER99]).

Cada zona tiene dos termopares de Pt/PtRh 13%, uno de control y otro de seguridad. Un instrumento electroneumático regula la temperatura, de acuerdo con la señal que recibe del termopar de control, actuando sobre las válvulas de aire y gas de la zona de mecheros.

La temperatura de la banda se mide mediante un pirómetro de radiación. Un instrumento electroneumático que recibe la señal eléctrica del pirómetro y la transforma en señal neumática. Mediante esta señal gobierna el funcionamiento de las cuatro zonas de mecheros en función de las variaciones respecto a la temperatura de banda consignada en el instrumento.

Conseguir una perfecta limpieza de banda es fundamental para lograr una buena adherencia. Si esta sección no consigue limpiar la banda, o si, por el contrario, se oxida por una mala composición de la llama (llama oxidante), será imposible lograr un recubrimiento de calidad.

2.3.2.2 ZONA DE CALENTAMIENTO (F-2)

La banda limpia procedente de F-1, y a una temperatura de 450-800ºC, es calentada en esta sección hasta una temperatura superior a 780ºC. El objetivo de este calentamiento es recristalizar el metal endurecido que sale de la laminación en frío y homogeneizar la estructura cristalina. Para ello, se hace pasar la banda por un horno estanco bajo una atmósfera neutra no oxidante (gas 95% de N_2 y 5% de H_2).

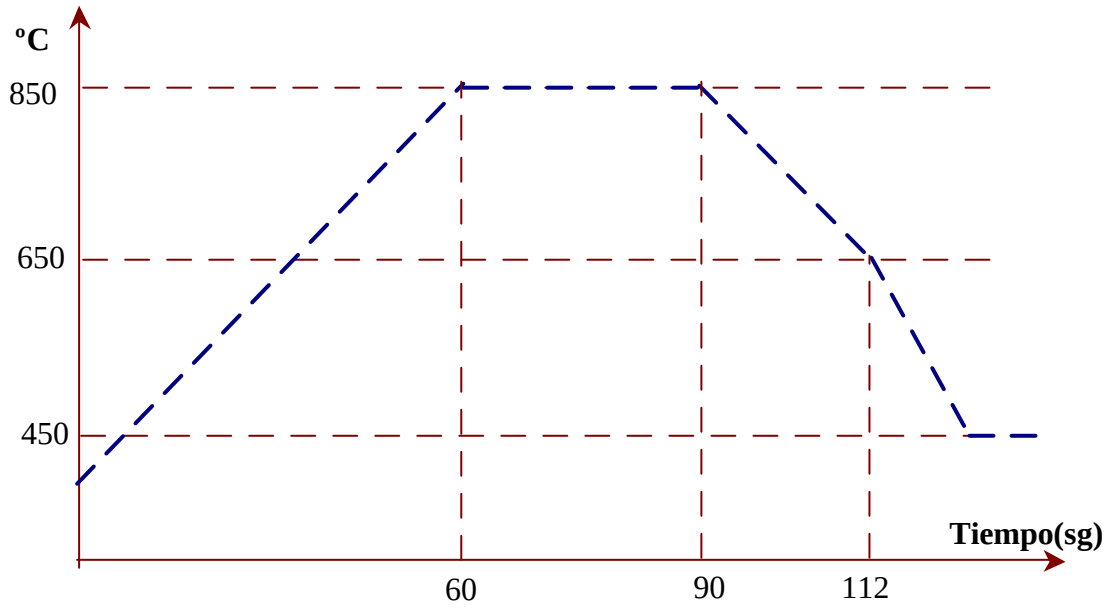


Figura 4. Ejemplo de un ciclo de recocido.

El calentamiento se realiza por radiación. El sistema de calentamiento está compuesto por 52 tubos radiantes en U equipados con mecheros de gas de hornos de cok de 75.000 Kcal. La mitad de estos tubos están situados sobre el paso de banda y la mitad restante en la parte inferior. A efectos de regulación y control, los tubos radiantes se encuentran divididos en cuatro zonas. En cada zona hay dos termopares de chromel-alumel, uno de seguridad y otro de control. Un instrumento

electroneumático de regulación por zona, recibe la señal eléctrica del termopar de control y la transforma en señal neumática que actúa sobre la válvula de aire de combustión, regulando su caudal en función de las desviaciones respecto a la temperatura de consigna. Los reguladores de gas de zona son gobernados mediante la señal variable que procede de los colectores de aire de combustión de los mecheros.

A su paso por esta sección, la banda está protegida con atmósfera de HN contra la oxidación. La oxidación de la banda podría producirse como consecuencia de la entrada de aire por alguna fuga o por rotura de un tubo radiante.

A su paso por esta sección, la banda alcanza su máxima temperatura en todo el proceso (aproximadamente 850ºC), siendo la temperatura máxima de la zona 980ºC.

Una vez alcanzada la temperatura buscada, se mantiene durante un tiempo para permitir el crecimiento del grano.

2.3.2.3 ZONA DE ENFRIAMIENTO LENTO CONTROLADO (F-3)

La banda, que a su paso por la sección anterior alcanzó su máxima temperatura, se enfriará de modo controlado durante el recorrido por F-3 para conseguir unas características mecánicas adecuadas buscando mejorar la textura del metal. El enfriamiento se realiza mediante un intercambio de calor banda-atmósfera. La atmósfera se refrigera por medio de tubos enfriados por aire.

La sección está dividida en dos zonas. La primera tiene 18 tubos de refrigeración distribuidos en igual número en la parte superior y en la inferior del paso de banda. La segunda tiene 18 tubos en la parte superior y el mismo número en la inferior. El aire de refrigeración lo suministra un ventilador para cada zona.

Para el secado del refractario, calentamiento y mantenimiento de temperatura, cada zona tiene instalado en su bóveda un circuito de resistencias eléctricas.

En cada zona hay dos termopares de *chromel-alumel*, uno de seguridad y otro de control. La señal eléctrica del termopar de control se transforma, mediante un instrumento electroneumático, en una señal neumática que acciona la válvula de mariposa, que regula el caudal de aire que circula por los tubos de refrigeración, en función de las desviaciones respecto de la consigna de temperatura. Las resistencias solo entran en funcionamiento cuando, aún con la refrigeración al mínimo, la temperatura permanece excesivamente baja.

En esta sección, una oxidación de la banda sería irreversible por estar el hidrógeno de la atmósfera a una temperatura excesivamente baja. La temperatura de banda a la salida de esta sección, se mide mediante un pirómetro, y varía entre 600-800ºC según el ciclo térmico.

2.3.2.4 ZONA DE ENFRIAMIENTO RÁPIDO "JET COOLING" (C-1)

En esta sección se enfría la banda hasta una temperatura más adecuada para realizar el recubrimiento, esto es ligeramente superior a la del baño. Este enfriamiento rápido sirve para preparar el acero al tratamiento de envejecimiento congelando una cantidad máxima de carbono en sobresaturación.

La sección se divide en tres zonas iguales cuyo funcionamiento es gobernado por la señal neumática emitida por un instrumento electroneumático, que a su vez recibe una señal eléctrica de un pirómetro de radiación. Este pirómetro mide la temperatura de la banda inmediatamente antes de la entrada de la banda al baño.

La circulación de aire en cada zona se realiza por medio de un ventilador que aspira la atmósfera de la zona a través de un intercambiador de calor refrigerado por agua. El caudal de atmósfera, ya fría, se regula mediante una persiana que se posiciona en función de la señal neumática procedente del pirómetro. Este caudal regulado es enviado por el ventilador a dos colectores de superficie perforada, a través de cuyos orificios se proyecta sobre ambas caras de la banda, enfriándola. El ciclo vuelve a empezar a partir de este momento, pero no es realmente un circuito cerrado, ya que la atmósfera del horno es renovada a razón de 300 m^3/h. Es imprescindible mantener esta sección totalmente exenta de fugas debido a que la presencia de oxígeno oxidaría la banda de forma irreversible, impidiendo la adherencia.

La temperatura de la banda, a la entrada del baño, debe ser lo más estable posible, ya que las variaciones de temperatura favorecen la disolución del acero en el zinc.

Por último, se realiza un envejecimiento o Igualación, garantizando una precipitación del carbono, permitiendo minimizar los fenómenos de envejecimientos del acero.

2.3.2.5 ZONA DEL "TURN DOWN"

Aquí se desvía la banda hacia el pote. El conducto de bajada cierra estanco con el nivel del metal fundido en el pote.

El pote de zinc ha sido diseñado para mantener fundido el zinc de revestimiento. La fusión se realiza mediante calentamiento eléctrico. La temperatura de fusión se mantiene mediante el calor que aporta la banda, o mediante resistencias eléctricas, si éste no fuera suficiente.

Un rodillo, denominado "*skin-roll*", se encuentra sumergido en el baño de zinc, a profundidad regulable. Su objeto es cambiar la dirección de la banda en el interior del pote.

Seis conductos, montados uno vertical y cinco horizontales, enfrían la banda mediante corriente de aire forzado. A continuación del conducto vertical, está localizado un rodillo deflector para cambiar la dirección de la banda.

Seguidamente, se encuentra un rodillo deflector que desvía la banda hacia el equipo de tensión nº 3 (brida). A la cabeza de este equipo está localizado un rodillo deflector.

El tanque de enfriamiento enfría la banda mediante la proyección de agua pulverizada sobre ambas caras.

El sistema de control del revestimiento controla el espesor del revestimiento mediante la proyección de aire a alta presión contra ambos lados de la banda.

La unidad de tratamiento químico tiene por objeto recubrir ambos lados de la banda con una leve película de CrO_4H_2, para prevenir la oxidación blanca que se podría producir en condiciones suaves de oxidación, tales como el almacenamiento en la nave o lluvia durante el transporte.

Para mantener constante la velocidad en esta sección se dispone del carro de lazos nº 2.

Cuatro rodillos deflectores conducen la banda por debajo del pote hasta el carro de lazos y el equipo de rodillos guía. En dos de estos rodillos deflectores están localizados los equipos de la galga para medir el espesor del revestimiento.

Cuando el espesor de la banda sea superior a 2mm el aplanado se realizará por medio de la C.S.I. (*Continuous Stretch Leveling*) o aplanadora. Los rodillos de flexión de esta unidad se abrirán al paso de la soldadura por una señal emitida por una célula fotoeléctrica montada a la salida del carro de lazos. Cuatro juegos de rodillos de tensión producen la tensión suficiente para obtener alargamientos de hasta el 2%. A la salida de esta unidad existe un equipo de rodillos para la medida de la tensión de la banda, indicando la uniformidad de la medida la planitud de la banda.

En otras líneas este aplanado puede venir precedido de un *'Skin-pass*", que dote al material de las características mecánicas, y rugosidad superficial adecuadas.

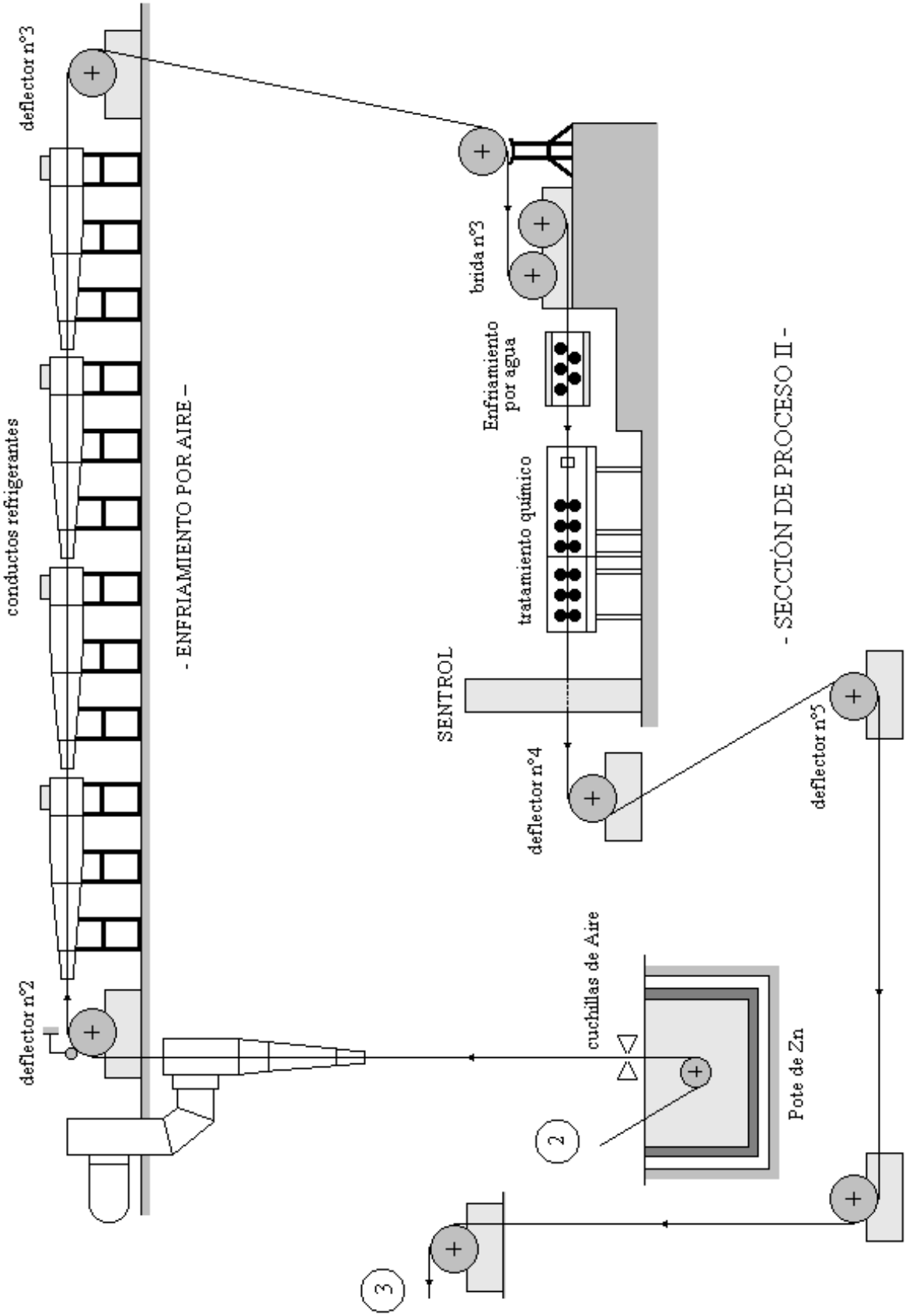


Figura 5. Esquema general de la sección de proceso (fuente: [VER99]).

2.3.3 SECCIÓN DE SALIDA

Se dispone de 2 bobinadoras de mandril expandible dotadas de un control hidráulico automático para el centrado de la banda. Una de ellas puede actuar como debobinadora para alimentar la sección de corte de chapa. Según pedido, las bobinas pueden ser aceitadas en máquinas aceitadoras dispuestas a la entrada de las bobinadoras. Cada bobinadora dispone, para su descarga, de un *skid* con capacidad para el almacenamiento de 2 bobinas de ancho máximo.

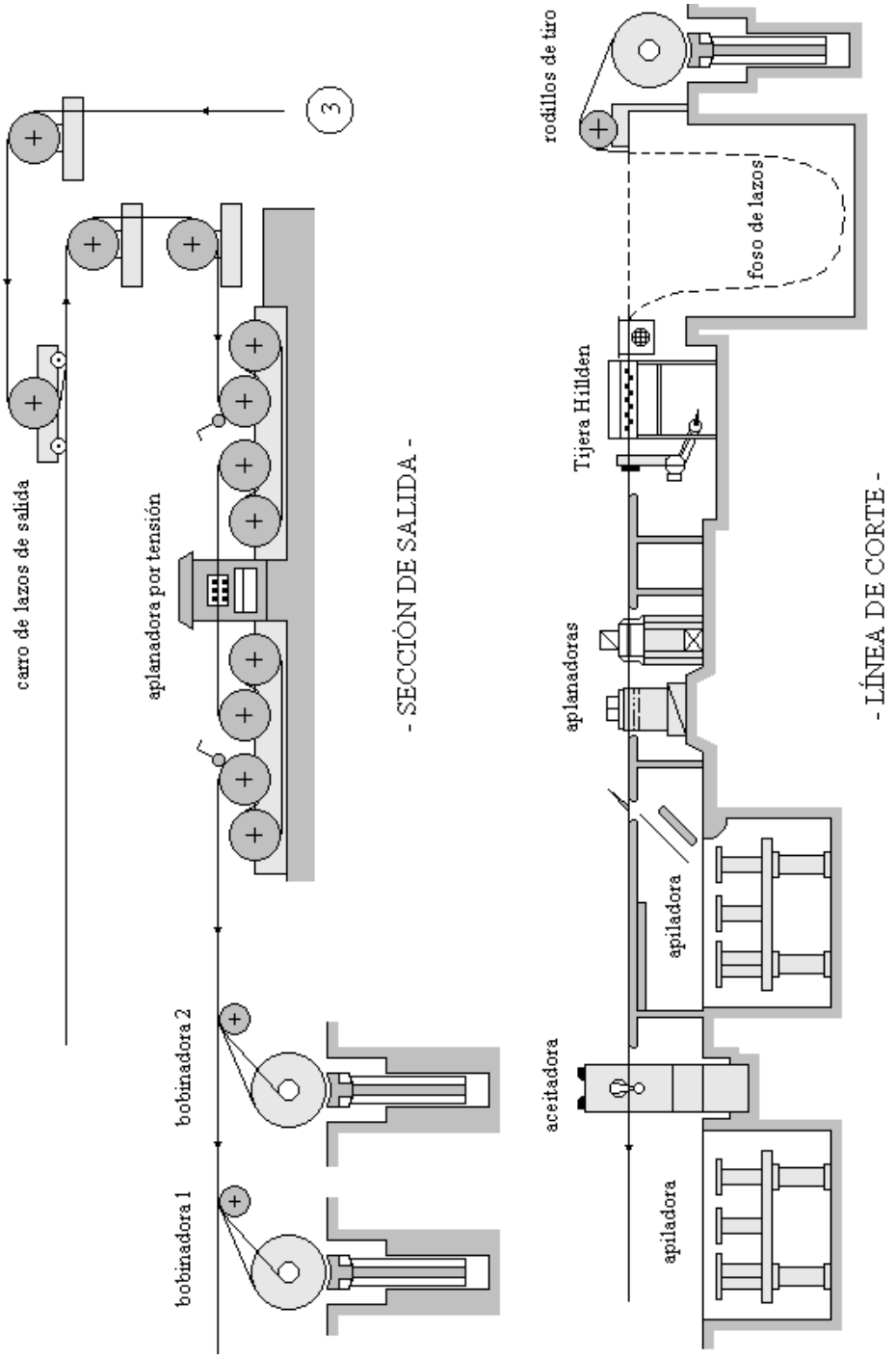


Figura 6. Esquema general de la sección de salida y la línea de corte (fuente: [VER99]).

A continuación, se dispone de una unidad de rodillos de tiro (*pinch roll unit*) y de un foso de lazos a la entrada de la sección de corte para quitar tensión a la banda y favorecer el guiado de la misma a la entrada de la tijera.

Las chapas, de longitud predeterminada, se obtienen mediante una tijera. Cuando el espesor de las chapas está comprendido entre 0.3 mm (0.012") y 1.5 mm (0.060"), el proceso se realizará mediante la aplanadora de 38mm (1½"). Los espesores mayores deberán procesarse con la aplanadora de 63.5mm (2½"). El espesor se verifica mediante una galga de rayos X instalada a la entrada de la tijera volante. Las chapas serán impresas con la marca de fabricación mediante un marcador operado mecánicamente, instalado en la mesa de inspección.

2.3.4 CONTROL DEL RECUBRIMIENTO

El sistema de recubrimiento utilizado en la práctica totalidad de las líneas de galvanizado en continuo es el de "cuchillas de aire". Este sistema no ha sido aun mejorado siendo el vigente actualmente. En este apartado se describirá de forma detallada su modo de actuación.

Las cuchillas de control del revestimiento están diseñadas para producir un chorro de aire con las características de un flujo laminar. Se dispone de una cuchilla a cada lado de la banda localizadas sobre el nivel del baño. El chorro laminar de aire se dirige contra la banda que emerge del metal fundido, impactando perpendicularmente sobre la misma, cortando de este modo la película húmeda del metal del revestimiento; el exceso de zinc escurre hacia abajo, dirigiéndose nuevamente al pote. La película del metal que permanece sobre la banda, presenta excelentes propiedades de aspecto superficial y uniformidad a todo lo ancho de la banda.

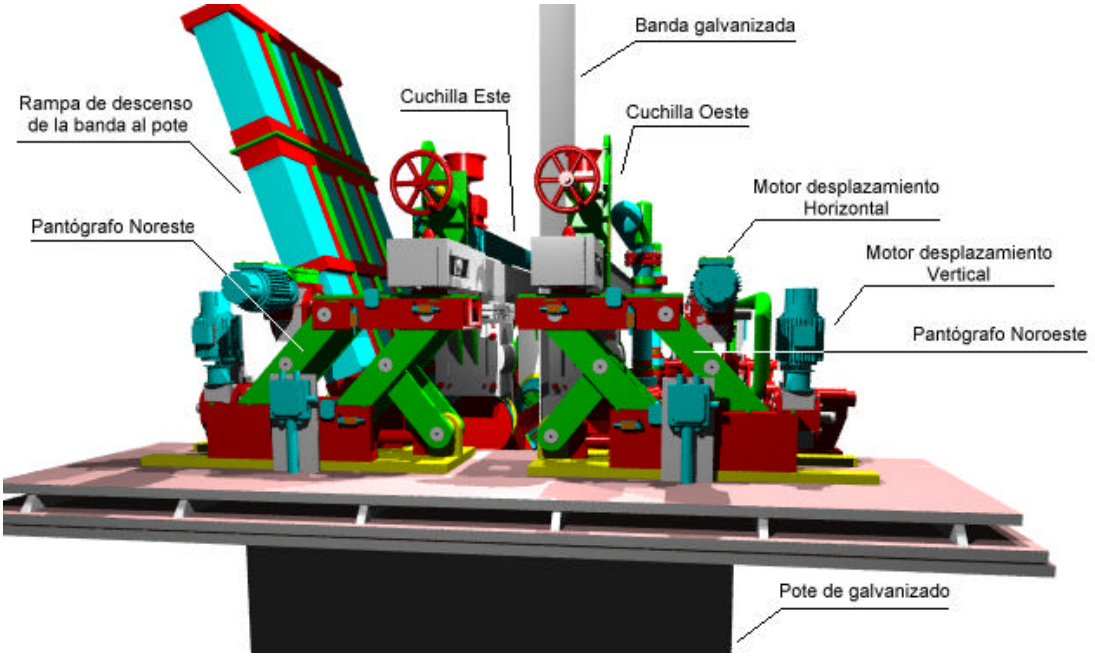


Figura 7. Esquema del sistema de control de recubrimiento mediante cuchillas de aire (fuente: [VER99]).

2.4 MODELO DE CONTROL DEL HORNO

En este punto se describe el modelo matemático de la zona de calentamiento del horno de recocido en continuo de la línea de galvanizado anteriormente descrita [DRE98][GON99][GON94][TOW88], aunque existen otras aproximaciones que usan redes neuronales [LU97] o sistemas expertos [REN99] para mejorar y optimizar el control del mismo.

El control de la zona de calentamiento del horno de recocido es fundamental, ya que tiene como objetivo recristalizar el metal endurecido que sale del laminado en frío y homogeneizar la estructura cristalina del acero. El estudio se centra en esta zona por ser la de mayor dificultad.

El modelo físico del horno se fundamenta en los mecanismos de transmisión de calor siguientes:

1. Conducción.
2. Convección.
3. Radiación.

2.4.1 CONDUCCIÓN

Explica el mecanismo de transferencia de calor, debidos a movimientos moleculares, entre dos cuerpos de íntimo contacto.

Se define como:

$$dQ = -K \cdot A \cdot \frac{dT}{dx} \cdot dt \qquad (2.1)$$

donde:

- dQ: Cantidad de calor (Kcal).
- K: Conductividad térmica del medio de transimisión (Kcal/m·ºC·h)
- A: Área a través de la que pasa Q (m^2)
- dT: Variación de la temperatura (ºC).
- x: Dirección en la que varía la temperatura (m).
- dt: Intervalo de tiempo en que dQ atraviesa A (h).

2.4.2 CONVECCIÓN

Corresponde con el mecanismo de transferencia de calor en el seno de un fluido debido a los movimientos de masa del mismo. Cuando se trata de explicar el fenómeno de convección de un sólido y el fluido que lo rodea, se desarrolla un coeficiente empírico, llamado *coeficiente de película*, que se obtiene del propio proceso de convección y no depende de las propiedades físicas del material.

Se define con la siguiente ecuación:

$$dQ = h \cdot dA \cdot (T_s - T_f) \tag{2.2}$$

donde:

- dQ: Cantidad de calor (Kcal).
- h: Coeficiente de película (Kcal/m^2·ºC).
- dA: Área a través de la que pasa Q (m^2)
- T_S: Temperatura del sólido (ºC).
- T_f: Temperatura del fluido (ºC).

2.4.3 RADIACIÓN

Explica el mecanismo de transmisión de calor que tiene su origen en la emisión de energía interna de un cuerpo en forma de ondas electromagnéticas. Ésta depende de la naturaleza del cuerpo y de su temperatura.

La ecuación básica se define como:

$$Q = s \cdot S \cdot e \cdot (T_f^4 - T_s^4) \tag{2.3}$$

donde:

- Q : Cantidad de calor transferida (Kcal).
- s : Constante de Stefan Boltzmann (4,88x10^{-8} Kcal/m^2·h).
- S: Área de intercambio (m^2).
- e : Coeficiente de emisividad.
- T_f: Temperatura del elemento radiador (K).
- T_S: Temperatura del elemento receptor (K).

2.4.4 ECUACIÓN DEL MODELO FÍSICO

De todos estos mecanismos, es importante destacar que el flujo de calor intercambiado entre dos cuerpos debido al mecanismo de radiación térmica, depende de las temperaturas de cada uno de ellos y no solo de su diferencia, tal y como sucede en los procesos de convección y conducción. Esto indica que, estos dos últimos procesos son predominantes cuando las temperaturas son bajas, mientras que, para temperaturas elevadas, la radiación térmica es el mecanismo de transmisión de calor más importante.

Debido a que la sección de calentamiento del horno se mueve entre los 750ºC y 950ºC, se **considerará solamente el mecanismo de radiación para el cálculo de los modelos** [GON94].

De esta forma, la primera aproximación considera la temperatura de la atmósfera del horno equivalente a la de los tubos radiantes y que la temperatura de las paredes del horno es del orden de la de la atmósfera del horno.

Así se llega a la siguiente ecuación:

$$s \cdot S \cdot RC \cdot (T_f^4 - T_s^4) = r \cdot d \cdot L_s \cdot W \cdot (QS_d - QS_e) \qquad (2.4)$$

donde:

- s : Constante de Stefan Boltzmann ($4{,}88 \times 10^{-8}$ Kcal/m^2·h).
- S: Área de intercambio de calor (m^2).
- RC : Coeficiente de emisividad.
- T_f: Temperatura del horno (K).
- T_s: Temperatura de la banda(K).
- r : Densidad del acero (7.859 Kg/m^3).
- d : Espesor de la banda (m).
- L_s : Velocidad de la banda (m/min).
- W : Ancho de banda (m).
- QS_d : Capacidad de calentamiento de la banda a la salida del horno (Kcal/Kg).
- QS_e : Capacidad de calentamiento de la banda a la entrada del horno (Kcal/Kg).

2.4.5 MODELIZACIÓN MATEMÁTICA DEL CALENTAMIENTO DE LA BANDA

Dentro de las diferentes formas de resolver la ecuación del modelo físico, DREVER [DRE98], propone la ecuación lineal interpolada siguiente:

$$TF = A \cdot TSot + B \cdot (LS \cdot d) + C \cdot \frac{TSot}{RC} + D \cdot \frac{(LS \cdot d)}{RC} + E \qquad (2.5)$$

donde:

- *TF* : Temperatura de la Banda.
- *TSot*: Temperatura de Consigna para cada Bobina.
- *d* : Espesor de la banda (m).
- *LS* : Velocidad de la banda (m/min).
- *RC* : Coeficiente de emisividad.
- *A, B, C, D, E, y RC* : Son coeficientes tomados de una tabla empírica dependiente de la temperatura de consigna, del producto *LS·d* y de los perfiles de calentamiento *(ST1 y ST2)* que corresponden con las diferencias de temperaturas entre zonas de calentamiento.

Una vez obtenidos los coeficientes, se calculan las temperaturas de cada una de las zonas mediante ecuaciones interpoladas, el momento de envío de las nuevas consignas, la velocidad de la banda, etc. [DRE98][GON99][GON94].

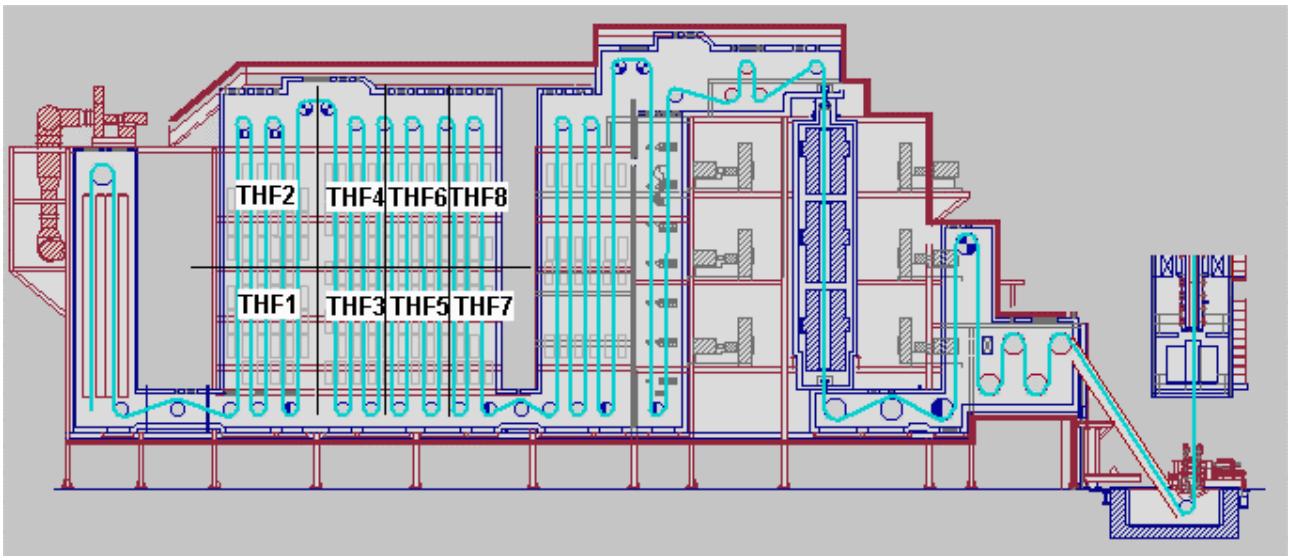


Figura 8. Temperaturas de consigna de cada uno de las zonas de calentamiento del horno.

2.5 CONCLUSIONES

En este capítulo, hemos podido ver cómo las líneas de galvanizado en continuo responden a un esquema común. Las bobinas de acero son convertidas en una banda continua que es limpiada mediante calentamiento no oxidante. Al finalizar la limpieza, a la banda se le dan las características mecánicas deseadas mediante un ciclo de recocido, que consta de un calentamiento y enfriamiento controlado en cuanto a temperaturas y a intervalos de tiempo. El control de estas temperaturas es indispensable para obtener unas bobinas con características adecuadas.

Después del ciclo de recocido, se realiza el galvanizado a través de la inmersión en un pote con zinc, recubriendo la banda por adherencia con una capa de espesor uniforme. El sistema empleado en la práctica totalidad de líneas de este tipo instaladas en el mundo es el control por cuchillas de aire. Sobre la banda inciden dos chorros de aire que eliminan el zinc sobrante devolviéndolo al pote.

Con el fin de evitar defectos superficiales, tras el enfriamiento de la banda, ésta es sometida a un tratamiento químico.

Resumiendo, la calidad del producto se puede focalizar en dos aspectos fundamentales:

- En cuanto a las propiedades del acero, depende fundamentalmente de: **la composición del acero, el proceso de fundición, los procesos de laminación y el ciclo de recocido.**
- En cuanto a las características anticorrosivas, viene marcada por el espesor del recubrimiento de zinc y por la uniformidad del mismo, y depende básicamente de: **la preparación superficial del metal base, el control de la temperatura de recubrimiento y homogeneización de la misma, la composición del baño, el control de las cuchillas de aire y velocidad de la banda.**

De esta forma, tal como se ha comentado anteriormente, **las propiedades finales del acero y el grado de adherencia del espesor de la capa de zinc, van a estar muy influenciadas por el grado de efectividad con que se aplique el ciclo térmico para cada tipo de bobina**. Éste es, por lo tanto, uno de los objetivos de esta tesis: **analizar, controlar y predecir de la mejor manera posible, las señales de consigna para cada tipo de bobina dentro de la zona de calentamiento del horno.**

En capítulos posteriores se describirán los pasos realizados para el análisis y mejora del control del horno mediante técnicas de Minería de Datos y de Inteligencia Artificial.

CAPÍTULO 3

ESTADO DEL ARTE: EL "DATA MINING"

3.1 INTRODUCCIÓN

Cada vez más, y gracias al desarrollo de las comunicaciones y la implementación y mejora de las redes informáticas; la cantidad de información que fluye en las empresas[2] y la capacidad de acceso a la misma ha aumentado considerablemente, pero en cambio, **cada vez se tiene menos tiempo y capacidad para asimilarla** [PER01]. Muchas veces, las empresas no saben obtener información valiosa de toda la cantidad ingente de datos que tienen almacenados[3], aunque **intuyen que el conocimiento que podrían extraer de ellos podría ser de gran ayuda en muchas de las áreas y facetas en que se desenvuelven** (toma de decisiones, mejoras en la producción, mejor conocimiento de los gustos del cliente, etc.). La competitividad de las empresas, y por lo tanto su supervivencia, depende de que este conocimiento pueda preservarse y utilizarse de forma eficiente.

En la industria, igualmente, la preocupación de las empresas por producir "mejor y más barato", la búsqueda constante de reducir "incertidumbre" en el proceso de fabricación y el aumento creciente de la información que se tiene de los procesos productivos, hace que crezca cada vez más, el interés por analizarla [CAS01]. Bien es cierto, que este interés solo aparece cuando la empresa tiene un volumen de históricos realmente importante del proceso y una cultura[4] de mejora arraigada.

Por otro lado, lógicamente, el tener un aceptable grado de automatización y *Data Warehouse* es requisito indispensable, ya que si no se dispone de la infraestructura necesaria para capturar y almacenar convenientemente la información, difícilmente se podrá obtener nada de ella [ORD00]. Esto implica que las empresas, antes de poder mejorar el proceso de producción con la minería de datos, deben invertir en mejorar los sistemas de automatización y control del mismo de forma que puedan construir una base de datos con históricos del proceso coherente, exhaustiva y de buena calidad.

---

[2] Se ha estimado que cada 20 meses se duplica la cantidad de información en el mundo [BOR96].
[3] Se estima que solo entre el 5% y 10% de las bases de datos comerciales han sido analizadas [FAY96a].
[4] Muchas veces la necesidad de mejora surge cuando se plantean soluciones para llevar a cabo eficazmente la trazabilidad del producto.

Las herramientas de *data mining* y estadística multivariante son útiles en este momento, cuando ya tenemos un volumen de información importante y de buena calidad. Los campos de aplicación de estas nuevas técnicas dentro de la industria son numerosos: control de calidad, identificación de sistemas, determinación de causas en fallos del proceso, detección de anomalías, prevención de fallos, modelización de sistemas, obtención de reglas y patrones de comportamiento, búsqueda de causas y relaciones entre variables, etc.

3.2 ¿QUÉ ES DATA MINING?

Como se comentaba anteriormente, para que las empresas trabajen eficientemente y mejoren su competitividad, se necesita contar con el mínimo de información necesaria y presentarla de la forma más fácil de interpretar y manejar. La Gestión del Conocimiento (*Knowledge Discovery)* abarca todas aquellas tecnologías relativamente nuevas que surgen de esta necesidad de procesar, analizar y aprovechar esta información escondida en grandes volúmenes de datos. Esta capacidad de captación, estructuración y transmisión de conocimiento es lo que se requiere de la *Gestión del Conocimiento.* Resumiendo, la Gestión del Conocimiento es una ciencia relativamente nueva que desarrolla herramientas que permiten tratar datos, **garantizando a quién las use, obtener información útil de la forma lo más eficiente posible a partir de los mismos.**

Vemos por lo tanto, que dentro de las múltiples áreas que se agrupan en torno a la gestión del conocimiento, aparece la *Minería de Datos* o *Data Mining* como una de las disciplinas que más está influyendo en nuestros días dentro del ámbito del análisis de datos. A grandes rasgos se puede decir que **la *minería de datos* es un conjunto de metodologías y herramientas que permiten extraer el conocimiento útil (patrones de comportamiento, modos de operación, información útil para descubrir fallos, tendencias, etc.) para la ayuda en la toma de decisiones, comprensión y mejora de procesos o sistemas, etc.; partiendo de grandes cantidades de datos.** Para alcanzar buenos resultados es necesario comprender que la minería de datos no se basa en una metodología estándar y genérica que resuelve todo tipo de problemas, sino que consiste en una metodología dinámica e iterativa que va a depender del problema planteado, de la disponibilidad de la fuentes de datos, del conocimiento de las herramientas necesarias y de los requerimientos y recursos de la empresa.

Claramente se identifican numerosos campos de aplicación de estas nuevas técnicas: control, optimización y supervisión de procesos industriales, control de calidad, modelado e identificación de sistemas, obtención de tendencias de la bolsa de valores, obtención del grado de siniestralidad posible que puede tener un asegurado, correlación entre indicadores financieros, diagnóstico de enfermedades, determinación de los efectos de un medicamento, clasificación de señales biomédicas, predicción de ventas, planificación de campañas publicitarias, detección de fraudes, detección de evasión de impuestos, detección de redes de narcotráfico, hallazgos de patrones de comportamientos criminales, etc.

Pero, vamos a tratar con más profundidad cada uno de estos conceptos...

3.2.1 DEFINICIÓN

El nombre de *Data Mining*, o Minería de Datos, deriva de la similitud que se encuentra entre buscar valiosa información de negocios en grandes bases de datos con la búsqueda de vetas de metales preciosos dentro de una montaña. Ambos procesos requieren examinar inteligentemente una inmensidad de material hasta encontrar algo que pueda resultarnos útil y valiosa.

La definición del concepto de *Data Mining* (DM) puede variar entre unos investigadores y otros. Por ejemplo, los estadísticos, analistas de datos y la comunidad de sistemas de gestión de la información adoptan mayoritariamente este término para referirse **al proceso genérico correspondiente a las técnicas y herramientas de investigación usadas para extraer información útil de una base de datos.** Dentro de estas técnicas podemos considerar todos aquellos métodos matemáticos y técnicas software para el análisis inteligente de los datos y búsqueda de patrones o tendencias en los mismos aplicados de forma iterativa e interactiva.

Dentro de las definiciones que se pueden encontrar en la literatura relacionada se muestran algunas de las más significativas:

- "*Data Mining* es la exploración y análisis, mediante métodos automáticos o semiautomáticos, de grandes cantidades de datos para descubrir reglas o patrones significativos" [BER97].
- "*Data Mining* es el proceso analítico diseñado para explorar grandes cantidades de datos (típicamente relacionados con el mercado o los negocios) con el fin de investigar patrones consistentes y/o relaciones sistemáticas entre variables y, a continuación, validar los resultados aplicando modelos detectados para nuevos subgrupos de datos" [STA01].
- "*Data Mining* es el conjunto de técnicas y herramientas aplicadas al proceso trivial de extraer y presentar el conocimiento implícito, previamente desconocido, potencialmente útil y humanamente comprensible, a partir de grandes conjuntos de datos, con el objeto de predecir de forma automatizada tendencias y comportamientos y/o descubrir de forma automatizada modelos previamente desconocidos" [PIA91]
- "*Data Mining* es el descubrimiento eficiente de información valiosa, no obvia, de una gran colección de datos" [BIG96].
- "*Data Mining* es la extracción de información implícita, previamente desconocida y potencialmente útil de una base de datos" [WIT00].
- "*Data Mining* combina técnicas de la estadística, inteligencia artificial, Bases de Datos, Visualización y otras áreas, para descubrir, de forma automática o semiautomática, modelos de (algunas veces enormes) series de datos." [SIE00].

- "*Data Mining* es el proceso de plantear varias preguntas y extraer información útil, patrones y tendencias de grandes cantidades de datos generalmente almacenados en bases de datos." [THU99].
- "*Data Mining* es el análisis de, habitualmente grandes, series de datos (observaciones) para encontrar relaciones inesperadas y resumir la información de nuevas maneras que sean entendibles y útiles por el propietario de los datos." [HAN01].

En cambio, en el ámbito del *Knowledge Discovery in Databases* (KDD) o descubrimiento de conocimiento en bases de datos, el *Data Mining* tiene otro significado. Efectivamente, el término KDD se empezó a utilizar en 1989 [PIA91] popularizándose por lo expertos en inteligencia artificial (IA) y aprendizaje de ordenadores (*Machine Learning)* **para referirse al amplio proceso de búsqueda de conocimiento en bases de datos y para enfatizar de que este "conocimiento" es el producto final del proceso del KDD.** La definición de KDD más representativa surge de diversos autores especialistas en ese campo:

- "Descubrimiento en bases de datos (KDD): es el proceso no trivial de identificar patrones en datos que sean válidos, novedosos, potencialmente útiles y por último comprensibles" [FAY96] [FAY96b].

Las fases en que se divide el KDD según [DIS02] [PER01a] son: exploración del dominio, recolección de los datos, **extracción de patrones en los datos**, inducir generalizaciones, verificación del conocimiento, transformación del conocimiento.

- De esta forma, y según los autores provenientes del campo de la IA o del Machine Learning (ML), el *Data Mining* corresponde a un paso del KDD y se define en la literatura especialista de las siguientes formas:
- "*Data Mining* consiste en obtener modelos comprensibles o patrones de una base de datos" [SIE00].
- "*Data Mining:* búsqueda de patrones de interés mediante árboles o reglas de clasificación, técnicas de regresión, clusterizado, modelizado secuencial, dependencias, etc." [WAN99].
- "El proceso de extraer patrones o modelos a partir de los datos" [FAY96].

En esta última definición coinciden la mayor parte de los autores que se dedican al *Data Mining*, el KDD, la IA o el ML, aunque también otros autores, como se ve en definiciones anteriores, describen el DM como el proceso completo.

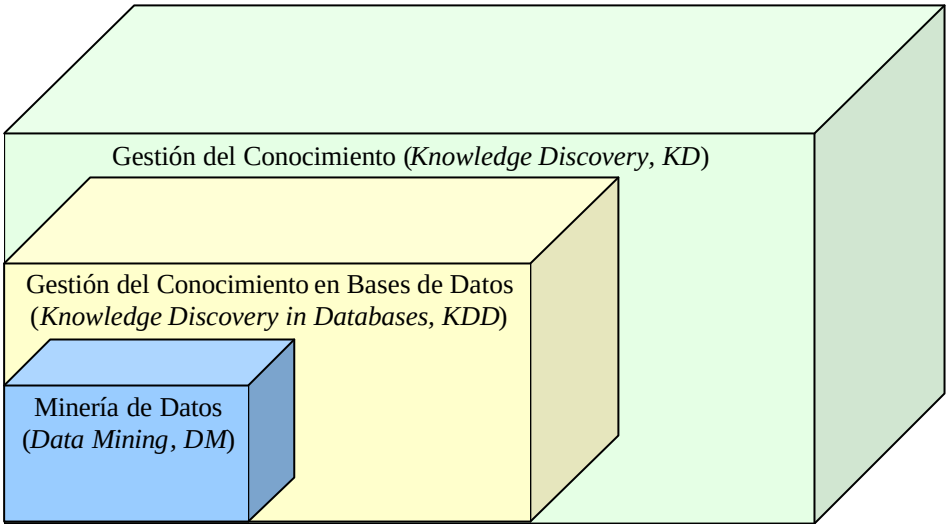


Figura 9. La Minería de Datos frente al KD y KDD.

En la Figura 9 resume cómo el *DM* está incluido en el *KDD*, y cómo éste se incluye a su vez en el *KD*.

3.2.2 CRONOLOGÍA DEL DM

Aunque los componentes clave del *data mining* existen desde hace décadas en áreas de investigación como inteligencia artificial, estadística o el aprendizaje automático, se puede afirmar que es ahora cuando se asiste al reconocimiento de la madurez de estas técnicas.

Las raíces del DM se remontan a los años 50. En esa época, los departamentos de informática preparaban resúmenes de la información, principalmente de tipo comercial, que se encontraba en los ficheros del ordenador central, con el propósito de facilitar la labor directiva. Así nacieron los sistemas de información para la dirección (EIS), que, sin embargo, eran voluminosos, poco flexibles, y difíciles de leer para los no informáticos.

En los años 60 nacen los sistemas gestores de base de datos que aún se mostraban rígidos y carecían de flexibilidad para realizar consultas. Posteriormente aparecieron los motores relacionales resolviendo estos problemas, aunque los informes resultaban muy laboriosos de preparar y depurar, perdiéndose relevancia por su bajo nivel de actualización. Otro grave problema era la diversidad de bases de datos no integradas, establecidas por los diferentes departamentos de una organización.

Nadie reparaba en la posible utilidad futura de un sistema interdependiente. El *Data Warehouse* vino solucionar este problema a finales de los años 80. El concepto de *Data Mart* permite responder a necesidades específicas sin renunciar al futuro almacén de datos global de la organización. La existencia del DW ha estimulado el desarrollo de los enfoques del DM, en los que las tareas de análisis se automatizan y dan un paso más al posibilitar la extracción de conocimiento inductivo.

El término "*descubrimiento de conocimiento en bases de datos*" (Knowledge Discovered in Databases o KDD para abreviar) empezó a usarse entre los expertos de inteligencia artificial y

aprendizaje de ordenadores, para referirse al amplio proceso de búsqueda de conocimiento en bases de datos y para enfatizar el hecho de que "el conocimiento" es el producto del incremento del ritmo de adquisición de datos. El crecimiento de la cantidad de datos almacenados se ve favorecido no sólo por el abaratamiento de los discos y sistemas de almacenamiento masivo, sino también por la automatización de muchos trabajos y técnicas de recogida de datos.

De esta forma, surge el término *Data Mining* a finales de la década de los 80, de las similitudes que existen entre buscar valiosa información de negocio en grandes bases de datos y minar una montaña, para encontrar una veta de metales preciosos.

Fundamentalmente, el avance de la Minería de Datos en las empresas se debe fundamentalmente a estos aspectos:

- Uso de la información para la búsqueda de la mejora de la competitividad en aspectos como: mejora de la calidad, reducción de costes, control de la producción, optimización de los recursos, etc.
- Incremento de la potencia de los ordenadores y abaratamiento de los mismos.
- Incremento del ritmo de adquisición de datos. El crecimiento de la cantidad de datos almacenados se ve favorecido no sólo por el abaratamiento de los discos y sistemas de almacenamiento masivo, sino también por la automatización de muchos trabajos y técnicas de recogida de datos.
- Aparición de nuevos métodos de técnicas de aprendizaje y almacenamiento de datos.

Etapa	Preguntas	Tecnologías	Características
Recolección de datos (Años 60)	**¿Cuál fue mi volumen total de ventas en los últimos 5 años?**	**Ordenadores. cintas, discos**	**Retrospectivo, datos estáticos**
Acceso a los datos (Años 80)	**¿Cuáles fueron las ventas unitarias en las Región Central el pasado mes de Marzo?**	**Bases de datos relacionales (RDBMS), Lenguaje *Query* Estructurado SQL, ODBC.**	**Retrospectivo, datos dinámicos**
Data Warehouse y soporte a la toma de decisiones (Años 90)	**¿Cuáles fueron las ventas unitarias en la Región Central el pasado mes de Marzo? Procedimiento para obtener datos de la zona de Caracas.**	**OLAP, bases de datos multidimensionales, datawarehouses.**	**Retrospectivo, obtención dinámica de datos a múltiples niveles**
Data Mining (Minería de Datos). Actualmente en desarrollo.	**¿Qué podrá pasar con las ventas unitarias en Caracas el próximo mes? ¿Por qué?**	**Algoritmos avanzados, ordenadores multiprocesador, bases de datos masivas.**	**Entrega de la Información proactiva y predictivamente.**

Tabla 7. Evolución de la Minería de Datos

3.2.3 ARQUITECTURA DE APLICACIÓN

Como se ha comentado en párrafos anteriores, para poder llevar a cabo una búsqueda de conocimiento mediante las técnicas de *Data Mining*, se necesita de una base de datos coherente y con datos relevantes del proceso. Esto implica, una necesidad clara de disponer de un sistema de adquisición, almacenamiento y manejo de la información lo suficientemente eficiente. Aquí es donde entran el campo del *Data Warehouse*, los sistemas OLAP y otros sistemas que se definen a continuación.

3.2.3.1 EL DATA WAREHOUSING

Se define *Data Warehousing* como el proceso a través del cual se organiza una gran cantidad de datos variados y almacenados, de tal forma que facilite la recuperación de información para llevar a cabo el proceso analítico [KIM98].

Los almacenes de datos (o *Data Warehouses*) generan bases de datos tangibles con una perspectiva histórica, utilizando datos de múltiples fuentes que se fusionan en forma congruente. Estos datos se mantienen actualizados, pero no cambian al ritmo de los sistemas transaccionales.

Muchos *Data Warehouses* se diseñan para contener un nivel de detalle hasta el nivel de transacción, con la intención de hacer disponible todo tipo de datos y características, para informar y analizar. Así un *Data Warehouse* se puede identificar como un recipiente de datos transaccionales que proporciona consultas operativas e información para poder llevar a cabo análisis multidimensionales. **De esta forma. dentro de una almacén de datos existen dos tecnologías complementarias: una relacional para consultas y una multidimensional para análisis**.

3.2.3.2 SISTEMAS OLAP

El *Data Warehousing* puede permitir la explotación sobre un procesamiento denominado OLAP (*On Line Analytical Processing*). **Los sistemas OLAP son aplicaciones de bases de datos orientadas a realizar un análisis multidimensional de datos mediante navegación del usuario por los mismos de modo asistido**. La información es vista como cubos, que contienen categorías descriptivas (dimensiones) y valores cuantitativos (medidas). El usuario puede acceder a su información bajo diferentes niveles de abstracción: desde el detalle más bajo hasta agregaciones bajo diferentes dimensiones.

Por lo tanto, un sistema OLAP se puede entender como **la generalización de un generador de informes.** Los sistemas OLAP evitan la necesidad de desarrollar interfaces de consulta, y ofrecen un entorno único valido para el análisis de cualquier información histórica, orientado a la toma de decisiones. A cambio, es necesario definir dimensiones, jerarquías y variables, organizando de esta forma los datos.

Según [BER97][COD93] se denominan OLAP a aquellos sistemas que:

- Soportan requerimientos complejos de análisis.
- Analizan datos desde diferentes perspectivas.
- Soportan análisis complejos multidimensionales dentro de volúmenes ingentes de información.
- Permiten la posibilidad de navegar sobre la información mediante informes jerarquizados.
- Disponen de funciones como: proyectar a una tabla, expandir, colapsar, girar, rotar, etc.

Se dividen fundamentalmente en dos tipos:

- **Sistemas MOLAP (Multidimensional OLAP):** Se utiliza sobre bases de datos multidimensionales Para los desarrolladores de aplicaciones acostumbrados a trabajar con bases de datos relacionales, el diseño de una base de datos multidimensional puede ser complejo o al menos, extraño. Pero en general, el diseño de dimensiones y variables es mucho más sencillo e intuitivo que un diseño relacional. Esto es debido a que las dimensiones y variables son reflejo directo de los informes en papel utilizados por la organización. En un sistema MOLAP los datos se encuentran almacenados en archivos con estructura multidimensional, los cuales reservan espacio para todas las combinaciones de todos los posibles valores de todas las dimensiones de cada una de las variables, incluyendo los valores de dimensión que representan acumulados. Es decir, un sistema MOLAP contiene precalculados (almacenados) los resultados de todas las posibles consultas a la base de datos. MOLAP consigue consultas muy rápidas a costa de mayores necesidades de almacenamiento, y retardos en las modificaciones (que no deberían producirse salvo excepciones), y largos procesos *batch* de carga y cálculo de acumulados.

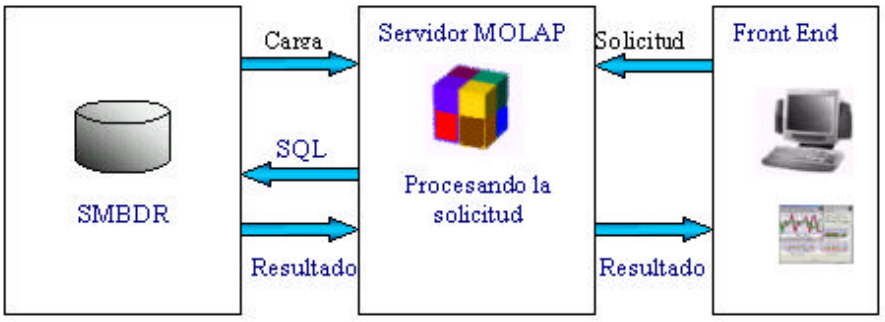


Figura 10. Arquitectura MOLAP[5] (fuente: [CUE02]).

[5] SMBDR significa Sistema de Manejo de Base de Datos Relacional.

- **Sistemas ROLAP (Relacional OLAP):** Corresponde a una arquitectura de base de datos multidimensional en la que los datos se encuentran almacenados en una base de datos relacional, la cual tiene forma de estrella (también llamada copo de nieve o araña). En ROLAP, en principio la base de datos sólo almacena información relativa a los datos en detalle, evitando acumulados (evitando redundancia). En ROLAP, al contener sólo las combinaciones de valores de dimensión que representan detalle, es decir, al no haber redundancia, el archivo de base de datos es pequeño. Los procesos *batch* de carga son rápidos (ya que no se requiere agregación), y sin embargo, las consultas pueden ser muy lentas, por lo que se aplica la solución de tener al menos algunas consultas precalculadas.

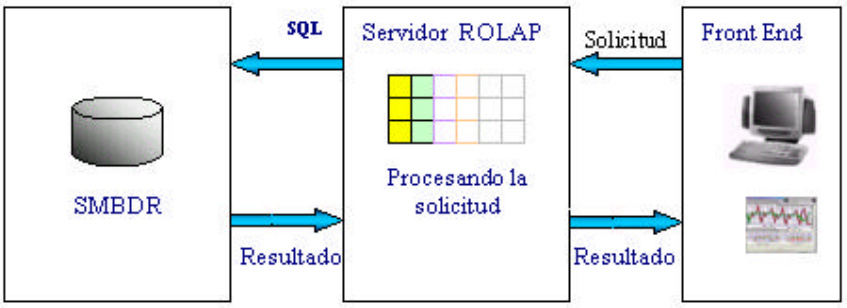


Figura 11. Arquitectura ROLAP (fuente: [CUE02]).

Resumiendo, podemos concluir que **los sistemas MOLAP son más adecuados para sistemas con pequeñas cantidades de datos que requieren de gran velocidad, mientras que los ROLAP son más adecuados para aquellas bases de datos grandes con requerimientos de velocidad menores**.

3.2.3.3 DIFERENCIAS ENTRE OLAP Y DSS

De todas formas es conveniente diferenciar entre un sistema OLAP y un sistema orientado a la toma de decisiones (DSS[6]) donde entraría el proceso de minería de datos, pues se trata de niveles jerárquicos distintos.

Con relación a los DSS, la minería de datos es el proceso en el que se apoyan las decisiones que buscan, a través de patrones de comportamientos, patrones de información en los datos a partir de los que se podrán determinar tendencias, mientras que un sistema OLAP nos ayuda a localizar rápidamente la información.

Por ejemplo, mientras que un sistema OLAP nos ayudaría a responder preguntas del tipo [ABA01]: "¿Compraron más vehículos del modelo X los habitantes del norte de España o del Sur en el año 1998?", un sistema DSS apoyado en técnicas de minería de datos, nos ayudaría a responder preguntas del tipo: "Quiero un modelo que identifique las características predictivas más importantes de las personas que compraron vehículos de la marca X".

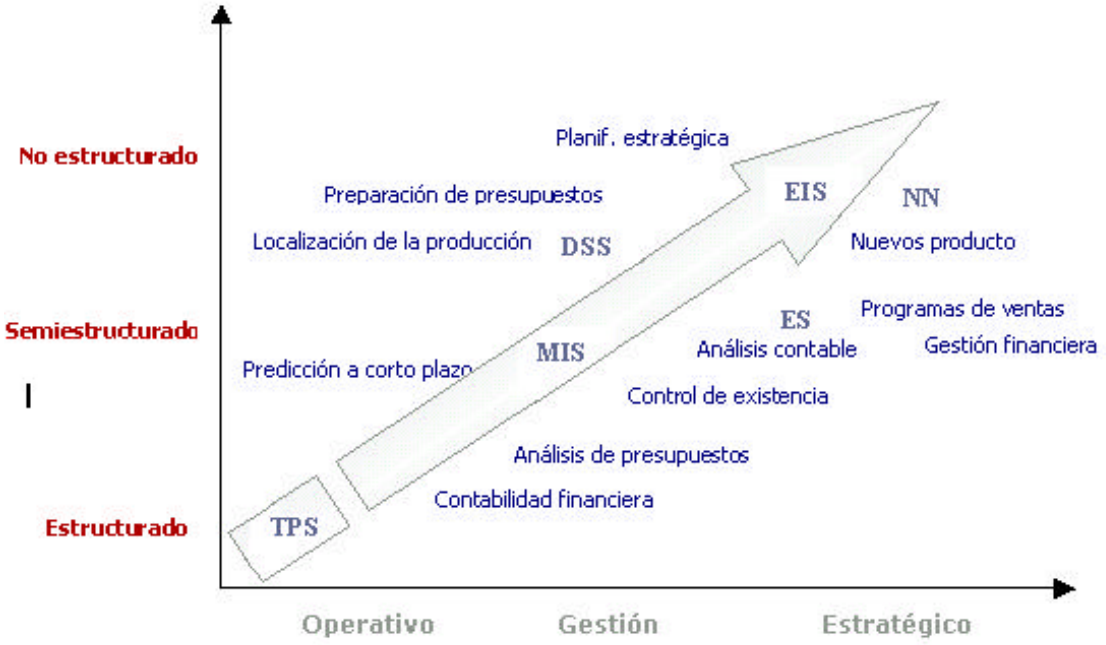


Figura 12. Herramientas a utilizar según el tipo de decisión y el nivel organizativo. ((TPS) Sistema de Proceso de Transacciones, (MIS) Sistema de Gestión de Información, (EIS) Sistema Informático para Ejecutivos, (RN) Red Neuronal(ES) Sistema Experto (DSS) Sistema de Soporte a la Toma de Decisiones) [CUE02].

[6] Decisión Support System (sistema de soporte a la decisión).

3.2.3.4 OTROS SISTEMAS

Realmente, los sistemas arriba descritos, se han estado aplicando en estos últimos años. Como es lógico, muchos de los sistemas que han existido desde hace tiempo han realizado fundamentalmente tareas de extracción de información más que de extracción de conocimiento. Así no encontramos con conceptos como por ejemplo:

- Sistemas de información gerencial MIS (*Management Information System*): Que corresponden con los sistemas generadores de informes de apoyo a la gerencia, y que se están viendo desplazados por los sistemas DSS.
- Sistemas de información para ejecutivos EIS (*Executive Information System*): Que proporcionan una clara visión a todos los niveles de lo que está ocurriendo en la empresa, aunque no están orientadas a la toma de decisiones.
- Cuadros de Mando : Que monitorizan en tiempo real los parámetros más importantes del sistema.

También encontramos aplicaciones más específicas, como por ejemplo:

- Query & Reporting: Que corresponden a sistemas con alta complejidad de consultas pero con altísimos tiempos de respuesta. Generalmente interactúan con otros sistemas del entorno.

3.2.3.5 OTRAS DEFINICIONES Y CONCEPTOS ACTUALES

Últimamente, están surgiendo otros conceptos que giran entorno a los sistemas que dan soporte a la toma de decisiones. Se definen brevemente:

- Sistemas DGSS (*Groupware Decission Support Systems*): Que corresponden a sistemas de apoyo a la toma de decisión en grupos, y que tratan de potenciar la toma de decisiones en un grupo de personas mediante la realimentación de la información generada por cada uno de los miembros del mismo.
- Sistemas Expertos ES (*Expert Systems*): Que son diseñados a partir de la experiencia de una serie de expertos y permiten tomar decisiones tal y como lo haría un humano. Ahora bien, para diseñar un sistema experto se necesita de un ingeniero de conocimiento (*knowledge engineer*), alguien que estudia cómo los expertos humanos toman decisiones y traduce las reglas en términos que un equipo de computación puede entender.

3.2.4 FASES DE UN PROCESO CLÁSICO DE DATA MINING

Las fases en el proceso global de *data mining* no están claramente diferenciadas lo que hace que sea un proceso iterativo e interactivo con el usuario experto. Las interacciones entre las decisiones tomadas en diferentes fases, así como los parámetros de los métodos utilizados y la forma de representar el problema suelen ser extremadamente complejos. Pequeños cambios en una parte pueden afectar fuertemente al resultado final. En [Bra961] se estructura el proceso en seis fases (tal como se muestra en la figura) que serán desarrolladas con detalle durante este capítulo.

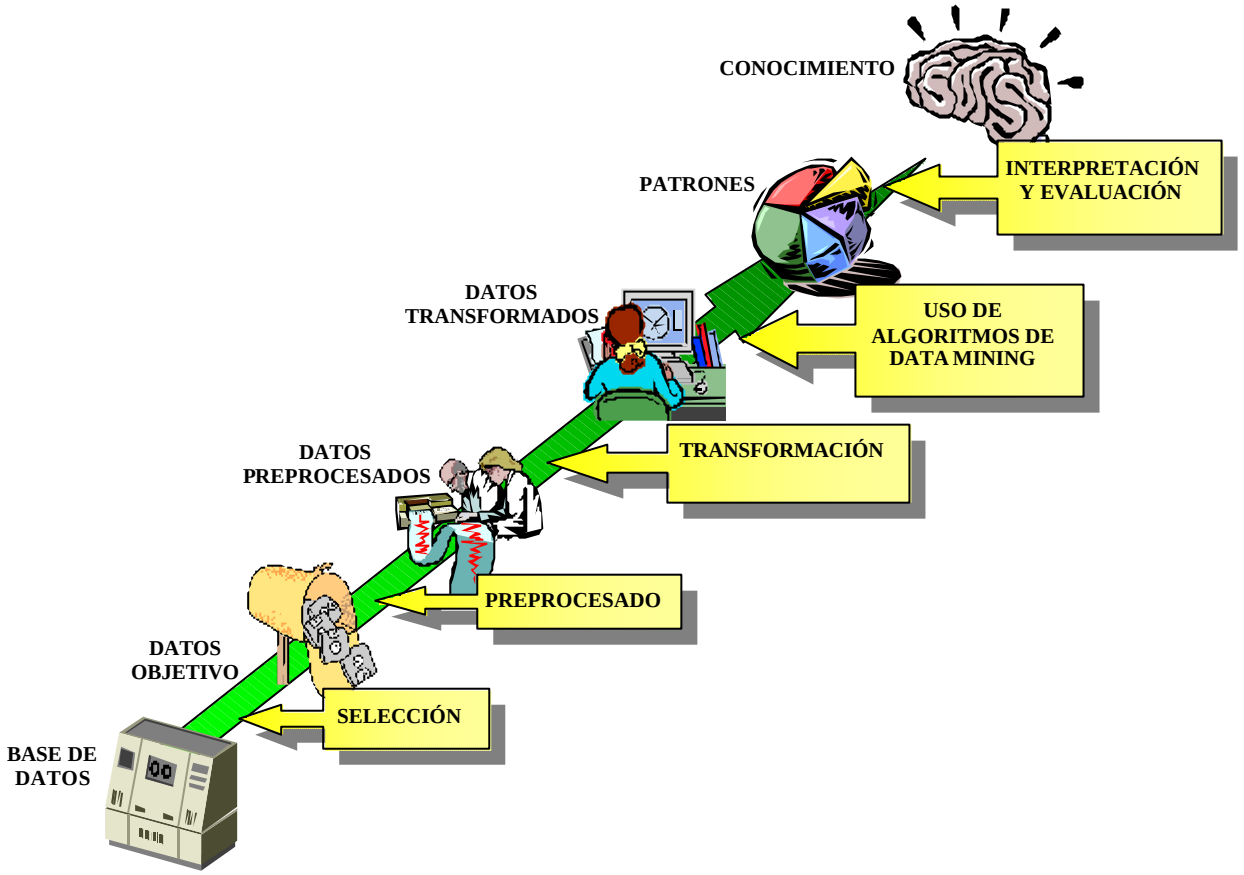


Figura 13. Fases típicas de un proceso de Data Mining.

3.2.4.1 DEFINICIÓN DEL ALCANCE Y OBJETIVOS

El primer paso de un proyecto de *data mining* consiste en **conocer el desarrollo y dominio de la aplicación, determinar el conocimiento relevante a usar, así como establecer los objetivos del usuario final**. El desarrollo de esta primera fase, establece las bases para la realización de las posteriores fases del proceso. por lo que el éxito o fracaso del proceso va a depender en gran medida de las decisiones que se adopten en esta etapa.

En esta fase se determinan los factores que son susceptibles de un procesado automático, los cuellos de botella del dominio, los conocimientos a priori que se tienen del proceso, así como cuáles son los objetivos finales que se pretenden lograr y cuáles van a ser los criterios de rendimiento

exigibles. Por lo tanto, esta fase requiere cierta dependencia usuario-analista. siendo necesario el establecimiento de unos canales de comunicación entre ambas partes.

Otro factor clave corresponde al conocimiento que se tiene del sistema. Muchas veces, lagunas de entendimiento dentro del proceso a analizar, pueden implicar pérdidas significativas de tiempo en fases posteriores. Generalmente, es conveniente volcar todos los esfuerzos iniciales en comprender el sistema en todos sus detalles, para evitar llegar a callejones sin salida debido a una falta de comprensión de algunas de las partes del proceso.

La importancia de este último factor queda reflejado en una de las metodologías de implantación de minería de datos más usada actualmente, el método CRISP-DM (Cross-Industry Standard Process for Data Mining) que se estudiará con más detalle en apartados posteriores.

En [PYL99], Dorian Pyle considera que un 80% de la importancia para llegar al éxito proviene en la forma de abordar el problema, definir cuales pueden ser las pautas para llegar a la solución y la forma de implementarlas para solucionar el problema con éxito.

Tarea	Porcentaje del Tiempo dedicado (en %)	Importancia para llegar al éxito final (en %)
1. Definir el Problema	**10 %**	**15 %**
2. Explorar la Solución	**9 %**	**14 %**
3. Implementación de los resultados	**1 %**	**51 %**
4.1. Data Mining: Preparación de los datos.	**60 %**	**15 %**
4.2. Data Mining: Procesamiento de los datos	**15 %**	**3 %**
4.3. Data Mining: Modelizado y testeo de los datos.	**5 %**	**2 %**

Tabla 8. Tiempo e importancia de cada una de las fases según [PYL99].

Fundamentalmente, según Pyle, los pasos que llevan al éxito van a depender de los siguientes aspectos:

- **Identificación correcta de los problemas a resolver:** Muchas veces, esta tarea puede parecer trivial, pero puede suceder que el problema no se comprenda completamente. En [PYL99] se cuenta el caso de una empresa de telecomunicaciones que pretendía mejorar un modelo de predicción de los clientes que tenían un alto porcentaje de probabilidades en darse de baja. Después de pasar por alto esta primera fase, ya que consideraban que tenían muy bien definido el problema, se desarrolló un modelo predictivo que parecía tener una eficiencia del 80% frente al 50% del modelo anterior y se planteó una campaña de marketing, con un elevado gasto de dinero, dirigida hacia esos clientes que, según el modelo, podían darse de baja en poco tiempo. El resultado final de esa campaña de marketing fue desastroso, ya que, por ejemplo, se habían considerado dentro del modelo que, personas desempleadas mayores de 80 años tenían muchas posibilidades en anular su contrato con la empresa. Y, claro está, muchas de estas personas morían y por lo tanto, ningún programa de incentivos ni de buzoneo podía evitar que estas personas se dieran de baja de la compañía de telecomunicaciones. Esto, condujo a pensar a la

empresa, que el problema había estado mal definido desde el principio y que había que considerarlo según diferentes segmentos de mercado.

- **Definición con precisión de los problemas**. Será necesario dividir las descripciones del problema que son demasiado generalistas en componentes más pequeños que puedan ser contrastados por la información examinada. Por ejemplo, un problema del tipo: "Me gustaría tener un modelo matemático de los índices de fallos que se producen en mi línea de ensamblaje los lunes y los viernes para poder eliminarlos o reducirlos...", puede ser necesario recomponerlo en pequeños y más abordables subproblemas del tipo: "detectar los tipos de fallos que se producen", "por qué los lunes y viernes sobresalen de los demás días", "determinar qué hay que examinar (empleados, maquinaria, etc.)", "cuáles son las piezas donde se producen los fallos", etc.

- **Uso de Mapas Cognitivos**. Cuando la cantidad de información es relevante puede ser necesario estructuras la información mediante un *mapa cognitivo* [PYL99]. En éste se representan lo objetos en círculos y con líneas las interrelaciones causa-efecto positivas o negativas que se producen entre ellos. La Figura 14 es un ejemplo sencillo de las relaciones causa-efecto, positivas o negativas que existen entre varios elementos interrelacionados.

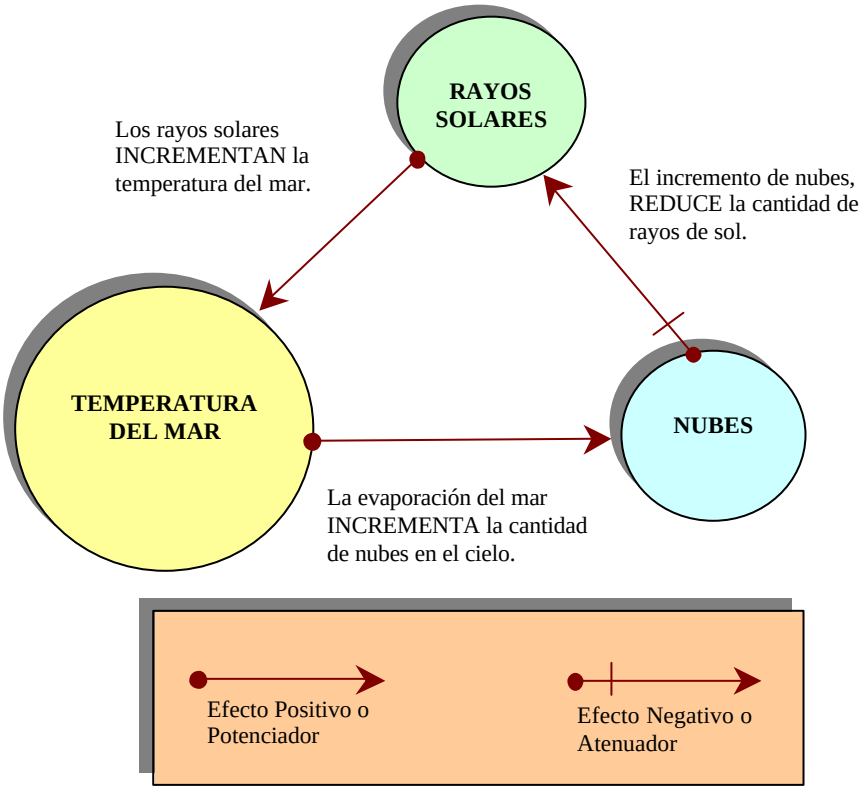


Figura 14. Mapa cognitivo simple de la relación entre las nubes, la temperatura del mar y los rayos solares.

- **Resolver las ambigüedades.** Es conveniente resolver las ambigüedades que puedan surgir debido a que la imagen mental del problema en la propia mente del cliente está formada por una gran cantidad de conceptos asociados, que él tiene asumidos, pero

que probablemente, pueden no ser tan claros para aquellas personas que no conocen el proceso con profundidad.

- **Determinar, dentro del número de problemas, el grado de importancia y dificultad de cada uno de ellos**.
- **Definir qué resultados se esperan conseguir.** También resulta de especial importancia determinar el tipo de resultado que se busca: un modelo matemático, unas gráficas, unos informes, etc; definiéndolo de la forma más completa posible. Esto permite que las tareas posteriores se dirijan directamente hacia el objetivo buscado.
- **Implementar los resultados obtenidos**. Para conseguir el éxito, no solo es importante obtener unos resultados sino también tener en cuenta la forma con la que se aplicarán.

Como es lógico, toda esta información será desarrollada y contrastada con diferentes expertos hasta alcanzar un consenso entre las partes implicadas, pudiéndose utilizar algunas de las conocidas técnicas de consultas a expertos [MAO02].

En el apartado 3.4.4, se tratan algunas de las diferentes formas de realizar consultas a expertos.

3.2.4.2 SELECCIÓN DE LOS DATOS RELEVANTES

La identificación de los datos relevantes para una operación de *data mining* es una tarea que no puede ser automatizada y que por lo tanto debe ser realizada por el analista. Esta tarea consiste en la creación del conjunto de datos objetivo, enfocando la búsqueda en subconjuntos de variables y/o muestras de datos en donde realizar el proceso de análisis.

En esta fase deben de ser seleccionados, de forma coordinada por el analista y el usuario, los datos más relevantes del proceso, así como su disponibilidad. Esto implica consideraciones sobre la homogeneidad y variación a lo largo del tiempo de los datos, los grados de libertad o la estrategia de muestreo.

La obtención de datos puede realizarse directamente desde sistemas transaccionales, archivos o a partir de un almacén de datos (*Data Warehouse*). Habitualmente esta obtención viene predeterminada en función de la disponibilidad de los datos: existencia de bases de datos para el proceso, datos almacenados en archivos, necesidad de implantar un nuevo sistema de adquisición de datos, etc.

La situación ideal es la obtención de datos a partir de un *Data Warehouse*, lo que permite importantes ventajas en relación a la validez, cumplimiento con las especificaciones y localización centralizada de los datos. El tomar datos a partir de sistemas de operación puede suponer el riesgo de encontrarse con problemas de ruido o sin datos.

3.2.4.3 PREPROCESADO Y LIMPIEZA DE DATOS

El objetivo del preprocesado de datos es la transformación del conjunto original de datos en un nuevo conjunto de datos más significativo y manejable. Según [PYL99], un 60% del tiempo se dedica al preprocesado de los datos.

Según [FAM97]: El preproceso es una transformación *T* que transforma la matriz que contiene los datos reales del proceso, *X* , en una nueva matriz *Y* tal que:

- *Y* conserva la información de *X*.
- *Y* elimina al menos uno de los problemas contenidos en *X*.
- Y es más útil que X

El preproceso de los datos incluye cuatro etapas principales: Identificación y conversión de tipos, imputación (rellenar los datos inexistentes), identificación de espurios (outliers), eliminación de ruido y datos incompletos.

IDENTIFICACIÓN Y CONVERSIÓN DE ATRIBUTOS

Las primeras tareas de preprocesado, son las más arduas ya que, generalmente, deben consistir en identificar, casi manualmente, los diferentes tipos de atributos[7] existentes en la base de datos y convertirlos a otro tipo dependiendo de las necesidades posteriores. Fundamentalmente, podemos clasificarlos en los siguientes dos grupos [WIT00] [WAL99]:

- **Numéricos o Cuantitativos.** También algunas veces llamados “continuos”.
- **Nominales o Cualitativos.** También algunas veces llamados “discretos”. Aunque la literatura estadística [HAI99] introduce unos “niveles de medida” clasificados en los siguientes subgrupos (aunque fundamentalmente se usan solamente los dos primeros):
  - *Nominales*. Que corresponden a valores que tienen distintos símbolos generalmente denominados etiquetas o nombres[8]. Por ejemplo: colores. Un caso especial son los datos binarios (que solo pueden tener dos valores).
  - *Ordinales*. Que determinan un cierto ranking en las categorías. Por ejemplo: frío < templado < caliente, bajo < medio < alto, etc.
  - *Intervalos*. Que son valores que no solo están ordenados sino también medidos en unidades iguales con un cero arbitrario. Por ejemplo: temperaturas, años, etc.

[7] En este trabajo, se denominarán atributos a las variables correspondientes a las columnas de la tabla de la base de datos y observaciones a cada una de las filas de las misma.
[8] De ahí la palabra “nominal”.

- *Ratios*. Que corresponden con medidas donde está definido un punto cero inherente en si mismo. Por ejemplo: la distancia de un objeto a otro, tiene como cero la distancia del objeto a si mismo, temperatura en grados absolutos, edad desde el *Big Bang*, etc.

Los atributos, según el tipo que sean, deben ser acomodados a los algoritmos que se vayan a utilizar. De esta forma, muchas veces resulta necesaria la conversión de los datos para que puedan ser tratados convenientemente. Este proceso de conversión depende en gran manera de los esquemas utilizados. Por ejemplo [HAI99], algunos de los esquemas utilizan valores formados por escalas ordinales y solamente usan comparaciones *mayor-que*, *menor-que* para compararlos. Otros en cambio, usan escalas tipo ratios y usan distancias entre ellas. Es decir, **es necesario comprender cómo trabajan los algoritmos de minería de datos que vamos a utilizar para saber como preparar los datos.**

Conversión de Tipos de Variables

Dependiendo de los algoritmos que vayamos a utilizar deberemos transformar los datos de un tipo a otro. Por ejemplo, una atributo nominal no puede ser tratado por una red neuronal o un clasificador basado en árboles puede necesitar que los datos sea nominales. En [WIT00] y [PYL99] se profundiza con detalle en los diferentes tipos de transformaciones.

Muchas veces un atributo nominal puede ser convertido a un atributo ordinal simplemente indicándole al sistema unas reglas que relaciones estos. Por ejemplo, una serie de la forma: {bajo, alto, medio}, fácilmente puede ser convertida en una serie ordinal simplemente mediante la regla: *bajo* < *medio* < *alto*, o *alto* > *medio* > *bajo*, aunque otras veces las reglas no son tan claras.

La conversión de datos nominales a numéricos dependerá del conocimiento que tengamos sobre el grado de cercanía o alejamiento de unos con otros. Por ejemplo, si tenemos una serie de datos del tipo: {*error insignificante*, *error medio*, *error peligroso*}, y queremos alimentar con ellos una variable numérica de un modelo matemático, será necesario desarrollar una escala de medidas numéricas que se adapten convenientemente.

Un caso más interesante, es la conversión de una serie de valores numéricos a una serie de datos nominales. Esta transformación consistirá fundamentalmente en la creación de clases agrupando los conjuntos de datos según algún criterio preestablecido: distancia, similitud, en relación a otra variable, etc.

Otro tipo de transformaciones más avanzadas se basan en reglas *fuzzy* o difusas, capaces de tratar las incertidumbres mediante funciones aplicadas a cada valor del campo.

Una vez tenemos los tipos de atributos adaptados a nuestras necesidades, será conveniente realizar las siguientes fases:

- Detectar los espurios y eliminarlos.
- Rellenar los datos inexistentes.
- Eliminar el ruido.

Tareas que se tratan con más profundidad en los apartados 3.4.1 y 3.4.2.

3.2.4.4 TRANSFORMACIÓN DE LOS DATOS

La fase de transformación y reducción de los datos, es otra de las fases críticas dentro del proceso global que necesita de un buen conocimiento y una buena intuición que determinará el éxito o el fracaso del proceso de *data mining*.

Se busca, por un lado, preparar la información que se tiene para que pueda ser procesada por los algoritmos de minería de datos y además, reducir la cantidad de información redundante para simplificar las tareas posteriores.

Se busca por lo tanto:

- Extracción de las características (o atributos) útiles de los datos (reducción de dimensionalidad).
- Transformación de los datos con el objetivo de proporcionar una representación de los datos mas intuitiva y manejable.
- Fundamentalmente podemos destacar tres tareas específicas:
- Reducción de los Datos.
- Creación de Datos Derivados.
- Transformación de la distribución de los Datos.

Las técnicas y algoritmos más utilizados en estas tareas, se tratan con más detalle en los puntos 3.4.1 y 3.4.3.

3.2.4.5 USO DE LOS ALGORITMOS DE DATA MINING

Una vez se tienen los datos transformados y preparados en una base de datos normalizada, con variables poco correladas entre si, con los espurios y el ruido eliminados, y con una dimensión adecuada; sería el momento del uso de los algoritmos de minería de datos.

Las herramientas de *data mining* empleadas en el proceso de extracción de conocimiento se pueden clasificar en dos grandes grupos:

- Técnicas de verificación (en las que el sistema se limita a comprobar hipótesis suministradas por el usuario).
- Métodos de descubrimiento (en los que se han de encontrar patrones potencialmente interesantes de forma automática, incluyendo en este grupo todas las técnicas de predicción) [DAE02]. El resultado obtenido con la aplicación de algoritmos de descubrimiento **puede ser de carácter descriptivo o predictivo**. Las predicciones sirven para prever el comportamiento futuro de algún tipo de entidad mientras que una descripción puede ayudar a su comprensión.

Antes de poder utilizar los datos, casi siempre es necesario preprocesarlos para adecuarlos a las necesidades de las técnicas que se van a utilizar sobre ellos. Las técnicas de visualización son muy útiles en este momento, para aumentar el conocimiento previo de los datos y como paso previo a procesos posteriores. También ayudan a descubrir la estructura de clusters de los datos y posibles correlaciones entre ellos, así como en la detección de espurios..

Los algoritmos de minería de datos pueden ser utilizados para alguna de las siguientes tareas [WES98]:

- Agrupamiento o segmentación: Se busca la identificación de tipologías o grupos en los cuales los elementos guardan similitud entre sí y se diferencias de los otros grupos. Esto permite el tratamiento particularizado de cada una de estas agrupaciones.
- Asociación: Consiste en establecer las posibles relaciones entre acciones o sucesos aparentemente independientes. Así, se puede reconocer cómo la ocurrencia de un determinado suceso puede inducir la aparición de otro u otros.
- Secuenciamiento: Es un concepto similar al anterior, pero en el que se incluye el factor tiempo. Es decir, permite reconocer el tiempo que transcurre o suele transcurrir entre el suceso inductor y los sucesos inducidos.
- Reconocimiento de patrones: Se trata de analizar la asociación de una señal o información de entrada con aquella o aquellas con las que guarda mayor similitud, y que están ya catalogadas en el sistema. Generalmente se usan para identificar las causas de problemas o incidencias y buscar las posibles soluciones, siempre y cuando se disponga de la base de información necesaria en la que buscar.

- Previsión: Se busca establecer el comportamiento futuro más probable de una variable o una serie de variables a partir de la evolución pasada y presente de las mismas o de otras de las cuales dependan. Las técnicas asociadas a estas herramientas tienen ya un elevado grado de madurez.
- Simulación: Comparan la situación actual de una variable y su posible evolución futura según la variación probable de las que depende.
- Optimización: resuelve el problema de la minimización o maximización de una función que depende de una serie de variables, encontrando los valores de éstas que satisfacen la condición de máximo (típicamente beneficios), o mínimo (típicamente costes). Normalmente suele haber unas restricciones, que hacen que no todas las posibles soluciones sean aceptables, de modo que el universo de búsqueda se reduce a aquellas soluciones que satisfagan las restricciones.
- Clasificación: Agrupa a todas las herramientas que permiten asignar a un elemento la pertenencia a un determinado grupo o clase. Esto se lleva a cabo a través de la dependencia de la pertenencia a cada clase en los valores de una sede de atributos o variables. Se establece un perfil característico de cada clase y su expresión, en términos de un algoritmo o reglas, en función de las distintas variables. Se establece también el grado de discriminación o influencia de estas últimas. Con ello, es posible clasificar un nuevo elemento una vez conocidos los valores de las variables presentes en él.

Para desarrollar todos estos procesos, se dispone de una extensa gama de técnicas que le pueden ayudar en cada una de las fases de dicho proceso.

3.2.4.6 INTERPRETACIÓN DE LOS RESULTADOS

La interpretación y verificación de resultados es un proceso complejo. La obtención de resultados aceptables dependerá de factores como: definición de medidas del interés del conocimiento (de tipo estadístico. en función de su sencillez) que permitan filtrarlo de forma automática, existencia de técnicas de visualización para facilitar la valoración de los resultados o búsqueda manual de conocimiento útil entre los resultados obtenidos.

Un factor muy importante en esta fase, es el grado de experiencia y conocimiento del analista. La cantidad de información extraída depende en gran medida. del grado de conocimiento que el analista tenga del problema. así como de sus experiencias en la resolución de problemas similares.

Las decisiones tomadas durante esta fase irán encaminadas en dos direcciones:

- Verificación de resultados: La verificación de resultados incluye determinar el grado de cumplimiento de los objetivos finales establecidos durante la primera fase del proceso de *data mining*, así como la validación de la información extraída. Durante esta fase se debe verificar la coherencia de la información obtenida con otros tipos de

conocimiento ya previamente asentado y aceptado, resolviendo las posibles inconsistencias existentes. Si los objetivos finales han sido alcanzados, se procederá a la consolidación del conocimiento descubierto, incorporándolo al sistema. o simplemente documentándolo y enviándolo a la parte interesada. En caso contrario se procederá a la obtención de más información.

- Obtención de más información: La información extraída se utilizará como información a priori para la extracción de más información. Para ello será necesario retornar a alguna de las fases anteriores del proceso de *data mining* y modificar algunas de las decisiones tomadas durante esas fases, haciendo para ello uso de la nueva información obtenida. De esta forma el proceso de *data mining* se convierte en un proceso potencialmente iterativo. Alguna de las decisiones que pueden ser tomadas para la obtención de más información son, por ejemplo: recolección de nuevos datos, separación de datos en clases, transformaciones de las variables, eliminación de datos, selección de otros algoritmos de *data mining*, cambio en los parámetros introducidos en los algoritmos, delimitación del campo de búsqueda, etc.

3.2.5 HERRAMIENTAS DE MINERÍA DE DATOS

En la Figura 15 podemos apreciar, el resultado de una encuesta hecha en el conocido portal sobre Minería de Datos y Gestión del Conocimiento, KDnuggets [KDN02], donde se pregunta al encuestado sobre la herramienta de *Data Mining* que habitualmente usa.

Este tipo de encuesta es particularmente importante, porque nos da una idea de las aplicaciones que más están usando los profesionales y nos puede ayudar a decidir correctamente cuando tengamos que adquirir uno de estos programas.

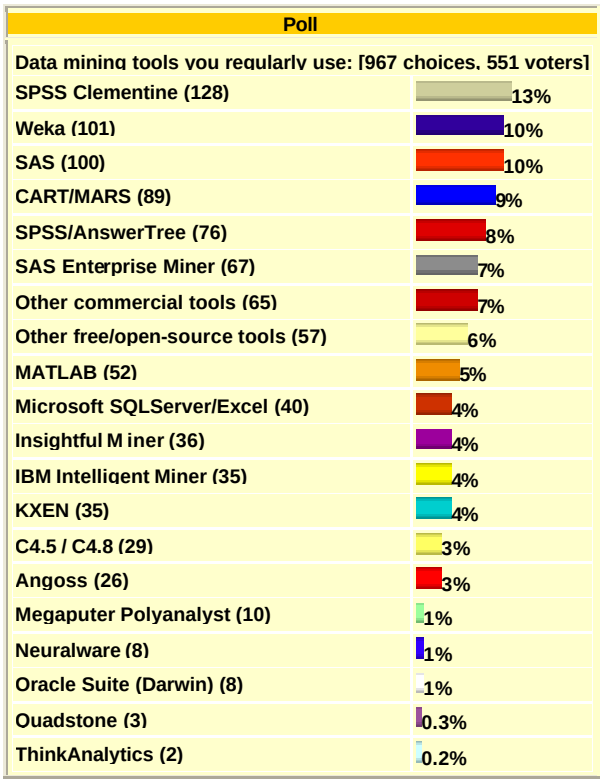


Figura 15. Herramienta de Minería de Datos usadas habitualmente (Junio de 2002). (http://www.kdnuggets.com/polls/data_mining_tools_2002_june2.htm)

La lista que aparece en la Figura 15 es una pequeña muestra de las múltiples aplicaciones que existen en el mercado. De ella destacan programas comerciales que forman parte de familias de aplicaciones estadísticas como por ejemplo: SAS (SAS, SAS EnterpriseMiner), o SPSS (SPPS Clementine, SPSS AnswerTree) y que son preferencia de aquellos que habitualmente trabajan con estos paquetes.

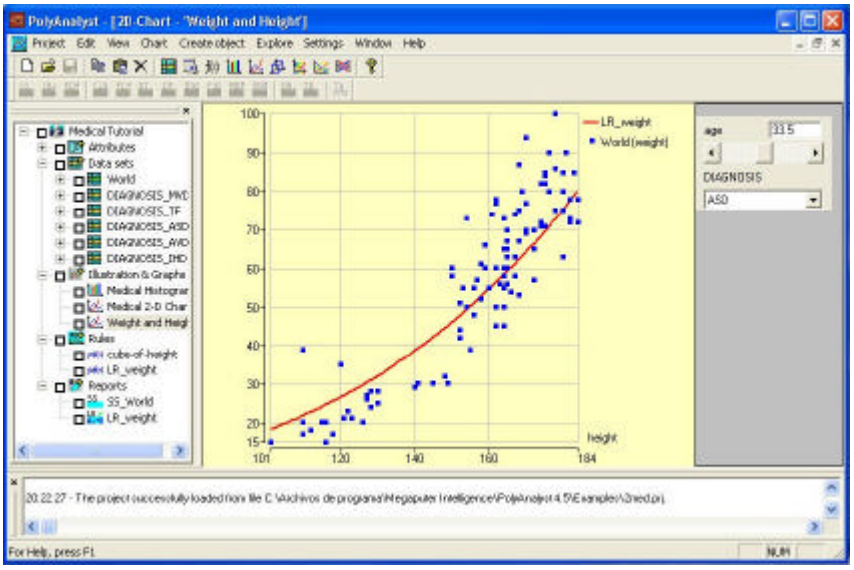


Figura 16. Ejemplo del programa comercial (PolyAnalyst (www.megaputer.com)).

Por otro lado, este tipo de aplicaciones comerciales contrastan con otras desarrolladas íntegramente en el campo de la Minería de Datos como por ejemplo: CART/MARS, IBM-I-Miner, Angoss, Megaputer PolyAnalyst, KXEN, etc.; y que fundamentalmente abarcan métodos estadísticos y de visualización combinados con algoritmos, bastante eficientes, más propios de Minería de Datos (clasificadores, generadores de reglas, clusterizado, etc.).

Habitualmente, estas herramientas disponen de sus propios entornos gráficos y suelen permitir al usuario hacer múltiples tareas, pero siempre acotados a las especificaciones de cada aplicación [GOE99]. El grado de eficiencia de cada herramienta depende de múltiples factores: tipos de algoritmos, funciones de tratamiento de la información, eficiencia de los algoritmos, generadores de informes, formas de pasar la información, etc.; aunque generalmente, los primeros de la lista cubren bastante bien las expectativas que se espera de ellos. Algunos de ellos, como el que se muestra en la Figura 16, pueden ser descargados de la red y evaluados durante un corto periodo de tiempo.

Por otro lado, en la segunda posición de la lista, se alza la herramienta WEKA [WEK02]. Esta aplicación es de libre distribución (licencia GPL) y destaca por la cantidad de algoritmos que presenta así como por la eficiencia de los mismos. Esta aplicación está desarrollada por miembros de la Universidad de Waikato (Nueva Zelanda) y es una muy buena opción, tal y como muestra la encuesta, frente a las costosas distribuciones comerciales.

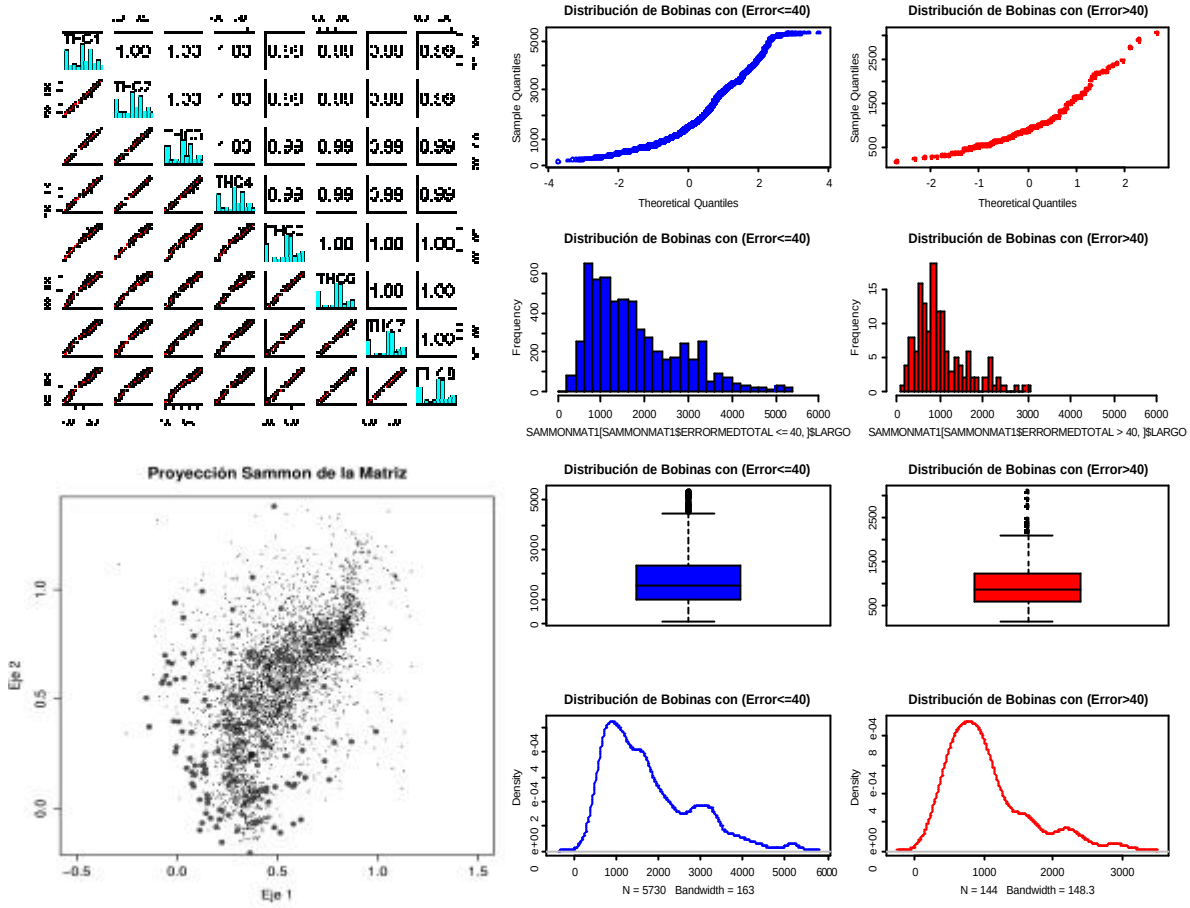


Figura 17. Algunas de múltiples posibilidades que ofrece el programa R *para visualización de datos.*

Se han obtenido excelentes resultados con las herramientas de libre distribución siguientes:

- R: Herramienta excelente para el análisis de datos basada en el conocido programa estadístico S-Plus y con un manejo de las matrices y variables equivalente a MATLAB. Este programa es muy útil para el análisis estadístico, transformación y manipulación de los datos. Está compuesto de múltiples librerías para realizar: gráficos y análisis estadísticos de todo tipo, regresiones lineales y no lineales, modelizado, clusterizado, etc.; y sigue en continua evolución. Cabe destacar la excelente asesoría técnica (responden las preguntas en pocas horas) llevada a cabo principalmente por algunos de los principales profesores e investigadores en estadística del mundo.
- WEKA: Programa de libre distribución que abarca algoritmos clasificadores de todo tipo, generadores de reglas, herramientas de clusterizado, etc. Esta aplicación proporciona gran cantidad de herramientas para la realización de tareas propias de minería de datos y permite la programación en JAVA de algoritmos más sofisticados.

- XELOPES: Otra librería de libre distribución con cantidad de funciones para minería de datos. Permite la implementación en JAVA o C++.
- SNNS: Aplicación de libre distribución para el desarrollo, entrenamiento y testeo de multitud de tipos diferentes de redes neuronales. Muy útil para desarrollar clasificadores sofisticados y modelos basados en redes neuronales.
- XmdvTool, Xgobi, IBM-OpenDX, Visipoint: Otras herramientas con licencia GPL que tienen diferentes funciones de visualización muy útiles para encontrar patrones ocultos en los datos.

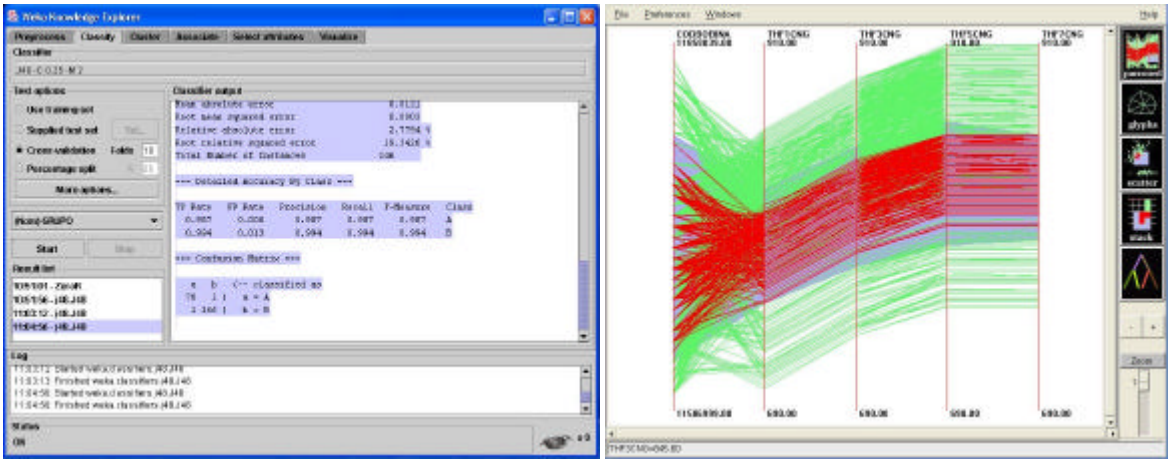


Figura 18. Generación de un árbol con un clasificador del programa WEKA *(izquierda) y diagrama de coordenadas paralelas realizado con el programa* XmdvTool *(derecha)*

Hoy en día, existen herramientas de libre distribución, realmente sorprendentes. Las que se acaban de enumerar, y muchas otras, permiten múltiples posibilidades. Los programas R y WEKA, usados conjuntamente, no solo se pueden utilizar como herramientas de aplicación, sino también, como auténticos entornos de programación. Esta característica, como es lógico, unido a que su coste es cero por ser programas con licencia GPL, aporta múltiples ventajas para los campos de investigación y docencia en el aprendizaje y desarrollo de la Minería de Datos [MAR02].

La experiencia obtenida con estas últimas herramientas, nos ha demostrado que las ventajas en el campo de la investigación son muchas...

3.2.6 APLICACIONES DEL DM Y TENDENCIAS

Hasta ahora esta tecnología ha sido de gran ayuda en áreas como la banca (detección de fraudes, análisis de morosidad o segmentación del mercado), telecomunicaciones (control de fugas de clientes, control de redes, ventas cruzadas), seguros (riesgos, mercadeo) y comercial. Entre todos estos usos destacan los referenciados en [BER97][BER00] [BRA96][WES98][PAR01].

Actualmente hay un número creciente de organizaciones inmersas en proyectos de *data mining*. La tecnología se puede aplicar a cualquier organización que disponga de una gran cantidad de datos y que se plantee explotarlos para obtener reglas de negocio o mejorar el servicio que presta.

Se presentan a continuación algunos ejemplos [KAR00]:

- Predicción automática de tendencias y comportamientos [CHA01]:
  - Marketing dirigido: analizar datos sobre envíos por correo publicitarios para identificar el segmento más apropiado para realizar un nuevo mailing.
  - Comportamiento del cliente en supermercados en ciertos días de la semana, de forma que se puedan promocionar ciertos productos en fechas determinadas.
  - Análisis de las ventas de una compañía farmacéutica para reforzar las acciones de marketing en los hospitales y médicos de mayor impacto.
  - Identificación de mejores clientes para el lanzamiento de una nueva tarjeta de crédito.
  - Detección de fraudes en distribuidores de una empresa multinacional.
- Descubrimiento automático de patrones ocultos:
  - Análisis de datos de ventas de productos para identificar aquellos que sin estar relacionados entre sí, se compran juntos a menudo.
  - Detección de transacciones fraudulentas realizadas con tarjeta de crédito
  - Detección de errores de grabación de datos.
  - Búsqueda de tendencias en la bolsa.
  - Determinación de las causas que producen los fallos en sistemas de producción.
  - Descriptores que “expliquen” los fallos de calidad en el producto.
- Prospectiva:
  - Conseguir modelos aplicables a bases de datos para la selección priorizada de nuevos clientes.

- Estudios de respuesta ante un posible cambio de precios.
- Generar nuevos modelos de control de un proceso.

➢ Segmentación y Clustering:

- Dividir la base de datos de clientes en segmentos relativamente homogéneos basados en conductas estudiadas.
- Una organización bancaria puede estudiar qué grupo de usuarios tiene una alta probabilidad de cancelar su cuenta en función de determinados parámetros y a continuación realizar acciones especificas para evitar que ocurra.
- Clasificar los tipos de clientes de una empresa de seguros.

➢ Aplicaciones científicas:

- Análisis de los datos obtenidos a partir de instrumental científico. Esto permite el análisis de los datos para investigación, la formación de hipótesis y teorías.
- Aplicaciones en Biología y Medicina, bioinformación que se traduce en minería de bases de datos distribuidas (por ejemplo el proyecto Genoma).

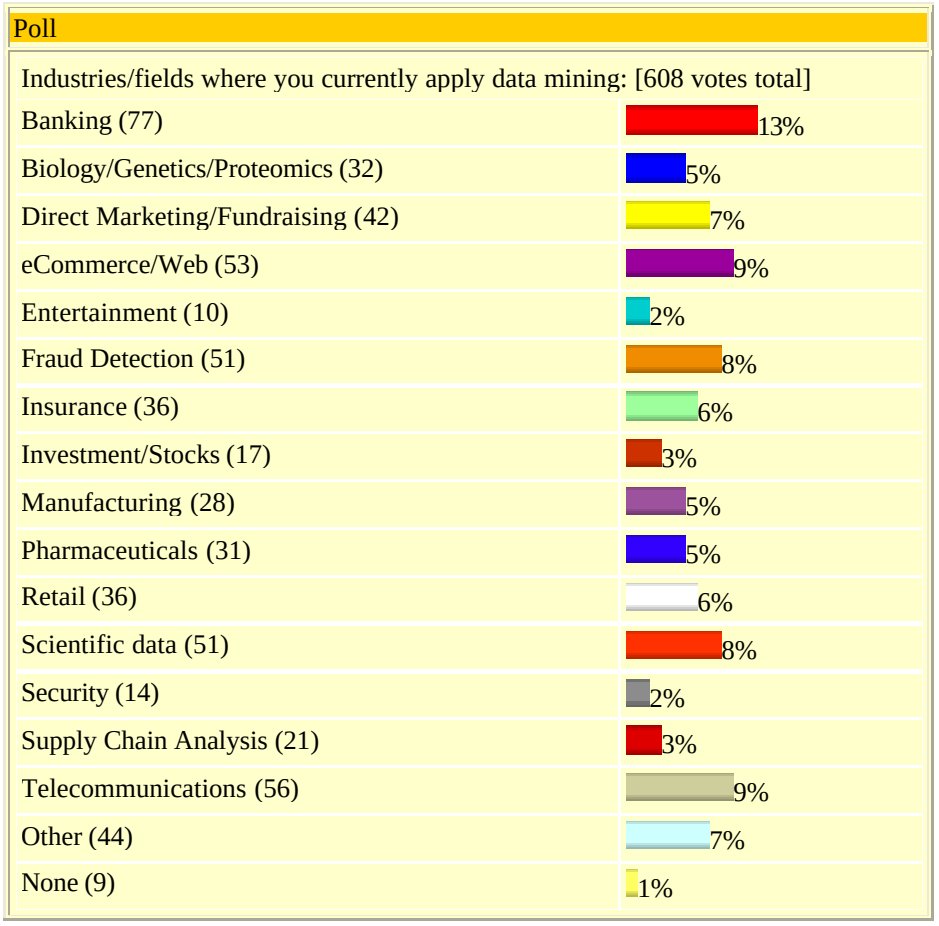


Figura 19. Campos donde se aplica la Minería de Datos (encuesta realizada en Junio de 2002 [KDN02]).

3.2.7 DIFICULTADES EN LA APLICACIÓN DEL DM

Se enumeran a continuación algunos de los problemas más habituales a los que se enfrenta cualquier proyecto de *data mining*:

- Uno de los mayores problemas es que el número de posibles relaciones es demasiado grande, y resulta prácticamente imposible validar cada una de ellas. Para resolver este problema, se utilizan estrategias de búsqueda, extraídas del área de aprendizaje automático (ML) [BER97][SPS99].
- Además todas estas herramientas siguen funcionando mejor fijándoles objetivos de búsqueda concretos. Si bien la minería de datos da la impresión de que se puede simplemente aplicar como herramienta a los datos, se debe tener un objetivo, o al menos una idea general de lo que busca.
- El coste de esta prospección de datos debe ser coherente con el beneficio esperado. Si bien las herramientas (hardware y software) han bajado su precio, el coste en tiempo, personal y consultoría se ha incrementado, llegando en algunos casos a hacer inviable el proyecto.
- Suele funcionar mejor en problemas ligados a empresas de éxito que en otros casos, debido a la gran dependencia que estas herramientas tienen respecto a todos los estamentos de la empresa, desde mantenimiento a compras.
- Es necesario trabajar en estrecha colaboración con expertos en el negocio para definir modelos. Su ausencia y/o disponibilidad marca el proyecto.
- Otro problema es que la información muchas veces está corrompida, tiene ruido, o simplemente le faltan partes. Para esto, se aplican técnicas estadísticas que ayudan a estimar la confiabilidad de las relaciones halladas.

3.3 METODOLOGÍAS DE APLICACIÓN DEL DM

A la vista de las dificultades anteriores y para utilizar estas técnicas de forma eficiente y ordenada, es preciso aplicar una metodología estructurada. A este respecto se proponen las siguientes metodologías, siempre adaptables a la situación a la que se aplique [BRA96] [WES98].

3.3.1 METODOLOGÍA CRISP-DM

CRISP-DM (*CRoss-Industry Standard Process for Data Mining*) [CRI00][ABA01], es una metodología para el desarrollo de proyectos de *data mining* que se ha convertido en un estándar de facto.

El consorcio CRISP-DM, responsable de esta metodología, está integrado por importantes empresas europeas y estadounidenses que poseen una amplia experiencia en proyectos de análisis de datos relacionados con muy diversos campos de la industria.

La metodología para minería de datos CRISP-DM, está descrita como un proceso jerárquico, que consiste en un conjunto de tareas descritas en cuatro niveles de abstracción, desde el general hasta el específico: fase, tareas generales, tareas específicas e instancias de proceso.

Al nivel más alto, el proceso está organizado en un número de fases; cada fase consiste en varias tareas generales de segundo nivel. Este segundo nivel se denomina genérico porque se pretende que sea lo suficientemente general como para cubrir todas las posibles situaciones. Las tareas generales deben ser lo más completas y estables posibles. Se entenderán tareas completas a aquellas que cubran completamente el proceso de análisis y sus posibles aplicaciones. Por otro lado, se entiende como estables aquellas tareas que cubran incluso desarrollos aún no conocidos.

El tercer nivel, el nivel de tareas especializadas, es el lugar en el que se describe cómo las acciones de las tareas generales (nivel 2) se deberían desarrollar en ciertas situaciones específicas. Por ejemplo, en el segundo nivel puede existir una tarea general llamada "limpieza de datos". El tercer nivel describe cómo difiere esta tarea de unas situaciones a otras, por ejemplo la limpieza de valores numéricos y la limpieza de valores categóricos o si el tipo de problema es un clusterizado a un modelo predictivo.

La descripción de fases y tareas en pasos discretos desarrollados en un orden específico representa una secuencia idealizada de eventos. En la práctica, muchas de estas tareas pueden ser desarrolladas en un orden diferente y frecuentemente será necesario volver atrás a tareas previas y repetir ciertas acciones. El modelo de procedimiento no pretende abarcar todas estas posibles rutas a lo largo del proyecto porque esto requeriría un modelo enormemente complejo.

El cuarto nivel, el nivel de instancias de proceso, es un conjunto de acciones, decisiones y resultados sobre el proceso de *data mining* en curso. Una instancia de proceso se organiza de acuerdo con las tareas definidas en los niveles superiores, pero representa lo que pasa en realidad en un proceso particular, más que lo que pasa en general.

Horizontalmente, la metodología CRISP-DM distingue entre el modelo de referencia y la guía del usuario. El modelo de referencia presenta una vista rápida de las fases, tareas y sus salidas y describe lo que hay que hacer en un proyecto de *data mining*. La guía del usuario da consejos y trucos mucho más detallados para cada fase y para cada tarea dentro de una fase y describe cómo desarrollar un proyecto de análisis de datos.

3.3.1.1 CONTEXTO DEL PROYECTO

El contexto del proyecto dirige el paso entre el nivel general y el especializado en el CRISP-DM. Actualmente se distinguen cuatro dimensiones diferentes de contextos:

- El dominio de aplicación: es el área específica en la cuál el proyecto tiene lugar.
- El tipo de problema: describe la clase(s) específica(s) de objetivo(s) que el proyecto va a abarcar.
- El aspecto técnico: cubre temas específicos que describen diferentes desafíos técnicos que puedan ocurrir durante el proceso.
- La dimensión de herramientas y técnica: especifica qué herramientas y/o qué técnicas se van a aplicar durante el proyecto.

Un contexto específico de minería de datos es un valor concreto para una o más de estas dimensiones. Por ejemplo, un proyecto que abarca un problema de clasificación en estimación de producción constituye un contexto específico. Cuantos más valores de dimensiones de diferentes contextos se cubran, más concreto es el contexto.

3.3.1.2 PROYECCIÓN

Se distinguen 2 tipos diferentes de proyecciones entre los niveles genérico y especializado en CRISP-DM:

- Proyección para el presente: si sólo se está aplicando el modelo genérico del proceso para llevar a cabo un solo proyecto y se pretende proyectar las tareas generales y sus descripciones para ese proyecto concreto, se habla entonces de una proyección sencilla para (probablemente) un solo uso.
- Proyección para el futuro: si se especializa el modelo genérico del proceso de acuerdo a un contexto predefinido encaminándolo a un modelo de proceso especializado para usar en el futuro en contextos similares.

Siendo evidente que el tipo de proyección apropiado depende del contexto específico y de las necesidades de cada organización.

3.3.1.3 CÓMO PROYECTAR

La estrategia básica para proyectar el modelo genérico de proceso al nivel especializado es la misma para todos los tipos de proyecciones:

- Analizar el contexto específico.
- Eliminar cualquier detalle que no sea aplicable en dicho contexto.
- Añadir detalles específicos al contexto.
- Especializar (o instanciar) contenidos genéricos de acuerdo a características concretas del contexto.
- Posiblemente, y para una mayor claridad, renombrar contenidos genéricos.

CRISP-DIM define las diferentes fases de las que consta un proyecto, las tareas correspondientes y las relaciones entre ellas.

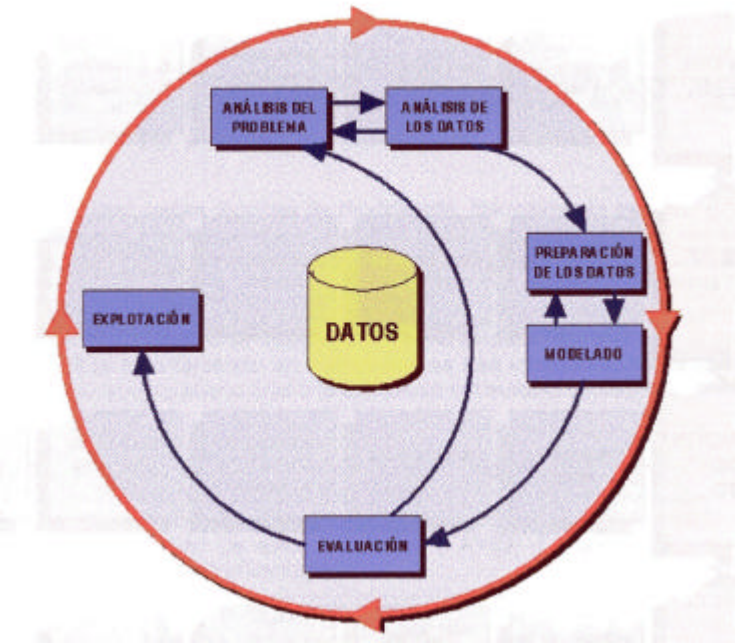


Figura 20. Fases del modelo de referencia CRISP-DM.

En la Figura 20 se muestran las 6 fases definidas. El orden de las mismas no es estricto, ya que frecuentemente a lo largo del desarrollo del proyecto, es necesario volver atrás en numerosas ocasiones, dependiendo de los resultados obtenidos en las fases previas. Las flechas indican las relaciones más habituales entre las fases. El círculo exterior simboliza la naturaleza cíclica del d*ata mining*, ya que la solución a la que finalmente se llega puede conducir al planteamiento de nuevas cuestiones que den origen a otros proyectos.

A continuación, se resumen las tareas genéricas en las que se desglosan cada una de las fases y las salidas generadas por cada una de ellas. A continuación se describen detalladamente.

El ciclo de vida de un proyecto de *data mining* consiste en 6 fases:

- **Análisis del Problema**: Fase inicial que incluye la comprensión de los objetivos y requerimientos del proyecto desde una perspectiva de negocio, con el fin de convertirlos en objetivos y en una planificación.
- **Análisis de los Datos**: Recolección inicial de datos para familiarizarse con ellos, identificar su calidad y descubrir las relaciones entre los más evidentes para las primeras hipótesis de relaciones ocultas entre ellos.
- **Preparación de los Datos**: Construcción de la base de datos a partir de los datos primarios. Estas tareas se desarrollan en numerosas ocasiones y no de una forma muy estructurada. Incluye la selección de tablas, registros y atributos, así como su transformación y preparación para las herramientas de modelizado.
- **Modelizado**: Se seleccionan y aplican varias técnicas de modelizado. Normalmente existen varias técnicas para el mismo problema y cada una exige una entrada de datos particular por ello es necesario interactuar con la fase anterior para adecuar la base de datos de trabajo. Los parámetros son calibrados.
- **Evaluación**: Una vez creado un buen modelo se debe evaluar el rendimiento del mismo y la integridad de todos los pasos sobre todo teniendo en cuenta que se han introducido todos los criterios de negocio. Se debe dar el visto bueno final a la aplicación del modelo de DM.
- **Desarrollo**: Normalmente los proyectos de DM no terminan en la implantación del modelo sino en el incremento de conocimiento obtenido de los datos. Para ello es imprescindible documentar y presentar los resultados de manera comprensible. Además debe asegurarse el mantenimiento de la aplicación y la posible difusión de estos resultados [FAY02].

La tabla siguiente resume los pasos más importantes de esta metodología.

Análsis del Problema	Análisis de los Datos	Preparación de los datos	Modelado	Evaluación	Explotación
Determinación de los objetivos Conocimiento previo Objetivos Criterios de éxito Evaluación de la situación Recursos disponibles Requerimientos, supuestos y restricciones Riesgos y contingencias Terminología Costes / Beneficios Determinación de los objetivos del data mining Objetivos del data mining Criterios de éxito del data mining Elaboración de la planificación Planificación del proyecto Valoración inicial Técnicas / herramientas	Adquisición de datos Adquisición de datos Descripción de los datos Descripción de los datos Exploración de datos Resultados de la exploración de los datos Verificación de la calidad de los datos Calidad de los datos	Datos iniciales Descripción de los datos Selección de datos Criterios de selección de datos Limpieza de los datos Limpieza de los datos Generación de variables adicionales Variables adicionales Integración de orígenes de datos Datos agrupados Cambios de formato de los datos Datos modificados	Selección de la técnica de modelado Técnica de modelado Supuestos de la técnica de modelado Diseño del método de evaluación Método de evaluación del modelo Generación del modelo Parámetros del modelo Modelos Descripción del modelo Evaluación del modelo Evaluación del modelo Revisión de Los parámetros del modelo	Evaluación de los resultados Valoración de los resultados del data mining Modelos válidos Revisión del proceso Revisión del proceso Determinación de las siguientes acciones Lista de posibilidades Decisión	Planificación de la explotación Plan de utilización Planificación de la monitorización y mantenimiento Plan de monitorización y mantenimiento Generación del informe final Informe final Presentación final Revisión del proyecto Experiencia adquirida

Tabla 9. Pasos más importantes dentro del Modelo CRISP-DM.

3.3.2 METODOLOGÍA SEMMA

SAS Institute desarrollador de esta metodología [SAS01], la define como el proceso de selección, exploración y modelizado de grandes cantidades de datos para descubrir patrones de negocio desconocidos.

El nombre de esta terminología es el acrónimo correspondiente a los cinco pasos básicos del proceso "*Sample, Explore, Modify, Model and Assess*". El esquema siguiente presenta la dinámica del sistema.

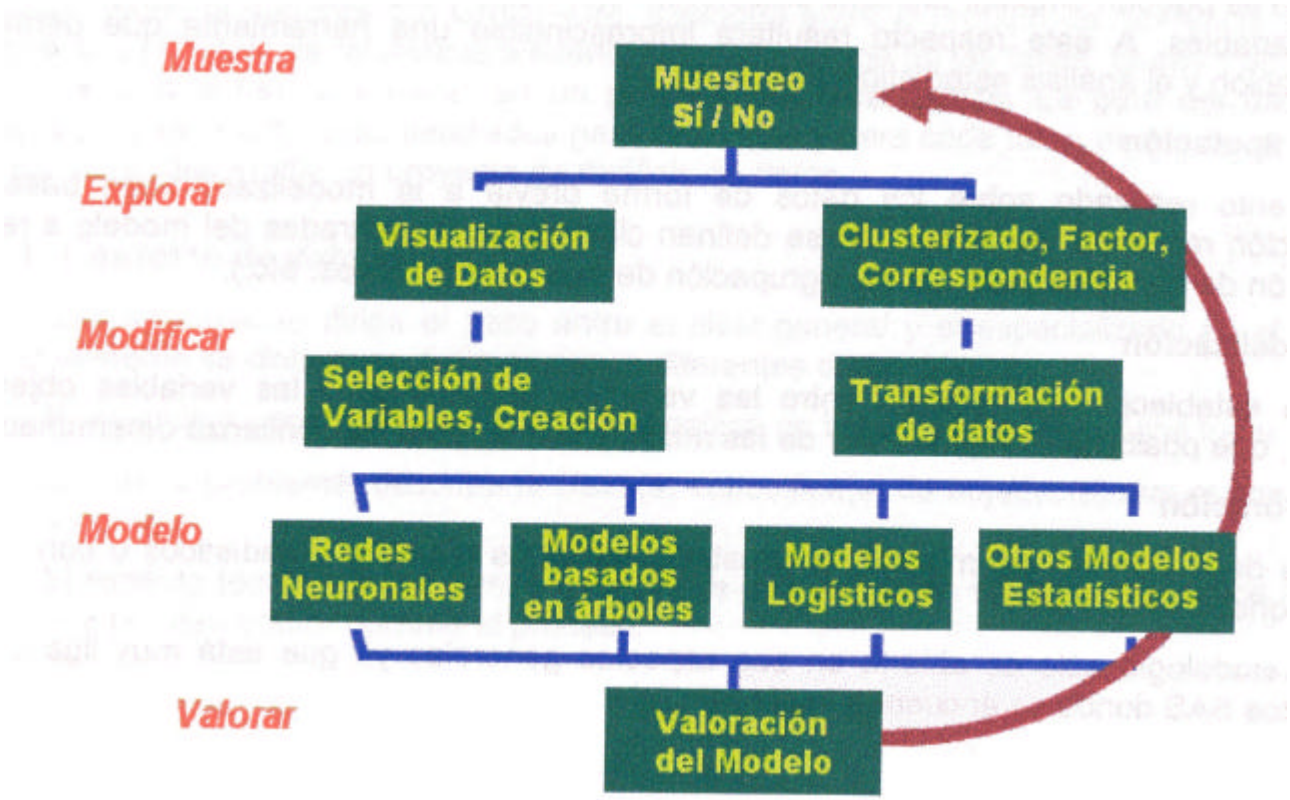


Figura 21. Metodología de SEMMA.

3.3.2.1 MUESTREO

Extracción de la población muestral sobre la que se va a aplicar el análisis. En ocasiones se trata de una muestra aleatoria, pero puede ser también un subconjunto de datos del *data warehouse* que cumplan unas condiciones determinadas. El objeto de trabajar con una muestra de la población en lugar de con toda ella, es la simplificación del estudio y la disminución de la carga de proceso. La muestra más óptima será aquella que, teniendo un error asumible, **contenga el número mínimo de observaciones.**

En el caso de que se recurra a un muestreo aleatorio, se debería tener la opción de elegir entre:

- El nivel de confianza de la muestra (usualmente el 95% o el 99%).
- El tamaño máximo de la muestra (número máximo de registros), en cuyo caso el sistema deberá informar del error cometido y la representatividad de la muestra sobre la población original.
- El error muestral que está dispuesto a cometer, en cuyo caso el sistema informará del número de observaciones que debe contener la muestra y su representatividad sobre la población original.

Para facilitar este paso se debe disponer de herramientas de extracción dinámica de información con o sin muestreo (simple o estratificado). En el caso del muestreo, dichas herramientas deben tener la opción de, dado un nivel de confianza, fijar el tamaño de la muestra y obtener el error o bien fijar el error y obtener el tamaño mínimo de la muestra que proporcione este grado de error.

3.3.2.2 EXPLORACIÓN

Una vez determinada la población que sirve para la obtención del modelo se deberá determinar cuáles son las variables explicativas que van a servir como entradas al modelo. Para ello es importante hacer una exploración de la información disponible de la población que permita eliminar variables que no influyen y agrupar aquellas que presentan efectos similares.

El objetivo es simplificar en lo posible el problema con el fin de optimizar la eficiencia del modelo. En este paso se pueden emplear herramientas que permitan visualizar de forma gráfica la información, utilizando las variables explicativas como dimensiones.

También se pueden emplear técnicas estadísticas que ayuden a poner de manifiesto relaciones entre variables. A este respecto resultará imprescindible una herramienta que permita la visualización y el análisis estadístico integrados.

3.3.2.3 MANIPULACIÓN

Tratamiento realizado sobre los datos de forma previa a la modelización, en base a la exploración realizada, de forma que se definan claramente las entradas del modelo a realizar (selección de variables explicativas, agrupación de variables similares, etc.).

3.3.2.4 MODELIZACIÓN

Permite establecer una relación entre las variables explicativas y las variables objeto del estudio, que posibilitan inferir el valor de las mismas con un nivel de confianza determinado.

3.3.3 METODOLOGÍA CRITIKAL

Desarrollada en el marco de un proyecto ESPRIT 22700 [ITI99][SCO99] se caracteriza por su fuerte integración con el desarrollo del *data warehouse* y no es de completa distribución libre. Los pasos que plantea son similares a los del CRISP-DM. Su principal fortaleza radica en la extensa valoración de otras herramientas antes de comenzar uno de los cuatro proyectos de DM en los que clasifica todos los problemas.

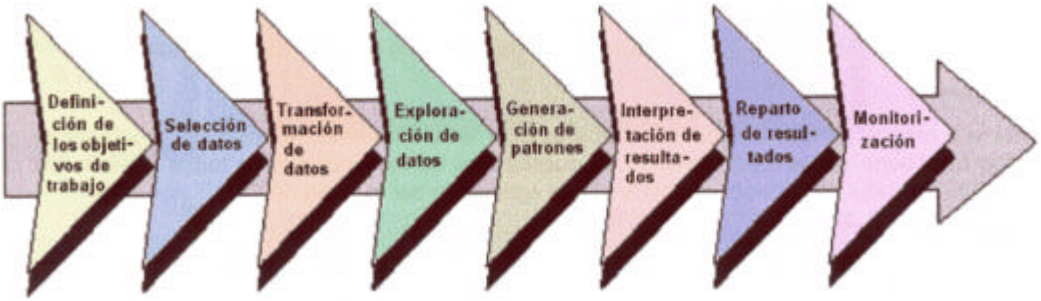


Figura 22. Pasos de la metodología CRITIKAL.

La valoración del coste se concreta en un formulado como el siguiente:

Lista de potenciales áreas de aplicación

Criterio \ Áreas de Aplicación (AA)	Peso (Ex.)	AA_1	AA_2	AA_3
Beneficio Potencial	0,25			
Facilidad de validación	0,10			
Duración del proyecto	0,05			
Tiempo de retorno de la inversión	0,10			
Disponibilidad de los datos	0,20			
Probabilidad de cambios en el entorno	0,15			
Usabilidad de los resultados	0,10			
Repetibilidad	0,05			
Σ	1,00			
Ranking				

Ranking de las diferentes áreas de aplicación

Figura 23. Formulario para valoración del coste.

Como puede observarse, no sólo se tiene en cuenta el beneficio potencial obtenible sino también elementos intrínsecos del proyecto de minería de datos como la posibilidad de validación, la duración y el tiempo de retorno de la inversión, el acceso a los datos, entorno cambiante, los usos de los resultados y, por último, la capacidad de repetición de este análisis para garantizar la futura transferencia del estudio.

3.3.4 METODOLOGÍA DE LAS "5 A'S"

- A título anecdótico, la metodología "5A's" la definió SPSS [SPS02] antes de desarrollar junto con otras empresas la metodología "CRISP-DM". Su nombre viene de las cinco palabras siguientes: "*Assess*, *Access*, *Analyze*, *Act* y *Automate*". El significado de estas cinco palabras al contexto del *data mining* se explica a continuación:
- Asesorar (*Assess*): La clave de una minería de datos no solo está en la tecnología que se va a usar, sino también en la forma en que se va a manejar y transmitir la información, ya que al final el objetivo **es asesorar en la toma de decisiones**. El *data warehouse*, la tecnología de manejo de la información, las herramientas a utilizar, etc.; deben estar orientadas a los procesos, estrategias y objetivos de la organización.
- Acceso (*Access*): Una vez que el contexto está planteado, entramos en la parte del proceso donde la tecnología puede ayudarnos. Es indispensable un sistema que nos ayude a recolectar la información de la mejor forma y con la mayor calidad posible.
- Analizar (*Analyze*): En esta fase, se hace uso de las diferentes herramientas de data mining para analizar la información y extraer el conocimiento deseado.
- Actuar (*Act*): Una vez se extraen conclusiones importantes, es conveniente plantear las posibles soluciones o aplicaciones. Generalmente hay que hacer uso de informes y gráficos que puedan ser interpretados por los agentes que toman las decisión de actuar.
- Automatizar (Automate): La actividad del DM no termina cuando las decisiones se han tomado, sino que es necesario monitorizar los efectos de las mismas. Como esto necesita ser validado continuamente, muchas veces es necesario desarrollar un sistema automático que permita con "una simple pulsación de un botón", monitorizar los resultados de las decisiones adoptadas.

3.3.5 METODOLOGÍAS DE DM: CONCLUSIONES

Últimamente la metodología CRISP-DM es la más aplicada a nivel mundial (ver Figura 24). Las razones fundamentales son debidas a su generalización y practicidad, además de su libre utilización.

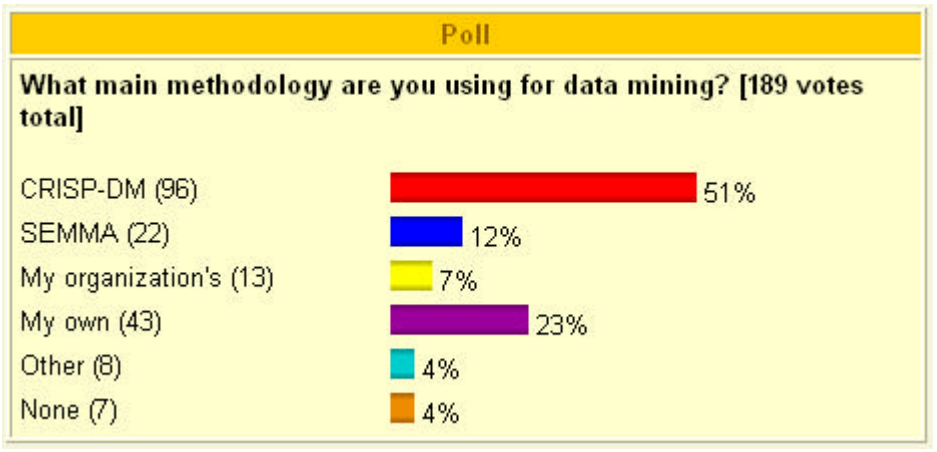


Figura 24. Encuesta realizada en [KDN02] sobre la metodología usada en agosto del 2002.

Un punto en común que se puede observar en todas estas metodologías **es que todas se basan en un modelo espiral, de forma que se retorna a las primeras fases del proceso pero a un nivel superior ya que la comprensión alcanzada es mayor. Además, en todas vemos la importancia del análisis inicial del contexto y de la validación final de la toma de decisiones.**

3.4 TÉCNICAS Y ALGORITMOS DE DATA MINING

La clasificación de las técnicas y algoritmos de *data mining* puede ser efectuada de múltiples formas [WAN99][WIT00].

En la práctica, quizá, una de las clasificaciones más interesantes de los algoritmos de DM es la que corresponde con su función. Como es lógico, ésta dependerá de la definición que adoptemos con respecto al proceso del *data mining*.

Si consideramos la parte de la minería de datos de las técnicas de extracción de conocimiento dentro del proceso del *KDD* (*Knowledge Discovery Process*) podemos clasificarlas según la función que desempeñan fundamentalmente en [WIT00] [PRU02] [FAY96a] [AND98] [AND99] [WAN99]:

- Clasificadores: Que clasifican datos en clases predefinidas.
- Algoritmos de regresión: A partir de los datos generan una función predictora.
- Descubrimiento de Reglas de Asociación: Búsqueda de relaciones entre variables.
- Modelado de Dependencias: Generación de modelos que "expliquen" las dependencias entre atributos.
- Clusterizado o agrupamiento: Búsqueda de conjuntos en los que agrupar los datos cuando las clases son desconocidas.
- Aprendizajes basados en casos: Se basa en indexar y recordar los casos más significativos, de forma que los nuevos casos son clasificados según el descriptor más próximo.
- Compactación: Búsqueda de descripciones más compactas de los datos. Técnicas de reducción de dimensión.
- Detección de desviaciones: Buscan desviaciones importantes de los datos respecto a valores anteriores.
- Sumarización: Describe las propiedades que comparten aquellas observaciones que pertenecen a una misma clase.
- Técnicas de Minería Datos Aplicados a Datos Secuenciales. Orientadas a la búsqueda de relaciones en datos que transcurren secuencialmente.

Y si consideramos también los algoritmos que pueden ayudar a las tareas previas de preprocesado y preparación de los datos, podemos añadir [MAR02]:

- Técnicas de visualización multivariante [FAY02].
- Algoritmos de detección y eliminación de espurios.
- Algoritmos de detección de datos ausentes y rellenado de los mismos.
- Otros algoritmos para el tratamiento y preprocesado de la información.

En la figura siguiente, se muestra un esquema que intentan agrupar las familias de algoritmos y técnicas basándose en lo visto anteriormente. Esta clasificación se ha realizado según el uso más habitual de cada algoritmo [HAN01][CRI00][THU99][CAB97], **aunque algunas de estas herramientas pueden ser utilizadas en varias de las aplicaciones mostradas**. Por ejemplo, las redes neuronales pueden servir no solo para desarrollar modelos clasificadores o predictores, sino también como algoritmos de segmentación, proyectores, filtros, etc.; así como algunos tipos de proyectores, generadores de reglas, etc.

El primer grupo abarca el **proceso de exploración de los datos (Exploratory Data Análisis EDA)** mediante técnicas iterativas y visuales, que permitan hacerse una idea de la estructura intrínseca de los datos, dominios, tipos de agrupamientos, puntos de operación, espurios, etc. Fundamentalmente se hace uso de descriptores estadísticos y métodos de visualización multivariante o basados en proyectores que permitan proyectar en 2 o 3 dimensiones datos embebidos en dimensiones superiores. Muchas de las técnicas descritas en este grupo son utilizadas en fases posteriores para analizar resultados intermedios, validarlos, extraer nuevas conclusiones, detectar espurios o patrones anómalos, etc.

El segundo grupo corresponde con todas aquellas técnicas de **preprocesado y tratamiento de la información** que sirvan para:

- Detectar, analizar y eliminar espurios.
- Rellenar los datos ausentes mediante técnicas de imputación y eliminar patrones incompletos.
- Transformar los datos y normalizarlos.
- Reducción de la dimensión mediante proyectores.

El tercer grupo describe algunas de las técnicas que buscan obtener **modelos descriptores de todos los datos o parte de los datos**, fundamentalmente buscando agrupamientos (segmentación) mediante técnicas de clusterizado, descubriendo patrones o reglas que describan dependencias o asociaciones entre variables mediante métodos generadores de reglas, etc.; o reglas que describan conceptos o clases inducidos de la relación de los datos existentes. Cabe decir, que muchos de los algoritmos clasificadores que se ven en el siguiente punto, pueden ser utilizados para estas tareas.

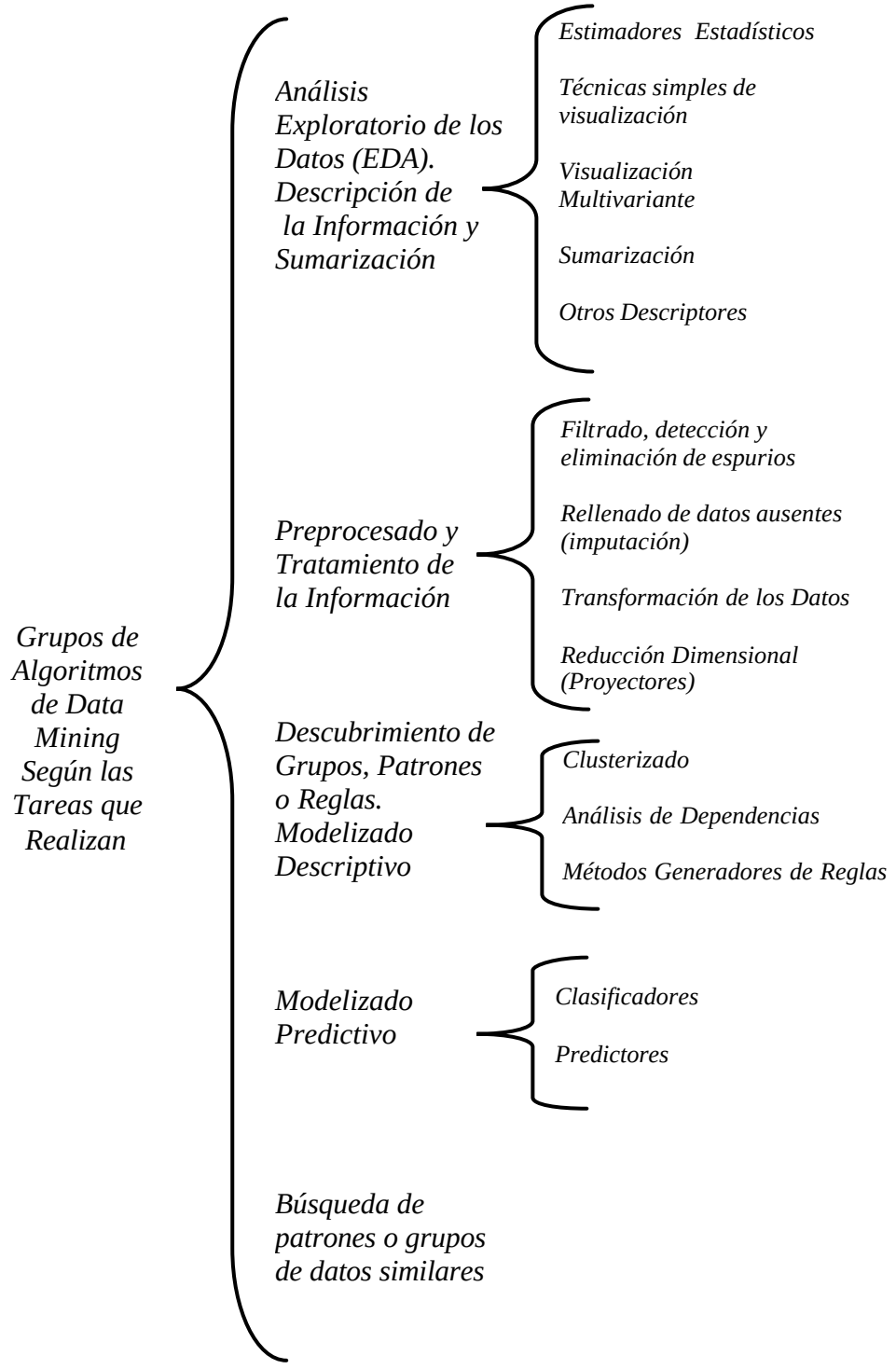


Figura 25. Grupos de algoritmos de data mining según las tareas que realizan.

En el cuarto grupo se tratan los algoritmos encargados de la generación de **modelos predictivos** fundamentalmente agrupados en:

- Clasificadores: Tienen como objetivo construir un modelo a partir de la información que se le suministra en el proceso de creación y una serie de clases que se le indican generalmente en una variable cualitativa, de forma, que sea capaz de clasificar nuevas observaciones en su clase correcta. Las técnicas existentes son muchas: árboles de decisión, generadores de reglas, algoritmos genéticos, redes neuronales, etc.; y su aplicación, además de la de clasificación, puede extrapolarse a otras tareas vistas en grupos anteriores.
- Predictores: El objetivo consiste en generar un modelo a partir de toda la información suministrada en su proceso de creación que permita, con nuevas observaciones, predecir un valor numérico acertado. Para ello, se aplican técnicas de regresión lineal, no lineal, árboles regresores, redes neuronales, neurodifusas, etc.

El último grupo de técnicas tratan de métodos de búsqueda, dentro de una base de datos, de datos o patrones que sean muy próximos a uno de interés. Estas técnicas son muy utilizadas en la búsqueda de textos o imágenes dentro de grandes volúmenes de datos.

En puntos posteriores se pasarán a describir algunos de los algoritmos y técnicas más representativos de cada grupo. Hay que tener en cuenta, que el número existente de métodos es enorme y cada día crece más y más. Muchos de ellos, son combinaciones de diversas técnicas con nuevos ajustes que los hacen más robustos y precisos, o variantes de los ya existentes pero más especializados.

3.4.1 ALGORITMOS Y TÉCNICAS PARA EL ANÁLISIS EXPLORATORIO DE LOS DATOS (EDA), DESCRIPCIÓN DE LA INFORMACIÓN Y SUMARIZACIÓN

Las técnicas y algoritmos que se mostrarán a continuación ayudan a explorar y describir la información de ingentes cantidades de observaciones con alta dimensionalidad. En la figura siguiente se muestra una agrupación básica de algunas de ellas.

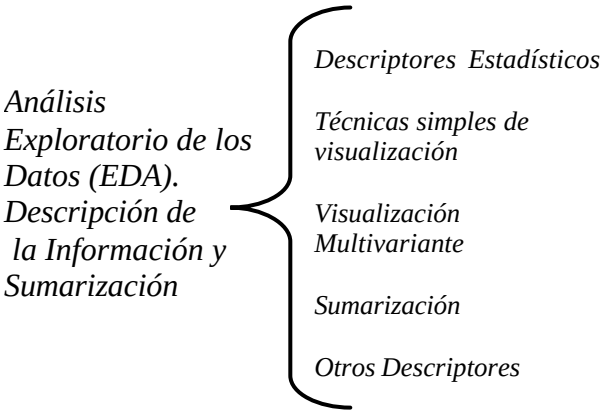


Figura 26. Algoritmos y Técnicas para el análisis exploratorio de los datos.

Generalmente, algunas de estas herramientas se utilizan en cada una de las fases de todo el proceso del *Data Mining*. Por ejemplo, cada vez que se aplica una técnica o algoritmo del DM los usaremos para:

- Comprender la estructura de los datos resultantes.
- Determinar agrupamientos en los nuevos datos.
- Detectar puntos anómalos.
- Descubrir agrupamientos.
- Analizar tendencias.
- Transformar los datos.
- Etc.

Por ello, las técnicas que se describe, a continuación son de gran utilidad.

3.4.1.1 DESCRIPTORES ESTADÍSTICOS

Los descriptores estadísticos son una de la herramientas más útiles para describir una serie de datos.

DESCRIPTORES PARA UNA VARIABLE

Como es lógico, en estadística existen multitud de técnicas y herramientas para el análisis de los datos cuya descripción se sale de este capítulo. Aún así, se recordarán algunas de las medidas más comunes utilizadas en estadística [STA01][SIE00][HAI99][PRU02]:

- **Mínimo:** El valor más pequeño.
- **Máximo:** El valor más grande.
- **Rango de la Muestra:** Corresponde al valor máximo menos el valor mínimo.
- **Media:** Corresponde al valor medio de los datos. La media es sensible a los espurios y puede ser falsificada por ellos.

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} \tag{3.1}$$

- **El percentil 25 o primer cuartil:** Valor del grupo de datos bajo el cuál está el 25% de todos los valores de la distribución. También se le denomina cuartil inferior o menor.
- **Mediana, percentil 50 o segundo cuartil:** Valor central de los datos ordenados, de forma que el 50% de los datos están por debajo del mismo. Si el número de datos es impar, corresponde con el valor central de los datos sino, se obtiene la media de los dos valores del centro.
- **El percentil 75 o tercer cuartil:** Valor del grupo de datos bajo el cuál está el 75% de todos los valores de la distribución. También se le denomina cuartil superior o mayor.
- **Rango Intercuantil:** Corresponde al tercer cuantil menos el primero.
- **Moda:** Corresponde al valor de la muestra que presenta mayor frecuencia.
- **Media absoluta de la desviación usando la Mediana**: Corresponde a otra medida de dispersión, pero donde se usa la mediana en vez de la media, ya que es más robusta frente a espurios.

$$MedianaDesv = \frac{1}{n}\sum_{i=1}^{n} \left| x_i - mediana \right| \tag{3.2}$$

- **Coeficiente de Variación**: Corresponde a otra medida de dispersión, pero donde se usa dos medidas: la desviación estándar y la media.

$$CoeficienteVariación = \frac{s}{\bar{x}} \tag{3.3}$$

- **Desviación estándar**: Corresponde con la medida de la dispersión de los datos y determina la variabilidad de los datos frente a la media. Si el valor es pequeño, los datos están próximos a la media, si es grande los datos están separados de la media.

$$s = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1}} \tag{3.4}$$

- **Varianza de la Muestra**: Otra medida similar a la varianza.

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x}) \tag{3.5}$$

- **Skewness o Sesgo**: Mide la desviación de la distribución frente a una distribución simétrica. Es una medida de simetría, de tal forma que si es positiva, la cola más larga de la distribución se encuentra en la derecha, si es negativa en la izquierda, y si es simétrica el valor de esta medida es cero.

$$Skewness = \frac{\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^3}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^3}} \tag{3.6}$$

- **Kurtosis**: es una medida estadística que describe el apuntamiento o achatamiento de una cierta distribución con respecto a una distribución normal. La kurtosis positiva indica una distribución relativamente apuntada, y la negativa indica una distribución relativamente achatada . En una distribución normal la kurtosis es igual a 3, a los valores mayores a 3 se los llama kurtosis excesiva. El caso de kurtosis excesiva indica que hay una mayor probabilidad de que los retornos observados estén más alejados de la media que en una distribución normal.

$$K = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \left[\frac{x_i - \bar{x}}{s} \right]^4 - \frac{3(n-1)^2}{(n-2)(n-3)} \tag{3.7}$$

- **Coeficiente de Asímetria**: Igual que las medidas anteriores, sirve para indicar la forma de la distribución. Si es positivo, la curva presenta asimetría hacia la derecha, si es negativo, la curva presenta asimetría hacia la izquierda; y si es cero, la distribución es simétrica.

$$CA = \frac{n}{(n-1)(n-2)} \sum_{i=1}^{n} \left[\frac{x_i - \bar{x}}{s} \right]^3 \tag{3.8}$$

- **Entropía**: Se utiliza para medir la cantidad de información existente en una muestra de variables categóricas. En este caso, H_i corresponde a la probabilidad del elemento i y q el número de diferentes valores categóricos, y N al número de patrones.

$$Entropía = \sum_{i=1}^{q} \frac{H_i}{N} \log_2 \left(\frac{H_i}{N} \right) \tag{3.9}$$

DESCRIPTORES PARA DOS VARIABLES

El análisis de correlación **muestra el grado en el que esas variables se relacionan**. La correlación entre X y Y puede calcularse sin necesidad de referirse a:

Los efectos de X sobre Y, o viceversa

Ningún efecto de una sobre la otra, sino que ellas se muevan juntas, debido a que una tercera variable influye en ambas.

El coeficiente de correlación (r) de una serie de pares de puntos ajustados sobre una línea recta, expresado en términos de las variables $x_i = X_i - \bar{X}$ y $y_i = Y_i - \bar{Y}$ es

$$r = \frac{1}{n-1} \sum x_i y_i \tag{3.10}$$

o en términos de las observaciones originales (X, Y)

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{X})}{\sqrt{\sum \left(X_i - \bar{X}\right)^2 \sum \left(Y_i - \bar{Y}\right)^2}} \tag{3.11}$$

Como el coeficiente de correlación r muestra el grado en el cual se relacionan X e Y (tiempo y demanda), si la correlación es perfecta y se ajusta a una línea recta r=1, esto indica que a una variación determinada de X (tiempo) corresponde exactamente una variación proporcional sobre Y (demanda). Si no existe correlación r=0 y si es perfecta pero inversamente relacionada se obtendrá r=-1.

Aquí surge un problema de apreciación. Los fenómenos sociales o económicos (relación tiempo-demanda) pertenecen a los llamados "sistemas ligeros", donde nunca habrá correlaciones perfectas (r=+1 o r=-1). Entonces, si el investigador de mercados encuentra un valor de, por ejemplo, r=0.7, dado que se está trabajando con sistemas reales donde únicamente se pueden pedir "r" cercanas a 1, la pregunta que surge es ¿Cuánto le sirve a un investigador el conocer ese valor de correlación para hacer sus predicciones? Es decir, si él sabe que su ajuste tiene un error de 30% ¿se queda con su ajuste de línea recta o busca un ajuste no lineal que eleve el grado de la correlación para que sus predicciones sean mejores?

Cuando existen relaciones multivariantes, se pueden establecer otros tipos de coeficientes de correlaciones, por ejemplo, el coeficiente de correlación parcial $r_{XY:Z}$ calcula el grado en el cual X y

Y se mueven juntos si *Z* permanece constante. Para ello, es necesario hacer la suposición de que la población de la muestra de las distribuciones de *X*, *Y* y *Z* son normales y multivariadas.

Al calcular su estimador $r_{YX:Z}$ surge un problema. Puesto que *Z* es una variable aleatoria, simplemente no es posible fijar un solo valor de Z_0. Así, a menos que la muestra sea extremadamente grande, es poco probable que más de una sola combinación *Y*, *X*, Z_0 implicando Z_0 sea observada. La alternativa es calcular $r_{YX.Z}$ como la correlación de *Y* y *X* después de que la influencia de *Z* se ha eliminado de cada una de ellas.

La correlación parcial resultante $r_{YX.Z}$ después de considerables manipulaciones, puede expresarse como la correlación simple de *Y* y *X* ajustada por la aplicación de dos correlaciones simples, implicando *Z* (llamadas r_{xz} y r_{yz}) como sigue:

$$r_{YX.Z} = \frac{r_{YX} - r_{YZ}\, r_{XZ}}{\sqrt{1-r_{XZ}^2}\,\sqrt{1-r_{YZ}^2}} \tag{3.12}$$

donde:

$$r^2 = \frac{\sum (\hat{Y}_i - \overline{Y})^2}{\sum (Y_i - \overline{Y})^2} = \text{cada una respecto de X y Z.} \tag{3.13}$$

Esta fórmula muestra que no necesita haber una correspondencia cercana entre los coeficientes que correlación parcial y simple; sin embargo, en el caso especial de que tanto *X* y *Y* no se relacionen por completo con *Z*

$$r_{XZ} = r_{YX} \tag{3.14}$$

y como se supondría, los coeficientes de correlación parcial simple son los mismos.

3.4.1.2 TÉCNICAS SIMPLES DE VISUALIZACIÓN

Las técnicas más comunes de visualización de series de datos [HAN01] son las siguientes:

HISTOGRAMAS

Muestran cómo está distribuidos los los datos.

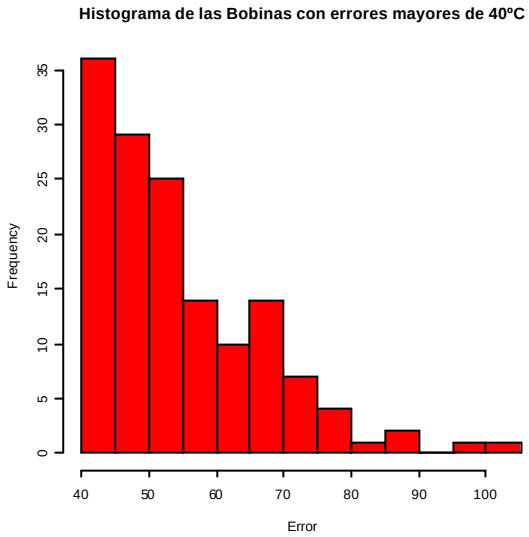


Figura 27. Histograma donde se muestra la distribución de las bobinas según el error.

DIAGRAMAS BOX-PLOT

Muestran en un solo gráfico, la distribución, el rango intercuartil, la media, los puntos extremos, etc.

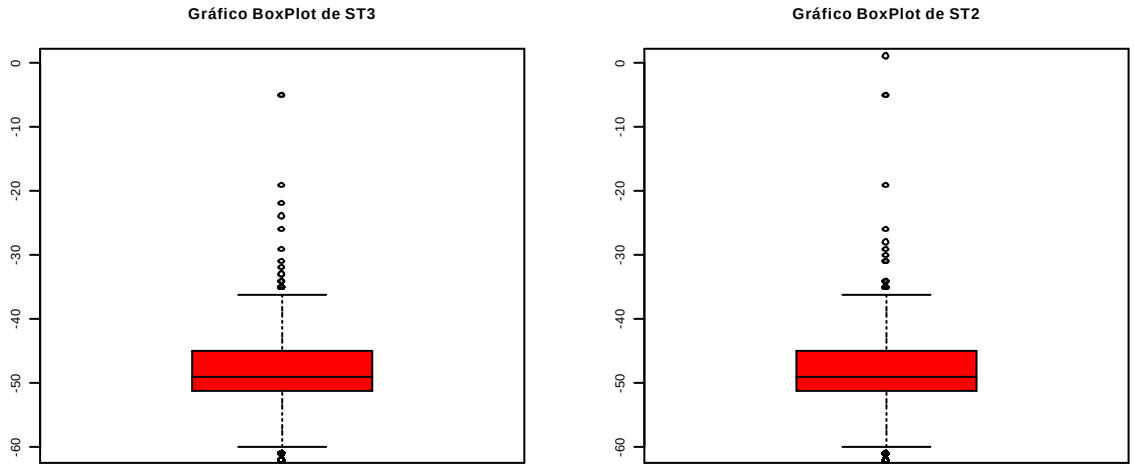


Figura 28. Dos gráficos de caja.

La caja central del gráfico está formada por tres líneas horizontales. La superior representa el tercer cuartil, la central la media y la inferior el primer cuartil.

Las líneas horizontales fuera de la caja, están situadas a 1,5 veces la distancia intercuartil.

Los puntos que están más alejados de esa distancia, se representan con círculos.

Estos gráficos ayudan a comprender cómo están distribuidos los datos de una variable pudiéndose contrastar con varias variables.

LOS SCATTERPLOTS

Permiten visualizar la relación entre dos o más variables.

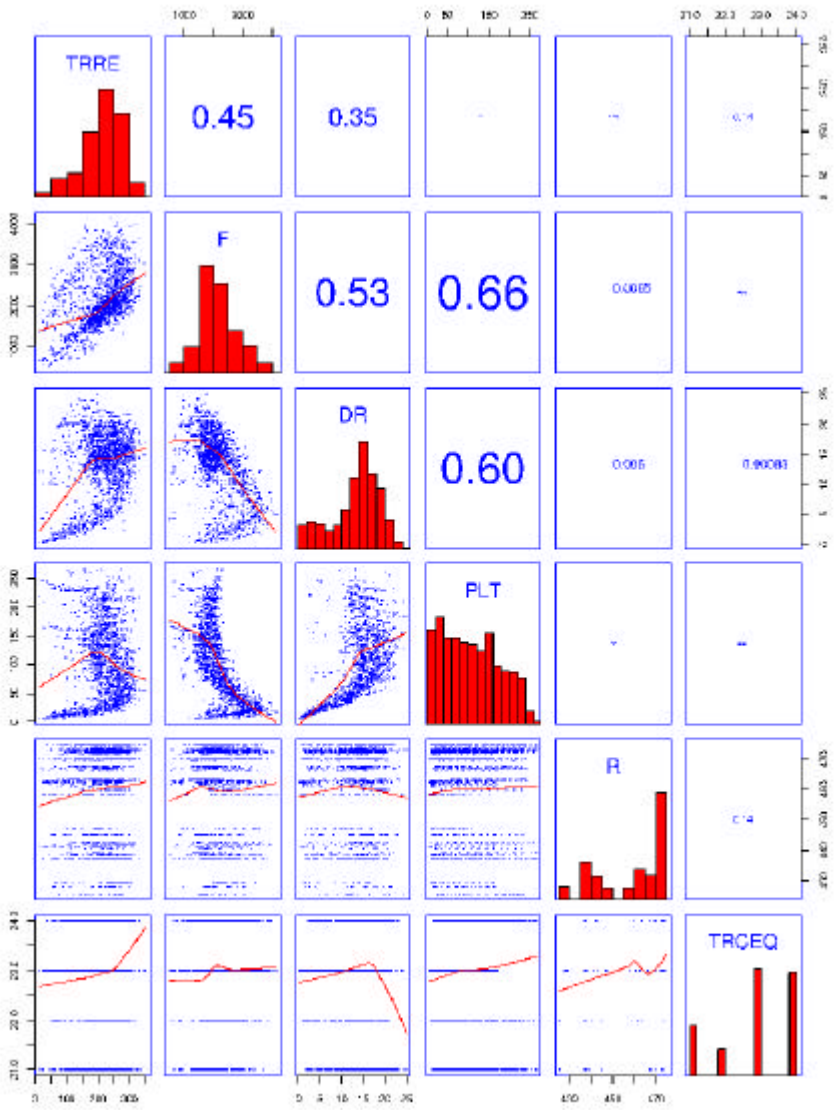


Figura 29. Diagrama scatterplot avanzado que nos representa la relación entre las variables muestreadas. Por un lado se muestra el histograma de cada una de ellas (gráficas de la diagonal), la distribución entre parejas de variables (triangulo inferior) y el coeficiente de correlación de cada pareja de variables (triangulo superior).

En la Figura 29 se muestra un gráfico avanzado de scatterplot donde se muestra a la vez:

- El histograma y nombre de cada variable (diagonal).
- La distribución de los puntos entre dos variables y la curva regresada (matriz triangular inferior).
- El coeficiente de correlación entre parejas de variables (matriz triangular superior).

Este gráfico permite rápidamente detectar variables muy correladas y el tipo de distribución existente entre ellas. Es muy útil para tareas de selección de variables y análisis de correlaciones.

OTRAS VARIANTES

Como es lógico, los tipos y combinaciones pueden ser enormes.

En la figura siguiente se muestran algunos ejemplos de gráficos en 2D, 3D, contorno, etc.

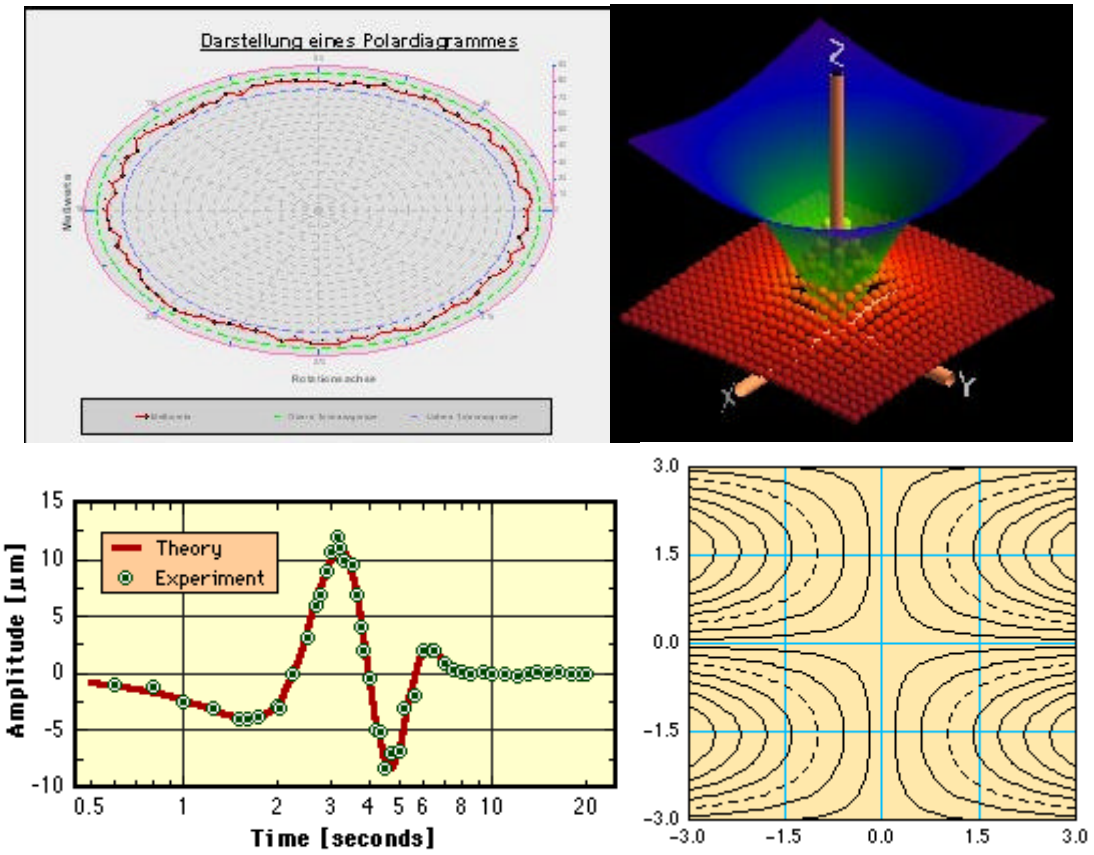


Figura 30. Varios tipos de Gráficos: diagrama polar (superior izquierda), 3D (superior derecha), diagrama temporal X-Y (inferior izquierda) y diagrama de contornos (inferior derecha).

3.4.1.3 TÉCNICAS DE VISUALIZACIÓN MULTIVARIANTE

Las técnicas de visualización anteriores, permiten ver la distribución o características de una muestra o cómo máximo la relación de dos variables, pero cuando pasamos de las tres dimensiones (variables o atributos) y tenemos observaciones con cuatro, cinco, seis o muchas más dimensiones, necesitamos nuevas herramientas que nos permitan observar su estructura. La posibilidad de visualizar la estructura de datos con alta dimensionalidad es una de las características buscadas en las **técnicas de visualización multivariante [FAY02]** tan necesarias y utilizadas en el proceso del *data mining* [REN02].

Efectivamente, esto es debido a que con las técnicas clásicas solo podemos visualizar puntos tridimensionales en un dibujo bidimensional mediante perspectivas, y se nos queda corto cuando tenemos bases de datos con gran cantidad de atributos. Para ello, existen diferentes técnicas y herramientas que pueden ayudar en el análisis de datos con elevado número de dimensiones y que se describen las más significativas.

GRÁFICO DE COORDENADAS PARALELAS

Es uno de los más fáciles de implementar y utilizar [INS87]. Consiste en disponer tantos ejes paralelos como variables diferentes tenemos y representar cada observación como una línea que uno los valores de cada una de los atributos en sus respectivos ejes.

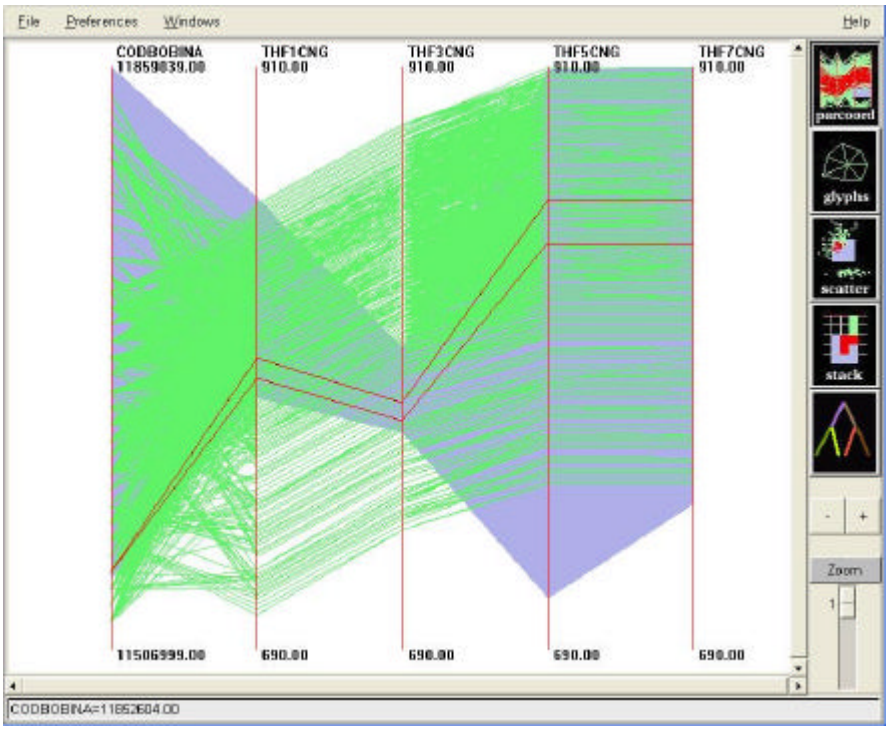


Figura 31. Diagrama de coordenadas paralelas multitud de observaciones.

La ventaja de esta forma de visualización, es que se pueden colorear las diferentes observaciones y detectar aquellas que son inusuales, descubrir grupos, patrones, relación entre variables, etc.

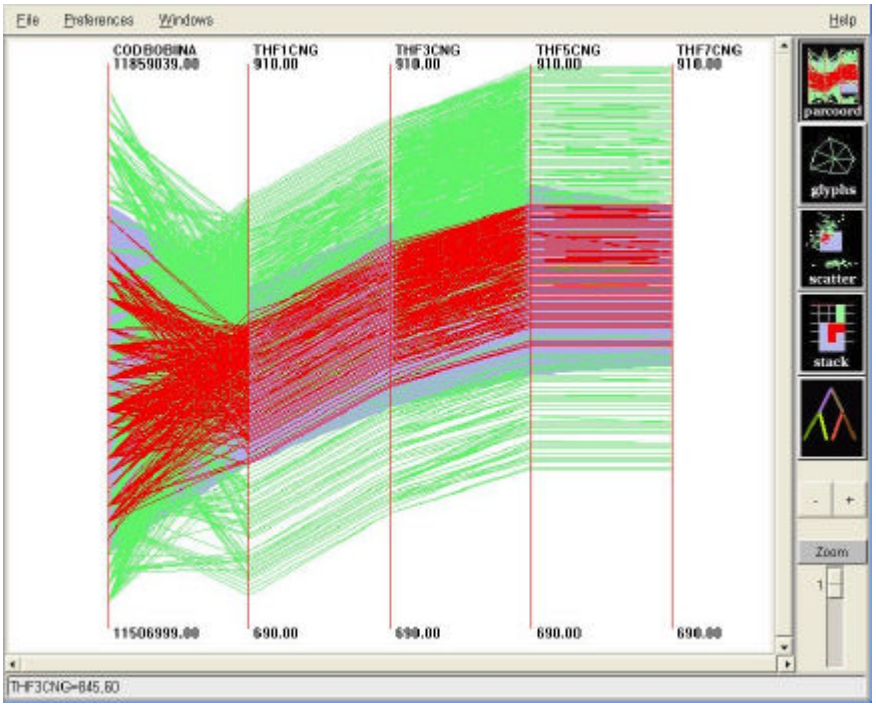


Figura 32. Gráfica donde se observa claramente la dependencia lineal de las cuatro variables de la derecha.

Existen variantes donde los ejes son radios de una circunferencia: coordenadas paralelas circulares, son ejes que se cortan en un punto, etc.

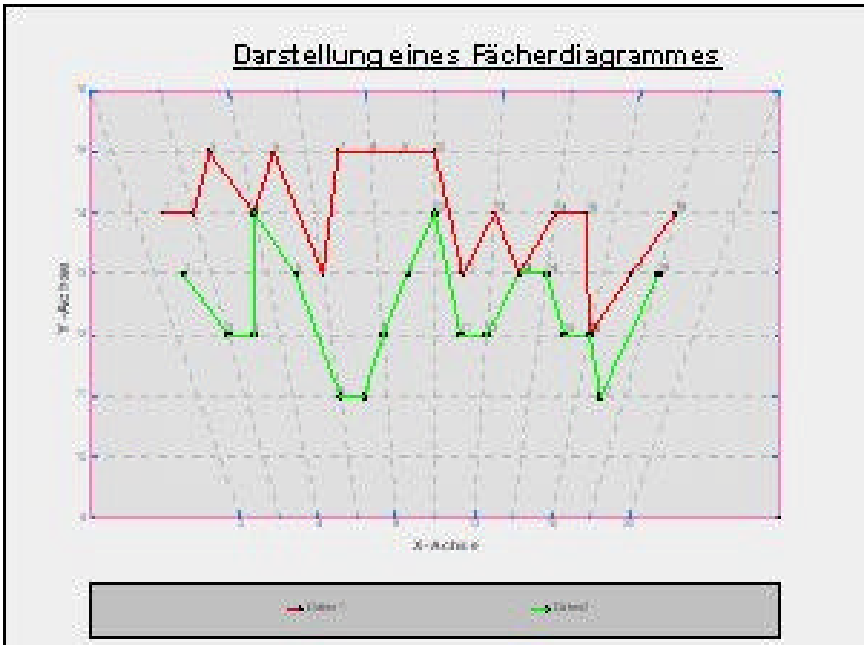


Figura 33. Otra variante de las coordenadas paralelas.

CARAS DE CHERNOFF

Las técnicas de iconografías se basan en la aptitud natural del cerebro humano para clasificar objetos parecidos. De esta forma, las "Caras de Chernoff" muestran una cara para cada observación, de tal forma que cada valor de un atributo modifica una de las características faciales (boca más grande, ojos más separados, nariz más grande, etc).

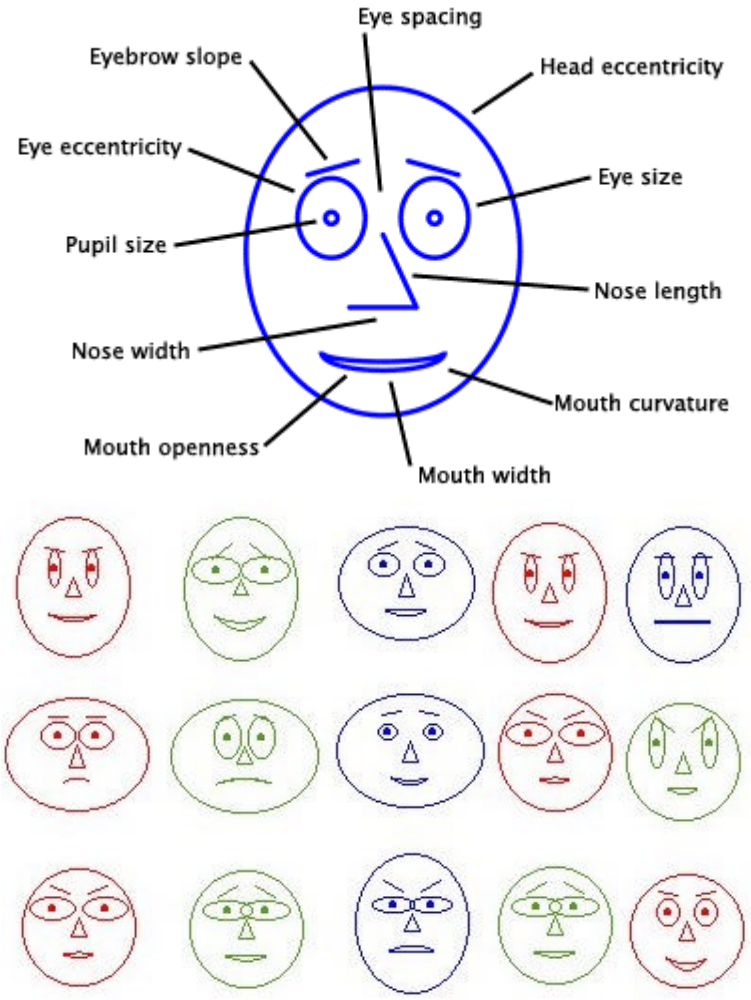


Figura 34. Caras de Chernoff.

Lamentablemente, aunque puede servir para clasificar diferentes observaciones, el cambio de un rasgo u otro no afectan de igual forma a la vista del clasificador humano, por lo que, pueden existir atributos que influyan más en el cambio de un cara que otros. Por ejemplo, el grosor de una boca puede influir menos que el estiramiento de los ojos ya que puede inducir a clasificar como de "rasgos orientales" a aquellas caras que tengan los ojos achinados.

ICONOS DE ESTRELLAS

Igual que con las caras de Chernoff, se representa cada observación como una estrella compuesta por *n* radios equidistantes, donde *n* es el número de atributos y cada uno de ellos con la longitud del valor de la variable correspondiente.

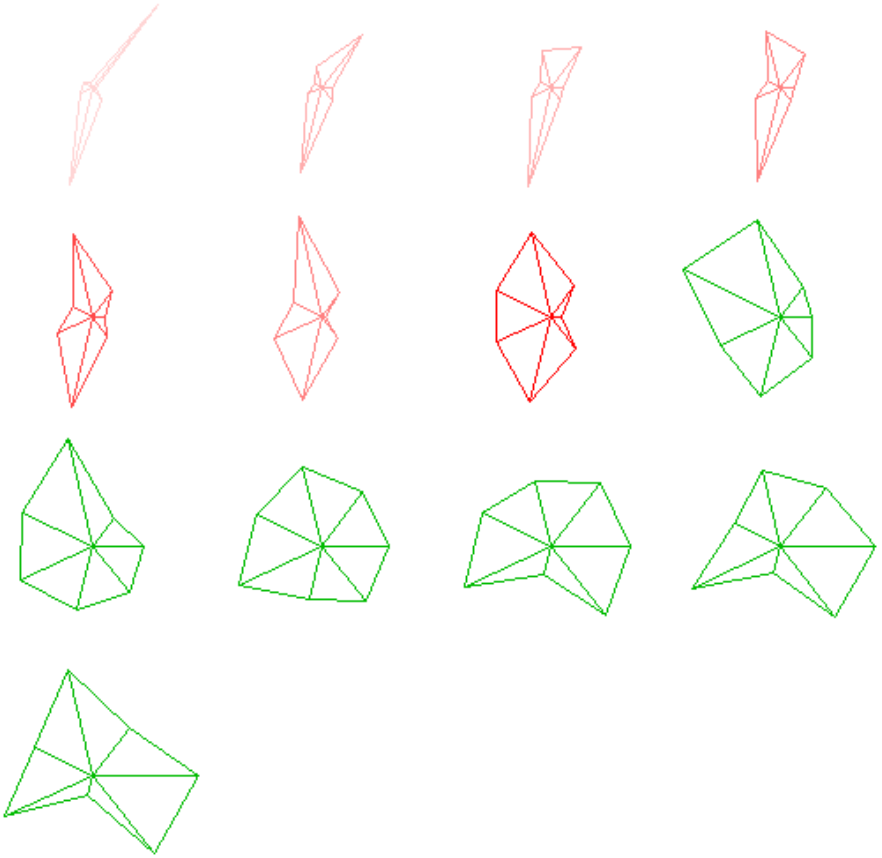


Figura 35. Iconos de estrellas de varias observaciones.

Este método es más aséptico que el anterior, aunque la clasificación se complica a medida que aumentan el número de observaciones.

OTROS MÉTODOS BASADOS EN ICONOS

Como se puede deducir, existen gran cantidad de métodos basados en iconos donde se cambian las dimensiones, formas, colores, etc.; de esferas, cilindros y otras formas geométricas. Casi cada software de DM tiene las propias y cada día aparecen nuevas.

Al final todas ellas necesitan de la red neuronal del cerebro humano para poder clasificarlas.

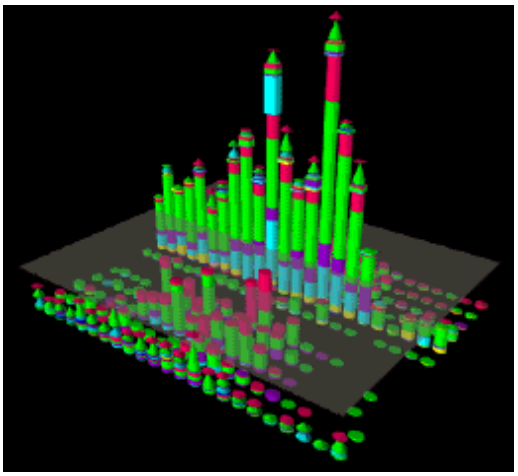


Figura 36. Ejemplo de un software de visualización multiariante.

TÉCNICAS DIMENSIONAL STACKING O REPRESENTACIÓN MULTIDIMENSIONAL PLANA

Otra técnica clásica de visualización multivariante, consiste en dividir una rejilla con los valores de dos atributos, y dentro de cada casilla, volver a dividir con los valores de otros dos atributos y así sucesivamente [MIC98]. De esta forma, se representan con puntos o cuadros rellenos las observaciones correspondientes.

La ventaja de las herramientas informáticas, es que permiten cambiar la posición de los atributos y así poder estudiar los grupos que se obtienen desde diferentes posiciones.

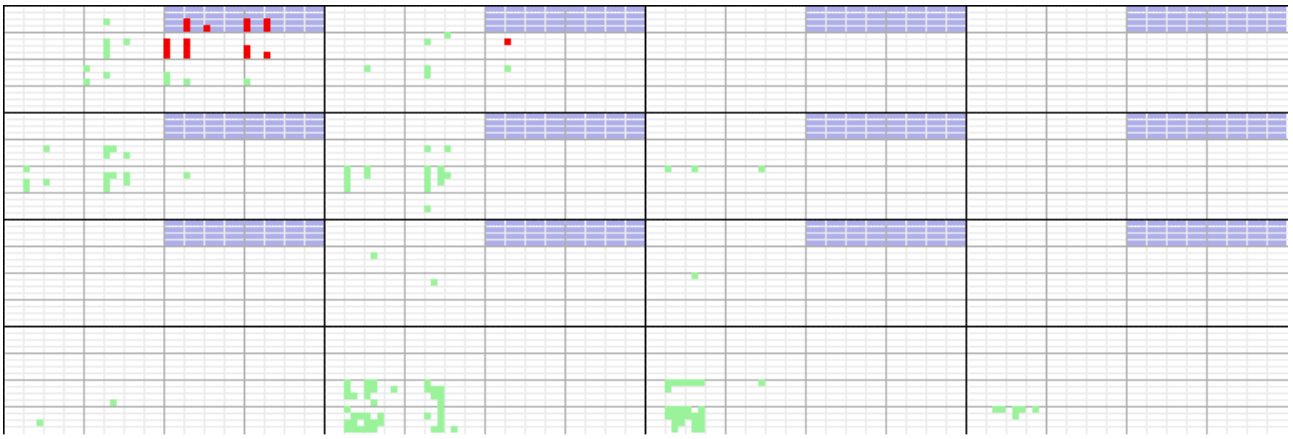


Figura 37. Diagrama plano multidimensional.

DENDOGRAMAS

Son diagramas tipo árbol que se desarrollan basándose en cómo de cercanos o lejanos están los elementos entre si. De esta forma, a partir de una medida de distancias entre las observaciones se genera un árbol que describe los diferentes agrupamientos y el grado de alejamiento o acercamiento que existe entre ellos.

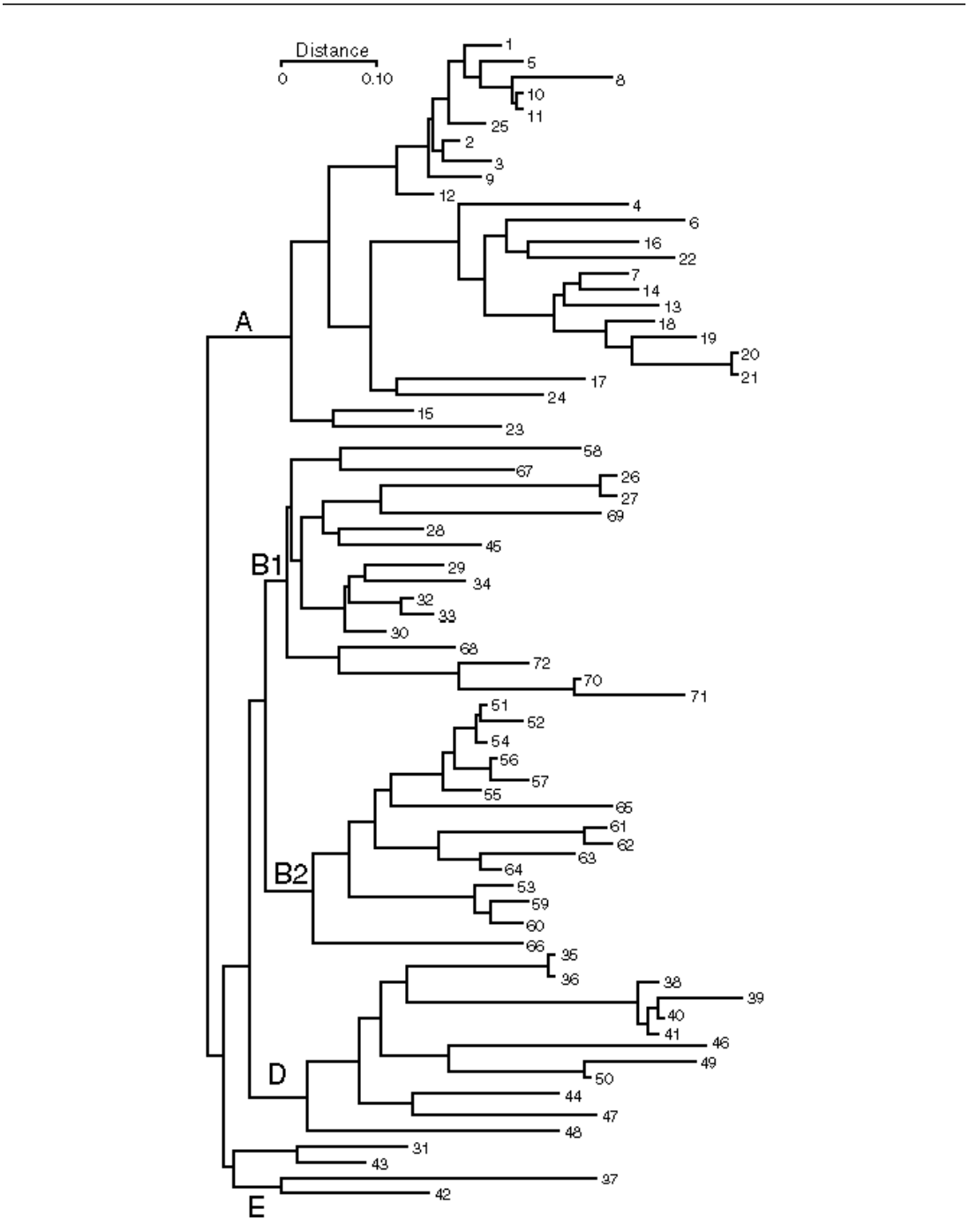


Figura 38. Dendograma de diferentes observaciones.

3.4.1.4 TÉCNICAS DE PROYECCIÓN

En muchas ocasiones es interesante visualizar de algún modo los datos para intentar obtener relaciones entre los mismos de modo natural. Cuando los patrones tienen dimensiones superiores a tres, su visualización, con diagramas clásicos de dos o tres dimensiones, resultan ineficaces. Es en estos casos cuando se recurren a las técnicas de proyección y reducción de dimensión.

Este tipo de proyección puede además suponer una reducción dimensional, si es que se consigue mantener la mayor parte de la información en los puntos proyectados.

Fundamentalmente, existen dos causas para utilizar estos métodos:

- **La reducción exclusivamente con el objetivo de visualizar**. En este caso la dimensión resultante deberá ser dos o tres. Se buscará representar en un plano o 3D los patrones "para ver" o "intuir" la disposición de los mismos o al menos tener una "representación" aproximada de su disposición siempre lo más fiel posible a la estructura real de los datos.
- **Reducir la dimensión de los datos** para mejorar el proceso de análisis, reducir la cantidad de información a manejar, determinar las variables sobrantes, etc; eliminando o transformando determinadas variables pero siempre **tratando de no "perder" información "útil".**

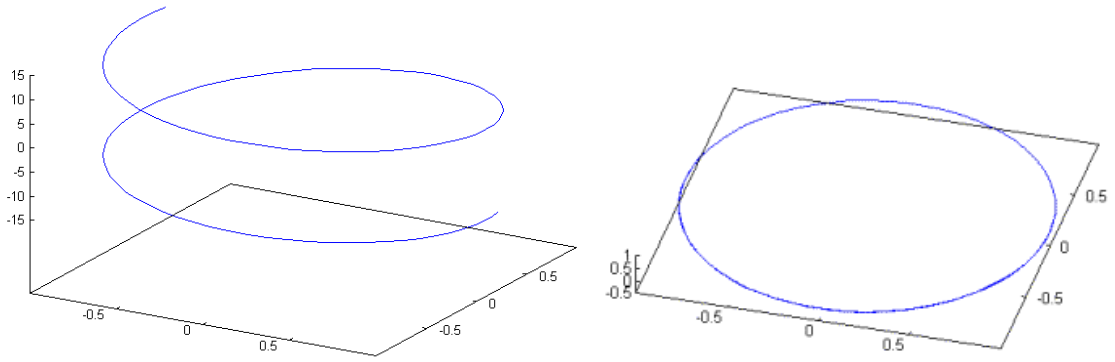


Figura 39. Ejemplo de un caso donde se pierde la estructura intrínseca en la proyección resultante.

En ambos casos se busca la "fidelidad" de la proyección, es decir, se trata de reducir dimensionalidad en los patrones sin perder la estructura "intrínseca" de los mismos (ver Figura 39). Es aquí cuando entran de lleno las técnicas que se describen a continuación.

Podemos comprender la necesidad de proyectar los datos para observar las estructura de los mismos, pero la primera pregunta que surgirá será la siguiente: ¿Cual es el espacio más pequeño sobre el que puedo proyectar?

Una de las técnicas que nos permite determinar la dimensión intrínseca de los datos es la *dimensión fractal*.

LA DIMENSIÓN FRACTAL

Basada en la teoría de los fractales e íntimamente ligada a la geometría diferencial, intenta aprovechar la idea de subvariedades de un espacio algebraico. Este algoritmo se basa en la **autosemejanza**, que es una propiedad que indica que **dos vistas en dos escalas diferentes del mismo objeto son semejantes**. Con ella podemos encontrar la dimensión fractal basada en el estudio de la variación de una propiedad medible de los datos como una función del cambio de escala.

La idea básica es muy simple. A medida que el número de dimensiones del espacio de las soluciones crece, aumenta la complejidad de la búsqueda de una posible solución para una predicción, por lo que ésta llega a hacerse imposible. Sin embargo, supongamos una barra en un espacio tridimensional. Aunque las coordenadas de la barra vengan expresadas en tres dimensiones, en realidad una sola es suficiente para representarla (ya que es una línea recta). Si conseguimos conocer la dimensión intrínseca de los datos, podremos disminuir los datos hasta esa dimensión sin pérdida alguna.

Supongamos que dividimos un espacio n-dimensional en zonas mediante una cuadrícula n-dimensional de longitud r (a cada hipercubo de esa cuadrícula le llamaremos ladrillo n-dimensional o hipercubo de dimensión n y lado r). Si $N(r)$ es el número de estos ladrillos o zonas del espacio con, al menos un punto en su interior, podemos obtener la dimensión fractal mediante la siguiente ecuación:

$$d_q = \frac{1}{q-1} \lim_{r \to 0} \frac{\log \sum_{i=1}^{N(r)} p_i(r)^q}{\log(r)} \qquad (3.15)$$

donde $p_i(r)$ es la probabilidad de la cuadrícula i (número de puntos dentro de la cuadrícula dividido por M), r es el lado del hipercubo de dimensión n y q es un coeficiente que generalmente es 0.

El algoritmo **consistirá en obtener parejas de datos formados por (el logaritmo del número de hipercubos que contienen puntos en su interior y el logaritmo del radio de esos hipercubos) representarlos en una gráfica y obtener la pendiente de la línea de regresión de esos puntos**. La pendiente de esa recta será la dimensión buscada.

Como ejemplo, sea una curva como la de la Figura 40. Su dimensión intrínseca será 1 (ya que es una línea), aunque esté en un espacio de dos dimensiones. Si se divide el espacio donde están los puntos (cuadrado grande) a la mitad, se obtienen 4 secciones, de las cuales sólo 3 están ocupadas por datos. Si se divide el espacio según una progresión $(1/2)^n$, y se cuentan las zonas que están realmente ocupadas, se obtendrán 3, 8, 16 y 30 cuadros con puntos en su interior. Se puede seguir dividiendo hasta que los cuadros llenos tengan uno o dos puntos como máximo.

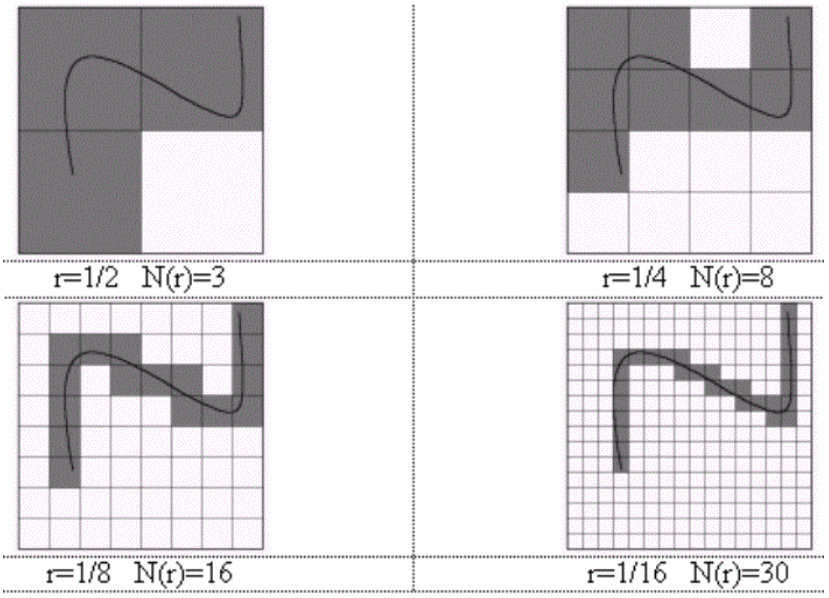


Figura 40. División del espacio bidimensional de los datos en cuadrado de lado r.

Si ahora se representan los pares de puntos (log (r), log(N(r))) en una gráfica como la de la Figura 41 y se obtiene la recta mediante un algoritmo de regresión lineal. Se obtendrá la dimensión intrínseca de los datos (determinada de la ecuación 6.16 cuando q=0) calculando la pendiente de la misma. En este ejemplo, la pendiente de la recta regresada es de 1.06, es decir, el algoritmo de la dimensión fractal nos indica que los puntos forman un objeto de un dimensión.

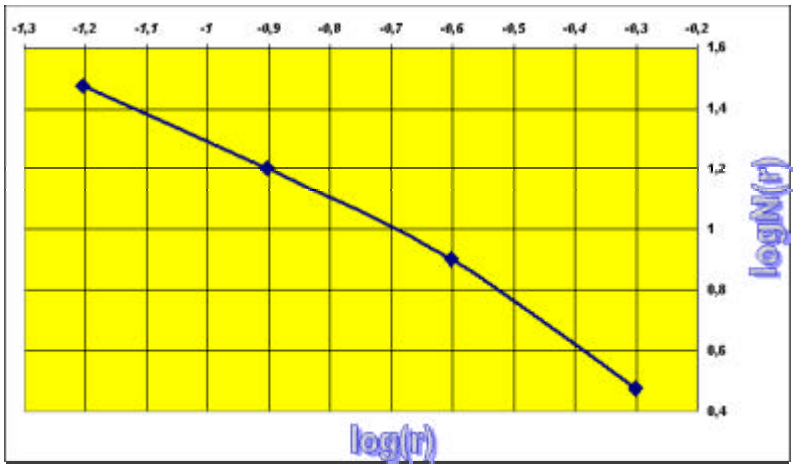


Figura 41. Curva obtenida representando los puntos obtenidos (log(r), log(N(r)) de donde se obtiene la recta regresada cuya pendiente es 1.06 la dimensión intrínseca de los datos buscada.

Este algoritmo permite fácilmente y con un coste de computación relativamente pequeño, determinar la estructural intrínseca de los puntos a analizar.

PROYECTOR LINEAL BASADO EN ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

El propósito del análisis de componentes principales (PCA) es transformar una matriz *X* de *p* variables en otra matriz *Y* de variables virtuales incorreladas ordenadas de mayor a menor varianza.

Sean *n* patrones de dimensión *p*, es decir:

$$X^* = \left[x_{ij}\right]_{i=1,p}^{j=1,n} \tag{3.16}$$

Podemos también representar la información "centrada" sobre su media, y obtener su media y varianza:

$$X = \left[x_{ij} - \bar{x}_i\right]_{i=1,p}^{j=1,n} \tag{3.17}$$

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \tag{3.18}$$

$$s = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}\left(x_{ij} - \bar{x}_i\right)^2} \tag{3.19}$$

$$Z = \left[\frac{x_{ij} - \bar{x}_i}{s_i}\right]_{i=1,p}^{j=1,n} \tag{3.20}$$

Se define la matriz de covarianza, como:

$$S = \left(X \;\; X^T\right)/(n-1) = \left(S_{ij}\right)_{i=1,p}^{j=1,p} \tag{3.21}$$

Cumpliendo la matriz *S* las siguientes propiedades:

- S es una matriz simétrica.
- S es definida NO negativa (autovalores no negativos).
- La traza de S es igual a la Inercia de los puntos respecto al origen.

El primer eje factorial U_1 relativo a los puntos en estudio lo es cuando este eje maximiza la inercia explicada.

$$F_{kj} = Z_j^t U_k \tag{3.22}$$

$$F_k = \left\{F_{kj}\right\}^{j=1,n} = Z\vec{U}_k \tag{3.23}$$

Así la selección de ejes factoriales se lleva a cabo en orden de "relevancia" en términos de aportación de información (autovalores), así cada eje que se determina debe aportar cada vez menor información.

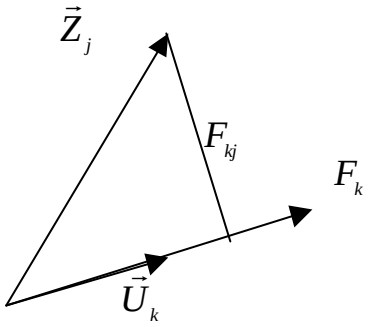


Figura 42. Ejes obtenidos de las matriz S.

En la Figura 43 podemos observar, que el primer autovalor de la nube de puntos corresponde **al eje que maximiza la inercia de los puntos**. Los siguientes autovalores corresponderán a ejes que irán aportando cada vez menor información.

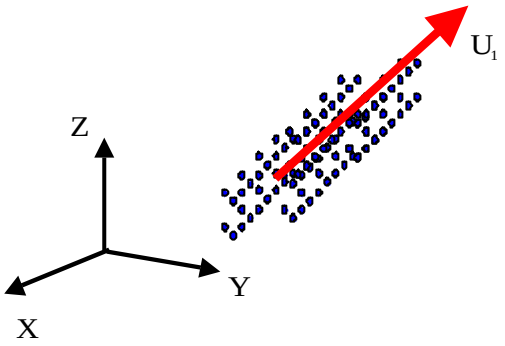


Figura 43. Primer Autovalor obtenido de la nube de puntos.

Lo que se busca, por lo tanto, es seleccionar los suficientes autovalores de forma que la pérdida de la información resultante no exceda del 10%.

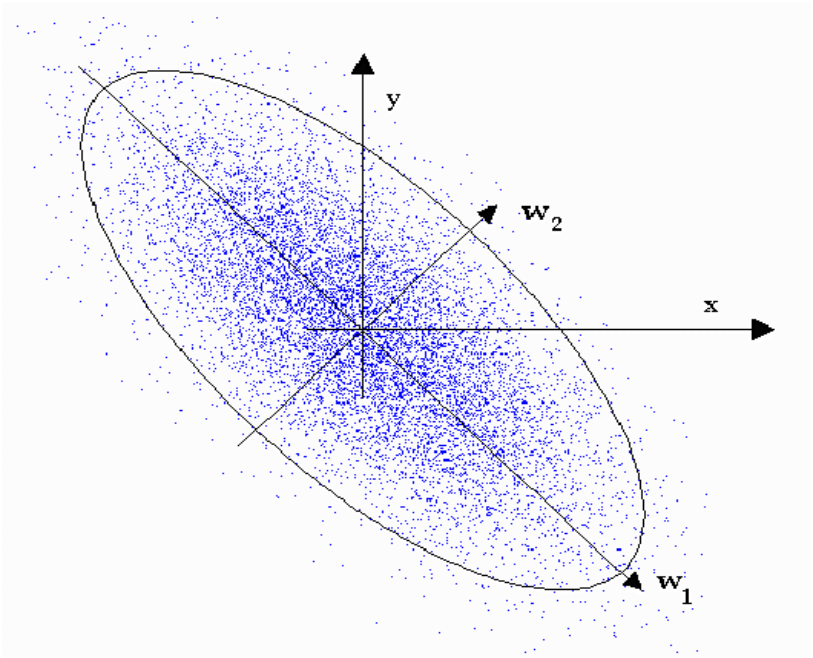


Figura 44. Obtención de los Autovalores Principales de la estructura de una nube de puntos.

Este método permite reducir considerablemente los datos, sustituyendo las variables observadas por un pequeño grupo de variables derivadas, pero tiene los siguientes inconvenientes:

- No hace ninguna mención específica a los errores de medida de datos, por lo que debería ser idealmente restringida a situaciones en las cuales se presume que esos errores son pequeños.
- El análisis de componentes principales no es invariante a cambios de escala en las variables.
- La proyección sobre los ejes es lineal, por lo que no se pueden inferir otras relaciones más complejas. Es un **método de proyección lineal, no es muy conveniente para estructuras de datos no lineales.**
- Las variables no representan ninguna cualidad física concreta, por lo que su interpretación es compleja.

PROYECTOR LINEAL: PROYECCIÓN PURSUIT

La técnica de *Proyección Pursuit (PP)* [JON87] es una técnica no supervisada que busca proyecciones lineales de baja densidad interesantes dentro de datos de alta densidad mediante la optimización de una función objetivo llamada índice de proyección (*Projection Index PI*).

La notación de *interesante* varía con la aplicación. El objetivo del *data mining* (búsqueda de grupos, estimación de densidades, etc.) debe ser trasladado a un índice numérico de los múltiples propuestos. Dependiendo de las propiedades estadísticas usadas, pueden ser más o menos sensibles en la detección de espurios, etc.

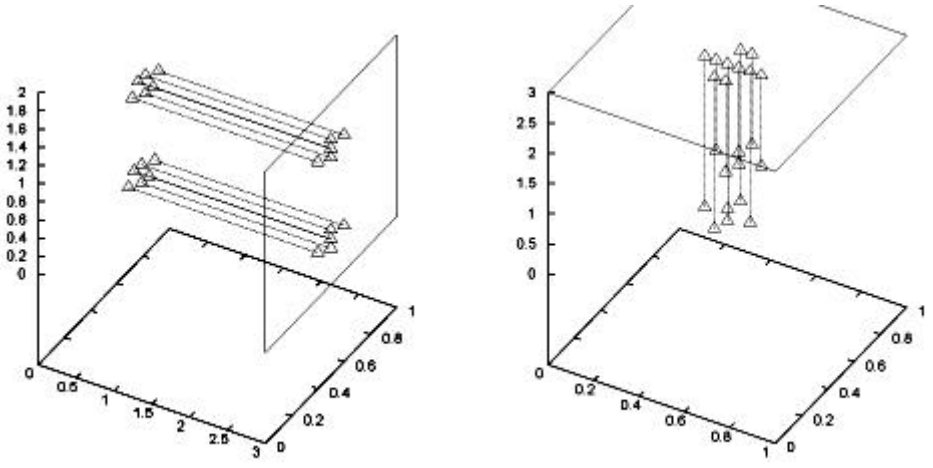


Figura 45. Dos proyecciones en 2D de puntos en 3D. En el primero (izquierda) se pueden apreciar mejor los dos grupos de puntos, mientras que en el segundo (derecha) los puntos proyectados están todos superpuestos.

Por ejemplo, supongamos que deseamos detectar agrupamientos. Un estimador estadístico muy utilizado para clusterizado es la media de las distancias de los vecinos más próximos (*Mean Nearest Neighbour Distance MNND*). De esta forma, el proceso consistirá en buscar la proyección que encuentre el menor valor final de este estimador, ya que cuánto más pequeño más concentrados serán los grupos y más fácil será detectarlos.

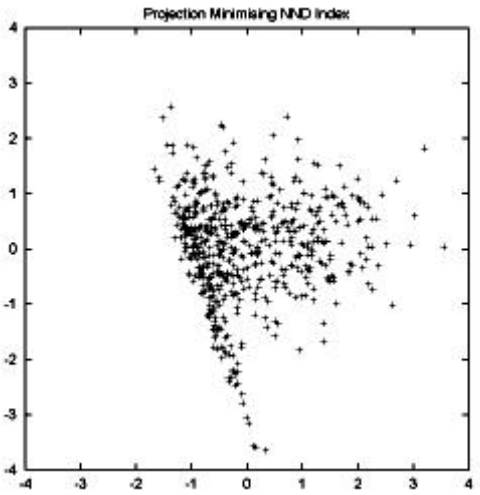


Figura 46. Detección de grupos minimizando el MNND.

Otra posibilidad estriba en detectar los espurios buscando la proyección que de un máximo MNND.

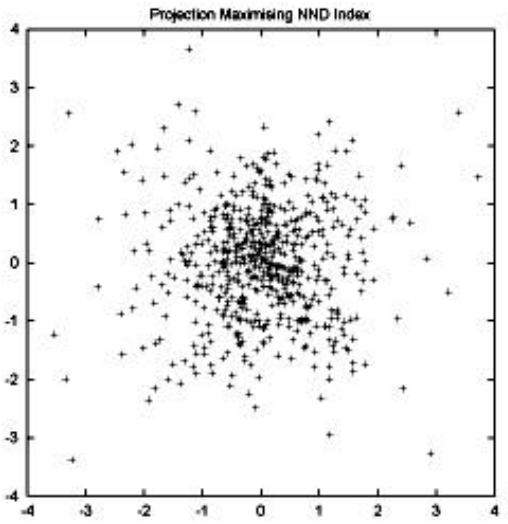


Figure 5: Maximised MNND Projection of Census Data

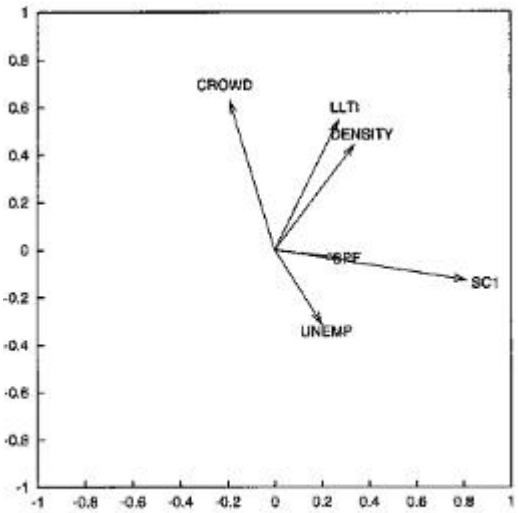


Figure 6: Maximised MNND Projection of Census Data - Interpretation Plot

Figura 47. Ejemplo de una proyección y los ejes proyectados [AGO99]. Detección de espurios.

Lógicamente, **esta técnica tiene sus limitaciones y deben estudiarse los resultados con mucha precaución.**

PROYECTOR NO LINEAL: PROYECCIÓN SAMMON

Este método [SAM69] pretende proyectar el espacio de patrones en un espacio R^2 (plano), tratando de mantener las distancias relativas entre patrones existente en el espacio original. Se pretende dibujar en un plano, los puntos relativos a cada patrón intentando mantener la distancia entre ellos. Esto permite poder "intuir" la estructura de los datos.

Sean N vectores en un espacio L-dimensional, (X_i), $i=1,...,N$. Tomemos N vectores en un espacio M-dimensional $(M=2$ ó $3)$ (Y_i) $i=1,...,N$. y definimos una distancia en cada espacio $d_{ij}^*[^oX_i, X_j]$ y $d_{ij}[^oY_i, Y_j]$.

$$Y_1 = \begin{bmatrix} y_{11} \\ \cdot \\ \cdot \\ \cdot \\ y_{1M} \end{bmatrix} \quad Y_2 = \begin{bmatrix} y_{21} \\ \cdot \\ \cdot \\ \cdot \\ y_{2M} \end{bmatrix} \quad Y_N = \begin{bmatrix} y_{N1} \\ \cdot \\ \cdot \\ \cdot \\ y_{NM} \end{bmatrix} \tag{3.24}$$

Podemos calcular las distancias d_{ij} entre los puntos del espacio de dimensión d, con el fin de definir un error E, que representa la calidad con la que los N puntos del espacio d-dimensional representan a los N puntos iniciales del espacio de dimensión B. Se define esta función de error como:

$$E = \frac{1}{\sum_{i<j} [d_{ij}^*]} \sum_{i<j}^{N} \frac{[d_{ij}^* - d_{ij}]^2}{d_{ij}^*} \tag{3.25}$$

Y se buscan los valores de Y_k para minimizar ese error (E).

Generalizando el problema:

$$E_s = \frac{1}{g_p} \sum_{i<j} d(x_i, x_j)^p \cdot [d(x_i, x_j) - d(y_i, y_j)]^2 \tag{3.26}$$

$$g_p = \sum_{i>j} d(x_i, x_j)^{p+2} \tag{3.27}$$

Dependiendo del valor de p distinguiremos :

- *Método de Proyección Local* ($p < 0$). Tiende a proyectar puntos que están próximos de una forma más precisa que los que están algo más alejados entre sí, y donde el caso más habitual es el *Método de Sammon* ($p = -1$).
- *Método de Proyección Neutra* ($p = 0$). Tanto los puntos próximos como los alejados entre sí, se proyectan con la misma precisión

- *Método de Proyección Global* (p > 0). Proyecta mejor los puntos que están más alejados entre sí.

Como ventajas podemos encontrar:

- La función no depende de ningún parámetro de control.
- Es altamente eficaz en identificar estructuras complejas de datos no lineales, incluso de tipo hiperelipsoidal.
- La proyección resultante es fácilmente evaluable por el investigador.
- La clasificación humana puede evitar puntos dudosos por su carácter más global.
- El algoritmo es simple y eficiente.

La mayor desventaja que tiene este método, es su alto coste computacional.

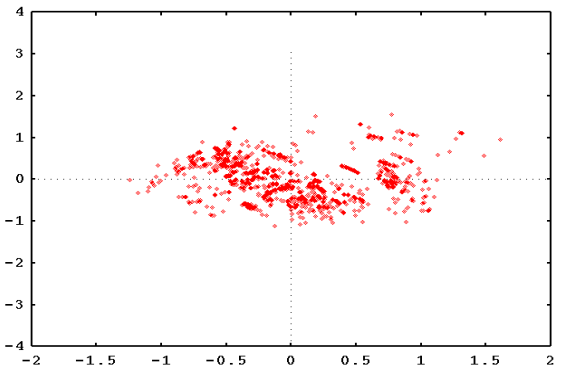


Figura 48. Ejemplo de una proyección Sammon de patrones con 96 componentes pasados a un espacio R^2.

A efectos prácticos **debe vigilarse el error final, ya que si este es elevado, el resultado puede ser poco fiable.**

PROYECTOR NO LINEAL BASADO EN COMPONENTES PRINCIPALES (NLPCA)

Este método trata de generalizar la idea base de los PCA, para permitir la aproximación NO LINEAL. Para ello, el *Análisis No Lineal de Componentes Principales* (NLPCA) estima una curva o superficie que pasa por el medio de las observaciones utilizando el criterio de mínimos cuadrados:

$$\min_{f,s_f} \sum_{i=1}^{n} \left\| x_i - f(s_f(x_i)) \right\|^2 \tag{3.28}$$

La composición de las funciones $f(s_f(x_i))$ representa las coordenadas p-dimensionales de la proyección x_i en una curva o superficie f.

La función $s_f : \Re^p \rightarrow \Re^r$ se denomina *índice de proyección* y da las coordenadas r-dimensionales de la proyección de x_i en f. La función $f : \Re^r \rightarrow \Re^p$ es una curva o superficie r-dimensional en $\Re^p$.

La modelización no paramétrica de las funciones s_f y f puede realizarse mediante una red neuronal. Su topología es una red de tres capas como la de la Figura 49.

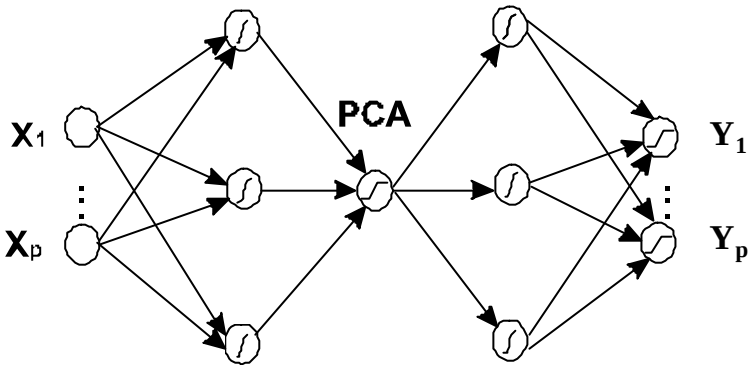


Figura 49. Estructura de la red neuronal para realizar el NLPCA.

Para entrenar la red neuronal, introducimos los vectores de puntos como patrones de entrenamiento a la entrada y la salida. Es decir, le indicamos a la red que ajuste los pesos y *bias* de sus neuronas para que la entrada y la salida sea la misma. **Si el error final obtenido en las fases de entrenamiento y validación de la misma es pequeño, habremos obtenido en la capa intermedia una ecuación no lineal donde se obtienen los ejes buscados.**

PROYECTOR NO LINEAL: PROYECCIÓN DE ANDREWS

Andrews (1972) describe un método muy sencillo para obtener gráficas de funciones que también puede aplicarse para obtener una representación visual de datos multivariantes. Cada punto $X=(x_1,x_2,...,x_p)$, de dimensión p, define una función:

$$f_X(t) = \frac{x_1}{\sqrt{2}} + x_2 sent + x_3 \cos t + x_4 sen2t + x_5 \cos 2t + ... \qquad (3.29)$$

Esta función se dibuja en el intervalo $-p \le t \le p$ de modo que un grupo de puntos p-dimensionales aparecerán como un grupo de líneas sobre el dibujo.

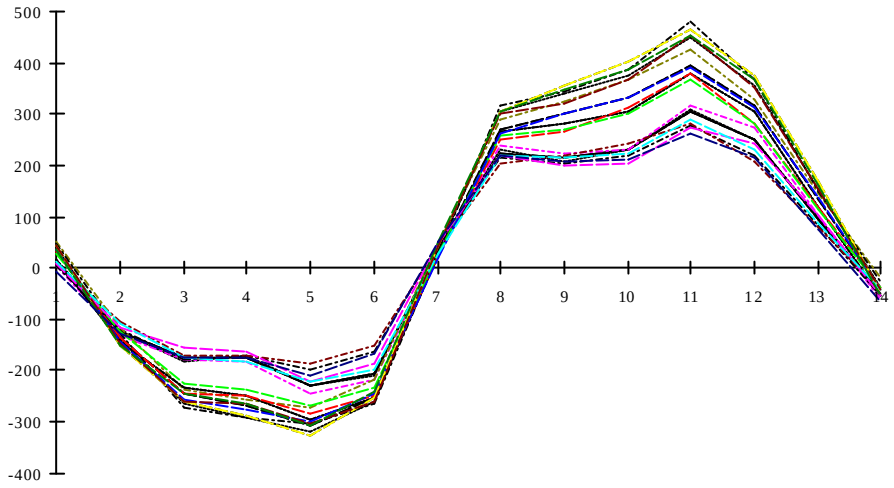


Figura 50. Ejemplo de la Proyección Andrews (Fuente: [ORT95]).

Esta particular función conserva las distancias, los puntos cercanos aparecen como líneas que se aproximan para todos los valores de t, mientras que puntos distantes se representarán como líneas que difieren al menos para algún intervalo de t

El examen de las gráficas por bandas de líneas **permite distinguir grupos de datos del espacio original.**

PROYECTOR NO LINEAL BASADO EN REDES KOHONEN O SOM (SELF-ORGANIZED MAPS)

Pertenece a la categoría de las redes neuronales competitivas o mapas de autoorganización, es decir, con aprendizaje no supervisado de tipo competitivo. Poseen una arquitectura de dos capas (entrada-salida) (una sola capa de conexiones), funciones de activación lineales y flujo de información unidireccional (son redes en cascada).

La red neuronal se entrena con una base de datos *n*-dimensional, de forma que el número de entradas de la red neuronal es igual al número de atributos que tengamos.

La salida de la red neuronal está compuesta por una rejilla de celdas en las que están conectadas todas las entradas.

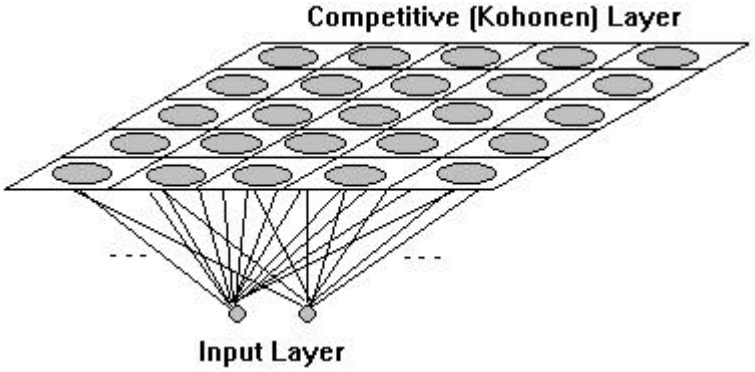


Figura 51. Estructura de la Red Kohonen.

El objetivo de este tipo de redes es clasificar los patrones de entrada en grupos de características similares, de manera que **cada grupo activará siempre la(s) misma(s) salida(s)**. Cada grupo de entradas queda representado en los pesos de las conexiones de la unidad de salida triunfante. La unidad de salida ganadora para cada grupo de entradas no se conoce a priori, es necesario averiguarlo después de entrenar a la red.

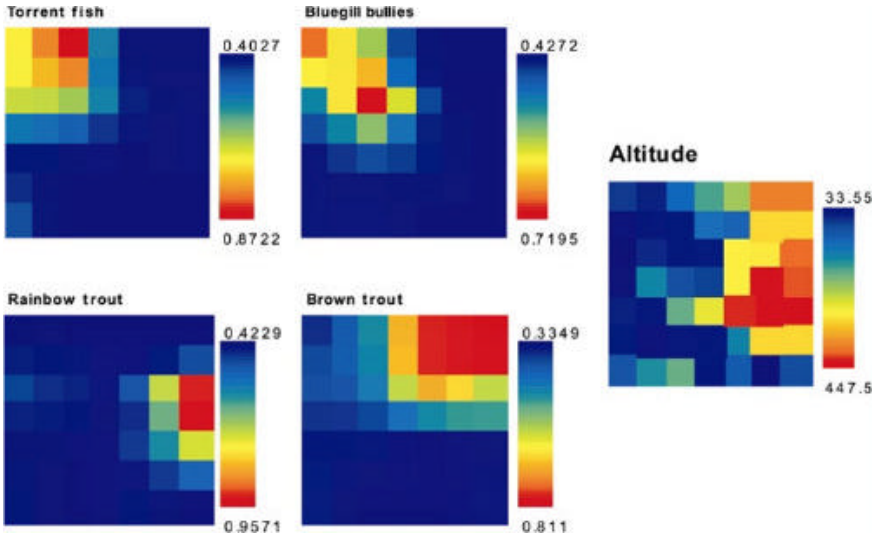


Figura 52. Resultado de clasificar cinco variables.

Una de las cualidades de este tipo de redes es la incorporación a la regla de aprendizaje de cierto grado de sensibilidad con respecto al vecindario o historia. Esto hace que el número de neuronas que no aprenden desaparezca, aumentando así su capacidad de extraer o mapear características topológicas de los datos.

De esta forma, la proyección de las observaciones y la visualización con un escalado de colores de cada una de las variables permite poder comparar las zonas en que se dividen y el número de las mismas.

Estas técnicas tienen una gran aplicabilidad en análisis de procesos industriales [CUA02]

Existen métodos que se basan en la misma filosofía. Por ejemplo, existe una técnica, denominada *Generative Topographic Maps (GTM)* [BIS96] que se basa en una variante probabilística de las redes autoorganizadas SOM de *Kohonen*.

PROYECTOR NO LINEAL RADVIZ

Este método [ANK96] puede comprenderse fácilmente con una explicación física.

Imaginemos que tenemos m puntos dispuestos alrededor de un circunferencia de radio unidad y separados a la misma distancia. Llamaremos a estos puntos S_1 a S_m respectivamente.

Supongamos ahora que tenemos m gomas unidas todas ellas por un extremo a una bola y por el otro, a cada uno de los puntos S de la circunferencia (ver figura).

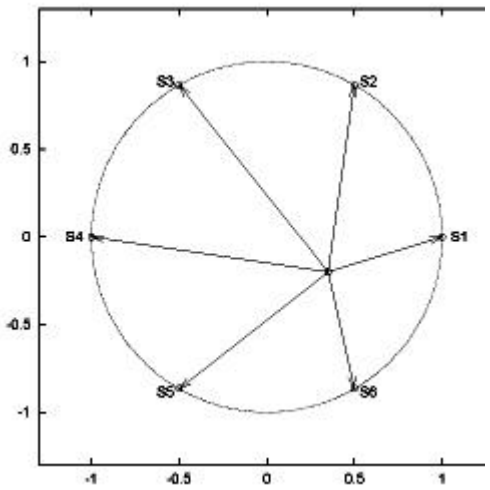


Figura 53. Un punto m*-dimensional proyectado en 2D.*

Finalmente asumamos que la constante de alargamiento (según los términos de la ley de Hooke) de la goma j es x_{ij} del punto i de la base de datos. Si soltáramos la bola llegaría a una posición de equilibrio, de tal forma que ésta denominada $(u_i, v_i)^T$ corresponde con la proyección en el espacio de dos dimensiones del punto $(x_{i1}, ..., x_{im})^T$ del espacio m-dimensional.

El método de proyección, por lo tanto, consiste en encontrar la transformación no lineal que cumpla ese principio. Para ello, se consideran las fuerzas que actúan en las gomas bidimensionales, cuya suma, cuando está la bola en equilibrio, será cero.

Cada una de las fuerzas será el producto de la constante de alargamiento de la goma por el vector en el que se extiende la goma. Es decir, si denotamos a $\mathbf{S_j}$ como el vector posición de cada una de las gomas y $\mathbf{u_i}=(u_i, v_i)^T$ tenemos:

$$\sum_{j=1}^{m}(\mathrm{S_j}-\mathrm{u_i})\cdot x_{ij}=0 \tag{3.30}$$

que puede ser resuelto,

$$\mathrm{u_i}=\sum_{j=1}^{m} w_{ij}\cdot \mathrm{S_j} \tag{3.31}$$

donde,

$$w_{ij}=\left(\sum_{j=1}^{m} x_{ij}\right)^{-1}\cdot x_{ij} \tag{3.32}$$

De forma que, para cada caso de i, u_i es simplemente una media ajustada de los S_j's cuyos pesos son los valores de las m variables normalizados a una suma igual a uno. Nótese que esta normalización hace que la proyección $\Re^m \rightarrow \Re^2$ sea no lineal.

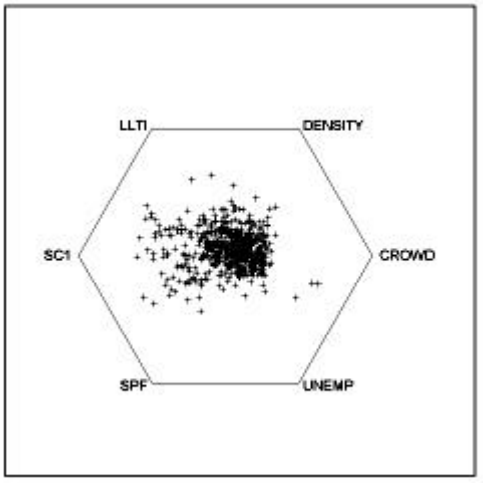


Figura 54. La proyección No Lineal mediante RADVIZ.

Esta técnica tiene como gran ventaja el bajo conste computacional, pero tiene una desventaja fundamental. Nótese que si dos puntos son por ejemplo: {2,2,2,2,...,2} y {8,8,8,8,...,8}, comparten el mismo punto en el espacio Â².

PROYECTOR NO LINEAL BASADO EN EL ANÁLISIS DE COMPONENTES CURVILÍNEAS

El *Análisis de Componentes Curvilíneas (CC),* también denominado *Cuantificación y Proyección Vectorial (VQP)* es una estrategia para la reducción dimensional y representación de conjuntos de datos multidimensionales.

El propósito de VQP es dar una representación conveniente de los datos en una dimensión baja, para un posterior agrupamiento y clasificación.

El principio de *VQP* es una red neuronal auto-organizada que realiza dos misiones:

- Cuantificación Vectorial (VQ) de la estructura de los datos de entrada.
- Proyección no lineal (P) de estos vectores cuantificados hacia un espacio de salida, realizando un despliegue demostrativo de la estructura

Cada una de las N neuronas tiene dos vectores de peso. Los vectores de entrada $\{x_i,\ i=1,...,N\}$ son de dimensión n, mientras que los correspondientes vectores de salida $\{y_i\}$son de dimensión p, con $p \le n$.

Los vectores de entrada se convierten en los prototipos de la distribución usando uno de los varios métodos de Cuantificación Vectorial que existen.

La capa de salida debe construir una proyección no lineal de los vectores de entrada, para conseguirlo, se consideran las distancias euclídeas entre los puntos de partida $x_i : X_{ij} = d(x_i, x_j)$. Las distancias correspondientes en el espacio de salida son $Y_{ij} = d(y_i, y_j)$. El objetivo es forzar a que Y_{ij} se iguale a X_{ij} para cada par de i y de j. Dado que una igualación perfecta no es posible en el caso de que los datos no sean lineales, se introduce una función de ponderación, dando como resultado una función cuadrática de energía:

$$E = \frac{1}{2}\sum_i \sum_{j \neq i} (X_{ij} - Y_{ij})^2 F(Y_{ij}, l_y) \tag{3.33}$$

Generalmente se escoge como una función acotada y monótona decreciente, para así favorecer la conservación de la topología local. Varios tipos son aceptables, pero es válida la simple función escalón:

$$F(Y_{ij}, l_y) = \begin{cases} 1 & si\ Y_{ij} \le l_y \\ 0 & si\ Y_{ij} > l_y \end{cases} \tag{3.34}$$

El algoritmo fue originalmente propuesto como una mejora a los mapas auto-organizados de Kohonen (SOM). La salida en este caso no es una estructura fijada de antemano, sino un espacio continuo capaz de tomar la forma de la estructura de los datos de partida.

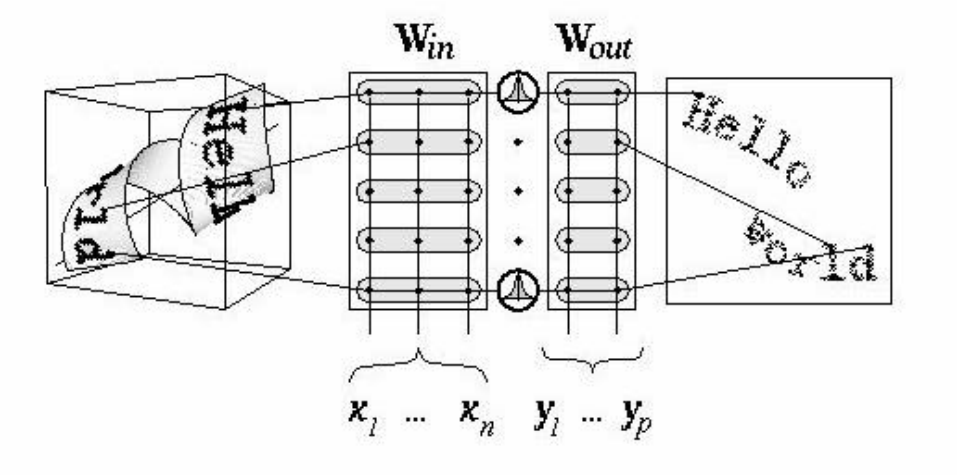


Figura 55. Proyección en dos dimensiones de un papel doblado en tres dimensiones.

En la Figura 55 se puede apreciar la estructura de una red ACC y un ejemplo de su utilidad: proyección en dos dimensiones de un papel doblado en tres dimensiones. Primero se 'aprende' como desdoblar el papel y después se proyecta el texto por medio de la relación construida entre los espacios de entrada y de salida.

3.4.1.5 OTRAS TÉCNICAS DE VISUALIZACIÓN

Realmente, cada día que pasa, aparecen nuevas técnicas o combinaciones de las ya explicadas. En [FAY02] se detallan gran cantidad de ellas.

Aún así, se considera que las técnicas descritas son las más utilizadas y su uso puede abarcar gran cantidad de usos dentro del proceso del *data mining*.

3.4.2 ALGORITMOS Y TÉCNICAS DE PREPROCESADO Y TRATAMIENTO DE LA INFORMACIÓN

La figura siguiente muestra algunas de las tareas más típicas dentro del preprocesado de los datos y tratamiento de la información. De esta forma, los grupos de algoritmos y técnicas que podemos encontrar pueden clasificarse dentro de cada una de éstas.

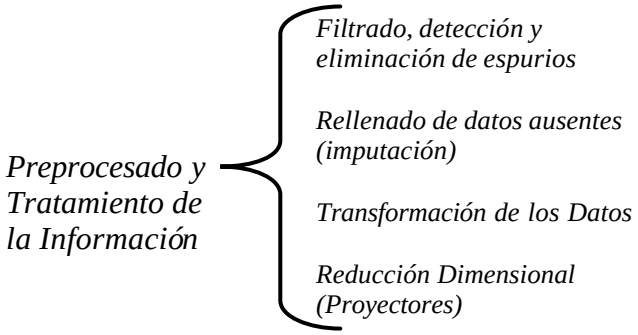


Figura 56. Grupos de Algoritmos y Técnicas en el Preprocesado y Tratamiento de la Información

3.4.2.1 FILTRADO, DETECCIÓN Y ELIMINACIÓN DE ESPURIOS

Se consideran valores no usuales a un conjunto de datos cuya disposición espacial resulta sensiblemente extraña frente al comportamiento general del conjunto [CAS01]. Este tipo de datos no usuales se denominan *outliers* o espurios y deben ser tratados convenientemente ya que pueden afectar enormemente a los resultados finales. Por ejemplo, si éstos son elevados, pueden influir en la creación de un clasificador o estimador, un modelo matemático, etc.

Los *outliers* pueden venir originados por errores en la medida, en la inserción de datos o en la transformación de los mismos aunque también puede ser debidos a sucesos inusuales que pueden ser interesante estudiar, ya que un estudio detallado puede aportar información valiosa sobre el proceso.

En los tres primeros casos, en cambio, sería deseable que los espurios fueran detectados, apartados y eliminados, aunque antes de poder ser eliminados, **es conveniente determinar su origen y solamente hacerlo cuando se tiene la certeza de las causas que los generan**.

La técnicas que se describen en [PAR01] [ROU87] [CAS01] [PER02] para detectar *outliers* en muestras normales multivariantes son :

- Usando técnicas de regresión para estimar el modelo que define los datos y determinando la desviación de los puntos frente al mismo según diversos criterios de calidad [PER02] (ver Tabla 10).
- Uso de histogramas o gráficos *boxplots* para detectarlos gráficamente.
- Mediante el uso de los autovalores de la muestra [CAS02].

- Mediante el cálculo de la distancia de Mahalanobis, que se define como [CAS02]:

$$d_j^2 = (x_j - \bar{x}) \cdot S^{-1} \cdot (x_j - \bar{x})^t \qquad (3.\ 35)$$

- donde x corresponde a una matriz de observaciones muestrales que sigue una distribución normal multivariante, y d_j^2 la distancia de Mahalanobis de la observación x_j al centro de la muestra.
- Usando proyectores lineales como: Análisis de Componentes Principales, Proyección Pursuit, etc.

Y para muestras con comportamiento no lineal[9]:

- Proyección Sammon.
- Redes de mapas autoorganizados (SOM).
- Proyectores PCA No Lineales.
- Generative Topographic Maps (GTM) [DAS03].
- Otros proyectores No Lineales basados en redes neuronales.
- Otras técnicas de Visualización Multivariante [FAY02]: Coordenadas paralelas, dendogramas, curvas Andrews, iconos, Radviz, etc.
- Otros métodos como el algoritmo PAELLA [CAS03].

El algoritmo PAELLA [CAS03], desarrollado por el Área de Proyectos de Ingeniería de la Universidad de La Rioja, consiste en un nuevo método para detección de espurios en bases de datos multivariantes, tanto para el caso normal como para el caso no normal (más frecuentemente este último en los procesos industriales). El algoritmo permite obtener un conjunto de datos de elevada pureza a partir de otro conjunto original "sucio" o sujeto a perturbaciones. La agresividad del algoritmo depende de parámetros que controlan su eficiencia según las necesidades especificas del analista.

En [CAS03] se compara este algoritmo con uno de los mejores existentes hasta el momento, el algoritmo BACON, superando a éste y demostrando una mayor estabilidad e independencia de la dimensión del problema. El algoritmo funciona tomando como datos de partida los resultados de un análisis cluster y construyendo en cada uno de estos dominios el soporte de la estructura intrínseca a los datos.

Debido a la necesidad de cómputo, en principio se proponen modelos lineales, sin que exista ningún impedimento que impida proponer otro tipo de modelos salvo la necesaria velocidad

[9] Todas estas técnicas se tratarán en apartados posteriores.

CRITERIO	Fórmula	Metodología
Residuos	$r_i = y_i - \hat{y}_i$	**Grafica de los residuos r_i versus los valores estimados $\hat{y}_i$. [HAI99]**
Residuos Estudentizados (residuo basado en la t-Student)	$rs_i = \frac{r_i\sqrt{n-p}}{\sqrt{(1-h_{ii})\sum_{i=1}^{n} r_i^2}}$	**Gráficas normales de los residuos estudentizados [ATK00]. Se recomienda fijan los límites superior e inferior en un intervalo de confianza del 95 %, es decir, en los valores ±1.96. Los residuos estadísticamente significativos son aquellos que no entren en estos límites [HAI99].**
Punto de Ruptura	$e_n^*(T,Z) = \min\left\{\frac{k}{n}; \sup_{Z'} \left\|T(Z) - T(Z')\right\| > d\right\}$ **Z conjunto de datos sin espúreos** **Z' conj. datos con k espúreos.** $T(Z) = \hat{?}$	**Es la menor fracción de contaminación (k/n) que puede causar que el estimador T tome valores arbitrariamente lejanos a T(Z). [DON83].**
Valores de Apalancamiento o Valores Sombrero **(en inglés: leverage value)**	$h_{ii} = x_i^T (X^T X) x_i$	**Gráfica de residuos estudentizados versus valores h_{ii}. Valores grandes de rs_i revelan la presencia de espúreos y valores grandes de h_{ii} revelan apalancamiento. Se puede decir, en general, que para problemas donde p sea mayor que 10 y el tamaño muestral n excede las 50 observaciones, los valores $h_{ii}>2p/n$ son considerados como grandes. Si por el contrario p < 10 y n < 50 se dice entonces que los valores $h_{ii}>3p/n$ son considerados grandes [HAI99]**
Prueba t	$t_j = \frac{\hat{q}_j\sqrt{n-p}}{\sqrt{u_j \sum_{i=1}^{n} r_i^2}}$ **u_j : j-esimo elemento de la diag** $(X^T X)^{-1}$	**H_0 ($q_j = 0$) será rechazada a un nivel de significancia a, cuando el valor absoluto de t_j sea mayor a $t_{n-p,1-a/2}$. En este caso se dirá que el coeficiente de regresión j-esimo es significantemente diferente de cero. [ROU87].**
Coeficiente de Correlación Múltiple al Cuadrado	$R^2 = 1 - \frac{SSE}{SST}$ **SSE y SST se definirán luego para cada estimador**	**Es el porcentaje de variabilidad de y explicada por la recta de regresión. $R^2 \leq 1$. Un valor cercano a 1 indica que una gran proporción de la suma total de los cuadrados ha sido explicada por la regresión.**

Tabla 10. Criterios de calidad para detectar espurios en modelos lineales.

3.4.2.2 RELLENADO DE DATOS INEXISTENTES

Los datos inexistentes son un mal endémico en las bases de datos de sistemas reales. Estos casos son muy comunes, por ejemplo en los procesos industriales donde se producen fallos de muestreo, se introducen mal los datos manuales, se realizan conversiones incorrectas o simplemente unas variables se muestrean a diferente velocidad que otras. Algunas aproximaciones simples [WIT00] [DIX79] [PAR01] y otras más complejas [DEM77] [PYL99] han sido propuestas para solucionar este problema.

Principalmente estas aproximaciones se pueden clasificar en los siguientes grandes grupos:

- Eliminar los patrones incompletos o llenar los datos mediante técnicas estadísticas que calculen los valores de reemplazo a partir del resto de los datos o asignar algún valor predicho. Por ejemplo:
  - Rellenar con la mediana o la moda.
  - **Rellenar con la media para preservar el valor medio o con el valor que menos influya en la desviación estándar**. Aunque [PYL99] demuestra que el segundo método es mejor que el primero ya que la media mide la tendencia central mientras que la desviación estándar no solo indica la tendencia central sino también la variabilidad dentro de la distribución (ver Tabla 10).
- **Usando estimadores adecuados** para cada tipo de datos y según la información que se tiene del proceso. Por ejemplo:
  - Obtener la relación entre variables mediante técnicas de regresión lineal, no lineal, redes neuronales, etc.; y rellenar los datos mediante estos estimadores.
  - Si son datos de series temporales, usar el dato anterior o posterior, desarrollar estimadores mediante Transformadas de Fourier, Transformadas Wavelets, Autocorrelación, etc.
- Mantener los patrones incompletos utilizando algoritmos específicos de *data mining* que contemplen la situación, estudiando los datos ausentes para detectar sus causas o descartando los patrones incompletos.

Posición	Muestra Original	Posición 11 Perdida	Rellenamos con el valor que Preserva la Media	Rellenamos con el valor que Preserva la Desviación Estándar
1	0.0886	0.0886	0.0886	0.0886
2	0.0684	0.0684	0.0684	0.0684
3	0.3515	0.3515	0.3515	0.3515
4	0.9874	0.9874	0.9874	0.9874
5	0.4713	0.4713	0.4713	0.4713
6	0.6115	0.6115	0.6115	0.6115
7	0.2573	0.2573	0.2573	0.2573
8	0.2914	0.2914	0.2914	0.2914
9	0.1662	0.1662	0.1662	0.1662
10	0.4400	0.4400	0.4400	0.4400
11	0.6939	???	0.3731	0.6939
Media	0.4023	0.3731	0.3731	0.3994
Desviación Estándar	0.2785	0.2753	0.2612	0.2753
Error estimado de la casilla 11			0.3208	0.0317

Tabla 11. Demostración de las ventajas de rellenar con el valor que preserva la desviación estándar frente al valor que preserva la media de las 10 primeras muestras (obtenido de [PYL99]).

3.4.2.3 TÉCNICAS DE ELIMINACIÓN DE RUIDO

El ruido existente en los datos puede provenir de diferentes sitios. Generalmente depende de los elementos de medición, la medición misma, factores externos que influyen en el sistema. Es decir, fundamentalmente depende de:

- Forma de realizar el muestreo.
- Los elementos de medición.
- Los factores externos que influyen en el sistema medido.

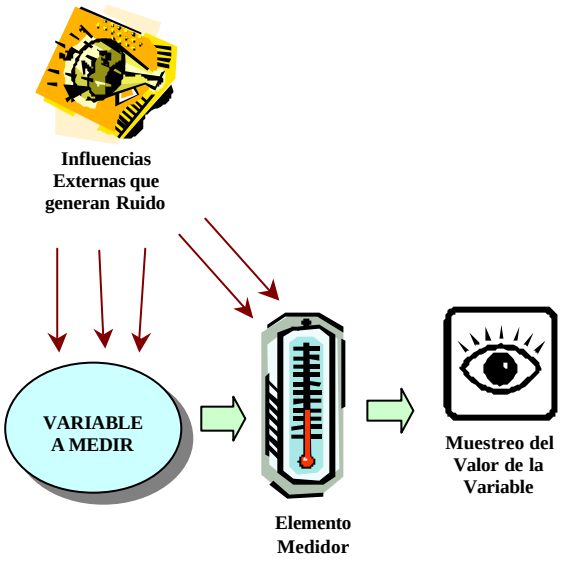


Figura 57. Elementos donde se produce ruido.

TÉCNICAS DE MUESTREO

Los dos tipos de problemas que resuelven las técnicas estadísticas son fundamentalmente:

- Problemas de Estimación.
- Problemas de contraste de hipótesis.

En ambos casos se trata de generalizar la información obtenida de una muestra a una población. Estas técnicas exigen que la muestra sea aleatoria.

En la práctica rara vez se dispone de muestras aleatorias, por la tanto la situación habitual es la que se esquematiza en la Figura 58.

Entre la muestra con la que se trabaja y la población de interés, o población *diana*, aparece la denominada población de muestreo: población (la mayor parte de las veces no definida con precisión) de la cual nuestra muestra es una muestra aleatoria.

En consecuencia la generalización está amenazada por dos posibles tipos de errores:

- **Error Aleatorio**: que es el que las técnicas estadísticas permiten cuantificar críticamente dependiente del tamaño muestral, pero también de la variabilidad de la variable a estudiar.
- **Error Sistemático**: que tiene que ver con la diferencia entre la población de muestreo y la población diana y que sólo puede ser controlado por el diseño del estudio.

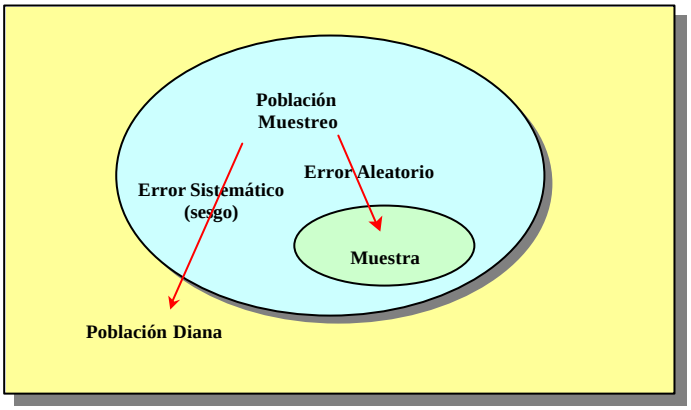


Figura 58. Errores que aparecen entre la población de muestreo y la población diana.

El tamaño muestral juega el mismo papel en estadística que el aumento de la lente en microscopía. Es decir, si no se ve una bacteria al microscopio, puede ocurrir que la preparación no la contenga o que el aumento de la lente sea insuficiente. Del mismo modo, para decidir el tamaño muestral para un problema de estimación, hay que tener una idea de la magnitud a estimar y del error aceptable, mientras si el problema consiste en un contraste de hipótesis, hay que saber el tamaño del efecto que se quiere ver. En un caso y en otro, hay que darse cuenta que el 100% de la inspección del espacio muestral puede ser muy costoso.

Existen dos tipos generales de muestreo: **El probabilístico** y **el no probabilístico**. En el primero, cada uno de los elementos de la muestra tiene la misma probabilidad de ser muestreado, y en el muestreo no probabilístico, la probabilidad de ser muestreado no es igual para todos los elementos del espacio muestral.

Aunque pareciera que el muestreo probabilístico es el más usado, en las investigaciones de mercado esto no es así. Un estudio de mercado siempre está enfocado a investigar ciertas características de, por ejemplo, empresas, productos y usuarios. Es decir, antes de iniciar la investigación siempre se hace una estratificación. Aunque se investiguen características que pueda tener toda la población, tales como usar calzado, fumar, hábitos de vestir, etc., siempre se estratifica antes de encuestar. No se debe confundir, por ejemplo, con investigar el % de gente que fuma, porque esto no sería una investigación de mercado. Una investigación acerca de los fumadores tendría como primera pregunta si la persona fuma y, seguiría una serie de cuestionamientos sobre sus gustos, preferencia de marcas, estrato social, etc. La estratificación implícita está en aplicar el cuestionario a quienes fuman, pues quien no fuma difícilmente opinaría con propiedad acerca de

gustos o marcas preferidas. A cualquier persona se le puede preguntar si fuma, pero no a cualquiera se le aplicará el cuestionario, que es la verdadera investigación.

Si se examinan más casos de investigación de mercados a base de encuestas, se encontrará siempre una estratificación preliminar implícita, y esto es un muestreo no probabilístico. Esta teoría es muy interesante y de gran aplicación de control de calidad, donde el universo de la muestra es finito y conocido.

Cuando trabajamos con una hipótesis de distribución normal, el cálculo de la muestra se deben tomar en cuenta algunas propiedades de ella y el error máximo que se permitirá en los resultados. Para el cálculo de *n* (tamaño de la muestra) se puede emplear la siguiente fórmula:

$$n = \frac{s^2 Z^2}{E^2} \tag{3.36}$$

Donde σ es la desviación estándar, que puede calcularse por criterio, por referencia a otros estudios o utilizando una prueba piloto. El nivel de confianza deseado se denota por *Z*, el cual se acepta que sea de un 95% en la mayoría de las investigaciones. El valor de *Z* indica el número de errores estándar asociados con el nivel de confianza. Su valor se obtiene mediante el uso de la tabla de probabilidades de una distribución normal. Para un nivel de confianza de 95%, *Z*=1.96, lo que significa que con una probabilidad total de 0.05 la media de la población caería fuera del intervalo 3·σ (σ es la desviación estándar de la muestra). Finalmente, *E* es el error máximo permitido y se puede interpretar como la mayor diferencia permitida entre la media de la muestra y la media de la población (*X*±*E)*.

ELIMINACIÓN DE RUIDO

Para poder eliminar el ruido, será necesario analizar sistemáticamente los diferentes factores que lo pueden generar de la siguiente manera:

- Determinar el tipo de variable y las posibles causas de ruido que pueden influirle. Por ejemplo, cuando se analizan series temporales es conveniente considerar las cuatro componentes básicas que las componen y que son: **la tendencia, el factor cíclico, las fluctuaciones estacionales y las variaciones no sistemáticas.**
- El componente de la tendencia se refiere al crecimiento o declinación en el largo plazo del valor promedio de la variable estudiada. Su importancia se deriva de considerar fluctuaciones en el nivel de la variable en el tiempo, con lo cual el estudio del nivel promedio de la variable a lo largo del tiempo es mejor que el estudio de esa variable en un momento específico de tiempo.
- Aún cuando puede definirse una tendencia de largo plazo en la variable, pueden darse divergencias significativas entre la línea de tendencia proyectada y el valor real que exhiba la variable. Esta divergencia se conoce como componente cíclico, y se admite entre sus causas el comportamiento del efecto combinado de fuerzas

económicas, sociales, políticas, tecnológicas, culturales y otras existentes en el entorno. La mayoría de estos ciclos no tiene patrones constantes que permitan prever su ocurrencia, magnitud y duración.

- En contraste con los componentes cíclicos, existen otros llamados estacionales, que exhiben fluctuaciones que se repiten periódicamente y que por lo regular dependen de factores externos.
- Aún conociendo los tres componentes señalados, una variable puede tener todavía un comportamiento real distinto del previsible por su línea de tendencia y por los factores cíclicos y estacionales. A esta desviación se le asigna el carácter de no sistemática y corresponde al llamado **componente aleatorio**.
- Analizar las técnicas de muestreo y el tamaño muestral obtenido [HAI99][PEÑ97]. El siguiente paso a realizar consiste en estimar el tamaño concreto de muestra a utilizar en los métodos siguientes, éste es un aspecto importante en tanto en cuanto determina el número de muestras o la información relevante de campo a utilizar.
- Estudiar las posibles deficiencias de los elementos de medición y almacenamiento de la misma [BAL97][SMI97][CRE93][SOD89]: personas, sensores, modo de almacenamiento en la base de datos, soportes hardware y software que lo realizan, etc.
- Identificar los diferentes ruidos y determinar la forma de eliminarlos mediante [OGA93][VIT91][WAN99]: filtros digitales (IIR, FIR, etc.), tranformadas inversas (Wavelet, Fourier, etc.), algoritmos a medida, manualmente, etc.

3.4.2.4 TRANSFORMACIÓN DE LOS DATOS

Dentro del Proceso de Transformación de los Datos, se utilizan técnicas y algoritmos para:

- Reducción de los Datos.
- Transformación de los Datos con

Se busca por lo tanto:

- Extracción de las características (o atributos) útiles de los datos (reducción de dimensionalidad).
- Transformación de los datos con el objetivo de proporcionar una representación de los datos mas intuitiva y manejable.

LA REDUCCIÓN DE LOS DATOS

Cuando el número de variables es considerable, con encontramos con múltiples dificultades debido a la alta dimensionalidad. Fundamentalmente, estas dificultades se describen en los siguientes aspectos [PYL99]:

- **Aumento del coste computacional**, necesidad de memoria, tamaño muestral necesario, etc.
- **Baja densidad en la población de muestras en el espacio de estado**. En un espacio de alta dimensionalidad, los conjuntos de datos están muy dispersos y "no rellenan" convenientemente el volumen dimensional [ORD00a]. Esto implica que la densidad de puntos dentro del espacio no es muy elevada y los conjuntos de datos están muy separados entre si, lo que puede generar importantes dificultades para detectarlos.
- Aumento del número de combinaciones posibles entre las variables, es decir, el número posible de estados del sistema se incrementa exponencialmente de tal forma que aumenta también exponencialmente las dificultades para generar un modelo que explique completamente todos los posibles estados que se puedan producir.
- Los conjuntos de datos muestran multicolinealidad (o concurvidad si generalizamos a nivel no paramétrico). Esto puede producir grandes problemas en muchas de las técnicas estadísticas y de minería de datos que se comentarán más adelante.

Debido a estos aspectos, el proceso de extracción de características trata de reducir la cantidad de datos (reducción de dimensionalidad) intentando mantener la mayor cantidad posible de la información que contienen los datos originales [WAN99].

La extracción de atributos debe cumplir con las siguientes condiciones:

- La dimensionalidad del vector de atributos debe ser menor que la del patrón original.
- Las nuevas variables deben presentar una interrelación lo más reducida posible.
- Los atributos deben representar una codificación óptima de la entrada, prescindiendo de la información que no sea importante.
- Si los datos proceden de un proceso del cual se posee conocimiento a priori, esta información puede ser utilizada en el proceso de reducción de información.

Las tres principales reducciones de la dimensionalidad son:

- Eliminación de atributos o variables (usualmente las columnas de la base de datos).
- Reducción de observaciones (generalmente las filas de la matriz de datos).
- Regularización del atributo. Que consiste en la eliminación de los valores extremos reduciendo el número de valores posibles para un atributo.
- Fundamentalmente, el proceso de reducción de dimensión se basa en:
- Análisis de correlación entre variables, de forma que se pueda eliminar una de ellas si es "explicada convenientemente" por otras.
- Uso de sistema de representación gráfica: gráficos *boxplots*, *scatterplots*, proyectores multidimensionales, etc.; para determinar interrelaciones lineales o no lineales, de los diferentes atributos.
- Métodos de proyección lineal o no lineal que generan nuevos ejes multidimensionales que se ajustan mejor a la estructura intrínseca de los datos y que permiten reducir el número de variables a utilizar. Por ejemplo: PCA lineal o no lineal, otros proyectores basados en redes neuronales (SOM, LVQ), etc. **Estas técnicas ya se han explicado anteriormente.**

Pero no solo es necesario la reducción de la dimensión de los datos, sino también adaptar los mismos a los algoritmos con que van a ser analizados.

Las operaciones de transformación de los datos se pueden clasificar en dos tipos:

- Creación de datos derivados.
- Transformación de la distribución de los datos.

CREACIÓN DE DATOS DERIVADOS

Consiste en crear nuevas variables, que pueden estar asociados a nuevas características, mediante la combinación de otras ya existentes. Este tipo de asociaciones resultan muy útiles y pueden mejorar considerablemente los resultados finales.

Se pueden clasificar en:

- Variables obtenidas de proyecciones derivadas de los algoritmos de reducción de dimensión. Por ejemplo, es muy usual utilizar PCA para determinar unos nuevos ejes y trabajar con las variables que los definen [WAN99] [SEB01].
- Términos derivados del tiempo. Si alguna de las variables es dependiente del tiempo, y el modelo depende también del tiempo, suele resultar útil calcular la primera y segunda derivada de la misma con respecto al tiempo pues marcan la 'velocidad' y 'aceleración' de la señal.
- Uso de *Metadata* para crear nuevas variables[10]. En [WES98] se define la *MetaData* como datos que están dentro de otros datos. Es decir, variables de las que se puede deducir otra información de la que a simple vista se observa. Por ejemplo de una simple fecha indicada en la forma siguiente: DDMMAAAA, donde los dos primeros dígitos indican el día, los dos siguientes el mes, y los cuatro siguientes el año, podríamos extraer mucha más información: estación del año, semana del mes, día de la semana, etc. Lo mismo podríamos decir de un simple DNI de donde podríamos sacar: edad aproximada de la persona, zona de donde proviene, etc.; con un cierto grado de seguridad.
- Valores estadísticos: medias, moda, varianza, curtosis, etc.; muy utilizadas en análisis de datos.
- Creación de nuevos datos a partir de la información obtenida de herramientas como: transformadas de Fourier, Wavelets, etc.

[10] Esta definición difiere un poco de la estadística o de almacenamiento de bases de datos donde se refiere a variables que incluyen otras variables. Mientras, que aquí se refiere a información que se puede extraer de una variable y puede servir para generar otros atributos distintos.

TRANSFORMACIÓN DE LA DISTRIBUCIÓN DE LOS DATOS

Se pueden enumerar en los siguientes tipos de transformaciones:

- Transformaciones Matemáticas. Las transformaciones matemáticas de los datos mediante la aplicación de logaritmos o raíz cuadrada has resultado ser un método muy eficaz para estandarizar las distribuciones de variables muy sesgadas. La estandarización de una variable es un proceso importante puesto que muchos de los métodos estadísticos necesitan que la distribución de las variables a tratar sea normal.
- Discretización de las variables. En ocasiones, la discretización de variables continuas proporciona una mejor interpretación de los datos. Generalmente, muchos de los algoritmos de minería de datos, como algunos generadores de reglas o árboles clasificadores, utilizan variables nominales lo que nos obliga a transformar las variables continuas que tengamos. Por ejemplo, para discretizar una variable se pueden utilizar los cuantiles 10, 25, 50, 75 y 90, para designar las clases: muy bajo, bajo, debajo de la media, encima de la media, alto, muy alto. Otros discretizadores más avanzados pueden estar basados en redes neuronales, sistemas *fuzzy* o algoritmos de clusterizado.
- Normalización de las variables. Muchos algoritmos que tratan atributos numéricos necesitan que los rangos de las variables sea fijados, para ello, generalmente se recurre a la *normalización*.

La Normalización

La normalización es uno de los procedimientos que más se utilizan para escalar las variables. **Consiste en ajustar el rango de los valores entre 0 y 1** (o también usualmente entre 0.1 y 0.9 cuando se trabajan con redes neuronales y funciones sigmoideas), dividiendo por el máximo todos los valores, o restando el mínimo y dividiendo por la diferencia del máximo y mínimo. De esta forma, dada una variable numérica x cuyo valor x_i que corresponde a la observación i de N observaciones, se define normalización como:

$$Norm(x_i) = \frac{x_i}{\max\limits_{i=1,..,N}(x_i)} \tag{3.37}$$

o también:

$$Norm(x_i) = \frac{x_i - \min\limits_{i=1,..,N}(x_i)}{\max\limits_{i=1,..,N}(x_i) - \min\limits_{i=1,..,N}(x_i)} \tag{3.38}$$

Otro tipo de técnica de normalización, llamada *estandarización* o *normalización estándar*, consiste en calcular la media estadística y la desviación estándar, restar la media y dividir el resultado por la desviación.

$$\text{Stand}(x_i) = \frac{x_i - \overline{x_i}}{s} \qquad (3.\ 39)$$

Esto, produce un nuevo tipo de variables con media cero y desviación estándar uno.

Algunas veces, simplemente se realiza la normalización decimal, que consiste en desplazar el punto decimal de forma que el rango quede entre 0 y 1.

Existen otras funciones más sofisticadas, basadas generalmente en funciones en "S" o en funciones "logistic" que intentan minimizar los valores extremos [PYL99].

3.4.3 DESCUBRIMIENTO DE GRUPOS, PATRONES YO REGLAS. MODELIZADO DESCRIPTIVO.

El objetivo del Modelizado Descriptivo, es la descripción total de los datos o del proceso que los generan.

Debido a que, pocas veces son aplicables modelos univariantes para explicar el comportamiento de algo porque, normalmente, existen más de una variable dependiente y existe incertidumbre sobre cuáles son las autenticas dependencias existentes entre ellas.

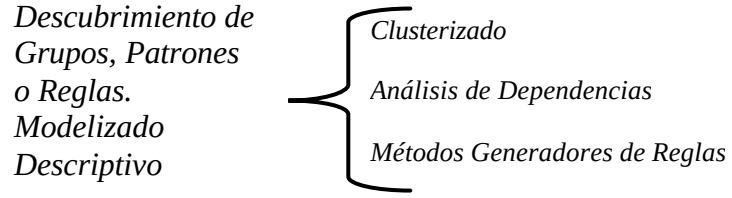


Figura 59. Grupos de Métodos y Algoritmos del Modelizado Predictivo

Dentro de esta serie de técnicas, nos encontramos fundamentalmente con:

- Algoritmos de segmentación (o también llamados de clusterizado): Que buscan grupos de datos que tienen similares características.
- Técnicas que describen densidades de probabilidad.
- Técnicas que describen las dependencias entre variables (correlación, gráficos de relación entre variables, etc.).
- Técnicas que buscan dentro de los datos, patrones, reglas como por ejemplo *las reglas de asociación* que analizan las relaciones entre las variables o las observaciones, y buscan combinaciones que se repiten usualmente.

3.4.3.1 ALGORITMOS DE CLUSTERIZADO

Es usual, asimismo, que existan diferentes estructuras de comportamiento, susceptibles de ser "explicadas" con diferentes modelos. Así pues, lo primero será estudiar como determinar estas microagrupaciones o clases. En este punto, se exponen diferentes técnicas que pueden ser utilizadas para estas tareas. Estas técnicas tratan de clasificar los patrones obtenidos en clases, ya sean estas conocidas o no.

La diferencia fundamental entre los métodos de agrupamiento (*clustering*) y clasificación, que se verán posteriormente, radica en que los segundos necesitan el conocimiento previo de la pertenencia a una determinada clase. En los métodos de clasificación, los patrones de entrenamiento contienen la información de la clase a la que pertenece cada patrón y buscan relaciones que permitan clasificarlos. Sin embargo, los métodos de agrupamiento deben de definir una función útil de clasificación sobre el conjunto x_i (donde $i=1, \ldots, N$), cuando N generalmente no es determinada. El número de grupos que se van a formar es desconocido, por lo que los algoritmos de *clustering* emplean una técnica basada en dos pasos: un bucle exterior considera posibles números de grupos, mientras que un bucle interior ajusta de la mejor forma posible los datos a ese número fijo de grupos.

Dado un número k de grupos, el objetivo de los algoritmos de vecinos más cercanos es encontrar la mejor k-partición de forma que los patrones de cada grupo de la partición estén más cercanos entre sí que de los patrones de los otros grupos. Una vez determinada esta partición, se puede intentar representar cualquier nuevo patrón en función del más cercano.

También, dados una serie de patrones ejemplo, podemos intentar representar cualquier nuevo patrón en función del más cercano, entendiendo la noción de distancia con cualquier métrica. Los individuos que queden clasificados en el mismo grupo serán lo más similares posible [BRA98].

Los clasificadores de vecinos más cercanos realizan la clasificación exclusivamente en función de estas distancias, siendo por lo tanto en principio no supervisados. No obstante de ellos han surgido una serie de variaciones que desembocan en cuatro grandes grupos:

- *K*-medias
- *K-NN* o K vecinos más cercanos.
- *LVQ*: Learning Vector Quantization
- *MC* o mapas de características.

A continuación, para aclarar mejor esta serie de métodos, se explicarán algunos de los más utilizados y que pertenecen a alguna de las familias antes descritas.

MÉTODO DE LAS K-MEDIAS

Es el más sencillo de los algoritmos de agrupamiento (*clustering*) habituales [FUK72][DUD73].

Sean p vectores de n características o vectores característicos X_j $(j=1, ..., p)$ que pueden agruparse en N clases cada uno de sus miembros N_i $(i=1, ..., N)$:

Se eligen una serie de valores del espacio como centros, a partir de los cuales se comenzará a generar clases o grupos. Cada vez que se presenta un patrón, se calcula su distancia a todas las medias y se le asigna la clase cuya media sea más cercana,

Se recalcula entonces la media de esa clase como el baricentro de todos los puntos que pertenecen a ella, incluido el último asignado de la forma siguiente,

$$M_t(t+1) = \frac{1}{N_i}\sum_{j=1}^{N_i} X_j \qquad (3.40)$$

siendo X_j $(j=1, ..., N_i)$ patrones asignados a M_t; y se repite la operación tantas veces como puntos se quiera clasificar o hasta que la media en el paso $t+1$ sea igual a la del paso t.

MÉTODO DE LOS K-VECINOS O K-NN

Es uno de los algoritmos más antiguos, surgió cuando se observó que el fijarse en un sólo patrón provoca que la existencia de un único punto defectuoso desvíe la clasificación sin remedio. Al ser no supervisado, no exige entrenamiento, durante la clasificación se calculan las distancias entre los patrones de entrada y los ejemplos almacenados. Se buscan los k ejemplares más cercanos y se asigna al patrón de entrada la clase más abundante entre estos k ejemplos.

ALGORITMOS LVQ (LEARNING VECTOR QUANTIZATION)

Frente a los K-Vecinos, los clasificadores ***LVQ*** -Learning Vector Quantization- sólo almacenan un número controlable de patrones. El entrenamiento del algoritmo *LVQ* (supervisado) se realiza en varias etapas. En primer lugar se determinan el número de ejemplos a almacenar, generalmente con el algoritmo de las K-medias antes descrito u otro procedimiento cualquiera de *clustering*. A partir de estos ejemplares se asigna cada patrón de entrenamiento al ejemplar más cercano, penalizándolo si es de la clase incorrecta y beneficiándolo si es de la correcta.

MÉTODO DE LAS DISTANCIAS ENCADENADAS (CHAIN-MAP)

Consiste en:

- Elegir un vector característico al azar X_i de los p que se tienen y colocarlo en la primera posición de una lista.

- Después se coloca en posición siguiente de la lista el vector más cercano al primero.
- Se elige el siguiente más cercano al último de la lista, y así sucesivamente, quedando ésta de la siguiente manera:

$$X_i(0),\ X_i(1),\ X_i(2),\ X_i(3),\ ...,\ X_i(p-1) \qquad (3.41)$$

  donde *X(1)* es el vector más cercano al *X(0)*, *X(2)* es el más cercano a *X(1)*, y así sucesivamente.

- Una vez obtenido este vector, se calculan las distancias euclídeas entre ellos d_1=distancia entre *X(0) y X(1)*, d_2=distancia entre *X(1) y X(2)*, etc. Y se representan gráficamente (), donde se pueden intuir las clases rápidamente.

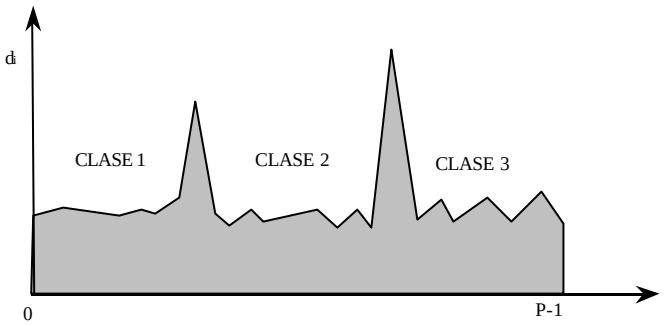


Figura 60. Clusterizado mediante chain-map.

MÉTODO MÁX-MIN

Los pasos del algoritmo son:

- Se escoge un elemento al azar X_i y se crea la primera clase a_1.
- Se calcula la distancia euclídea con todos los vectores, y se elige aquél que tenga la mayor. Este formara la segunda clase a_2.
- Para cada vector no agrupado, se obtiene las distancias euclídeas con los dos vectores de las clases a_1 y a_2; y se toma la mínima de las dos (para cada vector).
- Del conjunto de distancias, se elige la mayor. Si ésta es mayor que una determinada fracción de la distancia entre a_1 y a_2, se crea una tercera clase a_3.
- Se realiza el punto 3 con las tres clases y los (p-3) vectores, y después el punto 4, considerando que ahora es la fracción de la mínima distancia entre las tres clases.
- Se realiza el punto 5, hasta que ya no se definen más clases.
- Los elementos sin clasificar, se asignan a la clase cuya distancia sea más cercana.

El funcionamiento de este algoritmo depende de la fracción elegida.

ALGORITMO FUZZY C-MEDIAS *O* FUZZY ISODATA

Este método se basa en el algoritmo de *K-Medias*, pero es más sofisticado. Los pasos son los siguientes:

Se inicializa N_c=K, donde N_c es el número actual de clusters y K el deseado. Se eligen N_c vectores de forma aleatoria que formará cada uno de ellos un cluster inicial.

Se agrupan los demás elementos en cada cluster según el criterio de la distancia euclídea.

Se eliminan las clases cuyo número de miembros sea inferior a q_c previamente definido. Se actualiza N_c.

Se actualizan los centroides mediante la media muestral de cada clase.

$$Z_i = \frac{1}{N_i} \cdot \sum_{j=1}^{N_i} X_j \qquad para \quad i = 1,2,...,N_c \tag{3.42}$$

Se calcula la distancia euclídea media de todos los vectores de cada clase con su centroide correspondiente. Este parámetro sirve para obtener un valor que nos indique la dispersión que existe en cada grupo o clase.

$$\overline{D_i} = \frac{1}{N_i} \cdot \sum_{j=1}^{N_i} \left\| X_j - Z_i \right\| \qquad para \quad i = 1,2,...,N_c \tag{3.43}$$

Se obtiene también la distancia euclídea media de todos los clusters.

$$\overline{D} = \frac{1}{N_c} \cdot \sum_{i=1}^{N_c} N_i \cdot \overline{D_i} \tag{3.44}$$

Se comprueba si la iteración actual es la última (el número de iteraciones viene definido por el usuario). Si es así se iguala q_c a cero y se salta al punto 11. En segundo lugar, se realiza un test de posible uniones de clusters. Si N_c ³ 2·K entonces se salta al punto 11, sino se continúa con el punto 8.

Se obtiene un vector de características *n*-dimensional, según la expresión:

$$s_i = \begin{pmatrix} s_{i1} \\ s_{i2} \\ ... \\ s_{in} \end{pmatrix}; \qquad s_{ij} = \sqrt{\frac{1}{N_i} \cdot \sum_{k=1}^{N_i} \left(X_{kj} - Z_{ij} \right)^2} \tag{3.45}$$

$i = 1,2,...,N_c$ *(clases)*

$j = 1,2,...,n$ *(características)*

$k = 1,2,...,N_i$ *(elementos de la clase* a_i *)*

Se obtienen las desviaciones típicas máximas de cada clase. Es decir, se selecciona la componente mayor de las desviaciones de cada grupo, obteniéndose:

$$\{ s_{1max}, s_{2max}, \ldots, s_{N_c max}) \tag{3.46}$$

Se mira si hay que dividir una clase con las siguientes condiciones:

Si la dispersión media de la clase a_i candidata a dividirse en dos es superior a la media y, además, el número de elementos es al menos superior al doble del número mínimo de elementos permitido por grupo entonces se produce la división de esa clase. Resumiendo, la clase se divide si se cumple la relación:

$$D_j > D \quad y \quad N_j > 2 \cdot (\theta_N + 1 \tag{3.47}$$

Por otro lado, también se realiza la división si se cumple la condición:

$$N_c \leq \frac{K}{2} \tag{3.48}$$

El proceso de división se puede realizar de múltiples formas: crear dos centroides a partir del centroide principal con las mismas componentes que ésta, excepto la componente que produce más dispersión aplicando un coeficiente positivo para una y otro negativo para la componente de cada uno de los nuevos centroides; otra forma consiste en obtener las dos muestras más alejadas del centroide y se consideran como los centroides de las dos nuevas clases.

Se calcula la distancia entre parejas de clusters:

$$\begin{aligned} & D_{ij} = D_{ji} = \left\| Z_i - Z_j \right\| \\ & i = 1,2,\ldots,N_c - 1 \qquad j = i+1, i+2, \ldots, N_c \end{aligned} \tag{3.49}$$

Se comparan las distancias menores que el parámetro q_c y se toman las más pequeñas (si existen) en orden creciente.

Se unen las parejas de clusters con distancias menores, sólo si, ninguna de estas dos clases ha sido fusionada con otra en la misma iteración. El centroide calculado es:

$$Z_{ij} = \frac{1}{N_i + N_j} \cdot \left(N_i \cdot Z_i + N_j \cdot Z_j \right) \tag{3.50}$$

Lógicamente, cuando se produce una unión, hay que actualizar el parámetro N_c.

Se comprueba si se ha llegado a la última iteración. Si no es así, se salta al punto 2.

EL MÉTODO DE CLUSTERIZADO DE MONTAÑA

Existen otras técnicas de clusterizado más avanzadas, como por ejemplo: *el método de clusterizado de montaña* y *el clusterizado substractivo.*

- El primero consiste en crear una rejilla, donde las regiones entre intersecciones de las líneas son posibles candidatos a clusters, con centro en la intersecciones de las líneas.
- Después se crea una función montaña que representa la densidad de datos en cada punto de la rejilla. Esta densidad viene dada por la función:

$$m(v) = \sum_{i=1}^{N} \exp\left(-\frac{\|v - X_i\|^2}{2 \cdot s^2} \right) \tag{3.51}$$

  donde x_i es el elemento i de los N datos, v cada punto de la rejilla y s una constante. La altura de la montaña en cada punto de la rejilla $m(v)$ dependerá de las distancias de todos los puntos de datos a ese punto de la rejilla v.

- Se selecciona, el cluster con mayor altura de todos c_1.
- Se realiza una substracción de la montaña original con una montaña con centro en c_1 y distribución Gaussiana. La ecuación de substracción es:

$$m_{nueva}(v) = m(v) - m(c_1) \cdot exp\left(-\frac{\|v - c_1\|^2}{2 \cdot b^2} \right) \tag{3.52}$$

- El resultado del punto 4, es una nueva montaña. Se repite los puntos 3 y 4 hasta que no quede ningún punto sin clasificar o la montaña que se obtenga tenga una altura menor que un umbral definido.

CLUSTERIZADO SUBSTRACTIVO

El problema del clusterizado en montaña, es que el proceso de cómputo crece exponencialmente a medida que aumenta las dimensiones de los puntos.

En el clusterizado substractivo, los cluster iniciales son los mismos puntos de clasificación.

El funcionamiento de este algoritmo consiste en:

- Obtener el potencial de cada punto con respecto a todos los demás. Este valor depende inversamente de la distancia entre dimensiones y del número de vecinos que tenga. El potencial se calcula de la siguiente manera:

$$p_i = \sum_{j=1}^{n} \exp\left(-\frac{\|X_i - X_j\|^2}{(r_a/2)^2} \right) \tag{3.53}$$

donde r_a define el radio de los vecinos.

- Una vez obtenido los valores para cada punto, se busca el que mayor potencial tiene. Éste será el primer cluster c_1.
- Después se anulan todos aquellos puntos que estén dentro de su rango de influencia y se busca el segundo máximo para obtener el segundo *cluster*. Si el potencial de este segundo punto es mayor que la fracción estimada se marca como centro, continuando con el mismo proceso hasta que no queden más. Para realizar la anulación, se realiza la operación de substracción de todos los potenciales calculados:

$$p_i = p_i - p_{c_1} \cdot \exp\left(-\frac{\left\| X_i - X_j \right\|^2}{(r_b / 2)^2} \right) \tag{3.54}$$

Este algoritmo permite realizar con un grado óptimo de eficiencia el proceso de clusterizado, dependiendo esta eficacia de una elección correcta de los radios r_a y r_b.

MÉTODO DE LAS HIPERESFERAS

La tarea de clasificación **consiste básicamente en dotar a determinadas zonas del espacio muestral de unas etiquetas de clase de tal forma que seamos capaces de asignarle esta etiqueta a un nuevo punto**. Esto puede hacerse de una forma simplificada dividiendo el espacio en figuras más manejables como esferas o cuadrados de varios tamaños; combinando todas esas zonas simples podemos formar regiones tan complicadas como queramos. Esta es la base del algoritmo llamado de las esferas para tres dimensiones o de las hiperesferas de forma más general. Cada hiperesfera queda completamente definida conociendo su centro, el radio y una etiqueta de clase, de modo que cuesta poco mantenerla en memoria, por lo que pueden generarse muchas, permitiendo formar regiones complejas.

De forma simplificada, el algoritmo comienza iniciando a 0 el número de esferas. Al presentar un nuevo patrón se hace una búsqueda para determinar:

- Una hiperesfera H_1 que cubra al patrón (punto).
- Una hiperesfera H_2 que cubra al patrón y que sea de la misma clase que éste.
- La hiperesfera H_3 de la clase adecuada que se encuentre más cercana al patrón lo cubra o no.

Si las tres coinciden, quiere decir que el punto está en una zona claramente definida correspondiente al mismo grupo. Si la más cercana es de la clase acertada, $H_1 = H_2$, la hiperesfera es "recompensada", desplazando su centro en dirección al nuevo punto y ampliando su radio. Si H_1 es de clase incorrecta, se recompensa la H_2 y se penaliza la primera, disminuyendo su radio y alejando su centro en el sentido opuesto al punto. Si no existe ningún punto de la misma clase H_3, ésta es de nueva aparición, y se crea una nueva hiperesfera con centro en el patrón y un radio inicial que se pasa como parámetro al algoritmo. En el caso de que no exista H_2 y se recompense H_3 , se analiza

previamente su distancia al punto. Si ésta es demasiado grande, se crea una nueva hiperesfera en su lugar.

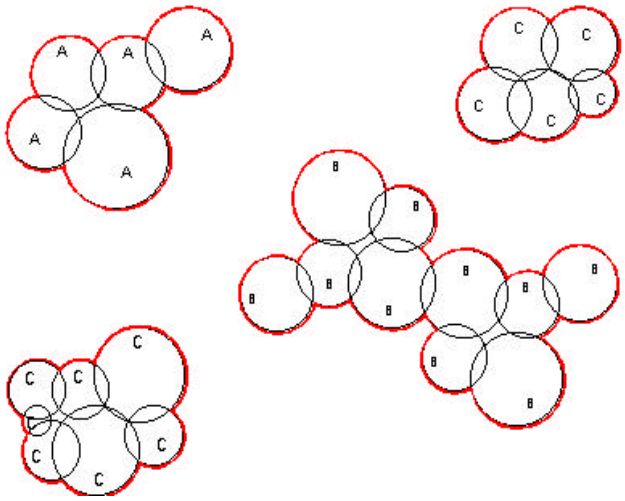


Figura 61. Clusterizado mediante hiperesferas.

MAPA DE CARACTERÍSTICAS

Por otra parte, los mapas de características siguen un algoritmo similar al LVQ, si bien su característica más importante es la rápida convergencia, a costa de una mayor dependencia del algoritmo de búsqueda de los centros.

VISUALIZACIÓN BASADA EN OTROS PROYECTORES

El uso de los proyectores (SOM, Sammon, RADVIZ, etc.), vistos en el apartado anterior, son uno de los métodos más utilizados en visualización de casos. Estos métodos se pueden usar para realizar tres tareas fundamentales:

- La primera es conseguir una apreciación global de la forma de los datos y su posible *clustering*.
- La segunda es ver características de los clusters y posibles correlaciones.
- La tercera es examinar nuevas muestras para clasificar y detectar novedades en los datos.

Este método de buscar correlaciones no es muy exacto pero es fácil de llevar a cabo ya que el cerebro humano es muy eficiente buscando patrones coincidentes. Este es un buen método para seleccionar componentes de cara a un estudio más detallado.

3.4.3.2 REGLAS DE ASOCIACIÓN

Las reglas de asociación [AGR93], derivan de un tipo de análisis que extrae información por coincidencias. Este análisis permite descubrir correlaciones o concurrencias en los sucesos de los datos a analizar y se formaliza en la obtención de reglas de tipo: *SI...ENTONCES*.

Por ejemplo si tenemos una tabla que hemos creado de las veces que fuimos a jugar a golf y pudimos finalmente jugar o no según las condiciones ambientales (ver tabla).

Ambiente	Temp.	Humedad	Viento	Jugar
soleado	alta	alta	no	No
soleado	alta	alta	si	No
nublado	alta	alta	no	Si
lluvia	media	alta	no	Si
lluvia	baja	normal	no	Si
lluvia	baja	normal	si	No
nublado	baja	normal	si	Si
soleado	media	alta	no	No
soleado	baja	normal	no	Si
luvia	media	normal	No	Si
soleado	media	normal	Si	Si
nublado	media	alta	Si	Si
nublado	alta	normal	No	Si
lluvia	media	alta	Si	No

Tabla 12. Histórico de sucesos de un juego de golf.

Podríamos deducir dentro todas las reglas posibles las que más han aparecido:

```
If humedad=normal and viento=no Then Jugar=SI 4/4
If humedad=normal and Jugar=SI Then viento=no 4/6
If viento=no and Jugar=SI Then humedad=normal 4/6
If humedad=normal Then viento=no and Jugar=SI 4/7
If viento=no Then Jugar=SI and humedad=normal 4/8
If Jugar=SI Then viento=no and humedad=normal 4/9
If true Then humedad=normal and viento=no and Jugar=SI 4/12
```

Figura 62. Reglas de asociación más importantes encontradas de la tabla de los juegos de golf.

Si pensamos en 100% de éxito, entonces sólo la primera regla cumple (cuatro de cuatro), pero el estudio de las demás reglas, nos puede dar una gran cantidad de información de reglas y patrones asociados.

Sea A el conjunto de todos los atributos de los casos a analizar, esta modalidad de exploración de datos, encuentra relaciones del tipo:

$$f(X) \Rightarrow q(Y) \tag{3.55}$$

donde:

X e Y son subconjuntos de A.

f y g son cualquier tipo de fórmulas que puedan definirse sobre los atributos de X e Y respectivamente.

De manera que la regla viene a decir que cuando se satisface la fórmula f, para un conjunto determinado de atributos, también se va a satisfacer la fórmula g para otro conjunto concreto de atributos de A. Cada regla obtenida, tiene asociado un índice de veracidad, que viene a mostrar la confidencia o fiabilidad de la regla, o lo que es lo mismo, cuantas veces se cumple esta regla en la muestra de datos explorados, en relación con el número de veces que se cumple la parte izquierda, *f(X)*.

Un enfoque un poco más concreto sería restringirse a fórmulas *booleanas*, constituidas por formulas atómicas del tipo x=1 o x=0 (siendo x perteneciente a A) y por conectivas ($\wedge$,$\vee$, $\neg$) (and, or, not).

Si la fórmula es booleana. el conjunto total de atributos A=(A_1,A_2, ..., A_p), está formado por atributos binarios, de modo que el dominio de cada A_i, es {0,1}. Sobre A se puede definir una relación r = {t_1, t_2, ..., t_n} dada por una matriz n x p, siendo n el número de casos disponibles y p el número de atributos.

Una regla de asociación acerca de una relación r es una expresión de la forma $X \Rightarrow Y$, donde X es un subconjunto de A e Y es un atributo perteneciente al subconjunto A-Y, El significado intuitivo de la regla consiste en que si en una fila de r cada uno de los atributos de X vale 1, entonces el atributo Y vale 1 en dicha fila.

La confidencia o fiabilidad de una regla, $X \Rightarrow Y$ en r viene dada por:

$$\frac{s(X \cup \{Y\}, r)}{s(X, r)} \tag{3.56}$$

donde:

s(X,r) $\rightarrow$ frecuencia del conjunto de atributos X en r ($\frac{n^o\, de\, ocurrencias\, de\, X}{n}$) esto es, es la fracción de filas de r que tienen a 1 los atributos de X.

s(X $\cup$ {Y}, r) $\rightarrow$ frecuencia de la regla X $\Rightarrow$ Y en r.

Por lo tanto. la confidencia de la regla $X \Rightarrow Y$ es la probabilidad de que se verifique el predicado Y condicionada a que satisfacen los predicados en X.

El descubrimiento de reglas de asociación consiste en encontrar todas las reglas de asociación $X \Rightarrow Y$, tales que la frecuencia de la regla alcance un umbral mínimo u_1, y la confidencia sea superior a un umbral u_2. Los umbrales u_1 y u_2, son fijados por el analista dependiendo del

tamaño de la muestra y del nivel de significación y fiabilidad establecido para las reglas descubiertas. Al subconjunto X de atributos de A cuya frecuencia es mayor que u_1, se denomina conjunto frecuente.

Para descubrir las reglas de asociación se puede seguir el siguiente procedimiento. Se buscan todos los conjuntos frecuentes de r formados por un único atributo, X_1. Sobre X_1, se forma los conjuntos candidatos a conjuntos frecuentes mediante todas las posibles combinaciones de dos atributos de X_1. Sobre cada uno de los conjuntos frecuentes candidatos se comprueba la frecuencia, de modo que se selecciona únicamente los conjuntos frecuentes, dando lugar a X_2. X_2 sirve a su vez para formar nuevas combinaciones de conjuntos candidatos de tres atributos y así sucesivamente hasta que ya no se puedan formar más candidatos

Las reglas obtenidas mediante este procedimiento pueden no ser interesantes para un usuario o en una situación particular debido a varios factores:

- La regla puede corresponderse con un conocimiento previo (ya conocido por el usuario).
- La regla puede referirse a un atributo o combinación de atributos no interesantes.
- Las reglas pueden ofrecer información redundante.

Por lo tanto, las reglas asociadas pueden someterse a procesos que permitan extraer la información más interesante. Para ello pueden realizarse tres operaciones:

- Poda: reducción de número de reglas.
- Ordenar: establecer un orden de reglas en función de algún criterio, como por ejemplo, nivel de significancia estadística.
- Clasificar: organizar las reglas en grupos con características similares.

Los criterios utilizados para podar y ordenar, usualmente son el nivel de confidencia, la frecuencia y la significancia estadística.

3.4.3.3 OTROS MÉTODOS

Otras técnicas que se explicarán más adelante, como por ejemplo los clasificadores (árboles, otras reglas, redes neuronales, etc.) pueden ser también utilizados para las tareas descritas en este apartado.

3.4.4 MODELIZADO PREDICTIVO

El modelizado predictivo es una de las facultades que tiene el aprendizaje humano. Efectivamente, éste a partir del uso de la observación desarrolla y valida continuamente un modelo que sirve para explicar cualquier fenómeno. Por ejemplo, desde edad temprana, un niño observando diferentes ejemplos de perros y usando las características principales de los mismos, va generando un modelo en su cerebro que le permita clasificar nuevos animales como perros. Y no solo eso, sino que ese modelo lo va afinando a medida que va validando nuevos ejemplos de perros, a través de lo que le dicen sus padres y su propia experiencia.

Es decir, nuestro cerebro continuamente esta desarrollando y validando modelos que nos permiten realizar generalizaciones a cerca del mundo que nos rodea y de esa forma poder comprenderlo [CAB97].

En *data mining* usamos el modelado predictivo para analizar una base de datos existente y determinar algunas de las características esenciales de los datos. Por supuesto, los datos deben incluir observaciones completas y válidas de forma que se pueda aprender de ellas y generar un modelo que pueda realizar predicciones acertadas.

Supongamos que tenemos M observaciones compuestas por n variables explicativas que denominaremos x_i (i=1, ..., M) y una variable dependiente de las mismas denominada y_i (i=1, ..., M).

Los algoritmos clasificadores tratan de encontrar, a partir de la información disponible en esa serie x_i observaciones, un modelo, patrón o estructura que explique, con un cierto grado de fidelidad, la variable *y* a partir de las *n* explicativas.

En términos estadísticos [HAI99], estos algoritmos entran dentro de las tareas de análisis de dependencia y que pueden definirse como "*aquellos en los que una variable o conjunto de variables se identifican como variables dependientes que van a ser explicadas por otras variables conocidas denominadas variables independientes*". Como contraste, un análisis interdependiente es aquel en el que ninguna variable o grupo de variables es definido como dependiente o independientes, y dónde el análisis del conjunto se realiza simultáneamente.

Los clasificadores pueden ser utilizados para:

- Determinar la relación causa-efecto de unas variables con respecto a otra, analizando
- Analizar cuales son las variables que más influyen en la variable a explicar para seleccionarlas como parámetros de entrada.
- Determinar agrupamiento de variables de entrada.
- Etc.

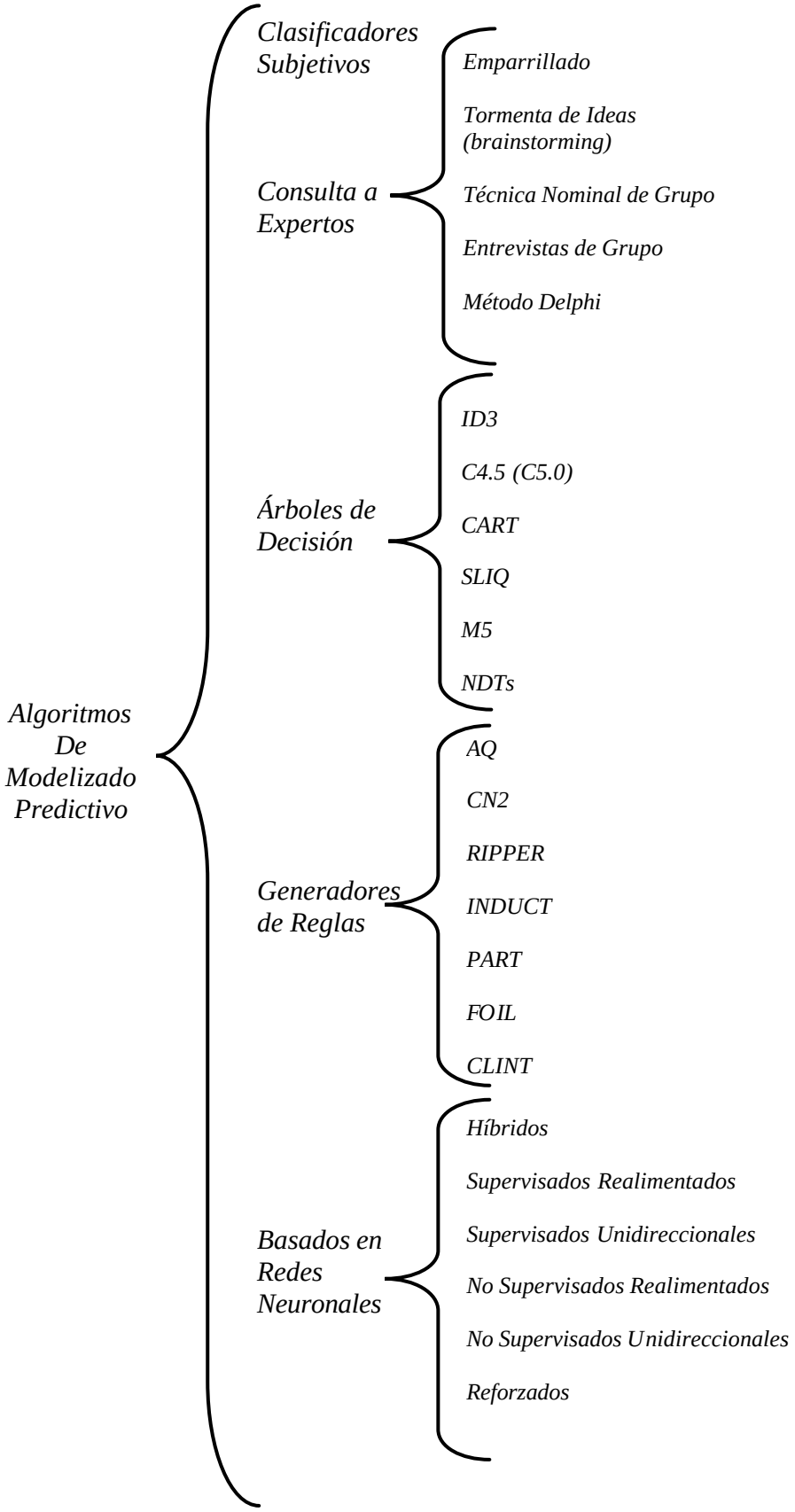


Figura 63. Algunos de los algoritmos de modelizado predictivo (I).

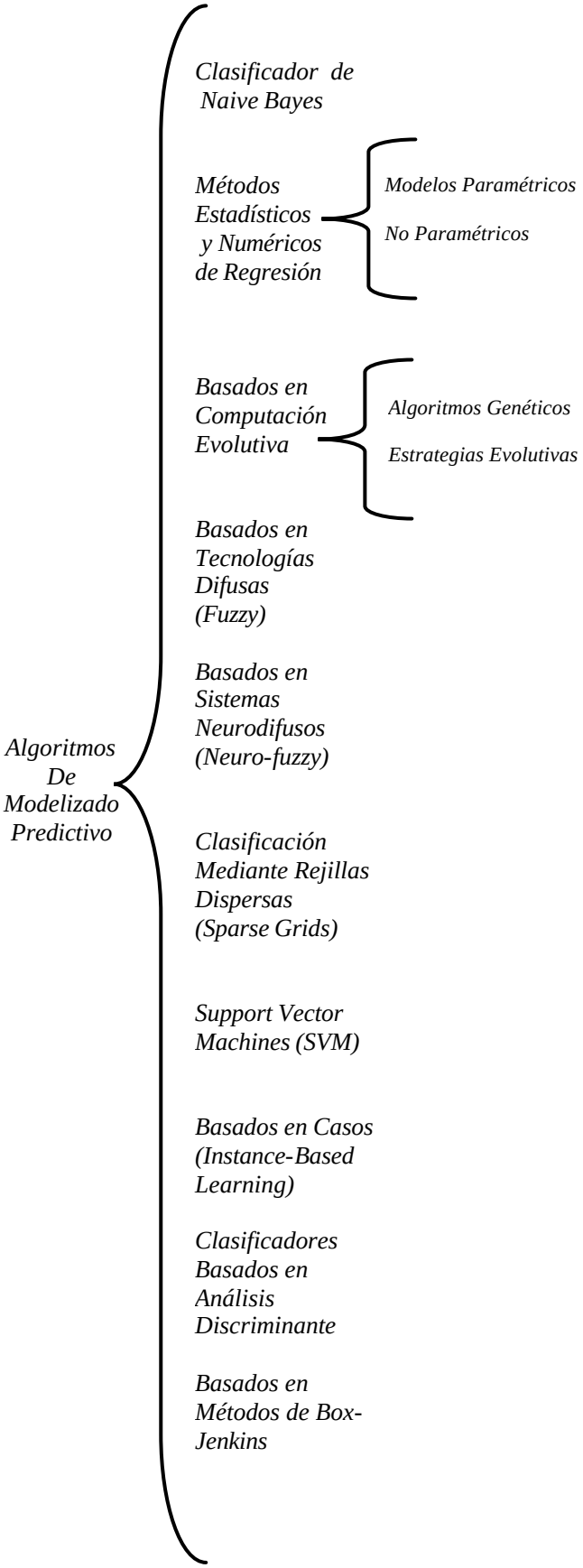


Figura 64. Algunos de los algoritmos de modelizado predictivo (y II).

Fundamentalmente se pueden dividir en:

- **Clasificadores:** Donde se genera un modelo que cada nueva observación la intenta clasificar en una de las clases previamente establecida. Por ejemplo, una empresa aseguradora necesita desarrollar, a partir de toda la información de sus clientes, un modelo clasificador que le indique si un nuevo asegurado, según la información del mismo, supone "poco riesgo" o "mucho riesgo" para la empresa. Dentro de este tipo de algoritmos, nos encontramos con: árboles de decisión, generadores de reglas, algunas redes neuronales utilizadas como clasificadores, etc.
- **Predicción del Valor:** En este caso, el modelo generado intenta, para cada nueva observación, predecir el valor continuo más probable. Por ejemplo, a partir de una serie de variables de un proceso térmico, queremos predecir la temperatura que va a alcanzar en un cierto momento. Dentro de este tipo de técnicas entran: regresión lineal o no lineal, redes neuronales utilizadas para generar modelos lineales o no lineales, etc.

3.4.4.1 LA METODOLOGÍA DE MODELIZADO Y VALIDACIÓN

La correcta selección de variables es muy importante en los problemas de modelado, ya que una mala elección puede dificultar e incluso impedir alcanzar resultados satisfactorios.

Tras la selección de las variables que van a intervenir en el modelado, es necesario un conjunto de datos que van a ser utilizados por la herramienta de modelado para obtener los nuevos modelos. También es necesario otro conjunto de datos para comprobar la validez de esos resultados. Para obtener dichos conjuntos es necesario un método de Selección de Muestras Representativas.

Para la Validación de Modelos no sólo es necesario un conjunto de datos sino también una estrategia que evite, en la medida de lo posible, los efectos derivados de modelar basándose en un conjunto finito de datos.

A la hora de abordar el diseño de un sistema, hay que tener en cuenta las distintas técnicas aplicables al problema planteado: selección de variables significativas, recogida y selección de los datos, generación de un modelo y sus parámetros y validación de ese modelo.

GENERALIDADES

Como se ha comentado, las técnicas y herramientas de data mining pueden ser empleadas con dos finalidades: predictiva y descriptiva. Los algoritmos se evalúan en función de la exactitud de predicción, grado de innovación de los resultados obtenidos, utilidad e interpretabilidad de la información obtenida.

Los algoritmos utilizables presentan tres componentes principales en su construcción que permiten caracterizarlos:

- Un lenguaje de representación.
- Unos métodos de búsqueda.
- Unos criterios de evaluación del modelo.

La integración de estas tres características conforma el algoritmo, diferenciándolo del resto. En los siguientes puntos se explicará con detalle cada una de estas características y las diferentes variedades existentes [ABA01].

Lenguaje de Representación del Modelo

Cada algoritmo usa un determinado lenguaje para representar el conocimiento. Así, dado un conjunto H de hechos (datos), un lenguaje L y alguna medida de la certidumbre C, el objetivo de los algoritmos de minería de datos consiste en encontrar sentencias S en L que describan relaciones dentro de un subconjunto Ḥ de H con una certidumbre c, de forma que S sea más sencillo que la enumeración de todos los hechos de H.

La característica global más importante de una representación del conocimiento es la eficiencia con la que lleva a cabo su tarea.

Otra característica interesante del lenguaje de representación es su generalidad, es decir, debe de servir para resolver una clase de problemas lo más amplia posible. Sin embargo, la eficiencia y la generalidad son objetivos contrapuestos, esto es, cuanto más eficiente es un método menor cantidad de problemas es capaz de resolver y viceversa. El conflicto entre generalidad y eficiencia suele resolverse optando por un compromiso o equilibrio entre ambas cualidades.

Un modelo $\boldsymbol{m_1}$ se dice que está sobreentrenado o sobreajustado si, dado un espacio de modelos $\boldsymbol{M}$, existe otro modelo $\boldsymbol{m_2}$ que obtiene mejores resultados que $\boldsymbol{m_1}$ sobre el conjunto completo donde va a ser implementado (población), obteniendo $\boldsymbol{m_1}$ mejores resultados sobre el conjunto de entrenamiento. Existen técnicas para evitar el sobreentrenamiento de los datos que serán tratadas en el punto evaluación del error.

En la construcción o selección del algoritmo de *data mining* deben de tenerse en cuenta las siguientes consideraciones:

- Las suposiciones de representación hechas sobre un algoritmo particular. Por ejemplo, si se considera la representación de un árbol de decisión mediante la división de nodos por un solo campo y particiones del espacio de entrada en hiperplanos paralelos a los ejes de los atributos, no se podría descubrir la fórmula $x=y$ en los datos.
- Representaciones muy generales del modelo son poco eficientes.
- Representaciones excesivamente adaptadas al problema implican riesgo de sobreentrenamiento.
- Las representaciones complejas mejoran el factor predictivo del modelo, pero aumentan la dificultad en la búsqueda y empeoran la interpretabilidad del modelo.

Métodos de búsqueda y optimización

Una vez fijada la representación del modelo (o familia de representaciones) y el criterio de evaluación del modelo, el problema puede ser reducido a un problema de optimización del tipo:

"*Encontrar los parámetros/modelos de la familia seleccionada que optimicen el criterio de evaluación del modelo*"

De modo general se pueden establecer cinco criterios de clasificación para los procedimientos de optimización:

Basándose en la naturaleza de las soluciones

- Numéricas: Si la solución queda completamente especificada en términos de un conjunto de ***m*** parámetros o atributos.
- Combinatorias: Si para especificar la solución no sólo hay que especificar un conjunto de ***m*** parámetros, sino también el orden (total o parcial) con que estos se combinan para dar dicha solución. En último término puede ocurrir que sea irrelevante la naturaleza de los atributos y sólo importe su orden. En tal caso se habla de problemas basados en el orden.

Basándose en el grado de aleatoriedad que se le da al proceso de búsqueda

- Deterministas o dirigidas: el procedimiento de búsqueda es completamente determinista, es decir, en las mismas condiciones de partida proporciona idénticos resultados. Las técnicas deterministas, por lo común son las más eficientes, pero son muy específicas y precisan mucho conocimiento adicional sobre la función objetivo así como el uso de hipótesis suplementarias de buen comportamiento, entre otros inconvenientes. Las técnicas clásicas de optimización, ya sean analíticas o enumerativas, son todas deterministas.
- Aleatorias o al azar: El procedimiento de búsqueda es completamente aleatorio. Habitualmente, se delimita una región de búsqueda y se toman puntos al azar dentro de ella. Después, mediante argumentos estadísticos, se puede dar una estimación de máxima verosimilitud para el valor del óptimo. Estas técnicas no requieren ninguna información adicional y se pueden aplicar a cualquier tipo de problemas, pero son poco eficientes.
- Estocásticas u orientadas: se combinan en proporción variable la búsqueda determinista con la búsqueda aleatoria. La componente determinista orienta la dirección de búsqueda y la aleatoria se encarga de la búsqueda local, buscando un punto intermedio entre la máxima eficiencia de las técnicas deterministas y la máxima eficacia de las aleatorias.

Basándose en la dirección preferente de búsqueda

- En Profundidad o explotadoras: la búsqueda da prioridad a la explotación de las soluciones disponibles antes que a la exploración de nuevas soluciones.
- En anchura o exploradoras: la búsqueda da prioridad a la exploración de nuevas soluciones antes que a la explotación de las disponibles.

En pocas palabras, la exploración trata de obtener más conocimiento del espacio de búsqueda y la explotación trata de aprovechar el que ya se tiene. Muchos métodos de búsqueda no son explotadores o exploradores puros, sino que combinan ambos procedimientos. De hecho es deseable tener control sobre la relación explotación-exploración. Idealmente, esa relación debería poderse expresar como un cociente entre dos parámetros al que se llama grado de penetración.

Basándose en el número de candidatos a solución que se mantienen simultáneamente

- Simple: se mantiene un sólo candidato a solución que se va actualizando sucesivamente para proporcionar, presumiblemente, soluciones cada vez más exactas del problema.
- Múltiples: se mantienen simultáneamente varios candidatos a solución con los cuales se va acotando cada vez con más precisión la región (o regiones) donde se encuentra los óptimos. Son las más apropiadas para implantaciones en paralelo. Aunque son computacionalmente más costosas, las búsquedas múltiples presentan tal cantidad de ventajas sobre las búsquedas simples que se deberían usar siempre que fuera posible, aunque no se disponga de procesadores en paralelo.

Basándose en la información disponible sobre la función a optimizar

- Ciegas: el proceso a optimizar funciona como una caja negra que ante ciertos valores de los parámetros devuelve un valor del objetivo. Es decir, no se dispone de ninguna información explícita sobre la aplicación *f(x)*. En la bibliografía inglesa, a estos métodos de optimización se les designa como *blackbox optimization*, que se puede traducir como optimización de cajas negras. La ventaja de estos métodos es que proporcionan algoritmos de búsqueda de propósito general, los cuales son muy fáciles de implantar para un problema específico.
- Heurísticas: se dispone de cierta información explícita sobre el proceso a optimizar, pudiéndose aprovechar dicha información para guiar la búsqueda. A dicha información útil para la búsqueda se le llama conocimiento específico. Las técnicas heurísticas proporcionan algoritmos dedicados de búsqueda, esto es, específicos para un problema concreto y difícilmente adaptables para cualquier otro. Tradicionalmente se ha tratado de añadir la máxima cantidad de conocimiento específico en los problemas de optimización, dado que es lo que más eficacia da a la búsqueda. Sin embargo, en problemas reales de mediana complejidad resulta muy difícil, cuando no imposible, encontrar tal información y la que se encuentra no suele ser de muy buena calidad. A efectos prácticos, el conocimiento especifico sólo sirve verdaderamente cuando es de buena calidad, y aun así se debe tener cuidado porque acentúa la tendencia a estancar la búsqueda en óptimos locales.

Criterios de evaluación del modelo

Para cada algoritmo de *data mining* es necesario establecer criterios de evaluación que proporcionen medidas de la diferencia entre el resultado del modelo y el proceso real. Para ello se distinguirá entre el error real y el error aparente.

Si se supone que existe una población de la cual han sido extraídos los datos de entrenamiento de forma aleatoria, se define el error real o error poblacional como el error que se comete al ser evaluado el modelo sobre los datos de la población.

Se define error aparente o error muestral como el error del modelo sobre la muestra de casos utilizados en el diseño o construcción del mismo.

El objetivo del modelo es minimizar el error real cometido que al ser desconocido debe de ser estimado. Todos los métodos de estimación se basan en la suposición de la existencia de un conjunto de entrenamiento, *T*, compuesto por *N* patrones. Este conjunto se utilizará para construir y evaluar el modelo, por lo que la existencia de un buen conjunto de aprendizaje resulta fundamental.

Para la estimación de los errores se utilizarán diferentes procedimientos en función de los objetivos del método. En particular, si el objetivo del método es realizar una regresión (la variable de salida es continua) es necesario el uso de una medida de distancia para realizar la evaluación del error. La figura siguiente, muestra las medidas de distancia más comúnmente utilizadas [ABA01] [PRU02]:

Minkowsky $D(x,y)=\sqrt[p]{\sum_{i=1}^{m}\left\|x_i-y_i\right\|^p}$	Euclídea $D(x,y)=\sqrt[2]{\sum_{i=1}^{m}\left\|x_i-y_i\right\|^2}$
Manhattan $D(x,y)=\sum_{i=1}^{m}\left\|x_i-y_i\right\|$	Camberra $D(x,y)=\sum_{i=1}^{m}\left\|\frac{x_i-y_i}{x_i+y_i}\right\|^p$
Chebychev o Maximum $D(x,y)=\max_{1,\ldots,m}\left\|x_i-y_i\right\|$	Cuadrática $D(x,y)=(x-y)^T\cdot Q(x-y)=\sum_{j=1}^{m}\left(\sum_{i=1}^{m}\left((x_i-y_i)\cdot q_{ji}\right)\left(x_j-y_j\right)\right)$ Q:matriz mxm de pesos definida positiva
Mahalanobis $D(x,y)=\sqrt[m]{\det(V)}\cdot(x-y)^T\cdot V^{-1}\cdot(x-y)$ V: matriz de covarianza de A_1, ..., A_m donde A_j es el vector de valores para el atributo j del conjunto de entrenamiento	Correlación $D(x,y)=\frac{\sum_{i=1}^{m}(x_i-\bar{x}_i)(y_i-\bar{y}_i)}{\sqrt{\sum_{i=1}^{m}(x_i-\bar{x}_i)^2\sum_{i=1}^{m}(y_i-\bar{y}_i)^2}}$
Chi-cuadrado $D(x,y)=\sum_{i=1}^{m}\frac{1}{suma_i}\left(\frac{x_i}{tam_x}-\frac{y_i}{tam_y}\right)^2$ $suma_i$ = suma de todos los valores del atributo i. tam_x = suma de todos los valores en el vector x.	Correlación del rango Kendall $D(x,y)=\frac{2}{n(n-1)}\sum_{i=1}^{m}\sum_{j=1}^{i-1}sign(x_i-x_j)\cdot sign(y_i-y_j)$ Sign(x)=signo de x

Figura 65. Algunas de las medidas de distancias más comunes.

Una vez seleccionada la medida de distancia a utilizar, se puede realizar una estimación del error del modelo mediante el cálculo de la distancia entre la salida predicha por el modelo y la salida real. Habitualmente, además de proporcionar esta estimación del error, se suelen fijar unas

tolerancias (11%, 5% , 10%, 15%, 20% y 25% del rango de la variable dependiente), calculando el porcentaje de aciertos obtenidos por el algoritmo con estas tolerancias.

En el caso de algoritmos de clasificación (variable dependiente discreta) los estimadores de error se basan en calcular la proporción de prototipos incorrectamente etiquetados por el clasificador. Una herramienta muy utilizada para la presentación y el análisis del resultado de una clasificación es la matriz de confusión, también llamada matriz de contingencia.

El estimador más simple del error real es el error aparente o error cometido sobre los patrones de entrenamiento. El problema fundamental de este estimador es que se calcula usando el mismo conjunto de patrones que el utilizado para construir el modelo, por lo que proporciona un estimador sesgado optimista de la bondad del modelo [WIT00].

Método entrenamiento prueba

Para solventar el problema de la dependencia entre el conjunto usado para construir el clasificador y el usado para realizar la estimación se puede dividir el conjunto inicial de patrones, T, en dos conjuntos independientes T_a y T_p de forma que:

- Los patrones de T_a, constituyen el conjunto de aprendizaje y se usan únicamente para construir el modelo.
- Los patrones de T_p constituyen el conjunto de validación y se usan únicamente para estimar el error.

Para que resulte eficaz debe de asegurarse que los patrones de T_a sean independientes de los de T_p pero que sigan la misma distribución. La manera habitual de asegurar estas condiciones es realizar una partición de T seleccionando los patrones aleatoriamente, de forma que $T_a \cup T_p = T$ y $T_a \cap T_p = \varnothing$. Este conjunto suele se suele seleccionar de forma que T_a esté compuesto por 2/3 de T, y T_p sea el otro 1/3 de T.

Una vez construido el modelo a partir de los patrones de T_a, se calcula el porcentaje de fallos del modelo que proporciona el estimador de error sobre todos los patrones de T_p (conjunto de prueba).

El inconveniente (en ocasiones muy serio) de este estimador es que se reduce el tamaño efectivo del conjunto de aprendizaje a 2/3 de su tamaño inicial. Para conjuntos con numerosos patrones esto no supone un gran problema pero si no se dispone de un conjunto numeroso, la construcción de un modelo consistente puede estar realmente comprometida. En estos casos puede emplearse el estimador por validación cruzada o estimadores del error tipo *bootstraping*.

Método del estimador de validación cruzada

El estimador por validación cruzada con V conjuntos, distribuye aleatoriamente los patrones de *T* en *V* conjuntos disjuntos T_1, T_2, … T_v de un tamaño similar de forma que, siendo *card(T)* el cardinal o tamaño del conjunto *T*:

$$\underset{i=1,\ldots,v}{card}(T_i) \approx \frac{car(T)}{V} \tag{3.57}$$

El procedimiento de estimación puede plantearse como sigue:

Para todo *v*, con *v* = 1, 2, ..., *V*, construir un modelo, que se denotará por f_v, usando T-T_v como conjunto de aprendizaje. Como ninguno de los patrones de T_v se ha usado para construir f_v, este conjunto puede ser utilizado para realizar una estimación del error del modelo mediante el conjunto de prueba.

Al finalizar este paso se obtienen *V* modelos, f_v con sus correspondientes estimaciones de error.

Usamos el mismo procedimiento, construir el modelo f_v usando todos los patrones de T.

Los errores obtenidos en *v*=1 se usan para estimar el error *v*=2 como valores de una distribución normal. Para valores grandes de *V*, cada uno de los *V* clasificadores se construye usando un conjunto de patrones de tamaño aproximado a *N(1-1/V)*, casi tan grande como b es *T*. La suposición básica de la validación cruzada, es que es un procedimiento estable ya que todos los modelos f_v desarrollados con casi todos los patrones de *T*, tienen una tasa de error aproximadamente igual a la del modelo buscado (construido con todos los patrones de T).

Cuando *V*=*N*, el estimador por validación cruzada con *N* conjuntos, se conoce como el estimador que deja uno fuera, del término inglés leave-one-out, [CRA79]. Para cada *n*, (con n = 1, 2,..., N) el enésimo patrón es descartado y el clasificador se construye utilizando los restantes N-1 patrones. Entonces, el patrón descartado se usa para prueba y se estima el error.

El gran inconveniente de este estimador es el gran esfuerzo computacional que requiere ya que todos los patrones de ***T*** se usan para construir ***f***, y cada uno de ellos se usa exactamente una vez para prueba. Los estimadores de validación cruzada son estimadores centrados pero con una gran varianza.

Método del estimador por *bootstrapping*

El estimador b*ootstrap* utiliza técnicas de reemplazo para construir un conjunto de patrones de aprendizaje T_a de igual tamaño que el conjunto original de patrones disponibles: Card(T_a)=Card(*T*)=*N*.

Para la construcción del conjunto de patrones T_a se extraen con reemplazamiento *N* patrones aleatorios. El conjunto de patrones no representados en el conjunto T_a constituye el conjunto de

patrones de prueba T_p. De esta forma, el conjunto T_a contiene j patrones distintos del conjunto T, mientras que el conjunto T_p contiene los N-j patrones restantes. Mediante simulaciones de Montecarlo se ha estimado que el valor esperado de j es del 63,2 %, siendo por lo tanto la esperanza de N-j el 36,8% restante: $E(j)$=63,2%, $E(N-j)$=36,8%.

Una vez generados los conjuntos de patrones de entrenamiento y prueba se procede a entrenar el algoritmo de *data mining* con el conjunto de patrones de entrenamiento, realizando una estimación del error real con el conjunto de patrones prueba. Este proceso se repite de forma iterativa k veces, obteniendo k estimaciones del error real (empíricamente se ha comprobado que para una muestra de 30 datos son necesarias aproximadamente 200 iteraciones para que las estimaciones sean satisfactorias). El error real se calcula en función de estos k errores, mediante técnicas de resumen de datos (error medio, error cuadrático, etc.), obteniéndose que para muestras de tamaño pequeño la estimación del error real está sesgada pesimistamente.

El estimador *0,632 bootstrap*, se construye considerando una combinación lineal entre el error *e0 bootstrap* y el error aparente:

$$0{,}632 \cdot B = 0{,}368 \cdot \psi + 0{,}632 \cdot e0 \qquad (3.58)$$

donde:

- ψ es la tasa de error aparente sobre todos los casos (tanto de entrenamiento como de prueba).
- e0 es la estimación *bootstrap e0*.
- Este tipo de estimación del error es sesgada optimistamente, por lo que a partir del estimador *e0 bootstrap* y del estimador .632B se pueden obtener las cotas superior e inferior del error real.

Los estimadores por *bootstrapping*, son sesgados y tienen una varianza baja, pudiendo ser utilizados como alternativa a los estimadores de validación cruzada.

A continuación se pasará a describir algunas de las técnicas y algoritmos de modelizado predictivo.

3.4.4.2 CLASIFICADORES SUBJETIVOS

Es el método más intuitivo. Se basa en la información previa del sistema y en la selección por parte de las variables que considera más importantes [ABA01][DVO91]. Este método tiene la ventaja de la sencillez y la desventaja de que no tiene ningún fundamento matemático.

A pesar de no tener ningún respaldo matemático, es útil para reducir un conjunto inicial muy elevado de variables a un subconjunto de variables candidatas. Para esta selección manual se suelen utilizar herramientas de visualización multidimensional de los datos, que permiten hacerse una idea más clara de la "forma" de los mismos.

3.4.4.3 CONSULTA A EXPERTOS

Una de las formas más comunes y lógicas de desarrollar clasificadores o de seleccionar las variables de entrada, se basa en el aprovechamiento del conocimiento de los expertos del sistema.

Generalmente, es uno de los primeros pasos que deben realizarse en el proceso de *data mining*.

Algunas de las más conocidas técnicas de consulta a expertos son [MAO02]:

- **Emparrillado**: Que consiste en escoger objetos que sean relevantes y elaborar distinciones bipolares que destaquen las diferencias que el experto tiene en su modelo mental. Para ello, cada experto da valores de 1 a *n* a cada característica bipolar y posteriormente se contrastan para poder definir el modelo mental.
- **Tormenta de ideas (*brainstorming*):** Se plantean las ideas individuales de cada experto, se discuten aportando comentarios, se adoptan las ideas ajenas y se desarrollan, se seleccionan las mejores según criterios de tiempo, costes, técnica, disponibilidad, etc. Por último, se rellenan fichas, se distribuyen aleatoriamente y se critican.
- **Técnica Nominal de Grupo:** Se generan las ideas por escrito, se hace una lista de ideas (por tema u otros criterios), se discuten y se votan.
- **Entrevistas de grupo**: Un coordinador dirige la entrevista con unas reglas de funcionamiento definidas, una discusión dirigida y un sumario de respuestas que hay que rellenar.
- **Método *Delphi*** [DEL02].

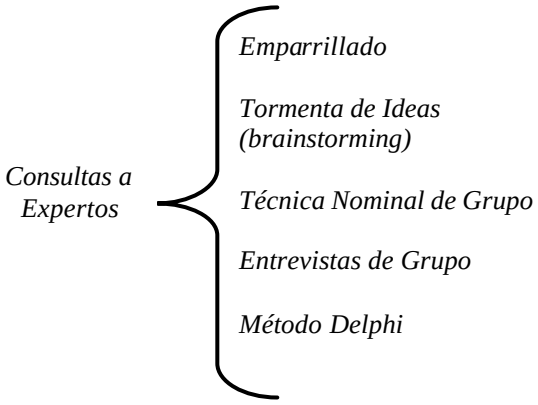


Figura 66. Clasificación de algunas de las técnicas de consultas a expertos.

EL MÉTODO DELPHI

El método *Delphi* pretende extraer y maximizar las ventajas que presentan los métodos basados en grupos de expertos y minimizar sus inconvenientes. Para ello se aprovecha la sinergia del debate en el grupo y se eliminan las interacciones sociales indeseables que existen dentro de todo grupo. De esta forma se espera obtener un consenso lo más fiable posible del grupo de expertos.

Es una técnica de previsión cualitativa donde se combina la opinión de los expertos en una serie de iteraciones. El resultado de cada iteración se utiliza para desarrollar la siguiente y obtener el conjunto de las opiniones de los expertos. Este método presenta tres características fundamentales:

- **Anonimato**: Durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:
  - Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
  - Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
  - El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- **Iteración y realimentación controlada**: La iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.

- **Respuesta del grupo en forma estadística**: La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.
    - En la realización de un Delphi aparece una terminología específica:
    - Circulación: Es cada uno de los sucesivos cuestionarios que se presenta al grupo de expertos.
    - Cuestionario: El cuestionario es el documento que se envía a los expertos. No es sólo un documento que contiene una lista de preguntas, sino que es el documento con el que se consigue que los expertos interactúen, ya que en él se presentarán los resultados de anteriores circulaciones.
    - Panel: Es el conjunto de expertos que toma parte en el Delphi.
    - Moderador: Es la persona responsable de recoger las respuestas del panel y preparar los cuestionarios.
- **Fases de Realización**
    - Antes de iniciar un Delphi se realizan una serie de tareas previas, como son:
    - Delimitar el contexto y el horizonte temporal en el que se desea realizar la previsión sobre el tema en estudio.
    - Seleccionar el panel de expertos y conseguir su compromiso de colaboración. Las personas que sean elegidas no sólo deben ser grandes conocedores del tema sobre el que se realiza el estudio, sino que deben presentar una pluralidad en sus planteamientos. Esta pluralidad debe evitar la aparición de sesgos en la información disponible en el panel.
    - Explicar a los expertos en qué consiste el método. Con esto se pretende conseguir la obtención de previsiones fiables, pues van los expertos van a conocer en todo momento cuál es el objetivo de la cada una de los procesos que requiere la metodología.

En un Delphi clásico se pueden distinguir cuatro circulaciones o fases:

- **Primera circulación.** El primer cuestionario es desestructurado, no existe un guión prefijado, sino que se pide a los expertos que establezcan cuáles son los eventos y tendencias más importantes que van a suceder en el futuro referentes al área en estudio. Cuando los cuestionarios son devueltos, éste realiza una labor de síntesis y selección, obteniéndose un conjunto manejable de eventos, en el que cada uno está definido de la forma más clara posible. Este conjunto formará el cuestionario de la segunda circulación.
- **Segunda circulación.** Los expertos reciben el cuestionario con los sucesos y se les pregunta por la fecha de ocurrencia. Una vez contestados, los cuestionarios son devueltos al moderador, que realiza un análisis estadístico de las previsiones de cada

evento. El análisis se centra en el cálculo de la mediana (año en que hay un 50% de expertos que piensan que va a suceder en ese año o antes), el primer cuartil o cuartil inferior (en el que se produce lo mismo para el 25% de los expertos) y tercer cuartil o cuartil superior (para el 75%). El moderador confecciona el cuestionario de la tercera circulación que comprende la lista de eventos y los estadísticos calculados para cada evento.

- **Tercera circulación**. Los expertos reciben el tercer cuestionario y se les solicita que realicen nuevas previsiones. Si se reafirman en su previsión anterior y ésta queda fuera de los márgenes entre los cuartiles inferior y superior, deben dar una explicación del motivo por el que creen que su previsión es correcta y la del resto del panel no. Estos argumentos se realimentarán al panel en la siguiente circulación. Al ser estos comentarios anónimos, los expertos pueden expresarse con total libertad, no estando sometidos a los problemas que aparecen en las reuniones cara a cara. Cuando el moderador recibe las respuestas, realiza de nuevo el análisis estadístico y, además, organiza los argumentos dados por los expertos cuyas previsiones se salen de los márgenes intercuartiles. El cuestionario de la cuarta circulación va a contener el análisis estadístico y el resumen de los argumentos.
- **Cuarta circulación**. Se solicita a los expertos que hagan nuevas previsiones, teniendo en cuenta las explicaciones dadas por los expertos. Se pide a todos los expertos que den su opinión en relación con las discrepancias que han surgido en el cuestionario. Cuando el moderador recibe los cuestionarios, realiza un nuevo análisis y sintetiza los argumentos utilizados por los expertos.

Figura 67. Fases del método Delphi [MAO02].

Teóricamente, ya habría terminado el Delphi, quedando tan sólo la elaboración de un informe en el que se indicarían las fechas calculadas a partir del análisis de las respuestas de los expertos y los comentarios realizados por los panelistas. Sin embargo, si no se hubiese llegado a un consenso, existiendo posturas muy distantes, el moderador debería confrontar los distintos argumentos para averiguar si se ha cometido algún error en el proceso.

Este método permite determinar cuáles son los elementos dentro de un grupo que tienen más importancia o son más prioritarios. Un ejemplo de aplicación es la priorización de variables relevantes en un proceso. Para aplicar este método se debe tener en cuenta que los elementos que se van a considerar deben ser comparables. Una vez que se tiene la lista de elementos se utiliza una semi-matriz para comparar cada elemento con los demás que pertenecen al grupo, colocando en el lugar correspondiente el número de puntos obtenidos mediante la votación de cada uno de los participantes en la sesión de priorización de los elementos. Al finalizar se suman los puntos obtenidos por cada uno de las variables y se seleccionan aquellas que obtuvieron mayor votación.

3.4.4.4 ÁRBOLES DE DECISIÓN

Los árboles de decisión son unos de los algoritmos clasificadores más conocidos y usados en las tareas de *data mining*[WIT00][DAE02], ya que son una forma de representación sencilla para clasificar ejemplos de un número finito de clases. Se basan en la partición del conjunto de ejemplos según ciertas condiciones que se aplican a los valores de las características. Su potencia descriptiva viene limitada por las condiciones o reglas con las que se divide el conjunto de entrenamiento.

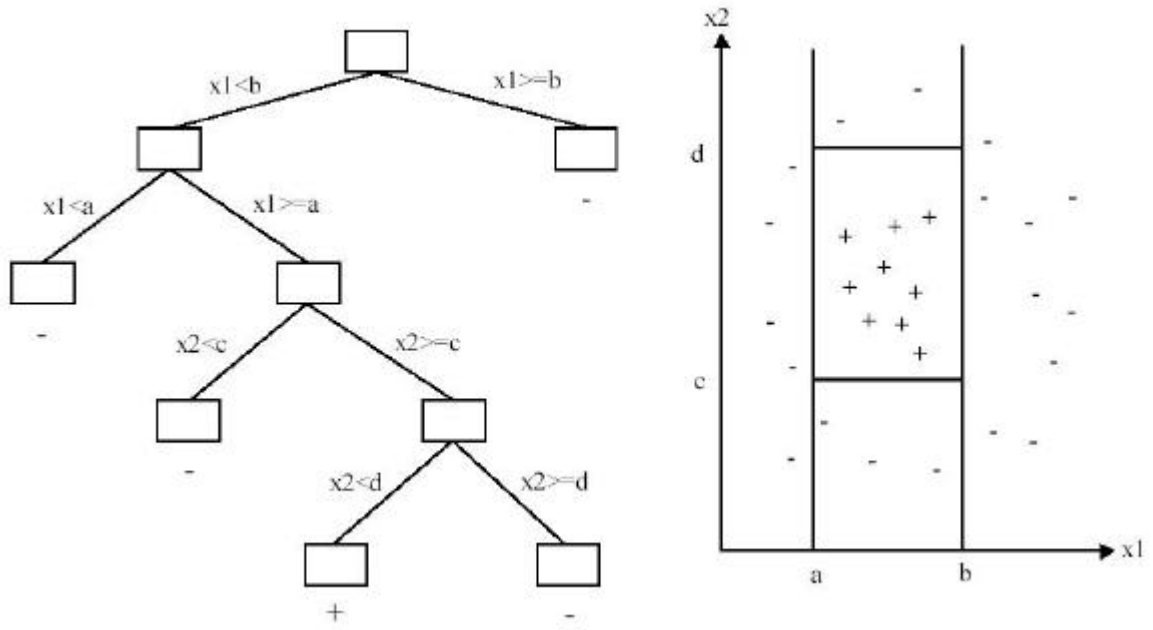


Figura 68. División del espacio para clasificar dos clases: menos y cruces.

Los árboles de decisión efectúan la clasificación utilizando un árbol construido durante el entrenamiento, el cual incorpora codificada toda la información de la partición *M* espacio. Tras la clasificación, un patrón es asignado a una clase determinada si cumple todos los criterios a lo largo de su camino dentro *M* árbol. Los árboles son capaces de codificar tanto valores simbólicos como numéricos, de forma explícita.

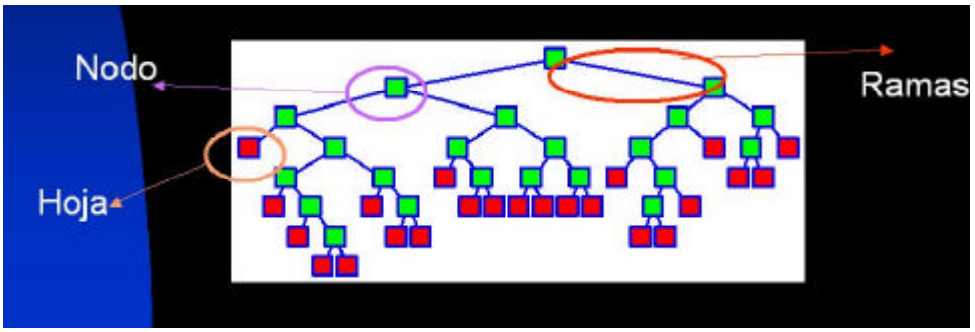


Figura 69. Partes más importantes de un árbol.

Los árboles están compuestos de nodos que nunca se unen en bucle cerrado. Se suele distinguir entre nodos terminales (pertenecientes al último nivel), nodo raíz (inicial) y nodos

intermedios o no terminales. También se establecen relaciones entre ellos, de modo que se denomina padre de un nodo a su antecesor y descendiente a cualquiera de los que dependen de él. Cada nodo tiene asociada una regla, de modo que se sigue hacia uno u otro descendiente en función de la respuesta a ésta, hasta que se llega a un nodo terminal, donde se efectúa realmente la clasificación.

El desarrollo de los árboles de decisión se basa, inicialmente, en el método "divide y conquistarás". Para ello, en cada nodo se va interrogando a cada atributo y según el valor del mismo, se va particionando el eje espacial que representa, de tal forma, que va generando subzonas del espacio muestral.

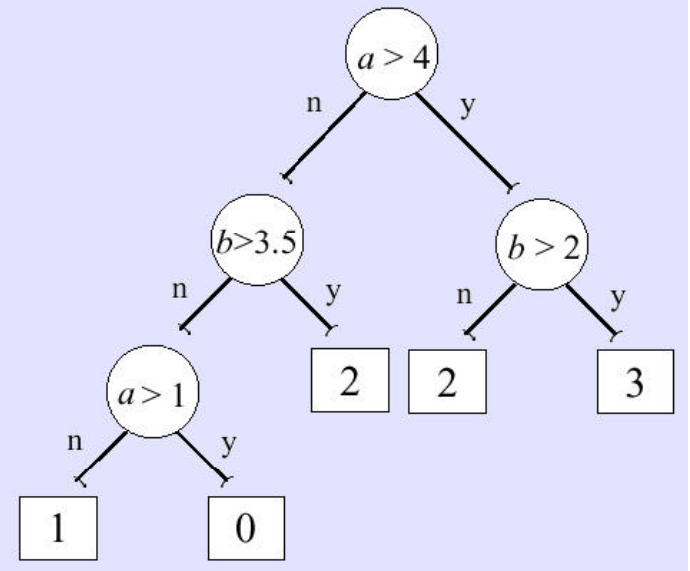


Figura 70. Árbol clasificador de cuatro clases a partir de dos atributos.

Generalmente, en cada nodo se evalúa cada atributo según una constante (ver Figura 70) aunque algunos árboles clasifican atributos con valores nominales, o comparan dos o más atributos con otro, o utilizan alguna función que incluye uno o varios atributos [WIT00].

De esta forma, si el atributo que se chequea es nominal, el número de hijos que salen de ese nodo será igual al de posibles valores de esa variable. En cambio, si el valor del mismo es continuo, lo más usual es que se divida en dos caminos: si es menor o igual que una constante, o si es mayor; aunque eso no signifique que se utilicen otros criterios de decisión que generen, en cada nodo, tres o más ramas.

En conclusión podemos encontrarnos con:

- Árboles que predicen una clase determinada a partir de atributos numéricos, nominales o ambos. Generalmente denominados árboles clasificadores.
- Árboles que predicen un valor numérico a partir de atributos numéricos, nominales o ambos. Llamados árboles de regresión.

El algoritmo general de creación de un árbol de decisión viene dado por los cuatro pasos siguientes:

- Paso 1: Asignar al nodo raíz todos los datos de entrenamiento (conjunto que se denotará por *T*).
- Paso 2: Si todos los elementos *M* nodo raíz pertenecen a la misma clase parar, en caso contrario ir al paso 3.
- Paso 3. Seleccionar una característica *X* con clases C_1, C_2, ... C_N distintas, creando *N* nodos T_1, T_2, ..., T_N descendientes del nodo padre *T*, de forma que T_i sea una partición de *T* en función de la regla derivada de la clase C_i.
- Paso 4: Para cada T_i hacer $T=T_i$ e ir al paso 2.

El problema de construir árboles de decisión es así expresado recursivamente. Primero, se selecciona un atributo como nodo principal o raíz del árbol. A partir del mismo, se divide el conjunto de observaciones en dos o más subseries de datos según el valor del atributo, y se repite recursivamente para cada rama hasta que en cada nodo.

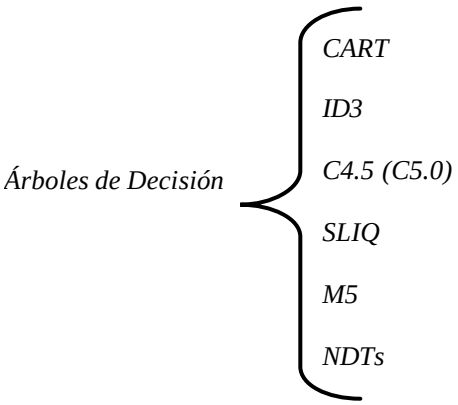


Figura 71. Clasificación de los principales algoritmos generadores de árboles de decisión.

Los árboles de decisión generados por el algoritmo anterior llegan usualmente a una solución demasiado ajustada al conjunto de patrones de aprendizaje, presentando componentes excesivamente específicos originados por la presencia de *outliers* o ruido en el conjunto de patrones de aprendizaje. Para subsanar el anterior problema se puede detener el crecimiento del árbol mediante algún criterio de parada o podar las ramas de los árboles que clasifican los casos menos numerosos (*pruning*).

Las principales diferencias entre los distintos algoritmos de construcción de árboles de decisión radican en las estrategias de poda y en la regla adoptada para dividir los nodos.

El primer paso de construcción de un árbol, consistirá en seleccionar el atributo más idóneo. Éste será aquél que genere hijos más puros, es decir, subgrupos donde las clases estén más definidas, es decir, las particiones dividen un conjunto de patrones en conjuntos disjuntos, con el fin

de incrementar la homogeneidad (en términos de clase) de los subconjuntos resultantes, o lo que es lo mismo, que los subconjuntos sean más puros que el conjunto originario.

Esta medida de pureza generalmente se denomina *medida de información* o *función de impureza* y las unidades de medida *bits*. Asociado con un nodo de un árbol, representa la esperada cantidad de información que deberíamos necesitar para especificar si una nueva serie de atributos puede ser clasificado dentro de una clase u otra.

FUNCIÓN DE IMPUREZA Y MEDIDA DE IMPUREZA

En primer lugar definiremos lo que se entiende por función de impureza.

Se dice que una función f definida sobre J-uplas de la forma (c_1, c_2,..., c_J) tales que:

$c_J \geq 0$ para j=1, 2, ..., J

$$\sum_{j=1}^{J} c_j = 1$$

es una *función de impureza*, si se verifica las siguientes propiedades:

- f alcanza su único máximo en $\left(\frac{1}{J}, \frac{1}{J}, ..., \frac{1}{J}\right)$
- f alcanza su alcanza su mínimo en (1, 0, 0,..., 0),(0, 1, 0,..., 0),...,(0, 0, 0,..., 1) y el valor del mínimo es 0.
- f es una función simétrica de c_1, c_2,..., c_J.

Dada una función de impureza f, definimos *la medida de impureza de cualquier nodo* **t**, y se escribe *i(t)* como:

$$i(t) = \Phi\big(p(1|t), p(2|t), ..., p(J|t)\big) \tag{3.59}$$

donde *p(j | t)* es la probabilidad de que un caso del nodo *t* (un patrón asociado al nodo *t*) sea de clase j. Estas probabilidades pueden calcularse empíricamente como la proporción de casos de clase *j* en el nodo *t*:

$$p(j|t) = \frac{N_j(t)}{N(t)} \tag{3.60}$$

Dicho de otra forma, la medida de impureza de un nodo es el resultado de evaluar la función de impureza sobre ese nodo tomando las proporciones relativas de cada clase como los c_j. Se puede observar que:

$$p(j|t) \geq 0 \tag{3.61}$$

$$\sum j \cdot p(j|t) = \sum j \cdot \frac{N_j(t)}{N(t)} = \frac{1}{N(t)} \sum j \cdot N_j(t) = 1 \tag{3.62}$$

Lo que garantiza que los componentes de la *J-upla*, calculados en términos de proporción relativa son válidos para evaluar la función de impureza.

Por otro lado, se puede deducir que:

- La máxima impureza se obtiene cuando todas las clases están igualmente representadas en *t*.
- La mínima impureza se obtiene cuando en *t* sólo hay casos de una sola clase (máxima homogeneidad).
- Cualquier permutación de los c_j produce el mismo resultado en el valor de impureza.

BONDAD DE UNA PARTICIÓN

La bondad de la partición *s* en un nodo *t*, *f(s, t)*, se define como el decrecimiento en impureza conseguido con ella:

$$\Phi(s,t) = \Delta i(s,t) = i(t) - \sum_{j=1}^{J} p_j \cdot i(t_j) \tag{3.63}$$

Para establecer el efecto que produce la selección de la mejor partición en cada nodo sobre el árbol final necesitamos una medida de la impureza global del árbol. Así, si conocemos cómo calcular *i(t)*, podemos calcular *f(s, t)*, para cada partición *s* y seleccionar la mejor partición como la que proporciona la mayor bondad *f(s, t)*.

El procedimiento general de poda consiste en construir un árbol muy grande, T_{max} donde los nodos terminales sean perfectamente homogéneos o están asociados a pocos patrones. Una vez obtenido T_{max} se tratará de podar este árbol, lo que significa la sustitución de una parte del árbol (sub-árbol) por una hoja. La poda tendrá lugar si el valor esperado de error en el sub-árbol es mayor que con el nodo que lo sustituya. Si *T'* se obtiene a partir de *T* por poda, *T'* es un subárbol podado de *T* denominándose como $T' \prec T$. Procediendo de forma recursiva se obtiene una sucesión decreciente y anidada de árboles: $T_{\max} \succ T_1 \succ T_2 \succ \ldots$ de entre los que se selecciona el árbol que tenga asociado menor error.

Según el tipo de medida de impureza que se use y el algoritmo para generar el árbol, podemos encontrarnos con diferentes tipo de métodos de generación de árboles. A continuación se explican algunos de los más importantes.

CART (CLASSIFICATION AND REGRESSION TREES)

El método CART (*Classification And Regression Trees*) [BRE84] es un algoritmo de árbol de decisión que realiza particiones binarlas con una estrategia de poda basada en un criterio de coste-complejidad.

Las particiones binarlas son el resultado de evaluar una condición que tiene dos únicas respuestas. La formulación de la regla de partición se realiza a partir del conjunto estándar de preguntas. En CART se define el conjunto estándar de preguntas, Q, de la siguiente forma:

Cada partición depende de un único atributo.

Si X_i es un atributo categórico, que toma valores en { c_1, c_2, …, c_L } , Q incluye las preguntas: ¿X_i $\hat{I}$ C? donde C es un conjunto de entre los subconjuntos de { c_1, c_2, ..., c_L }. Si X_i es un atributo continuo, Q incluye preguntas del tipo: ¿ X_i £ v ? donde v es un valor real cualquiera. No obstante, en orden de simplificar los cálculos permitiendo que los problemas sean tratables, el algoritmo CART toma v como el punto medio de dos valores consecutivos de X_i.

Cada pregunta da lugar a una partición con una medida de impureza asociada. La función de impureza utilizada por el algoritmo CART es el índice de *Gini*, que para un nodo dado t, se define como:

$$i(t) = Gini(t) = \sum_{\substack{i,j=1 \\ i \neq j}}^{J} p(i \mid t) \cdot p(j \mid t) = 1 - \sum_{j=1}^{J} p(j \mid t)^2 \tag{3.64}$$

Para cada nodo, CART determina la mejor partición para cada atributo, seleccionando la mejor partición, como aquella que cause la mayor reducción de impureza. De esta forma se construye un árbol muy grande, T_{max}. donde los nodos terminales sean perfectamente homogéneos o estén asociados a pocos patrones.

Una vez obtenido T_{max} se procede a la poda de este árbol mediante la construcción de una secuencia decreciente y anidada de árboles: $T_{\max} = T_0 \succ T_1 \succ T_2 \succ ... \succ \{t_1\}$ de manera que el árbol $\{t_i\}$ sea un árbol que conste de un único nodo.

Cada árbol podado de esta sucesión, T_{i+1} se construye seleccionando de entre todos los subárboles de T_i, el sub-árbol con una menor medida de coste-complejidad asociada. Es decir, se penalizan los árboles de decisión con un alto coste de complejidad, donde la medida del coste de complejidad para un determinado sub-árbol T, viene dada por.

$$R_a(T) = R(T) + a\left|\tilde{T}\right| \tag{3.65}$$

donde:

- *R(T)* es el error de clasificación asociado al árbol *T*.
- α es un valor real ($\alpha \geq 0$) (parámetro de complejidad) que se interpreta como el coste de complejidad por nodo terminal.
- $\left|\tilde{T}\right|$ es la complejidad del subárbol, $T \prec T_{max}$ que se define como el número de nodos terminales de *T*.

Así, la medida de coste-complejidad es una combinación lineal del coste del árbol y su complejidad, ponderada apropiadamente.

Una vez construida la sucesión decreciente y anidada de árboles, se calcula sobre cada árbol de la sucesión la tasa de error de clasificación para un conjunto de patrones de prueba independiente del conjunto de patrones utilizado en el entrenamiento, y así, se selecciona el árbol con menor tasa de error asociada.

ID3 (INTERACTIVE DICHOTOMIZER) O TDIDT (TOP-DOWN INDUCTION OF DECISION TREES)

El algoritmo ID3 (*Interactive Dichotomizer*) [QUI86] también llamado TDIDT (*Top-Down Induction of Decisión Trees*) [MIC98] se basa en la utilización de la entropía como función de impureza. La entropía, también denominada valor de información, se define como la medida de la incertidumbre que hay en un sistema, es decir, ante una determinada situación, la probabilidad de que ocurra cada uno de los posibles resultados.

Su expresión para un nodo *t* es:

$$S(t) = i(t) = \sum_{j=1}^{J} p(j \mid t) \cdot \log p(j \mid t) \tag{3.66}$$

A cada nodo se le asocia aquel atributo con mayor decrecimiento en la función de impureza que aún no haya sido considerado en la trayectoria desde la raíz.

El ID3 es capaz de tratar con atributos cuyos valores sean discretos o continuos. En el primer caso, el árbol de decisión generado tendrá tantas ramas como valores posibles tome el atributo, por otro lado, si los valores del atributo son continuos, el ID3 no clasifica muy adecuadamente los ejemplos dados.

Otro inconveniente del algoritmo ID3 es su predisposición a favorecer indirectamente a aquellos atributos con muchos valores, los cuales no necesariamente tienen por qué ser los más útiles.

Por todos estos inconvenientes, *Quinlan* (1993) propuso el C4.5, como extensión y ampliación del ID3 [QUI93].

C4.5

C4.5 [QUI93] [WIT00], y su posterior versión comercial C5.0, es una extensión de ID3, que incluye múltiples mejoras como por ejemplo:

- Construye árboles de decisión cuando algunos de los ejemplos presentan valores desconocidos en algunos de los atributos.
- Puede trabajar con atributos que presenten valores continuos.
- Tolerancia a datos con ruido.
- Genera reglas a partir de árboles.

La función de impureza utilizada en el algoritmo C4.5 viene dada por:

$$TasaGanancia(t) = i(t) = \frac{Gan(t)}{Infpart(t)} \tag{3.67}$$

Donde:

$Gan(t) = \Delta i(t) = S(t) - \sum_{j=1}^{J} p_j \cdot S(t_j) \Rightarrow$ Información ganada en la función impureza de entropía.

$Infpart = -\sum_{j=1}^{J} p_j \cdot \log p_j \Rightarrow$ Información de la partición.

Para cada nodo, C4.5 determina la mejor partición para cada atributo, considerando como mejor partición la que cause mayor reducción de impureza. De esta forma se construye un árbol muy grande, T_{max}, donde los nodos terminales sean perfectamente homogéneos o estén asociados a pocos patrones.

Una vez obtenido T_{max} se procede a la poda de este árbol mediante la eliminación de los nodos hijos que no resulten significativamente distintos de su nodo padre. La prueba de significancia entre los nodos hijos con su padre, puede realizarse mediante el cálculo de errores estándar sobre cada nodo, no siendo por lo tanto necesaria la utilización de un conjunto de patrones de prueba independiente del conjunto de patrones de entrenamiento. Sin embargo, es recomendable el uso de patrones de prueba puesto que en general se obtienen mejores resultados.

Fundamentalmente, la versión comercial C5.0 con respecto a la de código libre C4.5 no presenta diferencias sustanciales, si se exceptúa la mejora de la velocidad para generar reglas y los árboles.

SLIQ

El algoritmo SLIQ (Supervised Learning In Quest) [MEH96] es un árbol de decisión ideado para abordar problemas con grandes cantidades de datos. Para la fase de construcción del árbol T_{max} utiliza la misma función de impureza que el algoritmo de CART, es decir el índice de *Gini*.

El esquema utilizado por SLIQ en la fase de poda, es un esquema alternativo a los vistos en los otros métodos y que se fundamenta en el principio de longitud de descripción mínima, MDL. El MDL establece que el coste total de la codificación de unos datos D mediante un modelo M viene dado por:

$$\cos t(M, D) = \cos t(D \mid M) + \cos t(M) \tag{3.68}$$

donde:

- cost($D \mid M$), corresponde al coste en bits, de codificar los datos dado un modelo M.
- cost(M), que corresponde con el coste de codificar el modelo M.

Es decir, el coste total de codificación es la suma del coste de describir los datos en términos del modelo más el coste de describir el modelo. Dados dos modelos alternativos, el principio MDL establece como mejor modelo, aquel que tenga un menor coste de descripción asociado.

En el contexto de los árboles de decisión, se considera que los modelos son el conjunto de árboles obtenidos mediante la poda del árbol inicial, y los datos son el conjunto de patrones de entrenamiento. El objetivo de la poda MDL es encontrar el subárbol de T que mejor describa al conjunto de patrones de entrenamiento.

Existen dos componentes en el algoritmo de poda MDL:

- El esquema de codificación que determina el coste de codificar los datos y el modelo.
- El algoritmo usado para comparar varios subárboles de T.

Codificación de los datos y del modelo

El coste de codificar el conjunto de patrones de entrenamiento mediante un árbol de decisión T se define como la suma de todos los errores de clasificación, donde el conteo del número de errores de clasificación se realiza durante la fase de construcción.

Codificación del modelo

El coste de codificación del modelo viene dado en función del coste de codificación del árbol, más el coste de la descripción de las preguntas usadas para cada nodo interno del árbol:

• Codificación del árbol: Dado un árbol de decisión construido con el modelo SLIQ cada nodo del árbol de decisión puede ser un nodo interno con uno o dos hijos, o un nodo hoja. El

número de bits necesarios para codificar el árbol depende de lo permisible de la estructura del árbol. Existen tres formas posibles de codificar el árbol:

- Código$_1$: A un nodo se le permite únicamente tener 0 o dos hijos. Como sólo hay dos posibilidades, para codificar cada nodo se necesita un bit.
- Código$_2$: Cada nodo tiene 0 hijos, un hijo a la izquierda, un hijo a la derecha, o ambos hijos. Se necesitan por lo tanto 2 bits para codificar los cuatro posibles valores de cada nodo.
- Código$_3$: Solo se examinan nodos internos. Por lo tanto cada nodo puede tener un hijo a la izquierda, un hijo a la derecha, o ambos hijos, lo que requiere *log(3)* bits.

• Codificación de las particiones: El coste de codificar las particiones depende del tipo de atributo utilizado en la pregunta de la división:

Para atributos numéricos se asumirá un coste de 1 para cada pregunta que utilice un atributo numérico.

Atributos categóricos: El coste de una pregunta formulada sobre un atributo categórico viene dado por Log n_x, donde n_x, es el número de preguntas usadas en el árbol que utilizan el atributo.

Algoritmos de poda

La poda MDL evalúa la longitud de código en cada nodo del árbol de decisión para determinar si el nodo se convierte en una hoja, se poda el hijo de la izquierda o de la derecha. o el nodo permanece intacto. La longitud de código C(t) , para cada tipo de nodo se calcula mediante:

$C_{hoja}\ (t) = L(t) + Errores$ si t es una hoja

$C_{ambos}\ (t) = L(t) + L_{pregunta} + C(t_1) + C(t_2)$ si t tiene dos hijos

$C_{dcha}\ (t) = L(t) + L_{pregunta} + C(t_1) + C'(t_2)$ si t tiene solo un hijo t_1 a la dcha.

$C_{izda}\ (t) = L(t) + L_{pregunta} + C'(t_1) + C(t_2)$ si t tiene solo un hijo t_2 a la izda.

Donde:

$L_{pregunta}$ es el coste de codificar una pregunta.

$C'(t_i)$es el coste de codificar los ejemplos de los nodos hijos utilizando la clasificación del nodo padre.

Existen tres estrategias de poda:

- Poda completa: Considera solo para la poda nodos hoja o con dos hijos. Si $C_{hoja}(t) \pounds C_{ambos}(t)$ entonces se podan los dos hijos, convirtiéndose el nodo en nodo hoja.
- Poda parcial: Considera todo tipo de nodos, convirtiendo el nodo al tipo de nodo que posea un menor coste de codificación.
- Híbrido: Realiza la poda en dos fases. En la primera fase se realiza una poda completa, mientras que en la segunda fase considera todos los nodos excepto los tipo hoja, y los convierte en el nodo con menor coste de codificación.

M5

Este algoritmo [WIT01] desarrollado e implementado en WEKA [WEK02], corresponde a la categoría de árboles de regresió explicados en el sistema CART de Breiman. En este caso, este tipo de árboles son como los árboles ordinarios, pero donde al final de cada rama la clase se representa mediante el promedio del valor de las observaciones que han llegado hasta ella (de regresión) o mediante un modelo de combinación lineal (árbol de modelizado).

Primero se construyen usando un algoritmo de inducción para construir el árbol inicial, pero en este caso, se busca minimizar la variación en la subserie de datos que están por debajo del nodo.

Seguidamente, se realiza el podado como un árbol ordinario con la sola diferencia que para un árbol de regresión se sustituye el valor de esa subrama por el valor constante promedio, mientras que en los árboles de modelizado se sustituye por un modelo lineal del tipo:

$$w_0+w_1 \cdot a_1+w_2 \cdot a_2+\ldots+w_k \cdot a_k \qquad (3.\ 69)$$

donde a_k corresponde con el valor de los atributos, y w_k corresponde a los pesos obtenidos por regresión.

El algoritmo M5 tiene, igual que los algoritmos precedentes C4.5 y CART, mecanismos para trabajar eficientemente con valores inexistentes, ruido, incluso con valores nominales que son convertidos previamente en valores numéricos binarios.

Generalmente, los modelos numéricos generados por árboles no han sido muy utilizados ya que generalmente las redes neuronales son las que se suelen aplicar para desarrollar modelos predictivos de valores numéricos. Quizá una de las ventajas de los árboles de predicción numérica frente a las redes neuronales, es que estas últimas generan un modelo[11] tipo "caja negra" que no ayuda mucho a comprender la naturaleza de la solución mientras que el árbol, además de predecir puede ser usado para obtener conocimiento del sistema.

[11] Es cierto, que hoy en día hay diferentes estudios y técnicas para extraer reglas y conocimiento de los modelos obtenidos mediante redes neuronales, aunque aún no están muy extendidas.

Por ejemplo, en [WIT00] se muestra un ejemplo de un árbol que modeliza el tiempo de establecimiento de un servomotor a partir de dos variables numéricas (ganancia de la velocidad y del par) y dos nominales (tipo de motor={A, B, C, D, E} y tipo de tornillo={A,B,C,D,E}).

Nueva Variable	LM1	LM2	LM3	LM4	LM5	LM6	LM7
Constante	**-0,44**	**2,60**	**3,50**	**0,18**	**0,52**	**0,36**	**0,23**
Gpar							
GVel	**0,82**		**0,42**				**0,06**
M1=("1" SI *motor*=D o "0" SI *motor*=E,C,B,A)		**3,30**		**0,24**	**0,42**		
M2=("1" SI *motor*=D, E o "0" SI *motor*=C,B,A)	**1,80**			**-0,16**		**0,15**	**0,22**
M3=("1" SI *motor*=D,E,C o "0" SI *motor*=B,A)				**0,10**	**0,09**		**0,07**
M4=("1" SI *motor*=D,E,C,B o "0" SI *motor*=A)			**0,18**				
S1=("1" SI *tornillo*=D o "0" SI *tornillo*=E,C,B,A)							
S2=("1" SI *tornillo*=D, E o "0" SI *tornillo*=C,B,A)	**0,47**						
S3=("1" SI *tornillo*=D,E,C o "0" SI *tornillo*=B,A)	**0,63**		**0,28**	**0,34**			
S4=("1" SI *tornillo*=D,E,C,B o "0" SI *tornillo*=A)			**0,90**	**0,16**	**0,14**		

Tabla 13. Modelos lineales calculados para el árbol de la figura siguiente [WIT00].

De esta forma, el algoritmo para cada variable nominal genera cuatro nuevas variables binarias que son aplicadas a cada modelo lineal tal y como se muestra en la Tabla 13. Por ejemplo, el modelo *LM1* sería igual a:

$$LM1 = -0{,}44 \cdot GVel + 1{,}80 \cdot M2 + 0{,}47 \cdot S2 + 0{,}63 \cdot S3 \qquad (3.\ 70)$$

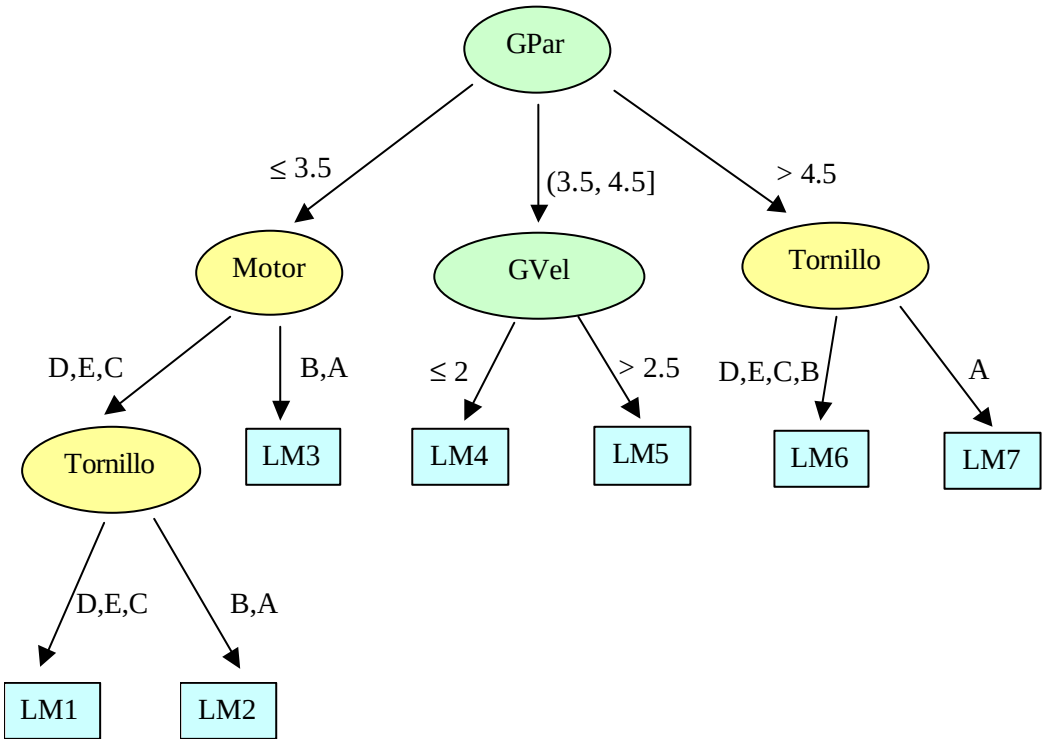


Figura 72. Árbol generado que utiliza siete modelos lineales para predecir el tiempo de establecimiento de un servomotor a partir de dos variables numéricas y dos nominales [WIT00].

Y que se utilizará para calcular el tiempo de establecimiento del servo siempre que se cumplan las condiciones de la rama izquierda de la Figura 72, y se calculen los valores binarios de las variables *M2*, *S2* y *S3* a partir de los valores nominales de *Motor* y *Tornillo*.

NDTs (NON-LINEAR DECISIÓN TREES)

Realmente este tipo de árboles se basan en las técnicas vistas anteriormente para desarrollar el árbol, pero donde en cada nodo se sustituye por un clasificador cualquiera: lineales, no lineales, basados en rejillas dispersas (Sparse-Grids based nodes), etc; que clasifican cada subárbol que parte de cada nodo [PRU02].

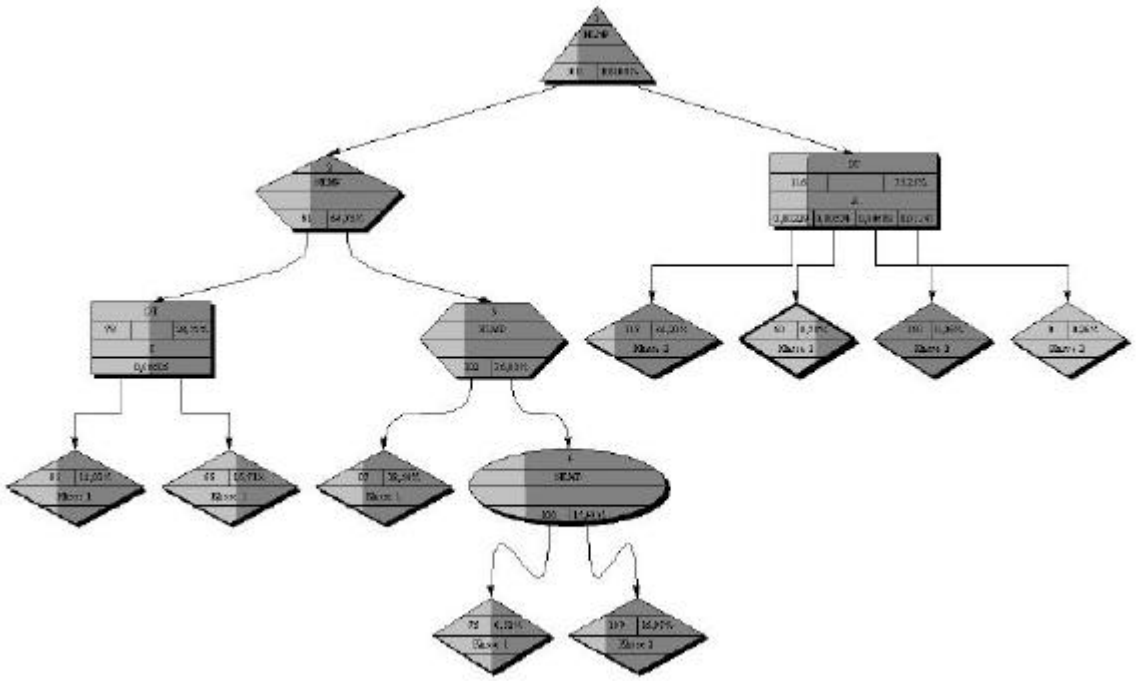


Figura 73. Árboles no lineales implementados por PRUDSYS [PRU02].

OTROS ALGORITMOS GENERADORES DE ÁRBOLES DE DECISIÓN

Como es lógico, de la combinación de estas y otras técnicas se pueden desarrollar algoritmos que presenten más eficiencia fundamentalmente en velocidad y capacidad de predicción.

3.4.4.5 GENERADORES DE REGLAS

Una desventaja de los árboles de decisión es que tienden a ser demasiado grandes en aplicaciones reales y, por tanto, se hacen difíciles de interpretar desde el punto de vista humano. Por ello, se han realizado diversos intentos para convertir los árboles de decisión en otras formas de representación, como las reglas inducidas [ADA01].

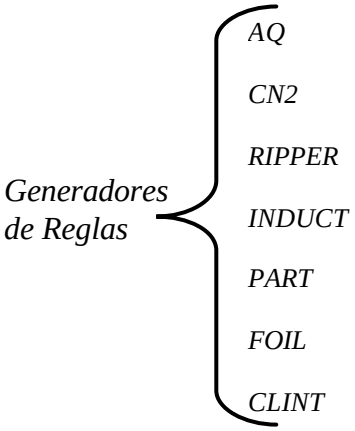


Figura 74. Algunos de los algoritmos básicos para generar reglas.

Las reglas inducidas pueden derivar de la construcción de un árbol de decisión, siendo primero generado el árbol de decisión y después trasladado a un conjunto de reglas.

Ejemplo:

Para el árbol de decisión presentado en la figura:

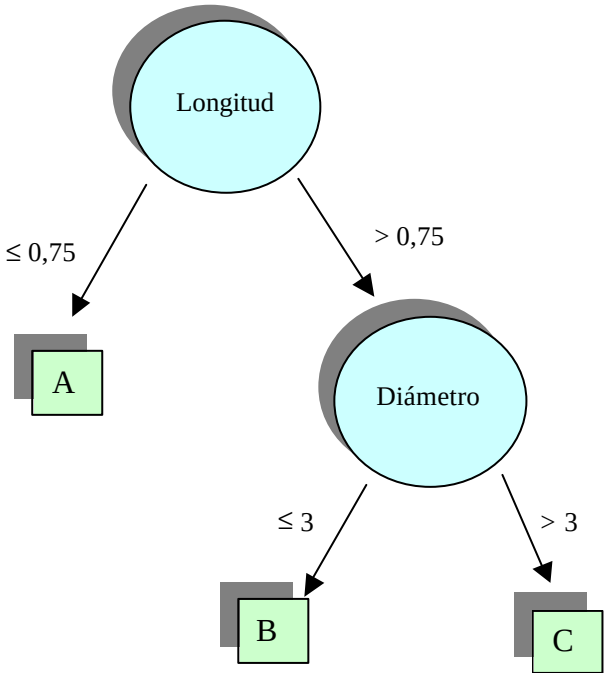


Figura 75. Ejemplo de un pequeño árbol de decisión para clasificar tres objetos según su diámetro y longitud.

La solución equivalente de reglas inducidas es:

- SI (*longitud* ≤ 0,75) ENTONCES "A".
- SI (No (*longitud* ≤ 0,75) Y (*diámetro* ≤ 3)) ENTONCES "B".
- SI (No (*longitud* ≤ 0,75) Y (No(*diámetro* ≤ 3)) ENTONCES "C".

Sin embargo, también se pueden inducir reglas directamente de los patrones mediante estrategias de abajo-hacia-arriba (bottom-up) con un estilo desde los específico a lo general, o con una estrategia de arriba-hacia-abajo (top-down) con un estilo de lo general a lo específico.

Habitualmente la descripción de las clases se realiza mediante reglas disjuntas, recibiendo las reglas inducidas en este caso el nombre de reglas de decisión.

AQ

La familia de algoritmos AQ [MIC86] tiene sus raíces e influencias en métodos de ingeniería eléctrica utilizados para la simplificación de circuitos eléctricos.

Antes de desarrollar el algoritmo AQ se va a proceder a introducir la terminología utilizada por la familia de algoritmos AQ. La pregunta sobre un atributo se denomina "selector", un conjunto de preguntas se denomina "complejo", y una combinación booleana de preguntas (disyunción) se denomina "recubrimiento".

Para cada clase, el algoritmo AQ induce una regla de decisión de la forma:

si <recubrimiento> entonces <clase> (3.71)

Almacenando el recubrimiento junto con el valor de la clase asociada.

La estrategia seguida por el algoritmo AQ es de tipo abajo-hacia-arriba. Inicialmente cada uno de los patrones de entrenamiento se considera un complejo. Los complejos entonces son examinados, eliminando selectores de forma selectiva y garantizando la consistencia del complejo resultante (se eliminan únicamente patrones de la misma clase y nunca de diferentes clases). De esta forma en cada etapa se produce un complejo, y mediante la combinación de los complejos generalizados, se construye un recubrimiento completo que cubre todos los patrones de una determinada clase.

La eliminación de selectores (generalización de complejos) se realiza con la ayuda de una función de evaluación. Lo usual es eliminar selectores de forma que se maximice el número de patrones positivos cubiertos, es decir, la función de evaluación para cada complejo viene dada por la proporción del número de patrones clasificados correctamente dividido por el número total de patrones cubiertos.

El objetivo por lo tanto del algoritmo AQ es encontrar un conjunto de recubrimientos compacto que cubra todos los posibles casos. clasificando de forma correcta cada uno de los

patrones en su clase correspondiente. El conjunto de recubrimientos se construye mediante una búsqueda heurística de la "mejor" pregunta para una clase *c*. Para ello se selecciona de forma aleatoria un patrón semilla perteneciente a la clase *c*. Comenzando con la regla más general "*todos los patrones son de clase c*", se exploran diferentes especializaciones de la regla hasta encontrar la pregunta que optimiza el número de patrones cubiertos en la clase *c*, de forma que no cubra patrones de ninguna otra clase. Esta pregunta se añade al conjunto de preguntas de la clase *c*, y se eliminan todos los patrones de la clase *c* que la satisfacen. El proceso se repite hasta que todos los patrones de la clase *c* son cubiertos.

Se pueden almacenar varias "mejores especializaciones hasta el momento", realizando la exploración en paralelo. Al conjunto de soluciones exploradas se le llama "*star*".

AQ garantiza encontrar un conjunto completamente consistente de reglas con los datos de entrenamiento (si es que estas reglas existen), pero no es capaz de clasificar correctamente patrones con ruido, ni considera ninguna estrategia para evitar el sobreentrenamiento.

CN2

El algoritmo CN2 [NIB89] se desarrolló como una extensión del algoritmo AQ que permite el tratamiento de problemas de ruido y sobreentrenamiento.

CN2 retiene un conjunto de complejos durante su búsqueda , de forma que estos complejos cubren un gran número de casos de una clase, aunque también pueden cubrir casos de otras clases. Adicionalmente, el algoritmo CN2 ejecuta un proceso de especialización, de forma que en cada paso de especialización o se añaden nuevas preguntas al complejo, o se elimina todo el complejo.

En la búsqueda de los mejores complejos, CN2 emplea dos tipos de heurística: significancia y bondad. La significancia es el umbral por debajo del cual no se considera un complejo para ser seleccionado como mejor complejo. La significancia es calculada mediante la función de entropía:

$$S(t) = 2 \cdot \sum_{i=1}^{n} p_i \cdot \log(p_i / q_i) \qquad (3.72)$$

donde:

- p_1, p_2, …, p_i es la distribución de frecuencias entre la clase de patrones que satisfacen un complejo.
- q_1,q_2,...,q_i es la distribución de frecuencias esperada del mismo número de patrones bajo la suposición de que le complejo seleccione patrones aleatoriamente.

Los complejos cuya función de entropía sea inferior a un umbral mínimo preestablecido son rechazados.

La bondad es una medida de la cualidad del complejo utilizada para establecer un orden entre los complejos candidatos a la inclusión final en el recubrimiento. La medida usual de bondad es la estimación de error Laplaciano que viene dada por:

$$\Delta E = \frac{n - n_c + k - 1}{n + k} \tag{3.73}$$

donde:

- n es el número total de patrones cubiertos por la regla.
- n_c es el número de patrones positivos cubiertos por la regla.
- k es el número de clases distintas.

Relación entre CN2 y AQ

AQ	CN2
Busca reglas que sean completamente consistentes con los datos de entrenamiento.	**Puede parar prematuramente una especialización cuando no existen reglas generadas por especialización por arriba de un umbral mínimo de significancia estadística dada por la función de entropía**
Considera sólo especializaciones que excluyen un ejemplo negativo específico que cubre la regla.	**Considera todas las especializaciones. Las especializaciones generadas por CN2 que no excluyen ningún ejemplo negativo son rechazadas.**
La función de evaluación es el número de ejemplos clasificados correctamente dividido entre el número total de ejemplos cubiertos.	**La función de evaluación es la estimación M error Laplaciano**
Genera reglas no ordenadas	**Genera reglas ordenadas**

Tabla 14. Comparación entre AQ y CN2.

RIPPER

Este algoritmo, desarrollado en [COH95], junto con CN2 y C4.5 son los métodos básicos que han servido para la implementación de nuevos algoritmos más complejos y robustos. Por ejemplo, SLIPPER es un algoritmo desarrollado por los mismos autores que el de RIPPER, pero más rápido y robusto.

Básicamente RIPPER es muy parecido al C4.5. Inicialmente los dos algoritmos generan una serie inicial de reglas que luego son depuradas en el C4.5 y combinadas en el RIPPER. Mientras que las reglas que genera C4.5 provienen de un árbol de decisión y son, por lo tanto, largas y redundantes, en RIPPER, en cambio, se generan reglas muy simples que luego son recombinadas y reemplazadas en otras más completas y más complejas.

INDUCT

INDUCT [GAI95] es otro algoritmo que se basa en una versión más sencilla llamada PRISM [CEN87] que usa AQ y ID3 (y su evolución C4.5) para generar reglas diferentes para cada clase a clasificar.

En este caso, PRISM es un algoritmo bastante sencillo que requiere que los datos estén libres de ruido. En cambio, INDUCT es una evolución de PRISM de tal forma que usa la distribución binomial para determinar la bondad de una regla, lo que permite generar mejores reglas con datos con cierto porcentaje de ruido.

Este algoritmo se ha visto mejorado para incluir reglas con excepciones [COM99], llamado RDR "Ripple-Down Rules", donde se demostraba que las reglas con excepciones generadas de una base de datos médica eran más comprensible para los médicos que las reglas básicas generadas mediante los métodos clásicos. En este caso, se demostraba que el personal médico estaba familiarizado con los diagnósticos complejos que incluyen varias excepciones.

PART

Un algoritmo generador de reglas a partir de subárboles, denominado PART [FRA98] está implementado en WEKA [WEK02]. En éste algoritmo se basa en las técnicas para generar árboles y reglas, de forma que genera subárboles que luego son convertidos a reglas.

Es un algoritmo bastante robusto a ruidos y valores ausentes.

FOIL

El algoritmo FOIL [QUI90] [QUI93a] se basa en el uso de técnicas de aprendizaje con lógica de predicados (ILP, *Inductive Logic Programming*). En este caso los conceptos son definidos mediante cláusulas *Horn*, de la forma:

$$\begin{aligned} &C_1 \text{ :- } L_{11}, L_{12}, ..., L_{1m} \\ &C_2 \text{ :- } L_{21}, L_{22}, ..., L_{2m} \\ &... \end{aligned} \tag{3.74}$$

donde el predicado es C_i es la cabeza de cada cláusula y los literales L_{ij} corresponden al cuerpo de la misma. Las comas representan una unión tipo "Y".

El algoritmo FOIL realiza los siguientes pasos:

- Inicializa la cláusula definiendo la cabeza como el concepto a aprender y deja el cuerpo de la misma vacío.
- Mientras la cláusula cubra ejemplos negativos haz:

- Encuentra un "buen" literal que se pueda añadir al cuerpo de la cláusula. Para ello se basa en el uso del criterio de Información utilizado en la generación de árboles de decisión.

- Elimina todos los ejemplos que cubre la nueva cláusula.
- Añade la cláusula a la definición del concepto. Si hay, algún ejemplo positivo sin cubrir comienza de nuevo en el paso 1.

Es interesante analizar cómo este algoritmo usa la metodología AQ para cubrir los casos positivos utilizando el método "divide y conquistarás" de los árboles de decisión.

La lógica inductiva de predicados (ILP) es especialmente útil cuando se dispone de una base de conocimiento que aplicar. De esta forma, el sistema puede ser "alimentado" con una serie de reglas que ayudan a obtener nuevos patrones de comportamiento en los datos.

CLINT

En [MIC98] se describe una solución, denominada CLINT, que se basa en los siguientes pasos:

- Construir el árbol que explica los ejemplos negativos.
- Identificar la cláusula que cubra los casos negativos.
- Borra la cláusula de la base de conocimiento.
- Recompone la estructura de conocimiento estudiando las cláusulas positivas que habían sido cubiertas por la cláusula anterior.

OTROS ALGORITMOS GENERADORES DE REGLAS

El número de algoritmos crece considerablemente, fundamentalmente mediante combinaciones y afinamientos de los algoritmos. Los campos de investigación se basan fundamentalmente en:

- Mejorar la velocidad de generación de árboles y reglas.
- Obtener reglas y árboles más robustos al ruido o datos inexistentes.
- Crear reglas y árboles que sea más eficientes y fáciles de comprender.

3.4.4.6 REDES NEURONALES

Las redes neuronales representan una forma especial de procesamiento de la información. Una red neuronal es una estructura compuesta por muchas unidades, muy simples, de procesamiento o neuronas, cada una con memoria local, habitualmente pequeña. Las neuronas se conectan mediante canales de comunicación denominados conexiones, que manejan datos numéricos. Operan sólo con los datos locales, por lo que tienen un gran potencial para el procesamiento paralelo, dado que los cálculos de los componentes en cada neurona son independientes. Las *redes neuronales*, también llamadas "*redes de neuronas artificiales*", son *modelos* bastante simplificados de las redes de neuronas que forman el cerebro. Y, al igual que este, intentan "aprender" a partir de los *datos* que se le suministran.

DEFINICIÓN

Las redes neuronales, o sistemas conexionistas, representan una forma especial de procesamiento de la información. Una red neuronal es una estructura compuesta por muchas unidades, muy simples, de procesamiento o *neuronas*, cada una con memoria local, habitualmente pequeña. Las neuronas se conectan mediante canales de comunicación denominados *conexiones*, que manejan datos numéricos. Operan sólo con los datos locales, por lo que tienen un gran potencial para el procesamiento paralelo, dado que los cálculos de los componentes en cada neurona son independientes.

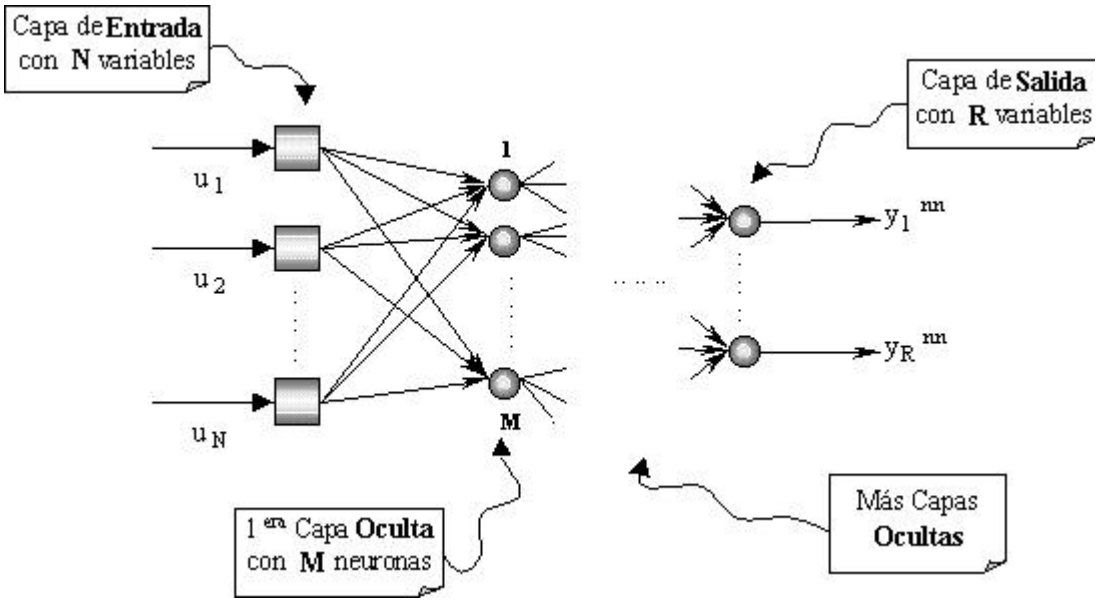


Figura 76. Partes de una Red Neuronal.

La mayoría de las redes neuronales tienen algún tipo de regla de aprendizaje, de forma que los pesos de las conexiones se ajustan dependiendo de los patrones presentados. Actualmente existen muchos métodos de aprendizaje y su número se incrementa día a día.

Según el método de aprendizaje, se distinguen dos grandes grupos:

- Redes Supervisadas: Durante la fase de aprendizaje, se indica a la red qué salida debe producir cada patrón, ajustando los pesos en función de ese valor. El aprendizaje supervisado es el típico de las redes concebidas para el ajuste de datos.
- Redes No Supervisadas: La red localiza en los datos de entrada propiedades que utiliza para separar los patrones en clases. El aprendizaje no supervisado es característico de las redes utilizadas en los casos en que los datos no tienen a priori ningún tipo de clasificación. La red se utiliza para detectar las regularidades intrínsecas de los datos estableciendo así la mejor clasificación posible.

En algunos casos se utilizan redes para cálculos de series temporales, en los cuales es necesario introducir valores de una misma variable en diferentes instantes de tiempo, para lo cual existen diversas combinaciones de redes: recurrentes y *Feedforward*, que denominamos *Time Delay*.

Hasta el momento, se han desarrollado muchos modelos de redes neuronales, pensados para una gran variedad de aplicaciones. Aunque se diferencian en su modo de activación, modelo de neurona, implementación y modo de funcionamiento, todos ellos tienen rasgos comunes: son sistemas de computación generados a partir de un conjunto compacto de unidades simples de procesamiento de información -neuronas artificiales- que comparten las siguientes características:

- Las neuronas están densamente conectadas, de tal forma que el estado de una neurona afecta al potencial de todas las neuronas a que está conectada, de acuerdo con los pesos de sus conexiones.
- Habitualmente los pesos sinápticos se adaptan según reglas de optimización, y puesto que la variación puede realizarse en cualquier parte de la red, se dice que ésta tiene memoria distribuida.
- Las neuronas tienen funciones de activación no lineales, Esto es que el nuevo estado de una neurona es una función no lineal de las señales generadas por las activaciones de otras neuronas.
- Aunque la red usa elementos simples, se caracteriza por su gran inmunidad al ruido en los datos de entrada.

ARQUITECTURA DE LAS REDES NEURONALES

Los diferentes modos de conectar las neuronas para generar una red se denominan arquitecturas. Las arquitecturas de las redes neuronales se dividen en tres grandes categorías (Figura 77):

Redes Progresivas o Unidireccionales (*Feedforward Networks***)**: Las neuronas se organizan de forma lineal (habitualmente en capas) de manera que cada neurona puede recibir una entrada del exterior o de las neuronas precedentes, pero no de las posteriores. Este tipo de redes da un patrón de salida en respuesta a un determinado patrón de entrada. Una vez entrenada, se fijan los pesos y la respuesta a un determinado patrón de entrada será la misma, independientemente de cualquier actividad anterior de la red. De esta forma, las redes progresivas no tienen dinámica real y no padecen por tanto problemas de estabilidad. Su dinámica se ha reducido a una simple aplicación no lineal instantánea. Estas redes están concebidas para transformar un determinado conjunto de datos de entrada en uno de salida. Cada conjunto de datos de entrada tiene uno correspondiente de salida. El objetivo es obtener una red que sintetice esta transformación, permitiendo la generalización de la función de transferencia a base de extrapolar a pares de datos semejantes. Así pues, los datos de aprendizaje de la red son pares entrada/salida, que la red debe seguir tras la fase de entrenamiento.

El aprendizaje es supervisado y tiene lugar a través de un proceso de ajuste de los pesos sinápticos de las neuronas de la red, de forma tal que se satisfaga algún tipo de criterio de aproximación u optimización. Típicamente, la arquitectura de una red de este tipo es la de una red multicapa, donde las neuronas de cada capa están enlazadas con las de la siguiente. El prototipo de estas redes es el perceptrón multicapa, que después se estudiará en detalle.

Redes Recurrentes o Realimentadas (*Feedback Networks*): Presentan varios estados en cada ciclo, de modo que son necesarias varias iteraciones para que converja la activación de cada neurona. En la forma de esta convergencia y en su propia dinámica puede residir el comportamiento temporal de las variables, de modo que su uso es especialmente adecuado en series temporales. Principalmente estas redes están concebidas para almacenar eficientemente información, de manera que cada ítem registrado se recupere con facilidad cuando la red se excita con una entrada similar. La red funciona como un sistema dinámico cuyos puntos de equilibrio representan los registros almacenados. Un dato de entrada a la red se interpreta como el estado inicial del sistema, a partir del cual evoluciona hasta alcanzar el equilibrio, cuyo registro se toma como salida o respuesta de la red. El entrenamiento de la red consiste en ajustar sus pesos de manera que los puntos de equilibrio se ajusten a los registros especificados. Típicamente la arquitectura de estas redes es la de una red monocapa con una gran realimentación entre todas las neuronas. El modelo más

conocido es la red de *Hopfield*, de la que se han derivado otros más complejos como las *memorias bidireccionales* o la red de *Boltzmann*.

Redes Celulares : Están formadas por neuronas artificiales regularmente espaciadas (llamadas células) que se comunican directamente con otras neuronas, pero sólo con las de su entorno. Las células adyacentes actúan entre sí mediante conexiones laterales múltiples. Incluso aquellas no conectadas pueden interactuar indirectamente a causa de la propagación de la señal durante el régimen dinámico (transitorio). Debido a su conectividad local, cada celda es excitada por su propia señal y por las señales de las celdas de su entorno. Además, la interacción mutua origina que el procesamiento de la señal se propague en el tiempo y alcance a todas las células de la red.

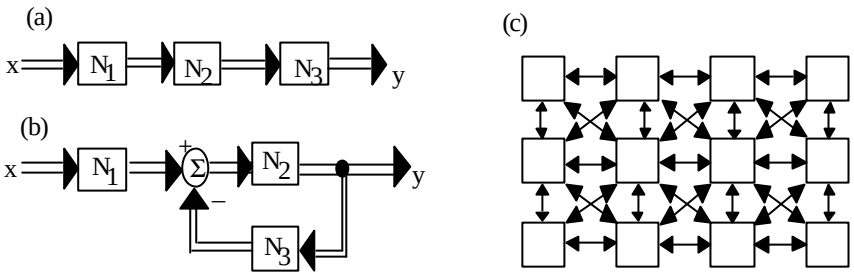


Figura 77. Los tres tipos de arquitecturas principales de las redes neuronales: 1.- Redes Progresivas, 2.- Redes Recurrentes, 3.- Redes Celulares.

Generalmente, una red neuronal está definida no sólo por su arquitectura, sino también por el tipo de neuronas usadas, la regla de aprendizaje o de entrenamiento, y su forma de operación.

TIPOS DE REDES NEURONALES

Existen varias clasificaciones de las múltiples variantes existente. Por ejemplo, se pueden dividir en nueve tipos de redes neuronales que se agrupan de acuerdo a las siguientes características:

- Con entradas binarias:
  - Aprendizaje supervisado:
    - Redes de Hopfield
    - Redes de Hamming
    - ART-3
  - Aprendizaje no supervisado:
    - Clasificador de Carpenter/Grossberg o ART-1
- Con entradas de valores continuos:
  - Aprendizaje supervisado
    - Perceptrón (Clasificador Gaussiano)

- Perceptrón Multicapa

o Aprendizaje no supervisado

- Mapas autoorganizativos de Kohonen
- ART-2
- Masking Fields

Otra clasificación es la que se muestra en la Figura 78.

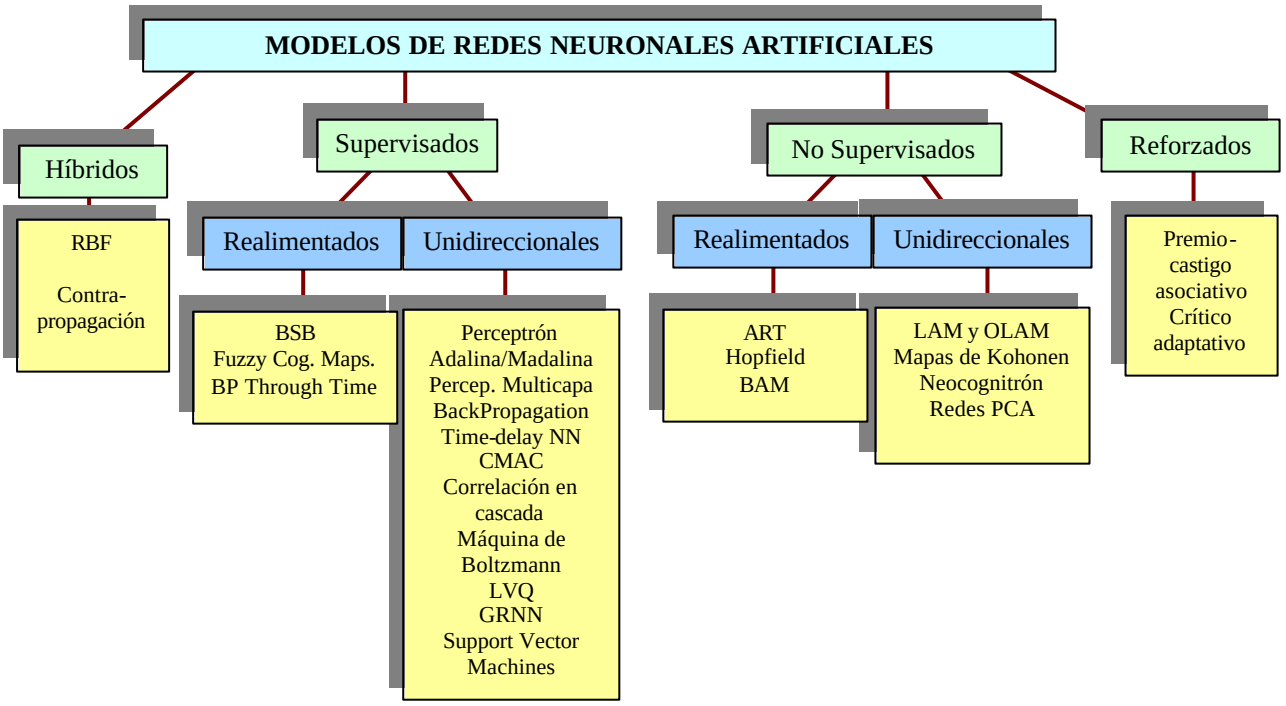


Figura 78. Clasificación de las Redes Neuronales Artificiales según [MAR01].

EVOLUCIÓN HISTÓRICA

Se considera el inicio de la utilización de redes neuronales en el área de la computación el trabajo publicado por *W. McCulloch* y *W. Pitts* [MCC43], en el que presentan un modelo simple de cálculo neuronal basado en el comportamiento de las neuronas biológicas y que realizaba cualquier función lógica o aritmética. Es la primera vez que se asimila el cerebro a un organismo computacional. Tal como indican *Anderson* y *Rosenfeld* [AND88], hay una idea fundamental que no se ponía de manifiesto en el artículo de *McCulloch* y *Pitts*: aunque las neuronas son dispositivos sencillos, se puede obtener una gran potencia de cálculo cuando se conectan adecuadamente entre sí.

Entre los años 50 y 60 se llevan a cabo diferentes trabajos que pondrán las bases para futuros desarrollos. En 1.951 *Marvin Minsky* [MIN54] construye el primer ordenador, *Snark*, orientado a redes, que sin realizar ningún procesamiento de información especial, sirvió de ayuda para diseños posteriores. *Nils Nilsson* [NIL65] resume la mayor parte del trabajo realizado en esta época.

En 1.959 *Widrow* y *Hoff* [WID60] desarrollan el *Adaline* (*Adaptive Linear Element*), que es uno de los modelos más sencillos de neurona con capacidad de aprendizaje: es un dispositivo que consta de un único elemento de procesamiento. En este caso se debe hablar de un tipo de elemento de procesamiento más que de una red neuronal propiamente dicha. Posteriormente, para aumentar la capacidad de decisión, se emplean varios de estos elementos trabajando en paralelo, lo que da origen al *Madaline* (*Multiple Adaline*).

En 1.959 *Frank Rosenblatt* [ROS59] publica el primer resultado importante en el cálculo de redes neuronales: el desarrollo del Perceptrón. El Perceptrón es un sistema de clasificación de patrones que puede identificar patrones geométricos y abstractos, debido a que tiene cierta capacidad de aprendizaje. Además, *Rosenblatt* prueba que el Perceptrón es capaz de clasificar cualquier conjunto de entrenamiento linealmente separable, mientras que el *Adaline* falla en algunos casos.

A mediados de los años 60, *M. Minsky* y *S. Papert* del MIT, [MIN69] realizan un análisis minucioso y riguroso del Perceptrón en términos de sus capacidades y limitaciones. El principal resultado de esta investigación es que el Perceptrón sólo puede distinguir tramas si éstas son linealmente separables. Dado que hay muchos problemas de clasificación que no son linealmente separables, esta condición impone unos límites bastante restrictivos a la aplicabilidad del Perceptrón. Este problema se resuelve introduciendo una nueva capa, lo que da lugar al Perceptrón Multicapa (*MultiLayer Perceptron*, *MLP*). De este modo se puede construir una función discriminante válida para las regiones que no son linealmente separables. En este caso aparece un problema serio, y es que el ajuste de los pesos en la red no puede realizarse de forma directa, debido a la existencia de la capa intermedia.

En el período que va de 1.967 a 1.982 se investigó poco en EE.UU. en el campo de los sistemas neuronales. La investigación en Japón, Europa y la Unión Soviética se vio menos afectada destacando los trabajos sobre construcción de modelos neuronales basados en datos neurológicos de Stephen Grossberg [GRO76]. La clase de modelos que comienza a desarrollar cae dentro de las redes denominadas ART [CAR88].

Desde principios de los 70 *Teuvo Kohonen* realiza investigaciones importantes con memorias asociativas y aprendizaje adaptativo [KOH77]. Una de sus contribuciones es el principio de aprendizaje competitivo. En él los elementos de proceso compiten para responder a un estímulo y el elemento ganador se autoadapta para tener más responsabilidad en la respuesta al estímulo siguiente.

La computación neuronal recibe entre los años 1.983 y 1.986 el apoyo de un investigador mundialmente considerado, *John Hopfield*, que se había interesado por este campo unos pocos años atrás. *Hopfield* publicó dos trabajos [HOP82] [HOP84], que, junto a múltiples conferencias dadas en todo el mundo, persuadiendo a muchos científicos para realizar investigaciones en el campo de las redes neuronales.

En 1.986 se produce la consolidación definitiva con la publicación del libro de *D. Rumelhart* y *J. McClelland*, *"Parallel Distributed Processing"* [RUM86]. En estos trabajos se da respuesta al problema del aprendizaje planteado por el Perceptrón Multicapa, desarrollando el método de retropropagación (*Backpropagation*).

MODELOS DE REDES NEURONALES

La variedad en cuanto a topologías, funciones de activación, etc., hace que la simple descripción de las distintas redes neuronales se convierta en un tema demasiado amplio como para abordarlo aquí. Por ello se tratarán exclusivamente los tipos de redes más básicos y conocidos.

Redes Progresivas. El Perceptrón Multicapa

La topología mas conocida dentro de las redes de propagación hacia adelante es el Perceptrón Multicapa. Como se ha visto, un sólo Perceptrón tiene muchas limitaciones. La más importante es que no puede distinguir clases que no sean linealmente separables. Si se organizan un conjunto de ellos en capas, esta limitación queda superada. Cada capa consta de varias neuronas cuya entrada proviene de las unidades de la capa anterior, y cuya salida va a las unidades de la capa posterior, sin ningún contacto con cualquier otra capa, ni entre las neuronas de una misma capa. La primera capa, denominada capa de entrada, recoge la señal de entrada y no realiza sobre ésta ningún tipo de procesamiento. El resultado de la red viene dado por el estado de las neuronas de la última capa, denominada capa de salida. Las capas intermedias se denominan capas ocultas.

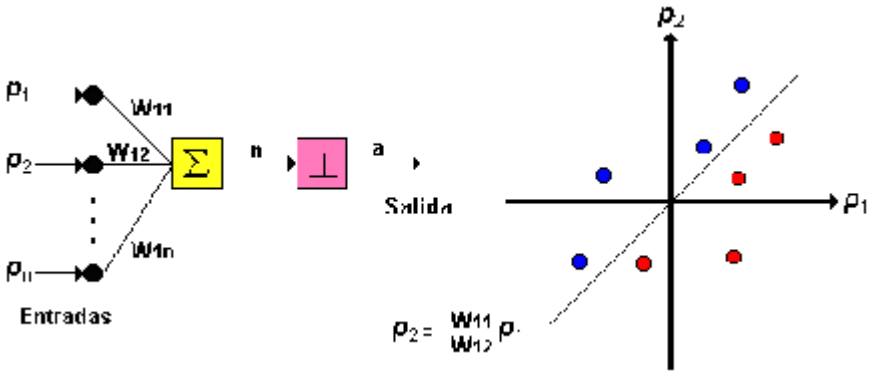


Figura 79. La red perceptrón [ACO00].

Uno de los ejes centrales de las investigaciones en el campo de las redes neuronales es cómo ajustar los pesos de los enlaces para obtener el comportamiento deseado del sistema. Esta modificación esta basada a menudo en la regla de Hebb, que indica que el enlace entre dos unidades debe ser reforzado si ambas unidades se activan al mismo tiempo. La forma más general de la regla de Hebb es:

$$\Delta w_{ij} = g\left(a_j(t), t_j\right) \cdot h\left(o_i(t), w_{ij}\right) \tag{3.75}$$

donde:

w_{ij}	**Peso de la conexión desde la neurona *i* hasta la neurona *j*.**
$a_j(t)$	**Activación de la unidad *j* en el tiempo *t*.**
t_j	**Salida deseada de la unidad *j* en un patrón de aprendizaje.**
$o_i(t)$	**Salida de la unidad *i* en el tiempo *t*.**
$g(...)$	**Función, depende de la activación de la unidad y de la función de aprendizaje.**
$h(...)$	**Función, depende de la salida de la unidad precedente y del peso de la conexión..**

En las redes con varias capas, tanto el número de capas como el de neuronas en cada capa son parte esencial del diseño. El número de capas está íntimamente relacionado con la velocidad de aprendizaje, por lo que comúnmente el número de capas ocultas se reduce a una o dos. Del número de neuronas en las capas ocultas depende el número de parámetros libres en el modelo y su operatividad. Si son pocas es imposible lograr el ajuste y si son excesivas se pierde generalidad.

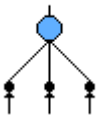
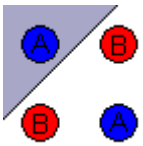
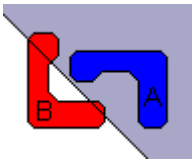
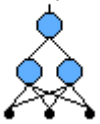
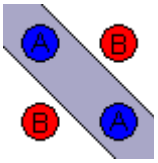
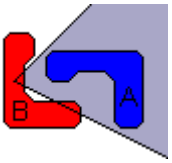
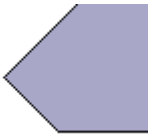
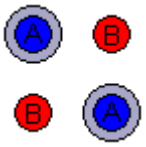
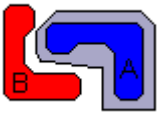
Estructura	Regiones de Decisión	Problema de la XOR	Clases con Regiones Mezcladas	Formas de Regiones más Generales
1 Capa 	Medio Plano Limitado por un Hiperplano			
2 Capas 	Regiones Cerradas o Convexas			
3 Capas 	Complejidad Arbitraria Limitada por el Número de Neuronas			

Figura 80. Problemas que pueden abordar según el número de capas [ACO00].

En general, se emplean redes neuronales multicapa con aprendizaje por retropropagación del error para simular una función desconocida a partir de los pares de señales de entrada y salida obtenidos de los ejemplos de aprendizaje [SON92]. Se espera en estos casos que la red sea capaz de generalizar adecuadamente, de forma que ante señales de entrada que no hayan sido presentadas previamente como patrones de aprendizaje se obtenga la salida adecuada. Aunque hay otros algoritmos basados en la misma topología [BRA92], [FAL91a], [FAL91b], las más extendidas son las siguientes:

Redes de Retropropagación (*Backpropagation*)

Es el algoritmo más famoso en redes progresivas supervisadas. La regla de actualización de los pesos, también denominada regla delta generalizada [ZEL93], se escribe como:

$$\Delta w_{ij} = x q_j o_i$$
$$d_j = \begin{cases} f'_j(s_j)(t_j - o_i) & \text{si la unidad } j \text{ es de salida} \\ f'_j(s_j)\sum_k d_k w_{jk} & \text{si la unidad } j \text{ es oculta} \end{cases} \qquad (3.76)$$

donde:

s_j	**Suma de las entradas ponderadas por los pesos de las conexiones.**
x	**Factor de aprendizaje (constante).**
q_j	**Error en la unidad *j* (diferencia entre la salida real y la deseada).**
t_j	**Salida deseada de la unidad *j*.**
$o_i(t)$	**Salida de la unidad y en el tiempo *t*.**
i	**Índice de las unidades precedentes de la *j* con enlaces w_{ij} de *i* hacia *j*.**
j	**Índice de la unidad actual.**
k	**Índice de las unidades posteriores a la *j* con enlaces w_{jk} de *j* hacia *k*.**
d_j	**Variación de los pesos del enlace de *i* hacia *j*.**
d_k	**Variación de los pesos del enlace de *j* hacia *k* (capa de salida).**
f'	**Derivada de la función de activación.**

En ocasiones esta fórmula se corrige introduciendo un término denominado momento. Los pesos se actualizan de acuerdo a la regla precedente, y se evalúa el error. Si el error se decrementa, el cambio se reitera hasta que el error comienza a aumentar. En ese momento se evalúa el nuevo gradiente y continúa el aprendizaje. El término del momento introduce el cambio de los pesos antiguos como parámetro en el cálculo del cambio en los pesos actuales. Esto evita los problemas habituales de oscilación cuando la superficie de error tiene un área mínima muy reducida.

El nuevo cambio de pesos será:

$$\Delta w_{ij}(t+1) = x q_j o_i + a \Delta w_{ij}(t) \qquad (3.77)$$

donde α es una constante que indica la influencia del momento.

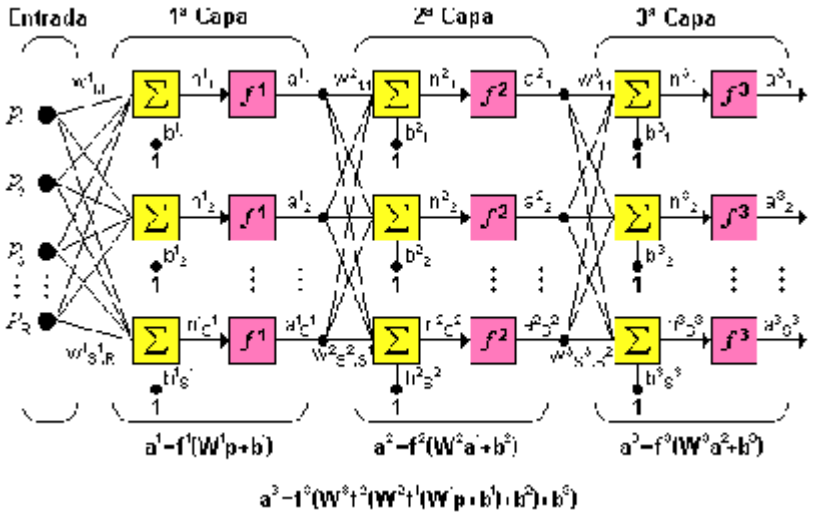


Figura 81. Red backpropagation [ACO00].

Algunos modelos incorporan variables aleatorias a la regla de actualización con la esperanza de que con algunos grados de aleatoriedad, el sistema pueda ser capaz de escapar al mínimo local [RUM86]. Este enfoque ha tenido algún éxito, pero añade complejidad al clasificador que puede dejar de ser válido. En general, concediendo el tiempo necesario y una η lo suficientemente pequeña, puede encontrarse un mínimo adecuado. Quedar atrapado en un mínimo local es habitualmente un problema menos importante si lo comparamos con el de encontrar la arquitectura correcta, el α y x adecuados, así como las otras variables del algoritmo.

QuickProp

Otro método para acelerar el aprendizaje es usar información sobre la curvatura de la superficie de error. Esto requiere el cálculo de la derivada de segundo orden de la función de error. *Quickprop* [FAL88] supone que la superficie de error tiene un comportamiento local cuadrático y realiza en un sólo paso la modificación de los pesos para alcanzar el mínimo de la parábola. Después de calcular el gradiente como en *Backpropagation*, la variación de los pesos se estima como [ZEL93]:

$$\Delta(t+1)w_{ij} = \frac{S(t+1)}{S(t)-S(t+1)}\Delta(t)w_{ij} \qquad (3.78)$$

donde S representa la derivada parcial de la función error en la dirección del peso w_{ij}.

Redes Recurrentes

Las redes progresivas son capaces de realizar una aplicación entre el espacio de entrada y el de salida, una vez entrenadas con pares de datos entrada-salida. Sin embargo, hay problemas donde no existen tales pares, esto es, se dispone de datos de entrada, pero no sus correspondientes datos de salida. Algunos ejemplos de este tipo de problemas pueden ser:

Agrupamiento: los datos de entrada deben ser agrupados en conjuntos cuyas características son inherentes a los datos. La salida de la red es una etiqueta de identificación del conjunto al que pertenece cada dato de entrada.

Cuantificación vectorial: este problema surge de la discretización de un espacio continuo. La entrada del sistema es un vector en un espacio n-dimensional, y su salida es una representación discreta del espacio de entrada. El sistema debe encontrar la discretización óptima de dicho espacio.

Reducción dimensional: intentan reflejar en un menor número de datos la mayor cantidad de información sobre la variación de los datos de entrada. En estos tipos de redes, el entrenamiento se realiza sin la presencia de un tutor externo, de modo que la adaptación de los pesos se basa, normalmente, en algún tipo de competición global entre las neuronas. Hay muchos tipos de redes auto organizadas, dependiendo del problema al que se desee aplicar. Uno de los esquemas más básicos es el propuesto por Rumelhart y Zipser [RUM85]. Otra red similar, pero con propiedades distintas, es el mapa topológico de Kohonen [KOH77], [KOH82], [KOH88]. Dentro de las redes auto organizadas cabe también destacar las redes ART, propuestas por Carpenter y Grossberg [CAR87] [GRO76], y el Cognitrón de Fukushima [FUK75] y [FUK88].

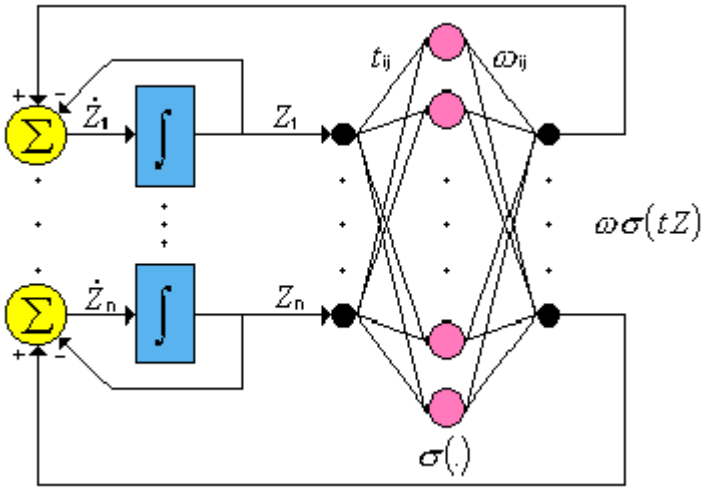


Figura 82. Estructura de una red multicapa [ACO00].

Mapas Autoorganizados - SOM

Las redes con aprendizaje no supervisado (también conocido como aprendizaje autosupervisado) no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por ello, suele decirse que estas redes son capaces de organizarse.

En el caso de las redes SOM, lo que se realiza es una identificación de características, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada. De este modo si se presentan a la red informaciones similares, siempre serán afectadas neuronas de salida próximas entre sí en la misma zona del mapa.

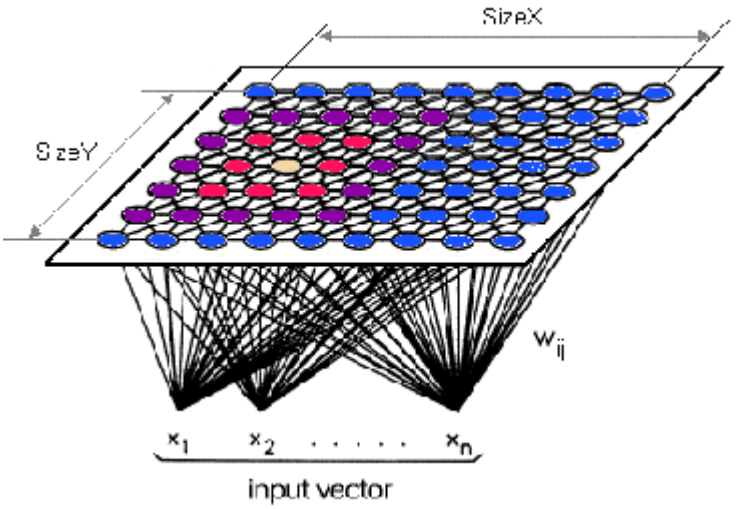


Figura 83. Estructura de una red SOM

La red SOM, tal y como muestra la Figura 83, consiste en un mapa formado por neuronas ubicadas en una malla regular donde la topología de la malla puede ser hexagonal o rectangular. Cada neurona tiene asociado un vector prototipo de tal forma que después del entrenamiento, las neuronas vecinas tienen vectores prototipo similares.

Para entrenar una red SOM lo primero que se debe de hacer es inicializar los vectores prototipo asociados a cada neurona. Esta inicialización se puede realizar con valores aleatorios o mediante una inicialización lineal. La inicialización lineal se realiza mediante el cálculo de los autovalores y autovectores de los datos que van a ser organizados, seleccionando el mayor autovector de los calculados para la inicialización.

El entrenamiento de la red se basa en dos principios:

- Aprendizaje competitivo: la neurona ganadora será aquella que tenga el vector prototipo más similar al vector de datos. Este vector prototipo se modificará de forma que se aproxime aún más a los datos.
- Aprendizaje cooperativo: se modifican los vectores prototipo correspondientes a las neuronas vecinas a la neurona ganadora. El tamaño del radio de las neuronas rodeando a la neurona ganadora es inicialmente grande y va decreciendo a medida que la red se entrena.

En la práctica, se pueden considerar los valores asociados a los puntos de la rejilla como centros de los clusters, más centros donde hay más datos y menos centros donde hay menos datos. Es necesario tener en cuenta que la estructura de malla del SOM hace que se sitúen centros donde

no hay datos. Si hay dos grupos de datos separados por una gran distancia, el hecho de que los puntos están unidos en una malla hace posible que queden puntos de la malla en medio, donde no hay datos.

El funcionamiento de esta red es relativamente simple. Cuando se presenta a la entrada una información $E_k=(e_1^{(k)},...,e_N^{(k)})$, cada una de las M neuronas de la capa de salida la recibe a través de las conexiones *Feedforward* con pesos w_{ji}. También estas neuronas reciben las correspondientes entradas debidas a las conexiones laterales con el resto de las neuronas de salida y cuya influencia dependerá de la distancia a la que se encuentren.

La formulación matemática del funcionamiento puede simplificarse mediante la siguiente expresión, que representa cuál de las M neuronas se activará al introducir la información E_k:

$$s_j=1 \Leftrightarrow min\left\|E_k-w_j\right\|=min\left(\sqrt{\sum_{i=1}^{N}\left(e_i^{(k)}-w_{ij}\right)^2}\right)$$
$$s_j=0 \; en\, otro\, caso \qquad (3.79)$$

donde

s_j	**Salida generada por una neurona de salida *j* ante un vector de entrada E_k.**
w_j	**Vector de pesos de las conexiones entre cada una de las neuronas de entrada y la neurona de salida *j*.**

Lo que hace la red SOM es realizar una tarea de clasificación, ya que la neurona de salida activada ante una entrada representa la clase a la que pertenece dicha información de entrada.

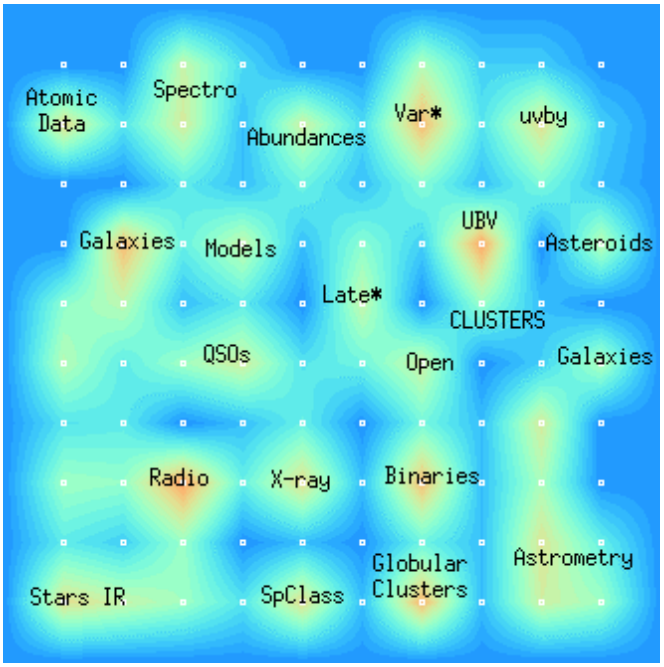


Figura 84. Típico resultado obtenido de la clasificación con una red SOM.

Como se ve, existen, por lo tanto, muchos modelos diferentes de redes neuronales y habrá que elegir entre ellos dependiendo del tipo de problema a tratar considerando sus ventajas y sus inconvenientes. La descripción de cada una de ellas, la forma de entrenarlas y usarlas, así como la enumeración de las aplicaciones más idóneas de cada una de ellas se puede encontrar en la abundante bibliografía que existe del tema.

Funciones de base radial (RBF)

Una Red Neuronal de Funciones de Base Radial (RBF) [STE97][POT96] tiene como objetivo centrar funciones de base radial alrededor de los datos a aproximar (una de las funciones más utilizadas es la de tipo Gausiana), utilizando la propiedad de estas redes neuronales de aproximar cualquier función continua mediante *n* funciones gaussianas que combinadas son la salida de una red RBF.

El problema que se plantea cuando se utiliza este tipo de aproximación radica en la cantidad de funciones radiales, centros c_i y anchos r_i a utilizar. La función de transferencia de las neuronas en la capa escondida es típicamente (existen otras posibilidades) una exponencial conformando una gaussiana (Función de Base Radial). Es de notar que cada neurona de la capa escondida en la red de RBF define un campo receptivo, en el espacio de patrones de entrada, caracterizado por una posición establecida por los valores medios w_{jk} y un alcance dado por la varlanza σ_j. La función de transferencia gaussiana determina la respuesta a cualquier entrada dentro de los campos receptivos y la salida de la red de RBF es la superposición lineal de estas respuestas.

En estas redes no se considera explícitamente un umbral; éste puede incluirse complementando al patrón de entrada con una componente adicional de valor 1 y agregando en la capa intermedia una unidad adicional con actividad constante igual a uno.

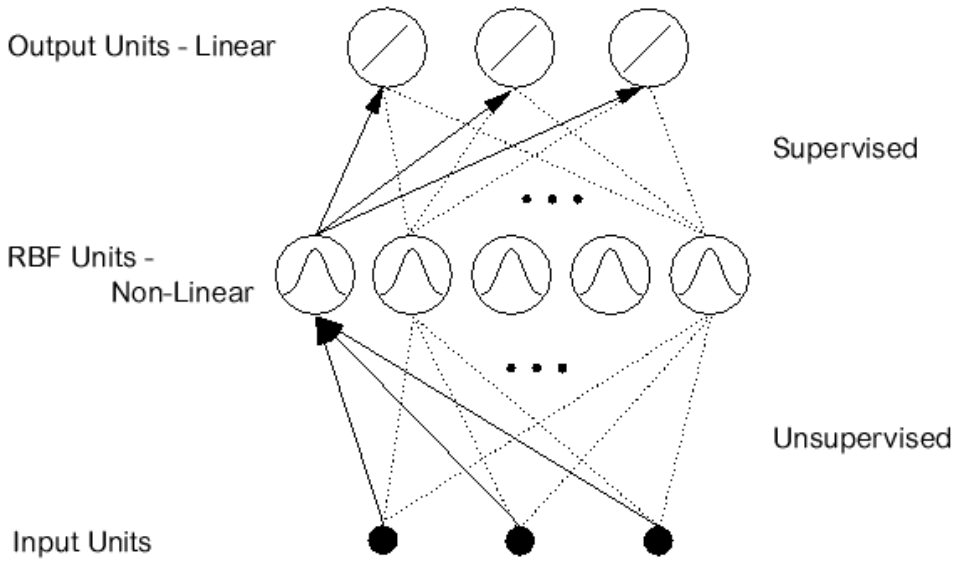


Figura 85. Esquema de una red neuronal RBF.

USO DE LAS REDES NEURONALES

Mediante el empleo de estas técnicas, será posible abordar la identificación de modelos sin proponer a priori su estructura, lo que significa un avance realmente notable desde el punto de vista práctico. Actualmente puede asegurarse que las redes neuronales están definitivamente instaladas como una tecnología firmemente asentada. Sin embargo, no son una herramienta de validez universal y conviene delimitar sensatamente sus aplicaciones. Debe tenerse en cuenta que la computación neuronal está indicada sólo para un determinado tipo de problemas. Existen una serie de criterios básicos para decidir si la solución de un determinado problema es o no susceptible de ser afrontada mediante una red neuronal, de si merece o no la pena considerar su utilización. Los más relevantes son los tres siguientes [COT98]:

- La solución del problema no es descriptible explícitamente mediante un algoritmo o un conjunto de reglas de decisión.
- Existe alguna evidencia de que entre las variables medibles de un proceso existe algún tipo de correlación o dependencia cuya forma, sin embargo, no es explícita, pudiendo enunciarse solamente de manera ambigua.
- Se dispone de una amplia base de datos representativos de las posibles realizaciones del proceso que cubren todas las situaciones de interés.

Hay que destacar la importancia del tercer criterio relativo a la disponibilidad de datos. En términos generales puede decirse que una red neuronal será a lo sumo tan buena como lo sean los datos empleados en su entrenamiento. Si hay dudas sobre la posibilidad de utilizar datos convenientemente representativos del problema es, con toda probabilidad, inútil intentar una solución utilizando la computación neuronal.

A modo general se ofrecen a continuación una serie de aspectos prácticos relacionado con el desarrollo de aplicaciones de redes neuronales [COT98]:

a) Adecuación de los recursos utilizables. Los elementos conceptuales básicos de la computación neuronal son sencillos y no requieren gran esfuerzo para su comprensión. Existe, además, una gran cantidad de software, en buena medida de dominio público, de manera que en general no se requiere gran esfuerzo de programación.

b) Existencia de objeciones serias a la utilización de la computación neuronal. El empleo de una red neuronal puede estar limitado por consideraciones de seguridad de funcionamiento, que puede ser un requisito insoslayable en ciertas aplicaciones.

c) Problemas prácticos relacionados con la obtención de datos. Como ya se ha comentado, el entrenamiento de una red neuronal requiere una amplia colección de datos representativos del problema objeto de estudio.

Como siempre que se introduce una nueva tecnología, es importante la ponderación cuidadosa de los factores que pueden conducir al fracaso de una iniciativa. Algunos puntos que hay que considerar en este sentido son los siguientes [COT98]:

- La estimación de los costes de recopilación y preprocesamiento de datos puede subestimarse, en especial si es necesario repetir experimentos con equipos costosos.
- Es difícil anticipar con precisión las prestaciones finales de una red neuronal. Por ello, siempre es posible que se produzca un déficit de resultados en el comportamiento de la red.
- El deficiente conocimiento de la tecnología neuronal puede repercutir en una elección inadecuada de la aplicación o un diseño ineficiente de la red neuronal.
- Los usuarios potenciales pueden resistirse a emplear una tecnología cuya justificación no es estándar, en la medida en que no se puede explicar, con las explicaciones al uso.

Ventajas del Uso de Redes Neuronales

La utilización de redes neuronales en el campo del análisis y la modelización es muy común debido a sus múltiples ventajas, entre las que cabe destacar [GON99].

- **Aprendizaje de la experiencia**. Las redes neuronales tienen una aplicación típica en aquellos sistemas en los que resulta complejo tanto especificarlos como encontrar una solución organizativa de los mismos, pero que tienen la particularidad de que generan gran cantidad y variedad de datos, de los que se puede inferir una respuesta de la que el sistema aprenderá.
- **Generalización a partir de ejemplos**. Una propiedad de cualquier sistema de auto-aprendizaje es la habilidad de interpolar a partir de casos previos. Con un diseño cuidadoso, una red neuronal puede proporcionar altos niveles de generalización y dar la respuesta correcta a datos que nunca se habían presentado.
- **Extracción de información de datos con ruido**. Puesto que en esencia son sistemas estadísticos, las redes pueden reconocer patrones subyacentes del ruido del proceso, una propiedad que es frecuentemente explotada en aplicaciones como la monitorización del proceso de las máquinas.
- **Desarrollo de soluciones más rápido y con menos dependencia de expertos**. Las redes neuronales aprenden de los ejemplos de modo que, siempre que existan ejemplos suficientes y se adopte un diseño apropiado, se pueden construir soluciones efectivas mucho más rápidamente que con los procedimientos tradicionales.
- **Adaptabilidad**. La naturaleza de las redes neuronales les permiten adaptarse a cualquier tipo de situación operativa. Por ejemplo, en una aplicación industrial pueden evitar las variaciones debidas a desgaste, etc.

- **Eficiencia computacional**. El entrenamiento de las redes neuronales demanda una gran potencia de computación, pero los requisitos en el modo de operación (tras el entrenamiento) son muy modestos. Para problemas muy grandes la velocidad se puede aumentar mediante el procesamiento paralelo, aprovechando su estructura intrínsecamente paralela. Sin embargo esto no suele ser necesario como se verá posteriormente.
- **No linealidad**. Muchas otras técnicas están basadas en asumir cierta linealidad, lo que limitas su aplicación a problemas del mundo real. Por su forma de construcción, las redes neuronales son grandes procesadores no lineales que pueden ser entrenados para su uso en un amplio rango de situaciones complejas.

Sin embargo, tal y como hemos visto en párrafos anteriores, es necesario tener en cuenta, que todas estas ventajas dejarían de serlo si el problema a tratar no se adaptase a su modelado mediante esta técnica, o existiese algún inconveniente de índole práctica que impidiese su implantación.

Diferencias Frente a otras Técnicas de IA

Así, las principales *características* que diferencian a las *redes neuronales* de otras tecnologías de IA son:

- Su capacidad de **aprendizaje** a partir de la experiencia (*entrenamiento*). Normalmente, para la elaboración de un *programa informático* es necesario un *estudio* detallado de la tarea a realizar para después *codificarla* en un lenguaje de programación. Pero, las *redes neuronales* pueden ser *entrenadas* para realizar una determinada tarea sin necesidad de un estudiar esta a fondo ni programarla usando un lenguaje de programación. Además; las redes neuronales pueden volver a entrenarse para ajustarse a nuevas necesidades de la tarea que realizan, sin tenerse que reescribir o revisar el código (cosa frecuente en programas tradicionales).
- Su **velocidad** de respuesta una vez concluido el entrenamiento. Se comportan también en este caso de manera similar a como lo hace el cerebro: los seres humanos no necesitamos pensar mucho para identificar un objeto, una palabra,... una vez hemos aprendido a hacerlo.
- Su **robustez**, en el sentido de que el conocimiento adquirido se encuentra repartido por toda la red, de forma que si se lesiona una parte se continúan generando cierto numero de respuestas correctas (en este caso también hay cierta analogía con los cerebros parcialmente dañados).

3.4.4.7 CLASIFICADOR BAYESIANO 'ELEMENTAL'

Dada una hipótesis *H* y una evidencia *E* que conduce a esa hipótesis, entonces podemos decir según la regla de *Bayes* que:

$$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]} \tag{3.80}$$

donde *Pr[A]* corresponde con la probabilidad de que suceda el suceso *A* y *Pr[A|B]* denota la probabilidad del suceso *A* condicionado a que suceda *B*.

Si tenemos E_1, E_2, ..., *En* evidencias independientes de un suceso *H*, la probabilidad de que suceda frente a una nueva evidencia *E* se generaliza como:

$$\Pr[H \mid E] = \frac{\Pr[E_1 \mid H]\Pr[E_2 \mid H]...\Pr[E_n \mid H]\Pr[H]}{\Pr[E]} \tag{3.81}$$

Que nos permite modelizar cada clase, a partir de las probabilidades que existen de que se produzca para cada uno de los evidencias que se conocen. Este método se denomina de *Naive Bayes* y se puede utilizar siempre que los eventos sean independientes.

Este método, aunque puede resultar muy simplista, trabaja muy bien con bases de datos reales cuando se ha hecho una selección previa de las variables y se obtiene una serie de atributos con muy poca interdependencia [WIT01].

Otra de las pegas de este método, es que si una de las probabilidades es cero, ya que no existe ningún caso positivo para alguna de las evidencias, la probabilidad final, debido al producto, sería cero. Soluciones fáciles pueden consistir en añadir un valor 1 a cada numerador de cada cálculo de probabilidad y dividir finalmente por el denominador más el número de probabilidades (denominado *estimados de Laplace*),o añadir una constante muy pequeña μ en numerador y denominador. Por ejemplo, si tuviésemos las probabilidades {2/9, 4/9, 3/9} el producto sería cero, pero podríamos resolverlo de estas dos formas:

$$\frac{2+1}{9+3}\cdot\frac{4+1}{9+3}\cdot\frac{3+1}{9+3} \approx \frac{2}{9}\cdot\frac{4}{9}\cdot\frac{3}{9} \tag{3.82}$$

$$\frac{2+\frac{m}{3}}{9+m}\cdot\frac{4+\frac{m}{3}}{9+m}\cdot\frac{3+\frac{m}{3}}{9+m} \approx \frac{2}{9}\cdot\frac{4}{9}\cdot\frac{3}{9} \tag{3.83}$$

3.4.4.8 MÉTODOS ESTADÍSTICOS Y NUMÉRICOS DE REGRESIÓN

El ajuste de modelos por técnicas clásicas puede llevarse a cabo atendiendo a distintos patrones de ajuste. Básicamente se establecen cinco tipos de ajustes:

- Modelos lineales.
- Modelos de análisis de la varianza.
- Modelos lineales generalizados.
- Modelos aditivos generalizados.
- Modelos de regresión local.

Los modelos lineales de regresión son aplicables a predictores continuos y categóricos que explican variables continuas. Se formulan matemáticamente como un modelo donde, para cada observación i, $i=1..N$, el valor y_i de la variable a explicar, se ajusta linealmente a los valores observados de las variables predictoras x_{ij}. El error cometido por este ajuste se absorbe en el residuo e_i.

$$y_i = b_0 + \sum_{j=1}^{N} b_j x_{ij} + e_i = \hat{y}_i + e_i \tag{3.84}$$

En media, la mejor predicción de la variable a explicar se obtiene mediante una ecuación del tipo,

$$y_i = b_0 + \sum_{j=1}^{N} b_j x_{ij} \tag{3.85}$$

El siguiente paso a la hora de proponer un modelo más complejo es proponer la posible transformación de la variable a explicar con un modelo lineal generalizado. Estos modelos son aplicables a predictores tanto continuos como categóricos en la explicación de variables tanto continuas como categóricas. Se asume, eso sí, como condición que la varianza de la variable a explicar sea función de su media.

$$h\left(E\left(y\right)\right) = b_0 + \sum_{j=1}^{N} b_j x_j \tag{3.86}$$

$$s_y^2 = fV\left(m\right) \tag{3.87}$$

Los modelos lineales generalizados permiten modelar datos que siguen distribuciones como la *normal*, *binomial*, *Poisson*, *gamma* y *normal inversa*, pero aun requieren la relación lineal en los parámetros.

Los modelos aditivos generalizados van más allá y permiten formular relaciones más complejas en las variables predictoras, a través de funciones no lineales (ver Figura 86).

$$h(E(y)) = \sum_{j=1}^{N} f_j(x_j) \tag{3.88}$$

El tipo de variables predictoras y explicadas es el mismo que para los modelos generalizados.

Si se permite la interacción entre distintas variables se llega a los modelos de regresión local generalizada.

$$y_i = g(x_{i1}, x_{i2}, \ldots, x_{in}) + e \tag{3.89}$$

El tipo de variables predictoras explicadas es el mismo que para los modelos generalizados y modelos aditivos.

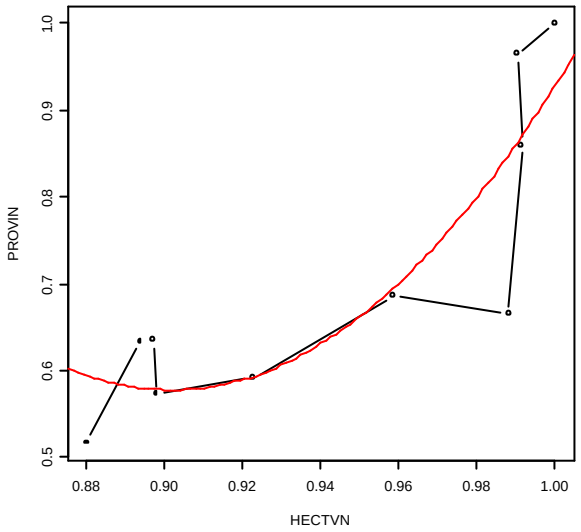


Figura 86. Ajuste no lineal de una serie de puntos.

Los modelos clásicos de regresión se basan en hipótesis que pudiera ser que no cumplieran los datos, tal es el caso de la condición de normalidad. Además los modelos obtenidos pueden verse desplazados en presencia de *outliers*. En ese caso es adecuada la utilización de métodos robustos que no se vean afectados por la debilidad de los datos.

AJUSTE DE MODELOS LINEALES POR TÉCNICAS CLÁSICAS

El ajuste de un modelo lineal a unos datos puede enfocarse desde dos puntos de vista.

- Regresión Lineal Multivariada.
- Ajuste de Mínimos Cuadrados.

Ambos enfoques conducen a un mismo modelo. Se detallará a continuación los fundamentos del ajuste desde el punto de vista de regresión lineal multivariada.

Regresión Lineal Multivariada

El vector correspondiente a una determinada observación x, de dimensión $(n\times 1)$ puede ser dividido en dos partes

$$x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.90}$$

donde x_1 hace referencia a las primeras q coordenadas de x, y x_2 denota las ultimas $s=(n-q)$ coordenadas. x sigue entonces una distribución cuyas matrices de media y covarianzas pueden ser expresadas como

$$m=\begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \text{ y } \Sigma=\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{3.91}$$

donde m_1 es $(q\times 1)$, m_2 es $(s\times 1)$, Σ_{11} es $(q\times q)$, Σ_{12} es $(q\times s)$ y $\Sigma_{21}=\Sigma_{12}^t$. Así definidas la distribución marginal que sigue x_1 es $N_q(m_1,\Sigma_{11})$. La distribución marginal de x_2 es $N_s(m_2,\Sigma_{22})$. Asimismo la distribución de x_2 condicionada a $x_1=x_1^*$ es una normal multivariante con vector de medias

$$m_{2\cdot 1}(x_1^*)=(m_2-\Sigma_{21}\Sigma_{11}^{-1}m_1)+\Sigma_{21}\Sigma_{21}^{-1}x_1^* \tag{3.92}$$

y matriz de covarianzas

$$\Sigma_{22\cdot 1}=(\Sigma_{22}-\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}) \tag{3.93}$$

Nótese que el vector de medias condicionado es función de x_1^*, no así la matriz de varianzas condicionada.

La función de densidad de probabilidad x_2 condicionada a $x_1=x_1^*$ es:

$$f(x_2\mid x_1=x_1^*)=(2p)^{-\frac{s}{2}}|\Sigma_{22\cdot 1}|^{-\frac{1}{2}}e^{-\frac{1}{2}(x_2-m_{2\cdot 1})'\Sigma_{22\cdot 1}^{-1}(x_2-m_{2\cdot 1})} \tag{3.94}$$

Por tanto la función de regresión multivariada de x_2 en función de x_1 es lineal de la forma $m_{2\cdot 1}(x_1)=b_0^t+x_1^tB^*$ donde:

$$b_0^t=(m_2-\Sigma_{21}\Sigma_{11}^{-1}m_1)^t\text{: Vector } (1\times s) \text{ de intersección con el origen.} \tag{3.95}$$

$$B^*=\Sigma_{11}^{-1}\Sigma_{12}\text{: Matriz } (q\times s) \text{de coeficientes de pendiente.} \tag{3.96}$$

Utilizándose de forma conjunta como:

$$B=\begin{bmatrix} b_o^t \\ B^* \end{bmatrix} \tag{3.97}$$

Para utilizar una notación más convencional puede llamarse:

$$y_{(s\times 1)} = x_2 \tag{3.98}$$

$$x_{((q+1)\times 1)} = \begin{bmatrix} 1 \\ x_1 \end{bmatrix} \tag{3.99}$$

$$m_{2\cdot 1}^t = xB \tag{3.100}$$

$$u^t = y^t - m_{2\cdot 1} \tag{3.101}$$

y entonces

$$y^t = x^t B + u^t \tag{3.102}$$

De esta manera el valor de y se descompone en dos partes: la proyección sobre el espacio vectorial generado por las columnas de x y el residuo u, complemento ortogonal al anterior.

La matriz de covarianzas de u es

$$\Sigma_{yy\cdot x} = \Sigma_{yy} - \Sigma_{yz_1}\Sigma_{x_1x_1}^{-1}\Sigma_{x_1y} \tag{3.103}$$

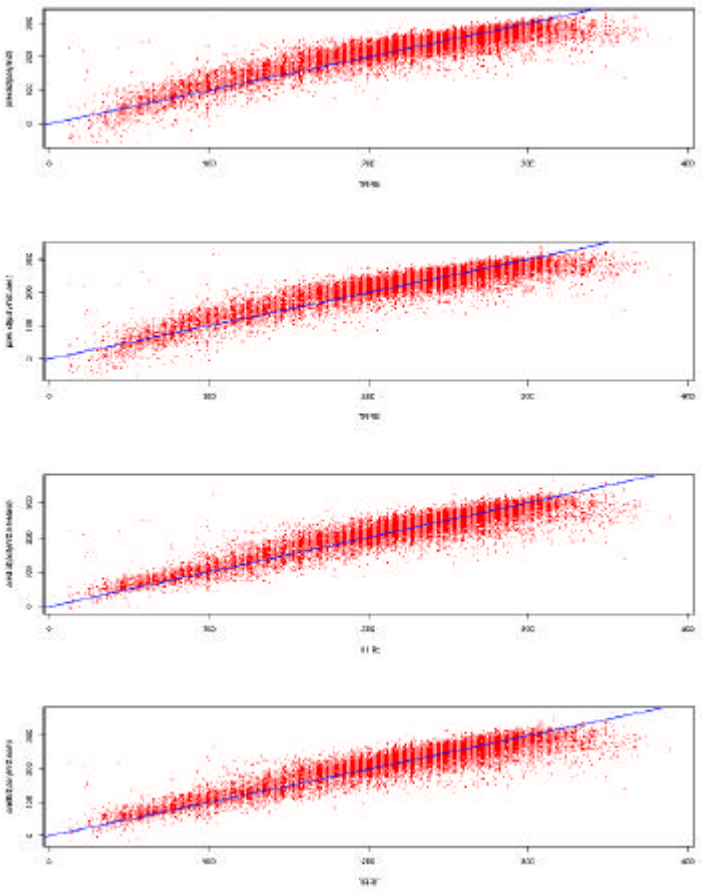


Figura 87. Obtención de un modelo lineal a partir de un banco de observaciones.

La componente correspondiente a los residuos debe ser analizada cuidadosamente una vez estimado el modelo, pues de su estudio se desprenderá la información necesaria para criticar la bondad del modelo.

La otra vía de aproximación al problema de la predicción lineal es el de ajuste por mínimos cuadrados.

El Ajuste por Mínimos Cuadrados

Este método se basa en calcular la ecuación de una curva para una serie de puntos dispersos sobre una gráfica. Esta curva se considera el mejor ajuste, entendiéndose por tal, aquel en el que la suma del cuadrado de las desviaciones de los puntos individuales respecto a la media es cero.

Para ello, se supone que los pares de puntos ajustados se asemejan a una recta, cuya ecuación es:

$$\hat{Y} = a \cdot X + b \tag{3.104}$$

donde b es la desviación al origen de la recta, a es la pendiente de la recta, X_i es el valor dado de la variable X (el tiempo) y $\hat{Y}$ es el valor calculado de la variable Y (demanda, oferta, etc.).

De esta forma se cambia de base a x: $x = X - \overline{X}$ y se calcula la recta de regresión de la siguiente forma:

$$b = \overline{Y} \qquad a = \frac{\sum Y_i \cdot x_i}{\sum x_i^2} \tag{3.105}$$

A pesar de los escrito en la teoría estadística sobre el método de mínimos cuadrados, a veces trabajar con dos variables no es muy útil al hacer un estudio de mercado. El tiempo como variable independiente no influye por sí mismo en el comportamiento de una variable como la oferta o la demanda. Esto quiere decir que existe la necesidad de considerar una u varias variables más, además de las dos mencionadas (T, D), que verdaderamente influyan en forma directa en el comportamiento de la variable dependiente (demanda u oferta).

El objetivo es, en base a una serie de variables independientes, tratar de estimar los coeficientes que explican el hiperplano de regresión:

$$\hat{Z}_i = a + b \cdot x_i + g \cdot y_i \tag{3.106}$$

de modo que el problema se presenta con objetivo de buscar a, b y g que cumplan:

$$\min \sum \left(Z_i - \hat{Z}_i \right)^2 \tag{3.107}$$

Generalizando, se plantea el siguiente modelo:

$$Y_{(N\times s)} = X_{(N\times(q+1))}B_{((q+1)\times s)} + U_{(N\times s)} \tag{3.108}$$

La solución óptima será aquella que minimice la suma de los cuadrados de las desviaciones producidas entre el valor estimado y el real de las variables a explicar. La solución emanará por tanto de

$$\frac{d\sum_{j=1}^{s}\sum_{i=1}^{N}\left(y_{ij}-\bar{y}_s\right)^2}{dB} = 0 \tag{3.109}$$

El estimador obtenido por esta vía de

$$B = \begin{bmatrix} b_o^t \\ B^* \end{bmatrix} \text{ es } \hat{B} = \left(X^t X\right)^{-1} X^t Y \tag{3.110}$$

que también puede ser escrito como

$$\hat{B} = \begin{bmatrix} \bar{y}^t - \bar{x}_{\cdot 1}^t S_{x_1x_1}^{-1} S_{x_1y} \\ S_{x_1x_1}^{-1} S_{x_1y} \end{bmatrix} \tag{3.111}$$

resultado que concuerda con el obtenido por la vía estadística. Sin embargo, la aproximación estadística proporciona un entorno más robusto para profundizar en la estructura de los residuos y aplicar técnicas de inferencia estadística como, por ejemplo, la estimación de intervalos de confianza de los coeficientes obtenidos.

Un estimador no sesgado de la matriz de covarianzas del residuo, $\Gamma = \Sigma_{yy\cdot x_1}$, viene dado por

$$\hat{\Gamma} = \frac{Y - X\hat{B}}{N - q - 1} \tag{3.112}$$

A la hora de seleccionar qué variables resultan más adecuadas en nuestro modelo, es útil definir el valor $AIC = \hat{s}^2\left(C_p + N\right)$, estimación que depende del estadístico C_p de *Mallow*. Este valor indica qué se debe hacer con cada una de las variables predictoras involucradas. El error cuadrático medio puede ser escrito como la suma de la varianza y el cuadrado del sesgo. No obstante, en una selección de variables es posible optimizar el error cuadrático medio y aun así continuar con un fuerte sesgo. El estadístico C_p de *Mallow* tiene en cuenta este efecto. El estadístico C_j representa el valor del estadístico C_p cuando se consideran j variables explicativas,

$$C_j = (N - j - 1)\frac{MSE_j}{MSE} - \left[N - 2(j+1)\right] \tag{3.113}$$

donde MSE y MSE_j representan respectivamente el valor del error cuadrático medio del modelo completo y del modelo que considera únicamente j variables explicativas.

Si el valor *AIC* correspondiente a una determinada variable es inferior al correspondiente al modelo completo, resulta favorable su inclusión o exclusión. En este caso parece adecuado incluir en el modelo los efectos del resto de variables.

CLASIFICACIÓN DE LOS MÉTODOS DE REGRESIÓN

El objetivo de los métodos de regresión [KEI94][HOS89][NET96][DRA98] es estimar una función $\hat{f}$ que proyecte los puntos o vectores de las características desde un espacio de entrada $X \subset R^n$, a un espacio de salida numérico o continuo $Y \subset R^m$, a partir de una muestra (conocida) finita de tamaño *N* de la proyección, $\{y^i, x^i\}_{j=1}^N \subset \Re^{N,m+n}$:

$$f : X \subset R^n \rightarrow Y \subset R^m \qquad (3.114)$$
$$x \rightarrow f(x) = y$$

El sistema generado por los datos será del tipo:

$$y^i = f(x^i) + e \qquad (3.115)$$

donde

$$x=(x_1,x_2,...,x_n) \in D \subset \Re^n \qquad (3.116)$$
$$y=(y_1,y_2,...,y_n) \in D \subset \Re^m$$

La función f captura la influencia de las características de x sobre la salida y. La componente aditiva estocástica ε, cuyo valor medio tiende a cero, refleja la dependencia de y respecto a otras características distintas de x que no son ni controladas ni medidas. El objetivo de los métodos de regresión es utilizar los datos para construir una función $\hat{f}(x)$ que pueda servir como una aproximación razonable de la función *f(x)* sobre el dominio *D* de interés. Por aproximación razonable se entiende que los resultados obtenidos con la función $\hat{f}(x)$ sean similares a los que se obtendrían con la función *f(x)*, es decir que el nivel de generalización de $\hat{f}(x)$ sea adecuado, permitiendo utilizar $\hat{f}(x)$ para aproximar el valor de f en puntos desconocidos. Además se pretende que la función $\hat{f}$ sea interpretable, permitiendo comprender las propiedades de f, así como fácil de evaluar. Por último, para determinados casos se exigirá que la función $\hat{f}(x)$ sea una función suave de sus argumentos; esto es, que tenga derivadas de bajo grado para cualquier punto de *D*.

A continuación se repasan los métodos habituales de regresión en función del método de búsqueda utilizado.

APROXIMACIÓN PARAMÉTRICA

La aproximación paramétrica consiste en aproximar la función $\hat{f}(x)$ mediante una función paramétrica del tipo $\hat{f}(x) = g\left(x \middle| \{a_j\}_1^p\right)$.La estimación de los parámetros $\{a_j\}_1^p$ de la función g se calcula mediante el método de mínimos cuadrados aplicado a los datos de entrenamiento. Es decir:

$$\hat{f}(x) = g\left(x \middle| \{a_j\}_1^p\right) \tag{3.117}$$

donde la estimación de los parámetros viene dada por:

$$\{a_j\}_1^p = \min_{\{a_j\}_1^p \in R^p} \sum_{i=1}^{N} \left[y^i - g\left(x^i \middle| \{a_j\}_1^p\right)\right]^2 \tag{3.118}$$

Existen numerosas parametrizaciones utilizadas para representar la función *g*. La parametrización más simple utilizada es la lineal:

$$g\left(x^i \middle| \{a_j\}_1^p\right) = a_0 + \sum_{j=1}^{p} a_j \cdot x_j; \qquad p \le n \tag{3.119}$$

Esta parametrización no produce una aproximación demasiado buena, pero tiene la virtud de requerir relativamente pocos datos, es fácil de interpretar y se evalúa rápidamente. Además, si el componente aleatorio ε es importante comparado con *f*, la varianza del estimador predomina, y el error sistemático asociado con la aproximación carece de importancia.

El conocimiento previo del problema puede resultar de gran ayuda en la selección de la parametrización de la función. En general debido a la limitación de flexibilidad de las aproximaciones de tipo paramétrico, únicamente se obtiene una aproximación convergente de] problema si la forma de la función *f(x)* es del tipo representado por la parametrización.

Para problemas de dimensionalidad baja ($n \le 2$), los métodos paramétricos se han generalizado, obteniéndose excelentes resultados, mediante el uso de tres paradigmas: segmentación, promedios locales y métodos de penalización.

SEGMENTACIÓN

La idea básica de la interpolación polinomial segmentaria consiste en aproximar *f* mediante varios polinomios de orden bajo, cada uno definido sobre una subregión (o intervalo) distinto del dominio *D*. La aproximación obtenida es continua y, dependiendo del orden de los polinomios, mantiene también continuas sus derivadas. El compromiso entre suavidad y flexibilidad de la función obtenida se controla mediante el número de subregiones (número de nodos) y el orden de discontinuidad permitido en la vecindad de los nodos. Las funciones polinomiales segmentarias más usuales son los *splines* [DEB78].

Una función *spline* está formada por varios polinomios de grado *q*, cada uno definido sobre una subregión, unidos entre sí de forma que la curva resultante sea derivable *q-1* veces con derivadas continuas. La elección más usual para *q* es 3. en cuyo caso, al *spline* se lo denomina *spline* cúbico.

La ventaja de los *splines* frente a la interpolación polinómica clásica estriba en la ausencia de inestabilidad para cualquier número de puntos. Presentan también una ventaja mecánica añadida: "de todas las curvas que pasan por esos puntos, las *splines* son las que minimizan la curvatura".

Una generalización de los *splines* se puede realizar considerando curvas de interpolación a trozos donde los puntos se utilizan para controlar la curva, pero sin exigirle que pase por ella. Es decir, estos puntos dan forma a la curva pero a excepción del punto inicial y del final, la curva no pasa por ellos. Esta generalización de los *spline* se denomina *B-spline*. En función del comportamiento del *B-spline* en los puntos inicial y final, se distingue entre *B-spline* abiertos (si el *B-spline* pasa por estos dos puntos) y *B-splines* uniformes (si el *B-spline* no llegan a tocarlos).

Promedios locales

Las aproximaciones mediante promedios locales son de la forma:

$$\hat{f}(x) = \sum_{i=1}^{N} K(x, x^i) y^i \tag{3.120}$$

donde $K(x, x^i)$, llamada función de kernel, tiene un valor máximo para $x = x^i$, y va decreciendo a medida que aumenta la distancia entre x y x^i. Así, $\hat{f}$ se obtiene como un promedio compensado de los valores de y^i donde los pesos son mayores a medida que se acerca a puntos donde el valor de la función es conocido. Cuando $n>l$, se suele tomar como kernel la distancia euclídea entre los puntos:

$$K(x, z) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2} \tag{3.121}$$

Los métodos de promedios locales han recibido atención constante en la literatura estadística desde su introducción por *Parzen* en 1962 [PAR62]. Los métodos de promedios locales tienen propiedades asintóticas deseables [STO77].

Métodos de penalización

Las aproximaciones mediante penalización se definen como:

$$\hat{f}(x) = \min \left\{ \sum_{i=1}^{N} \left[y^i - g(x^i) \right]^2 + l \cdot R(g) \right\} \tag{3.122}$$

donde *R(g)* es un funcional que se incrementa cuando disminuye la suavidad de la función *g*. La minimización se realiza entre todas las funciones *g* para las cuales se encuentra definida *R(g)*. El parámetro l controla la relación entre la suavidad de *g* y su fidelidad a los datos.

El funcional *R(g)* más estudiado y utilizado en métodos de penalización es la integral del cuadrado del *laplaciano*:

$$R(g) = \sum_{k=1}^{n} \sum_{l=1}^{n} \int \left| \frac{\partial^2 g}{\partial x_k \partial x_l} \right|^2 dx \tag{3.123}$$

Las propiedades de los métodos de penalización son similares a las de los métodos de kernel basados en distancias euclídeas [SIL85].

APROXIMACIÓN NO PARAMÉTRICA

La generalización de los métodos de segmentación y promedios locales a espacios de mayor dimensión es directa en la teoría, pero difícil en la práctica. Las dificultades vienen debidas a la maldición de la dimensionalidad, que como ya se ha explicado, expresa el hecho de que para poder poblar de forma densa los espacios euclídeos se necesite un número de puntos que crece exponencialmente con la dimensión del espacio. En el caso de aproximación mediante *splines*, la generalización viene dada como el producto tensorial de funciones *spline* de una dimensión. Esas funciones vienen asociadas a una malla de puntos obtenida mediante el producto externo de las posiciones de los nodos para cada variable independiente.

Los métodos de promedio local padecen problemas similares cuando aumenta la dimensión del espacio. Por ejemplo, consideremos D como un hipercubo unitario de R^n y tomemos un kernel uniforme con distancia lineal y funciones asintóticas a un 10%, del ancho de D. Entonces, suponiendo que los datos estén uniformemente distribuidos, el kernel sólo contendrá $0.1 \cdot n$ datos de la muestra en promedio. Lo cual hace que esté casi vacío para valores moderados de n. Si se pretende ajustar el tamaño del entorno (ancho de banda del kernel) para que contenga el 10% de la muestra, cubrirá (en promedio) el $100 \cdot \frac{0.1}{n}$ del rango en cada variable, resultando una aproximación demasiado ruda.

El problema inherente de la difusión de los datos en las muestras cuando se trabaja con funciones de dimensión elevada, limita drásticamente la generalización de los métodos anteriores por lo que es necesaria la introducción de otro tipo de métodos no paramétricos.

Modelo Aditivo

Recientemente, se ha generalizado el modelo paramétrico lineal al modelo no paramétrico conocido como modelo aditivo. El modelo aditivo [HAS90] construye la función $\hat{f}$ mediante la siguiente expresión:

$$\hat{f}(x_1, x_2, ..., x_n) = a_0 + \sum_{i=1}^{n} g_i(x_i) \qquad (3.124)$$

donde las funciones $\{g_i(x_i)\}_{i=1}^{n}$ son funciones unidimensionales suaves, pero diferentes. Estas funciones son desconocidas y tienen que ser calculadas a partir de los datos.

Para estimar cada g_i se necesita una muestra de tamaño razonablemente grande, pero como las funciones g_i son unidimensionales, los requerimientos de tamaño de la muestra crecen solamente de forma lineal respecto a la dimensión (este método no sufre la maldición de la dimensionalidad). Aunque los modelos aditivos no son capaces de obtener aproximaciones adecuadas para funciones muy generales de R^n resultan fáciles de interpretar y evaluar.

Computación Adaptativa

La computación adaptativa es aquella que ajusta dinámicamente su estrategia para tener en cuenta el comportamiento del problema específico que intenta resolver. La computación adaptativa

no se utiliza únicamente para la regresión de funciones en altas dimensiones, sino que se tienen ejemplos de algoritmos adaptativos en el cálculo de integrales mediante cuadratura numérica [FRI81]. En estadística, los algoritmos adaptativos para aproximar funciones se han basado en:

- **Proyección activa.** Ya hemos visto anteriormente, que el modelo PPR, (*Projection Pursut Regresión*) [FRI81] construye la función $\hat{f}(x_1, x_2, ..., x_n)$ mediante la siguiente expresión:

$$\hat{f}(x_1, x_2, ..., x_n) = \sum_{m=1}^{M} g_m \left(\sum_{i=1}^{n} a_{im} \cdot x_i \right) \tag{3.125}$$

es decir, mediante funciones aditivas de combinaciones lineales de las variables. Como en el modelo aditivo, a la función univariable g_m se le exige simplemente que sea suave. Estas funciones, y los coeficientes correspondientes a las combinaciones lineales, son optimizadas de forma conjunta para producir un buen ajuste de los datos basado en la minoración de alguna distancia. Se demuestra [DIA84] que cualquier función suave de *n* variables puede representarse de esta forma para un valor suficientemente grande de *M*. La ventaja de este enfoque está en que incluso para valores moderados de *M* muchas clases de funciones pueden aproximarse de esta manera [DON85]. Otra ventaja de esta proyección está en su equivarianza afín, esto es, la solución es invariante frente a cualquier transformación afín no singular (rotaciones) de las variables originales. Es el único método práctico que parece poseer esta propiedad.

- **Proyección recursiva**: El algoritmo de la partición recursiva ajusta un modelo de la forma:

$$\hat{f}(x_1, x_2, ..., x_n) = \sum_{m=1}^{M} a_i \cdot B_i(x) \tag{3.126}$$

donde x y a_i son los coeficientes de las funciones base B_i y M es el número de funciones base del modelo.

Las funciones base utilizadas por el método de proyección recursiva son de1 tipo:

$$B_i(x) = I_{R_{ij}}(x) = \begin{cases} 1 & x \in R_i \\ 0 & x \notin R_i \end{cases} \tag{3.127}$$

donde R_1, R_2, ..., R_m son regiones rectangulares que realizan una partición de $\hat{A}^p$.

Los datos son utilizados para estimar de forma simultánea las subregiones R_i y los parámetros asociados con las funciones construidas sobre cada región. Para la construcción de las subregiones se procede realizando particiones recursivas: Inicialmente la partición es *D*. Para cada paso del algoritmo, todos los intervalos existentes son divididos de forma óptima en dos subregiones, iterando el proceso hasta conseguir un gran número de subregiones. En la segunda fase del algoritmo, las

subregiones se van recombinando entre sí, mediante criterios de penalización del número de subregiones y grado de convergencia, hasta lograr un conjunto óptimo.

El principal inconveniente de los métodos de proyección recursiva reside en la discontinuidad de la función de aproximación obtenida.

Regresión Multivariante Adaptativa con Splines (MARS)

Un problema común en muchas disciplinas es la adecuada aproximación de funciones de muchas variables, conocido únicamente el valor de dicha función en un reducido grupo de puntos del espacio de la variable independiente y, a menudo, perturbado por el ruido. El objetivo es encontrar el modelo de dependencia entre la variable respuesta y las variables de entrada $x_i,...,x_n$ una vez que se han realizado unas muestras $\{y_i, x_i,..., x_n\}_1^N$. El sistema que genera los datos se puede describir como:

$$y = f(x_1,..., x_n) + \varepsilon \tag{3.128}$$

sobre un dominio $(x_1, K, x_n) \in D \subset R^n$, el cual, contiene los datos.

Donde la función f relaciona la variable de salida con las variables de entrada y ε es el ruido estocástico. El objetivo del análisis de regresión es encontrar una función $\tilde{f}(x_1, L, x_n)$ que sirva como una razonable aproximación de $f(x_1, \Lambda, x_n)$ sobre el dominio D de interés.

Para ello se considera un tipo de funciones denominadas *funciones básicas* B_m de la forma:

$$B_m(x) = I[x \in R_m] \tag{3.129}$$

I es una función que toma el valor 1 si el argumento es cierto, y el valor 0 en caso contrario. Los $\{a_m\}_1^M$ son los coeficientes de expansión cuyos valores son ajustados para obtener una buena adaptación a los datos. Los $\{R_m\}_1^M$ son las subregiones de espacio donde está definida la función. Si estas subregiones son disjuntas, sólo una función básica es distinta de 0 para cada x.

La principal limitación del método anterior es su falta de continuidad entre subregiones vecinas. Esta falta de continuidad limita severamente la precisión de la adaptación. Para conseguir modelos continuos, con derivadas continuas, se desarrollo el método de *splines* regresivos adaptativos (*Multivariable Adaptative Regressive Splines*, MARS).

El método MARS, [FRI91][CHE99][STE01] combina los métodos de proyección activa y proyección recursiva, utilizando regresión multivariante adaptativa con *splines*. El modelo utilizado por el MARS es el mismo que el dado en proyección recursiva. pero con esas funciones base distintas, tal y como hemos visto anteriormente. Las funciones base utilizadas por el MARS son *splines* multivariantes, es decir, el producto tensorial de funciones *spline* de una dimensión.

El único aspecto que introduce discontinuidades en el modelo es la función escalón. Si se reemplaza esta función por otra que sea continua, el algoritmo 1 debería de producir modelos continuos. La función elegida para reemplazar a la función escalón es un *spline*.

Las dos partes de la división de la función básica tienen la forma:

$$b_q^{\pm}(x-t^n)=\left[\pm(x-t^n)\right]_+^{q_S} \tag{3.130}$$

donde t^n es la localización del nodo, q_S es el orden del *spline*, y el subíndice indica la parte positiva del argumento.

Para $q_S > 0$ la aproximación por *splines* es continua, y con q_S *-1* derivadas continuas.

Las funciones escalón son un caso particular en que los *splines* son de grado cero, $q_S = 0$.

Este método produce unas funciones básicas son el producto de *splines* univariantes. Estas funciones básicas tienen la forma:

$$B_m^{(q)}(x)=\prod_{K=1}^{K_m}\left[s_{km}\cdot\left(x_{v(k,m)}-t_{km}\right)\right] \tag{3.131}$$

Es decir, reemplazando las funciones escalón por *splines* de grado q_S, se consiguen modelos continuos, con q_S *-1* derivadas continuas.

El modelo MARS se escribe de la siguiente forma:

$$\tilde{f}(x)=a_0+\sum_{K_m=1} f_i(x_i)+\sum_{K_m=2} f_{ij}(x_i,x_j)+\sum_{K_m=3} f_{ijk}(x_i,x_j,x_k)+L \tag{3.132}$$

El primer sumatorio contiene todas aquellas funciones que dependen de una sola variable. El segundo contiene las funciones básicas que dependen de dos variables, y representa las interacciones entre dos variables. El tercer sumatorio representa la contribución de las interacciones entre tres variables, y así sucesivamente.

Sea $V(m)=\{v(k,m)\}_1^{Km}$ el conjunto de variables asociada con la función básica m, $B_m(x)$.
Cada función del primer sumatorio puede ser expresada como

$$f_i(x_i)=\sum_{\substack{K_m=1\\ i\in V(m)}} a_m B_m(x_i) \tag{3.133}$$

Esto es la suma de todas las funciones básicas que envuelven solamente la variable x_i y es el *spline* que representa la función univariante correspondiente.

Cada función bivariante del segundo sumatorio puede ser expresada como:

$$f_i(x_i,x_j)=\sum_{\substack{K_m=2\\ (i,j)\in V(m)}} a_m B_m(x_i,x_j) \tag{3.134}$$

Lo cual representa la suma de todas las funciones básicas que envuelven un determinado par de variables x_i y x_j. Sumándole la correspondiente contribución univariante para esas mismas variables se tendrá:

$$f_i(x_i, x_j) = \sum_{\substack{K_m=2 \\ (i,j)\in V(m)}} a_m B_m(x_i, x_j) \tag{3.135}$$

que representa el conjunto de la contribución bivariante de x_i y x_j al modelo. Procediendo de la misma manera se obtienes las contribuciones de los términos correspondientes a grupos de tres variables y más variables.

3.4.4.9 MÉTODOS BASADOS EN COMPUTACIÓN EVOLUTIVA

La idea de computación evolutiva fue introducida por primera vez en los años 60, por el Dr. *Rechenberg* en su ensayo *Evolution Strategies*. Las estrategias de computación evolutiva suponen un enfoque alternativo para abordar problemas complejos de búsqueda y aprendizaje a través de modelos computacionales de procesos evolutivos. Las implantaciones concretas de tales estrategias se conocen como algoritmos evolutivos [DAV91] [HOL92][OPE01].

La principal aportación de la computación evolutiva a la metodología de resolución de problemas, consiste en el uso de mecanismos de selección de soluciones potenciales y de construcción de nuevos candidatos por recombinación de características de otros ya presentes, de modo parecido a como ocurre en la evolución de los organismos naturales. Una de las características de la Naturaleza es la existencia de organismos adaptados para la supervivencia en prácticamente cualquier ecosistema, incluso en los más inhóspitos. El medio ambiente se encuentra sometido a continuos cambios, lo cual motiva que ciertas especies se extingan y otras evolucionen y adquieran preponderancia gracias a su adaptación a la nueva situación. El motor del cambio está controlado por mecanismos supraorgánicos como la selección natural. Los algoritmos evolutivos surgen al intentar trasladar la versatilidad y elegancia de estos procesos naturales al terreno algorítmico.

Para ello se identifica el espacio n-dimensional de posibles soluciones (población inicial) en un lenguaje de representación de conocimiento denominado cromosoma (cadena cromosómica o genotipo). El proceso desarrollado por los algoritmos evolutivos es estocástico e iterativo. Inicialmente se crea de forma aleatoria o mediante algún heurístico de construcción una población inicial de individuos, cada uno de los cuales contiene uno o más cromosomas. Dichos cromosomas permiten que cada individuo represente una posible solución al problema que se está considerando. Un proceso de codificación/decodificación (ρ) permite obtener la solución que los cromosomas de cada individuo contienen.

Cada uno de los individuos de la población recibe, a través de una función de adecuación (*fitness*), una medida de su bondad con respecto al problema que se desea resolver. Este valor es empleado por el algoritmo para guiar la búsqueda.

El algoritmo está estructurado en tres fases principales que se ejecutan de manera circular, selección (s), reproducción (w_*) y reemplazo (y), las cuales se llevan a cabo de manera repetitiva. Cada una de las iteraciones del algoritmo se denomina ciclo reproductivo básico o generación.

Durante la fase de selección se crea una población temporal P' en la que aquellos individuos más aptos (los correspondientes a las mejores soluciones contenidas en la población) estarán representados un mayor número de veces que los poco aptos (principio de selección natural). En la fase de reproducción se aplican sobre los individuos contenidos en la población temporal, P', diferentes operadores de cambio (también denominados operadores reproductivos). El objetivo de esta fase es producir individuos con nuevas características, idealmente mejores (principio de

adaptación). Finalmente, durante la fase de reemplazo, se sustituyen individuos de la población original por los nuevos individuos creados. Este reemplazo afecta a los peores individuos y tiende a conservar los mejores (supervivencia de los más adaptados). Todo este proceso descrito se repite hasta que se cumple un determinado criterio de terminación (normalmente al completar un cierto número de iteraciones).

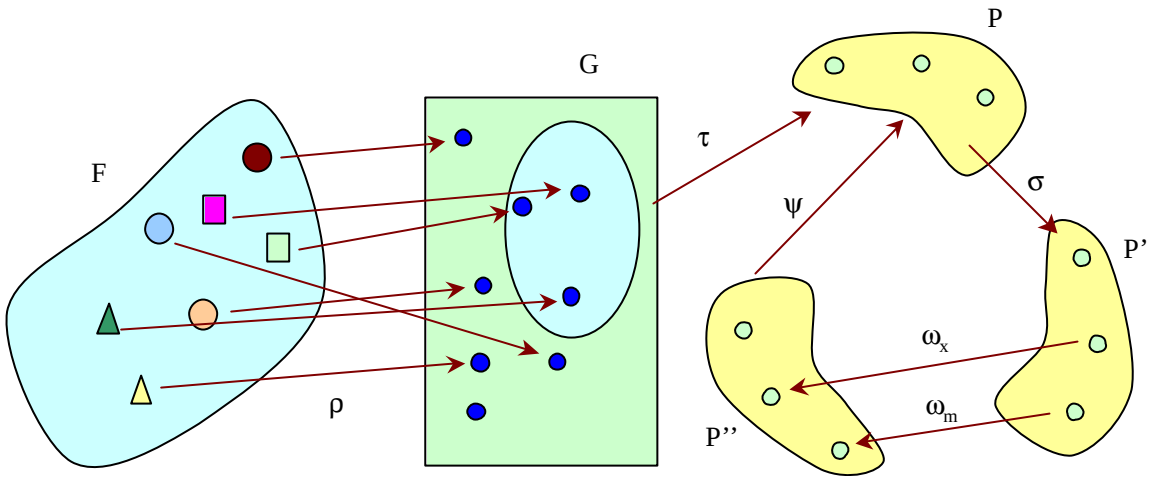


Figura 88. Esquema de un algoritmo evolutivo.

Sobre los algoritmos evolutivos se han desarrollado dos paradigmas fundamentales:

Los Algoritmos Genéticos: Se hace evolucionar una población de enteros binarios sometiéndolos a transformaciones unitarias y binarias genéricas y a un proceso de selección. Estas técnicas son probablemente el representante más conocido de los algoritmos evolutivos, y aquellas cuyo uso está más extendido.

Las Estrategias Evolutivas: Su objetivo inicial era servir de herramienta para optimización de parámetros en problemas de ingeniería. Debido a este objetivo primordial, estas técnicas se caracterizan por manejar vectores de números codificados en punto flotante (si bien existen versiones de las mismas que se aplican a problemas discretos).

ALGORITMOS GENÉTICOS

El concepto de algoritmo genético fue introducido en 1975 por *John Holland* en el libro *Adaption in Natural and Artificial System* [BLI97].

Los algoritmos genéticos constituyen el paradigma más completo de la computación evolutiva, tanto a nivel teórico como a nivel práctico, debido a las siguientes características:

- Flexibilidad: pueden adoptar con facilidad nuevas ideas, generales o específicas, que surjan en el campo de la computación evolutiva, pudiendo además hibridarse con otros paradigmas y enfoques.

- Sencillez: la base teórica es sencilla en su desarrollo y con grandes posibilidades de ampliación.
- Versatilidad: necesitan poco conocimiento específico, pudiendo sin embargo incorporar conocimiento específico con poco esfuerzo adicional.
- Fácil implementación.

Como mayor desventaja encontramos la elevada capacidad de procesado y tiempo necesarios habitualmente para llegar a una solución óptima.

Un algoritmo genético tradicional puede ser definido como una tupla con los siguientes elementos: GA$(S, l, f, I, S, R, X, K_X, M, K_M, t)$,

Donde:

- Σ es el alfabeto binario empleado para construir los cromosomas ($\Sigma=\{0,1\}$)
- λ es la longitud de las cadenas de cromosomas. Dichos cromosomas forman el espacio de genotipos, $G \rightarrow \Sigma^l$.
- f es la función de adecuación que relaciona los cromosomas con la bondad de las soluciones que representan.
- I es el operador de inicialización. responsable de generar los individuos que formarán parte de la población inicial. Usualmente, dichos individuos se generan de manera totalmente aleatoria, aunque es posible considerar otras alternativas.
- S es el operador de selección que determina los cromosomas presentes en la población que pasan a la fase reproductiva.
- R es el operador de reemplazo cuya misión es formar una nueva población a partir de la ya existente y de los nuevos individuos creados durante la fase reproductiva.
- X y K_X son los operadores de cruce y parámetros del mismo respectivamente. La misión del operador de cruce es producir nuevas soluciones a partir de la combinación de porciones de soluciones ya existentes, es decir, $X: G^{k1} \times K_X \text{ ® } G^{k2}$.
- M y $K_M \rightarrow K_M$ indica la proporción de posiciones de la cadena que sufrirán modificación.
- t es el criterio de terminación que determina cuando se debe concluir la ejecución del algoritmo.

A continuación se describen de forma más detallada los operadores más importantes que intervienen en los algoritmos genéticos: selección, cruce, mutación y reemplazamiento.

Selección

Consiste en seleccionar dos padres de la población original de acuerdo al grado de bondad que aporten al problema (a mayor bondad mayores oportunidades de ser seleccionado).

La selección puede realizarse mediante tres mecanismos:

- Selección proporcional a la adecuación (*fitness-proportionate selection*): Cada componente de la población seleccionada para reproducirse es escogido de la población actual (de tamaño N) de manera independiente, realizando una selección aleatoria en la que el individuo i-ésimo cuya adecuación es f_i tiene una probabilidad p_i de ser seleccionado de:

$$p_i = \frac{f_i}{\sum_{1 \le j \le N} f_j} \tag{3.136}$$

- Selección por ranking: la población seleccionada se construye atendiendo a la ordenación de los valores de adecuación en lugar de a sus valores absolutos: el mejor individuo es seleccionado con probabilidad p_1, el siguiente con p_2, etc. Dichas probabilidades pueden ser calculadas de muy diversas maneras siendo la más extendida la denominada ranking lineal:

$$p_i = \frac{1}{N}\left[h^- + \left(h^+ - h^-\right)\frac{i-1}{N-1}\right] \tag{3.137}$$

- Selección por torneo (*Tournament Selection*): consiste en tomar un subgrupo de t individuos de la población actual, copiando el mejor de éstos a la población seleccionada. Frecuentemente se toma t=2 (torneo binario).

Cruce

Una vez realizada la selección. se procede a la reproducción o cruce de los individuos seleccionados. La población nueva intercambia material cromosómico y sus descendientes forman la siguiente generación.

Los operadores de cruce más utilizados se basan en la linealidad de los cromosomas y, usualmente, en la ortogonalidad de los mismos. Dichos operadores son los siguientes:

- Cruce de un punto (*one-point crossover* o *single-point crossover*): se selecciona una posición interior de las cadenas y se intercambian los segmentos de ambas cadenas a la izquierda de las mismas.
- Cruce de n puntos (*n-point crossover*): este operador constituye una generalización del anterior en el que se seleccionan n puntos en el interior de las cadenas y se intercambian los segmentos entre puntos de corte alternos. Usualmente, n = 2, es decir, cruce de doble punto (*double-point crossover*).

- Cruce uniforme (*uniform crossover*): funcionamiento similar al del cruce de *n* puntos con $n = \lambda$. Para ser precisos, se realiza un test aleatorio para decidir de cuál de los antecesores se toma cada posición de la cadena.

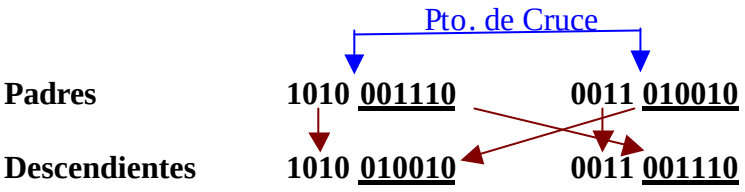


Figura 89. Ejemplo de operador de cruce basado en un punto.

Mutación

Los operadores de mutación son el arquetipo de operadores de alteración dado que actúan sobre individuos solos, realizando una pequeña modificación en alguno de sus genes o en el conjunto. Esta alteración puede consistir en cambiar un *bit* del valor binario, un dígito, o una serie de dígitos.

Estos cambios serán favorables si aumentan la salud del individuo que los lleva, es decir, si le hacen ser una solución mejor. Si los cambios son perjudiciales, estos genes no se propagarán de una generación a otra, porque los individuos que los portan se extinguirán, al no ser elegidos como padres y morirán.

Por lo tanto, mientras que los operadores de cruce se encargan de la búsqueda en profundidad explotando las mejores características de que disponga la población actual, los operadores de mutación realizan la búsqueda en anchura encargándose de explorar nuevos dominios en busca de mejores soluciones.

Esta división tiene un valor práctico para diferenciar dos modos complementarios de implantar la reproducción: el cruce se encarga de realizar un intercambio estructurado de información, la mutación proporciona una garantía de accesibilidad para todos los puntos del espacio de búsqueda.

Figura 90. Mutación binaria.

Reemplazamiento

A partir de los *n* miembros de la población de criadores y de los K miembros de la población de descendientes se debe obtener una nueva población de n miembros.

El reemplazamiento puede realizarse según varias técnicas:

- *Reemplazo del peor*: Los nuevos individuos creados substituyen a los peores individuos de la población actual.
- *Reemplazo aleatorio*: Se seleccionan al azar los individuos que serán reemplazados.
- *Reemplazo por torneo*: Se selecciona un subgrupo, de *t* individuos y se substituye al peor de éstos.
- *Reemplazo directo*: Cada pareja de descendientes reemplaza a sus progenitores.

En el reemplazamiento suele forzarse la permanencia de los *k* mejores individuos de la población (*k*-elitismo), para asegurar que las mejores soluciones encontradas no se pierdan.

Estos operadores frecuentemente producirán individuos que representan soluciones inválidas cuando se aplican a problemas con restricciones. En dicha situación existen diversas posibilidades:

- Eliminar los individuos inválidos. Esta opción tiene el inconveniente de desperdiciar el esfuerzo computacional empleado en generar dichas soluciones.
- Emplear un mecanismo de reparación para producir soluciones correctas.
- Usar una función de penalización que haga que las soluciones inválidas tengan una adecuación menor que las válidas.

La elección de los operadores de selección y reemplazo determina uno de los parámetros fundamentales del algoritmo: el modelo de evolución. En este sentido pueden considerarse tres opciones:

- Modelo generacional: en cada iteración del algoritmo se renueva completamente la población del algoritmo.
- Modelo de estado estacionario: en cada iteración del algoritmo únicamente se generan uno o dos individuos nuevos, los cuales son reinsertados en la población siguiendo el mecanismo de reemplazo elegido.
- Modelo gradual: este modelo generaliza los dos anteriores. En cada iteración se genera un porcentaje *g* de la población (*gap*), el cual es acto seguido reinsertado en la población.

En general, cuanto menor sea el porcentaje de solapamiento, esto es, cuantos menos individuos se generen en cada iteración. más rápida será la convergencia del algoritmo. Esto puede ser ventajoso, aunque debe tenerse en cuenta que puede redundar en una prematura degeneración de la población.

El funcionamiento de un algoritmo genético, por lo tanto, se basa en los siguientes pasos:

- Paso 1: Generar una población mediante el operador de inicialización.
- Paso 2: Si se cumple el criterio de parada, parar, si no ir al paso 3.
- Paso 3: Seleccionar dos padres de la población original.
- Cruzar los individuos seleccionados.
- Realizar mutaciones de los hijos en cuanto a la aportación de cada uno de ellos a la nueva población.
- Aceptar los nuevos cromosomas en la población.
- Reemplazar a los nuevos individuos en la población e ir al paso 2.

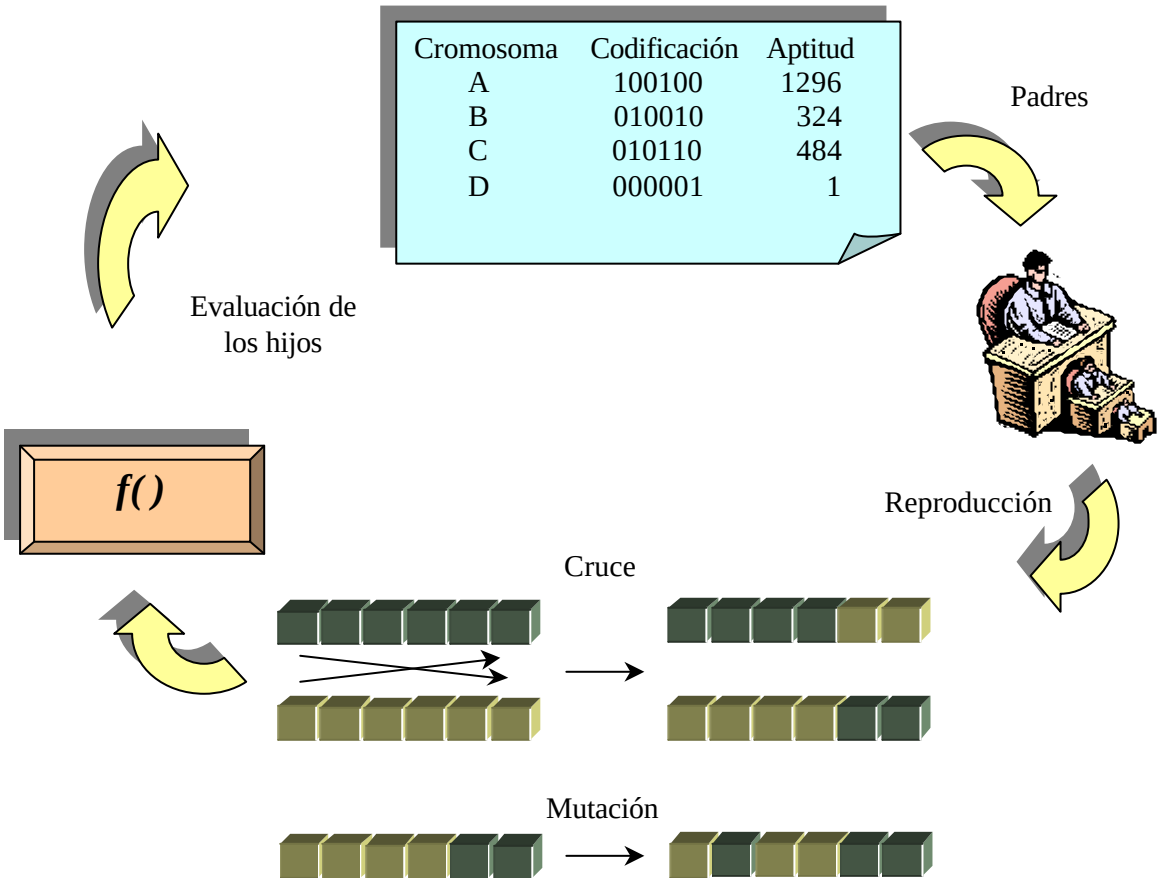


Figura 91. Ciclo de los algoritmos genéticos.

ESTRATEGIAS EVOLUTIVAS

Las Estrategias evolutivas surgieron inicialmente como métodos estocásticos de escalada con paso adaptativo específicamente diseñados para resolver problemas de optimización paramétrica [SCH89]. Con el paso del tiempo se han ido incorporando procedimientos propios de la computación evolutiva hasta convertirse en un paradigma más de dicha metodología. En el presente, las ES son algoritmos evolutivos enfocados preferentemente hacia la optimización paramétrica que utilizan una representación a través de vectores reales, selección determinista y operadores específicos de mutación y cruce, el cual ocupa ahora un lugar secundario con respecto a la mutación.

Estrategias evolutivas simples

Las Estrategias evolutivas simples (también llamadas de dos miembros) son el modelo más sencillo de su paradigma. Pese a su sencillez, tienen utilidad práctica y aun se siguen utilizando versiones mejoradas para resolver problemas reales de ingeniería.

Se trata de procedimientos estocásticos de optimización paramétrica con paso adaptativo. En la terminología del ramo se denotan como (1 + 1) y se describen así: evoluciona un solo individuo haciendo uso únicamente de un operador genético, la mutación.

Los individuos son bicrosomáticos y se representan a través de un par de vectores reales. El primer vector representa a un punto del espacio de búsqueda, y el segundo es un vector de desviaciones típicas que se usa al realizar la mutación. Al haber un solo progenitor no hay selección, y el criterio de reemplazo consiste en que el descendiente reemplaza a su progenitor si y sólo si, es más apto que éste; en caso contrario, el descendiente es eliminado y la población permanece sin cambios.

El segundo cromosoma no se somete a evolución, puesto que se adapta en función del éxito que haya tenido la mutación del primero.

Estrategias evolutivas múltiples

Las estrategias evolutivas múltiples surgen para corregir la tendencia de los métodos de búsqueda simple hacia subóptimos. Para corregir esa tendencia es conveniente realizar una búsqueda múltiple y una selección de los mejores descendientes. Ahora bien, la mutación por sí sola no tiene la capacidad de recombinar información de buenos individuos; por eso también es necesario introducir operadores de cruce.

Se deben definir dos parámetros: el tamaño de la población y el tamaño de la descendencia. Asimismo existen dos posibilidades de reemplazo: reemplazo determinista por inclusión (o de tipo más) o reemplazo determinista por inserción (o de tipo coma).

Las Estrategias Evolutivas múltiples incorporan los siguientes criterios:

- Criterio de representación: Es el ya comentado: mediante parejas de vectores reales. No es infrecuente usar una representación mediante tríos, en la que el último vector controla la correlación entre las mutaciones de cada componente.
- Criterio de tratamiento de los individuos no factibles: Como el anterior, es decir, se ignoran los individuos no factibles (filtrado).
- Operadores genéticos: Son por defecto el cruce y la mutación, aunque se pueden introducir otros. En lo que respecta al cruce, se usa habitualmente el cruce uniforme (para ambos cromosomas) o el cruce intermedio, que consiste en generar un único descendiente promediando los dos progenitores En ocasiones se realiza el cruce en modo global, esto es, tomando un nuevo par de progenitores para cada componente de la descendencia. La mutación es la misma de antes, salvo que, como ahora no se verifica la hipótesis del 1/5 de éxitos, el segundo cromosoma se somete también a evolución (mutación).
- Criterio de selección: Cualquier miembro de la población puede ser elegido como progenitor con igual probabilidad (muestreo aleatorio simple).
- Criterios de reemplazo: El reemplazo siempre es determinista: se eligen los mejores miembros. Existen dos posibilidades de hacer esa elección, por inserción o por inclusión.

3.4.4.10 MÉTODOS BASADOS EN TECNOLOGÍAS DIFUSAS

La Teoría de los conjuntos difusos se inició en 1965, en la Universidad de California en Berkeley por *Lotfi, A. Zadeh* [ZAD65], con el propósito de intentar modelar la incertidumbre de algunos tipos de lenguaje natural y la complejidad de las relaciones entre las distintas percepciones de la realidad. Uno de los productos más importantes de las investigaciones desarrolladas en este campo, es lo que se conoce como lógica difusa. La lógica difusa designa un conjunto de herramientas de la lógica convencional (booleana), que ha sido extendido para incluir el concepto de verdad parcial (valores de verdad entre completamente cierto y completamente falso).

El concepto de pertenencia manejado en la teoría de conjuntos clásica, difícilmente puede ser utilizado para muchas de las situaciones en las que no existe una distinción binaria de las categorías. Por ejemplo, si se define el conjunto de gente joven estableciendo como edad límite 20 años, se puede plantea el dilema de por qué una persona es joven en su cumpleaños número 20 y no es joven al día siguiente. Este problema se presentaría ante cualquier límite de edad establecido. Una construcción más natural del conjunto de jóvenes se obtendría relajando la separación entre joven y no joven.

Para enfrentar este problema, la lógica difusa propone la utilización de un concepto de pertenencia a conjuntos que permita el manejo de distintas graduaciones; apareciendo de esta forma los grados de pertenencia de un elemento a un conjunto. De esta forma la distinción bivalente de inclusión / no inclusión manejada en la Teoría de Conjuntos clásica se convierte en la lógica difusa en una distinción polivalente en la que cabe la posibilidad de muchos grados de pertenencia.

La convención utilizada para denotar los grados de pertenencia es de asignar el valor 1 al grado de pertenencia mas fuerte, el valor 0 al grado de pertenencia mas débil y valores reales en el interior del intervalo [0,1] a grados de pertenencia intermedios. El rol que juega el intervalo unitario en la asignación de grados de pertenencia es solo homegeneizador y puede ser reemplazado por cualquier otro conjunto ordenado. Sin embargo, esta convención permite observar a la lógica difusa como una extensión natural de la binaria.

CONJUNTOS DIFUSOS

Sea X un conjunto convencional de elementos, al que se denominará conjunto universo. Se define el conjunto difuso $\tilde{A}$ sobre X como:

$$\tilde{A} := \left\{ \left(x, m_{\tilde{A}}(x); x \in X \right) \right\} \tag{3.138}$$

donde $m_{\tilde{A}} : X \to [0,1]$ es la función de pertenencia que a cada elemento x dentro del universo del discurso X le asigna un número $m_{\tilde{A}}(x)$ entre cero y uno, que representa su grado de pertenencia al conjunto difuso $\tilde{A}$.

Nótese que si $m_{\tilde{A}} : X \to \{0,1\}$ esto es, si se restringen los posibles grados de pertenencia únicamente a los valores 0 (no pertenencia) y 1 (pertenencia). $m_{\tilde{A}}(x)$ es la función característica del

conjunto (no difuso) *A* [HAL65]. Así pues, todo conjunto en el sentido usual es también un conjunto difuso.

La función de pertenencia se establece de una manera arbitraria, lo cual es uno de los aspectos más flexibles de los Conjuntos Difusos. Por ejemplo, se puede convenir que el grado de pertenencia de una temperatura de "45ºC" al conjunto $\tilde{A}$ es 1, el de "25ºC" es 0.4 . el de "6ºC" es 0, etc.; cuanto mayor es el valor de una temperatura, mayor es su grado de pertenencia al conjunto.

Para operar en la práctica con los Conjuntos Difusos se suelen emplear funciones de pertenencia del tipo representado en la figura:

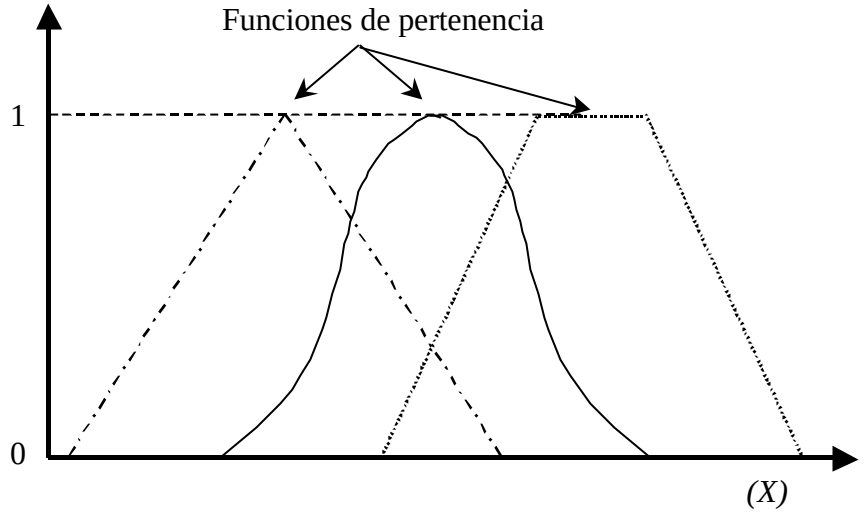


Figura 92. Funciones de pertenencia.

En la figura se pueden observar tres tipos de funciones de pertenencia muy usuales: el tipo triangular, que puede ser un caso concreto del trapezoidal (derecha) en el que los dos valores centrales son iguales, y el de forma de campana *gaussiana*.

La noción tradicional que se maneja de conjunto vacío es la de "aquel que no contiene elementos". En conjuntos difusos se dice que un conjunto difuso, $\tilde{A}$, es vacío si todos los elementos *x* del universo del discurso *X* tienen grados de pertenencia $m_{\tilde{A}}(x)$ nulos, esto es:

$$\tilde{A} = f \Leftrightarrow m_{\tilde{A}}(x) = 0 \qquad \forall x \in X \tag{3.139}$$

Dados dos conjuntos difusos $\tilde{A}$ y $\tilde{B}$ en un universo X se dice que $\tilde{A}$ es un subconjunto de $\tilde{B}$ si para todo elemento x del universo se cumple la desigualdad $m_{\tilde{A}}(x) \le m_{\tilde{B}}(x)$. Los conjuntos serán iguales si cada uno es un subconjunto del otro, en otras palabras, si para todo x, $m_{\tilde{A}}(x) = m_{\tilde{B}}(x)$.

El tamaño. o cardinalidad, de un conjunto difuso $\tilde{A}$ en un universo dado es la suma, sobre los elementos del universo, de los grados de pertenencia a $\tilde{A}$:

$$T(\tilde{A}) = \sum_{x \in X} m_{\tilde{A}}(x) \tag{3.140}$$

Cada elemento x del universo X tiene asociado un peso relativo. respecto a $\tilde{A}$, que viene dado por:

$$p_{\tilde{A}}(x) = \frac{m_{\tilde{A}}(x)}{T(\tilde{A})} \tag{3.141}$$

El valor esperado de $\tilde{A}$ viene dado por:

$$E(\tilde{A}) = \sum_{x \in X} x p_{\tilde{A}}(x) \tag{3.142}$$

Pudiendo definirse de modo general los momentos de orden k del conjunto difuso $\tilde{A}$ sobre X como:

$$E_k(\tilde{A}) = \sum_{x \in X} x^k p_{\tilde{A}}(x) \tag{3.143}$$

Operaciones Básicas sobre Conjuntos Difusos

Con el fin de hacer significativa la lógica difusa los operadores booleanos básicos (complemento, intersección y unión) han sido redefinidos para que puedan tomar valores entre cero y uno y generar resultados entre cero y uno:

1. Complemento: $m_{\neg\tilde{A}}(x) = 1 - m_{\tilde{A}}(x) \quad \forall x \in X$ (se corresponde con el conectivo NO).
2. Intersección: $m_{\tilde{A}\cap\tilde{B}} = Min\{m_{\tilde{A}}(x), m_{\tilde{B}}(x)\} \quad \forall x \in X$ (se corresponde con el conectivo Y).
3. Unión: $m_{\tilde{A}\cup\tilde{B}} = Max\{m_{\tilde{A}}(x), m_{\tilde{B}}(x)\} \quad \forall x \in X$ (se corresponde con el conectivo O)

El principio de extensión

El principio de extensión establece que si $f : X \to Y$ es una función (operación) y $\tilde{A}$ es un conjunto difuso en X entonces $\tilde{A}$ induce vía f un conjunto difuso $\tilde{B} = f(\tilde{A})$ en Y dado por:

$$m_{\tilde{B}}(y) = \begin{cases} Max_{x \in f^{-1}(y)} m_{\tilde{A}}(x), & si\ f^{-1}(y) \neq f \\ 0, & en\ otro\ caso \end{cases} \tag{3.144}$$

donde $f^{-1}(y) = \{x \in X \mid f(x) = y\}$.

El principio de extensión puede ser generalizado considerando $f : X_1 \times ... \times X_n \to Y$ y $\tilde{A}_1, ..., \tilde{A}_n$ conjuntos difusos de $X_1 ... X_n$, respectivamente, entonces $\tilde{A}_1, ..., \tilde{A}_n$ induce vía f un conjunto difuso $\tilde{B} = f(\tilde{A}_1, ..., \tilde{A}_n)$ en Y dado por:

$$m_{\tilde{B}}(y) = \begin{cases} Max\{\min\{m_{\tilde{A}_1}(x), ..., m_{\tilde{A}_n}(x)\} \mid x \in f^{-1}(y)\} & si\ f^{-1}(y) \neq f \\ 0, & en\ otro\ caso \end{cases} \tag{3.145}$$

Por lo tanto el principio de extensión generalizado permite extender las relaciones no difusas a sus contrapartes difusas. Esto puede ser usado, por ejemplo, para implementar la aritmética difusa considerando la función $f : X \times X \to X$ se tienes los siguientes operadores:

- Adición: $m_{f(\tilde{A},\tilde{B})=\tilde{A}+\tilde{B}}(x) = \underset{z=x+y}{Max} \min\left\{m_{\tilde{A}}(x), m_{\tilde{B}}(y)\right\}$
- Sustracción: $m_{f(\tilde{A},\tilde{B})=\tilde{A}-\tilde{B}}(x) = \underset{z=x-y}{Max} \min\left\{m_{\tilde{A}}(x), m_{\tilde{B}}(y)\right\}$
- Multiplicación: $m_{f(\tilde{A},\tilde{B})=\tilde{A}\cdot\tilde{B}}(x) = \underset{z=x\cdot y}{Max} \min\left\{m_{\tilde{A}}(x), m_{\tilde{B}}(y)\right\}$
- División: $m_{f(\tilde{A},\tilde{B})=\tilde{A}/\tilde{B}}(x) = \underset{z=x/y}{Max} \min\left\{m_{\tilde{A}}(x), m_{\tilde{B}}(y)\right\}$

Nivel o corte de un conjunto difuso

Dado un número $a \hat{I} (0, 1)$ y un conjunto difuso $\tilde{A}$, se definen:

- El *corte-a* de $\tilde{A}, \tilde{A}_a$, como el conjunto, en el sentido usual, consistente de aquellos objetos cuyos grados de pertenencia a $\tilde{A}$ superen, estrictamente, el valor a:

$$\tilde{A}_a = \left\{x \in X : m_{\tilde{A}}(x) > a\right\} \tag{3.146}$$

- El *corte-a* cerrado de $\tilde{A}, \tilde{A}^a$, como el conjunto, en el sentido usual, consistente de aquellos objetos cuyos grados de pertenencia a $\tilde{A}$ no es inferior al valor a:

$$\tilde{A}_a = \left\{x \in X : m_{\tilde{A}}(x) \geq a\right\} \tag{3.147}$$

El valor *a* recibe el nombre umbral de corte.

Ejemplo: Si el conjunto difuso A está dado por:

X	$m_{\tilde{A}}(x)$
1	1
2	1
3	0.7
4	0.5
5	0.1

entonces $\tilde{A}_{0.1} = \{1,2,3,4\}, \tilde{A}_{0.5} = \{1,2,3\}, \tilde{A}_{0.8} = \{1,2\}, \tilde{A}_1 = \{1\}$.

El concepto de un *a-corte* para un conjunto difuso es crucial para el llamado teorema de descomposición el cual establece que un conjunto difuso $\tilde{A}$ en *X* puede representarse como:

$$\tilde{A} = \bigcup_{a \in [0,1]} a \cdot \tilde{A}_a \tag{3.148}$$

Hedges (Realces)

Un aspecto importante en el desarrollo de los sistemas difusos es su habilidad para definir "*hedges*" o transformadores de valores difusos. Estas operaciones constituyen un esfuerzo para mantener lazos cercanos al lenguaje natural y llegar a la generación de afirmaciones difusas a través de cálculos matemáticos.

El papel de adverbio en un conjunto difuso lo realiza la función unaria *r:[0,1]* ® *[0,1*], denominada *hedge* o realce. El realce de] conjunto difuso $\tilde{A}$ sobre *r*, es un subconjunto difuso en el intervalo unitario [0,1], obtenido mediante la composición de la función de ajuste *r* sobre $\tilde{A}$, es decir, $r \circ \tilde{A}$. Se dice que *r* es un realce diminutivo (respectivamente aumentativo) si para cada *t*, *r(t)³t* (respectivamente *r(t)£ t*).

Ejemplo:

Para cada p>0, la función $\begin{array}{l} r_p : [0,1] \to [0,1] \\ t \mapsto t^p \end{array}$ es un realce. Para p≤ 1 el realce r_p es diminutivo y para p≥ 1, r_p es aumentativo.

Los adverbios muy y poco usualmente vienen dados por *p=2* y *p=1/2* respectivamente. Así si μ_{viejo}(Pedro) = 0.8 se tiene que $\mu_{muy\ viejo}$(Pedro)= 0.64 muy viejo.

RELACIONES DIFUSAS

Existen conjuntos difusos que difícilmente pueden ser caracterizados por la observación de una sola variable. En este contexto, la definición de conjuntos difusos a partir de funciones de pertenencia requiere que éstas posean dominios en los que se permita la participación de varias variables. Ahora. la coexistencia de varias variables en el dominio de una función de pertenencia denota una interacción dentro de un sistema, estas interacciones siguen ciertos patrones de comportamiento, denominados relaciones causales entre las variables difusas.

Para dos conjuntos usuales *A, B* su producto cartesiano consta de todas las parejas ordenadas de la forma *(a,b)* donde *a Î A* y *b Î B.* Así pues, si • es un operador conjuntor y $\tilde{A}$ y $\tilde{B}$ son conjuntos difusos en sendos universos *X* e *Y*, su producto cartesiano es el conjunto difuso:

$$\begin{array}{l} m_{\tilde{A}\bullet\tilde{B}} : X \times Y \to [0,1 \\ (x,y) \mapsto m_{\tilde{A}}(x) * m_{\tilde{B}}(y) \end{array} \qquad (3.149)$$

Una relación, en el sentido usual, entre dos conjuntos es un subconjunto de su producto cartesiano. Por tanto, se puede considerar a una relación difusa entre dos universos como un conjunto difuso en su producto cartesiano.

Si R es una relación difusa en $X \times Y$ las respectivas proyecciones de R en *X* y en *Y* son los conjuntos difusos:

$$\begin{array}{l} \prod_X [R] : x \mapsto Max\{R(x,y) | y \in Y\} \\ \prod_Y [R] : y \mapsto Max\{R(x,y) | x \in X\} \end{array} \qquad (3.150)$$

Si R es una relación difusa $X \times Y$ en y S es una relación difusa en $Y \times Z$, la composición de R con S en $X \times Z$ es el conjunto difuso:

$$S \circ R : (x,z) \mapsto Max\left\{R(x,y) \circ S(y,z) \middle| y \in Y\right\} \tag{3.151}$$

Si R es una relación difusa en $X \times Y$, entonces todo conjunto difuso $\tilde{B}$ en Y determina un conjunto difuso $\tilde{A} = \tilde{B} \circ R$ en X, denominado composición de R con $\tilde{B}$, que viene dado por:

$$\tilde{A} = \tilde{B} \circ R : x \mapsto Max\left\{R(x,y) \circ \tilde{B}(y) \middle| y \in Y\right\} \tag{3.152}$$

Una etiqueta lingüística es un nombre a un conjunto difuso. Es decir, es una tema (Nombre, $\tilde{A}$, X), donde *Nombre* es el nombre asociado al conjunto difuso $\tilde{A}$ en el universo X. Es convencional confundir a la etiqueta lingüística con su propio nombre.

Lógicas proposicionales difusas

Las lógicas proposicionales difusas son de una forma similar a la siguiente:

"Si x es bajo e y es alto entonces z es mediano",

donde:

- x e y son variables insumo (nombres para valores conocidos de los datos).
- z es una variable resultado (un nombre para el valor de los datos a ser computado).
- bajo es una función de pertenencia (subconjunto difuso) definida sobre x.
- alto es una función similar definida sobre y.

En general una regla difusa tendrá la estructura:

"Si A entonces B"

Donde A se llamará antecedente y B el consecuente de la regla, siendo A y B etiquetas de conjuntos difusos representando valores de x e y, A y B pueden ser cualquier proposición, tan complicada como se quiera.

El objetivo de las lógicas proposicionales difusas es encontrar una relación difusa R, cuya función de pertenencia exprese el grado de verdad de *"Si A entonces B"*. A esta acción se la denomina implicar, y aunque existen muchas formas de representar la relación entre antecedente y consecuente las más útiles son las de *Mandami* (también denominada *Min*) y *Larsen* (también denominada producto):

$$\text{Mandami:} \quad m_{A \Rightarrow B}(x,y) = \min\left(m_A(x), m_B(y)\right) \tag{3.153}$$

$$\text{Larsen:} \quad m_{A \Rightarrow B}(x,y) = \left(m_A(x) \cdot m_B(y)\right) \tag{3.154}$$

Implicar es el paso previo a inferir, que consiste en extraer una conclusión a partir de la relación generada por la implicación y de un conjunto difuso de entrada. Dicha conclusión es un nuevo conjunto difuso, y se obtendrá mediante la composición de R con $\tilde{A}$. En la composición.

todos los subconjuntos difusos asignados a cada variable resultado son combinados para formar un único subconjunto difuso con cada una de estas variables.

Un ejemplo, es la regla del *Modus Ponens Composicional*:

x *es* $\tilde{A}$.

Si x es A entonces y es B

y *es* $\tilde{B}$

se tiene que $\tilde{B} = \tilde{A} \circ (A \to B)$, es decir:

$$m_{\tilde{B}}(y) = \sup_{x \in \tilde{A}} \left[\min \left\{ m_{\tilde{A}}(x), m_{A \to B}(x, y) \right\} \right] \tag{3.155}$$

El conjunto de reglas en un sistema es conocido como la base de reglas o conocimientos.

Difuminado y perfilado

Cuando se utiliza la lógica difusa para el razonamiento, gran cantidad de veces los valores de entrada no son conjuntos difusos. sino valores numéricos concretos, por lo que se debe obtener un conjunto difuso correspondiente a esa entrada (difuminado o *fuzzification*), por otro lado, se debe obtener un valor concreto de salida a partir del conjunto borroso originado durante el proceso de inferencia (perfilado o *defuzzification*).

El primer paso, difuminado, es sencillo puesto que si se tiene un valor de entrada x_0 se puede definir el correspondiente conjunto unitario (*singleton*) como

$$m(x) = \begin{cases} 1 & si\, x = x_0 \\ 0 & en\, otro\, caso \end{cases} \tag{3.156}$$

Esto permite simplificar el supremo del proceso de composición, y de ahí su popularidad y amplio uso en aplicaciones. Pero este mecanismo puede ser desastroso cuando la información puede ser alterada por ruido, en cuyo caso sería mejor utilizar un conjunto no unitario en el cual se tomaría $m(x_0) = 1$ y $m(x)$ decrecería desde la unidad hasta 0, a partir de x_0 a izquierda y derecha.

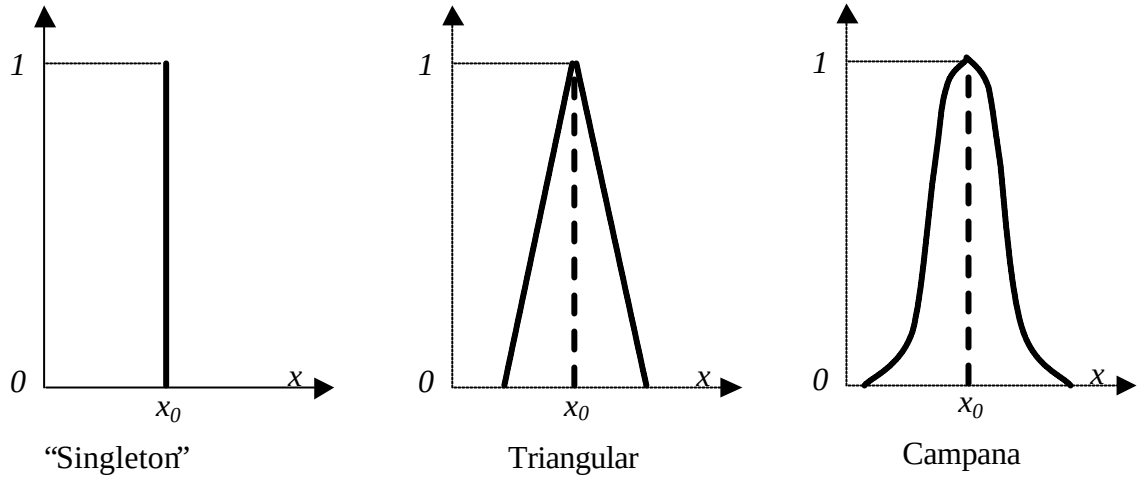


Figura 93. Ejemplos de funciones de este tipo pueden ser la de campana (Gauss) y la triangular.

El perfilado consiste en la transformación de la curva (función de pertenencia) del conjunto difuso obteniendo como salida un único valor concreto (*crisp*). Para perfilar existen diferentes métodos encontrándose su mayor o menor adecuación en función de su simplicidad computacional. Algunos de estos son:

Primer máximo: Se toma como representante de un conjunto difuso al primer elemento $x_{\tilde{A}}$, en el universo X de acuerdo con un orden dado, tal que $m(x_{\tilde{A}}) = \underset{x \in X}{Max}\, m_{\tilde{A}}(x)$. Este criterio conlleva la dificultad de calcular un valor máximo de una función real.

Corte-a: Se elige un elemento $x_0 \in A_a$ en el corte-a de $\tilde{A}$.

Centroide: Se toma el elemento $x_0 \in X$ tal que:

$$\left| m_{\tilde{A}}(x_0) - E(\tilde{A}) \right| = \underset{x \in X}{Min} \left| m_{\tilde{A}}(x) - E(\tilde{A}) \right| \tag{3.157}$$

es decir, x_0 es uno de los elementos en el universo X cuyo grado de pertenencia a $\tilde{A}$ es el más cercano al valor esperado de los valores de $\tilde{A}$.

De manera más general, para $k \geq 1$, se puede elegir al elemento $x_0 \in X$ tal que:

$$\left| m_{\tilde{A}}(x_0)^k - E_k(\tilde{A}) \right| = \underset{x \in X}{Min} \left| m_{\tilde{A}}(x)^k - E_k(\tilde{A}) \right| \tag{3.158}$$

es decir, x_0 es uno de los elementos cuyo grado de pertenencia a $\tilde{A}$ tiene una *k-ésima* potencia más cercana al *k-ésimo* momento de $\tilde{A}$.

Centro de gravedad: El centro de gravedad de altura a de $\tilde{A}$ es:

$$C(\tilde{A}, a) = \begin{cases} \dfrac{1}{T(\tilde{A}^a)} \displaystyle\sum_{x \in \tilde{A}^a} m_{\tilde{A}}(x)\, x & \text{si X es finito o numerable} \\ \dfrac{1}{T(\tilde{A}^a)} \displaystyle\int_{x \in \tilde{A}} m_{\tilde{A}}(x)\, x \cdot dx & \text{si X es un espacio de integración} \end{cases} \tag{3.159}$$

El centro de gravedad $C(\tilde{A}, a)$ es pues el promedio de los elementos en el corte-a de $\tilde{A}$.

Dos centros de gravedad importantes para perfilar son el centro de gravedad básico (centro de gravedad de altura 0) y el centro de gravedad máximo (centro de gravedad de altura $\underset{x \in X}{Max}\, m_{\tilde{A}}(x)$).

Un conjunto difuso $\tilde{A}$ en X se dice ser convexo si para cualesquiera n puntos (x_1, x_2, ..., x_n) y cualesquiera n coeficientes $a_1, a_2, ..., a_n \in [0,1]$ tales que $\sum_{i=1}^{n} a_i = 1$ que tiene:

$$m_{\tilde{A}}\left(\sum_{i=1}^{n} a_i x_i\right) \geq \sum_{i=1}^{n} a_i m_{\tilde{A}}(x_i) \qquad (3.160)$$

En el caso de que $\tilde{A}$ sea un conjunto convexo, cualquiera de los centros básico o máximo puede ser un buen representante del conjunto difuso $\tilde{A}$. Sin embargo, si $\tilde{A}$ no es convexo, la selección por centros puede ser muy desafortunada. Por ejemplo, si $\tilde{A}$ fuese un conjunto usual, entonces se podría elegir a un elemento fuera de $\tilde{A}$ con este criterio.

SISTEMA DE INFERENCIA DIFUSA

Un sistema de inferencia difusa es aquel que usa un conjunto de funciones de pertenencia y reglas difusas para dar razón de un grupo de datos. A los sistemas de inferencia difusa. se les conoce también con los nombres de sistemas basado en reglas difusas, modelos difusos, controladores lógicos difusos o simplemente sistemas difusos.

La lógica difusa es ampliamente utilizada para modelar sistemas de control y de procesamiento de señales. Los sistemas basados en lógica difusa permiten relacionar entradas (*crisp* inputs) y salidas (*crisp outputs*) con funciones lineales, y están formados por cuatro componentes:

- Reglas: proporcionadas por expertos.
- Difuminador.
- Perfilador.
- Motor de inferencia.

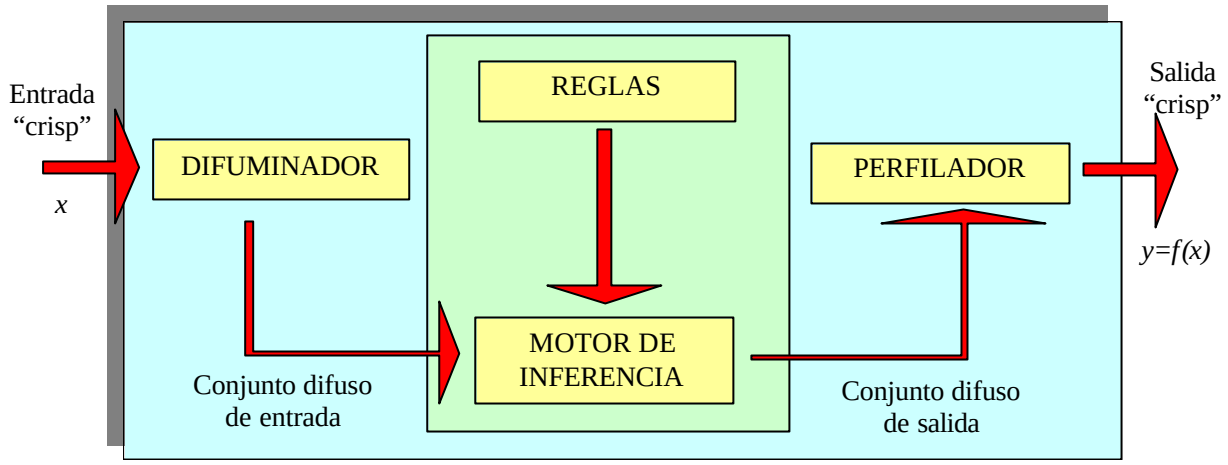


Figura 94. Esquema de un sistema de inferencia

Los sistema de inferencia difusa más utilizados en las aplicaciones prácticas son el Modelo de *Mandami* y el Modelo de *Sugeno*.

Modelo de Mandami

El modelo de Mandami [MAN75] fue propuesto como una primera aproximación a sistemas de inferencia difusa. Cada regla difusa R_i dice que:

$$R_i\text{: si } x \text{ es } \tilde{A}_i \text{ entonces } y \text{ es } \tilde{B}_i .$$

expresa una relación difusa $\tilde{C}_i$ que representa la intersección difusa de los conjuntos difusos $\tilde{A}_i$ y $\tilde{B}_i$. ($\tilde{C}_i = \tilde{A}_i \cap \tilde{B}_i$). Entonces la función de pertenencia de $\tilde{C}_i$, viene dada por:

$$m_{\tilde{C}_i}(x,y) = \min\left(m_{\tilde{A}_i}(x), m_{\tilde{B}_i}(y)\right) \tag{3.161}$$

Para la contribución de m reglas difusas el modelo de *Mandami* considera la relación difusa $\tilde{C}$ construida como la unión de cada una de las relaciones difusas $\tilde{C}_i$ (i=1,...,m). Por lo tanto la función de pertenencia viene dada por:

$$m_{\tilde{C}}(x,y) = \max_{i=1,\ldots,m}\left(m_{C_i}(x,y)\right) = \max_{i=1,\ldots,m}\left(\min\left(m_{\tilde{A}_i}(x), m_{\tilde{B}_i}(y)\right)\right) \tag{3.162}$$

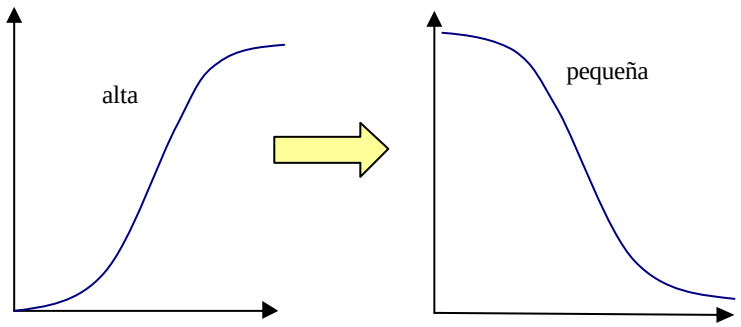


Figura 95. Ejemplo de regla tipo Mandami: si la presión es alta entonces el volumen es pequeño.

Modelo Sugeno

El modelo de inferencia difuso Sugeno también conocido como TSK (*Takagi-Sugeno-Kang model*) fue propuesto por Takagi, Sugeno y Kang [SUG88] [TAK85]. Las reglas en el modelo de *Sugeno* son del tipo:

$$Si\ x_1\ es\ A_1\ y\ ...\ y\ x_N\ es\ A_N\ entonces\ y=f(x_1, ..., x_N)$$

donde A_k, k=1,...,N son las etiquetas de las premisas o antecedentes.

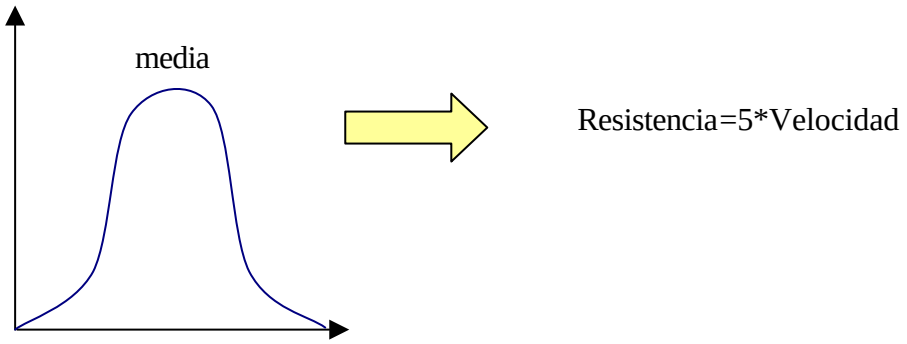


Figura 96. Ejemplo modelo Sugeno: si la velocidad es media entonces la resistencia es cinco veces la velocidad.

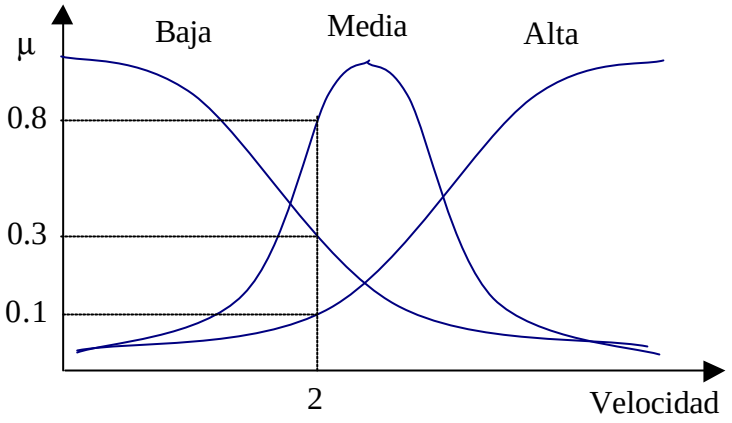
La diferencia entre el modelo de *Sugeno* y el modelo de *Mandami*, radica en el tipo de consecuencia obtenida: un funcional para el caso de *Sugeno* y una consecuencia difusa para el caso de *Mandami*. Habitualmente el funcional f es una función lineal de las variables x_k, es decir:

$$f(x_1, x_2, ...x_N) = p_1 \cdot x_1 + ... + p_N \cdot x_N + p_0 \tag{3.163}$$

Si se considera un conjunto de m reglas difusas, el modelo de inferencia calcula una salida y (*crisp*) mediante la media ponderada de las salidas individuales y_i (i= 1, ..., m):

$$y = \sum_{i=1}^{m} \frac{w_i \cdot y_i}{\sum_{j=1}^{m} w_j} = \sum_{i=1}^{m} \frac{w_i}{\sum_{j=1}^{m} w_j} p_{i1} \cdot x_1 + ... + p_{iN} \cdot x_N + p_{i0} \tag{3.164}$$

donde w_i denota el peso de la regla i.



$$\begin{cases} R1: w1 = 0.3; r1 = 2 \\ R2: w2 = 0.8; r2 = 4*2 \\ R3: w3 = 0.1; r3 = 8*2 \end{cases} \Rightarrow \text{Resistencia} = \sum (w_i \cdot r_i) / \sum w_i = 7.12$$

$$\begin{cases} \text{Si la velocidad es baja entonces la resistencia es 2} \\ \text{Si la velocidad es media entonces la resistencia es 4 veces la velocidad} \\ \text{Si la velocidad es alta entonces la resistencia es 8 veces la velocidad} \end{cases}$$

Figura 97.Ejemplo de obtención de la resistencia a partir de una velocidad igual a 2 y tres reglas.

Desde un punto de vista geométrico, la base de reglas de un modelo *Sugeno* proporciona una aproximación de la proyección mediante una función lineal a trozos. En el caso general estas funciones podrían ser no lineales.

El modelo de inferencia difuso *Sugeno* permite la descomposición del sistema en subsistemas más simples, permitiendo además realizar una partición del espacio de entrada, lo que permite la descripción de sistemas complejos [WAT95].

APLICACIONES DE LA LÓGICA DIFUSA AL PROCESO DEL DATA MINING

Las técnicas de lógica difusa pueden ser aplicadas en diferentes fases del proceso de *data mining*. Por ejemplo durante la etapa de preproceso y limpieza de datos las técnicas de lógica difusa pueden ser aplicadas para la detección de outliers mediante el uso de algoritmos de *clustering* difusos, o durante la etapa de evaluación y verificación de resultados pueden ser utilizados para contrastar y verificar las conclusiones obtenidas por los expertos.

Sin embargo, la principal aplicación de la lógica difusa en el proceso de *data mining* se centra en los algoritmos. Esta contribución de las técnicas de lógica difusa va dirigida en dos direcciones:

- Difuminado de datos: Las técnicas de lógica difusa permiten el tratamiento de datos difusos provenientes de medidas imprecisas o de las descripciones de expertos del

dominio. Estos datos se convierten a conjuntos difusos y son tratados mediante algoritmos de *data mining*, procediendo a realizar el perfilado para la interpretación de los resultados.

- Difuminado de algoritmos: Métodos que utilizan técnicas difusas para estructurar y tratar conjuntos de datos usuales. Dentro de este grupo se encuentran por ejemplo, los algoritmos de *clustering* difusos como el KNN-fuzzy, las redes neuro-fuzzy o las reglas asociadas

3.4.4.11 MÉTODOS BASADOS EN TÉCNICAS NEURODIFUSAS

Los sistemas híbridos de redes neuronales con lógica difusa dan lugar a un tipo de métodos denominados *neuro-fuzzy*, que permiten un tratamiento más eficaz, que la aplicación directa de redes neuronales y lógica difusa, para una gran variedad de problemas.

Los problemas abordados por el *data mining* tienen generalmente una componente de datos empíricos y otra de conocimiento previo. La estructura *neuro-fuzzy* permite el uso de métodos cualitativos y cuantitativos en la construcción de modelos tanto en la etapa de aprendizaje como en la de funcionamiento y realimentación de [BOS98].

Además los sistemas *neuro-fuzzy* heredan las propiedades de los sistemas difusos y de las redes neuronales, tales como:

- Interpretación lingüística.
- Introducción de conocimiento previo.
- Autoaprendizaje.
- Generalización.
- Interpolación.
- Etc.

Los sistemas neuro-fuzzy pueden clasificarse en función de las diferentes interacciones entre las redes neuronales y sistemas difusos en la siguiente taxonomía [NAU95]:

- Modelos Concurrentes: Cuando la red neuronal y el sistema difuso trabajan juntos pero sin interactuar el uno en el otro, es decir, ninguno determina las características del otro.
- Modelos Cooperativos: Cuando la red neuronal se usa para determinar los parámetros de funcionamiento del sistema difuso. En estos modelos, se distinguen dos fases: la de entrenamiento y la de funcionamiento. En la primera, la red neuronal interactúa con el sistema difuso determinando los parámetros del mismo, mientras en la segunda, la red neuronal desaparece dejando sólo el sistema difuso.

- Modelos Híbridos: En esta aproximación, los sistemas difusos y de red neuronal, trabajan juntos en una arquitectura homogénea que puede ser interpretada como una red neuronal con parámetros difusos o como un sistema difuso con parámetros o funcionamiento distribuidos. El difuminado de la red neuronal convencional se realiza mediante la extensión de los pesos y/o las entradas y/o las salidas objetivo en números difusos. Estas extensiones pueden ser de los siguientes tipos:

Red neuronal difusa	Pesos	Entradas	Salidas
Tipo 1	Crisp	Difuso	Crisp
Tipo 2	Crisp	Difuso	Difuso
Tipo 3	Difuso	Difuso	Difuso
Tipo 4	Difuso	Crisp	Difuso
Tipo 5	Crisp,	Crisp	Difuso
Tipo 6	Difuso	Crisp	Crisp
Tipo 7	Difuso	Difuso	Crisp

Tabla 15. Tipos de modelos de redes neuronales difusas.

Las redes neuro-fuzzy tipo 1 se utilizan en problemas de clasificación de un vector de entrada difuso en una clase convencional (crisp). Las redes de tipo 2, 3 y 4 se utilizan para implementar reglas del tipo SI...ENTONCES...

El proceso computacional neuro-fuzzy se desarrolla en tres etapas:

1. Desarrollo de un modelo neuronal difuso compuesto por arquitecturas que pueden incorporar neuronas, neuronas difusas y/o inferencia difusa. Dos posibles ejemplos de modelos neuronales difusos son por ejemplo:

Los términos lingüísticos son interpretados por un interface difuso que proporciona la entrada para una red neuronal de varias capas. La red neuronal es entrenada para lograr las salidas objetivo:

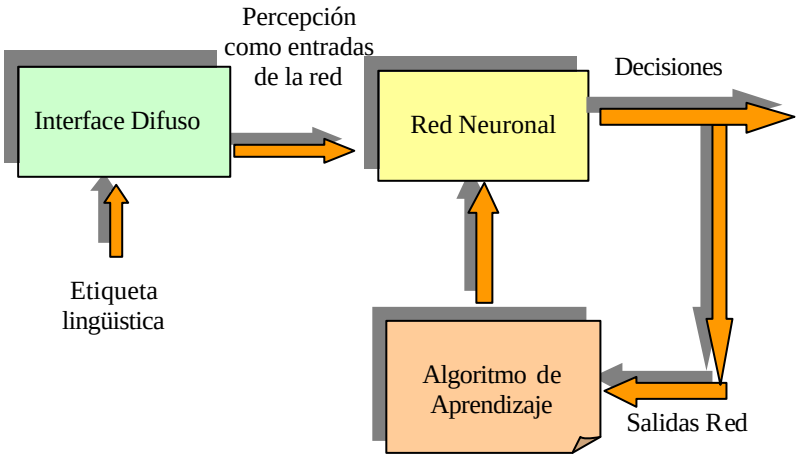


Figura 98. Un ejemplo de modelo neuro-fuzzy.

Una red neuronal actúa como motor de inferencia de un sistema difuso:

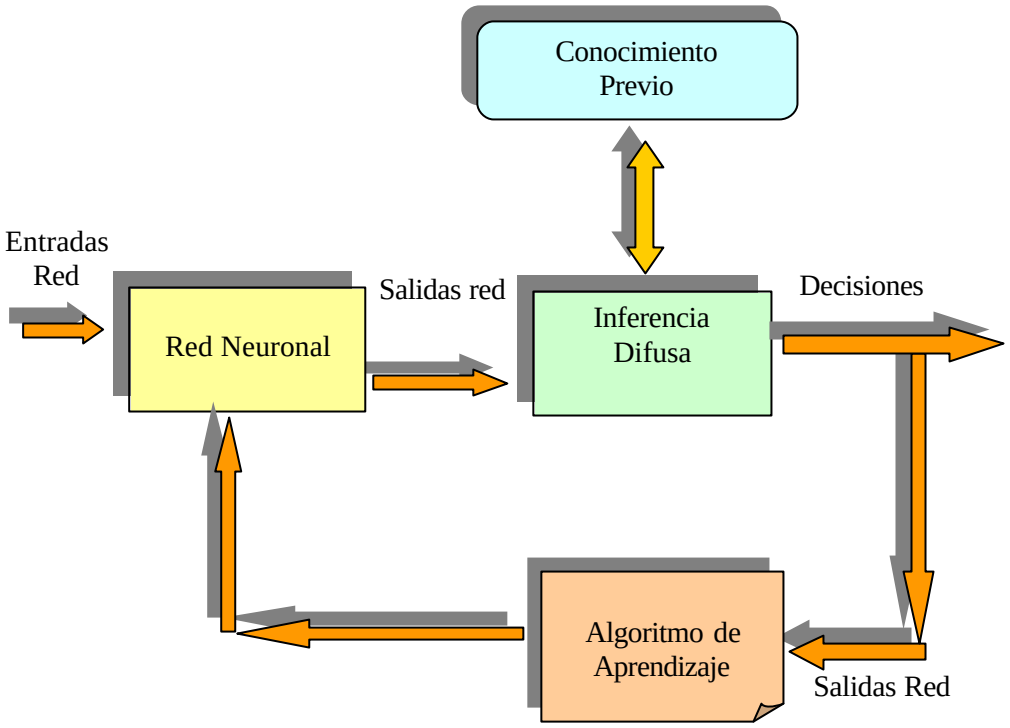


Figura 99. Otro ejemplo de modelo neuro-fuzzy.

2. Creación de modelos sinápticos que permitan las conexiones difusas en la red neuronal.

3. Desarrollo de algoritmos de aprendizaje (métodos de ajuste de los pesos).

Obsérvese que cualquier red neurona] puede ser aproximada con cualquier grado de convergencia por un sistema experto difuso. En efecto, supóngase que v_j (j=1, ..., n) son las señales de entrada e y_i (i=1,...,n) son la salidas de la red neuronal pertenecientes al intervalo [0,1]. Por tanto, o=G(v) con $v \in [0,1]^n$, $o \in [0,1]^m$ y G continua, representa la red. Dado cualquier entrada-salida, v-o, la regla correspondiente en el sistema experto difuso viene dada por:

$$\text{R: Si } x \text{ es } \tilde{A} \text{ entonces } z \text{ es } \tilde{C}.$$

donde:

- el conjunto difuso $\tilde{A}$ se define como $m_{\tilde{A}}(j) = v_j,\ j = 1,...,n$ y 0 en otro caso.
- el conjunto difuso $\tilde{C}$ se define como $m_{\tilde{C}}(i) = o_i, i = 1,...,m$ y 0 en otro caso

3.4.4.12 CLASIFICADORES MEDIANTE REJILLAS DISPERSAS

Desarrollado por PRUDSYS [PRU02], trata de resolver el problema de modelado no lineal aproximándolo a una rejilla espacial, a modo del cálculo de elementos finitos, de tal forma que, para cada punto de la rejilla de un subespacio $V_n \subset V$ define una función que se ajusta a la nube de puntos a clasificar y donde *n* es el número de puntos de la rejilla en ese subespacio.

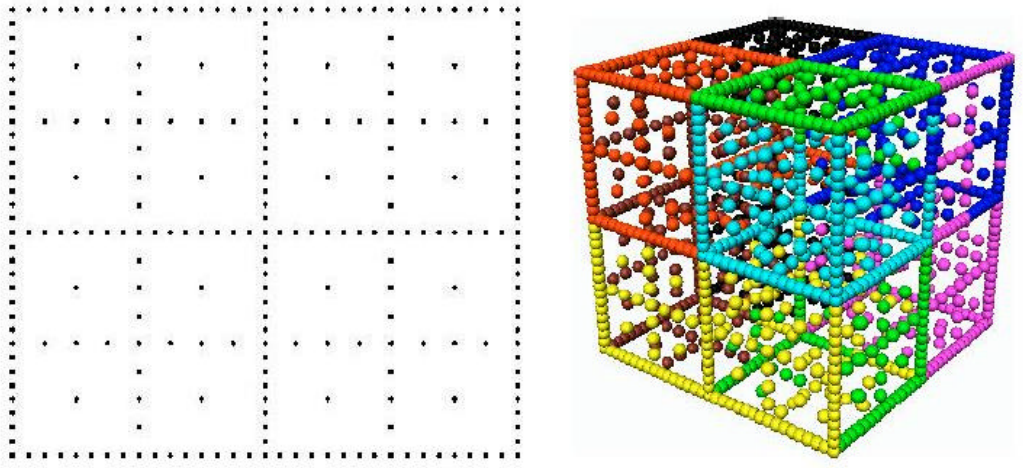


Figura 100.Rejillas dispersas 2D y 3D.

El problema surge con la dimensionalidad, ya que, a medida que aumenta ésta, el número de puntos de la rejilla necesarios para cubrir cada espacio crece exponencialmente y por lo tanto la capacidad de procesamiento necesaria.

Para resolver esto, los clasificadores mediante rejillas dispersas se basan en dos ideas [GAR01][GAR02]:

- La aplicación de un jerarquía de rejillas anidadas mediante la combinación de técnicas de rejillas dispersas. Éstas se basan en la combinación jerarquizada de rejillas no-isotrópicas. De esta forma, la dimensión del espacio de una rejilla dispersa $V_n^{(s)}$ es del orden de $O(n^{d-1} \cdot 2^n)$.
- El uso, en cada punto de la rejilla, de funciones básicas jerárquicas, como por ejemplo las *Wavelets* que combinan las ventajas de las funciones clásicas de elementos finitos, que trabajan en el dominio del espacio (o el tiempo), y las de Fourier, que trabajan en el dominio de la frecuencia.

La aplicación práctica se basa en el uso de técnicas combinatorias de varias rejillas clásicas, tal y como se muestra en la figura siguiente.

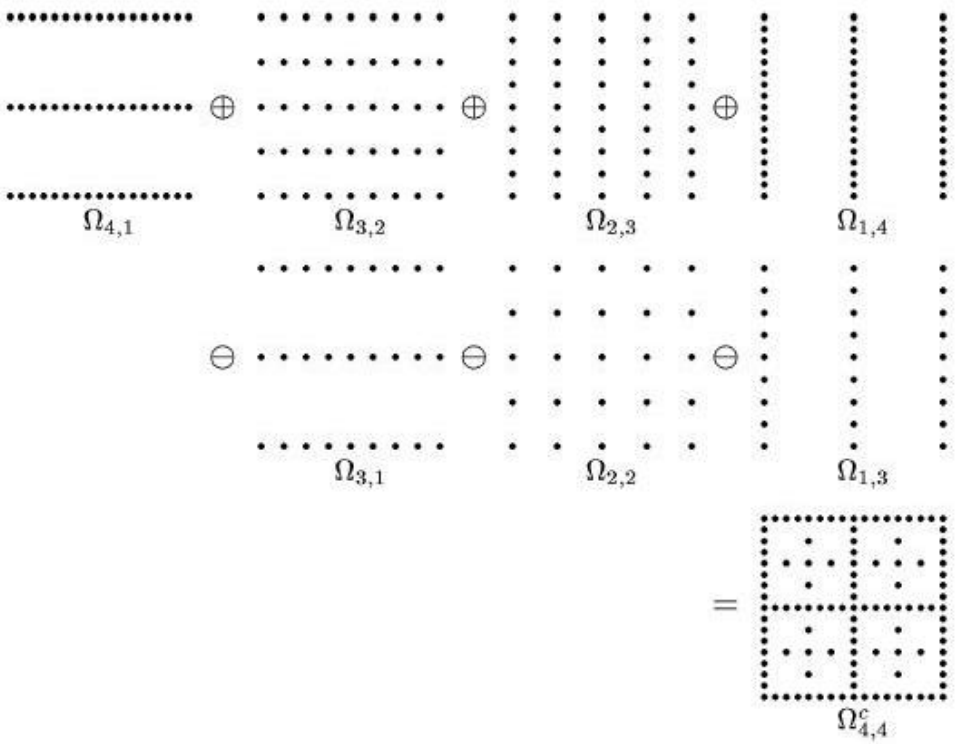


Figura 101.Construcción de una rejilla dispersa mediante otras más básicas [PRU01].

De esta forma y según la teoría de las Rejillas Dispersas, es posible construir *wavelet* básicas de alta dimensionalidad con todas las ventajas de las *wavelets*. De esta forma, se puede construir un subespacio V_n óptimo y resolverlo.

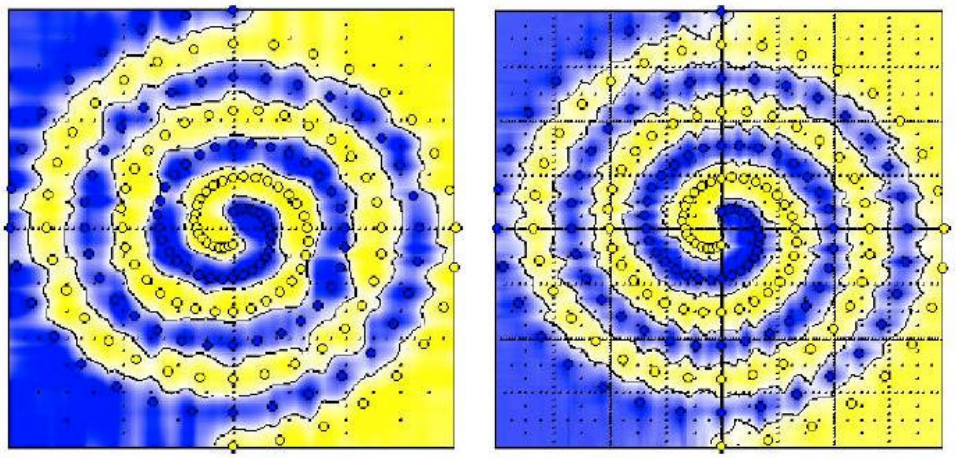


Figura 102. El problema de clasificar dos espirales entrelazadas [GAR01].

Las ventajas principales son:

- Debido a que el clasificador f_n (f_n Ì V_n) no se construye según los puntos sino según la forma del espacio, la eficiencia computacional solo depende linealmente del número de puntos, lo que lo hace muy adecuado para el tratamiento de grandes volúmenes de datos.
- Debido a que la función de clasificación f_n se representa mediante funciones básicas *wavelets* de diversos niveles, puede analizarse en cada nivel.
- Puede paralelizarse el proceso para mejorar el tiempo de creación del clasificador.

3.4.4.13 MÉTODOS BASADOS EN MÁQUINAS DE VECTORES SOPORTE (SUPPORT VECTOR MACHINES SVM)

Los métodos basados en Máquinas de Vectores Soporte o más comúnmente llamados *Support Vector Machines* (SVM) han llegado a ser extremadamente populares en los últimos años, debido a que ellos son capaces de simular un montón de métodos no-lineales dentro de un número ilimitado de dimensiones [CRI00a][WIT00][PRU02].

Los SVM pueden ser expresados como métodos de solución del problema de regularización siguiente:

$$\min_{f \in V} R(f) \tag{3.165}$$

$$R(f) = \frac{1}{M}\sum_{i=1}^{M} C(f(x_i), y_i) + l \cdot f(f) \tag{3.166}$$

donde:

- $C(x,y)$: es una función de error. Por ejemplo $C(x,y)=(x-y)^2$.
- V: es una función en el espacio $\Re^d$.
- $f(f)$: Es un operador de suavizado, $f(f) = \|Pf\|_2^2$. Por ejemplo: $Pf = \nabla f$
- f: Es una función de clasificación o regresión con las requeridas propiedades de suavizado del operador P.
- l: es un parametro de regularización.

En el caso de los Support Vector Machine la función de coste utilizada es:

$$C(x, y) = |y - x|_e = \begin{cases} 0 & si\ |x-y| < e \\ |y-x| - e & \text{en otro caso} \end{cases} \tag{3.167}$$

que, dentro de la formulación anteriormente empleada, puede demostrarse que se aproxima a la propuesta del método SVM. De esta forma, la función toma la forma siguiente:

$$f(x,a,a^*)=\sum_{i=1}^{M}(a_i^*-a_i)\cdot K(x,x_i)+b \tag{3.168}$$

con una constante b y donde a_i^* y a_i son constantes positivas que resuelven el siguiente problema de programación cuadrática,

$$\min_{a,a^*} R(a^*,a)=e\sum_{i=1}^{M}(a_i^*+a_i)-\sum_{i=1}^{M}y_i(a_i^*-a_i)+\frac{1}{2}\sum_{i,k=1}^{M}(a_i^*-a_i)(a_k^*-a_k)K(x_i,x_k) \tag{3.169}$$

sujeto a las siguientes restricciones:

$$0\le a,a^*\le\frac{1}{2lM} \qquad y \qquad \sum_{i=1}^{M}\left(a_i^*-a_i\right)=0 \tag{3.170}$$

Debido a la naturaleza de este problema de programación cuadrática, solo un número de coeficientes $a_i^*-a_i$ serán diferentes de cero y los puntos de entrada x_i asociados a ellos serán llamados vectores soporte (support vectors). El número de vectores depende de $\frac{1}{2lM}$ y de ε; y la elección del Kernel K determinará las propiedades de suavizado de la solución.

Función Kernel	Esquema de Aproximación
$K(x,y)=x\cdot y$	Lineal
$K(x,y)=(g\ x\cdot y+c_o)^d$	Polinómica de grado d
$K(x,y)=\exp(-g\lVert x-y\rVert)$	Función *de* Base Radial Gaussiana
$K(x,y)=\tanh(g\ x\cdot y+c_0)$	Perceptrón Multicapa (sigmoidea)

Tabla 16. Algunos de los esquemas de aproximación más utilizados.

Debido a que los atributos son utilizados solamente para el cálculo del Kernel, las escalas en SVM son lineales con el número de estos, y por lo tanto el número de dimensiones puede ser muy grande.

La desventaja fundamental estriba en que el número de vectores que pueden ser utilizados debe ser pequeño. Esto es debido a que la matriz $\{K(x_i,x_j)\}_{i,j=1...m}$ es generalmente densa y su número de elementos crece cuadráticamente según el número de vectores.

3.4.4.14 MÉTODOS DE APRENDIZAJE BASADOS EN CASOS (INSTANCE BASED LEARNING)

En este tipo de aprendizaje [AHA92][WIT00], se almacenan los ejemplos de entrenamiento y cuando se quiere clasificar un nuevo objeto, se extraen los objetos más parecidos y se usan para clasificar al nuevo objeto. En este método, los ejemplos iniciales son almacenados y utilizados como "fuente de conocimiento". De esta forma, cuando aparecen nuevos ejemplos, se intentan clasificar mediante alguna medida de distancias o similar, y si no se puede asignar a ninguno de los ya existentes, se almacena como un ejemplo nuevo.

Contrario a los otros esquemas vistos, el proceso de aprendizaje es trivial y sencillo de implementar pero tiene la desventaja de que consume bastante tiempo en la etapa de trabajo del clasificador. Éste aumenta considerablemente según la cantidad de ejemplos almacenados ya que el clasificador cada vez que aparece un nuevo ejemplo, debe determinar si pertenece alguna de las clases ya existentes o no escaneando toda la base de datos de entrenamiento.

También, el ruido afecta considerablemente a este tipo de clasificadores, ya que son almacenados como nuevos casos aunque su generalidad sea mínima.

Por supuesto, estas técnicas pueden basarse en el uso de técnicas avanzadas de clusterizado combinadas con técnicas de eliminación de ruido y atributos ponderados, de forma que, pueden obtenerse métodos de aprendizaje óptimos para muchas aplicaciones de *data mining* [AHA92].

Actualmente se están desarrollando nuevas técnicas basados en este método de aprendizaje aplicando nuevas formas de particionado del espacio, eliminación de ruido, etc [MAR95].

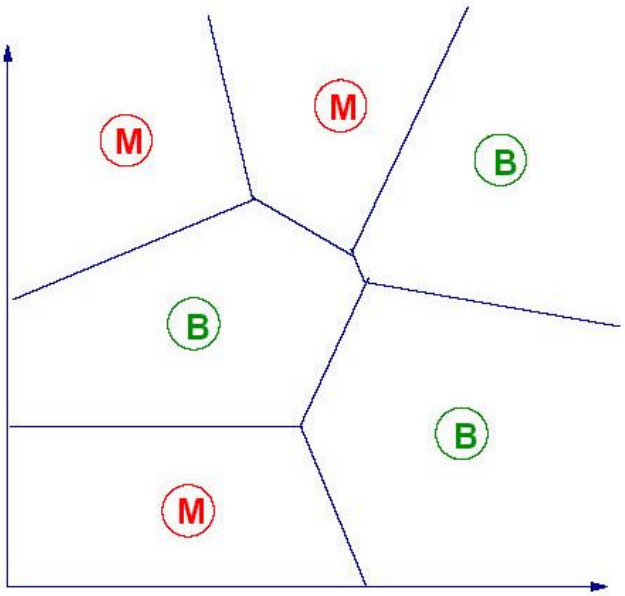


Figura 103.Particionado del espacio en un aprendizaje basado en casos.

3.4.4.15 CLASIFICADORES BASADOS EN ANÁLISIS DISCRIMINANTE

Sea un conjunto de n objetos divididos en q grupos $\{Gi;\ i=1,...,q\}$ de tamaños $\{ng;g=1,...,q\}$ que constituyen una partición de la población de la que dichos objetos proceden.

Sea $Y = (Y_1,..., Y_p)$ un conjunto de variables numéricas observadas sobre dichos objetos con el fin de utilizar dicha información para discriminar entre los q grupos anteriores. Mientras no se diga lo contrario, se supone que dichas variables son cuantitativas.

El Análisis Discriminante [TAB96][SAL00] es una técnica estadística multivadante con una finalidad doble:

- Un fin descriptivo consistente en analizar si existen diferencias entre una sede de grupos en los que se divide una población, con respecto a un conjunto de variables y, en caso afirmativo, averiguar a qué se deben.
- Un fin predictivo consistente en proporcionar procedimientos sistemáticos de clasificación de nuevas observaciones de origen desconocido en algunos de los grupos considerados.

Para llevar a cabo un análisis de este tipo se deben seguir los siguientes pasos:

1. Plantear el problema a resolver.

2. Analizar si existen diferencias significativas entre las grupos.

3. Establecer el número y composición de dimensiones de discriminación para grupos analizados.

4. Evaluar los resultados obtenidos desde un punto de vista predictivo, analizando la significación estadística y práctica del proceso de discriminación.

En cuanto a los pasos matemáticos del desarrollo, se desglosan en los seis puntos siguientes.

- Cálculo de las funciones discriminantes que permitirán discriminar entre los q grupos.
- Determinar el número de funciones discriminantes significativas mediante un contraste de hipótesis secuencial.
- Interpretar los resultados obtenidos desde dos ópticas:
- Significado de las dimensiones de discriminación entre los grupos proporcionadas por las funciones discriminantes mediante el análisis de la matriz de estructura y de la de los coeficientes estandarizados de las funciones discriminantes.
- Análisis del sentido de la discriminación entre dichos grupos, es decir, averiguar qué grupos separa cada función discriminante y en qué sentido. Este análisis se lleva a

cabo mediante representaciones gráficas del espacio de discriminación así como de perfiles multivariantes correspondientes a cada grupo.

- Selección de variables. El problema de selección de variables intenta responder a la pregunta: "¿Son necesarias todas las variables clasificadoras para discriminar?". Para responderla existen, esencialmente, tres tipos de algoritmos: algoritmos de selección de variables hacia adelante, eliminación hacia atrás y de regresión por pasos.

  Los algoritmos de selección hacia adelante comienzan eligiendo la variable que más discrimina entre los *q* grupos. A continuación seleccionan la segunda más discriminante y así sucesivamente. Si de las variables que quedan por elegir ninguna discrimina de forma significativa entre los grupos analizados el algoritmo finaliza. Los algoritmos de eliminación hacia atrás proceden de forma inversa a los anteriores. Se comienza suponiendo que todas las variables son necesarias para discriminar y se elimina la menos discriminante entre los grupos analizados y así sucesivamente. Si las variables no eliminadas discriminan significativamente entre los grupos analizados el algoritmo finaliza.

  Los algoritmos de regresión por pasos utilizan una combinación de los dos algoritmos anteriores permitiendo la posibilidad de arrepentirse de decisiones tomadas con precipitación (bien sea eliminando del conjunto seleccionado una variable introducida en el conjunto de discriminación en un paso anterior del algoritmo, bien sea introduciendo en dicho conjunto una variable eliminada con anterioridad).

  Para determinar qué variables entran y salen en cada paso de este tipo de algoritmos se utilizan diversos criterios de entrada y salida.

  Conviene destacar los siguientes inconvenientes de los procedimientos de selección de variables: No tienen por qué llegar a la solución óptima, utilizan como criterios de selección la separación de grupos y no la clasificación, el nivel de significación global es superior al establecido para introducir y sacar variables debido a la realización simultánea de varios test de hipótesis.

- Clasificación. Existen vados métodos de clasificación dependiendo del número de grupos a clasificar (dos o más grupos), de las hipótesis hechas acerca del comportamiento de las variables en cada grupo (normalidad conjunta, homocedasticidad) así como del criterio utilizado para llevar a cabo dicha clasificación.

Uno de los criterios más utilizados es el criterio Bayes distinguiendo entre los casos de dos y más de dos grupos, si la discriminación se lleva a cabo bajo hipótesis de normalidad o no normalidad y/o bajo hipótesis de horno y heterocedasticidad.

La evaluación del procedimiento de clasificación se hace según tres aspectos del mismo: su eficiencia, su significación estadística y su significación práctica.

Para evaluar su eficiencia se construye la tabla de confusión, que es una tabla de frecuencias cruzadas que refleja los resultados de aplicar dicho procedimiento a los casos observados. Así, en el caso de la discriminación de dos grupos, dicha tabla sería de la forma:

		Grupo Predicho	
		1	2
Grupo Real	1	n_{11}	n_{12}
	2	n_{21}	n_{22}

Tabla 17. Distribución del análisis discriminante para dos grupos.

donde n_{ij} es el número de casos pertenecientes al grupo i y para los cuales el mecanismo de clasificación ha predicho que pertenecen al grupo j. La proporción de bien clasificados vendrá dada por la relación entre aciertos ($n_{11}+n_{22}$) y el total (n).

El proceso de evaluación se puede llevar a cabo de varias formas. Tres de las más utilizadas son las siguientes:

- Con los casos utilizados en el análisis.
- Dividiendo la muestra en dos partes: una para estimar las funciones discriminantes y otra para evaluarla.
- Utilizando, para cada caso, las funciones discriminantes estimadas mediante el resto de los casos.
- Significación estadística. Se evalúa comparando los resultados obtenidos con los que se obtendrían aplicando un mecanismo aleatorio. Los dos mecanismos más utilizados son el criterio de aleatoriedad proporciona y el de máxima aleatoriedad que clasifica todas las observaciones asignándolas al grupo de mayor tamaño.

3.4.4.16 BASADOS EN LA METODOLOGÍAS DE BOX-JENKINS

Una de las herramientas estadísticas más utilizadas para realizar previsiones en series temporales es la metodología de *Box-Jenkins*, de la cual se va a explicar a continuación su método, así como sus ventajas e inconvenientes. Otras aproximaciones se basan en el espacio de estados -filtro de Kalman-, modelos GARCH, modelo bilineal y modelo TAR entre otras, así como el uso de redes neuronales.

La metodología de *Box-Jenkins* [BOX76] consiste en encontrar un modelo matemático que represente el comportamiento de una serie temporal de datos, de modo que para hacer previsiones no haya más que introducir en dicho modelo el periodo de tiempo para el cual se quiere hacer la previsión. Los modelos más utilizados son los conocidos modelos ARIMA univariantes, en los cuales se explica el comportamiento de una serie temporal a partir de las observaciones pasadas de la propia serie y a partir de los errores pasados de previsión (o diferencias entre valores reales del pasado y las correspondientes previsiones utilizando el modelo).

Una ventaja de los modelos de *Box-Jenkins* de previsión es que una vez adquirida experiencia en su metodología resulta más o menos rápido el mecanismo de búsqueda de los modelos, gracias al uso del ordenador. Además, una vez encontrado el modelo resulta inmediato hacer previsiones y comparaciones entre datos reales y previsiones para observaciones pertenecientes al pasado, de modo que resulta fácil ver gráficamente la bondad del modelo elegido. Otra ventaja es que con dichos modelos se obtienen muy buenas previsiones a corto plazo, bajando la calidad de las mismas para plazos muy largos, debido fundamentalmente a la propia estructura de los modelos ARIMA. De todos modos esta conclusión es una generalización ya que cada serie tiene sus propias particularidades. El inconveniente principal de esta metodología es el ya reseñado de realizar peores previsiones a largo plazo. Lo que ocurre es que este inconveniente no lo es tanto debido a la ya indicada relativa facilidad para encontrar modelos nuevos, con lo que resulta muy fácil y práctico ir obteniendo nuevos modelos según se van obteniendo nuevos datos reales, que puedan hacer que el modelo varíe ligeramente, y con los cuales se pueden ir realizando de forma casi continua previsiones a corto plazo.

Como ocurre siempre que se trabaja con la metodología de *Box-Jenkins*, hay que elegir uno entre varios modelos alternativos que sirven para representar a una serie de datos. Se puede tomar como criterio para dicha elección escoger el modelo para el cual sea menor el error estándar de los residuos (RSE). Éste es el criterio más aceptado universalmente, aunque no es siempre perfecto, ya que nos podemos encontrar con circunstancias como que un modelo con mayor RSE hace mejores previsiones a muy corto plazo que otro con un error menor.

3.4.5 BÚSQUEDA DE PATRONES O GRUPOS DE DATOS SIMILARES

El número de técnicas existentes en este grupo son considerables [HAN02].

El uso más común, y donde más se está investigando, es en el terreno de:

- Búsqueda de documentos escritos similares en la web o en bases de datos ingentes.
- Reconocimiento de imágenes similares.
- Búsqueda de información relevante en bases de datos voluminosas.

Generalmente se fundamentan en el uso:

- Modelos para caracterizar los patrones o estructuras de datos a buscar.
- Proyectores y herramientas de clusterizado para clasificar los patrones.
- Métricas de distancia o similares para determinar el grado de similitud.

Estas técnicas, hacen uso de los algoritmos anteriormente descritos combinándolos y adaptándolos a las necesidades que surgan.

3.5 CONCLUSIONES

En este capítulo se ha contextualizado el ambiguo y actual término de *data mining*. Para ello se recurre a una detallada descripción de los términos relacionados con él y a una definición por oposición frente a términos anteriores con los que puede confundirse con relativa facilidad. Se selecciona la acepción más cercana a la extracción de conocimiento de grandes bases de datos.

Dentro de la descripción de las tareas del *data mining*, se ha procedido a exponer las fase principales de este proceso, las herramientas comerciales más utilizadas y metodologías más usadas para el desarrollo de proyectos de *data mining*. Se hace especial hincapié en el CRISP-DM, que es la elegida finalmente por su disponibilidad y generalidad. Se desglosan sus actividades según varios bloques de desarrollo que se seguirán posteriormente.

Por último, se ha hecho un esfuerzo de describir y organizar las técnicas y algoritmos de *data mining* más comunes organizados según las tareas en las que se suelen utilizar. Tras una introducción general de los grupos, se pasa a describir los siguientes grupos de técnicas:

- Herramientas más utilizadas dentro del proceso de análisis exploratorio de los datos: Fundalmentalmente descriptores estadísticos y técnicas de visualización.
- Técnicas para el preprocesado de los datos: filtrado y detección de espurios, rellenado de datos ausentes, transformación y reducción dimensional.
- Modelizado Descriptivo y búsqueda de patrones o estructuras: Organizados en algoritmos de clusterizado y reglas asociativas.
- Modelizado Predictivo: Que corresponde a gran cantidad de algoritmos y técnicas para el desarrollo de modelo que clasifique clases o permitan predecir valores .
- Búsqueda de patrones o estructuras similares: Que permiten buscar dentro de la base de datos, patrones o estructuras de datos similares.

Muchas de estas técnicas, como ya se ha comentado anteriormente, aunque han sido clasificadas en algún grupo, pueden ser utilizadas en cualquiera de las fases del proceso del *data mining*. Por lo tanto, es conveniente considerarlas en su conjunto para poder aplicarlas cuando sea necesario.

Lógicamente, en este capítulo se han descrito los algoritmos o técnicas más comunes, ya que realmente el número de ellos es enorme. Aún así, se considera que la descripción de los mismos puede ayudar a comprender con bastante profundidad las posibilidades que estas herramientas entrañan y cómo el uso del *data mining* puede ayudar considerablemente a mejorar y analizar cantidad de procesos mediante el análisis de sus históricos y de la información periférica de los mismos.

CAPÍTULO 4

ANÁLISIS DEL PROBLEMA: ESTUDIO DEL CONTEXTO Y DETERMINACIÓN DE LOS OBJETIVOS

4.1 INTRODUCCIÓN

A partir de este capítulo, entramos de lleno en la aplicación de la metodología *CRISP-DM* [CRI00] para la comprensión del proceso de galvanizado mediante la adquisición de conocimiento del horno en la fase de calentamiento de la banda y mejora del control del mismo.

Para ello, y según esta metodología, se comenzará por analizar el contexto describiendo las partes claves del proceso, enumerando los recursos disponibles y estudiando el problema. Posteriormente, se plantearán los objetivos reales de este trabajo.

4.2 FASE I: ANÁLISIS DEL PROBLEMA

Según [CRI00], como se comentó en el capítulo anterior, el estándar *CRISP-DM* permite definir todo el proceso de descubrimiento de conocimiento en bases de datos, definiendo etapas y tareas y el significado de cada una de ellas. Lógicamente, la primera de esta etapas corresponde con el **análisis del problema**, donde se busca garantizar la perfecta comprensión del problema planteado y poder llegar así a una definición lo más completa posible de los objetivos finales.

Las tareas propias de esta primera etapa constan de los siguientes pasos:

- Determinación de los Objetivos.
- Evaluación de la Situación.
- Determinación de los Objetivos del *Data Mining*.
- Elaboración de la Planificación.

4.2.1 DETERMINACIÓN DE LOS OBJETIVOS DE NEGOCIO

En este primer paso [CRI00][ABA01], se deben especificar claramente el problema que da origen a la realización del proyecto junto con los objetivos de negocio perseguidos. Si existen objetivos contrapuestos, habrá que ponderarlos adecuadamente.

Como es lógico, esta tarea es fundamental, ya que un error en la definición de los objetivos se arrastrará en todos los procesos posteriores.

Las fases fundamentales son:

1. **Conocimiento Previo**. Donde se trata de recoger toda la información disponible sobre el entorno en el que se desarrolla el proyecto al comienzo del mismo, con el fin de identificar los objetivos y los recursos disponibles (humanos y materiales). Es decir, se analizará la Organización y el Problema.

    a. **Organización**: Departamentos y grupos de trabajo implicados. Personas responsables en cada uno de ellos y conocimiento de cada una de ellas del proceso. Otras personas implicadas y conocimiento de cada una de ellas. Tareas asignadas.

    b. **Problema**: Descripción general del problema. Estado actual del proyecto. Destinatarios de cada tipo de informe o resultado del proyecto. Necesidades y expectativas de los usuarios.

2. **Objetivos de Negocio**: Se describen detalladamente los objetivos del proyecto, tanto los principales como cualquier otro requerimiento secundario. Para ello:

    a. Se describirá informalmente el problema de *Data Mining*.

    b. Se detallarán los aspectos del problema.

    c. Se describirán otros requerimientos esperados.

    d. Se enumerarán los beneficios esperados.

3. **Criterios de Éxito:** Para cada uno de los objetivos planteados en el punto anterior, será necesario especificar un criterio que permita determinar el grado de éxito conseguido. Éstos podrán ser tanto objetivos como subjetivos. En este segundo caso será necesario indicar quién los juzgará.

4.2.1.1 APLICACIÓN

ORGANIZACIÓN

- Departamentos implicados:
  - Equipo Técnico.
  - Mantenimiento.
  - Informática nivel de proceso y de negocio.
- Unidades afectadas:
  - Producción.
  - Mantenimiento.
  - Informática nivel de proceso.
  - Comercial.

PROBLEMÁTICA A SOLUCIONAR

- Área de negocio directamente implicada: Producción.
- **Descripción somera del problema: Detección de las causas que producen paradas y roturas de la banda y, búsqueda de un modelo y de conocimiento que mejore el ajuste de las temperaturas del horno de galvanizado, dentro de la fase de calentamiento, para que la temperatura alcanzada por cada bobina, al final de la zona de calentamiento del horno, sea lo más próxima a la objetivo [LU97].** El sistema debe mejorar la predicción de las temperaturas de consigna de zonas del horno para el tratamiento de la bobina actual y las siguientes. El estudio se centra en la zona de calentamiento por ser la de mayor dificultad.
- Situación actual: El modelo de control del horno corresponde a un modelo matemático empleado para el control de la temperatura de la zona de calentamiento del mismo. Este modelo, desarrollado mediante ecuaciones diferenciales, trata de explicar el comportamiento físico de transmisión de calor de los radiadores del horno a la banda de acero considerando fundamentalmente:
  - Temperatura objetivo de la bobina actual y de las futuras a tratar en la salida de la zona de calentamiento del horno. Estas temperaturas son introducidas al modelo a partir de las curvas de tratamiento térmico preestablecidas para el proceso de galvanizados de cada tipo de acero.
  - La velocidad de la banda.
  - Dimensiones de la misma (anchura y espesor).
  - Temperatura actual de las zonas del horno.

- Temperatura y salto térmico entre subzonas.
- Coeficientes de emisividad, etc.

El modelo actual, utiliza la ecuación básica para calcular las temperaturas de consigna de zonas del horno y velocidad de la banda. Posteriormente realiza un proceso iterativo, donde introduce términos de convección para el afino de la solución. El modelo matemático se ha ido ajustando y mejorando paralelamente en toda la fase de estudio de los datos y desarrollo de este trabajo.

- **Necesidades y Expectativas**: Fundamentalmente son dos:
  - Búsqueda de la mejora en la predicción de las temperaturas de zona del horno y velocidad de la banda para la mejora de la calidad del proceso de galvanizado.
  - Análisis de las causas de reducción de paradas para el aumento de la producción y la reducción de costes. Búsqueda de conocimiento y comprensión de las faltas del sistema actual.
- **Equipo de trabajo**: Es un equipo reducido claramente sensibilizado por la mejora del proceso lo que simplifica la transmisión de conceptos propios del *DM*.

OBJETIVOS DE NEGOCIO

Los objetivos principales son:

- **Detección de las causas que producen paradas o roturas de la banda.**
- **Obtención de conocimiento, mediante técnicas de *DM*, de las causas que pueden inducir fallos o errores elevados entre la temperatura de la banda objetivo y la real. Determinación del grado de eficiencia del modelo actual.**
- **Desarrollo de un modelo no lineal que mejore las predicciones de temperatura de consigna de las zonas de calentamiento del horno y velocidad de la banda adecuada para la bobina actual y siguientes.**
- **Desarrollo de técnicas de análisis y supervisión *on-line* del proceso de galvanizado.**

De estos, se pueden desglosar objetivos más específicos:

- Reducción del número de paradas o roturas de la banda.
- Mejora de la calidad en el proceso de galvanizado gracias a un mejor ajuste del mismo.
- Mejor control del proceso.
- Reducción del número de bobinas tratadas erróneamente.

- Mejor comprensión del proceso de galvanizado mediante el análisis de los históricos del mismo. Obtención de conocimiento que explique "errores elevados" entre la temperatura esperada de la banda y la real.
- Clasificación de las bobinas según el proceso de galvanizado.
- Conocimiento de las variables más características que influyen en el proceso y en qué grado. Calidad y relevancia de cada una de ellas.
- Desarrollo de herramientas que mejoren el control y la supervisión del proceso. Identificación de futuras fallas.
- Determinación y visualización de los diferentes puntos de operación del horno.
- Mejora de la productividad.
- Aumento de la confianza del operario de producción.
- Mejora de la confianza para el cliente.

CRITERIOS DE ÉXITO

El grado de cumplimiento de los objetivos anteriormente descritos será medido por los siguientes criterios:

Criterios Medibles

- Grado de Eficiencia del Modelo Actual y del Nuevo Modelo: Tanto por ciento de bobinas cuyo error entre las temperaturas objetivo y real de la banda a la salida de la zona de calentamiento del horno, sea menor de un umbral predeterminado.
- Número de paradas por mes producidas por errores. Número de horas de No Producción evitadas y existentes.
- Número de veces que ha sido necesario el paso a modo manual debido a imposibilidades de cada modelo.
- Calidad final promedio obtenida en el galvanizado de las bobinas.
- Uso que se realiza de los nuevos métodos de supervisión y monitorización. Aumento de la calidad en el control.

Criterios No Medibles

- Mejora obtenida del conocimiento del proceso de galvanizado en la fase de calentamiento del horno.
- Grado de confiabilidad de los operarios al nuevo sistema.
- Mejora del nivel de confianza del cliente.

Los criterios medibles como no medibles deberán ser valorados por el equipo técnico tanto en el momento de estudio *off-line* como en los procesos posteriores de aplicación *on-line*.

4.2.2 EVALUACIÓN DE LA SITUACIÓN

En esta segunda fase se analizan con más detalle todos los factores que pueden influir y afectar al desarrollo posterior del proceso de *data mining*.

Las fases en que se divide son:

1. **Recursos Disponibles**: Se describe una lista de todos los recursos disponibles en:

    a. **Datos**: Se describe cada una de las fuentes de datos: su origen, el grado de fiabilidad, el modo de acceso, el conocimiento previo sobre las mismas, el tipo de datos y cantidad, etc.

    b. **Hardware**: Se enumera todo el hardware del que se dispone para cada una de las fases del proceso: adquisición, análisis, control, monitorización, etc. Se analiza la posibilidad de acceso, mantenimiento necesario, debilidades y fortalezas, etc.

    c. **Software**: Se realiza una descripción de las herramientas software que se tienen para los procesos de adquisición, análisis, monitorización, etc. Se analizarán las posibilidades y carencias.

    d. **Personal**: Se definen las personas implicadas en el proyecto: administradores, analistas, expertos en *DM*, etc. Se determinará la disponibilidad de cada uno de ellos.

2. **Requerimientos, Supuestos y Restricciones:** Se realiza una lista de requerimientos, restricciones y supuestos relativos tanto a la planificación del proyecto como a los datos o recursos disponibles.

    a. **Requerimientos:** Se establecen los requerimientos del proyecto, incluyendo tanto los temporales como aquellos que tengan que ver con los resultados (comprensibilidad, precisión, capacidad de explotación, mantenibilidad, repetitibilidad, etc.). También todos aquellos que tengan que ver con aspectos de seguridad, restricciones legales y privacidad.

    b. **Supuestos:** Se clarificarán todos aquellos supuestos relativos a los datos (calidad necesaria, tipos de datos a utilizar, cantidad de los datos, etc.) y a otros factores externos. Se determinará la forma en que deben ser presentados los resultados.

    c. **Restricciones:** Restricciones debidas a: Posibilidades de acceso, tiempo, acceso a datos confidenciales, acceso al conocimiento de los expertos, costes, etc.

3. **Riesgos y Contingencias**: Se realizará una lista de circunstancias que puedan retrasar o impedir la realización del proyecto y se planificarán las acciones a llevar a cabo si se producen. Los más comunes pueden ser:

    a. **Riesgos relativos a la Organización**: Debidos a falta de motivación, pérdida de personal asignado al proyecto, cambios en las estrategias de la organización, etc.

    b. **Riesgos Financieros**.

    c. **Riesgos Técnicos**.

    d. **Riesgos Relativos a los Datos**: Datos que no pueden conseguirse, mala calidad de los mismos, etc.

4. **Terminología**: Se elaborará un glosario con toda la terminología relevante del proceso industrial y del proceso de *DM*, de forma que sea fácilmente comprensible por cualquier miembro del equipo involucrado en el proyecto.

5. **Costes y Beneficios**: Se realizará una comparación de los costes del proyecto con los beneficios esperados, cuantificándose todo lo que sea posible. Se incluirán los costes de adquisición, desarrollo e implementación de los resultados, así como los beneficios obtenidos.

4.2.2.1 APLICACIÓN

RECURSOS DISPONIBLES

A continuación se realiza una descripción de los recursos disponibles:

Hardware

Para el análisis y desarrollo de las técnicas de *DM* se disponen de:

- Dos equipos de sobremesa con arquitectura PC-Ofimática, con sistema operativo Windows XP Profesional y 512 Megabytes de memoria.
- Otro equipo con Linux Red Hat 7.1 y 1 Gigabyte de memoria RAM para el entrenamiento de las Redes Neuronales.

Software

Se disponen de las siguientes herramientas *sofware*:

- Microsoft Office: Access, Word, Excel, etc.
- Herramienta GNU de análisis estadístico y tratamiento de datos: R.
- Herramientas GNU de Data Mining: WEKA y XELOPES.
- Matlab 6.1 con diversas toolboxes.
- Herramientas GNU de diseño de Redes Neuronales: SNNS.
- SPSS.
- Herramientas GNU de Visualización Multivariante.
- Herramientas de manejo de bases de datos en Linux Red Hat 7.1.
- Etc.

Fuentes de los Datos

Los sistemas de adquisición de datos se dividen en varios niveles:

- Nivel I: Automatización Básica:
  - Sistema de Control Distribuido Fisher & Porter.
  - PLCs de Telemecánica.

  - Datos de operador: Registrados en *data logger* y en papel.

- Nivel II: Ordenador de Proceso.
  - Seguimiento de bobinas.
  - Base de datos (históricos).
- Nivel III: Ordenador de Negocio.
  - *Test-tracking* de laboratorio.
  - Control Integrado de Producción.
  - Diseño de Producto.

Tipos de Datos

Los datos son obtenidos del proceso de galvanizado a partir de:

- Datos *on-line* de la zona de calentamiento del horno de galvanizado:
- Datos obtenidos del proceso anterior de laminación: tipo de bobina, tipo de acero, dimensiones, etc.
- Datos obtenidos de la tijera después del cromatado.

Inicialmente, el número de variables disponibles en la base de datos es de 6.890 que corresponden a todo el proceso desde el laminado, aunque solo un número reducido de variables son de interés para este trabajo.

Las variables se dividen en:

- Numéricas: Que corresponden a todas aquellas que implican la medida de sensores.
- Categóricas: Indican el tipo de bobina, tipo de acero, código de bobina, etc.

Otros Datos corresponde a Fuentes de Conocimiento que se obtendrán de:

- Entrevistas con los expertos.
- Informes.
- Documentos descriptores del sistema.
- Artículos y bibliografía especializada.
- Participación en foros de *CRISP-DM*.

REQUERIMIENTOS, SUPUESTOS Y RESTRICCIONES

- Debido a motivos de confidencialidad, NO SE PUEDE DISPONER DE LA IMPLEMENTACIÓN ACTUAL DEL MODELO MATEMÁTICO. Por lo tanto, habrá que estudiar y deducir el comportamiento del modelo mediante el análisis de los históricos y el estudio de la documentación suministrada.

- Se requerirá una base de datos lo suficientemente grande para poder entrenar y testear una red neuronal con fiabilidad.

- El 99% de los datos deberán estar libres de errores.

- Algunos datos deberán ser enmascarados para evitar problemas de confidencialidad.

- La agenda del proyecto, debido a las distancias, deberá tomar en cuenta la disponibilidad de tiempo del personal, de forma que, las reuniones se limitarán a aquellos momentos en que los resultados intermedios necesiten ser estudiados para determinar pasos posteriores.

- Se aprovechará la comunicación vía *e-mail* para la resolución de dudas puntuales.

- **Todos los estudios del proceso se realizarán *off-line***. Posteriormente, una vez comprobados los resultados durante un tiempo prudencial, se decidirá la implantación o no del control del horno. Para ello, y siempre si la empresa lo estima oportuno, se implementará en paralelo monitorizando las predicciones y ajustes que realiza y simulando los resultados. La decisión de implantación final recae en la empresa.

- Todos los resultados obtenidos serán debidamente cotejados por el personal experto de la empresa.

- Los procesos finales serán documentados con los listados correspondientes.

RIESGOS Y CONTINGENCIAS

Riesgos Relativos a la Organización

- Debido a las numerosas tareas del personal especializado, será necesario motivarlo adecuadamente mediante informes periódicos de los avances que se están realizando.

- Será necesario una comunicación lo más flexible y fluida posible entre los diferentes agentes encargados. Los requerimientos pedidos deberán ser especificados con entera claridad.

Riesgos Técnicos

- Será necesario que la comunicación entre los distintos niveles de operación del proceso sea correcta, debido a que una deficiencia en la adquisición o almacenamiento de los datos puede bloquear el proceso de análisis.

- Las herramientas hardware-software deben poder tratar gran cantidad de información con tiempo reducidos de procesado.

- Los datos deberán estar sincronizados en el tiempo y asignados correctamente a cada bobina. Tendrán que ser de buena calidad y completos. Para ello, será necesario que el personal encargado de la adquisición de la información, verifique la calidad de los mismos.

COSTES Y BENEFICIOS

Los costes planteados son reducidos, debido a que solamente se tomarán en cuenta los relativos al personal de la empresa encargado de la adquisición de la información.

Como el proceso relativo a este trabajo es *off-line* no se tomarán en cuenta los costes derivados de las pruebas *on-line*. Estos deberán ser estudiados a *posteriori*.

Los beneficios deberán ser calculados mediante simulación usando la siguiente metodología:

1. Determinación de la calidad final de cada una de las bobinas y su valor real.

2. Selección de las bobinas cuya calidad a empeorado debido a problemas en el modelo del horno. Determinación del tanto por ciento de bobinas con baja calidad.

3. Simulación del tratamiento con el nuevo modelo.

4. Estimación de la calidad que se habría obtenido con el nuevo modelo. Obtención del valor teórico de las bobinas tratadas.

5. Cálculo del beneficio posible a partir del valor teórico menos el real.

También se determinarán los beneficios por reducción de número de paradas y tiempo en las mismas, o eliminación de otros sucesos anómalos.

4.2.3 DETERMINACIÓN DE LOS OBJETIVOS DEL *DATA MINING*

Una vez se han planteado los objetivos de la empresa y realizado un inventario de los recursos humanos y no humanos de que se dispone, la metodología *CRISP-DM* [CRI00] considera que ahora se deben desarrollar los objetivos desde una perspectiva más técnica. De esta forma, se pretende adaptar los objetivos planteados anteriormente a unos nuevos desde el punto de vista del *Data Mining*.

Las fases en que se dividen son:

1. **Objetivos del *Data Mining***: Se desglosará cada uno de los objetivos anteriores en las tareas necesarias propias del proceso de *DM*. Así mismo, se describirán las herramientas o técnicas que podrán ser usadas para la consecución con éxito de las tareas descritas.

2. **Criterios de éxito del *Data Mining***: Se expondrán los criterios que permitan determinar, de forma técnica y lo más objetiva posible, el grado de alcance de los resultados obtenidos en cada una de las tareas de *DM* realizadas. Si alguno de los criterios es subjetivo, se designará la persona o personas que lo evaluarán.

4.2.3.1 APLICACIÓN

A continuación se pasan a exponer, en orden temporal, los objetivos del proceso del *DM*, los pasos necesarios y las técnicas a emplear, así como los criterios para valorar el grado de éxito conseguido.

OBJETIVOS DEL PROYECTO DE DATA MINING

1. Determinación, mediante técnicas estadísticas y de Minería de Datos, **de las variables más importantes del proceso y del grado de importancia de cada una de ellas**. Tareas:

   - Preprocesado y transformación de los datos: eliminación de ruidos, análisis visual, uso de técnicas de visualización multivariante para la eliminación de espurios, transformación de la información.

   - Uso de gráficos multivariantes, índices estadísticos y de correlación para identificar las variables más importantes.

   - Reducción dimensional mediante proyectores lineales o no lineales.

2. **Obtención de conocimiento** mediante técnicas extractoras de patrones de comportamiento, **de los sucesos que pueden generar errores en el proceso de galvanizado.**

- Preparación de una base de datos con las variables más características.
- Búsqueda de grupos mediante algoritmos de clusterizado y métodos de visualización multivariante. Identificación de casos anómalos.
- Clasificación de los grupos según el tipo de acero y el grado de error.
- Uso de algoritmos y técnicas extractoras de patrones de comportamiento.

3. **Extracción de reglas que permitan modelizar categóricamente las variables y los sucesos producidos.**
   - Preparación de la bases de datos anterior para el tratamiento correcto con los clasificadores.
   - Uso de clasificadores para obtención de reglas o modelos que expliquen los diferentes grupos obtenidos.
4. **Generación de un modelo No-Lineal, basado en redes neuronales, que mejore el ajuste de las temperaturas de consigna de las zonas del horno y velocidad de la banda, reduciendo el error existente entre la temperatura de la banda, esperada y real, a la salida de la zona de calentamiento del horno** [CHE02]**.**
   - Selección de la red neuronal apropiada.
   - Preparación de la base de datos para el entrenamiento y validación de la misma.
   - Entrenamiento y testeo del modelo no lineal obtenido.
5. **Desarrollo de *Sensores-Software*** para la monitorización y generación de alarmas [CHI03]:
   - Uso de proyectores lineales o no lineales para la caracterización del punto de operación.
   - Clasificación de los diferentes puntos de operación.
   - Determinación de los puntos de operación anómalos.
   - Generación de *Sensores-Sofware* que generen alarmas e informes de estado del proceso.

CRITERIOS DE ÉXITO DEL PROYECTO DE DATA MINING

Para verificar los puntos anteriores, será necesario establecer unos criterios que determinen el grado de éxito de los resultados anteriores. De esta forma, para cada uno de los puntos anteriores se formulan los siguientes baremos:

1. Para el caso de determinación de las variables más significativas del proceso, se tomarán en cuenta los siguientes puntos:
    a. Grado de eficacia de los modelos obtenidos con esas variables.
    b. Acciones llevadas a cabo debido al conocimiento obtenido.
    c. Estimación del grado de confiabilidad de los expertos según su experiencia.
2. Para la extracción del conocimiento de las causas que puedan producir errores:
    a. Tanto por ciento de casos anómalos explicados.
    b. Valoración de los expertos del conocimiento adquirido.
    c. Mejora de la calidad debida a este conocimiento.
    d. Reducción de tiempos muertos y de incidencias por la mejora del conocimiento del sistema.
    e. Uso que se realiza de esos conocimientos.
3. Para el proceso de extracción de reglas de conocimiento:
    a. Grado de uso que realizan los expertos de las mismas.
    b. Contingencias y nuevas estrategias obtenidas a partir de las mismas.
    c. Mejora del proceso debido al uso de estas reglas.
    d. Número de prevenciones positivas de fallos o sucesos anómalos.
4. Para el modelo desarrollado:
    a. Error medio entre las temperaturas de la banda reales y las de consigna en las fases de entrenamiento y de validación. Se usará validación cruzada.
    b. Número de bobinas con errores altos frente al modelo actual.
    c. Eficiencia en la predicción de temperaturas reales frente al modelo actual.

d. Capacidad de reacción del modelo.

5. Para los Sensores-Software:

   a. Aceptabilidad por parte de los expertos.

   b. Capacidad de predicción de contingencias.

   c. Conocimientos adquiridos de su uso.

4.2.4 ELABORACIÓN DE LA PLANIFICACIÓN

Por último, dentro de la primera fase de la metodología CRISP-DM, se procede a realizar un planificación inicial teórica de las tareas a realizar y de las herramientas a usar.

Está compuesto por dos fases:

1. **Planificación del proyecto**: Donde se enumeran todas las tareas a desarrollar en el proyecto, indicando si es posible, la duración estimada, recursos necesarios, entradas, salidas y dependencias, riesgos asociados, etc. También se podrá definir los momentos en que se deberá revisar la evolución del proyecto, que pueda producir modificaciones de la planificación inicial.

2. **Valoración inicial de técnicas y herramientas**: Se enumerarán las técnicas y herramientas de que se dispondrán. La elección final de las mismas dependerá de fases posteriores.

4.2.4.1 APLICACIÓN

PLANIFICACIÓN Y TÉCNICAS PREVISTAS

Dentro de la planificación del proyecto se plantearon las siguientes tareas:

1. Desarrollo de la lista inicial de variables a tratar a partir del conocimiento de los expertos, documentación del modelo matemático, etc.

   - Tiempo Previsto: 2 semanas.

   - Métodos y Técnicas Previstas: *Delphi*, contraste de opiniones, estudio de documentación.

   - Recursos Necesarios Previstos: Expertos y documentación explicativa del funcionamiento del horno y del modelo actual.

   - Entrada: 6.890 variables.

- Salida Prevista: Primera lista de variables con la que trabajar.

2. Recogida de los datos de la base de datos de históricos.

   - Tiempo Previsto: 1 mes.
   - Métodos y Técnicas Previstas: Tratamiento con herramientas de base de datos (*Access*, *Oracle*, etc.).
   - Recursos Necesarios Previstos: Operador de planta para la adquisición de los históricos, ordenadores conectados al proceso.
   - Entrada: Históricos diarios del procesado de las bobinas.
   - Salida Prevista: Base de Datos del tratamiento de mínimo 2.500 bobinas.

3. Estudio exploratorio de los datos mediante técnicas estadísticas y de visualización multivariante, selección de variables más características, estudio y filtrado de espurios, reducción dimensional, transformación de los datos:

   - Tiempo Previsto: 9 meses.
   - Métodos y Técnicas Previstas: Técnicas de visualización estadística y multivariante.
   - Recursos Necesarios Previstos:
     - Herramienta R de análisis estadístico: diagramas *box-plots*, diagramas *scatter-plots*, índices estadísticos, clusterizado, proyectores lineales y no lineales, etc.
     - Herramientas de visualización multivariante.
     - Matlab 6.1.
     - Microsoft Access.
     - Ordenador PC-Ofimático 500 Megas.
   - Entrada: Base de Datos obtenida en el paso anterior.
   - Salida Prevista: Base de datos filtrada, con las variables más importantes, libre de ruidos y con los datos transformados.

   Si los datos no son adecuados, será necesario volver al punto 2. Si se supone necesaria alguna variable no considerada inicialmente, habrá que recomenzar en el punto 1.

4. Extracción de patrones de comportamiento, reglas, etc.

   - Tiempo Previsto: 4 meses.
   - Métodos y Técnicas Previstas: Árboles clasificadores, reglas de decisión, reglas asociativas, herramientas de clusterizado y visualización multivariante, etc.
   - Recursos Necesarios Previstos:
     - Herramienta GNU WEKA y librería XELOPES con algoritmos de *DM*.
     - Herramienta R de análisis estadístico: diagramas *box-plots*, diagramas *scatter-plots*, índices estadísticos, clusterizado, proyectores lineales y no lineales (PCA, SOM, SAMMON), etc. [TOM02].
     - Herramientas de visualización multivariante: coordenadas paralelas, RADVIZ, etc.
     - Matlab 6.1.
     - Microsoft Access.
     - Ordenador PC-Ofimático 500 Megas.
   - Entrada: Base de datos filtrada, con las variables más importantes, libre de ruidos y con los datos transformados.
   - Salida Prevista: Reglas y árboles que expliquen los comportamientos anómalos.

   Si los datos no son adecuados, será necesario volver al punto 3.

5. Creación del modelo no lineal mediante redes neuronales: entrenamiento y validación, búsqueda del modelo óptimo.

   - Tiempo Previsto: 4 meses.
   - Métodos y Técnicas Previstas: redes neuronales supervisadas basadas en Backpropagation o no supervisadas basadas en ART1, ART2 o Fuzzy-ART.
   - Recursos Necesarios Previstos:
     - Herramienta SNNS para el entrenamiento y testeo de redes neuronales.

- Herramienta R de análisis estadístico.
- Matlab 6.1: Toolboxes de lógica difusa y redes neuronales.
- Ordenador PC-Ofimático 500 Megas.

- Entrada: Base de datos filtrada, con las variables más importantes, libre de ruidos y con los datos transformados para la creación de redes.
- Salida Prevista: Modelo basado en red neuronal o *neuro-fuzzy* con mejor calidad de predicción que el modelo actual.

6. Desarrollo de *Sensores-Software* para monitorización y control de puntos de operación, detección de sucesos anómalos, etc.
   - Tiempo Previsto: 2 meses.
   - Métodos y Técnicas Previstas: proyectores lineales o no lineales, redes neuronales, técnicas de análisis estadístico, métodos de visualización multivariante, etc.
   - Recursos Necesarios Previstos:
     - Herramienta R de análisis estadístico: diagramas *box-plots*, diagramas *scatter-plots*, índices estadísticos, clusterizado, proyectores lineales y no lineales (PCA, SOM, SAMMON), etc.
     - Herramientas de visualización multivariante: coordenadas paralelas, RADVIZ, etc.
     - Herramienta SNNS para el entrenamiento y testeo de redes neuronales.
     - Matlab 6.1: toolboxes de lógica difusa y redes neuronales.
     - Ordenador PC-Ofimático 500 Megas.
   - Entrada: Base de datos inicial para la fase de simulación. Datos on-line para el proceso de monitorización y ajustes.
   - Salida Prevista: Sensor-Software que permita monitorizar el punto de operación del horno y prever fallos anómalos.

4.3 CONCLUSIONES

En este capítulo, se ha desarrollado la primera fase de aplicación de la metodología *CRISP-DM* [CRI00], donde se analiza el contexto y se definen los objetivos del proyecto de *Data Mining*.

Para ello, inicialmente se han planteado los objetivos de negocio y estudiado los recursos disponibles, para pasar a desarrollar con más detalle, los objetivos propios del proyecto de *DM*.

Una vez enumerados los objetivos del proyecto de *DM*, se ha planteado una planificación inicial junto con una previsión de las herramientas que serán necesarias.

En los capítulos siguientes, se describen los resultados que se han obtenido para cada una de las fases posteriores de la metodología *CRISP-DM* y el grado de cumplimiento de los objetivos aquí descritos.

CAPÍTULO 5

ANÁLISIS Y PREPARACIÓN DE LOS DATOS

5.1 INTRODUCCIÓN

El preprocesado de los datos es una de las tareas más importantes dentro de todo proceso que pretenda extraer conocimiento, modelar un sistema o evaluarlo. Efectivamente, varios autores subrayan la importancia, para la consecución con éxito del trabajo de *DM*, de las tareas iniciales de definición del problema, análisis de los datos y preparación de los datos [PYL99][WAN99].

Tarea	Porcentaje del Tiempo dedicado (en %)	Importancia para llegar al éxito final (en %)
1. Definir el Problema	**10 %**	**15 %**
2. Explorar la Solución	**9 %**	**14 %**
3. Implementación de las especificaciones	**1 %**	**51 %**
4.1. Data Mining: Análisis de los datos.	**60 %**	**15 %**
4.2. Data Mining: Preparación de los datos	**15 %**	**3 %**
4.3. Data Mining: Modelizado y testeo de los datos.	**5 %**	**2 %**

Tabla 18. Tiempo e Importancia de cada una de las fases del DM (según [PYL99]).

La primera de estas fases, desarrollada en el capítulo anterior, corresponde con el **análisis del problema**, donde se buscaba garantizar la perfecta comprensión del problema planteado y poder llegar así a una definición lo más completa posible de los objetivos finales.

En este capítulo y en el siguiente, se describen los pasos realizados en la segunda y tercera fases del proceso de *CRISP-DM:* el análisis y comprensión de los datos y, la preparación de los mismos.

5.2 OBJETIVOS

Las fases dos y tres de la metodología CRISP-DM, tal y como hemos visto en el capítulo tres, tratan de preparar la información de la mejor forma posible para la posterior etapa correspondiente al modelizado.

Podemos decir que **el objetivo fundamental de la segunda fase consiste en analizar la información que se tiene verificando la calidad final. Si ésta no es aceptable, será necesario realizar nuevas adquisiciones de datos hasta que la calidad de éstos sea la adecuada.** Una vez conseguidos, **se definen los objetivos de la tercera fase que consisten en preparar la información, seleccionando primeramente las variables más importantes, filtrar y eliminar el ruido y transformar los datos para la posterior etapa de modelizado (fase IV).**

5.3 FASE II: ANÁLISIS DE LOS DATOS

Siguiendo la metodología definida en el estándar *CRISP-DM* [CRI00], la segunda de estas fases comprende al **análisis de los datos**, donde se adquieren los datos, se describen y se analiza la calidad de los mismos.

Ésta se divide en varias etapas secuenciales:

- Adquisición de la información.
- Descripción y exploración de los datos.
- Verificación de la calidad de los mismos.

Este proceso debe realizarse hasta que la calidad de la información sea lo suficientemente buena para que permita la consecución de las fases siguientes con garantías de éxito.

5.3.1 ADQUISICIÓN DE LOS DATOS

El primer paso consiste en la obtención de los datos necesarios de todas las fuentes de que se disponen. **Será necesario integrar las diferentes bases de datos en unas pocas, con las variables más útiles.**

Fundamentalmente se busca:

- Generar un conjunto inicial de datos con los que trabajar.
- Desarrollar un informe en el que se describe la forma de conseguir los datos de las diferentes fuentes y los problemas encontrados.

Se plantean los siguientes pasos:

- **Planificar requerimientos**: Se analiza el tipo de información requerida:
  - Variables necesarias, tipos de rangos de cada variables, etc.
  - Se determina si es posible adquirirla y está disponible.
- **Criterios de Selección**: Se definen los siguientes pasos:
  - Criterios de selección de las variables.
  - Se seleccionan las tablas o ficheros de interés.
  - Longitud y periodo de tiempo de los datos (meses, últimos años, etc.).
- **Inserción de los datos**: Se describe la forma de introducir la información no disponible electrónicamente, ya que puede existir en otros soportes: conocimientos de expertos, papel, informes escritos, etc:
  - Forma de codificarla.
  - Modo de adquirirla.

5.3.2 DESCRIPCIÓN DE LOS DATOS

En esta etapa se describen las características fundamentales de los datos: tablas, variables individuales (cantidad de registros, tipo de datos, descriptores estadísticos, gráficos, etc.), de forma que **se comprenda cómo son los datos**.

Se desarrollarán los siguientes pasos:

- **Análisis volumétrico de los datos**: Identificación de los datos y métodos de captura, forma de acceso a los datos, tablas usadas, volumen de los datos, complejidad, características de los mismos, etc.
- **Tipos y valores de las variables**: Tipos (numéricas, simbólicas, etc.), rango de los valores, disponibilidad de los datos, uso de descriptores estadísticos (máximo, mínimo, media, mediana, varianza, etc.), uso de gráficos descriptores, grado de consistencia de los datos, importancia, etc.
- **Claves**: Analizar relaciones entre tablas, solapación de información, etc.
- **Revisión de objetivos**: Analizar los objetivos iniciales y determinar si las variables son adecuadas.

5.3.3 EXPLORACIÓN DE LOS DATOS

A partir de la descripción de los datos, **se realiza un primer análisis superficial de las características de los datos**, como por ejemplo:

- Relaciones entre variables.
- Tipo de distribución de los datos.
- Agrupamientos.
- Etc.

Para ello, se pueden usar:

- Técnicas de visualización.
- Análisis de correlaciones.
- Técnicas estadísticas.
- Otras Técnicas.

5.3.4 VERIFICAR LA CALIDAD DE LOS DATOS

Al final de esta fase del proceso CRISP DM, se debe determina la calidad de los datos disponibles de forma que si no son suficientemente buenos, será necesario volver a repetir todos los pasos anteriores.

Para ello se analizará si: Los datos tienen errores, describen realmente la realidad, cubren todo el rango, la cantidad de ruido existente, cantidad de datos inexistentes, etc.

Fundamentalmente se realizan las siguientes tareas:

- Revisión de las variables, comprobando:
  - Si se representa la realidad y son consistentes.
  - La cantidad de campos vacíos y el por qué de los mismos.
  - Si existen espurios y sus causas.
  - Variables que no son necesarias.
  - Etc.
- Análisis de los archivos de texto:
  - Comprobando los separadores.
  - Analizando ficheros vacíos.
  - Determinando si el número de campos es el mismo en cada registro.

- Etc.

- Ruido e inconsistencia entre fuentes de datos:
  - Comprobar la consistencia entre las diferentes fuentes.
  - Determinar cantidad de datos redundantes.
  - Detectar el ruido, su procedencia y las variables afectadas.
  - Etc.

5.4 FASE III: PREPARACIÓN DE LOS DATOS

Como ya se ha comentado anteriormente, la fase III de la metodología *CRISP-DM* trata de generar una base de datos óptima para la posterior fase de modelizado (fase IV).

Esta fase consta de las siguientes etapas:

- Selección de los datos.
- Limpieza de los datos.
- Generación de variables adicionales.
- Integración de orígenes de datos.
- Cambios de formato de los mismos.

Estas etapas **se realizarán repetidamente hasta que se obtenga una base de datos lo suficientemente adecuada para las posteriores fases de la metodología *CRISP-DM*.**

5.4.1 SELECCIÓN DE LOS DATOS

En esta primera etapa, se parte de los datos obtenidos en la fase anterior y de la descripción de los mismos. A partir de toda esta información, se realiza una selección de las variables más importantes según los siguientes requerimientos:

- Que sean lo más independientes entre si. Se eliminarán las variables muy dependientes de otras.
- Que describan casi completamente el sistema a describir o analizar.
- Que tengan una relevancia individual destacada. Se descartarán aquellas cuya influencia en el sistema sea nulo o muy escaso.
- Que estén exentas de ruido y con datos fiables.

Así mismo, si el volumen de datos es suficientemente grande, se decidirá si es necesario reducir el número de muestras.

Por ejemplo, un caso muy destacado aparece cuando decidimos dividir los datos en grupos, ya que habrá que homogeneizar la densidad de cada uno de ellos reduciendo los datos en aquellos grupos numerosos. Esto es necesario cuando queremos modelizar varios grupos y en uno de ellos tenemos una densidad mucho más elevada que en los demás, lo que puede influir negativamente en la creación de un modelo que los clasifique, ya que éste dará más peso a los grupos con mayor densidad de individuos.

5.4.2 LIMPIEZA DE LOS DATOS

Como se ha comentado en el punto anterior, las variables deben ser lo más fiables posibles. Para ello habrá que:

Tratar el ruido en los datos:

- Corrigiendo, ignorando o eliminando aquellos datos con ruido.
- Se estudiarán las posibles causas que generan ese ruido y la forma de resolverlo.
- Se usarán técnicas de filtrado para mejorar la calidad de los datos.

Tratar los espurios:

- Analizándolos por separado y determinando las causas que los generaron.
- Eliminándolos.
- Clasificándolos en otros grupos.
- Tratar los valores incompletos:
- Eliminándolos.
- Ignorándolos.
- Completándolos con técnicas estadísticas.
- Rellenándolos con otras técnicas.

5.4.3 GENERACIÓN DE VARIABLES ADICIONALES

Se generarán nuevas variables a partir de las ya existentes siempre que permitan agilizar los estudios posteriores.

Dentro este proceso, se pueden incluir tareas de transformación de los datos como:

- Estandarizar o normalizar variables.
- Asignar pesos según la importancia de cada variable.
- Cambiar la codificación de alguna variable.
- Uso de transformadas.

- Uso de proyectores.
- Adición de nuevas variables a partir de otras.
- Uso de indicadores estadísticos.
- Uso de otros indicadores.

5.4.4 INTEGRACIÓN DE ORÍGENES DE DATOS

Se combinarán los datos procedentes de diferentes orígenes, siempre que no se haya hecho ya, para obtener una base de datos más compacta y útil.

5.4.5 CAMBIOS DE FORMATOS DE DATOS

Se adecuará el formato de los datos para que puedan ser usados por las herramientas que se vayan a utilizar en fases posteriores. Por ejemplo:

- Cambiando el orden de las variables de un registro.
- Cambiando el tipo de variables. Convirtiendo variables numéricas a categóricas, o viceversa.
- Reordenando los datos.
- Etc.

5.5 APLICACIÓN PRÁCTICA DE LAS FASES II Y III DE LA METODOLOGÍA CRISP-DM

La aplicación de la fases II y III de la metodología CRISP-DM ha consistido en un proceso iterativo de ajuste y afinamiento de los datos a manejar. Éste se puede resumir en los siguientes pasos:

- Inicialmente, de las 6.890 variables disponibles, se seleccionaron manualmente aquellas que podían influir en el comportamiento del sistema y se dividieron en grupos para su análisis posterior.
- Se efectuó un análisis volumétrico y de rango de los datos.
- Una vez seleccionados por grupos de variables se realizó un estudio exploratorio, mediante diversas técnicas de visualización, de los valores de las mismas analizando los resultados y los errores que se observaban en ellos.
- Se verificó la calidad de los mismos.
- Las conclusiones obtenidas del análisis inicial se contrastaron con la opinión de los expertos de la planta modificándose las mismas hasta llegar a un acuerdo mayoritario. Se buscaron las causas que originaron los patrones defectuosos y los comportamientos anómalos y se corrigieron.
- Se generaron nuevas variables y nuevas bases datos, repitiendo los pasos anteriores[12], hasta tener una base de datos consistente, de buena calidad y adecuada.
- De las conclusiones obtenidas y de la información contrastada con el personal de la empresa se extrajeron las pautas que servirían para identificar las variables a estudiar en procesos posteriores.

Una vez identificadas las causas de los comportamientos anómalos, se continuó con la fase III de la metodología *CRISP-DM*:

- Primero, se realizó un proceso de filtrado que eliminaba los patrones defectuosos. Aún así, éstos se almacenaron por separado para posteriores estudios.
- Se caracterizaron las curvas dinámicas y se crearon nuevas variables.
- Se definieron las nuevas características y variables necesarias para una nueva base de datos. Se repitieron los pasos anteriores hasta obtener unos datos con una calidad adecuada.
- Por último, se realizó una selección global y se preparó la base de datos de todas las variables con la que se trabajaría en adelante.

[12] Correspondientes a la fase II de la metodología *CRISP-DM*.

5.5.1 EL PROCESO DE ADQUISICIÓN

El proceso de adquisición se ha realizado por personal experto de la empresa del área de informática de procesos. Éste se ha centrado, fundamentalmente, en el volumen de históricos que se genera continuamente en todo el proceso de galvanizado.

El proceso de adquisición se ha realizado de la siguiente forma:

- Se seleccionaron, según el conocimiento de los expertos y de la información disponible del modelo, una serie de variables lo más completa posible de las 6.890 disponibles.
- El personal encargado de la informática de procesos, realizó la selección de las variables pedidas en diferentes tablas de Access.
- Las tablas fueron enviadas al personal investigador encargado del proceso del *DM*.

Este proceso se ha prolongado en el tiempo ya que, la bases de datos adquiridas se han tenido que realizar en tres ocasiones debido a las siguientes causas:

- El modelo del horno y todo el sistema de adquisición se ha ido mejorando y actualizando paralelamente a esta fase de estudio. Debido a esto, los datos iniciales no eran muy adecuados, por lo que las primeras bases de datos fueron descartadas por la cantidad de datos incorrectos.
- No existían algunas variables fundamentales para el proceso del *DM*: modo automático o manual del proceso, espesor real de la banda a la entrada, etc.
- Faltaban variables que no habían sido consideradas inicialmente.

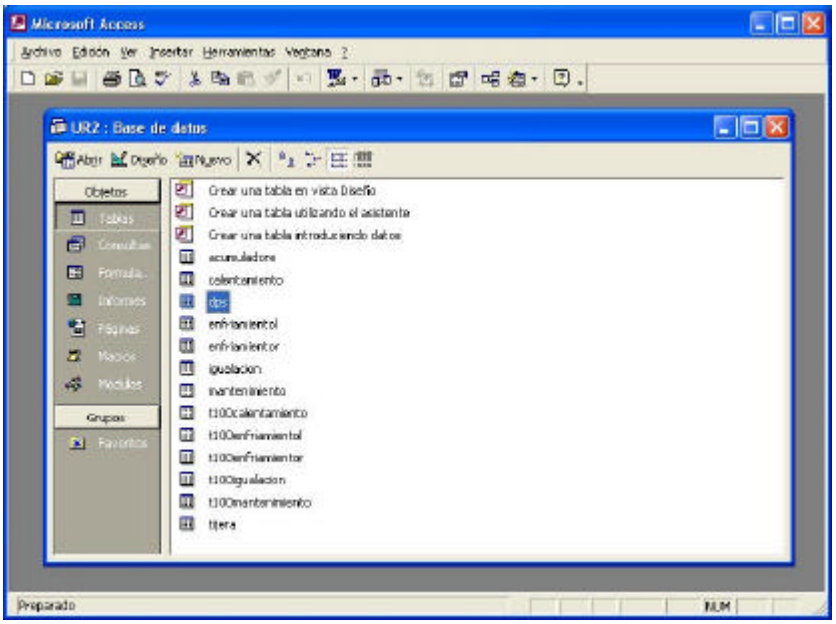


Figura 104. Visualización de las tablas de la primera base de datos utilizada.

5.5.1.1 ETAPAS DEL PROCESO DE ADQUISICIÓN

Las bases de datos se han desarrollado en tres fases:

- La primera de ellas estaba compuesta por catorce tablas con las variables más significativas del proceso de galvanizado, incluyendo las etapas del horno (calentamiento, mantenimiento, igualación, enfriamiento lento, enfriamiento rápido, etc.), la de la tijera, y datos de fabricación de cada bobina procesada. Incluían datos medios por bobina (tablas *"dps"*, *"calentamiento"*, etc.), medidos cada 100 metros de banda (tablas *"T100calentamiento"* , *"T100enfriamiento"*, etc.) y cada 30 metros (Tabla *"Tijera"*). El tiempo de preprocesado y análisis de esta base de datos fue de tres meses aproximadamente.
- De la primera base de datos, se preseleccionaron algunas variables y se detectaron falta de alguna otras. Para ello se desarrolló la segunda base de datos que completaba los datos de la primera. El análisis de los nuevos datos se prolongó unos cinco meses más.
- De los estudios más intensos, realizados en esta segunda base de datos, se detectaron algunas fallas importantes y variables necesarias que no existían dentro del proceso de generación de históricos. De esta necesidad surgió la tercera base de datos. En este caso, debido a que se había trabajado paralelamente en la mejora de todo el proceso de adquisición y control del horno, los datos eran mucho más consistentes presentando muchos menos errores.

Los procesos y estudios realizados en estas tres etapas, se describen con más detalle en apartados posteriores. Éstos, debido a la cantidad de variables tan grandes, se han abordado en grupos de variables.

5.5.2 ANÁLISIS INICIAL DE LAS OBSERVACIONES. PRIMERA Y SEGUNDA BASE DE DATOS

Una vez disponemos de la base de datos de información a procesar, debemos proceder a limpiar la información no relevante y seleccionar las variables más significativas reduciendo lo más posible la dimensión de los datos, para posteriormente pasar al proceso de modelizado.

Lo primero que hacemos, es conectarnos mediante ODBC (*Open DataBase Connectivity*) al servidor de bases de datos en *MySql* para obtener la información a procesar.

```
# Cargamos la Librería RODBC
library(RODBC);

# Abrimos el Canal de comunicación con la base de datos
canal <- odbcConnect("aceralia","aceralia","aceralia","dim-api3.unirioja.es");

# Obtenemos mediante SQL las variables de consigna
VZona1 <- sqlQuery(canal,"SELECT INSTANTE,THF1VALCNG as THC1,THF2VALCNG as
THC2,THF3VALCNG as THC3,THF4VALCNG as THC4,THF5VALCNG as THC5,THF6VALCNG as
THC6,THF7VALCNG as THC7,THF8VALCNG as THC8,CODBOBINA FROM MODELO_JUN");
```

Figura 105. Conexión en R a la base de datos mediante ODBC.

Para comenzar el análisis, es conveniente "echar un vistazo" a los datos con algunas de las herramientas de visualización multivariante disponibles[13], como por ejemplo: gráficos de cajas o *box plots*, histogramas, rectas cuantil-cuartil, etc.

La función *pairs* del entorno de programación estadístico *R*, es otra de las herramientas más útiles para realizar una exploración inicial. Esta función permite comparar varias variables entre si mostrándonos en un mismo gráfico, y con una adecuada programación el histograma, la correlación y la distribución de los puntos emparejados en variables.

A continuación se explicarán los estudios realizados en los datos de la primera base de datos suministrada y completada con la segunda base de datos.

[13] Inicialmente se utilizará el programa estadístico R con licencia GNU [RPR02] que dispone de un sin fin de herramientas para análisis y manipulación de datos.

5.5.3 ESTUDIO EXPLORATORIO DE LAS VARIABLES DE TEMPERATURA (*THF*) DE LA ZONA DE CALENTAMIENTO

Primeramente se estudia la relación existente entre las variables consigna de temperatura de la zona de calentamiento para contrastar los resultados obtenidos con los teóricos del modelo. Para ello, se analiza inicialmente los datos de procesado de bobinas de un mes completo que corresponden con la primera base de datos suministrada.

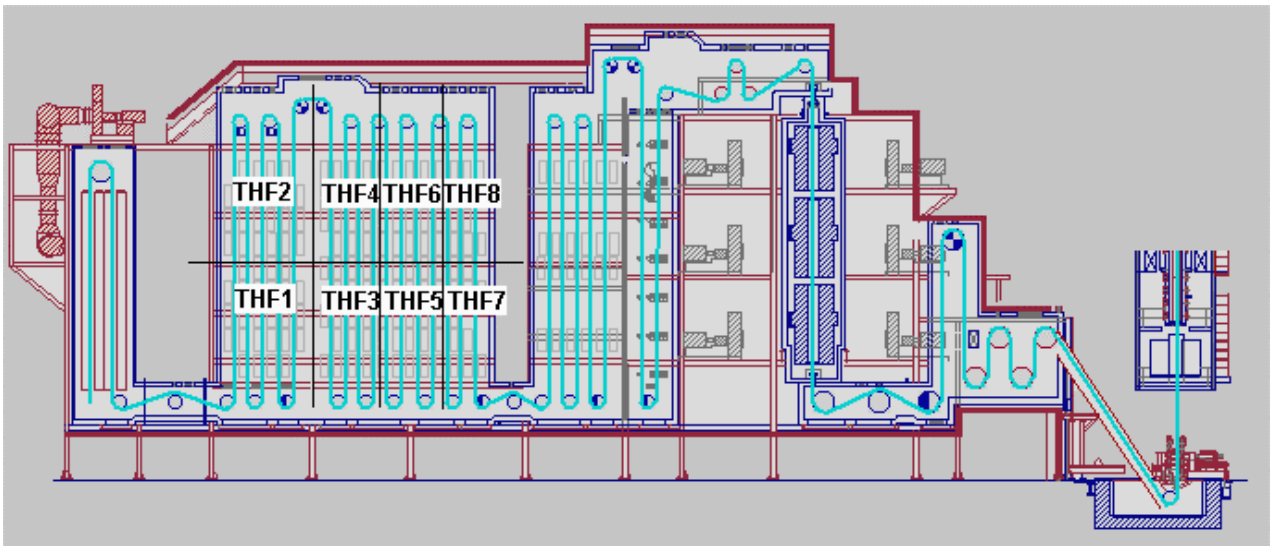


Figura 106. Variables de temperatura en las subzonas de la zona de calentamiento.

5.5.3.1 PRIMERA SELECCIÓN DE LAS VARIABLES A UTILIZAR

Dentro de las variables *THFxVALMED*, *THFxVALMIN* y *THFxVALMAX* (valores[14] de temperatura medio, mínimo y máximo de la zona *x*), advertimos rápidamente que algún fallo existe en la captura de las mismas, ya que en muchos instantes, la temperatura media (*THFxVALMED)* no está entre los valores mínimo y máximo, como cabía suponer (ver Figura 107).

Detectado este error y hablado con el personal de la empresa, nos corroboraron que **el dato más adecuado era el de la media de temperatura**, y que los mínimos y máximos no se utilizaban ya que no eran muy fiables. Por lo tanto, para los estudios posteriores, se prescinde del uso de las variables *THFxVALMIN* y *THFxVALMAX*.

Otra de las variables que se va a utilizar en el estudio, es la temperatura de consigna para cada zona (*THFxVALCNG).*

[14] Las temperaturas de cada una de estas zonas son establecidas mediante radiadores de gas.

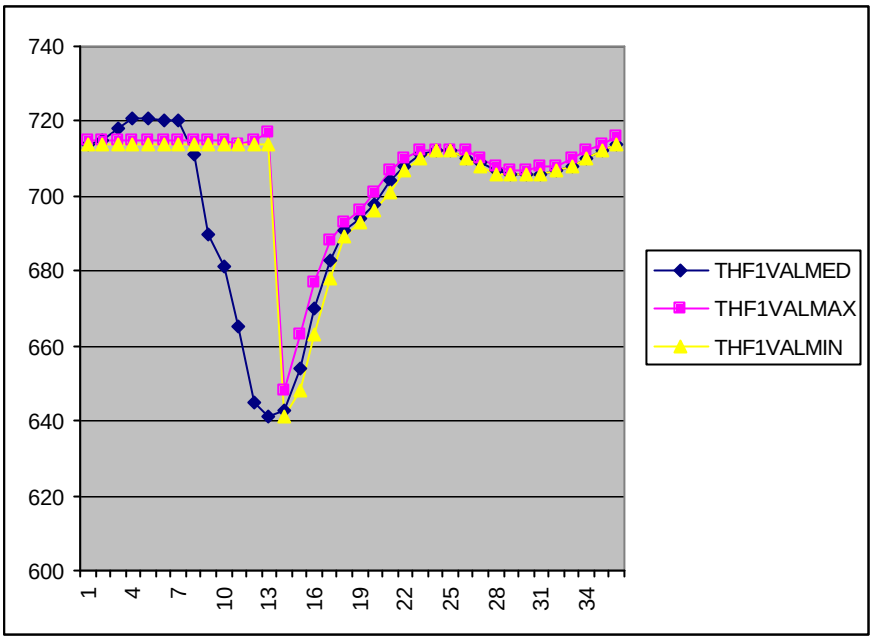


Figura 107. Evolución de las temperaturas de las subzonas media, máxima y mínima de una bobina.

Comenzamos el análisis, visualizando dos variables de consigna en dos de las zonas de calentamiento.

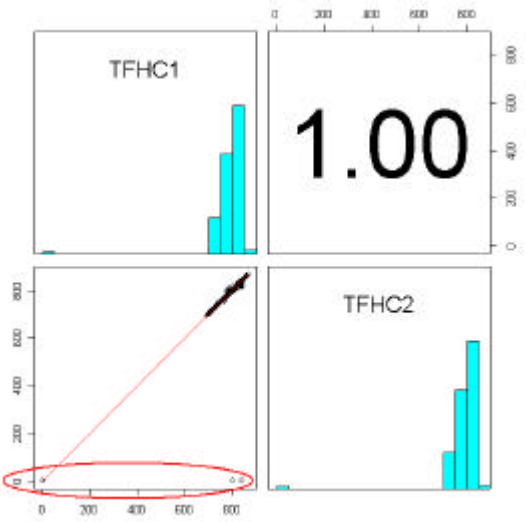


Figura 108. Comparativa entre las temperaturas de consigna THF1VALCNG y THF2VALCNG.

Lo primero que se observa, es que existen unos espurios claramente identificables que impiden observar correctamente la distribución de los demás puntos. Se procede por lo tanto, a la eliminación de todas las observaciones que tengan estos puntos. También se comprobarán los puntos eliminados.

```
# Realizamos un Attachment de las variables de VZona1
attach(VZona1)

# Obtenemos los valores buenos si las temperaturas de consigna son mayores de
100
VZ1Buenos <- VZona1[TFHC1>100 & TFHC2>100 & TFHC3>100 & TFHC4>100 & TFHC5>100 &
TFHC6>100 & TFHC7>100 & TFHC8>100,]
dim(VZona1)
      [1] 30753    10
dim(VZ1Buenos)
      [1] 30447    10

# Obtenemos las observaciones que tienen algún valor de consigna menor de 100
VZ1Malos <- VZona1[!(TFHC1>100 & TFHC2>100 & TFHC3>100 & TFHC4>100 & TFHC5>100 &
TFHC6>100 & TFHC7>100 & TFHC8>100),]

dim(VZ1Malos)
      [1] 306  10
```

Figura 109. Programa en R donde se eliminan los puntos espurios que tengan una variable de consigna menor de 100 grados.

5.5.3.2 ANÁLISIS DE LOS ESPURIOS

Después de realizado el filtrado, observamos rápidamente un número inicial de 306 observaciones erróneas de un total de 30.753 lo que nos da un 0,995% de observaciones erróneas en 45 bobinas de 1.712, o lo que es lo mismo, **un 2,62% de bobinas en las que han aparecido problemas en las temperaturas de consigna o reales.**

Para ver el instante en que se producen, visualizamos las temperaturas de consigna máximas para cada bobina ordenándolas en el eje *x* de forma temporal.

```
# Realizamos un attachment de las variables de VZona1
attach(VZona1)
# Obtenemos las posiciones de las variables con observaciones erróneas
ids <- !(TFHC1>100 & TFHC2>100 & TFHC3>100 & TFHC4>100 & TFHC5>100 & TFHC6>100 &
TFHC7>100 & TFHC8>100)
length(ids)
      [1] 30753
# Creamos un vector con 4 (azul) de 30753 valores y rellenamos con 2 (rojo) las
# posiciones erróneas
cc <- rep (4,30753)
cc[ids] <- 2
ColorBobina <- tapply(cc,VZona1$CODBOBINA,min)
# Obtenemos la máxima temperatura de consigna de la subzona 1 por bobina
MAXTEMPCNG <- tapply(VZona1$THF1,VZona1$CODBOBINA,max)
# Obtenemos un vector para los círculos rellenos (malos) y vacíos (buenos)
BobinasTotC <- rep(21,30753)
BobinasTotC[ColorBobina==2]<-19
# Dibujamos la máxima temperatura de cada bobina en azul-vacio (correcta) o
# rojo-lleno(errónea)
plot(MAXTEMPCNG,col=ColorBobina,xlab="BOBINAS",ylab="Máxima Temperatura Consigna
1 THF1VALCNG",pch=BobinasTotC)
```

Figura 110. Programa para visualizar las máximas temperaturas de consigna de cada bobina.

En la Figura 111 podemos observar en puntos rojos-rellenos, como las observaciones erróneas aparecen en varios instantes del mes. Esta gráfica nos permite comprender, **que las observaciones erróneas que se van a analizar, son debidas a efectos esporádicos y que no se producen de forma continua.**

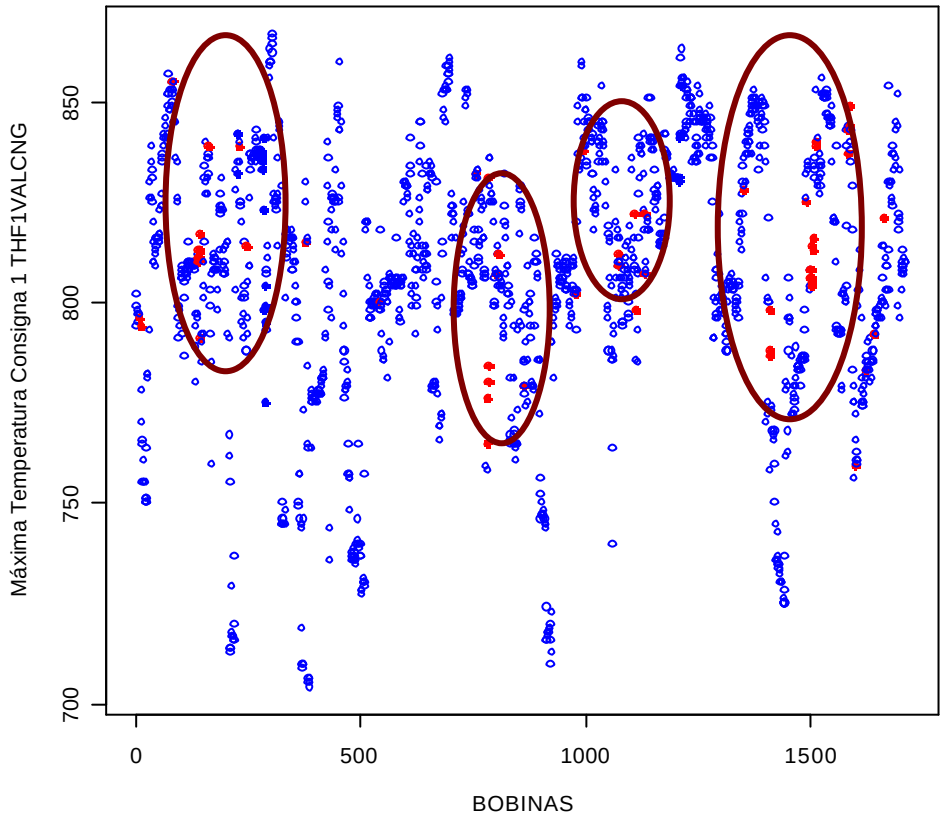


Figura 111. Máximas temperaturas de consigna para cada bobina en orden según el día de procesado. En los círculos rellenos se pueden ver las bobinas en las que aparecen observaciones de temperaturas de consigna o reales del horno defectuosas.

Si visualizamos una muestra de ellas, podemos advertir que en la mayoría, las temperaturas son cero excepto en la temperatura de consigna *THFC6*, esto puede indicar fundamentalmente dos cosas:

- Que las señales de consigna no se han podido almacenar.
- Que el modelo o el software no calcula correctamente algunos valores de consigna.

```
VZ1Malos[1:50,]
     INSTANTE THFC1 THFC2 THFC3 THFC4 THFC5 THFC6 THFC7 THFC8 CODBOBINA
226        19     0     0     0     0     0   850     0     0  11523009
227        20     0     0     0     0     0   850     0     0  11523009
228        21     0     0     0     0     0   850     0     0  11523009
229        22     0     0     0     0     0   850     0     0  11523009
270        19     0     0     0     0     0   850     0     0  11523011
271        20     0     0     0     0     0   850     0     0  11523011
272        21     0     0     0     0     0   850     0     0  11523011
273        22     0     0     0     0     0   850     0     0  11523011
274        23     0     0     0     0     0   850     0     0  11523011
275        24     0     0     0     0     0   850     0     0  11523011
276        25     0     0     0     0     0   850     0     0  11523011
277        26     0     0     0     0     0   850     0     0  11523011
278        27     0     0     0     0     0   850     0     0  11523011
279        28     0     0     0     0     0   850     0     0  11523011
280        29     0     0     0     0     0   853     0     0  11523011
281        30     0     0     0     0     0   795     0     0  11523011
282        31     0     0     0     0     0   780     0     0  11523011
283        32     0     0     0     0     0   781     0     0  11523011
284        33     0     0     0     0     0   781     0     0  11523011
285        34     0     0     0     0     0   781     0     0  11523011
286        35     0     0     0     0     0   781     0     0  11523011
287        36     0     0     0     0     0   782     0     0  11523011
288        37     0     0     0     0     0   782     0     0  11523011
289        38     0     0     0     0     0   782     0     0  11523011
1205       14     0     0     0     0     0   899     0     0  11523081
1206       15     0     0     0     0     0   899     0     0  11523081
1207       16     0     0     0     0     0   899     0     0  11523081
1208       17     0     0     0     0     0   880     0     0  11523081
1209       18     0     0     0     0     0   835     0     0  11523081
1210       19     0     0     0     0     0   835     0     0  11523081
1211       20     0     0     0     0     0   835     0     0  11523081
```

Figura 112. Muestra representativa de las observaciones erróneas de las temperaturas de consigna.

En las figuras siguientes podemos contrastar claramente que la muestra representativa de la Figura 112 se cumple en toda la matriz de observaciones erróneas. Esto queda claramente indicado en el resumen de la Figura 113 donde obtenemos los valores medios iguales a cero de todas las variables exceptuando la *TFHC6*. También se puede apreciar visualmente en la Figura 114 la distribución de los pares de puntos de las variables estudiadas.

```
summary(VZ1Malos)
    INSTANTE         THFC1            THFC2            THFC3
 Min.   : 1.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
 1st Qu.:20.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00
 Median :28.00   Median :  0.00   Median :  0.00   Median :  0.00
 Mean   :25.25   Mean   : 32.44   Mean   : 24.35   Mean   : 28.21
 3rd Qu.:31.00   3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.00
 Max.   :42.00   Max.   :842.00   Max.   :842.00   Max.   :875.00
     THFC4            THFC5            THFC6            THFC7
 Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :  0.00
 1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:822.0   1st Qu.:  0.00
 Median :  0.00   Median :  0.00   Median :846.0   Median :  0.00
 Mean   : 25.27   Mean   : 15.15   Mean   :815.5   Mean   : 26.32
 3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:860.0   3rd Qu.:  0.00
 Max.   :875.00   Max.   :900.00   Max.   :900.0   Max.   :900.00
     THFC8          CODBOBINA
 Min.   :  0.00   Min.   :11523009
 1st Qu.:  0.00   1st Qu.:11533061
 Median :  0.00   Median :11643005
 Mean   : 26.32   Mean   :      NA
 3rd Qu.:  0.00   3rd Qu.:11763020
 Max.   :900.00   Max.   :11813048
```

Figura 113. Resumen de las observaciones eliminadas.

Otro aspecto realmente interesante podemos apreciarlo si comparamos los instantes máximos en los que las temperaturas de consigna se hacen cero para cada bobina comparándolos con los datos obtenidos. Tal y como podemos observar en las figuras siguientes se llega a la conclusión de que **las consignas erróneas se producen y continúan generalmente hasta el final de cada bobina ya que los instantes finales de éstas coinciden con los de cada bobina.** Esto induce a pensar que, **una vez producido el error, el sistema no trabaja correctamente durante el tratamiento de una bobina, y que solamente "vuelve a la normalidad" cuando calcula las consignas para la siguiente bobina[15].** De esta forma, si se produce un error en el sistema de control, éste generará unas consignas equivocadas durante todo el tratamiento de la bobina y no será corregido hasta la siguiente.

Al parecer, **este tipo de error parece que puede localizarse en el sistema de monitorización que se encarga de llamar al modelo al comienzo de cada bobina, de monitorizar los datos y de almacenarlos en la base de datos.**

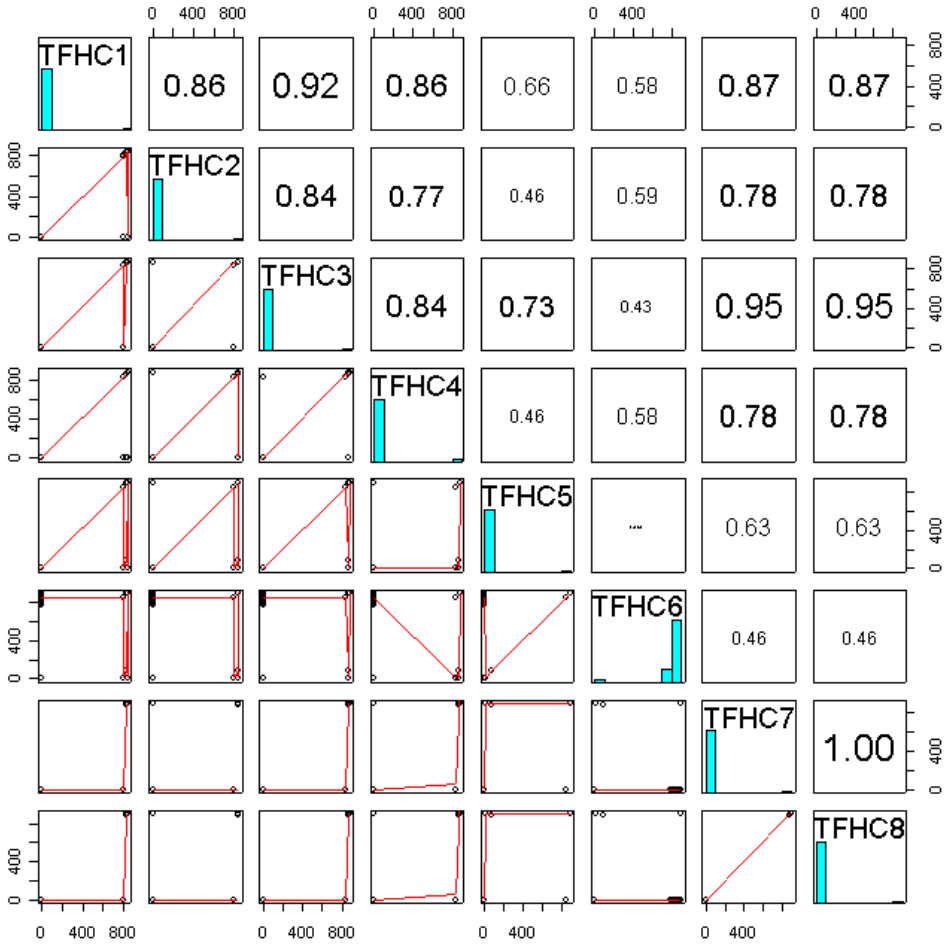


Figura 114. Gráfico comparativo entre variables de consigna erróneas.

```
# Obtenemos el instante mayor de cada bobina con observaciones erróneas
tapply(VZ1Malos[,1],VZ1Malos$CODBOBINA,max)
11523009 11523011 11523081 11533057 11533058 11533059 11533060 11533061
      22       38       20       38       39       35       34       36
11533062 11543017 11553010 11553026 11573032 11603013 11623059 11633012
      39       21        3        6        1       22        7       30
11633013 11633014 11633015 11633016 11633030 11643005 11643059 11673001
      30       30       30       28       24       21       23       24
11673019 11683026 11683027 11693007 11693008 11693029 11693030 11713075
      13        3        3       28       30       22       20       10
11723053 11723054 11723055 11743031 11763017 11763018 11763019 11763020
       4        4        4       28       42       33       33       33
11763021 11763022 11763023 11763024 11763025 11763078 11763079 11803031
      28       34       34       34       31        3        3        5
11803038 11803040 11803041 11803056 11803081 11813019 11813048
      11       14        1       35       20       21       37
```

Figura 115. Máximo instante en que las temperaturas de consigna se hacen cero.

También se puede observar en la evolución de las variables de temperatura de consigna y reales de alguna de las bobinas con observaciones defectuosas, cómo el valor a cero de las consignas se produce después de un error generado en la lectura de las temperaturas reales.

INSTANTE	THF1VALMED	THF1VALCNG	THF2VALMED	THF2VALCNG	CODBOBINA
19	803	805	803	805	11533057
20	802	805	802	805	11533057
21	802	805	803	805	11533057
22	-1	805	-1	805	11533057
23	-1	805	-1	805	11533057
24	-1	806	-1	806	11533057
25	-1	806	-1	806	11533057
26	-1	806	-1	806	11533057
27	-1	806	-1	806	11533057
28	-1	806	-1	806	11533057
29	805	0	805	0	11533057
30	806	0	806	0	11533057
31	808	0	807	0	11533057
32	808	0	808	0	11533057
33	809	0	808	0	11533057
34	810	0	809	0	11533057
35	809	0	808	0	11533057
36	808	0	808	0	11533057
37	808	0	807	0	11533057
38	807	0	807	0	11533057

Tabla 19. Evolución de las variables de consigna y reales de una bobina.

Como se ve en la Tabla 19, un error producido en la lectura de las temperaturas de los instantes 22 a 28, produce unos valores de consigna erróneos en instantes posteriores. En este ejemplo, **se podría llegar a la conclusión de que el sistema de adquisición y almacenamiento de las temperaturas reales ha fallado temporalmente, ya que todas las lecturas de las variables *THFxVALMED* han sido erróneas**.

5.5.3.3 ESTUDIO DEL COMPORTAMIENTO DE LAS BOBINAS CON OBSERVACIONES ERRÓNEAS

A continuación, se estudiará la evolución en el tiempo de las señales en todas las bobinas con observaciones erróneas.

```
unique(VZ1Malos$CODBOBINA)
 [1] 11523009 11523011 11523081 11533057 11533058 11533059 11533060 11533061
 [9] 11533062 11543017 11553010 11553026 11573032 11603013 11623059 11633012
[17] 11633013 11633014 11633015 11633016 11633030 11643005 11643059 11673001
[25] 11673019 11683026 11683027 11693007 11693008 11693029 11693030 11713075
[33] 11723053 11723054 11723055 11743031 11763017 11763018 11763019 11763020
[41] 11763021 11763022 11763023 11763024 11763025 11763078 11763079 11803031
[49] 11803038 11803040 11803041 11803056 11803081 11813019 11813048
```

Figura 116. Obtención de las bobinas con observaciones erróneas.

En las tablas siguientes, se muestra el comportamiento observado en cada una de las bobinas estudiadas.

Código Bobina	Patrón de Comportamiento	Observaciones
11523009	A	4 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11523011	A	20 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11523081	B	5 Temperaturas Medias a –1 y después 7 Últimas Temperaturas de Consigna a Cero
11533057	B	7 Temperaturas Medias a –1 y después 10 Últimas Temperaturas de Consigna a Cero
11533058	A	15 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11533059	A	9 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11533060	A	8 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11533061	A	10 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11533062	A	19 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11543017	C	2 Temperaturas de Consigna (Algunas THF) a Cero
11553010	C	2 Observaciones de Temperaturas de Consigna (Algunas THF) a Cero
11553026	C	2 Observaciones de Temperaturas de Consigna (Algunas THF) a Cero
11573032	C	Pocas Observaciones de Temperaturas de Consigna (Algunas THF) a Cero
11603013	A	7 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11633012	B	23 Temperaturas Medias a –1 y después 4 Últimas Temperaturas de Consigna a Cero
11633013	A	7 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11633014	A	4 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11633015	A	4 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11633016	A	3 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11633030	A	3 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11643005	A	12 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11643059	A	6 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11673001	B	1 Temperaturas Medias a –1 y después 1 Últimas Temperaturas De Consigna a Cero

Tabla 20. Patrones de comportamiento detectado en las bobinas con observaciones erróneas.

Código Bobina	Patrón de Comportamiento	Observaciones
11673019	B	2 Temperaturas Medias a –1 y después 2 Últimas Temperaturas de Consigna a Cero
11683026	B	4 Temperaturas Medias a –1 y después 4 Últimas Temperaturas de Consigna a Cero
11683027	B	4 Temperaturas Medias a –1 y después 4 Últimas Temperaturas de Consigna a Cero
11693007	B	4 Temperaturas Medias a –1 y después 4 Últimas Temperaturas de Consigna a Cero
11693008	B	5 Temperaturas Medias a –1 y después 7 Últimas Temperaturas de Consigna a Cero
11693029	A	7 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11693030	A	4 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11713075	A	2 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11723053	C	Pocas Observaciones de Temperaturas de Consigna (Algunas THF) a Cero
11723054	C	Pocas Observaciones de Temperaturas de Consigna (Algunas THF) a Cero
11723055	C	Pocas Observaciones de Temperaturas de Consigna (Algunas THF) a Cero
11743031	A	2 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11763017	A	20 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11763018	A	6 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11763019	A	5 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11763020	A	6 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11803040	A	7 Últimas Temperaturas de Consigna (THF1 a THF8) a Cero
11803041	C	Pocas Observaciones de Temperaturas de Consigna (Algunas THF) a Diez
11803056	B	34 Temperaturas Medias a –1 y después 1 Últimas Temperaturas de Consigna a Cero
11813019	B	2 Temperaturas Medias a –1 y después 4 Últimas Temperaturas de Consigna a Cero
11813048	B	3 Temperaturas Medias a –1 y después 16 Últimas Temperaturas de Consigna a Cero

Tabla 21. Patrones de comportamiento detectado en las bobinas con observaciones erróneas (continuación).

Explicación de los Patrones Observados

Como se puede observar en las tablas anteriores, se han detectado tres tipos fundamentales de patrones de comportamiento en las temperaturas de las bobinas con observaciones defectuosas. Estos se han clasificado en:

1. Patrón Tipo A Corresponde al patrón que más aparece y que además resulta ser, junto con el tipo B, el más problemático (aparece en 25 de las 1.712 bobinas, es decir, un **1,46 % de las bobinas analizadas**). Las temperaturas de consigna se ponen repentinamente a cero y continúan hasta el final de cada bobina. En la Tabla 22 y en la Tabla 23 se pueden ver ejemplos de varias de las bobinas encontradas.

Curiosamente, los valores de consigna se ponen a cero y no se recuperan hasta que se inicia el cálculo para otra bobina. Esto claramente descarta el posible ruido o fallo aleatorio e **induce a pensar que es debido a un error del software que calcula las mismas o en el sistema que almacena los valores de consigna.** Se descartan posibles fallos en el modelo matemático, pues éste solamente trabaja en el comienzo de cada bobina.

Comparando este tipo de error con el siguiente (Tipo B), se deduce que **el sistema no guarda o trata correctamente la información, curiosamente cuando se producen valores de temperatura leídos a -1**.

INSTANTE	CODBOBINA	VELHFVALMIN	THF1VALMED	THF1VALCNG	TMPP1VALMED	TMPP2VALMED
14	11523009	32000	795	795	218	823
15	11523009	32000	795	795	217	823
16	11523009	32000	795	795	217	823
17	11523009	32000	794	795	216	822
18	11523009	32000	794	795	217	822
19	11523009	0	794	0	220	821
20	11523009	0	795	0	221	821
21	11523009	0	795	0	220	821
22	11523009	0	796	0	219	821

Tabla 22. Bobina con error tipo A.

Efectivamente, si observamos la Tabla 22 podemos apreciar que **el error en las temperaturas de consigna es debido a que se detecta un valor cero en la velocidad de la banda (*VELHFVALMIN*) y este mismo error, se arrastra hasta el final de la bobina.**

INSTANTE	THF1VALMED	THF1VALCNG	THF2VALMED	THF2VALCNG	THF3VALMED	THF3VALCNG
11	788	790	788	790	820	820
12	788	790	788	790	820	820
13	788	790	788	790	819	820
14	789	790	788	790	819	820
15	790	790	789	790	819	820
16	790	790	789	790	819	820
17	791	790	790	790	819	820
18	791	790	790	790	819	820
19	791	0	790	0	819	0
20	791	0	790	0	820	0
21	790	0	791	0	820	0
22	790	0	791	0	820	0
23	790	0	791	0	820	0
24	790	0	791	0	820	0
25	789	0	790	0	820	0
26	788	0	789	0	820	0
27	788	0	788	0	820	0
28	787	0	788	0	820	0
29	787	0	787	0	819	0
30	785	0	786	0	818	0
31	783	0	783	0	815	0
32	781	0	781	0	812	0

Tabla 23. Evolución de las temperaturas de consigna de la bobina 11523011 (patrón A).

Probablemente, este error se produce cuando el dispositivo que mide la velocidad de la banda **deja de funcionar o no captura correctamente su valor, o cuando el sistema que almacena los datos correspondientes a esa variable no actúa según lo convenido.**

Inicialmente se llegó a la siguiente conclusión:

"Si el número de instantes en que las consignas están a cero es considerable, la temperatura del horno en cada zona comienza a bajar peligrosamente. ***Lo que puede afectar claramente a las propiedades mecánicas de esa bobina y de las posteriores, ya que la inercia del horno es elevada y por lo tanto, cuando comience el tratamiento de la siguiente bobina, pasará un tiempo hasta que la temperatura del horno alcance su valor óptimo.***

Por otro lado, la misma inercia del horno impide que, patrones de error tipo A con pocos instantes de valores de consigna a cero puedan producir bajadas significativas en la temperatura del horno. Así se advierte en las temperaturas el horno de todas aquellas bobinas observadas correspondientes al patrón tipo A pero con pocas observaciones erróneas, ***lo que significa que la magnitud del efecto de este tipo de error y el siguiente (el tipo B) dependerá del instante (dentro de cada bobina) en que se produzca.".***

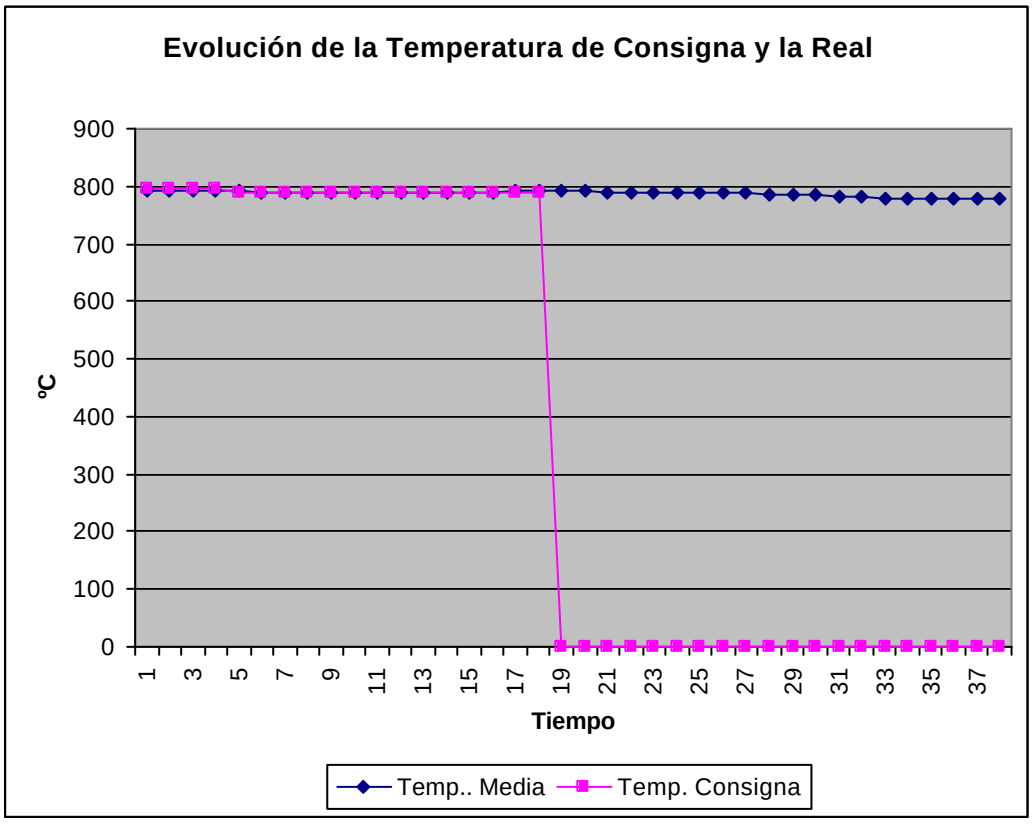


Figura 117. Evolución de las temperaturas de consigna TFH1CNG y real TFH1MED de la zona de calentamiento 1 del horno en la bobina 11523011 (patrón de comportamiento A).

Pero posteriormente, consultados los expertos y analizados más casos, fue descartada, ya que se llegó a la conclusión de que el fallo solamente era de adquisición. En el caso de la Figura 117 se demostró que la temperatura de la banda bajaba debido a que en las siguientes bobinas la temperatura objetivo era menor.

2. Patrón Tipo B: Este tipo de error se detecta en aquellos casos en que la lectura de las temperaturas *THFxVALMED* en la parte de calentamiento es incorrecta (valor a "-1") y que produce posteriormente unos valores de consigna *THFxVALCNG* igual a cero en los instantes siguientes (aparece en 12 de las 1.712 bobinas analizadas, es decir, un **0,70 % de las bobinas analizadas**) hasta el final de la bobina.

Este tipo de error inicialmente lo achacamos a errores en la medida de las temperaturas producidos por el autómata, problemas en las comunicaciones o en los elementos de adquisición de las señales. Como éste se producía generalmente en casi todas las medidas de temperatura de las ocho zonas de la fase de calentamiento del horno a la vez, se consideró que el problema podía ser debido a un efecto esporádico de comunicación o de inicialización del dispositivo de adquisición, mas que del propio termopar o voltímetro de cada zona, ya que muchos sistemas de adquisición cuando se resetean o entran en ejecución en "modo error" producen una salida en "alta impedancia" lo que explicaría el valor a "-1" es decir "0xFFFF" en hexadecimal, típico en algunos autómatas cuando la entrada analógica del mismo está sin alimentar o en "alta impedancia".

Pero cuando se presentó esta conclusión al personal experto de la empresa, nos explicaron que esos valores a "-1" los ponía el programa de monitorización (que llama al modelo matemático y almacena los datos en la Base de Datos) para inicializar las tablas y que seguramente lo que había pasado es que había existido algún error en el programa de monitorización y éste no había actualizado esos datos.

3. Patrón Tipo C: Se han detectado esporádicas observaciones de temperatura de consigna a cero (*THFxVALCNG*) en algunos valores de consigna (aparece en 8 de las 1.712 bobinas analizadas, es decir, un **0,46 % de las bobinas analizadas**). Este tipo de patrón se puede achacar a fallos en la escritura de los datos, o a un pequeño ruido que afecta a algún sensor y que se refleja en el cálculo de las variables de consigna, o a errores en la adquisición.

Este tipo de error, al producirse solamente en uno o dos instantes dentro de cada una de las 8 bobinas, **se ve claramente que no afecta en las temperaturas del horno** y por lo que se considera un fallo leve y ocasional.

Es probable, que las medidas de mejora del sistema de monitorización necesarias para corregir los errores tipo A y B, solventen algunos de los producidos en el tipo C, aunque seguramente los errores achacados al tipo C, sean debidos a pequeños defectos de ruido en las comunicaciones.

CONCLUSIONES INICIALES

Por lo tanto, y después de analizados los tres tipos de errores encontrados, podemos llegar a las siguientes conclusiones.

- Los errores se han producido en varios momentos del mes tal y como lo muestra la Figura 111. Esta gráfica nos permite comprender, **que las observaciones erróneas son debidas a efectos esporádicos y no se producen de forma continuada.**
- Observamos un número inicial de 306 observaciones erróneas de un total de 30.753, es decir, un 0,995% de observaciones erróneas. De 1.712 bobinas estudiadas se han encontrado errores en 45 de ellas, es decir, **un 2,62% donde han aparecido problemas en las temperaturas de consigna o reales.**
- **Se considera que, para evitar los errores tipo A y B, debe analizarse y mejorarse el sistema de monitorización encargado del almacenamiento de las variables en la base de datos y de la gestión de las variables de consigna. Sería interesante depurar el software, simulando "los comportamientos anómalos" detectados para "descubrir" posibles fallos.**
- Resultaría conveniente revisar todo el sistema de adquisición y almacenamiento de las variables más significativas del proceso buscando la eliminación de ruido en las comunicaciones. Probablemente el error tipo C se solucionará con estas medidas.
- Falta una variable que nos indique si sistema está trabajando en modo manual o en modo automático. Es decir, se determina la necesidad de tener una variable que indique **si el modelo matemático está trabajando o no.**

5.5.3.4 ANÁLISIS DE LA RELACIÓN ENTRE LAS TEMPERATURAS REALES Y DE CONSIGNA DE LA ZONA DE CALENTAMIENTO

Una vez, hemos analizado los espurios, detectado las posibles causas que los generaron y buscado modos de solucionar esos fallos, procedemos a analizar la relación existente entre las variables de temperatura de las subzonas de la zona de calentamiento del horno (*TFHxVALMED)* y las de consigna (*TFHxVALCNG).*

Para una mejor manipulación en la programación, se han reducido los nombres de las variables de la siguiente forma:

- VELHFVALMIN a VELS (velocidad media consigna en la banda).
- TMPPxVALMED a TMPPx (temperatura de la banda siendo *x* un número indicando el pirómetro que lo lee (pirómetro 1 corresponde con la temperatura de la banda a la entrada de la zona de calentamiento, pirómetro 2 indica la temperatura de la banda a la salida de la zona de calentamiento y el pirómetro 3 a la salida de la zona de mantenimiento).
- THFxVALMED a THFx (temperaturas reales de las zonas *x* en la parte de calentamiento del horno).
- THFxVALCNG a THCx (temperaturas de consigna de las zonas *x* en la parte de calentamiento del horno).

Lo primero que hacemos, es eliminar todas las bobinas con valores defectuosos.

```
# Cargamos de la base de datos las variables
library(RODBC);
canal <- odbcConnect("aceralia","aceralia","aceralia","dim-api3.unirioja.es");
VZona1 <- sqlQuery(canal,"SELECT INSTANTE,VELHFVALMIN as VELS, TMPP1VALMED as
TMPP1, TMPP2VALMED as TMPP2, TMPP3VALMED as TMPP3, THF1VALCNG as THC1,THF2VALCNG
as THC2,THF3VALCNG as THC3,THF4VALCNG as THC4,THF5VALCNG as THC5,THF6VALCNG as
THC6,THF7VALCNG as THC7,THF8VALCNG as THC8, THF1VALMED as THF1,THF2VALMED as
THF2,THF3VALMED as THF3,THF4VALMED as THF4,THF5VALMED as THF5,THF6VALMED as
THF6,THF7VALMED as THF7,THF8VALMED as THF8,CODBOBINA FROM MODELO_JUN");
dim(VZona1)
[1] 30753    22


# Obtenemos las variables con observaciones defectuosas y correctas
attach(VZona1)
VZonaMalos <- VZona1[!(THC1>100 & THC2>100 & THC3>100 & THC4>100 & THC5>100 &
THC6>100 & THC7>100 & THC8>100 & THF1>100 & THF2>100 & THF3>100 & THF4>100 &
THF5>100 & THF6>100 & THF7>100 & THF8>100),]
VZonaBuenos <- VZona1[(THC1>100 & THC2>100 & THC3>100 & THC4>100 & THC5>100 &
THC6>100 & THC7>100 & THC8>100 & THF1>100 & THF2>100 & THF3>100 & THF4>100 &
THF5>100 & THF6>100 & THF7>100 & THF8>100),]
dim(VZonaMalos)
[1] 423  22
```

```
dim(VZonaBuenos)
[1] 30330    22

# Realizamos un vistazo general
summary(VZonaBuenos)
    INSTANTE          VELS           TMPP1           TMPP2
 Min.   : 1.00   Min.   :32000   Min.   : -1.0   Min.   : -1.0
 1st Qu.: 5.00   1st Qu.:32000   1st Qu.:216.0   1st Qu.:798.0
 Median :10.00   Median :32000   Median :226.0   Median :820.0
 Mean   :11.61   Mean   :32000   Mean   :224.9   Mean   :806.6
 3rd Qu.:17.00   3rd Qu.:32000   3rd Qu.:236.0   3rd Qu.:825.0
 Max.   :49.00   Max.   :32000   Max.   :348.0   Max.   :885.0
     TMPP3            THC1            THC2            THC3
 Min.   : -1.0   Min.   :694.0   Min.   :695.0   Min.   :723.0
 1st Qu.:796.0   1st Qu.:781.0   1st Qu.:782.0   1st Qu.:811.0
 Median :817.0   Median :802.0   Median :802.0   Median :832.0
 Mean   :802.6   Mean   :796.7   Mean   :796.7   Mean   :826.7
 3rd Qu.:825.0   3rd Qu.:817.0   3rd Qu.:817.0   3rd Qu.:848.0
 Max.   :878.0   Max.   :865.0   Max.   :865.0   Max.   :895.0
      THC4            THC5            THC6            THC7
 Min.   :723.0   Min.   :742.0   Min.   :742.0   Min.   :742.0
 1st Qu.:811.0   1st Qu.:830.0   1st Qu.:830.0   1st Qu.:830.0
 Median :832.0   Median :851.0   Median :851.0   Median :851.0
 Mean   :826.6   Mean   :845.9   Mean   :845.9   Mean   :845.9
 3rd Qu.:848.0   3rd Qu.:868.0   3rd Qu.:868.0   3rd Qu.:868.0
 Max.   :895.0   Max.   :920.0   Max.   :920.0   Max.   :920.0
      THC8            THF1           THF2           THF3
 Min.   :742.0   Min.   :673   Min.   :676   Min.   :716.0
 1st Qu.:830.0   1st Qu.:781   1st Qu.:781   1st Qu.:811.0
 Median :851.0   Median :802   Median :802   Median :832.0
 Mean   :845.9   Mean   :796   Mean   :796   Mean   :826.1
 3rd Qu.:868.0   3rd Qu.:817   3rd Qu.:817   3rd Qu.:848.0
 Max.   :920.0   Max.   :867   Max.   :867   Max.   :896.0
      THF4           THF5            THF6            THF7
 Min.   :705   Min.   :732.0   Min.   :730.0   Min.   :735.0
 1st Qu.:811   1st Qu.:830.0   1st Qu.:830.0   1st Qu.:830.0
 Median :832   Median :851.0   Median :851.0   Median :850.0
 Mean   :826   Mean   :845.3   Mean   :845.5   Mean   :845.4
 3rd Qu.:848   3rd Qu.:868.0   3rd Qu.:868.0   3rd Qu.:867.0
 Max.   :896   Max.   :921.0   Max.   :921.0   Max.   :921.0
      THF8          CODBOBINA
 Min.   :734.0   Min.   :11523001
 1st Qu.:830.0   1st Qu.:11583060
 Median :850.0   Median :11643067
 Mean   :845.3   Mean   :      NA
 3rd Qu.:867.0   3rd Qu.:11713010
 Max.   :921.0   Max.   :11843038
```

Figura 118. Programa que realiza el filtrado de los puntos defectuosos.

RELACIÓN ENTRE LAS TEMPERATURAS DE CONSIGNA

Usamos el comando *pairs* para visualizar la relación entre las temperaturas de consigna.

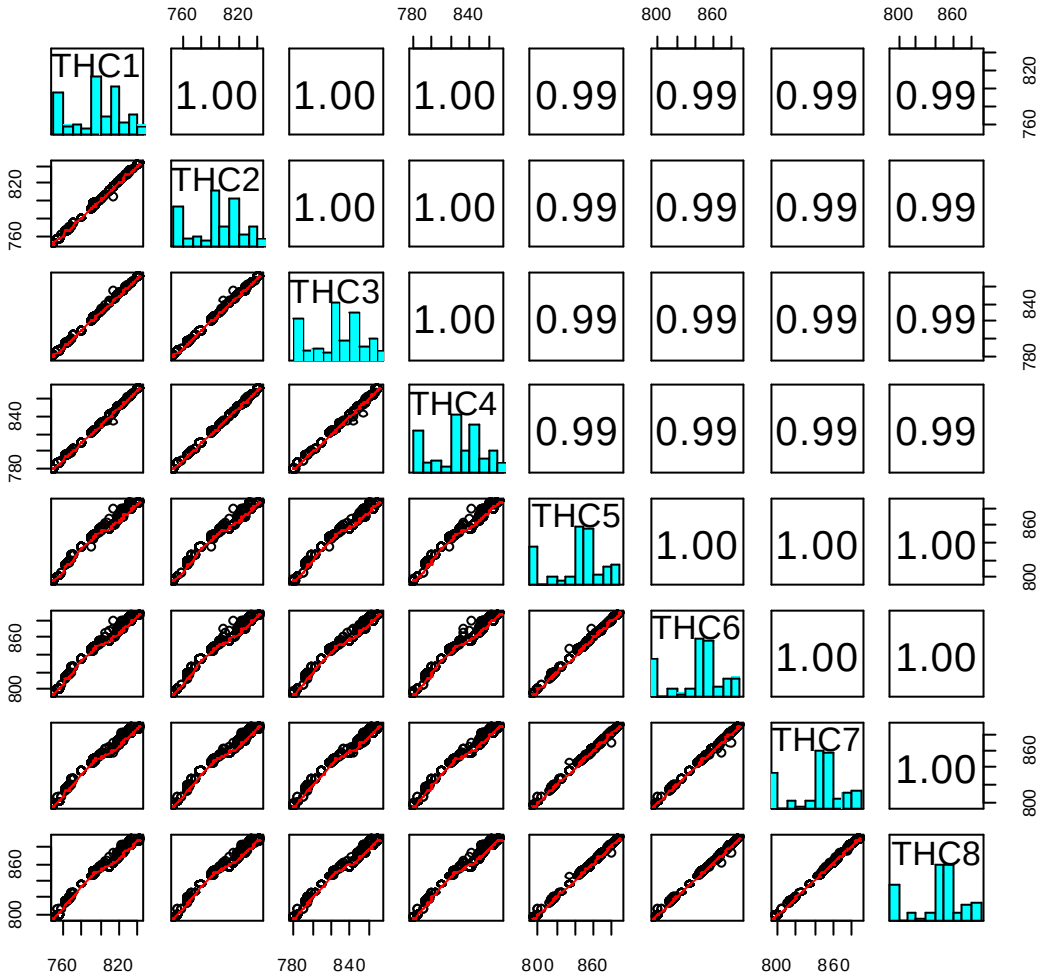


Figura 119. Relación entre las diferentes variables de temperatura de consigna de la zona de calentamiento del horno (TFHxVALCNG llamadas *THCx*).

En la figura anterior, podemos observar que la correlación entre las variables es uno o prácticamente uno, tal y como lo dice el modelo [DRE98]. Éste especifica el cálculo de las temperaturas de consigna de la siguiente forma:

TFH1VALCNG=TFH2VALCNG=TF (temperatura calculada por el modelo)
TFH3VALCNG=TFH4VALCNG=TF+ST1
TFH5VALCNG=TFH6VALCNG=TF+ST2 (5.1)
TFH7VALCNG=TFH8VALCNG=TF+ST3

Siendo *TF* la temperatura objetivo calculada por el modelo matemático, *ST1* la diferencia de temperaturas entre las zonas 3-4 y las 1-2, *ST2* la diferencia de temperaturas entre las zonas 5-6 y 1-2, y *ST3* la diferencia entre las zonas 7-8 y 1-2.

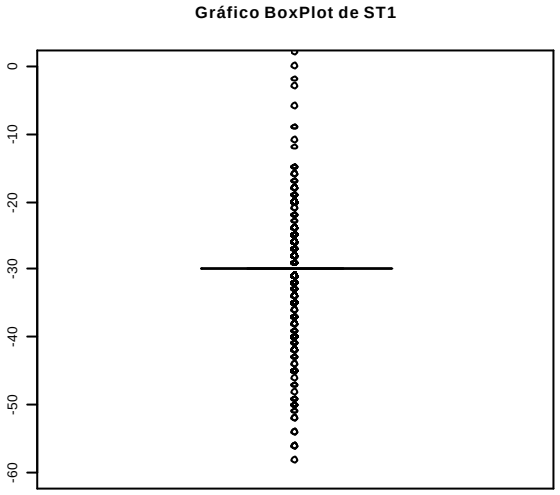


Figura 120. Gráfico de caja de la variable ST1.

A nivel empírico y consultado personal de la empresa, nos confirman que las temperaturas de las ocho zonas "físicas" se agrupan de dos en dos formando cuatro "lógicas", tal y como se muestra en la ecuación (5.1), además nos indican **que los valores más usuales de "steps" iniciales que se suelen usar inicialmente oscilan entre los 30ºC, 50ºC y 50ºC, para *ST1*, *ST2* y *ST3* respectivamente, y que el modelo realmente calcula solamente la incógnita *THF1VALCNG***. Por otro lado, ST2 y ST3 se comprueba que, prácticamente en casi todos los casos, son iguales.

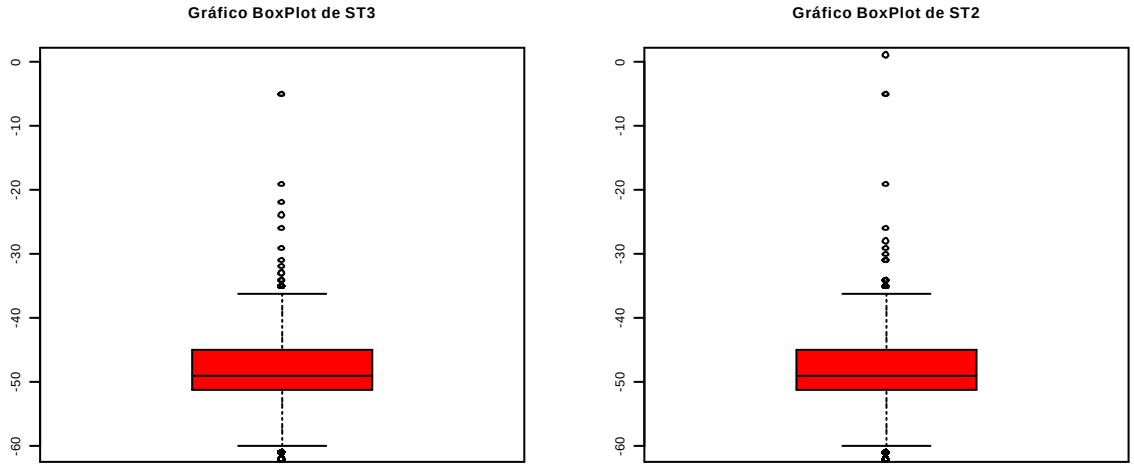


Figura 121. Gráficos de caja de las variables ST2 y ST3.

La elección de esta serie de incrementos de temperatura depende del formato de la banda y surge de una búsqueda de equilibrio entre la eficiencia del horno y un calentamiento progresivo de la banda. Este perfil de incrementos puede ser variado, en el "modo manual", por los operarios cuando, debido a características especiales de la banda, se quiera "forzar" el ciclo térmico.

```
%Calculamos los pasos
STEP1<-THC1-THC3
STEP2<-THC1-THC5
STEP3<-THC1-THC7
boxplot(Pasos[,2],col="red",ylim=c(-60,0),main="Gráfico BoxPlot de ST1")
boxplot(Pasos[,2],col="red",ylim=c(-60,0),main="Gráfico BoxPlot de ST2")
boxplot(Pasos[,3],col="red",ylim=c(-60,0),main="Gráfico BoxPlot de ST3")
table(Pasos[,1])
  -70   -27   -24   -23   -18   -14   -10    -6    -2     0     2     3     6
    4     1     3     2     1     1     1     1     1     1     1     1     1
    9    11    12    15    16    17    18    19    20    21    22    23    24
    2     1     1     5     3     3     6    10   220     2     7     2    12
   25    26    27    28    29    30    31    32    33    34    35    36    37
  633    57    47    41    24 28269   105   121    16     9   369     3    41
   38    39    40    41    42    43    44    45    46    47    48    49    50
    7     5   289     8    10     5     2    70     1     3     1     5     8
   51    52    54    56    58    76    82    86
    2     2     2     3     1     2     1     2

table(Pasos[,2])
-834 -833 -832 -751  -61  -14   -1    5   19   26   28   29   30   31   34   35
   1    1    1    2    2    1    1    2    1    1    1    1    1    3    6   17
  36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51
  12   29   17   55   30   44   84  113   85 7335 2377 2260 2473 2765 3333 1961
  52   53   54   55   56   57   58   59   60   61   62   63   64   65   66   67
1604 1435 1023 2037   57   44  144   87  101   64  157  142  102  129   32   38
  68   69   70   71   72   73   74   75   76   77   78   79   80   81   82   97
  24   19   28   25   34   21    2    2   26   17    7   25    4    1    1    2
  99  100  103  105  106
   1    1    2    2    2

table(Pasos[,3])
-795   -5    5   19   22   24   26   29   31   32   33   34   35   36   37   38
   1    1    2    1    1    1    1    1    1    1    2    3   20   18   28   18
  39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54
  40   25   41   91  126   96 7277 2370 2275 2462 2761 3345 1966 1612 1441 1022
  55   56   57   58   59   60   61   62   63   64   65   66   67   68   69   70
2059   61   49  148   95   98   69  159  141  103  125   31   38   23   19   23
  71   72   73   75   76   77   78   79   80   97   99  100  103  105  106
  25   34   21    2   23   19    6   22    4    2    1    1    2    2    2
```

Figura 122. Programa que calcula y visualiza en diagramas de cajas los pasos.

Si nos fijamos en la Figura 120 podemos ver que los 30ºC de diferencia en la variable *ST1* se mantiene con ese valor en un número elevado de observaciones (28.269). En cambio los pasos 2 y 3 (variables *ST2* y *ST3*) oscilan 20 grados entorno a los 50ºC. **Claramente podemos ver que los**

gráficos de cajas de estas dos variables son muy parecidos, lo que nos invita a deducir que el sistema asigna, generalmente, los mismos valores a estas dos variables.

Curiosamente, advertimos que existen puntos que no cumplen estas igualdades. Podemos visualizar las diferencias entre los pares de variables de consigna siguientes:

DST0=TFH1VALCNG-TFH2VALCNG
DST1=TFH3VALCNG-TFH4VALCNG
DST2=TFH5VALCNG-TFH6VALCNG (5.2)
DST3=TFH7VALCNG-TFH8VALCNG

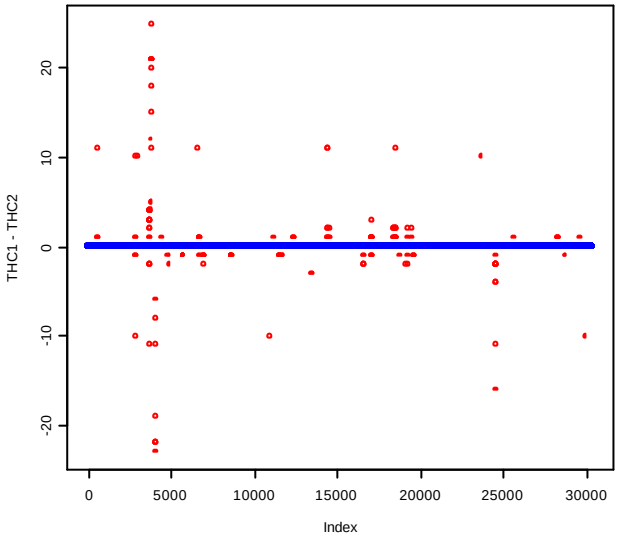


Figura 123. Diferencia entre las variables de consigna THC1 y THC2.

La Figura 123 se repite para los diferentes pares de temperaturas de consigna donde vemos que **la mayoría de los datos son iguales tal y como dice el modelo, pero existen una serie de puntos de consigna que inexplicablemente son distintos.**

Podemos comprobar si existe relación entre las bobinas erróneas del apartado anterior con los puntos con pares de temperaturas diferentes.

```
# Obtenemos datos con observaciones erróneas
VZMalos <- VZona1[!(THC1>100 & THC2>100 & THC3>100 & THC4>100 & THC5>100 &
THC6>100 & THC7>100 & THC8>100),]
dim(VZMalos)
      [1] 306  22

# Obtenemos variables con temperaturas de consigna diferentes entre 1 y 2, ...
VZDiferentes <- VZona1[!(THC1==THC2 & THC3==THC4 & THC5==THC6 & THC7==THC8),]
dim(VZDiferentes)
[1] 765  22

# Obtenemos las bobinas de cada una de las matrices y las comparamos
BobinasMal <- unique(VZMalos$CODBOBINA)
```

```
BobinasDif <- unique(VZDiferentes$CODBOBINA)
for (i in 1:length(BobinasDif))
+ for (j in 1:length(BobinasMal))
+ if (BobinasDif[i]==BobinasMal[j]) print(BobinasDif[i]);
[1] 11523009
[1] 11523011
[1] 11523081
[1] 11533057
[1] 11533058
[1] 11533059
[1] 11533060
[1] 11533061
[1] 11533062
[1] 11543017
[1] 11553010
[1] 11553026
[1] 11573032
[1] 11603013
[1] 11633012
[1] 11633013
[1] 11633014
[1] 11633015
[1] 11633016
[1] 11633030
[1] 11643005
[1] 11643059
[1] 11673001
[1] 11673019
[1] 11683026
[1] 11683027
[1] 11693007
[1] 11693008
[1] 11693029
[1] 11693030
[1] 11713075
[1] 11723053
[1] 11723054
[1] 11723055
[1] 11743031
[1] 11763017
[1] 11763018
[1] 11763019
[1] 11763020
[1] 11763021
[1] 11763022
[1] 11763023
[1] 11763024
[1] 11763025
[1] 11763078
[1] 11763079
[1] 11803031
[1] 11803038
[1] 11803041
[1] 11803056
[1] 11803081
```

Figura 124. Comparación entre las bobinas de erróneas con temperaturas de consigna menor de 100 y distintas entre pares de consigna según el modelo.

Como se puede ver en la Figura 124, todas las bobinas detectadas corresponden a bobinas en las que se han encontrado observaciones defectuosas. Efectivamente, **los valores distintos entre pares de consignas se producen justo después de valores de consigna a cero**. Esto indica claramente, que **el sistema integrado de supervisión y monitorización del horno no almacena correctamente los valores del proceso en la Base de Datos con lo que se pierde la trazabilidad de todo el proceso.**

COMPARACIÓN ENTRE LAS TEMPERATURAS REALES Y DE CONSIGNA

Después de filtrados los casos anteriores, procedemos a comparar los valores de temperaturas de consigna con los reales.

```
# Dibujamos los puntos de consigna y el valor medio leído
plot (DatosBuenos[1:1000,]$THF1VALCNG,col=DatosBuenos[1:1000,]$CODBOBINA)
lines (DatosBuenos[1:1000,]$THF1VALMED,col=DatosBuenos[1:1000,]$CODBOBINA)
```

Figura 125. Programa que representa con puntos los valores de consigna y con líneas los reales.

En la figura siguiente podemos observar los puntos de consigna (con un color diferente para cada bobina) y el comportamiento de las temperaturas de las subzonas (representados con línea continua). Ésta nos muestra claramente **que los controladores que alimentan a los radiadores de gas en cada subzona de la zona de calentamiento del horno, siguen con bastante fidelidad la temperatura de consigna que se les pide por lo que solo se considerarán éstas en posteriores estudios.**

En las Figura 126 y Figura 127 podemos ver que existen bobinas "con fuertes cambios de temperaturas de consigna" donde ésta crece o decrece progresivamente para adecuar la temperatura del horno a las próximas bobinas a tratar. Este aumento o descenso progresivo de temperatura, claramente afectará, hasta que se estabilice la temperatura de tratamiento ideal, a esa bobina y siguientes.

Este tipo de comportamiento, ya considerado en el modelo matemático inicial, debe ser analizado e introducido dentro del estudio y desarrollo del modelo a conseguir. Es por ello, que se propone la creación de una serie de nuevas variables que estudien estas transiciones e indiquen, de alguna forma, cómo discurre la transición de temperatura entre una bobina y otra.

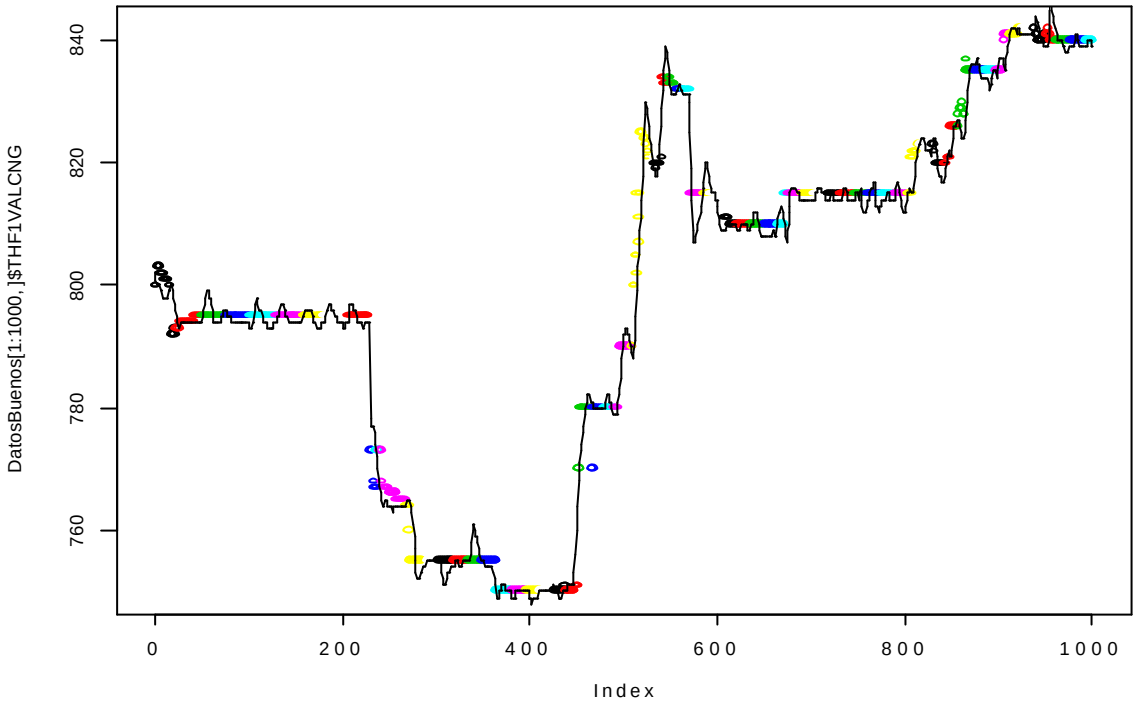


Figura 126. 1000 primeros valores de temperaturas reales THF1VALMED de la subzona 1 (línea continua) frente a los valores de temperatura de consigna THF1VALCNG (puntos) de la zona de calentamiento.

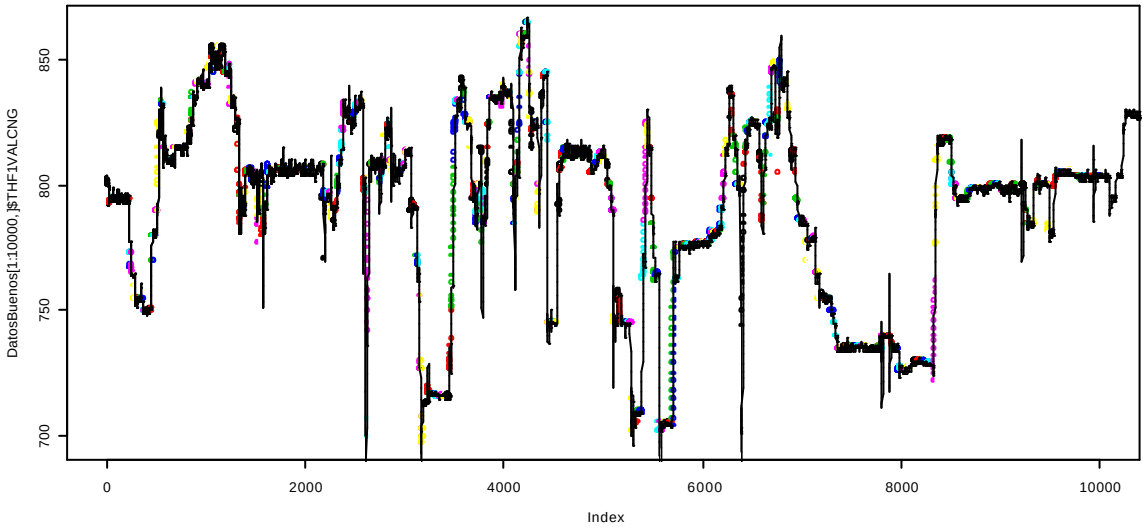


Figura 127. Representación de la evolución de los 20.000 primeros puntos (1.078 bobinas) de temperatura de consigna y reales de las zonas de calentamiento del horno.

EVOLUCIÓN DE LA TRANSICIÓN DE LAS TEMPERATURAS ENTRE BOBINAS

En la Figura 127 podemos ver la evolución de las temperaturas de consigna y reales de 1.078 bobinas de la base de datos.

Se ve claramente que existen en algunos momentos cambios bruscos entre las temperaturas de consigna de una bobina y la siguiente. Procedemos a analizar algunas de ellas mediante el programa *"R"*.

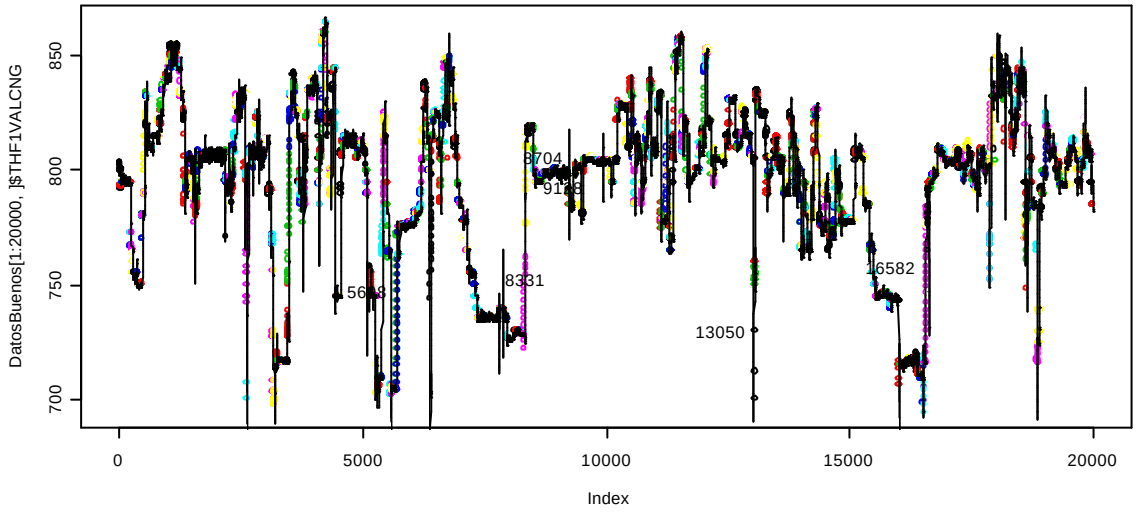


Figura 128. Identificación de algunos puntos en el gráfico.

```
# Dibujamos los puntos de consigna y medios de cada bobina (con color dif)
plot(DatosBuenos[1:20000,]$THF1VALCNG,col=DatosBuenos[1:20000,]$CODBOBINA)
lines(DatosBuenos[1:20000,]$THF1VALMED,col=DatosBuenos[1:20000,]$CODBOBINA)


# Identificamos manualmente algunos puntos
identify(DatosBuenos[1:20000,]$THF1VALCNG)
      [1]  5698  8331  8704  9128 13050 16582

DatosBuenos[5698,]$CODBOBINA
      [1] 11573043
DatosBuenos[8331,]$CODBOBINA
      [1] 11593030
DatosBuenos[8704,]$CODBOBINA
      [1] 11593044
DatosBuenos[9128,]$CODBOBINA
      [1] 11603010
DatosBuenos[13050,]$CODBOBINA
      [1] 11633009
DatosBuenos[16582,]$CODBOBINA
      [1] 11663006
```

Figura 129. Identificación de algunas bobinas con temperaturas de transición elevadas.

Para comprobar esas transiciones de temperaturas, se han analizado las temperaturas de consigna de veinte en veinte. En las figuras siguientes podemos apreciar claramente **cómo las temperaturas de consigna que genera el modelo son bastante irregulares cuando se producen cambios bruscos entre una bobina y otra.**

```
# Obtenemos las bobinas
Bobinas <- tapply(DatosBuenos$CODBOBINA,DatosBuenos$CODBOBINA,max)

# Dibujamos las bobinas desde la 'bobini' hasta 'bobini+20'
bobini <- 40
plot(DatosBuenos[CODBOBINA>Bobinas[bobini] &
CODBOBINA<Bobinas[bobini+20],]$THF1VALCNG,col=DatosBuenos[CODBOBINA>Bobinas[bobi
ni] & CODBOBINA<Bobinas[bobini+20],]$CODBOBINA)

# Obtenemos el código de la bobina primera y la ultima dibujadas
Bobinas[bobini]
      11523042
Bobinas[bobini+20]
      11523062
```

Figura 130. Programa que nos permite visualizar las temperaturas de consigna de las bobinas deseadas.

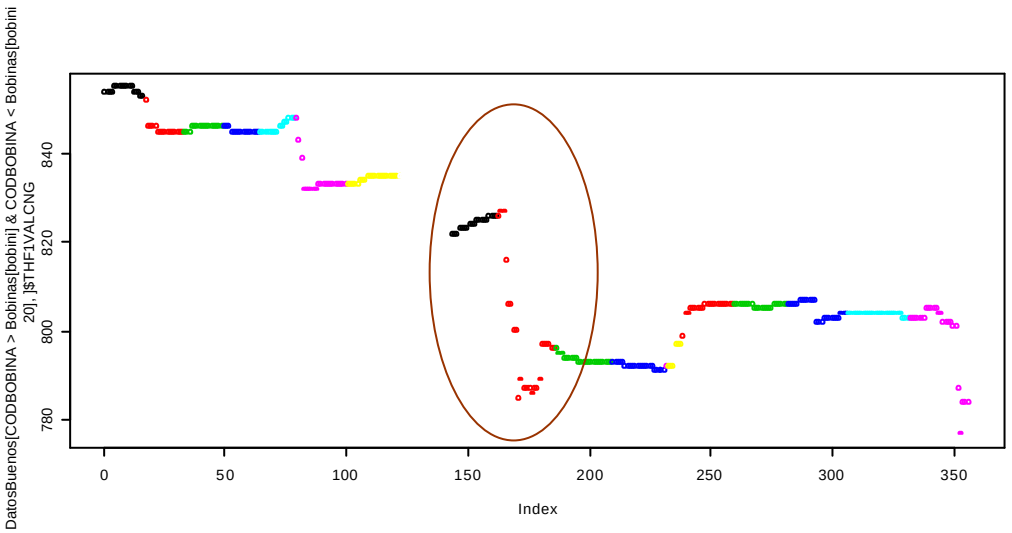


Figura 131. Evolución de las temperaturas de consigna de la bobina 11523083 a la bobina 11533024.

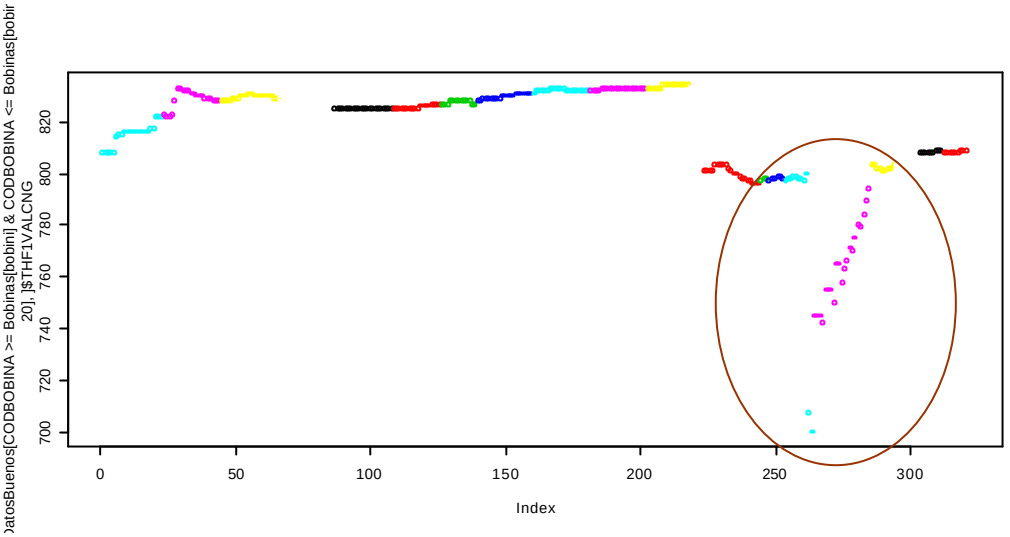


Figura 132. Evolución de las temperaturas de consigna de la bobina 11543005 a la bobina 11543026.

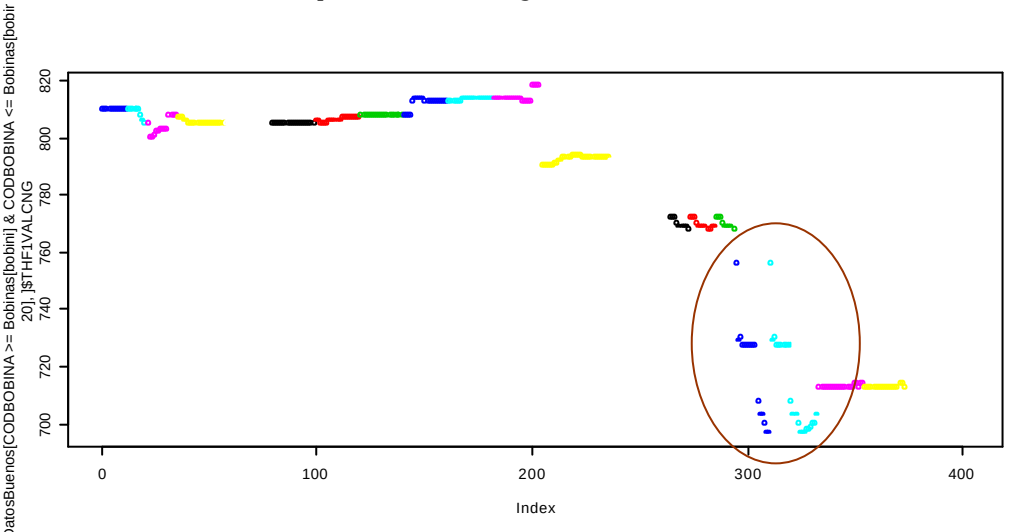


Figura 133. Evolución de las temperaturas de consigna de la bobina 11543046 a la bobina 11543066.

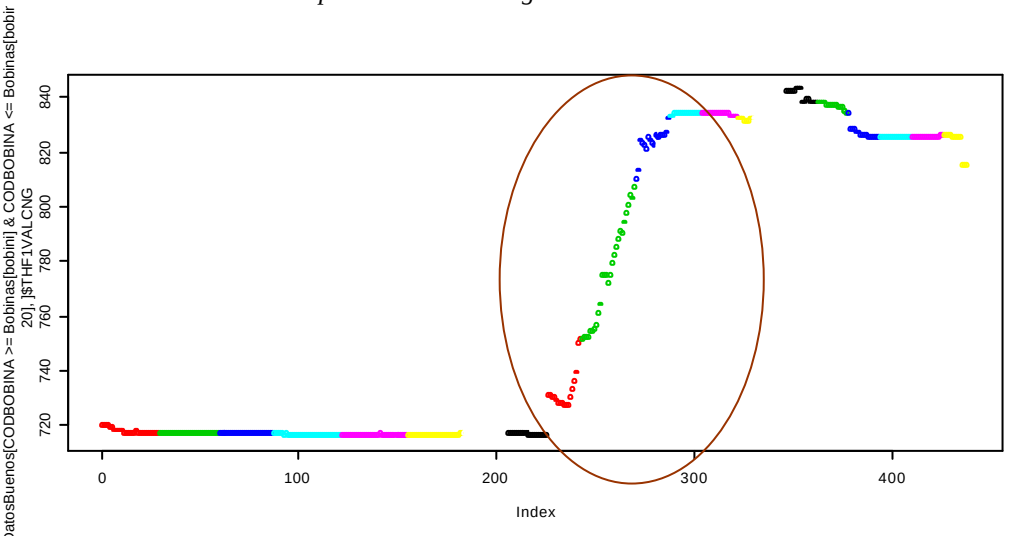


Figura 134. Evolución de las temperaturas de consigna de la bobina 11543066 a la bobina 11553015.

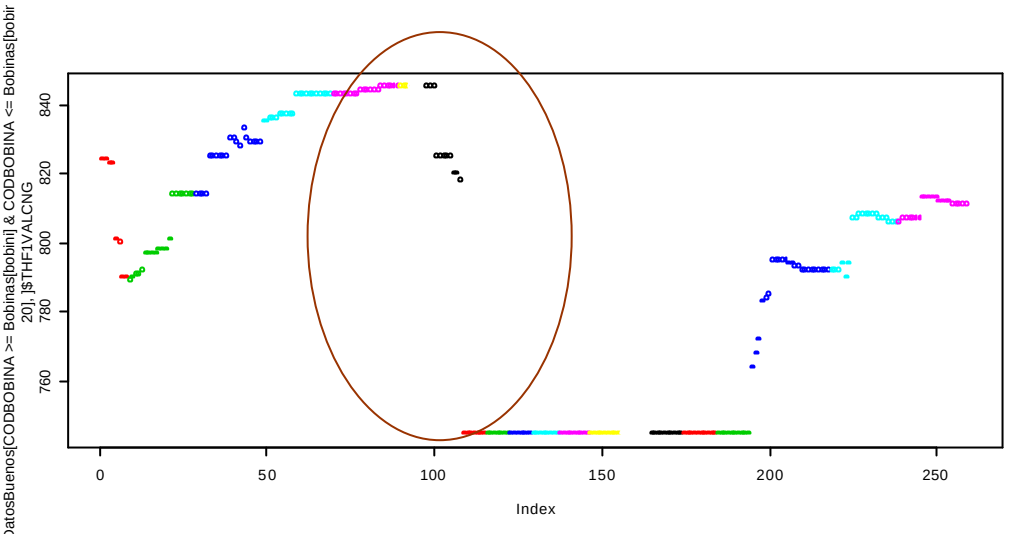


Figura 135. Evolución de las temperaturas de consigna de la bobina 11563006 a la bobina 11563026.

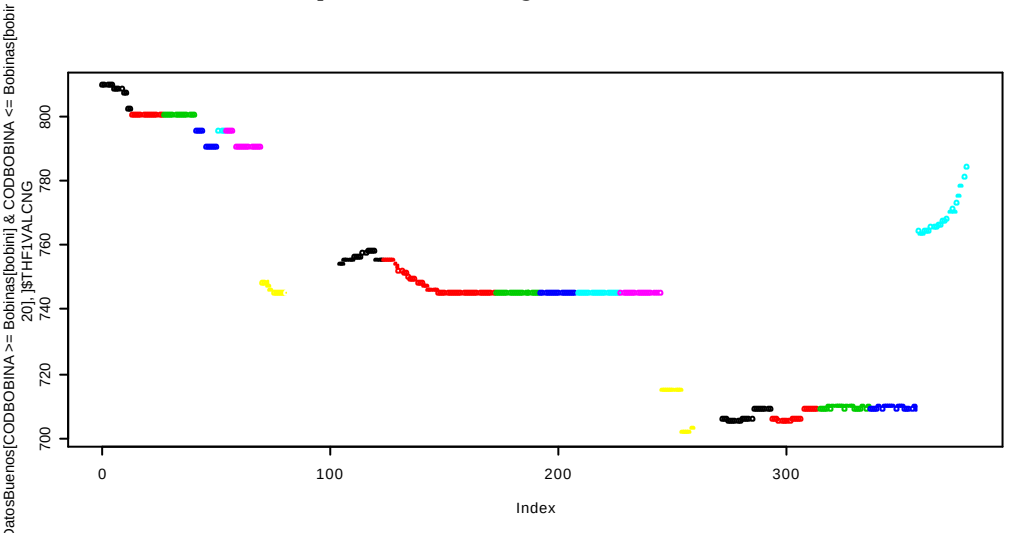


Figura 136. Evolución de las temperaturas de consigna de la bobina 11573009 a la bobina 11573029.

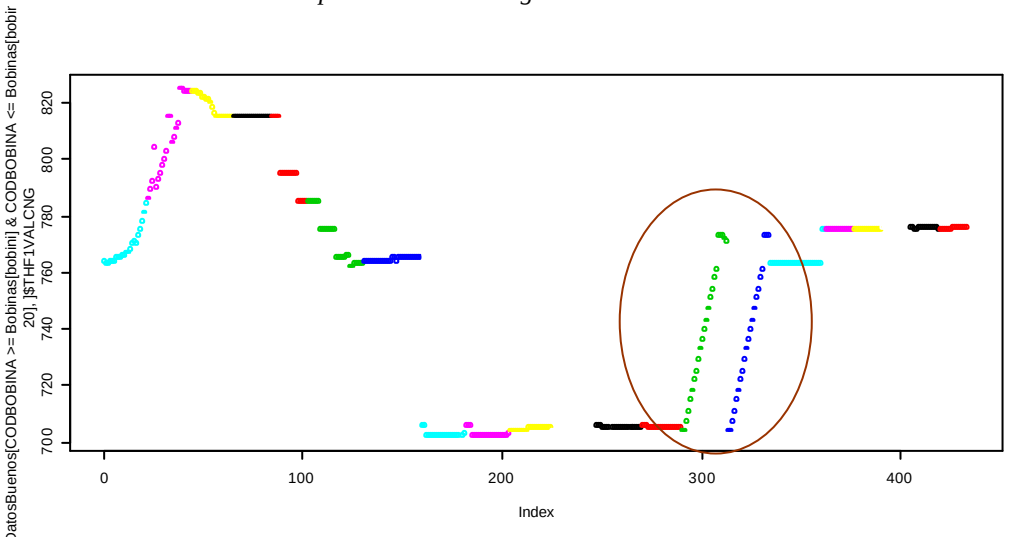


Figura 137. Evolución de las temperaturas de consigna de la bobina 11573029 a la bobina 11573050.

En las figuras anteriores, se muestra la evolución de las temperaturas de consigna de la zona 1 para diversos grupos de bobinas. Claramente se puede apreciar que hay unas bobinas en las que las temperaturas de consigna apenas varían unos grados mientras que en otras, que suelen ser de transición, muestran un incremento o decremento progresivo de estas temperaturas.

También se advierten comportamientos extraños como los de la Figura 133 y la Figura 137, donde las temperaturas de consigna de la zona 1 no siguen una evolución lógica que correspondería a un paso progresivo entre dos temperaturas distintas, sino que "inexplicablemente" se repite el proceso en dos bobinas consecutivas. **Estos comportamientos pueden achacarse a funcionamientos en "modo manual", aunque debido a una falta de variable que indique ese modo, no es posible determinarlo con seguridad.** Claramente se vislumbra **la necesidad de una variable que indique el modo de funcionamiento del horno.**

5.5.3.5 CARACTERIZACIÓN DEL COMPORTAMIENTO DE LAS TEMPERATURAS DE CONSIGNA

Para caracterizar este comportamiento, se propone la generación de unas nuevas variables que definan el comportamiento de las temperaturas aplicadas a cada bobina. Estas nuevas variables serán las siguientes (ver Figura 138):

- Temperatura de consigna media, de todos los instantes, (*THFxMEDTOTAL*) para cada bobina.
- Diferencia entre el valor máximo y el mínimo de la temperatura de consigna para cada bobina (*THFxDIFTOTAL*) que indica "el grado de variación" de la temperatura de consigna en esa bobina.

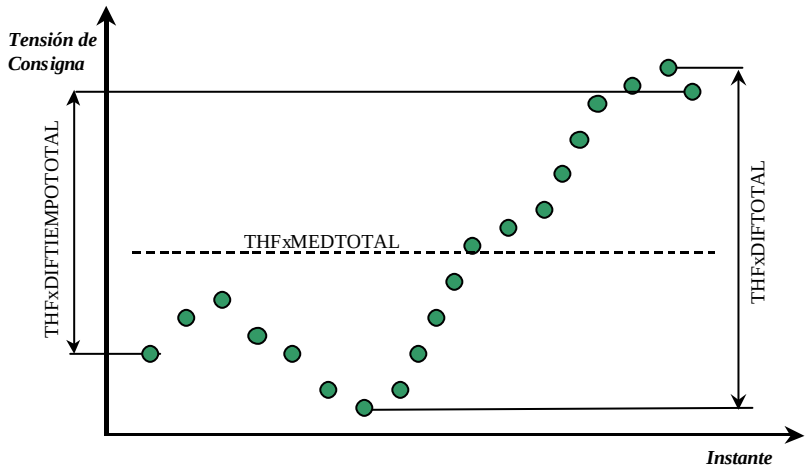


Figura 138. Caracterización del comportamiento de las temperaturas de consigna en una subzona para una bobina.

- Diferencia entre el valor de consigna del primer instante y del último de la temperatura de consigna para cada bobina. (*THFxDIFTIEMPOTOTAL*).
- Máximo instante de la bobina (*MAXINSTANTE*) que corresponde al máximo instante de cada bobina (cada instante corresponde a 100m.).

También se considera interesante el estudio del comportamiento de las temperaturas de consigna con respecto a la bobina anterior y la posterior.

```
# Obtenemos el instante máximo de cada bobina
MAXINSTANTE <- tapply(DatosBuenos$INSTANTE,DatosBuenos$CODBOBINA,max)

# Obtenemos el valor de consigna máximo y minimo de temperatura por bobina
MINCNG <- tapply(DatosBuenos$THF1VALCNG,DatosBuenos$CODBOBINA,min)
MAXCNG <- tapply(DatosBuenos$THF1VALCNG,DatosBuenos$CODBOBINA,max)


# Obtenemos el código de cada bobina
CODBOB <- tapply(DatosBuenos$CODBOBINA,DatosBuenos$CODBOBINA,max)

# Obtenemos la consigna de temperatura al principio y al final de cada bobina
MATCNGINI <- DatosBuenos[c(TRUE,(DatosBuenos[1:29260,]$CODBOBINA!=
DatosBuenos[2:29261,]$CODBOBINA)),]$THF1VALCNG
MATCNGFIN <- DatosBuenos[DatosBuenos[1:29260,]
$CODBOBINA!=DatosBuenos[2:29261,]$CODBOBINA,]$THF1VALCNG

# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(DatosBuenos$THF1VALCNG,DatosBuenos$CODBOBINA,mean)

# Obtenemos las variables finales
THF1MEDTOTAL <- round(VALMEAN)
THF1DIFTOTAL <- MAXCNG-MINCNG
THF1DIFTIEMPOTOTAL <- MATCNGFIN-MATCNGINI[1:1650]

# Creamos una matriz con las variables y el código de cada bobina
CONSIGNAS1 <- as.matrix(cbind(CODBOB[1:1650],MAXINSTANTE[1:1650],
THF1MEDTOTAL[1:1650],THF1DIFTOTAL[1:1650],THF1DIFTIEMPOTOTAL[1:1650]))


# Visualizamos las 15 primeras bobinas
CONSIGNAS1[1:10,]
             [,1] [,2] [,3] [,4] [,5]
11523001 11523001   24  799   11   -7
11523002 11523002   26  794    2    2
11523003 11523003   26  795    0    0
11523004 11523004   26  795    0    0
11523005 11523005   28  795    0    0
11523006 11523006   27  795    0    0
11523007 11523007   23  795    0    0
11523008 11523008   27  795    0    0
11523010 11523010   22  795    0    0

# Pasamos la matriz 'CONSIGNAS1' a un archivo de texto
write.table(CONSIGNAS1,"C:\\PISON\\DOCTORADO\\TESIS_ESTUDIO\\consignas.txt",quot
e=FALSE,sep=",",row.names=FALSE,col.names=FALSE)
```

Figura 139. Programa que obtiene las variables anteriormente descritas.

Procedemos a estudiar este tipo de variables, para ello pasamos la matriz *CONSIGNAS1* a un archivo de texto y lo modificamos para que pueda ser leído por el programa de visualización *freeware DAVIS* [DAV02].

Una vez realizada la visualización mediante coordenadas paralelas de las matriz *CONSIGNAS1* podemos extraer las siguientes conclusiones (Figura 140):

- La mayoría de las curvas de consignas de temperatura tienen un comportamiento aceptable (líneas en rojo). Se puede observar, cómo la mayoría de las bobinas tienen una diferencia entre valores de consigna máximos y mínimos bastante reducida (*THF1DIFTOTAL*) y una diferencia entre el primer valor y el último próxima a cero (*THF1DIFTIEMPOTOTAL*).
- Las líneas en azul, muestran bobinas con valores de *THF1DIFTOTAL* elevados y *THF1DIFTIEMPOTOTAL* bastante separados de cero. Siguiendo la trayectoria de esas líneas en los otros atributos, vemos claramente que tanto el valor medio de la temperatura aplicada (*THF1MEDTOTAL)* y la longitud de la bobina *(MAXINSTANTE)* no influyen en las mismas ya que los cortes de las líneas azules se distribuyen de forma homogénea por cada uno de estos ejes.

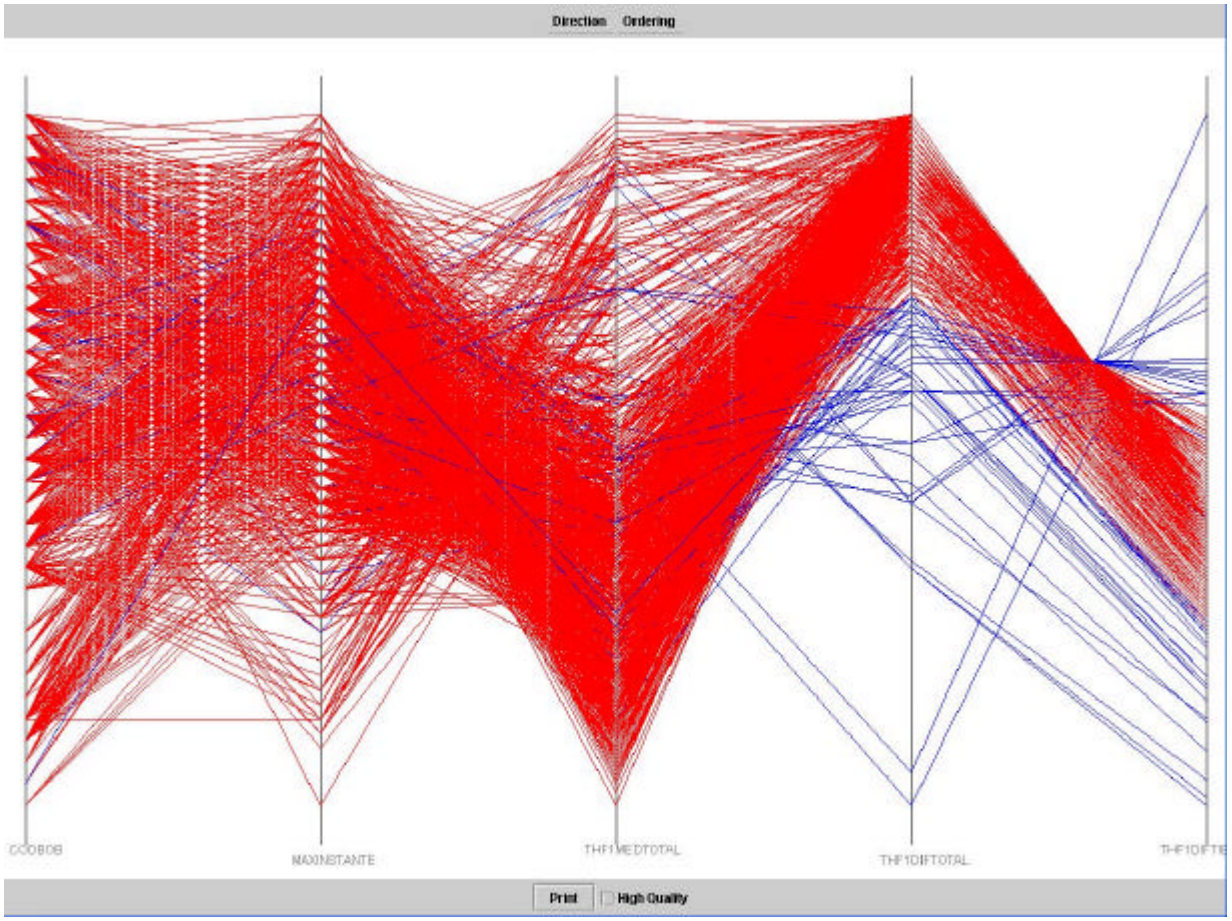


Figura 140. Grafico en coordenadas paralelas de la matriz CONSIGNAS1.

- También se aprecia, que las variables *THF1DIFTOTAL* y *THF1DIFTIEMPOTOTAL* están bastante correladas, por lo que podemos eliminar una de estas variables.

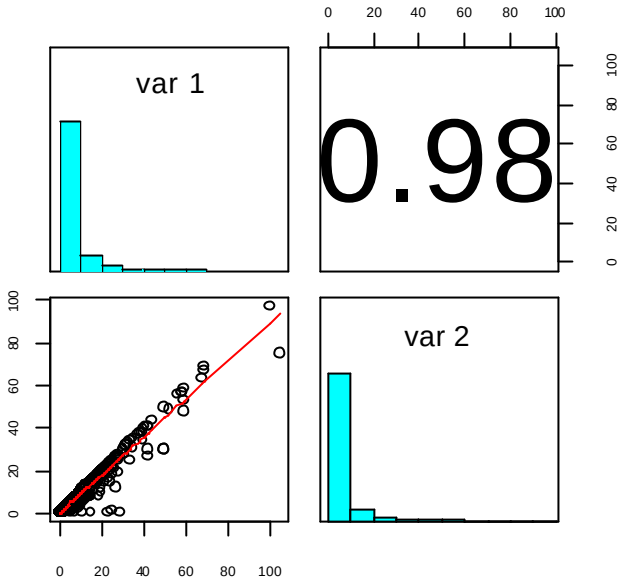


Figura 141. Correlación entre las variables THF1DIFTOTAL y THF1DIFTIEMPOTOTAL

En conclusión, **vemos que solo unas pocas bobinas, tienen una diferencia sustancial entre sus temperaturas de consigna y que la mayoría cumplen un régimen de curvas con temperaturas bastante homogéneas.**

Como se verá en apartados posteriores, resultará interesante comprobar si las variaciones de temperatura entre los distintos instantes de una bobina pueden afectar claramente a la temperatura final de la misma.

CATEGORIZACIÓN DEL COMPORTAMIENTO DE LAS TEMPERATURAS DE CONSIGNA DE CADA BOBINA

Para mejorar los estudios posteriores, se procede a clasificar cada curva según la siguiente tabla (*TIPOCURVATHFx)*:

Valor de *THF1DIFTOTAL*	Signo de *THF1DIFTIEMPOTOTAL*	NOMBRE
< 10 ºC	Positivo o Negativo	HORIZONTAL (H)
>= 10 ºC y < 30ºC	Positivo	BAJACRECIENTE (BC)
>= 10 ºC y < 30ºC	Negativo	BAJADECRECIENTE (BD)
>= 30 ºC y < 60ºC	Positivo	MEDIACRECIENTE (MC)
>= 30 ºC y < 60ºC	Negativo	MEDIADECRECIENTE (MD)
>= 60 ºC y < 200ºC	Positivo	ALTACRECIENTE (AC)
>= 60 ºC y < 200ºC	Negativo	ALTADECRECIENTE (AD)
>= 200	Positivo o Negativo	ERROR (E)

Tabla 24. Categorización de las curvas de temperatura de consigna aplicadas a cada bobina.

5.5.3.6 EVOLUCIÓN DE LAS TEMPERATURAS DE CONSIGNA DE LAS SUBZONAS 1,3,5 Y 7 DE LA ZONA DE CALENTAMIENTO DEL HORNO (THF1VALCNG, THF3VALCNG, THF5VALCNG, THF7CALCNG)

A continuación estudiaremos la relación entre las variables de consigna de las subzonas 1, 3, 5 y 7 de la zona de calentamiento del horno.

```
# Obtenemos las variables de temperatura de consigna de las subzonas 1,3,5 y 7
VALMEAN <- tapply(DatosBuenos$THF1VALCNG,DatosBuenos$CODBOBINA,mean)
THF1MEDTOTAL <- round(VALMEAN)

VALMEAN <- tapply(DatosBuenos$THF3VALCNG,DatosBuenos$CODBOBINA,mean)
THF3MEDTOTAL <- round(VALMEAN)

VALMEAN <- tapply(DatosBuenos$THF5VALCNG,DatosBuenos$CODBOBINA,mean)
THF5MEDTOTAL <- round(VALMEAN)

VALMEAN <- tapply(DatosBuenos$THF7VALCNG,DatosBuenos$CODBOBINA,mean)
THF7MEDTOTAL <- round(VALMEAN)

# Obtenemos el código de cada bobina
CODBOB <- tapply(DatosBuenos$CODBOBINA,DatosBuenos$CODBOBINA,max)

# Creamos una matriz con las variables y el código de cada bobina

CONSIGNAS1 <- as.matrix(cbind(CODBOB[1:1650],THF1MEDTOTAL[1:1650],
THF3MEDTOTAL[1:1650],THF5MEDTOTAL[1:1650],THF7MEDTOTAL[1:1650]))

# Visualizamos las 20 primeras bobinas
CONSIGNAS1[1:20,]
             [,1] [,2] [,3] [,4] [,5]
11523001 11523001  799  829  852  852
11523002 11523002  794  824  849  849
```

```
11523003 11523003  795  825  850  850
11523004 11523004  795  825  850  850
11523005 11523005  795  825  850  850
11523006 11523006  795  825  850  850
11523007 11523007  795  825  850  850
11523008 11523008  795  825  850  850
11523010 11523010  795  825  850  850
11523012 11523012  769  799  822  822
11523013 11523013  773  803  826  826
11523014 11523014  767  797  818  818
11523015 11523015  757  787  803  804
11523016 11523016  757  787  804  804
11523017 11523017  755  785  800  800
11523018 11523018  755  785  800  800
11523019 11523019  755  785  800  800
11523020 11523020  755  785  800  800
11523021 11523021  750  780  795  795
11523022 11523022  750  780  795  795

# Pasamos la matriz 'CONSIGNAS1' a un archivo de texto
write.table(CONSIGNAS1,"C:\\PISON\\DOCTORADO\\TESIS_ESTUDIO\\consignas.txt",quot
e=FALSE,sep=",",row.names=FALSE,col.names=FALSE)
```

Figura 142. Programa para obtener las temperaturas de consigna de las subzonas 1, 3, 5 y 7.

Como se ve en la Figura 142, solamente se han elegido las temperaturas de consigna de las zonas impares (1, 3, 5 y 7), porque las temperaturas de las pares, tal y como las especifica el modelo, son iguales a las de las subzona anterior.

El objetivo principal que perseguimos, es el de comprender **cómo evolucionan en la mayoría de las bobinas las temperaturas de consigna del horno**. Para ver si la evolución es progresiva o no, volvemos a utilizar el gráfico de coordenadas paralelas anteriormente estudiado. Para desarrollar este gráfico, necesitamos convertir el archivo de datos a un nuevo archivo con la cabecera siguiente.

```
5 1650 (Dimension, Número Observaciones)
CODBOBINA
THF1CNG
THF3CNG
THF5CNG
THF7CNG
11523001 11843037 4  (Minimo y Máximo)
700. 900. 4
700. 900. 4
700. 900. 4
700. 900. 4
11523001 799 829 852 852 (Primera Observación)
11523002 794 824 849 849
11523003 795 825 850 850
11523004 795 825 850 850
11523005 795 825 850 850
11523006 795 825 850 850
```

```
11523007 795 825 850 850
11523008 795 825 850 850
11523010 795 825 850 850
11523012 769 799 822 822
11523013 773 803 826 826
11523014 767 797 818 818
11523015 757 787 803 804
11523016 757 787 804 804
11523017 755 785 800 800
11523018 755 785 800 800
.....
```

Figura 143. Cabecera necesaria para el programa XMDVTOOL de visualización.

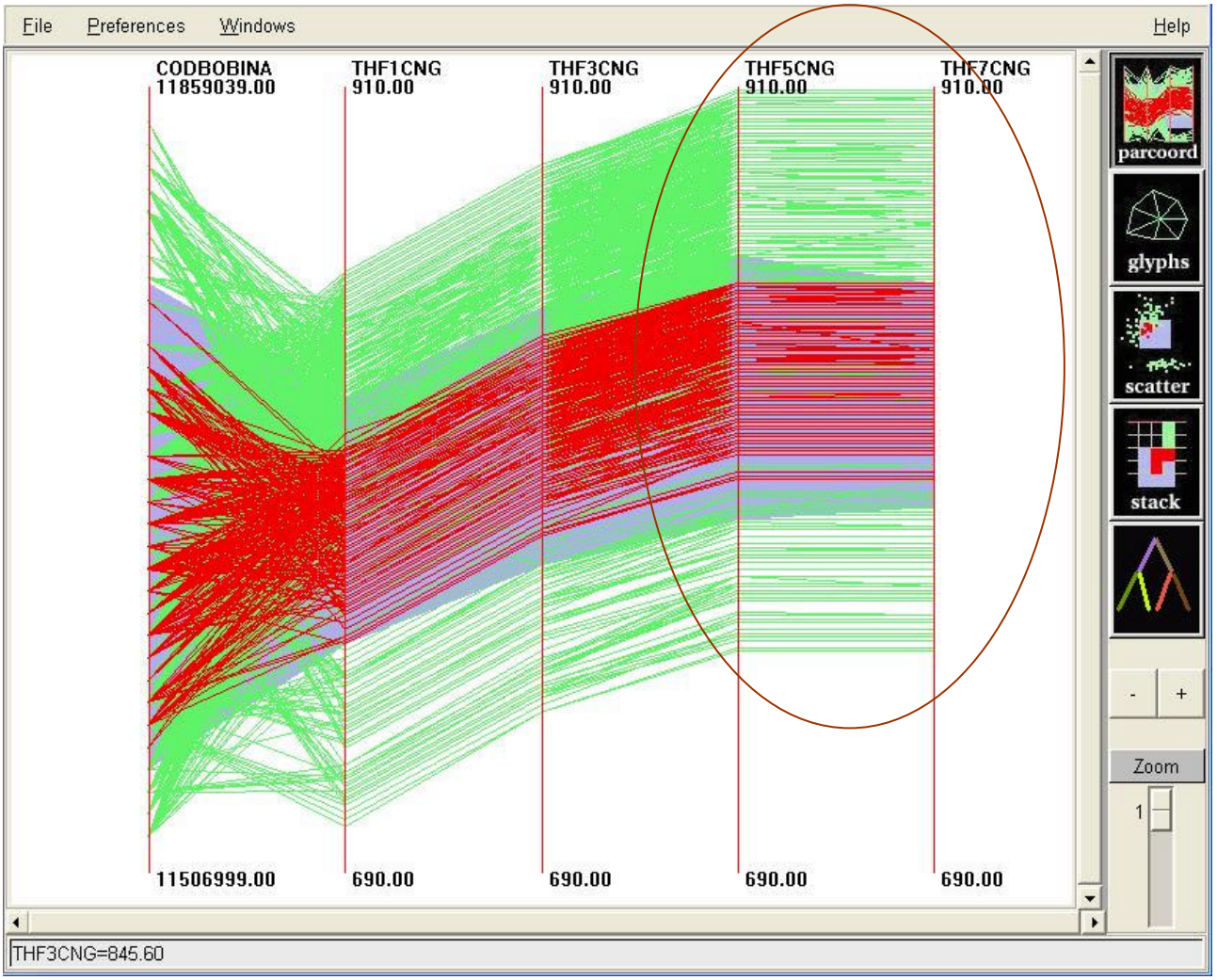


Figura 144. Consignas de temperatura de las subzonas 1,3, 5 y 7 de la zona de temperatura del horno.

En la figura anterior, podemos apreciar claramente la evolución de las temperaturas de consigna a lo largo de la zona de calentamiento del horno. Cabe destacar, **que las temperaturas *THF5VALCNG* y *THF7VALCNG* son prácticamente iguales para casi todas las bobinas**.

Esta figura, también nos sirve para corroborar que los valores de incremento asignados a los *STEPS* son casi siempre los mismos, tal y como se ha explicado en párrafos anteriores.

Por lo tanto inicialmente y para reducir el número de variables a utilizar, se estudiarán solamente las temperaturas de consigna de las variables 1, 3 y 5.

Trabajando con la herramienta *XMDVTOOL* [XMD02] con los gráficos de coordenadas paralelas, hemos podido ver el comportamiento extraño de dos bobinas de las inicialmente estudiadas.

Como se puede observar la mayoría de las temperaturas de consigna van creciendo a medida que se pasa de una subzona a otra. En cambio, podemos apreciar claramente, que se han encontrado dos bobinas (ver figura siguiente) cuya temperatura de consigna *THF3CNG* es menor que la *THF1CNG*. Este comportamiento se estudiará con más detenimiento en apartados posteriores.

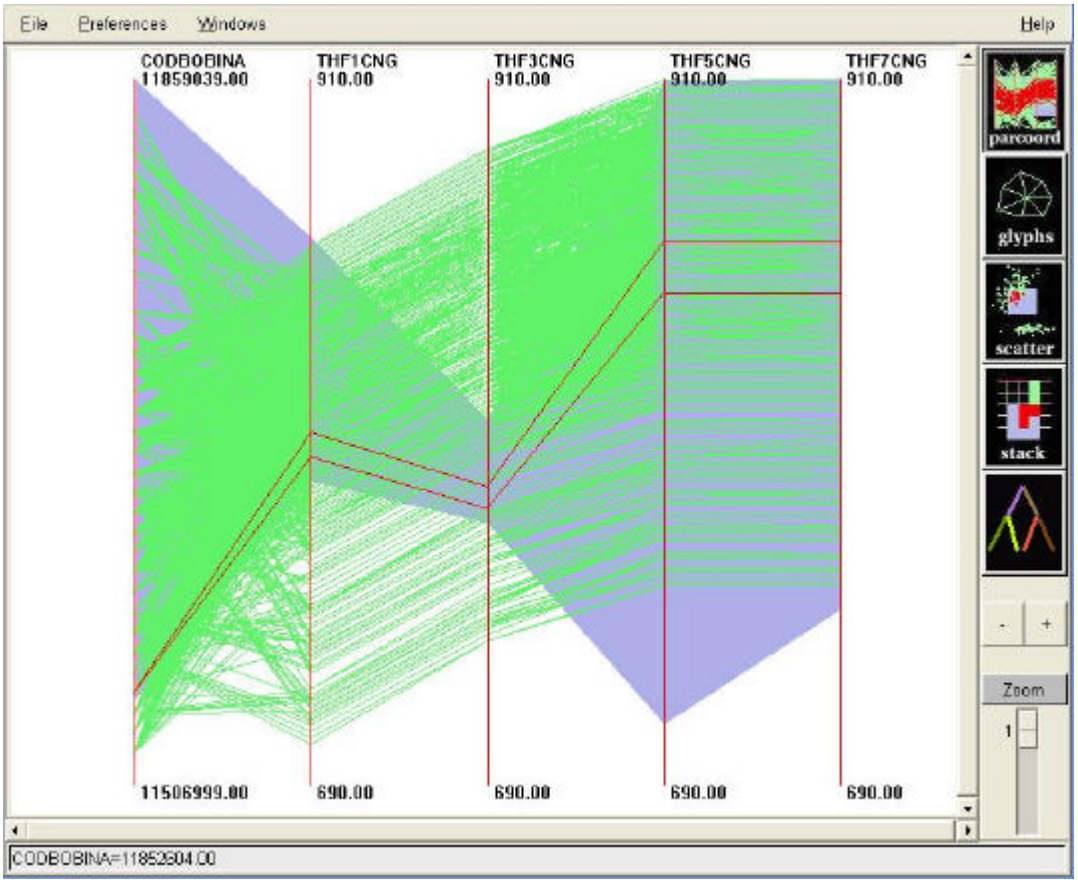


Figura 145. Diagrama de coordenadas paralelas con dos bobinas cuyas temperaturas de consigna tienen un comportamiento extraño.

5.5.3.7 CONCLUSIONES INICIALES DEL ESTUDIO EXPLORATORIO DE LAS VARIABLES DE TEMPERATURA THF EN LA ZONA DE CALENTAMIENTO DEL HORNO

Por lo tanto, después del análisis realizado a las variables de temperatura de consigna de la zona de calentamiento del horno, podemos concluir que:

- **Las variables más fiables para cada instante de la bobina son las que nos dan el valor medio** (THFxVALMED). Se desestima el uso de las variables MAX y MIN por dar valores erróneos.
- Para los estudios posteriores, **es conveniente eliminar aquellas bobinas que tienen valores de observación erróneos**. Las causas y posibles formas de solución de esos errores ya se han explicado anteriormente. Se eliminarán las bobinas con temperaturas con valores menores de 100.
- Los dispositivos que controlan la temperatura de cada zona en la parte de calentamiento del horno siguen con bastante fidelidad la temperatura de consigna tal y como se puede observar en la correlación que muestra la Figura 146. Por lo tanto, inicialmente, **se utilizarán para el estudio únicamente las temperaturas de consigna** (*THFxVALCNG*).

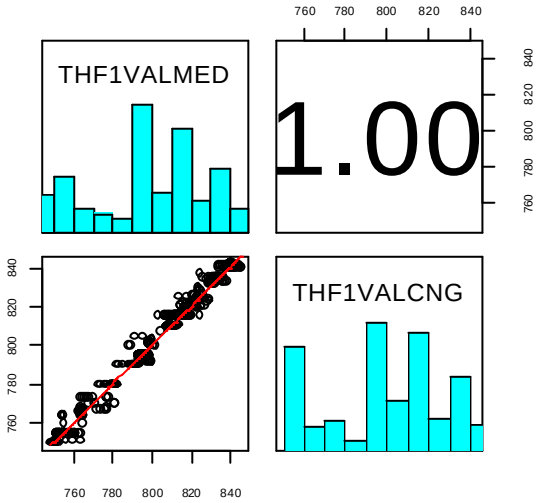


Figura 146. Correlación y distribución entre la temperatura real y la de consigna de la subzona 1.

- **Para cada bobina, será conveniente caracterizar el comportamiento** de las temperaturas de consigna de las zonas (creación de las variables *THFxMEDTOTAL, THFxDIFTOTAL, TIPOCURVATHFx).*
- **Sólo se utilizarán para el estudio las variables de consigna correspondientes a las zonas 1, 3 y 5,** ya que las temperaturas de las zonas 2, 4, 6 y 8 son iguales que las de las 1, 3, 5 y 7; y además, la temperatura de consigna de la zona 5 no difiere mucho de la 7. Aún así, para analizar otro tipo de errores, se considera que puede utilizarse solamente el valor la zona 1 como muestra representativa de las demás zonas, ya que en casi todas las bobinas, los *STEPS* no varían en demasía.
- **Es necesaria una variable que describa si el funcionamiento del horno está en "Modo Manual" o "Modo Automático" ya que muchos comportamientos anómalos pueden no ser achacados al sistema de control.**
- **Las temperaturas de consigna de la zona 1 debe ser menor que la zona 3, y ésta menor que la zona 5.**

5.5.4 ESTUDIO EXPLORATORIO DE LAS VARIABLES DE TEMPERATURA DE LOS PIRÓMETROS 1, 2 Y 3 (TMPP1, TMPP2, TMPP3)

Una vez se han estudiado los datos correspondientes a las variables de temperatura de consigna y medias de las subzonas correspondientes a la etapa de calentamiento del horno, procedemos a analizar las temperaturas de consigna y medias de los pirómetros 1, 2 y 3 (TMPP1, TMPP2 y TMPP3).

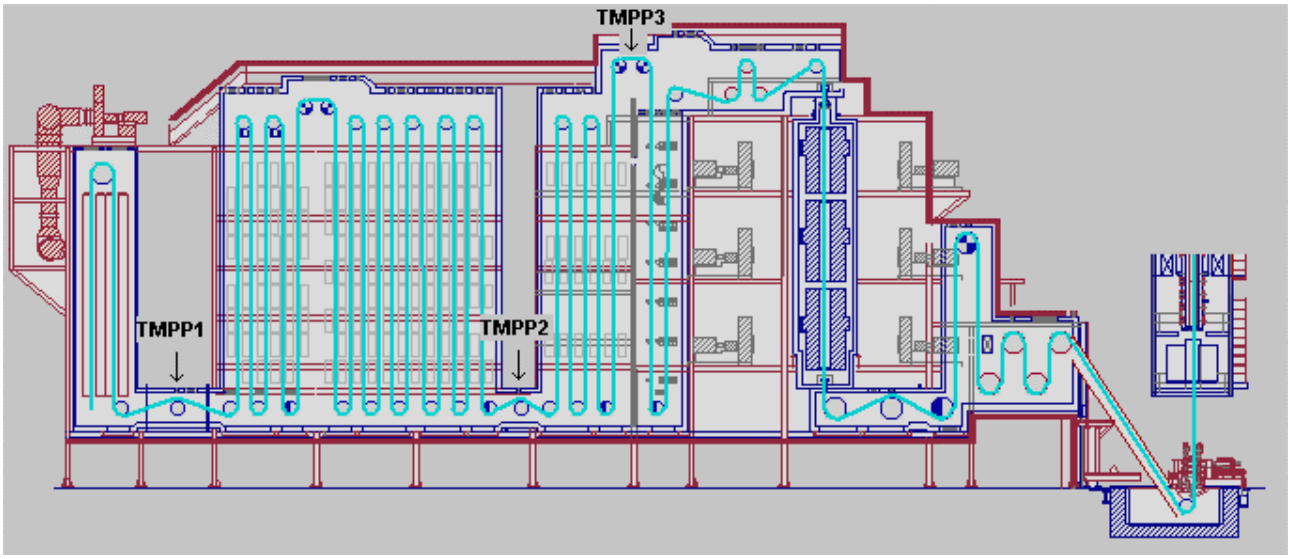


Figura 147. Posición de los pirómetros 1, 2 y 3.

Estos pirómetros miden la temperatura de la banda en tres puntos del horno, tal y como se muestra en la Figura 147. El pirómetro uno mide la temperatura de la banda de acero a la entrada de la zona de calentamiento, el segundo a la salida de la misma zona y el tercero a la salida de la zona de mantenimiento.

	TMPP1VALME	TMPP1VALCN	TMPP2VALME	TMPP2VALCN	TMPP3VALME	TMPP3VALCN	CODBOBINA
979	239	0	805	800	799	0	11523068
980	239	0	806	800	799	0	11523068
981	239	0	806	800	799	0	11523068
982	239	0	806	800	799	0	11523068
983	240	0	806	800	799	0	11523068
984	240	0	806	800	799	0	11523068
985	240	0	806	800	800	0	11523068
986	241	0	807	800	800	0	11523068
987	240	0	807	800	800	0	11523068
988	241	0	808	800	800	0	11523068
989	243	0	808	800	800	0	11523068
990	242	0	808	800	800	0	11523068
991	242	0	808	800	800	0	11523068
992	243	0	-1	0	-1	0	11523068

Figura 148. Algunos valores de las temperaturas de los pirómetros uno, dos y tres de una bobina.

Si visualizamos los datos obtenidos medios y de consigna para los tres pirómetros, advertimos que:

- **No se almacenan los valores de consigna de los pirómetros uno y tres**, ya que el modelo sólo establece como valor de consigna el de la temperatura de la banda a la salida de la zona de calentamiento y supone el valor de consigna del pirómetro tres igual al del pirómetro dos.
- **Existen valores erróneos al final de algunas bobina**. Parece que algunas veces, en el instante final de la bobina, no se almacena el valor de las temperaturas medias y de consigna de los pirómetros dos y tres.

5.5.4.1 ANÁLISIS DE LAS OBSERVACIONES ERRÓNEAS

Lo primero que se plantea y estudia, es el análisis de las observaciones erróneas mediante los programas y gráficas siguientes.

```
# Obtenemos el máximo instante para cada bobina (es decir la longitud)
BobinasMaxInstante <- tapply(DatosBuenos$INSTANTE,DatosBuenos$CODBOBINA,max)
dim(BobinasMaxInstante)
      [1] 1651
# Eliminamos las observaciones con temperaturas menores de 100
DatMalos <- DatosBuenos[!(TMPP1VALME>100 & TMPP2VALME>100 & TMPP3VALME>100),]
dim(DatMalos)
      [1] 228  25

# Calculamos el tanto por ciento de observaciones defectuosas
228*100/1651
      [1] 13.80981
```

Figura 149. Obtención de los datos erróneos de las temperaturas de los pirómetros.

Para realizar el análisis, se utiliza el programa de la Figura 149 que sirve para obtener las bobinas que presentan lecturas erróneas en los pirómetros uno, dos y tres.

```
DatMalos[1:29,19:25]
     TMPP1VALME TMPP1VALCN TMPP2VALME TMPP2VALCN TMPP3VALME TMPP3VALCN CODBOBINA
76           -1          0        820        825        825          0  11523003
130          -1          0         -1          0        824          0  11523005
267          -1          0        801        800        799          0  11523014
363          -1          0         -1        750         -1          0  11523020
451          -1          0        768        750        764          0  11523026
526          -1          0        812        800        801          0  11523031
555         224          0         -1        800         -1          0  11523035
570          -1          0        775        800        773          0  11523037
842          -1          0        772        775        767          0  11523057
992         243          0         -1          0         -1          0  11523068
1098         -1          0        824        825        820          0  11523076
1131         -1          0        826        825        824          0  11523080
1282         -1          0         -1          0         -1          0  11533007
1322         -1          0        819        825        826          0  11533009
1516         -1          0        818        825        824          0  11533022
1741         -1          0         -1          0         -1          0  11533033
1906         -1          0         -1          0         -1          0  11533039
```

2020	-1	0	825	825	825	0	11533043
2215	-1	0	-1	0	-1	0	11533052
2289	-1	0	803	800	797	0	11533064
2378	-1	0	824	825	824	0	11543005
2515	-1	0	822	828	827	0	11543012
2640	-1	0	842	830	832	0	11543022
2933	211	0	-1	0	-1	0	11543050
2974	-1	0	-1	0	-1	0	11543052
3015	-1	0	-1	0	817	0	11543054
3286	-1	0	-1	0	-1	0	11543067
3453	-1	0	-1	750	-1	0	11553001
3497	-1	0	-1	0	-1	0	11553003

Figura 150. Muestra de algunos de los valores de lectura erróneos encontrados.

El siguiente paso consiste en ver como están distribuidos.

```
# Obtenemos las bobinas con observaciones erróneas y las dibujamos en rojo
BobMalas <- !(TMPP1VALME>100 & TMPP2VALME>100 & TMPP3VALME>100)

# Creamos un vector de colores azul (4) y rojo (2)
cc <- rep(4,29261)
cc[BobMalas]<-2

# Añadimos la columna de colores y obtenemos para cada bobina el azul o rojo
ValBuenos <- cbind(DatosBuenos$CODBOBINA, cc)
BobinasTot <- tapply(ValBuenos[,2],ValBuenos[,1],min)

# Obtenemos la temperatura máxima de las medias del pirómetro 2
ValBuenos2 <- cbind(DatosBuenos$CODBOBINA, DatosBuenos$TMPP2VALME)
BobinasTot2 <- tapply(ValBuenos2[,2],ValBuenos2[,1],max)

# Creamos un vector de puntos negros (malos) y puntos blancos (buenos)
BobinasTotC <- rep(21,1651)
BobinasTotC[BobinasTot==2]<-19

# Dibujamos los datos
plot(BobinasTot2,col=BobinasTot,xlab="BOBINAS",ylab="max(TMPP2VALMED)",pch=Bobin
asTotC)
```

Figura 151. Programa que detecta y muestra las bobinas con valores defectuosos.

Cómo se puede apreciar, en la Figura 152 podemos ver como la distribución de los observaciones defectuosas se produce de una forma homogénea durante todo el mes estudiado. Por lo tanto, se llega a la conclusión de que es un error que se genera **durante todo el proceso y en el último instante de algunas bobinas**. Este error, con bastante seguridad, **se producirá por un defecto en el almacenamiento de los datos en el último momento o en el momento de cambio de alguna bobina y por lo tanto, no se considera problemático ya que se produce en un solo instante y de forma esporádica.**

Como estas observaciones defectuosas pueden perjudicar el desarrollo de estudios posteriores, se plantea su eliminación o cambio. Debido a que estos valores aparecen una vez y solamente en un 13% de las bobinas, **se decide rellenarlos con el valor aparecido en el instante anterior, ya que los mismos, por la inercia del sistema y del modelo, no varían significativamente con respecto a los del instante anterior.**

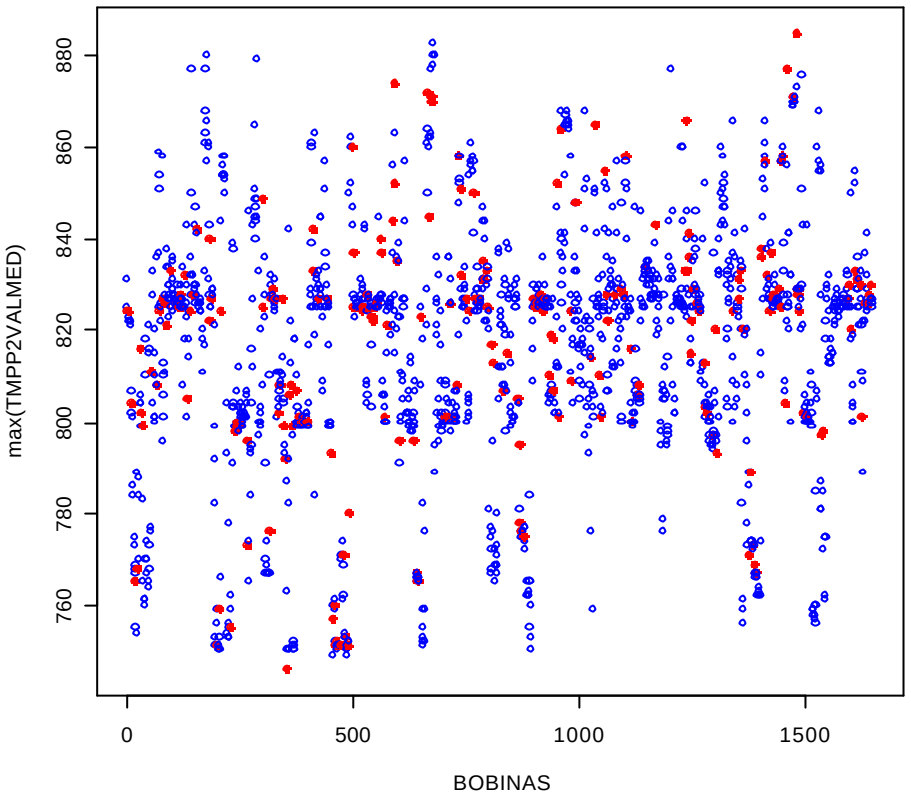


Figura 152. Distribución de las temperaturas máximas de cada bobina a la salida de la zona de calentamiento. En los puntos rellenos se muestran las bobinas que tienen la última lectura de temperaturas de los pirómetros defectuosa.

```
# Número de observaciones de bobinas malas
length(BobMalas)
[1] 29261

# Obtenemos las observaciones posteriores a las bobinas malas
BobMalasInsMenos <- BobMalas[2:29261]
length(BobMalasInsMenos)
[1] 29260

# Rellenamos los datos erróneos con los siguientes datos
DatosBuenos[BobMalas[1:29260],19:24] <- DatosBuenos[BobMalasInsMenos,19:24]
```

Figura 153. Programa que rellena los datos erróneos de las temperaturas de los pirómetros media y de consigna con los datos de temperatura del instante anterior.

5.5.4.2 ANÁLISIS DEL ERROR ENTRE LA TEMPERATURA DE CONSIGNA DE LA BANDA (TMPP2VALCNG) Y LA TEMPERATURA REAL (TMPP2VALMED) DE LA BANDA A LA SALIDA DE LA ZONA DE CALENTAMIENTO

Una vez hemos eliminado todos los datos espurios de la base de datos, procedemos a analizar el error que se produce entre la temperatura esperada de la banda y la temperatura real de la misma cuando sale de la zona de calentamiento.

```
# Cálculo del error medio de cada bobina MEAN(TMPP2VALMED-TMPP2VALCNG)
error <- tapply(DatosBuenos$TMPP2VALME-DatosBuenos$TMPP2VALCN,
DatosBuenos$CODBOBINA,mean)

# Cálculo del error máximo de cada bobina MAX(TMPP2VALMED-TMPP2VALCNG)
errormax <- tapply(abs(DatosBuenos$TMPP2VALME-DatosBuenos$TMPP2VALCN)
,DatosBuenos$CODBOBINA,max)
plot(error,xlab="BOBINAS",ylab="Max(TMPP2VALME-TMPP2VALCN)")

# Cálculo del error mínimo de cada bobina MIN(TMPP2VALMED-TMPP2VALCNG)
errormin <- tapply(abs(DatosBuenos$TMPP2VALME-DatosBuenos$TMPP2VALCN)
,DatosBuenos$CODBOBINA,min)

# Cálculo del la media del error del valor absoluto
sumabs <- function(x) mean(abs(x))
errorabs <- tapply(DatosBuenos$TMPP2VALME-DatosBuenos$TMPP2VALCN,
DatosBuenos$CODBOBINA,sumabs)
plot(errorabs,xlab="BOBINAS",ylab="Mean(abs(TMPP2VALME-TMPP2VALCN))")

# Dibujamos el error
plot(error,xlab="BOBINAS",ylab="Min(TMPP2VALME-TMPP2VALCN)")
```

Figura 154. Programa que visualiza los errores media, máximo y mínimo de cada bobina.

Primero, visualizamos para cada bobina el error[16] "medio" y "medio absoluto" y después, el error "máximo" y "mínimo".

[16] A partir de ahora y en todo el documento, llamaremos "error de una bobina" a la diferencia entre la temperatura esperada de la banda a la salida de la zona de calentamiento (temperatura de consigna del pirómetro 2) y la real (temperatura leída en el pirómetro 2). Hay que recordar, que uno de los objetivos finales de todo este trabajo consiste en comprender las causas que generan este "error" y desarrollar soluciones que lo minimicen.

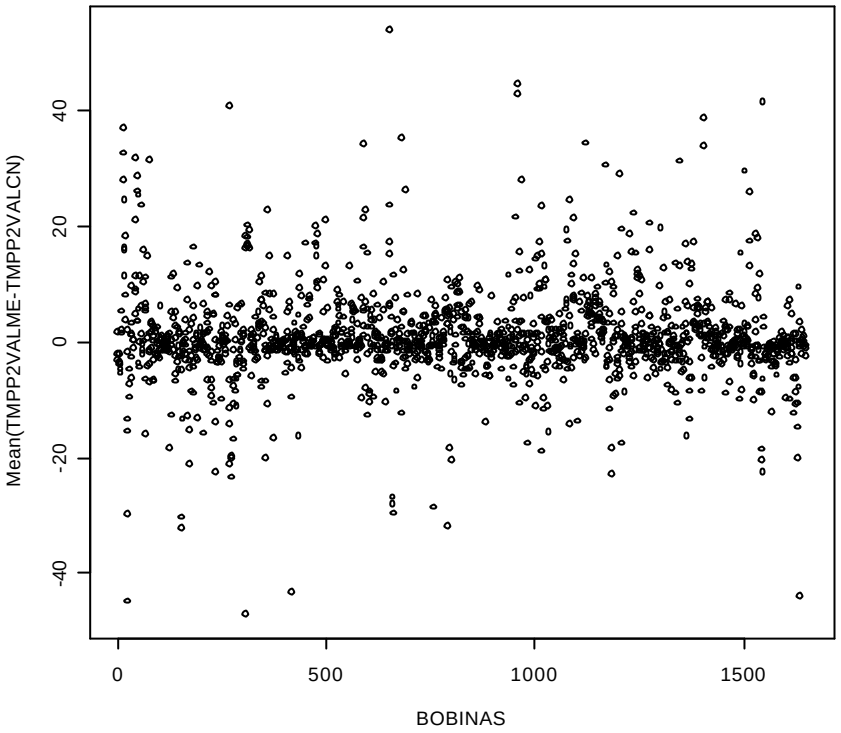


Figura 155. Error para cada bobina de la temperatura de la bobina en la salida de la zona de calentamiento.

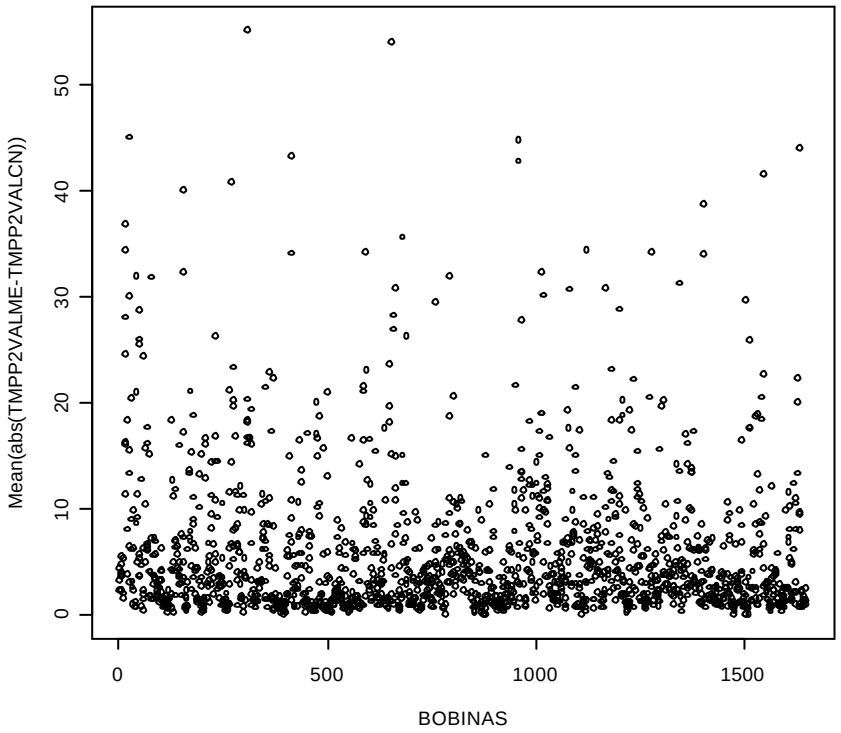


Figura 156. Error medio absoluto para cada bobina de la temperatura de la bobina de la zona de calentamiento.

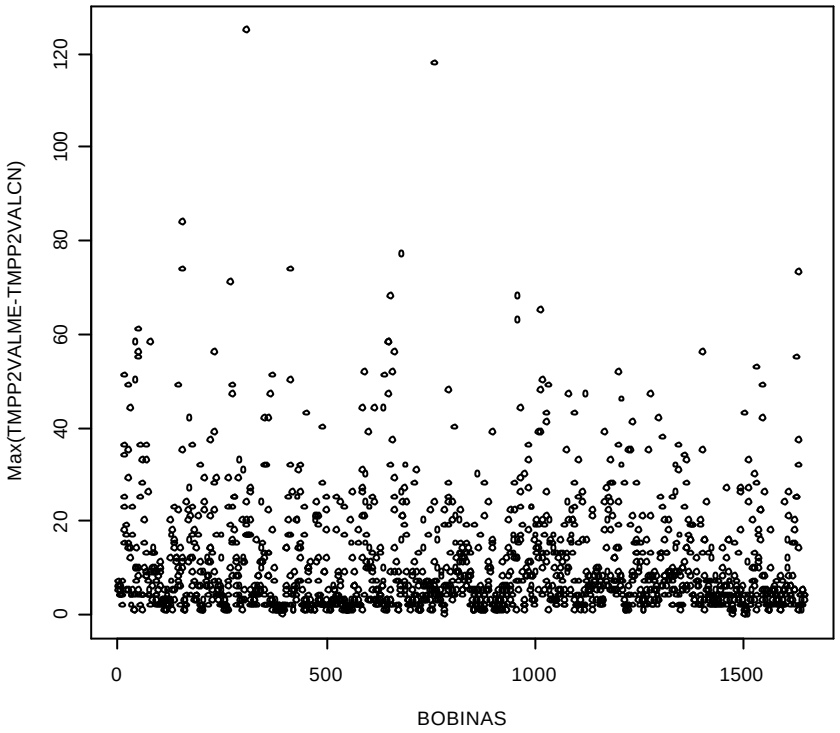


Figura 157. Error máximo para cada bobina de la temperatura de la bobina en la salida de la zona de calentamiento.

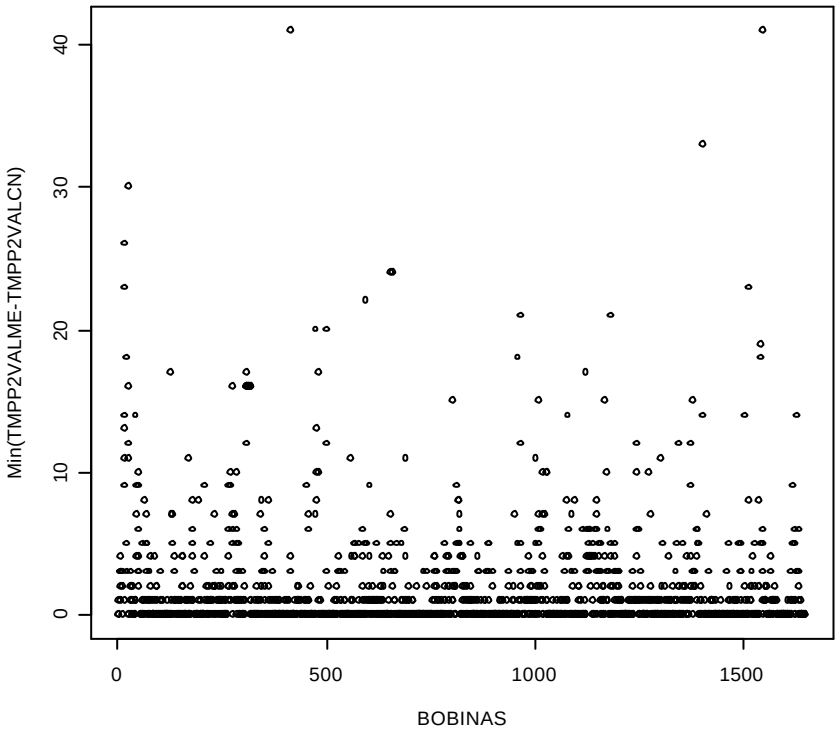


Figura 158. Error mínimo para cada bobina de la temperatura de la bobina en la salida de la zona de calentamiento.

Como nos muestra la Figura 157 y la Figura 158, existen numerosas bobinas **en las que el error mínimo oscila entre los 10 y 40 grados y el máximo entre los 40 y 120 grados centígrados de diferencia entre las temperaturas de consigna y las reales.**

ESTUDIO DE LA DISTRIBUCIÓN DEL ERROR

La distribución del error, va a **mostrarnos el grado de eficiencia del modelo actual y nos va a ayudar a detectar qué parámetros influyen y en qué grado en el mismo**.

Antes de estudiar este error, es conveniente ver si cumple una distribución normal o no. Para ellos nos ayudamos del comando "*hist*" del lenguaje de programación *R*.

```
# Calculo del error medio de cada bobina (TMPP2VALMED-TMPP2VALCNG)
error <- tapply(DatosBuenos$TMPP2VALME-DatosBuenos$TMPP2VALCN,
DatosBuenos$CODBOBINA,mean)

# Mostramos el histograma
hist(error,breaks=100,col=2)
```

Figura 159. Uso del comando "hist" para ver la distribución del error.

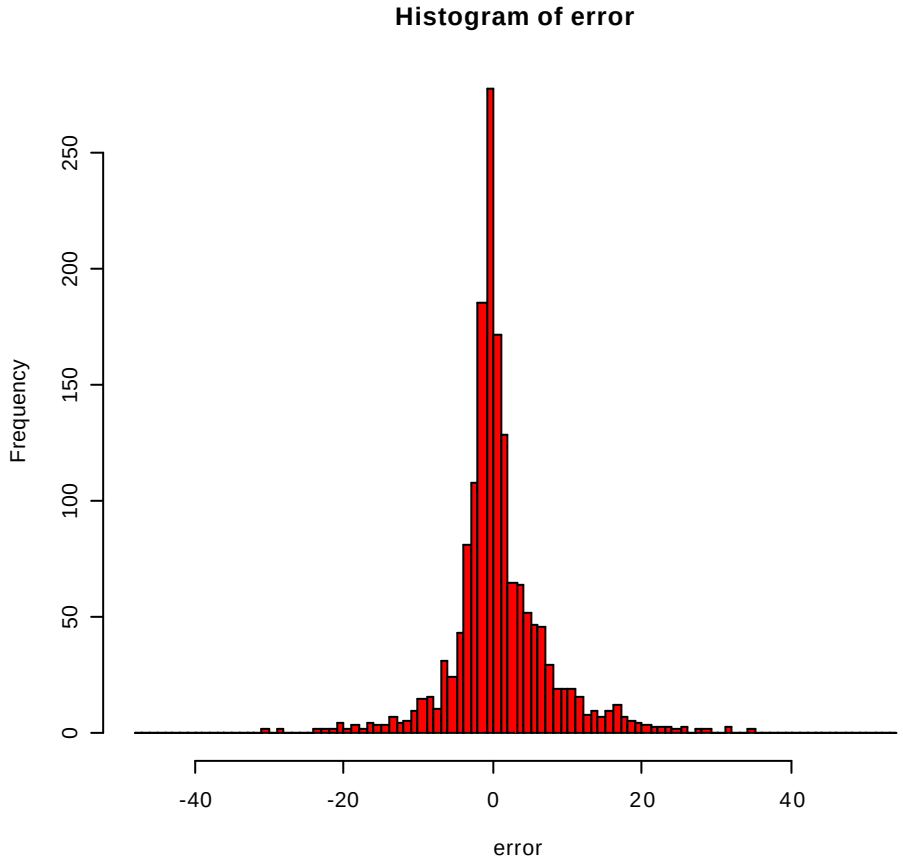


Figura 160. Histograma del error.

En la Figura 160 podemos apreciar claramente que el error cumple una distribución normal, tal y como se esperaba.

Esto garantiza los corolarios estadísticos que determinan que todo residuo de un modelo debe tener para todos sus ejes:

- Una Distribución Normal.
- Media Cero.
- Tener una varianza igual entre ejes.

Esto indica que el modelo actual **explica más o menos bien el sistema físico ya que en los residuos no se advierte ningún tipo de estructura no lineal adyacente.** Aún así, y como veremos en posteriores estudios, éste puede ser mejorado sustancialmente.

5.5.4.3 ESTUDIO DE LA EVOLUCIÓN DE LAS TEMPERATURAS DE CONSIGNA Y REALES DEL PIRÓMETRO 2 PARA CADA BOBINA

A continuación se muestra la evolución de las temperaturas de consigna y reales del pirómetro 2 para cada bobina.

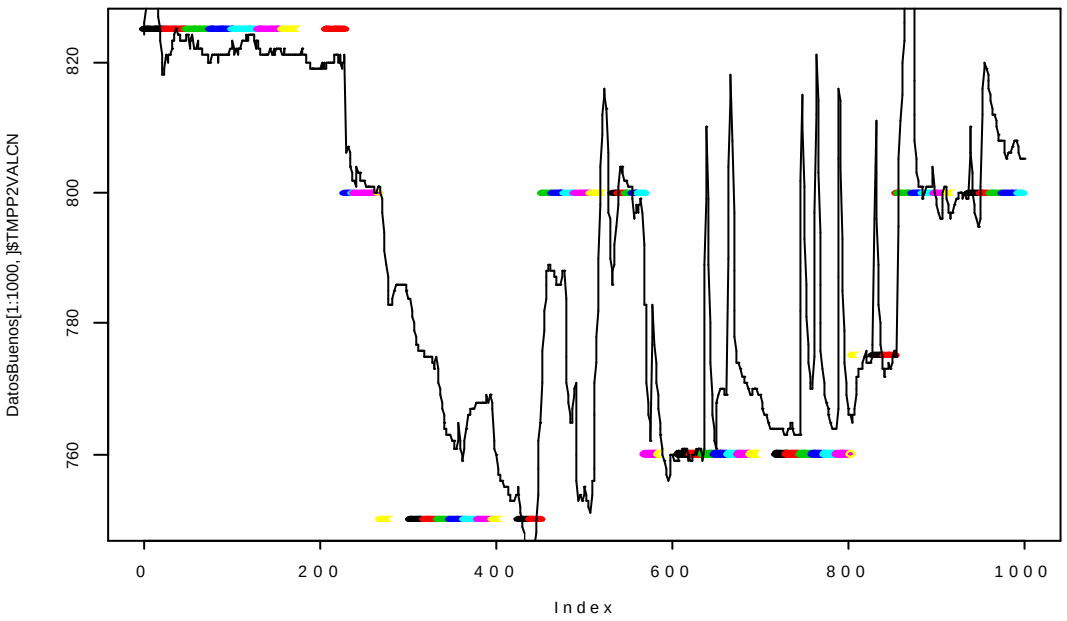


Figura 161. Temperaturas de consigna de la banda a la salida de la zona de calentamiento (TMPP2VALCN con puntos de tonalidad diferente por bobina) y temperatura real leída en el centro de la banda por el pirómetro 2 (TMPP2VALMED, línea continua).

En la Figura 162 podemos apreciar claramente, **que la temperatura objetivo de la banda a la salida de la zona de calentamiento del horno se mantiene constante en casi todas las**

bobinas, mientras que el valor real de la temperatura del centro de la banda leída por el pirómetro 2 oscila con errores más o menos pronunciados según la bobina estudiada.

```
# Dibujamos los puntos de consigna y el valor medio leido
plot (DatosBuenos[1:1000,]$TMPP2VALCN,col=DatosBuenos[1:1000,]$CODBOBINA)
lines (DatosBuenos[1:1000,]$TMPP2VALME,col=DatosBuenos[1:1000,]$CODBOBINA)
```

Figura 162. Programa que representa con puntos los valores de consigna y con líneas los reales.

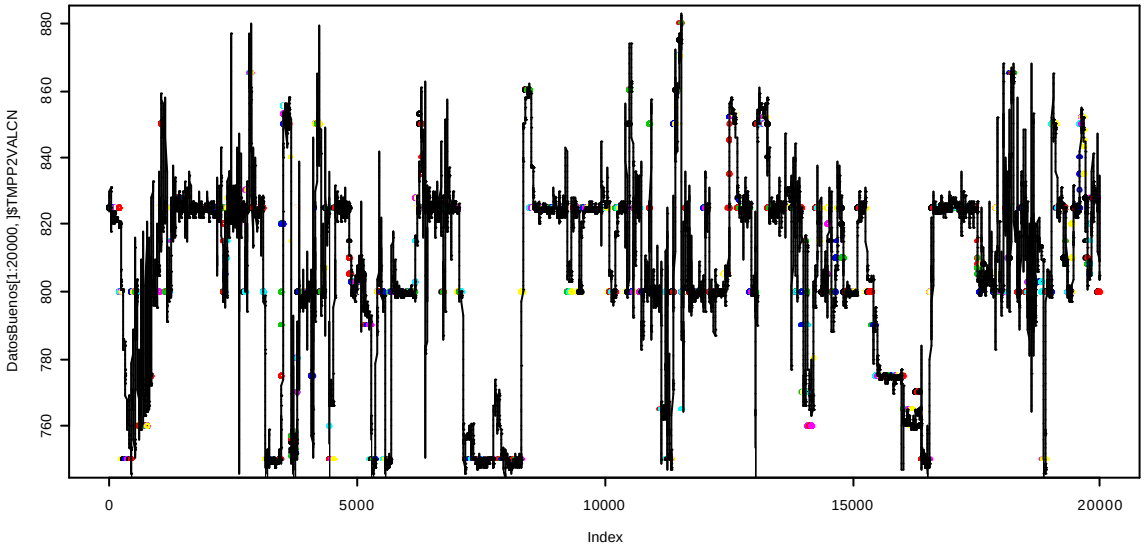


Figura 163. Representación de la evolución de los 20.000 primeros puntos (1078 bobinas) de temperatura de consigna y reales del pirómetro 2 de la zona de calentamiento del horno.

ANÁLISIS DE LA EVOLUCIÓN PARA ALGUNAS DE LAS BOBINAS

A continuación, y para comprender mejor la lectura que se obtiene de la temperatura de la banda de las bobinas a la salida de la zona de calentamiento, se procede a analizarlas en grupos de veinte.

En las gráficas siguientes **se observan diferentes tipos de comportamientos**. En algunos casos, la temperatura de la banda se aproxima con bastante precisión a la temperatura de consigna, sobre todo en aquellos casos en que la temperatura de consigna se mantiene constante en las bobinas anteriores y posteriores.

En otros casos, en cambio, cuando se produce una transición brusca de las temperaturas de consigna entre una bobina y la siguiente, **la temperatura de la banda varía más bruscamente**.

Por otro lado, existen casos en que el comportamiento de la temperatura de la banda **oscila gravemente** entre los valores de consigna aún cuando estos sean constantes en todas las bobinas.

```
# Obtenemos las bobinas
Bobinas <- tapply(DatosBuenos$CODBOBINA,DatosBuenos$CODBOBINA,max)

# Dibujamos las bobinas desde la 'bobini' hasta 'bobini+20'
bobini <- 40
plot(DatosBuenos[CODBOBINA>Bobinas[bobini] &
CODBOBINA<Bobinas[bobini+20],]$TMPP2VALME,col=DatosBuenos[CODBOBINA>Bobinas[bobi
ni] & CODBOBINA<Bobinas[bobini+20],]$CODBOBINA);
lines(DatosBuenos[CODBOBINA>Bobinas[bobini] &
CODBOBINA<Bobinas[bobini+20],]$TMPP2VALME);points(DatosBuenos[CODBOBINA>Bobinas[
bobini] &
CODBOBINA<Bobinas[bobini+20],]$TMPP2VALCN,col=DatosBuenos[CODBOBINA>Bobinas[bobi
ni] & CODBOBINA<Bobinas[bobini+20],]$CODBOBINA)

# Obtenemos el código de la bobina primera y la ultima dibujadas
Bobinas[bobini]
      11523042
Bobinas[bobini+20]
      11523062
```

Figura 164. Programa que nos permite visualizar las temperaturas reales de salida de la zona de calentamiento del horno de las bobinas deseadas.

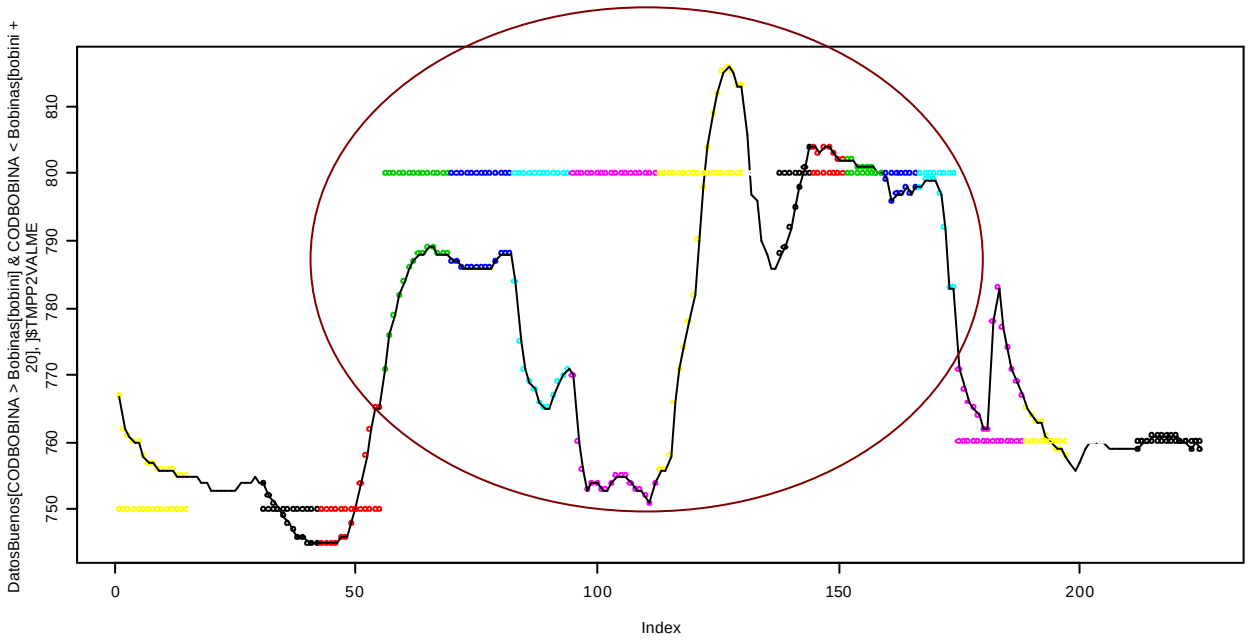


Figura 165. Evolución de la temperatura de las bobinas 11523022 a 11523042.

En la Figura 165 vemos que cuando se produce un cambio brusco de las temperaturas de consigna, **la temperatura de las bobinas posteriores oscilan con errores de hasta 50 grados hasta que se estabiliza después de pasadas siete bobinas**.

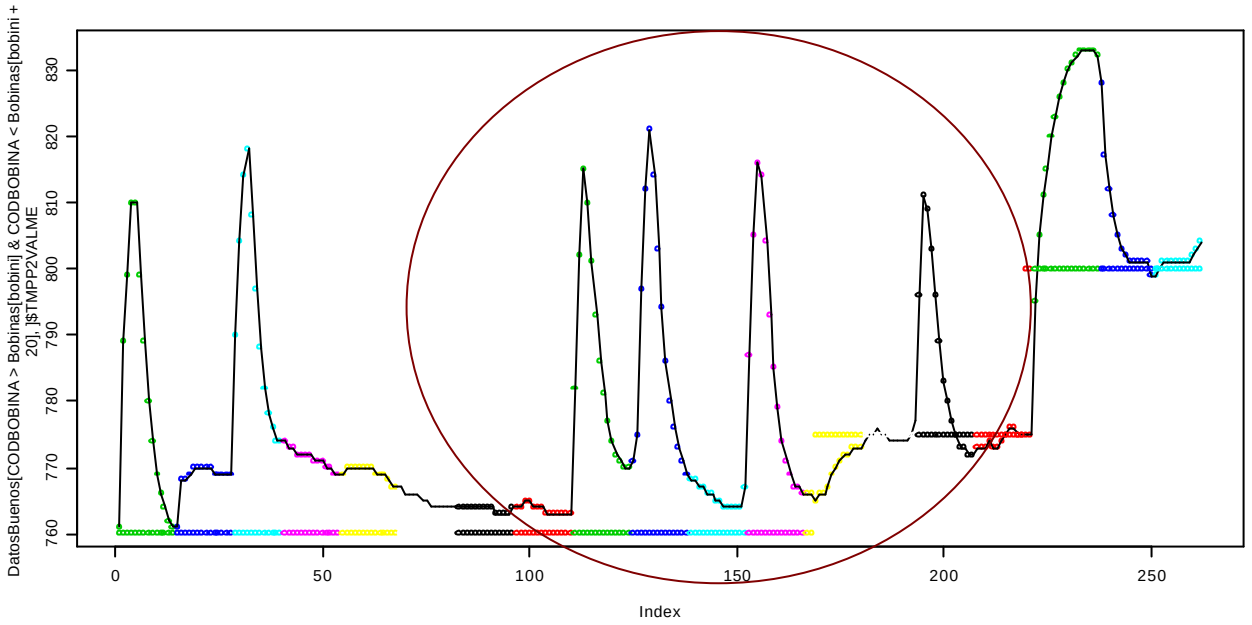


Figura 166. Evolución de la temperatura de las bobinas 11523042 a 11523062.

En la Figura 166 **se ve un comportamiento "extraño" de la temperatura del pirómetro con picos bastante elevados**, aún cuando la temperatura de consigna se mantiene constante en diez bobinas. Seguramente podrá ser achacado a un uso en "modo manual".

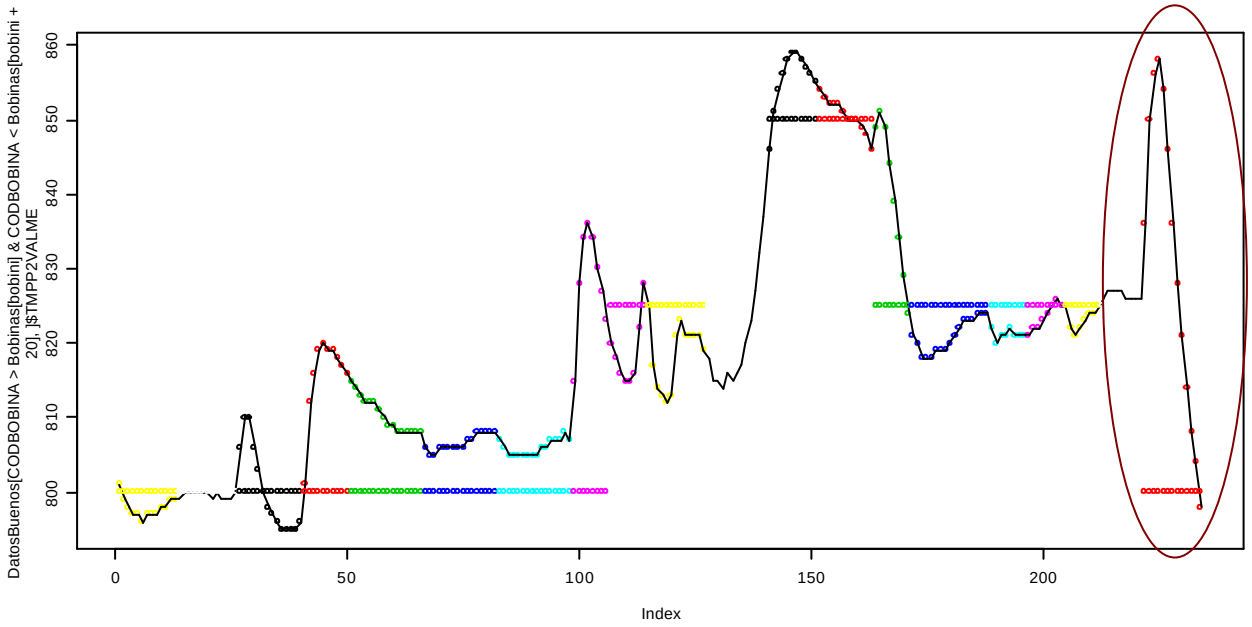


Figura 167. Evolución de la temperatura de las bobinas 11523062 a 11523082.

También, en la Figura 167 vemos como el comportamiento es bastante bueno hasta la última bobina, en donde **la transición brusca de las temperaturas de consigna produce un crecimiento del error bastante pronunciado**.

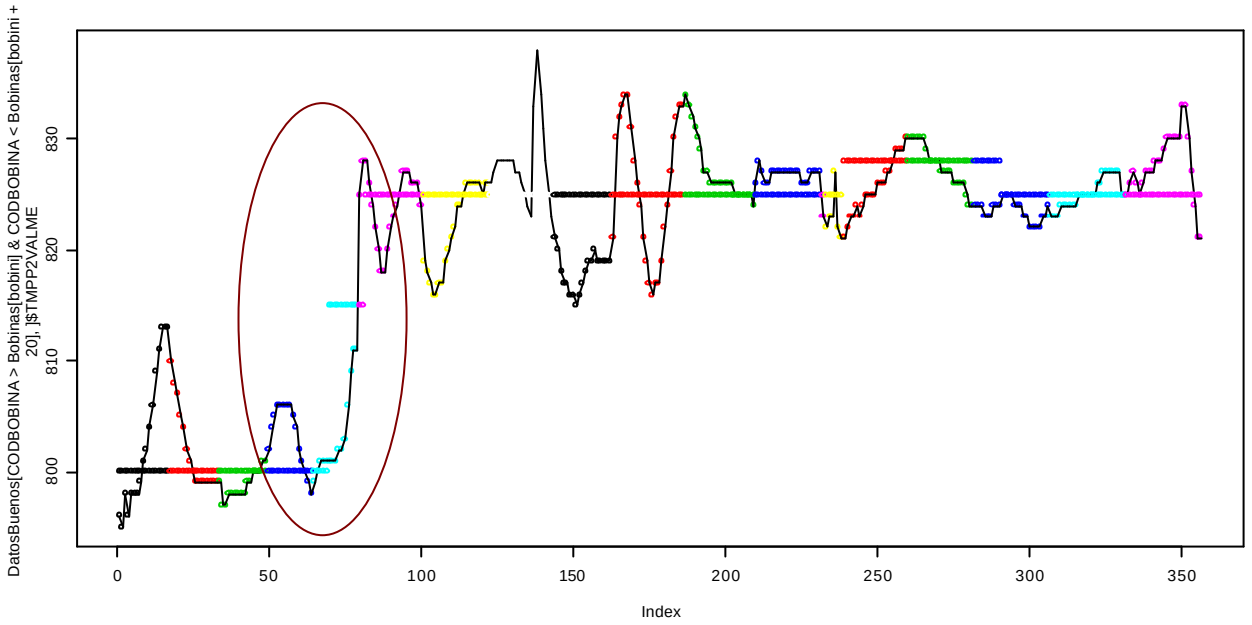


Figura 168. Evolución de la temperatura de las bobinas 11523083 a 11533023.

En la Figura 168 se muestra **un comportamiento ideal de la temperatura de la banda**. Claramente se puede apreciar en la transición**, cómo el escalonamiento de las temperaturas de consigna de esa bobina, produce un crecimiento progresivo y adecuado de las temperaturas de las siguientes bobinas.**

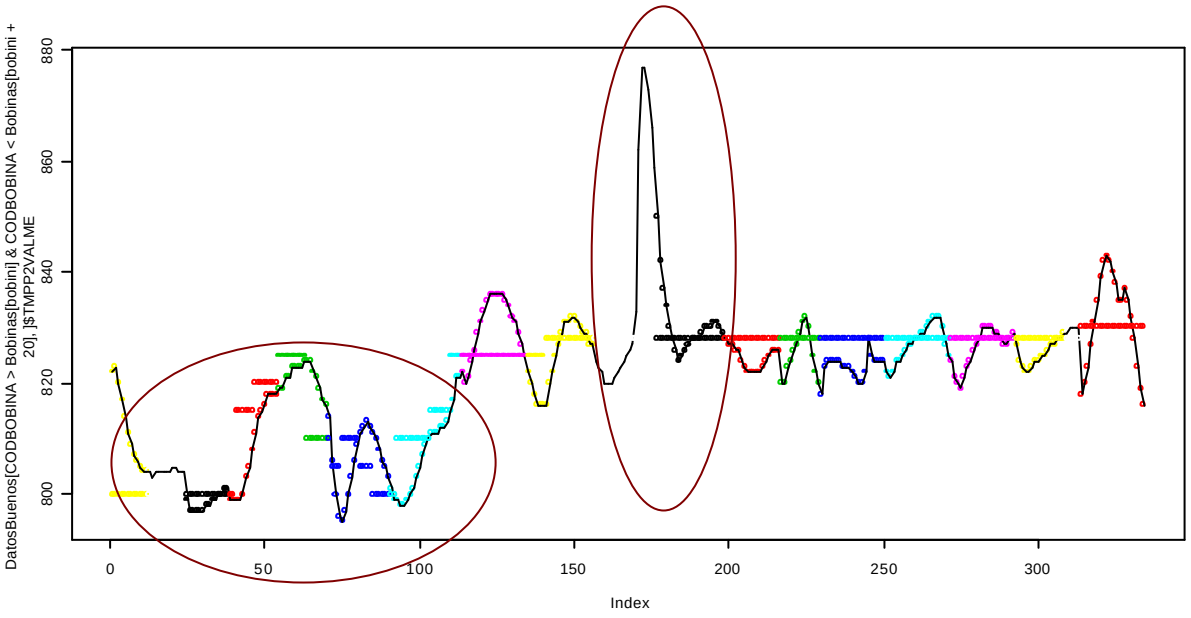


Figura 169. Evolución de la temperatura de las bobinas 11533056 a 11543019.

En la Figura 169 vemos otro caso en que las temperaturas de consignas de varias bobinas han sido escalonadas, y cómo el comportamiento de la temperatura de la banda oscila cerca de las mismas.

Por otro lado, vemos que aún así, en una de las bobinas se produce un pico de temperatura que inicialmente resulta inexplicable.

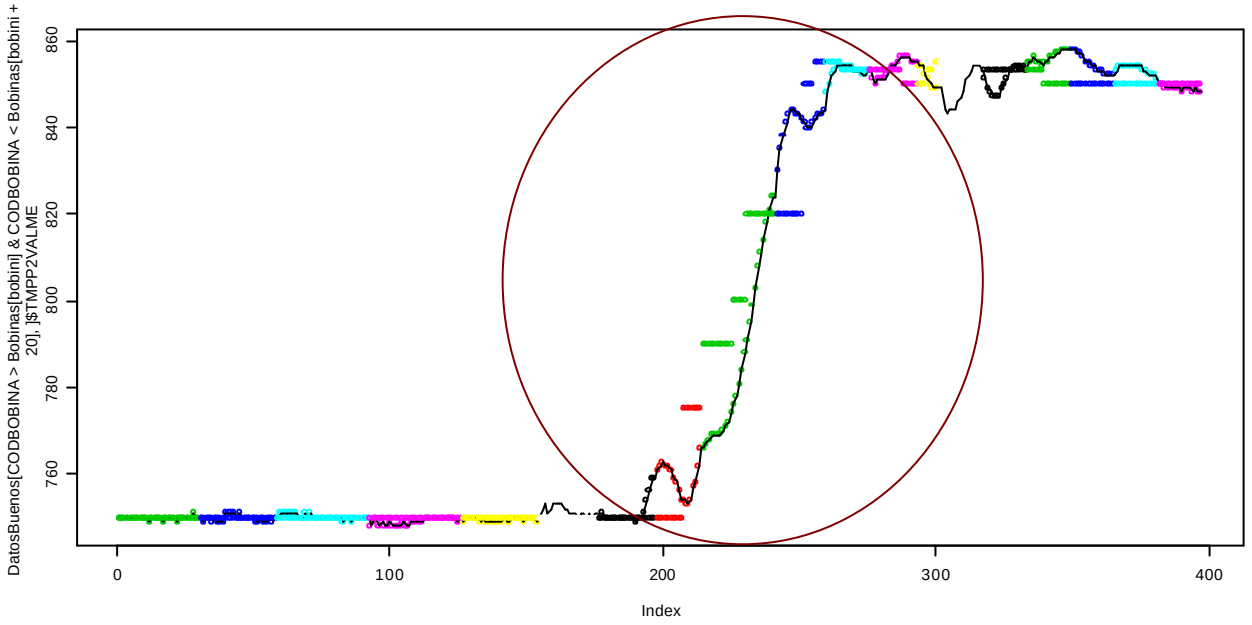


Figura 170. Evolución de la temperatura de las bobinas 11543066 a 11553015.

Por último, la Figura 170 muestra como la transición, con una diferencia de temperaturas de consigna de 100 grados centígrados, se produce progresivamente y sin cambios bruscos de temperatura.

CONCLUSIONES DEL ANÁLISIS

De las figuras anteriores, podemos concluir **que muchos de los "errores" se producen en transiciones bruscas de temperaturas de consigna, aunque, existen muchos otros que no pueden ser explicados solamente con el análisis de esas dos variables. Volvemos a ver, la necesidad de disponer de una variable que indique si el sistema está trabajando en "modo manual" o en "modo automático".**

Lógicamente, habrá otros tipos de parámetros que afecten a la temperatura de la banda: velocidad de la misma, temperatura del horno, dimensiones de la banda, etc.; **que tendrán que ser considerados para poder explicar correctamente el comportamiento de la temperatura de salida de la banda**. Estos estudios se realizarán en capítulos posteriores.

5.5.4.4 CARACTERIZACIÓN DE LA EVOLUCIÓN DEL ERROR Y TEMPERATURAS DE LOS PIRÓMETROS

Para caracterizar el comportamiento de las temperaturas leídas por los pirómetros y del error, se propone la generación de una serie de nuevas variables por bobina que tratarán de describir la forma de la curva para cada bobina.

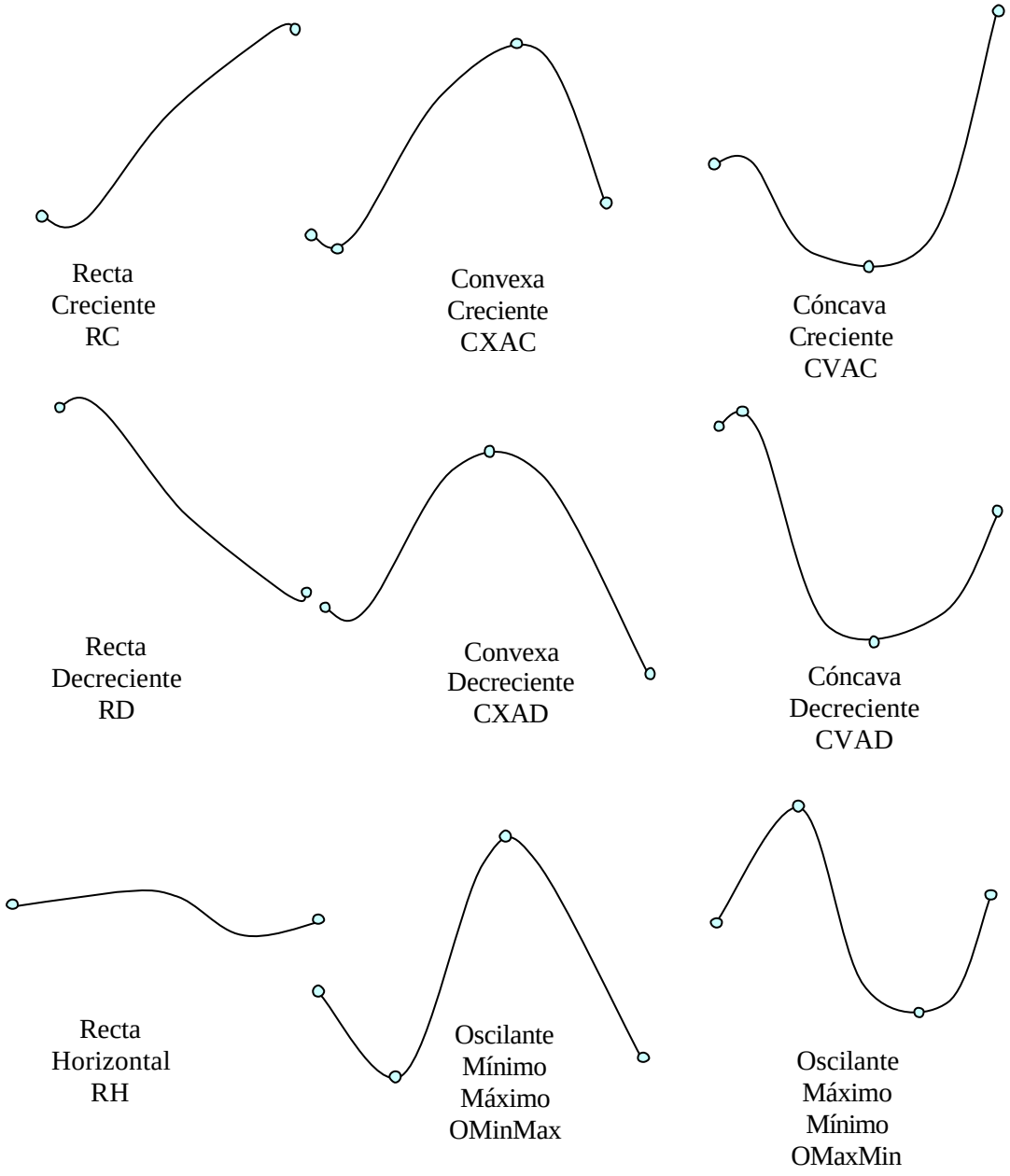


Figura 171. Clasificación de las curvas de temperatura de los pirómetros y del error.

También se considera oportuno la creación de unas variables que indiquen la evolución de las temperaturas en las bobinas anteriores y posteriores.

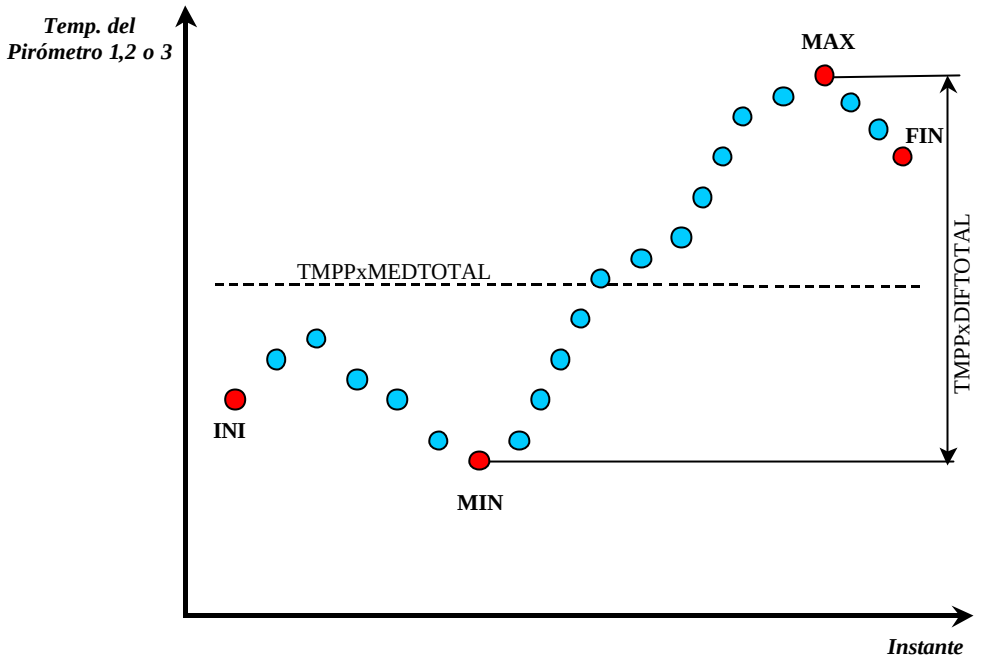


Figura 172. Variables que caracterizan la curva de temperatura por bobina de cada pirómetro

TEMPERATURAS LEÍDAS DE LOS PIRÓMETROS 1, 2 Y 3

Las Variables Nuevas a Generar son las siguientes:

- **Temperatura del pirómetro media**, de todos los instantes, (*TMPPxMEDTOTAL*) para cada bobina.
- **Diferencia entre el valor máximo y el mínimo** de la temperatura del pirómetro para cada bobina (*TMPPxDIFTOTAL*).
- **Tipo de curva** (*TIPOCURVATMPPx*).

Donde *x* representa el número del pirómetro que obtiene la temperatura de la banda.

El tipo de curva para cada bobina se designa a partir de los siguientes parámetros:

- Valor de la diferencia entre el valor máximo y el mínimo (ver Figura 172).
- Orden en que aparecen los valores Máximo y Mínimo según el tiempo.
- Si la distancia de la temperatura máxima o mínima a las temperaturas de los puntos iniciales y finales sobrepasa ¼ de la distancia entre el máximo o el mínimo según el orden en que estén los mismos.

Es decir, tal y como muestra la Tabla 25, la curva *MCXAD* (MEDIA CONVEXA DECRECIENTE) vendrá determinada por aquella cuya *TMPPxDIFTOTAL* esté entre 20 y 40 grados centígrados, que primero aparezca el valor máximo y después el mínimo, y que la diferencia entre el valor máximo y el valor del punto inicial sea mayor de ¼ del valor de *TMPPxDIFTOTAL*.

Valor de TMPPxDIFTOTAL	Aparece Primero	Aparece Segundo	ABS(T(INI)-T(1º)) > ¼ de TMPPxDIFTOTAL	ABS(T(FIN)-T(2º)) > ¼ de TMPPxDIFTOTAL	VALOR DE TIPOCURVATMPPx
<= 10ºC	-	-	Indiferente	Indiferente	HORIZONTAL (H)
>10ºC y <=20ºC	MIN	MAX	Indiferente	Indiferente	BAJA RECTA CRECIENTE (BRC)
>10ºC y <=20ºC	MAX	MIN	Indiferente	Indiferente	BAJA RECTA DECRECIENTE (BRD)
>20ºC y <=40ºC	MIN	MAX	NO	NO	MEDIA RECTA CRECIENTE (MRC)
>20ºC y <=40ºC	MAX	MIN	NO	NO	MEDIA RECTA DECRECIENTE (MRD)
>20ºC y <=40ºC	MIN	MAX	SI	NO	MEDIA CÓNCAVA CRECIENTE (MCVAC)
>20ºC y <=40ºC	MAX	MIN	SI	NO	MEDIA CONVEXA DECRECIENTE (MCXAD)
>20ºC y <=40ºC	MIN	MAX	NO	SI	MEDIA CONVEXA CRECIENTE (MCXAC)
>20ºC y <=40ºC	MAX	MIN	NO	SI	MEDIA CÓNCAVA DECRECIENTE (MCVAD)
>20ºC y <=40ºC	MIN	MAX	SI	SI	MEDIA OSCILANTE MINIMO MÁXIMO (MOMINMAX)
>20ºC y <=40ºC	MAX	MIN	SI	SI	MEDIA OSCILANTE MÁXIMO MINIMO (MOMAXMIN)
>40ºC y <=200ºC	MIN	MAX	NO	NO	ALTA RECTA CRECIENTE (ARC)
>40ºC y <=200ºC	MAX	MIN	NO	NO	ALTA RECTA DECRECIENTE (ARD)
>40ºC y <=200ºC	MIN	MAX	SI	NO	ALTA CÓNCAVA CRECIENTE (ACVAC)
>40ºC y <=200ºC	MAX	MIN	SI	NO	ALTA CONVEXA DECRECIENTE (ACXAD)
>40ºC y <=200ºC	MIN	MAX	NO	SI	ALTA CONVEXA CRECIENTE (ACXAC)
>40ºC y <=200ºC	MAX	MIN	NO	SI	ALTA CÓNCAVA DECRECIENTE (ACVAD)
>40ºC y <=200ºC	MIN	MAX	SI	SI	ALTA OSCILANTE MINIMO MÁXIMO (AOMINMAX)
>40ºC y <=200ºC	MAX	MIN	SI	SI	ALTA OSCILANTE MÁXIMO MINIMO (AOMAXMIN)
>200ºC	-	-	-	-	ERROR (E)

Tabla 25. Valores que se asignarán a TIPOCURVATMPPx según la distancia de los máximos y mínimos a los puntos inicial o final y del valor de la diferencia entre el valor máximo y mínimo.

TEMPERATURA DE CONSIGNA DEL PIRÓMETRO 2

Igual que en el apartado anterior, las variables se denominarán:

- **Temperatura del pirómetro media** de consigna, de todos los instantes, (*TMPP2CNGMEDTOTAL*) para cada bobina.
- **Diferencia entre el valor máximo y el mínimo** de la temperatura de consigna del pirómetro para cada bobina (*TMPP2CNGDIFTOTAL*).
- **Tipo de curva** (*TIPOCURVATMPP2CNG*) que se definen igual que *TIPOCURVATMPPx*

La creación de estas nuevas variables servirá para reducir el número de datos a manipular en procesos posteriores.

EVOLUCIÓN DEL ERROR ENTRE LA TEMPERATURA DE CONSIGNA Y REAL DEL PIRÓMETRO DOS

También se caracteriza la curva de la diferencia entre el valor de consigna del pirómetro 2 (temperatura de banda deseada para cada instante a la salida de la zona de calentamiento) y la real:

- **Error medio del pirómetro**: media de todos los instantes (*ERRORMEDTOTAL*) para cada bobina y error absoluto (*ERRORMEDTOTALABS*).
- **Diferencia entre el valor máximo y el mínimo del error** del pirómetro para cada bobina (*ERRORDIFTOTAL*).
- **Tipo de curva** (*TIPOCURVAERROR*) que se definen igual que *TIPOCURVATMPPx*

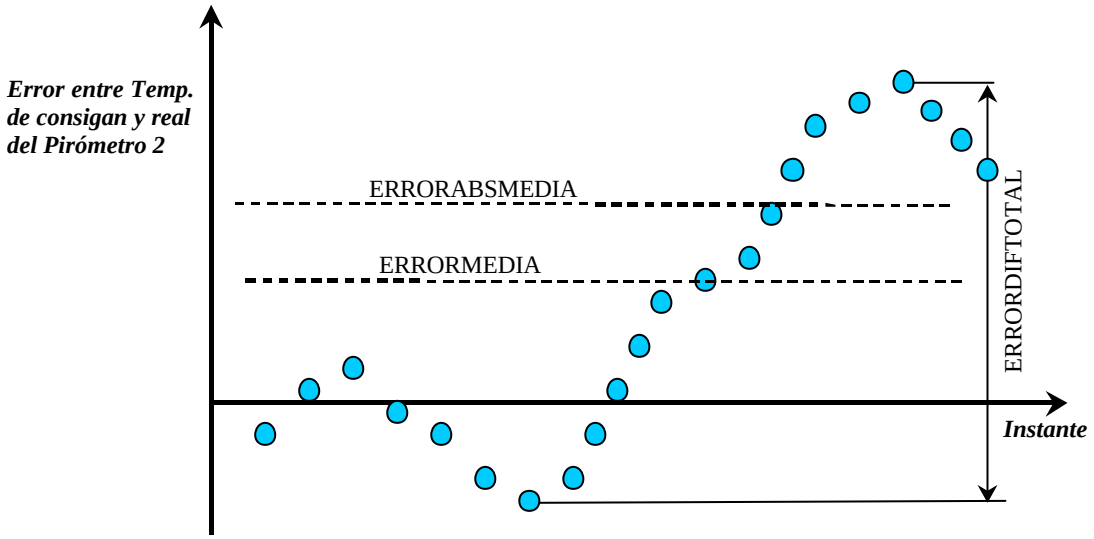


Figura 173. Caracterización del error entre la temperatura leída del pirómetro dos y la de consigna.

5.5.5 ESTUDIO EXPLORATORIO DE LAS VELOCIDADES DE LA BANDA

El estudio se realiza a partir del análisis de la variable *VELCENMED* de la tabla "*acumuladore*".

El valor que indica esta variable, corresponde con el valor de la velocidad de la banda dentro del horno en metros por minuto. Esta medida se realiza en el centro de la línea de galvanizado.

```
# Realizamos un attachement de la matriz DATBUENOS
attach(T100CALB)

# Obtenemos las bobinas
CODBOB <- tapply(T100CALB$CODBOBINA,T100CALB$CODBOBINA,min)

# Cargamos la Librería RODBC
library(RODBC);

# Abrimos el Canal de comunicación con la base de datos
canal <- odbcConnect("aceralia3","","","localhost");

# Cargamos la variable VELCENMED (velocidad de la banda)
T100CALACUM <- sqlQuery(canal,"SELECT CODBOBINA, VELCENVALMED as VELCENMED FROM
acumuladore");

# Eliminamos las bobinas quitadas en trabajos anteriores
T100CALACUMBIEN <- T100CALACUM [(T100CALACUM$CODBOBINA %in% CODBOB),]

# Estudiamos la distribución de las velocidades
table(T100CALACUMBIEN$VELCENMED)

   -1     0    10    12    13    14    15    16    17    18    19    20    21
34155   482    32     1     6     6    12     5     5     5     9    10     5
   22    23    24    25    26    27    28    29    30    31    32    33    34
   15    88    19   102    23    19    20    15   147    76    24    20    23
   35    36    37    38    39    40    41    42    43    44    45    46    47
   88    24    32    48    80  1556    87    44    44    63   138    41    65
   48    49    50    51    52    53    54    55    56    57    58    59    60
   59    60   713    96    49   108    80  1372   305   563   535   558  5658
   61    62    63    64    65    66    67    68    69    70    71    72    73
  287  1021  1823   853  4069  1559   928   370   297  5182   492  1533   885
   74    75    76    77    78    79    80    81    82    83    84    85    86
  705  4625   303   490  1645   818  8365  1449   607   613   552  5358   337
   87    88    89    90    91    92    93    94    95    96    97    98    99
 2488  1069   476  7743   349  2018  1161  1016  4938   974  1076  1118   556
  100   101   102   103   104   105   106   107   108   109   110   111   112
13466   288   528   802   203  3781   193   983   809   286 11719   491  1643
  113   114   115   116   117   118   119   120   121   122   123   124   125
  657   920  9907   396  1391   941   476 15665   460   844   848   557  9036
  126   127   128   129   130   131   132   133   134   135   136   137   138
 1910  1621  1669  1698 22785   398  1603  1360  1727 10743  1392  4375  3644
  139   140   141   142   143   144   145   146   147   148   149   150   314
  946 42560   342   822   613   627 15729  3807  1338   854   667 35427     2
```

Figura 174. Programa que obtiene las velocidades de la base de datos.

A continuación vamos a explorar las curvas de velocidades por bobina.

```
# Obtenemos las bobinas
Bobinas <- tapply(T100CALACUMBIEN$CODBOBINA,T100CALACUMBIEN$CODBOBINA,max)

# Dibujamos las bobinas desde la 'bobini' hasta 'bobini+20'
bobini <- 1
plot(T100CALACUMBIEN[CODBOBINA>=Bobinas[bobini] &
CODBOBINA<Bobinas[bobini+20],]$VELCENMED,ylim=c(0,155),col=T100CALACUMBIEN[CODBO
BINA>=Bobinas[bobini] & CODBOBINA<Bobinas[bobini+20],]$CODBOBINA);
lines(T100CALACUMBIEN[CODBOBINA>=Bobinas[bobini] &
CODBOBINA<Bobinas[bobini+20],]$VELCENMED);

# Obtenemos el código de la bobina primera y la ultima dibujadas
Bobinas[bobini]
      20103001
Bobinas[bobini+20]
      20103021
```

Figura 175. Programa que sirve para representar en gráficas las curvas de velocidades.

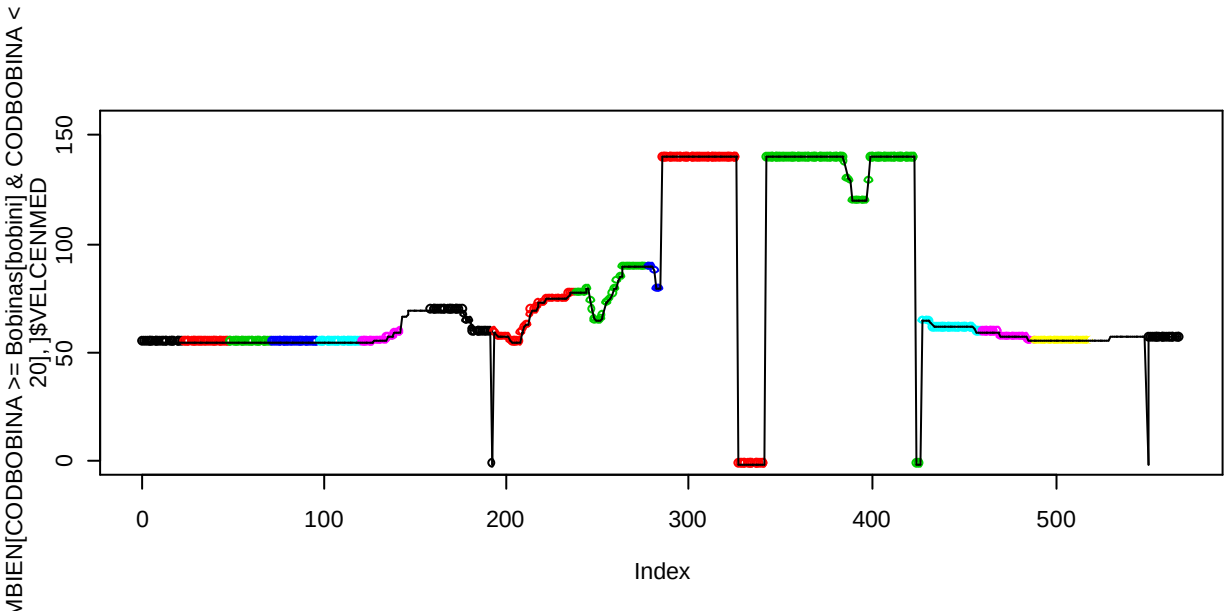


Figura 176. Curvas de velocidades de las bobinas 20103001 a 20103021.

Estudiando los resultados de la distribución de las velocidades dada por el programa de la Figura 175 y observando detenidamente la Figura 176 y siguientes, podemos ver claramente que:

- El sensor que determina la velocidad de la banda o los sistemas de captura que almacenan el valor, **generan bastante ruido (34.155 valores a –1 de 353.887, es decir un 9,7% de los datos de velocidad)** . Estos espurios será necesario tomar en cuenta y eliminarlos cuando sea necesario.

- Existe una gran mayoría de curvas de velocidades con un comportamiento horizontal, aunque entre transiciones aparecen diferentes tipos de curvas. Por lo tanto, se estima conveniente caracterizar las curvas de velocidades con el mismo método utilizado en apartados anteriores.

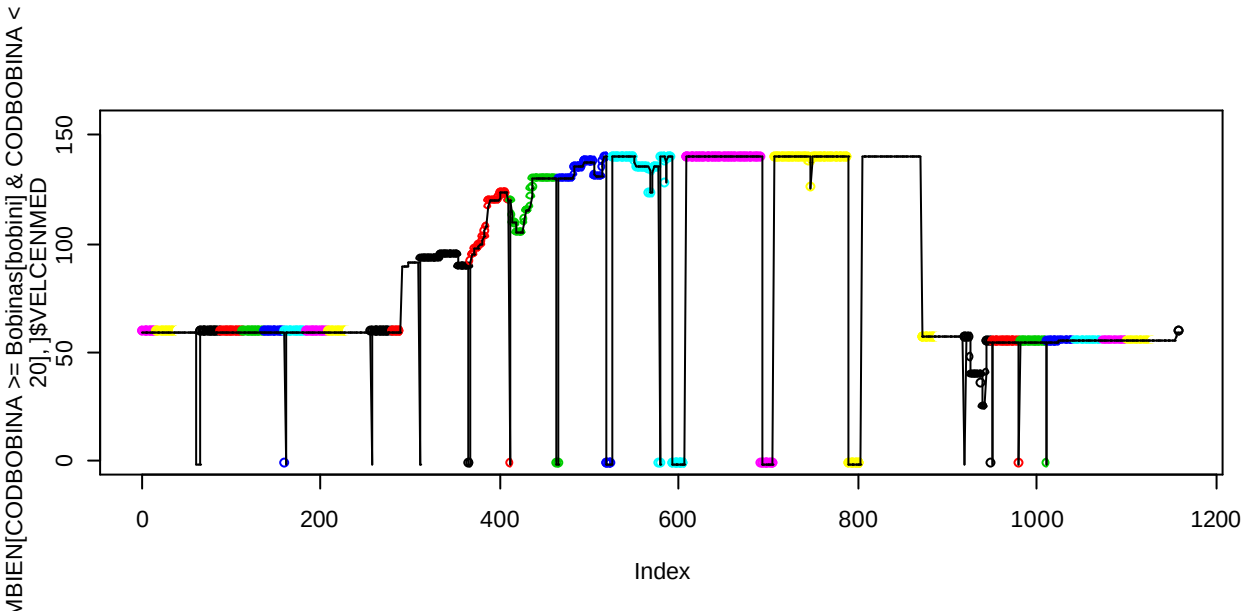


Figura 177. Curvas de velocidades de las bobinas 20103041 a 20103061.

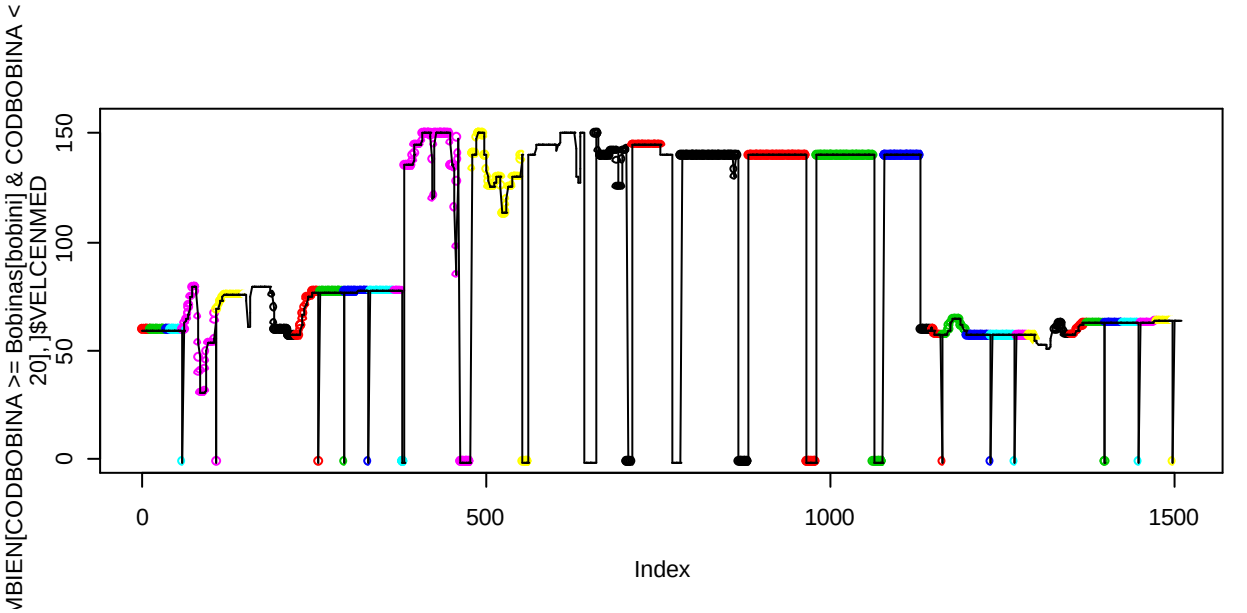


Figura 178. Curvas de velocidades de las bobinas 20103061 a 20103081.

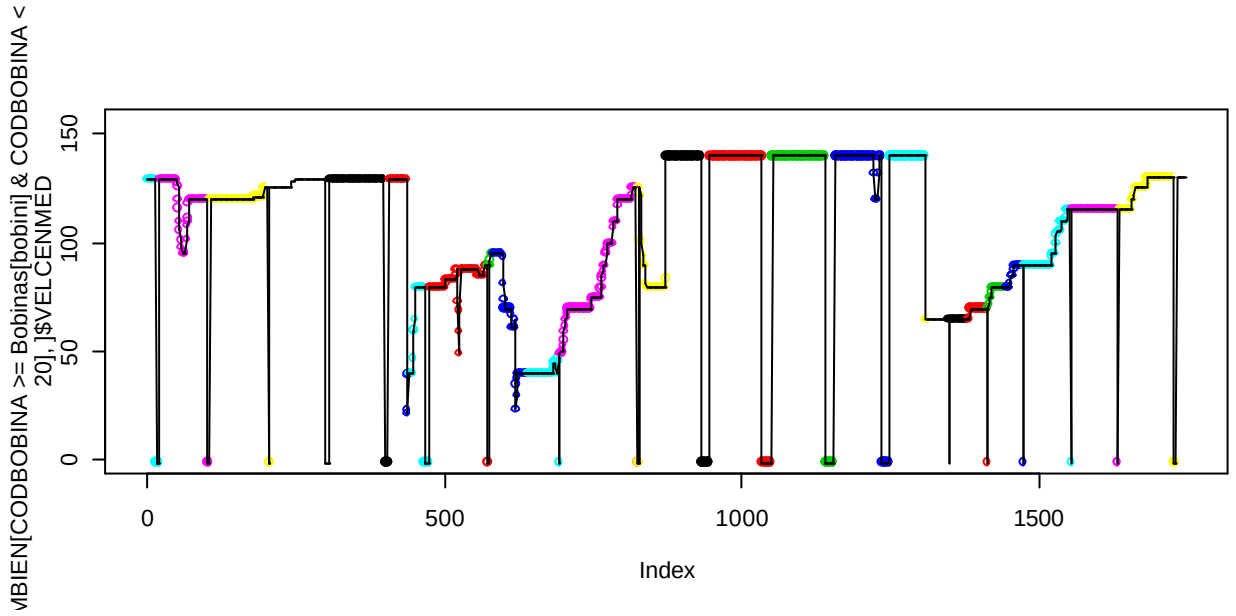


Figura 179. Curvas de velocidades de las bobinas 20133084 a 20143018.

5.5.5.1 CARACTERIZACIÓN DE LA CURVA DE LA VELOCIDAD DE LA BANDA

Para caracterizar el comportamiento de la curva de velocidades de cada bobina, se plantean las siguientes variables:

Igual que en el apartado anterior, las variables se denominarán:

- **Velocidad media** de todos los instantes, (*VELMEDTOTAL*) para cada bobina.
- **Diferencia entre el valor máximo y el mínimo** de la velocidad para cada bobina (*VELDIFTOTAL*).
- **Tipo de curva** (*TIPOCURVAVEL*) que se definen igual que las curvas anteriores, excepto por el margen de velocidades que indican el tamaño de la curva (horizontal, baja, media y alta) ya que habrá que acomodarlo a la nueva magnitud. Para ello, se estudia el histograma de diferencias entre máximo y mínimo de velocidades (ver Figura 181) y se divide siguiendo unos criterios lo más lógicos posibles.

```
# Determinamos qué bobinas tienen todos a –1 o a 0
# Obtenemos el máximo de velocidad de cada bobina
MAXVEL <- tapply(T100CALACUMBIEN$VELCENMED,T100CALACUMBIEN$CODBOBINA,max)

# Detectamos cuantas bobinas tienen a –1 o a 0 todos los valores
# Vemos que son 3 con todo (-1) y 8 con todo (0)
table(MAXVEL)
MAXVEL
 -1   0  40  41  44  45  46  47  48  50  51  53  54  55  56  57  58  59  60  61
  3   8  10   1   1   2   1   1   1  10   1   2   1  35   6  18   8  12 181   6
 62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81
 41  72  37 168  63  46  13   4 148  13  43  23  17 130   7  13  45  16 170  43
 82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101
 18  14  18 151  10  59  19  11 156   9  50  19  20 125  22  24  22  16 236   7
102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
 11  14   2  88   1  23  17   4 245   5  30  15  22 182  12  29  13   6 278   4
122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141
 16  12   7 152  31  29  34  26 369   6  18  28  34 214  21  70  60   8 632   5
142 143 144 145 146 147 148 149 150 314
 11   6   6 261  46  20  17   5 552   2

# Vemos que bobinas tienen todas las velocidades a cero o a -1
bobmalas <- c(row.names(as.matrix(MAXVEL[MAXVEL %in% -
1])),row.names(as.matrix(MAXVEL[MAXVEL %in% -0])))

# Ponemos a 100 las velocidades de esas bobinas
T100CALACUMBIEN[T100CALACUMBIEN$CODBOBINA %in% bobmalas,]$VELCENMED <- 100

# Eliminamos las observaciones de velocidad que son menores de 10
DATBUENOS<- T100CALACUMBIEN[T100CALACUMBIEN$VELCENMED>10,]

############################################################
# Caracterización de la curva de velocidades
############################################################

# Obtenemos el valor de velocidad máximo y mínimo por bobina
MINTMPP <- tapply(DATBUENOS$VELCENMED,DATBUENOS$CODBOBINA,min)
MAXTMPP <- tapply(DATBUENOS$VELCENMED,DATBUENOS$CODBOBINA,max)

# Obtenemos el valor medio de velocidad de cada bobina
VALMEAN <- tapply(DATBUENOS$VELCENMED,DATBUENOS$CODBOBINA,mean)

# Obtenemos las variables finales
VELMEDTOTAL <- round(VALMEAN)
VELDIFTOTAL <- MAXTMPP-MINTMPP

#Obtenemos el histograma de las diferencias de velocidades
hist(VELDIFTOTAL,breaks=15,col=2)
```

Figura 180. Programa que calcula las variables VELMEDTOTAL y VELDIFTOTAL.

Valor de VELDIFTOTAL	Aparece Primero	Aparece Segundo	ABS(VEL(INI)-VEL(1º)) > ¼ de VELxDIFTOTAL	ABS(VEL(FIN)-VEL(2º)) > ¼ de VELxDIFTOTAL	VALOR DE TIPOCURVAVEL
<= 10	-	-	Indiferente	Indiferente	HORIZONTAL (H)
>10 y <=20	MIN	MAX	Indiferente	Indiferente	BAJA RECTA CRECIENTE (BRC)
>10 y <=20	MAX	MIN	Indiferente	Indiferente	BAJA RECTA DECRECIENTE (BRD)
>20 y <=50	MIN	MAX	NO	NO	MEDIA RECTA CRECIENTE (MRC)
>20 y <=50	MAX	MIN	NO	NO	MEDIA RECTA DECRECIENTE (MRD)
>20 y <=50	MIN	MAX	SI	NO	MEDIA CÓNCAVA CRECIENTE (MCVAC)
>20 y <=50	MAX	MIN	SI	NO	MEDIA CONVEXA DECRECIENTE (MCXAD)
>20 y <=50	MIN	MAX	NO	SI	MEDIA CONVEXA CRECIENTE (MCXAC)
>20 y <=50	MAX	MIN	NO	SI	MEDIA CÓNCAVA DECRECIENTE (MCVAD)
>20 y <=50	MIN	MAX	SI	SI	MEDIA OSCILANTE MINIMO MÁXIMO (MOMINMAX)
>20 y <=50	MAX	MIN	SI	SI	MEDIA OSCILANTE MÁXIMO MINIMO (MOMAXMIN)
>50 y <=200	MIN	MAX	NO	NO	ALTA RECTA CRECIENTE (ARC)
>50 y <=200	MAX	MIN	NO	NO	ALTA RECTA DECRECIENTE (ARD)
>50 y <=200	MIN	MAX	SI	NO	ALTA CÓNCAVA CRECIENTE (ACVAC)
>50 y <=200	MAX	MIN	SI	NO	ALTA CONVEXA DECRECIENTE (ACXAD)
>50 y <=200	MIN	MAX	NO	SI	ALTA CONVEXA CRECIENTE (ACXAC)
>50 y <=200	MAX	MIN	NO	SI	ALTA CÓNCAVA DECRECIENTE (ACVAD)
>50 y <=200	MIN	MAX	SI	SI	ALTA OSCILANTE MINIMO MÁXIMO (AOMINMAX)
>50 y <=200	MAX	MIN	SI	SI	ALTA OSCILANTE MÁXIMO MINIMO (AOMAXMIN)
>200	-	-	-	-	ERROR (E)

Tabla 26. Valores que se asignarán a TIPOCURVAVEL según la distancia de los máximos y mínimos a los puntos inicial o final y del valor de la diferencia entre el valor máximo y mínimo.

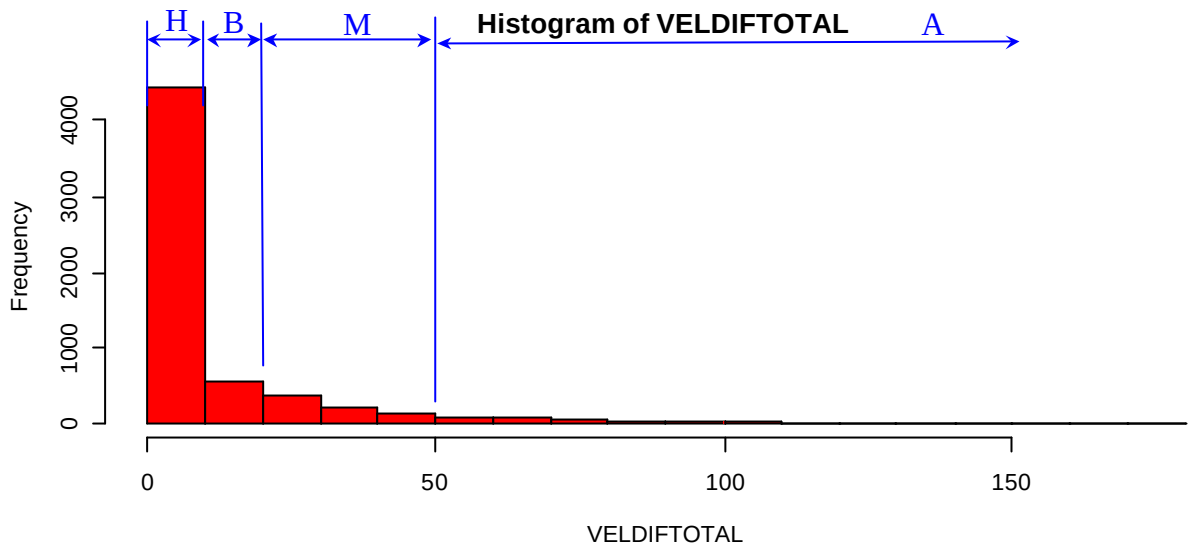


Figura 181. Selección del tipo de curva según el histograma de las diferencias entre velocidad máxima y mínima.

En la Figura 181, se muestra el histograma que permite clasificar las diferencias entre la velocidad máxima y mínima de cada curva según el siguiente esquema:

- Horizontal (H): Cuando es menor o igual de 10.
- Baja (B): Mayor de 10 y menor o igual de 20.
- Media (M): Mayor de 20 y menor o igual de 50.
- Alta (A): Mayor de 50 y menor o igual de 200.

La Tabla 26 utiliza los márgenes arriba descritos para caracterizar las curvas de velocidad.

5.5.6 ESTUDIO EXPLORATORIO DE LOS ESPESORES DE LAS BOBINAS

Por último, se procede a analizar las variables correspondientes a los espesores de las bandas antes y después del galvanizado.

Para ello, se obtienen las variables *ESPESOR* de la tabla "*dps*" (la llamaremos *ESPESOR_INI*), que corresponde al espesor objetivo de fabricación de la bobina antes del galvanizado; y la variable *ESPTOTVALMED* de la tabla *"tijera"* (la llamaremos *ESPESOR_FIN*), que es medida por una galga de rayos x, después del proceso de galvanizado y que indica el espesor total de la bobina junto con la capa que la recubre.

```
 load("T100CALB.Rdata");
# Realizamos un attachement de la matriz DATBUENOS
attach(T100CALB)

# Obtenemos las bobinas
CODBOB <- tapply(T100CALB$CODBOBINA,T100CALB$CODBOBINA,min)

# Cargamos la Librería RODBC
library(RODBC);

# Abrimos el Canal de comunicación con la base de datos
canal <- odbcConnect("aceralia2","","","localhost");

# Cargamos la variable ESPE_INI (Espesor SIN recubrimiento)
T100ESPE_INI <- sqlQuery(canal,"SELECT CODBOBINA, ESPESOR as ESPE_INI FROM
dps");

# Cargamos la variable ESPE_FIN (Espesor CON recubrimiento)
T100ESPE_FIN <- sqlQuery(canal,"SELECT CODBOBINA, ESPTOTVALMED as ESPE_FIN FROM
tijera");

# Eliminamos las bobinas quitadas en trabajos anteriores
T100ESPE_INI <- T100ESPE_INI[(T100ESPE_INI$CODBOBINA %in% CODBOB),]

# Eliminamos los valores de espesores finales erróneos
T100ESPE_FIN <- T100ESPE_FIN[T100ESPE_FIN[,2]>=0,]

# Eliminamos las bobinas quitadas en trabajos anteriores
T100ESPE_FIN <- T100ESPE_FIN[(T100ESPE_FIN$CODBOBINA %in% CODBOB),]

# Obtenemos la media de cada bobina y el código de bobina
CODBOBESPFIN <- tapply(T100ESPE_FIN$CODBOBINA, T100ESPE_FIN$CODBOBINA,min)
ESPEFINMEAN <- tapply(T100ESPE_FIN[,2], T100ESPE_FIN[,1], mean)

# Eliminamos las bobinas que no hay en ESPE_FIN
CODBOBESPINI <- tapply(T100ESPE_INI$CODBOBINA, T100ESPE_INI$CODBOBINA,min)
ESPESORESINI <- T100ESPE_INI[(CODBOBESPINI %in% CODBOBESPFIN),2]


# Guardamos la matriz ESPESORES
```

```
CAPA <- ESPEFINMEAN- ESPESORESINI
ESPESORES <- cbind(CODBOBESPFIN, ESPESORESINI, ESPEFINMEAN, CAPA)
save(ESPESORES,file="ESPESORES.Rdata")

# Visualizamos el histograma de espesores
hist(CAPA,100,xlim=c(-0.25,0.25),col="red")
hist(ESPEFINMEAN,100,xlim=c(0,5),col="red")
hist(ESPESORESINI,100,xlim=c(0,5),col="red")
```

Figura 182. Programa utilizado para la obtención de los espesores.

Tal y como muestra la Figura 182, se obtiene la variable *CAPA* resultado de la resta del espesor final menos el inicial y que corresponde al espesor de la capa de metal después del baño. Esta variable, como es lógico, debería ser siempre positiva, pero como se advierte en el histograma de la Figura 183, existe un buen número de espesores de la capa que son negativos. **Esto es debido a que el espesor inicial es un valor aproximado del espesor de la bobina mientras que el espesor final si que corresponde con el valor real.** Debido a esto, es fácil comprender, que puede ocurrir que el espesor inicial real de la bobina sea menor del esperado, y por lo tanto, que el resultado sea negativo. Claramente se puede deducir la necesidad de disponer de una variable que indique el espesor real de la banda cuando entra en el horno de galvanizado.

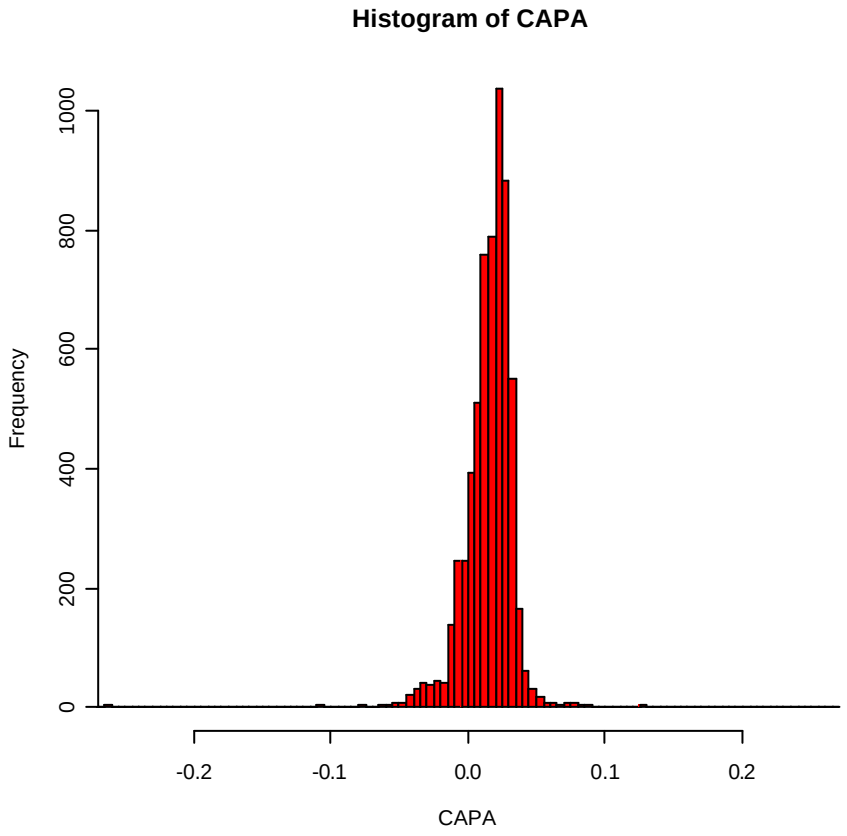


Figura 183. Histograma de la variable "CAPA".

Lo que claramente nos invita a comprender, es que **el valor de estas variables no es muy fiable,** y **solamente pueden ser utilizadas, cuando se analizan bobinas consecutivas con espesores similares.**

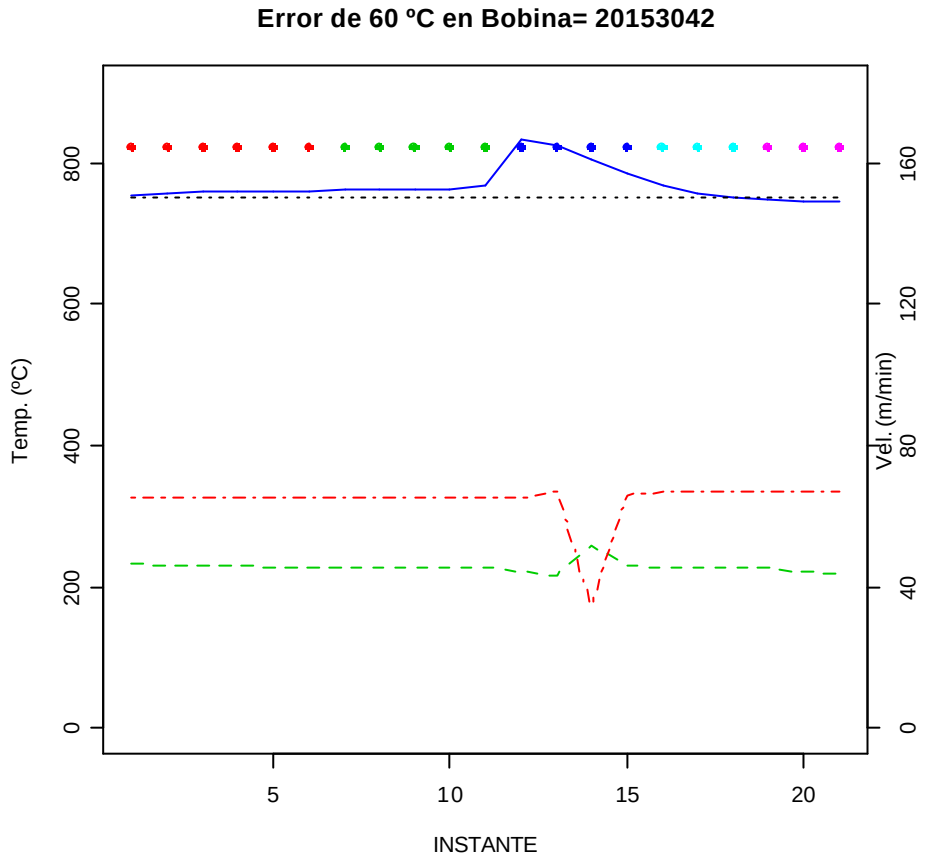


Figura 184. Error de 60ºC esperado entre la temperatura esperada de la banda y la final.

Por ejemplo, en la Figura 184, podemos observar cinco bobinas consecutivas, y como la temperatura de la bobina central (línea azul en trazo continuo) experimenta una subida elevada durante un tiempo.

Código	ESPESOR	Código	ESPESOR	Código	ESPESOR	Código	ESPESOR	Código	ESPESOR
20153040	2,0414	20153041	2,0294	20153042	1,9967	20153043	2,0145	20153044	2,0279
20153040	2,0383	20153041	2,0301	20153042	2,0013	20153043	2,0167	20153044	2,0186
20153040	2,0326	20153041	2,0343	20153042	2,0014	20153043	2,0169	20153044	2,0218
20153040	2,0360	20153041	2,0394	20153042	1,9979	20153043	2,0190	20153044	2,0205
20153040	2,0321	20153041	2,0417	20153042	1,9986	20153043	2,0187	20153044	2,0189
20153040	2,0367	20153041	2,0393	20153042	2,0009	20153043	2,0189	20153044	2,0198
20153040	2,0347	20153041	2,0393	20153042	2,0030	20153043	2,0206	20153044	2,0293
20153040	2,0351	20153041	2,0421	20153042	2,0080	20153043	2,0191	20153044	2,0288
20153040	2,0305	20153041	2,0394	20153042	2,0108	20153043	2,0166	20153044	2,0260
20153040	...	20153041	...	20153042	...	20153043	...	20153044	...
MEDIA	**2,034**	**MEDIA**	**2,032**	**MEDIA**	**2,003**	**MEDIA**	**2,018**	**MEDIA**	**2,024**

Tabla 27. Espesores finales (espesor más capa de recubrimiento) de las bobinas de la Figura 184.

Si vemos los espesores finales de las cinco bobinas de la Figura 184, y obtenemos la media para cada una de las bobinas (Tabla 27), podemos ver claramente que el espesor final de la bobina 20153042 es mucho menor al de las anteriores y posteriores. Lo que no se puede saber, es si esto es debido al fallo del tratamiento térmico o si el fallo del tratamiento térmico es debido a que el espesor inicial de esa bobina es menor que el de las anteriores. Lamentablemente, el espesor inicial tal y como aparecen en la tabla "*dps*" indica un valor de 2 para las cinco bobinas.

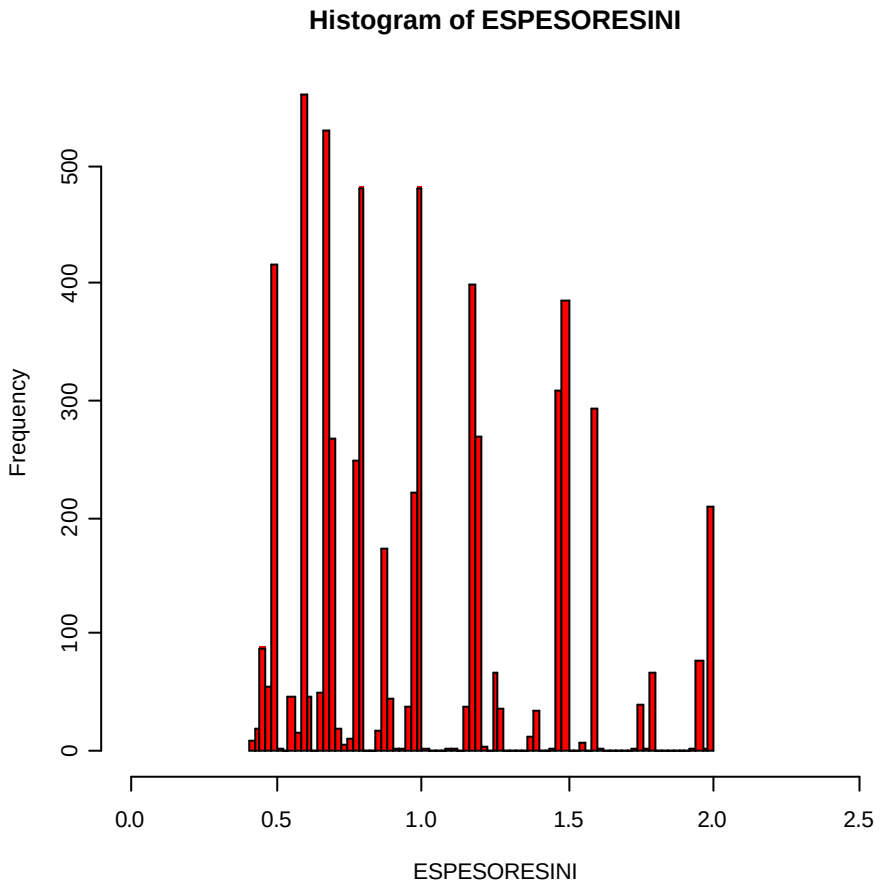


Figura 185. Histograma de los espesores de las diferentes bobinas utilizadas.

Una información muy valiosa nos la muestra el histograma de la Figura 185, donde se observan las diferentes "familias" de bobinas según el espesor. Estas varían entre 0.4 mm y 2 mm.

5.6 PREPARACIÓN DE LA BASE DE DATOS A UTILIZAR

Como resultado de todos los estudios anteriores y conclusiones efectuadas, se detectaron algunas anomalías en las dos primeras bases de datos que debían ser resueltas, ya que:

- Existían muchas variables con ruido o valores incompletos.
- No se conocía el modo de funcionamiento del horno.
- No se conocía el espesor real de la banda.
- Faltaban datos propios de cada bobina: ciclo térmico, tipo de acero, etc.

Debido a esto y a que, paralelamente a los primeros estudios aquí expuestos, se fueron desarrollando trabajos de mejora del sistema de adquisición, almacenamiento y control del horno; se propuso la realización de una nueva base de datos más completa, con menos cantidad de ruido y mejor calidad de los datos.

Todo el conocimiento obtenido de estos primeros estudios, también sirvió para que las consultas a expertos fueran más precisas y que las variables elegidas para esta nueva base de datos fueran mucho menores.

A esta base de datos, se le añadieron las siguientes nuevas variables:

- *BOBENT*: Código de fabricación de la bobina.
- *ESPENT*: Espesor real de la bobina a la entrada del horno.
- *CLASACERO*: Clase de acero de la bobina tratada.
- *DUREZA*: Dureza del acero.
- *CICREC*: Ciclo de recocido realizado.
- *MODHF*: Indica si el sistema está trabajando en "Modo Automático" (Valor a –1) o "Modo Manual" (Valor a 0).

En la figura siguiente podemos observar el listado del programa que captura las variables a utilizar, las preprocesa y las almacena en dos matrices para posteriores estudios.

```
# Cargamos las librerías de análisis multivariante
library(mva)
library(multiv)

# Cargamos las matrices con los datos de las 2.628 bobinas
library(RODBC)
canal <- odbcConnect("aceralia2003","","","localhost");

#############################################################
# Obtenemos los nuevos datos dinámicos de la tabla T100cal #
#############################################################

T100CALB <- sqlQuery(canal, "SELECT CODBOBINA, INSTANTE, THF1VALCNG as THC1,
TMPP1VALMED as TMPP1M, TMPP2VALMED as TMPP2M, TMPP2VALCNG as TMPP2C, MODHF FROM
T100cal");

#Eliminamos las bobinas con MODHF inexistente (12581 primeros valores)
T100CALB <- T100CALB[!is.na(T100CALB$MODHF),]

#Sacamos el código de bobina
LISTABOBINAST100 <- unique(T100CALB$CODBOBINA)

# Obtenemos los colores de las bobinas según el código
# y eliminamos el blanco
CODBOBT100 <- T100CALB$CODBOBINA
COLORBOB <- round((CODBOBT100-4) %% 7)+1

#############################################################
##############################
# Obtenemos datos de bobinas #
##############################

DATBOBINAS <- sqlQuery(canal, "SELECT CODBOBINA, BOBENT, ESPENT, CLASACERO,
DUREZA, CICREC, ANCHO, ESPESOR, LARGO, PESO, CALIDAD, FECFAB, HORFAB FROM dps")

# Detectamos las bobinas repetidas
table(DATBOBINAS$CODBOBINA)

# Obtenemos una lista de las bobinas sin repeticiones
LISTABOBINAS <- unique(DATBOBINAS$CODBOBINA)

# Sacamos la lista de la posición de las bobinas sin repetir
POSLISTA <- match(LISTABOBINAS,DATBOBINAS$CODBOBINA)

# Obtenemos una nueva base de datos sin las bobinas repetidas
DATBOBINAS <- DATBOBINAS[POSLISTA,]

# Obtenemos una nueva base sincronizada con las bobinas de la otra tabla
POSLISTA <- match(LISTABOBINAST100,DATBOBINAS$CODBOBINA)
DATBOBINAS <- DATBOBINAS[POSLISTA,]

# Obtenemos una nueva lista
LISTABOBINAS <- unique(DATBOBINAS$CODBOBINA)
# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINAS==LISTABOBINAST100)
TRUE
2090
#############################################################################
```

```
#########################################################################
# Obtenemos las velocidades comprimiéndolas de la tabla "T30Acumuladore" #
#########################################################################
T30CALVEL <- sqlQuery(canal, "SELECT CODBOBINA, INSTANTE, VELCEN as VELCENMED
FROM T30Acumuladore");

# Sacamos la lista de la posición de los datos de las bobinas de la
# otra base de datos
POSLISTA <- T30CALVEL$CODBOBINA %in% LISTABOBINAST100

# Obtenemos los datos solamente de las bobinas coincidentes con las otras BD
T30CALVEL <- T30CALVEL[POSLISTA,]

# Obtenemos una nueva lista de bobinas
LISTABOBINASVEL <- unique(T30CALVEL$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINAS==LISTABOBINASVEL)
TRUE
2090

#Obtenemos el número de datos de cada bobina
LONGITUDBOBT100 <- tapply(CODBOBT100,CODBOBT100,length)
LONGITUDBOBVEL <- tapply(T30CALVEL$CODBOBINA, T30CALVEL$CODBOBINA,length)

# Como la tabla de velocidades es más larga, debemos comprimirla #
#################################################################

# Obtenemos el paso en cada bobina
PASOVEL <- LONGITUDBOBVEL/ LONGITUDBOBT100

# Obtenemos el nuevo paso y la lista de bobinas sin contar las menores de 1
# Ya que esas no pueden ser comprimidas
LISTABOBINASVELBUENAS <- LISTABOBINASVEL[PASOVEL>=1]
PASOVEL <- PASOVEL[PASOVEL>=1]
# ---------------------------------------------------------
#Eliminamos las bobinas que no están en la lista
POSLISTA <- T30CALVEL$CODBOBINA %in% LISTABOBINASVELBUENAS
T30CALVEL <- T30CALVEL[POSLISTA,]

#Eliminamos las bobinas que no están en la lista de las otras bases de datos
POSLISTA <- DATBOBINAS$CODBOBINA %in% LISTABOBINASVELBUENAS
DATBOBINAS <- DATBOBINAS[POSLISTA,]

#Eliminamos las bobinas que no están en la lista de las otras bases de datos
POSLISTA <- T100CALB$CODBOBINA %in% LISTABOBINASVELBUENAS
T100CALB <- T100CALB[POSLISTA,]

# Obtenemos los colores de las bobinas según el código
# y eliminamos el blanco
CODBOBT100 <- T100CALB$CODBOBINA
COLORBOB <- round((CODBOBT100-4) %% 7)+1
# ---------------------------------------------------------
# Comprimimos la base de datos para que sea como la de T100CALB
# Volvemos a obtener el número de datos de cada bobina

LONGITUDBOBT100 <- tapply(CODBOBT100,CODBOBT100,length)
LONGITUDBOBVEL <- tapply(T30CALVEL$CODBOBINA, T30CALVEL$CODBOBINA,length)
```

```
# Como la tabla de velocidades es más larga, debemos comprimirla #
#################################################################

# Obtenemos de nuevo el paso en cada bobina, ahora si las bobinas erróneas
PASOVEL <- LONGITUDBOBVEL/ LONGITUDBOBT100
BOBVELD <- T30CALVEL$CODBOBINA
VELOCIDAD <- T30CALVEL$VELCENMED

DIVIDE <- PASOVEL[match(BOBVELD, LISTABOBINASVELBUENAS)]

PUESTOS <- floor(1: length(VELOCIDAD)%% DIVIDE)
VELOCIDADFIN <- VELOCIDAD[PUESTOS==0]
VELOCIDADFIN <- as.numeric(VELOCIDADFIN[1: length(CODBOBT100)])

CODBOBVEL <- BOBVELD[PUESTOS==0]
CODBOBVEL <- as.numeric(CODBOBVEL[1: length(CODBOBT100)])
CODAMBOS <- cbind(CODBOBT100,CODBOBVEL)
#########################################################################
###################################
# Obtenemos datos de espesor final #
###################################

DATESPESOR <- sqlQuery(canal, "SELECT CODBOBINA, INSTANTE, ESPTOTMED FROM
T30Tijera")

#Eliminamos las bobinas que no están en la lista de las otras bases de datos
POSLISTA <- DATESPESOR$CODBOBINA %in% LISTABOBINASVELBUENAS
DATESPESORFINAL <- DATESPESOR[POSLISTA,]

# Ordenamos las listas
NUMORDER <- order(DATESPESORFINAL$CODBOBINA)

# Espesores Ordenados
ESPORDENADOS <- DATESPESORFINAL[NUMORDER,]

# Eliminamos
# ESPORDENADOS <- ESPORDENADOS[ESPORDENADOS$ESPTOTMED>-10,]

# Obtenemos una nueva lista de bobinas
LISTABOBINASESP <- unique(ESPORDENADOS$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTABOBINASESP)
TRUE
1979
# Determinamos el valor del ESPESOR FINAL
ESPFINAL <- tapply(ESPORDENADOS$ESPTOTMED, ESPORDENADOS$CODBOBINA, median)

# Determinamos que bobinas han sido trabajadas en "modo manual" y "modo autom"
MODOBOB <- tapply(T100CALB$MODHF, T100CALB$CODBOBINA, median)

# Generamos unas nuevas matrices con los datos
attach(T100CALB)

MATDINAMIC <- data.frame(cbind(CODBOBINA, INSTANTE, THC1, TMPP1M, TMPP2M,
TMPP2C, VELOCIDADFIN, COLORBOB, MODHF))
MATBOBINAS <- cbind(DATBOBINAS$CODBOBINA, ESPFINAL, MODOBOB)
```

Figura 186. Listado que genera las nuevas matrices de la base de datos tercera.

5.6.1 ANÁLISIS INICIAL

Lo primero que se observa, es que en la tabla *"dps"* donde se muestra las características de cada bobina, existen muchas de ellas repetidas. Por ello, se procede a eliminar las observaciones repetidas.

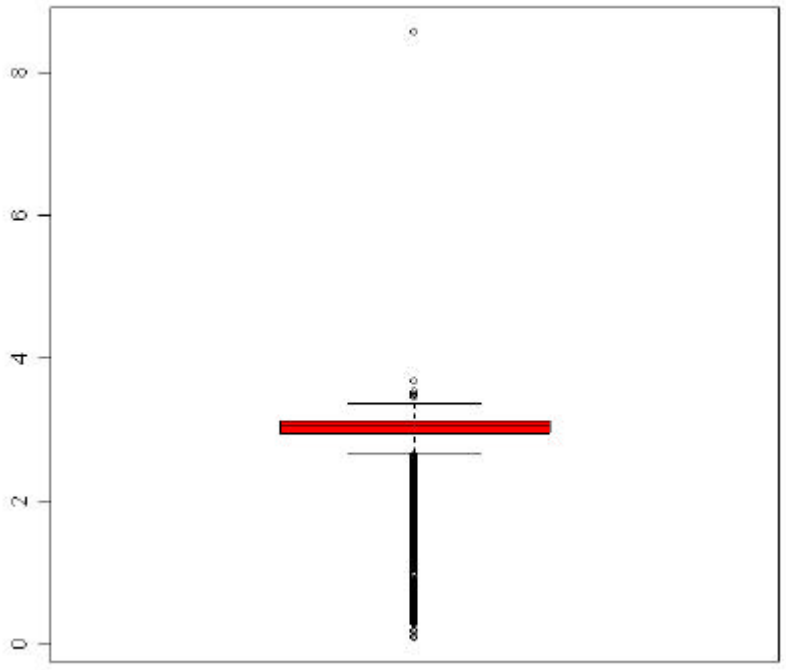


Figura 187. Boxplot, de la relación entre los datos de las tablas T100 y T30.

De la observación y tratamiento de la base de datos, ha aparecido una problemática cuando se ha pretendido comprimir los datos de la tabla *"T30Acumuladore"* y *"T30Tijera"* (correspondientes a medidas cada 30 metros) según los de la tabla "T100cal" (correspondientes a medidas cada 100 metros), ya que la relación de compresión tendría que estar en 3,33 pero como se demuestra en la figura siguiente, existen muchas bobinas cuya relación se desvía mucho de esa media, e incluso es menor que 1. Estas últimas (111 de 2.090), han tenido que ser eliminadas, ya que no se podía comprimir. **Este error es debido a que las medidas son realizadas en el acumulador donde la velocidad es diferente que la del horno.**

5.6.1.1 ANÁLISIS EXPLORATORIO INICIAL DE LA NUEVA BASE DE DATOS

Se estudian los datos y se determina que:

- De 71.842 datos existentes:
  - Los 12.581 primeros (17,6%) corresponden a datos con la variable *MODHF* inexistente. Se eliminan de la base de datos.
  - 40.496 observaciones (56,3%) son en datos del "modo manual" (*MODHF*=0).
  - 18.765 (26,1%) corresponden al modo automático" (*MODHF*=-1).
- La compresión de las tablas con datos medidos cada 30 metros, ha sido bastante problemática, debido a que existían bobinas con menor número de datos que en las tablas de medidas de 100 metros. De esta forma, ha sido necesario eliminar esas bobinas (111 de 2.090).
- Finalmente quedan un total de 1.979 bobinas a estudiar.

```
# Realizamos un primer análisis de los datos

# Número de Bobinas
LONGBOB <- length(MATBOBINAS[,1])
LONGBOB
[1] 1979

# Datos bobinas primera
DATBOBINAS[1,]
    CODBOBINA   BOBENT ESPENT CLASACERO DUREZA CICREC ANCHO ESPESOR
    23293006  2324D517  0.583  B011F97    17    <NA>  1250   0.6
    LARGO  PESO CALIDAD   FECFAB     HORFAB
    3683   21770   NA    25-11-2002  08:51
MATBOBINAS[1,]
                ESPFINAL      MODOBOB
2.329301e+07 5.861333e-01 0.000000e+00

# Datos bobinas última
DATBOBINAS[LONGBOB,]
     CODBOBINA   BOBENT ESPENT CLASACERO DUREZA CICREC ANCHO ESPESOR
     23653024  2362D068  0.993  B100F55    50    <NA>  1090    1
     LARGO  PESO CALIDAD     FECFAB     HORFAB
     1826   15350  NA       31-12-2002  17:46
MATBOBINAS[LONGBOB,]
                ESPFINAL      MODOBOB
2.365302e+07 9.817930e-01 0.000000e+00
```

```
# Resumen de Datos
summary(DATBOBINAS)
   CODBOBINA             BOBENT          ESPENT         CLASACERO
 Min.   :23293006   146494  :   3   Min.   :0.417   B100F55:650
 1st Qu.:23383060   2341D541:   3   1st Qu.:0.670   B105F55:295
 Median :23473038   2342T502:   3   Median :0.760   B102G33:152
 Mean   :23476003   144079  :   2   Mean   :0.883   B012F53:129
 3rd Qu.:23563035   145430  :   2   3rd Qu.:1.188   C114G55:113
 Max.   :23653024   145702  :   2   Max.   :2.016   B102G55: 82
                    (Other) :1964                   (Other):558
    DUREZA      CICREC          ANCHO          ESPESOR
 50     :1269   A1  :  33   Min.   : 750   Min.   :0.4300
 E8     : 166   B1  :   2   1st Qu.:1000   1st Qu.:0.6700
 19     : 135   D1  :   5   Median :1210   Median :0.7600
 14     :  73   D2  :   1   Mean   :1166   Mean   :0.8813
 32     :  51   NA's:1938   3rd Qu.:1300   3rd Qu.:1.1700
 (Other): 260               Max.   :1525   Max.   :2.0000
 NA's   :  25
    LARGO           PESO          CALIDAD                FECFAB
 Min.   : 350   Min.   : 1380   Min.   :   1.000   11-12-2002:  82
 1st Qu.:1654   1st Qu.:14160   1st Qu.:   1.000   27-12-2002:  78
 Median :2200   Median :18510   Median :   1.000   12-12-2002:  74
 Mean   :2418   Mean   :17414   Mean   :   1.031   22-12-2002:  74
 3rd Qu.:3170   3rd Qu.:21295   3rd Qu.:   1.000   07-12-2002:  72
 Max.   :5538   Max.   :26340   Max.   :   2.000   01-12-2002:  70
                                NA's   :1914.000   (Other)   :1529
    HORFAB
 01:12  :   6
 08:51  :   6
 09:38  :   6
 23:31  :   6
 04:58  :   5
 05:09  :   5
 (Other):1945
summary(MATBOBINAS)

      V1              ESPFINAL             MODOBOB
 Min.   :23293006   Min.   :-1.701e+38   Min.   :-1.0000
 1st Qu.:23383060   1st Qu.: 6.629e-01   1st Qu.:-1.0000
 Median :23473038   Median : 7.557e-01   Median : 0.0000
 Mean   :23476003   Mean   :-3.439e+35   Mean   :-0.2883
 3rd Qu.:23563035   3rd Qu.: 1.166e+00   3rd Qu.: 0.0000
 Max.   :23653024   Max.   : 1.970e+00   Max.   : 0.0000
```

Figura 188. Resumen de los datos relativos a cada bobina.

De las figuras siguientes se extraen otras conclusiones:

- Datos Relativos a 35 días de proceso.
- 48 clases de acero con 32 tipos de durezas.
- Variables *CICREC* (ciclo de recocido) y *CALIDAD*, prácticamente sin datos.
- Bobinas entre 350 y 5.538 metros de longitud.

- Rango de espesores de 0,417 mm. a 2,016 mm. La mayoría de los espesores de las bobinas se centran en el rango 0,417 mm. a 1 mm.

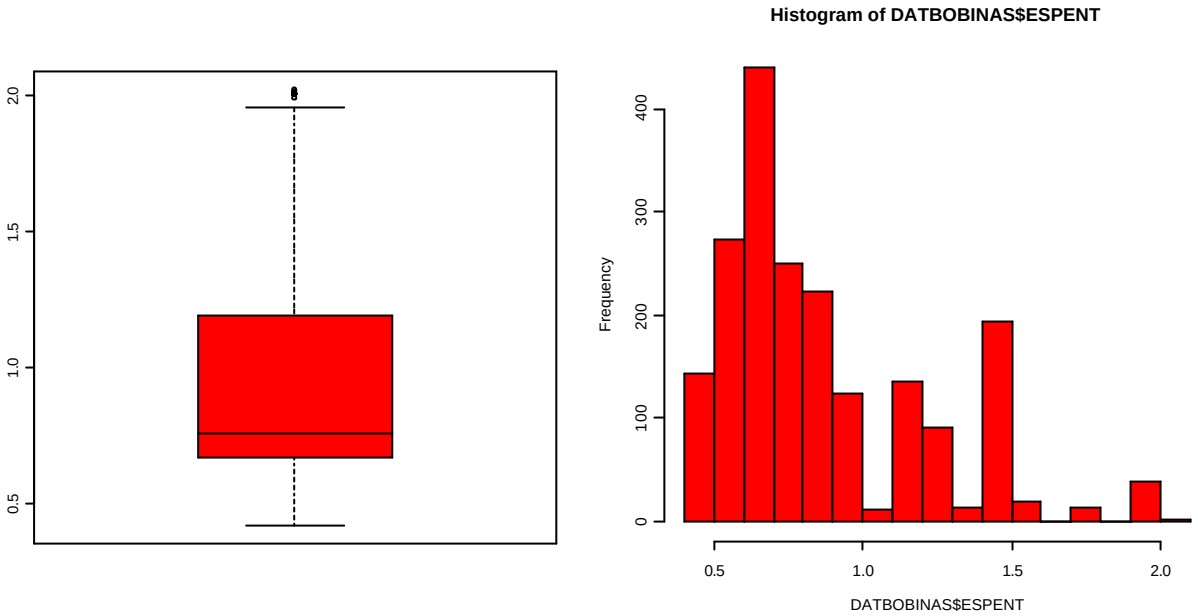


Figura 189. Gráfico box-plot y histograma del espesor entrante de las bobinas.

Analizamos los datos dinámicos...

```
# Obtenemos un resumen de los datos dinámicos
summary(MATDINAMIC)
   CODBOBINA           INSTANTE          THC1           TMPP1M
 Min.   :23293006   Min.   : 1.00   Min.   :  0.0   Min.   : -1
 1st Qu.:23393012   1st Qu.: 8.00   1st Qu.:783.0   1st Qu.:230
 Median :23473063   Median :15.00   Median :816.0   Median :245
 Mean   :23476682   Mean   :16.55   Mean   :804.7   Mean   :243
 3rd Qu.:23563051   3rd Qu.:24.00   3rd Qu.:837.0   3rd Qu.:261
 Max.   :23653024   Max.   :58.00   Max.   :878.0   Max.   :371
    TMPP2M          TMPP2C       VELOCIDADFIN      COLORBOB
 Min.   : -1.0   Min.   :  0.0   Min.   : -1.0   Min.   :1.000
 1st Qu.:772.0   1st Qu.:770.0   1st Qu.:100.0   1st Qu.:2.000
 Median :814.0   Median :815.0   Median :120.0   Median :4.000
 Mean   :789.5   Mean   :788.6   Mean   :113.1   Mean   :4.005
 3rd Qu.:825.0   3rd Qu.:825.0   3rd Qu.:130.0   3rd Qu.:6.000
 Max.   :889.0   Max.   :865.0   Max.   :150.0   Max.   :7.000


     MODHF
 Min.   :-1.0000
 1st Qu.:-1.0000
 Median : 0.0000
 Mean   :-0.3166
 3rd Qu.: 0.0000
 Max.   : 0.0000
```

Figura 190. Resumen de la matriz de datos dinámicos.

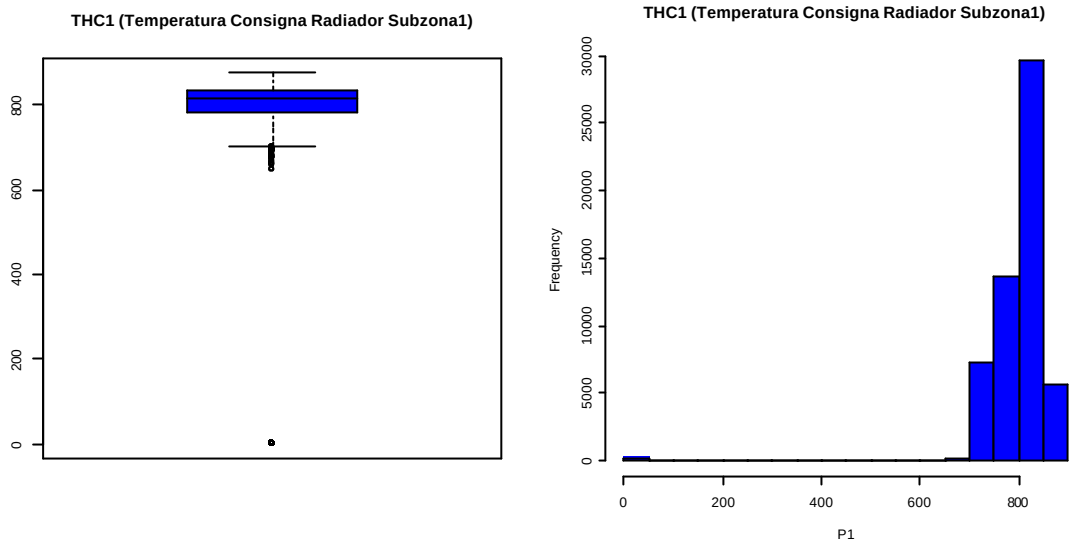


Figura 191. Gráfico box-plot y histograma de la temperatura de la zona 1 del horno.

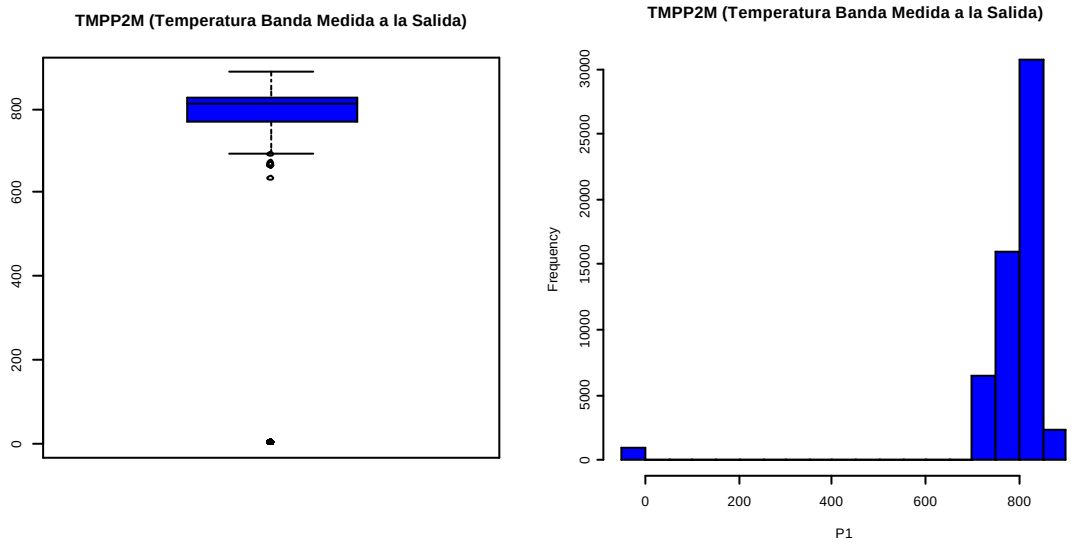


Figura 192. Gráfico box-plot y histograma de la temp. de la banda a la salida de la zona de calentamiento del horno.

En las figuras anteriores se ve claramente que aún siguen existiendo algunos valores a cero resultado, seguramente, de una pérdida de los datos o a paradas imprevistas.

Será necesario eliminar esos puntos o tomar en cuenta que existen en la futura creación de variables.

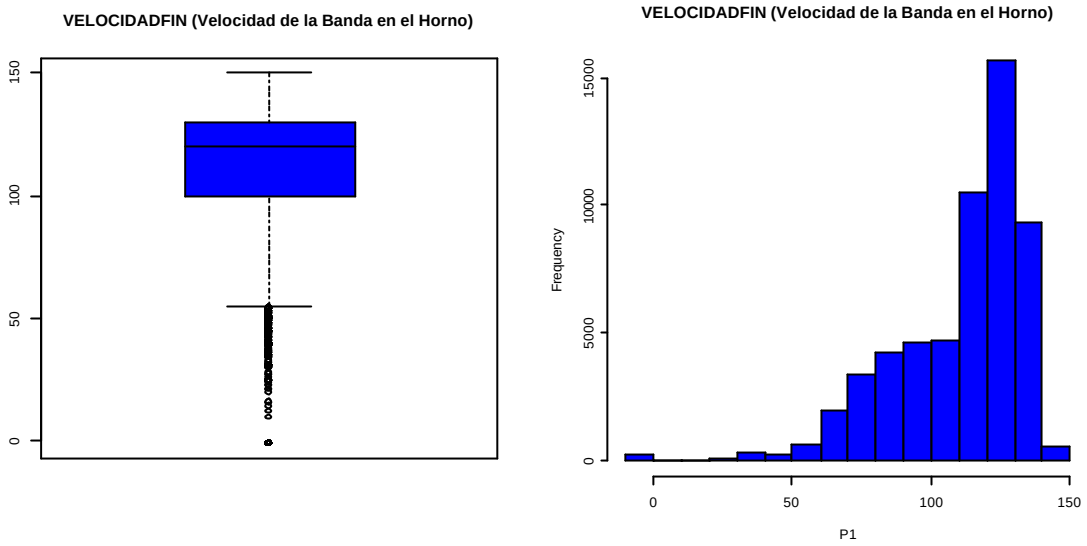


Figura 193. Gráfico box-plot y histograma de la velocidad de la banda dentro del horno.

Los gráficos de la Figura 193 muestran cómo la variable velocidad presenta una cantidad importante de valores próximos a cero y una varianza mayor que las anteriores.

A continuación se analizan con más detalle los datos obtenidos.

```
# Realizamos un segundo análisis de los datos

# Visualizamos el número de clases de acero
K1 <-  DATBOBINAS$CLASACERO
table(K1)
B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53
      2      51       4     129       7       2       3       3
B014F55 B016F35 B017F53 B023H53 B025F55 B032H53 B042H53 B044H53
      0       8       0       4      45       4       2       5
B081B99 B085F97 B085G99 B100B95 B100F33 B100F55 B101F55 B102G33
      4      10      37      16       1     650      12     152
B102G55 B103G33 B103G55 B105F55 B120G55 C107G55 C114G55 C115G55
     82       2      15     295      14      52     113       2
C116G55 D012F55 D012G99 D031B33 D032F55 D071F55 D094B33 D094G55
      2      14       6      19      27      18       3      11
K011B55 K011F57 K021H43 K021H53 K022H53 N013H53 N017B97 X100G99
     10      63       1      30       2       7       1      39


# Visualizamos el número de clases de acero
K2 <-  DATBOBINAS$DUREZA
table(K2)
  11   13   14   15   16   17   19   20   24   29   30   32   37
  10   12   73   43   34   48  135    2    3    4    7   51    2
  50   E1   E8   F8   G0   G4
1269   35  166   27   19   14
```

```
# Visualizamos K1 frente a K2 (CLASE DE ACERO ~ DUREZA)
table(K1,K2)
                                K2
K1        11 13 14 15 16 17  19 20 24 29 30 32 37  50 E1  E8 F8 G0 G4
  B011B99  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B011F97  0  0  0  0  0 48   0  0  0  0  0  0  0   0  0   0  0  0  0
  B012B97  0  4  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B012F53  0  0  0  0  0  0 129  0  0  0  0  0  0   0  0   0  0  0  0
  B012F55  0  0  0  0  0  0   6  0  0  0  0  0  0   0  0   0  0  0  0
  B013B55  2  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B013C55  0  3  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B014F53  0  0  0  0  0  0   0  0  3  0  0  0  0   0  0   0  0  0  0
  B014F55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B016F35  8  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B017F53  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B023H53  0  0  0  0  0  0   0  0  0  4  0  0  0   0  0   0  0  0  0
  B025F55  0  0  0 43  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B032H53  0  0  0  0  0  0   0  0  0  0  0  4  0   0  0   0  0  0  0
  B042H53  0  0  0  0  0  0   0  2  0  0  0  0  0   0  0   0  0  0  0
  B044H53  0  5  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  B081B99  0  0  0  0  0  0   0  0  0  0  0  0  2   0  0   0  0  0  0
  B085F97  0  0  0  0  0  0   0  0  0  0  0 10  0   0  0   0  0  0  0
  B085G99  0  0  0  0  0  0   0  0  0  0  0 37  0   0  0   0  0  0  0
  B100B95  0  0  0  0  0  0   0  0  0  0  0  0  0  16  0   0  0  0  0
  B100F33  0  0  0  0  0  0   0  0  0  0  0  0  0   1  0   0  0  0  0
  B100F55  0  0  0  0  0  0   0  0  0  0  0  0  0 647  0   0  0  0  0
  B101F55  0  0  0  0  0  0   0  0  0  0  0  0  0  12  0   0  0  0  0
  B102G33  0  0  0  0  0  0   0  0  0  0  0  0  0 149  0   0  0  0  0
  B102G55  0  0  0  0  0  0   0  0  0  0  0  0  0  82  0   0  0  0  0
  B103G33  0  0  0  0  0  0   0  0  0  0  0  0  0   2  0   0  0  0  0
  B103G55  0  0  0  0  0  0   0  0  0  0  0  0  0  15  0   0  0  0  0
  B105F55  0  0  0  0  0  0   0  0  0  0  0  0  0 294  0   0  0  0  0
  B120G55  0  0  0  0  0  0   0  0  0  0  0  0  0  13  0   0  0  0  0
  C107G55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0  51  0  0  0
  C114G55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0 113  0  0  0
  C115G55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   2  0  0  0
  C116G55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  D012F55  0  0  0  0  0  0   0  0  0  0  0  0  0   0 14   0  0  0  0
  D012G99  0  0  0  0  0  0   0  0  0  0  0  0  0   0  3   0  0  0  0
  D031B33  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0 19  0
  D032F55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0 27  0  0
  D071F55  0  0  0  0  0  0   0  0  0  0  0  0  0   0 18   0  0  0  0
  D094B33  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  3
  D094G55  0  0  0  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0 11
  K011B55  0  0 10  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  K011F57  0  0 63  0  0  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  K021H43  0  0  0  0  1  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  K021H53  0  0  0  0 30  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  K022H53  0  0  0  0  2  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  N013H53  0  0  0  0  0  0   0  0  0  0  7  0  0   0  0   0  0  0  0
  N017B97  0  0  0  0  1  0   0  0  0  0  0  0  0   0  0   0  0  0  0
  X100G99  0  0  0  0  0  0   0  0  0  0  0  0  0  38  0   0  0  0  0

# Modo de operación de la bobina
K3 <- MATBOBINAS[,3]
table(K3)
  -1 -0.5    0
 569    3 1407
```

```
# Visualizamos K1 frente a K2 (CLASE DE ACERO ~ MODO (AUTOMATICO=-1 o MANUAL=0)
table(K1,K3)
             K3
K1          -1 -0.5   0
  B011B99    0    0   2
  B011F97    3    1  47
  B012B97    0    0   4
  B012F53   60    0  69
  B012F55    0    0   7
  B013B55    1    0   1
  B013C55    0    0   3
  B014F53    2    0   1
  B014F55    0    0   0
  B016F35    4    0   4
  B017F53    0    0   0
  B023H53    0    0   4
  B025F55   23    0  22
  B032H53    0    0   4
  B042H53    0    0   2
  B044H53    0    0   5
  B081B99    1    0   3
  B085F97    0    0  10
  B085G99    1    0  36
  B100B95    7    0   9
  B100F33    1    0   0
  B100F55  183    0 467
  B101F55    1    0  11
  B102G33   84    0  68
  B102G55   17    0  65
  B103G33    1    0   1
  B103G55    1    0  14
  B105F55   94    0 201
  B120G55    2    0  12
  C107G55    4    0  48
  C114G55   24    1  88
  C115G55    1    0   1
  C116G55    0    0   2
  D012F55    0    0  14
  D012G99    0    0   6
  D031B33    2    0  17
  D032F55    2    0  25
  D071F55    0    0  18
  D094B33    0    0   3
  D094G55    4    0   7
  K011B55    2    1   7
  K011F57   38    0  25
  K021H43    1    0   0
  K021H53    4    0  26
  K022H53    0    0   2
  N013H53    0    0   7
  N017B97    0    0   1
  X100G99    1    0  38
```

```
# Observamos el tratamiento que se hace de cada tipo de dureza
# K2 es la dureza del acero, K3 es el tipo de modo "Automático=-1 ~ Manual=0"
table(K2,K3)
          K3
K2     -1 -0.5   0
  11    5    0   5
  13    0    0  12 *
  14   40    1  32
  15   23    0  20
  16    5    0  29 *
  17    3    1  44 *
  19   60    0  75
  20    0    0   2
  24    2    0   1
  29    0    0   4 *
  30    0    0   7 *
  32    1    0  50 *
  37    1    0   1
  50  392    0 877
  E1    0    0  35 *
  E8   29    1 136
  F8    2    0  25 *
  G0    2    0  17
  G4    4    0  10

# Obtenemos una lista de la clase de aceros de la bobinas
O <- table(K1)

# Sacamos las más abundantes
O2 <- O[O>40]
B011F97 B012F53 B025F55 B100F55 B102G33 B102G55 B105F55 C107G55 C114G55 K011F57
     51     129      45     650     152      82     295      52     113      63

# Determinamos la dureza de cada bobina
DUREZA <- K2[match(row.names(as.matrix(O2)),K1)]

DUREZA
[1] 17 19 15 50 50 50 50 E8 E8 14
```

Figura 194. Primeros resultados del análisis visual.

De la Figura 194 podemos extraer otras conclusiones:

- Existe una relación 1/3 de bobinas tratadas en "modo automático" frente al "modo manual", es decir, el tratamiento en "modo manual" es tres veces superior al "modo automático" de uso del modelo, lo que implica que el uso del modelo matemático no es muy intensivo.
- 569 bobinas tratadas en "modo automático".
- 1.407 bobinas tratadas en "modo manual".

- Las clases de acero más utilizadas han sido las de la tabla siguiente (de las cuales 1.179, un 59,6%, son de dureza 50):

Clase Acero	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
Número de Bobinas	51	129	45	650	152	82	295	52	113	63
Dureza del Acero	17	19	15	50	50	50	50	E8	E8	14

Tabla 28. Número de bobinas más tratadas.

- Existen bobinas con un alto porcentaje del uso en "Modo Manual" frente al "Modo Automático" (ver tabla), lo que indica claramente que el modelo no es usado en varios tipos de bobinas.

Dureza	Bobinas en Modo Automático	Bobinas en Modo Manual	Porcentaje de Bobinas en Modo Manual
11	5	5	50,00%
13	0	12	100,00%
14	40	32	44,44%
15	23	20	46,51%
16	5	29	85,29%
17	3	44	93,62%
19	60	75	55,56%
20	0	2	100,00%
24	2	1	33,33%
29	0	4	100,00%
30	0	7	100,00%
32	1	50	98,04%
37	1	1	50,00%
50	392	877	69,11%
E1	0	35	100,00%
E8	29	136	82,42%
F8	2	25	92,59%
G0	2	17	89,47%
G4	4	10	71,43%

Figura 195. Porcentaje de bobinas tratadas en "modo manual" frente al "modo automático" según la dureza del acero.

5.6.1.2 ANÁLISIS VISUAL DE LOS DATOS

Siguiendo con el análisis, se propone el programa de la Figura 196 para representar gráficamente el comportamiento de las curvas más importantes.

```
# Función que dibuja la evolución de las curvas dinámicamente
# NumIni=Primer dato
# NumDatos=Numero de datos
#MATDINAMIC <- cbind(CODBOBINA, INSTANTE, THC1, TMPP1M, TMPP2M, TMPP2C,
#VELOCIDADFIN, COLORBOB, MODHF)

dibujadina <- function(NumIni=1, NumDatos=1000)
{
      # Dibujamos los 'NumDatos' desde la posición 'NumIni'
      NumFin= NumIni+NumDatos-1
      plot(MATDINAMIC[NumIni:NumFin,3], col= MATDINAMIC[NumIni:NumFin,8],pch=20,
ylim=c(200,900),xlab="INSTANTE",ylab="Temp. (ºC)",main=paste("Datos desde ",
NumIni," hasta ", NumFin))
      axis(4,c(0,200,400,600,800),c("0","40","80","120","160"))
      mtext("Vel. (m/min)",4)

      # Dibuja temperatura entrada de banda en verde
      lines(MATDINAMIC[NumIni:NumFin,4],col=3,lty=1,lwd=2)

      # Dibuja temperatura salida de banda en azul
      lines(MATDINAMIC[NumIni:NumFin,5],col=4,lty=1,lwd=2)

      # Dibuja temperatura salida objetivo de banda en negro
      lines(MATDINAMIC[NumIni:NumFin,6],col=1,lty=3,lwd=3)

      # Dibuja velocidad de banda en rojo
      lines(MATDINAMIC[NumIni:NumFin,7]*5,col=2,lty=4,lwd=2)

      # Dibuja el modo de operación
      lines(400-200*MATDINAMIC[NumIni:NumFin,9],col=6,lty=1,lwd=1)
}
```

Figura 196. Programa que dibuja el comportamiento dinámico del proceso.

Debido a las altas inercias del horno y los cambios bruscos de consignas que se originan entre unas bobinas y otras, se considera necesario analizar la evolución de las curvas de forma continua, focalizando el estudio en grupos de varias bobinas.

Se visualizarán las variables seleccionadas en estudios anteriores siguientes:

- *TMPP1M*: Temperatura media de la banda a la entrada del horno (*línea continua verde inferior*).
- *TMPP2M*: Temperatura media de la banda a la salida de la zona de calentamiento del horno (*línea continua azul*).
- *TMPP2C*: Temperatura esperada de la banda a la salida de la zona de calentamiento del horno (*línea de puntos negros*).

- *THC1*: Temperatura de consigna de la subzona 1 (puntos gruesos de diferente color para cada bobina).
- *VELOCIDADFIN*: Velocidad medida de la banda en el centro del horno (línea punto – raya – punto roja). En este caso, **se ha multiplicado la velocidad por cinco para poder observar con mejor detalle las variaciones en la misma dentro de la escala de temperaturas**. A la derecha del dibujo, se ha dispuesto un segundo eje vertical donde se muestra la escala de velocidades en metros por minuto.
- *MODHF*: Modo de trabajo del horno (0 en "Modo Manual" y –1 en "Modo Automático" (línea magenta fina).

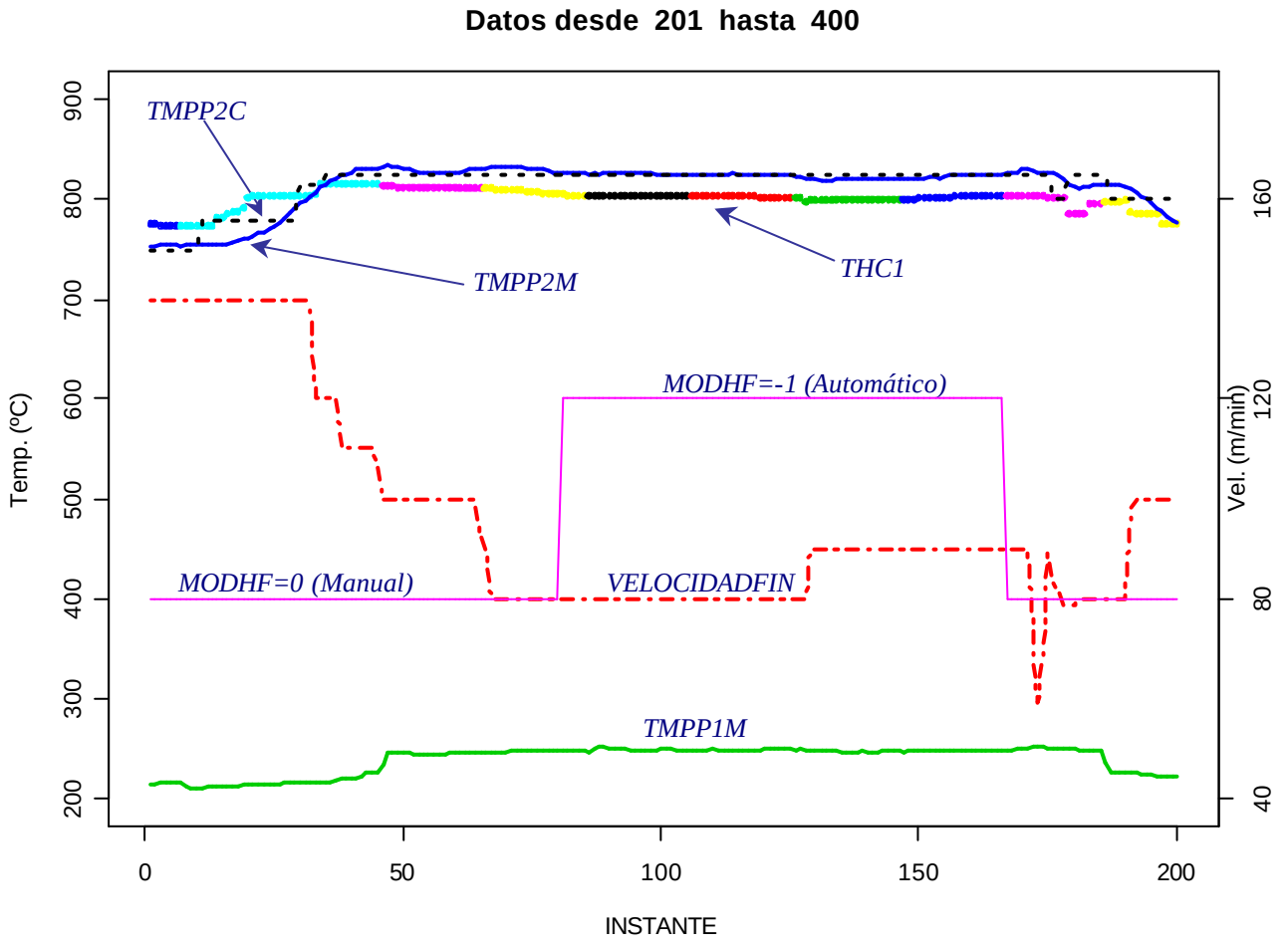


Figura 197. Gráfico explicativo de las curvas más importantes a estudiar.

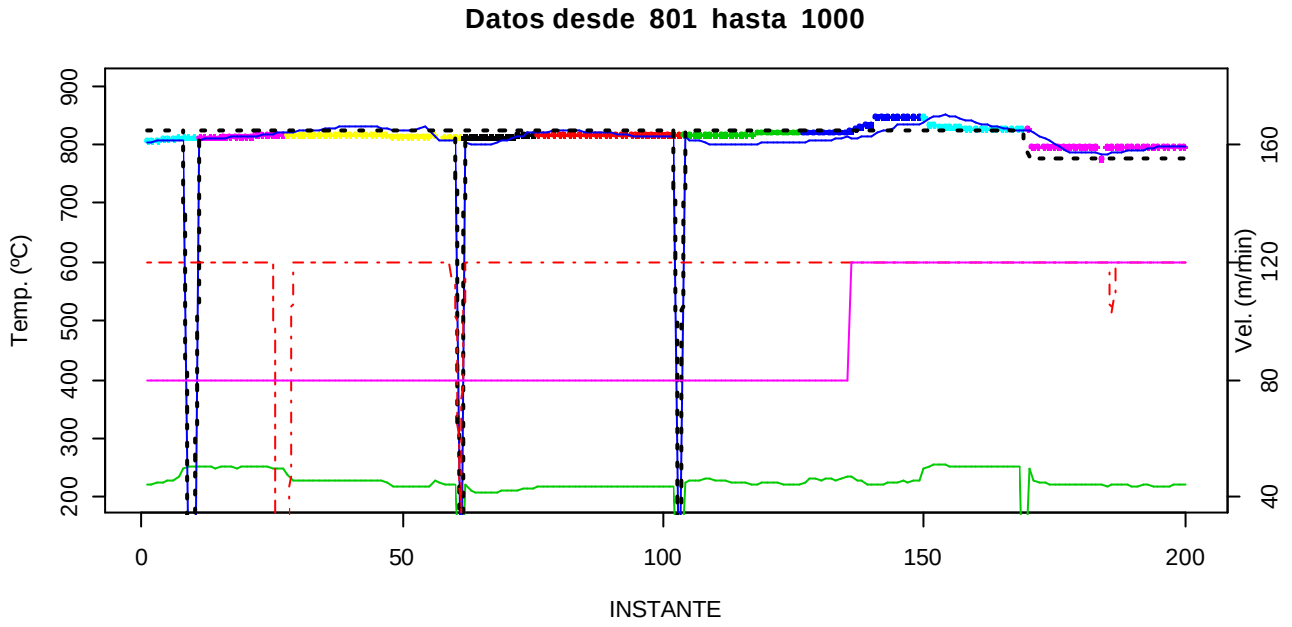


Figura 198. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

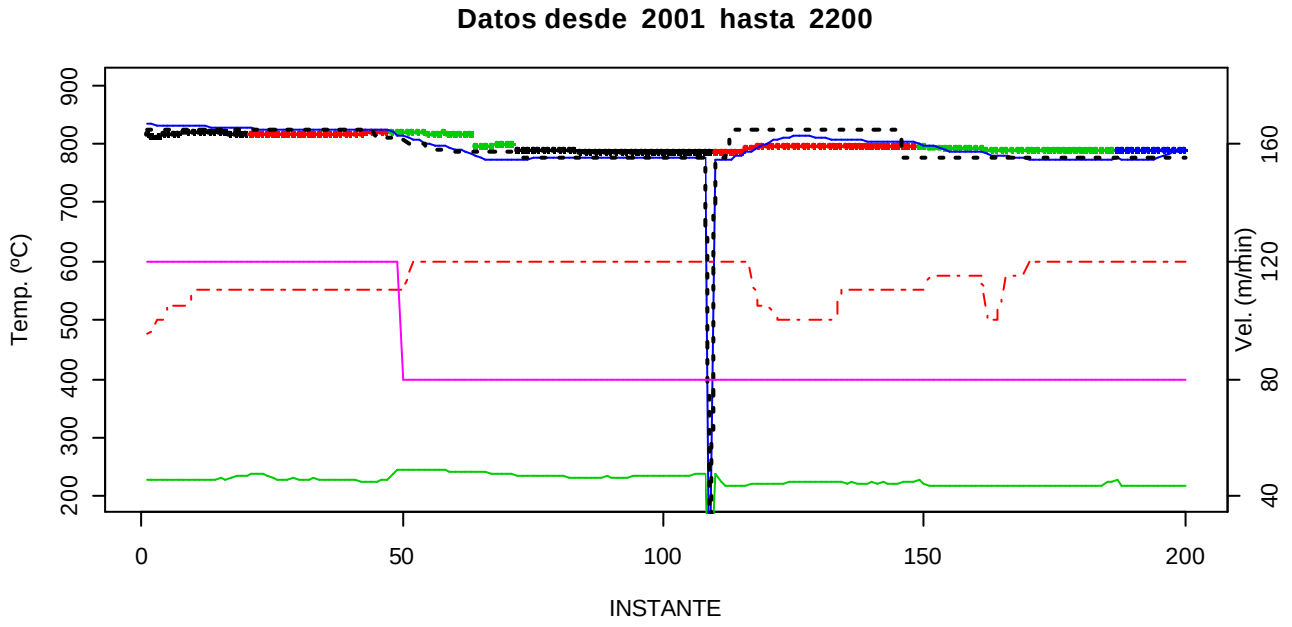


Figura 199. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

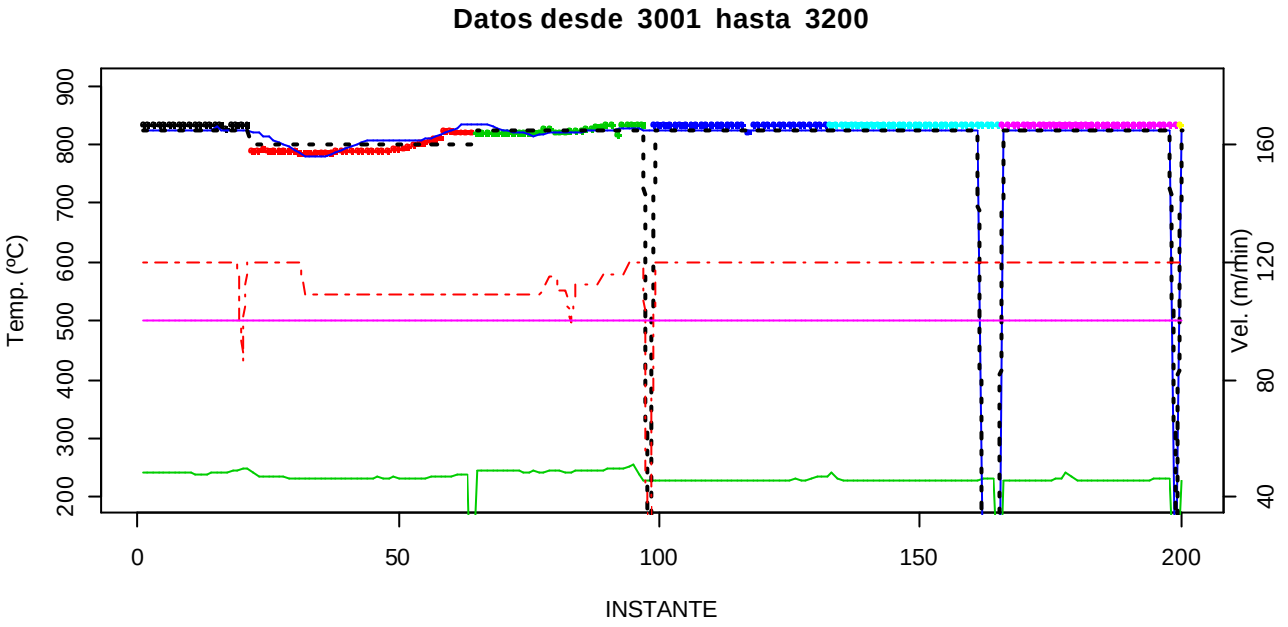


Figura 200. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

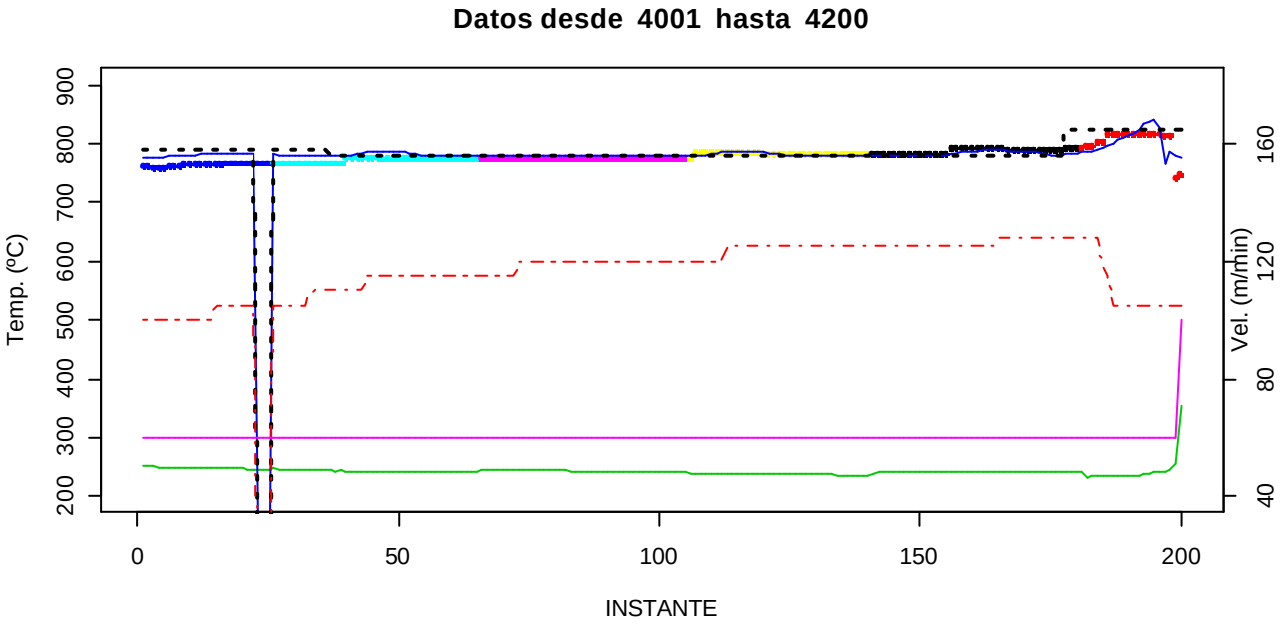


Figura 201. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

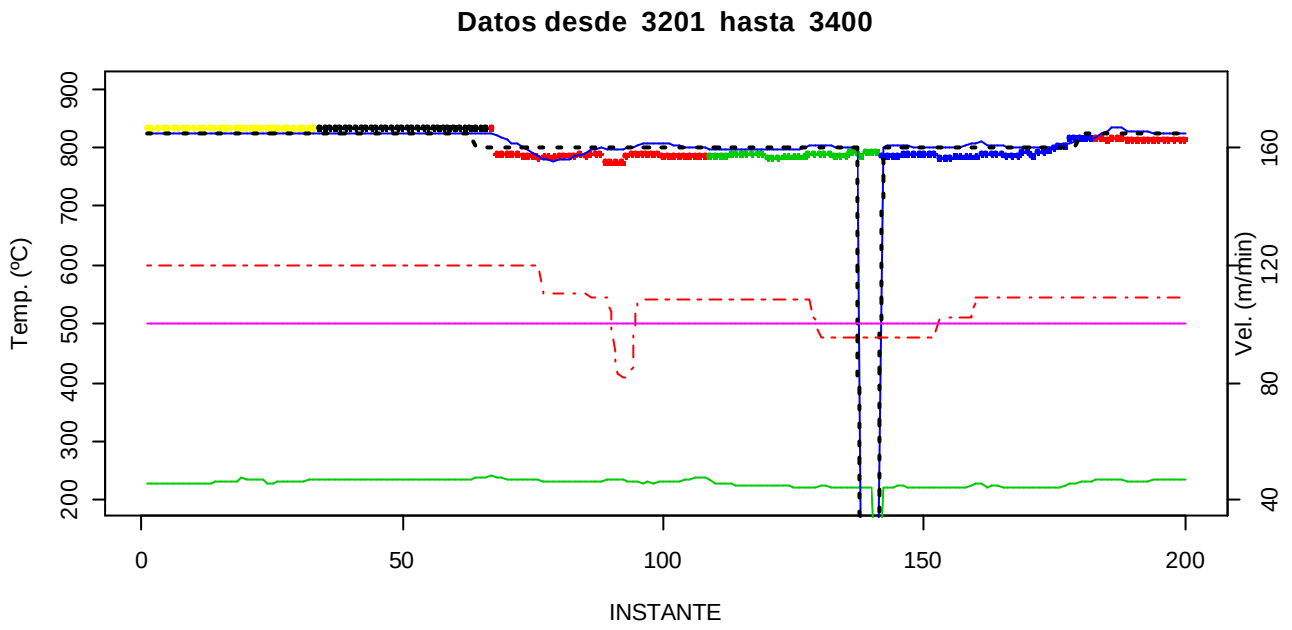


Figura 202. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

Como se puede observar, el comportamiento en los "dos modos" ha mejorado sustancialmente gracias al trabajo de mejora del personal experto de la empresa, aunque existen momentos en que la temperatura de la banda no alcanza con rapidez la temperatura objetivo (ver Figura 199 y Figura 201).

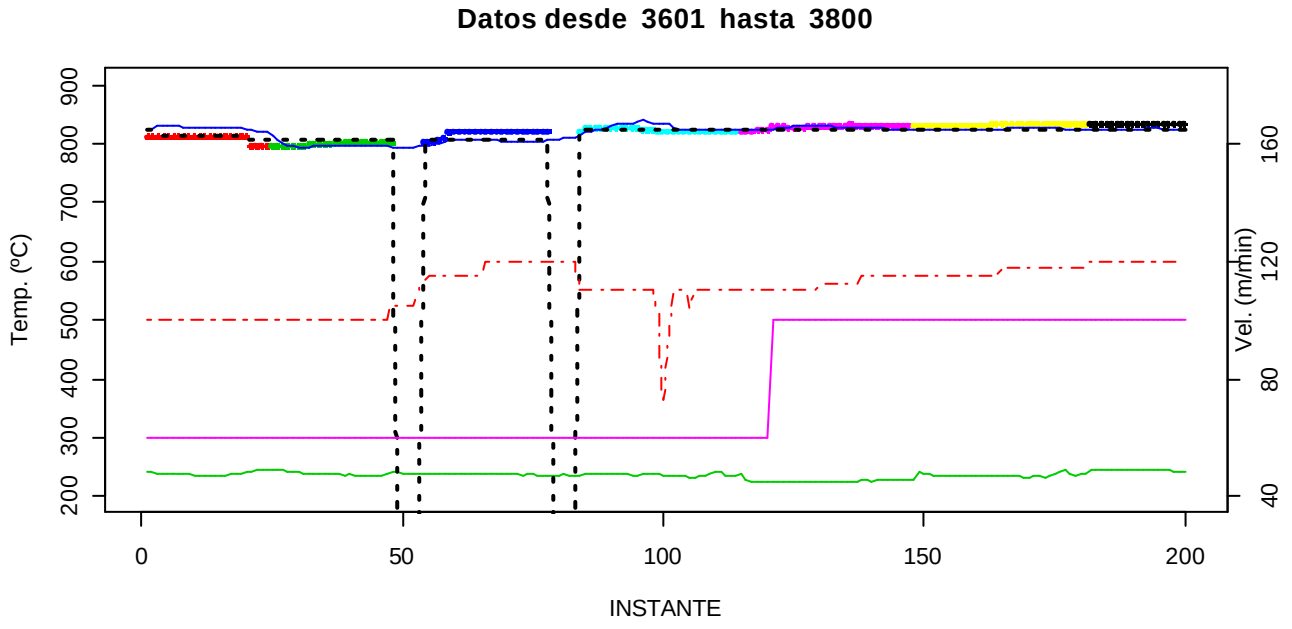


Figura 203. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

Aún así, se observa que siguen existiendo pérdidas de datos. Fundamentalmente estas pérdidas se producen en el las transiciones de bobinas, al final de una bobina hasta el inicio de la siguiente (ver figuras anteriores y siguientes).

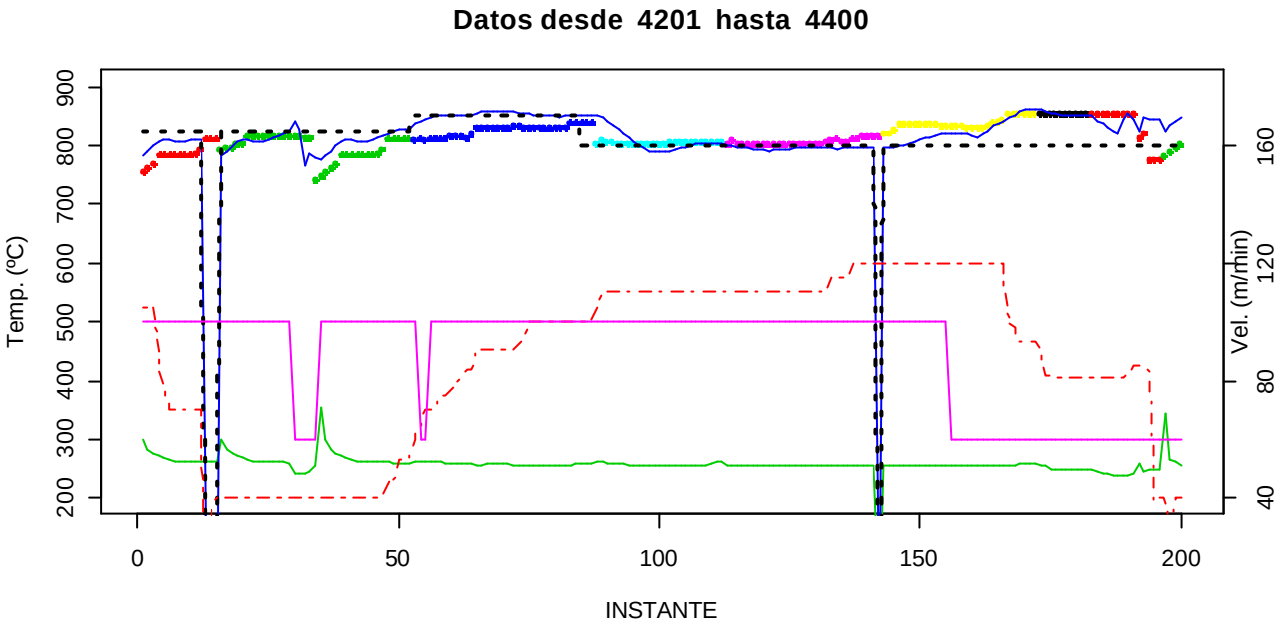


Figura 204. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

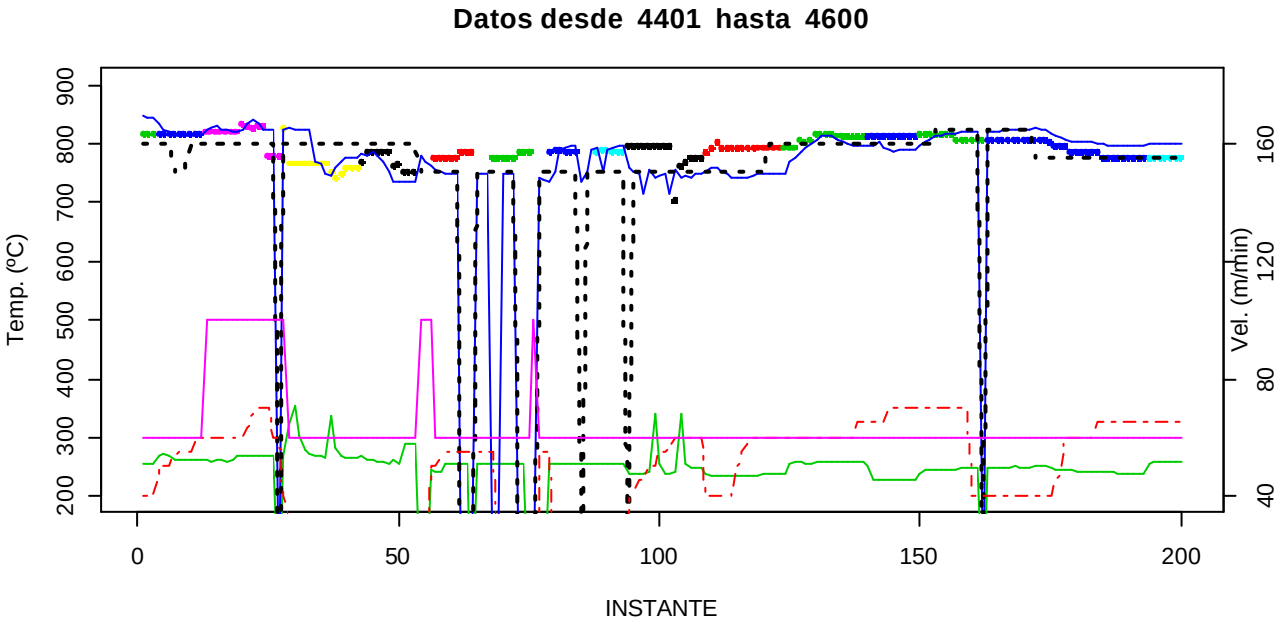


Figura 205. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

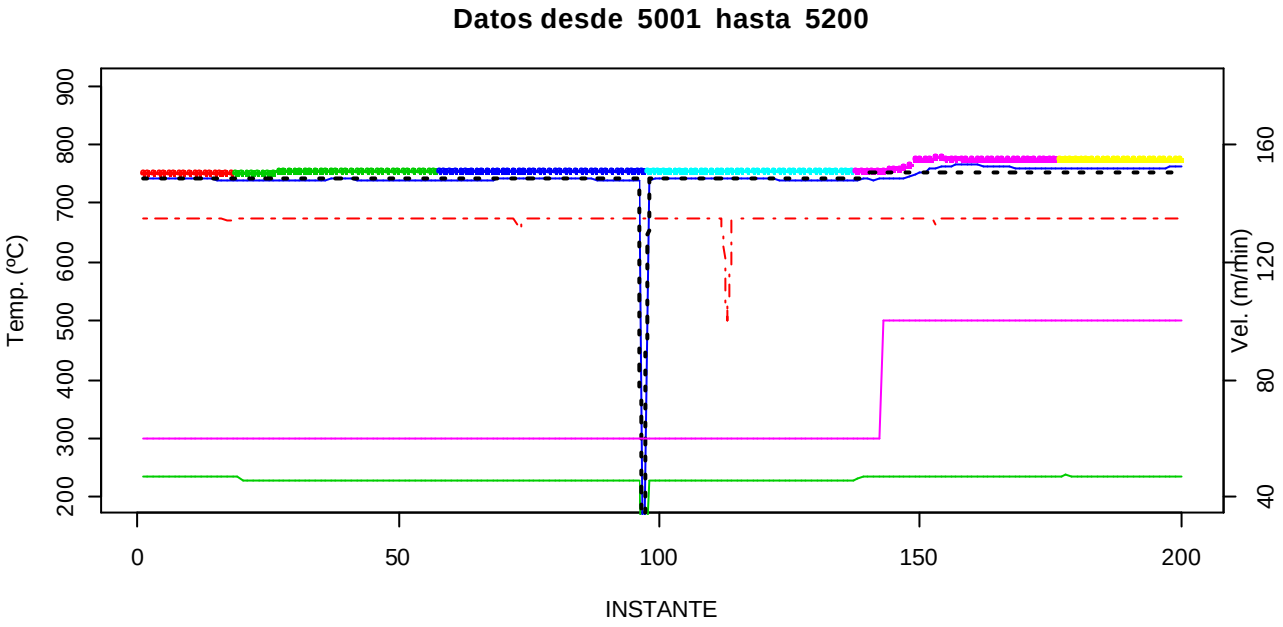


Figura 206. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

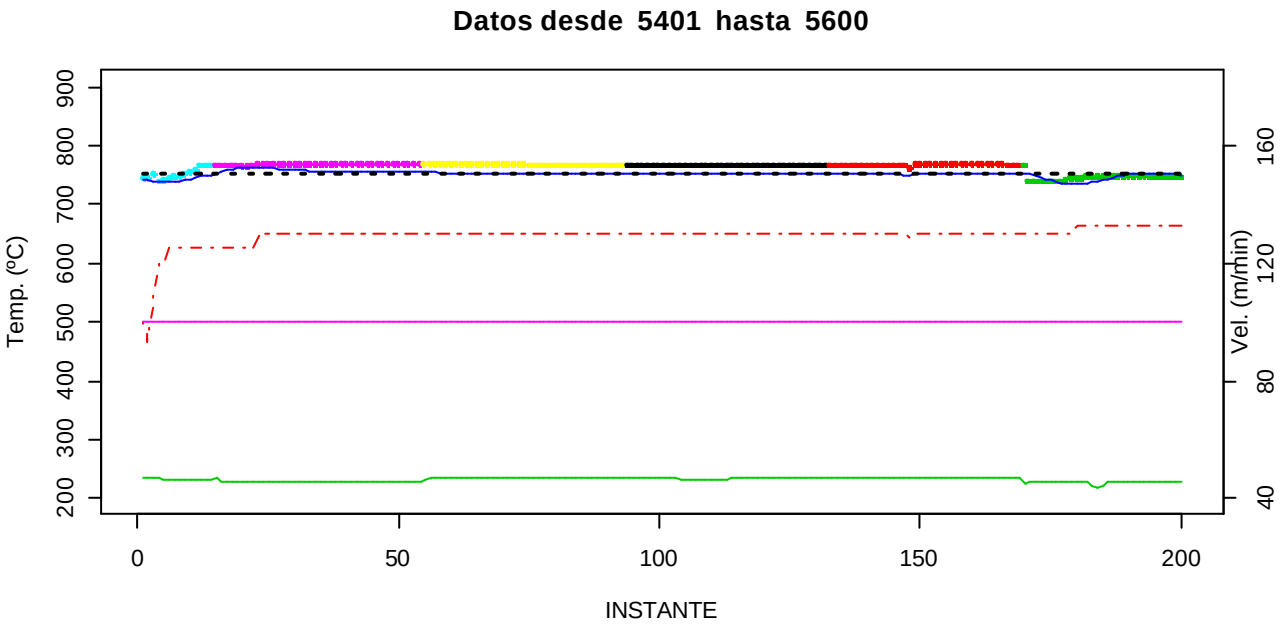


Figura 207. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

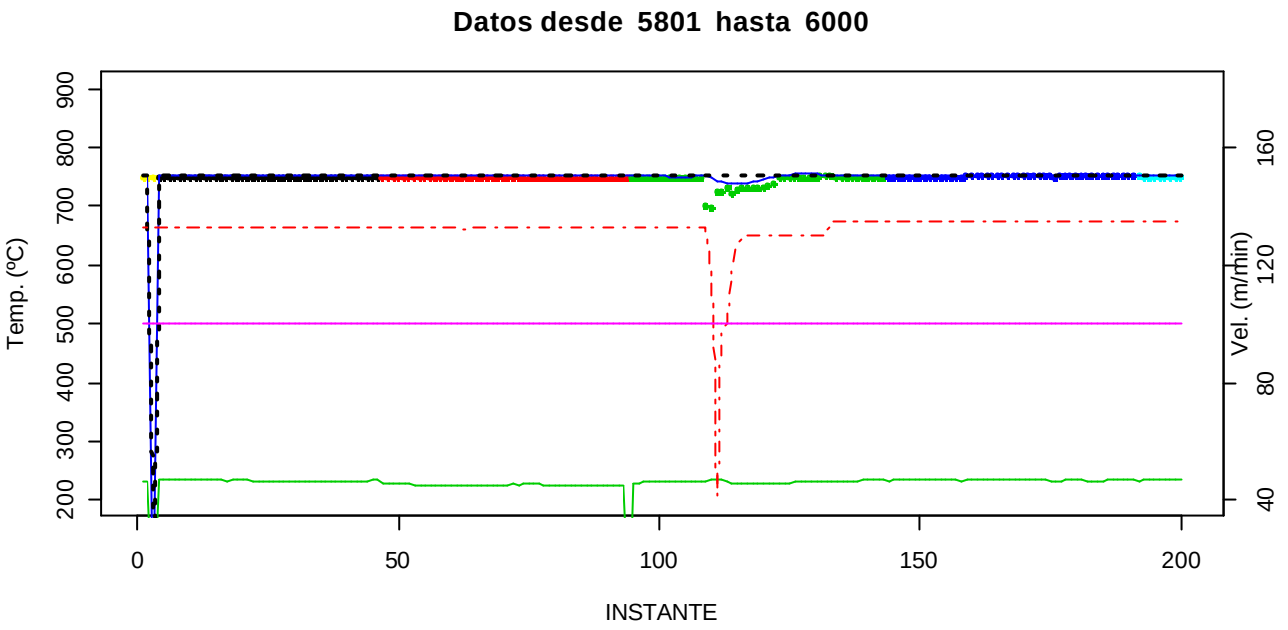


Figura 208. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

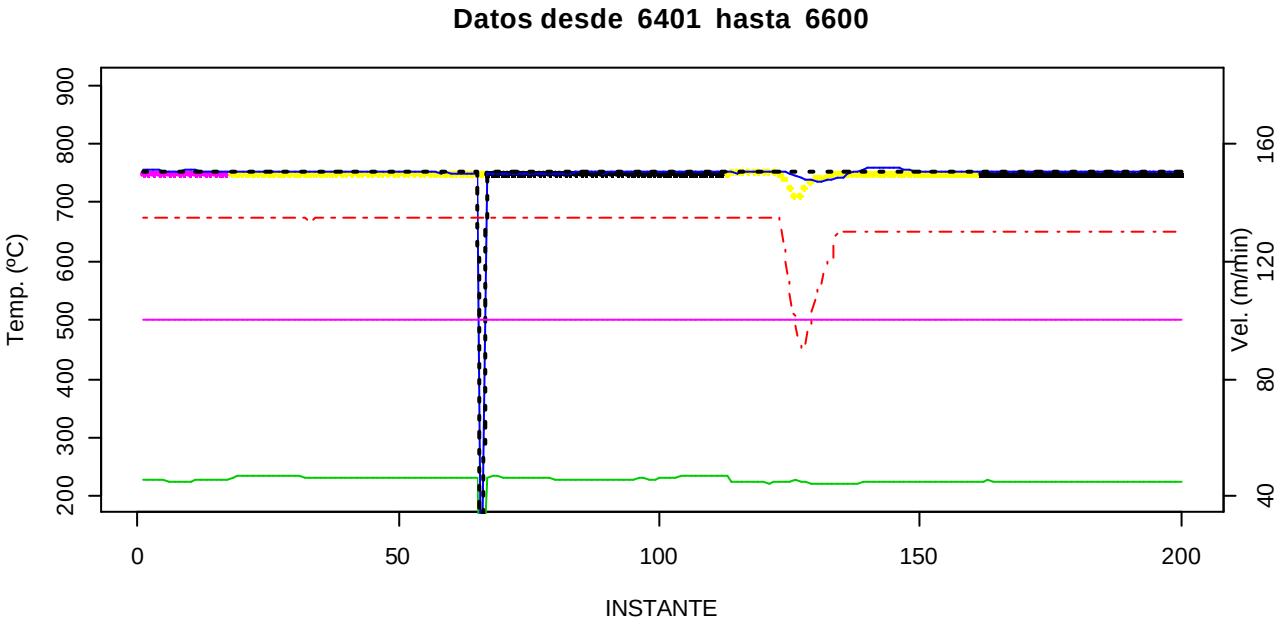


Figura 209. Gráfico ejemplo del comportamiento dinámico de una serie de bobinas.

5.6.2 SELECCIÓN DE LAS VARIABLES FINALES A UTILIZAR PARA ESTUDIAR LA ZONA DE CALENTAMIENTO DEL HORNO

Después de realizado el estudio exploratorio y analizadas las diferentes bases de datos de históricos del proceso de galvanizado del proceso, se han seleccionado, a partir de la última base de datos, las siguientes variables:

Tabla de la B.D. a la que pertenece	Nombre de la Variable	Formato	Descripción
T100cal	CODBOBINA	Código Numérico	Código de la Bobina
T100cal	INSTANTE	Entero	Instante en que se hace cada medida (cada 100 metros)
T100cal	THF1VALCNG	Entero	Temp. de consigna de la subzona 1 (cada 100 metros)
T100cal	TMPP1VALMED	Entero	Valor de temperatura del pirómetro 1 (cada 100 metros)
T100cal	TMPP2VALMED	Entero	Valor de temperatura del pirómetro 2 (cada 100 metros)
T100cal	TMPP2VALCNG	Entero	Valor de consigna de temperatura del pirómetro 2 (cada 100 metros)
T100cal	MODHF	Entero	Valor de temperatura del pirómetro 3 (cada 100 metros)
T30Acumuladore	VELCENMED	Entero	Velocidad de la banda medida en el centro del horno. Esta velocidad es la misma dentro de la zona comprendida entre los acumuladores de entrada y de salida (en metros por minuto). (cada 30 metros)
dps	ESPENT	Real	Espesor Real de la Bobina a la entrada del horno (en mm.)
dps	CLASACERO	Categórica	Tipo de Acero de la Bobina
dps	DUREZA	Categórica	Dureza del Acero
dps	CICREC	Categórica	Ciclo de Recocido
dps	CALIDAD	Categórica	Calidad de la bobina
dps	ANCHO	Entero	Anchura en mm. de la banda.
dps	ESPESOR	Real	Espesor objetivo de fabricación en mm.
dps	LARGO	Entero	Longitud de la Bobina (en metros)
dps	FEC_FAB	DD-MM-AAAA	Fecha De Fabricación
dps	HOR_FAB	HH:MM	Hora de Fabricación
T30Tijera	ESPTOTMED	Numérica	Espesor + recubrimiento en cada instante (en mm.) (cada 30 metros)

Tabla 29. Variables a capturar para estudiar la zona de calentamiento del horno.

5.6.2.1 JUSTIFICACIÓN

Las variables mostradas en la Tabla 29 se han seleccionado a partir de los estudios realizados en puntos anteriores. Su justificación se ha basado en las conclusiones siguientes:

- Se observó que las variables *THFxVALMAX*, *THFxVALMIN*, *TMPPxVALMIN*, *TMPPxVALMAX* **no daban valores fiables**, ya que el valor medio sobrepasaba en muchos momentos por encima o debajo el valor de estas variables[17]. Por lo tanto, **las variables más fiables para cada instante de la bobina son las que nos dan el valor medio.**
- **Sólo se utilizarán para el estudio las variables de consigna de temperatura del horno correspondientes a la zona 1 (THF1VALCNG),** ya que se considera que puede utilizarse como muestra representativa de las demás zonas porque en casi todas las bobinas los *STEPS* no varían en demasía. Además, como se ha verificado que la temperatura real del horno, gracias a la básica, sigue fielmente la temperatura de consigna, solo se utiliza esta última.
- **El valor de las temperaturas de consigna *TMPP1CNG* y *TMPP3CNG* son cero**, ya que el sistema no tiene valores de consigna para estos pirómetros. El valor de consigna apropiado, es decir, la temperatura buscada que debe tener la banda a la salida de la zona de calentamiento, debe ser *TMPP2CNG* y ésta es la misma que la del pirómetro 3. Como conclusión, solo se utiliza la temperatura de consiga del pirómetro 2.
- La velocidad de la banda, es la misma para las diferentes zonas del horno, por lo tanto, **solo es necesario utilizar *VELCENMED***. En la nueva base de datos, el número de errores ha descendido considerablemente aunque, aún así, será conveniente realizar un filtrado para eliminar los existentes.
- **Se utilizarán también las dimensiones de la banda**, ya que la temperatura de ésta dependerá, como es lógico, de sus características físicas además de otras variables.
- El espesor del recubrimiento de zinc se utiliza para determinar si errores elevados en la temperatura de la banda afectan a la calidad y espesor del mismo. **Se analizarán los espesores reales de la banda frente a los finales, para determinar el espesor de la capa de galvanizado.**
- Se desarrollarán nuevas variables numéricas basadas en indicadores y categóricas que indiquen el tipo de curvas para cada bobina.

[17] Consultado personal de la empresa, nos ratificaron que el valor más fiable era el del valor medio, ya que la electrónica de adquisición no daba buenos valores máximos y mínimos.

5.6.2.2 VARIABLES FINALES

Las variables seleccionadas se han incluido en dos matrices:

- DATBOBINAS: Datos de cada bobina.
- MATDINAMIC: Datos dinámicos ya preparados del proceso (medias de valores cada 100 metros de banda)

Tabla de la B.D. de donde procede	Nombre de la Variable	Formato	Descripción
Dps	CODBOBINA	Código Numérico	Código de la Bobina
Dps	BOBENT	Categórica	Código de fabricación Bobina
Dps	ESPENT	Real	Espesor Real de la Bobina a la entrada del horno (en mm.)
Dps	CLASACERO	Categórica	Tipo de Acero de la Bobina
Dps	DUREZA	Categórica	Dureza del Acero
Dps	CICREC	Categórica	Ciclo de Recocido
Dps	ANCHO	Entero	Anchura en mm. de la banda.
Dps	ESPESOR	Real	Espesor objetivo de fabricación en mm.
dps	LARGO	Entero	Longitud de la Bobina (en metros)
dps	PESO	Real	Peso en kg.
dps	CALIDAD	Categórica	Calidad de la bobina
dps	FEC_FAB	DD-MM-AAAA	Fecha De Fabricación
dps	HOR_FAB	HH:MM	Hora de Fabricación

Tabla 30. Variables de la matriz DATBOBINAS.

Tabla de la B.D. de donde procede	Nombre de la Variable	Formato	Descripción
T100cal	CODBOBINA	Código Numérico	Código de la Bobina
T100cal	INSTANTE	Entero	Instante en que se hace cada medida (cada 100 metros)
T100cal (de THF1VALCNG)	THC1	Entero	Temp. de consigna de la subzona 1 (cada 100 metros)
T100cal (TMPP1VALMED)	TMPP1M	Entero	Valor de temperatura del pirómetro 1 (cada 100 metros)
T100cal (TMPP2VALMED)	TMPP2M	Entero	Valor de temperatura del pirómetro 2 (cada 100 metros)
T100cal (TMPP2VALCNG)	TMPP2C	Entero	Valor de consigna de temperatura del pirómetro 2 (cada 100 metros)
T30Acumuladore (de VELCENMED)	VELOCIDADFIN	Entero	Velocidad de la banda medida en el centro del horno. (en metros por minuto). (ajustada a medidas cada 100 metros)
Nueva Variable	COLORBOB	Entero	Color asignado a la bobina. Número de 1 a 7 obtenido del código de bobina (CODBOBINA%7+1)
T100cal	MODHF	Entero	Valor de temperatura del pirómetro 3 (cada 100 metros)

Tabla 31. Variables de la matriz MATDINAMIC.

5.6.3 CREACIÓN DE NUEVAS VARIABLES

Además de las variables seleccionadas, se ha considerado conveniente la creación de nuevas variables que faciliten los futuros análisis. Éstas se basan en la metodología ya expuesta para la caracterización de las curvas de temperaturas y velocidades.

Una vez creadas, se incluyen en una nueva matriz llamada *MATBOBINAS*.

Tabla de la B.D. a la que pertenece	Nombre de la Variable	Formato	Descripción
T100calentamiento	CODBOBINA	Código Numérico	Código de la Bobina
Nueva Variable	MAXINSTANTE	Entero	Instante máximo de la bobina
T30Tijera	ESPFINAL	Real	Espesor medio final de la banda con el recubrimiento
Nueva Variable	MODOBOB	Categórica	Modo de operación en el que se ha trabajado la bobina con un tiempo mayor del 50%. (0="Manual", -1="Automático")
Nueva Variable	SECCION	Real	Espesor * Anchura (en mm^2.)
Nueva Variable	THF1DIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura de consigna para cada bobina en la zona 1.
Nueva Variable	THF1MEDTOTAL	Entero	Temperatura de consigna media para cada bobina en la zona 1.
Nueva Variable	TIPOCURVATHF1	Categórica	Tipo de Curva de la temperatura de consigna en la zona 1.
Nueva Variable	TMPPxMEDTOTAL	Entero	Temperatura media leída por el pirómetro x (x vale 1 o 2)
Nueva Variable	TMPPxDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura del pirómetro x para cada bobina (x vale 1, 2)
Nueva Variable	TIPOCURVATMPPx	Categórica	Tipo de Curva de la temperatura del pirómetro x (x vale 1, 2).
Nueva Variable	TMPP2CNGMEDTOTAL	Entero	Temperatura de consigna del pirómetro 2.
Nueva Variable	TMPP2CNGDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura de consigna del pirómetro 2
Nueva Variable	TIPOCURVATMPP2CNG	Categórica	Tipo de Curva de la temperatura de consigna del pirómetro 2.
Nueva Variable	VELMEDTOTAL	Entero	Velocidad de la banda.
Nueva Variable	VELDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la velocidad.
Nueva Variable	TIPOCURVAVEL	Categórica	Tipo de Curva de la velocidad de la banda.
Nueva Variable	ERRORMEDTOTAL	Entero	Valor del error medio obtenido de la diferencia del valor medido por el pirómetro 2 y el de consigna.
Nueva Variable	ERRORMEDTOTALABS	Entero	Valor del error medio absoluto del valor absoluto de la diferencia del valor medido por el pirómetro 2 y el de consigna.
Nueva Variable	ERRORDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo del error del pirómetro para cada bobina.
Nueva Variable	TIPOCURVAERROR	Categórica	Tipo de Curva de la temperatura del error.

Tabla 32. Variables de la matriz MATBOBINAS.

5.6.3.1 CREACIÓN DE VARIABLES MAXINSTANTE, ESPFINAL, MODOBOB, SECCIÓN

El programa siguiente, muestra los pasos realizados para la obtención de las nuevas variables:

- MAXINSTANTE: Que corresponde con el último instante de cada bobina.
- ESPFINAL: Espesor final de la banda con el recubrimiento de zinc.
- MODOBOB: Modo de operación del horno en esa bobina.
- SECCIÓN: Anchura por espesor inicial de la banda (en mm^2).

```
# ...  continúa del programa anterior

# Determinamos que bobinas han sido trabajadas en “modo manual” y “modo autom”
MODOBOB <- tapply(MATDINAMIC$MODHF, MATDINAMIC$CODBOBINA, median)

# Obtenemos el instante máximo de cada bobina
MAXINSTANTE <- tapply(MATDINAMIC$INSTANTE, MATDINAMIC$CODBOBINA,max)

# Obtenemos la sección de cada bobina
SECCION <- DATBOBINAS$ESPENT * DATBOBINAS$ANCHO

################################################
# Obtenemos datos de espesor final (ESPFINAL) #
################################################

DATESPESOR <- sqlQuery(canal, “SELECT CODBOBINA, INSTANTE, ESPTOTMED FROM
T30Tijera”)

#Eliminamos las bobinas que no están en la lista de las otras bases de datos
POSLISTA <- DATESPESOR$CODBOBINA %in% LISTABOBINASVELBUENAS
DATESPESORFINAL <- DATESPESOR[POSLISTA,]

# Ordenamos las listas
NUMORDER <- order(DATESPESORFINAL$CODBOBINA)

# Espesores Ordenados
ESPORDENADOS <- DATESPESORFINAL[NUMORDER,]

# Obtenemos una nueva lista de bobinas
LISTABOBINASESP <- unique(ESPORDENADOS$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTABOBINASESP)
TRUE
1979

# Determinamos el valor medio del ESPESOR FINAL usando la mediana
ESPFINAL <- tapply(ESPORDENADOS$ESPTOTMED, ESPORDENADOS$CODBOBINA, median)
```

Figura 210. Programa que obtiene las variables MAXINSTANTE, ESPFINAL, MODOBOB y SECCIÓN.

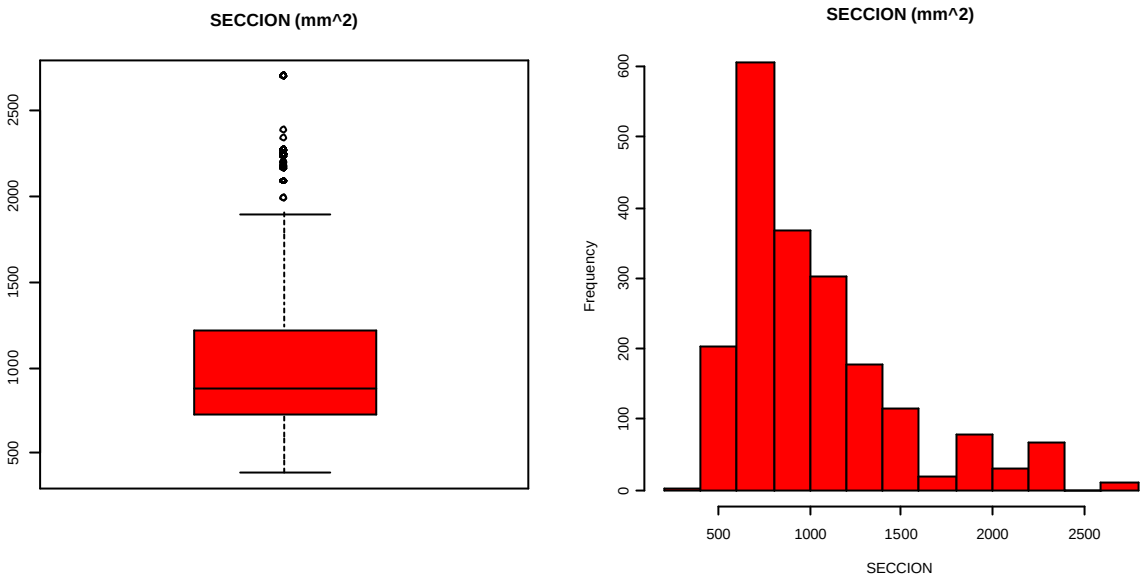


Figura 211. Diagrama boxplot e histograma de las secciones de cada bobina.

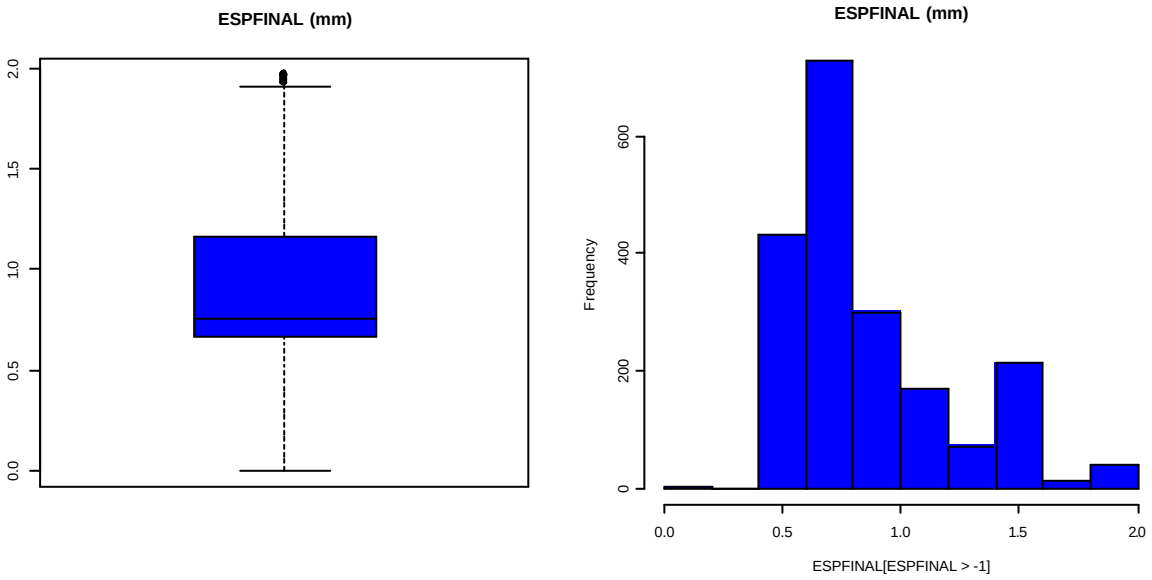


Figura 212. Diagrama boxplot e histograma de los espesores finales de cada bobina.

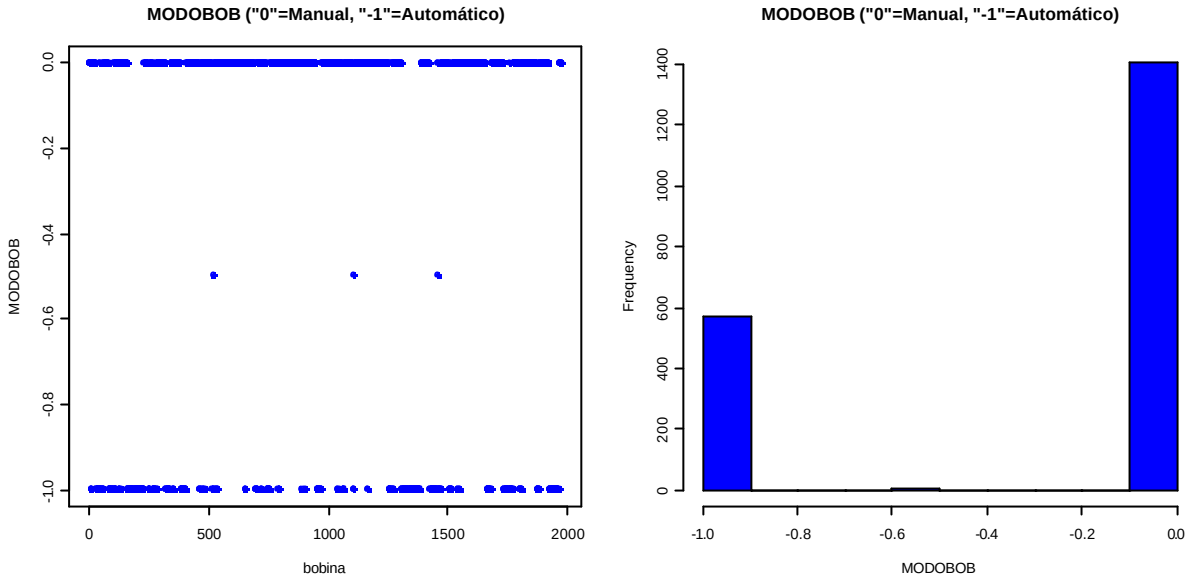


Figura 213. Distribución en el tiempo de los modos de operación de cada bobina. El valor "0" es "Modo Manual", el "-1" es "Modo Automático" y el "-0.5" corresponde a bobinas donde se ha trabajado el 50% en cada modo.

Tal y como se explicó en apartados anteriores, se procede a generar un nuevo tipo de variables, para cada bobina, que expliquen las curvas de comportamiento de las temperaturas de las subzonas de la zona de calentamiento del horno.

5.6.3.2 CARACTERIZACIÓN DE LAS CURVAS DE LAS TEMPERATURAS DE CONSIGNA DE LAS SUBZONA 1 DEL HORNO

Procedemos a obtener las nuevas variables que caracterizan las curvas de las temperaturas de consigna de la subzona 1.

```
# Caracterización de la curva de temperaturas de consigna
# de subzona 1
############################################################

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC$THC1>100

MATSINRUIDO <- MATDINAMIC[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(ESPORDENADOS$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979
# Obtenemos el valor de consigna máximo y mínimo de temperatura por bobina
MINCNG <- tapply(MATSINRUIDO$THC1, MATSINRUIDO$CODBOBINA,min)
MAXCNG <- tapply(MATSINRUIDO$THC1, MATSINRUIDO$CODBOBINA,max)
DIFTOTAL <- MAXCNG-MINCNG
MAXDIFTOTAL <- max(DIFTOTAL)

# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(MATSINRUIDO$THC1, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos la consigna de temperatura al principio y al final de cada bobina
MATCNGINI <- MATSINRUIDO[c(TRUE,(MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-
1),]$CODBOBINA!= MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA)),]$THC1
MATCNGFIN <- MATSINRUIDO[MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-1),]$CODBOBINA!=
MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA,]$THC1

# Obtenemos las variables finales
THF1MEDTOTAL <- round(VALMEAN)
THF1DIFTOTAL <- MAXCNG-MINCNG
THF1DIFTIEMPOTOTAL <- MATCNGFIN[1:length(THF1DIFTOTAL)]-
MATCNGINI[1:length(THF1DIFTOTAL)]

# Obtenemos la variable categórica
TIPOCURVATHF1 <- rep("H",length(THF1DIFTOTAL))
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=10 & abs(THF1DIFTOTAL)<30 &
THF1DIFTIEMPOTOTAL>=0] <- “BC”
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=10 & abs(THF1DIFTOTAL)<30 &
THF1DIFTIEMPOTOTAL<0] <- “BD”
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=30 & abs(THF1DIFTOTAL)<60 &
THF1DIFTIEMPOTOTAL>=0] <- “MC”
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=30 & abs(THF1DIFTOTAL)<60 &
THF1DIFTIEMPOTOTAL<0] <- “MD”
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=60 & abs(THF1DIFTOTAL)<200 &
THF1DIFTIEMPOTOTAL>=0] <- “AC”
```

```
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=60 & abs(THF1DIFTOTAL)<200 &
THF1DIFTIEMPOTOTAL<0] <- "AD"
TIPOCURVATHF1[abs(THF1DIFTOTAL)>=200] <- "E"

table(TIPOCURVATHF1)
TIPOCURVATHF1
  AC   AD   BC   BD    H   MC   MD
   6   34  210  133 1449   82   65
```

Figura 214. Programa que obtiene las nuevas variables TIPOCURVATHF1, THF1DIFTOTAL y THF1MEDTOTAL que caracterizan la curva de la temperatura de consigna de la subzona 1.

El programa de la Figura 214 nos genera un archivo con las variables que explican el comportamiento de las curvas de temperatura de consigna de la subzona 1 de la zona de calentamiento del horno para cada bobina.

	Alto Decr. (AD)	Medio Decr. (MD)	Bajo Decr. [BD]	Horizontal (H)	Bajo Crec. (BC)	Medio Crec. (MC)	Alto Crec. (AC)	Error (E)
Tipos de curvas de THFCNG1 en la subzona 1	**34**	**65**	**133**	**1449**	**210**	**82**	**65**	**184**

Tabla 33. Distribución de los tipos de curvas de temperaturas de consigna de la subzona 1.

Claramente se aprecia, que existe un alto porcentaje de curvas horizontales que destaca delos demás.

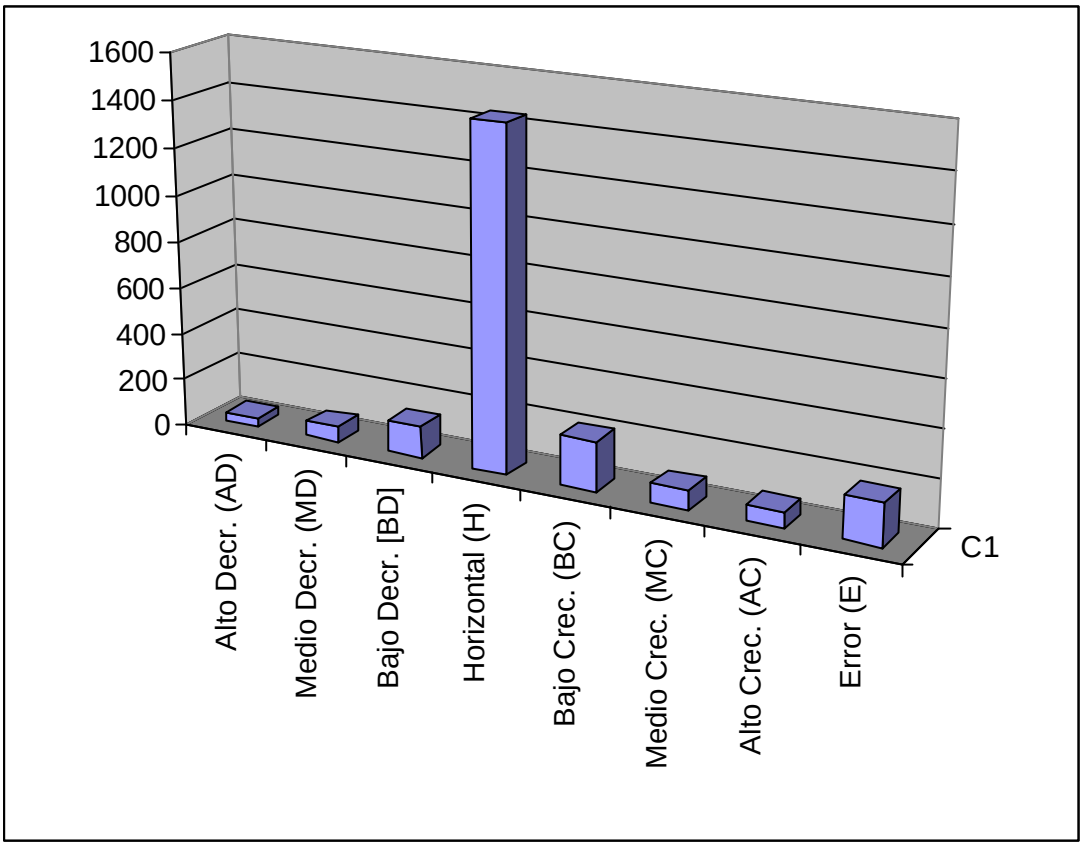


Figura 215. Tipos de curvas de THC1.

Valor de THF1DIFTOTAL	Signo de THF1DIFTIEMPOTOTAL	NOMBRE
< 10 ºC	Positivo o Negativo	HORIZONTAL (H)
>= 10 ºC y < 30ºC	Positivo	BAJACRECIENTE (BC)
>= 10 ºC y < 30ºC	Negativo	BAJADECRECIENTE (BD)
>= 30 ºC y < 60ºC	Positivo	MEDIACRECIENTE (MC)
>= 30 ºC y < 60ºC	Negativo	MEDIADECRECIENTE (MD)
>= 60 ºC y < 200ºC	Positivo	ALTACRECIENTE (AC)
>= 60 ºC y < 200ºC	Negativo	ALTADECRECIENTE (AD)
>= 200	Positivo o Negativo	ERROR (E)

Tabla 34. Categorización de las curvas de temperatura de consigna aplicadas a cada bobina.

5.6.3.3 CARACTERIZACIÓN DE LAS CURVAS DE LAS TEMPERATURAS DE LOS PIRÓMETROS 1 Y 2

Igual que en el caso anterior, se obtienen las variables que nos ayudarán a caracterizar las curvas, para cada bobina, de las temperaturas de la banda que leen los pirómetros 1 y 2.

```
#########################################################
# Caracterización de la curva de temperaturas de     #
# pirómetros 1 y 2                                    #
#########################################################

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC$TMPP1M>100

MATSINRUIDO <- MATDINAMIC[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979


#############################################################
# Caracterización de la curva de temperaturas leídas
# por el pirómetro 1
#############################################################

# Obtenemos el valor de consigna máximo y mínimo de temperatura por bobina
MINTMPP <- tapply(MATSINRUIDO$TMPP1M, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(MATSINRUIDO$TMPP1M, MATSINRUIDO$CODBOBINA,max)

# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(MATSINRUIDO$TMPP1M, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos la consigna de temperatura al principio y al final de cada bobina
MATTMPPINI <- MATSINRUIDO[c(TRUE,(MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-
1),]$CODBOBINA!= MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA)),]$TMPP1M
MATTMPPFIN <- MATSINRUIDO[MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-1),]$CODBOBINA!=
MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA,]$TMPP1M

# Buscamos la posición de los máximos y mínimos de cada bobina
BOBMASTMPP1 <- (MATSINRUIDO$CODBOBINA*1000)+ MATSINRUIDO$TMPP1M
MAXEXP <- tapply(BOBMASTMPP1, MATSINRUIDO$CODBOBINA,max)
MINEXP <- tapply(BOBMASTMPP1, MATSINRUIDO$CODBOBINA,min)

#Detectamos la posición de los valores MAXEXP y MINEXP
POSMINEXP <- match (MINEXP,BOBMASTMPP1)
POSMAXEXP <- match (MAXEXP,BOBMASTMPP1)
```

```
# Determinamos si MAX está antes que MIN
MAXANTESMIN <- (POSMAXEXP <= POSMINEXP)

# Obtenemos las variables finales
TMPP1MEDTOTAL <- round(VALMEAN)
TMPP1DIFTOTAL <- MAXTMPP-MINTMPP

#Calculamos si la diferencia entre los máximos, mínimos, con
#inicio o final es mayor de 1/4 de TMPP1DIFTOTAL
MAXCONINI <- abs(MAXTMPP- MATTMPPINI)>(abs(TMPP1DIFTOTAL/4))
MINCONINI <- abs(MINTMPP- MATTMPPINI)>(abs(TMPP1DIFTOTAL/4))
MAXCONFIN <- abs(MAXTMPP- MATTMPPFIN)>(abs(TMPP1DIFTOTAL/4))
MINCONFIN <- abs(MINTMPP- MATTMPPFIN)>(abs(TMPP1DIFTOTAL/4))

# Obtenemos la variable categórica
TIPOCURVATMPP1 <- rep("H",length(TMPP1DIFTOTAL))
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>10 & abs(TMPP1DIFTOTAL)<=20 &
MAXANTESMIN==FALSE] <- “BRC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>10 & abs(TMPP1DIFTOTAL)<=20 &
MAXANTESMIN==TRUE] <- “BRD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- “MRC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- “MRD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- “MCVAC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- “MCXAD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- “MCXAC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- “MCVAD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- “MOMINMAX”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>20 & abs(TMPP1DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- “MOMAXMIN”


TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- “ARC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- “ARD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- “ACVAC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- “ACXAD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- “ACXAC”
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- “ACVAD”

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- “AOMINMAX”
```

```
TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>40 & abs(TMPP1DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- "AOMAXMIN"

TIPOCURVATMPP1[abs(TMPP1DIFTOTAL)>200] <- "E"

#####################################################################




#############################################################
# Caracterización de la curva de temperaturas leídas
# por el pirómetro 2
#############################################################


# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC$TMPP2M>100

MATSINRUIDO <- MATDINAMIC[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979

# Obtenemos el valor de consigna máximo y mínimo de temperatura por bobina
MINTMPP <- tapply(MATSINRUIDO$TMPP2M, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(MATSINRUIDO$TMPP2M, MATSINRUIDO$CODBOBINA,max)

# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(MATSINRUIDO$TMPP2M, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos la consigna de temperatura al principio y al final de cada bobina
MATTMPPINI <- MATSINRUIDO[c(TRUE,(MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-
1),]$CODBOBINA!= MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA)),]$TMPP2M
MATTMPPFIN <- MATSINRUIDO[MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-1),]$CODBOBINA!=
MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA,]$TMPP2M

# Buscamos la posición de los máximos y mínimos de cada bobina
BOBMASTMPP2 <- (MATSINRUIDO$CODBOBINA*1000)+ MATSINRUIDO$TMPP2M
MAXEXP <- tapply(BOBMASTMPP2, MATSINRUIDO$CODBOBINA,max)
MINEXP <- tapply(BOBMASTMPP2, MATSINRUIDO$CODBOBINA,min)

#Detectamos la posición de los valores MAXEXP y MINEXP
POSMINEXP <- match (MINEXP,BOBMASTMPP2)
POSMAXEXP <- match (MAXEXP,BOBMASTMPP2)

# Determinamos si MAX está antes que MIN
MAXANTESMIN <- (POSMAXEXP <= POSMINEXP)

# Obtenemos las variables finales
TMPP2MEDTOTAL <- round(VALMEAN)
TMPP2DIFTOTAL <- MAXTMPP-MINTMPP
```

```
#Calculamos si la diferencia entre los máximos, mínimos, con
#inicio o final es mayor de 1/4 de TMPP2DIFTOTAL
MAXCONINI <- abs(MAXTMPP- MATTMPPINI)>(abs(TMPP2DIFTOTAL/4))
MINCONINI <- abs(MINTMPP- MATTMPPINI)>(abs(TMPP2DIFTOTAL/4))
MAXCONFIN <- abs(MAXTMPP- MATTMPPFIN)>(abs(TMPP2DIFTOTAL/4))
MINCONFIN <- abs(MINTMPP- MATTMPPFIN)>(abs(TMPP2DIFTOTAL/4))


# Obtenemos la variable categórica
TIPOCURVATMPP2 <- rep("H",length(TMPP2DIFTOTAL))
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>10 & abs(TMPP2DIFTOTAL)<=20 &
MAXANTESMIN==FALSE] <- "BRC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>10 & abs(TMPP2DIFTOTAL)<=20 &
MAXANTESMIN==TRUE] <- "BRD"


TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- "MRC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- "MRD"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- "MCVAC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- "MCXAD"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- "MCXAC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- "MCVAD"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- "MOMINMAX"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>20 & abs(TMPP2DIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- "MOMAXMIN"


TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- "ARC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- "ARD"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- "ACVAC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- "ACXAD"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- "ACXAC"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- "ACVAD"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- "AOMINMAX"
TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>40 & abs(TMPP2DIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- "AOMAXMIN"

TIPOCURVATMPP2[abs(TMPP2DIFTOTAL)>200] <- "E"

#################################################################
```

```
############################################################
# Caracterización de la curva de temperaturas de consigna
# del pirómetro 2
############################################################


# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC$TMPP2C>100

MATSINRUIDO <- MATDINAMIC[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979

# Obtenemos el valor de consigna máximo y mínimo de temperatura por bobina
MINTMPP <- tapply(MATSINRUIDO$TMPP2C, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(MATSINRUIDO$TMPP2C, MATSINRUIDO$CODBOBINA,max)

# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(MATSINRUIDO$TMPP2C, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos la consigna de temperatura al principio y al final de cada bobina
MATTMPPINI <- MATSINRUIDO[c(TRUE,(MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-
1),]$CODBOBINA!= MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA)),]$TMPP2C
MATTMPPFIN <- MATSINRUIDO[MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-1),]$CODBOBINA!=
MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA,]$TMPP2C

# Buscamos la posición de los máximos y mínimos de cada bobina
BOBMASTMPP2CNG <- (MATSINRUIDO$CODBOBINA*1000)+ MATSINRUIDO$TMPP2C
MAXEXP <- tapply(BOBMASTMPP2CNG, MATSINRUIDO$CODBOBINA,max)
MINEXP <- tapply(BOBMASTMPP2CNG, MATSINRUIDO$CODBOBINA,min)

#Detectamos la posición de los valores MAXEXP y MINEXP
POSMINEXP <- match (MINEXP,BOBMASTMPP2CNG)
POSMAXEXP <- match (MAXEXP,BOBMASTMPP2CNG)

# Determinamos si MAX está antes que MIN
MAXANTESMIN <- (POSMAXEXP <= POSMINEXP)

# Obtenemos las variables finales
TMPP2CNGMEDTOTAL <- round(VALMEAN)
TMPP2CNGDIFTOTAL <- MAXTMPP-MINTMPP

#Calculamos si la diferencia entre los máximos, mínimos, con
#inicio o final es mayor de 1/4 de TMPP2CNGDIFTOTAL
MAXCONINI <- abs(MAXTMPP- MATTMPPINI)>(abs(TMPP2CNGDIFTOTAL/4))
MINCONINI <- abs(MINTMPP- MATTMPPINI)>(abs(TMPP2CNGDIFTOTAL/4))
MAXCONFIN <- abs(MAXTMPP- MATTMPPFIN)>(abs(TMPP2CNGDIFTOTAL/4))
MINCONFIN <- abs(MINTMPP- MATTMPPFIN)>(abs(TMPP2CNGDIFTOTAL/4))
```

```
# Obtenemos la variable categórica
TIPOCURVATMPP2CNG <- rep("H",length(TMPP2CNGDIFTOTAL))
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>10 & abs(TMPP2CNGDIFTOTAL)<=20 &
MAXANTESMIN==FALSE] <- “BRC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>10 & abs(TMPP2CNGDIFTOTAL)<=20 &
MAXANTESMIN==TRUE] <- “BRD”


TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- “MRC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- “MRD”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- “MCVAC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- “MCXAD”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- “MCXAC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- “MCVAD”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- “MOMINMAX”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>20 & abs(TMPP2CNGDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- “MOMAXMIN”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- “ARC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- “ARD”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- “ACVAC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- “ACXAD”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- “ACXAC”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- “ACVAD”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- “AOMINMAX”
TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>40 & abs(TMPP2CNGDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- “AOMAXMIN”

TIPOCURVATMPP2CNG[abs(TMPP2CNGDIFTOTAL)>200] <- “E”

#################################################################
```

```
################################
# Obtenemos los tipos de Curvas #
################################


table(TIPOCURVATMPP1)
TIPOCURVATMPP1
   ACVAC     ACXAC     ACXAD AOMAXMIN AOMINMAX       ARC       ARD       BRC
       1        15         2        1         1         4         5       178
     BRD         H     MCVAC     MCVAD     MCXAC     MCXAD MOMAXMIN MOMINMAX
     263      1441         5         8         9         3         1         4
     MRC       MRD
      15        23

table(TIPOCURVATMPP2)
TIPOCURVATMPP2
   ACVAC     ACVAD     ACXAC     ACXAD AOMAXMIN AOMINMAX       ARC       ARD
       9        17        12        10         6         5        44        35
     BRC       BRD         H     MCVAC     MCVAD     MCXAC     MCXAD MOMAXMIN
     174       199      1179        28        33        45        25        10
MOMINMAX       MRC       MRD
      11        80        57

table(TIPOCURVATMPP2CNG)
TIPOCURVATMPP2CNG
   ACVAD     ACXAC AOMAXMIN       ARC       ARD       BRC       BRD         H
       2         1         1        34        26        15        18      1767
   MCVAC     MCVAD     MCXAC     MCXAD       MRC       MRD
       1         3         1         1        57        52
```

Figura 216. Programa que obtiene las nuevas variables TIPOCURVATMPPx, TMPPxDIFTOTAL y TMPPxMEDTOTAL que caracterizan la curva de la temperatura dada por los pirómetros 1 y 2; así como la temperatura de consigna del pirómetro 2.

En este caso, el programa de la Figura 216 nos genera un archivo con las variables que explican el comportamiento de las curvas de temperatura de la banda en dos puntos dentro de la zona de calentamiento del horno para cada bobina.

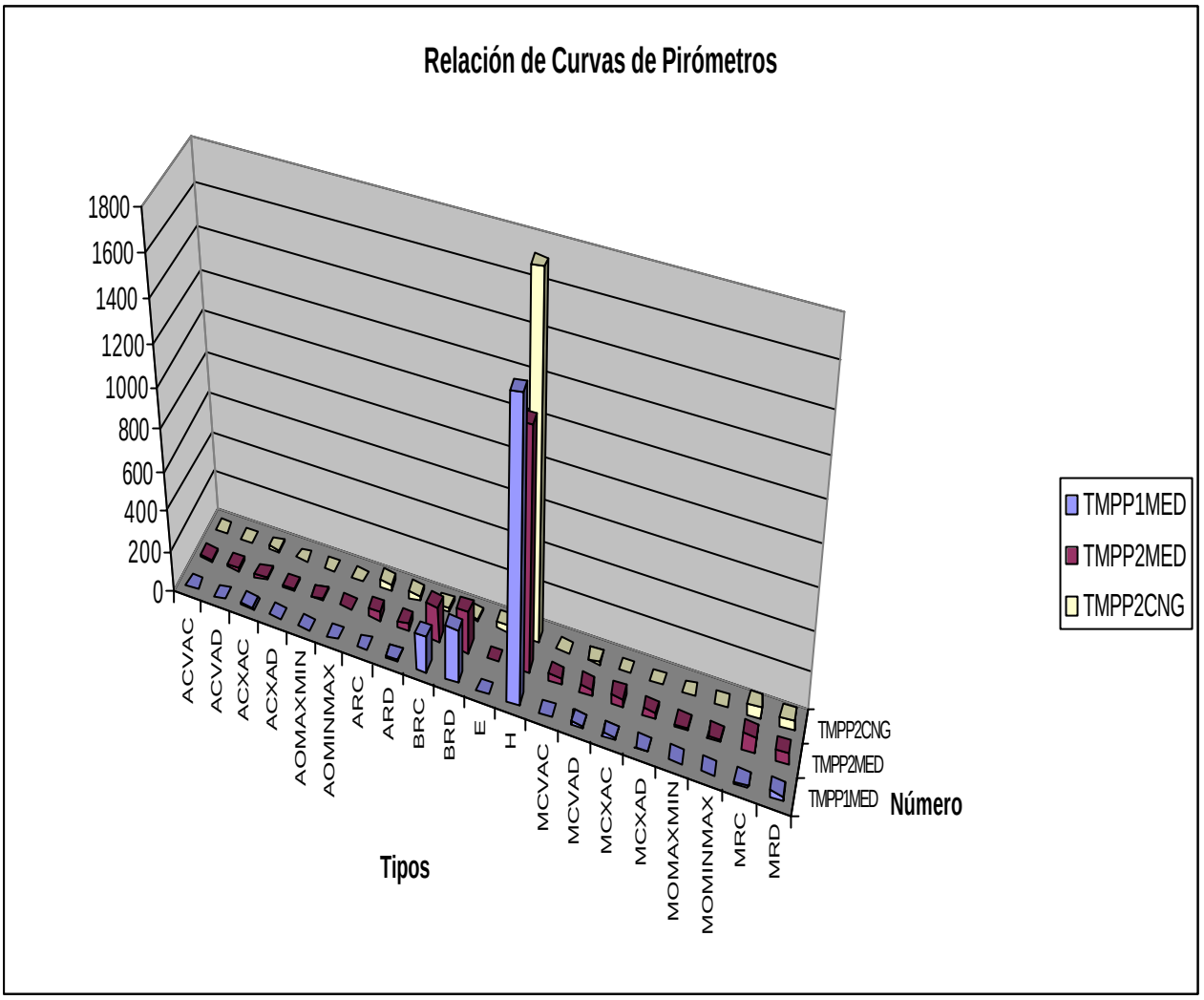


Figura 217. Distribución de las curvas de temperatura de los pirómetros.

	ACVAC	ACVAD	ACXAC	ACXAD	AOMAXMIN	AOMINMAX	ARC	ARD	BRC	BRD	E
TMPP1MED	1	0	15	2	1	1	4	5	178	263	0
TMPP2MED	9	17	12	10	6	5	44	35	174	199	0
TMPP2CNG	0	2	1	0	1	0	34	26	15	18	0

Tabla 35. Distribución de los tipos de curvas de temperaturas de consigna de los pirómetros 1, 2 y 3.

	H	MCVAC	MCVAD	MCXAC	MCXAD	MOMAXMIN	MOMINMAX	MRC	MRD
TMPP1MED	1441	5	8	9	3	1	4	15	23
TMPP2MED	1179	28	33	45	25	10	11	80	57
TMPP2CNG	1767	1	3	1	1	0	0	57	52

Tabla 36. Distribución de los tipos de curvas de temperaturas de consigna de los pirómetros 1, 2 y 3.

Valor de TMPPxDIFTOTAL	Aparece Primero	Aparece Segundo	ABS(T(INI)-T(1º)) > ¼ de TMPPxDIFTOTAL	ABS(T(FIN)-T(2º)) > ¼ de TMPPxDIFTOTAL	VALOR DE TIPOCURVATMPPx
<= 10ºC	-	-	Indiferente	Indiferente	HORIZONTAL (H)
>10ºC y <=20ºC	MIN	MAX	Indiferente	Indiferente	BAJA RECTA CRECIENTE (BRC)
>10ºC y <=20ºC	MAX	MIN	Indiferente	Indiferente	BAJA RECTA DECRECIENTE (BRD)
>20ºC y <=40ºC	MIN	MAX	NO	NO	MEDIA RECTA CRECIENTE (MRC)
>20ºC y <=40ºC	MAX	MIN	NO	NO	MEDIA RECTA DECRECIENTE (MRD)
>20ºC y <=40ºC	MIN	MAX	SI	NO	MEDIA CÓNCAVA CRECIENTE (MCVAC)
>20ºC y <=40ºC	MAX	MIN	SI	NO	MEDIA CONVEXA DECRECIENTE (MCXAD)
>20ºC y <=40ºC	MIN	MAX	NO	SI	MEDIA CONVEXA CRECIENTE (MCXAC)
>20ºC y <=40ºC	MAX	MIN	NO	SI	MEDIA CÓNCAVA DECRECIENTE (MCVAD)
>20ºC y <=40ºC	MIN	MAX	SI	SI	MEDIA OSCILANTE MINIMO MÁXIMO (MOMINMAX)
>20ºC y <=40ºC	MAX	MIN	SI	SI	MEDIA OSCILANTE MÁXIMO MINIMO (MOMAXMIN)
>40ºC y <=200ºC	MIN	MAX	NO	NO	ALTA RECTA CRECIENTE (ARC)
>40ºC y <=200ºC	MAX	MIN	NO	NO	ALTA RECTA DECRECIENTE (ARD)
>40ºC y <=200ºC	MIN	MAX	SI	NO	ALTA CÓNCAVA CRECIENTE (ACVAC)
>40ºC y <=200ºC	MAX	MIN	SI	NO	ALTA CONVEXA DECRECIENTE (ACXAD)
>40ºC y <=200ºC	MIN	MAX	NO	SI	ALTA CONVEXA CRECIENTE (ACXAC)
>40ºC y <=200ºC	MAX	MIN	NO	SI	ALTA CÓNCAVA DECRECIENTE (ACVAD)
>40ºC y <=200ºC	MIN	MAX	SI	SI	ALTA OSCILANTE MINIMO MÁXIMO (AOMINMAX)
>40ºC y <=200ºC	MAX	MIN	SI	SI	ALTA OSCILANTE MÁXIMO MINIMO (AOMAXMIN)
>200ºC	-	-	-	-	ERROR (E)

Tabla 37. Valores que se asignarán a TIPOCURVATMPPx según la distancia de los máximos y mínimos a los puntos inicial o final y del valor de la diferencia entre el valor máximo y mínimo.

5.6.3.4 CARACTERIZACIÓN DE LAS CURVAS DE LAS TEMPERATURAS DE LA DIFERENCIA ENTRE EL VALOR DE CONSIGNA DEL PIRÓMETRO 2 Y EL REAL

Continuamos caracterizando las curvas de error.

```
############################################################
# Caracterización de la curva de ERROR=(TMPP2C-TMPP2M)   #
############################################################

ERRORTCAL <- round(MATDINAMIC$TMPP2C- MATDINAMIC$TMPP2M)

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- abs(ERRORTCAL)<200

MATSINRUIDO <- MATDINAMIC[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979

# Obtenemos el nuevo error sin los espurios
ERRORTCAL <- round(MATSINRUIDO$TMPP2C- MATSINRUIDO$TMPP2M)
# Obtenemos el valor de consigna máximo y mínimo de error por bobina
MINTMPP <- tapply(ERRORTCAL, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(ERRORTCAL, MATSINRUIDO$CODBOBINA,max)

# Obtenemos el valor medio de la consigna de error de cada bobina
VALMEAN <- tapply(ERRORTCAL, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos la consigna de error al principio y al final de cada bobina
MATTMPPINI <- ERRORTCAL[c(TRUE,(MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-
1),]$CODBOBINA!= MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA))]
MATTMPPFIN <- ERRORTCAL[MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-1),]$CODBOBINA!=
MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA]

# Buscamos la posición de los máximos y mínimos de cada bobina
BOBMASERROR <- (MATSINRUIDO$CODBOBINA*1000)+ ERRORTCAL
MAXEXP <- tapply(BOBMASERROR, MATSINRUIDO$CODBOBINA,max)
MINEXP <- tapply(BOBMASERROR, MATSINRUIDO$CODBOBINA,min)

#Detectamos la posición de los valores MAXEXP y MINEXP
POSMINEXP <- match (MINEXP,BOBMASERROR)
POSMAXEXP <- match (MAXEXP,BOBMASERROR)

# Determinamos si MAX está antes que MIN
MAXANTESMIN <- (POSMAXEXP <= POSMINEXP)

# Obtenemos las variables finales
ERRORMEDTOTAL <- round(VALMEAN)
ERRORDIFTOTAL <- MAXTMPP-MINTMPP
```

```
# Calculamos el ERROR MEDIO ABSOLUTO (área del error)
sumabs <- function(x) mean(abs(x))
ERRORMEDTOTALABS <- tapply(ERRORTCAL, MATSINRUIDO$CODBOBINA,sumabs)
ERRORMEDTOTALABS <- round(ERRORMEDTOTALABS)

#Calculamos si la diferencia entre los máximos, mínimos, con
#inicio o final es mayor de 1/4 de ERRORDIFTOTAL
MAXCONINI <- abs(MAXTMPP- MATTMPPINI)>(abs(ERRORDIFTOTAL/4))
MINCONINI <- abs(MINTMPP- MATTMPPINI)>(abs(ERRORDIFTOTAL/4))
MAXCONFIN <- abs(MAXTMPP- MATTMPPFIN)>(abs(ERRORDIFTOTAL/4))
MINCONFIN <- abs(MINTMPP- MATTMPPFIN)>(abs(ERRORDIFTOTAL/4))

# Obtenemos la variable categórica
TIPOCURVAERROR <- rep("H",length(ERRORDIFTOTAL))
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>10 & abs(ERRORDIFTOTAL)<=20 &
MAXANTESMIN==FALSE] <- "BRC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>10 & abs(ERRORDIFTOTAL)<=20 &
MAXANTESMIN==TRUE] <- "BRD"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- "MRC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- "MRD"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- "MCVAC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- "MCXAD"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- "MCXAC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- "MCVAD"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- "MOMINMAX"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>20 & abs(ERRORDIFTOTAL)<=40 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- "MOMAXMIN"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==FALSE] <- "ARC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==FALSE] <- "ARD"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==TRUE & MAXCONFIN==FALSE] <- "ACVAC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==TRUE & MINCONFIN==FALSE] <- "ACXAD"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI==FALSE & MAXCONFIN==TRUE] <- "ACXAC"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI==FALSE & MINCONFIN==TRUE] <- "ACVAD"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==FALSE & MINCONINI== TRUE & MAXCONFIN==TRUE] <- "AOMINMAX"

TIPOCURVAERROR[abs(ERRORDIFTOTAL)>40 & abs(ERRORDIFTOTAL)<=200 &
MAXANTESMIN==TRUE & MAXCONINI== TRUE & MINCONFIN==TRUE] <- "AOMAXMIN"
TIPOCURVAERROR[abs(ERRORDIFTOTAL)>200] <- "E"
```

```
##################################################################

table(TIPOCURVAERROR)

TIPOCURVAERROR
   ACVAC     ACVAD     ACXAC     ACXAD AOMAXMIN AOMINMAX       ARC       ARD
      15        30        33        16       14       11        31        31
     BRC       BRD         E         H    MCVAC    MCVAD    MCXAC    MCXAD
     174       171        32      1097       31       61        53        29
MOMAXMIN MOMINMAX       MRC       MRD
      18        22        53        57
```

Figura 218. Programa que obtiene las nuevas variables ERRORDIFTOTAL, ERRORMEDTOTAL, ERRORMEDTOTALABS, TIPOCURVAERROR que caracterizan la curva de la diferencia entre la temperatura de consigna del pirómetro dos y la real.

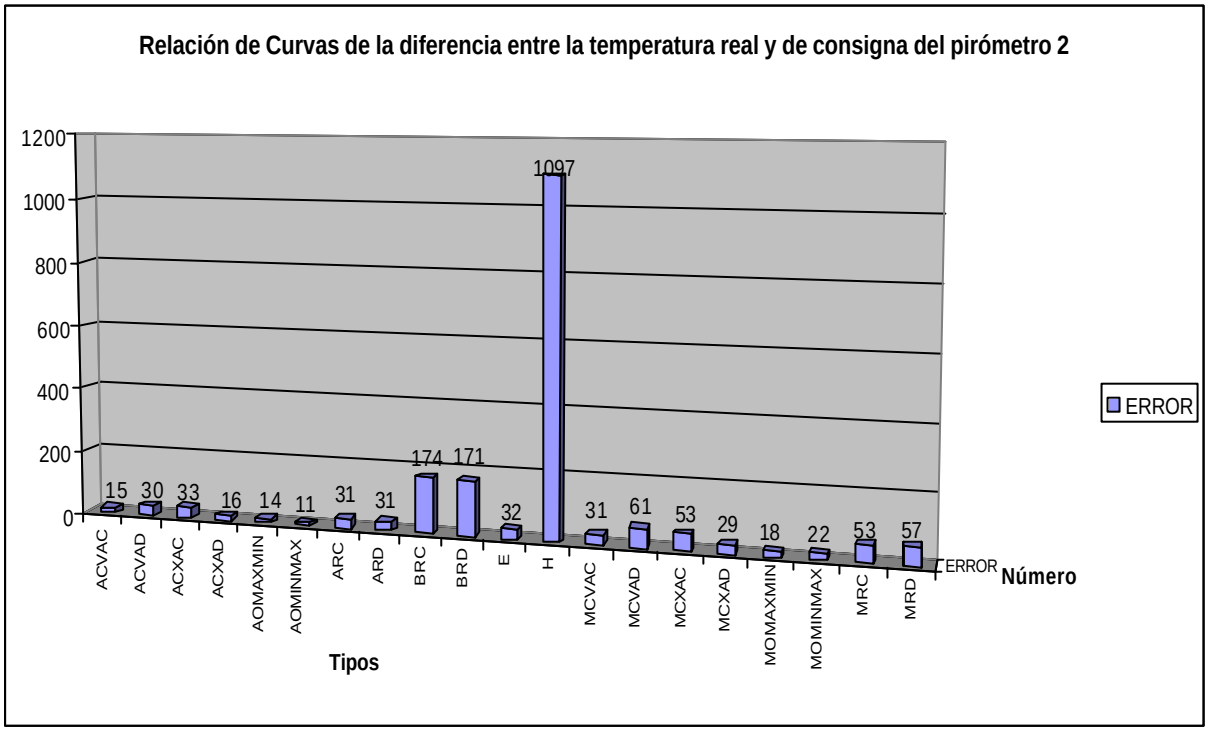


Figura 219. Distribución de las curvas de la diferencia entre la temperatura de consigna del pirómetro dos y la real.

	ACVAC	ACVAD	ACXAC	ACXAD	AOMAXMIN	AOMINMAX	ARC	ARD	BRC	BRD	E	H
TIPOCURVAERROR	15	30	33	16	14	11	31	31	174	171	32	1097

Tabla 38. Distribución de las curvas de la diferencia entre la temperatura de consigna del pirómetro dos y la real.

	MCVAC	MCVAD	MCXAC	MCXAD	MOMAXMIN	MOMINMAX	MRC	MRD
TIPOCURVAERROR	31	61	53	29	18	22	53	57

Tabla 39. Distribución de las curvas de la diferencia entre la temperatura de consigna del pirómetro dos y la real.

5.6.3.5 OBTENCIÓN DE LAS VARIABLES QUE DEFINEN LA VELOCIDAD DE LA BOBINA

Otra variable importante, es la velocidad de la banda en el horno. Las variables que definen la curva de velocidad de cada bobina se obtienen con el programa siguiente.

```
############################################################
# Caracterización de la curva de VELOCIDAD=(TMPP2C-TMPP2M)   #
############################################################

# Determinamos que bobinas tienen todos a –1 o a 0
# Obtenemos el máximo de velocidad de cada bobina
VELTCAL <- round(MATDINAMIC$VELOCIDADFIN)

MAXVEL <- tapply(VELTCAL,MATDINAMIC$CODBOBINA,max)
table(MAXVEL)
MAXVEL
 -1  31  40  49  50  53  55  57  60  62  65  67  69  70  72  73  74  75  76  77
  4   1   3   1   3   1   6   1  22   1  24  19   2  49   5  14  11  26   2  13
 78  79  80  81  82  83  84  85  86  87  88  89  90  92  93  94  95  97  98  99
 11   5  75   1   4  24   5  72   2   7   7   2  82   6   1   3  33   6  27   1
100 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
 93   4   5   1  35   1   1  21   6  91   1   7   3   1 128   1   2   7   3 198
121 122 123 125 126 127 128 129 130 131 132 133 134 135 137 138 139 140 142 143
  1   7   5 180   3  13   7   1 269   1  20  15   4 194   2   3   7  36   2   2
145 146 147 149 150
 12   1   1   1   1


# Vemos que bobinas tienen todas las velocidades a cero o a -1
bobmalas <- c(row.names(as.matrix(MAXVEL[MAXVEL %in% -
1])),row.names(as.matrix(MAXVEL[MAXVEL %in% -0])))

# Ponemos a 999 las velocidades de esas bobinas
MATDINAMIC[MATDINAMIC$CODBOBINA %in% bobmalas,]$VELOCIDADFIN <- 999

# Agarramos las nuevas velocidades
VELTCAL <- round(MATDINAMIC$VELOCIDADFIN)

# Obtenemos las observaciones de velocidad que son mayores de 10
LISTASIN <- VELTCAL>10

# Eliminamos los valores debidos a fallos de adquisición
MATSINRUIDO <- MATDINAMIC[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979
```

```
# Obtenemos el nuevo error sin los espurios
VELTCAL <- round(MATSINRUIDO$VELOCIDADFIN)
# Obtenemos el valor de consigna máximo y mínimo de error por bobina
MINTMPP <- tapply(VELTCAL, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(VELTCAL, MATSINRUIDO$CODBOBINA,max)

# Obtenemos el valor medio de la consigna de error de cada bobina
VALMEAN <- tapply(VELTCAL, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos la velocidad al principio y al final de cada bobina
MATTMPPINI <- VELTCAL[c(TRUE,(MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-
1),]$CODBOBINA!= MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA))]
MATTMPPFIN <- VELTCAL[MATSINRUIDO[1: (dim(MATSINRUIDO)[1]-1),]$CODBOBINA!=
MATSINRUIDO[2: (dim(MATSINRUIDO)[1]),]$CODBOBINA]

# Buscamos la posición de los máximos y mínimos de cada bobina
BOBMASERROR <- (MATSINRUIDO$CODBOBINA*1000)+ VELTCAL
MAXEXP <- tapply(BOBMASERROR, MATSINRUIDO$CODBOBINA,max)
MINEXP <- tapply(BOBMASERROR, MATSINRUIDO$CODBOBINA,min)

#Detectamos la posición de los valores MAXEXP y MINEXP
POSMINEXP <- match (MINEXP,BOBMASERROR)
POSMAXEXP <- match (MAXEXP,BOBMASERROR)

# Determinamos si MAX está antes que MIN
MAXANTESMIN <- (POSMAXEXP <= POSMINEXP)

# Obtenemos las variables finales
VELMEDTOTAL <- round(VALMEAN)
VELDIFTOTAL <- MAXTMPP-MINTMPP

#Calculamos si la diferencia entre los máximos, mínimos, con
#inicio o final es mayor de 1/4 de VELDIFTOTAL
MAXCONINI <- abs(MAXTMPP- MATTMPPINI)>(abs(VELDIFTOTAL/4))
MINCONINI <- abs(MINTMPP- MATTMPPINI)>(abs(VELDIFTOTAL/4))
MAXCONFIN <- abs(MAXTMPP- MATTMPPFIN)>(abs(VELDIFTOTAL/4))
MINCONFIN <- abs(MINTMPP- MATTMPPFIN)>(abs(VELDIFTOTAL/4))


# Obtenemos la variable categórica
TIPOCURVAVEL <- rep("H",length(VELDIFTOTAL))
TIPOCURVAVEL[abs(VELDIFTOTAL)>10 & abs(VELDIFTOTAL)<=20 & MAXANTESMIN==FALSE] <-
"BRC"
TIPOCURVAVEL[abs(VELDIFTOTAL)>10 & abs(VELDIFTOTAL)<=20 & MAXANTESMIN==TRUE] <-
"BRD"


TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==FALSE &
MINCONINI==FALSE & MAXCONFIN==FALSE] <- "MRC"
TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==TRUE &
MAXCONINI==FALSE & MINCONFIN==FALSE] <- "MRD"

TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==FALSE &
MINCONINI==TRUE & MAXCONFIN==FALSE] <- "MCVAC"
TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==TRUE &
MAXCONINI==TRUE & MINCONFIN==FALSE] <- "MCXAD"

TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==FALSE &
MINCONINI==FALSE & MAXCONFIN==TRUE] <- "MCXAC"
```

```
TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==TRUE &
MAXCONINI==FALSE & MINCONFIN==TRUE] <- "MCVAD"

TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==FALSE &
MINCONINI== TRUE & MAXCONFIN==TRUE] <- "MOMINMAX"
TIPOCURVAVEL[abs(VELDIFTOTAL)>20 & abs(VELDIFTOTAL)<=50 & MAXANTESMIN==TRUE &
MAXCONINI== TRUE & MINCONFIN==TRUE] <- "MOMAXMIN"


TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==FALSE &
MINCONINI==FALSE & MAXCONFIN==FALSE] <- "ARC"
TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==TRUE &
MAXCONINI==FALSE & MINCONFIN==FALSE] <- "ARD"

TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==FALSE &
MINCONINI==TRUE & MAXCONFIN==FALSE] <- "ACVAC"
TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==TRUE &
MAXCONINI==TRUE & MINCONFIN==FALSE] <- "ACXAD"

TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==FALSE &
MINCONINI==FALSE & MAXCONFIN==TRUE] <- "ACXAC"
TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==TRUE &
MAXCONINI==FALSE & MINCONFIN==TRUE] <- "ACVAD"

TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==FALSE &
MINCONINI== TRUE & MAXCONFIN==TRUE] <- "AOMINMAX"
TIPOCURVAVEL[abs(VELDIFTOTAL)>50 & abs(VELDIFTOTAL)<=200 & MAXANTESMIN==TRUE &
MAXCONINI== TRUE & MINCONFIN==TRUE] <- "AOMAXMIN"

TIPOCURVAVEL[abs(VELDIFTOTAL)>200] <- "E"

#################################################################

table(TIPOCURVAVEL)
TIPOCURVAVEL
   ACVAC    ACVAD    ACXAD AOMINMAX      ARC      ARD      BRC      BRD
       3       59        1        1       10       10       89       82
       H    MCVAC    MCVAD    MCXAC    MCXAD MOMAXMIN      MRC      MRD
    1500       21       99        7        7        5       48       37
```

Figura 220. Programa que obtiene las nuevas variables VELDIFTOTAL,VELMEDTOTAL,TIPOCURVAVEL que caracterizan la curva de la velocidad de la banda en el horno.

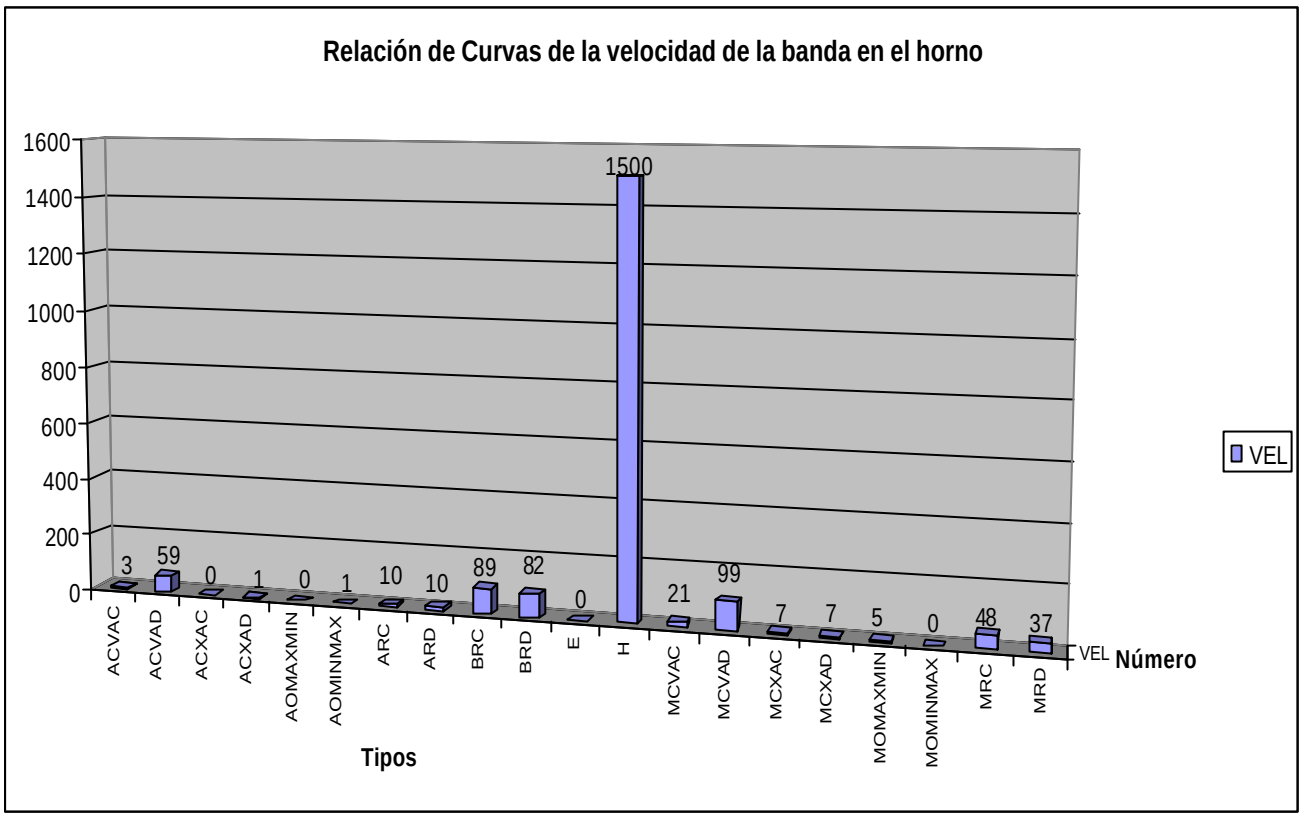


Figura 221. Distribución de las curvas de la velocidad de la bobina en el horno.

	ACVAC	ACVAD	ACXAC	ACXAD	AOMAXMIN	AOMINMAX	ARC	ARD	BRC	BRD	E	H
TIPOCURVAVEL	3	59	0	1	0	1	10	10	89	82	0	1500

Tabla 40. Distribución de las curvas de la velocidad de la bobina en el horno.

	MCVAC	MCVAD	MCXAC	MCXAD	MOMAXMIN	MOMINMAX	MRC	MRD
TIPOCURVAVEL	21	99	7	7	5	0	48	37

Tabla 41. Distribución de las curvas de la velocidad de la bobina en el horno.

5.6.4 CREACIÓN DE LA MATRIZ CON TODOS LAS NUEVAS VARIABLES

Una vez creadas, se incluyen en una nueva matriz llamada *MATBOBINAS*:

Tabla de la B.D. a la que pertenece	Nombre de la Variable	Formato	Descripción
T100calentamiento	CODBOBINA	Código Numérico	Código de la Bobina
Nueva Variable	MAXINSTANTE	Entero	Instante máximo de la bobina
T30Tijera	ESPFINAL	Real	Espesor medio final de la banda con el recubrimiento
Nueva Variable	MODOBOB	Categórica	Modo de operación en el que se ha trabajado la bobina con un tiempo mayor del 50%. (0=”Manual”, -1=”Automático”)
Nueva Variable	SECCION	Real	Espesor * Anchura (en mm^2.)
Nueva Variable	THF1DIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura de consigna para cada bobina en la zona 1.
Nueva Variable	THF1MEDTOTAL	Entero	Temperatura de consigna media para cada bobina en la zona 1.
Nueva Variable	TIPOCURVATHF1	Categórica	Tipo de Curva de la temperatura de consigna en la zona 1.
Nueva Variable	TMPPxMEDTOTAL	Entero	Temperatura media leída por el pirómetro x (x vale 1 o 2)
Nueva Variable	TMPPxDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura del pirómetro x para cada bobina (x vale 1, 2)
Nueva Variable	TIPOCURVATMPPx	Categórica	Tipo de Curva de la temperatura del pirómetro x (x vale 1, 2).
Nueva Variable	TMPP2CNGMEDTOTAL	Entero	Temperatura de consigna del pirómetro 2.
Nueva Variable	TMPP2CNGDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura de consigna del pirómetro 2
Nueva Variable	TIPOCURVATMPP2CNG	Categórica	Tipo de Curva de la temperatura de consigna del pirómetro 2.
Nueva Variable	VELMEDTOTAL	Entero	Velocidad de la banda.
Nueva Variable	VELDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la velocidad.
Nueva Variable	TIPOCURVAVEL	Categórica	Tipo de Curva de la velocidad de la banda.
Nueva Variable	ERRORMEDTOTAL	Entero	Valor del error medio obtenido de la diferencia del valor medido por el pirómetro 2 y el de consigna.
Nueva Variable	ERRORMEDTOTALABS	Entero	Valor del error medio absoluto del valor absoluto de la diferencia del valor medido por el pirómetro 2 y el de consigna.
Nueva Variable	ERRORDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo del error del pirómetro para cada bobina.
Nueva Variable	TIPOCURVAERROR	Categórica	Tipo de Curva de la temperatura del error.

Tabla 42. Variables de la matriz MATBOBINAS.

```
# Creamos algunas nuevas variables
# Determinamos que bobinas han sido trabajadas en “modo manual” y “modo autom”
MODOBOBMEAN <- tapply(T100CALB$MODHF, T100CALB$CODBOBINA, mean)

# Obtenemos el instante máximo de cada bobina
MAXINSTANTE <- tapply(MATDINAMIC$INSTANTE, MATDINAMIC$CODBOBINA,max)

# Creamos la nueva matriz MATBOBINAS

MATBOBINAS <- data.frame(cbind(DATBOBINAS$CODBOBINA, MAXINSTANTE,ESPFINAL,
MODOBOB, MODOBOBMEAN, THF1MEDTOTAL, THF1DIFTOTAL, TMPP1MEDTOTAL, TMPP1DIFTOTAL,
TMPP2MEDTOTAL, TMPP2DIFTOTAL, TMPP2CNGMEDTOTAL, TMPP2CNGDIFTOTAL, ERRORMEDTOTAL,
ERRORMEDTOTALABS, ERRORDIFTOTAL, VELMEDTOTAL, VELDIFTOTAL, TIPOCURVATHF1,
TIPOCURVATMPP1, TIPOCURVATMPP2, TIPOCURVATMPP2CNG ,TIPOCURVAERROR,
TIPOCURVAVEL))

# Visualizamos las primeras 4 bobinas
MATBOBINAS[1:4,]
```

```
              V1 MAXINSTANTE          ESPFINAL MODOBOB MODOBOBMEAN
23293006 23293006          39 0.586133301258087       0           0
23293007 23293007          33 0.589666664600372       0           0
23293008 23293008          38 0.590000033378601       0           0
23293009 23293009          23 0.587403804063797       0           0
         THF1MEDTOTAL THF1DIFTOTAL TMPP1MEDTOTAL TMPP1DIFTOTAL TMPP2MEDTOTAL
23293006          770            3           212             4           754
23293007          772            9           210             8           749
23293008          778           32           211             8           761
23293009          758           73           221             7           755
         TMPP2DIFTOTAL TMPP2CNGMEDTOTAL TMPP2CNGDIFTOTAL ERRORMEDTOTAL
23293006            17              750                0            -4
23293007             4              750                0             1
23293008            22              770                0             9
23293009            20              750                0            -4
         ERRORMEDTOTALABS ERRORDIFTOTAL VELMEDTOTAL VELDIFTOTAL TIPOCURVATHF1
23293006                4            17         145           1             H
23293007                2             4         145           0             H
23293008                9            22         138          61            MC
23293009                7            20         145           0            AD
         TIPOCURVATMPP1 TIPOCURVATMPP2 TIPOCURVATMPP2CNG TIPOCURVAERROR
23293006              H            BRD                H            BRC
23293007              H              H                H              H
23293008              H          MCXAC                H          MCVAD
23293009              H            BRD                H            BRC
         TIPOCURVAVEL
23293006            H
23293007            H
23293008        ACVAD
23293009            H
```

```
#Obtenemos los nombres finales de las variables de las matrices a usar
names(MATDINAMIC)
```

```
[1] "CODBOBINA"    "INSTANTE"     "THC1"         "TMPP1M"       "TMPP2M"
[6] "TMPP2C"       "VELOCIDADFIN" "COLORBOB"     "MODHF"
```

```
names(DATBOBINAS)
```

```
 [1] "CODBOBINA" "BOBENT"    "ESPENT"    "CLASACERO" "DUREZA"    "CICREC"
 [7] "ANCHO"     "ESPESOR"   "LARGO"     "PESO"      "CALIDAD"   "FECFAB"
[13] "HORFAB"
```

```
names(MATBOBINAS)
 [1] "V1"                 "MAXINSTANTE"        "ESPFINAL"
 [4] "MODOBOB"            "MODOBOBMEAN"        "THF1MEDTOTAL"
 [7] "THF1DIFTOTAL"       "TMPP1MEDTOTAL"      "TMPP1DIFTOTAL"
[10] "TMPP2MEDTOTAL"      "TMPP2DIFTOTAL"      "TMPP2CNGMEDTOTAL"
[13] "TMPP2CNGDIFTOTAL"   "ERRORMEDTOTAL"      "ERRORMEDTOTALABS"
[16] "ERRORDIFTOTAL"      "VELMEDTOTAL"        "VELDIFTOTAL"
[19] "TIPOCURVATHF1"      "TIPOCURVATMPP1"     "TIPOCURVATMPP2"
[22] "TIPOCURVATMPP2CNG" "TIPOCURVAERROR"     "TIPOCURVAVEL"

save(MATDINAMIC,DATBOBINAS,MATBOBINAS,file="C:\\JAVI_CASA\\PISON_DESPACHO\\DOCTO
RADO\\TESIS\\Tesis_25_01_03\\apoyo\\2003\\DATOS2003_FINAL2.RData")
```

Figura 222. Creación final de la matriz MATBOBINAS.

5.7 CONCLUSIONES

En este capítulo se han descrito todos los pasos que han sido necesarios para preparar una nueva base de datos más fiable que sirva para la consecución con éxito de las fases posteriores de estudio. Este trabajo ha sido arduo ya que se han tenido que analizar gran cantidad de variables, filtrarlas y adaptarlas.

Del análisis inicial de los datos, se obtuvieron las siguientes conclusiones:

- Las observaciones erróneas eran debidas a efectos esporádicos y no se producían de forma continuada.
- Se observó un número inicial de 306 observaciones erróneas de un total de 30.753, es decir, un 0,995% de observaciones erróneas.
- De 1.712 bobinas estudiadas se encontraron errores en 45 de ellas, es decir, **un 2,62% donde han aparecido problemas en las temperaturas de consigna o reales.**
- Se consideró que se debía analizar y mejorar el sistema de monitorización encargado del almacenamiento de las variables en la base de datos y de la gestión de las variables de consigna. Era necesario obtener una nueva base de datos.
- Faltaba una variable que indicara si el sistema estaba trabajando en "modo manual" o en "modo automático".

Después se trabajó separadamente con grupos de variables. Las conclusiones y acciones más importantes para cada uno de estos grupos fueron las siguientes:

5.7.1 TEMPERATURAS DE CONSIGNA DE ZONAS DEL HORNO

- **Las variables más fiables para cada instante de la bobina eran las que corresponden con el valor medio** (THFxVALMED). Se desestimó el uso de las variables MAX y MIN por dar valores erróneos.
- **Resultó conveniente eliminar aquellas bobinas que tenían valores de observación erróneos**. Se eliminaron las bobinas con temperaturas con valores menores de 100.
- Los dispositivos que controlan la temperatura de cada zona en la parte de calentamiento del horno siguen con bastante fidelidad la temperatura de consigna. Por lo tanto **se seleccionaron para el estudio únicamente las temperaturas de consigna** (*THFxVALCNG*).
- Para cada bobina, se caracterizó el comportamiento de las temperaturas de consigna de las zonas (creación de las variables *THFxMEDTOTAL, THFxDIFTOTAL, TIPOCURVATHFx).*

- **Sólo son necesarias para el estudio, las variables de consigna correspondientes a las zonas 1, 3 y 5,** ya que las temperaturas de las zonas 2, 4, 6 y 8 son iguales que las de las 1, 3, 5 y 7; y además, la temperatura de consigna de la zona 5 no difiere mucho de la 7.
- **Se creó una variable que indica si el funcionamiento del horno está en "Modo Manual" o "Modo Automático" ya que muchos comportamientos anómalos pueden no ser achacados al sistema de control.**

5.7.2 TEMPERATURAS DE PIRÓMETROS Y ANÁLISIS DEL ERROR

De los datos obtenidos medios y de consigna de los tres pirómetros se establecieron las siguientes conclusiones y acciones a realizar:

- **El valor de las temperaturas de consigna *TMPP1CNG* y *TMPP3CNG* es cero**, ya que el sistema no tienen valores de consigna para estos pirómetros. El valor de consigna apropiado, es decir, la temperatura buscada que debe tener la banda a la salida de la zona de calentamiento, debe ser *TMPP2CNG* y ésta es la misma que la del pirómetro 3. Como conclusión, solo se utilizó la temperatura de consiga del pirómetro 2.
- **Existían valores erróneos al final de algunas bobina**. Parece que algunas veces, en el instante final de la bobina, no se almacena el valor de las temperaturas medias y de consigna de los pirómetros dos y tres. Por lo tanto, se llegó a la conclusión de que es un error que se genera **durante todo el proceso y en el último instante de algunas bobinas**. Se concluyó, que este error, con bastante seguridad, **se produce por un defecto en el almacenamiento de los datos en el último momento o en el momento de cambio de alguna bobina y por lo tanto, no se consideró problemático ya que se produce en un solo instante y de forma esporádica.** Se decidió rellenarlos con el valor aparecido en el instante anterior, ya que los mismos, por la inercia del sistema y del modelo, no varían significativamente con respecto a los del instante anterior.
- La distribución normal del error indica que **el modelo actual explica más o menos bien el sistema físico,** ya que en los residuos no se advierte ningún tipo de estructura no lineal.
- Aún así, existen numerosas bobinas en las que el error mínimo oscila entre los 10 y 40 grados y el máximo entre los 40 y 120 grados centígrados de diferencia entre las temperaturas de consigna y las reales. **Muchos de estos, se producen en transiciones bruscas de temperaturas de consigna, aunque existen otros, que no pueden ser explicados solamente con el análisis de esas dos variables.**
- También se caracterizaron las curvas de temperatura de pirómetros y del error.

5.7.3 VELOCIDAD DE LA BANDA Y DIMENSIONES DE LA MISMA

Analizando la velocidad y las dimensiones de la banda se obtuvieron las siguientes conclusiones:

- La velocidad de la banda era la misma para las diferentes zonas del horno, por lo tanto, **solo se utilizó *VELCENMED***.
- El sensor que determina la velocidad de la banda o los sistemas de captura que almacenan el valor, **genera bastante ruido (34.155 valores a –1 de 353.887, es decir un 9,7% de los datos de velocidad)**. Estos espurios tuvieron que ser eliminados.
- Existe una gran mayoría de curvas de velocidades con un comportamiento horizontal, aunque entre transiciones aparecen diferentes tipos de curvas. Por lo tanto, se estimó conveniente caracterizar las curvas de velocidades.
- **Se utilizaron también las dimensiones de la banda**, ya que la temperatura de ésta dependerá, como es lógico, de sus características físicas además de otras variables.

5.7.4 USO DE UNA NUEVA BASE DE DATOS

Se obtuvo una nueva base de datos nueva con valores más fiables y algunas nuevas variables con las siguientes características:

- Datos Relativos a 35 días de proceso.
- 48 clases de acero con 32 tipos de durezas.
- Variables *CICREC* (ciclo de recocido) y *CALIDAD*, prácticamente sin datos.
- Bobinas entre 350 y 5.538 metros de longitud.
- Rango de espesores de 0,417 mm. a 2,016 mm. La mayoría de los espesores de las bobinas se centran en el rango 0,417 mm. a 1 mm.
- 40.496 observaciones (56,3%) son en datos del "modo manual" (*MODHF*=0) y 18.765 (26,1%) corresponden al modo automático" (*MODHF*=-1) (569 bobinas tratadas en "modo automático" frente a 1.407 bobinas tratadas en "modo manual"). Existe una relación 1/3 de bobinas tratadas en "modo automático" frente al "modo manual", es decir, el tratamiento en "modo manual" es tres veces superior al "modo automático" de uso del modelo, lo que implica que el uso del modelo matemático no es muy intensivo.

- Las clases de acero más utilizadas han sido las de la tabla siguiente (de las cuales 1.179, 59,6%, son de dureza 50):

Clase Acero	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
Número de Bobinas	51	129	45	650	152	82	295	52	113	63
Dureza del Acero	17	19	15	50	50	50	50	E8	E8	14

Tabla 43. Número de bobinas más tratadas.

- Existen bobinas con un alto porcentaje del uso en "Modo Manual" frente al "Modo Automático" (ver tabla), lo que indica claramente que el modelo no es usado en varios tipos de bobinas.

Dureza	Bobinas en Modo Automático	Bobinas en Modo Manual	Porcentaje de Bobinas en Modo Manual
11	5	5	50,00%
13	0	12	100,00%
14	40	32	44,44%
15	23	20	46,51%
16	5	29	85,29%
17	3	44	93,62%
19	60	75	55,56%
20	0	2	100,00%
24	2	1	33,33%
29	0	4	100,00%
30	0	7	100,00%
32	1	50	98,04%
37	1	1	50,00%
50	392	877	69,11%
E1	0	35	100,00%
E8	29	136	82,42%
F8	2	25	92,59%
G0	2	17	89,47%
G4	4	10	71,43%

Figura 223. Porcentaje de bobinas tratadas en "modo manual" frente al "modo automático" según la dureza del acero.

- La compresión de las tablas con datos medidos cada 30 metros, fue bastante problemática, debido a que existían bobinas con menor número de datos que en las tablas de medidas de 100 metros. De esta forma, fue necesario eliminar esas bobinas (111 de 2.090).

- Finalmente quedaron un total de 1.979 bobinas a estudiar.

5.7.5 CREACIÓN DE LAS TABLAS FINALES

De todas las conclusiones anteriores y partiendo de la base de datos final, se crearon las siguientes tablas:

- DATBOBINAS: Datos de cada bobina.
- MATDINAMIC: Datos dinámicos ya preparados del proceso (medias de valores cada 100 metros de banda)
- MATBOBINAS: Datos con nuevas variables que caracterizan los tipos de curvas.

Tabla de la B.D. de donde procede	Nombre de la Variable	Formato	Descripción
Dps	CODBOBINA	Código Numérico	Código de la Bobina
Dps	BOBENT	Categórica	Código de fabricación Bobina
Dps	ESPENT	Real	Espesor Real de la Bobina a la entrada del horno (en mm.)
Dps	CLASACERO	Categórica	Tipo de Acero de la Bobina
Dps	DUREZA	Categórica	Dureza del Acero
Dps	CICREC	Categórica	Ciclo de Recocido
Dps	ANCHO	Entero	Anchura en mm. de la banda.
Dps	ESPESOR	Real	Espesor objetivo de fabricación en mm.
dps	LARGO	Entero	Longitud de la Bobina (en metros)
dps	PESO	Real	Peso en kg.
dps	CALIDAD	Categórica	Calidad de la bobina
dps	FEC_FAB	DD-MM-AAAA	Fecha De Fabricación
dps	HOR_FAB	HH:MM	Hora de Fabricación

Tabla 44. Variables de la matriz DATBOBINAS.

Tabla de la B.D. de donde procede	Nombre de la Variable	Formato	Descripción
T100cal	CODBOBINA	Código Numérico	Código de la Bobina
T100cal	INSTANTE	Entero	Instante en que se hace cada medida (cada 100 metros)
T100cal (de THF1VALCNG)	THC1	Entero	Temp. de consigna de la subzona 1 (cada 100 metros)
T100cal (TMPP1VALMED)	TMPP1M	Entero	Valor de temperatura del pirómetro 1 (cada 100 metros)
T100cal (TMPP2VALMED)	TMPP2M	Entero	Valor de temperatura del pirómetro 2 (cada 100 metros)
T100cal (TMPP2VALCNG)	TMPP2C	Entero	Valor de consigna de temperatura del pirómetro 2. (cada 100 metros)
T30Acumuladore (de VELCENMED)	VELOCIDADFIN	Entero	Velocidad de la banda medida en el centro del horno. (en m./min). (ajustada a medidas cada 100 metros)
Nueva Variable	COLORBOB	Entero	Color asignado a la bobina. Número de 1 a 7 obtenido del código de bobina (CODBOBINA%7+1)
T100cal	MODHF	Entero	Valor de Temp. del pirómetro 3 (cada 100 metros)

Tabla 45. Variables de la matriz MATDINAMIC.

Tabla de la B.D. a la que pertenece	Nombre de la Variable	Formato	Descripción
T100calentamiento	CODBOBINA	Código Numérico	Código de la Bobina
Nueva Variable	MAXINSTANTE	Entero	Instante máximo de la bobina
T30Tijera	ESPFINAL	Real	Espesor medio final de la banda con el recubrimiento
Nueva Variable	MODOBOB	Categórica	Modo de operación en el que se ha trabajado la bobina con un tiempo mayor del 50%. (0="Manual", -1="Automático")
Nueva Variable	SECCION	Real	Espesor * Anchura (en mm^2.)
Nueva Variable	THF1DIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura de consigna para cada bobina en la zona 1.
Nueva Variable	THF1MEDTOTAL	Entero	Temperatura de consigna media para cada bobina en la zona 1.
Nueva Variable	TIPOCURVATHF1	Categórica	Tipo de Curva de la temperatura de consigna en la zona 1.
Nueva Variable	TMPPxMEDTOTAL	Entero	Temperatura media leída por el pirómetro x (x vale 1 o 2)
Nueva Variable	TMPPxDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura del pirómetro x para cada bobina (x vale 1, 2)
Nueva Variable	TIPOCURVATMPPx	Categórica	Tipo de Curva de la temperatura del pirómetro x (x vale 1, 2).
Nueva Variable	TMPP2CNGMEDTOTAL	Entero	Temperatura de consigna del pirómetro 2.
Nueva Variable	TMPP2CNGDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la temperatura de consigna del pirómetro 2
Nueva Variable	TIPOCURVATMPP2CNG	Categórica	Tipo de Curva de la temperatura de consigna del pirómetro 2.
Nueva Variable	VELMEDTOTAL	Entero	Velocidad de la banda.
Nueva Variable	VELDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo de la velocidad.
Nueva Variable	TIPOCURVAVEL	Categórica	Tipo de Curva de la velocidad de la banda.
Nueva Variable	ERRORMEDTOTAL	Entero	Valor del error medio obtenido de la diferencia del valor medido por el pirómetro 2 y el de consigna.
Nueva Variable	ERRORMEDTOTALABS	Entero	Valor del error medio absoluto del valor absoluto de la diferencia del valor medido por el pirómetro 2 y el de consigna.
Nueva Variable	ERRORDIFTOTAL	Entero	Diferencia entre el valor máximo y el mínimo del error del pirómetro para cada bobina.
Nueva Variable	TIPOCURVAERROR	Categórica	Tipo de Curva de la temperatura del error.

Tabla 46. Variables de la matriz MATBOBINAS.

En el capítulo siguiente, se procede a analizar la información obtenida buscando conocimiento útil.

CAPÍTULO 6

ANÁLISIS DE LOS DATOS: ESTUDIO DE LA INFORMACIÓN MEDIANTE TÉCNICAS DE MINERÍA DE DATOS

6.1 INTRODUCCIÓN

Todos los pasos realizados en la preparación de la información, nos ha permitido obtener una base de datos consistente con la que poder trabajar con garantías.

De la simple observación de los datos ya hemos obtenido algunas conclusiones, más o menos importantes, aunque no es sino en este momento cuando podemos intentar extraer información más útil que nos ayude a comprender mejor la interrelación entre las variables a analizar y conclusiones que nos ayuden a entender el proceso.

En este capítulo, se aplican diferentes técnicas de minería de datos para intentar obtener información práctica, oculta en la base de datos, que nos permita extraer conclusiones importantes y nos ayuden a resolver algunas de estas dudas:

- ¿Qué hace que la temperatura real de la banda y la objetivo sean muy diferentes? ¿Cuáles son las causas principales?
- ¿Cuál es el modo más fiable (“modo manual” o “modo automático”).
- ¿En qué bobinas se trabaja mejor y cuáles peor?
- ¿Cómo se pueden clasificar los tipos de bobinas?
- ¿Por qué se producen esos errores?
- ¿Qué variables influyen más y cuáles menos dentro del proceso de galvanizado?
- ¿Qué grado de dependencia tienen las variables entre si? ¿Cuáles se pueden eliminar?

- ¿Qué variables son las más interesantes para el modelado?
- ¿Qué grado de eficiencia tiene el modelo actual para cada tipo de bobina?
- ¿Cuándo el sistema de control trabaja mejor o peor?
- ¿Cómo se puede explicar el comportamiento en las transiciones entre bobinas?
- ¿Podemos obtener algunas reglas que nos permitan minimizar con cierto grado de precisión el error?
- ¿Qué tipo de estructura tiene la información?

Y otras muchas preguntas que, si son resueltas, nos pueden ayudar a comprender mejor el proceso estudiado.

En los apartados siguientes, se muestra el empleo de una batería de herramientas de Estadística y de Minería de Datos que nos van a ayudar a resolver las preguntas antes planteadas.

6.2 ANÁLISIS DE DEPENDENCIAS ENTRE VARIABLES

Lo primero que se estudia es la dependencia entre variables más significativas.

En este punto se intentará, mediante diversas técnicas de minería de datos, analizar y buscar una relación entre la variable[18] "error" y el comportamiento de las diversas variables de proceso para cada bobina sin considerar en su plenitud, la relación de las variables de proceso de una bobina con las anteriores o posteriores. Es en el punto siguiente, donde se tratarán con más detalle estas relaciones.

6.2.1 ESTUDIO DE LA RELACIÓN ENTRE EL ERROR DE TEMPERATURA, LA VELOCIDAD Y LAS DIMENSIONES DE LA BANDA

Las variables que van a entrar en el estudio son:

- *VELDIFTOTAL*: Diferencia entre velocidad mínima y máxima de banda para cada bobina.
- *VELMEDTOTAL*: Velocidad media de la banda en esa bobina.
- *LARGO*: Longitud en metros de esa bobina.
- *SECCION*: Sección (en mm^2) de la banda para esa bobina. Resultado de multiplicar el ancho de la banda por el espesor.
- *Abs(ERRORMEDTOTAL)*: Valor absoluto de la diferencia entre la temperatura de consigna del pirómetro dos (temperatura esperada de la bobina a la salida de la zona de calentamiento) y la real.

Lo primero que realizamos, es un estudio de los gráficos box-plots de las variables que se van a estudiar (ver Figura 224).

[18] Como se ha definido en el apartado 5.5.4.2 y en otras partes de este trabajo, el "error" corresponde con el valor medio absoluto, para cada bobina, de la diferencia de la temperatura real de la banda, medida por el pirómetro dos a la salida de la zona de calentamiento del horno, frente a la temperatura objetivo que debe alcanzar dicha banda.

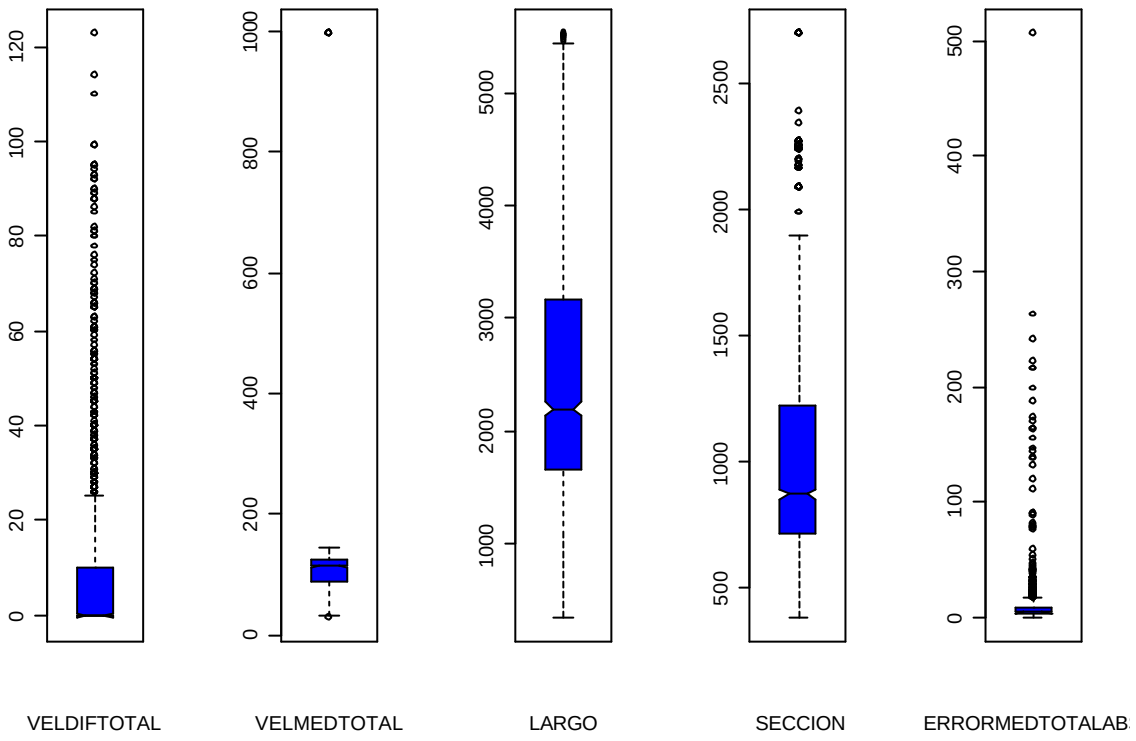


Figura 224. Gráficos box-plots de las variables estudiadas.

Vemos claramente que las dimensiones de las bobinas en las que se mueve el 50% de ellas oscilan entre 1.800 m. y 3.200 metros de longitud (variable *LARGO)*, y 750 mm^2. y 1.250 mm^2. cuadrados de sección (variable *SECCIÓN*), un rango bastante considerable.

Por otro lado, se puede advertir cómo la velocidad media total varía considerablemente entre un rango de 30 a 130 metros por minuto.

```
#dibujamos los gráficos de box-plots de las variables
SECCION<- as.numeric(as.matrix(DATBOBINAS$ESPENT))*
as.numeric(as.matrix(DATBOBINAS$ANCHO))
par(mfrow=c(1,5),cex=1)
boxplot(as.numeric(as.matrix(MATBOBINAS$VELDIFTOTAL)),col="blue",notch=TRUE,xlab
="VELDIFTOTAL")
boxplot(as.numeric(as.matrix(MATBOBINAS$VELMEDTOTAL)),col="blue",notch=TRUE,
xlab="VELMEDTOTAL")
boxplot(as.numeric(as.matrix(DATBOBINAS$LARGO)),col="blue",notch=TRUE,
xlab="LARGO")
boxplot(SECCION,col="blue",notch=TRUE, xlab="SECCION")
boxplot(as.numeric(as.matrix(MATBOBINAS$ERRORMEDTOTALABS)),col="blue",notch=TRUE
, xlab="ERRORMEDTOTALABS")

# Obtenemos la lista de errores medios absolutos
J<-as.numeric(as.matrix(MATBOBINAS$ERRORMEDTOTALABS))
```

```
# Medimos el % de errores > 30ºC
table(J>30)
FALSE  TRUE
 1917    62
> 62/(1917+62)
[1] 0.03132895

# Medimos el % de errores > 40ºC
table(J>40)
FALSE  TRUE
 1940    39
39/(1940+39)
[1] 0.01970692

# Medimos el % de errores > 50ºC
table(J>50)
FALSE  TRUE
 1947    32
32/(1947+32)
[1] 0.01616978
```

Figura 225. Programa que obtiene los gráficos box-plots de las variables estudiadas.

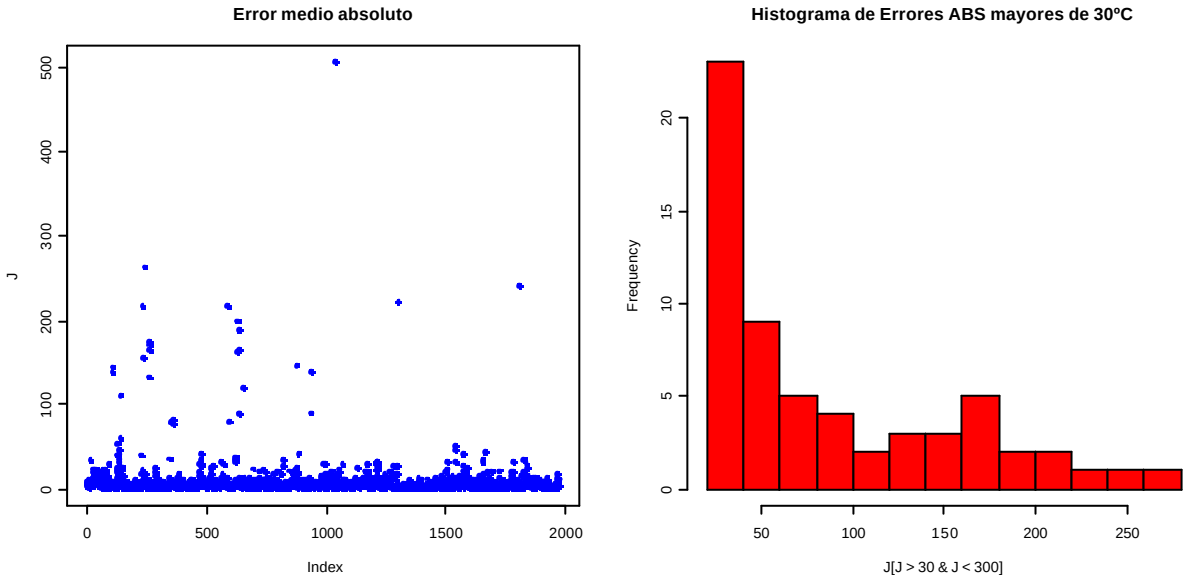


Figura 226. Error medio absoluto para todas las bobinas.

Las figuras anteriores que representan el "error medio absoluto" de cada bobina (variable *ERRORMEDTOTALABS*)[19], muestra claramente como el 1,6% de las bobinas tienen una media del error absoluto mayor de 50ºC, y si consideramos como umbral 30ºC, obtenemos un 3,1% de bobinas a analizar.

[19] Este error es bastante significativo ya que muestra la media del error (temperatura esperada de la banda y temperatura real).

6.2.1.1 ANÁLISIS DEL SCATTER-PLOTS

Podemos obtener más información estudiando el gráfico de *scatter-plots*.

```
Dibujamos el scatter-plot de las variables
# ponemos el histograma en la diagonal
panel.hist <- function(x, ...)
     {
          usr <- par("usr"); on.exit(par(usr))
          par(usr = c(usr[1:2], 0, 1.5) )
          h <- hist(x, plot = FALSE)
          breaks <- h$breaks; nB <- length(breaks)
          y <- h$counts; y <- y/max(y)
          rect(breaks[-nB], 0, breaks[-1], y, col="cyan", ...)
     }
## ponemos las correlaciones absolutas en las casillas superiores
     panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
     {
          usr <- par("usr"); on.exit(par(usr))
          par(usr = c(0, 1, 0, 1))
          r <- abs(cor(x, y))
          txt <- format(c(r, 0.123456789), digits=digits)[1]
          txt <- paste(prefix, txt, sep="")
          if(missing(cex.cor)) cex <- 0.8/strwidth(txt)
          text(0.5, 0.5, txt, cex = cex * r)
     }

# Dibujamos el scatter-plots
VELDIFTOTAL <- as.numeric(as.matrix(MATBOBINAS$VELDIFTOTAL))
VELMEDTOTAL <- as.numeric(as.matrix(MATBOBINAS$VELMEDTOTAL))
LARGO <- as.numeric(as.matrix(DATBOBINAS$LARGO))
ERRORMEDTOTALABS <- as.numeric(as.matrix(MATBOBINAS$ERRORMEDTOTALABS))
ESPENT <- as.numeric(as.matrix(DATBOBINAS$ESPENT))
ANCHO <- as.numeric(as.matrix(DATBOBINAS$ANCHO))
SECCION <- ANCHO*ESPENT

# Eliminamos las Velocidades Espúreos
INDVEL <- VELMEDTOTAL<600 & ERRORMEDTOTALABS<100

MAT <- as.matrix(cbind(VELDIFTOTAL[INDVEL], VELMEDTOTAL[INDVEL], LARGO[INDVEL],
SECCION[INDVEL], ERRORMEDTOTALABS[INDVEL]))
colnames(MAT) <- c("VELDIF","VELMED","LARGO","SECCION","ERRORMEDABS")

pairs(MAT, lower.panel=panel.smooth, upper.panel=panel.cor,diag.panel=
panel.hist)
```

Figura 227. Programa que realiza el scatter-plots de las variables estudiadas.

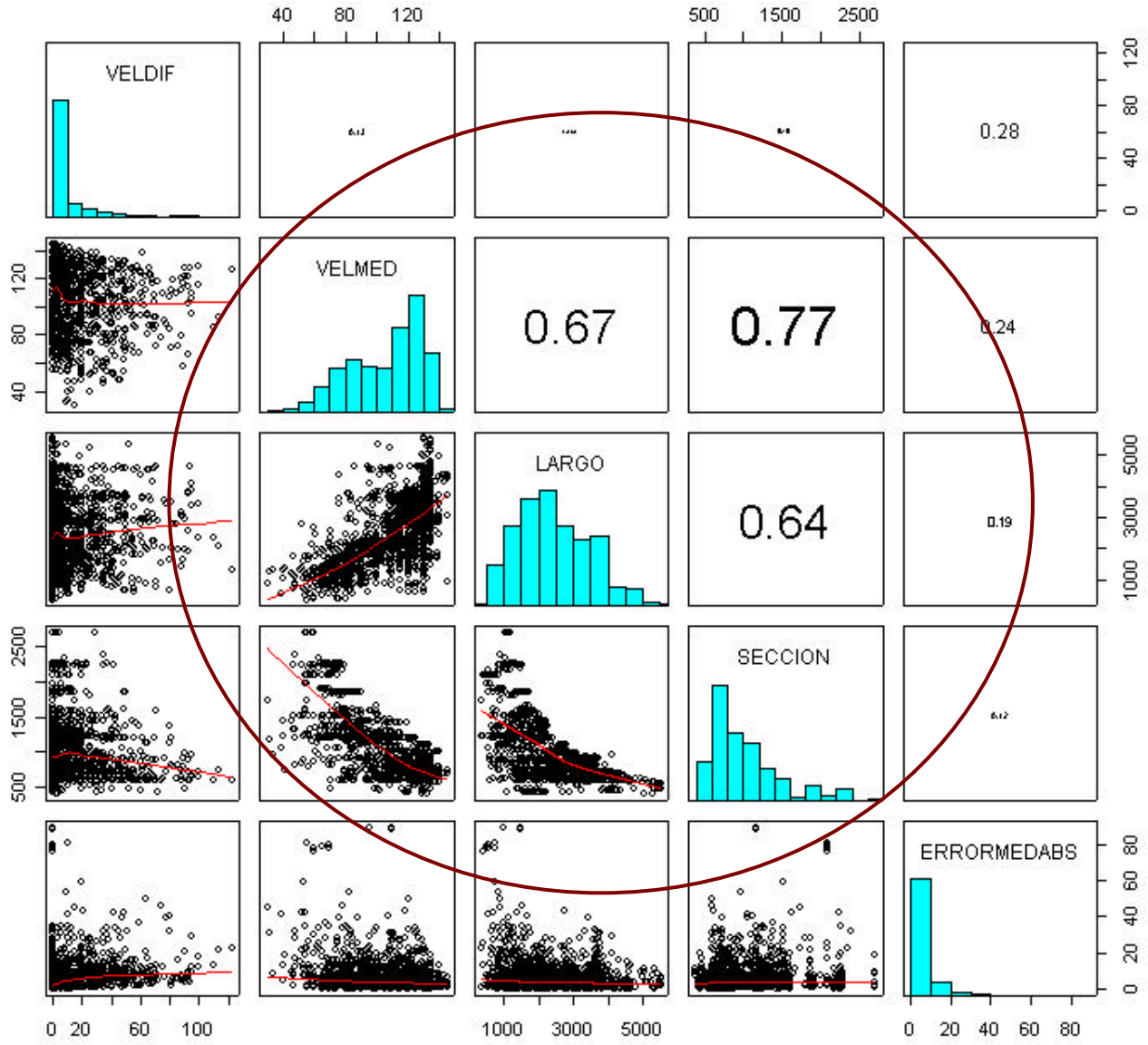


Figura 228. Comparación entre las variables estudiadas.

La Figura 228 nos muestra la correlación y distribución entre las diferentes variables estudiadas. Vemos que solo son destacables las correlaciones entre la sección de la banda, la longitud y la velocidad media de la misma. **El error no presenta ninguna correlación lineal importante con las dimensiones de la banda ni con velocidad de la misma, es decir, se aprecia que estas variables, inicialmente, no son causa directa de los errores de temperatura en la banda. Bien es cierto, que aún así, se advierte una pequeña correlación del error medio absoluto con la diferencia de velocidades.**

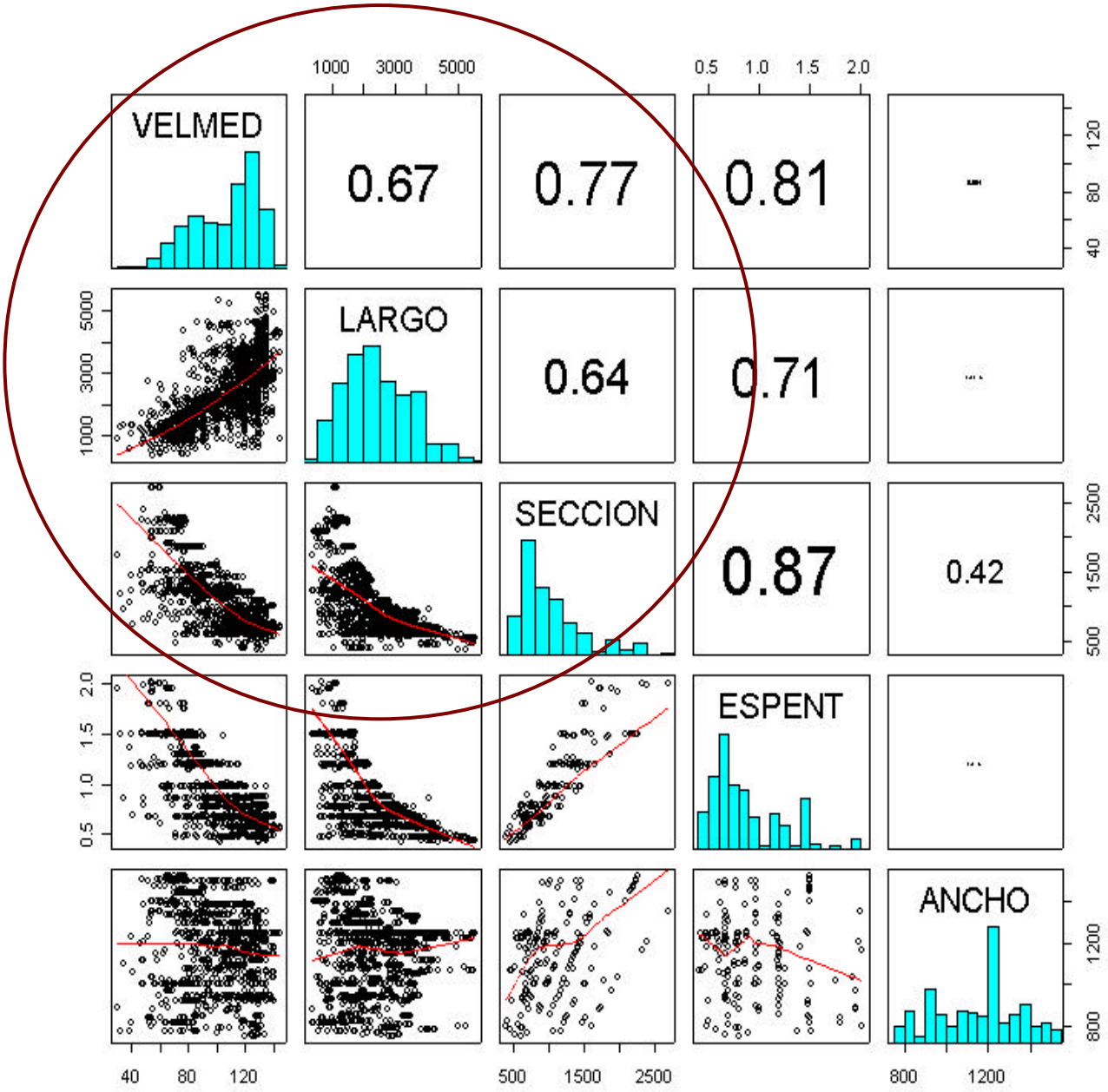


Figura 229. Relación entre la velocidad y dimensiones de las bobinas.

Claramente se advierte una relación inversa entre la sección y la velocidad media que cumple el siguiente supuesto:

"La velocidad será inversamente proporcional a la sección de la banda para poder transmitir la misma cantidad de energía calorífica por mm^3"

Aún así, también dependerá del tipo de acero y a capacidad de calentamiento de la banda, lo que claramente significa, que el estudio deberá ser realizado para cada grupo de bobinas por separado.

De la Figura 228 y Figura 229 comprobamos relaciones más obvias, como por ejemplo:

- La relación inversa entre el espesor de la banda y su longitud, ya que las mismas bobinas según su espesor serán más o menos largas debido a que el *tocho* de donde se obtienen, tiene la mismas dimensiones. Es lógico, que estas variables son muy dependientes entre si por lo que para evitar efectos de multicolinealidad será conveniente no emplear más que una de ellas.
- Al depender la longitud del espesor, la correlación se propaga con la velocidad media.

Posteriores estudios deberán tener en cuenta estas dependencias, de forma que, de las variables que determinan las dimensiones de la bobina (espesor de la bobina a la entrada, anchura de la bobina, longitud, sección) solamente se usará una o dos, como por ejemplo, **el espesor de la bobina a la entrada del horno (*ESPENT*) o la sección (*SECCION*).**

Lo que llama la atención es que el error medio total no presenta ninguna correlación representativa con las variables estudiadas, **excepto una pequeña correlación con la diferencia entre velocidades dentro de una misma bobina *VELDIFTOTAL*, (variable que nos da una idea de cómo han variado las velocidades en cada bobina). Esto es también lógico porque variaciones bruscas en la velocidad de la banda afectarán claramente a la cantidad de energía calorífica que se le suministrará a la banda.**

6.2.1.2 ESTUDIO DE LAS RELACIÓN ENTRE LAS VELOCIDADES, EL ERROR Y EL TIPO DE BOBINA

Antes de continuar con otros estudios, resulta interesante comprobar el grado de relación de las dimensiones de cada bobina con respecto al tipo de la misma.

Clase Acero	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
Número de Bobinas	51	129	45	650	152	82	295	52	113	63
Dureza del Acero	17	19	15	50	50	50	50	E8	E8	14

Tabla 47. Número de bobinas más tratadas.

Podemos ver las velocidades de tratamiento para cada tipo de bobina. Para ello, realizamos una tabla comparadora entre las diferentes clases de aceros y las velocidades (agrupadas en decenas de metros por minuto).

```
table(10*round(as.numeric(as.matrix(MATBOBINAS$VELMEDTOTAL))/10),DATBOBINAS$CLAS
ACERO)

      B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53 B014F55
  30        0       0       0       0       0       0       0       0       0
  40        0       0       0       0       1       0       0       0       0
  50        0       1       0       0       0       0       0       0       0
  60        1       0       0       0       0       1       0       0       0
  70        0       0       0       2       0       0       0       0       0
  80        1       3       0       2       1       0       1       0       0
  90        0       1       0       1       0       0       0       0       0
  100       0       2       0       5       0       0       0       0       0
  110       0       0       0      13       0       0       0       0       0
  120       0       9       1      32       0       0       1       1       0
  130       0      19       3      57       5       1       1       2       0
  140       0      16       0      16       0       0       0       0       0
  1000      0       0       0       1       0       0       0       0       0
```

	B016F35	B017F53	B023H53	B025F55	B032H53	B042H53	B044H53	B081B99	B085F97
30	0	0	0	1	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	1
90	0	0	0	1	0	0	0	0	0
100	0	0	0	5	1	0	0	1	1
110	0	0	0	0	3	0	0	0	0
120	0	0	3	3	0	0	1	0	0
130	2	0	1	18	0	2	4	1	4
140	6	0	0	17	0	0	0	1	4
1000	0	0	0	0	0	0	0	1	0

	B085G99	B100B95	B100F33	B100F55	B101F55	B102G33	B102G55	B103G33	B103G55
30	0	0	0	0	0	0	0	0	0
40	0	0	0	2	0	0	0	0	0
50	0	0	0	8	0	0	0	0	0
60	0	0	0	32	1	1	1	0	0
70	0	0	0	75	0	2	3	0	0
80	0	0	0	128	2	0	7	0	1
90	1	0	0	47	0	2	10	1	6
100	1	4	0	67	7	3	11	0	0
110	1	1	1	58	1	24	18	0	2
120	1	3	0	110	1	115	18	1	2
130	7	8	0	70	0	3	8	0	2
140	26	0	0	52	0	2	6	0	2
1000	0	0	0	1	0	0	0	0	0

	B105F55	B120G55	C107G55	C114G55	C115G55	C116G55	D012F55	D012G99	D031B33
30	0	0	0	0	1	0	0	1	0
40	0	0	0	0	0	0	0	0	0
50	5	0	1	0	0	1	2	0	1
60	5	1	0	3	1	1	2	1	8
70	20	1	0	1	0	0	1	1	6
80	51	2	1	4	0	0	3	2	3
90	27	0	0	19	0	0	1	0	1
100	63	2	2	15	0	0	3	0	0
110	26	1	0	7	0	0	2	0	0
120	73	3	35	42	0	0	0	0	0
130	15	4	12	20	0	0	0	0	0
140	10	0	1	2	0	0	0	0	0
1000	0	0	0	0	0	0	0	1	0

	D032F55	D071F55	D094B33	D094G55	K011B55	K011F57	K021H43	K021H53	K022H53
30	0	0	0	0	0	0	0	0	0
40	0	0	1	0	0	0	0	0	0
50	0	0	0	0	0	1	0	0	0
60	1	0	0	0	0	0	1	0	0
70	1	0	1	1	0	0	0	1	0
80	19	2	1	9	1	5	0	0	0
90	5	11	0	1	0	3	0	0	0
100	1	1	0	0	0	8	0	1	0
110	0	0	0	0	1	6	0	1	0
120	0	4	0	0	3	5	0	5	2
130	0	0	0	0	4	33	0	22	0
140	0	0	0	0	1	2	0	0	0
1000	0	0	0	0	0	0	0	0	0

	N013H53	N017B97	X100G99
30	0	0	0
40	0	0	0
50	1	0	0
60	0	0	0
70	1	0	1
80	0	1	0
90	2	0	0
100	1	0	0
110	1	0	0
120	0	0	5
130	1	0	14
140	0	0	19
1000	0	0	0

Figura 230. Tabla comparativa entre las velocidades (agrupadas por decenas de metros/minuto) y las clases de acero.

La tabla obtenida en la Figura 230 puede ser simplificada para las bobinas más comunes (el 73,43%), ya que las restantes, debido a su escasez, no pueden ser estudiadas con profundidad.

	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57	Acero
VELCENMED	17	19	15	50	50	50	50	E8	E8	14	Dureza
30	0	0	1	0	0	0	0	0	0	0	
40	0	0	0	2	0	0	0	0	0	0	
50	1	0	0	8	0	0	5	1	0	0	
60	0	0	0	32	1	1	5	0	3	0	
70	0	2	0	75	2	3	20	0	1	0	
80	3	2	0	128	0	7	51	1	4	1	
90	1	1	1	47	2	10	27	0	19	0	
100	2	5	5	67	3	11	63	2	15	0	
110	0	13	0	58	24	18	26	0	7	1	
120	9	32	3	110	115	18	73	35	42	3	
130	19	57	18	70	3	8	15	12	20	4	
140	16	16	17	52	2	6	10	1	2	1	TOTAL
SUMA:	**35**	**112**	**28**	**597**	**150**	**76**	**285**	**51**	**111**	**9**	**1454**
% del TOTAL:	**1,77%**	**5,66%**	**1,41%**	**30,17%**	**7,58%**	**3,84%**	**14,40%**	**2,58%**	**5,61%**	**0,45%**	**73,47%**

Tabla 48. Relación entre las velocidades y las clases de acero de las bobinas.

De la Tabla 48, se puede apreciar claramente que las bobinas con dureza: 17, 19, 15 y 14 (un 9,3%) son tratadas a alta velocidad., mientras que las bobinas con acero de dureza 50 (un 56%), tienen una distribución de velocidades más variable.

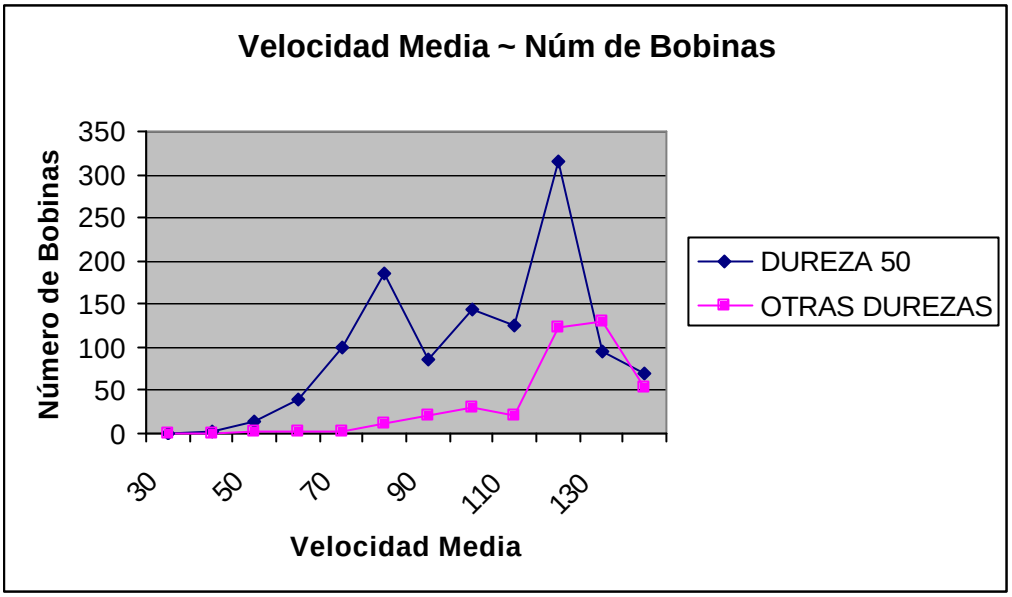


Figura 231. Distribución de número de bobinas tratadas frente a la velocidad.

6.2.1.3 COMPARACIÓN FRENTE AL "MODO DE USO"

Podemos suponer, que las velocidades bajas pueden ser debidas a tratamientos en "modo manual" sobretodo en las bobinas que no son de dureza 50. Para determinar la veracidad de esta suposición, realizamos otras tablas comparadoras.

```
# Extraemos los datos de las bobinas más usuales
BOBINASESTUDIAR <- c("B011F97", "B012F53", "B025F55", "B100F55", "B102G33",
"B102G55", "B105F55", "C107G55", "C114G55", "K011F57")

DATBOBINASESTU <- DATBOBINAS[DATBOBINAS$CLASACERO %in% BOBINASESTUDIAR,]
MATBOBINASESTU <- MATBOBINAS[DATBOBINAS$CLASACERO %in% BOBINASESTUDIAR,]

# Estudiamos el número de bobinas tratadas manualmente y automáticamente
table(DATBOBINASESTU$CLASACERO,MATBOBINASESTU$MODOBOB)

         -0.5  -1   0
  B011F97    1   3  47
  B012F53    0  60  69
  B025F55    0  23  22
  B100F55    0 183 467
  B102G33    0  84  68
  B102G55    0  17  65
  B105F55    0  94 201
  C107G55    0   4  48
  C114G55    1  24  88
  K011F57    0  38  25
```

```
# Analizamos los modos frente a la dureza
table(DATBOBINASESTU$DUREZA,MATBOBINASESTU$MODOBOB)

     -0.5  -1   0
  14    0  38  25
  15    0  23  20
  17    1   3  44
  19    0  60  69
  50    0 378 794
  E8    1  28 135

# Analizamos la velocidad frente al modo de operación
table(10*round(as.numeric(as.matrix(MATBOBINAS$VELMEDTOTAL))/10),MATBOBINAS$MODO
BOB)
       -0.5  -1   0
  30      0   0   3
  40      0   1   3
  50      0   4  17
  60      0   9  52
  70      0  32  86
  80      0  57 194
  90      0  33 107
  100     1  57 147
  110     0  50 117
  120     1 173 305
  130     0 105 238
  140     1  48 134
```

Figura 232. Comparación entre bobinas tratadas manualmente y automáticamente.

Del estudio de las tablas de la Figura 232 podemos concluir que:

- El grado de uso del "modo manual" es mayoritario frente al uso "automático". Así se advierte de la Tabla 49, donde la mayoría de las bobinas tienen un índice mayor de bobinas tratadas en "modo manual" frente al "modo automático", incluso algunas prácticamente sobrepasan el 90%.

Código	Modo Auto.	Modo Manual	M. Manual ~ M. Auto.
B011F97	3	47	94%
B012F53	60	69	53%
B025F55	23	22	49%
B100F55	183	467	72%
B102G33	84	68	45%
B102G55	17	65	79%
B105F55	94	201	68%
C107G55	4	48	92%
C114G55	24	88	79%
K011F57	38	25	40%

Tabla 49. Relación entre el modo manual y el automático con el tipo de acero.

- Si analizamos el tipo de uso según la dureza, podemos advertir que dos tipos de durezas tienen un grado elevado de uso en "modo manual" frente al "automático" (17 y E8).

Dureza	Modo Auto.	Modo Manual	M. Manual ~ M. Auto.
14	**38**	**25**	**40%**
15	**23**	**20**	**47%**
17	**3**	**44**	**94%**
19	**60**	**69**	**53%**
50	**378**	**794**	**68%**
E8	**28**	**135**	**83%**

Tabla 50. Relación entre el modo manual y el automático con la dureza del acero.

- La velocidad media de tratamiento de las bobinas tiene un rango mayor en el modo manual que en el automático (ver Tabla 51), lo que indica claramente, que incluso a velocidades elevadas las bobinas son tratadas manualmente.

Vel	Modo Auto.	Modo Manual	M. Manual ~ M. Auto.
30	0	3	100%
40	1	3	75%
50	4	17	81%
60	9	52	85%
70	32	86	73%
80	57	194	77%
90	33	107	76%
100	57	147	72%
110	50	117	70%
120	173	305	64%
130	105	238	69%
140	48	134	74%

Tabla 51. Relación entre el modo manual y el automático según la velocidad media.

6.2.1.4 CONCLUSIONES DEL ANÁLISIS ENTRE LA VELOCIDAD Y EL TIPO DE ACERO

De las tablas anteriores, podemos concluir que existen algunos tipo de aceros que prácticamente no son usados en "modo automático" seguramente porque el modelo matemático no es capaz de tratarlo convenientemente. **Claramente se aprecia la necesidad de tener un modelo adaptado a cada tipo de clase de acero o tipo de dureza.**

6.2.2 ESTUDIO DE LA RELACIÓN ENTRE EL ERROR DE TEMPERATURA, EL TIPO DE BOBINA Y EL "MODO DE USO"

De los estudios realizados en el punto anterior, surge la pregunta de cuál es el grado de fiabilidad de cada uno de los modos. Para ello, comparamos el error absoluto de cada tipo de bobinas frente a cada uno de los modos.

```
# Separamos las bobinas tratadas en "Modo Manual" frente al "Modo Automático"

# Bobinas en "modo manual"
DATBOBINASESTUMANUAL <- DATBOBINASESTU[MATBOBINASESTU$MODOBOB==0,]
dim(DATBOBINASESTUMANUAL)
[1] 1100   13
MATBOBINASESTUMANUAL <- MATBOBINASESTU[MATBOBINASESTU$MODOBOB==0,]

# Bobinas en "modo automático"
MATBOBINASESTUAUTOMA <- MATBOBINASESTU[MATBOBINASESTU$MODOBOB==-1,]
DATBOBINASESTUAUTOMA <- DATBOBINASESTU[MATBOBINASESTU$MODOBOB==-1,]
dim(DATBOBINASESTUAUTOMA)
[1] 530  13

# Comparamos el error absoluto frente para cada bobina en modo manual
table(20*round(as.numeric(as.matrix(MATBOBINASESTUMANUAL$ERRORMEDTOTALABS))/20),
DATBOBINASESTUMANUAL$CLASACERO)

    B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53 B014F55
  0       0      42       0      58       0       0       0       0       0
  20      0       4       0       7       0       0       0       0       0
  40      0       1       0       4       0       0       0       0       0
  80      0       0       0       0       0       0       0       0       0
  140     0       0       0       0       0       0       0       0       0
  160     0       0       0       0       0       0       0       0       0
  180     0       0       0       0       0       0       0       0       0
  200     0       0       0       0       0       0       0       0       0
  220     0       0       0       0       0       0       0       0       0
  240     0       0       0       0       0       0       0       0       0
  260     0       0       0       0       0       0       0       0       0
  500     0       0       0       0       0       0       0       0       0

    B016F35 B017F53 B023H53 B025F55 B032H53 B042H53 B044H53 B081B99 B085F97
  0       0       0       0      20       0       0       0       0       0
  20      0       0       0       2       0       0       0       0       0
  40      0       0       0       0       0       0       0       0       0
  80      0       0       0       0       0       0       0       0       0
  140     0       0       0       0       0       0       0       0       0
  160     0       0       0       0       0       0       0       0       0
  180     0       0       0       0       0       0       0       0       0
  200     0       0       0       0       0       0       0       0       0
  220     0       0       0       0       0       0       0       0       0
  240     0       0       0       0       0       0       0       0       0
  260     0       0       0       0       0       0       0       0       0
  500     0       0       0       0       0       0       0       0       0
```

	B085G99	B100B95	B100F33	B100F55	B101F55	B102G33	B102G55	B103G33	B103G55
0	0	0	0	385	0	54	39	0	0
20	0	0	0	58	0	12	16	0	0
40	0	0	0	9	0	1	2	0	0
80	0	0	0	7	0	0	0	0	0
140	0	0	0	2	0	0	1	0	0
160	0	0	0	2	0	0	3	0	0
180	0	0	0	1	0	0	1	0	0
200	0	0	0	1	0	0	0	0	0
220	0	0	0	1	0	1	1	0	0
240	0	0	0	0	0	0	1	0	0
260	0	0	0	0	0	0	1	0	0
500	0	0	0	1	0	0	0	0	0

	B105F55	B120G55	C107G55	C114G55	C115G55	C116G55	D012F55	D012G99	D031B33
0	182	0	44	76	0	0	0	0	0
20	17	0	4	10	0	0	0	0	0
40	2	0	0	2	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0

	D032F55	D071F55	D094B33	D094G55	K011B55	K011F57	K021H43	K021H53	K022H53
0	0	0	0	0	0	18	0	0	0
20	0	0	0	0	0	6	0	0	0
40	0	0	0	0	0	1	0	0	0
80	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0

	N013H53	N017B97	X100G99
0	0	0	0
20	0	0	0
40	0	0	0
80	0	0	0
140	0	0	0
160	0	0	0
180	0	0	0
200	0	0	0
220	0	0	0
240	0	0	0
260	0	0	0
500	0	0	0

```
# Comparamos el error absoluto frente para cada bobina en modo automatico
table(20*round(as.numeric(as.matrix(MATBOBINASESTUAUTOMA$ERRORMEDTOTALABS))/20),
DATBOBINASESTUAUTOMA$CLASACERO)

    B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53 B014F55
0         0       3       0      57       0       0       0       0       0
20        0       0       0       3       0       0       0       0       0
40        0       0       0       0       0       0       0       0       0
80        0       0       0       0       0       0       0       0       0

    B016F35 B017F53 B023H53 B025F55 B032H53 B042H53 B044H53 B081B99 B085F97
0         0       0       0      22       0       0       0       0       0
20        0       0       0       1       0       0       0       0       0
40        0       0       0       0       0       0       0       0       0
80        0       0       0       0       0       0       0       0       0

    B085G99 B100B95 B100F33 B100F55 B101F55 B102G33 B102G55 B103G33 B103G55
0         0       0       0     176       0      65      15       0       0
20        0       0       0       5       0      19       2       0       0
40        0       0       0       0       0       0       0       0       0
80        0       0       0       2       0       0       0       0       0

    B105F55 B120G55 C107G55 C114G55 C115G55 C116G55 D012F55 D012G99 D031B33
0        88       0       1      16       0       0       0       0       0
20        5       0       3       8       0       0       0       0       0
40        1       0       0       0       0       0       0       0       0
80        0       0       0       0       0       0       0       0       0

    D032F55 D071F55 D094B33 D094G55 K011B55 K011F57 K021H43 K021H53 K022H53
0         0       0       0       0       0      37       0       0       0
20        0       0       0       0       0       1       0       0       0
40        0       0       0       0       0       0       0       0       0
80        0       0       0       0       0       0       0       0       0

    N013H53 N017B97 X100G99
0         0       0       0
20        0       0       0
40        0       0       0
80        0       0       0
```

Figura 233. Comparación del error medio absoluto frente al tipo de acero de las bobinas para los dos modos.

Los resultados pueden ser resumidos en las siguientes tablas...

ERRORABS	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
0	42	58	20	385	54	39	182	44	76	18
20	4	7	2	58	12	16	17	4	10	6
40	1	4	0	9	1	2	2	0	2	1
80	0	0	0	7	0	0	0	0	0	0
140	0	0	0	2	0	1	0	0	0	0
160	0	0	0	2	0	3	0	0	0	0
180	0	0	0	1	0	1	0	0	0	0
200	0	0	0	1	0	0	0	0	0	0
220	0	0	0	1	1	1	0	0	0	0
240	0	0	0	0	0	1	0	0	0	0
260	0	0	0	0	0	1	0	0	0	0
500	0	0	0	1	0	0	0	0	0	0
SUMA=	47	69	22	467	68	65	201	48	88	25

Tabla 52. Bobinas tratadas en modo manual.

ERRORABS	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
0	3	57	22	176	65	15	88	1	16	37
20	0	3	1	5	19	2	5	3	8	1
40	0	0	0	0	0	0	1	0	0	0
80	0	0	0	2	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0
SUMA MAN=	3	60	23	183	84	17	94	4	24	38

Tabla 53. Bobinas tratadas en modo automático.

En las figuras siguientes podemos apreciar el número de bobinas con errores elevados frente a bobinas con errores bajos comparándolos con cada "modo". En este caso, las curvas corresponden con el número de bobinas tratadas en "modo manual" (zona de azul claro) más el número de bobinas tratadas en modo automático (zona rojo oscuro), de forma que la superior marca el total de bobinas con ese tipo de error, y la distancia entre ellas, para cada error medio absoluto, corresponde con el número de bobinas del "modo automático".

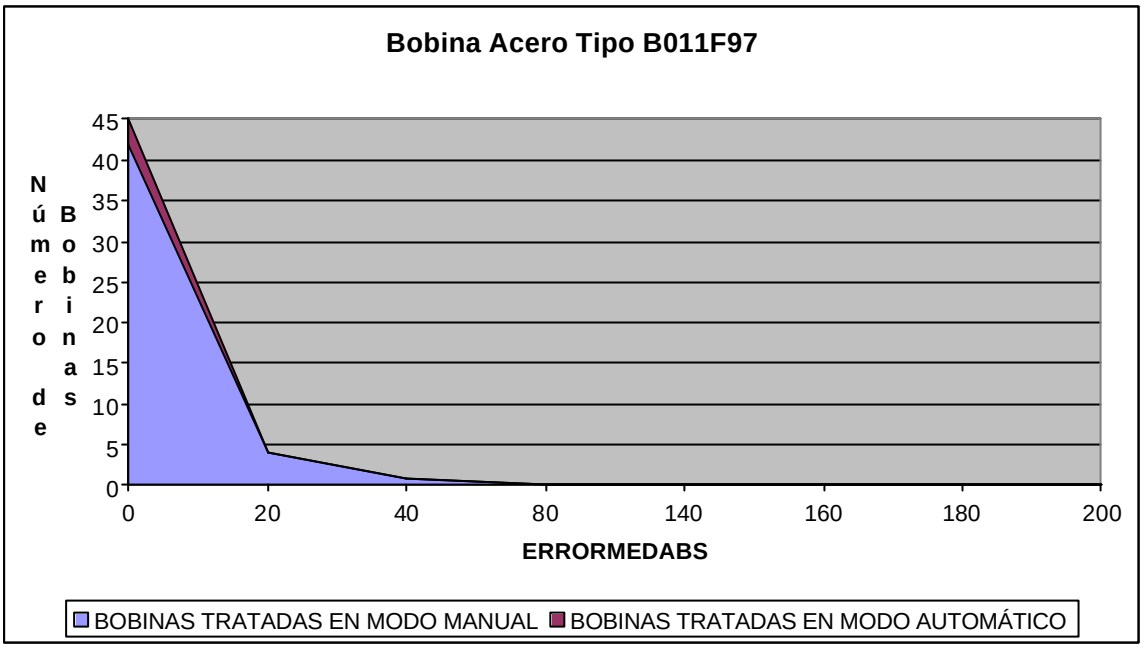


Figura 234. Número de bobinas B011F97 para cada modo.

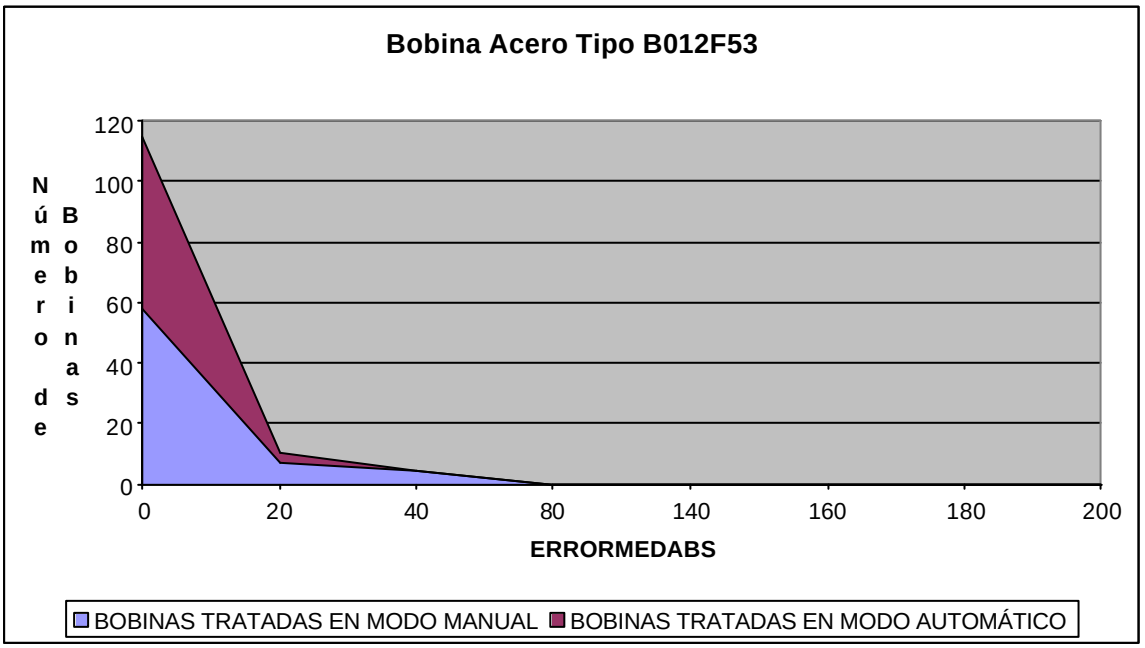


Figura 235. Número de bobinas B012F53 para cada modo.

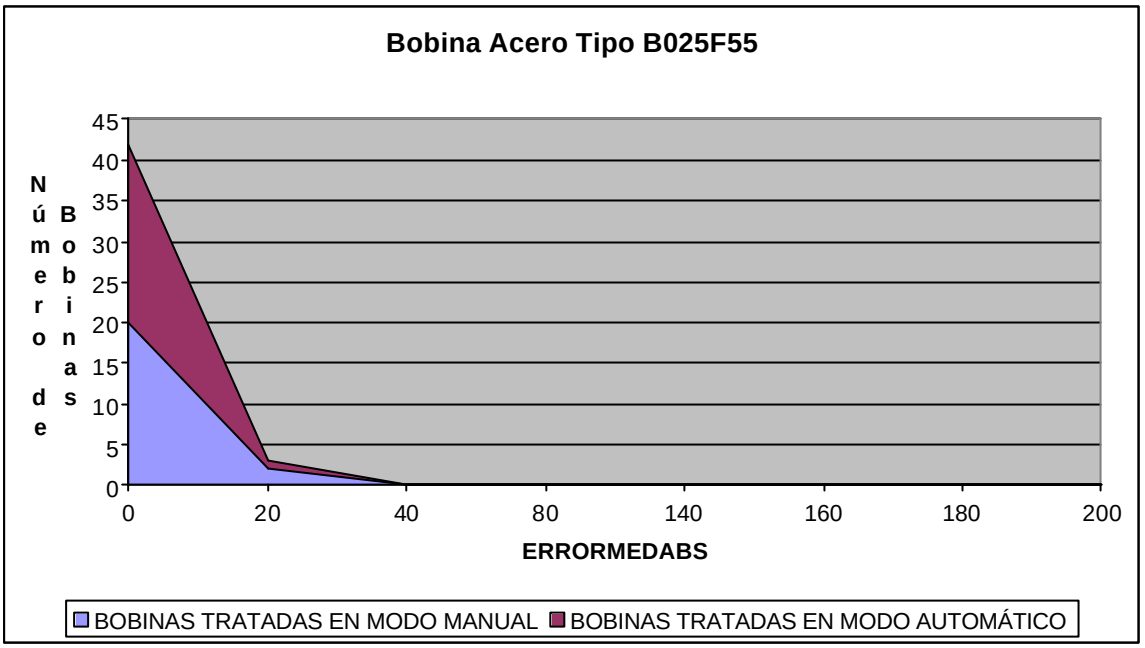


Figura 236. Número de bobinas B025F55 para cada modo.

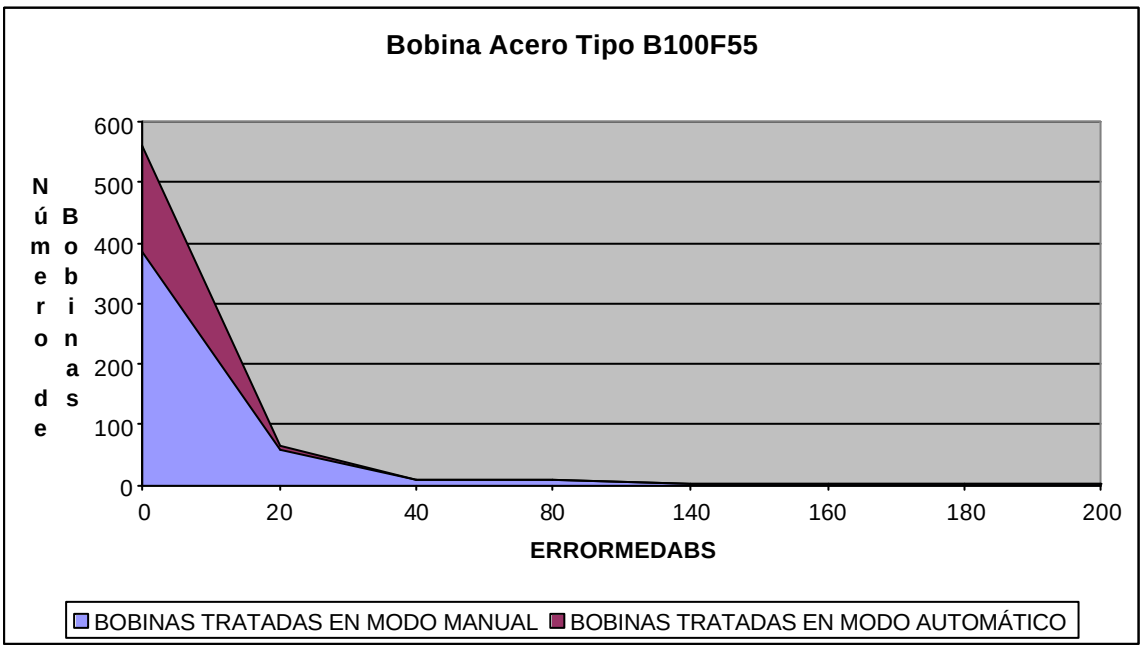


Figura 237. Número de bobinas B100F55 para cada modo.

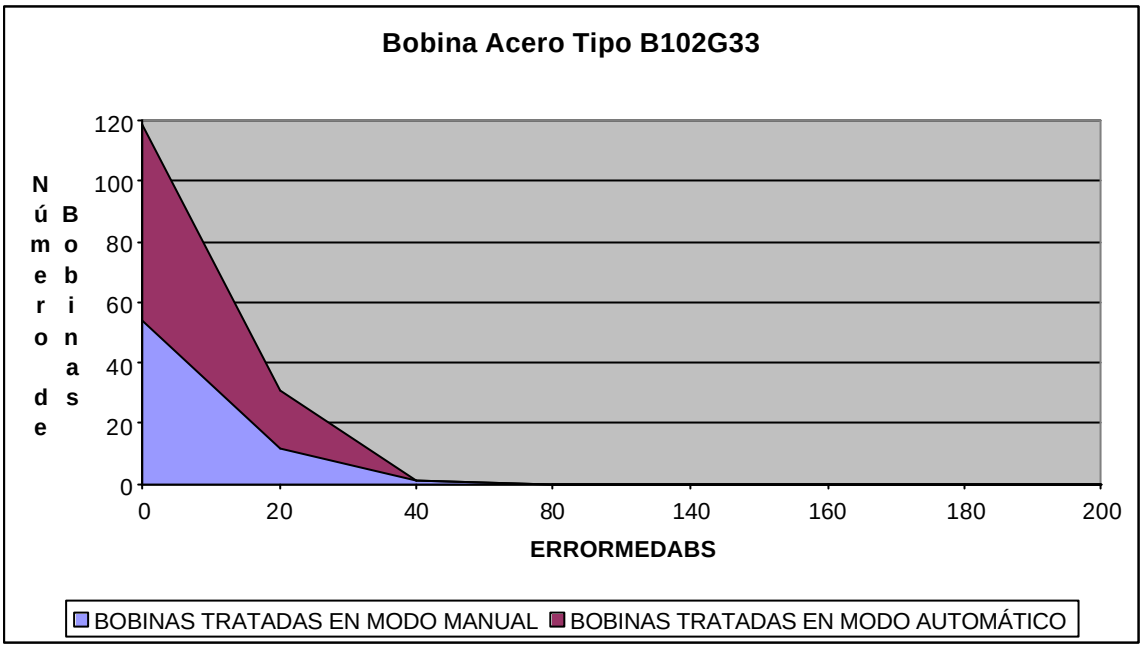


Figura 238. Número de bobinas B102G33 para cada modo.

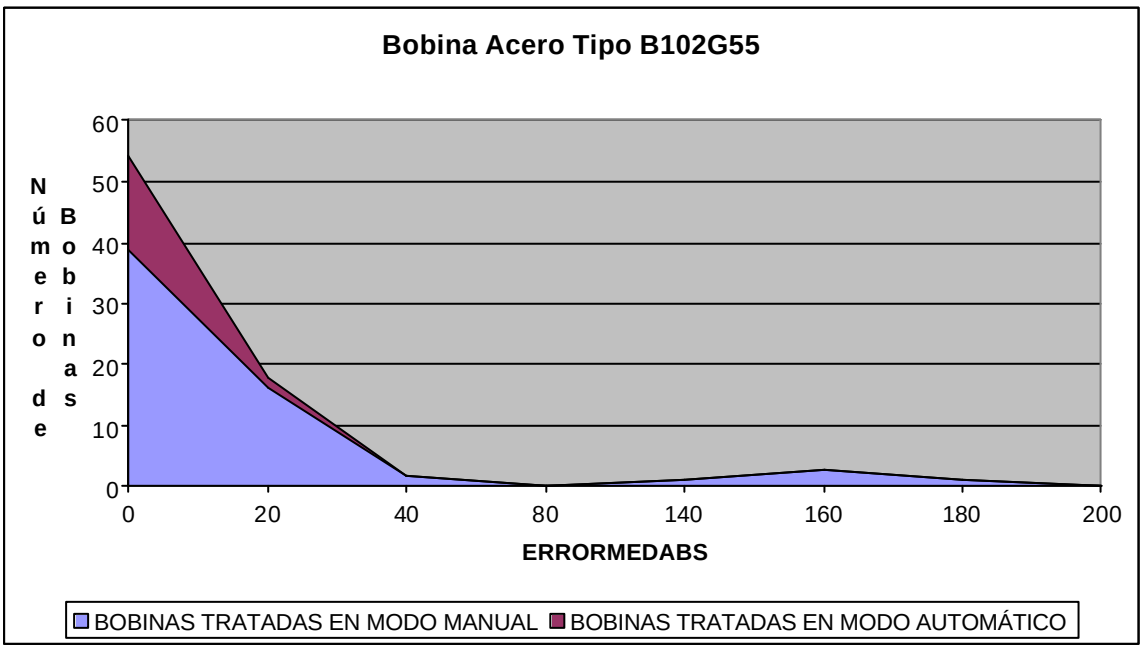


Figura 239. Número de bobinas B102G55 para cada modo.

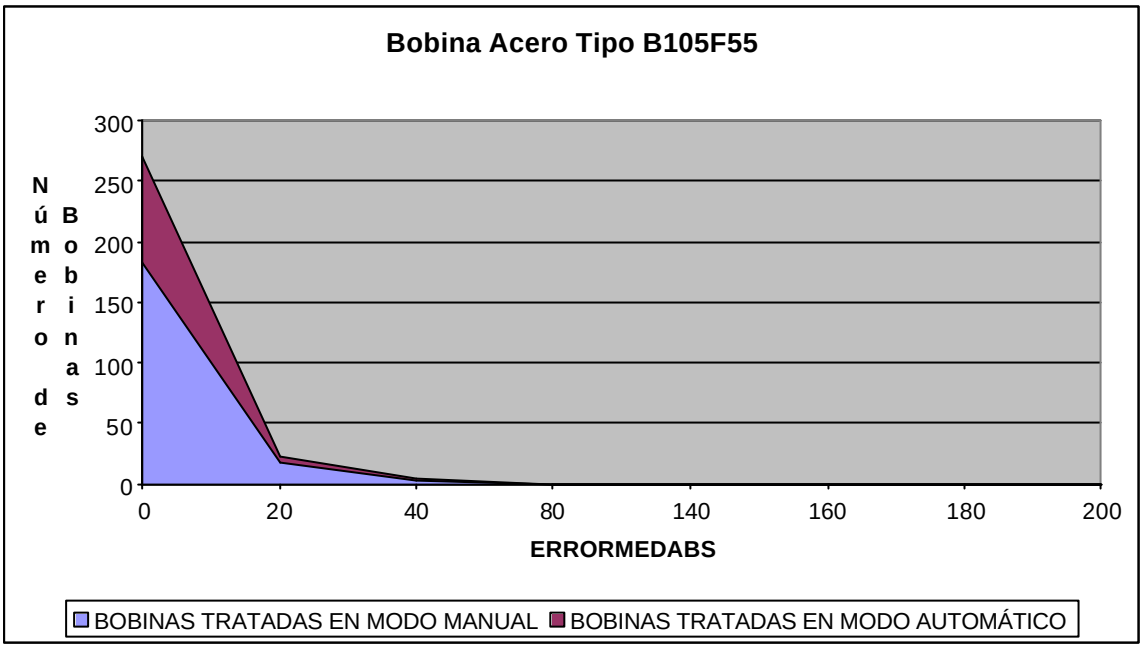


Figura 240. Número de bobinas B105F55 para cada modo.

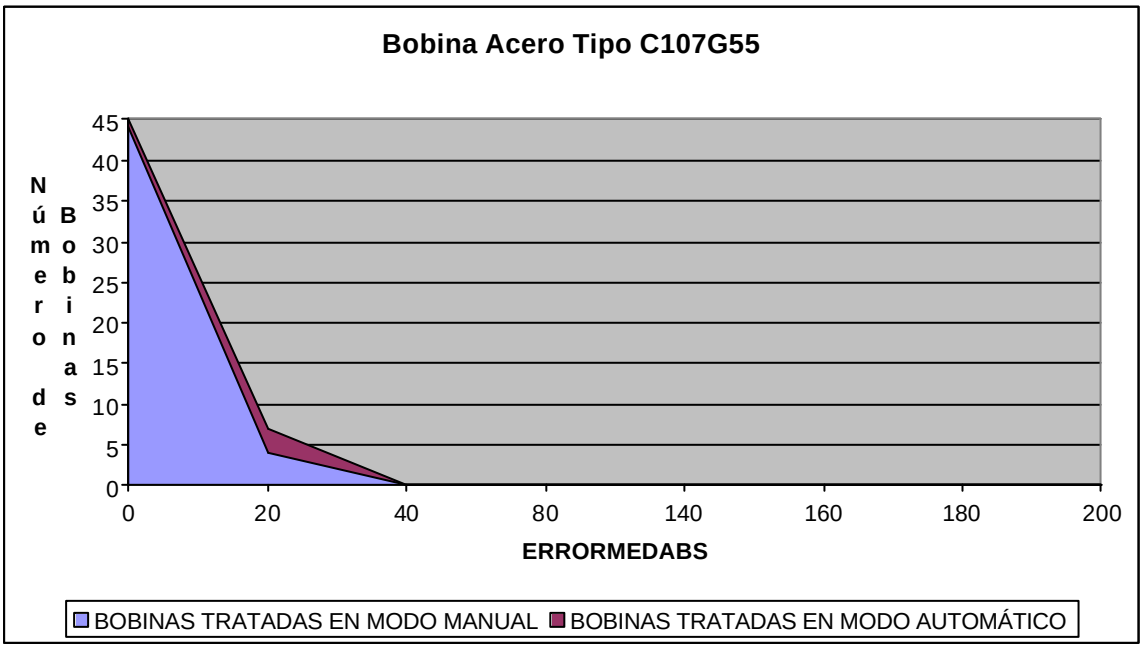


Figura 241. Porcentaje de bobinas C107G55 para cada modo.

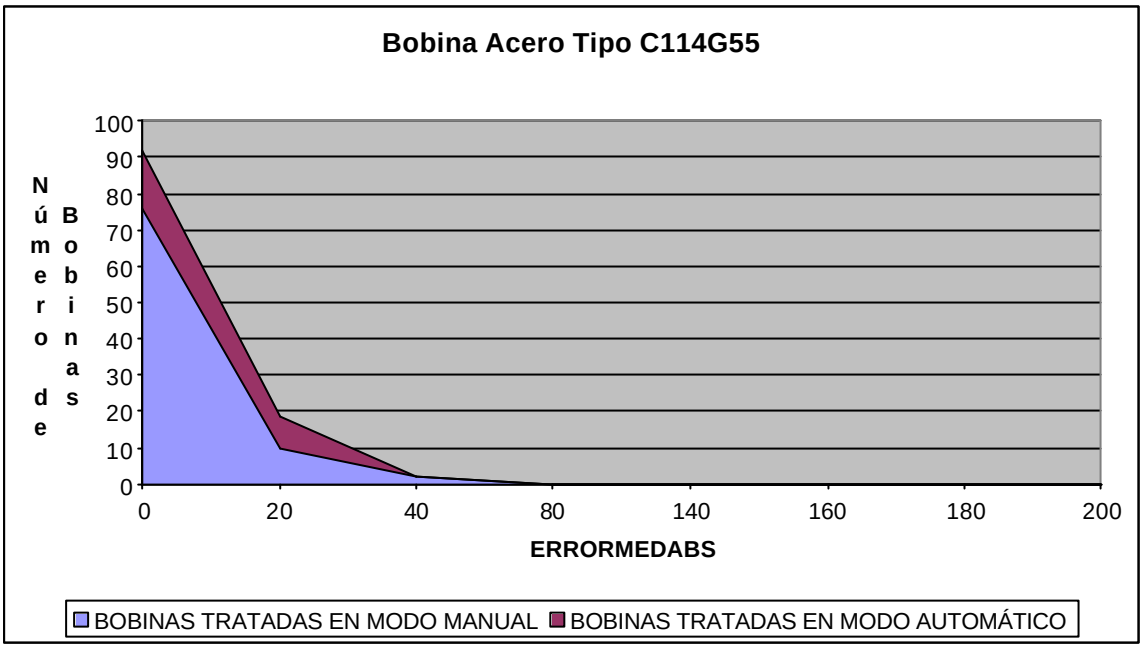


Figura 242. Porcentaje de bobinas C114G55 para cada modo.

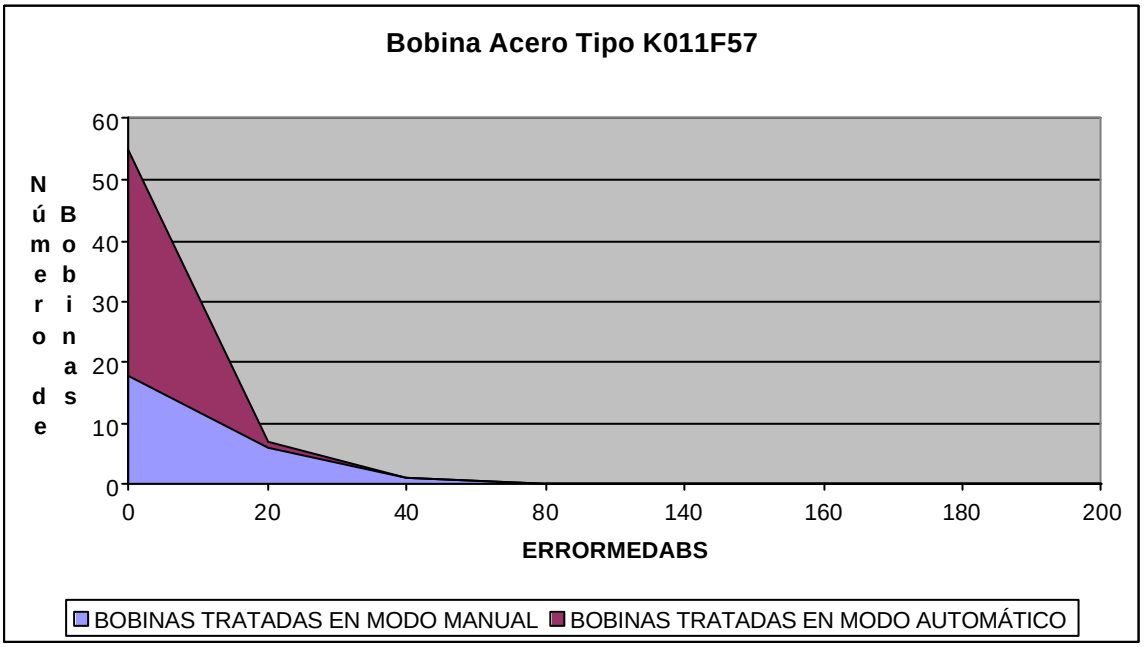


Figura 243. Porcentaje de bobinas K011F57 para cada modo.

Para resumir, agrupamos los errores en bajos, medios y altos.

```
# Caracterizamos los errores de las bobinas tratadas automaticamente
TIPOERRORMANUAL <- rep("BAJO",length(MATBOBINASESTUMANUAL$ERRORMEDTOTALABS))
TIPOERRORMANUAL[as.numeric(as.matrix(MATBOBINASESTUMANUAL$ERRORMEDTOTALABS))>20
& as.numeric(as.matrix(MATBOBINASESTUMANUAL$ERRORMEDTOTALABS))<=50]="MEDIO"
TIPOERRORMANUAL[as.numeric(as.matrix(MATBOBINASESTUMANUAL$ERRORMEDTOTALABS))>50]
="ALTO"
table(TIPOERRORMANUAL)
TIPOERRORMANUAL
 ALTO  BAJO MEDIO
   24  1024    52
# Caracterizamos los errores de las bobinas tratadas automaticamente
TIPOERRORAUTOMAT <- rep("BAJO",length(MATBOBINASESTUAUTOMA$ERRORMEDTOTALABS))
TIPOERRORAUTOMAT[as.numeric(as.matrix(MATBOBINASESTUAUTOMA$ERRORMEDTOTALABS))>20
& as.numeric(as.matrix(MATBOBINASESTUAUTOMA$ERRORMEDTOTALABS))<=50]="MEDIO"
TIPOERRORAUTOMAT[as.numeric(as.matrix(MATBOBINASESTUAUTOMA$ERRORMEDTOTALABS))>50
]="ALTO"
table(TIPOERRORAUTOMAT)
TIPOERRORAUTOMAT
 ALTO  BAJO MEDIO
    2   521     7
```

Figura 244. Programa que caracteriza los errores en tres tipos.

	MODO MANUAL		MODO AUTOMATICO	
Tipo de Error	Núm.	En %	Núm.	En %.
BAJO <=20ºC	1024	93,1%	521	98,3%
MEDIO (>20ºC & <=50ºC)	52	4,7%	7	1,3%
ALTO (>50ºC)	24	2,2%	2	0,4%

Tabla 54. Porcentajes globales de tipos de error medio absoluto para los dos modos.

En la Tabla 54 podemos observar que el porcentaje de bobinas en las que se observan errores medios y altos de diferentes tipos son mayores en el "modo manual" que en el "modo automático". **Claramente se observa, que en el "modo automático", se asegura un 98,3% de bobinas con errores BAJOS frente al 93,1% del "modo manual".**

Más conveniente, es estudiar el resumen para cada tipo de acero.

```
# Comparamos el error absoluto frente para cada bobina en modo manual
table(TIPOERRORMANUAL,DATBOBINASESTUMANUAL$CLASACERO)

TIPOERRORMANUAL B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53
          ALTO        0       0       0       0       0       0       0       0
          BAJO        0      45       0      62       0       0       0       0
          MEDIO       0       2       0       7       0       0       0       0

TIPOERRORMANUAL B014F55 B016F35 B017F53 B023H53 B025F55 B032H53 B042H53 B044H53
          ALTO        0       0       0       0       0       0       0       0
          BAJO        0       0       0       0      21       0       0       0
          MEDIO       0       0       0       0       1       0       0       0
```

```
TIPOERRORMANUAL B081B99 B085F97 B085G99 B100B95 B100F33 B100F55 B101F55 B102G33
          ALTO        0       0       0       0       0      15       0       1
          BAJO        0       0       0       0       0     428       0      65
          MEDIO       0       0       0       0       0      24       0       2

TIPOERRORMANUAL B102G55 B103G33 B103G55 B105F55 B120G55 C107G55 C114G55 C115G55
          ALTO        8       0       0       0       0       0       0       0
          BAJO       51       0       0     198       0      48      83       0
          MEDIO       6       0       0       3       0       0       5       0

TIPOERRORMANUAL C116G55 D012F55 D012G99 D031B33 D032F55 D071F55 D094B33 D094G55
          ALTO        0       0       0       0       0       0       0       0
          BAJO        0       0       0       0       0       0       0       0
          MEDIO       0       0       0       0       0       0       0       0

TIPOERRORMANUAL K011B55 K011F57 K021H43 K021H53 K022H53 N013H53 N017B97 X100G99
          ALTO        0       0       0       0       0       0       0       0
          BAJO        0      23       0       0       0       0       0       0
          MEDIO       0       2       0       0       0       0       0       0


# Comparamos el error absoluto frente para cada bobina en modo automático
table(TIPOERRORAUTOMAT, DATBOBINASESTUAUTOMA$CLASACERO)

TIPOERRORAUTOMAT B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53
           ALTO        0       0       0       0       0       0       0       0
           BAJO        0       3       0      59       0       0       0       0
           MEDIO       0       0       0       1       0       0       0       0

TIPOERRORAUTOMAT B014F55 B016F35 B017F53 B023H53 B025F55 B032H53 B042H53 B044H53
           ALTO        0       0       0       0       0       0       0       0
           BAJO        0       0       0       0      23       0       0       0
           MEDIO       0       0       0       0       0       0       0       0

TIPOERRORAUTOMAT B081B99 B085F97 B085G99 B100B95 B100F33 B100F55 B101F55 B102G33
           ALTO        0       0       0       0       0       2       0       0
           BAJO        0       0       0       0       0     181       0      82
           MEDIO       0       0       0       0       0       0       0       2

TIPOERRORAUTOMAT B102G55 B103G33 B103G55 B105F55 B120G55 C107G55 C114G55 C115G55
           ALTO        0       0       0       0       0       0       0       0
           BAJO       17       0       0      93       0       4      21       0
           MEDIO       0       0       0       1       0       0       3       0

TIPOERRORAUTOMAT C116G55 D012F55 D012G99 D031B33 D032F55 D071F55 D094B33 D094G55
           ALTO        0       0       0       0       0       0       0       0
           BAJO        0       0       0       0       0       0       0       0
           MEDIO       0       0       0       0       0       0       0       0

TIPOERRORAUTOMAT K011B55 K011F57 K021H43 K021H53 K022H53 N013H53 N017B97 X100G99
           ALTO        0       0       0       0       0       0       0       0
           BAJO        0      38       0       0       0       0       0       0
           MEDIO       0       0       0       0       0       0       0       0
```

Figura 245. Obtención de tipos de errores para cada bobina.

TIPOERRORMANUAL	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
ALTO	0	0	0	15	1	8	0	0	0	0
MEDIO	2	7	1	24	2	6	3	0	5	2
BAJO	45	62	21	428	65	51	198	48	83	23

Tabla 55. Número de bobinas de cada tipo de error para el modo manual.

TIPOERRORAUTOMAT	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
ALTO	0	0	0	2	0	0	0	0	0	0
MEDIO	0	1	0	0	2	0	1	0	3	0
BAJO	3	59	23	181	82	17	93	4	21	38

Tabla 56. Número de bobinas de cada tipo de error para el modo automático.

% ERRORABSMANUAL	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	4,3%	10,1%	4,5%	8,4%	4,4%	21,5%	1,5%	0,0%	5,7%	8,0%
BAJO	95,7%	89,9%	95,5%	91,6%	95,6%	78,5%	98,5%	100,0%	94,3%	92,0%

Tabla 57. Comparación con los tipos de error para el modo manual (en porcentajes relativos).

% ERRORABSAUTOMAT	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	0,0%	1,7%	0,0%	1,1%	2,4%	0,0%	1,1%	0,0%	12,5%	0,0%
BAJO	100,0%	98,3%	100,0%	98,9%	97,6%	100,0%	98,9%	100,0%	87,5%	100,0%

Tabla 58. Comparación con los tipos de error para el modo automático (en porcentajes relativos).

6.2.2.1 CONCLUSIONES DEL ESTUDIO ENTRE EL ERROR DE TEMPERATURA, EL TIPO DE ACERO Y EL "MODO DE USO"

Analizando todas las tablas obtenidas anteriormente y comparando los modos "automático y manual" frente a cada bobina podemos concluir que:

- El procesado en "modo automático" supera en casi todos los aceros al procesado en "modo manual", produciendo un porcentaje mayor de bobinas con errores BAJOS frente al "modo manual".
- Las bobinas con aceros B012F53 y B102G55 presentan una mayor cantidad de errores elevados y medios en el "modo manual" que en el "modo automático".
- La gran mayoría de bobinas con aceros tipo B011F97 (50 bobinas) y C107G55 (129 bobinas) apenas han sido tratadas en el "modo automático" (menos de un 10%) y cuando esto ha sucedido, el porcentaje de errores "BAJOS" ha sido del 100%. Sería conveniente estudiar el funcionamiento en "modo automático" para este tipo de bobinas, ya que su eficiencia es del 100%.
- En los aceros más utilizados B100F55 y B105F55 (945 bobinas) el uso del "modo manual" o "modo automático" no presenta grandes diferencias para el segundo tipo, **pero si se reduce claramente la efectividad del primero.**
- Solamente el "modo manual" para la bobina C114G55, mejora frente al "modo automático".

Concluyendo, podemos deducir que **el control automático del horno funciona mejor que el "modo manual" para casi todos los tipos de aceros estudiados.** Aunque sería conveniente analizar cuándo se producen los cambios del "modo automático" al "modo manual" o viceversa y por qué, ya que estas suposiciones pueden ser equivocadas. Por ejemplo, puede suceder que el operario deba pasar al "modo manual" para resolver contingencias y solo cuando el horno está funcionando correctamente, lo vuelven a pasar a "modo automático", lo que perjudica claramente las estadísticas del "modo manual", ya que los errores aparecerán siempre cuando se resuelven las contingencias.

6.2.3 ESTUDIO DE LA RELACIÓN ENTRE EL ERROR DE LA TEMPERATURA MEDIDA DE LA BOBINAS Y LAS TEMPERATURAS DE CONSIGNA DEL HORNO

Seguimos analizando el comportamiento del "error" con otras variables. En este caso se decide estudiar si existe alguna relación entre las variables de consigna del horno con el error.

Las variables que van a entrar en el estudio son:

- *THF1DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de consigna de zona 1 para cada bobina. Se utiliza para determinar el grado de estabilización de la curva de temperatura en ese punto.
- *THF1MEDTOTAL*: Temperatura de consigna de la zona 1.
- *THF3MEDTOTAL*: Temperatura de consigna de la zona 3.
- *THF5MEDTOTAL*: Temperatura de consigna de la zona 5.
- *THF1DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de consigna de zona 3 para cada bobina.
- *THF3DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de consigna de zona 3 para cada bobina.
- *THF5DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de consigna de zona 5 para cada bobina.
- *STEP1*: Diferencia de temperaturas entre la zona 1 y la 3.
- *STEP2*: Diferencia de temperaturas entre la zona 1 y la 5.
- *Abs(ERRORMEDTOTAL)*: Valor absoluto de la diferencia entre la temperatura de consigna del pirómetro dos (temperatura esperada de la bobina a la salida de la zona de calentamiento) y la real.

Lo primero que hacemos, es echar un vistazo a los gráficos de *box-plots* de las variables a estudiar.

```
# Obtenemos las nuevas variables THF3DIFTOTAL y THF5DIFTOTAL #
#############################################################

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC2$THC3>100
MATSINRUIDO <- MATDINAMIC2[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
LISTABOBINASVELBUENAS <- unique(MATBOBINAS2[,1])
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979

# Obtenemos el valor de consigna máximo y mínimo de temperatura por bobina
MINTMPP <- tapply(MATSINRUIDO$THC3, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(MATSINRUIDO$THC3, MATSINRUIDO$CODBOBINA,max)

THF3DIFTOTAL <- MAXTMPP-MINTMPP

# Obtenemos las nuevas variables

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC2$THC5>100
MATSINRUIDO <- MATDINAMIC2[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
LISTABOBINASVELBUENAS <- unique(MATBOBINAS2[,1])
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979

# Obtenemos el valor de consigna máximo y mínimo de temperatura por bobina
MINTMPP <- tapply(MATSINRUIDO$THC5, MATSINRUIDO$CODBOBINA,min)
MAXTMPP <- tapply(MATSINRUIDO$THC5, MATSINRUIDO$CODBOBINA,max)

THF5DIFTOTAL <- MAXTMPP-MINTMPP

STEP1 <- as.numeric(as.matrix(MATBOBINAS2$THF3MEDTOTAL))-
as.numeric(as.matrix(MATBOBINAS2$THF1MEDTOTAL))
STEP2 <- as.numeric(as.matrix(MATBOBINAS2$THF5MEDTOTAL))-
as.numeric(as.matrix(MATBOBINAS2$THF1MEDTOTAL))

#############################################################################

# Obtenemos la lista de errores medios absolutos
MATERRORNUM <-as.numeric(as.matrix(MATBOBINAS2$ERRORMEDTOTALABS))
# Dibujamos los gráficos de box-plot de las variables
# THF1DIFTOTAL, THF1MEDTOTAL, THF3DIFTOTAL, THF5DIFTOTAL, STEP1, STEP2,
ERRORMEDTOTAL)
par(mfrow=c(2,4),cex=1)
boxplot(as.numeric(as.matrix(MATBOBINAS2$THF1DIFTOTAL)),col="blue",notch=TRUE,xl
ab="THF1DIFTOTAL")
```

```
boxplot(as.numeric(as.matrix(MATBOBINAS2$THF1MEDTOTAL)),col="blue",notch=TRUE,
xlab="THF1MEDTOTAL")
boxplot(as.numeric(as.matrix(MATBOBINAS2$THF3MEDTOTAL)),col="blue",notch=TRUE,
xlab="THF3MEDTOTAL")
boxplot(THF3DIFTOTAL,col="blue",notch=TRUE, xlab="THF3DIFTOTAL")
boxplot(THF5DIFTOTAL,col="blue",notch=TRUE, xlab="THF5DIFTOTAL")
boxplot(STEP1,col="blue",notch=TRUE, xlab="STEP1")
boxplot(STEP2,col="blue",notch=TRUE, xlab="STEP2")
boxplot(MATERRORNUM,col="blue",notch=TRUE, xlab="ERRORMEDTOTAL")
# Obtenemos un resumen de los datos
summary(as.numeric(as.matrix(MATBOBINAS2$THF1DIFTOTAL)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    1.00    2.00    9.14   10.00   99.00
summary(as.numeric(as.matrix(MATBOBINAS2$THF1MEDTOTAL)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    702     795     825     816     844     877
summary(THF3DIFTOTAL)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    1.00    2.00    9.52   10.00  125.00
summary(as.numeric(as.matrix(MATBOBINAS2$THF3MEDTOTAL)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  734.0   825.0   855.0   846.1   875.0   907.0
summary(THF5DIFTOTAL)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    1.00    5.00   11.56   14.00  293.00
summary(as.numeric(as.matrix(MATBOBINAS2$THF5MEDTOTAL)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  734.0   841.0   874.0   865.5   894.0   932.0
summary(STEP1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -3.00   30.00   30.00   30.17   30.00   63.00
summary(STEP2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-116.00   46.00   49.00   49.56   52.00   85.00
summary(MATERRORNUM)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.00    4.00    7.96    8.00  507.00
THF1DIF <- as.numeric(as.matrix(MATBOBINAS2$THF1DIFTOTAL))
THF1MED <- as.numeric(as.matrix(MATBOBINAS2$THF1MEDTOTAL))
THF3MED <- as.numeric(as.matrix(MATBOBINAS2$THF3MEDTOTAL))
THF5MED <- as.numeric(as.matrix(MATBOBINAS2$THF5MEDTOTAL))
# Dibujamos el scatter-plots
ERRORMEDTOTALABS <- MATERRORNUM

# Eliminamos las Velocidades Espúreos
INDVEL <- ERRORMEDTOTALABS<100 & THF1DIF<30 & THF3DIFTOTAL<30 & THF5DIFTOTAL<30
& STEP1>0 & STEP2>0

MAT <- as.matrix(cbind(THF1MED[INDVEL], THF3MED[INDVEL], THF5MED[INDVEL],
THF1DIF[INDVEL],
THF3DIFTOTAL[INDVEL],THF5DIFTOTAL[INDVEL],STEP1[INDVEL],STEP2[INDVEL],ERRORMEDTO
TALABS[INDVEL]))
colnames(MAT) <-
c("THF1MED","THF3MED","THF5MED","THF1DIF","THF3DIF","THF5DIF","STEP1","STEP2","E
RRORMEDABS")

pairs(MAT, lower.panel=panel.smooth, upper.panel=panel.cor,diag.panel=
panel.hist)
```

Figura 246. Programa que dibuja los gráficos de box-plots de las variables a estudiar.

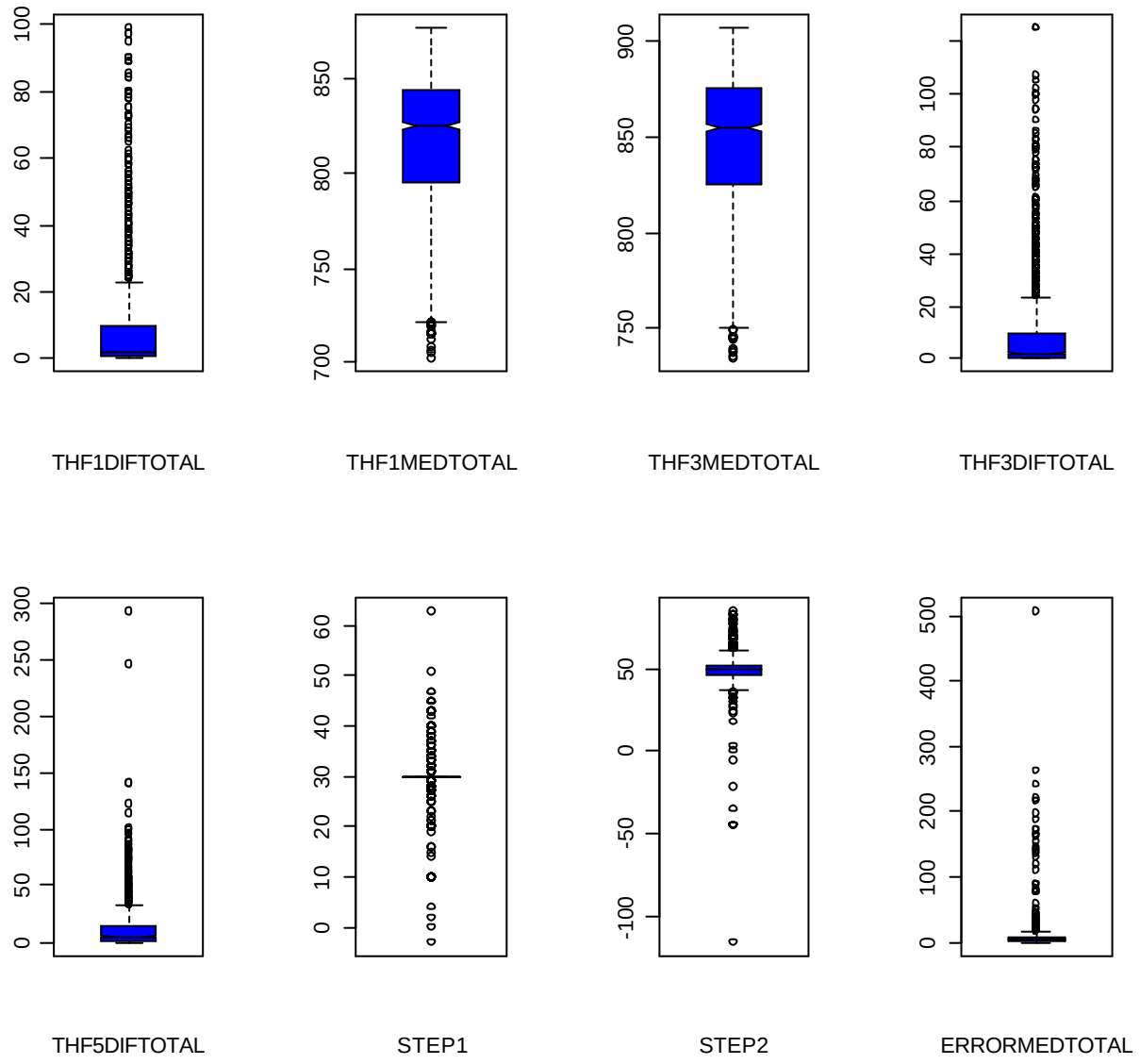


Figura 247. Gráfico de box-plots de las variables estudiadas.

De la Figura 247 podemos observar que la diferencia entre los valores máximo y mínimo de las temperaturas de consigna, es decir la variación de la temperatura de las zonas 1, 3 y 5 para cada bobina, puede llegar hasta los 60ºC.

Vemos también, que el *STEP1* se mantiene prácticamente constante en casi todas las bobinas a 30ºC, esto verifica la información suministrada por personal de la empresa. También el *STEP2* se mantiene entre el rango de los 50º, aunque éste varía un poco más entre los 46ºC y 52ºC (rango intercuartil).

Por otro lado, observamos en las variables THFxDIFTOTAL, un número elevado de bobinas en donde la diferencia entre temperaturas de consigna se mueve en un rango de 20ºC a valores mayores de los 100ºC.

Continuamos observando el gráfico de dispersión de las variables analizadas.

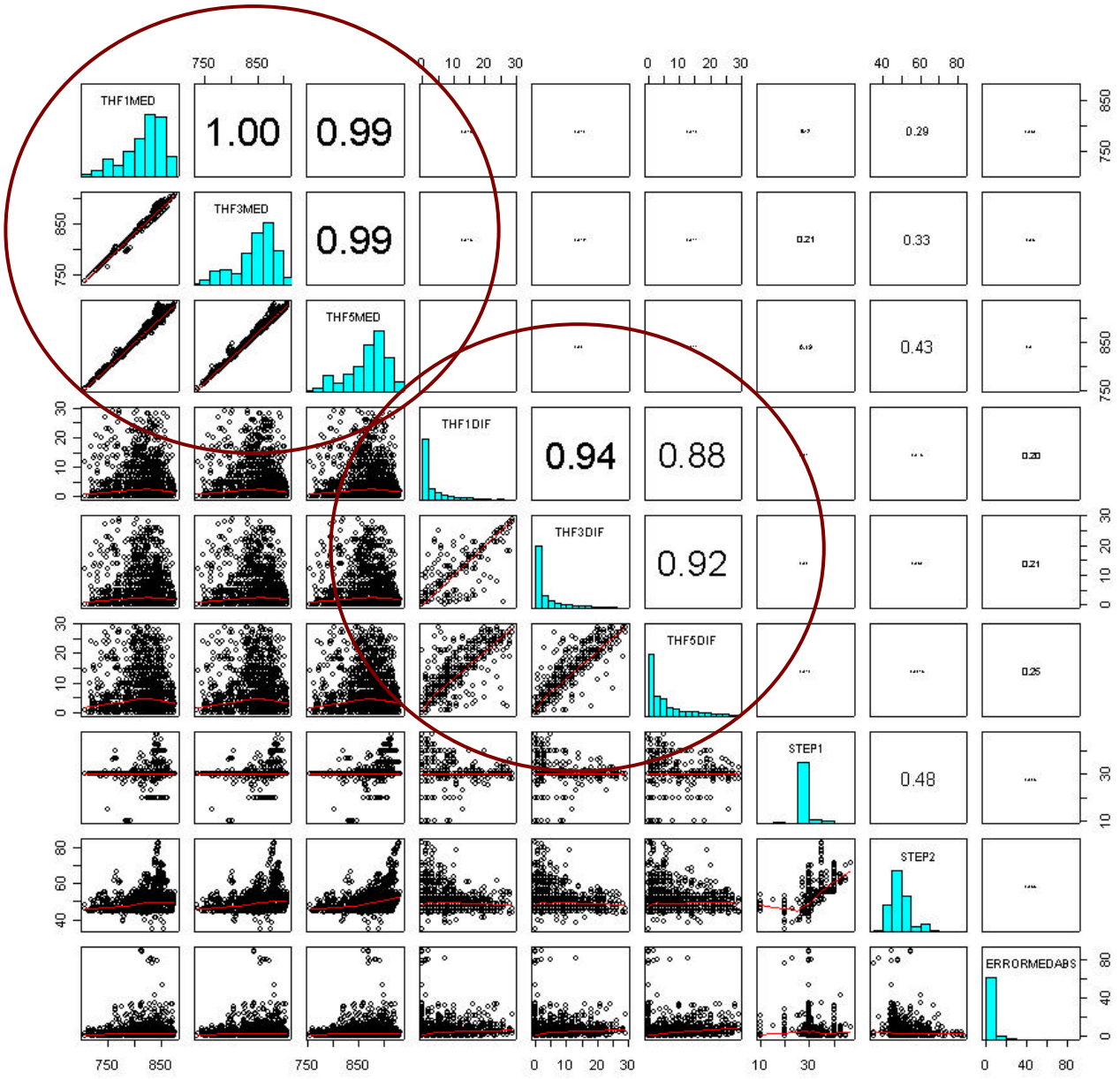


Figura 248. Gráfico de dispersión de las variables estudiadas.

En la Figura 248 podemos observar que, lógicamente, las variables *THFxMEDTOTAL y THFxDIFTOTAL* están bastante correladas entre sí. **Esto corrobora claramente que el sistema que genera las consignas solamente actúa en la *THF1* y las temperatura de consigna de las zonas 3 y 5 son la mismas que la de la zona 1 más los *STEPS*.**

Otro aspecto más significativo es el de la relación que existe entre las variables *STEP1* y *STEP2* con la variable *THF1MEDTOTAL*. Observando la distribución y correlación de estas variables, vemos claramente que los valores de los *STEPS* se mantienen más o menos constantes mientras la temperatura de consigna *THF1MEDTOTAL* se mueve entre los 700ºC y 820ºC

aproximadamente, pero que se mueven en un rango mayor cuando esta temperatura sobrepasa esos 820ºC.

Por otro lado, que queda claro, es que **"el error" (*ERRORMEDTOTALABS*) no tiene ninguna correlación lineal significativa con las demás variables.**

De todas formas, intentamos buscar alguna relación no lineal en la distribución de las bobinas con errores elevados dentro de la gráfica *sammon*.

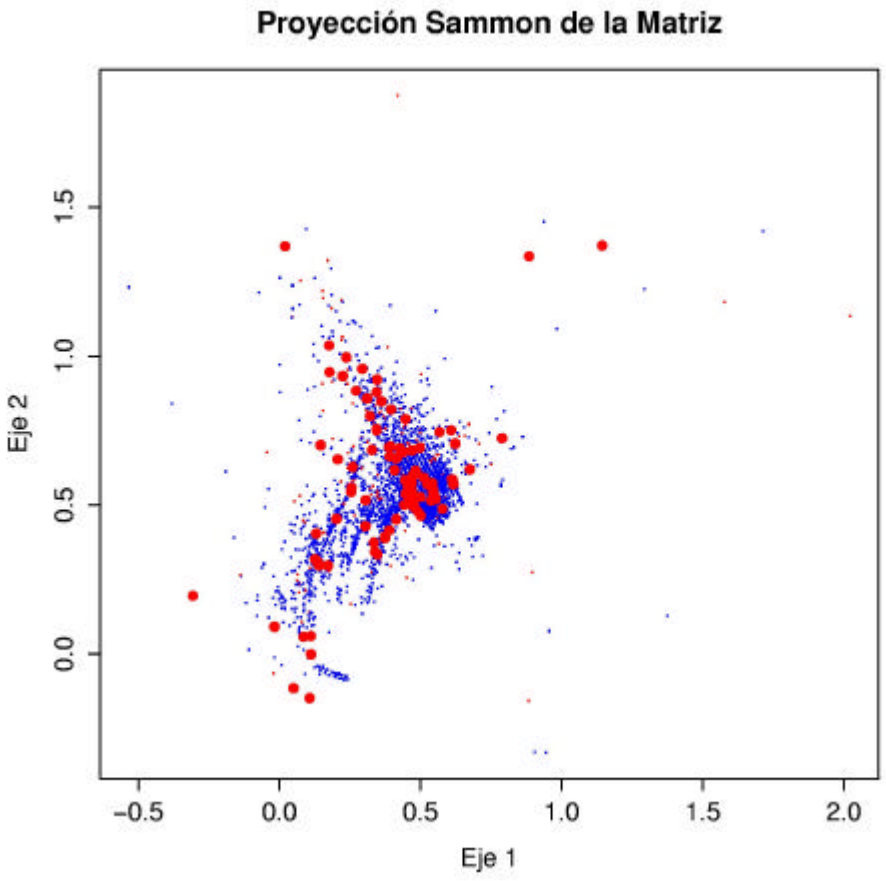


Figura 249. Diagrama sammon de las variables de consigna de temperatura de zonas del horno enfrentadas con el tipo de bobina. Los puntos grandes muestran los errores mayores de 40ºC.

En la Figura 249 podemos apreciar claramente, que la distribución de las bobinas con errores elevados "cubre" todo el espacio de las demás bobinas y que, por lo tanto, **en las variables estudiadas no subyace una estructura interna que pueda "explicar" las bobinas con "errores" elevados frente a las demás.**

6.2.3.1 CONCLUSIONES DEL ESTUDIO DE LA RELACIÓN ENTRE EL ERROR DE LA TEMPERATURA MEDIDA DE LAS BOBINAS Y LAS TEMPERATURAS DE CONSIGNA DE LAS ZONAS DEL HORNO

Después del análisis realizado, podemos llegar a las siguientes conclusiones:

- Como se ha visto, las variables estudiadas tienen una correlación lineal prácticamente nula con respecto al "error" de las bobinas. Además la gráfica *sammon* ha mostrado una falta de estructura en estas variables que pueda "ayudar" a separar las bobinas con "errores" elevados de las demás. Por lo tanto, podemos concluir en este punto, que **ni las temperaturas de las zonas del horno (*THF1MEDTOTAL*), ni las diferencias de variaciones de temperatura de las mismos (*THFxDIFTOTAL*), ni los "saltos térmicos" entre unas zonas y otras de la zona de calentamiento del horno (*STEPx*) son causa directa de las diferencias producidas entre la temperatura esperada de la banda y la real (*ERRORMEDTOTALABS*) según los resultados obtenidos.**
- También se ha visto claramente que la correlación entre las consignas de las zonas 1, 3 y 5 es prácticamente 1. Por lo tanto, **solamente se considerará la consigna de la zona 1 como variable a utilizar.**

6.2.4 ESTUDIO DE LA RELACIÓN ENTRE EL ERROR DE LA TEMPERATURA MEDIDA DE LA BANDA A LA SALIDA DE LA ZONA DE CALENTAMIENTO DEL HORNO Y LA TEMPERATURA DE LA BANDA A LA ENTRADA DEL MISMO

Por último analizaremos el comportamiento del "error" con otras variables. En este caso se decide estudiar si existe alguna relación entre la temperatura de la banda en la entrada y el "error" de la temperatura a la salida.

Las variables que van a entrar en el estudio son:

- *TMPP1MEDTOTAL*: Temperatura media de la banda leída por el pirómetro 1 en cada bobina.
- *TMPP2MEDTOTAL*: Temperatura media de la banda leída por el pirómetro 2 en cada bobina.
- *TMPP1DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de la banda leídas por el pirómetro 1 para cada bobina. Se utiliza para determinar el grado de estabilización de la curva de temperatura en ese punto.
- *TMPP2DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de la banda leídas por el pirómetro 2 para cada bobina. Se utiliza para determinar el grado de estabilización de la curva de temperatura en ese punto.
- *STEPPIRO*: Diferencia de temperaturas de la banda entre la temperatura de entrada de la banda al horno (pirómetro 1) y la temperatura de la banda a la salida de la zona de calentamiento del horno (pirómetro 2) (abs(*TMPP1MEDTOTAL-TMPP2MEDTOTAL)).*
- *ERRORMEDTOTAL*: Valor medio del valor absoluto de la diferencia entre la temperatura de consigna del pirómetro dos (temperatura esperada de la bobina a la salida de la zona de calentamiento) y la real.

Igual que en los puntos anteriores, procedemos a estudiar los gráficos de cajas y el *scatter plots.*

```
# Obtenemos las variables
TMPP1MED <- as.numeric(as.matrix(MATBOBINAS2$TMPP1MEDTOTAL))
TMPP1DIF <- as.numeric(as.matrix(MATBOBINAS2$TMPP1DIFTOTAL))
TMPP2MED <- as.numeric(as.matrix(MATBOBINAS2$TMPP2MEDTOTAL))
TMPP2DIF <- as.numeric(as.matrix(MATBOBINAS2$TMPP2DIFTOTAL))
STEPPIRO <- abs(TMPP2MED- TMPP1MED)

# Dibujamos el box-plots
par(mfrow=c(1,6),cex=1)
boxplot(TMPP1MED,col="blue",notch=TRUE, xlab="TMPP1MED")
boxplot(TMPP2MED,col="blue",notch=TRUE, xlab="TMPP2MED")
boxplot(TMPP1DIF,col="blue",notch=TRUE, xlab="TMPP1DIF")
boxplot(TMPP2DIF,col="blue",notch=TRUE, xlab="TMPP2DIF")
boxplot(STEPPIRO,col="blue",notch=TRUE, xlab="STEPPIRO")
boxplot(MATERRORNUM,col="blue",notch=TRUE, xlab="ERRORMEDTOTAL")

# Eliminamos las Velocidades Espúreos
INDVEL <- ERRORMEDTOTALABS<100 & TMPP1MED>100 & TMPP1MED>100
MAT <- as.matrix(cbind(TMPP1MED[INDVEL], TMPP1DIF[INDVEL], TMPP2MED[INDVEL],
TMPP2DIF[INDVEL], STEPPIRO[INDVEL], ERRORMEDTOTALABS[INDVEL]))
colnames(MAT) <-
c("TMPP1MED","TMPP1DIF","TMPP2MED","TMPP2DIF","STEPPIRO","ERRORMEDABS")

pairs(MAT, lower.panel=panel.smooth, upper.panel=panel.cor,diag.panel=
panel.hist)

# Vemos un resumen de los datos
summary(TMPP1MED)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  200.0   234.0   250.0   249.5   264.5   300.0
summary(TMPP2MED)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  715.0   792.0   821.0   807.6   826.0   867.0
summary(TMPP1DIF)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   4.000   7.000   8.962  11.000 155.000
summary(TMPP2DIF)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    3.50    8.00   13.58   18.00  130.00
summary(STEPPIRO)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  478.0   542.0   561.0   558.1   575.0   630.0
summary(ERRORMEDTOTALABS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.00    4.00    7.96    8.00  507.00
```

Figura 250. Programa que dibuja los gráficos box-plots *y el gráfico de* scatter-plots *de las variables estudiadas.*

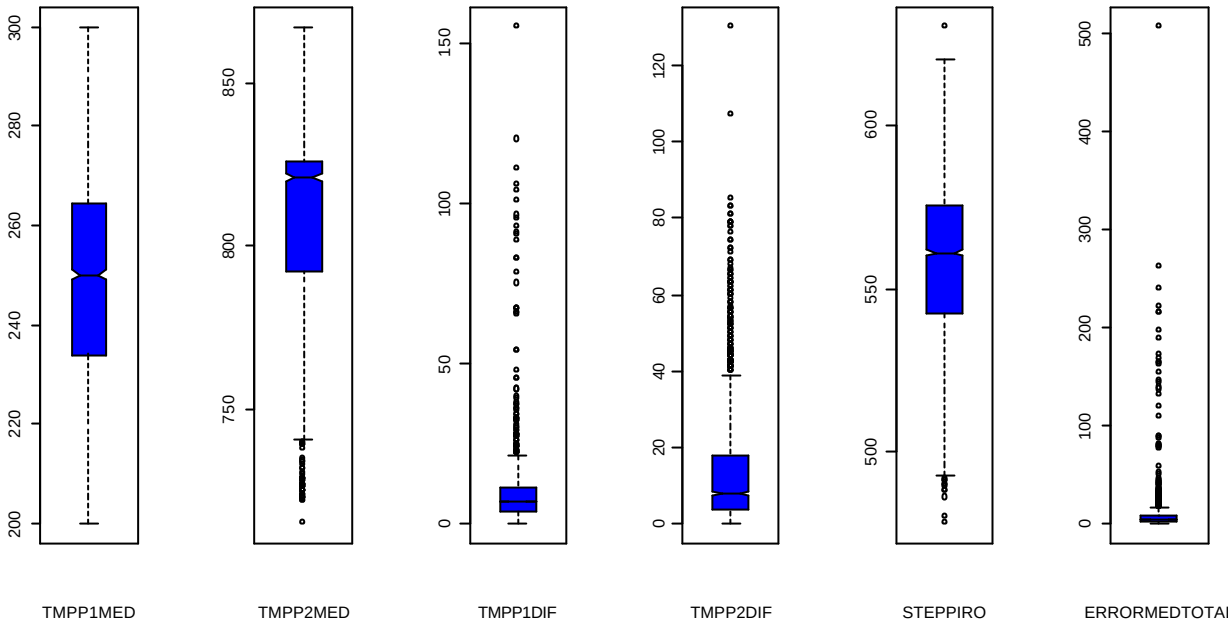


Figura 251. Gráficos box-plots de las variables TMPP1DIFTOTAL, TMPP2DIFTOTAL, TMPP1MEDTOTAL, TMPP2MEDTOTAL, STEPPIRO y ERRORMEDTOTAL.

En éstos se observa claramente, que existen algunas bobinas con diferencias notables de temperatura tanto en el pirómetro 1, como en el 2. Así, vemos diferencias mayores de 100ºC en algunas bobinas (ver Figura 251).

Además, la gráfica de cajas nos muestra el rango de [234ºC, 265ºC] de la temperatura de entrada de la banda y un rango intercuartil de la temperatura de salida de la banda de [792ºC, 826ºC].

Observamos que la variable *STEPPIRO*, que corresponde a la diferencia absoluta de la temperatura media de entrada de la banda en la zona de calentamiento del horno y la de salida, se mueve en un rango de [542ºC, 575ºC].

Por último, se advierte que la variación de las temperaturas de entrada y de salida para cada bobina (variables *TMPP1DIF* y *TMPP2DIF*) es grande, lo que indica que existen bobinas donde la temperatura oscila gravemente.

También se muestra el gráfico de *scatter-plot* de las variables estudiadas.

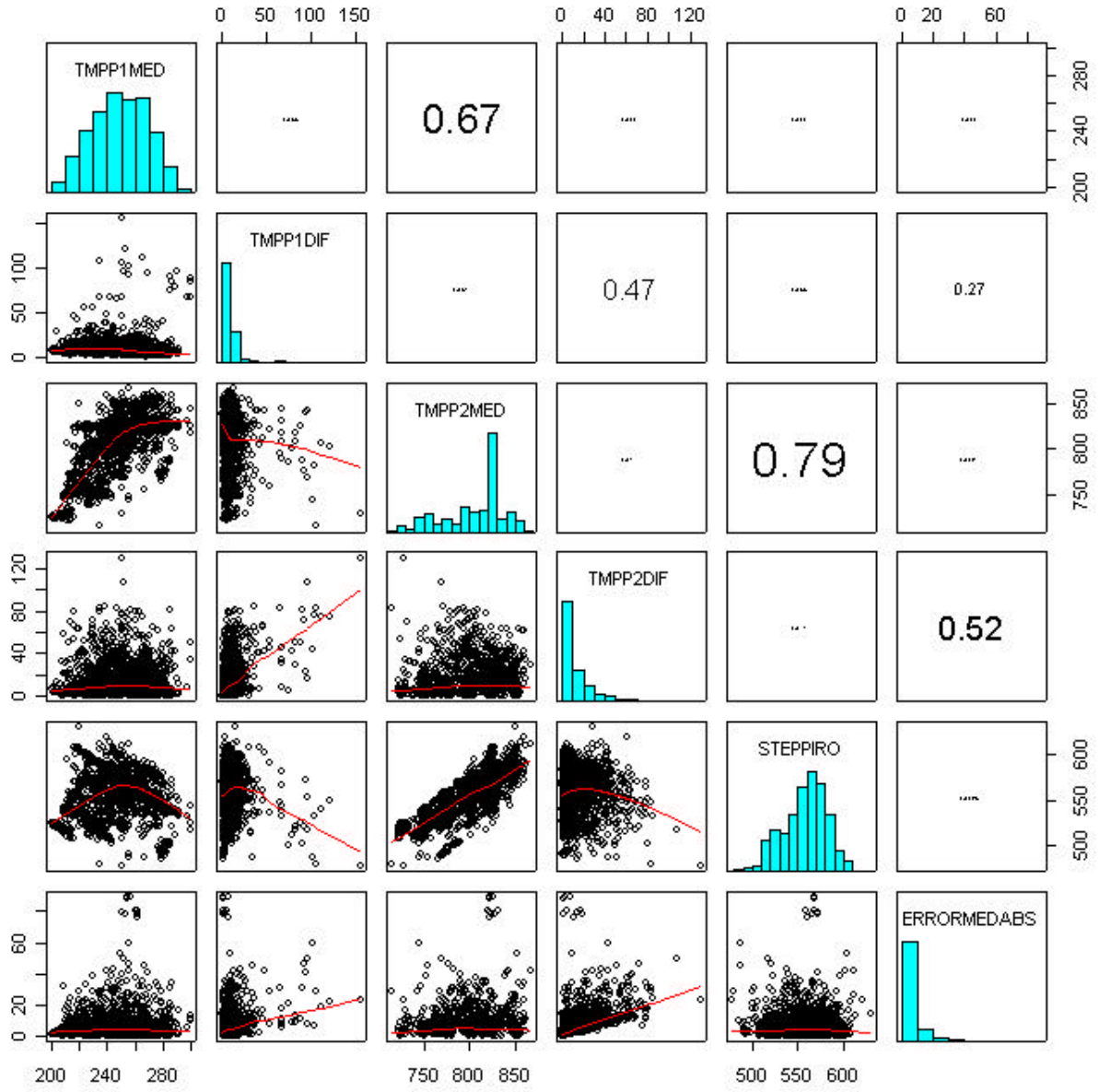


Figura 252. Scatter plot de las variables analizadas.

Igual que hemos hecho en puntos anteriores, realizamos el gráfico *sammon* de las variables analizadas separando las bobinas con "errores" elevados de las demás.

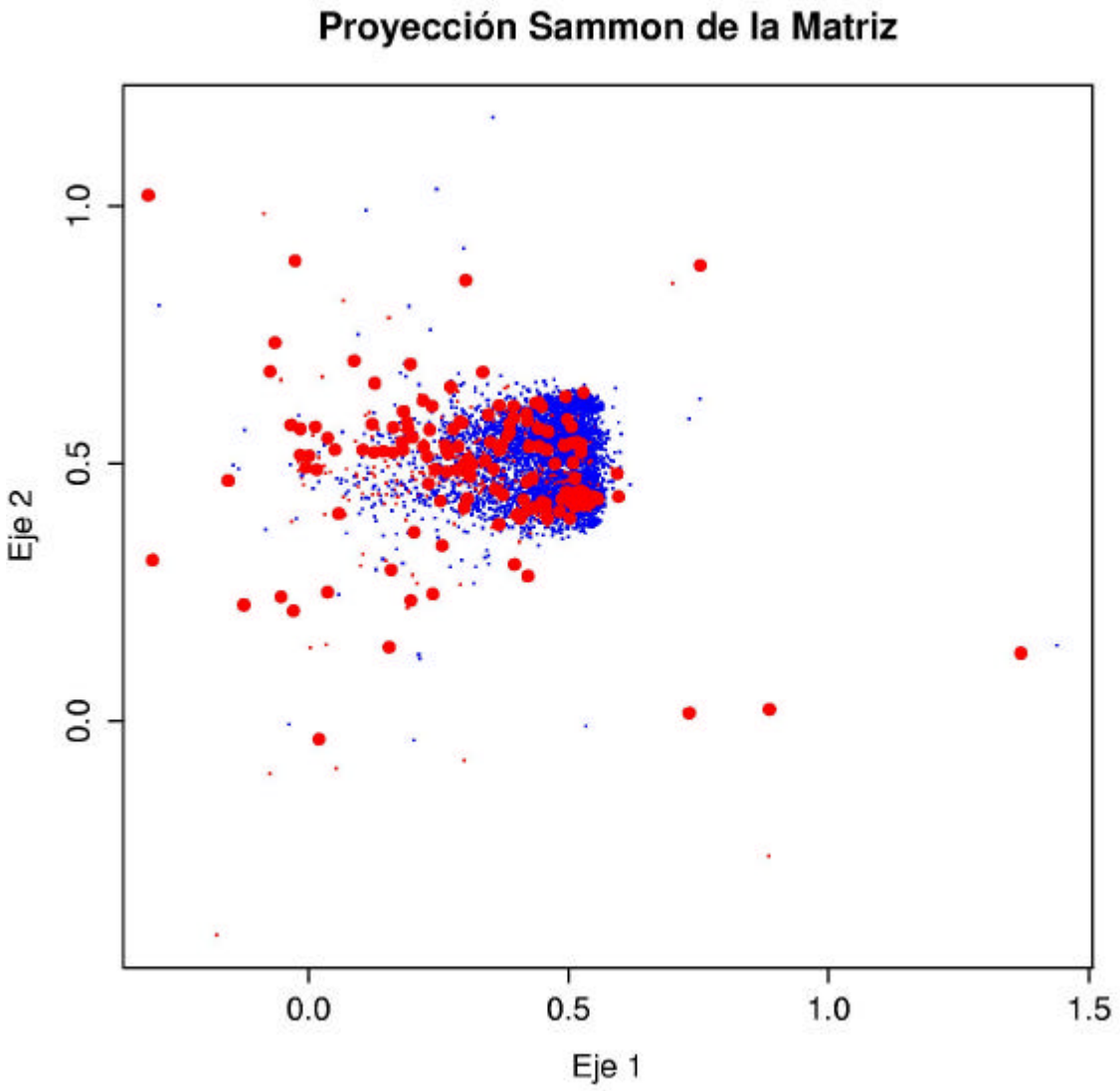


Figura 253. Diagrama sammon de las variables estudiadas. En puntos gruesos las bobinas con error elevado.

6.2.4.1 CONCLUSIONES DEL ESTUDIO DE LA RELACIÓN ENTRE EL ERROR DE LA TEMPERATURA MEDIDA DE LAS BOBINAS A LA SALIDA DEL HORNO Y LA TEMPERATURA MEDIDA A LA ENTRADA DEL HORNO

En el *scatter-plot* de la Figura 252 vemos una pequeña correlación de la variable *TMPP1DIFTOTAL* con el "error" (*ERRORMEDTOTAL)*. Aunque esta correlación es bastante pequeña (0,27) si que parece significativa si la comparamos con la correlación de la variable *TMPP2DIFTOTAL* (0,52) ya que ésta si que está directamente implicada en el cálculo del "error"[20] y no es muy elevada. Esto indica **que las variaciones bruscas de temperatura en la banda a la entrada del horno, influyen en el "error" final.** Sí se ve en la Figura 253 que existe una pequeña distribución de las bobinas "erróneas" y que tienden a moverse hacia la izquierda con respecto a las demás, probablemente debido a estas correlaciones. Aún así, esta gráfica no es muy significativa ya que se están utilizando algunas variables que son completamente dependientes de la variable a estudiar.

Las demás correlaciones que se advierten son bastante lógicas. Por un lado, vemos que la variable *TMPP2MEDTOTAL* y el incremento de temperatura de la banda entre el pirómetro 1 y el 2 (*STEPPIRO*) son muy dependientes entre sí.

También se advierte que las variables *TMPP1DIFTOTAL* y *TMPP2DIFTOTAL* tienen una cierta correlación, lo que indica que **esas "variaciones" de temperatura de la banda a la entrada del horno, no son completamente absorbidas por el salto térmico y perduran, en cierta forma, hasta la salida de la zona de calentamiento donde está situado el pirómetro 2.**

Por último, de la observación de la Figura 252 podemos concluir que el "error" no depende ni de la temperatura con la que entra la banda al horno ni del incremento de temperaturas al que se ve sometida la misma. Este detalle **permite descartar posibles deficiencias en la "capacidad" del horno para calentar algunas bobinas y llevarlas a una temperatura de consigna buscada e invita a pensar que estos "errores" viene más por problemas dinámicos de cambios bruscos de consigna entre bobinas que por problemas estáticos en la capacidad para transmitir la energía calorífica necesaria a la banda.**

[20] Realmente, cabe recordar que "el error" (variable *ERRORMEDTOTALABS*) se calcula a partir de la media del valor absoluto de la diferencia de la temperatura del pirómetro 2 (*TMMP2MEDTOTAL*) menos la temperatura de consigna. Por lo tanto, como es lógico, la diferencia entre la temperatura máxima y mínima de éste pirómetro por bobina influye bastante en el "error" final, ya que está indicando la variación de la variable *ERRORMEDTOTAL* con respecto a la de consigna.

6.2.5 CONCLUSIONES FINALES DEL ESTUDIO DE DEPENDENCIAS

De los análisis anteriores se pueden resumir las siguientes conclusiones finales:

- **Parece que el grado de eficiencia del "modo automático" es mayor que el "modo manual".** De los 10 tipos de aceros estudiados, 9 presentan un menor error en las bobinas tratadas en "modo automático" que en el "modo manual", aunque este último modo puede verse afectado porque se utiliza cuando hay que resolver contingencias o procesar bobinas difíciles de tratar.

	MODO MANUAL		MODO AUTOMATICO	
Tipo de Error	Núm.	En %	Núm.	En %.
BAJO <=20ºC	1024	93,1%	521	98,3%
MEDIO (>20ºC & <=50ºC)	52	4,7%	7	1,3%
ALTO (>50ºC)	24	2,2%	2	0,4%

Tabla 59. Porcentajes globales de tipos de error medio absoluto para los dos modos.

- **El tipo de acero de cada bobina y el modo en que se ha tratado cada una de ellas, deben ser considerados en los procesos posteriores de modelizado**. Será conveniente, buscar una relación de agrupamiento entre los diferentes tipos de bobinas.

% ERRORABSMANUAL	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	4,3%	10,1%	4,5%	8,4%	4,4%	21,5%	1,5%	0,0%	5,7%	8,0%
BAJO	95,7%	89,9%	95,5%	91,6%	95,6%	78,5%	98,5%	100,0%	94,3%	92,0%

Tabla 60. Comparación con los tipos de error para el modo manual (en porcentajes relativos).

% ERRORABSAUTOMAT	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	0,0%	1,7%	0,0%	1,1%	2,4%	0,0%	1,1%	0,0%	12,5%	0,0%
BAJO	100,0%	98,3%	100,0%	98,9%	97,6%	100,0%	98,9%	100,0%	87,5%	100,0%

Tabla 61. Comparación con los tipos de error para el modo automático (en porcentajes relativos).

- **Los comportamientos dinámicos de las variables de temperatura del horno, de la banda y de la velocidad de la misma, son los que parecen ser causantes de los "errores elevados".** La derivada de las variables parece ser la mejor elección para futuros estudios.

- **Las dimensiones de la banda NO presentan correlación lineal con el "error".** Las diferentes dimensiones (ESPESOR, ANCHURA y LONGITUD) son muy dependientes entre sí debido a que proceden de paralelepípedos de fundición de iguales dimensiones, por lo tanto, solo se usará una de ellas.
- **La velocidad de la banda, lógicamente, presenta una correlación significativa con las dimensiones de la misma pero no con el error.**
- **Los saltos térmicos entre unas zonas y otras, y las temperaturas de las zonas de la parte de calentamiento del horno, parecen ser adecuados siempre que la variables mantenga un régimen permanente en cada bobina.**
- **Las variaciones de temperatura de la bobina a la entrada, son mantenidas, en cierto modo, a la salida. Esto indica que el salto térmico, no es capaz de absorber algunos cambios bruscos de temperatura de la banda.**
- **El horno tiene capacidad para "llevar" a todas las bobinas a la temperatura de consigna buscada.**

6.3 BÚSQUEDA DE CONOCIMIENTO MEDIANTE TÉCNICAS DE MINERÍA DE DATOS

6.3.1 USO DE CLASIFICADORES Y REGLAS DE ASOCIACIÓN

Siguiendo con el estudio de los datos, podemos aplicar la herramienta *WEKA* [WEK02] que nos permitirá utilizar diferentes herramientas de DM para la búsqueda de conocimiento de la base de datos.

Lo primero que hacemos, es generar una matriz con las variables a estudiar y convertirla a un archivo de texto con extensión '*.arff*' tal y como nos lo exige el programa.

En los análisis anteriores, hemos visto la relación que existe entre las variables de la base de datos y la variable independiente a estudiar, el "error". Por lo tanto, de los estudios realizados en el punto anterior, se seleccionarán las variables que cumplan estas dos condiciones:

- Sean independientes entre sí.
- Tengan una cierta influencia en la variable "error".

Éstas son:

- *CLASACERO:* Clase de acero de cada bobina.
- *DUREZA*: Del acero.
- *ANCHO:* Anchura de la bobina. Debido a que influirá en la superficie por metro lineal de banda.
- *ESPENT*: Espesor a la entrada del horno. Aunque está bastante correlada con la anchura, es conveniente disponer de ella porque puede influir en el coeficiente de emisividad de la banda.
- *MODOBOB*: Modo de trabajo de esa bobina (manual o automático).
- *VELDIFTOTAL*: Diferencia entre velocidad mínima y máxima de banda para cada bobina. Ya que valores elevados de la misma, indican cambios de velocidad en la banda y esto puede llevar a errores en la distribución de la temperatura de cada bobina.
- *THF1DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de consigna de la subzona 1 para cada bobina. Se utiliza para determinar el grado de estabilización de la curva de temperatura en ese punto.
- *THF1MEDTOTAL*: Temperatura de consigna de la subzona 1. Aunque inicialmente no tiene relación con respecto a la bobina, se selecciona para determinar si puede clasificar los diferentes errores de las bobinas.

- *TMPP1DIFTOTAL*: Diferencia entre temperaturas mínima y máxima de la banda leídas por el pirómetro 1 para cada bobina. Se utiliza para determinar el grado de estabilización de la curva de temperatura en ese punto. Se selecciona porque **variaciones bruscas de temperatura en la banda a la entrada del horno, influyen en el "error" final.**
- *TMPP1MEDTOTAL*: Temperatura media de la banda leída por el pirómetro 1 en cada bobina.
- *ERRORMEDTOTALABS*: Como valor de la variable a explicar y que es la media del valor absoluto de la diferencia entre la temperatura de consigna del pirómetro dos (temperatura esperada de la bobina a la salida de la zona de calentamiento) y la real.

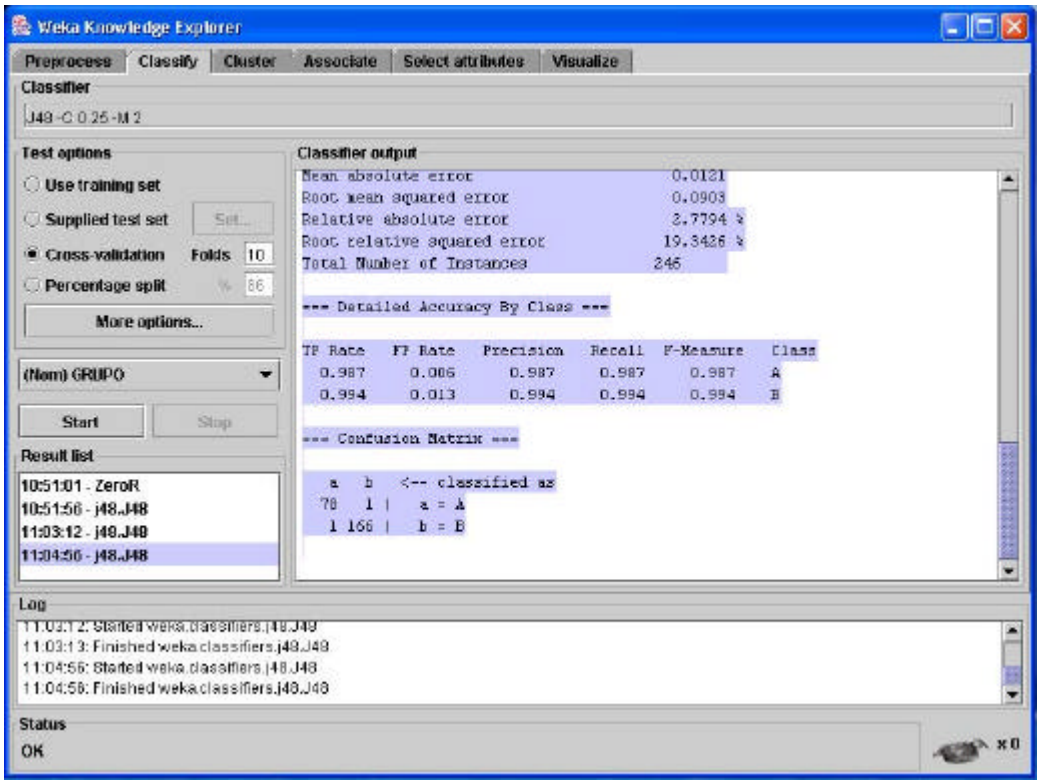


Figura 254. Entorno de los clasificadores el programa WEKA.

También seleccionamos las variables categóricas que indican los tipos de curvas de los parámetros más importantes:

- *TIPOCURVATHF1*: Tipo de curva de la subzona 1.
- *TIPOCURVATMPP1*: Tipo de curva del pirómetro 1.
- *TIPOCURVATMPP2CNG*: Tipo de curva de la temperatura de consigna del pirómetro.
- *TIPOCURVAVEL*: Tipo de curva de la velocidad de la banda.
- *TIPOCURVAERROR*: Tipo de curva del error. Es la variable a explicar.

Aunque simplificamos el número de categorías de cada una de estas variables, para simplificar los resultados.

Para ello, creamos una matriz con todas las variables y creamos un archivo de texto.

```
# Creamos las matrices

# Extraemos los datos de las bobinas más usuales
BOBINASESTUDIAR <- c("B011F97", "B012F53", "B025F55", "B100F55", "B102G33",
"B102G55", "B105F55", "C107G55", "C114G55", "K011F57")

DATBOBINASESTU <- DATBOBINAS[DATBOBINAS$CLASACERO %in% BOBINASESTUDIAR,]
MATBOBINASESTU <- MATBOBINAS[DATBOBINAS$CLASACERO %in% BOBINASESTUDIAR,]

# Simplificamos los tipos de grupos
MATBOBINASESTU$TIPOCURVATHF1[1:2]
[1] H H
Levels:  AC AD BC BD H MC MD

# Variables tipo curva GRUPOTIPOTHF1
ORDTHC1 <- as.numeric(MATBOBINASESTU$TIPOCURVATHF1) #< 30ºC
GRUPOTIPOTHF1 <- rep("H", length(ORDTHC1))
GRUPOTIPOTHF1[ORDTHC1<=2] <- "A" # >=60ºC
GRUPOTIPOTHF1[ORDTHC1>=6] <- "M" # >=30 ºC y <60ºC

# Simplificamos los tipos de grupos GRUPOTIPOTMPP1
MATBOBINASESTU$TIPOCURVATMPP1[1:2]
[1] H H
Levels:  ACVAC ACXAC ACXAD AOMAXMIN AOMINMAX ARC ARD BRC BRD H MCVAC MCVAD MCXAC
MCXAD MOMAXMIN MOMINMAX MRC MRD
ORDTMPP1 <- as.numeric(MATBOBINASESTU$ TIPOCURVATMPP1) #<= 20ºC
GRUPOTIPOTMPP1 <- rep("H", length(ORDTMPP1))
GRUPOTIPOTMPP1 [ORDTMPP1 <=7] <- "A" # >40ºC
GRUPOTIPOTMPP1 [ORDTMPP1 >=11] <- "M" # >20 ºC y <60ºC
```

```
# Simplificamos los tipos de grupos GRUPOTIPOTMPP2CNG
MATBOBINASESTU$TIPOCURVATMPP2CNG[1:2]
[1] H H
Levels:  ACVAD ACXAC AOMAXMIN ARC ARD BRC BRD H MCVAC MCVAD MCXAC MCXAD MRC MRD



ORDTMPP2CNG <- as.numeric(MATBOBINASESTU$TIPOCURVATMPP2CNG) #<= 20ºC
GRUPOTIPOTMPP2CNG <- rep("H", length(ORDTMPP2CNG))
GRUPOTIPOTMPP2CNG [ORDTMPP2CNG <=5] <- "A" # >40ºC
GRUPOTIPOTMPP2CNG [ORDTMPP2CNG >=9] <- "M" # >20 ºC y <60ºC

# Simplificamos los tipos de grupos GRUPOTIPOVEL
MATBOBINASESTU$TIPOCURVAVEL[1:2]
[1] H H
Levels:  ACVAC ACVAD ACXAD AOMINMAX ARC ARD BRC BRD H MCVAC MCVAD MCXAC MCXAD
MOMAXMIN MRC MRD

ORDVEL <- as.numeric(MATBOBINASESTU$TIPOCURVAVEL) #<= 20 m/min
GRUPOTIPOVEL <- rep("H", length(ORDVEL))
GRUPOTIPOVEL [ORDVEL <=6] <- "A" # >50 m/min
GRUPOTIPOVEL [ORDVEL >=10] <- "M" # >20 m/min y <=50 m/min


# Simplificamos los tipos de errores
GRUPOERROR <- rep("BAJO", length(MATBOBINASESTU$ERRORMEDTOTALABS)) #Errores
menores de 30ºC
GRUPOERROR[as.numeric(as.matrix(MATBOBINASESTU$ERRORMEDTOTALABS)) >=30] <-
"ALTO" #Errores >=30 de media absoluta

# Creamos la Matriz
MATJ2003 <- data.frame(DATBOBINASESTU$CODBOBINA, DATBOBINASESTU$CLASACERO,
DATBOBINASESTU$DUREZA, DATBOBINASESTU$ANCHO, DATBOBINASESTU$ESPENT,
MATBOBINASESTU$MODOBOB, MATBOBINASESTU$VELDIFTOTAL, MATBOBINASESTU$THF1DIFTOTAL,
MATBOBINASESTU$THF1MEDTOTAL, MATBOBINASESTU$TMPP1DIFTOTAL,
MATBOBINASESTU$TMPP1DIFTOTAL, MATBOBINASESTU$ERRORMEDTOTALABS, GRUPOTIPOTHF1,
GRUPOTIPOTMPP1, GRUPOTIPOTMPP2CNG, GRUPOTIPOVEL, GRUPOERROR)

# Pasamos la matriz 'MATJ48' a un archivo de texto
#write.table(MATJ2003[MATJ2003$GRUPOERROR=="ALTO",],"C:\\temp\\DATDOC\\MATJ2003_
ALTO.txt",quote=FALSE,sep=",",row.names=FALSE,col.names=FALSE)
write.table(MATJ2003,"C:\\temp\\DATDOC\\MATJ2003_ALTO.txt",quote=FALSE,sep=",",r
ow.names=FALSE,col.names=FALSE)
```

Figura 255. Programa que genera una nueva base de datos para poder ser analizada con el software WEKA.

Un ejemplo de la matriz resultante se muestra en la figura siguiente.

```
@relation MATJ2003

@attribute CODBOBINA real
@attribute CLASACERO
{B011B99,B011F97,B012B97,B012F53,B012F55,B013B55,B013C55,B014F53,B014F55,B016F35
,B017F53,B023H53,B025F55,B032H53,B042H53,B044H53,B081B99,B085F97,B085G99,B100B95
,B100F33,B100F55,B101F55,B102G33,B102G55,B103G33,B103G55,B105F55,B120G55,C107G55
,C114G55,C115G55,C116G55,D012F55,D012G99,D031B33,D032F55,D071F55,D094B33,D094G55
,K011B55,K011F57,K021H43,K021H53,K022H53,N013H53,N017B97,X100G99}
@attribute DUREZA {11,13,14,15,16,17,19,20,24,29,30,32,37,50,E1,E8,F8,G0,G4,NA}
@attribute ANCHO real
@attribute ESPENT real
@attribute MODOBOB {-1,-0.5,0}
@attribute VELDIFTOTAL real
@attribute THF1DIFTOTAL real
@attribute THF1MEDTOTAL real
@attribute TMPP1DIFTOTAL real
@attribute TMPP1DIFTOTAL real
@attribute ERRORMEDTOTALABS real
@attribute GRUPOTIPOTHF1 {A,H,M}
@attribute GRUPOTIPOTMPP1 {A,H,M}
@attribute GRUPOTIPOTMPP2CNG {A,H,M}
@attribute GRUPOTIPOVEL {A,H,M}
@attribute GRUPOERROR {ALTO,BAJO}

@data
23293006,B011F97,17,1250,0.583,0,1,3,770,4,4,4,H,H,H,H,ALTO
23293007,B011F97,17,1250,0.583,0,0,9,772,8,8,2,H,H,H,H,BAJO
23293008,B011F97,17,1250,0.583,0,61,32,778,8,8,9,M,H,H,A,ALTO
23293009,B011F97,17,1250,0.583,0,0,73,758,7,7,7,A,H,H,H,ALTO
23293010,B011F97,17,1250,0.583,0,35,73,752,6,6,4,A,H,H,M,ALTO
23293011,B011F97,17,1250,0.583,0,20,20,766,12,12,4,H,H,H,H,ALTO
23293012,B011F97,17,1250,0.583,0,33,41,799,14,14,11,M,H,A,M,BAJO
23293013,B100F55,50,1350,0.675,0,8,3,812,12,12,5,H,H,H,H,ALTO
23293014,B100F55,50,1350,0.675,0,10,7,808,3,3,5,H,H,H,H,ALTO
23293015,B100F55,50,1350,0.675,-1,0,1,804,5,5,1,H,H,H,H,BAJO
23293016,B100F55,50,1350,0.675,-1,0,2,803,2,2,1,H,H,H,H,BAJO
23293017,B100F55,50,1350,0.675,-1,10,4,800,3,3,4,H,H,H,H,ALTO
23293018,B100F55,50,1350,0.675,-1,0,2,802,2,2,2,H,H,H,H,BAJO
23293020,B100F55,50,1350,0.675,0,31,18,798,3,3,9,H,H,M,M,ALTO
23293021,B011F97,17,1250,0.583,0,30,24,782,18,18,8,H,H,A,M,ALTO
23293022,B011F97,17,1250,0.583,0,10,2,779,20,20,5,H,H,H,H,ALTO
...
```

Figura 256. Ejemplo de la matriz creada para analizarla con el programa WEKA.

Si aplicamos el algoritmo de reglas asociativas, obtenemos algunas reglas que pueden ser interesantes.

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.1
-S -1.0
Relation:     MATJ2003-weka.filters.AttributeFilter-V-R2,6,13-17
Instances:    1632
Attributes:   7
              CLASACERO
              MODOBOB
              GRUPOTIPOTHF1
              GRUPOTIPOTMPP1
              GRUPOTIPOTMPP2CNG
              GRUPOTIPOVEL
              GRUPOERROR
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.75
Minimum metric <confidence>: 0.7
Number of cycles performed: 5

Generated sets of large itemsets:

Size of set of large itemsets L(1): 5

Size of set of large itemsets L(2): 10

Size of set of large itemsets L(3): 10

Size of set of large itemsets L(4): 2

Best rules found:

  1. GRUPOTIPOTMPP2CNG=H GRUPOTIPOVEL=H 1281 ==> GRUPOERROR=BAJO 1253
conf:(0.98)
  2. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 1321 ==> GRUPOTIPOTMPP1=H 1292
conf:(0.98)
  3. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H GRUPOERROR=BAJO 1290 ==> GRUPOTIPOTMPP1=H
1261    conf:(0.98)
  4. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 1321 ==> GRUPOERROR=BAJO 1290    conf:(0.98)
  5. GRUPOTIPOVEL=H 1378 ==> GRUPOERROR=BAJO 1345    conf:(0.98)
  6. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 1292 ==> GRUPOERROR=BAJO
1261    conf:(0.98)
  7. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 1333 ==> GRUPOERROR=BAJO 1301
conf:(0.98)
  8. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 1425 ==> GRUPOERROR=BAJO 1388
conf:(0.97)
  9. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 1382 ==> GRUPOERROR=BAJO 1346
conf:(0.97)
 10. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 1343 ==>
GRUPOERROR=BAJO 1308    conf:(0.97)
 11. GRUPOTIPOTHF1=H 1486 ==> GRUPOERROR=BAJO 1445    conf:(0.97)
 12. GRUPOTIPOTHF1=H 1486 ==> GRUPOTIPOTMPP1=H 1445    conf:(0.97)
```

```
 13. GRUPOTIPOTMPP1=H 1558 ==> GRUPOERROR=BAJO 1515    conf:(0.97)
 14. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 1445 ==> GRUPOERROR=BAJO 1405
conf:(0.97)
 15. GRUPOTIPOTHF1=H GRUPOERROR=BAJO 1445 ==> GRUPOTIPOTMPP1=H 1405
conf:(0.97)
 16. GRUPOTIPOTMPP2CNG=H 1494 ==> GRUPOERROR=BAJO 1452    conf:(0.97)
 17. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 1382 ==> GRUPOTIPOTMPP1=H 1343
conf:(0.97)
 18. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H GRUPOERROR=BAJO 1346 ==>
GRUPOTIPOTMPP1=H 1308    conf:(0.97)
 19. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H GRUPOERROR=BAJO 1301 ==> GRUPOTIPOTHF1=H
1261    conf:(0.97)
 20. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 1333 ==> GRUPOTIPOTHF1=H 1292
conf:(0.97)
 21. GRUPOTIPOTMPP2CNG=H GRUPOTIPOVEL=H 1281 ==> GRUPOTIPOTHF1=H 1241
conf:(0.97)
 22. GRUPOTIPOTMPP2CNG=H GRUPOTIPOVEL=H 1281 ==> GRUPOTIPOTMPP1=H 1240
conf:(0.97)
 23. GRUPOTIPOVEL=H 1378 ==> GRUPOTIPOTMPP1=H 1333    conf:(0.97)
 24. GRUPOTIPOVEL=H GRUPOERROR=BAJO 1345 ==> GRUPOTIPOTMPP1=H 1301
conf:(0.97)
 25. GRUPOTIPOVEL=H GRUPOERROR=BAJO 1345 ==> GRUPOTIPOTHF1=H 1290    conf:(0.96)
 26. GRUPOTIPOVEL=H 1378 ==> GRUPOTIPOTHF1=H 1321    conf:(0.96)
 27. GRUPOERROR=BAJO 1583 ==> GRUPOTIPOTMPP1=H 1515    conf:(0.96)
 28. GRUPOTIPOTMPP2CNG=H GRUPOERROR=BAJO 1452 ==> GRUPOTIPOTMPP1=H 1388
conf:(0.96)
 29. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 1321 ==> GRUPOTIPOTMPP1=H GRUPOERROR=BAJO
1261    conf:(0.95)
 30. GRUPOTIPOTMPP2CNG=H 1494 ==> GRUPOTIPOTMPP1=H 1425    conf:(0.95)
 31. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 1382 ==> GRUPOTIPOTMPP1=H
GRUPOERROR=BAJO 1308    conf:(0.95)
 32. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 1333 ==> GRUPOTIPOTHF1=H GRUPOERROR=BAJO
1261    conf:(0.95)
 33. GRUPOTIPOTHF1=H 1486 ==> GRUPOTIPOTMPP1=H GRUPOERROR=BAJO 1405
conf:(0.95)
 34. GRUPOTIPOVEL=H 1378 ==> GRUPOTIPOTMPP1=H GRUPOERROR=BAJO 1301
conf:(0.94)
 35. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 1425 ==> GRUPOTIPOTHF1=H 1343
conf:(0.94)
 36. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H GRUPOERROR=BAJO 1388 ==>
GRUPOTIPOTHF1=H 1308    conf:(0.94)
 37. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 1321 ==> GRUPOTIPOTMPP2CNG=H 1241
conf:(0.94)
 38. GRUPOTIPOVEL=H 1378 ==> GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 1292
conf:(0.94)
 39. GRUPOTIPOVEL=H GRUPOERROR=BAJO 1345 ==> GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H
1261    conf:(0.94)
 40. GRUPOTIPOVEL=H 1378 ==> GRUPOTIPOTHF1=H GRUPOERROR=BAJO 1290    conf:(0.94)
 41. GRUPOTIPOVEL=H GRUPOERROR=BAJO 1345 ==> GRUPOTIPOTMPP2CNG=H 1253
conf:(0.93)
 42. GRUPOTIPOTHF1=H GRUPOERROR=BAJO 1445 ==> GRUPOTIPOTMPP2CNG=H 1346
conf:(0.93)
 43. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H GRUPOERROR=BAJO 1405 ==>
GRUPOTIPOTMPP2CNG=H 1308    conf:(0.93)
...
```

Figura 257. Reglas asociativas obtenidas de analizar las variables categóricas.

Las reglas asociativas [AGR94] nos indican el grado de asociación entre las variables existentes según el análisis de coincidencias en la base de datos. Por ejemplo, la regla número 1 de la Figura 257 que dice que

"1. GRUPOTIPOTMPP2CNG=H GRUPOTIPOVEL=H 1281 ==> GRUPOERROR=BAJO 1253 conf:(0.98)"

y nos indica que:

> *"Cuando el curva de consigna de temperatura de pirómetro 2 es horizontal y la curva de velocidad es horizontal, existe un 98% de probabilidades de que el tipo de curva de error sea BAJO."*

La mayoría de las reglas obtenidas NO aportan mucha información, ya que lógicamente se refieren a que cuando las curvas de temperaturas de consigna y de velocidad son constantes, el error suele ser bajo.

Más interesante, resulta estudiar simplemente las bobinas con errores mayores de 30ºC que son, en total, 49. **De éstas, 46 son en "modo manual" y 3 en "automático".**

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 100 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.1
-S -1.0
Relation:     MATJ2003-weka.filters.AttributeFilter-V-R2-3,6,13-16
Instances:    49
Attributes:   7
              CLASACERO
              DUREZA
              MODOBOB
              GRUPOTIPOTHF1
              GRUPOTIPOTMPP1
              GRUPOTIPOTMPP2CNG
              GRUPOTIPOVEL
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.6
Minimum metric <confidence>: 0.7
Number of cycles performed: 8

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 13

Size of set of large itemsets L(3): 13

Size of set of large itemsets L(4): 6
```

```
Size of set of large itemsets L(5): 1

Best rules found:

  1. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 31 ==> GRUPOTIPOTMPP1=H 31    conf:(1)
  2. MODOBOB=0 GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 29 ==> GRUPOTIPOTMPP1=H 29
conf:(1)
  3. GRUPOTIPOTHF1=H 41 ==> GRUPOTIPOTMPP1=H 40    conf:(0.98)
  4. MODOBOB=0 GRUPOTIPOTHF1=H 39 ==> GRUPOTIPOTMPP1=H 38    conf:(0.97)
  5. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 36 ==> GRUPOTIPOTMPP1=H 35
conf:(0.97)
  6. MODOBOB=0 GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 34 ==> GRUPOTIPOTMPP1=H 33
conf:(0.97)
  7. MODOBOB=0 GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 34 ==> GRUPOTIPOTHF1=H 33
conf:(0.97)
  8. DUREZA=50 GRUPOTIPOTMPP1=H 34 ==> GRUPOTIPOTMPP2CNG=H 33    conf:(0.97)
  9. DUREZA=50 GRUPOTIPOTHF1=H 33 ==> GRUPOTIPOTMPP2CNG=H 32    conf:(0.97)
 10. DUREZA=50 GRUPOTIPOTHF1=H 33 ==> GRUPOTIPOTMPP1=H 32    conf:(0.97)
 11. GRUPOTIPOVEL=H 33 ==> GRUPOTIPOTMPP1=H 32    conf:(0.97)
 12. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 32 ==> GRUPOTIPOTMPP2CNG=H 31
conf:(0.97)
 13. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 32 ==> GRUPOTIPOTMPP1=H 31
conf:(0.97)
 14. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 32 ==> GRUPOTIPOTHF1=H 31    conf:(0.97)
 15. DUREZA=50 MODOBOB=0 GRUPOTIPOTMPP1=H 31 ==> GRUPOTIPOTMPP2CNG=H 30
conf:(0.97)
 16. DUREZA=50 MODOBOB=0 GRUPOTIPOTHF1=H 31 ==> GRUPOTIPOTMPP2CNG=H 30
conf:(0.97)
 17. DUREZA=50 MODOBOB=0 GRUPOTIPOTHF1=H 31 ==> GRUPOTIPOTMPP1=H 30
conf:(0.97)
 18. DUREZA=50 MODOBOB=0 GRUPOTIPOTMPP1=H 31 ==> GRUPOTIPOTHF1=H 30
conf:(0.97)
 19. MODOBOB=0 GRUPOTIPOVEL=H 31 ==> GRUPOTIPOTMPP1=H 30    conf:(0.97)
 20. DUREZA=50 MODOBOB=0 GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 30 ==>
GRUPOTIPOTMPP2CNG=H 29    conf:(0.97)
 21. DUREZA=50 MODOBOB=0 GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 30 ==>
GRUPOTIPOTMPP1=H 29    conf:(0.97)
 22. DUREZA=50 MODOBOB=0 GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 30 ==>
GRUPOTIPOTHF1=H 29    conf:(0.97)
 23. MODOBOB=0 GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 30 ==> GRUPOTIPOTHF1=H 29
conf:(0.97)
 24. GRUPOTIPOTHF1=H 41 ==> MODOBOB=0 39    conf:(0.95)
 25. MODOBOB=0 GRUPOTIPOTMPP1=H 40 ==> GRUPOTIPOTHF1=H 38    conf:(0.95)
 26. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 40 ==> MODOBOB=0 38    conf:(0.95)
 27. DUREZA=50 39 ==> GRUPOTIPOTMPP2CNG=H 37    conf:(0.95)
 28. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 37 ==> GRUPOTIPOTHF1=H 35
conf:(0.95)
 29. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 36 ==> MODOBOB=0 34    conf:(0.94)
 30. DUREZA=50 MODOBOB=0 36 ==> GRUPOTIPOTMPP2CNG=H 34    conf:(0.94)
 31. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 35 ==> MODOBOB=0 33
conf:(0.94)
 32. DUREZA=50 GRUPOTIPOTMPP1=H 34 ==> GRUPOTIPOTHF1=H 32    conf:(0.94)
 33. DUREZA=50 GRUPOTIPOTHF1=H 33 ==> GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 31
conf:(0.94)
 34. DUREZA=50 GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 33 ==> GRUPOTIPOTHF1=H 31
conf:(0.94)
 35. GRUPOTIPOVEL=H 33 ==> GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 31    conf:(0.94)
 36. DUREZA=50 GRUPOTIPOTHF1=H 33 ==> MODOBOB=0 31    conf:(0.94)
 37. GRUPOTIPOVEL=H 33 ==> GRUPOTIPOTHF1=H 31    conf:(0.94)
```

```
 38. GRUPOTIPOVEL=H 33 ==> MODOBOB=0 31    conf:(0.94)
 39. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 32 ==> MODOBOB=0 30
conf:(0.94)
 40. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 32 ==> MODOBOB=0 30
conf:(0.94)
 41. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 32 ==> MODOBOB=0 30    conf:(0.94)
 42. DUREZA=50 MODOBOB=0 GRUPOTIPOTHF1=H 31 ==> GRUPOTIPOTMPP1=H
GRUPOTIPOTMPP2CNG=H 29    conf:(0.94)
 43. DUREZA=50 MODOBOB=0 GRUPOTIPOTMPP1=H 31 ==> GRUPOTIPOTHF1=H
GRUPOTIPOTMPP2CNG=H 29    conf:(0.94)
 44. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 31 ==>
MODOBOB=0 29    conf:(0.94)
 45. MODOBOB=0 GRUPOTIPOVEL=H 31 ==> GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 29
conf:(0.94)
 46. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 31 ==> MODOBOB=0 GRUPOTIPOTMPP1=H 29
conf:(0.94)
 47. GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 31 ==> MODOBOB=0 29
conf:(0.94)
 48. MODOBOB=0 GRUPOTIPOVEL=H 31 ==> GRUPOTIPOTHF1=H 29    conf:(0.94)
 49. GRUPOTIPOTHF1=H GRUPOTIPOVEL=H 31 ==> MODOBOB=0 29    conf:(0.94)
 50. GRUPOTIPOTMPP1=H 43 ==> GRUPOTIPOTHF1=H 40    conf:(0.93)
 51. GRUPOTIPOTMPP1=H 43 ==> MODOBOB=0 40    conf:(0.93)
 52. GRUPOTIPOTMPP2CNG=H 42 ==> MODOBOB=0 39    conf:(0.93)
 53. GRUPOTIPOTHF1=H 41 ==> MODOBOB=0 GRUPOTIPOTMPP1=H 38    conf:(0.93)
 54. DUREZA=50 39 ==> MODOBOB=0 36    conf:(0.92)
 55. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 37 ==> MODOBOB=0 34    conf:(0.92)
 56. DUREZA=50 GRUPOTIPOTMPP2CNG=H 37 ==> MODOBOB=0 34    conf:(0.92)
 57. GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 36 ==> MODOBOB=0 GRUPOTIPOTMPP1=H 33
conf:(0.92)
 58. DUREZA=50 GRUPOTIPOTMPP1=H 34 ==> GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 31
conf:(0.91)
 59. DUREZA=50 GRUPOTIPOTMPP1=H 34 ==> MODOBOB=0 31    conf:(0.91)
 60. DUREZA=50 GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 33 ==> MODOBOB=0 30
conf:(0.91)
 61. DUREZA=50 GRUPOTIPOTHF1=H 33 ==> MODOBOB=0 GRUPOTIPOTMPP2CNG=H 30
conf:(0.91)
 62. DUREZA=50 GRUPOTIPOTHF1=H 33 ==> MODOBOB=0 GRUPOTIPOTMPP1=H 30
conf:(0.91)
 63. GRUPOTIPOVEL=H 33 ==> MODOBOB=0 GRUPOTIPOTMPP1=H 30    conf:(0.91)
 64. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP1=H 32 ==> MODOBOB=0
GRUPOTIPOTMPP2CNG=H 29    conf:(0.91)
 65. DUREZA=50 GRUPOTIPOTHF1=H GRUPOTIPOTMPP2CNG=H 32 ==> MODOBOB=0
GRUPOTIPOTMPP1=H 29    conf:(0.91)
 66. GRUPOTIPOTMPP1=H GRUPOTIPOVEL=H 32 ==> MODOBOB=0 GRUPOTIPOTHF1=H 29
conf:(0.91)
 67. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 37 ==> MODOBOB=0 GRUPOTIPOTHF1=H 33
conf:(0.89)
 68. DUREZA=50 GRUPOTIPOTMPP2CNG=H 37 ==> GRUPOTIPOTMPP1=H 33    conf:(0.89)
 69. GRUPOTIPOTMPP1=H GRUPOTIPOTMPP2CNG=H 37 ==> DUREZA=50 33    conf:(0.89)
 ...
```

Figura 258. Reglas asociativas para los errores ALTOS.

En la mayoría de las reglas que aparece el *MODOBOB*=0, es decir, en "modo manual", se indican que existe una gran cantidad de curvas de temperatura de consigan de horno, de pirómetro dos y de velocidad HORIZONTALES. Es decir, bobinas que están en régimen permanente con casi todas sus curvas horizontales, pero que su error es ALTO.

Esto parece indicar que los errores altos en "el modo manual" son bastante frecuentes porque no se reacciona a tiempo en el manejo de las curvas de velocidades y temperatura de consigna de zonas del horno para reducir el error entre la medida de temperatura del pirómetro 2 y la de consigna.

Se observa que, resulta conveniente estudiar el error de una bobina con respecto a las curvas de las bobinas anteriores. Para ello, se obtienen unas variables con el comportamiento de las bobinas anteriores y alguna variable que indique si las bobinas están seguidas o no.

```
# Cargamos las matrices con los datos de las 2.628 bobinas
library(RODBC)
canal <- odbcConnect("aceralia2003","","","localhost");

# Obtenemos el orden de las bobinas iniciales
DATBOBINASX <- sqlQuery(canal, "SELECT CODBOBINA FROM dps")

# Detectamos las bobinas repetidas
table(DATBOBINASX$CODBOBINA)

# Obtenemos una lista de las bobinas sin repeticiones
LISTABOBINAS <- unique(DATBOBINASX$CODBOBINA)

# Creamos el orden de bobinas
ORDENBOBINAS <- (1:length(LISTABOBINAS))

# Determinamos cuales aparecen finalmente
EXISTEN <- ORDENBOBINAS[match(DATBOBINASESTU$CODBOBINA,LISTABOBINAS)]

# OBTENEMOS UNA LISTA DE BOBINAS SEGUIDAS (Seguidas=1 otro valor >1)
SALTO <- EXISTEN[2:length(EXISTEN)]-EXISTEN[1:length(EXISTEN)-1]

# OBTENEMOS UNA LISTA DONDE HAY CAMBIO DE CLASE DE ACERO
CAMBACERO <- DATBOBINASESTU$CLASACERO[2:length(DATBOBINASESTU$CLASACERO)] !=
DATBOBINASESTU$CLASACERO[1:length(DATBOBINASESTU$CLASACERO)-1]

# OBTENEMOS UNA LISTA DONDE HAY CAMBIO DE ANCHO DE ACERO
CAMBANCH <- DATBOBINASESTU$ANCHO[2:length(DATBOBINASESTU$ANCHO)] !=
DATBOBINASESTU$ANCHO[1:length(DATBOBINASESTU$ANCHO)-1]

CAMBESP <- DATBOBINASESTU$ESPENT[2:length(DATBOBINASESTU$ESPENT)] !=
DATBOBINASESTU$ESPENT[1:length(DATBOBINASESTU$ESPENT)-1]


# OBTENEMOS LOS DATOS DE LAS VARIABLES ANTERIORES
CLASACEROANTES <-  DATBOBINASESTU$CLASACERO[1:length(DATBOBINASESTU$CLASACERO)-
1]
MODOBOBANTES <- MATBOBINASESTU$MODOBOB[1:length(MATBOBINASESTU$MODOBOB)-1]
```

```
GRUPOTIPOTHF1ANTES <-  GRUPOTIPOTHF1[1:length(GRUPOTIPOTHF1)-1]
GRUPOTIPOTMPP1ANTES <-  GRUPOTIPOTMPP1[1:length(GRUPOTIPOTMPP1)-1]
GRUPOTIPOTMPP2CNGANTES <-  GRUPOTIPOTMPP2CNG[1:length(GRUPOTIPOTMPP2CNG)-1]
GRUPOTIPOVELANTES <-  GRUPOTIPOVEL[1:length(GRUPOTIPOVEL)-1]
GRUPOERRORANTES <-  GRUPOERROR[1:length(GRUPOERROR)-1]
CLASACEROAHORA <-  DATBOBINASESTU$CLASACERO[2:length(DATBOBINASESTU$CLASACERO)]
MODOBOBAHORA <- MATBOBINASESTU$MODOBOB[2:length(MATBOBINASESTU$MODOBOB)]
GRUPOTIPOTHF1AHORA <-  GRUPOTIPOTHF1[2:length(GRUPOTIPOTHF1)]
GRUPOTIPOTMPP1AHORA <-  GRUPOTIPOTMPP1[2:length(GRUPOTIPOTMPP1)]
GRUPOTIPOTMPP2CNGAHORA <-  GRUPOTIPOTMPP2CNG[2:length(GRUPOTIPOTMPP2CNG)]
GRUPOTIPOVELAHORA <-  GRUPOTIPOVEL[2:length(GRUPOTIPOVEL)]
GRUPOERRORAHORA <-  GRUPOERROR[2:length(GRUPOERROR)]

# Creamos la Matriz con variables categóricas de bobinas actuales y antes
MATJ2003 <- data.frame(SALTO, CAMBACERO, CLASACEROANTES, MODOBOBANTES,
GRUPOTIPOTHF1ANTES, GRUPOTIPOTMPP1ANTES, GRUPOTIPOTMPP2CNGANTES,
GRUPOTIPOVELANTES, GRUPOERRORANTES, CLASACEROAHORA, MODOBOBAHORA,
GRUPOTIPOTHF1AHORA, GRUPOTIPOTMPP1AHORA, GRUPOTIPOTMPP2CNGAHORA,
GRUPOTIPOVELAHORA, GRUPOERRORAHORA, CAMBANCH, CAMBESP)

# Pasamos la matriz a un archivo de texto solo bobinas consecutivas
# write.table(MATJ2003[MATJ2003$SALTO==1,],
"C:\\temp\\MATJ2003_CONSECUTIVAS.txt",quote=FALSE,sep=",",row.names=FALSE,col.na
mes=FALSE)
write.table(MATJ2003[MATJ2003$SALTO==1 &
MATJ2003$GRUPOERRORAHORA=="ALTO",],"C:\\temp\\MATJ2003_CONSECUTIVAS_ALTOS.txt",q
uote=FALSE,sep=",",row.names=FALSE,col.names=FALSE)
```

Figura 259. Programa que genera una nueva base de datos con bobinas anteriores y actuales consecutivas para poder ser analizada con el software WEKA.

Primero analizamos si el cambio de clase de acero genera errores ALTOS.

```
table(MATJ2003$CAMBACERO,MATJ2003$GRUPOERRORAHORA)
       ALTO BAJO
  FALSE   38 1255
  TRUE    11  327
11/(38+11);327/(1255+327)
[1] 0.2244898
[1] 0.2067004

table(MATJ2003$CAMBANCH,MATJ2003$GRUPOERRORAHORA)
        ALTO BAJO
  FALSE   30 1220
  TRUE    19  362
19/(30+19);362/(1220+362)
[1] 0.3877551
[1] 0.2288243

table(MATJ2003$CAMBESP,MATJ2003$GRUPOERRORAHORA)
        ALTO BAJO
  FALSE   34 1278
  TRUE    15  304
15/(34+15);304/(1278+304)
[1] 0.3061224
[1] 0.1921618
```

Figura 260. Comparación entre el cambio de acero y el tipo de error de la bobina.

Vemos que en un 22,45% de las bobinas con errores altos aparece en un cambio de bobina, aunque también se produce en un 20,67% de las bobinas con errores bajos. Por lo tanto, podemos observar que la causa de cambio de acero, no afecta en demasía al error de la siguiente bobina, aunque puede afectar a las posteriores.

En cambio, se observa que el cambio de anchura afecta a un 38,77% de las bobinas con errores ALTOS, frente al 22,89% de errores BAJOS, y que también el cambio de espesor, un 30,6% de errores ALTOS frente al 19,2% de errores BAJOS. Es decir, **parece que los cambios de anchura o de espesor son alguna de las causas de los errores elevados.**

	TIPO DE ERROR	
SUCESO	ALTOS (>30ºC)	BAJOS (<=30ºC)
Cambio de Acero	22,4%	20,7%
Cambio de Anchura	38,8%	22,9%
Cambio de Espesor	30,6%	19,3%

Tabla 62. Porcentajes de errores ALTOS y BAJOS para cambios de acero, anchura o espesor de bobinas.

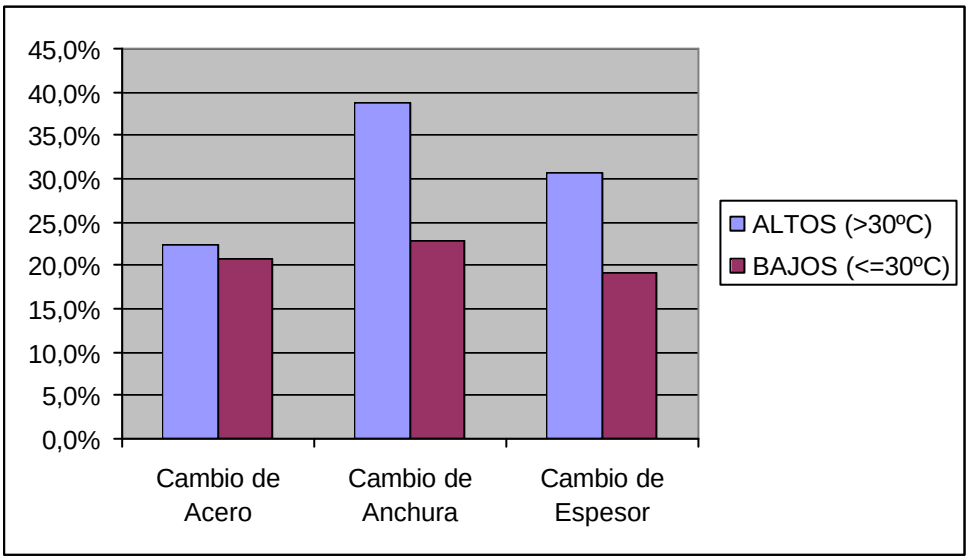


Figura 261. Gráfico de porcentajes de errores ALTOS y BAJOS para cambios de acero, anchura o espesor de bobinas.

Continuando con el estudio, realizamos múltiples pruebas generando diferentes reglas asociativas y clasificadores buscando obtener una relación que nos permita determinar qué bobinas son más incompatibles entre si.

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
-S -1.0
Relation:     MATJ2003-weka.filters.AttributeFilter-V-R2-15,17-18
Instances:    36
Attributes:   16
              CAMBACERO
              CLASACEROANTES
              MODOBOBANTES
              GRUPOTIPOTHF1ANTES
              GRUPOTIPOTMPP1ANTES
              GRUPOTIPOTMPP2CNGANTES
              GRUPOTIPOVELANTES
              GRUPOERRORANTES
              CLASACEROAHORA
              MODOBOBAHORA
              GRUPOTIPOTHF1AHORA
              GRUPOTIPOTMPP1AHORA
              GRUPOTIPOTMPP2CNGAHORA
              GRUPOTIPOVELAHORA
              CAMBANCH
              CAMBESP
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.8
Minimum metric <confidence>: 0.9
Number of cycles performed: 4

Generated sets of large itemsets:
Size of set of large itemsets L(1): 9
Size of set of large itemsets L(2): 19
Size of set of large itemsets L(3): 14
Size of set of large itemsets L(4): 3

Best rules found:

  1. MODOBOBANTES=0 32 ==> MODOBOBAHORA=0 32    conf:(1)
  2. GRUPOTIPOTHF1AHORA=H 31 ==> GRUPOTIPOTMPP1ANTES=H 31    conf:(1)
  3. GRUPOTIPOTMPP2CNGANTES=H 31 ==> CAMBACERO=FALSE 31    conf:(1)
  4. GRUPOTIPOTHF1AHORA=H GRUPOTIPOTMPP1AHORA=H 30 ==> GRUPOTIPOTMPP1ANTES=H 30
conf:(1)
  5. GRUPOTIPOTHF1ANTES=H 30 ==> GRUPOTIPOTMPP1ANTES=H GRUPOTIPOTHF1AHORA=H 30
conf:(1)
...
```

Figura 262. Reglas asociativas de las bobinas con errores ALTOS.

De la Figura 262, podemos obtener algunas conclusiones para las bobinas con errores ALTOS:

- La regla 1 indica que cuando el modo de la bobinas anterior es "Manual" en el 100% de los casos el modo actual es "Manual". Es decir, que la mayor parte de los errores se produce en "modo manual" y no en cambios de modo.
- Se corrobora que un alto porcentaje de errores ALTOS tienen curvas horizontales en casi todas sus temperaturas y velocidades de consigna.

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
-S -1.0
Relation:     MATJ2003-weka.filters.AttributeFilter-V-R2-4,10-11,17-18
Instances:    36
Attributes:   7
              CAMBACERO
              CLASACEROANTES
              MODOBOBANTES
              CLASACEROAHORA
              MODOBOBAHORA
              CAMBANCH
              CAMBESP
=== Associator model (full training set) ===
Apriori
=======

Minimum support: 0.5
Minimum metric <confidence>: 0.9
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L(1): 7
Size of set of large itemsets L(2): 17
Size of set of large itemsets L(3): 19
Size of set of large itemsets L(4): 10
Size of set of large itemsets L(5): 2

Best rules found:

  1. MODOBOBANTES=0 32 ==> MODOBOBAHORA=0 32    conf:(1)
  2. CAMBACERO=FALSE MODOBOBANTES=0 28 ==> MODOBOBAHORA=0 28    conf:(1)
  3. CAMBESP=FALSE 28 ==> CAMBACERO=FALSE 28    conf:(1)
  4. MODOBOBAHORA=0 CAMBESP=FALSE 25 ==> CAMBACERO=FALSE 25    conf:(1)
  5. MODOBOBANTES=0 CAMBESP=FALSE 24 ==> CAMBACERO=FALSE MODOBOBAHORA=0 24
conf:(1)
  6. CAMBACERO=FALSE MODOBOBANTES=0 CAMBESP=FALSE 24 ==> MODOBOBAHORA=0 24
conf:(1)
  7. MODOBOBANTES=0 MODOBOBAHORA=0 CAMBESP=FALSE 24 ==> CAMBACERO=FALSE 24
conf:(1)
  8. MODOBOBANTES=0 CAMBESP=FALSE 24 ==> MODOBOBAHORA=0 24    conf:(1)
  9. CAMBACERO=FALSE CAMBANCH=FALSE 24 ==> CAMBESP=FALSE 24    conf:(1)
 10. CAMBANCH=FALSE CAMBESP=FALSE 24 ==> CAMBACERO=FALSE 24    conf:(1)
 11. MODOBOBANTES=0 CAMBESP=FALSE 24 ==> CAMBACERO=FALSE 24    conf:(1)
```

Figura 263. Otras, reglas asociativas de las bobinas con errores ALTOS.

Analizando, la figura anterior, podemos observar en las reglas 5 y 10 que 24 bobinas de las 36 bobinas, es decir un 66,67% de las bobinas con errores ALTOS que:

"En "modo manual", un 66,67% de las bobinas con errores ALTOS, NO se producen por cambios de anchura, de espesor o tipo de acero de la bobina".

Lo que claramente indica, que un 67% de los errores altos pueden ser debidos a una falta de reajuste, en el "modo manual", de las bobinas que están en régimen permanente o con errores espontáneos.

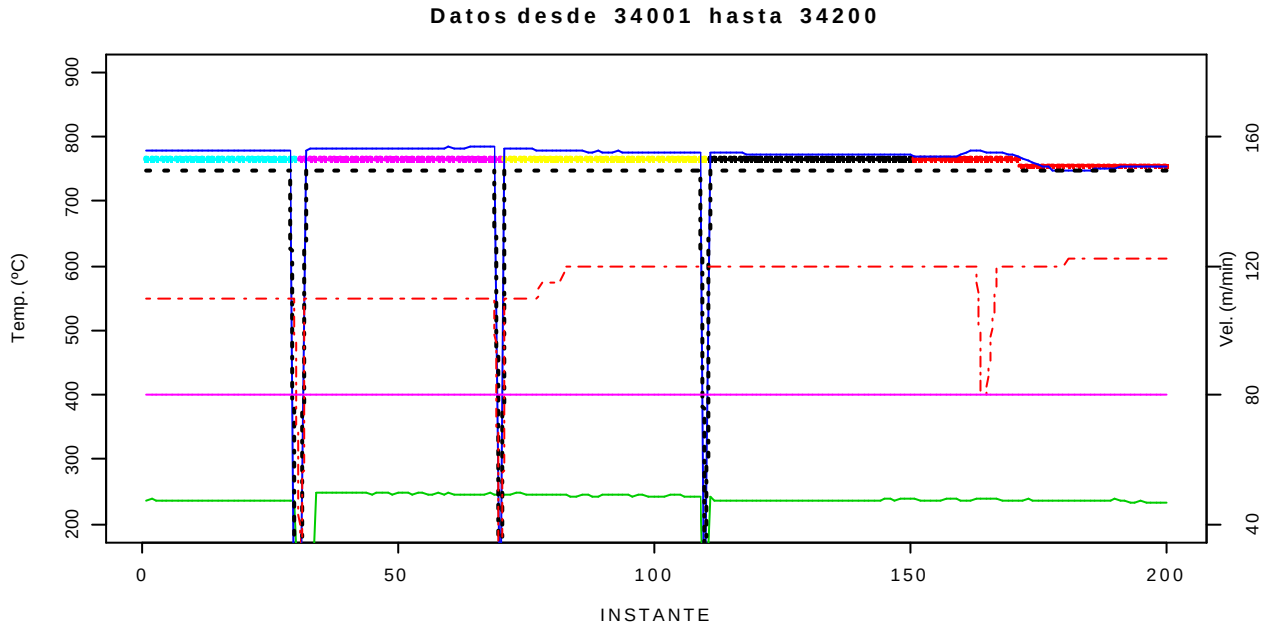


Figura 264. Curvas que explican ese 66% de errores ALTOS por una falta de reajuste en el "modo manual"

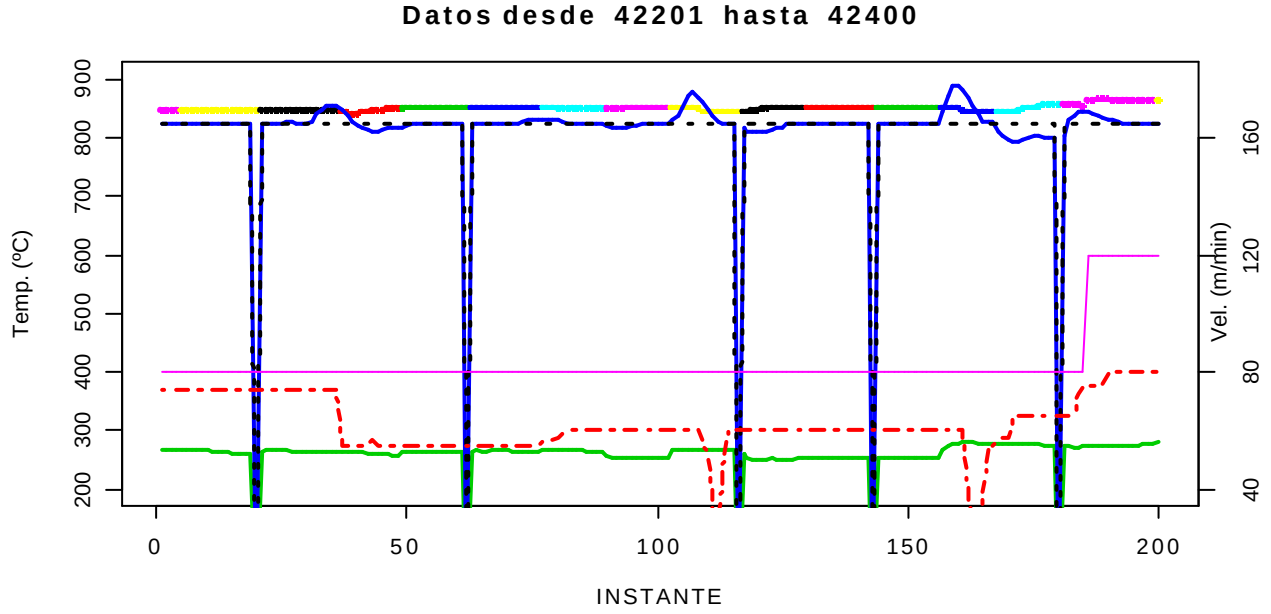


Figura 265. Curvas que explican ese 66% de errores ALTOS por una falta de reajuste en el "modo manual"

```
=== Run information ===

Scheme:       weka.classifiers.j48.PART -C 0.25 -M 2
Relation:     MATJ2003-weka.filters.AttributeFilter-V-R2-4,9-11,16-18
Instances:    115
Attributes:   9
              CAMBACERO
              CLASACEROANTES
              MODOBOBANTES
              GRUPOERRORANTES
              CLASACEROAHORA
              MODOBOBAHORA
              GRUPOERRORAHORA
              CAMBANCH
              CAMBESP
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===
PART decision list
------------------
GRUPOERRORANTES = BAJO AND
CAMBACERO = FALSE AND
CAMBANCH = FALSE: BAJO (72.0/11.0)

GRUPOERRORANTES = ALTO: ALTO (17.0)

CAMBACERO = TRUE: BAJO (17.0/1.0)

: ALTO (9.0/2.0)

Number of Rules  :      4


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          99               86.087  %
Incorrectly Classified Instances        16               13.913  %
Kappa statistic                          0.6382
Mean absolute error                      0.2259
Root mean squared error                  0.353
Relative absolute error                 52.3795 %
Root relative squared error             76.1192 %
Total Number of Instances              115
=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.583     0.013      0.955     0.583     0.724    ALTO
  0.987     0.417      0.839     0.987     0.907    BAJO

=== Confusion Matrix ===

  a  b   <-- classified as
 21 15 |  a = ALTO
  1 78 |  b = BAJO
```

Figura 266. Primeras reglas obtenidas para el cambio de bobina.

En la Figura 266 podemos observar unas reglas generadas mediante el algoritmo PART de la herramienta WEKA que genera reglas a partir del algoritmo clasificador C4.5. El algoritmo utilizado, dentro de los múltiples algoritmos que permite este programa para clasificar, es el algoritmo J4.8, que está basado en la última versión libre del conocido algoritmo de clasificación C4.5 R8 antes de que se pasara a la versión comercial C5.0. Este algoritmo genera un árbol podado y realiza una validación cruzada del mismo. Para evitar problemas entre la diferencia de muestras de errores BAJOS frente a los ALTOS, se reduce el número de observaciones de las bobinas con errores BAJOS a un número doble que el de ALTOS.

El interés estriba en poder generar un sistema experto que nos indique, mediante reglas, momentos que puedan ser problemáticos.

De este modo, las reglas que se muestra a continuación y que son obtenidas de los resultados mostrados en la Figura 266:

```
GRUPOERRORANTES = BAJO AND
CAMBACERO = FALSE AND
CAMBANCH = FALSE: BAJO (72.0/11.0)

GRUPOERRORANTES = ALTO: ALTO (17.0)

CAMBACERO = TRUE: BAJO (17.0/1.0)

: ALTO (9.0/2.0)
```

Figura 267. Una de las reglas obtenidas.

Que se podría implementar de la siguiente forma:

```
Si el error de la bobina anterior es BAJO, y no hay cambio de acero, ni de
   anchura, el error será BAJO en un 72/(11+72)=86,74% de los casos.
Si no se cumple la anterior y el error anterior era ALTO, será ALTO.
Si no se cumple lo anterior y hay cambio de acero, el error es BAJO en un
   17/(1+17)=94,44% de los casos.
Si no se cumple ninguno de los casos anteriores, el error es ALTO en un
   9/(2+9)=81% de los casos.
```

Figura 268. Implementación del sistema de decisión en el intercambio de bobinas.

Lamentablemente este sistema **no puede ser implementado debido a dos causas fundamentales**:

- El pequeño número de muestras utilizado que aleja al clasificador de ser un buen generalizador.
- El pequeño grado de eficiencia demostrado en la validación cruzada. Debido a que solamente se ha obtenido una tasa de acierto del 86,08% que no ha podido ser mejorada sustancialmente con ninguno de los diferentes clasificadores de que dispone la herramienta WEKA: PART, J48.8, ID3, etc.

Aún así, se considera que puede ser interesante que los expertos estudien con diferentes tipos de clasificadores las reglas obtenidas por ellos, pues quizás se puede obtener conocimiento interesante del proceso.

6.3.1.1 CONCLUSIONES DEL USO DE LA HERRAMIENTA WEKA

Los clasificadores generados no han resultado ser muy precisos, aunque, de la observación de las reglas asociativas, si se ha podido obtener las siguientes conclusiones:

- **Esto parece indicar que los errores altos en "el modo manual" son bastante frecuentes porque no se reacciona a tiempo en el manejo de las curvas de velocidades y temperaturas de consignas de horno para reducir el error entre la medida de temperatura del pirómetro 2 y la de consigna.**
- **Los cambios de espesor y anchura de banda pueden ser parte de las causas de los errores ALTOS en bobinas, pero se ha visto que en "modo manual", un 66,67% de las bobinas con errores ALTOS, NO se producen por cambios de anchura, de espesor o tipo de acero de la bobina.**

Aún así, es cierto, que podría mejorarse la base de datos a estudiar, en vez de con bobinas, con valores dinámicos separados en unidades de tiempo más constantes y con un agrupamiento mejor de los casos.

Lo que si se ha visto, que es conveniente determinar si la clasificación que realiza la empresa para las bobinas es adecuada o si, incluso, puede ser simplificada.

6.3.2 BÚSQUEDA DE GRUPOS DE BOBINAS

Como hemos visto en estudios anteriores, el problema del modelizado y la búsqueda de conocimiento que nos explique el funcionamiento del sistema, debe ser abordado separadamente para cada tipo de bobina. Esto es debido, a que el comportamiento de cada bobina frente al horno depende de sus dimensiones y del tipo de acero de la misma.

La primera duda que surge es si los tipos de aceros pueden ser agrupados en grupos más amplios según su comportamiento. Para intentar resolverla, hacemos uso de diferentes proyectores y técnicas de *Data Mining* visual.

6.3.2.1 ESTUDIO CON PROYECTOR "SAMMON"

Lo primero que hacemos, es utilizar el proyectos *Sammon*, para visualizar los puntos de operación del horno para las 10 bobinas más importantes.

```
# Cargamos librerias Multivariante
library(mva)
library(multiv)
library(sm)

# Creamos las matrices
# Extraemos los datos de las bobinas más usuales
BOBINASESTUDIAR <- c("B011F97", "B012F53", "B025F55", "B100F55", "B102G33",
"B102G55", "B105F55", "C107G55", "C114G55", "K011F57")

DATBOBINASESTU <- DATBOBINAS[DATBOBINAS$CLASACERO %in% BOBINASESTUDIAR,]
MATBOBINASESTU <- MATBOBINAS[DATBOBINAS$CLASACERO %in% BOBINASESTUDIAR,]
# Creamos la SECCION
SECCIONESTU <- as.numeric(as.matrix(DATBOBINASESTU$ESPENT))*
as.numeric(as.matrix(DATBOBINASESTU$ANCHO))

# Creamos una matriz de datos
table(DATBOBINASESTU$CLASACERO)

B011B99 B011F97 B012B97 B012F53 B012F55 B013B55 B013C55 B014F53 B014F55 B016F35
      0      51       0     129       0       0       0       0       0       0
B017F53 B023H53 B025F55 B032H53 B042H53 B044H53 B081B99 B085F97 B085G99 B100B95
      0       0      45       0       0       0       0       0       0       0
B100F33 B100F55 B101F55 B102G33 B102G55 B103G33 B103G55 B105F55 B120G55 C107G55
      0     650       0     152      82       0       0     295       0      52
C114G55 C115G55 C116G55 D012F55 D012G99 D031B33 D032F55 D071F55 D094B33 D094G55
    113       0       0       0       0       0       0       0       0       0
K011B55 K011F57 K021H43 K021H53 K022H53 N013H53 N017B97 X100G99
      0      63       0       0       0       0       0       0

MATSAM2 <- cbind(DATBOBINASESTU$CLASACERO, DATBOBINASESTU$DUREZA, SECCIONESTU,
as.numeric(as.matrix(MATBOBINASESTU$MODOBOB)),
as.numeric(as.matrix(MATBOBINASESTU$THF1MEDTOTAL)),
as.numeric(as.matrix(MATBOBINASESTU$TMPP2CNGMEDTOTAL)),
as.numeric(as.matrix(MATBOBINASESTU$VELMEDTOTAL)),
as.numeric(as.matrix(MATBOBINASESTU$TMPP1MEDTOTAL)),
```

```
as.numeric(as.matrix(MATBOBINASESTU$ERRORMEDTOTALABS)));

# Eliminamos unos NA
MATSAM2[is.na(MATSAM2[,2]),2]=50
# Normalizamos los datos
MATSAMNORM <- cbind((MATSAM2[,1]-min(MATSAM2[,1]))/(max(MATSAM2[,1])-
min(MATSAM2[,1])), (MATSAM2[,2]-min(MATSAM2[,2]))/(max(MATSAM2[,2])-
min(MATSAM2[,2])), (MATSAM2[,3]-min(MATSAM2[,3]))/(max(MATSAM2[,3])-
min(MATSAM2[,3])), (MATSAM2[,4]-min(MATSAM2[,4]))/(max(MATSAM2[,4])-
min(MATSAM2[,4])), (MATSAM2[,5]-min(MATSAM2[,5]))/(max(MATSAM2[,5])-
min(MATSAM2[,5])), (MATSAM2[,6]-min(MATSAM2[,6]))/(max(MATSAM2[,6])-
min(MATSAM2[,6])), (MATSAM2[,7]-min(MATSAM2[,7]))/(max(MATSAM2[,7])-
min(MATSAM2[,7])), (MATSAM2[,8]-min(MATSAM2[,8]))/(max(MATSAM2[,8])-
min(MATSAM2[,8])), (MATSAM2[,9]-min(MATSAM2[,9]))/(max(MATSAM2[,9])-
min(MATSAM2[,9])))

# Calculamos el sammon de DIMSAM observaciones obtenidas de la matriz MATSAM2
MATSAM <- sammon(as.matrix(MATSAMNORM[,5:8]), tol=0.03, maxit=1000,
diagnostics=TRUE)

# Tipos
TIPOBOBPUNT <- c(2,4,13,22,24,25,28,30,31,42)
TIPOPUNT <- match(MATSAM2[,1], TIPOBOBPUNT)
COLORB <- 4+as.numeric(MATSAM2[,4]*2)

ZONA <- MATSAM$rproj[,1]>0.2 & MATSAM$rproj[,1]<1.2 &
MATSAM$rproj[,2]>-0.5 & MATSAM$rproj[,2]<1
MATJ <- MATSAM$rproj[ZONA,]
plot(MATJ[,1], MATJ[,2], pch=TIPOPUNT[ZONA],col= COLORB[ZONA])

# Dibujamos gráfica de densidades
MJ <- MATJ
#MJ[match(max(MATSAM$rproj[,2]),MATSAM$rproj[,2]),2] <- mean(MATSAM$rproj[,2])
sm.density(MJ)
```

Figura 269. Programa que genera la proyección Sammon de los puntos de operación.

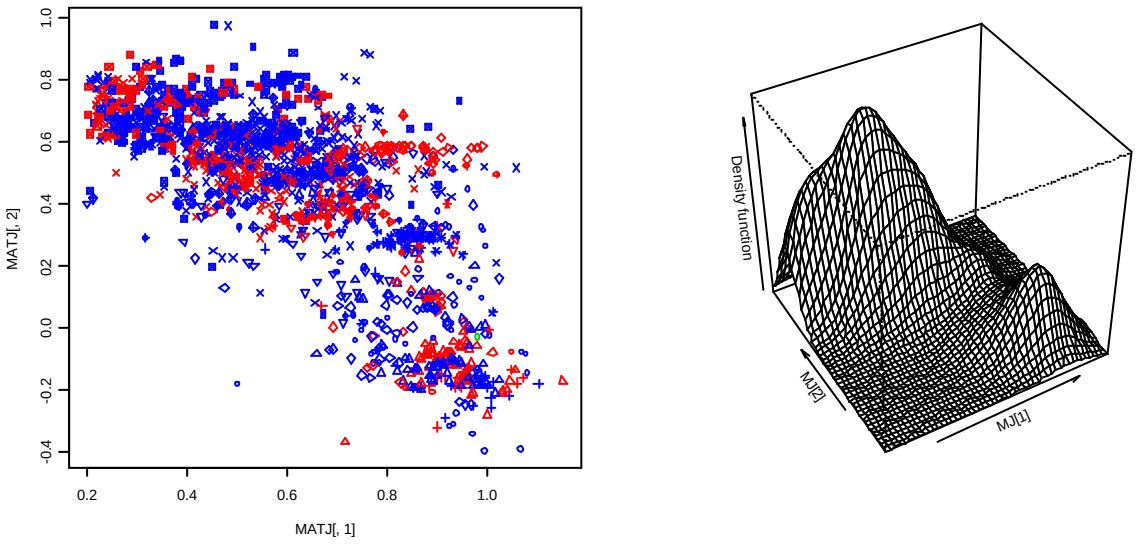


Figura 270. Figura Sammon con los puntos de operación del horno.

En la Figura 270 podemos ver los puntos de operación del proceso horno sacados de la proyección de las variables de consigna del mismo: *MATBOBINASESTU$THF1MEDTOTAL*, *MATBOBINASESTU$TMPP2CNGMEDTOTAL*, *MATBOBINASESTU$VELMEDTOTAL*, *MATBOBINASESTU$TMPP1MEDTOTAL*. En azul oscuro se muestran los puntos en "modo manual" y en rojo los que han sido realizados en "modo automático".

Se pueden apreciar dos zonas diferenciadas, aunque inicialmente no se puede saber cual es la variable que influye en esa separación. Por otro lado, sí se puede apreciar cómo los puntos de operación varían ostensiblemente para cada tipo de bobina.

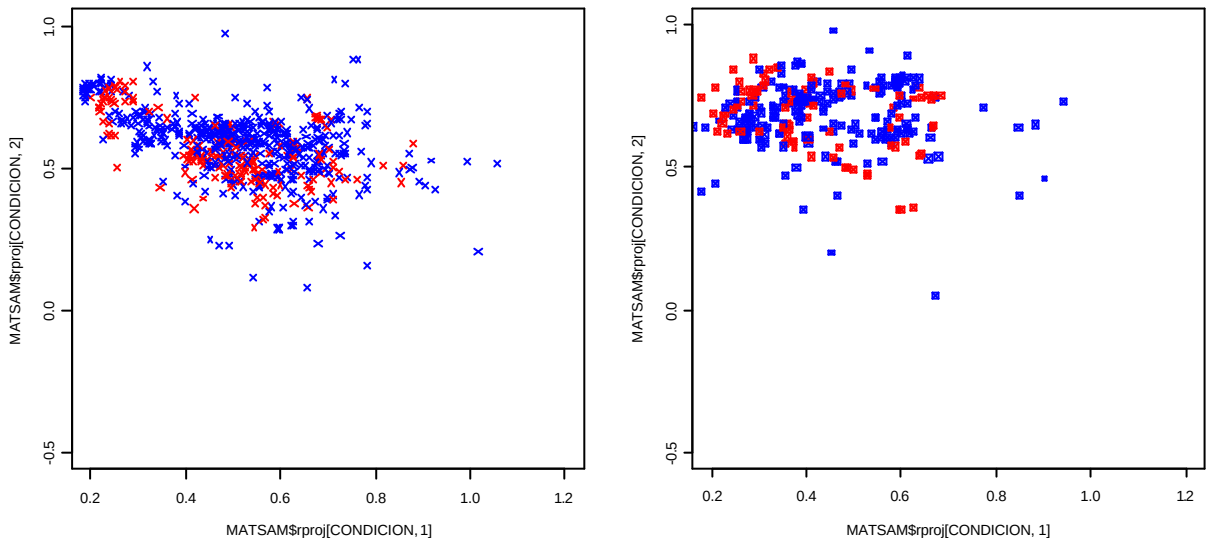


Figura 271. Puntos de operación en modo manual y automático para las bobinas con acero más comunes B100F55 (izquierda) y B105F55 (derecha) (650 y 295 respectivamente).

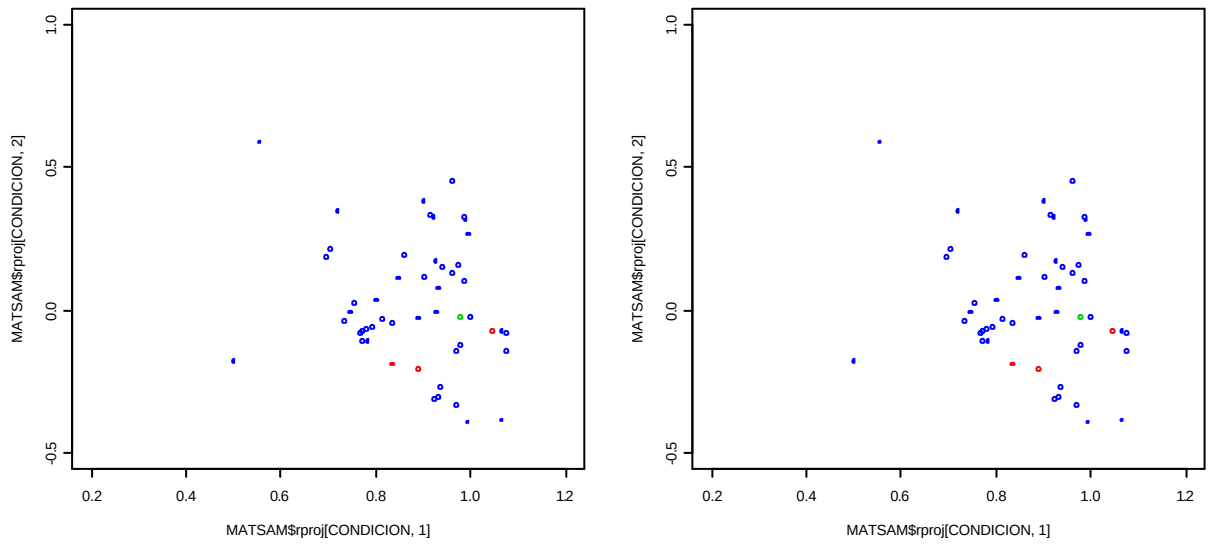


Figura 272. Puntos de operación en modo manual y automático para las bobinas con acero B011F97 (izquierda) y K011F57 (derecha) (650 y 295 respectivamente).

```
 # Obtenemos los tipos de bobinas
TIPOBOBPUNT <- c(2,4,13,22,24,25,28,30,31,42)
TIPOPUNT <- match(as.numeric(DATBOBINASESTU$CLASACERO), TIPOBOBPUNT)
COLORB <- 4+as.numeric(as.matrix(MATBOBINASESTU$MODOBOB))*2

# Dibujamos puntos de operación para diferentes bobinas
CONDICION <- DATBOBINASESTU$CLASACER == ”B100F55”
plot(MATSAM$rproj[CONDICION,1], MATSAM$rproj[CONDICION,2],ylim=c(-
0.5,1),xlim=c(0.2,1.2),pch=TIPOPUNT[CONDICION],col= COLORB[CONDICION])
CONDICION <- DATBOBINASESTU$CLASACER == ”B105F55”
plot(MATSAM$rproj[CONDICION,1], MATSAM$rproj[CONDICION,2],ylim=c(-
0.5,1),xlim=c(0.2,1.2),pch=TIPOPUNT[CONDICION],col= COLORB[CONDICION])

CONDICION <- DATBOBINASESTU$CLASACER == ”B011F97”
plot(MATSAM$rproj[CONDICION,1], MATSAM$rproj[CONDICION,2],ylim=c(-
0.5,1),xlim=c(0.2,1.2),pch=TIPOPUNT[CONDICION],col= COLORB[CONDICION])
CONDICION <- DATBOBINASESTU$CLASACER == ”K011F57”
plot(MATSAM$rproj[CONDICION,1], MATSAM$rproj[CONDICION,2],ylim=c(-
0.5,1),xlim=c(0.2,1.2),pch=TIPOPUNT[CONDICION],col= COLORB[CONDICION])

# Visualizamos según Tipos de durezas por colores
# 14=Negro
# 15=Rojo
# 17=Verde
# 19=Azul
# 50=Azul Claro
# E8=Magenta

TIPODURPUNT <- c(“14”,”15”,”17”,”19”,”50”,”E8”)
TIPOPUNTD <- match(DATBOBINASESTU$DUREZA, TIPODURPUNT)
plot(MATSAM$rproj[,1], MATSAM$rproj[,2], ylim=c(-0.5,1), xlim=c(0.2,1.2),
pch=19,col= TIPOPUNTD)

# Visualizamos puntos de elevado error según dureza
CONDICION <- as.numeric(as.matrix(MATBOBINASESTU$ERRORMEDTOTALABS))>=30
PCHNORM <- rep(3,length(MATBOBINASESTU$ERRORMEDTOTALABS))
PCHNORM[CONDICION] <- 19
plot(MATSAM$rproj[,1], MATSAM$rproj[,2],pch=’.’, ylim=c(-0.5,1),
xlim=c(0.2,1.2),col=TIPOPUNTD)
points(MATSAM$rproj[CONDICION,1],
MATSAM$rproj[CONDICION,2],pch=PCHNORM[CONDICION],col= TIPOPUNTD[CONDICION])
```

Figura 273. Programa que genera los diferentes gráficos Sammon.

Más interesante resulta observar la Figura 274, donde se muestran los puntos de operación para cada dureza del acero. Los colores utilizados son:

- Dureza 14=Negro
- Dureza 15=Rojo
- Dureza 17=Verde
- Dureza 19=Azul
- Dureza 50=Azul Claro
- Dureza E8=Magenta

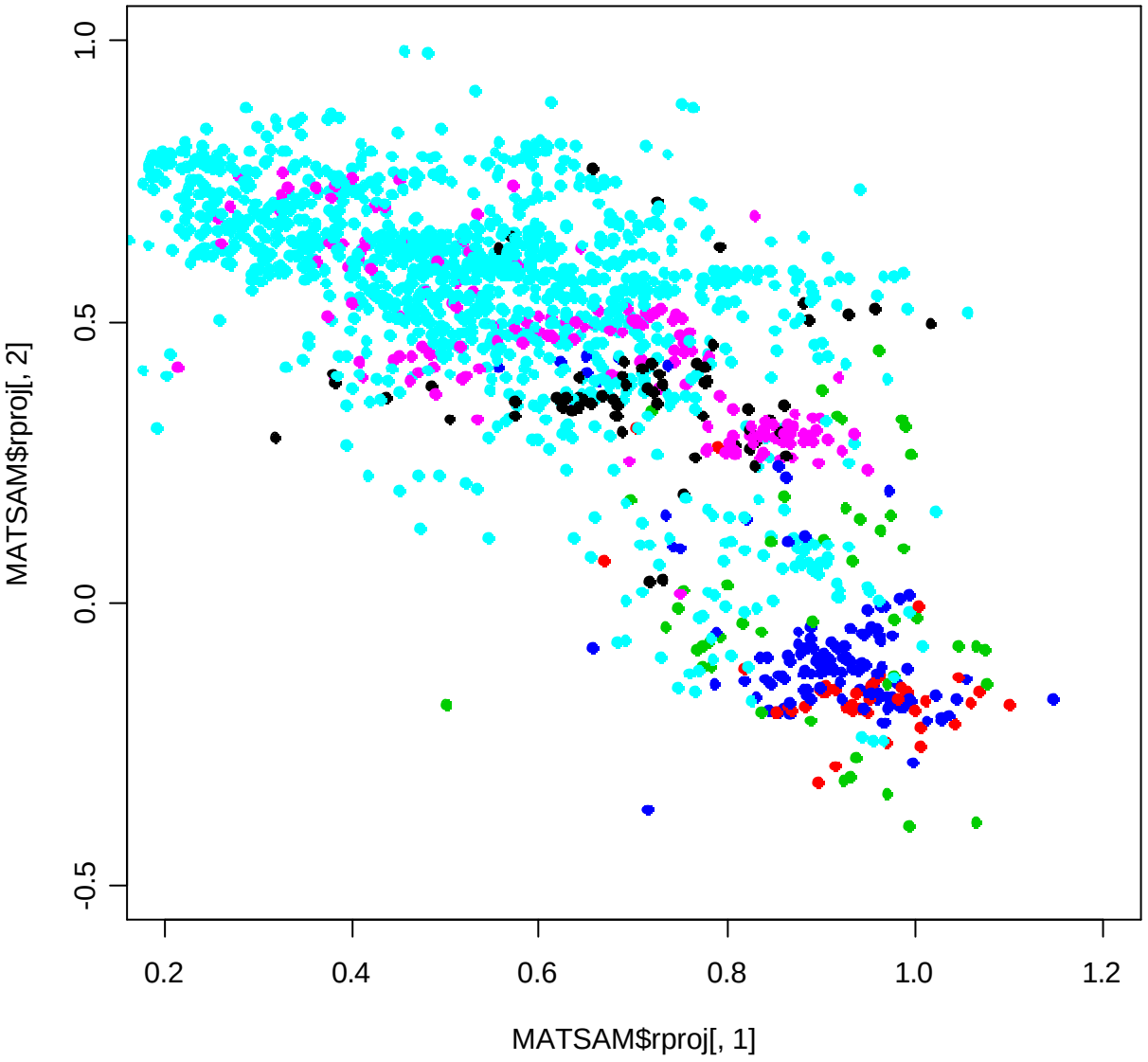


Figura 274. Visualización de puntos de operación para cada dureza.

Vemos claramente que los puntos de operación del proceso dependen de la dureza del acero. Así podemos observar como las durezas 19 (azul) y 15 (rojo) se focalizan en la esquina inferior derecha. Igualmente la 17 (verde) se sitúa en esa zona, pero de una manera más dispersa.

Por otro lado, la 14 (negro) y la E8 (magenta) están también focalizadas por encima de los grupos anteriores, aunque ésta última mucho más expandida. Por último, queda la dureza 50 (azul claro), que es la más abundante y cuyos puntos de operación se mueven fundamentalmente por toda la mitad superior del gráfico.

En la Figura 275 podemos observar los puntos de operación con errores elevados (puntos gruesos) frente a los demás (puntos finos).

La mayoría de los errores elevados se distribuyen en zonas periféricas de sus puntos de operación, excepto en los más numerosos (azul claro) donde existen algunos en zonas de operación periféricas y otros en la zona central.

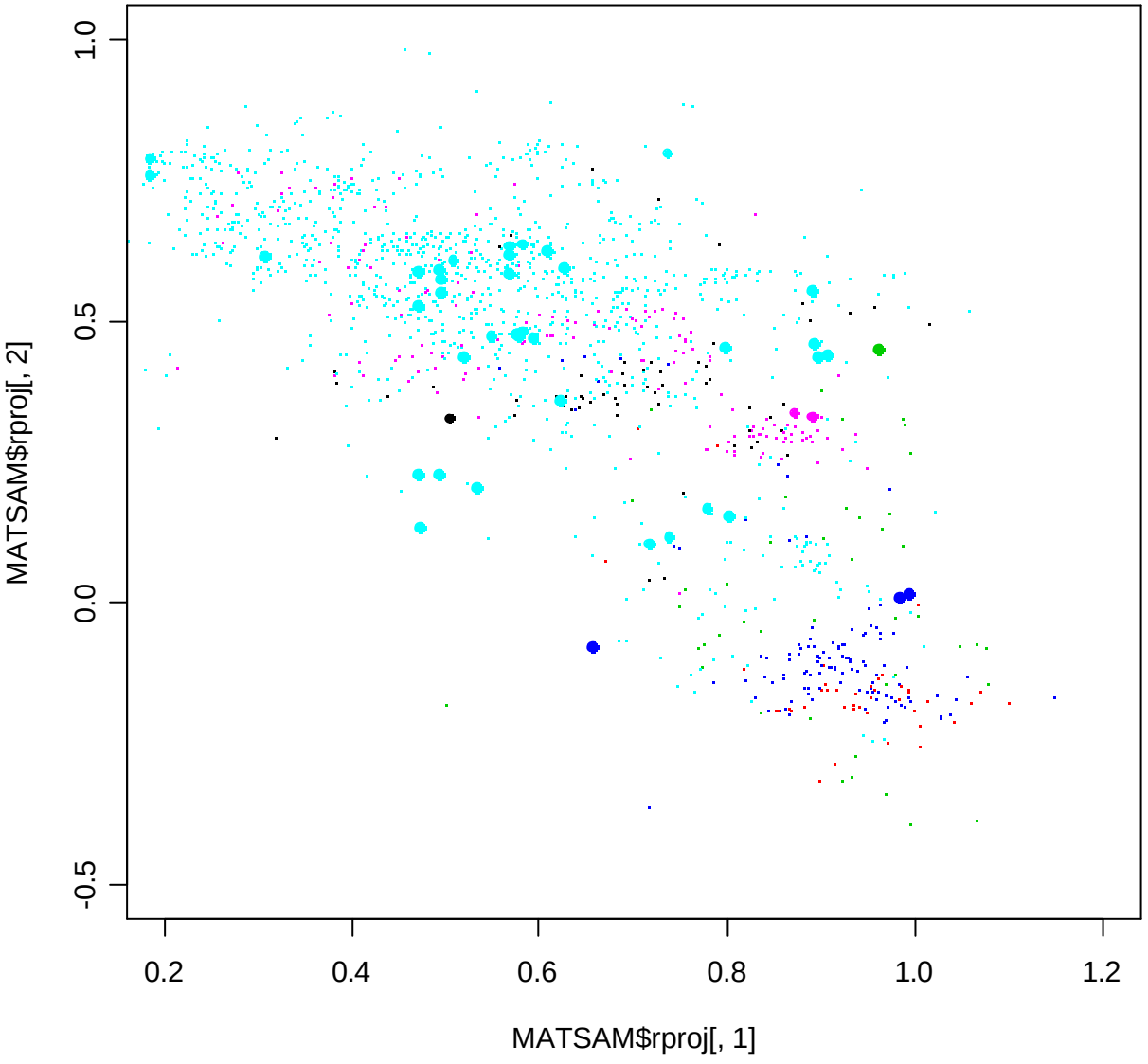


Figura 275. Distribución de puntos de operación con errores elevados (puntos gruesos) frente a los demás (puntos).

CONCLUSIONES

Hemos corroborado, **que los puntos de operación del proceso del galvanizado son muy dependientes del tipo de dureza del acero, de forma que, se podría usar ésta como clasificador de bobinas y seguir realizando el estudio agrupando las bobinas según sus similitudes de composición metalúrgica.** Esta suposición es muy lógica ya que dentro del modelo matemático del horno, se utiliza la capacidad de calentamiento de la banda como uno de los parámetros más importantes y éste lógicamente variará según su tipo.

Incluso se ha observado (aunque el número de muestras no era muy elevado) que, en algunos aceros, la variación de ese punto de operación fuera de la zona de proceso, puede indicar la posibilidad de "error ALTO".

Por otro lado, surgen las siguientes preguntas:

- ¿Es correcto el uso de la dureza del acero como "separador" de las diferentes familias de bobinas?
- Si vemos que el tipo de acero influyen en los puntos de operación del proceso. ¿Es correcto el uso que hace el modelo según la clasificación actual de los tipos de aceros?
- ¿Puede obtenerse otro criterio clasificador de bobinas más adecuado?

Estas preguntas, se intentarán responder en el apartado siguiente.

6.3.2.2 BÚSQUEDA DE FAMILIAS DE BOBINAS SEGÚN LA COMPOSICIÓN METALÚRGICA DEL ACERO

Mediante el uso de diversos proyectores y técnicas de clusterizado, se va a intentar obtener unos criterios objetivos de clasificación de bobinas según la composición metalúrgica de los aceros de las mismas.

Para ello, se trabajará con la tabla de composición del acero de cada bobina (Figura 276). En ésta aparece en la primera columna, el código de la bobina y en la segunda el código de la colada.

CIDPR	COLADA	C	Mn	Si	S	P	Al-Tot	Cu	Ni	Cr	Nb	V	Ti	B	N	Ceq
23223001	294544	0,0023	0,1223	0,0061	0,0084	0,0082	0,0252	0,0202	0,0164	0,0164	0,0013	0,002	0,0611	0,0001	0,004	0,0226
23223002	294512	0,0033	0,1367	0,0066	0,0102	0,0083	0,0373	0,0139	0,0175	0,0176	0,0014	0,0025	0,0737	0,0001	0,0034	0,026
23223003	294671	0,0044	0,1104	0,0058	0,0096	0,0101	0,0342	0,0184	0,0164	0,0152	0,0007	0,002	0,0704	0,0001	0,0037	0,0227
23223004	294758	0,0029	0,1273	0,0078	0,0085	0,0093	0,029	0,0111	0,0311	0,0265	0,0006	0,0026	0,0673	0,0001	0,0037	0,024
23223005	294757	0,0029	0,1204	0,0083	0,0092	0,0088	0,0276	0,0168	0,0336	0,0277	0,001	0,0019	0,0634	0,0001	0,0034	0,0229
23223006	295254	0,0028	0,1086	0,0089	0,0079	0,0103	0,0263	0,0189	0,0181	0,019	0,0005	0,0024	0,0641	0,0001	0,0033	0,0208
23223007	294756	0,0023	0,1442	0,0085	0,007	0,0118	0,0257	0,008	0,034	0,0199	0,0001	0,0014	0,0674	0,0001	0,0042	0,0262
23223008	294758	0,0029	0,1273	0,0078	0,0085	0,0093	0,029	0,0111	0,0311	0,0265	0,0006	0,0026	0,0673	0,0001	0,0037	0,024
23223009	294658	0,0025	0,1273	0,0064	0,0091	0,0102	0,0332	0,0212	0,017	0,017	0,0013	0,0026	0,07	0,0001	0,0038	0,0236

Figura 276. Tabla de composición de los aceros de cada bobina.

USO DEL PROYECTOR SAMMON

Lo primero que aplicamos, es el proyector sammon para intentar ver si existen familias.

```
# Leemos el archivo de datos en formato “csv2”
MATACEROS <-
read.csv2(file="C:\\JAVI_CASA\\PISON_DESPACHO\\DOCTORADO\\TESIS\\Tesis_31_01_03\
\apoyo\\2003\\aceros.csv")

# Calculamos el sammon
MATSAMACEROS <- sammon(as.matrix(MATACEROS[,3:17]), tol=0.005, maxit=1000,
diagnostics=TRUE)

# Dibujamos los puntos
plot(MATSAMACEROS$rproj[,1], MATSAMACEROS$rproj[,2])

# Dibujamos más ampliada la zona importante
plot(MATSAMACEROS $rproj[,1], MATSAMACEROS
$rproj[,2],xlim=c(0.4,1.0),ylim=c(0.15,0.6),pch=3)
ZONA <- MATSAMACEROS$rproj[,1]>0.4 & MATSAMACEROS$rproj[,1]<1.0 &
MATSAMACEROS$rproj[,2]>0.15 & MATSAMACEROS$rproj[,2]<0.6
MATJ <- MATSAMACEROS$rproj[ZONA,]
plot(MATJ$rproj[,1],MATJ$rproj[,2])
sm.density(MATJ)
```

Figura 277. Programa que realiza la proyección Sammon.

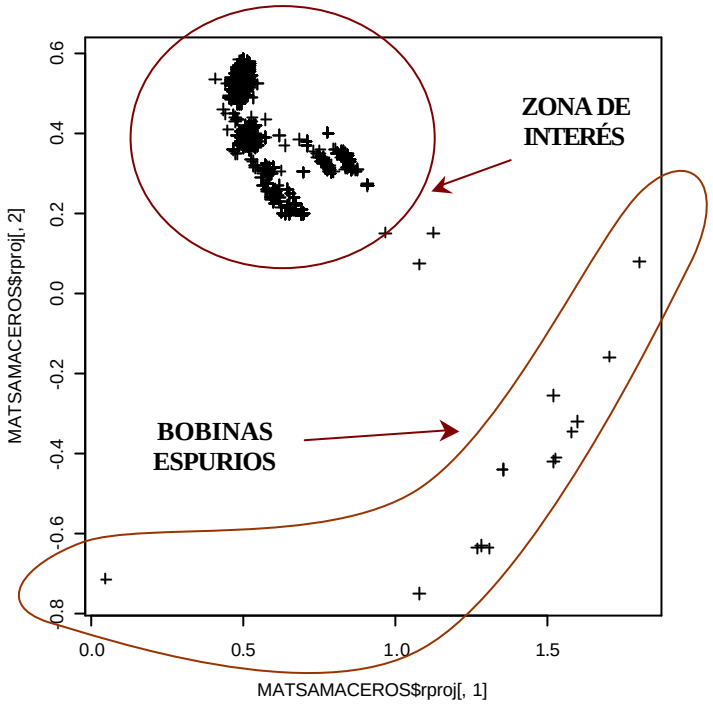


Figura 278. Proyección Sammon de las diferentes bobinas según la composición del acero.

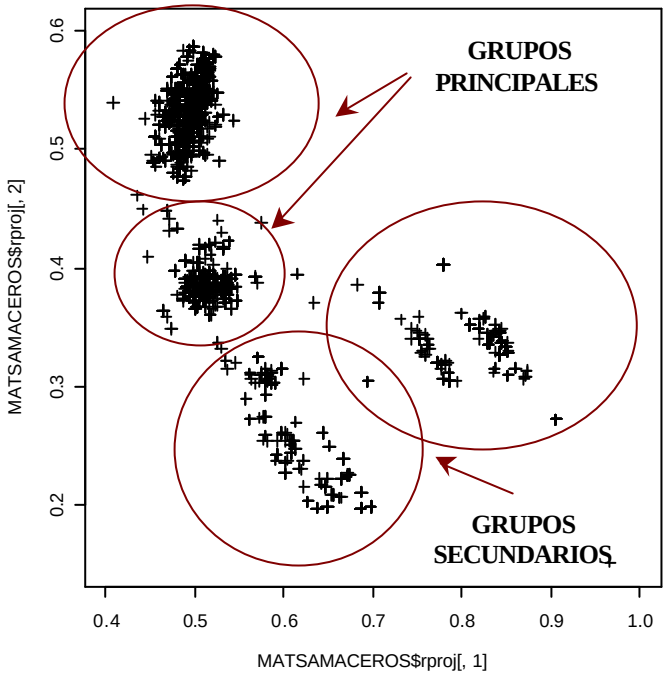


Figura 279. Ampliación de la zona más densa.

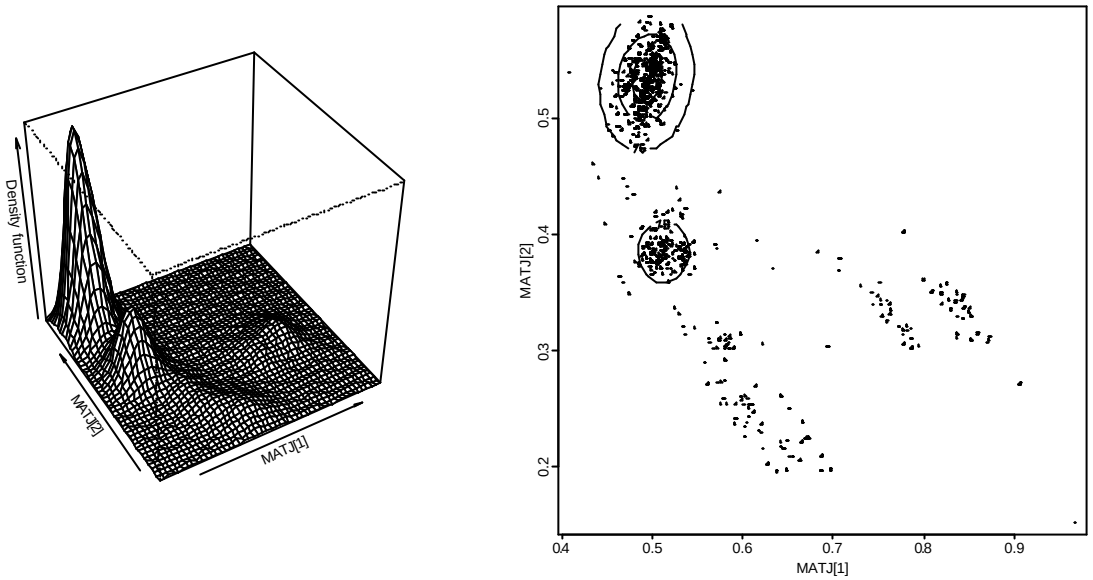


Figura 280. Densidad de bobinas en la zona ampliada.

De las figuras anteriores, podemos extraer las siguientes conclusiones:

- Claramente, hay dos familias de bobinas principales. Existe dos grupos principales y otros cuatro secundarios, que incluso pueden ser agrupados solamente en dos o en tres.
- Existen 13 bobinas que están fuera de toda clasificación.

Sería por lo tanto importante, poder obtener un sistema de clasificación de bobinas que permita, dada la composición del acero, determinar a qué familia pertenece.

CREACIÓN DE UN CLASIFICADOR DE BOBINAS

Lo primero que realizamos, es determinar la dimensión de la estructura intrínseca de los datos. Para ello, utilizamos el cálculo de la *dimensión fractal* [HAL86] con la librería *fdim* [ORD00a] para obtenerla.

```
# Eliminamos las bobinas espúreas
ZONA <- MATSAMACEROS$rproj[,1]>0.4 & MATSAMACEROS$rproj[,1]<1.0 &
MATSAMACEROS$rproj[,2]>0.15 & MATSAMACEROS$rproj[,2]<0.6
MATACERNUEV <- MATACEROS[ZONA,]
# Calculamos la Dimensión Fractal
df <- fdim(as.matrix(MATACERNUEV[,3:17]),q=0,Alpha=0.2,PlotF=TRUE)
print(df$fdim)
      X1
1.416724
```

Figura 281. Programa que calcula la dimensión fractal de los datos usados.

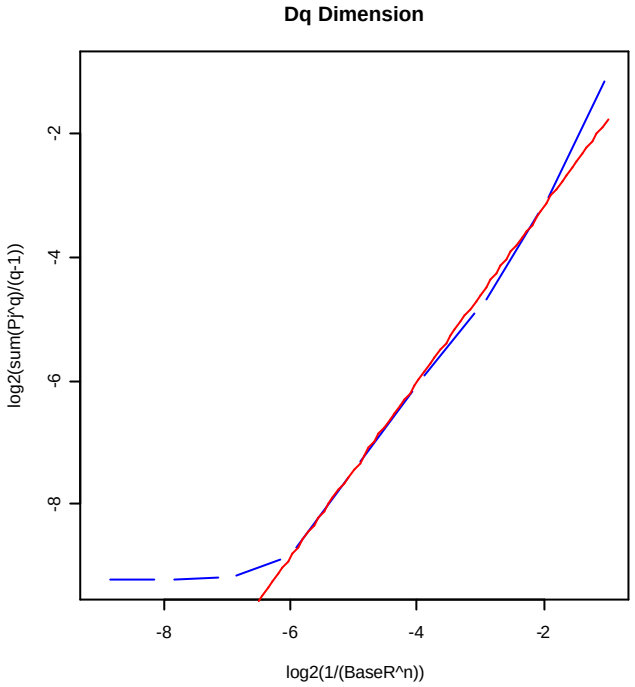


Figura 282. Pendiente obtenida del cálculo de la dimensión fractal de todas las bobinas.

Observamos que la dimensión intrínseca de los datos está entre 1 y 2. Lo que nos indica claramente que la estructura intrínseca de los datos está en una hipersuperficie y que posiblemente con dos ejes del "proyector PCA" o de un proyector "PCA no lineal" podamos explicarla. Aún así, como la cantidad de bobinas no es muy elevada, estos resultados deben ser tratados con cierta prudencia.

```
# Obtenemos las bobinas sin espúreos
ZONA <- MATSAMACEROS$rproj[,1]>0.4 & MATSAMACEROS$rproj[,1]<1.0 &
MATSAMACEROS$rproj[,2]>0.15 & MATSAMACEROS$rproj[,2]<0.6
MATACERNUEV <- MATACEROS[ZONA,]

# Obtenemos la proyección PCA
PCAACERO <- pca(as.matrix(MATACERNUEV[,3:17]),method=2)

# Vemos el grado de información de cada eje
# (los dos primeros abarcan el 97,37% de la varianza)
PCAACERO$evals/sum(PCAACERO$evals)
[1] 0.852458715 0.121311206 0.013563774 0.005591092 0.002940928 0.002383562
[7] 0.001750722

0.852458715+0.121311206
[1] 0.97377

# Visualizamos la proyección de las bobinas con los dos ejes principales PCA
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2],pch=3)
```

Figura 283. Cálculo del PCA y proyección de las bobinas en los dos ejes principales.

En la Figura 283 podemos ver, tal y como habíamos previsto con el cálculo de la dimensión fractal, que los dos primeros ejes de la proyección PCA abarcan un alto porcentaje de la varianza, el 97,37%, y que pueden ser utilizados con garantías para poder proyectar cada una de estas bobinas.

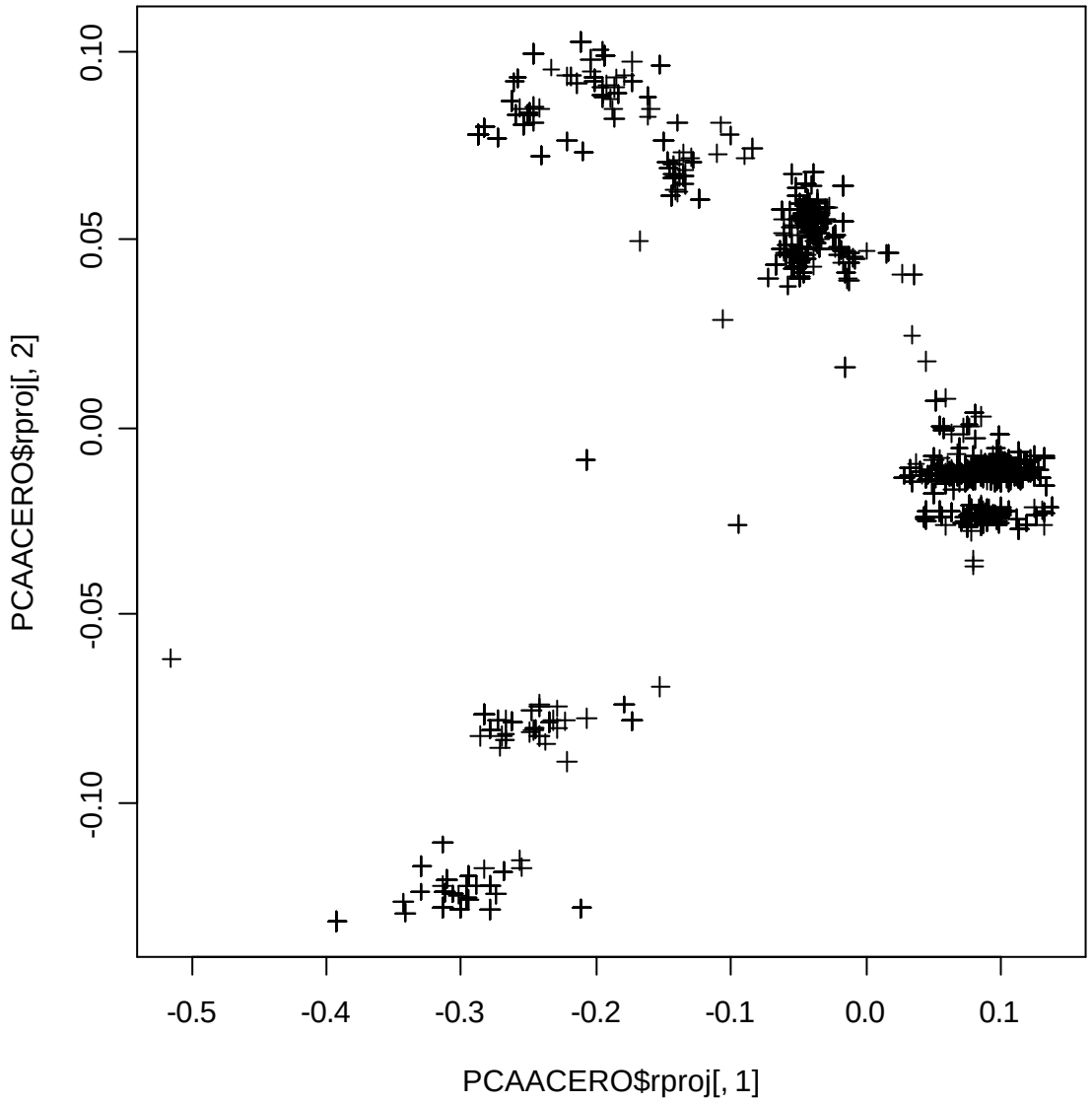


Figura 284. Proyección de las bobinas según la composición del acero usando los dos ejes principales del PCA.

En la Figura 284, vemos los mismos grupos anteriores pero, esta vez, obtenidos con el proyector PCA. **La ventaja principal, es que podemos utilizar los dos ejes principales para proyectar cualquier otra bobina y ver dónde está situada.**

```
PCAACERO$evecs
              Comp1         Comp2        Comp3         Comp4        Comp5
C      -0.1024804089  0.3990987459  0.018545810  0.4942042774 -0.005423465
Mn     -0.9198100130 -0.0527227803  0.233868101 -0.2763186801  0.032262943
Si     -0.2324455956 -0.6057458521 -0.537671713  0.3906571712 -0.033772071
S      -0.0034960441  0.0153275660 -0.063702379 -0.0209550485 -0.058773439
P      -0.0878961436 -0.2193394449 -0.164802144  0.1018262177 -0.045727689
Al.Tot -0.0111097784  0.0696342422 -0.086174415 -0.0801315215  0.854636576
Cu      0.0027200969 -0.0015921302  0.026088669 -0.0795425382 -0.416511894
Ni      0.0006906298  0.0123756541 -0.018642516  0.0136537040 -0.093709296
Cr     -0.0017298936 -0.0130193988  0.020027674 -0.0240693009 -0.069227397
Nb     -0.0206599001  0.0437644691  0.206401553 -0.1329976054 -0.237900706
V       0.0039525277 -0.0178896447  0.022927248  0.0152544118 -0.010621628
Ti      0.1313637707 -0.5130783911  0.753109714  0.3452067755  0.130292655
B      -0.0008623904 -0.0003938356 -0.011097136  0.0003007922 -0.019953822
N      -0.0024801141  0.0097156607 -0.006916984  0.0165151714  0.014166424
Ceq    -0.2526624469  0.3897617976  0.072975075  0.6042465978 -0.022349801
              Comp6        Comp7
C       0.021570869 -0.047147872
Mn      0.015810435  0.091636390
Si     -0.017815682 -0.213765221
S       0.118024248  0.038889575
P      -0.029281777  0.058083966
Al.Tot  0.319249913 -0.307696508
Cu      0.716528926 -0.030067142
Ni      0.438398959 -0.321133501
Cr      0.248464542 -0.105815144
Nb     -0.326073530 -0.851995579
V       0.015205489 -0.001919575
Ti      0.087837167  0.034087508
B      -0.006697698  0.012768572
N       0.002869957 -0.014858592
Ceq     0.024373781 -0.038045581
```

Figura 285. Vectores obtenidos mediante PCA (los dos primeros son los utilizados para la proyección final).

Así por ejemplo, para una bobina cualquiera, podemos proyectarla y ver a qué grupo pertenece. Esto puede servir para:

- Clasificar visualmente el tipo de acero.
- Ayudar al personal que planifique el comportamiento futuro de las bobinas a procesar.
- Detectar con anterioridad bobinas anómalas.
- Etc.

```
# Obtenemos el vector para centrar datos
VECCENT <- mean(MATACERNUEV[,3:17])
VECCENT
           C           Mn           Si            S            P
0.0170708264 0.2140858678 0.0206400413 0.0086775620 0.0163334711
      Al.Tot           Cu           Ni           Cr           Nb
0.0317890909 0.0152621488 0.0181692149 0.0181490909 0.0023759091
           V           Ti            B            N          Ceq
0.0017226033 0.0497402066 0.0002957025 0.0037799174 0.0530132645

# Código de la bobina 100
MATACERNUEV[100,1]
23233019

# Obtengo la composición del acero de la bobina 23233019
# Centramos el punto de la bobina 23233019
PUNTOCENT <- MATACERNUEV[100,3:17]- VECCENT
x <- sum(PUNTOCENT*PCAACERO$evecs[,1])
y <- sum(PUNTOCENT*PCAACERO$evecs[,2])

# Visualizamos la proyección de las bobinas con los dos ejes principales PCA
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2],col='blue',pch=3)
points(x,y,col='red',pch=15)
```

Figura 286. Programa mediante los dos ejes principales del PCA proyecta una bobina cualquiera.

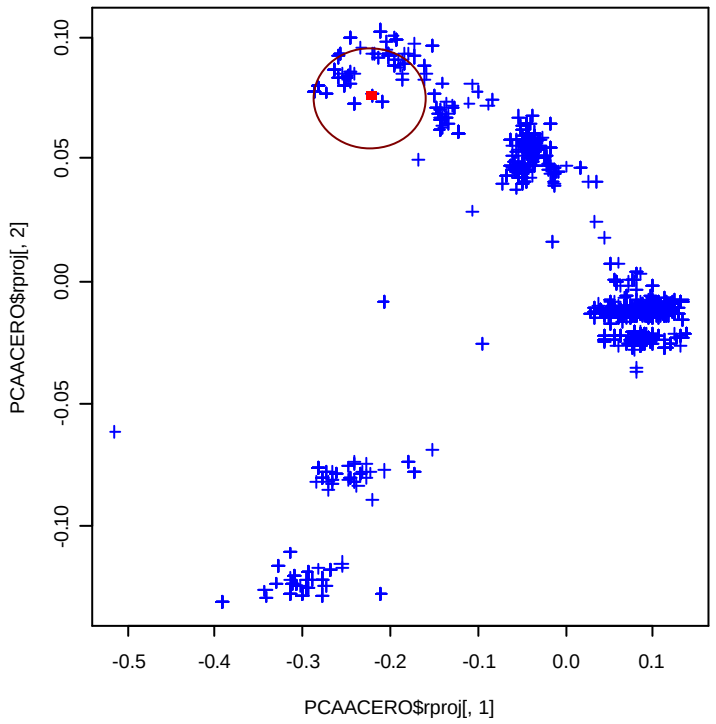


Figura 287. Situación de una bobina cualquiera en el mapa creado con los dos ejes PCA principales.

6.3.2.3 CREACIÓN DE UN CLASIFICADOR ON LINE DE BOBINAS SEGÚN EL TIPO DE ACERO

Utilizando los dos ejes principales PCA y definiendo zonas gráficas dentro del espacio de proyección, podemos generar un clasificador *On-Line* que nos ayude a determinar, en tiempo real, el tipo de una bobina a partir de la composición de su acero[21].

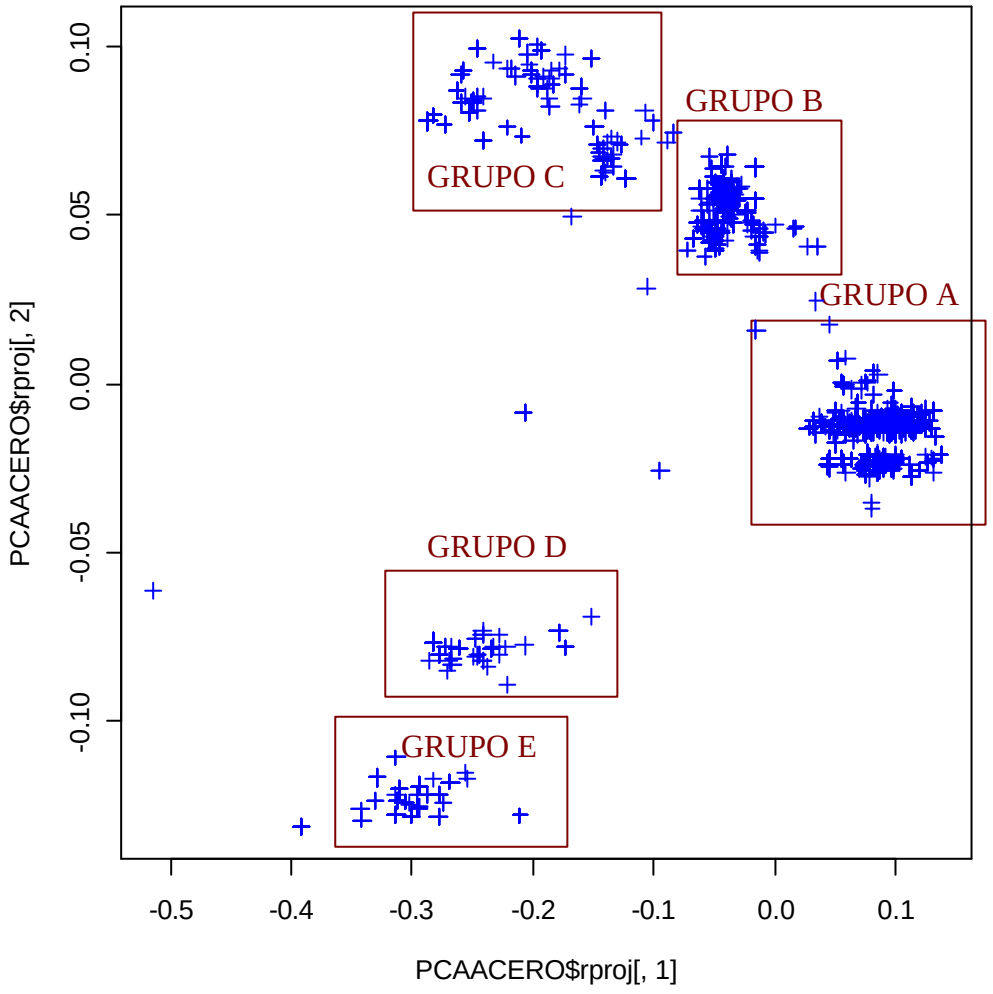


Figura 288. Particionado del espacio de proyección para la clasificación de las bobinas.

[21] Esta metodología está siendo usada ampliamente en procesos industriales de industrias químicas por WANG y otros autores [WAN99] [SEB01]. Éstos demuestran la gran utilidad y practicidad de estas técnicas para el desarrollo de técnicas de predicción y control on-line.

La implementación del clasificador puede ser muy sencilla, tal y como se muestra en la figura siguiente:

```
# Obtengo la composición del acero de la bobina XXX
# Centramos el punto de la bobina 23233019
PUNTOCENT <- DATX
x <- sum(PUNTOCENT*PCAACERO$evecs[,1])
y <- sum(PUNTOCENT*PCAACERO$evecs[,2])

# Clasificador de Bobinas
TIPOBOBINA <- "N" # No identificada
TIPOBOBINA[x>0.0 & x<0.15 & y>-0.06 & y<0.03] <- "A" # Bobina tipo A
TIPOBOBINA[x>-0.08 & x<0.05 & y>0.03 & y<0.08] <- "B" # Bobina tipo B
TIPOBOBINA[x>-0.3 & x<-0.08 & y>0.05 & y<0.12] <- "C" # Bobina tipo C
TIPOBOBINA[x>-0.3 & x<-0.1 & y>-0.09 & y<0.06] <- "D" # Bobina tipo D
TIPOBOBINA[x>-0.37 & x<-0.2 & y>-0.18 & y<0.11] <- "E" # Bobina tipo E
```

Figura 289. Clasificador que utiliza las subzonas del espacio proyectado de los dos ejes principales PCA.

USO DE ALGORITMOS DE CLUSTERIZADO

La anterior forma de implementación resulta bastante útil aunque es conveniente verificarla con algoritmos de clusterizado, ya que puede servir para determinar la validez o no de las proyecciones anteriores.

Además, la obtención de los centroides y de las distancias máximas de pertenencia a cada familia, pueden ser suficientes para desarrollar un algoritmo que detecte la pertenencia o no de una bobina. Es decir, el clasificador simplemente debe verificar que el punto *n-dimensional* de la bobina *x* pertenece a alguna de las hiperesferas situadas en el centroide de cada grupo y con radio la distancia de pertenencia del mismo.

Uso de K-Means

Inicialmente, usamos uno de los algoritmos más sencillos: el K-Means.

```
# Cargamos las librerías multivariantes y de clusterizado
library(mva)
library(cluster)

# Intentamos con 'k-means, buscar 5 familias de bobinas
KMEANSCLUS <- kmeans(as.matrix(MATACERNUEV[,3:17]),4)
COLORCLUST <- KMEANSCLUS$cluster

# Visualizamos con colores cada cluster
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2], col=COLORCLUST,pch=3)
```

Figura 290. Cálculo de clusters con k-means.

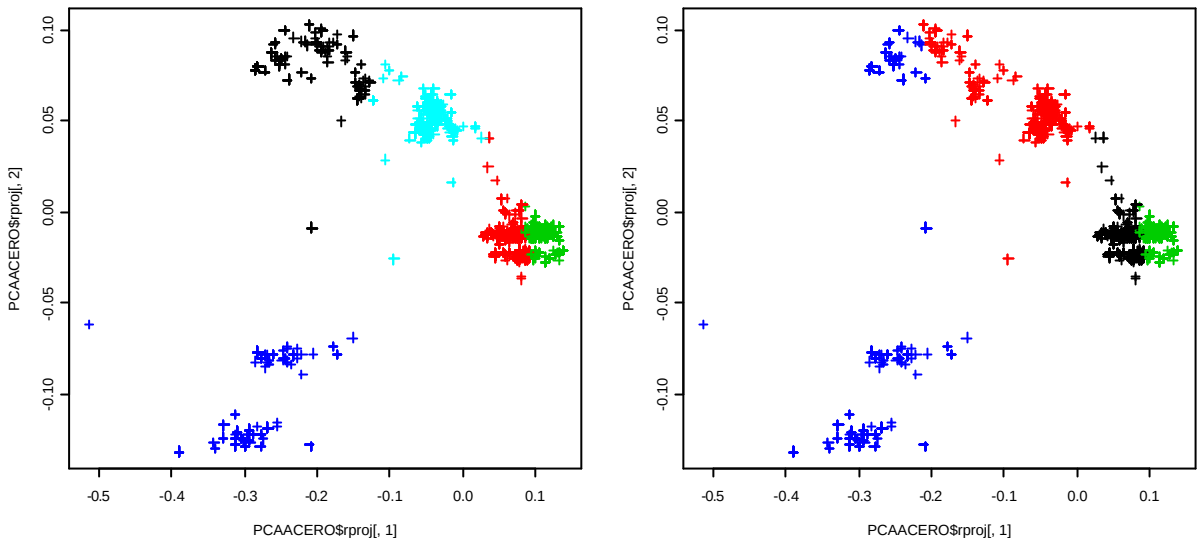


Figura 291. Clasificación con" k-means" con 5 grupos (izquierda) y 4 grupos (derecha).

De los resultados, podemos advertir que, aunque clasifica bien alguna familia, vemos que, la familia de la derecha (GRUPO A) se divide en dos grupos. Por otro lado, se observa que los dos subgrupos inferiores, no han podido ser separados.

Uso de Clara

Probamos un algoritmo más robusto a espúreos: el algoritmo "*Clara*".

```
# Cargamos las librerías multivariantes y de clusterizado
library(mva)
library(cluster)

# Intentamos con 'clara', buscar 5 familias de bobinas
CLARAX <- clara(as.matrix(MATACERNUEV[,3:17]),5)
COLORCLUST <- CLARAX$clustering
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2], col=COLORCLUST,pch=3)

# Visualizamos los cluster
CLUSTER <- CLARAX$clusinfo
     size   max_diss    av_diss isolation
[1,] 1484 0.09275796 0.02558672 0.6238290
[2,]  217 0.13400493 0.06127380 0.7602214
[3,]  504 0.12242639 0.02937021 0.8233593
[4,]  215 0.25015557 0.04856166 1.0623705
```

Figura 292. Programa que realiza el clusterizado con "Clara".

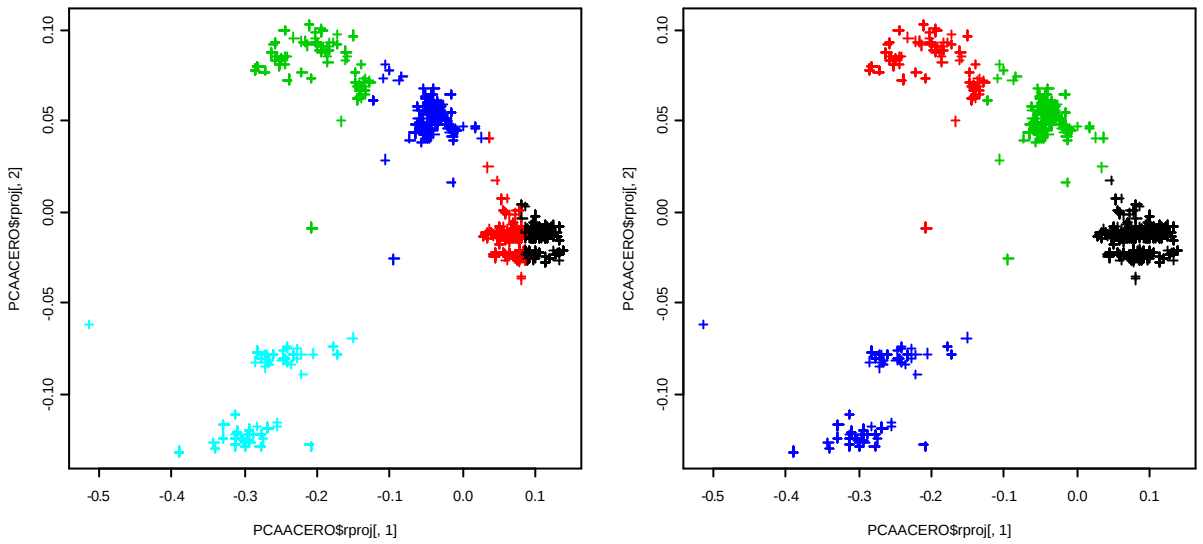


Figura 293. Clasificación con "Clara" con 5 grupos (izquierda) y 4 grupos (derecha).

En el caso del uso del algoritmo "clara", se vuelve a observar que cuando se le ha dicho que clasifique 5 o 6 subgrupos (Figura 293, parte izquierda), éste ha dividido la nube de puntos derecha en dos grupos.

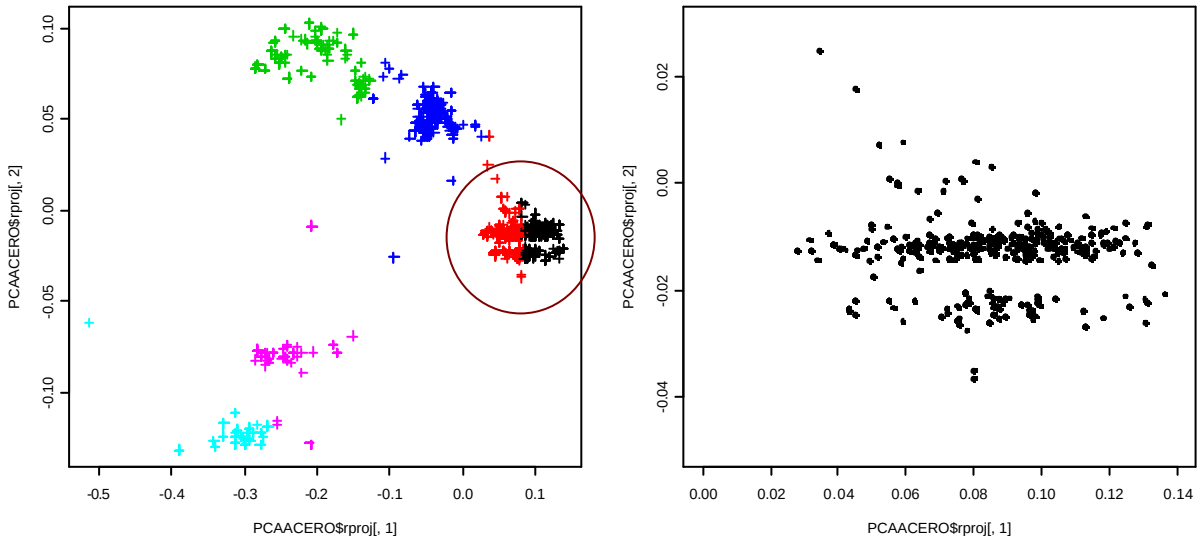


Figura 294. Separación con clara de 6 grupos (izquierda) y ampliación del grupo mayoritario, GRUPO A (derecha).

En cambio, la clasificación de 4 grupos (Figura 293, parte derecha), se ha realizado de forma correcta. Esto parece indicar, que esa nube de puntos puede estar conformada por dos subfamilias próximas entre si, y que no es detectada en la proyección de los dos ejes principales PCA.

Para verificar esto, visualizamos las proyecciones de los 5 subgrupos, con pares de ejes siguientes: (PCA1, PCA3), (PCA2, PCA3), (PCA1, PCA4) y (PCA2, PCA4).

```
# Proyecciones con otros ejes PCA
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,3], col=COLORCLUST,pch=3)
plot(PCAACERO$rproj[,2], PCAACERO$rproj[,3], col=COLORCLUST,pch=3)
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,4], col=COLORCLUST,pch=3)
plot(PCAACERO$rproj[,2], PCAACERO$rproj[,4], col=COLORCLUST,pch=3)
```

Figura 295. Buscamos proyecciones en otros ejes del PCA.

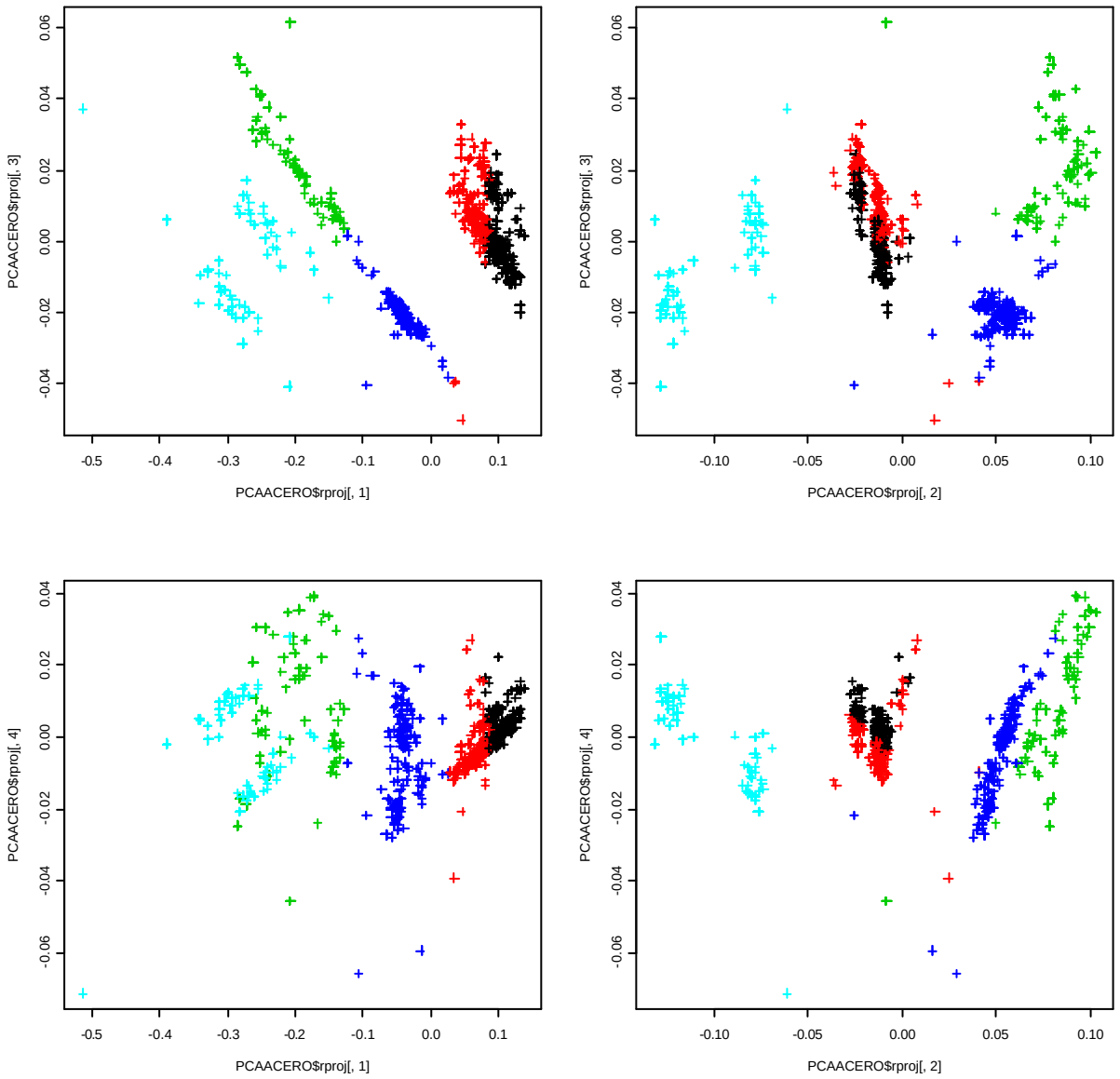


Figura 296. Proyección con otros ejes PCA.

La Figura 296 nos muestra las subfamilias proyectadas con diferentes ejes del PCA. **Vemos claramente que los dos subgrupos están muy pegados entre si, y que probablemente, esta división que realizan los algoritmos de clusterizado es debida a la gran densidad de puntos que existe en esa zona con respecto a otras, más que a distancias entre subfamilias.**

Por lo tanto, podemos concluir que **la proyección mediante el uso de los dos ejes principales del PCA representa con bastante fidelidad las familias de aceros existentes y que, por lo tanto, puede ser utilizada como clasificador mediante la detección de la subzona en que se encuenta el punto proyectado (tal y como se ha explicado anteriormente).**

ANÁLISIS DE LOS TIPOS DE ACEROS CON RESPECTO A LA NUEVA CLASIFICACIÓN

Ahora, solo queda determinar a qué grupos de los tipos de aceros antes clasificados, pertenecen la nueva clasificación de familias.

```
# Intentamos con 'clara', buscar 4 familias de bobinas
CLARAX <- clara(as.matrix(MATACERNUEV[,3:17]),4)
COLORCLUST <- CLARAX$clustering
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2], col=COLORCLUST,pch=3)

# Añadimos el tipo de subfamilia
MATACERESTU <- CLARAX$clustering
# Extraemos los datos de las bobinas existentes
GRUPOCLUS <- MATACERESTU[MATACERNUEV[,1] %in% DATBOBINAS$CODBOBINA]
BOBCLUS <-  MATACERNUEV[MATACERNUEV[,1] %in% DATBOBINAS$CODBOBINA,1]
# Extraemos el tipo de acero
DATACERBOB <- DATBOBINAS[DATBOBINAS$CODBOBINA %in% BOBCLUS,]

# Comprobamos que los codigos de bobina coinciden
table(DATACERBOB$CODBOBINA== BOBCLUS)
TRUE
1960
# Comparamos los grupos de bobinas con el tipo de acero
table(DATACERBOB$CLASACERO,GRUPOCLUS)
          GRUPOCLUS
             1  2   3   4
  B011B99    0  2   0   0
  B011F97    0  5  46   0
  B012B97    0  0   4   0
  B012F53    0  0 129   0
  B012F55    0  0   7   0
  B013B55    0  0   2   0
  B013C55    0  0   3   0
  B014F53    0  0   3   0
  B014F55    0  0   0   0
  B016F35    0  0   8   0
  B017F53    0  0   0   0
  B023H53    0  0   4   0
  B025F55    0  0  45   0
  B032H53    0  4   0   0
  B042H53    0  0   2   0
  B044H53    0  0   5   0
  B081B99    0  4   0   0
  B085F97    0 10   0   0
  B085G99    0 37   0   0
```

```
  B100B95  16  0   0   0
  B100F33   1  0   0   0
  B100F55 649  0   0   0
  B101F55  12  0   0   0
  B102G33 152  0   0   0
  B102G55  82  0   0   0
  B103G33   2  0   0   0
  B103G55  15  0   0   0
  B105F55 295  0   0   0
  B120G55  14  0   0   0
  C107G55   0  0   0  52
  C114G55   0  0   0 113
  C115G55   0  0   0   0
  C116G55   0  0   2   0
  D012F55   0 13   1   0
  D012G99   0  6   0   0
  D031B33   0 19   0   0
  D032F55   0 27   0   0
  D071F55   1 11   5   0
  D094B33   0  0   0   0
  D094G55   0  0   0   0
  K011B55   0  0  10   0
  K011F57   0  0  62   0
  K021H43   0  0   1   0
  K021H53   0  0  30   0
  K022H53   0  0   2   0
  N013H53   0  7   0   0
  N017B97   0  1   0   0
  X100G99   1  5   0  33

table(DATACERBOB$DUREZA,GRUPOCLUS)
    GRUPOCLUS
       1  2   3   4
  11   0  0  10   0
  13   0  0  12   0
  14   0  0  72   0
  15   0  0  43   0
  16   0  1  33   0
  17   0  5  43   0
  19   0  0 135   0
  20   0  0   2   0
  24   0  0   3   0
  29   0  0   4   0
  30   0  7   0   0
  32   0 51   0   0
  37   0  2   0   0
  50 1230  5   0  33
  E1   1 27   6   0
  E8   0  0   0 164
  F8   0 27   0   0
  G0   0 19   0   0
  G4   0  0   0   0
```

Figura 297. Comparación entre tipos de aceros y dureza, con las nuevas familias creadas.

Grupo 1		Grupo 2		Grupo 3		Grupo 4	
B100F55	649	B085G99	37	B012F53	129	C114G55	113
B105F55	295	D032F55	27	K011F57	62	C107G55	52
B102G33	152	D031B33	19	B011F97	46	X100G99	33
B102G55	82	D012F55	13	B025F55	45		
B100B95	16	D071F55	11	K021H53	30		
B103G55	15	B085F97	10	K011B55	10		
B120G55	14	N013H53	7	B016F35	8		
B101F55	12	D012G99	6	B012F55	7		
B103G33	2	B011F97	5	B044H53	5		
B100F33	1	X100G99	5	D071F55	5		
D071F55	1	B032H53	4	B012B97	4		
X100G99	1	B081B99	4	B023H53	4		
		B011B99	2	B013C55	3		
		N017B97	1	B014F53	3		
				B013B55	2		
				B042H53	2		
				C116G55	2		
				K022H53	2		
				D012F55	1		
				K021H43	1		

Tabla 63. Tipos de acero pertenecientes a cada grupo.

La mayoría de los tipos de aceros son clasificados en un solo grupo excepto cuatro bobinas que aparecen en dos o más grupos.

	1	2	3	4
B011F97	0	5	46	0
D012F55	0	13	1	0
D071F55	1	11	5	0
X100G99	1	5	0	33

Tabla 64. Tipos de aceros que han aparecido en varios grupos.

Vemos, que la mayoría de los tipos de aceros se pueden agrupar fácilmente en una de las familias, excepto las mostradas en la Tabla 64 que tienen entre si composiciones claramente diferenciadas y que, por lo tanto, **pueden dar problemas en el proceso de galvanizado**.

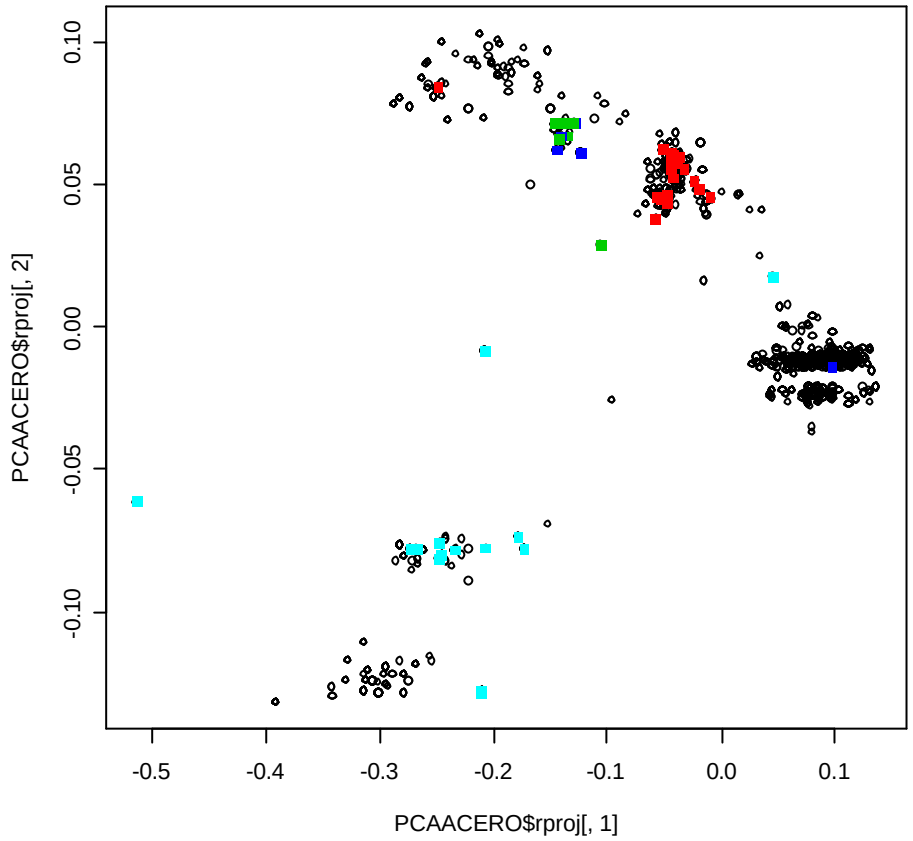


Figura 298. Posición de los aceros B011F97=”rojo”, D012F55=”verde”, D071F55=”azul”, X100G99=”azul claro”.

En este caso, todos los aceros forman agrupamientos claramente observables, aunque algunos de ellos tienen bobinas con composiciones muy distanciadas frente a las de su misma familia.

Grupo 1		Grupo 2		Grupo 3		Grupo 4	
50	1230	32	51	19	135	E8	164
E1	1	E1	27	14	72	50	33
		F8	27	15	43		
		G0	19	17	43		
		30	7	16	33		
		17	5	13	12		
		50	5	11	10		
		37	2	E1	6		
		16	1	29	4		
				24	3		
				20	2		

Tabla 65. Familias de aceros, frente a la dureza de los mismos.

También se observa que la dureza es un factor clave en el proceso de clasificación ya que:

- En el grupo 1 están la mayoría de las bobinas con dureza 50.
- En el grupo 3, están las bobinas con dureza baja.

Aunque existen durezas que aparecen en varios grupos.

```
# Determinamos el código de las bobinas que pertenecen a los aceros
# B011F97, D012F55, D071F55, X100G99
BOBACERRAR <- c("B011F97", "D012F55", "D071F55", "X100G99")
#BOBACERRAR <- c("B100F55","B105F55","B102G33","B102G55","B100B95")

# Extraemos los datos de las bobinas existentes con tipo de acero buscado
INDICEDATB <- DATBOBINAS$CLASACERO %in% BOBACERRAR
GRUPBBA <- DATBOBINAS[INDICEDATB,1]

# Detectamos la posición de las bobinas pertenecientes a esos aceros
INDICEDATBIG <- MATACERNUEV[,1] %in% GRUPBBA
# Obtenemos el tipo de cada una de ellas
TIPOPUNTD <- match (DATBOBINAS[DATBOBINAS$CODBOBINA %in%
MATACERNUEV[INDICEDATBIG,1],]$CLASACER, BOBACERRAR)+1

# Determinamos el color para cada clase
# B011F97="rojo", D012F55="verde", D071F55="azul", X100G99="azul claro"
# Visualizamos la proyección de las bobinas con los dos ejes principales PCA
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2],pch=1)
points(PCAACERO$rproj[INDICEDATBIG,1], PCAACERO$rproj[INDICEDATBIG,2], col=
TIPOPUNTD, pch=15)
```

Figura 299. Programa para visualizar las bobinas pertenecientes a varios grupos.

Análisis del Grupo Mayoritario

Por último, vamos a analizar la disposición de los principales aceros en la familia mayoritaria (GRUPO A). Estos aceros corresponden a los tipos: B100F55, B105F55, B102G33, B102G55, y B100B95, que en total suman 1.194 bobinas tratadas.

```
# Determinamos el código de las bobinas que pertenecen a los aceros
BOBACERRAR <- c("B100F55","B105F55","B102G33","B102G55","B100B95")

# Extraemos los datos de las bobinas existentes con tipo de acero buscado
INDICEDATB <- DATBOBINAS$CLASACERO %in% BOBACERRAR
GRUPBBA <- DATBOBINAS[INDICEDATB,1]
# Detectamos la posición de las bobinas pertenecientes a esos aceros
INDICEDATBIG <- MATACERNUEV[,1] %in% GRUPBBA
# Determinamos el color para cada clase
TIPOPUNTD <- match (DATBOBINAS[DATBOBINAS$CODBOBINA %in%
MATACERNUEV[INDICEDATBIG,1],]$CLASACER, BOBACERRAR)+1

# Visualizamos la proyección de las bobinas con los dos ejes principales PCA
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2],xlim=c(0,0.14),ylim=c(-
0.05,0.03),pch=1)
points(PCAACERO$rproj[INDICEDATBIG,1], PCAACERO$rproj[INDICEDATBIG,2], col=
TIPOPUNTD, pch=15)
```

Figura 300. Programa que dibuja los 5 grupos principales de aceros de la familia 1.

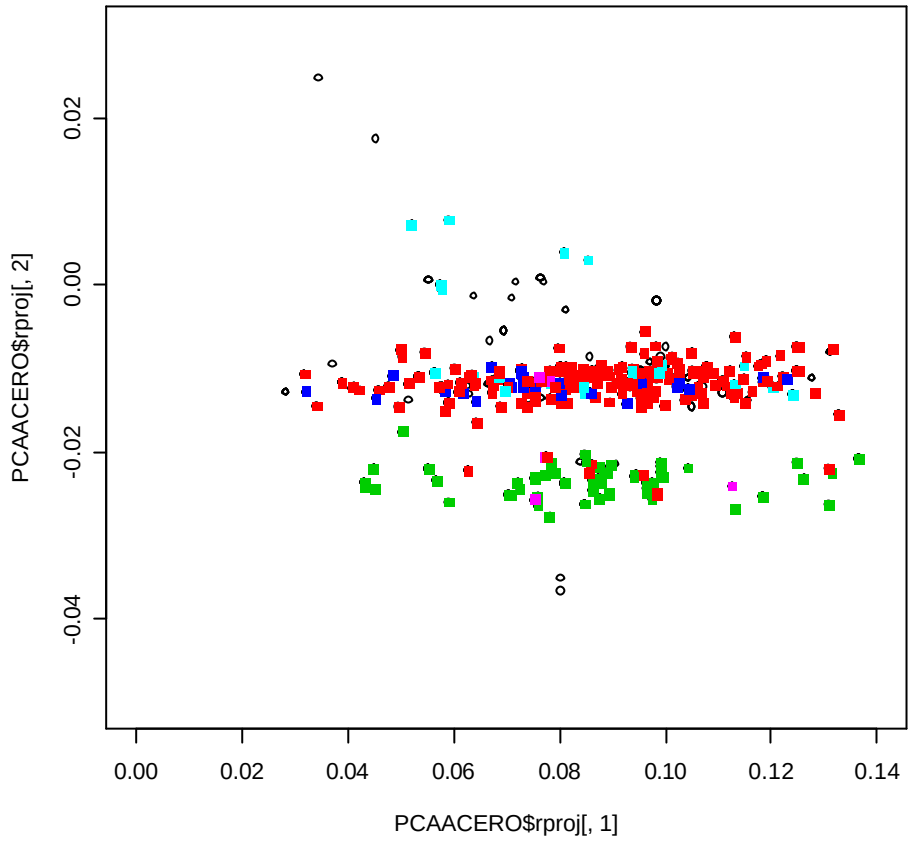


Figura 301. Disposición de los 5 aceros más significativos del GRUPO A: B100F55="rojo", B105F55="verde", B102G33="azul", B102G55="azul claro" y B100B95="magenta".

Donde vemos que casi todas las bobinas de los tipos de aceros, excepto el B105F55, pertenecen al subgrupo superior.

Inicialmente, se tomarán todos estos tipos de aceros como pertenecientes a la familia 1, aunque si se observan dificultades, se estudiará el acero B105F55 de forma diferente a los demás.

6.3.2.4 CONCLUSIONES DEL ESTUDIO DE TIPOS DE BOBINAS

El uso de proyectores (*Sammon* y PCA), algoritmos de clusterizado y métodos de visualización, ha permitido obtener conocimiento valioso del funcionamiento que tiene el sistema frente a los diferentes tipos de aceros de que están compuestas las bobinas tratadas.

Además, hemos podido observar los aceros según su composición, ver las familias en que pueden ser agrupados, comparar estas nuevas familias con los tipos de aceros en los que se clasifican actualmente y, sobretodo, se ha generado un nuevo clasificador de bobinas según la composición metalúrgica de las mismas.

Este nuevo clasificador, basado en proyectores PCA, puede ayudar considerablemente, en las tareas de predicción y planificación del proceso de galvanizado, ya que:

- Visualiza la bobina según su composición frente a los diferentes grupos de bobinas ya procesadas. Lo que permite, de un solo vistazo, ver el grado de pertenencia a una familia u a otra.
- Puede servir, para generar un sistema automático de alarma, que detecte aquellas bobinas cuya composición se escape de unos márgenes preestablecidos.
- Ayuda en reducir el número de familias de aceros.

Por ejemplo, en las figuras siguientes se detecta un error elevado en una de las bobinas que claramente se observa corresponde a un tipo de acero de diferente familia que las anteriores y posteriores.

	CODBOBINA	BOBENT	ESPENT	CLASACERO
2108	23563052	146725	2.000	B100F55
2109	23563053	146647	2.000	B100F55
2110	23563054	146726	2.000	B100F55
2111	23563055	146727	2.000	B100F55
2113	23563056	146728	2.000	B100F55
2114	23563057	147396	1.980	B100F55
2115	**23563058**	**147397**	**2.008**	**B013B55**
2116	23563059	147572	1.510	B100F55
2117	23563060	147485	1.510	B100F55

Figura 302. Bobinas tratadas.

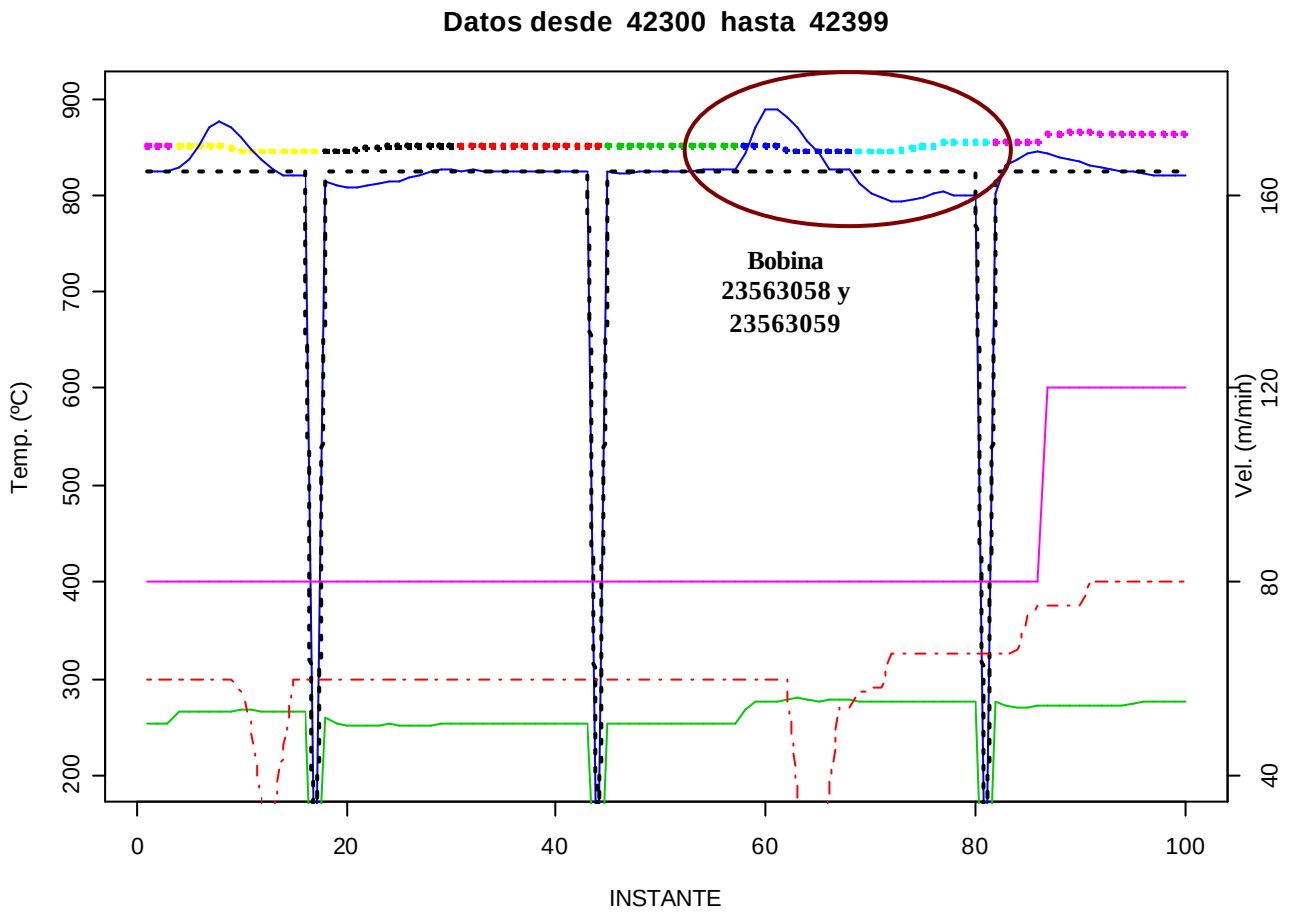


Figura 303. Ejemplo del tratamiento de una serie de bobinas y cómo el error producido en una de ellas es debido a una composición metalúrgica diferente. (bobinas de 23563052 a 23563060).

En la Figura 303, se puede observar cómo el cambio de tipo de acero confunde al sistema de control, que mantenía unas temperaturas de consigna y velocidades constantes, y hace que la temperatura de la banda oscile.

El clasificador de bobinas desarrollado (ver Figura 304), podría advertirnos de la existencia de bobinas anómalas que se salen del grupo que se está tratando en ese momento, lo que ayudaría a detectar cambios de tipo de acero y, por lo tanto, reducir el número de incidencias o errores tales como:

- Roturas de la banda por un cambio de resistencia de la misma.
- Errores altos de temperatura.
- Cambios bruscos que afectan a varias bobinas.
- Etc.

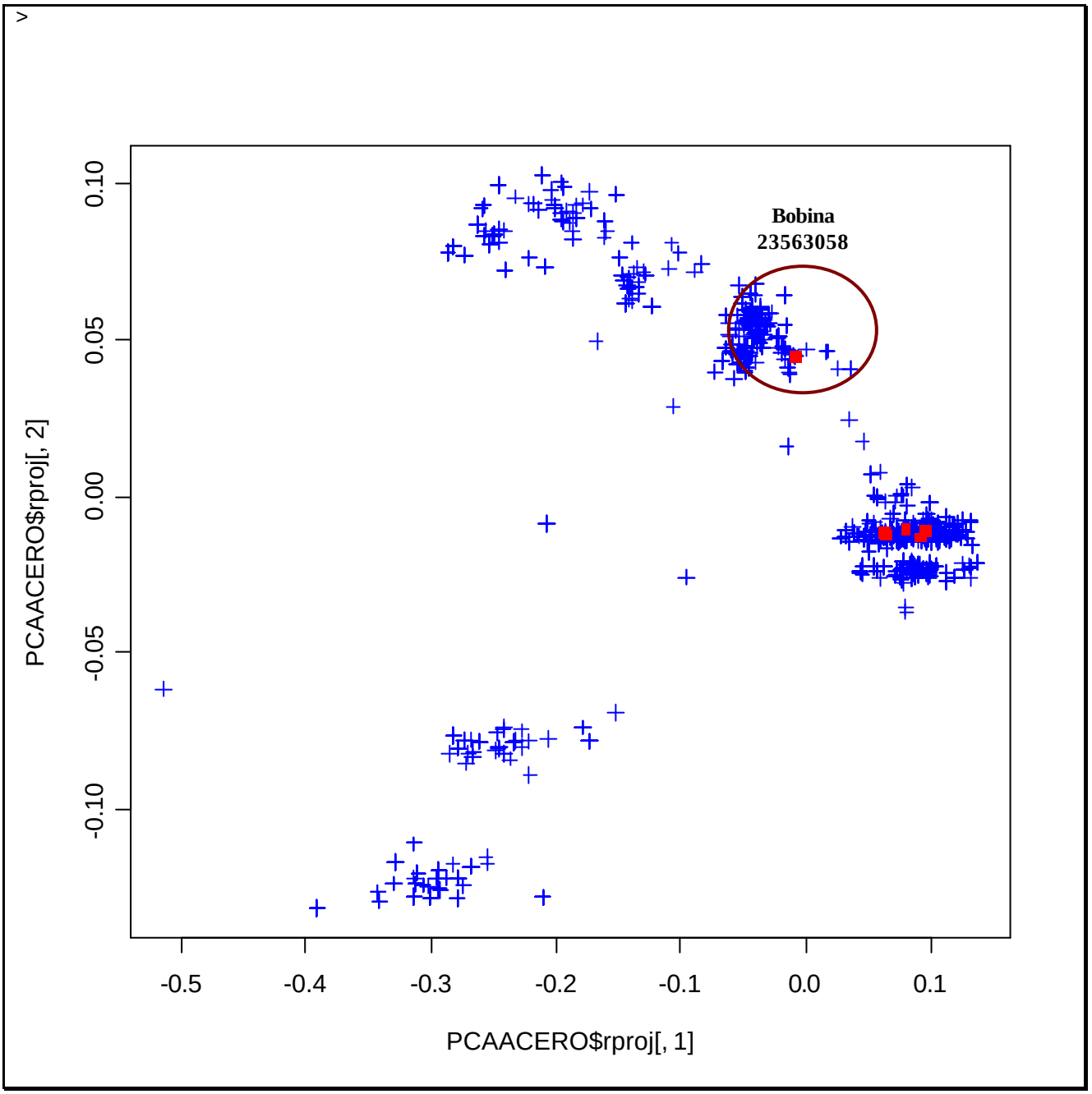


Figura 304. Localización de una bobina anómala dentro de una serie de bobinas tratadas.

6.3.3 ESTUDIO DE LAS VARIABLES MEDIANTE VISUALIZACIÓN EN COORDENADAS PARALELAS

La gráfica de coordenadas paralelas es una herramienta muy útil para la búsqueda de patrones en los datos. En este caso, se utiliza la herramienta visual de distribución libre "*XMDVTOOL*" [XMD02] y para ello, se crea un archivo con las siguientes variables: *VELDIFTOTAL*, *VELMEDTOTAL*, *THF1DIFTOTAL*, *THF1MEDTOTAL*, *STEP1*, *TMPP1DIFTOTAL*, *TMPP1MEDTOTAL*, *SECCION* y *ERRORMEDTOTAL*; con sus nombres simplificados.

El programa que vuelca los datos a un fichero se muestra a continuación:

```
# Cargamos las librerías de análisis multivariante
library(mva)
library(multiv)

# Cargamos las variables siguientes

VELDIFTOTAL <- as.numeric(as.matrix(MATBOBINAS2$VELDIFTOTAL))
VELMEDTOTAL <- as.numeric(as.matrix(MATBOBINAS2$VELMEDTOTAL))
THF1DIFTOTAL <- as.numeric(as.matrix(MATBOBINAS2$THF1DIFTOTAL))
THF1MEDTOTAL <- as.numeric(as.matrix(MATBOBINAS2$THF1MEDTOTAL))
THF3MEDTOTAL <- as.numeric(as.matrix(MATBOBINAS2$THF3MEDTOTAL))
THF5MEDTOTAL <- as.numeric(as.matrix(MATBOBINAS2$THF5MEDTOTAL))
TMPP1DIFTOTAL <- as.numeric(as.matrix(MATBOBINAS2$TMPP1DIFTOTAL))
TMPP1MEDTOTAL <- as.numeric(as.matrix(MATBOBINAS2$TMPP1MEDTOTAL))
TMPP2DIFTOTAL <- as.numeric(as.matrix(MATBOBINAS2$TMPP2DIFTOTAL))
TMPP2MEDTOTAL <- as.numeric(as.matrix(MATBOBINAS2$TMPP2MEDTOTAL))
ANCHO <- round(as.numeric(as.matrix(DATBOBINAS$ANCHO)))
ESPENT <- round(100*as.numeric(as.matrix(DATBOBINAS$ESPENT)))
ERRORMEDTOTALABS <- as.numeric(as.matrix(MATBOBINAS2$ERRORMEDTOTALABS))

#Creamos dos nuevas variables
STEP1 <- abs(THF1MEDTOTAL- THF3MEDTOTAL)
STEP2 <- abs(THF1MEDTOTAL- THF5MEDTOTAL)

# Creamos la Matriz
MATJ48 <- cbind(VELDIFTOTAL, VELMEDTOTAL, THF1DIFTOTAL , THF1MEDTOTAL, STEP1,
TMPP1DIFTOTAL, TMPP1MEDTOTAL, ANCHO, ESPENT, ERRORMEDTOTALABS)
dim(MATJ48)

# Pasamos la matriz 'MATJ48' a un archivo de texto
write.table(MATJ48,"C:\\temp\\ERRORES_GRAFICAPARALELOS2.txt",quote=FALSE,sep=","
,row.names=FALSE,col.names=FALSE)
```

Figura 305. Programa que vuelca las variables a un archivo.

Después se modifica el archivo generado añadiéndole una cabecera, tal y como indica la figura siguiente.

```
10 1979
VELDIF
VELMED
THF1DIF
THF1MED
STEP1
TMP1DIF
TMP1MED
ANCHO
ESPENT
ERRORABS
0. 123. 4
0. 123. 4
0. 99. 4
702. 877. 4
0. 63. 4
0. 155. 4
200. 300. 4
750. 1525. 4
42. 202. 4
0. 507. 4
1 1 3 770 30 4 212 1250 58 4
0 0 9 772 29 8 210 1250 58 2
61 61 32 778 30 8 211 1250 58 9
0 0 73 758 30 7 221 1250 58 7
35 35 73 752 31 6 223 1250 58 4
20 20 20 766 30 12 213 1250 58 4
33 33 41 799 30 14 216 1250 58 11
8 8 3 812 30 12 244 1350 68 5
10 10 7 808 30 3 248 1350 68 5
0 0 1 804 30 5 250 1350 68 1
0 0 2 803 30 2 249 1350 68 1
...
```

Figura 306. Parte inicial del archivo generado para el programa XMDVTool.

6.3.3.1 INFLUENCIA DE LA VARIABLE THF1MEDTOTAL

En las figuras siguientes, podemos observar en rojo las bobinas con errores absolutos mayores de 30ºC. Claramente, **se puede apreciar que los errores elevados tienen valores altos de temperaturas de zonas de horno y de temperatura de entrada de bobina**, lo que viene a indicar que el error es proporcional a la temperatura de entrada de la banda y de la zona del horno.

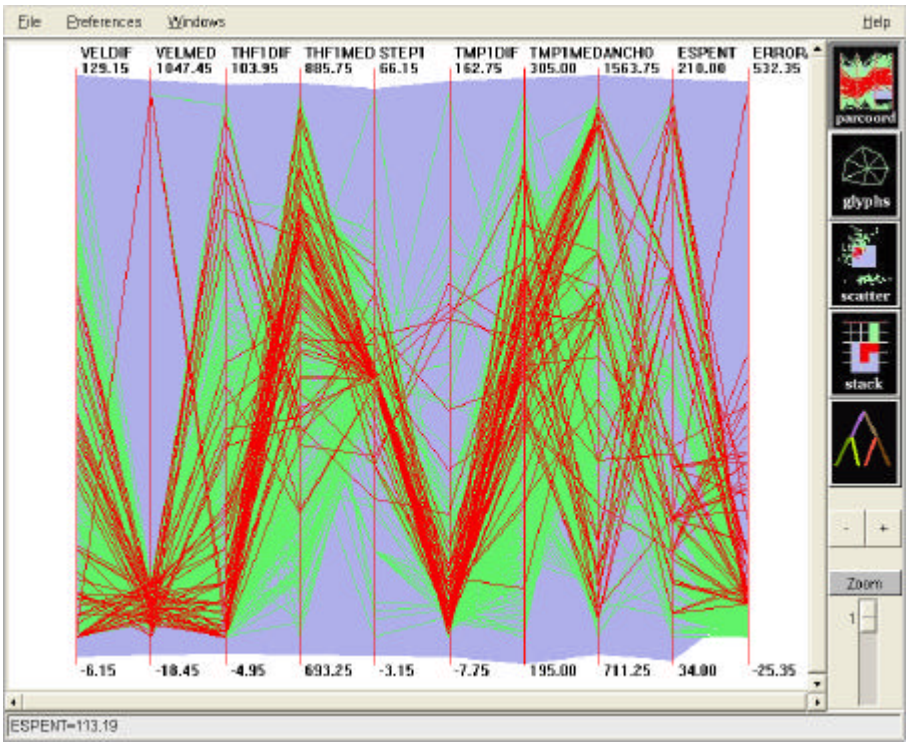


Figura 307. Gráfico en coordenadas paralelas de las bobinas con errores mayores de 30ºC (en rojo).

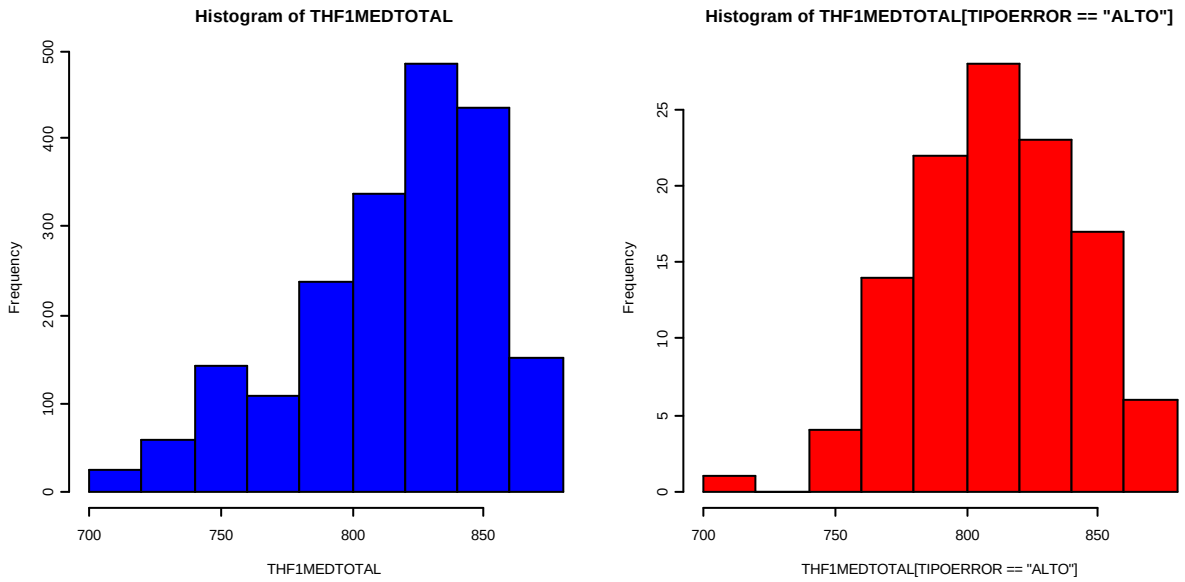


Figura 308. Histogramas de las temp. medias de horno de todas las bobinas (izquierda) y con error ALTO (derecha)

6.3.3.2 INFLUENCIA DE LA VARIABLE THF1DIFTOTAL

Del estudio de las figuras siguientes se pueden extraer algunas nuevas conclusiones.

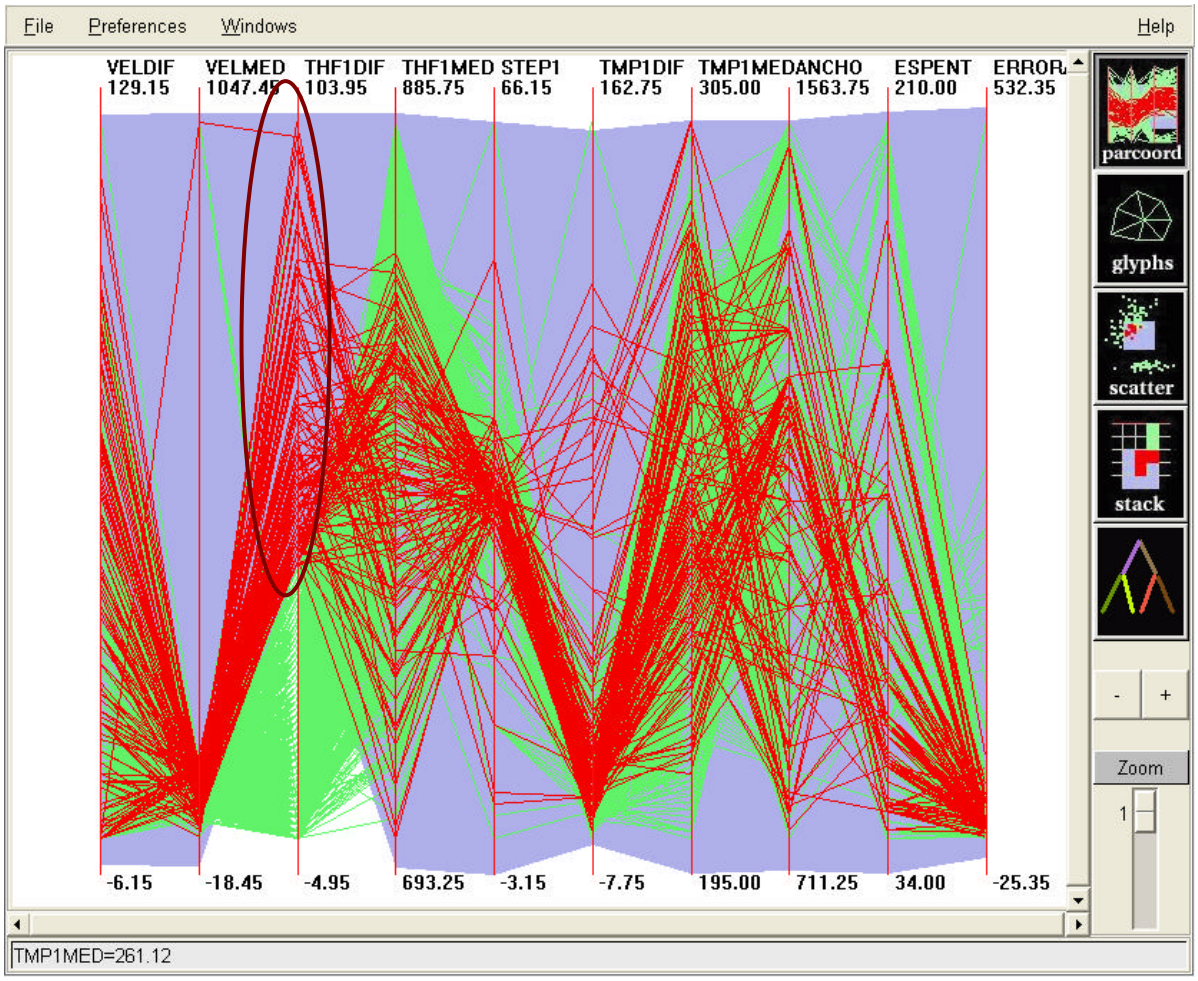


Figura 309. Grupo con THF1DIFTOTAL elevadas.

Lo primero que se puede observar, tal y como muestra la Figura 309, es que existen bastantes bobinas con diferencias en sus curvas de temperatura de zona 1 entre 30ºC y 100ºC. Lo que si se puede apreciar, es que las bobinas que tienen esas diferencias tienen errores finales muy diversos (entre 0 y 50ºC), **lo que indica claramente que la variación de temperaturas de zona no siempre es causa directa del error final de temperatura de banda, aunque puede influir en algunas de ellas.**

6.3.3.3 INFLUENCIA DE LAS VARIABLE ESPENT Y ANCHO

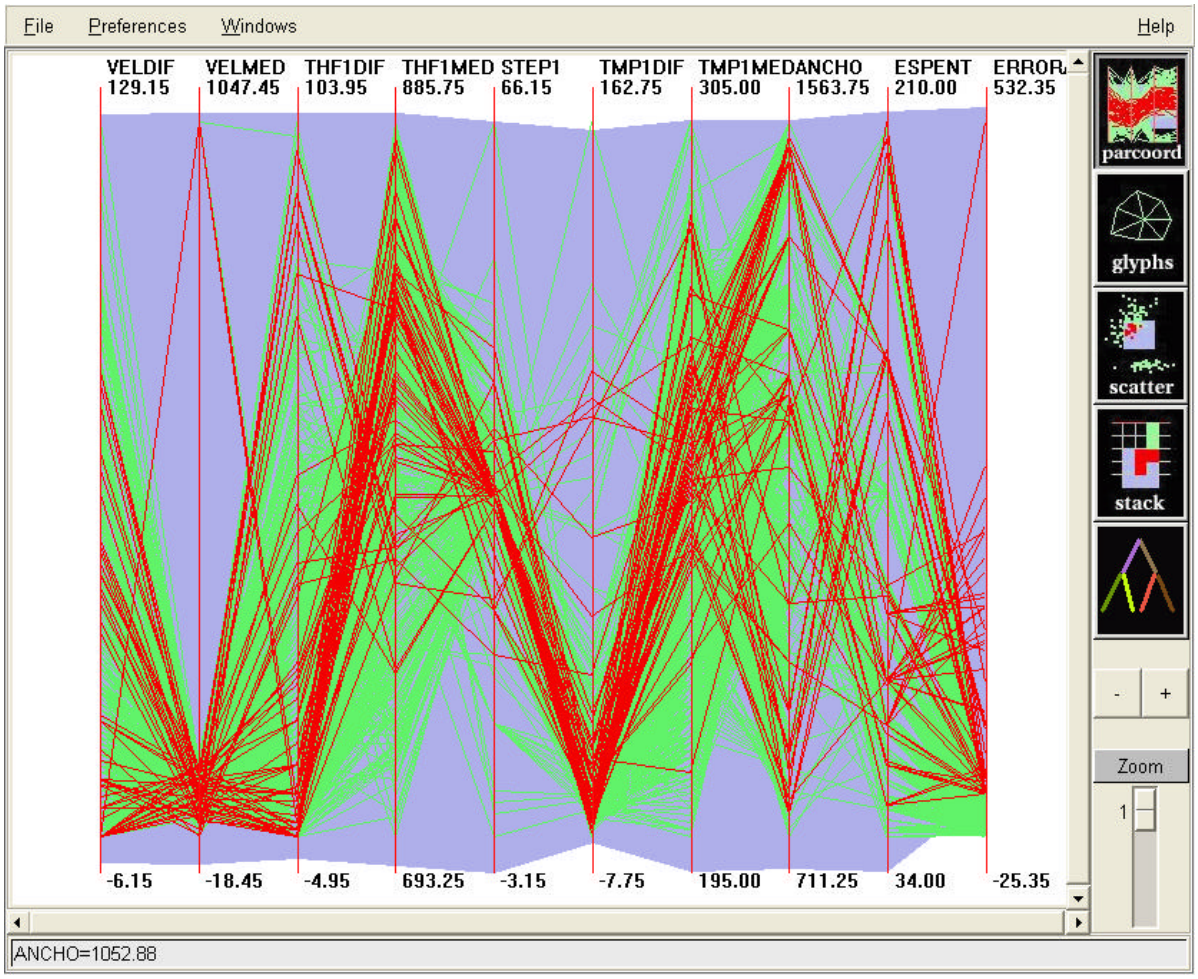


Figura 310. Bobinas con error ALTO frente a la anchura y el espesor de entrada.

También podemos advertir en la Figura 310, que la anchura de la bobina y el espesor de entrada es variable para las bobinas con errores ALTOS. Esto ya se había corroborado con el diagrama de *scatter-plots*, pero era conveniente visualizar estas relaciones con otras herramientas por si podía existir una estructura no lineal que explicara los errores ALTOS.

6.3.3.4 INFLUENCIA DE LA VARIABLE VELDIFTOTAL

Procedemos a estudiar la relación de la variable *VELDIFTOTAL* con los errores ALTOS y BAJOS.

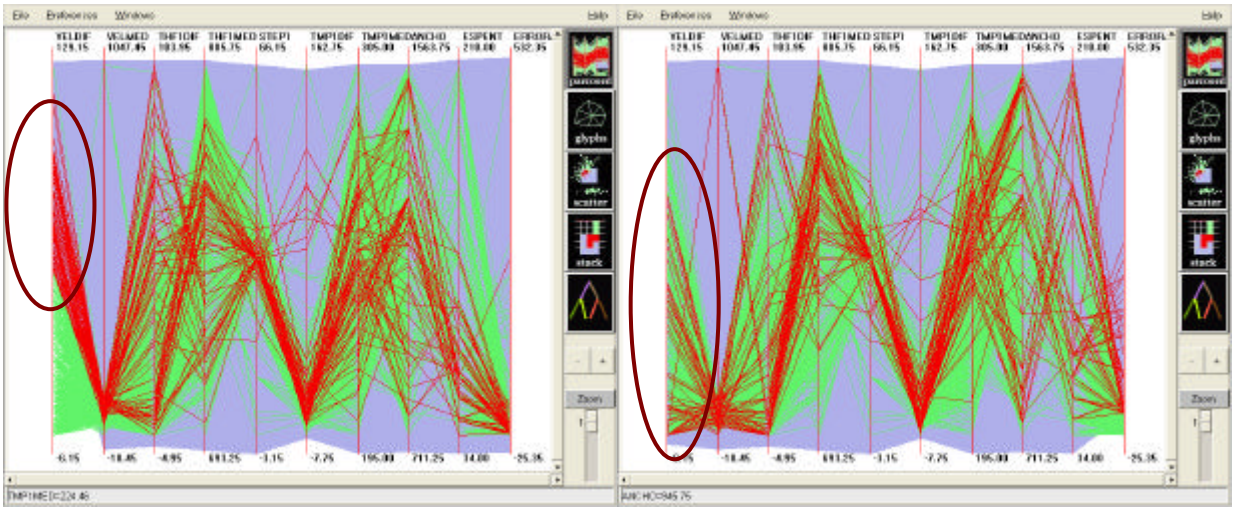


Figura 311. Gráfica donde se muestran las bobinas con "errores" mayores de 40ºC.

Según la gráfica izquierda de la Figura 311, podemos corroborar que multitud de bobinas con altos valores de *VELDIFTOTAL* tienen pequeños errores. Además se ve claramente que no existe relación entre las variaciones bruscas de la temperatura del pirómetro 1 y la de la velocidad.

Por otro lado, en la gráfica derecha de la Figura 311 podemos observar que el rango de la variable *VELDIFTOTAL* se mueve en todo el eje. Esto viene a indicarnos, que **esta variable no siempre puede ser la causa que produzca influencia en el error, aunque puede influir en bobinas posteriores.**

6.3.3.5 CONCLUSIONES DE LOS ANÁLISIS CON VISUALIZACIÓN DE COORDENADAS PARALELAS

En los análisis efectuados hasta ahora, se ha tratado de determinar las causas que producen los "errores" de temperatura de cada bobina a partir de las curvas de las variables más significativas de cada una de ellas.

El análisis efectuado con las herramientas de visualización utilizadas anteriormente ha ayudado a comprender un poco más las "causas" que generan algunos "errores" **aunque estos resultados no han sido concluyentes.**

6.4 CONCLUSIONES FINALES

En este capítulo, se han utilizado diversas técnicas de DM y estadísticas para analizar la base de datos ya preprocesada con la intención de:

- Estudiar la dependencia entre variables.
- Buscar conocimiento que explique el comportamiento del sistema.
- Creación de clasificadores que simplifiquen tareas posteriores y ayuden en la toma de decisiones.

Del análisis de dependencias, tanto con técnicas de estadística multivariante como de las técnicas de DM, se han obtenido las siguientes conclusiones:

- Prácticamente, tres de cada cuatro bobinas han sido tratadas en "modo manual", existiendo aceros que prácticamente no han sido tratados en "modo automático". **En el "modo automático", se asegura un 98,3% de bobinas con errores BAJOS frente al 93,1% del "modo manual" de los diez aceros más comunes. Para todos los aceros, el porcentaje es parecido: 98,2% frente al 92,5% del "modo manual".**
- **Comparando los errores entre los dos modos, se deduce que el grado de eficiencia del "modo automático" es mayor que el "modo manual".**

	MODO MANUAL		MODO AUTOMATICO	
Tipo de Error	Núm.	En %	Núm.	En %.
BAJO <=20ºC	1024	93,1%	521	98,3%
MEDIO (>20ºC & <=50ºC)	52	4,7%	7	1,3%
ALTO (>50ºC)	24	2,2%	2	0,4%

Tabla 66. Número y porcentajes globales de tipos de error medio absoluto para los dos modos.

- Se observa, que el estudio debe realizarse separadamente para cada familia de aceros. El tipo de acero de cada bobina y el modo en que se ha tratado cada una de ellas, **deben ser considerados en los procesos posteriores de modelizado**. Será conveniente, buscar una relación de agrupamiento entre los diferentes tipos de bobinas. Se observa que, de los 10 aceros principales, en 9 el modo automático produce menos errores que el "modo manual".

% ERRORABSMANUAL	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	4,3%	10,1%	4,5%	8,4%	4,4%	21,5%	1,5%	0,0%	5,7%	8,0%
BAJO	95,7%	89,9%	95,5%	91,6%	95,6%	78,5%	98,5%	100,0%	94,3%	92,0%

Tabla 67. Comparación con los tipos de error para el modo manual (en porcentajes relativos).

% ERRORABSAUTOMAT	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	0,0%	1,7%	0,0%	1,1%	2,4%	0,0%	1,1%	0,0%	12,5%	0,0%
BAJO	100,0%	98,3%	100,0%	98,9%	97,6%	100,0%	98,9%	100,0%	87,5%	100,0%

Tabla 68. Comparación con los tipos de error para el modo automático (en porcentajes relativos).

- Los comportamientos dinámicos de las variables de temperatura del horno, de la banda y de la velocidad de la misma, son los que parecen ser causantes de los "errores elevados". **La derivada de las variables parece ser la mejor elección para futuros estudios.** Las variables estáticas como: temperaturas medias, velocidades medias, dimensiones de la banda, etc.; no presentan ninguna causalidad significativa con el "error final". Esto nos permite prever que el estudio y control de las variables del horno debe orientarse hacia una análisis dinámico en modo continuo de las curvas, evitando la focalización del mismo en las bobinas.
- **Las dimensiones de la banda NO presentan correlación lineal con el "error".** Las diferentes dimensiones (ESPESOR, ANCHURA y LONGITUD) son muy dependientes entre sí debido a que proceden de paralelepípedos de fundición de iguales dimensiones, por lo tanto, solo se usará una o dos de ellas.
- **La velocidad de la banda, lógicamente, presenta una correlación significativa con las dimensiones de la misma pero no con el error.**
- Los saltos térmicos entre unas zonas y otras, y las temperaturas de las zonas de la parte de calentamiento del horno, **parecen ser adecuados siempre que la variables mantenga un régimen permanente en cada bobina.**
- **Se observa que el horno tiene capacidad para "llevar" a todas las bobinas a la temperatura de consigna buscada.** Aunque las variaciones de temperatura de la bobina a la entrada, son mantenidas, en cierto modo, a la salida. **Esto indica que el salto térmico, no es capaz de absorber algunos cambios bruscos de temperatura de la banda.** Es decir, se deduce, que el horno **tiene capacidad suficiente para "llevar" a las bobinas a la temperatura esperada, pero en cambio no es tan eficaz, ante las variaciones bruscas de temperatura o velocidad.**

Por otro lado, del análisis de los datos mediante técnicas de minería de datos, también se han extraído otras conclusiones:

- **Los cambios de anchura o de espesor son alguna de las causas de los errores elevados.**

SUCESO	TIPO DE ERROR	
	ALTOS (>30ºC)	BAJOS (<=30ºC)
Cambio de Acero	22,4%	20,7%
Cambio de Anchura	38,8%	22,9%
Cambio de Espesor	30,6%	19,3%

Tabla 69. Porcentajes de errores ALTOS y BAJOS para cambios de acero, anchura o espesor de bobinas.

- Los cambios de espesor y anchura de banda pueden ser parte de las causas de los errores ALTOS en bobinas, pero se ha visto que en "modo manual", un 66,67% de las bobinas con errores ALTOS, NO se producen por cambios de anchura, de espesor o tipo de acero de la bobina. Esto **parece indicar que los errores altos en "el modo manual" pueden ser debidos a que no se reacciona a tiempo en el manejo de las curvas de velocidades y temperatura de zona del horno para reducir el error entre la medida de temperatura del pirómetro 2 y la de consigna cuando estos se producen de forma espontánea o, en algunas ocasiones, no se supervisa durante un cierto tiempo.**

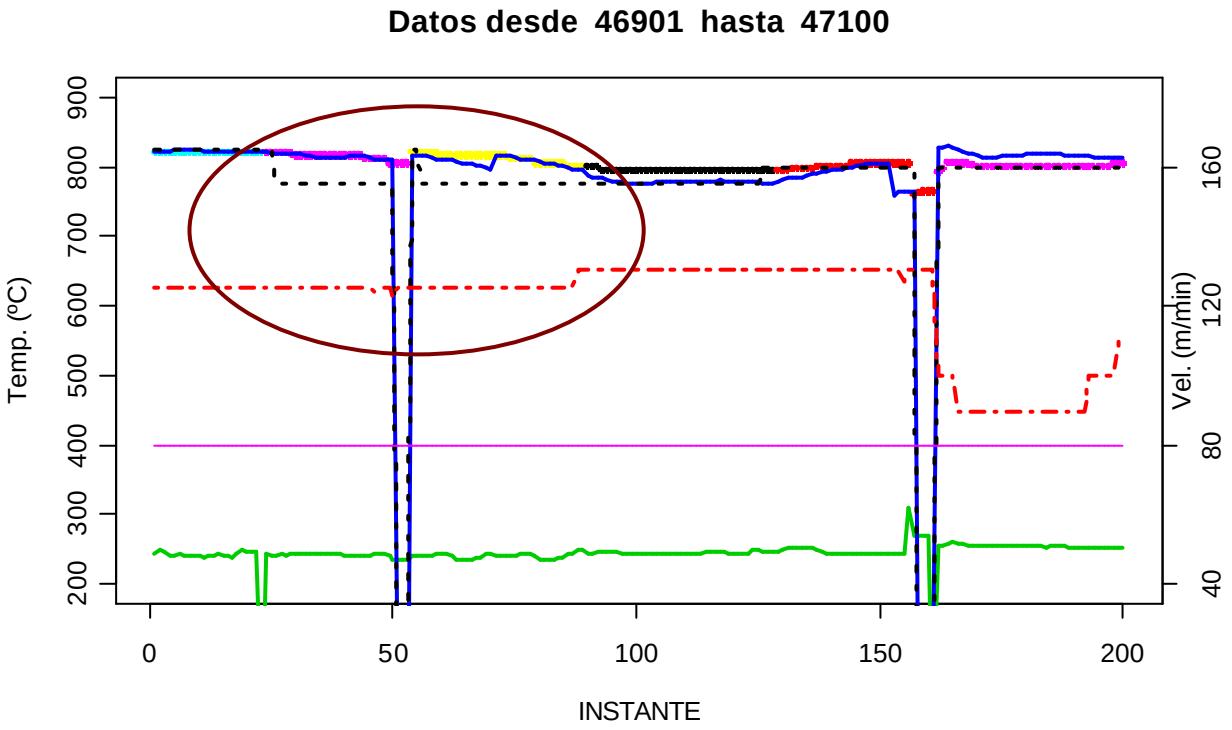


Figura 312. Falta de reacción en "modo manual" para dos bobinas consecutivas.

- El análisis de las bobinas mediante clasificadores generados con diferentes técnicas de DM, no han resultado ser muy precisos, aunque si se ha podido obtener alguna conclusión interesante con el uso de reglas asociativas.
- Los puntos de operación del horno dependen de la dureza del acero.

Por último, el uso de proyectores ha servido para:

- Simplificar el número de familias de aceros según su composición metalúrgica.
- Crear un nuevo clasificador, basado en proyectores PCA, que ayudará en las tareas de predicción y planificación del proceso de galvanizado. Este **clasificador podrá ayudar a detectar, de un solo vistazo, el grado de pertenencia a una familia u a otra.**

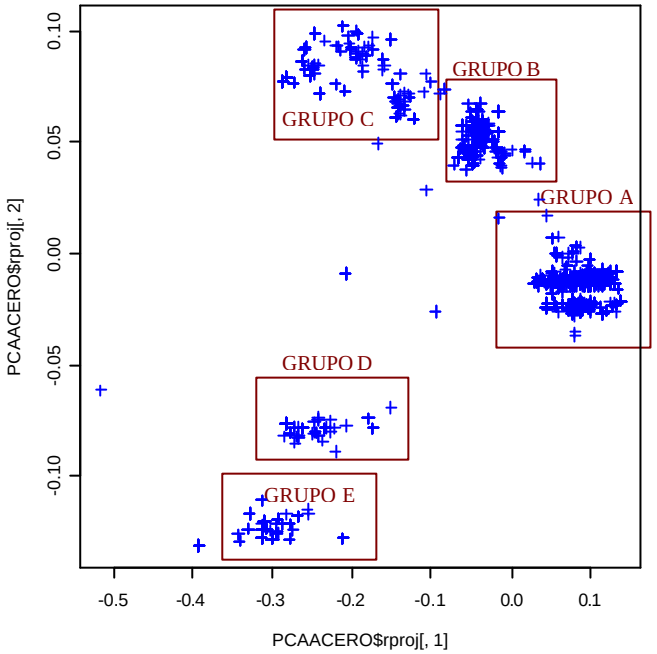


Figura 313. Ejemplo de sistema de supervisión de las bobinas.

- **Desarrollar un sistema automático de alarma, que detecte aquellas bobinas cuya composición metalúrgica se escape de unos márgenes preestablecidos. Lo que servirá para detectar anomalías en la composición metalúrgica de la banda que pueda ser causa de:**
  - **Roturas en la banda.**
  - **Paradas imprevistas debidas a fallos de otro tipo.**
  - **Pérdidas del punto de operación óptimo.**

En el capítulo siguiente, se tratará de generar un nuevo sistema de supervisión mediante el desarrollo de sensores-software y un modelo de control automático que intente mejorar los resultados actuales.

CAPÍTULO 7

MODELIZACIÓN PARA EL CONTROL Y SUPERVISIÓN DEL HORNO EN LA ZONA DE CALENTAMIENTO

7.1 INTRODUCCIÓN

Siguiendo con la aplicación de la metodología *CRISP-DM* [CRI00] y, una vez preparados y analizados los datos, procedemos a la etapa de modelizado de la fase de calentamiento de la banda para la mejora del control y supervisión del mismo.

Los objetivos principales son dos:

- **La generación de un modelo No-Lineal, basado en redes neuronales, que mejore el ajuste de las temperaturas de zonas del horno y velocidad de la banda**, reduciendo el error existente entre la temperatura de la banda, esperada y real, a la salida de la zona de calentamiento del horno.
- **El desarrollo de *Sensores-Software*** para la supervisión visual del punto de operación del horno.

Para ello, y según esta metodología, se continúa con la fase de modelado. En ésta, se seleccionarán las técnicas de modelizado más adecuadas, se generarán los modelos y se validarán los mismos.

7.2 FASE IV: MODELADO

En esta fase del proceso *CRISP-DM*, se seleccionan las técnicas para generar los modelos con los datos que se han preparado anteriormente. Es probable que sea necesario reajustar los mismos para que puedan ser manejados correctamente.

Las tareas propias de esta fase, constan de los siguientes pasos:

- Selección de las técnicas de modelado.
- Diseño del método de evaluación.
- Generación del modelo.
- Evaluación del modelo.

7.2.1 SELECCIÓN DE LAS TÉCNICAS DE MODELADO

En este primer paso [CRI00][ABA01], se deben seleccionar las técnicas que se utilizarán para el modelado en función de:

- El tipo de problema.
- Los datos a manejar.
- El tiempo necesario para obtener el modelo.
- El conocimiento de la técnica.
- Las herramientas de que se disponen.

Las técnicas seleccionadas inicialmente se deberán ordenar según el grado de cumplimiento de los puntos anteriores, comprobando que los datos que se necesitan están disponibles tanto en calidad y cantidad, como en un formato correcto.

7.2.2 DISEÑO DEL MÉTODO DE EVALUACIÓN

Antes de comenzar con el proceso de construcción del modelo, será necesario definir el mecanismo de validación del mismo.

Para ello se determinará:

- Función que determine el error cometido y umbral de calidad estimado: error cuadrático medio, FPE, etc.
- Tipo de método de evaluación de los modelos generados: división de dos grupos para entrenamiento y validación, validación cruzada, etc.
- Si es necesario, el tamaño de los grupos de validación y el tipo de entrenamiento.

7.2.3 GENERACIÓN DEL MODELO

Una vez que se han precisado todos los detalles relativos a la generación de los modelos, se aplicarán las técnicas de modelado a los datos preparados.

Se describirán:

- Los parámetros utilizados. Incluyendo su importancia, influencia en el resultado del modelo y valores iniciales asignados.
- Modelos obtenidos, gráficas de entrenamiento y validación, resultados numéricos.
- Descripción detallada del modelo, de los parámetros del mismo, de la exactitud y sensibilidad, y la forma de implementarlo.

7.2.4 EVALUACIÓN DEL MODELO

Por último, se evaluará la eficiencia de cada uno de los modelos, analizando:

- Grado de calidad de la predicción.
- Si está sobreentrenado o no.
- Potencial de predicción de información no conocida.
- Velocidad de proceso.
- Evolución de los resultados con nuevos datos.
- Influencia de cada uno de los parámetros en el modelo.
- Grado de generalización de lo explicado.

Si los resultados no son adecuados, se repetirán todos los pasos hasta llegar a una solución lo más óptima posible.

7.3 APLICACIÓN PRÁCTICA DE LA FASE IV DE LA METODOLOGÍA CRISP-DM

Para cumplir con los objetivos planteados, y después de las conclusiones desarrolladas en el capítulo anterior, **se decide realizar el proceso de modelizado por separado para cada una de las nuevas familias de bobinas que se obtienen con el nuevo clasificador desarrollado.**

En este caso, el estudio se centrará en los GRUPOS A y B, debido a que son los dos conjuntos con más cantidad de bobinas procesadas: 1.988 de 2.420 (82%), aunque puede ser desarrollado para cualquiera de los otros grupos, si se suministra una mayor cantidad de datos que los describan.

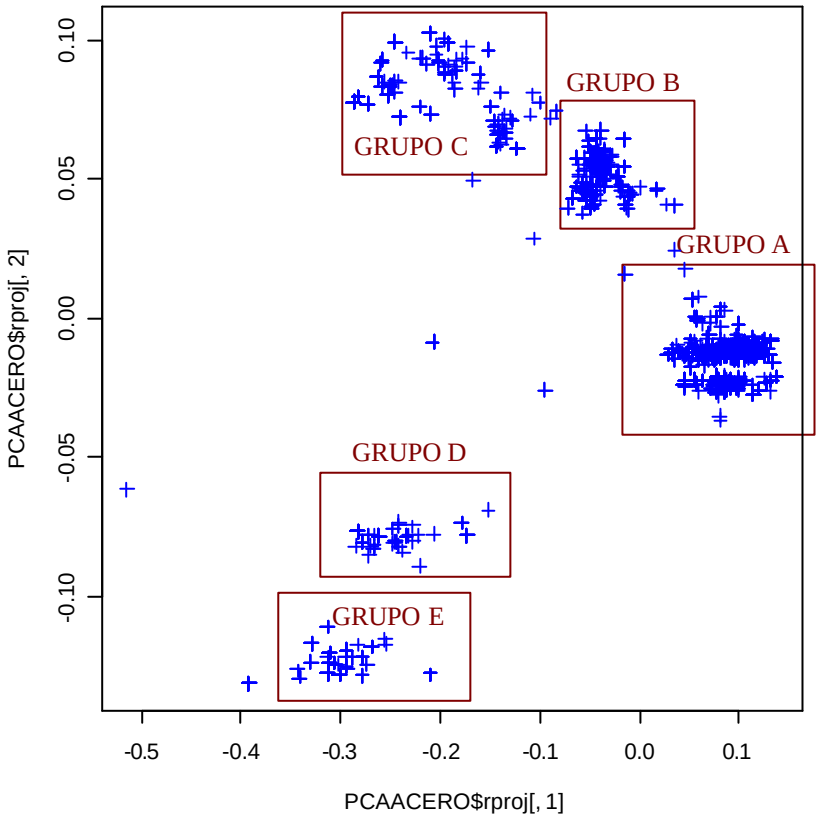


Figura 314. Grupos de bobinas según la composición del acero.

Para ello, el modelado se centrará en dos objetivos fundamentales:

- El desarrollo de un sensor-software que permita:
  - Visualizar el punto de operación del horno frente a diferentes dimensiones de bobinas.
  - Identificar zonas pertenecientes a regímenes permanentes y transitorios.
  - Detectar movimientos y tendencias hacia zonas de errores elevados para supervisión visual por parte del operario.
- La creación de varios modelos no lineales basados en redes neuronales que "expliquen" el comportamiento de las bobinas de cada familia a los cambios de temperaturas y velocidades de consigna y las mejores consignas, para:
  - Simular con anterioridad el comportamiento de la banda ante ciertas consignas del horno.
  - Predecir las consignas más adecuadas para el tratamiento de las bobinas con dimensiones o familias diferentes.
  - Crear nuevos métodos de control automático que reduzcan el "error" promedio.

En los puntos siguientes, se desarrollarán con detalle los pasos realizados para cada uno de estos dos objetivos.

7.3.1 DESARROLLO DE UN SENSOR-SOFTWARE PARA SUPERVISIÓN DEL PUNTO DE OPERACIÓN DEL HORNO

7.3.1.1 METODOLOGÍA

OBJETIVO

Como se ha explicado anteriormente, el objetivo consiste en desarrollar un sensor-software que nos indique el punto de operación actual del horno, la tendencia y zonas de regímenes permanente y transitorios.

SELECCIÓN DE LA TÉCNICA A UTILIZAR

Para el desarrollo de este sensor-software, se va a utilizar una metodología similar a la empleada para modelizar los tipos de aceros según su composición metalúrgica (capítulo anterior). Es decir, se hará uso de proyectores para situar gráficamente el punto de operación en un espacio bidimensional y se delimitarán visualmente las zonas de operación [MIS02][WAN02].

BASE DE DATOS A UTILIZAR

Se pretende modelizar el comportamiento en régimen permanente de las bobinas del GRUPO A, que tengan curvas de velocidad, temperatura de pirómetros 1 y 2, error absoluto y temperaturas de zona HORIZONTALES y con error medio absoluto BAJO. Es decir, aquellas bobinas que han presentado un comportamiento que definíamos como "BUENO" y cuyas temperaturas y velocidades de consigna han permanecido estables durante toda la bobina.

VARIABLES A UTILIZAR

Primeramente, se seleccionarán las variables siguientes:

- *ANCHO*: Ancho de la bobina.
- *ESPENT*: Espesor de la banda para esa bobina.
- *TMPP1VALMED*: Temperatura de entrada de esa bobina.

que definen el tipo de bobina.

Y:

- *THF1VALMED*: Temperatura media de zona 1.
- *THF3VALMED*: Temperatura media de zona 3.
- *THF5VALMED*: Temperatura media de zona 5.

- *TMPP2CNGMEDTOTAL*: Temperatura media de consigna.
- *VELMEDTOTAL*: Velocidad media de la banda en esa bobina.

Solamente para las bobinas del GRUPO A y que tengan todas sus curvas HORIZONTALES y un error medio absoluto < 20ºC.

CRITERIOS DE VALIDACIÓN

Se seleccionarán aleatoriamente, un 80% de esas bobinas para el modelado y un 20% para la validación.

También se observará el movimiento del punto de operación para bobinas con errores ALTOS y se intentará detectar las zonas de régimen permanente con error BAJO frente a zonas transitorias o de errores ALTOS.

7.3.1.2 PREPARACIÓN DE LOS DATOS

Lo primero que hacemos, es preparar las nuevas variables y adaptar la base de datos.

```
################################################
# Adaptamos la base de datos para el modelizado #
################################################

# Cargamos las librerías de análisis multivariante
library(mva)
library(multiv)
library(cluster)

# Cargamos las matrices con los datos de las 2.628 bobinas
library(RODBC)
canal <- odbcConnect("aceralia2003","","","localhost");

#############################################################
# Obtenemos los nuevos datos dinámicos de la tabla T100cal #
#############################################################

T100CALB2 <- sqlQuery(canal, "SELECT CODBOBINA, THF3VALCNG as THC3, THF5VALCNG
as THC5 FROM T100cal");

LISTABOBINASVELBUENAS <- unique(MATBOBINAS[,1])

#Eliminamos las bobinas que no están en la lista de las otras bases de datos
POSLISTA <- T100CALB2$CODBOBINA %in% LISTABOBINASVELBUENAS
T100CALB2 <- T100CALB2[POSLISTA,]
# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==unique(T100CALB2$CODBOBINA))
TRUE
1979
# Guardamos en MATDINAMIC2 las dos nuevas variables
attach(T100CALB2)
MATDINAMIC2 <- data.frame(cbind(MATDINAMIC, THC3, THC5))
```

```
################################################
# Creamos las variables TH3VALMED y THF5VALMED #
################################################

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC2$THC3>100
MATSINRUIDO <- MATDINAMIC2[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979
# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(MATSINRUIDO$THC3, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos las variables finales
THF3MEDTOTAL <- round(VALMEAN)

# Eliminamos los valores debidos a fallos de adquisición
LISTASIN <- MATDINAMIC2$THC5>100
MATSINRUIDO <- MATDINAMIC2[LISTASIN,]

# Obtenemos una nueva lista de bobinas
LISTASINRUIDO <- unique(MATSINRUIDO$CODBOBINA)

# Verificamos que la lista de esta tabla junto con la de la tabla
# anterior son iguales (Todos tienen que ser TRUE)
table(LISTABOBINASVELBUENAS==LISTASINRUIDO)
TRUE
1979
# Obtenemos el valor medio de la consigna de temperatura de cada bobina
VALMEAN <- tapply(MATSINRUIDO$THC5, MATSINRUIDO$CODBOBINA,mean)

# Obtenemos las variables finales
THF5MEDTOTAL <- round(VALMEAN)

# Intentamos con 'clara', buscar 4 familias de bobinas
CLARAX <- clara(as.matrix(MATACERNUEV[,3:17]),4)
COLORCLUST <- CLARAX$clustering
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2], col=COLORCLUST,pch=3)

# Añadimos el tipo de subfamilia
MATACERESTU <- CLARAX$clustering

#Buscamos el tipo para cada código de bobina
CODTIPOBOB <- match(DATBOBINAS$CODBOBINA, MATACERNUEV[,1])
FAMILIABOB <- MATACERESTU[CODTIPOBOB]
FAMILIABOB[is.na(FAMILIABOB)] <- 99

# Guardamos en una nueva matriz las variables creadas
MATBOBINAS2 <- data.frame(cbind(as.matrix(MATBOBINAS), THF3MEDTOTAL,
THF5MEDTOTAL, FAMILIABOB))
save(MATBOBINAS2,MATDINAMIC2,DATBOBINAS,file="DatosModelo.RData")
```

Figura 315. Creación de las nuevas variables para el modelizado.

7.3.1.3 PROYECCIÓN SAMMON DE LOS PUNTOS DE OPERACIÓN DE LAS BOBINAS DEL GRUPO-A

Ahora obtenemos la proyección de los puntos de operación de las bobinas con errores bajos, pertenecientes al GRUPO-A.

```
# Obtenemos las bobinas de la familia GRUPO-A
#INDGRUPOAESTAC <- MATBOBINAS2$FAMILIA==1 & MATBOBINAS2$TIPOCURVATHF1=="H" &
#MATBOBINAS2$TIPOCURVATMPP1=="H" & MATBOBINAS2$TIPOCURVATMPP2CNG=="H" &
#MATBOBINAS2$TIPOCURVAERROR=="H" &  MATBOBINAS2$TIPOCURVAVEL=="H"

INDGRUPOAESTAC <- MATBOBINAS2$FAMILIA==1

#Creamos una matriz con las variables a modelizar
CODBOBMATSAM3 <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$CODBOBINA))
ANCHOMATSAM3 <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ANCHO))
ESPENTMATSAM3 <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ESPENT))
TMPP1MATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$TMPP1MEDTOTAL))
THF1MATSAM3 <- as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$THF1MEDTOTAL))
THF3MATSAM3 <- as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$THF3MEDTOTAL))
THF5MATSAM3 <- as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$THF5MEDTOTAL))
TMPP2CNGMATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$TMPP2CNGMEDTOTAL))
VELMATSAM3 <- as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$VELMEDTOTAL))
ERRORSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$ERRORMEDTOTALABS))

MATSAM3 <- cbind(CODBOBMATSAM3, ANCHOMATSAM3, ESPENTMATSAM3, TMPP1MATSAM3,
THF1MATSAM3, THF3MATSAM3, THF5MATSAM3, TMPP2CNGMATSAM3, VELMATSAM3, ERRORSAM3)

# Normalizamos los datos
MATSAMNORM <- cbind(MATSAM3[,1], (MATSAM3[,2]-
min(MATSAM3[,2]))/(max(MATSAM3[,2])- min(MATSAM3[,2])), (MATSAM3[,3]-
min(MATSAM3[,3]))/(max(MATSAM3[,3])- min(MATSAM3[,3])), (MATSAM3[,4]-
min(MATSAM3[,4]))/(max(MATSAM3[,4])- min(MATSAM3[,4])), (MATSAM3[,5]-
min(MATSAM3[,5]))/(max(MATSAM3[,5])- min(MATSAM3[,5])), (MATSAM3[,6]-
min(MATSAM3[,6]))/(max(MATSAM3[,6])- min(MATSAM3[,6])), (MATSAM3[,7]-
min(MATSAM3[,7]))/(max(MATSAM3[,7])- min(MATSAM3[,7])), (MATSAM3[,8]-
min(MATSAM3[,8]))/(max(MATSAM3[,8])- min(MATSAM3[,8])), (MATSAM3[,9]-
min(MATSAM3[,9]))/(max(MATSAM3[,9])- min(MATSAM3[,9])))

# Calculamos el sammon observaciones obtenidas de la matriz MATSAM3
MATSAM <- sammon(as.matrix(MATSAMNORM[,2:9]), tol=0.031, maxit=1000,
diagnostics=TRUE)
# Coloreamos las bobinas con diferentes espesores
COLORERRORALT <- rep(1,length(INDGRUPOAESTAC))
COLORERRORALT[MATSAM3[,3]<0.7] <- 2
COLORERRORALT[MATSAM3[,3]>=0.7 & MATSAM3[,3]<1] <- 3
COLORERRORALT[MATSAM3[,3]>=1] <- 4
PCHTIPO <- rep(3,length(INDGRUPOAESTAC))
PCHTIPO[MATSAM3[,10]>20] <- 19

plot(MATSAM$rproj[,1], MATSAM$rproj[,2],col=COLORERRORALT,pch=PCHTIPO)
```

Figura 316. Programa que genera una proyección Sammon de los puntos de operación.

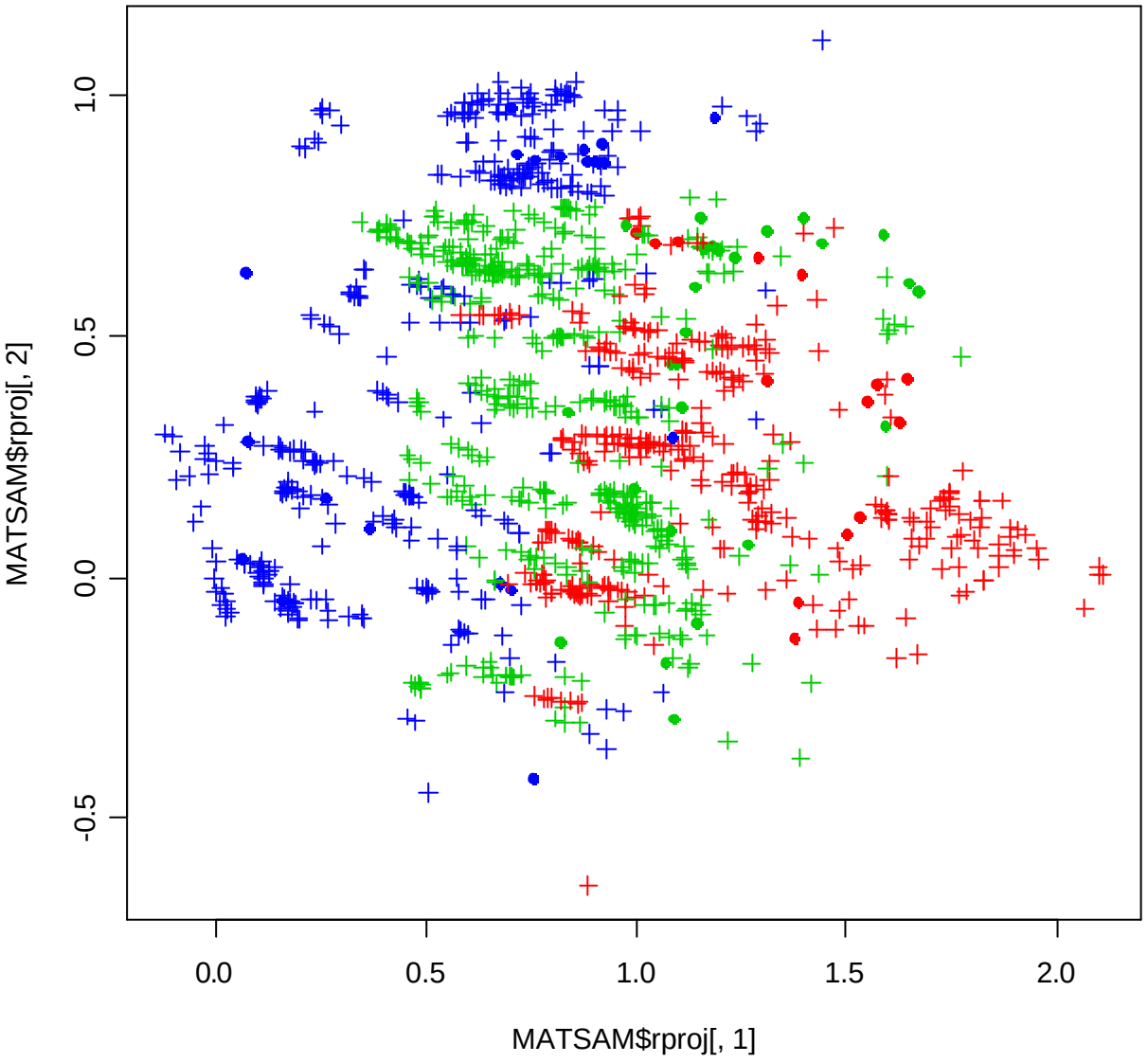


Figura 317. Proyección Sammon de los puntos de operación del GRUPO A para tres familias de espesores de bobinas: (ESPENT < 0.7 mm.)='rojo', (ESPENT >=0.7 & ESPENT <1 mm.)='verde', (ESPENT >= 1 mm.)='azul'.

En la Figura 317 se muestran los puntos de operación para el GRUPO-A de bobinas para tres familias de espesores de bobinas. También se muestran con puntos gruesos las bobinas con errores ALTOS frente a las demás bobinas.

7.3.1.4 PROYECCIÓN PCA DE LOS PUNTOS DE OPERACIÓN EN RÉGIMEN PERMANENTE

Continuando con la misma metodología utilizada anteriormente, procedemos a obtener los ejes PCA de las bobinas del GRUPO-A, que están en régimen permanente.

```
# Obtenemos las bobinas de la familia GRUPO-A
INDGRUPOAESTAC <- MATBOBINAS2$FAMILIA==1 & MATBOBINAS2$TIPOCURVATHF1=="H" &
MATBOBINAS2$TIPOCURVATMPP1=="H" & MATBOBINAS2$TIPOCURVATMPP2CNG=="H" &
MATBOBINAS2$TIPOCURVAERROR=="H" &  MATBOBINAS2$TIPOCURVAVEL=="H"
#Creamos una matriz con las variables a modelizar
#Escalamos los datos
CODBOBMATSAM3 <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$CODBOBINA))
ANCHOMATSAM3 <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ANCHO))/2000
ESPENTMATSAM3 <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ESPENT))/2.5
TMPP1MATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$TMPP1MEDTOTAL))/350
THF1MATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$THF1MEDTOTAL))/950
THF3MATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$THF3MEDTOTAL))/950
THF5MATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$THF5MEDTOTAL))/950
TMPP2CNGMATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$TMPP2CNGMEDTOTAL))/950
VELMATSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$VELMEDTOTAL))/200
ERRORSAM3 <-
as.numeric(as.matrix(MATBOBINAS2[INDGRUPOAESTAC,]$ERRORMEDTOTALABS))
MATSAM3 <- cbind(CODBOBMATSAM3, ANCHOMATSAM3, ESPENTMATSAM3, TMPP1MATSAM3,
THF1MATSAM3, THF3MATSAM3, THF5MATSAM3, TMPP2CNGMATSAM3, VELMATSAM3, ERRORSAM3)
# Cargamos la librería fdim
library(fdim)
# Calculamos la Dimensión Fractal
df <- fdim(MATSAM3[,2:9],q=0,Alpha=0.2,PlotF=TRUE)
print(df$fdim)
      X1
1.318207
# Obtenemos la proyección PCA
PCAPUNTOS <- pca(MATSAM3[,2:9],method=2)
# Vemos el grado de información de cada eje
PCAPUNTOS$evals/sum(PCAPUNTOS$evals)
[1] 0.6368717039 0.2510465555 0.0626472811 0.0344081151 0.0126651884
[6] 0.0013966956 0.0009644605

# Determinamos el tanto por ciento de información de los dos ejes principales
# (los dos primeros abarcan el 88,8% de la varianza)
J <- PCAPUNTOS$evals/sum(PCAPUNTOS$evals)
sum(J[1:2])
[1] 0.8879183

# Visualizamos los dos ejes de componentes principales
PCAPUNTOS$evecs[,1:2]
                     Comp1       Comp2
ANCHOMATSAM3     0.01208078  0.97447086
ESPENTMATSAM3   -0.80097162 -0.08692325
TMPP1MATSAM3    -0.11092694 -0.07876852
```

```
THF1MATSAM3     -0.08348075 -0.04706218
THF3MATSAM3     -0.08849894 -0.03675192
THF5MATSAM3     -0.08647533 -0.03340996
TMPP2CNGMATSAM3 -0.01414533  0.03514935
VELMATSAM3       0.56878340 -0.17529733

# Vemos el grado de aportación de cada variable en cada eje
round(abs(PCAPUNTOS$evecs[,1]*100/sum(abs(PCAPUNTOS$evecs[,1]))))
   ANCHOMATSAM3   ESPENTMATSAM3    TMPP1MATSAM3     THF1MATSAM3
              1              45               6               5
    THF3MATSAM3     THF5MATSAM3 TMPP2CNGMATSAM3      VELMATSAM3
              5              5                1              32
round(abs(PCAPUNTOS$evecs[,2]*100/sum(abs(PCAPUNTOS$evecs[,2]))))
   ANCHOMATSAM3   ESPENTMATSAM3    TMPP1MATSAM3     THF1MATSAM3
             66              6                5               3
    THF3MATSAM3     THF5MATSAM3 TMPP2CNGMATSAM3      VELMATSAM3
              3              2                2              12
# Coloreamos las bobinas con diferentes espesores
COLORERRORALT <- rep(1,length(INDGRUPOAESTAC))
ESPESORESESCAL <- MATSAM3[,3]*2.5
COLORERRORALT[ESPESORESESCAL<0.7] <- 2
COLORERRORALT[ESPESORESESCAL >=0.7 & ESPESORESESCAL <1] <- 3
COLORERRORALT[ESPESORESESCAL >=1] <- 4
# Visualizamos la proyección de las bobinas con los dos ejes principales PCA
plot(PCAPUNTOS$rproj[,1], PCAPUNTOS$rproj[,2],pch=19, col=COLORERRORALT)
```

Figura 318. Programa que calcula los ejes PCA de las bobinas del GRUPO-A en régimen permanente.

Lo primero que se obtiene, es el valor de la dimensión fractal para determinar la dimensión intrínseca de los puntos. Así, se puede ver que el valor de ésta es 1,31, lo que indica que la estructura puede ser una hipersuperficie y, por lo tanto, podrá ser explicado con dos ejes lineales o no lineales. Aún así, como el número de puntos es limitado, los resultados deben ser considerados "con reservas".

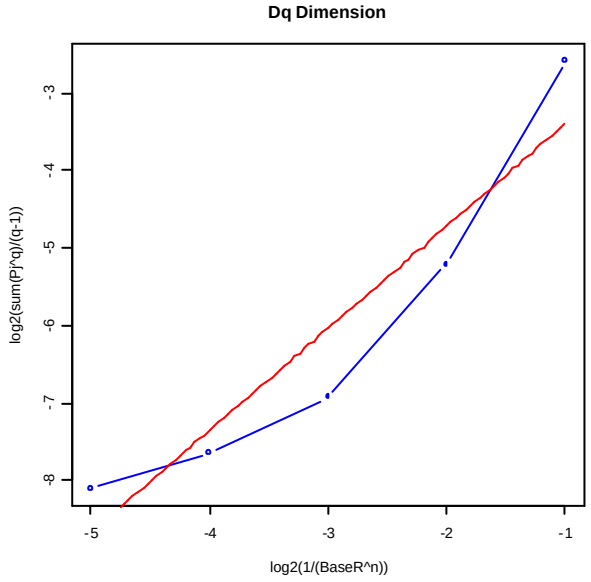


Figura 319. Curva de cálculo de la dimensión fractal.

Por otro lado, si observamos el grado de relevancia de las variables en cada eje, podemos observar que el espesor y la velocidad son relevantes en la posición *x* (eje 1) de la proyección, mientras que en la posición *y* (eje 2) influyen el ancho y velocidad respectivamente.

En los resultados obtenidos con el programa de la Figura 318, nos indican que el 88,8% de la varianza es explicada con los dos ejes principales PCA.

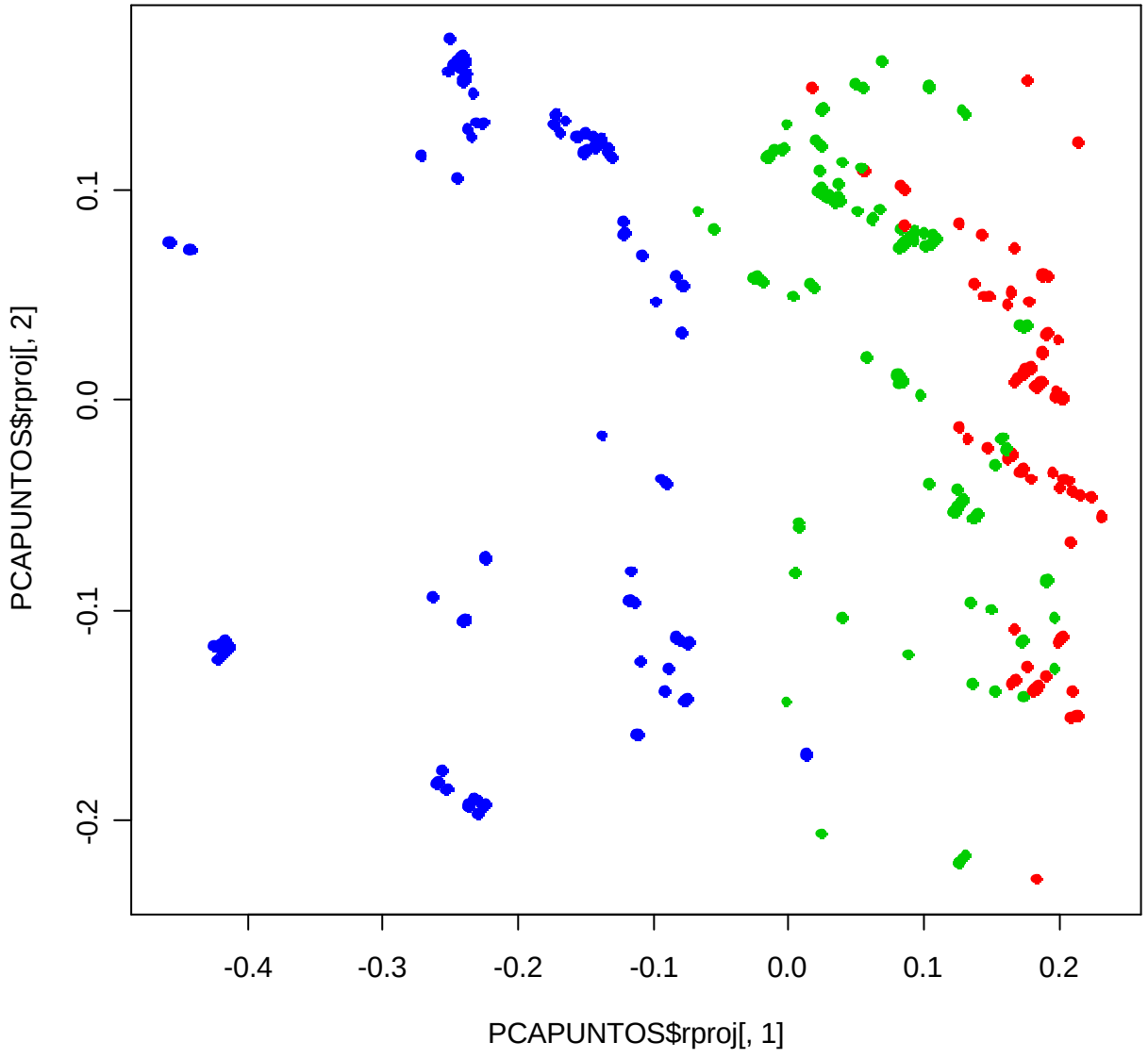


Figura 320. Proyección de las bobinas del GRUPO-A usando los dos ejes PCA principales. Colores: (ESPENT < 0.7 mm.)='rojo', (ESPENT >=0.7 & ESPENT <1 mm.)='verde', (ESPENT >= 1 mm.)='azul'.

En la Figura 320 se muestran los puntos de operación en régimen permanente de las bobinas del GRUPO-A, usando los dos ejes principales. Igual que en las figuras anteriores, se muestran con diferentes colores las bobinas según el espesor de las mismas.

Claramente, podemos clasificar a cada uno de los grupos de puntos como pertenecientes a un grupo de bobinas con características físicas parecidas. En este caso, la anchura y el espesor, separan la proyección de los puntos de operación en diferentes zonas dentro del espacio bidimensional.

7.3.1.5 USO DEL PROYECTOR PARA MONITORIZAR PUNTOS DE OPERACIÓN

El proyector obtenido de los puntos de operación de las bobinas del GRUPO-A en régimen permanente, puede ser utilizado como un sensor-software que nos permita:

- Localizar visualmente el punto de operación del horno.
- Detectar visualmente tendencias hacía estados transitorios.
- Predecir futuros comportamientos.

Vamos a ver con un ejemplo el funcionamiento de este sensor-software para una serie de bobinas del GRUPO-A.

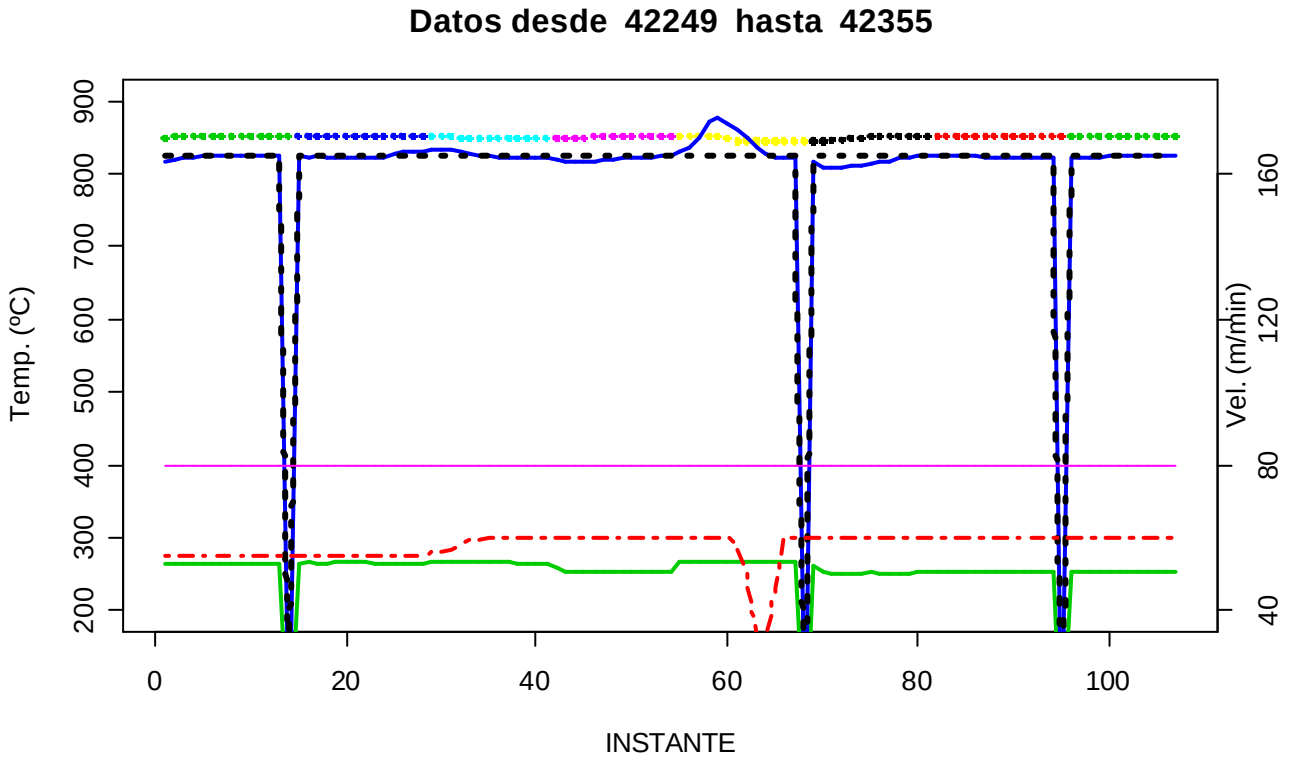


Figura 321. Comportamiento dinámico de las bobinas 23563049 hasta 23563056.

En la Figura 321 se representa el comportamiento dinámico de una serie de bobinas de códigos 23563049 hasta 23563056. Como se puede observar, en una de ellas de repente, aumenta la temperatura de la banda y posteriormente, en el modo manual se modifican las consignas de velocidad y temperatura de zona del horno para adaptarlas a la nueva bobina.

```
# Obtenemos los datos de las bobinas 23563049 a 23563056 #
# (posiciones 10526 a 10704)                             #
##########################################################

INDICEMD <- MATDINAMIC2$CODBOBINA>=23563049 & MATDINAMIC2$CODBOBINA<=23563056

#Creamos una matriz con las variables a visualizar
CODBOBMATD3 <- MATDINAMIC2[INDICEMD,]$CODBOBINA
ANCHOMATD3 <- rep(1350,length(CODBOBMATD3))/2000
ESPENTMAD3 <- rep(2.0,length(CODBOBMATD3))/2.5
TMPP1MATD3 <- MATDINAMIC2[INDICEMD,]$TMPP1M/350
THF1MATD3 <- MATDINAMIC2[INDICEMD,]$THC1/950
THF3MATD3 <- MATDINAMIC2[INDICEMD,]$THC3/950
THF5MATD3 <- MATDINAMIC2[INDICEMD,]$THC5/950
TMPP2CNGD3 <- MATDINAMIC2[INDICEMD,]$TMPP2C/950
VELMATD3 <- MATDINAMIC2[INDICEMD,]$VELOCIDADFIN/200
ERRORD3 <- abs((TMPP2CNGD3*950)-MATDINAMIC2[INDICEMD,]$TMPP2M)
MATD3 <- cbind(CODBOBMATD3, ANCHOMATD3, ESPENTMAD3, TMPP1MATD3, THF1MATD3,
THF3MATD3, THF5MATD3, TMPP2CNGD3, VELMATD3, ERRORD3)

# Eliminamos los espúreos
QUITAESP <- (350*TMPP1MATD3)>100 & (950*TMPP2CNGD3)>100

MATD3SIN <- MATD3[QUITAESP,]

#################################################################

# Visualizamos los puntos con el proyector PCA #
################################################
EJEPCA1 <- PCAPUNTOS$evecs[,1]
EJEPCA2 <- PCAPUNTOS$evecs[,2]

# Obtenemos un vector con la media de los puntos originales del proyector

VECTMEAN <- apply(MATSAM3[,2:9],2,mean)

# Obtenemos los nuevos puntos a proyectar
VECTCENT <- MATD3SIN[,2:9]-
matrix(rep(VECTMEAN,dim(MATD3SIN)[1]),ncol=8,byrow=TRUE)

XP <- VECTCENT*matrix(rep(EJEPCA1, dim(MATD3SIN)[1]),ncol=8,byrow=TRUE)
YP <- VECTCENT*matrix(rep(EJEPCA2, dim(MATD3SIN)[1]),ncol=8,byrow=TRUE)

XPS <- apply(XP,1,sum)
YPS <- apply(YP,1,sum)

# Visualizamos todos los puntos originales
plot(PCAPUNTOS$rproj[,1], PCAPUNTOS$rproj[,2])
points(XPS,YPS,col="red",pch=19)

# Ampliamos la zona y visualizamos en rojo los puntos con errores ALTOS
```

```
plot(PCAPUNTOS$rproj[,1], PCAPUNTOS$rproj[,2], xlim=c(-0.51,-
0.40),ylim=c(0.05,0.1))
COLOR <- rep(4,length(XPS))
COLOR[MATD3SIN[,10]>10 & MATD3SIN[,10]<=30] <- 3
COLOR[MATD3SIN[,10]>30] <- 2
points(XPS,YPS,pch=19,col=COLOR)
```

Figura 322. Programa que proyecta los puntos de operación de las bobinas 23563049 a 23563056.

El comportamiento de todos esos puntos de operación de consigna, pueden ser visualizados mediante el uso de los proyectores anteriormente obtenidos.

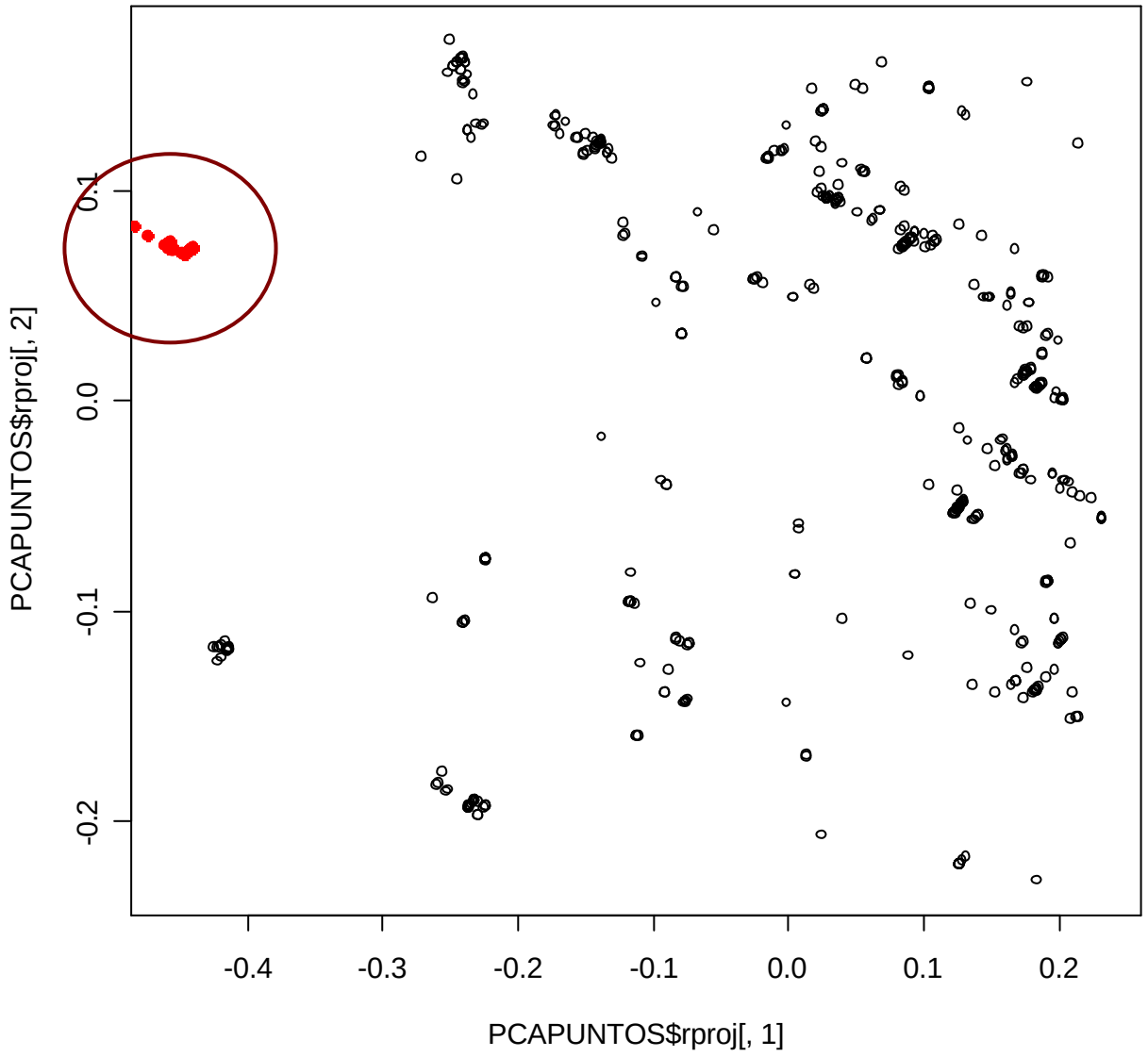


Figura 323. Representación de los puntos de operación de las bobinas 23563049 a 23563056.

De esta forma, en la Figura 323 podemos ver el movimiento de los puntos de operación del horno para esas bobinas frente a los puntos de operación de otras bobinas, con otros tipos de anchuras y espesores; pertenecientes al GRUPO-A de bobinas estudiadas.

En la figura siguiente, se muestra la zona de puntos ampliada. Los colores indican el error medio absoluto cada 100 metros de banda:

- ERROR <= 10ºC.: "Azul"
- ERROR >10ºC. & ERROR<=30ºC.: "Verde"
- ERROR>30ºC.: "Rojo"

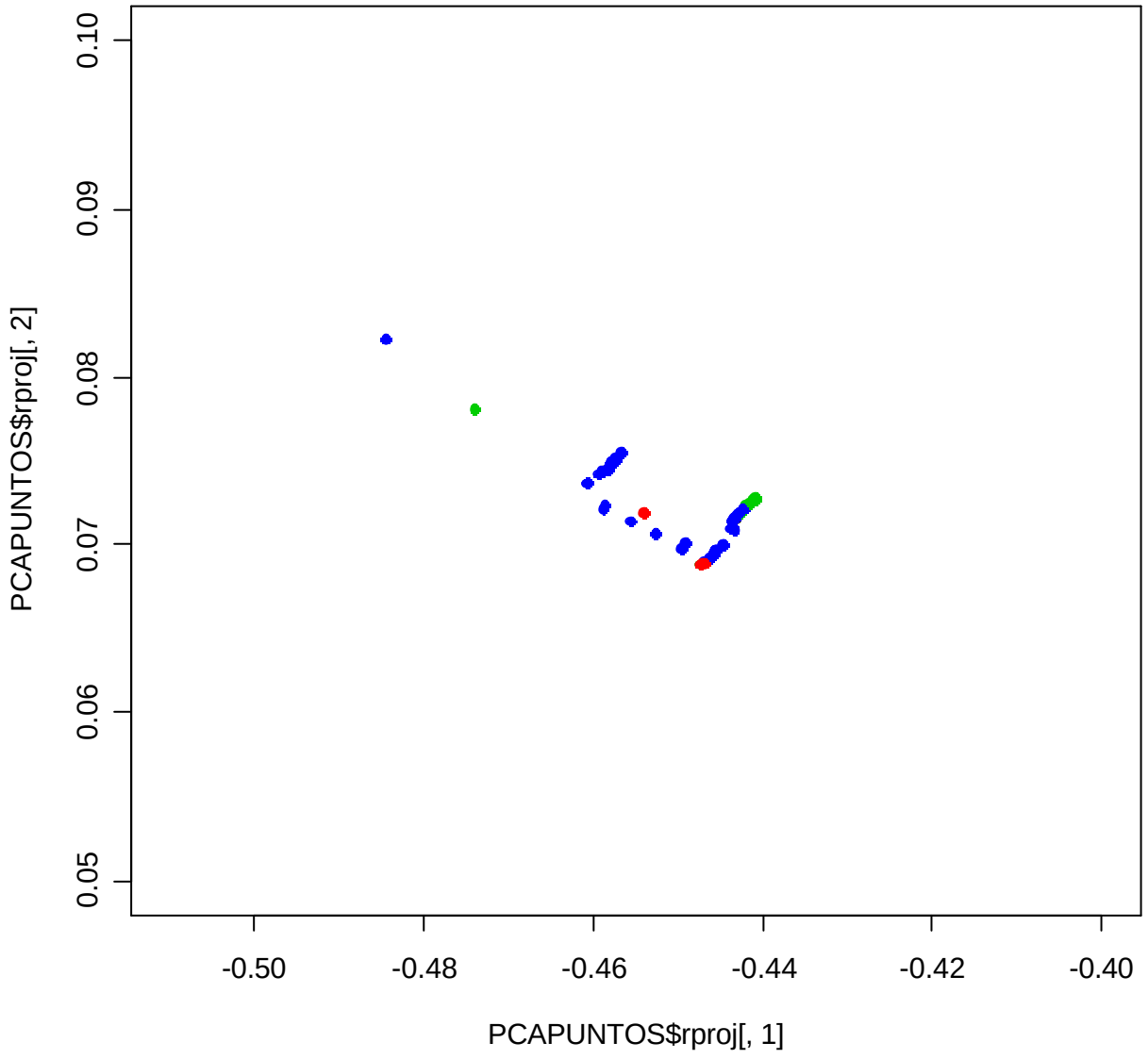


Figura 324. Zona ampliada de los puntos de operación de las bobinas tratadas.

Se puede observar, que los puntos de operación en régimen permanente se mueven dentro de una franja claramente limitada, hasta el momento en que se produce un error considerable y el sistema reacciona.

En las figuras siguientes, se pueden observar las tres fases de operación:

- En régimen permanente.
- Reacción frente un aumento del error.
- Vuelta al régimen permanente.

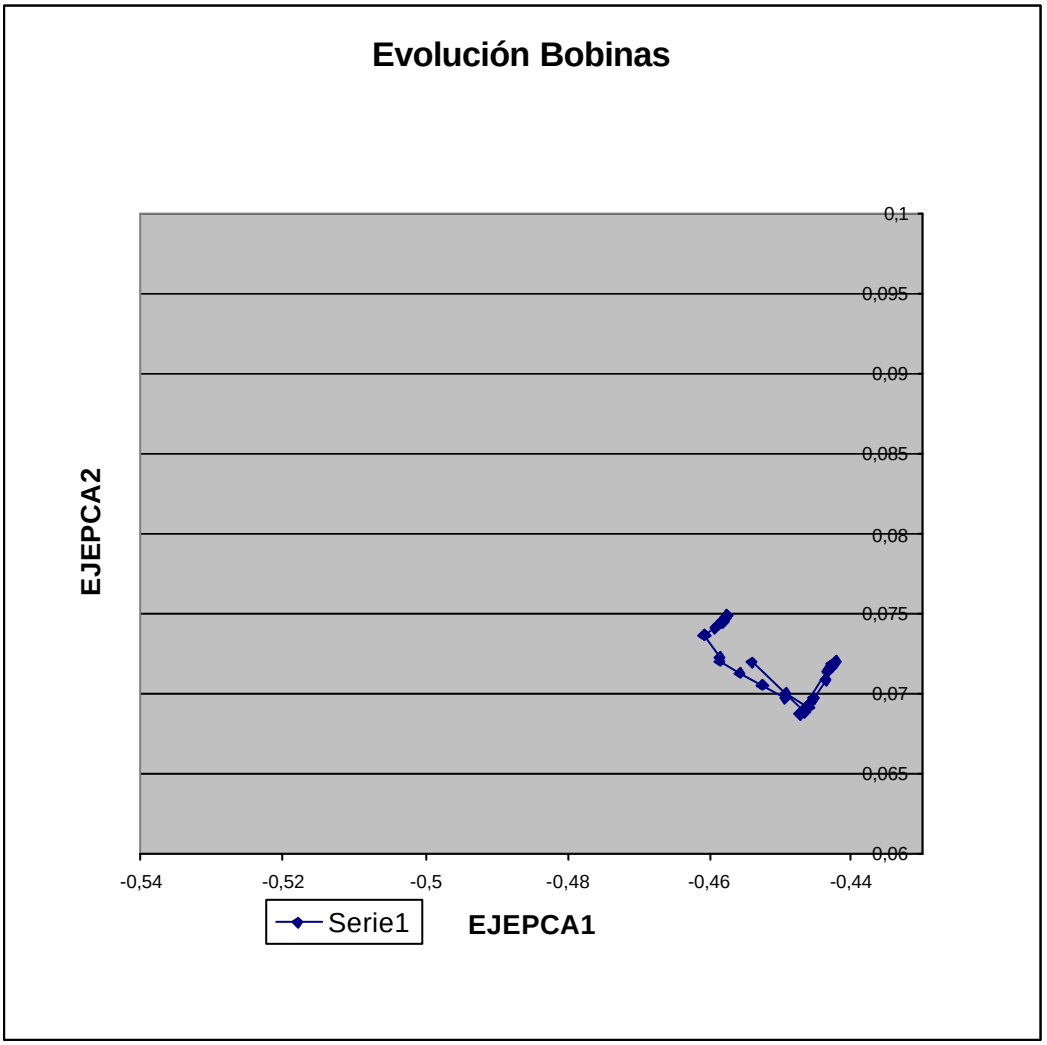


Figura 325. Puntos de operación en régimen permanente.

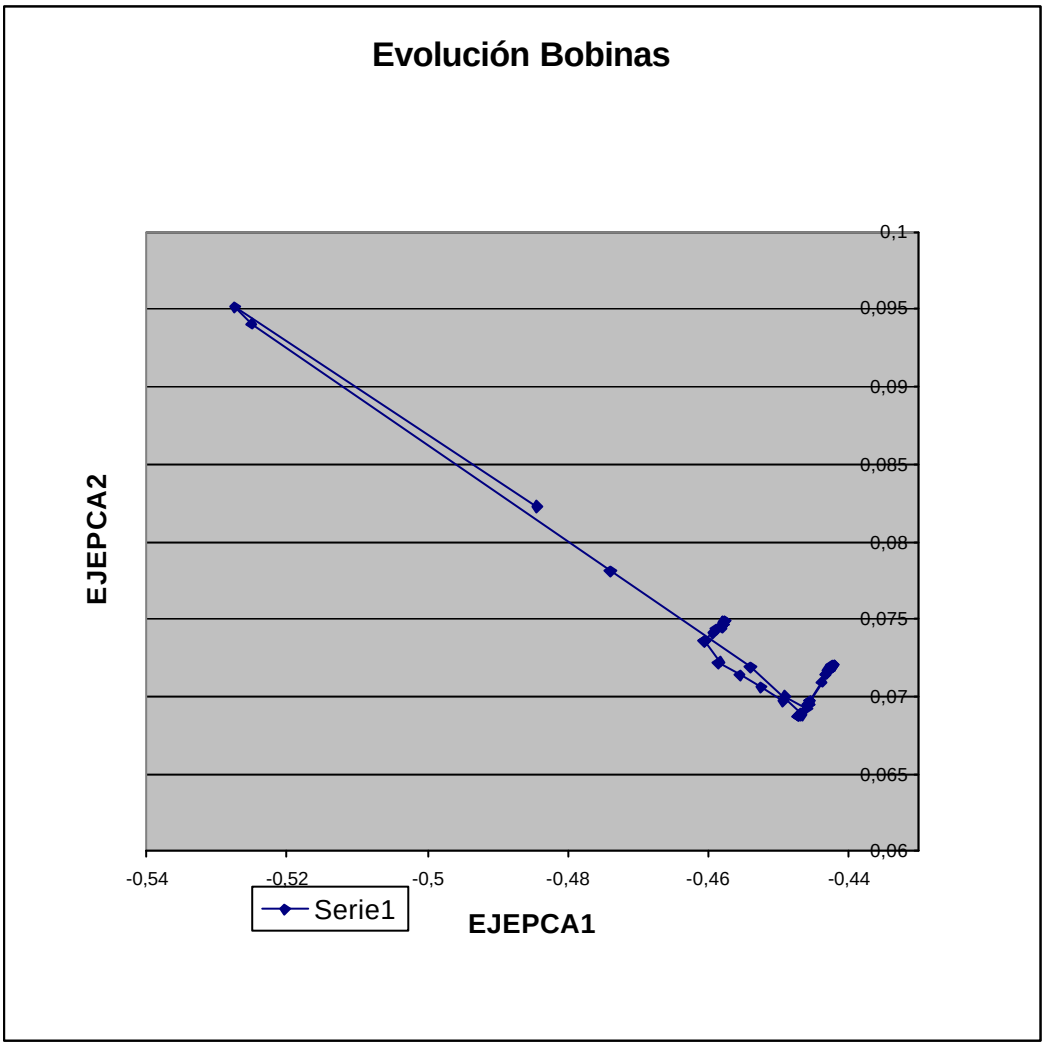


Figura 326. Reacción del sistema ante un aumento del error.

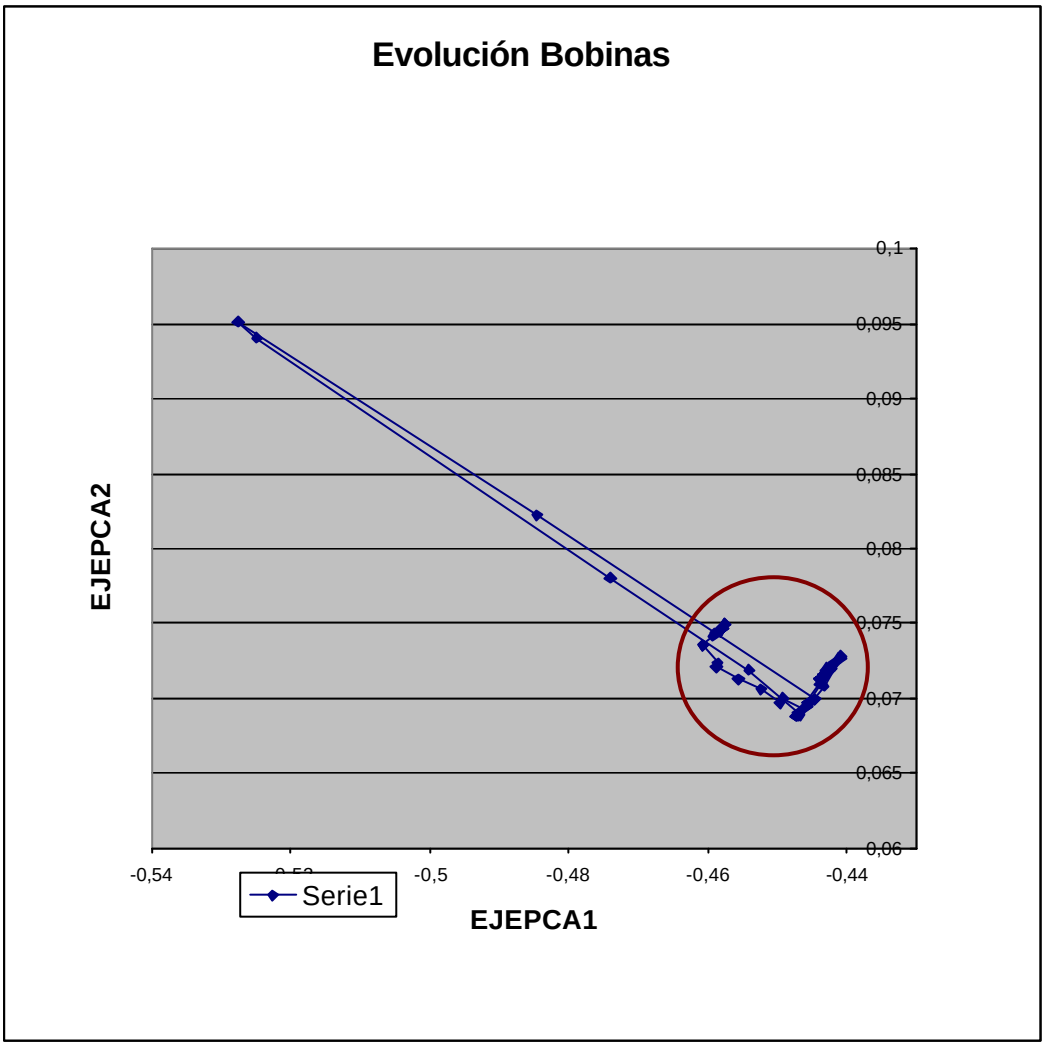


Figura 327. Movimiento de todos los puntos de operación y franja que delimita la zona de trabajo esperada.

7.3.1.6 CONCLUSIONES

Cuando las dimensiones de las bobinas (espesor y anchura) son constantes, las temperaturas y velocidades de consigna entran en juego en el proyector desarrollado.

En las figuras anteriores, se muestra la evolución del punto de operación del horno para un tipo de bobina determinado, de forma que, cuando se produce una situación anómala, se puede detectar rápidamente la reacción del sistema debido a que los puntos proyectados se salen de su "zona normal" de trabajo.

Esta característica del sensor-software puede ser muy útil en el trabajo en "modo manual", ya que permite que el operario pueda ajustar las consignas y visualizar en tiempo real la posición del punto de operación del horno detectando la tendencia del mismo. De esta forma, para cada tipo de bobina según su anchura y espesor, y partiendo de los históricos del proceso, se pueden definir radios de distancia a partir del centroide de puntos en régimen permanente para detectar visualmente cuándo los puntos de consigna que se están ajustando se escapan del centro.

En conclusión, este proyector puede ser una pequeña ayuda cuando se realizan ajustes en "modo manual", ya que muestra cómo se mueve el punto de operación ante pequeñas variaciones. Lógicamente éste no puede ser utilizado para el control en modo automático.

7.3.2 GENERACIÓN DE MODELOS CON REDES NEURONALES

Hasta este momento, el estudio y análisis de los datos ha servido para:

- Obtener conocimiento del proceso de funcionamiento del horno.
- Desarrollar sensores-software y otras herramientas, que puedan ayudar en la planificación y supervisión de todo el sistema.

En este apartado se propone desarrollar una metodología, basada en redes neuronales, que permita simular y planear *off-line* las mejores consignas de velocidad y temperatura de zona del horno, con el objetivo de reducir el error final entre la temperatura de la banda y la esperada.

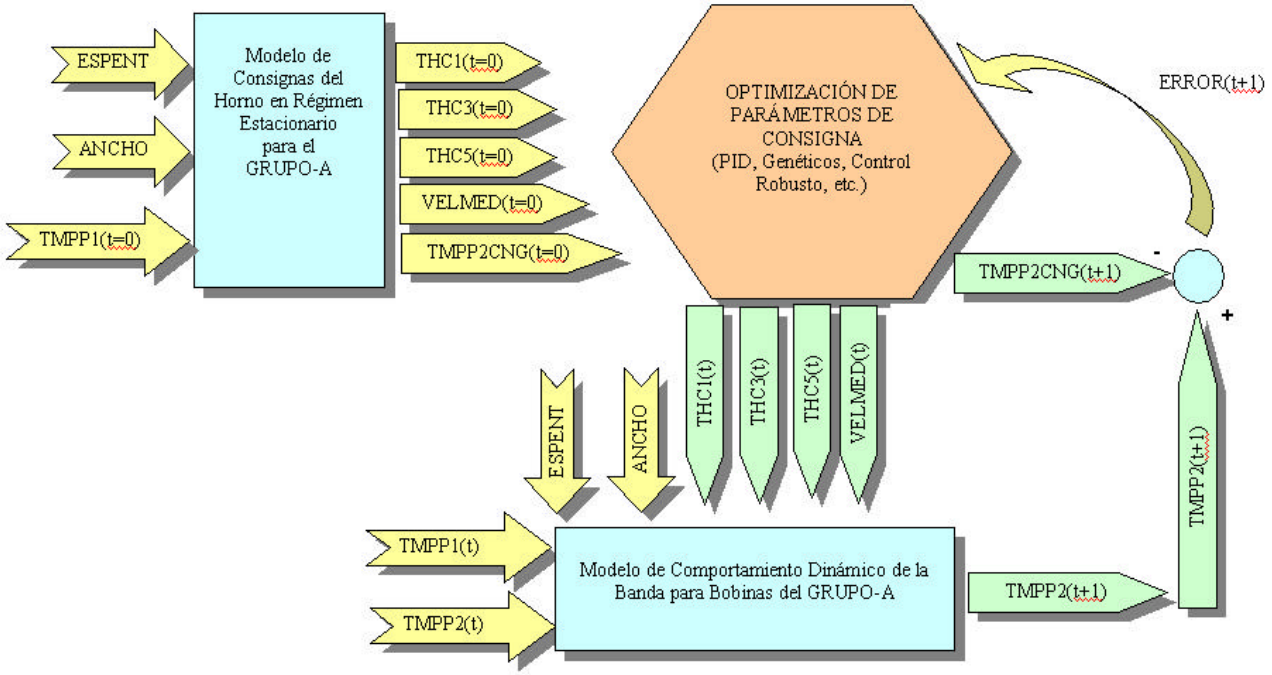


Figura 328. Sistema planteado de Control y Validación Off-Line.

Fundamentalmente, el sistema planteado pretende reducir los errores que se producen en la banda cuando existen:

- Cambios de espesor y anchura de la banda.
- Cambios de tipos de aceros.

Para ello, se pretende la creación de los siguientes modelos basados en redes neuronales:

- **Modelo de Consignas del Horno en Régimen Estacionario. Que aprenda el comportamiento del sistema, tanto en el "modo manual" como en el "modo automático"**, partiendo de la anchura de la bobina, el espesor de la misma y la temperatura de entrada de la banda; y siempre para aquellos casos en donde el tratamiento está en régimen permanente y con errores BAJOS.
- **Modelo de Comportamiento Dinámico de la Banda**: Que modelice el comportamiento de la banda ante fluctuaciones de temperatura de zona del horno, de entrada de banda y de velocidad media de la misma.

De esta forma, el primer modelo nos permitirá conocer las temperaturas y velocidades de consigna más adecuados para cada tipo de bobina, mientras que el segundo nos ayudará a conocer con anterioridad el comportamiento de la banda ante las variaciones de temperatura y velocidad.

7.3.2.1 METODOLOGÍA PLANTEADA

El sistema de control *off-line*, pretende ser una herramienta útil para planificar las temperaturas y velocidades de consigna entre cambios de dimensiones de banda y de tipos de aceros [KIM98b].

Para su realización, se desarrollará la siguiente metodología (ver Figura 328):

- Se usará el clasificador de bobinas, realizado en capítulos anteriores, para agrupar las bobinas en grupos con características mecánicas y térmicas similares.
- Para cada grupo de bobinas, se creará, a partir de los datos históricos de procesos en régimen estacionario y con errores BAJOS, el modelo no lineal de las temperaturas de consigna de zona del horno, temperaturas de consigna de banda y velocidades. Se validará el grado de generalización obtenido.
- También, separadamente para cada grupo de bobinas, se desarrollará el modelo de comportamiento dinámico ante cambios de velocidad o temperaturas. Se validará el grado de generalización del modelo creado.
- Una vez tenemos los dos tipos de modelos para cada grupo de bobinas, se simularán las temperaturas de zona y velocidades ante cambios de dimensiones o/y temperaturas de entrada de la banda.
- Mediante técnicas de control clásicas o avanzadas, se buscarán las mejores consignas que reduzcan el error en la transición.

A continuación, se muestra, para el GRUPO-A de bobinas, la aplicación práctica de la metodología planteada.

7.3.2.2 GENERACIÓN DE MODELOS NO LINEALES PARTIENDO DE VALORES DE CONSIGNAS EN RÉGIMEN ESTACIONARIO

En este punto se describen los pasos realizados para la creación de un modelo matemático no lineal que nos permita, a partir de los datos de históricos del proceso, obtener las variables de consigna más adecuadas para cada tipo de bobina del GRUPO-A.

El objetivo es aprender del proceso, tanto en "modo manual" como en "modo automático", de los **casos con errores BAJOS y en régimen permanente (CURVAS HORIZONTALES), generando una red neuronal no lineal que nos prediga las variables de consigna óptimas para cada bobina según: la anchura, espesor de entrada y temperatura de entrada de la banda.**

Para ello, se utilizan varias redes neuronales multicapa MLP entrenadas con el método de aprendizaje *Backpropagation* o el método *Levenberg-Marquardt* [MAT00].

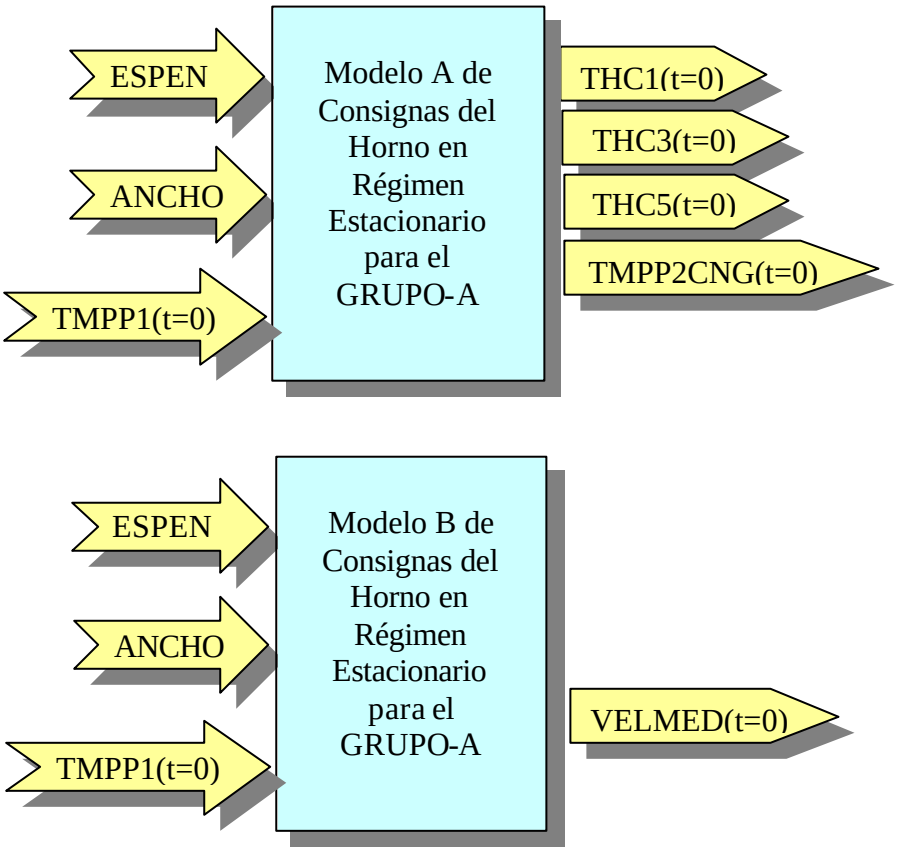


Figura 329. Modelos generados para la obtención de las variables de consigna.

La validación se realizará mediante el método de validación cruzada [MAR01], buscando el mínimo óptimo de testeo [PRE98]. Para ello, de la base de datos obtenida, se utilizará un 80% de los patrones para el entrenamiento y otro 20% como conjunto de test [HAY99].

CREACIÓN DE LA BASE DE DATOS

Lo primero que hacemos, es obtener una matriz con los datos de consigna de las bobinas en régimen permanente y con errores BAJOS de las bobinas del GRUPO-A.

```
# Obtenemos las bobinas de la familia GRUPO-A
INDGRUPOAESTAC <- MATBOBINAS2$FAMILIA==1 & MATBOBINAS2$TIPOCURVATHF1==”H” &
MATBOBINAS2$TIPOCURVATMPP1==”H” & MATBOBINAS2$TIPOCURVATMPP2CNG==”H” &
MATBOBINAS2$TIPOCURVAERROR==”H” &  MATBOBINAS2$TIPOCURVAVEL==”H”

# Obtenemos las bobinas
TIPCOD <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$CODBOBINA))

# Sacamos el índice de las bobinas en la matriz de datos dinámicos
INDGMATDINAMIC <- MATDINAMIC2$CODBOBINA %in% TIPCOD


#Creamos una matriz con las variables a modelizar
CODBOBMATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$CODBOBINA
TMPP1MATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$TMPP1M
THF1MATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$THC1
THF3MATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$THC3
THF5MATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$THC5
TMPP2CNGMATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$TMPP2C
VELMATSAM3 <- MATDINAMIC2[INDGMATDINAMIC,]$VELOCIDADFIN
ERRORSAM3 <- abs(MATDINAMIC2[INDGMATDINAMIC,]$TMPP2C-
MATDINAMIC2[INDGMATDINAMIC,]$TMPP2M)

# Obtenemos un número de posición para cada bobina
POSINCBOB <- match(CODBOBMATSAM3,TIPCOD)
ANCHOBOB <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ANCHO))
ESPENTBOB <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ESPENT))

# Creamos la anchura y espesor
ANCHOMATSAM3 <- ANCHOBOB[POSINCBOB]
ESPENTMATSAM3 <- round(ESPENTBOB[POSINCBOB]*1000)

MATSAM3 <- cbind(CODBOBMATSAM3, ANCHOMATSAM3, ESPENTMATSAM3, TMPP1MATSAM3,
THF1MATSAM3, THF3MATSAM3, THF5MATSAM3, TMPP2CNGMATSAM3, VELMATSAM3, ERRORSAM3)

# Eliminamos los espúreos
INDGHT <- MATSAM3[,4]>100 & MATSAM3[,5]>100 & MATSAM3[,6]>100 & MATSAM3[,7]>100
& MATSAM3[,8]>100 & MATSAM3[,9]>10
MATSAM3SIN <- MATSAM3[INDGHT,]

dim(MATSAM3SIN)
[1] 12221    10

summary(MATSAM3SIN)
CODBOBMATSAM3       ANCHOMATSAM3  ESPENTMATSAM3     TMPP1MATSAM3
 Min.   :23293014   Min.   : 760   Min.   : 601.0   Min.   :209.0
 1st Qu.:23413048   1st Qu.:1062   1st Qu.: 675.0   1st Qu.:247.0
 Median :23533031   Median :1200   Median : 775.0   Median :259.0
 Mean   :23499232   Mean   :1199   Mean   : 908.2   Mean   :257.0
 3rd Qu.:23573037   3rd Qu.:1390   3rd Qu.: 975.0   3rd Qu.:270.0
 Max.   :23653012   Max.   :1525   Max.   :2000.0   Max.   :292.0
```

```
  THF1MATSAM3     THF3MATSAM3     THF5MATSAM3    TMPP2CNGMATSAM3
 Min.   :761.0   Min.   :791.0   Min.   :808.0   Min.   :755.0
 1st Qu.:823.0   1st Qu.:853.0   1st Qu.:873.0   1st Qu.:825.0
 Median :837.0   Median :867.0   Median :886.0   Median :825.0
 Mean   :833.5   Mean   :863.8   Mean   :884.5   Mean   :823.7
 3rd Qu.:847.0   3rd Qu.:879.0   3rd Qu.:900.0   3rd Qu.:825.0
 Max.   :877.0   Max.   :907.0   Max.   :931.0   Max.   :855.0
   VELMATSAM3      ERRORSAM3
 Min.   : 55.0   Min.   : 0.000
 1st Qu.: 97.0   1st Qu.: 1.000
 Median :115.0   Median : 1.000
 Mean   :109.2   Mean   : 2.125
 3rd Qu.:125.0   3rd Qu.: 3.000
 Max.   :140.0   Max.   :19.000

# Guardamos la matriz en un archivo csv
write.table(MATSAM3SIN,"c:\\temp\\NEURONALPERM.CSV",quote=FALSE,sep=",",row.name
s=FALSE,col.names=FALSE)
```

Figura 330. Programa que obtiene los datos en régimen permanente de la base de datos de históricos.

DISEÑO DE LAS REDES NEURONALES

Una vez obtenidos los datos de aquellas bobinas que **están en régimen permanente y con curvas horizontales**, procedemos a generar varios modelos que nos permitan predecir las variables de consigna del horno a partir de las variables de entrada.

En este momento, es conveniente diseñar el tamaño de la red a la complejidad del problema que se está tratando [HAY99] en los siguientes aspectos:

- Tipo de red neuronal.
- Número de capas.
- Tipo de funciones de las neuronas.
- Tipo de Entrenamiento a realizar.
- Tipo de Testeo y error de testeo.
- Número de neuronas.

Tipo de Red Neuronal

Inicialmente, **se selecciona una red neuronal multicapa tipo perceptrón (MLP)**, ya que se ha comprobado experimentalmente que es capaz de representar complejos *mappings* y de abordar problemas de clasificación de gran envergadura, de una manera eficaz y relativamente simple [MAR01].

Este tipo de red neuronal se utiliza en la asociación y clasificación de elementos con un número finito de discontinuidades. Puede estar formada por múltiples capas con diferentes funciones de transferencia cada una de ellas. En las figuras siguientes, se puede ver el esquema de este tipo de red neuronal y las funciones de transferencia más comúnmente usadas.

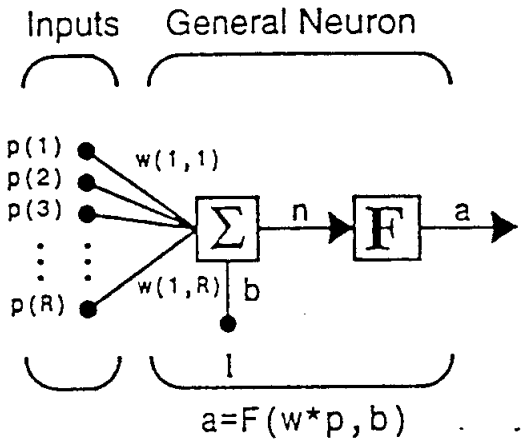


Figura 331. Esquema general de una neurona de una red tipo perceptrón.

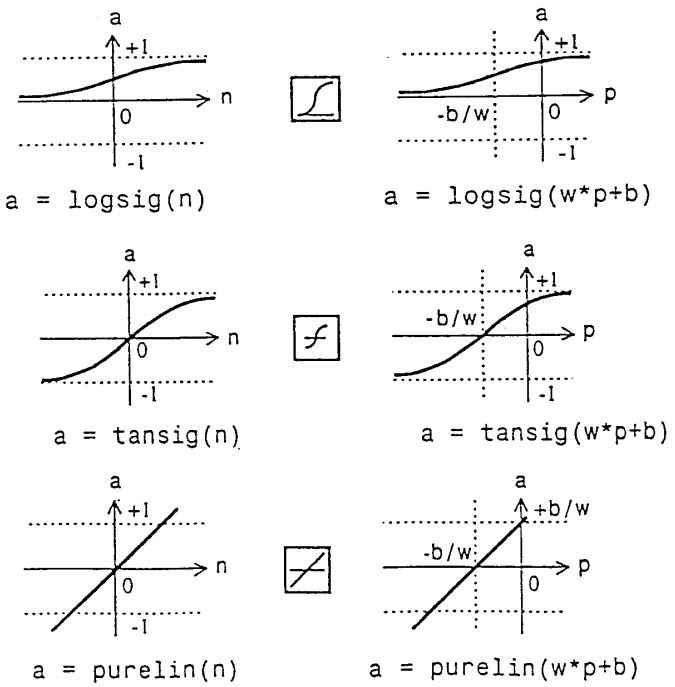


Figura 332. Funciones de transferencia más utilizadas en una red MLP.

Fundamentalmente las redes de este tipo, formadas por dos capas con funciones sigmoidea/lineal, pueden representar cualquier relación funcional entre entradas/salidas siempre que la capa sigmoidea tenga suficiente número de neuronas.

Número de Capas

Se selecciona inicialmente dos capas ocultas, aunque varios autores han demostrado que una arquitectura MLP con una sola capa oculta es un aproximador universal de funciones [HEC87][HEC90]. Aún así, si la solución no es óptima, se usarán dos capas ocultas.

Tipo de Entrenamiento y Validación

Se entrena la red, por ejemplo, con el 60% de los patrones extraídos de forma aleatoria de la base de datos usando otros 20% para la validación y el último 20% para testeo. Esto se realizará hasta que se alcance el error mínimo de generalización (error de testeo) siempre que se alcance un umbral próximo al 0,5%.

El aprendizaje se realizará en modo serie (*on line*), que consiste en que los pesos sinápticos se actualizan tras la presentación de cada patrón, ya que es más adecuado para aquellos problemas en que se dispone de un conjunto numeroso de patrones de entrenamiento en el que hay mucha redundancia de datos [MAR01].

Número de Neuronas

Según [BAU89][HAY99][MAR01] se demuestra que una red de n entradas y h neuronas ocultas, con un total de w pesos, requiere un número de patrones de aprendizaje del orden de $p=w/e$ para proporcionar un error generalizado del orden de e. Así, si queremos que una red alcance un error de generalización de, por ejemplo, $e=0.1$ (un 10%), el número de patrones de aprendizaje necesarios p será del orden de $p=10 \cdot w$, expresión que suele ser indicativa del número aproximado de patrones necesarios para entrenar adecuadamente una red neuronal de w pesos.

Como el número de patrones obtenidos es de 12.221, de los cuales se van a utilizar 9.777 (el 80%) para entrenamiento y el 20% para testeo, y el error mínimo buscado es de un 0,5%, el número de pesos de la capa oculta será como máximo de:

$$p = \frac{w}{e}; \qquad 9777 = \frac{w}{0{,}005}; \qquad w = 48{,}88 \tag{7.1}$$

Inicialmente, como el número de entradas no es muy grande y el número de salidas no excede de tres, se decide empezar con una capa oculta de 9 neuronas, ya que el número de pesos es: 12·3+12=48.

Error de Entrenamiento y Validación

El error utilizado será la media de la suma de los errores al cuadrado MSE (*Mean Sum of squares of Errors*) para los datos de entrenamiento, validación y testeo.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^{\,2} = \frac{1}{N}\sum_{i=1}^{N} (t_i - a_i)^2 \tag{7.2}$$

CREACIÓN Y TESTEO DE LOS MODELOS PARA LAS TEMPERATURAS DE CONSIGNA

Una vez tenemos diseñada la topología y número de capas de la red neuronal, procedemos a entrenarla mediante la *Neural Network Toolbox, versión 4 (R12)* perteneciente a la herramienta Matlab®. Ésta nos permite gran cantidad de métodos diferentes de entrenamiento. Se ha utilizado uno de los métodos más eficaces (aunque necesita más memoria), el método *Levenberg-Maquardt* que realiza el aprendizaje mediante una aproximación del método de Newton.

El programa de la figura siguiente desarrolla diferentes redes neuronales con una capa intermedia que varía entre 7 y 27 neuronas. Cada una de las redes es entrenada y validada, y se almacena aquella que tiene menor error de generalización.

```
function modelo_consignasTEMP(MAT)

%Programa que calcula la red neuronal Backpropagation (con diferentes neuronas)
%óptima para obtener un error de entrenamiento inferior a un 0,1
%y un error de generalizacion menor del 0,5%

%Cargamos los datos de MAT
% COL 1=CODBOBMATSAM3
% COL 2=ANCHOMATSAM3
% COL 3=ESPENTMATSAM3
% COL 4=TMPP1MATSAM3
% COL 5=THF1MATSAM3
% COL 6=THF3MATSAM3
% COL 7=THF5MATSAM3
% COL 8=TMPP2CNGMATSAM3
% COL 9=VELMATSAM3
% COL 10=ERRORSAM3

%MAT = csvread('c:\\temp\\NEURONALPERM.CSV');

% Normalizamos la Matriz
MinVect= [10000000, 700, 0, 100, 700, 700, 700, 700, 10, 0];
MinimosMAT = ones(size(MAT),1) * MinVect;

VectRang=[30000000, 1300, 2500, 300, 300, 300, 300, 300, 200, 200];
MATRange = ones(size(MAT),1) * VectRang;

MATNORM = (MAT-MinimosMAT)./MATRange;

%Obtenemos las Entradas y las Salidas de los datos
% 70% aleatorios de entrenamiento
% 20% aleatorios de validacion
% 10% aleatorios de testeo
        % Posiciones aleatorias para TEST
        NumPatrones = size(MATNORM,1);
        Aleat = 1000*rand(NumPatrones,1);
        EnumPos = 1:NumPatrones;
        PosAleat = [Aleat EnumPos'];
        PosAleatOrd = sortrows(PosAleat,1);

        PosIniTot=1;
        PosFinTot=round(0.9*NumPatrones);
```

```
        PosIniTest=PosFinTot+1;
        PosFinTest=NumPatrones;

        % Datos de testeo
        PosTest = PosAleatOrd(PosIniTest:PosFinTest,2);
        DatosInTest = MATNORM(PosTest,2:4);
        DatosOutTest = MATNORM(PosTest,5:8);

        % Los demas datos
        PosTot = PosAleatOrd(PosIniTot:PosFinTot,2);
        MATNORM2 = MATNORM(PosTot,1:10);

for j=1:3000
    % Creamos la red neuronal
      Neuronas=7+mod(j,20);
    net = newff([0 1;0 1;0 1],[3 Neuronas 4],{'logsig' 'logsig'
'purelin'},'trainlm');

    % Error MSE
    ERRORMSEENT = [0]
    ERRORMSEVAL = [0]
    ERRORMSTEST = [0]

    ERRORMAXEVAL = 9999
    ERRORMAXTEST = 9999
    %Inicializamos el bucle
    for h=1:1

        %Obtenemos las Entradas y las Salidas de los datos
        % 80% aleatorios de entrenamiento
        % 20% aleatorios de validacion

        % Posiciones aleatorias
        NumPatrones = size(MATNORM2,1);
        Aleat = 1000*rand(NumPatrones,1);
        EnumPos = 1:NumPatrones;
        PosAleat = [Aleat EnumPos'];
        PosAleatOrd = sortrows(PosAleat,1);

        PosIniEnt=1;
        PosFinEnt=round(0.8*NumPatrones);

        PosIniVal=PosFinEnt+1;
        PosFinVal=NumPatrones;

        % Datos de entrenamiento
        PosEnt = PosAleatOrd(PosIniEnt:PosFinEnt,2);
        DatosInEnt = MATNORM2(PosEnt,2:4);
        DatosOutEnt = MATNORM2(PosEnt,5:8);

        % Datos de Validacion
        PosVal = PosAleatOrd(PosIniVal:PosFinVal,2);
        DatosInVal = MATNORM2(PosVal,2:4);
        DatosOutVal = MATNORM2(PosVal,5:8);

        tic;
        %Entrenamos la red
        net.trainParam.epochs = 30;
        net.trainParam.goal = 0.0001;
        net.trainParam.show = 5;
```

```
        P=DatosInEnt';
        T=DatosOutEnt';
        [net,tr,Y,E,Pf,Af] = train(net,P,T);
        %net.adaptParam.passes = 100;
        %[net,Y,E,Pf,Af] = adapt(net,P,T);
        MSEENT = mse(E)
        ERRORMSEENT = [ERRORMSEENT MSEENT];

        % Simulamos con los datos de validacion
        P=DatosInVal';
        T=DatosOutVal';
        [Y,Pf,Af,E,perf] = sim(net,P,Pf,Af,T);
        MSEVAL = mse(E)
        ERRORMSEVAL = [ERRORMSEVAL MSEVAL];
        toc;

        if ERRORMAXEVAL>MSEVAL
            ERRORMAXEVAL=MSEVAL;
            Arch2='c:\temp\ModeloA\consignasTEMP\MATMEJOR';
            Archivo=strcat(Arch2,num2str(j),'.MData');
            save
(Archivo,'PosAleat','net','ERRORMSEENT','ERRORMSEVAL','ERRORMSTEST','ERRORMAXEVA
L');
        end
        % Simulamos con los datos de test
        P=DatosInTest';
        T=DatosOutTest';
        [Y,Pf,Af,E,perf] = sim(net,P,Pf,Af,T);
        MSTEST = mse(E)
        ERRORMSTEST = [ERRORMSTEST MSTEST];

        if ERRORMAXTEST>MSTEST
            ERRORMAXTEST=MSTEST;
            Arch2='c:\temp\ModeloA\consignasTEMP\MATMEJORTEST';
            Archivo=strcat(Arch2,num2str(j),'.MData');
            save
(Archivo,'PosAleat','net','ERRORMSEENT','ERRORMSEVAL','ERRORMSTEST','ERRORMAXEVA
L');
        end

        figure(3);
        plot(1:(h+1),ERRORMSEENT,1:(h+1),ERRORMSEVAL,1:(h+1),ERRORMSTEST);
        refresh(3);

        figure(2);
        %plot(1:size(T,2),T(1,:),1:size(Y,2),Y(1,:));
        plot(1:100,T(1,1:100),1:100,Y(1,1:100));
        refresh(2);


        Arch2='c:\temp\ModeloA\consignasTEMP\ERRORFINAL';
        Archivo=strcat(Arch2,num2str(j),'.MData');
        save (Archivo,'ERRORMSEENT','ERRORMSEVAL','ERRORMAXEVAL','ERRORMSTEST');

    end
end
```

Figura 333.. Programa que busca la red más óptima con una capa oculta variable.

En la Figura 333, se busca la topología más adecuada que permita resolver con mayor eficiencia el problema planteado.

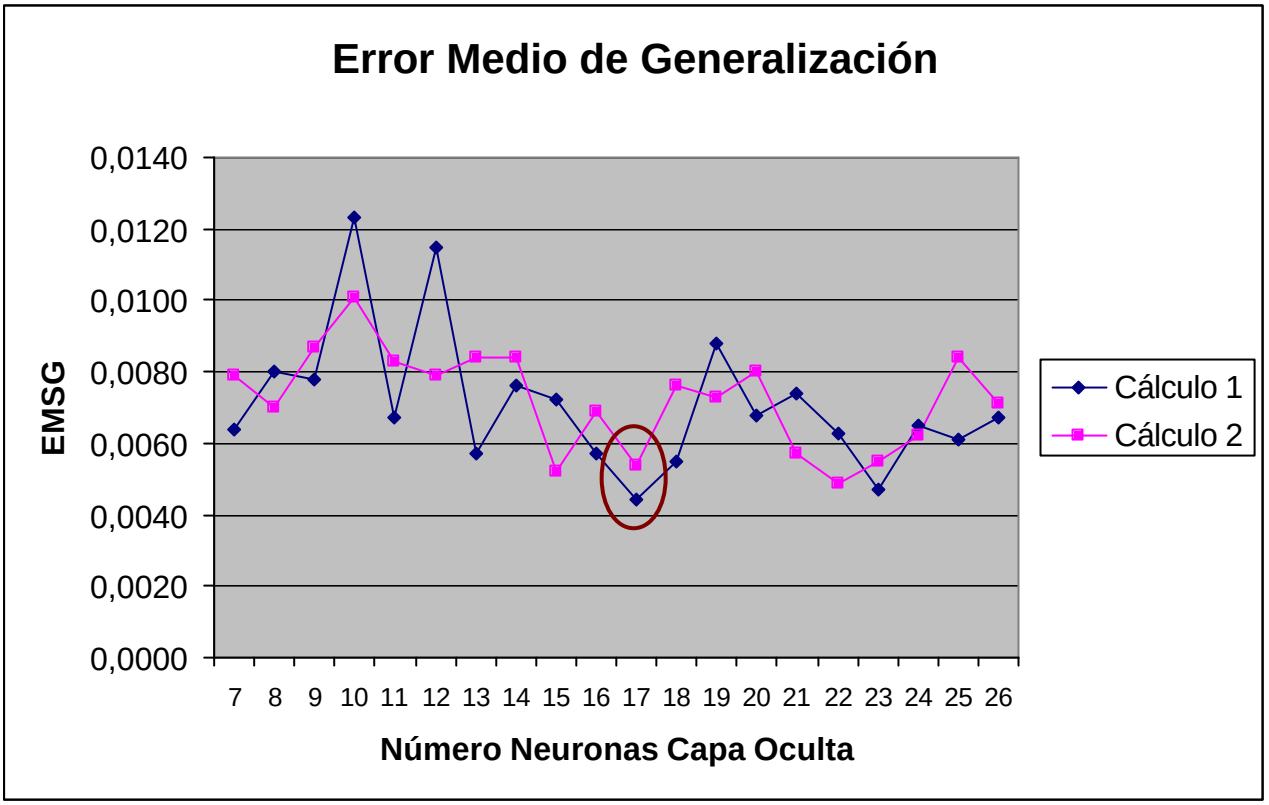


Figura 334. Error de validación obtenido según el número de neuronas de la capa oculta.

En la Figura 334, podemos observar que los menores errores de generalización obtenidos, se han realizado con una capa oculta de 17 y de 23 neuronas. Por lo tanto, procedemos a crear varias redes, entrenarlas y obtener aquella que de un error de generalización más óptimo. Para ello, se utiliza el 60% de los datos para entrenamiento, el 20% para validación y el 20% para testeo final, según el siguiente procedimiento:

- Se separa un 20% de los datos elegidos aleatoriamente para el testeo.
- Se entrena la red neuronal con un 60% de los datos elegidos aleatoriamente.
- Se valida con el otro 20%.
- Se almacena la red neuronal con menor error de validación.
- Se calcula para los datos de testeo.
- Se almacena la red neuronal con menor error de testeo.

Resultados Obtenidos

Se han probado redes neuronales de 17 y 23 neuronas en su capa intermedia obteniéndose los siguientes resultados:

- Se ha elegido una red de 17 neuronas después de un entrenamiento más exhaustivo y mejores datos de entrada, con los siguientes datos:
  - MSE de entrenamiento: 0,00070. Es decir, un 2,64 % de error de entrenamiento.
  - MSE de validación: 0,00072. Es decir, un 2,68% de error de validación.
  - MSE de testeo: 0,00075. Es decir, un 2,74% de error de testeo.

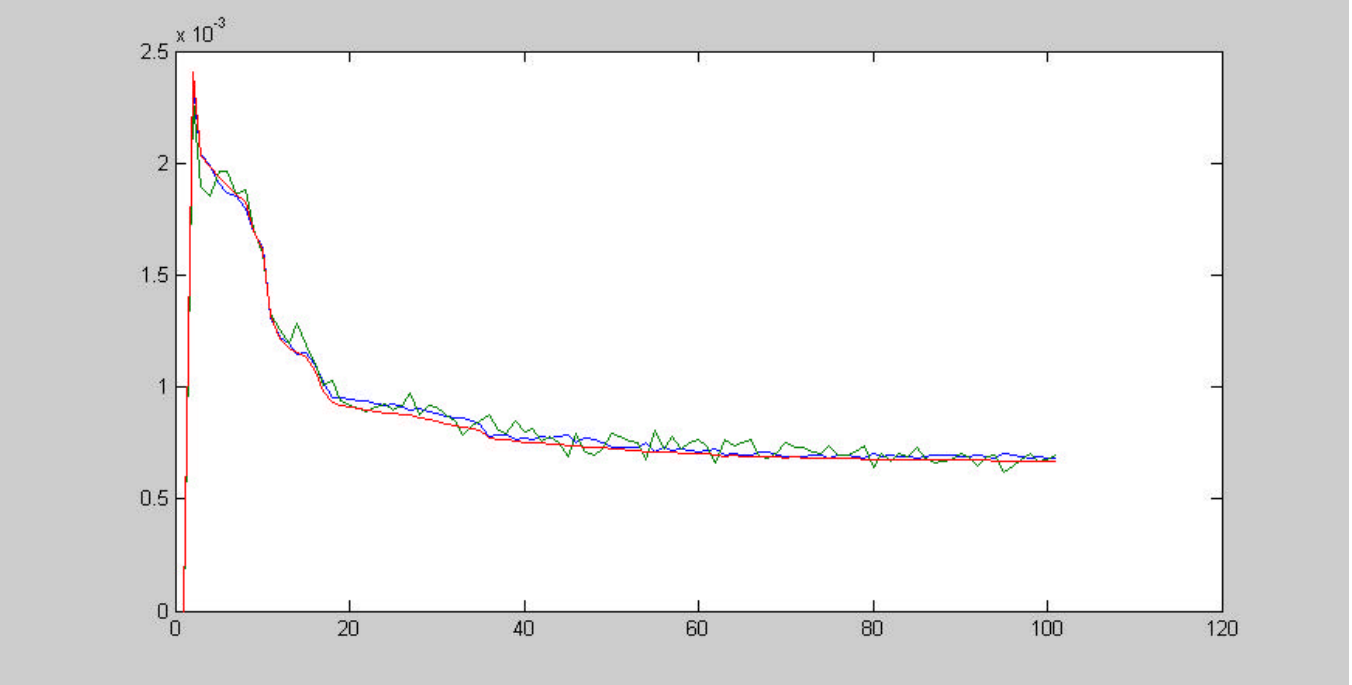


Figura 335. Evolución del error MSE de entrenamiento (azul) frente al de validación (izquierda en rojo) y testeo (derecha en verde).

```
% Pesos y Bias de las neuronas de la capa de Entrada
% Pesos
net.IW{1}
ans =

   -4.4280    8.2246    3.5723
    6.0853    2.8912    0.4239
    1.1510   -4.5137    6.8319

% Bias
net.b{1}
```

```
ans =
    1.2605
   -4.0398
    7.3941



% Pesos y Bias de las neuronas de la capa de Oculta
% Pesos
net.LW{2,1}

ans =
   10.2616    8.1402    7.5774
    0.8080   -9.8540  -12.6268
    9.6312    7.2357   -7.2686
   -8.6400    2.3859   10.9394
    5.8841  -11.1993   -5.8035
    2.0676   -9.4299   10.8093
   -9.9275    7.0629    5.6133
   -1.7384   -7.1309  -12.2447
  -14.1050    1.6172   -2.2569
    8.9588   -9.2876    6.4033
    3.6919   14.0872   -2.9327
    9.4901    2.9976   10.4058
    2.5112   -9.4811   10.5438
  -14.2111    1.4599   -1.8024
   13.4353    2.9004   -4.2700
    9.2953   10.4224    3.3985
   -1.2426    8.6221   11.1091
% Bias
net.b{2}
ans =
  -19.5354
   15.1282
  -12.0651
   -0.2557
   -0.8696
   -3.9789
   -3.4777
   11.8956
    7.4070
   -2.1044
   -6.2683
   -8.7474
    1.8286
    2.7774
   -0.6502
   -5.3194
  -16.9641

% Pesos y Bias de las neuronas de la capa de Salida
% Pesos
net.LW{3,2}
ans =
  Columns 1 through 8
    2.2947    2.7021   -0.7706   -0.9589    0.4879   -0.9985   -0.6959    1.7788
    2.2940    2.8589    0.7596   -1.0558    0.9346   -0.6735   -0.7720   -0.2067
    2.3683    2.9556    0.3851   -0.2369    0.7491   -0.6459   -0.5881    1.0042
```

```
    1.0890    2.1480    0.0617    1.1019    0.0233   -1.0515    0.5003   -0.0137
  Columns 9 through 16
    0.1575   -0.3214    1.1028   -0.1423    0.1543    0.4119   -0.2764   -0.4914
   -0.5923   -0.5941    1.1926    0.5441   -1.2184    0.4836    0.0907    0.3393
   -0.1006    0.4197    1.2284   -1.2713   -0.4706   -0.4367   -0.5963    0.4871
    0.4612    0.3023    1.2317    0.3192    0.4207   -0.3844   -1.3802   -0.8906



  Column 17
   -0.3269
   -0.1854
   -0.2197
   -0.8897

net.b{3}
ans =
    0.1165
   -0.3471
   -0.6192
   -0.6034
```

Figura 336. Pesos y bias de la red neuronal creada.

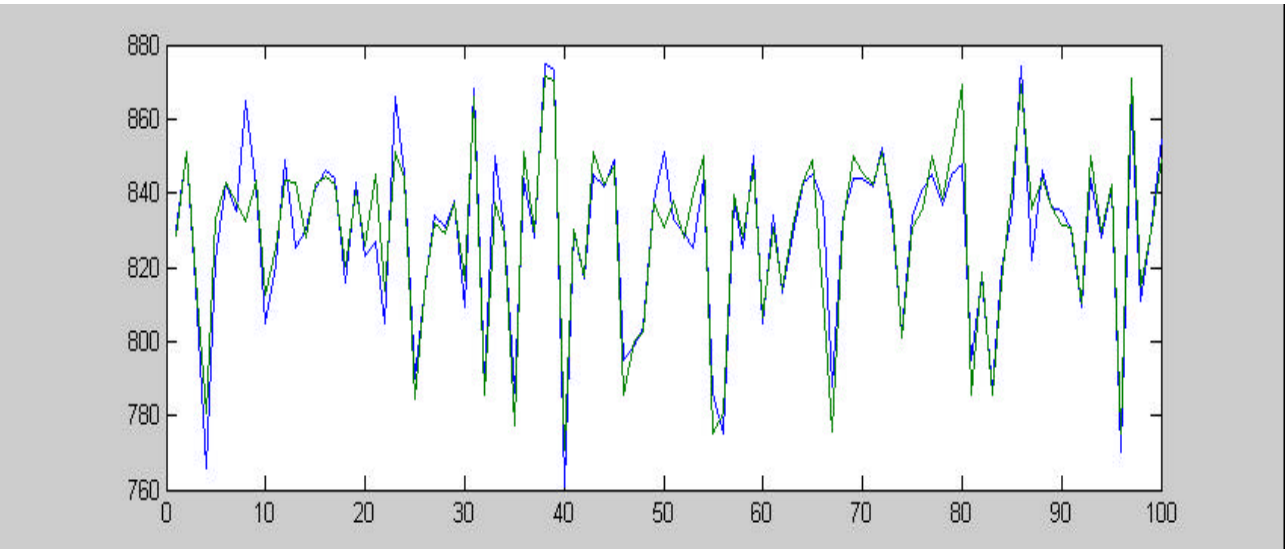


Figura 337. Resultados obtenidos del modelo para los datos de testeo (verde) frente a los datos reales (azul) de THC1.

CREACIÓN Y TESTEO DE LOS MODELOS DE CONSIGNA PARA LA VELOCIDAD DE BANDA

Igual que en el caso anterior, procedemos a crear el modelo que nos permita predecir la velocidad de la banda.

Resultados Obtenidos

Los resultados de la red que predice el valor de la velocidad de la banda:

- MSE de entrenamiento: 0,0009. Es decir, un 3,0% de error.
- MSE de validación: 0,0010. Es decir, un 3,2% de error.
- MSE de testeo: 0,0011. Es decir, un 3,3% de error.

También, se muestran las curvas de velocidad media de banda de los datos de testeo obtenidos aleatoriamente.

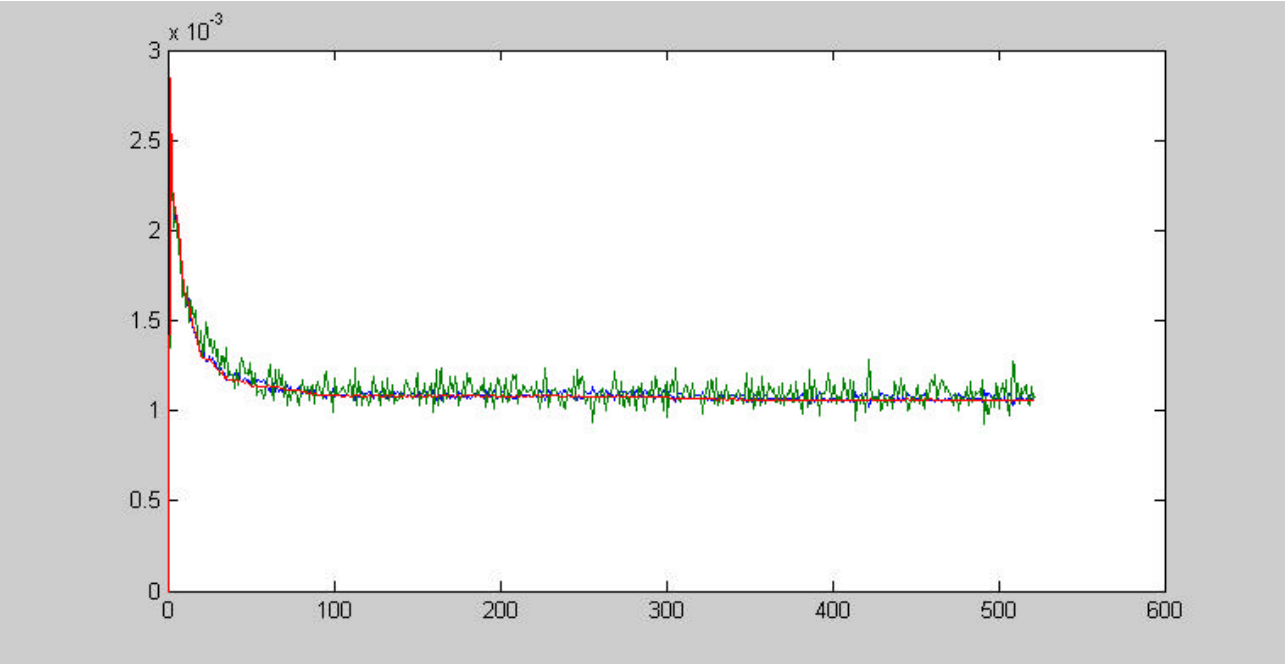


Figura 338. Evolución del error MSE de entrenamiento.

```
% Pesos y Bias de las neuronas de Entrada
% Pesos
net.IW{1}
ans =
   -7.9316    4.0319    1.9041
   -0.5048    6.7993    0.0716
   -0.7488    7.7771    3.6384

% Bias
net.b{1}
ans =
    3.9792
   -2.9977
   -8.6065
% Pesos y Bias de las neuronas de la Capa Oculta
% Pesos
net.LW{2,1}
ans =
    8.0212    8.8927   -8.0810
    5.2194  -12.5276   -3.6015
  -12.6739   -1.5534    0.6125
    9.5877   -2.9078  -10.4005
  -14.3487   -1.3138    2.2178
    5.7260   10.7966    7.8906
   -2.4646  -14.3093    2.1008
    7.7541   10.7063   -7.0648
    8.0174   10.7884    3.4664
    0.8657    3.2296  -14.0068
    7.2689   -9.7704   -7.6685
    8.8843  -11.4580    0.8152
    0.3261    6.1588  -12.9556
   -8.3833   -6.5070  -10.5399
   -1.9220  -13.8509    3.1388
    1.9131    2.7818  -13.9978
   -9.5385    9.6688    4.7818

%Bias
net.b{2}
ans =
  -11.2558
   -1.2194
   14.9447
   -2.6509
   10.3533
  -14.8522
    7.9675
   -4.9488
  -12.4025
    5.8498
    6.9572
    2.7479
    0.3336
    7.2566
    0.9327
   10.9511
   -9.6564
```

```
% Pesos y Bias de las neuronas de la Capa de Salida
% Pesos
net.LW{3,2}
ans =
  Columns 1 through 8
   -1.1946    0.2165   -1.2042    1.2734    1.9749   -0.1642   -0.1551    0.9557
  Columns 9 through 16
    0.6563    0.8658    0.5475   -1.0871   -1.6843   -1.9055   -0.7436    0.6206
  Column 17
   -0.7420

net.b{3}
ans =
    0.2397
```

Figura 339. Pesos y bias de la red neuronal creada.

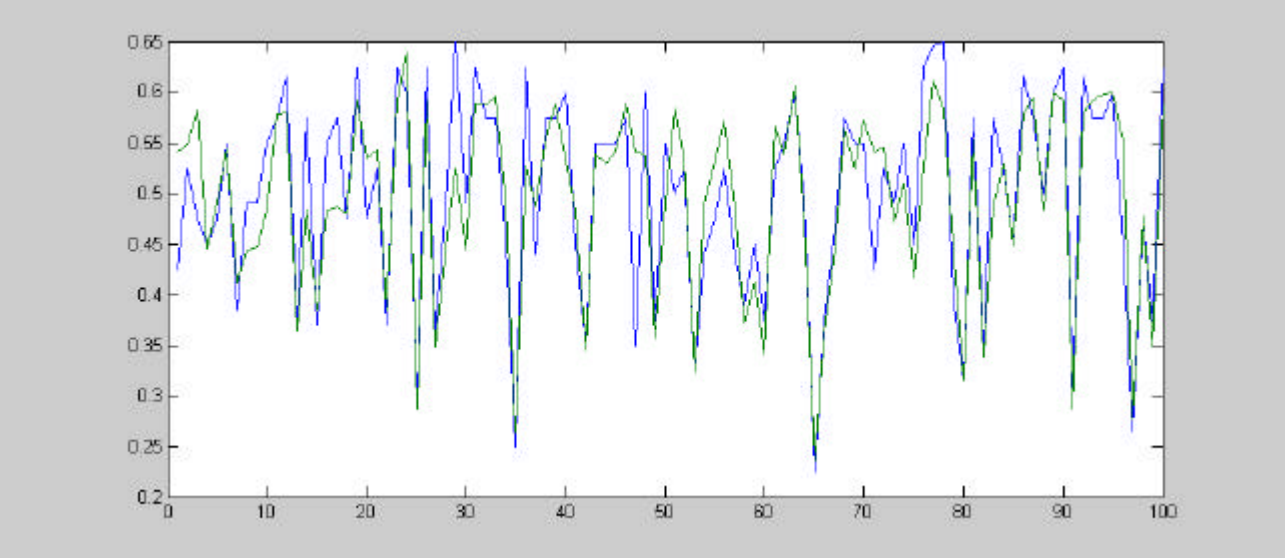


Figura 340. Resultados para los datos del modelo (verde) frente a los datos reales (azul) de VELOCIDADFIN.

```
function MSEVAL=visualiza_todo(MAT,Posini, Longdat)

%Programa que calcula la red neuronal Backpropagation con 9 neuronas
%óptima para obtener un error de entrenamiento inferior a un 0,1
%y un error de generalizaci´on menor del 0,5%
    % Normalizamos la Matriz
    MinimoVectMAT=min(MAT);
    MinimosMAT = ones(size(MAT),1) * min(MAT);
    Vectrang= max(MAT)-min(MAT);
    MATRange = ones(size(MAT),1) * Vectrang;
    MATNORM = (MAT-MinimosMAT)./MATRange;


    % Cargamos la red de zonas del horno
    load ('-MAT','C:\temp\ModeloA\neu17\Modelo Radiadores\MATMEJOR1.MData');
    DatosIn =MATNORM(Posini:(Posini+Longdat-1),2:4);
    DatosOut =MATNORM(Posini:(Posini+Longdat-1),5:7);

    % Simulamos con los datos
```

```
    P=DatosIn';
    T=DatosOut';
    [Y] = sim(net,P);
    MSEVAL = mse(Y-T)

    figure(2);

    % Cargamos la red de velocidad y tmpp2 de consigna
    load ('-MAT','C:\temp\ModeloA\neu17\Modelo Vel y CNGPIR2\MATMEJOR2.MData');
    DatosOut =MATNORM(Posini:(Posini+Longdat-1),8:9);

    % Simulamos con los datos
    T2=DatosOut';
    [Y2] = sim(net,P);
    MSEVAL2 = mse(Y2-T2)

    figure(2);

    % Desnormalizamos los datos de entrada y salida
    % Desnormalizamos los datos de entrada y salida
    P(1,:)=P(1,:)*Vectrang(2)+MinimoVectMAT(2); %ANCHO
    P(2,:)=P(2,:)*Vectrang(3)+MinimoVectMAT(3); %ESPENT
    P(3,:)=P(3,:)*Vectrang(4)+MinimoVectMAT(4); %TMPP1

    T(1,:)=T(1,:)*Vectrang(5)+MinimoVectMAT(5); %THF1
    Y(1,:)=Y(1,:)*Vectrang(5)+MinimoVectMAT(5);
    T(2,:)=T(2,:)*Vectrang(6)+MinimoVectMAT(6); %THF3
    Y(2,:)=Y(2,:)*Vectrang(6)+MinimoVectMAT(6);
    T(3,:)=T(3,:)*Vectrang(7)+MinimoVectMAT(7); %THF5
    Y(3,:)=Y(3,:)*Vectrang(7)+MinimoVectMAT(7);

    T2(1,:)=T2(1,:)*Vectrang(8)+MinimoVectMAT(8); %TMPP2CNG
    Y2(1,:)=Y2(1,:)*Vectrang(8)+MinimoVectMAT(8);
    T2(2,:)=5*(T2(2,:)*Vectrang(9)+MinimoVectMAT(9)); %Velocidad * 5
    Y2(2,:)=5*(Y2(2,:)*Vectrang(9)+MinimoVectMAT(9));

    %Dibujamos Las curvas reales y las obtenidas con los modelos
    figure(1)
    clf
    hold on
    plot(1:Longdat,P(3,:),'g',1:Longdat,T(1,:),'b');
    plot(1:Longdat,T(2,:),'b',1:Longdat,T(3,:),'b');
    plot(1:Longdat,T2(1,:),'r',1:Longdat,T2(2,:),'m');
    ylabel('Temperatura y Velocidad')
    title('Curvas Reales')
    hold off

    figure(2)
    clf
    hold on
    plot(1:Longdat,P(3,:),'g',1:Longdat,Y(1,:),'b');
    plot(1:Longdat,Y(2,:),'b',1:Longdat,Y(3,:),'b');
    plot(1:Longdat,Y2(1,:),'r',1:Longdat,Y2(2,:),'m');
    ylabel('Temperatura y Velocidad')
    title('Curvas Modelizadas')
    hold off
     return
end
```

Figura 341. Programa utilizado para la simulación de los datos mediante las redes obtenidas.

Por último, se muestra un ejemplo de curvas reales frente a las curvas modeladas. Vemos que el modelo predice con bastante precisión las temperaturas y velocidades de consigna a partir de las dimensiones y temperatura de entrada de la banda.

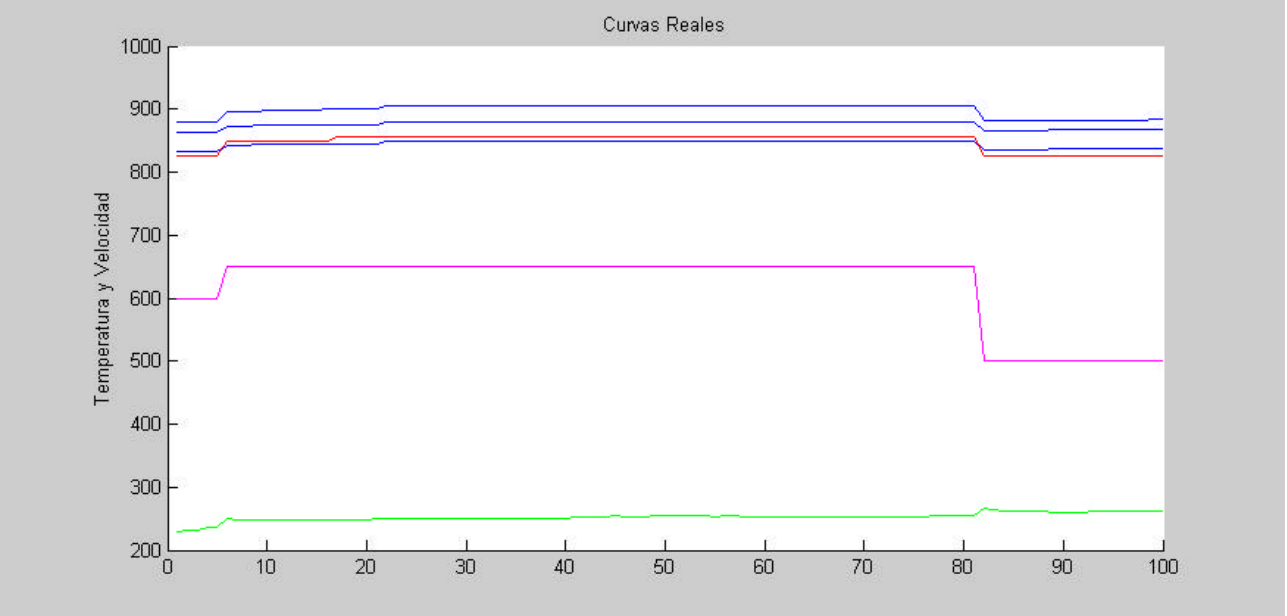


*Figura 342. Curvas reales de temperatura de consigna de zona del horno (THF1, THF3 y THF5 en 'azul'), temperatura de consigna de pirómetro 2 (TMPP2CNG en 'rojo') , velocidad magnificada (VELOCIDADFIN*5 en 'magenta') frente a la temperatura de entrada de banda(TMPP1 en 'verde claro').*

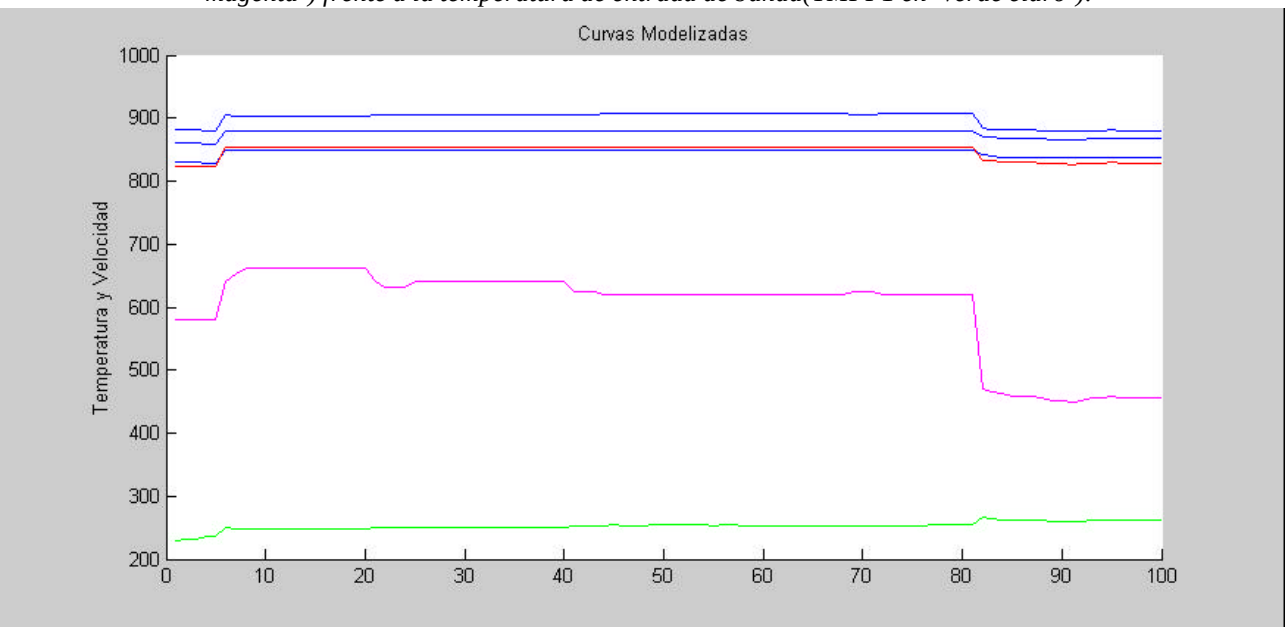


*Figura 343. Curvas modelizadas de temperatura de consigna de zona del horno (THF1, THF3 y THF5 en 'azul'), temperatura de consigna de pirómetro 2 (TMPP2CNG en 'rojo') , velocidad magnificada (VELOCIDADFIN*5 en 'magenta') frente a la temperatura de entrada de banda (TMPP1 en 'verde claro').*

7.3.2.3 GENERACIÓN DE MODELOS NO LINEALES DEL COMPORTAMIENTO DINÁMICO DE LA BANDA

Una vez obtenidos los modelos no lineales para determinar las temperaturas de consigna del horno y velocidades de consigna más adecuadas para la banda según el espesor, anchura y temperatura de entrada de la misma; se pretende obtener un modelo que nos explique el comportamiento dinámico de la banda ante la variaciones de temperatura de consigna del horno y de velocidad.

En este punto se describen los pasos realizados **para la creación de una red neuronal para las bobinas del GRUPO-A que explique la temperatura de salida de la banda *TMPP2(t+1)*, partiendo de las siguientes variables:**

- *THC1(t):* Temperatura de consigna de zona 1.
- *THC3(t):* Temperatura de consigna de zona 3.
- *THC5(t):* Temperatura de consigna de zona 5.
- *ANCHURA(t):* Anchura de la banda:
- *ESPENT(t):* Espesor de la banda.
- *VELMED(t):* Velocidad de la banda.
- *TMPP1(t):* Temperatura de entrada de la banda.
- *TMPP2(t):* Temperatura de salida de la banda.

y las **derivadas de todas estas variables**.

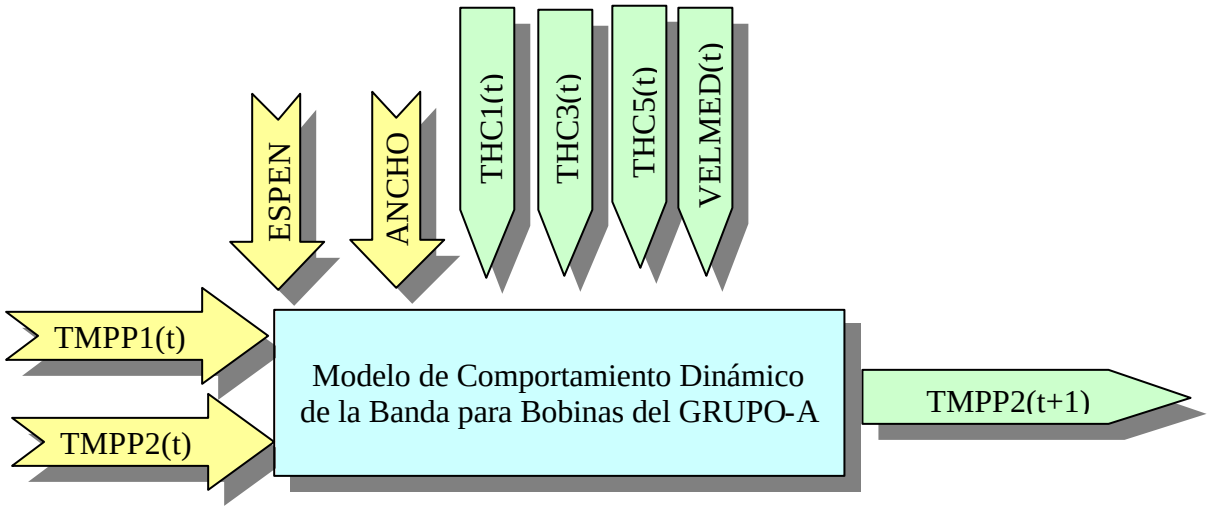


Figura 344. Esquema buscado para la generación de un modelo no lineal que explique la temperatura final de la banda.

CREACIÓN DE LA BASE DE DATOS

Lo primero que realizamos es la base de datos con las variables de históricos de las bobinas del grupo 1.

```
# Obtenemos una matriz con el número de orden de las bobinas
# Obtenemos un número de posición para cada bobina
POSINCREMENBOB <- match(MATDINAMIC2$CODBOBINA,DATBOBINAS$CODBOBINA)

# Obtenemos las bobinas de la familia GRUPO-A
INDGRUPOAESTAC <- MATBOBINAS2$FAMILIA==1

# Obtenemos las bobinas
TIPCOD <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$CODBOBINA))

# Sacamos el índice de las bobinas en la matriz de datos dinámicos
INDGMATDINAMIC <- MATDINAMIC2$CODBOBINA %in% TIPCOD


#Creamos una matriz con las variables a modelizar
CODBOBMATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$CODBOBINA
TMPP1MATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$TMPP1M
THF1MATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$THC1
THF3MATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$THC3
THF5MATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$THC5
TMPP2MATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$TMPP2M
TMPP2CNGMATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$TMPP2C
VELMATSAM4 <- MATDINAMIC2[INDGMATDINAMIC,]$VELOCIDADFIN
BOBINASSEGUI4 <- POSINCREMENBOB[INDGMATDINAMIC]

ERRORSAM4 <- abs(MATDINAMIC2[INDGMATDINAMIC,]$TMPP2C-
MATDINAMIC2[INDGMATDINAMIC,]$TMPP2M)

# Obtenemos un número de posición para cada bobina
POSINCBOB <- match(CODBOBMATSAM4,TIPCOD)
ANCHOBOB <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ANCHO))
ESPENTBOB <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ESPENT))

# Creamos la anchura y espesor
ANCHOMATSAM4 <- ANCHOBOB[POSINCBOB]
ESPENTMATSAM4 <- round(ESPENTBOB[POSINCBOB]*1000)


# Obtenemos el DIFERENCIAL
LONGIT <- length(TMPP1MATSAM4)

DIFTMPP1MATSAM4 <- TMPP1MATSAM4[2:LONGIT]- TMPP1MATSAM4[1:(LONGIT-1)]
DIFTMPP2MATSAM4 <- TMPP2MATSAM4[2:LONGIT]- TMPP2MATSAM4[1:(LONGIT-1)]
DIFTMPP2CNGMATSAM4 <- TMPP2CNGMATSAM4 [2:LONGIT]- TMPP2CNGMATSAM4 [1:(LONGIT-1)]
DIFTHF1MATSAM4 <- THF1MATSAM4[2:LONGIT]- THF1MATSAM4[1:(LONGIT-1)]
DIFTHF3MATSAM4 <- THF3MATSAM4[2:LONGIT]- THF3MATSAM4[1:(LONGIT-1)]
DIFTHF5MATSAM4 <- THF5MATSAM4[2:LONGIT]- THF5MATSAM4[1:(LONGIT-1)]
DIFVELMATSAM4 <- VELMATSAM4 [2:LONGIT]- VELMATSAM4 [1:(LONGIT-1)]
DIFANCHOMATSAM4 <- ANCHOMATSAM4[2:LONGIT]- ANCHOMATSAM4[1:(LONGIT-1)]
DIFESPENTMATSAM4 <- ESPENTMATSAM4[2:LONGIT]- ESPENTMATSAM4[1:(LONGIT-1)]
DIFBOBINASSEGUI4 <- BOBINASSEGUI4[2:LONGIT]- BOBINASSEGUI4[1:(LONGIT-1)]
```

```
# Eliminamos el ultimo dato
# Obtenemos el DIFERENCIAL
LONGIT <- length(TMPP1MATSAM4)

CODBOBMATSAM4 <- CODBOBMATSAM4[1:(LONGIT-1)]
TMPP1MATSAM4 <- TMPP1MATSAM4[1:(LONGIT-1)]
TMPP2MATSAM4 <- TMPP2MATSAM4[1:(LONGIT-1)]
TMPP2CNGMATSAM4 <- TMPP2CNGMATSAM4 [1:(LONGIT-1)]
THF1MATSAM4 <- THF1MATSAM4[1:(LONGIT-1)]
THF3MATSAM4 <- THF3MATSAM4[1:(LONGIT-1)]
THF5MATSAM4 <- THF5MATSAM4[1:(LONGIT-1)]
VELMATSAM4 <- VELMATSAM4 [1:(LONGIT-1)]
ANCHOMATSAM4 <- ANCHOMATSAM4[1:(LONGIT-1)]
ESPENTMATSAM4 <- ESPENTMATSAM4[1:(LONGIT-1)]
BOBINASSEGUI4 <- BOBINASSEGUI4[1:(LONGIT-1)]

# SALIDA
TMPP2MATSAM4SALIDA <- TMPP2MATSAM4[2:LONGIT]

# CREAMOS UNA MATRIZ CON LOS DATOS y OTRA CON SUS DIFERENCIALES

MATSAM4 <- cbind(CODBOBMATSAM4, ANCHOMATSAM4, ESPENTMATSAM4, VELMATSAM4,
TMPP1MATSAM4, THF1MATSAM4, THF3MATSAM4, THF5MATSAM4, TMPP2MATSAM4,
TMPP2MATSAM4SALIDA, TMPP2CNGMATSAM4, BOBINASSEGUI4)

DIFMATSAM4 <- cbind(CODBOBMATSAM4, DIFANCHOMATSAM4, DIFESPENTMATSAM4,
DIFVELMATSAM4, DIFTMPP1MATSAM4, DIFTHF1MATSAM4, DIFTHF3MATSAM4, DIFTHF5MATSAM4,
DIFTMPP2CNGMATSAM4, DIFTMPP2MATSAM4, DIFBOBINASSEGUI4)

# Eliminamos los espúreos y datos de bobinas NO seguidas
INDGHT <- MATSAM4[,4]>10 & MATSAM4[,5]>100 & MATSAM4[,6]>100 & MATSAM4[,7]>100 &
MATSAM4[,8]>100 & MATSAM4[,9]>100 & MATSAM4[,10]>100 & MATSAM4[,11]>100 &
DIFBOBINASSEGUI4<2 & abs(DIFMATSAM4[,4])<200 & abs(DIFMATSAM4[,5])<200 &
abs(DIFMATSAM4[,6])<200 & abs(DIFMATSAM4[,7])<200 & abs(DIFMATSAM4[,8])<200 &
abs(DIFMATSAM4[,9])<200 & abs(DIFMATSAM4[,10])<200

MATSAM4SIN <- MATSAM4[INDGHT,]
DIFMATSAM4SIN <- DIFMATSAM4[INDGHT,]

# Eliminamos las observaciones con variaciones muy pequeñas <=2ºC
INDGHT <- abs(DIFMATSAM4SIN[,4])>2 | abs(DIFMATSAM4SIN[,5])>2 |
abs(DIFMATSAM4SIN[,6])>2 | abs(DIFMATSAM4SIN[,7])>2 | abs(DIFMATSAM4SIN[,8])>2 |
abs(DIFMATSAM4SIN[,9])>2 | abs(DIFMATSAM4SIN[,10])>2

MATSAM4SIN <- MATSAM4SIN[INDGHT,]
DIFMATSAM4SIN <- DIFMATSAM4SIN[INDGHT,]

dim(MATSAM4SIN)
[1] 6589   12
dim(DIFMATSAM4SIN)
[1] 6589   11
```

```
summary(MATSAM4SIN)
 CODBOBMATSAM4       ANCHOMATSAM4  ESPENTMATSAM4      VELMATSAM4
 Min.   :23293013   Min.   : 750   Min.   : 476.0   Min.   : 21.0
 1st Qu.:23393009   1st Qu.:1000   1st Qu.: 675.0   1st Qu.: 82.0
 Median :23483035   Median :1175   Median : 775.0   Median :109.0
 Mean   :23483406   Mean   :1169   Mean   : 877.8   Mean   :101.4
 3rd Qu.:23573044   3rd Qu.:1333   3rd Qu.: 975.0   3rd Qu.:120.0
 Max.   :23653024   Max.   :1525   Max.   :2000.0   Max.   :150.0
  TMPP1MATSAM4    THF1MATSAM4   THF3MATSAM4     THF5MATSAM4     TMPP2MATSAM4
 Min.   :207.0   Min.   :716   Min.   :744.0   Min.   :616.0   Min.   :714.0
 1st Qu.:243.0   1st Qu.:805   1st Qu.:835.0   1st Qu.:854.0   1st Qu.:806.0
 Median :254.0   Median :825   Median :856.0   Median :877.0   Median :823.0
 Mean   :253.6   Mean   :822   Mean   :852.3   Mean   :872.2   Mean   :818.5
 3rd Qu.:266.0   3rd Qu.:841   3rd Qu.:873.0   3rd Qu.:891.0   3rd Qu.:832.0
 Max.   :364.0   Max.   :877   Max.   :907.0   Max.   :933.0   Max.   :889.0
 TMPP2MATSAM4SALIDA TMPP2CNGMATSAM4 BOBINASSEGUI4
 Min.   :714.0      Min.   :725.0   Min.   :   8
 1st Qu.:806.0      1st Qu.:810.0   1st Qu.: 518
 Median :823.0      Median :825.0   Median :1041
 Mean   :818.6      Mean   :818.3   Mean   :1035
 3rd Qu.:832.0      3rd Qu.:825.0   3rd Qu.:1564
 Max.   :889.0      Max.   :865.0   Max.   :1979

summary(DIFMATSAM4SIN)
 CODBOBMATSAM4      DIFANCHOMATSAM4      DIFESPENTMATSAM4
 Min.   :23293013   Min.   :-230.00000   Min.   :-329.00000
 1st Qu.:23393009   1st Qu.:   0.00000   1st Qu.:   0.00000
 Median :23483035   Median :   0.00000   Median :   0.00000
 Mean   :23483406   Mean   :   0.04401   Mean   :  -0.02459
 3rd Qu.:23573044   3rd Qu.:   0.00000   3rd Qu.:   0.00000
 Max.   :23653024   Max.   : 255.00000   Max.   : 510.00000
 DIFVELMATSAM4       DIFTMPP1MATSAM4     DIFTHF1MATSAM4
 Min.   :-138.0000   Min.   :-83.00000   Min.   :-76.000000
 1st Qu.:   0.0000   1st Qu.: -1.00000   1st Qu.:  0.000000
 Median :   0.0000   Median :  0.00000   Median :  0.000000
 Mean   :  -0.2879   Mean   :  0.08059   Mean   : -0.007437
 3rd Qu.:   0.0000   3rd Qu.:  1.00000   3rd Qu.:  0.000000
 Max.   :  74.0000   Max.   : 96.00000   Max.   : 80.000000
 DIFTHF3MATSAM4      DIFTHF5MATSAM4      DIFTMPP2CNGMATSAM4
 Min.   :-76.00000   Min.   :-90.00000   Min.   :-1.000e+02
 1st Qu.:  0.00000   1st Qu.:  0.00000   1st Qu.: 0.000e+00
 Median :  0.00000   Median :  0.00000   Median : 0.000e+00
 Mean   : -0.02064   Mean   : -0.07831   Mean   : 9.713e-03
 3rd Qu.:  0.00000   3rd Qu.:  1.00000   3rd Qu.: 0.000e+00
 Max.   : 80.00000   Max.   :100.00000   Max.   : 1.000e+02
 DIFTMPP2MATSAM4    DIFBOBINASSEGUI4
 Min.   :-96.0000   Min.   :0.00000
 1st Qu.: -2.0000   1st Qu.:0.00000
 Median :  0.0000   Median :0.00000
 Mean   :  0.0856   Mean   :0.07148
 3rd Qu.:  2.0000   3rd Qu.:0.00000
 Max.   : 93.0000   Max.   :1.00000

# Guardamos la matriz en un archivo csv
write.table(MATSAM4SIN,"c:\\temp\\DINMATENE2003TODO.CSV",quote=FALSE,sep=",",row
.names=FALSE,col.names=FALSE)
write.table(DIFMATSAM4SIN,"c:\\temp\\DINDIFMATENE2003TODO.CSV",quote=FALSE,sep="
,",row.names=FALSE,col.names=FALSE)
```

Figura 345. Programa que obtiene los datos en régimen dinámico de la base de datos de históricos.

REDUCCIÓN DE LA DIMENSIÓN DE LOS DATOS MEDIANTE PCA

Como se puede apreciar en la Figura 345, **el número de variables de entrada de la red neuronal, si consideramos las variables y sus derivadas, es demasiado elevado (15)**. Además, tal y como se ha concluido en los capítulos 5 y 6, muchas de ellas son muy dependientes entre si.

Para poder reducir el número de entradas, surgen diversos planteamientos de reducción de dimensión. De ellos, se determina como muy buena opción el uso de PCA o NLPCA para reducir el número de éstas [TAN95][KRA91] debido a que:

- Se generan variables linealmente independientes.
- Podemos elegir el número de variables nuevas que podemos crear según el porcentaje de varianza explicada.

Por lo tanto, para reducir el número de variables, se tratarán mediante PCA los datos correspondientes a la velocidad de banda y temperatura de consigna de zonas del horno y, separadamente, los datos correspondientes a la diferencia de estos valores [LI02][BRA00].

```
# Escalamos las variables antes de hacer la proyección PCA
MATSAM4SIN[,2] <- MATSAM4SIN[,2]/2000 #ANCHOMATSAM4
MATSAM4SIN[,3] <- MATSAM4SIN[,3]/2500 #ESPENTMATAM4
MATSAM4SIN[,4] <- MATSAM4SIN[,4]/200 #VELMATSAM4
MATSAM4SIN[,5] <- MATSAM4SIN[,5]/400 #TMPP1MATSAM4
MATSAM4SIN[,6] <- MATSAM4SIN[,6]/950 #THF1MATSAM4
MATSAM4SIN[,7] <- MATSAM4SIN[,7]/950 #THF3MATSAM4
MATSAM4SIN[,8] <- MATSAM4SIN[,8]/950 #THF5MATSAM4
MATSAM4SIN[,9] <- MATSAM4SIN[,9]/950 #TMPP2MATSAM4

DIFMATSAM4SIN[,2] <- DIFMATSAM4SIN[,2]/1500 #DIFANCHOMATSAM4
DIFMATSAM4SIN[,3] <- DIFMATSAM4SIN[,3]/1000 #DIFESPENTMATSAM4
DIFMATSAM4SIN[,4] <- DIFMATSAM4SIN[,4]/100 #DIFVELOCIDAD
DIFMATSAM4SIN[,5] <- DIFMATSAM4SIN[,5]/150 #DIFTMPP1
DIFMATSAM4SIN[,6] <- DIFMATSAM4SIN[,6]/150 #DIFTHF1
DIFMATSAM4SIN[,7] <- DIFMATSAM4SIN[,7]/150 #DIFTHF3
DIFMATSAM4SIN[,8] <- DIFMATSAM4SIN[,8]/150 #DIFTHF5


# Obtenemos la proyección PCA de MATSAM4SIN
PCASAM4 <- pca(as.matrix(MATSAM4SIN[,2:9]),method=2)

# Obtenemos el vector medias
VECTMEANMATSAM4 <- apply(MATSAM4SIN[,2:9],2,mean)
VECTMEANMATSAM4
VECTMEANMATSAM4
 ANCHOMATSAM4 ESPENTMATSAM4    VELMATSAM4  TMPP1MATSAM4   THF1MATSAM4
    0.5845106     0.3511300     0.5072181     0.6341091     0.8652004
  THF3MATSAM4   THF5MATSAM4  TMPP2MATSAM4
    0.8971105     0.9180641     0.8616191



# Vemos el grado de información de cada eje
# (los cuatro primeros abarcan el 97,1% de la varianza)
```

```
PCASAM4$evals/sum(PCASAM4$evals)
[1] 0.538337376 0.230677625 0.154631401 0.047843064 0.022580848 0.004116602
[7] 0.001813085

0.538337376+0.230677625+0.154631401+0.047843064
[1] 0.9714895

PCASAM4$evecs[,1:4]
                    Comp1       Comp2       Comp3       Comp4
ANCHOMATSAM4  -0.12078515 -0.92698748 -0.34976407 -0.03844801
ESPENTMATSAM4 -0.65939540  0.30735305 -0.54237010 -0.36464676
VELMATSAM4     0.72356908  0.15268884 -0.63215398 -0.13202124
TMPP1MATSAM4  -0.11553051  0.04678752 -0.08084021  0.67370815
THF1MATSAM4   -0.05886373  0.08421364 -0.22919670  0.26809930
THF3MATSAM4   -0.06294954  0.08016659 -0.23705516  0.28426244
THF5MATSAM4   -0.06110113  0.08362295 -0.24323174  0.30329526
TMPP2MATSAM4  -0.05047979 -0.01462697 -0.09725239  0.38674517

# Guardamos la matriz en un archivo csv
write.table(VECTMEANMATSAM4,"c:\\temp\\
VECTMEANENE2003.CSV",quote=FALSE,sep=",",row.names=FALSE,col.names=FALSE)
write.table(PCASAM4$evecs[,1:4],"c:\\temp\\EJESPCAENE2003.CSV",quote=FALSE,sep="
,",row.names=FALSE,col.names=FALSE)

# Obtenemos la proyección PCA de DIFSAM4
PCADIFSAM4 <- pca(as.matrix(DIFMATSAM4SIN[,2:8]),method=2)

# Obtenemos el vector medias
VECTMEANDIFMATSAM4 <- apply(DIFMATSAM4SIN[,2:8],2,mean)
VECTMEANDIFMATSAM4
 DIFANCHOMATSAM4 DIFESPENTMATSAM4    DIFVELMATSAM4  DIFTMPP1MATSAM4
    2.934183e-05    -2.458643e-05    -2.879041e-03     5.372591e-04
  DIFTHF1MATSAM4   DIFTHF3MATSAM4   DIFTHF5MATSAM4
   -4.957758e-05    -1.376031e-04    -5.220823e-04

# Vemos el grado de información de cada eje
# (los cinco primeros abarcan el 97% de la varianza)
PCADIFSAM4$evals/sum(PCADIFSAM4$evals)
[1] 0.539167458 0.278072097 0.089175805 0.033782937 0.029628673 0.022454528
[7] 0.007718503
0.539167458+0.278072097+0.089175805+0.033782937+0.029628673
[1] 0.969827

# Visualizamos los cinco vectores PCA principales
PCADIFSAM4$evecs[,1:5]
                       Comp1         Comp2        Comp3         Comp4
DIFANCHOMATSAM4  -0.002838288 -0.0003587765  0.005360753 -0.0450454089
DIFESPENTMATSAM4  0.005561315  0.0046497232 -0.024384568  0.2539561973
DIFVELMATSAM4    -0.998781431 -0.0476417699 -0.011232666  0.0004320745
DIFTMPP1MATSAM4  -0.004841414 -0.1297495214  0.990692498  0.0361086620
DIFTHF1MATSAM4   -0.027132397  0.5357164843  0.044318661  0.6553499024
DIFTHF3MATSAM4   -0.028162189  0.5788925764  0.079439271  0.1222062792
DIFTHF5MATSAM4   -0.029050895  0.5989697890  0.097489052 -0.6983947806
                       Comp5
DIFANCHOMATSAM4   0.178468235
DIFESPENTMATSAM4 -0.938124039
DIFVELMATSAM4    -0.005860737
DIFTMPP1MATSAM4  -0.017839648
DIFTHF1MATSAM4    0.099504509
DIFTHF3MATSAM4    0.152660505
```

```
DIFTHF5MATSAM4   -0.233481046

# Guardamos la matriz en un archivo csv
write.table(VECTMEANDIFMATSAM4,"c:\\temp\\
VECTMEANDIFENE2003.CSV",quote=FALSE,sep=",",row.names=FALSE,col.names=FALSE)
write.table(PCADIFSAM4$evecs[,1:5],"c:\\temp\\EJESPCADIFENE2003.CSV",quote=FALSE
,sep=",",row.names=FALSE,col.names=FALSE)

# Creamos una matriz con la proyección de los puntos de entrada y la salida
MATFINSAM4 <- cbind(PCASAM4$rproj[,1:4], PCADIFSAM4$rproj[,1:5],
MATSAM4SIN[,10])

write.table(MATFINSAM4,"c:\\temp\\NEURONALPROYPCAENE2003.CSV",quote=FALSE,sep=",
",row.names=FALSE,col.names=FALSE)
```

Figura 346. Obtención de los ejes PCA de las variables de entrada de la red neuronal.

En la Figura 346 podemos ver los ejes obtenidos para los dos grupos de variables:

- Variables de temperatura de consigna de horno y velocidad: Cuatro ejes principales PCA que abarcan 97,1% de la varianza de los datos.
- Variables de derivadas de temperatura de consigna y velocidad: Cinco ejes principales PCA que abarcan 97% de la varianza de los datos.

De esta forma, **seleccionando los ejes principales de cada proyección PCA para cada grupo, podemos reducir el número de entradas de la red neuronal a nueve**.

CREACIÓN Y TESTEO DEL MODELO DE COMPORTAMIENTO DE LA BANDA

Volvemos a realizar y testear la red neuronal mediante la librería *Neural Network Toolbox, versión 4 (R12)* de Matlab®.

Igual que en el punto anterior, creamos una base de datos de testeo con el 20% de los datos, y el 80% restante se utiliza mediante correlación cruzada con el 80% para entrenamiento y el 20% para evaluación. Usamos 6.589 patrones para entrenar y validar la red neuronal.

Se entrenan gran cantidad de redes con capas intermedias entre 10 y 19 neuronas, y se almacenan las que mejores resultados dan en la comprobación con el test. Es decir, se guardan aquellas que mejor generalizan.

Neuronas	MINENT	MINEVAL	MINTEST
11	0,001272	0,001577	0,002315
12	0,001269	0,001671	0,002193
13	**0,001280**	**0,001741**	**0,001879**
14	0,001219	0,001603	0,002395
15	0,001201	0,001543	0,002378
16	0,001270	0,001672	0,002445
17	0,001136	0,001899	0,002067
18	0,001121	0,001674	0,002041
19	0,001125	0,001679	0,002508
20	0,001357	0,001846	0,002360

Tabla 70. Algunos resultados de las mejores redes obtenidas.

Resultados Obtenidos

En la Tabla 70, se muestran los errores mínimos de entrenamiento, validación y testeo de redes con diferentes tamaños de capa intermedia. Se elige la red con 13 neuronas, debido a los pequeños errores de testeo que presenta.

Posteriormente, con el objetivo de reducir el error, se desarrolla un entrenamiento más exhaustivo. De esta forma, los errores de la red que predice el valor de temperatura de la banda a la salida de la zona de calentamiento del horno a partir de los datos anteriores de temperaturas de consigna de horno y velocidad, y sus derivadas, son los siguientes:

- MSE de entrenamiento: 0,0019. Es decir, un 4,36% de error.
- MSE de validación: 0,0020. Es decir, un 4,47% de error.
- MSE de testeo: 0,0033. Es decir, un 5,74% de error.

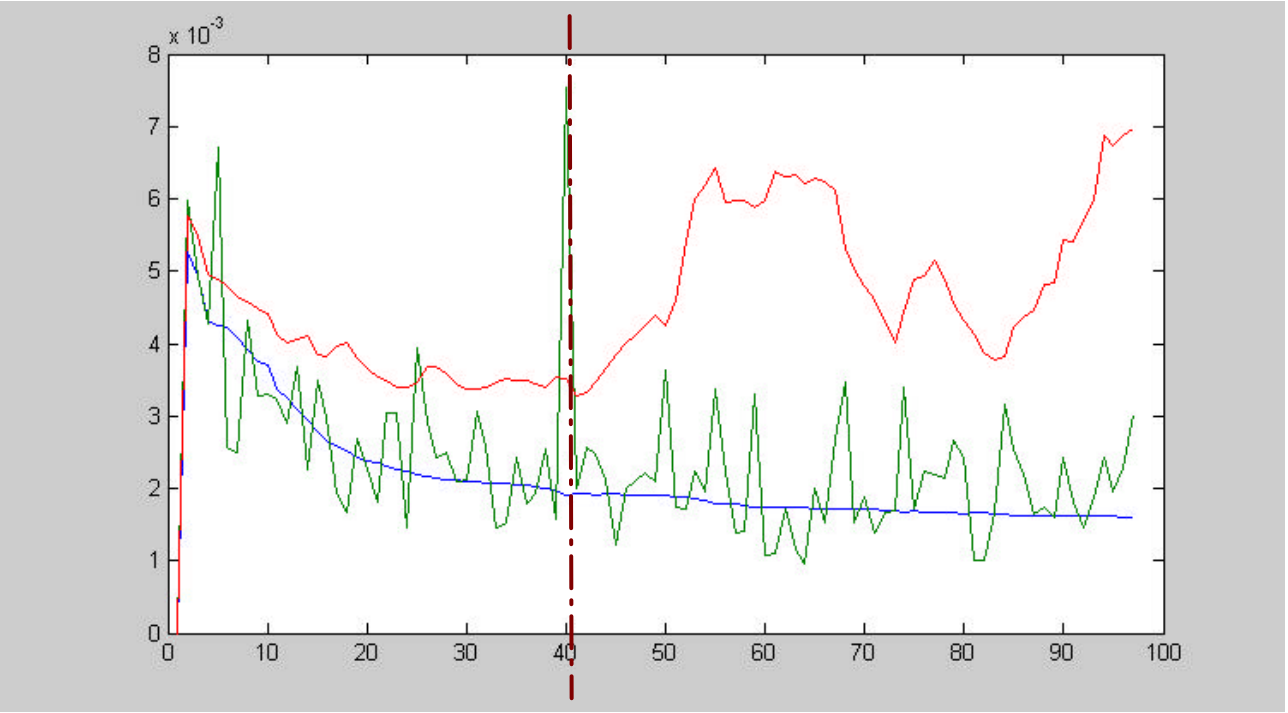


Figura 347. Evolución del error MSE de entrenamiento (azul) frente al de validación (verde) y testeo (rojo).

```
% Pesos y Bias de las neuronas de Entrada
% Pesos

net.IW{1}
ans =
  Columns 1 through 6
    3.5923   -3.8204   -2.8930    2.3089   -4.2397    0.6458
   -3.7692    2.5836    1.5991   -3.6453    4.5431    2.0097
    1.4393   -5.1963    2.3511    4.8913   -0.4311   -0.3866
   -2.3715    0.6372   -6.0295   -3.4115    0.0447    1.8556
   -0.3026    4.6336  -10.8990   -1.9215    0.9728    2.9089
   -1.4980   -0.6009    1.5468    0.7674    0.4256   -1.0215
   -5.7002    3.1698    4.3902   -0.1732    4.4466    1.5178
   10.1273   -2.7825   -0.6430   -2.5181   -0.5465   -2.8244
    0.5637   -1.4588   -1.6911   -2.2618   -0.0638   -1.3940
  Columns 7 through 9
    7.7244   -2.4034    2.3541
    3.0496    7.4012   -1.9029
    1.6741    1.4960    4.5480
    0.6413   -0.4675   -3.8961
    4.1417    0.4901    5.1314
    5.3588    4.7968   -8.8028
   -1.9384    2.1623    1.5233
    4.3070    0.2905   -1.2965
    6.2426    4.3253    3.1146

net.b{1}
ans =
   -5.0108
   -0.8078
   -3.3998
```

```
    9.1642
   -1.8386
    2.2772
   -4.1462
   -0.0426
   -4.3976

% Pesos y Bias de las neuronas de la Capa Oculta
% Pesos

net.LW{2,1}
ans =
  Columns 1 through 6
    2.3099    6.9278    0.3963   -6.9549    4.2444   -3.8599
    4.5577   -2.7330   -1.8825   -2.8705    1.2360    1.4588
   -2.8158   -5.0774   -0.7618   -0.1142    0.5508   -3.2939
    1.8791    0.8223   -2.9092   -5.8118    0.7441   -1.8765
    3.0651   -3.7357   -2.2638    4.5006   -5.4867   -1.5499
   -1.4499    0.0896   -1.1707   -8.2281    1.4571    1.9116
    4.5107   -3.8882    2.1876    0.4845   -0.3023    0.1669
    0.6472    4.5205    1.3333   -3.6365    4.7886    5.7659
   -4.1809   -2.7353   -2.5971   -2.2520    2.4999   -0.5857
   -5.3318    2.4287    0.7461   -3.3134   -0.4482   -2.0623
   -5.8518   -5.4913    2.9460    5.1878    3.3080    2.1880
   -5.9974    0.9175   -3.0688    1.7383    2.6631   -4.1576
    1.2999    2.6133   -2.2299    0.7588   -0.6056    2.1733



 Columns 7 through 9
   -5.8070   -1.0977   -7.2809
    3.3755    4.9080   -3.7749
   -0.8380    1.2652    0.4942
   -3.5446    4.1991    1.8547
   -2.5324    2.6421   -1.0032
   -0.0389    1.7457   -1.8826
    4.6069   -1.5640   -1.8684
   -2.5115   -0.5122   -3.7713
    0.7598   -4.5488    0.9243
    3.8664   -6.6503   -1.1641
   -3.4193   -3.4203   -2.8018
    0.1065   -0.0766   -2.3446
    0.7720    3.6683    1.9427

%Bias
net.b{2}
>> net.b{2}
ans =
   -0.7253
   -3.6944
    9.0865
   -1.7918
    1.2853
    8.0592
   -3.5402
    0.5804
    1.5315
   -3.0101
   -3.9998
```

```
   -6.0717
   -6.2815
% Pesos y Bias de las neuronas de la Capa de Salida
% Pesos
net.LW{3,2}
ans =
  Columns 1 through 6
   -1.0447    0.5405   -1.3829    0.5378   -0.2445    2.0484
  Columns 7 through 12
   -1.7292    1.9029   -2.0241    2.5722   -1.8746   -1.0880
  Column 13
   -0.7063
net.b{3}
ans =
   -1.6558
```

Figura 348. Pesos y bias de la red neuronal creada.

Los pasos anteriores han mostrado las fases de entrenamiento y validación del modelo que explica el comportamiento de la banda ante variaciones de temperatura y velocidad.

Este modelo permite simular previamente cómo se va a comportar la banda, según diferentes cambios de consigna, cuando está compuesta por bobinas del GRUPO A.

Estos mismos pasos deberán emplearse con las bobinas de los demás grupos, y así también, poder simular el sistema ante variaciones de aceros.

7.3.3 SIMULACIÓN DEL PROCESO MEDIANTE EL USO DE LOS MODELOS NO LINEALES OBTENIDOS

Una vez obtenida la red neuronal, procedemos a realizar la simulación del comportamiento de la banda frente al real.

En el programa de la figura siguiente se desarrolla la simulación de la banda comparándola con el comportamiento real.

```
function MSEVAL=simula_FINAL_MARZO2003(MATSAM4SIN,DIFMATSAM4SIN,Posini, Longdat)

%Programa que simula

%Cargamos los datos de MAT
% COL 1=CODBOBMATSAM4
% COL 2=ANCHOMATSAM4
% COL 3=ESPENTMATSAM4
% COL 4=VELMATSAM4
% COL 5=TMPP1MATSAM4
% COL 6=THF1MATSAM4
% COL 7=THF3MATSAM4
% COL 8=THF5MATSAM4
% COL 9=TMPP2MATSAM4
% COL 10=TMPP2MATSAM4SALIDA
% COL 11=TMPP2CNGMATSAM4

%MAT = csvread('c:\\temp\\NEURONALPERM.CSV');
%Guardamos la matriz en un archivo csv
%MATSAM4SINENE=csvread('c:\\temp\\DINMATENE2003.CSV');
%DIFMATSAM4SINENE=csvread('c:\\temp\\DINDIFMATENE2003.CSV');

    %Dibujamos las curvas reales
    PPI1=MATSAM4SIN(Posini:(Posini+Longdat-1),5)';   %Temp Piro1
    TRAD=MATSAM4SIN(Posini:(Posini+Longdat-1),6:8)'; %THF1, THF3, THF5
    TCNG2=MATSAM4SIN(Posini:(Posini+Longdat-1),11)'; %TCNG2
    TPI2=MATSAM4SIN(Posini:(Posini+Longdat-1),9)'; %Temp Piro2
    VELO=5*MATSAM4SIN(Posini:(Posini+Longdat-1),4)'; %Velocidad

    figure(3)
    clf
    hold on
    plot(1:Longdat,PPI1,'g',1:Longdat,TRAD(1,:),'b:');
    plot(1:Longdat,TRAD(1,:),'b:');
    plot(1:Longdat,TRAD(2,:),'b:',1:Longdat,TRAD(3,:),'b:');
    plot(1:Longdat,TCNG2,'r',1:Longdat,VELO,'m');
    plot(1:Longdat,TPI2,'k');
    ylabel('Temperatura y Velocidad')
    title('Curvas Reales')
    hold off



    %Cargamos la red neuronal de comportamiento dinamico
    load ('-MAT','C:\temp\ModeloA\dinamicoFINAL\MATMEJORTEST1.mat');
```

```
netdina = net;


% Calculamos la simulacion de las curvas con las redes neuronales
% Obtenemos con las redes neuronales la THFC1, THFC3 y THFC5 de zonas
% Y la velocidad y temp de consigna de pirometro2
% Normalizamos la Matriz

MinimoVectMAT= [10000000, 700, 0, 100, 700, 700, 700, 700, 10, 0];
Vectrang=[30000000, 1300, 2500, 300, 300, 300, 300, 300, 200, 200];

%Creamos las variables ANCH, TMPP1 y ESPENT normalizadas
TMPP1ESPERADA = ones(Longdat,1)*MATSAM4SIN(Posini,5);
DatosIn =[MATSAM4SIN(Posini:(Posini+Longdat-1),2:3) TMPP1ESPERADA];

MinimosMAT = ones(size(DatosIn),1) * MinimoVectMAT(2:4);
MATRange = ones(size(DatosIn),1) * Vectrang(2:4);
DatosInNorm = (DatosIn-MinimosMAT)./MATRange;

% Cargamos la red de THCxs
load ('-MAT','C:\temp\ModeloA\consignasTEMP\MATMEJORTEST11.MData');
netrad = net;

% Simulamos con los datos
P=DatosInNorm';
[Y] = sim(netrad,P);

% Cargamos la red de velocidad y tmpp2 de consigna
load ('-MAT','C:\temp\ModeloA\consignasVEL\MATMEJORTEST13.MData');
netvel = net;

% Simulamos con los datos
[Y2] = sim(netvel,P);


% Desnormalizamos los datos de entrada y salida
% Desnormalizamos los datos de entrada y salida
ANCHO=P(1,:)*Vectrang(2)+MinimoVectMAT(2); %ANCHO
ESPENT=P(2,:)*Vectrang(3)+MinimoVectMAT(3); %ESPENT
PPI1=P(3,:)*Vectrang(4)+MinimoVectMAT(4); %TMPP1


TRAD1=Y(1,:)*Vectrang(5)+MinimoVectMAT(5); %THF1
TRAD3=Y(2,:)*Vectrang(6)+MinimoVectMAT(6); %THF3
TRAD5=Y(3,:)*Vectrang(7)+MinimoVectMAT(7); %THF5
TCNG2=Y(4,:)*Vectrang(8)+MinimoVectMAT(8); %TMPP2CNG

VELO=(Y2(1,:)*Vectrang(9)+MinimoVectMAT(9)); %Velocidad

% ---------------------------------------------------------

%Obtenemos las curvas reales
ANCHO=MATSAM4SIN(Posini:(Posini+Longdat-1),2)';   %ANCHO
ESPENT=MATSAM4SIN(Posini:(Posini+Longdat-1),3)';   %ESPENT
VELO=MATSAM4SIN(Posini:(Posini+Longdat-1),4)'; %Velocidad
PPI1=MATSAM4SIN(Posini:(Posini+Longdat-1),5)';   %Temp Piro1
TRAD1=MATSAM4SIN(Posini:(Posini+Longdat-1),6)'; %THF1, THF3, THF5
TRAD3=MATSAM4SIN(Posini:(Posini+Longdat-1),7)'; %THF1, THF3, THF5
TRAD5=MATSAM4SIN(Posini:(Posini+Longdat-1),8)'; %THF1, THF3, THF5
TPI2_REAL=MATSAM4SIN(Posini:(Posini+Longdat-1),9)'; %Temp Piro2
```

```
    TCNG2=MATSAM4SIN(Posini:(Posini+Longdat-1),11)'; %TCNG2

    % Creamos los vectores medias y ejes del PCA para comprimir los datos de
entrada de la red neuronal din´amica

    VecMedSAM4 = csvread('c:\\temp\\VECTMEANENE2003.CSV');
    EjesPCASAM4 = csvread('c:\temp\EJESPCAENE2003.CSV');

    VecMedDIFSAM4 = csvread('c:\temp\VECTMEANDIFENE2003.CSV');
    EjesDIFPCASAM4 = csvread('c:\temp\EJESPCADIFENE2003.CSV');


    % Calculamos la simulacion de la temperatura de la banda para esas consignas
    TPI2 = TPI2_REAL;
    MSEVAL = 0;
    MSERROR=0;
    for h=2:Longdat
        % Obtenemos Todas las Variables de Entrada de la Red de Simulacion
        ANCHOMATSAM4 = ANCHO(h-1);
        ESPENTMATSAM4 = ESPENT(h-1);
        VELMATSAM4 = VELO(h-1);
        TMPP1MATSAM4 = PPI1(h-1);
        THF1MATSAM4 = TRAD1(h-1);
        THF3MATSAM4 = TRAD3(h-1);
        THF5MATSAM4 = TRAD5(h-1);
        TMPP2CNGMATSAM4 = TCNG2(h-1);
        TMPP2MATSAM4 = TPI2(h-1);

        % Obtenemos sus diferencias
        DIFANCHOMATSAM4 = ANCHO(h)-ANCHO(h-1);
        DIFESPENTMATSAM4 = ESPENT(h)-ESPENT(h-1);
        DIFVELMATSAM4 = VELO(h)-VELO(h-1);
        DIFTMPP1MATSAM4 = PPI1(h)-PPI1(h-1);
        DIFTHF1MATSAM4 = TRAD1(h)-TRAD1(h-1);
        DIFTHF3MATSAM4 = TRAD3(h)-TRAD3(h-1);
        DIFTHF5MATSAM4 = TRAD5(h)-TRAD5(h-1);
        DIFTMPP2CNGMATSAM4 = TCNG2(h)-TCNG2(h-1);

        %MATSAM4SIN[,2] <- MATSAM4SIN[,2]/2000 ANCHOMATSAM4

        INMATSAM4 = [ANCHOMATSAM4/2000 ESPENTMATSAM4/2500 VELMATSAM4/200
TMPP1MATSAM4/400 ...
            THF1MATSAM4/950 THF3MATSAM4/950 THF5MATSAM4/950 TMPP2MATSAM4/950];

        INDIFMATSAM4 = [DIFANCHOMATSAM4/1500 DIFESPENTMATSAM4/1000
DIFVELMATSAM4/100 DIFTMPP1MATSAM4/150 ...
            DIFTHF1MATSAM4/150 DIFTHF3MATSAM4/150 DIFTHF5MATSAM4/150];

        % Proyectamos con los ejes PCA
        PROYSAM4 = (INMATSAM4-VecMedSAM4')*EjesPCASAM4;
        PROYDIFSAM4 = (INDIFMATSAM4-VecMedDIFSAM4')*EjesDIFPCASAM4;

        % Simulamos el comportamiento dinamico de la banda
        % Creamos las entradas normalizadas
        %VectMinDin =1.0e+003*[-1.1288 -0.4175 -0.1861 -0.5351 -0.2146 -0.1381
-0.1351 -0.0640 0.7140];
        %VectrangDin =1.0e+003*[1.5432 0.7569 0.2860 0.8798 0.4384 0.2860 0.2078
0.1604 0.1750];

        load c:\temp\ModeloA\dinamicoFINAL\vectoresreddin.mat
```

```
        VectMinDin = VectMin;
        VectrangDin =Vectrang;

        DatosInDinamic = ([PROYSAM4 PROYDIFSAM4]-VectMinDin(1:9)) ./
VectrangDin(1:9);
            % Simulamos con los datos

        P=DatosInDinamic';

        if abs(DIFVELMATSAM4)>2 | abs(DIFTMPP1MATSAM4)>2 ...
         | abs(DIFTHF1MATSAM4)>2 | abs(DIFTHF3MATSAM4)>2 | abs(DIFTHF5MATSAM4)>2
              [YDINA] = sim(netdina,P);
               TPI2(h)= YDINA*VectrangDin(10)+VectMinDin(10);%TMPP2
        else
              TPI2(h)=TPI2(h-1);
        end
        if MSEVAL<abs(TPI2_REAL(h)-TPI2(h))
            MSEVAL=+abs(TPI2_REAL(h)-TPI2(h));
        end
        end
    end

    MERROR=mean(abs(TPI2_REAL-TPI2))

    % Dibujamos los resultados de la simulacion
    figure(4)
    clf
    hold on
    %plot(1:Longdat,PPI1,'g',1:Longdat,TRAD1,'b:');
    plot(1:Longdat,TRAD1,'b:');
    plot(1:Longdat,TRAD3,'b:',1:Longdat,TRAD5,'b:');
    plot(1:Longdat,TCNG2,'r',1:Longdat,VELO*5,'m');
    %plot(1:Longdat,TPI2,'k',1:Longdat,TPI2_REAL);
    plot(1:Longdat,TPI2,'k.',1:Longdat,TPI2_REAL,'k-');
    ylabel('Temperatura y Velocidad')
    title('Temp. Piro2 Simulada=Puntos Negros, Real=Linea Negra')
    xlabel('Tiempo');
    hold off
    MSEVAL

    return

end
```

Figura 349. Programa que realiza la simulación de TMPP2 de la banda ante variaciones de temperatura y velocidad.

En las figuras siguientes, **se puede ver el excelente comportamiento de la curva simulada (línea de puntos negros), frente al comportamiento real de la banda (línea continua).**

En el epígrafe de cada figura, se indica el error medio (en grados) y el máximo error entre la curva simulada y real.

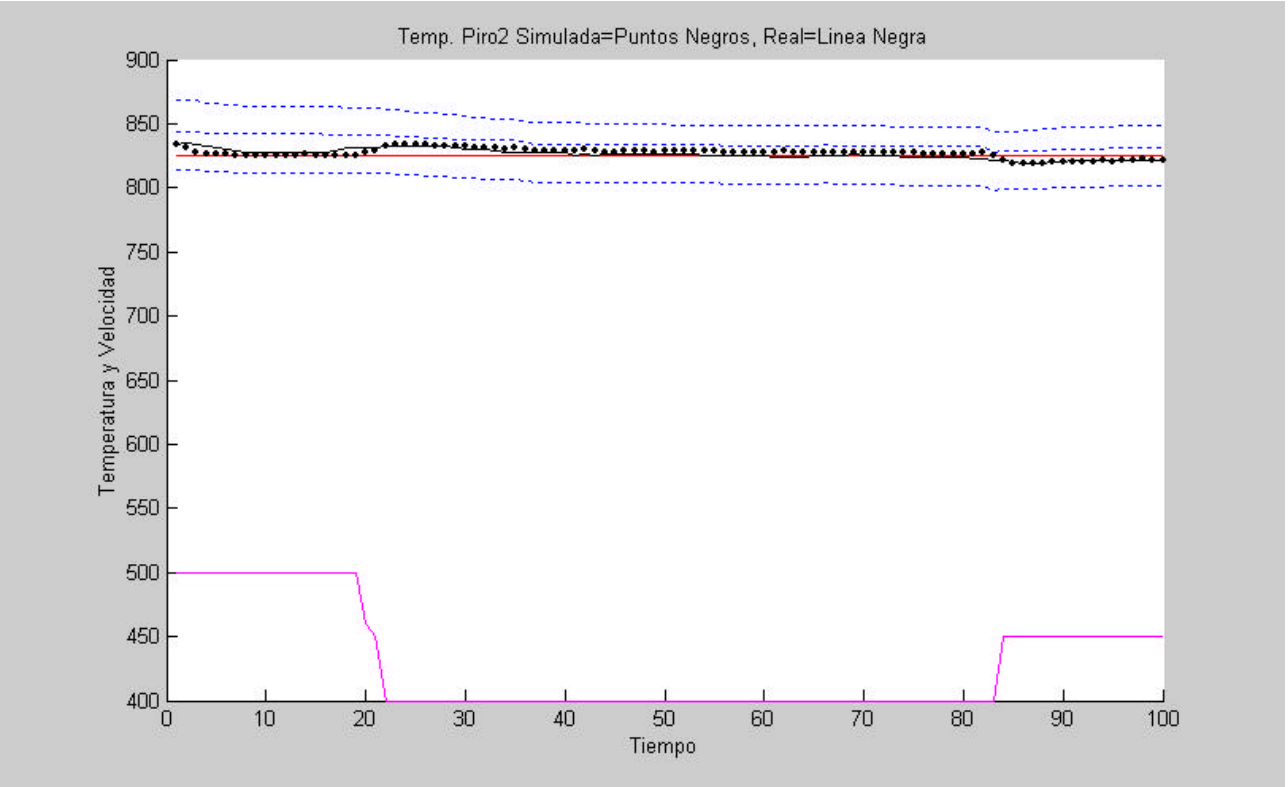


Figura 350. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=2ºC, Máximo=6ºC).

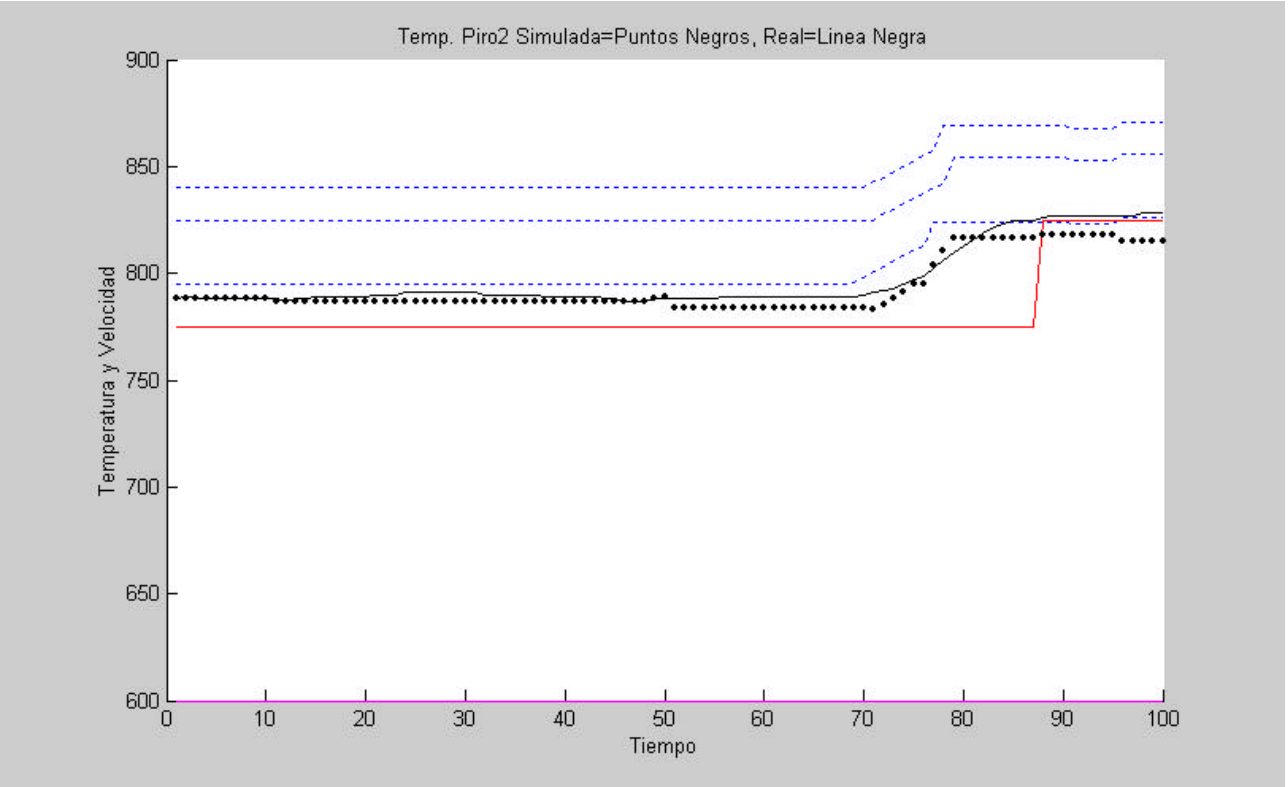


Figura 351. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=4ºC, Máximo=13ºC).

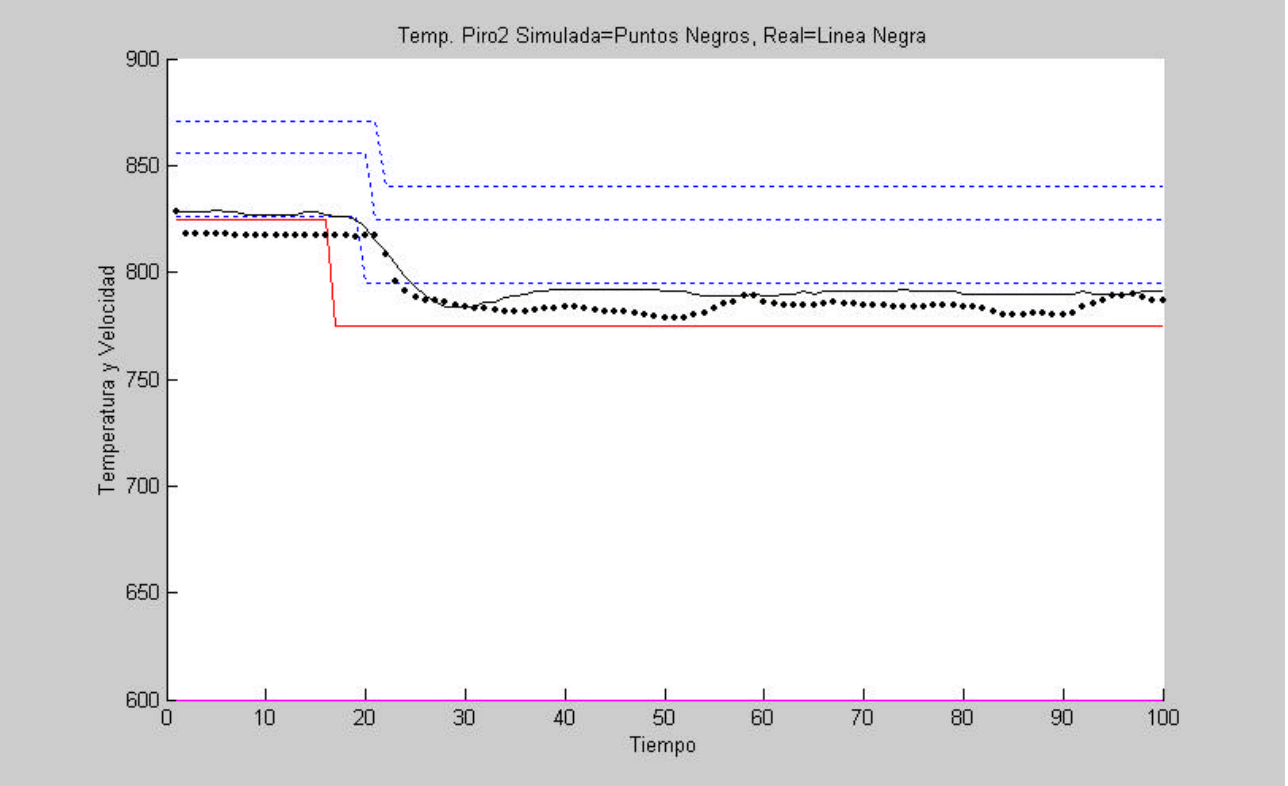


Figura 352. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=7ºC, Máximo=13ºC).

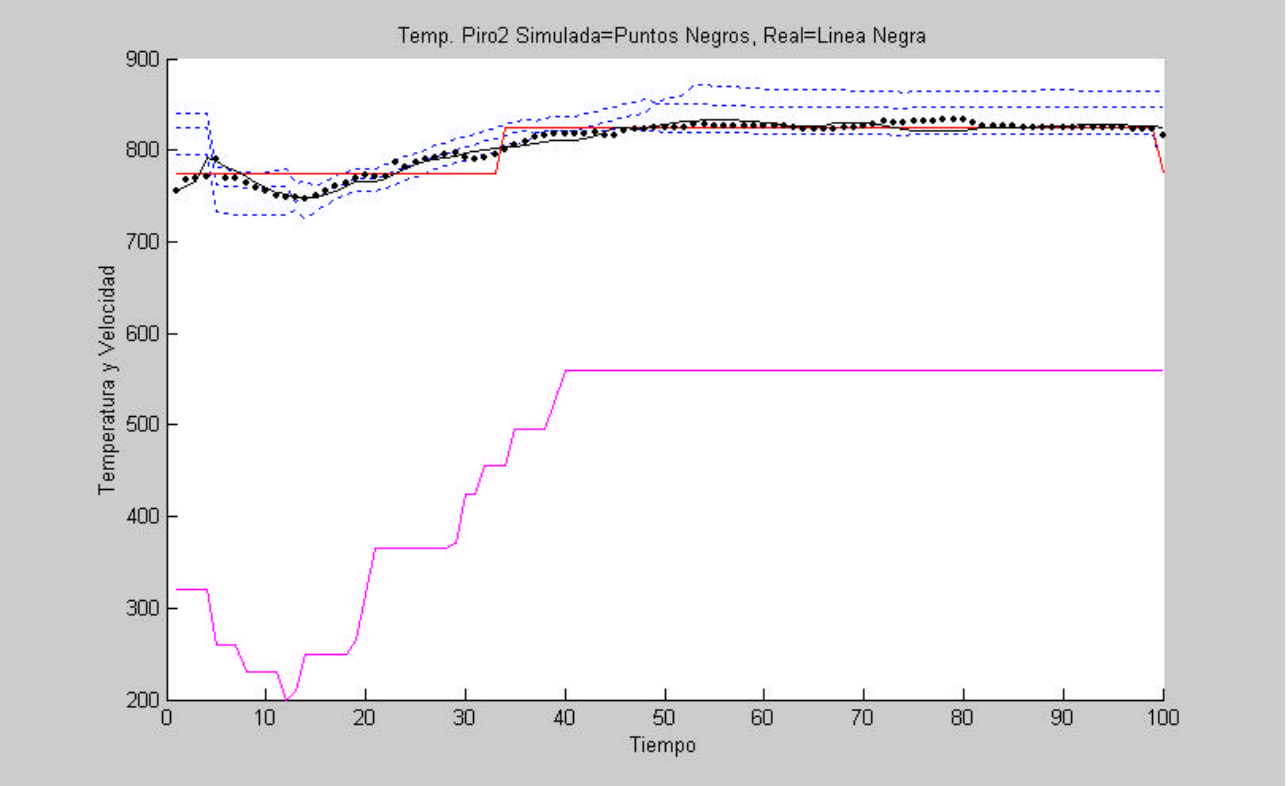


Figura 353. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=5ºC, Máximo=21ºC).

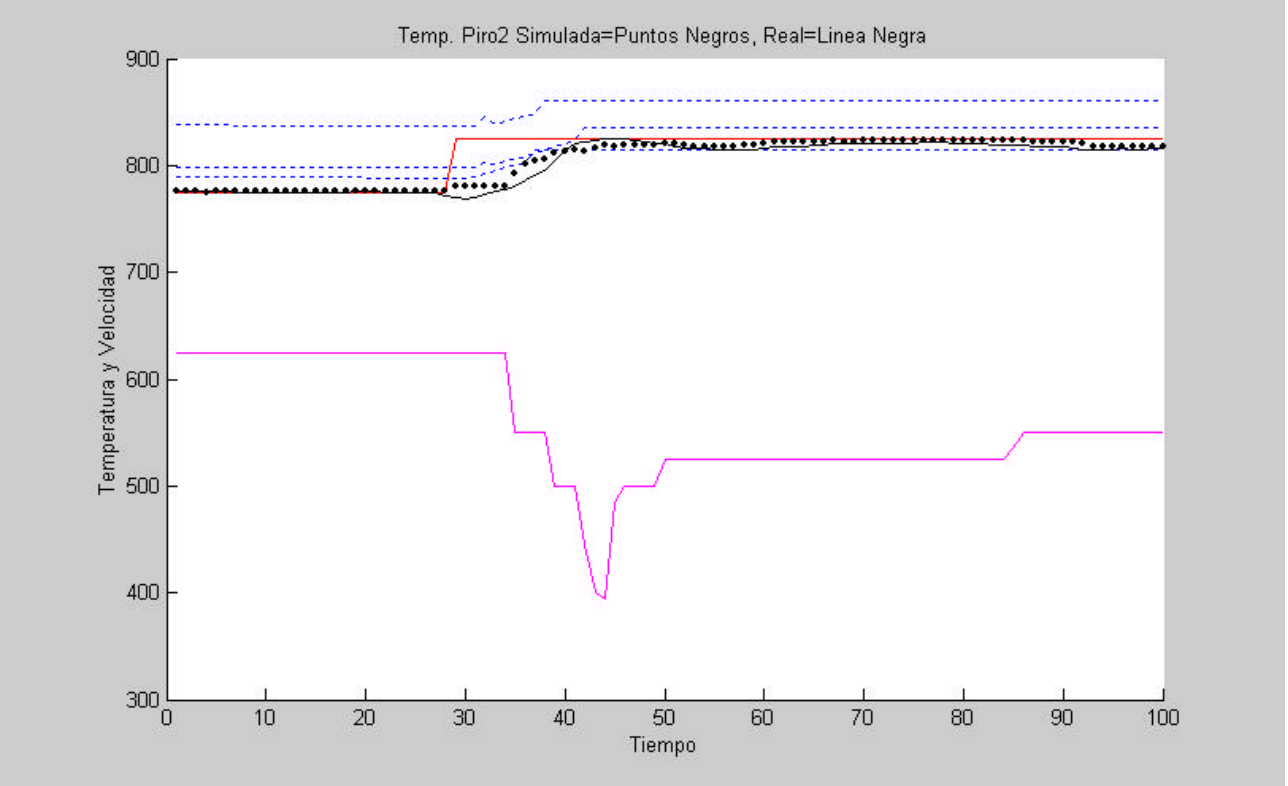


Figura 354. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=3,5ºC, Máximo=5,6ºC).

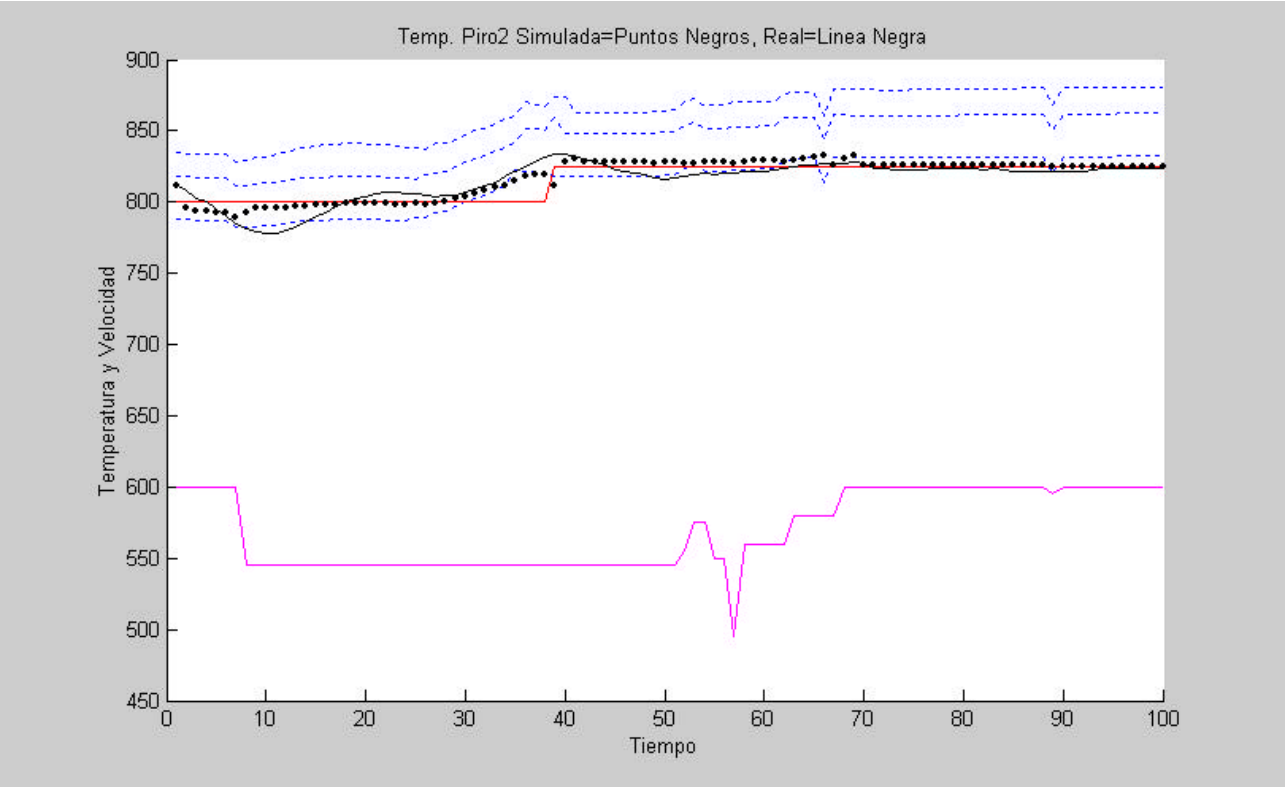


Figura 355. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=5,5ºC, Máximo=22,6ºC).

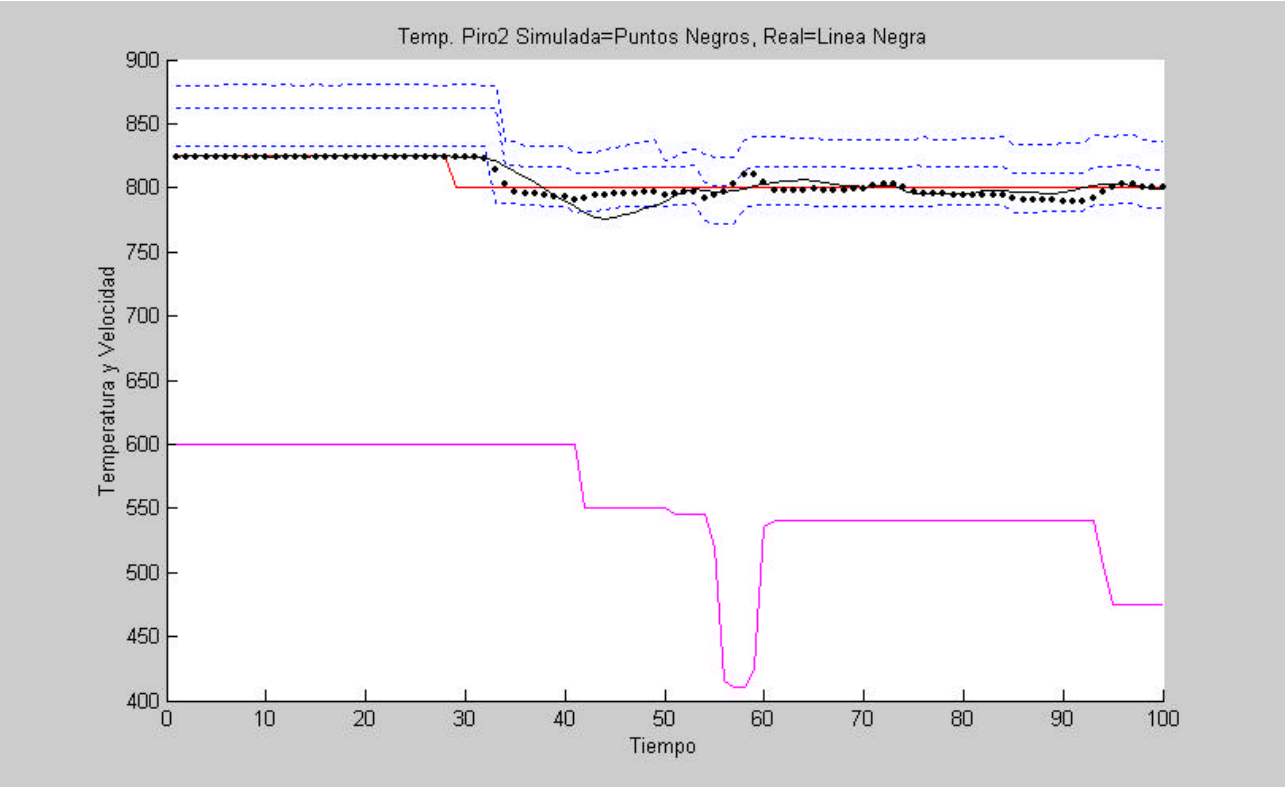


Figura 356. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=4,2ºC, Máximo=18,6ºC).

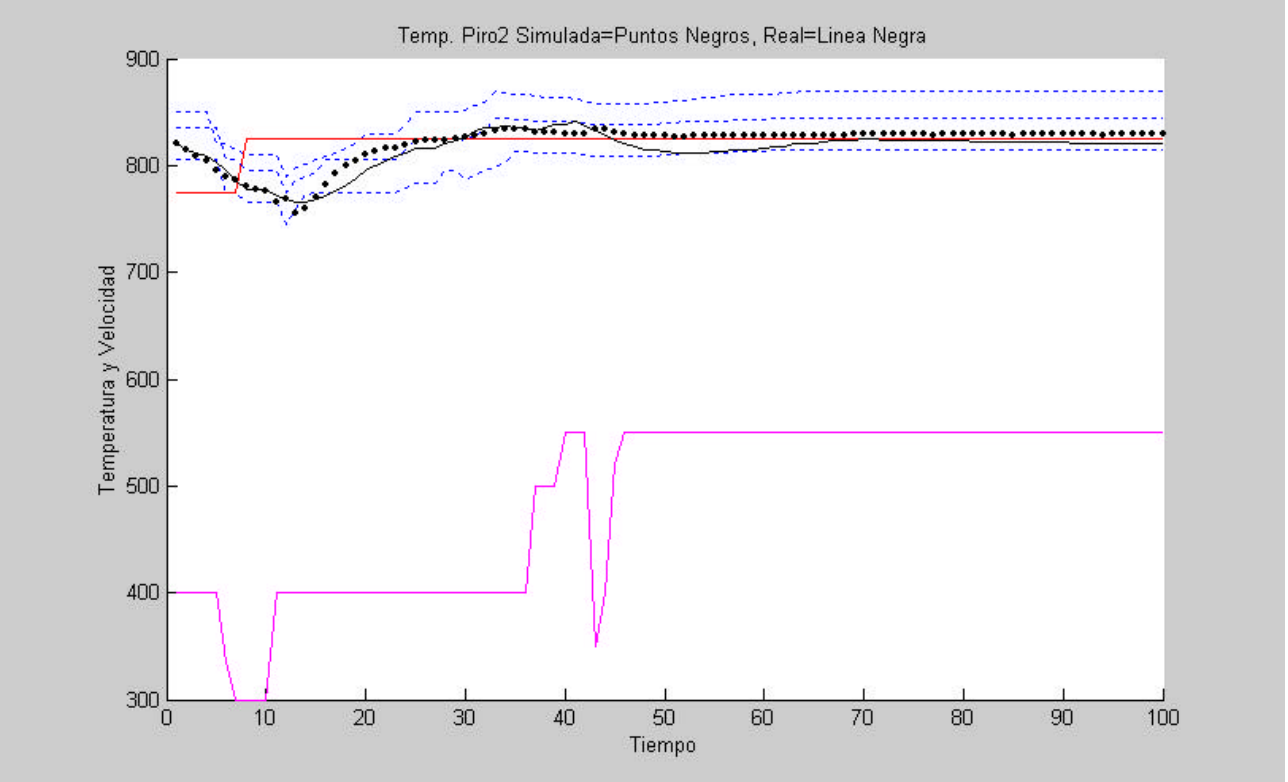


Figura 357. Comportamiento simulado (puntos) y real de la banda (ERROR Medio=7,5ºC, Máximo=17,5ºC).

CONCLUSIONES DE LA SIMULACIÓN

En las figuras anteriores, se han mostrado las simulaciones de la banda para diversos estados de transición.

El modelo matemático obtenido ha dado unos resultados excelentes. Estos han mejorado lo previsto, aunque en algunas variaciones bruscas de la banda, el modelo llega a simularla pero en menor grado. Aún así, éste puede ser mejorado fundamentalmente con:

- El uso de otros modelos de redes neuronales más adecuados para modelización de sistemas dinámicos [SUM01][HUL00][HAN02][LIU01].
- El entrenamiento con una base de datos mucho mayor que permita desarrollar un modelo más robusto y generalizado.

Este modelo, junto con los modelos obtenidos que predicen las variables de consigna, pueden ser unas herramientas de ayuda para:

- Prever el ajuste de las temperaturas de horno y velocidades de consigna para transiciones entre bobinas con diferentes anchuras, espesores o velocidades.
- Analizar los estados transitorios que pueden ser peligrosos.
- Aprender del comportamiento de la banda realizando diferentes simulaciones y pruebas.

En el punto siguiente, se muestra una aplicación de optimización de curvas de consigna.

7.3.4 MEJORA OFF-LINE DE LAS TRANSICIONES ENTRE BOBINAS DE DIFERENTE ANCHURA Y ESPESOR MEDIANTE EL USO DE ALGORITMOS GENÉTICOS

Una de las posibilidades más interesantes que nos provee el uso de los modelos anteriormente obtenidos, es **la planificación de las curvas de temperatura de horno y velocidad para reducir el error entre transiciones.**

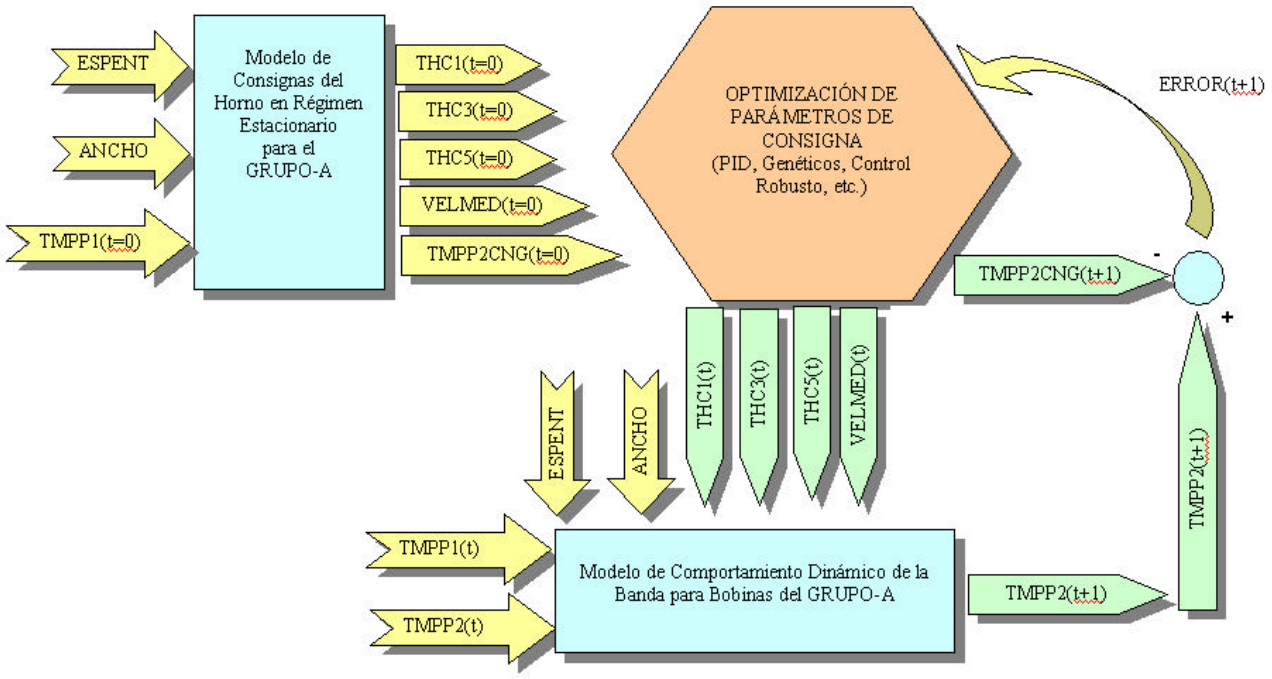


Figura 358. Sistema planteado de Control y Validación Off-Line.

La simulación OFF-LINE de la temperatura de la banda, ante diferentes curvas de temperatura de horno y velocidad de consigna, puede ayudar considerablemente a reducir el error de temperatura de la banda en los cambios de bobinas.

Técnicas como: algoritmos genéticos, control robusto, optimización, etc.; **pueden ser muy útiles en la búsqueda de las curvas de consigna más adecuadas.**

A continuación, se muestra unos ejemplos del uso de algoritmos genéticos en la búsqueda de unas curvas de transición lo más adecuadas posibles en un caso típico de bobinas con diferentes espesores y anchuras.

Para ello, seleccionamos dos bobinas consecutivas con diferentes anchuras.

CODBOB	ANCHURA	ESPESOR
...	...	...
23293048	1200	670
23293048	1200	670
23293048	1200	670
23293048	1200	670
23293049	1150	670
23293049	1150	670
23293049	1150	670
23293049	1150	670
...	...	...

Figura 359. Ejemplo de una transición con bobinas con diferentes anchuras.

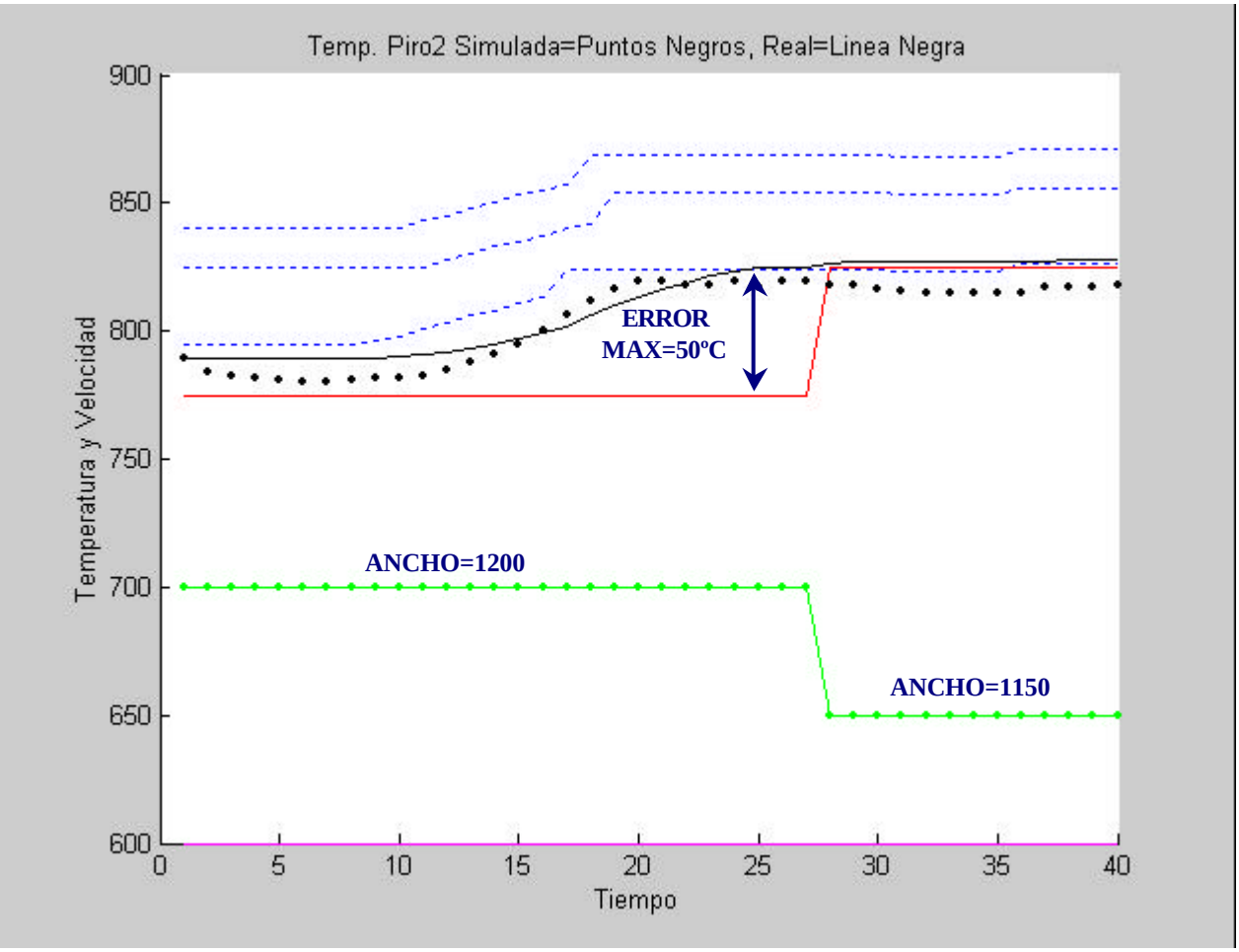


Figura 360. Ejemplo de curvas reales de temperatura y velocidad para bobinas de diferentes anchuras.

En la Figura 360, podemos ver el comportamiento real y simulado de la banda de dos bobinas con diferentes anchuras. Claramente se puede apreciar un error elevado entre la temperatura esperada de la banda (temperatura de consigna de pirómetro 2) y la real (puntos negros (temperatura de la banda simulada) o línea negra (temperatura de banda real)).

Mediante el uso de los algoritmos genéticos, el sistema que se presenta a continuación y según la función de coste que se elija, puede reducir:

- El error medio.
- El error máximo.
- Otros parámetros.

Inicialmente, el sistema puede utiliza los modelos de consignas para generar las curvas iniciales de consigna de temperaturas de zona de horno y velocidad según el espesor, la anchura de la banda y una temperatura inicial de entrada de la banda, o utilizar otras consignas preestablecidas.

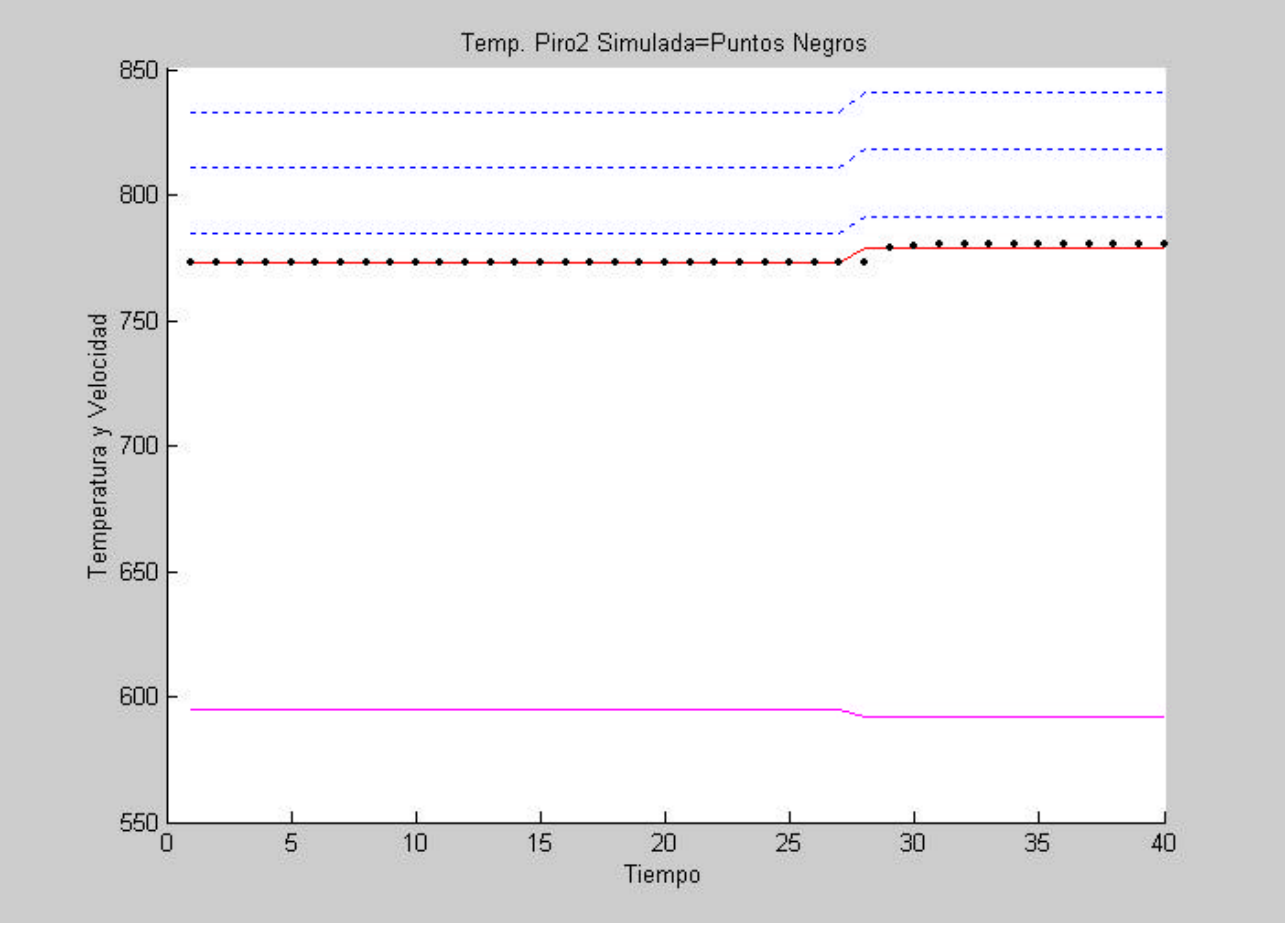


Figura 361. Ejemplo de curvas simuladas de temperatura y velocidad de consigna para las bobinas del caso estudiado.

En la Figura 361, podemos observar las curvas de consigna obtenidas con las redes neuronales previamente desarrolladas. Fácilmente se puede apreciar que las consignas que nos dan los modelos de consignas hacen que la temperatura de la banda no varíe tan bruscamente, lo que parece lógico, ya que la temperatura de consigna de la banda no tendría por qué incrementarse tanto.

Aún así, y para demostrar que el sistema puede ajustar cualquier tipo de variables de consigna, vamos a ver dos casos:

- Ajuste de curvas de consigna no obtenidas con los modelos de consignas.
- Ajuste de curvas de consigna obtenidas con los modelos de consignas.

7.3.4.1 AJUSTE DE CURVAS DE CONSIGNA NO OBTENIDAS CON LOS MODELOS DE CONSIGNAS

En el ejemplo actual, se trata de reducir el error medio entre la temperatura de consigna y la real, buscando rectas de ajuste en las curvas de consigna, ajustando mediante algoritmos genéticos las variables POSICIÓN y ANCHO de cada una de las rectas de transición entre consignas.

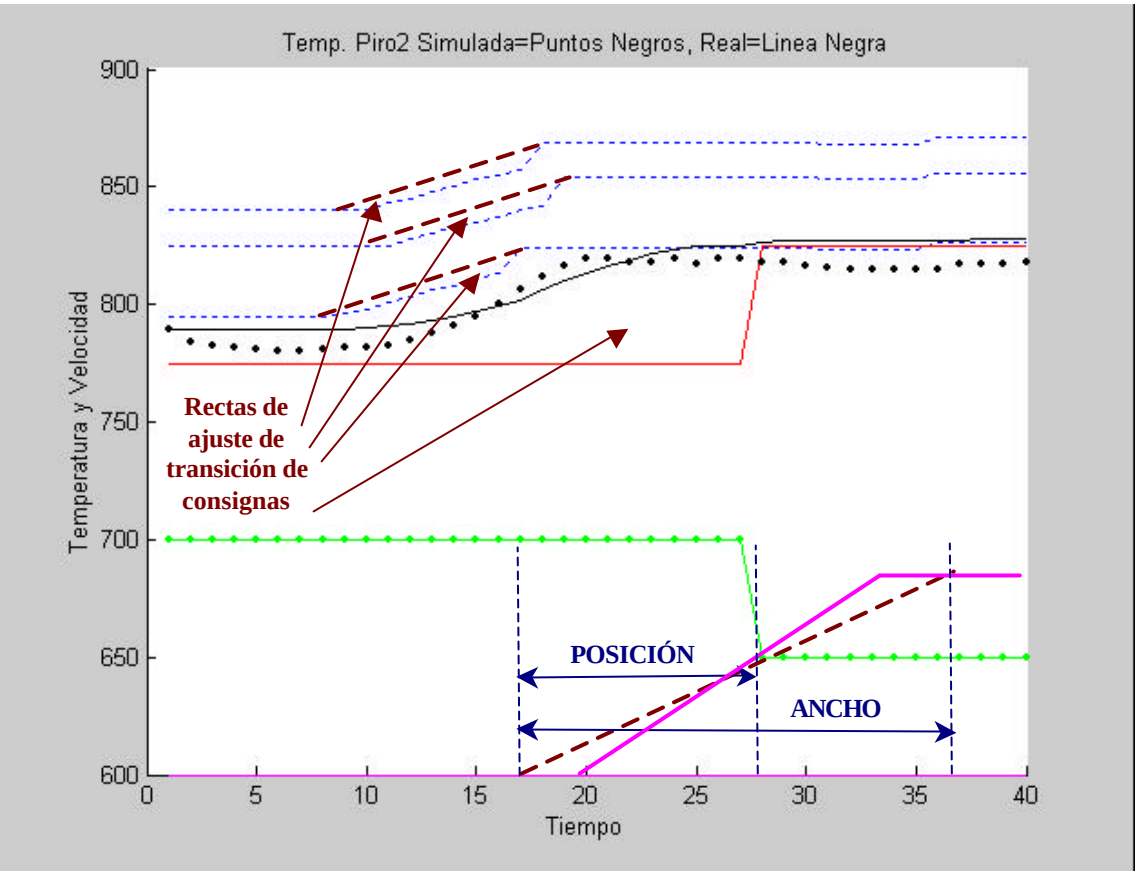


Figura 362. Ejemplo de ajustes de curvas de temperatura de consigna de zonas del horno y velocidad de consigna.

Los algoritmos genéticos, **tratarán de obtener nuevas rectas de ajuste en las curvas de consigna, para reducir el error final medio esperado en la banda. Es decir, se buscará minimizar el valor de la media de la diferencia entre la temperatura de la banda de consigna y la real.**

AJUSTE DE LAS CURVAS MEDIANTE ALGORITMOS GENÉTICOS

El programa que se muestra a continuación, mediante algoritmos genéticos, busca las curvas que generen el menor error medio con la siguiente metodología:

- Genera 100 individuos con valores aleatorios para las siguientes variables:
  - Posición y longitud de la recta de ajuste de la temperatura de consigna de la zona 1 del horno.
  - Posición y longitud de la recta de ajuste de la temperatura de consigna de la zona 3 del horno.
  - Posición y longitud de la recta de ajuste de la temperatura de consigna de la zona 5 del horno.
  - Posición y longitud de la recta de ajuste de la velocidad de consigna de la banda.
- Simula con las curvas de cada individuo, el comportamiento de la banda y obtiene el error máximo instantáneo entre la temperatura de la banda obtenida y la real.
- Obtiene los 20 mejores individuos (con menor error).
- Crea 70 nuevos individuos con el cruce de los 20 mejores.
- Crea otros 10 con mutaciones.
- Crea una nueva generación con los individuos de los pasos 4, 5 y 6.
- Vuelve a repetir los pasos 2 a 5, almacenando los resultados de cada generación.

```
function geneticos()
% Creamos unos 100 curvas aleatorios
% Curvas(1,1:2)=TRAD1 Pos y Long
% Curvas(2,1:2)=TRAD3 Pos y Long
% Curvas(3,1:2)=TRAD5 Pos y Long
% Curvas(4,1:2)=VELO Pos y Long

% Generamos 100 Posiciones aleatorias y Longitudes
% Posiciones entre -8 y +8
% Longitudes entre 1 y 10
MATSAM4SIN=csvread('c:\\temp\\DINMATENE2003TODO.CSV');
DIFMATSAM4SIN=csvread('c:\\temp\\DINDIFMATENE2003TODO.CSV');
NumPob=100;
Curvas=zeros(4,2,NumPob);
Error=zeros(NumPob,1);

for h=1:NumPob
    Curvas(1:4,1:2,h)=[round(rand(4,1)*12)-8 round(rand(4,1)*7)+1];
end

for gene=1:100
    for h=1:NumPob
        Curv =Curvas(:,:,h);

        Error(h)=simula_geneticos_MARZO(MATSAM4SIN,DIFMATSAM4SIN,560,50,Curv);
        Error(h)
    end

     % Ordenamos las posiciones segun los errores
     PosI = [(1:NumPob)' Error];
     PosIOrd = sortrows(PosI,2);

     Arch2='c:\temp\ModeloA\genetico2\GENERACION_';
     Archivo=strcat(Arch2,num2str(gene),'.mat');
     save (Archivo,'Curvas','PosIOrd');

     % Obtenemos los 20 mejores
     CurvasMejor = Curvas(:,:,PosIOrd(:,1));
     CurvasNuev=zeros(4,2,NumPob);
     CurvasNuev(:,:,1:20)=CurvasMejor(:,:,1:20);
     % Obtenemos 70 por mezcla de 20 mejores
     PosAleat10 = round((19*rand(70,4))+1);
     CurvasNuev(1,1:2,21:90)=CurvasMejor(1,1:2,PosAleat10(:,1));
     CurvasNuev(2,1:2,21:90)=CurvasMejor(2,1:2,PosAleat10(:,2));
     CurvasNuev(3,1:2,21:90)=CurvasMejor(3,1:2,PosAleat10(:,3));
     CurvasNuev(4,1:2,21:90)=CurvasMejor(4,1:2,PosAleat10(:,4));
     Curvas=CurvasNuev;
     % Cogemos un 100 y hacemos unas mutaciones
    for h=0:9
        Curvas(1:4,1:2,90+h)=[round(rand(4,1)*12)-8 round(rand(4,1)*9)+1];
    end

end
return;
```

Figura 363. Programa que optimiza las curvas.

```
function
MSERROR=simula_todo_real3(MATSAM4SIN,DIFMATSAM4SIN,Posini,Longdat,Curvas)

%Programa que simula

%Cargamos los datos de MAT
% COL 1=CODBOBMATSAM4
% COL 2=ANCHOMATSAM4
% COL 3=ESPENTMATSAM4
% COL 4=VELMATSAM4
% COL 5=TMPP1MATSAM4
% COL 6=THF1MATSAM4
% COL 7=THF3MATSAM4
% COL 8=THF5MATSAM4
% COL 9=TMPP2MATSAM4
% COL 10=TMPP2MATSAM4SALIDA
% COL 11=TMPP2CNGMATSAM4

%MAT = csvread('c:\\temp\\NEURONALPERM.CSV');
%Guardamos la matriz en un archivo csv
%MATSAM4SINENE=csvread('c:\\temp\\DINMATENE2003TODO.CSV');
%DIFMATSAM4SINENE=csvread('c:\\temp\\DINDIFMATENE2003TODO.CSV');


    %Dibujamos las curvas reales
    PPI1=MATSAM4SIN(Posini:(Posini+Longdat-1),5)';   %Temp Piro1
    TRAD=MATSAM4SIN(Posini:(Posini+Longdat-1),6:8)'; %THF1, THF3, THF5
    TCNG2=MATSAM4SIN(Posini:(Posini+Longdat-1),11)'; %TCNG2
    TPI2=MATSAM4SIN(Posini:(Posini+Longdat-1),9)'; %Temp Piro2
    VELO=5*MATSAM4SIN(Posini:(Posini+Longdat-1),4)'; %Velocidad

    figure(3)
    clf
    hold on
    plot(1:Longdat,PPI1,'g',1:Longdat,TRAD(1,:),'b:');
    plot(1:Longdat,TRAD(1,:),'b:');
    plot(1:Longdat,TRAD(2,:),'b:',1:Longdat,TRAD(3,:),'b:');
    plot(1:Longdat,TCNG2,'r',1:Longdat,VELO,'m');
    plot(1:Longdat,TPI2,'k');
    ylabel('Temperatura y Velocidad')
    title('Curvas Reales')
    hold off



    %Cargamos la red neuronal de comportamiento dinamico
    load ('-MAT','C:\temp\ModeloA\dinamicoFINAL\MATMEJORTEST1.mat');
    netdina = net;

    % Calculamos la simulacion de las curvas con las redes neuronales

    % Obtenemos con las redes neuronales la THFC1, THFC3 y THFC5
    % Y la velocidad y temp de consigna de pirometro2
    % Normalizamos la Matriz
```

```
    MinimoVectMAT= [10000000, 700, 0, 100, 700, 700, 700, 700, 10, 0];
    Vectrang=[30000000, 1300, 2500, 300, 300, 300, 300, 300, 200, 200];


    %Creamos las variables ANCH, TMPP1 y ESPENT normalizadas
    TMPP1ESPERADA = ones(Longdat,1)*MATSAM4SIN(Posini,5);
    DatosIn =[MATSAM4SIN(Posini:(Posini+Longdat-1),2:3) TMPP1ESPERADA];

    MinimosMAT = ones(size(DatosIn),1) * MinimoVectMAT(2:4);
    MATRange = ones(size(DatosIn),1) * Vectrang(2:4);
    DatosInNorm = (DatosIn-MinimosMAT)./MATRange;


% Cargamos la red THCxs
    load ('-MAT','C:\temp\ModeloA\consignasTEMP\MATMEJORTEST11.MData');
    netrad = net;

    % Simulamos con los datos
    P=DatosInNorm';
    [Y] = sim(netrad,P);

    % Cargamos la red de velocidad y tmpp2 de consigna
    load ('-MAT','C:\temp\ModeloA\consignasVEL\MATMEJORTEST13.MData');
    netvel = net;

    % Simulamos con los datos
    [Y2] = sim(netvel,P);


    % Desnormalizamos los datos de entrada y salida
    % Desnormalizamos los datos de entrada y salida
    ANCHO=P(1,:)*Vectrang(2)+MinimoVectMAT(2); %ANCHO
    ESPENT=P(2,:)*Vectrang(3)+MinimoVectMAT(3); %ESPENT
    PPI1=P(3,:)*Vectrang(4)+MinimoVectMAT(4); %TMPP1


    TRAD1=Y(1,:)*Vectrang(5)+MinimoVectMAT(5); %THF1
    TRAD3=Y(2,:)*Vectrang(6)+MinimoVectMAT(6); %THF3
    TRAD5=Y(3,:)*Vectrang(7)+MinimoVectMAT(7); %THF5
    TCNG2=Y(4,:)*Vectrang(8)+MinimoVectMAT(8); %TMPP2CNG

    VELO=(Y2(1,:)*Vectrang(9)+MinimoVectMAT(9)); %Velocidad

    % ---------------------------------------------------------

    %Obtenemos las curvas reales
    ANCHO=MATSAM4SIN(Posini:(Posini+Longdat-1),2)';   %ANCHO
    ESPENT=MATSAM4SIN(Posini:(Posini+Longdat-1),3)';   %ESPENT
    VELO=MATSAM4SIN(Posini:(Posini+Longdat-1),4)'; %Velocidad
    PPI1=MATSAM4SIN(Posini:(Posini+Longdat-1),5)';   %Temp Piro1
    TRAD1=MATSAM4SIN(Posini:(Posini+Longdat-1),6)'; %THF1, THF3, THF5
    TRAD3=MATSAM4SIN(Posini:(Posini+Longdat-1),7)'; %THF1, THF3, THF5
    TRAD5=MATSAM4SIN(Posini:(Posini+Longdat-1),8)'; %THF1, THF3, THF5
    TPI2_REAL=MATSAM4SIN(Posini:(Posini+Longdat-1),9)'; %Temp Piro2
    TCNG2=MATSAM4SIN(Posini:(Posini+Longdat-1),11)'; %TCNG2
```

```
    % Creamos los vectores medias y ejes del PCA para comprimir los datos de
entrada de la red neuronal din´amica

    VecMedSAM4 = csvread('c:\\temp\\VECTMEANENE2003.CSV');
    EjesPCASAM4 = csvread('c:\temp\EJESPCAENE2003.CSV');



    VecMedDIFSAM4 = csvread('c:\temp\VECTMEANDIFENE2003.CSV');
    EjesDIFPCASAM4 = csvread('c:\temp\EJESPCADIFENE2003.CSV');

    % Buscamos el punto de cambio
    SECCION=ANCHO(1)*ESPENT(1);
    POSICION_MEDIO=1;
    DIFANCHO=0;
    DIFESPENT=0;
    for h=2:Longdat
        SECCION2=ANCHO(h)*ESPENT(h);
        if SECCION~=SECCION2
            POSICION_MEDIO=h;
            DIFANCHO=ANCHO(h)-ANCHO(h-1);
            DIFESPENT=ESPENT(h)-ESPENT(h-1);
        end
        SECCION=SECCION2;
    end
    if DIFANCHO+DIFESPENT==0
        display('ERROR NO ENCONTRADO PUNTO DE CAMBIO!!!');
        return;
    end

    DIFCUR=ones(4,1);
    %Obtenemos la diferencia en TRAD1, TRAD3, TRAD5 y VELO
    DIFCUR(1)=TRAD1(POSICION_MEDIO)-TRAD1(POSICION_MEDIO-1);
    DIFCUR(2)=TRAD3(POSICION_MEDIO)-TRAD3(POSICION_MEDIO-1);
    DIFCUR(3)=TRAD5(POSICION_MEDIO)-TRAD5(POSICION_MEDIO-1);
    DIFCUR(4)=VELO(POSICION_MEDIO)-VELO(POSICION_MEDIO-1);


    %Curvas(1,1:2)=TRAD1 Pos y Long
    %Curvas(2,1:2)=TRAD3 Pos y Long
    %Curvas(3,1:2)=TRAD5 Pos y Long
    %Curvas(4,1:2)=VELO Pos y Long

    % Modificamos las curvas de consigna

    %TRAD1ANT=TRAD1(POSICION_MEDIO-1);
    %TRAD1DES=TRAD1(POSICION_MEDIO);
    %TRAD3ANT=TRAD3(POSICION_MEDIO-1);
    %TRAD3DES=TRAD3(POSICION_MEDIO);
    %TRAD5ANT=TRAD5(POSICION_MEDIO-1);
    %TRAD5DES=TRAD5(POSICION_MEDIO);
    %VELANT=VELO(POSICION_MEDIO-1);
    %VELDES=VELO(POSICION_MEDIO);

    TRAD1ANT=min(TRAD1);
    TRAD1DES=max(TRAD1);
```

```
    TRAD3ANT=min(TRAD3);
    TRAD3DES=max(TRAD3);
    TRAD5ANT=min(TRAD5);
    TRAD5DES=max(TRAD5);
    VELANT=min(VELO);
    VELDES=max(VELO);
    DIFCUR(1)=TRAD1DES-TRAD1ANT;
    DIFCUR(2)=TRAD1DES-TRAD1ANT;
    DIFCUR(3)=TRAD1DES-TRAD1ANT;
    DIFCUR(4)=VELDES-VELANT;

    for j=1:4
        for h=1:POSICION_MEDIO-Curvas(j,1)
                switch j
                    case 1
                        TRAD1(h)=TRAD1ANT;
                    case 2
                        TRAD3(h)=TRAD3ANT;
                    case 3
                        TRAD5(h)=TRAD5ANT;
                    case 4
                        VELO(h)=VELANT;
                end
        end
        for h=POSICION_MEDIO-Curvas(j,1)+1:Longdat
                switch j
                    case 1
                        TRAD1(h)=TRAD1DES;
                    case 2
                        TRAD3(h)=TRAD3DES;
                    case 3
                        TRAD5(h)=TRAD5DES;
                    case 4
                        VELO(h)=VELDES;
                end
         end
    end


    for j=1:4
        %Calculamos la pendiente
        Pendiente=DIFCUR(j)/Curvas(j,2);
        for h=0:(Curvas(j,2)-1)
            switch j
                case 1
                    TRAD1(POSICION_MEDIO-Curvas(j,1)+h)=TRAD1(POSICION_MEDIO-
Curvas(j,1)-1)+Pendiente*h;
                case 2
                    TRAD3(POSICION_MEDIO-Curvas(j,1)+h)=TRAD3(POSICION_MEDIO-
Curvas(j,1)-1)+Pendiente*h;
                case 3
                    TRAD5(POSICION_MEDIO-Curvas(j,1)+h)=TRAD5(POSICION_MEDIO-
Curvas(j,1)-1)+Pendiente*h;
                case 4
                    VELO(POSICION_MEDIO-Curvas(j,1)+h)=VELO(POSICION_MEDIO-
Curvas(j,1)-1)+Pendiente*h;
              end
          end
     end
```

```
    % ---------------------------------------------------------
    %Dibujamos las curvas reales
    %PPI1=MATSAM4SIN(Posini:(Posini+Longdat-1),5)';   %Temp Piro1
    %TRAD1=MATSAM4SIN(Posini:(Posini+Longdat-1),6)'; %THF1, THF3, THF5
    %TRAD3=MATSAM4SIN(Posini:(Posini+Longdat-1),7)'; %THF1, THF3, THF5
    %TRAD5=MATSAM4SIN(Posini:(Posini+Longdat-1),8)'; %THF1, THF3, THF5

    %TCNG2=MATSAM4SIN(Posini:(Posini+Longdat-1),9)'; %TCNG2
    TPI2_REAL=MATSAM4SIN(Posini:(Posini+Longdat-1),11)'; %Temp Piro2
    %VELO=MATSAM4SIN(Posini:(Posini+Longdat-1),4)'; %Velocidad


        % Calculamos la simulacion de la temperatura de la banda para esas
consignas
    %TPI2 = TPI2_REAL;
    TPI2=TCNG2;
    MSEVAL = 0;
    MSERROR=0;
    for h=2:Longdat
        % Obtenemos Todas las Variables de Entrada de la Red de Simulacion
        ANCHOMATSAM4 = ANCHO(h-1);
        ESPENTMATSAM4 = ESPENT(h-1);
        VELMATSAM4 = VELO(h-1);
        TMPP1MATSAM4 = PPI1(h-1);
        THF1MATSAM4 = TRAD1(h-1);
        THF3MATSAM4 = TRAD3(h-1);
        THF5MATSAM4 = TRAD5(h-1);
        TMPP2CNGMATSAM4 = TCNG2(h-1);
        TMPP2MATSAM4 = TPI2(h-1);

        % Obtenemos sus diferencias
        DIFANCHOMATSAM4 = ANCHO(h)-ANCHO(h-1);
        DIFESPENTMATSAM4 = ESPENT(h)-ESPENT(h-1);
        DIFVELMATSAM4 = VELO(h)-VELO(h-1);
        DIFTMPP1MATSAM4 = PPI1(h)-PPI1(h-1);
        DIFTHF1MATSAM4 = TRAD1(h)-TRAD1(h-1);
        DIFTHF3MATSAM4 = TRAD3(h)-TRAD3(h-1);
        DIFTHF5MATSAM4 = TRAD5(h)-TRAD5(h-1);
        DIFTMPP2CNGMATSAM4 = TCNG2(h)-TCNG2(h-1);

        INMATSAM4 = [ANCHOMATSAM4/2000 ESPENTMATSAM4/2500 VELMATSAM4/200
TMPP1MATSAM4/400 ...
            THF1MATSAM4/950 THF3MATSAM4/950 THF5MATSAM4/950 TMPP2MATSAM4/950];

        INDIFMATSAM4 = [DIFANCHOMATSAM4/1500 DIFESPENTMATSAM4/1000
DIFVELMATSAM4/100 DIFTMPP1MATSAM4/150 ...
            DIFTHF1MATSAM4/150 DIFTHF3MATSAM4/150 DIFTHF5MATSAM4/150];

        % Proyectamos con los ejes PCA
        PROYSAM4 = (INMATSAM4-VecMedSAM4')*EjesPCASAM4;
        PROYDIFSAM4 = (INDIFMATSAM4-VecMedDIFSAM4')*EjesDIFPCASAM4;

        % Simulamos el comportamiento dinamico de la banda
        % Creamos las entradas normalizadas
        %VectMinDin =1.0e+003*[-1.1288 -0.4175 -0.1861 -0.5351 -0.2146 -0.1381 -
0.1351 -0.0640 0.7140];
        %VectrangDin =1.0e+003*[1.5432 0.7569 0.2860 0.8798 0.4384 0.2860 0.2078
0.1604 0.1750];

        load c:\temp\ModeloA\dinamicoFINAL\vectoresreddin.mat
```

```
        VectMinDin = VectMin;
        VectrangDin =Vectrang;

        DatosInDinamic = ([PROYSAM4 PROYDIFSAM4]-
VectMinDin(1:9))./VectrangDin(1:9);
            % Simulamos con los datos

        P=DatosInDinamic';
        if abs(DIFVELMATSAM4)>1 | abs(DIFTMPP1MATSAM4)>1 ...
          | abs(DIFTHF1MATSAM4)>1 | abs(DIFTHF3MATSAM4)>1 |
abs(DIFTHF5MATSAM4)>1

                [YDINA] = sim(netdina,P);
                TPI2(h)= YDINA*VectrangDin(10)+VectMinDin(10);%TMPP2
            else
                TPI2(h)=TPI2(h-1);
            end


        [YDINA] = sim(netdina,P);
        TPI2(h)= YDINA*VectrangDin(10)+VectMinDin(10);%TMPP2

        %if MSEVAL<abs(TPI2_REAL(h)-TPI2(h))
        %    MSEVAL=+abs(TPI2_REAL(h)-TPI2(h));
        %end
        if MSEVAL<abs(TCNG2(h)-TPI2(h))
            MSEVAL=+abs(TCNG2(h)-TPI2(h));
        end


        end
    end

    %MERROR=mean(abs(TPI2_REAL-TPI2))
    MSERROR=mean(abs(TCNG2-TPI2))
    % Dibujamos los resultados de la simulacion
    figure(4)
    clf
    hold on
    %plot(1:Longdat,PPI1,'g',1:Longdat,TRAD1,'b:');
    plot(1:Longdat,TRAD1,'b:');
    plot(1:Longdat,TRAD3,'b:',1:Longdat,TRAD5,'b:');
    plot(1:Longdat,TCNG2,'r',1:Longdat,VELO*5,'m');
    %plot(1:Longdat,TPI2,'k.',1:Longdat,TPI2_REAL,'k-');
    plot(1:Longdat,TPI2,'k.-');
    ylabel('Temperatura y Velocidad')
    title('Temp. Piro2 Simulada=Puntos Negros')
    xlabel('Tiempo');
    hold off

    MSEVAL
    MSERROR=mean(abs(TCNG2-TPI2))

return;
end
```

Figura 364. Programa que simula el comportamiento de la banda ajustando las curvas de consigna.

RESULTADOS OBTENIDOS

Después de 23 minutos de optimización de la función de coste correspondiente al error medio absoluto y de 7 generaciones de individuos, el error de la curva converge a un valor de 7,31ºC de error medio, frente a los 15,1ºC del error medio inicial; y un valor de 19,38ºC de diferencia máxima, frente a los 44,6ºC máximos iniciales.

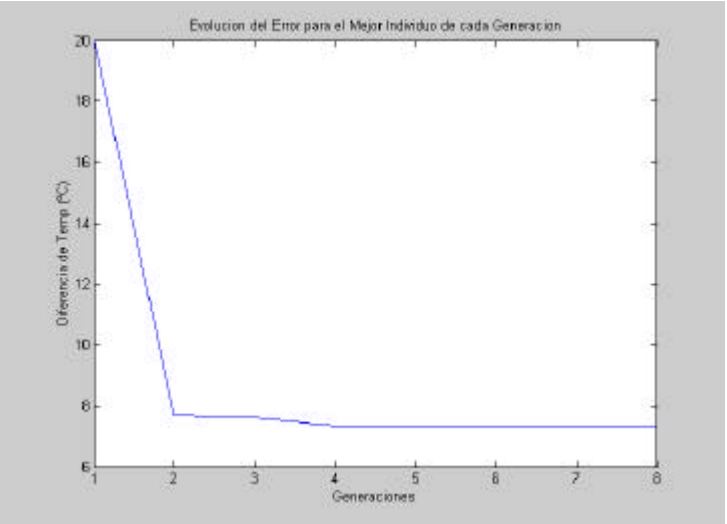


Figura 365.Evolución del error mediante los algoritmos genéticos.

De esta forma, **se ha conseguido reducir claramente la diferencia térmica entre la temperatura de la banda y la buscada.**

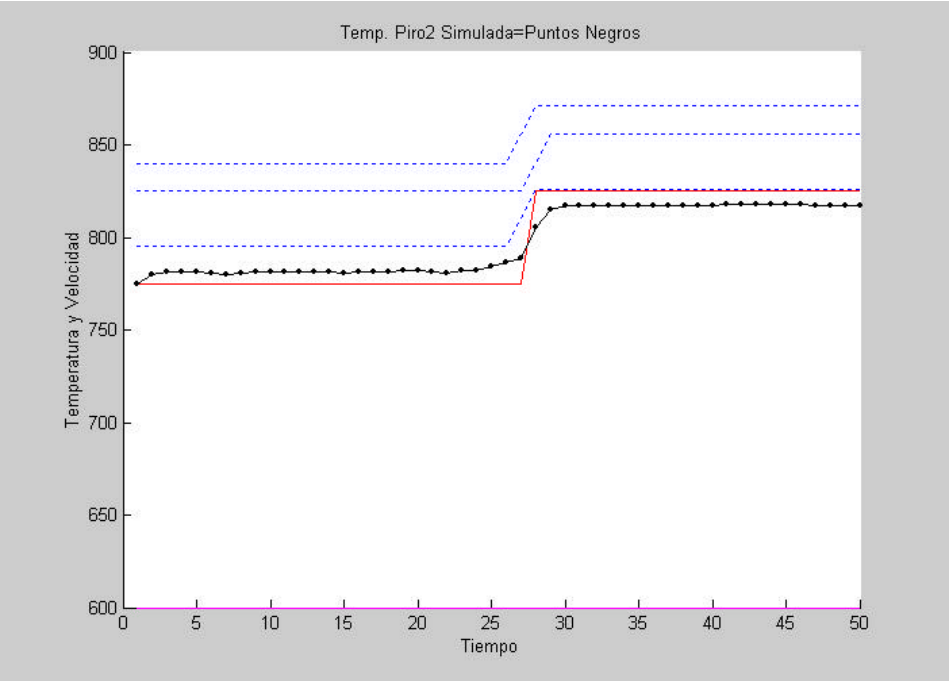


Figura 366. Curvas finales obtenidas que reducen el error medio al 7,31ºC.

7.3.4.2 AJUSTE DE CURVAS DE CONSIGNA OBTENIDAS CON LOS MODELOS DE CONSIGNAS

Para la planificación de las curvas de consigna es evidente que podemos utilizar los modelos de redes neuronales de consignas de temperatura del horno y velocidad.

Para este caso, utilizamos otro caso de cambio de espesor y anchura de bobinas.

CODBOB	ANCHURA	ESPESOR
...	...	...
23293040	1350	675
23293040	1350	675
23293040	1350	675
23293040	1350	675
23293040	1350	675
23293040	1350	675
23293041	1200	670
23293041	1200	670
23293041	1200	670
23293041	1200	670
...	...	...

Figura 367. Ejemplo de una transición con cambio de espesor y anchura.

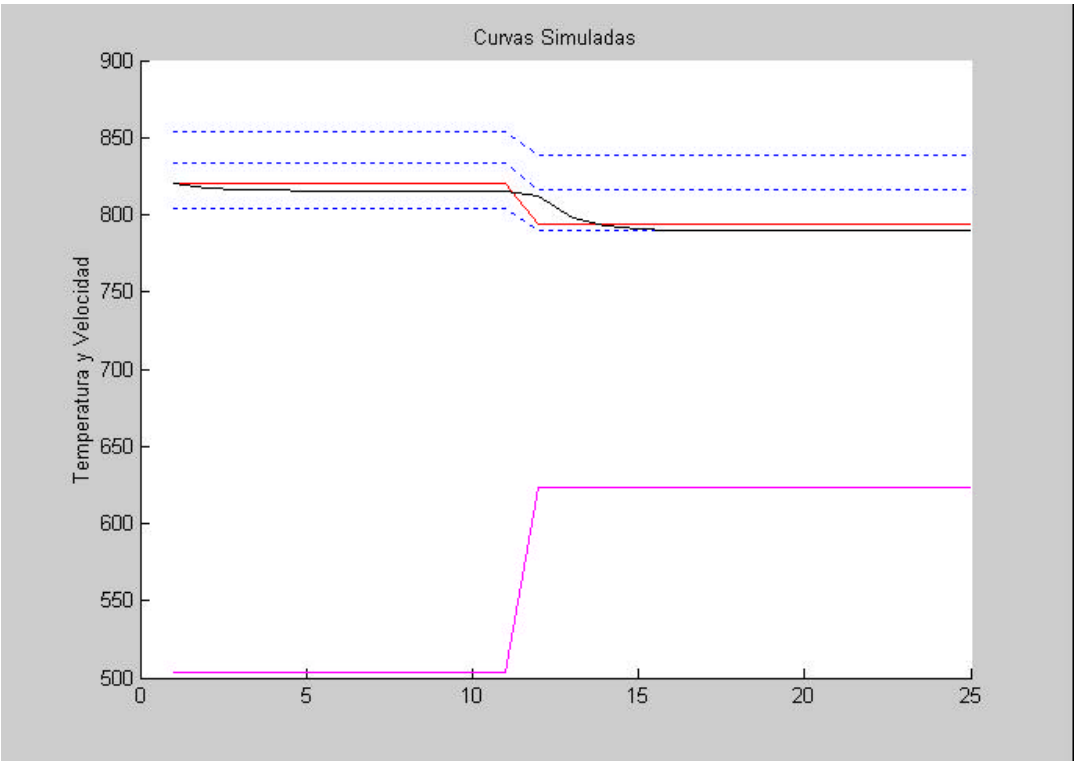


Figura 368. Ejemplo de curvas iniciales de temperatura y velocidad según el modelo de consignas.

El sistema utiliza los modelos de consignas para generar las curvas iniciales de consigna de temperaturas de zonas del horno y velocidad de la banda según el espesor, la anchura de la banda y una temperatura inicial de entrada de la banda.

En el ejemplo actual, se trata de reducir el error máximo de 18,49ºC buscando rectas de ajuste en las curvas de consigna, ajustando mediante algoritmos genéticos las variables POSICIÓN y ANCHO de cada una de las rectas de transición entre consignas.

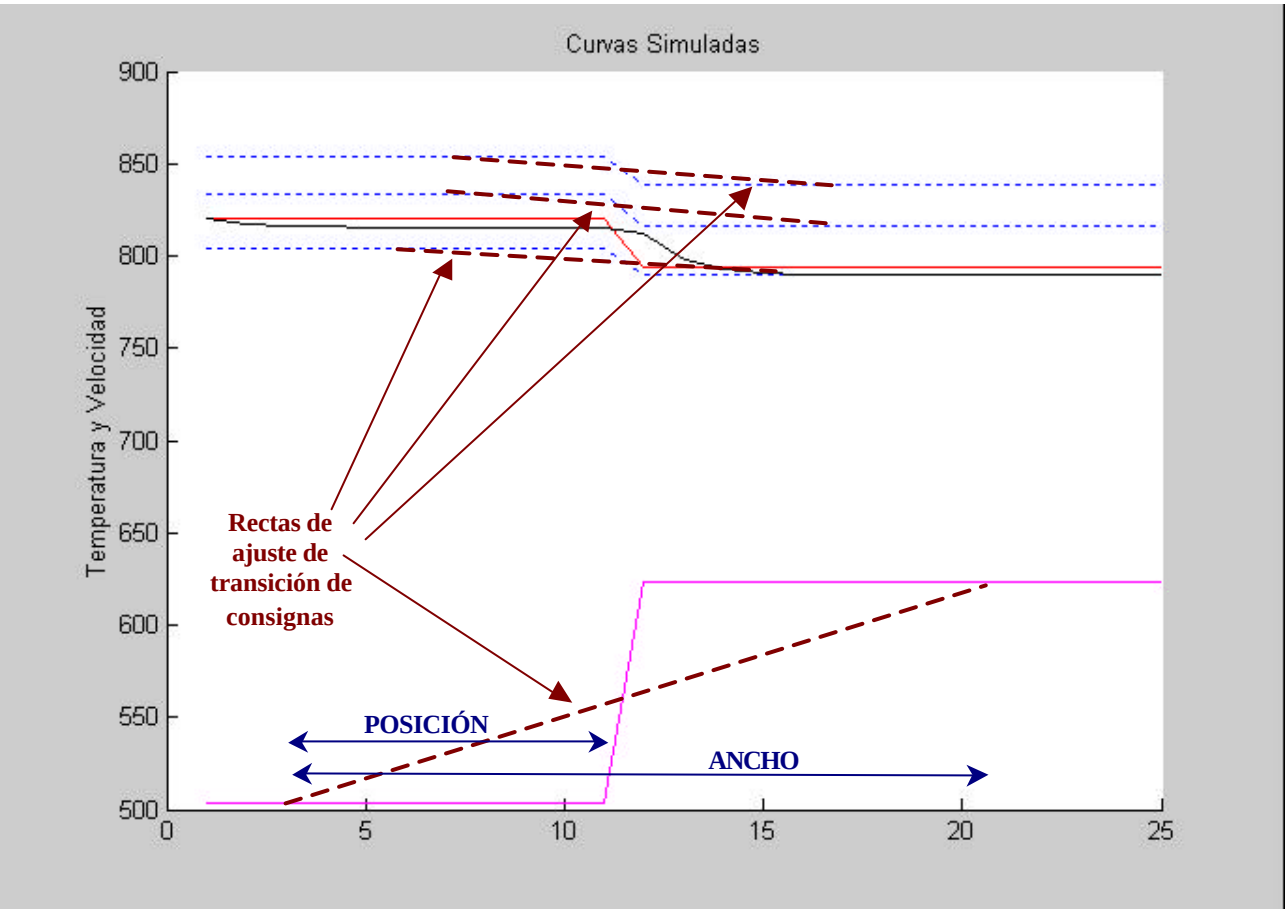


Figura 369. Ejemplo de ajuste de las curvas iniciales de temperatura y velocidad según el modelo de consignas.

RESULTADOS OBTENIDOS

Después de 15 minutos de entrenamiento y 9 generaciones de individuos, el error de la curva converge a un valor de 7,8ºC de diferencia máxima, frente a los 18,5ºC iniciales.

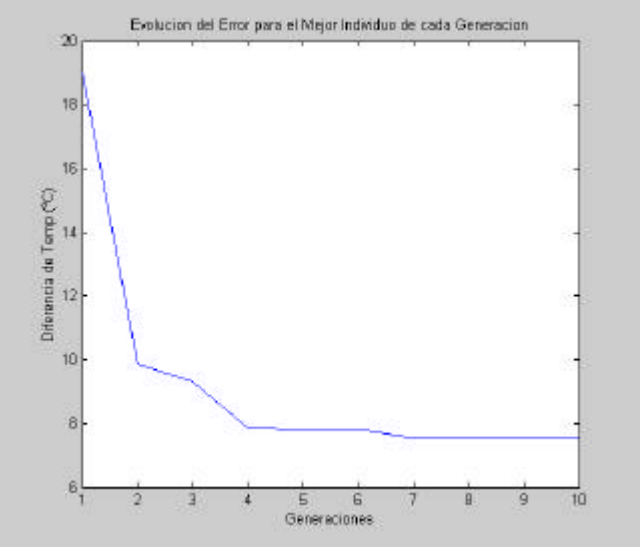


Figura 370.Evolución del error mediante los algoritmos genéticos.

También **se ha conseguido reducir claramente la diferencia de temperatura entre la temperatura de la banda y la buscada.**

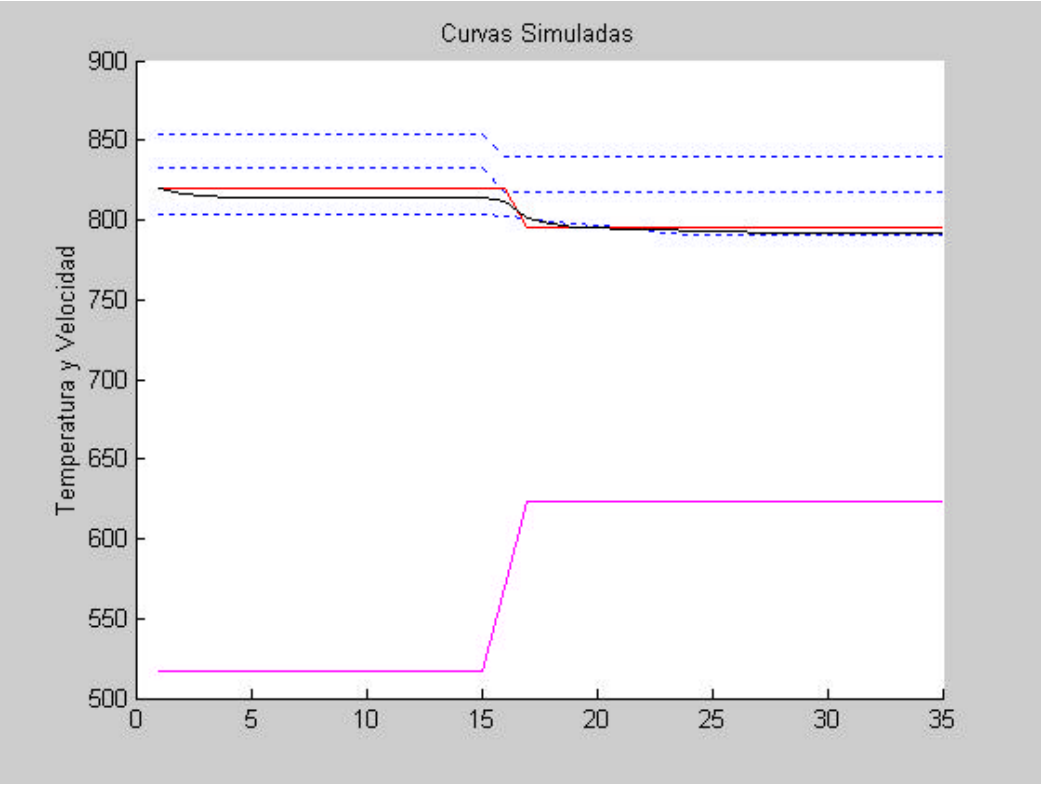


Figura 371. Curvas finales obtenidas que reducen el error al 7,81ºC máximos.

7.4 CONCLUSIONES DE LA FASE DE MODELADO

En este capítulo **se han desarrollado diversas técnicas que pueden ayudar en la predicción y mejora del control del horno de galvanizado.**

Hemos podido ver cómo el proyector de las consignas del horno es una herramienta que puede ser utilizada para visualizar tendencias de este punto de operación, cuando se ajustan las consignas en "modo manual". De esta forma, **el operario puede visualizar, ante pequeños cambios manuales de consignas, "cómo se mueve" el punto de operación en relación con el centroíde de puntos de operación en régimen permanente.**

Posteriormente, se han creado los modelos de consignas para el control del horno y del comportamiento dinámico de la banda, que **pueden ser de gran utilidad para la planificación y ajuste de las temperaturas de consigna del horno y velocidades, tanto en los regímenes permanentes como en las transiciones entre bobinas de diferentes espesores, anchuras o tipos de acero**.

Para estas transiciones, se ha desarrollado una metodología, basada en algoritmos genéticos y redes neuronales, **que permite simular y plantear con anterioridad las mejores consignas de velocidad y temperatura de consigna de zonas del horno, de forma que se reduzca el error final medio o máximo entre la temperatura de la banda y la esperada.**

En el capítulo siguiente, siguiendo con la metodología CRISP-DM, se validarán los resultados de todo el proceso y se plantearán las estrategias de aplicación de todas estas técnicas, en el entorno industrial.

CAPÍTULO 8

EVALUACIÓN DE LOS RESULTADOS OBTENIDOS

8.1 INTRODUCCIÓN

Una vez desarrollados los diferentes modelos y métodos de optimización del proceso, se realizan la siguiente fase de la metodología *CRISP-DM* [CRI00]: **la evaluación de los resultados.**

Los objetivos principales son:

- Evaluar los resultados obtenidos valorando los modelos y métodos desarrollados.
- Determinar las mejoras del proceso y acciones a considerar.

8.2 FASE V: EVALUACIÓN DE LOS RESULTADOS

En esta fase del proceso *CRISP-DM*, se evalúan los modelos obtenidos, se revisa el proceso desarrollado hasta el momento y se busca la forma de mejorar los resultados obtenidos.

Esta fase constan de los siguientes pasos:

- Evaluación de los Resultados.
- Revisión del Proceso.
- Determinación de las Siguientes Acciones a Tomar.

8.2.1 EVALUACIÓN DE LOS RESULTADOS

En este primer paso [CRI00][ABA01], se trata de comprobar que los modelos cumplen los objetivos buscados, así como determinar las flaquezas de los mismos y la forma de reducirlas.

Para ello, se analizan:

- **Los resultados del Data Mining.** Donde se observa:
  - El grado de comprensión obtenido del sistema.
  - El grado de aplicabilidad de los mismos.
  - Si se ha descubierto nueva información sobre el problema.
  - Si se cumplen los objetivos iniciales.
- **El funcionamiento de cada Modelo.** Analizando:
  - El grado de generalización del modelo.
  - La precisión del mismo.
  - Las flaquezas.

8.2.2 REVISIÓN DEL PROCESO

Una vez analizados los resultados del DM y de los modelos, se procede a revisar todo el proceso anterior buscando las alternativas de mejora.

Se pretende, una vez observados los resultados, mejorar toda la metodología para llegar a un afinamiento mayor de los resultados.

8.2.3 DETERMINACIÓN DE LAS ACCIONES SIGUIENTES

Si los resultados son adecuados, se pasará a la fase de explotación de los mismos, sino, se desarrollará una serie de alternativas de mejora de las fases previas de la Metodología *CRIPS-DM,* analizando:

- Las acciones de mejora a realizar.
- Los resultados que se conseguirán.
- El tiempo y recursos necesarios.
- El potencial de explotación final esperado.

De esta forma, según el grado de éxito de los resultados obtenidos, los plazos y recursos de que se disponen y el potencial esperado, se decide la alternativa mejor, justificando su elección.

8.3 APLICACIÓN PRÁCTICA DE LA FASE V DE LA METODOLOGÍA CRISP-DM

Para poder validar los resultados obtenidos, es conveniente utilizar una nueva base de datos que nos permita determinar el grado de eficiencia, ante nueva información, de:

- Las conclusiones obtenidas.
- Las herramientas de diagnóstico desarrolladas.
- Los Modelos entrenados.

Para ello, se trabaja con una nueva base de datos suministrada por la empresa, tratándola y transformándola adecuadamente según los pasos explicados en los capítulos anteriores.

Una vez tratada, se evaluarán las siguientes herramientas:

- **El clasificador de bobinas.** Se evaluará la capacidad que tiene para predecir paradas o roturas de banda.
- **El modelo generador de consignas de temperatura y velocidad.**
- **El modelo que explica el comportamiento dinámico de la banda.**

8.3.1 EVALUACIÓN DEL CLASIFICADOR DE BOBINAS

Para determinar la utilidad del uso del clasificador de bobinas que tiene para detectar roturas de la banda, se analizan las proyecciones de las bobinas para las paradas más significativas ocurridas en un mes de tratamiento.

Para ello, se detectan las paradas calculando la diferencia en horas entre bobinas.

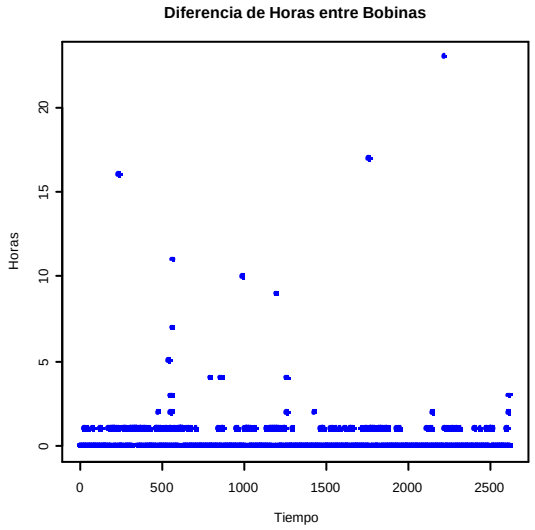


Figura 372. Detección de paradas entre bobinas por su diferencia de horas.

```
# Cargamos las librerías de análisis multivariante
library(mva)
library(multiv)

# Cargamos las matrices con los datos de las 2.628 bobinas
library(RODBC)
canal <- odbcConnect("aceralia2003","","","localhost");

##############################
# Obtenemos datos de bobinas #
##############################


DATBOBINASTODAS <- sqlQuery(canal, "SELECT CODBOBINA, BOBENT, ESPENT, CLASACERO,
DUREZA, CICREC, ANCHO, ESPESOR, LARGO, PESO, CALIDAD, FECFAB, HORFAB FROM dps")

# Detectamos las bobinas repetidas
table(DATBOBINASTODAS$CODBOBINA)

# Obtenemos una lista de las bobinas sin repeticiones
LISTABOBINAS <- unique(DATBOBINASTODAS$CODBOBINA)

# Convertimos fechas y horas a formato POSIX
dates <- DATBOBINASTODAS$FECFAB
times <- DATBOBINASTODAS$HORFAB
x <- paste(dates, times)
FECHAS <- strptime(x, "%d-%m-%Y %H:%M")

# Obtenemos la diferencia en horas entre bobinas
K <- length(dates)
DIFHOR <- round(difftime(FECHAS[2:K], FECHAS[1:(K-1)], units = "hours"))

# Dibujamos la diferencia de tiempo entre bobinas
plot(DIFHOR,col='blue',pch=19,xlab='Tiempo',ylab='Horas',main='Diferencia de
Horas entre Bobinas')
table(DIFHOR)
DIFHOR
   0    1    2    3    4    5    7    9   10   11   16   17   23
2252  354    7    3    3    1    1    1    1    1    1    1    1

# Cargamos los ejes
load(file="MATDAT310103.RData")

MATACERNUEV <- MATACEROS

# Obtenemos la proyección PCA
PCAACERO <- pca(as.matrix(MATACERNUEV[,3:17]),method=2)

#Detectamos las bobinas con errores grandes
BOBPARADAS <- DATBOBINASTODAS$CODBOBINA[DIFHOR>5]
BOBPARADAS
 [1] 23243047 23313038 23323006 23393002 23423036 23513001 23583033
DIFHOR[DIFHOR>5]
[1] 16  7 11 10  9 17 23
```

```
#Que bobina
QUEBOB <- 7

#Obtenemos las bobinas anteriores y posteriores
POSMED <- BOBPARADAS[QUEBOB]
POSCEN <- match(BOBPARADAS[QUEBOB], DATBOBINASTODAS$CODBOBINA)
POSICIONMED <- match(POSMED, MATACERNUEV[,1])

PASODIF <- 3
plot(PCAACERO$rproj[,1], PCAACERO$rproj[,2])
points(PCAACERO$rproj[(POSICIONMED-PASODIF):(POSICIONMED+PASODIF),1],
PCAACERO$rproj[(POSICIONMED-PASODIF):(POSICIONMED+PASODIF),2],col='red',pch=19)

# Datos Parada
DATPROJ <- cbind(MATACERNUEV[(POSICIONMED -PASODIF):( POSICIONMED
+PASODIF),1],PCAACERO$rproj[(POSICIONMED-PASODIF):(POSICIONMED+PASODIF),1],
PCAACERO$rproj[(POSICIONMED-PASODIF):(POSICIONMED+PASODIF),2], DIFHOR[(POSCEN-
PASODIF):( POSCEN+PASODIF)])

# Datos Bobinas
#Obtenemos las bobinas anteriores y posteriores
POSICIONMATBOB <- match(BOBPARADAS[QUEBOB], MATBOBINAS[,1])
MATBOBPARADA <- MATBOBINAS[(POSICIONMATBOB -PASODIF):( POSICIONMATBOB
+PASODIF),]

#Obtenemos el tipo de acero
POSICIONDATBOB <- match(BOBPARADAS[QUEBOB], DATBOBINAS[,1])
DATBOBPARADA <- DATBOBINAS[(POSICIONDATBOB -PASODIF):( POSICIONDATBOB
+PASODIF),]

DATOSPARADA <- cbind (DATPROJ, DATBOBPARADA, MATBOBPARADA$MODOBOB, MATBOBPARADA$
ERRORMEDTOTALABS, MATBOBPARADA$TIPOCURVAERROR)

names(DATOSPARADA)[1:4] <- c("CODBOB","XPROJ","YPROJ","DIFHORAS")
names(DATOSPARADA)[18:20] <- c("MODOBOB","ERRORMEDABS","TIPOERROR")

DATOSPARADA
```

Figura 373. Programa que detecta las paradas y muestra la proyección de las bobinas anteriores y posteriores.

Vamos a observar alguna de las paradas más significativas, que corresponden a paradas mayores de 5 horas.

23243047	23313038	23323006	23393002	23423036	23513001	23583033
16	7	11	10	9	17	23

Tabla 71. Bobinas y horas de parada.

8.3.1.1 RESULTADOS PARADA DE LA BOBINA 23313038 (7 HORAS)

En la figura y tabla siguientes, **se puede ver cómo la parada del proceso parecer ser debida a unas bobinas con composición muy diferente a las demás. Parece que la predicción del proyector puede ayudar a evitar estas paradas que pueden ser debidas a roturas de la banda u otra causa.**

Aún así, este estudio debería completarse con un análisis de los informes de producción para determinar el tipo de avería producida.

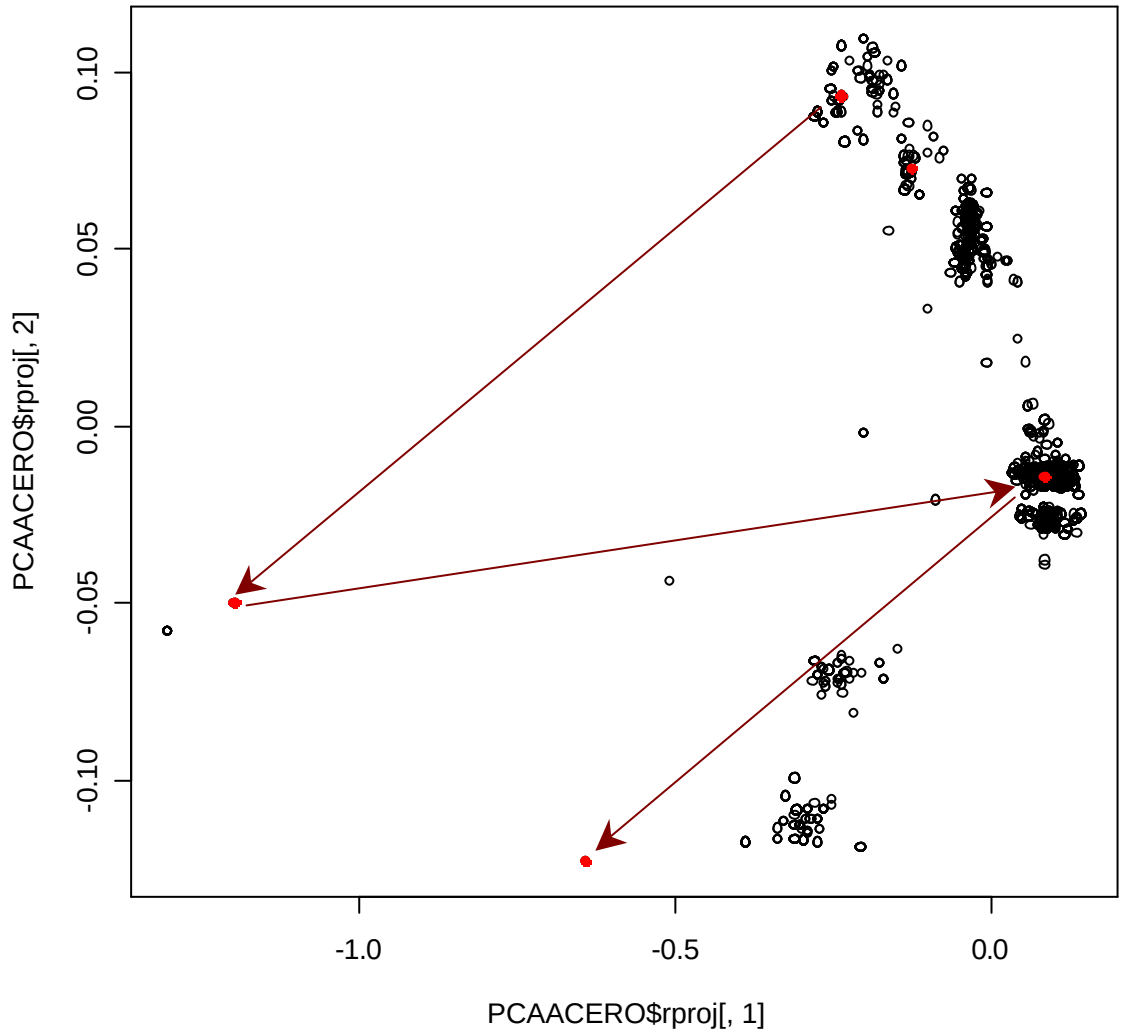


Figura 374. Proyección de las bobinas tratadas.

En este caso, claramente se aprecia cómo en la parada aparecen unas bobinas con composiciones muy diferenciadas.

DATOSPARADA

	CODBOB	XPROJ	YPROJ	DIFHORAS	CODBOBINA
478	23313035	-0.23568451	0.09325653	2	23313035
479	23313036	-1.19662028	-0.05007996	0	23313036
480	23313037	-1.19662028	-0.05007996	0	23313037
481	**23313038**	**0.08812127**	**-0.01439113**	**7**	**23313038**
482	23323001	-0.64039511	-0.12322459	2	23323001
483	23323002	-0.64039511	-0.12322459	3	23323002
484	23323003	-0.12331164	0.07291117	0	23323003

	BOBENT	ESPENT	CLASACERO	DUREZA	CICREC	ANCHO	ESPESOR
478	2328D513	0.961	B011B99	<NA>	A1	1003	0.961
479	2174T006	1.370	D094B33	G4	<NA>	775	1.370
480	2183D031	1.370	D094B33	G4	<NA>	775	1.370
481	**2221T001**	**1.490**	**B100F55**	**<NA>**	**A1**	**838**	**1.490**
482	2294D077	1.506	C115G55	E8	<NA>	775	1.500
483	2300D005	1.506	C115G55	E8	<NA>	775	1.500
484	2242D021	1.951	D012G99	<NA>	A1	889	1.951

	LARGO	PESO	CALIDAD	FECFAB	HORFAB	MODOBOB
478	824	6150	NA	28-11-2002	02:08	0
479	1152	9710	NA	28-11-2002	04:35	0
480	552	4610	NA	28-11-2002	04:44	0
481	**681**	**6580**	**NA**	**28-11-2002**	**04:55**	**0**
482	1260	11870	NA	28-11-2002	12:04	-1
483	1276	12090	NA	28-11-2002	13:37	0
484	891	12050	NA	28-11-2002	16:37	0

	ERRORMEDABS	TIPOERROR
478	53	H
479	39	MOMAXMIN
480	41	MRD
481	**35**	**ACVAC**
482	25	ACVAC
483	31	ACXAC
484	45	ARC

Figura 375. Datos principales de las bobinas tratadas antes y después de la parada.

8.3.1.2 RESULTADOS PARADA DE LA BOBINA 23323006 (11 HORAS)

En este caso, se visualiza claramente como el error parece que es debido a una bobina cuya composición se sale de la familia que se estaba tratando. Igual que en el caso anterior, y en los posteriores, se debería contrastar este tipo de información con los informes de incidencias en la producción.

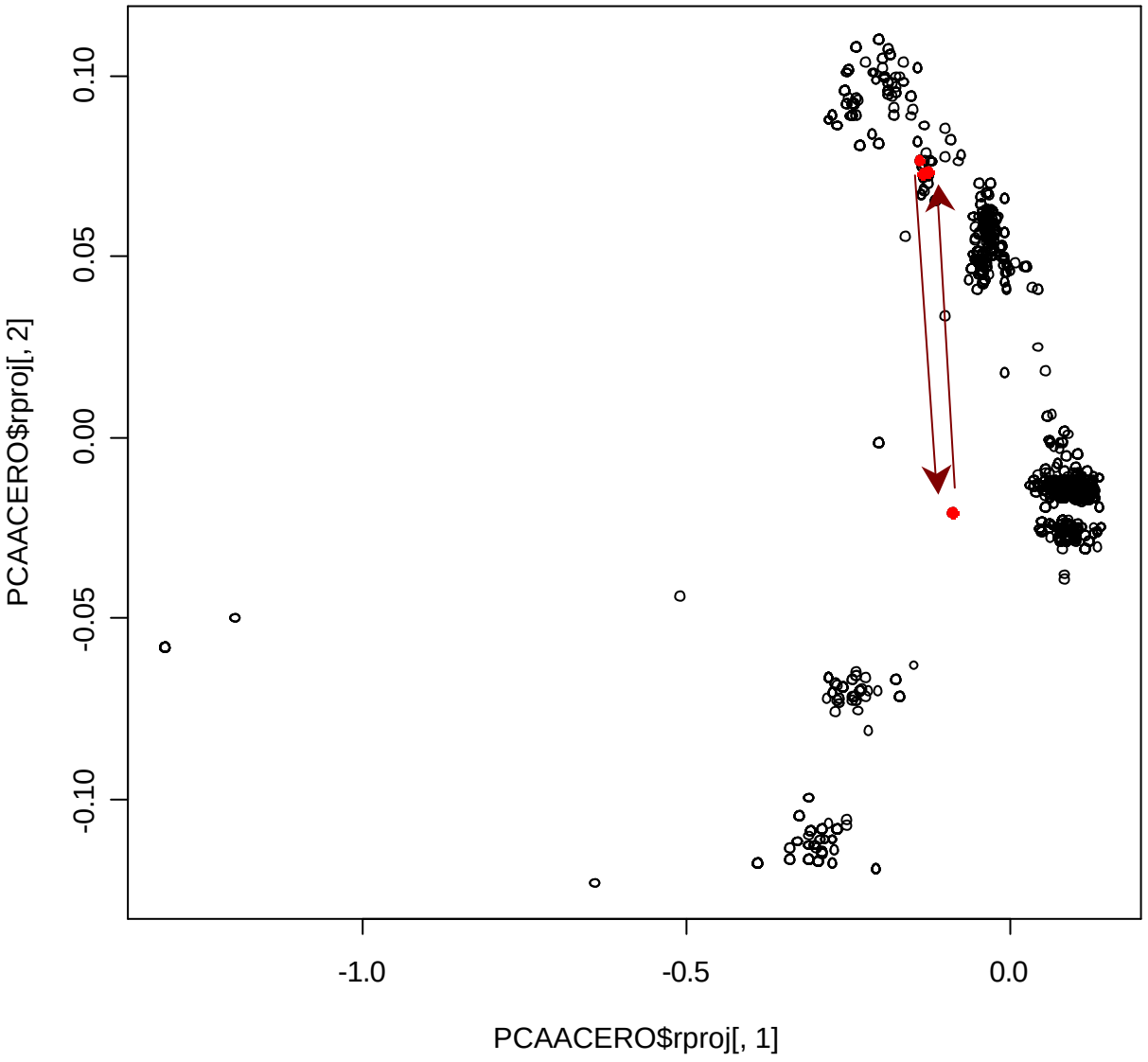


Figura 376. Proyección de las bobinas tratadas.

```
DATOSPARADA
      CODBOB      XPROJ       YPROJ DIFHORAS CODBOBINA
484 23323003 -0.1233116  0.07291117        0  23323003
485 23323004 -0.0861260 -0.02109201        0  23323004
486 23323005 -0.0861260 -0.02109201        0  23323005
487 23323006 -0.1330089  0.07205819       11  23323006
488 23323007 -0.1330089  0.07205819        0  23323007
489 23323008 -0.1363475  0.07614189        0  23323010
490 23323009 -0.1363475  0.07614189        0  23323011
      BOBENT ESPENT CLASACERO DUREZA CICREC ANCHO ESPESOR
484 2242D021  1.951   D012G99     <NA>     A1   889   1.951
485 2256D082  1.921   C116G55     <NA>     A1  1034   1.921
486 2256D082  1.921   C116G55     <NA>     A1  1034   1.921
487 2268D079  1.951   D012G99     <NA>     A1   886   1.951
488 2268D079  1.951   D012G99     <NA>     A1   886   1.951
489 2265D031  1.800   D012F55     E1     <NA>   845   1.770
490 2265D034  1.800   D012F55     E1     <NA>   845   1.770
    LARGO  PESO CALIDAD     FECFAB HORFAB MODOBOB
484   891 12050      NA 28-11-2002  16:37       0
485   503  7720      NA 28-11-2002  16:46       0
486   591  9090      NA 28-11-2002  17:02       0
487   350  4730      NA 28-11-2002  17:08       0
488   399  5460      NA 29-11-2002  04:27       0
489   736  9060      NA 29-11-2002  04:57       0
490  1330 16350      NA 29-11-2002  05:17       0
    ERRORMEDABS TIPOERROR
484          45       ARC
485           8       MRC
486           2         H
487          23       ARD
488         110         E
489          59         E
490          15       ARC
```

Figura 377. Datos principales de las bobinas tratadas antes y después de la parada.

8.3.1.3 RESULTADOS PARADA DE LA BOBINA 23393002 (10 HORAS)

En cambio, en este caso, la parada también parece debida a un cambio de composición del acero.

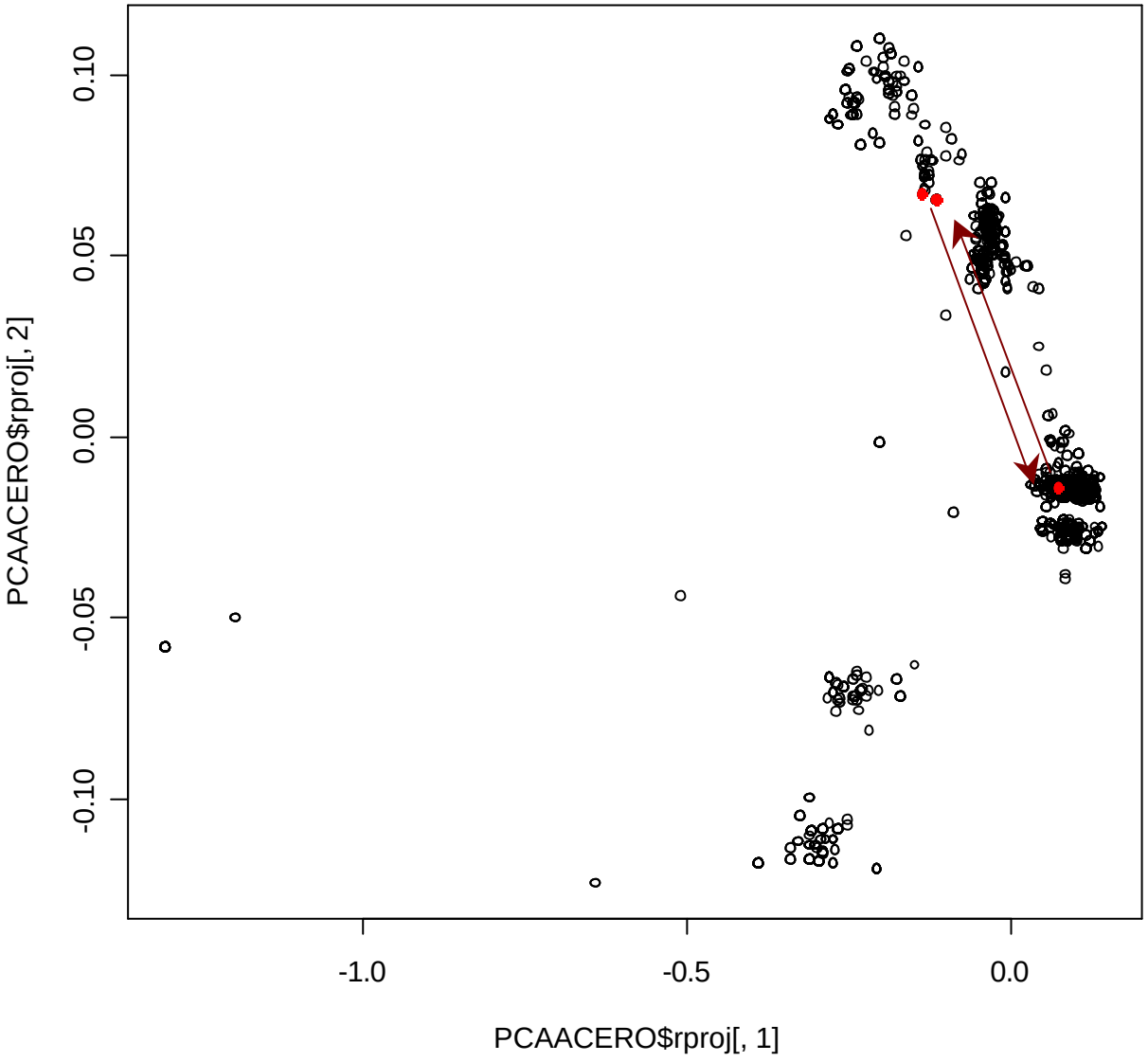


Figura 378. Proyección de las bobinas tratadas.

```
DATOSPARADA
      CODBOB       XPROJ       YPROJ DIFHORAS CODBOBINA
879 23383072 -0.13472881  0.06666723        0  23383072
880 23383073 -0.11303237  0.06532033        0  23383073
881 23393001 -0.13472881  0.06666723        0  23393001
882 23393002 -0.13472881  0.06666723       10  23393002
883 23393003 -0.13472881  0.06666723        0  23393003
884 23393004  0.07736478 -0.01444014        0  23393004
885 23393005  0.07736478 -0.01444014        0  23393005
      BOBENT ESPENT CLASACERO DUREZA CICREC ANCHO ESPESOR
879 145549     1.48   D071F55     E1     NA   850    1.47
880 145551     1.48   D071F55     E1     NA   850    1.47
881 146157     1.48   D071F55     E1     NA   850    1.47
882 146156     1.48   D071F55     E1     NA   850    1.47
883 146158     1.48   D071F55     E1     NA   850    1.47
884 146528     1.49   B100F55     50     NA   820    1.47
885 146528     1.49   B100F55     50     NA   820    1.47
    LARGO  PESO CALIDAD     FECFAB HORFAB MODOBOB
879  1619 15810      NA 05-12-2002  05:40       0
880  1648 16120      NA 05-12-2002  05:58       0
881  1612 16040      NA 05-12-2002  06:17       0
882  1623 16080      NA 05-12-2002  06:36       0
883  1611 15970      NA 05-12-2002  16:17       0
884   673  6510      NA 05-12-2002  16:33       0
885   396  3750      NA 05-12-2002  16:41       0
    ERRORMEDABS TIPOERROR
879           6       BRC
880           4       BRD
881           3         H
882           4       BRC
883           5     MCXAC
884          11     ACXAC
885          13  AOMAXMIN
```

Figura 379. Datos principales de las bobinas tratadas antes y después de la parada.

8.3.1.4 RESULTADOS PARADA DE LA BOBINA 23423036 (9 HORAS)

También en este caso, parece que la parada no es explicada por un cambio brusco de tipo de acero.

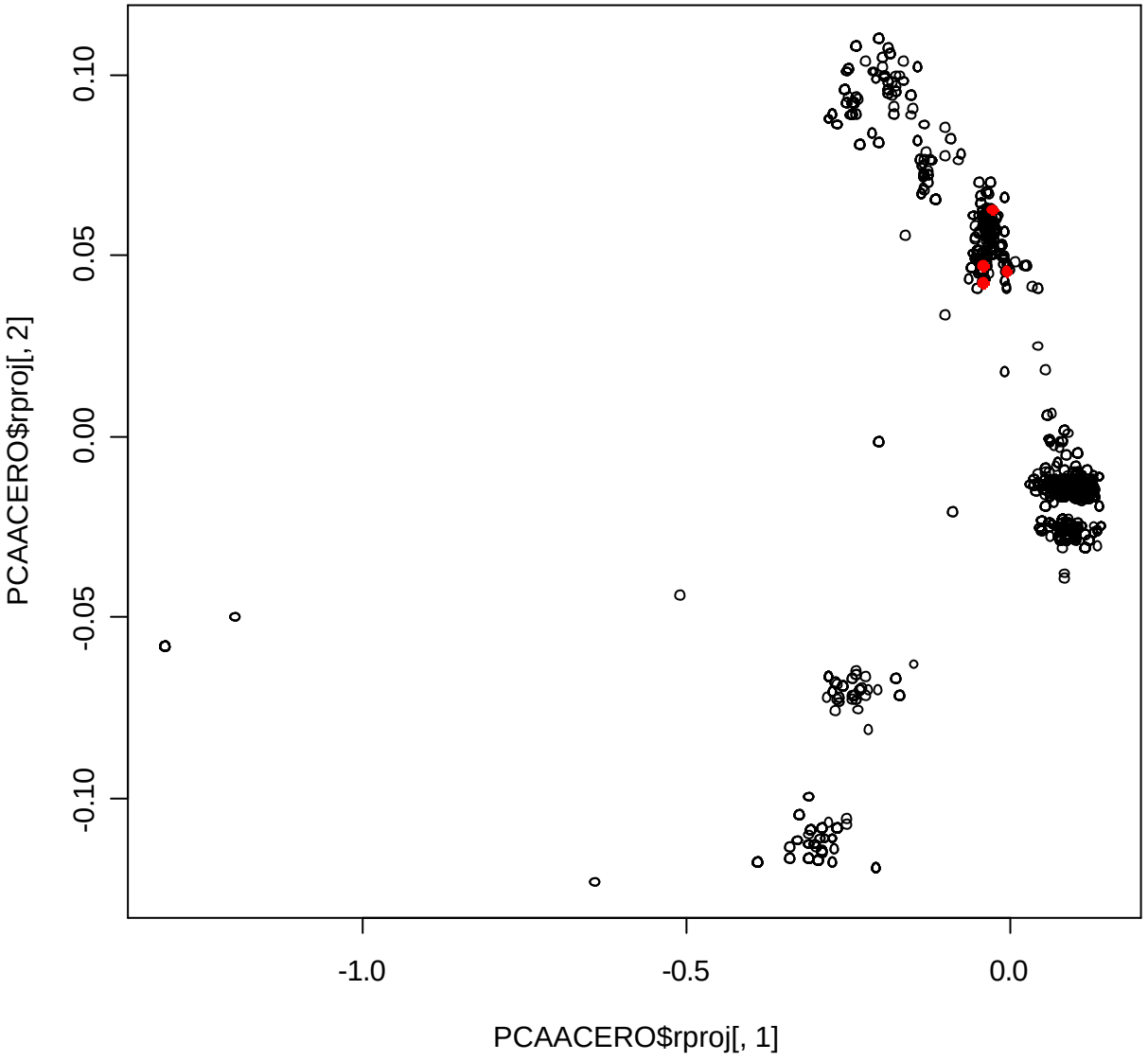


Figura 380. Proyección de las bobinas tratadas.

```
> DATOSPARADA
       CODBOB        XPROJ      YPROJ DIFHORAS CODBOBINA
1073 23423033 -0.002268993 0.04520298        0  23423031
1074 23423034 -0.002268993 0.04520298        1  23423034
1075 23423035 -0.037990634 0.04644348        1  23423035
1076 23423036 -0.039141127 0.04218879        9  23423036
1077 23433001 -0.025487274 0.06208967        0  23433001
1078 23433002 -0.025487274 0.06208967        0  23433003
1079 23433003 -0.025487274 0.06208967        1  23433004
       BOBENT ESPENT CLASACERO DUREZA CICREC ANCHO
1073 145694    0.476   B025F55     15     NA  1250
1074 145702    0.476   B025F55     15     NA  1250
1075 145702    0.476   B025F55     15     NA  1250
1076 145703    0.476   B025F55     15     NA  1250
1077 146492    0.476   B012F53     19     NA  1250
1078 146492    0.476   B012F53     19     NA  1250
1079 146507    0.476   B025F55     15     NA  1250
     ESPESOR LARGO  PESO CALIDAD     FECFAB HORFAB
1073     0.5  4517 21720      NA 08-12-2002  21:06
1074     0.5  3791 18250      NA 08-12-2002  21:48
1075     0.5  4565 21950      NA 08-12-2002  22:22
1076     0.5  4486 21550      NA 08-12-2002  23:17
1077     0.5  1328  6560      NA 09-12-2002  08:26
1078     0.5  2586 12510      NA 09-12-2002  09:06
1079     0.5  4501 21600      NA 09-12-2002  09:53
     MODOBOB ERRORMEDABS TIPOERROR
1073       0           3       BRC
1074       0           5       BRC
1075       0           1         H
1076       0           0         H
1077      -1          23     ACXAD
1078      -1          10       BRC
1079      -1           4       BRD
```

Figura 381. Datos principales de las bobinas tratadas antes y después de la parada.

8.3.1.5 RESULTADOS PARADA DE LA BOBINA 23513001 (17 HORAS)

Este otro caso de parada, también parece ser explicado mediante el proyector desarrollado.

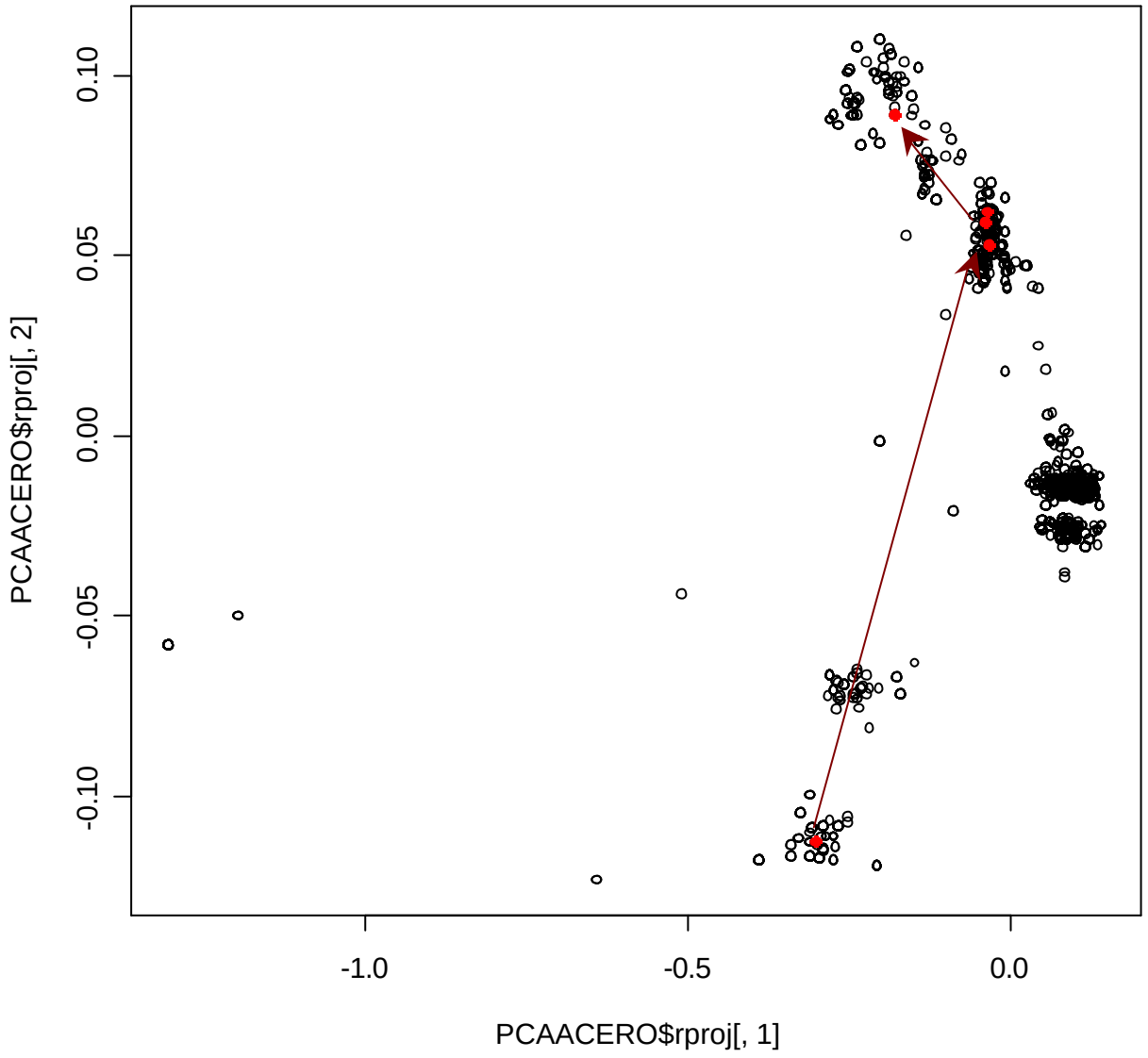


Figura 382. Proyección de las bobinas tratadas.

```
> DATOSPARADA
       CODBOB       XPROJ       YPROJ DIFHORAS CODBOBINA
1615 23503067 -0.30126143 -0.11287444        0  23503067
1616 23503068 -0.30126143 -0.11287444        0  23503068
1617 23503069 -0.03469319  0.06159851        0  23503069
1618 23513001 -0.03107716  0.05292280       17  23513001
1619 23513002 -0.17584447  0.08852034        1  23513002
1620 23513003 -0.17584447  0.08852034        0  23513003
1621 23513004 -0.03596421  0.05854230        1  23513004
       BOBENT ESPENT CLASACERO DUREZA CICREC ANCHO
1615 2345D007  0.670   C114G55     E8     <NA>   940
1616 2345D007  0.670   C114G55     E8     <NA>   940
1617 2348D589  0.801   K021H53     16     <NA>   935
1618 2348D592  0.801   K021H53     16     <NA>   935
1619 2340D586  0.861   B081B99     <NA>     A1  1104
1620 2340D586  0.861   B081B99     <NA>     A1  1104
1621 145284    0.577   B012F53     19     <NA>  1250
     ESPESOR LARGO  PESO CALIDAD     FECFAB HORFAB
1615   0.670  1900  9440      NA 17-12-2002  05:12
1616   0.670  1756  8710      NA 17-12-2002  05:25
1617   0.800  3016 17530      NA 17-12-2002  05:47
1618   0.800  2988 17440      NA 17-12-2002  06:14
1619   0.861  4110  1380      NA 17-12-2002  23:32
1620   0.861  1389 10430      NA 18-12-2002  00:58
1621   0.600   700  4050      NA 18-12-2002  01:06
     MODOBOB ERRORMEDABS TIPOERROR
1615       0           4       BRC
1616       0          10       ARD
1617       0          12       ARC
1618       0           2         H
1619       0          29     ACXAD
1620       0          21       MRC
1621       0          31     ACVAD
```

Figura 383. Datos principales de las bobinas tratadas antes y después de la parada.

8.3.1.6 RESULTADOS PARADA DE LA BOBINA 23583033 (23 HORAS)

Este es otro caso claro de cambio de composición de aceros.

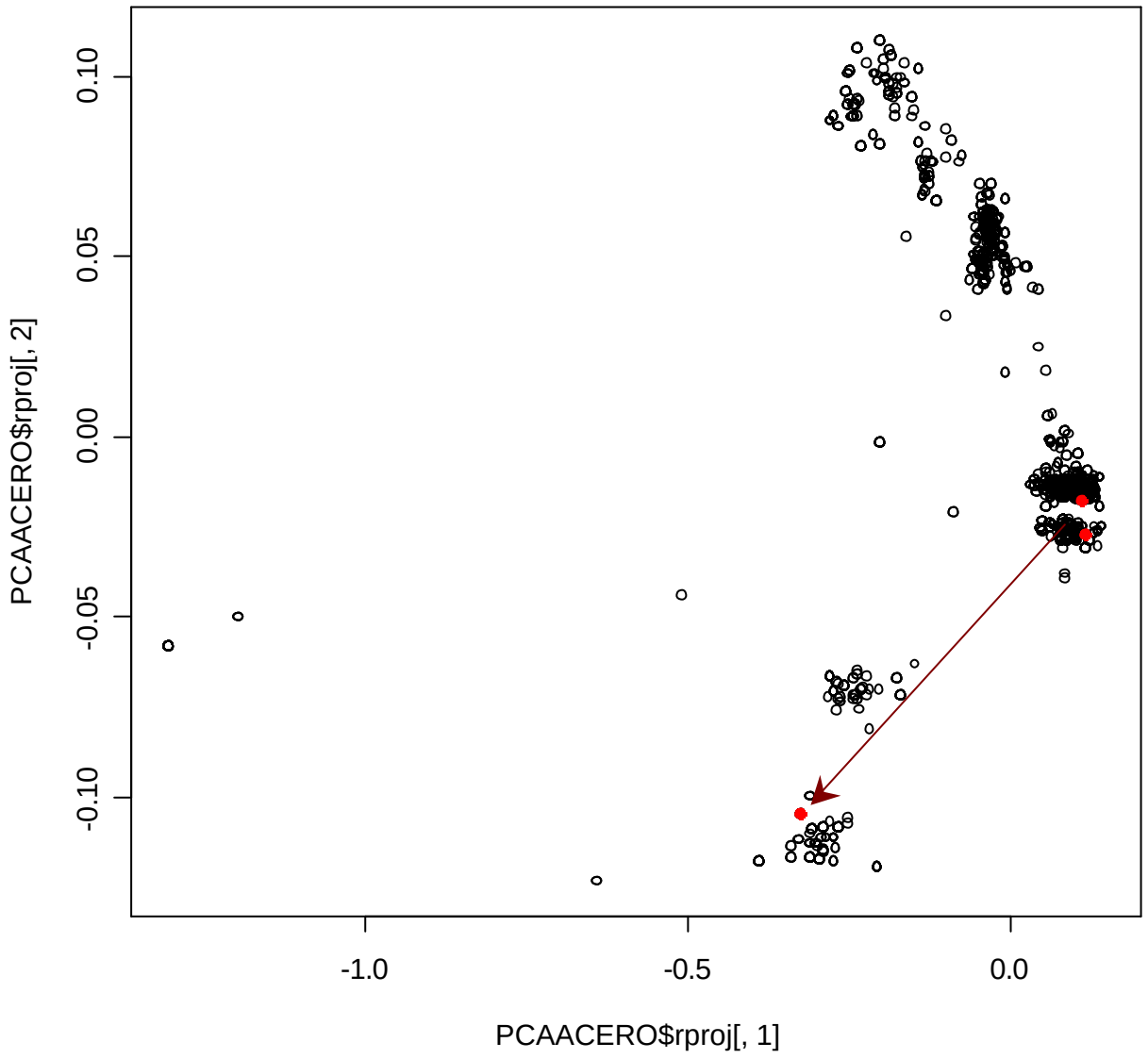


Figura 384. Proyección de las bobinas tratadas.

DATOSPARADA

	CODBOB	XPROJ	YPROJ	DIFHORAS	CODBOBINA
2045	23583030	0.1126794	-0.01782435	0	23583030
2046	23583031	0.1191356	-0.02768558	0	23583031
2047	23583032	0.1191356	-0.02768558	0	23583032
2048	23583033	0.1191356	-0.02768558	23	23583033
2049	23593001	-0.3235468	-0.10471079	0	23593006
2050	23593002	-0.3235468	-0.10471079	0	23593007
2051	23593003	-0.3235468	-0.10471079	0	23593008

	BOBENT	ESPENT	CLASACERO	DUREZA	CICREC	ANCHO
2045	2356D064	0.801	B120G55	50	NA	1028
2046	2356D061	0.801	B100B95	50	NA	1028
2047	2356D060	0.801	B100B95	50	NA	1028
2048	2356D062	0.801	B100B95	50	NA	1028
2049	2356D052	0.701	B120G55	50	NA	1070
2050	2356D053	0.701	B120G55	50	NA	1070
2051	2357T503	0.701	B120G55	50	NA	1070

	ESPESOR	LARGO	PESO	CALIDAD	FECFAB	HORFAB
2045	0.8	3052	19520	NA	24-12-2002	17:04
2046	0.8	2864	18340	NA	24-12-2002	17:25
2047	0.8	2891	18500	NA	24-12-2002	17:47
2048	0.8	2844	18180	NA	24-12-2002	18:10
2049	0.7	3110	18130	NA	25-12-2002	18:51
2050	0.7	3088	18010	NA	25-12-2002	19:22
2051	0.7	2886	17000	NA	25-12-2002	19:47

	MODOBOB	ERRORMEDABS	TIPOERROR
2045	0	2	H
2046	0	4	H
2047	0	2	H
2048	0	1	H
2049	0	11	ACVAD
2050	0	0	H
2051	0	10	MCXAD

Figura 385. Datos principales de las bobinas tratadas antes y después de la parada.

8.3.2 EVALUACIÓN DE LOS MODELOS DE CONSIGNAS

Para evaluar los modelos de consignas se obtienen, de la nueva base de datos: los espesores, anchuras y temperaturas de entrada de banda de aquellas bobinas con errores mínimos y curvas horizontales de temperatura y velocidad. Se introducen en los modelos, y se verifica el error frente a las temperaturas y velocidades de consigna reales.

```
# Obtenemos las bobinas de la familia GRUPO-A
INDGRUPOAESTAC <- MATBOBINAS$TIPOCURVATHF1==”H” & MATBOBINAS$TIPOCURVATMPP1==”H”
& MATBOBINAS$TIPOCURVATMPP2CNG==”H” & MATBOBINAS$TIPOCURVAERROR==”H” &
MATBOBINAS$TIPOCURVAVEL==”H”

# Obtenemos las bobinas
TIPCOD <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$CODBOBINA))

# Sacamos el índice de las bobinas en la matriz de datos dinámicos
INDGMATDINAMIC <- MATDINAMIC$CODBOBINA %in% TIPCOD

#Creamos una matriz con las variables a modelizar
CODBOBMATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$CODBOBINA
TMPP1MATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$TMPP1M
THF1MATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$THC1
THF3MATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$THC3
THF5MATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$THC5
TMPP2CNGMATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$TMPP2C
VELMATSAM3 <- MATDINAMIC[INDGMATDINAMIC,]$VELOCIDADFIN
ERRORSAM3 <- abs(MATDINAMIC[INDGMATDINAMIC,]$TMPP2C-
MATDINAMIC[INDGMATDINAMIC,]$TMPP2M)

# Obtenemos un número de posición para cada bobina
POSINCBOB <- match(CODBOBMATSAM3,TIPCOD)
ANCHOBOB <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ANCHO))
ESPENTBOB <- as.numeric(as.matrix(DATBOBINAS[INDGRUPOAESTAC,]$ESPENT))

# Creamos la anchura y espesor
ANCHOMATSAM3 <- ANCHOBOB[POSINCBOB]
ESPENTMATSAM3 <- round(ESPENTBOB[POSINCBOB]*1000)

MATSAM3 <- cbind(CODBOBMATSAM3, ANCHOMATSAM3, ESPENTMATSAM3, TMPP1MATSAM3,
THF1MATSAM3, THF3MATSAM3, THF5MATSAM3, TMPP2CNGMATSAM3, VELMATSAM3, ERRORSAM3)

# Eliminamos los espúreos
INDGHT <- MATSAM3[,3]>601 & MATSAM3[,4]>209 & MATSAM3[,5]>100 & MATSAM3[,6]>100
& MATSAM3[,7]>100 & MATSAM3[,8]>100 & MATSAM3[,9]>10 & MATSAM3[,9]<200
MATSAM3SIN <- MATSAM3[INDGHT,]

# Guardamos la matriz en un archivo csv
write.table(MATSAM3SIN,"c:\\temp\\NEURONALPERMFEB.CSV",quote=FALSE,sep=",",row.n
ames=FALSE,col.names=FALSE)
```

```
summary(MATSAM3SIN)
 CODBOBMATSAM3       ANCHOMATSAM3  ESPENTMATSAM3   TMPP1MATSAM3    THF1MATSAM3
 Min.   :30013008   Min.   : 770   Min.   : 487   Min.   :150.0   Min.   :783.0
 1st Qu.:30253031   1st Qu.: 978   1st Qu.: 701   1st Qu.:150.0   1st Qu.:832.0
 Median :30363007   Median :1230   Median :1190   Median :242.0   Median :847.0
 Mean   :30335223   Mean   :1173   Mean   :1132   Mean   :213.4   Mean   :843.8
 3rd Qu.:30433079   3rd Qu.:1400   3rd Qu.:1490   3rd Qu.:266.0   3rd Qu.:858.0
 Max.   :30483003   Max.   :1500   Max.   :1980   Max.   :292.0   Max.   :887.0
  THF3MATSAM3   THF5MATSAM3    TMPP2CNGMATSAM3   VELMATSAM3       ERRORSAM3
 Min.   :813   Min.   :835.0   Min.   :775.0   Min.   : 54.00   Min.   : 0.000
 1st Qu.:865   1st Qu.:882.0   1st Qu.:825.0   1st Qu.: 78.00   1st Qu.: 1.000
 Median :885   Median :907.0   Median :825.0   Median : 97.00   Median : 1.000
 Mean   :878   Mean   :898.2   Mean   :828.3   Mean   : 96.97   Mean   : 2.104
 3rd Qu.:895   3rd Qu.:915.0   3rd Qu.:825.0   3rd Qu.:120.00   3rd Qu.: 3.000
 Max.   :917   Max.   :933.0   Max.   :865.0   Max.   :999.00   Max.   :15.000
dim(MATSAM3SIN)
[1] 7515   10
```

Figura 386. Programa que obtiene las variables de aquellas bobinas en régimen permanente y con errores mínimos.

8.3.2.1 SIMULACIÓN Y OBTENCIÓN DEL ERROR

El programa siguiente introduce en los modelos no lineales, las variables de entrada, previamente normalizadas y determina el error final.

```
function MSERROR=simula_consignas(MAT)

%Programa que calcula la red neuronal Backpropagation
%óptima para obtener un error de entrenamiento inferior a un 0,1
%y un error de generalizaci´on menor del 0,5%

%Cargamos los datos de MAT
% COL 1=CODBOBMATSAM3
% COL 2=ANCHOMATSAM3
% COL 3=ESPENTMATSAM3
% COL 4=TMPP1MATSAM3
% COL 5=THF1MATSAM3
% COL 6=THF3MATSAM3
% COL 7=THF5MATSAM3
% COL 8=TMPP2CNGMATSAM3
% COL 9=VELMATSAM3
% COL 10=ERRORSAM3


%MAT = csvread('c:\\temp\\NEURONALPERMFEB.CSV');

    %figure(3)
    %clf
    %hold on
    %plot(1:Longdat,PPI1,'g',1:Longdat,TRAD(1,:),'b:');
    %plot(1:Longdat,TRAD(2,:),'b:',1:Longdat,TRAD(3,:),'b:');
    %plot(1:Longdat,TCNG2,'r',1:Longdat,VELO,'m');
    %plot(1:Longdat,TPI2,'k');
    %ylabel('Temperatura y Velocidad')
    %title('Curvas Reales')
    %hold off
```

```
    % Calculamos la simulacion de las curvas con las redes neuronales

    % Obtenemos con las redes neuronales la THFC1, THFC3 y THFC5
    % Y la velocidad y temp de consigna de pirometro2
    % Normalizamos la Matriz

    MinimoVectMAT= [10000000, 700, 0, 100, 700, 700, 700, 700, 10, 0];
    Vectrang=[30000000, 1300, 2500, 300, 300, 300, 300, 300, 200, 200];
    %Creamos las variables ANCH, TMPP1 y ESPENT normalizadas
    Longdat = size(MAT,1)

    DatosIn =MAT(:,2:4);
    MinimosMAT = ones(size(DatosIn),1) * MinimoVectMAT(2:4);
    MATRange = ones(size(DatosIn),1) * Vectrang(2:4);
    DatosInNorm = (DatosIn-MinimosMAT)./MATRange;

    DatosOut=MAT(:,5:8);
    MinimosMAT = ones(size(DatosOut),1) * MinimoVectMAT(5:8);
    MATRange = ones(size(DatosOut),1) * Vectrang(5:8);
    DatosOutNorm = (DatosOut-MinimosMAT)./MATRange;

    DatosOut2=MAT(:,9);
    MinimosMAT = ones(size(DatosOut2),1) * MinimoVectMAT(9);
    MATRange = ones(size(DatosOut2),1) * Vectrang(9);
    DatosOut2Norm = (DatosOut2-MinimosMAT)./MATRange;

% Cargamos la red de THCxs
    load ('-MAT','C:\temp\ModeloA\consignasTEMP\MATMEJORTEST1.MData');
    netrad = net;

    % Simulamos con los datos
    P=DatosInNorm';
    T=DatosOutNorm';
    [Y] = sim(netrad,P);
    MSEMODELO1 = mse(Y-T)

    % Cargamos la red de velocidad y tmpp2 de consigna
    load ('-MAT','C:\temp\ModeloA\consignasVEL\MATMEJORTEST6.MData');
    netvel = net;
    T2=DatosOut2Norm';
    % Simulamos con los datos
    [Y2] = sim(netvel,P);
    MSEMODELO2 = mse(Y2-T2)

    % Desnormalizamos los datos de entrada y salida
    % Desnormalizamos los datos de entrada y salida
    ANCHO=P(1,:)*Vectrang(2)+MinimoVectMAT(2); %ANCHO
    ESPENT=P(2,:)*Vectrang(3)+MinimoVectMAT(3); %ESPENT
    PPI1=P(3,:)*Vectrang(4)+MinimoVectMAT(4); %TMPP1

    TRAD1=Y(1,:)*Vectrang(5)+MinimoVectMAT(5); %THF1
    TRAD3=Y(2,:)*Vectrang(6)+MinimoVectMAT(6); %THF3
    TRAD5=Y(3,:)*Vectrang(7)+MinimoVectMAT(7); %THF5
    TCNG2=Y(4,:)*Vectrang(8)+MinimoVectMAT(8); %TMPP2CNG

    VELO=Y2(1,:)*Vectrang(9)+MinimoVectMAT(9); %Velocidad



    % ---------------------------------------------------------
```

```
    %Dibujamos las curvas reales
    % Dibujamos los resultados de la red neuronal
    figure(4)
    clf
    hold on
    Longdat=100;
    ini=300;
    plot(ini:ini+Longdat,TRAD1(ini:ini+Longdat),'b.');

plot(ini:ini+Longdat,TRAD3(ini:ini+Longdat),'b.',ini:ini+Longdat,TRAD5(ini:ini+L
ongdat),'b.');

    plot(ini:ini+Longdat,MAT(ini:ini+Longdat,5),'b');

plot(ini:ini+Longdat,MAT(ini:ini+Longdat,6),'b',ini:ini+Longdat,MAT(ini:ini+Long
dat,7),'b');
plot(ini:ini+Longdat,TCNG2(ini:ini+Longdat),'r.',ini:ini+Longdat,VELO(ini:ini+Lo
ngdat)*5,'m.');

plot(ini:ini+Longdat,MAT(ini:ini+Longdat,8),'r',ini:ini+Longdat,MAT(ini:ini+Long
dat,9)*5,'m');

    %plot(1:Longdat,TPI2,'k',1:Longdat,TPI2_REAL);
    %plot(1:Longdat,TPI2,'k');
    ylabel('Temperatura y Velocidad')
    title('Curvas Simuladas')
    hold off

end
```

Figura 387. Programa en MATLAB que evalúa los modelos de consignas finales.

RESULTADOS

Una vez introducidas las nuevas variables, se calculan las temperaturas de consigna de zona del horno y la velocidad de la banda, y se comparan con los valores reales.

Los resultados son lo siguientes:

- Modelo generador de consignas de zona del horno y de pirómetro 2: **Error MSE=0,0032, que corresponde con un 5,66%.**
- Modelo generador de consigna de velocidad de la banda: **Error MSE=0,0030, que corresponde con un 5,48%.**

Vemos que **el error no es muy alto**, aunque lógicamente es mayor con los datos nuevos. En las figuras siguientes se muestran las curvas de consigna reales (en líneas) frente a las simuladas (en puntos). Hay que advertir, que la velocidad está magnificada por cinco (en azul se muestran las temperaturas de consigna de zona del horno, en rojo las de consigna del pirómetro 2, en magenta la velocidad y en negro la temperatura real de la banda).

De todas formas, es probable que los resultados mejorarían si entrenamos las redes neuronales con una cantidad mayor de históricos.

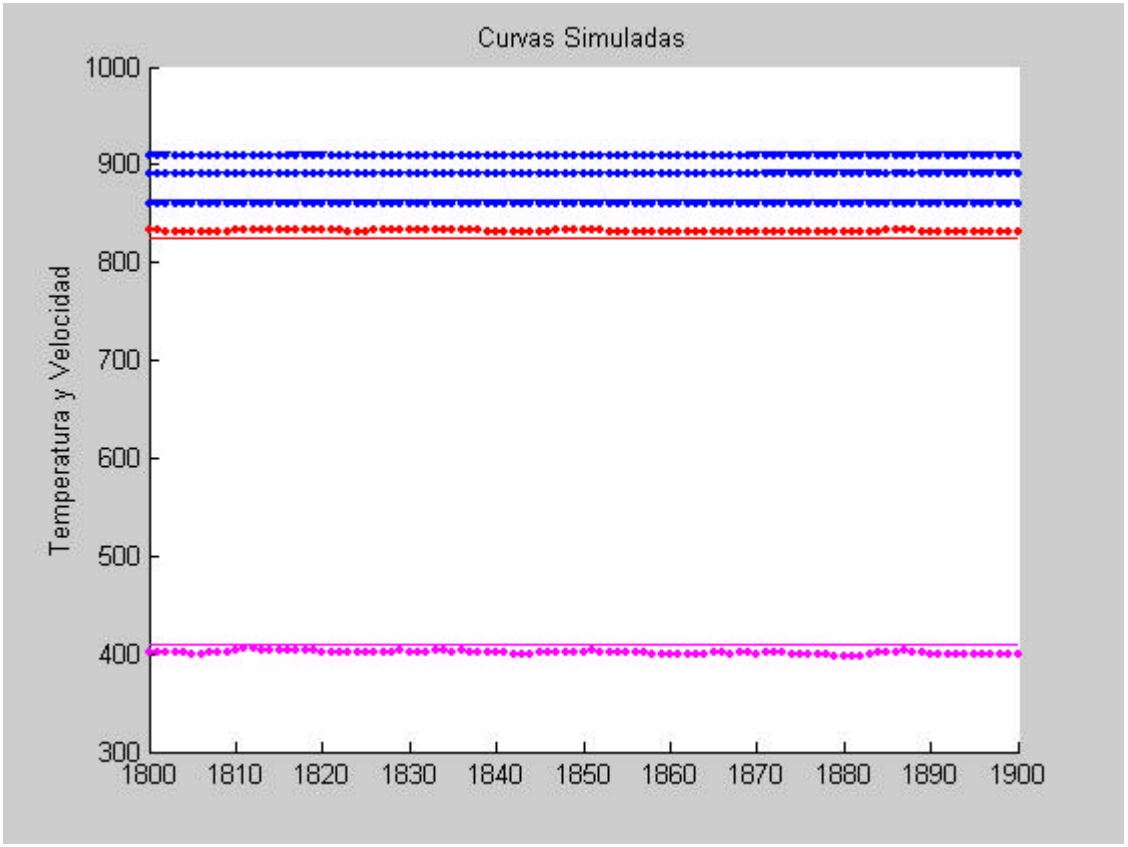


Figura 388. Temperaturas y velocidades de consigna reales(líneas) y simuladas (puntos).

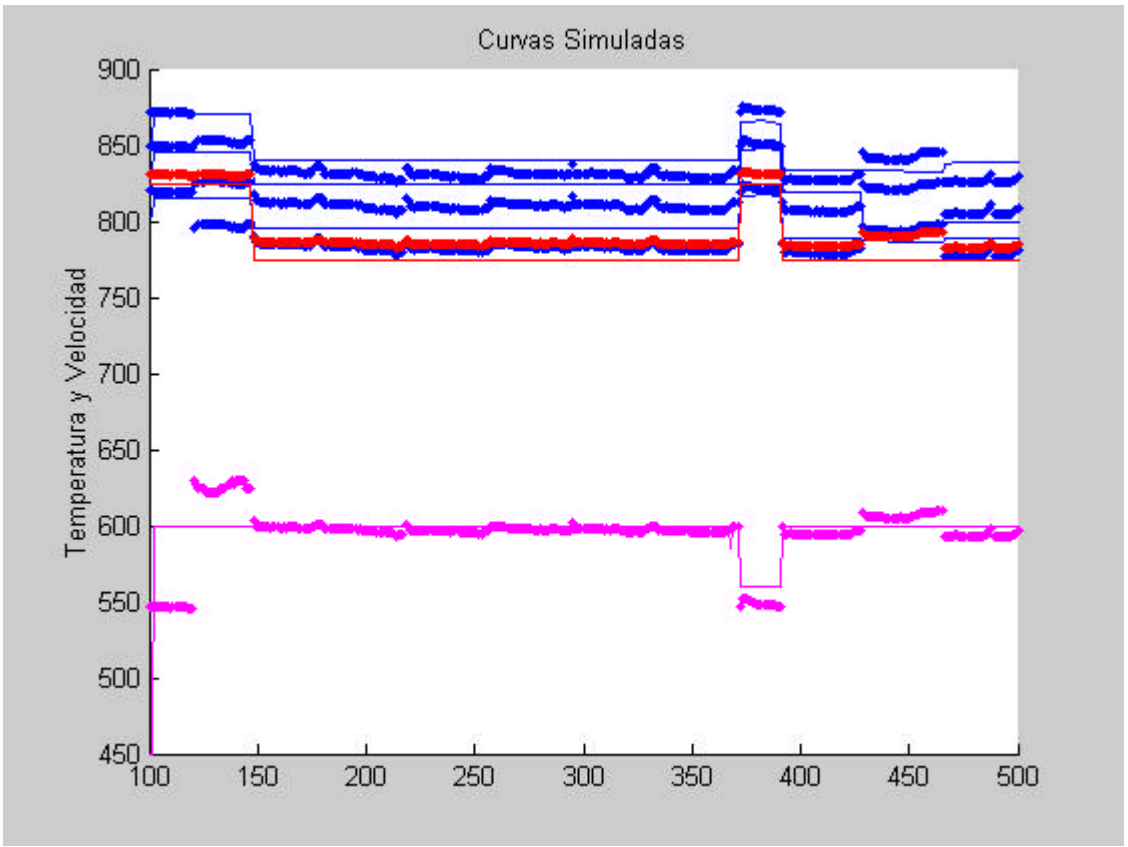


Figura 389. Temperaturas y velocidades de consigna reales(líneas) y simuladas (puntos).

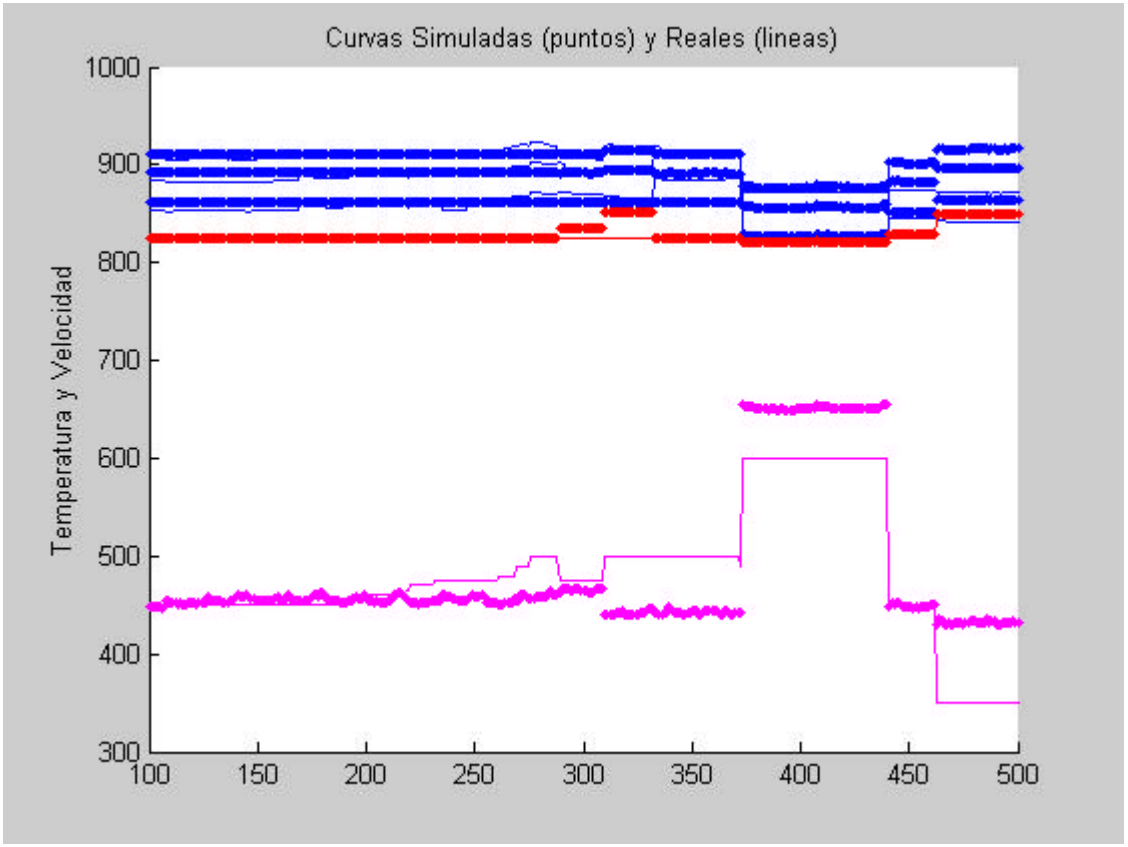


Figura 390. Temperaturas y velocidades de consigna reales(líneas) y simuladas (puntos).

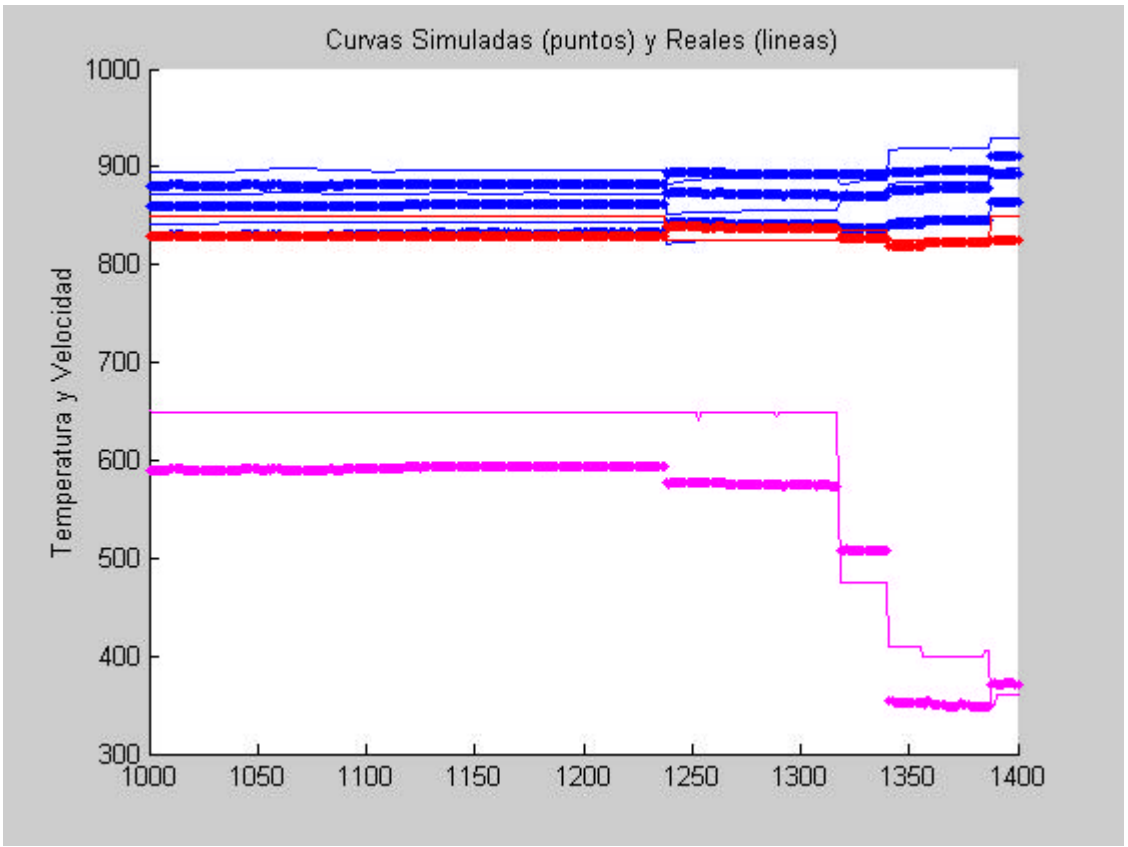


Figura 391. Temperaturas y velocidades de consigna reales(líneas) y simuladas (puntos).

8.3.3 EVALUACIÓN DEL MODELO DE COMPORTAMIENTO DINÁMICO DE LA BANDA

Para poder evaluar el comportamiento del modelo dinámico de la banda, se seleccionan las bobinas de las familias de aceros B105F55 y B100F55 de toda la base de datos, ya que son éstas las que pertenecen al GRUPO-A de bobinas modelizadas, se simula el comportamiento de éstas, siempre que sean consecutivas y, se visualizan las curvas reales frente a las simuladas.

8.3.3.1 RESULTADOS OBTENIDOS

Los resultados obtenidos parecen ser muy buenas, aunque como es lógico el error final es mayor que en la base de datos utilizada para entrenamiento y testeo.

En las figuras siguientes, se puede apreciar el comportamiento del modelo dinámico de la banda ante los nuevos datos de entrada. En azul se muestran las temperatura de consigna de zona de horno, en rojo las de consigna del pirómetro 2, en magenta la velocidad y en negro la temperatura real de la banda.

Los errores varían según la serie de datos analizada siendo cercanos al 5% cuando las variaciones no son muy bruscas y de un 10% cuando se producen variaciones elevadas.

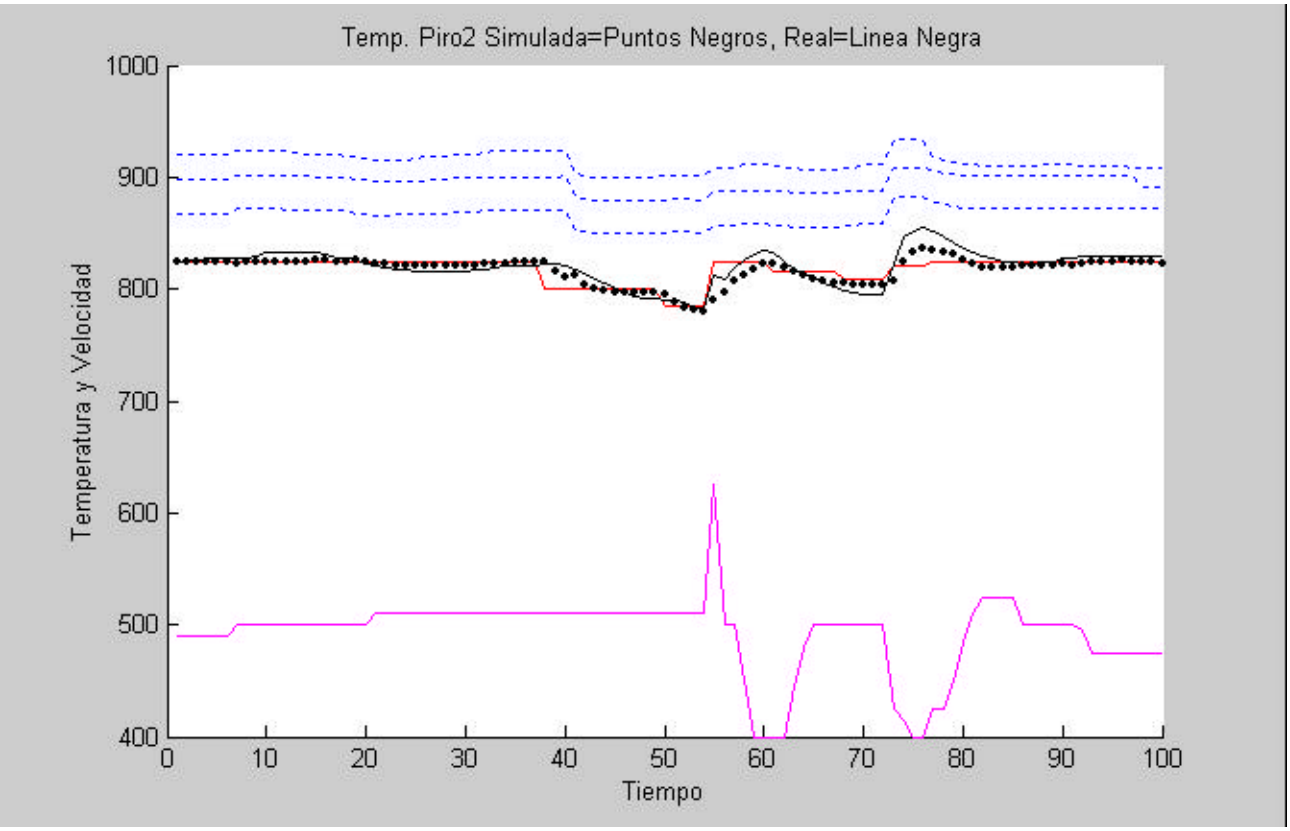


Figura 392. Resultados de la simulación con los nuevos datos: MEANERROR=6,23 ºC y MAXERROR=22 ºC.

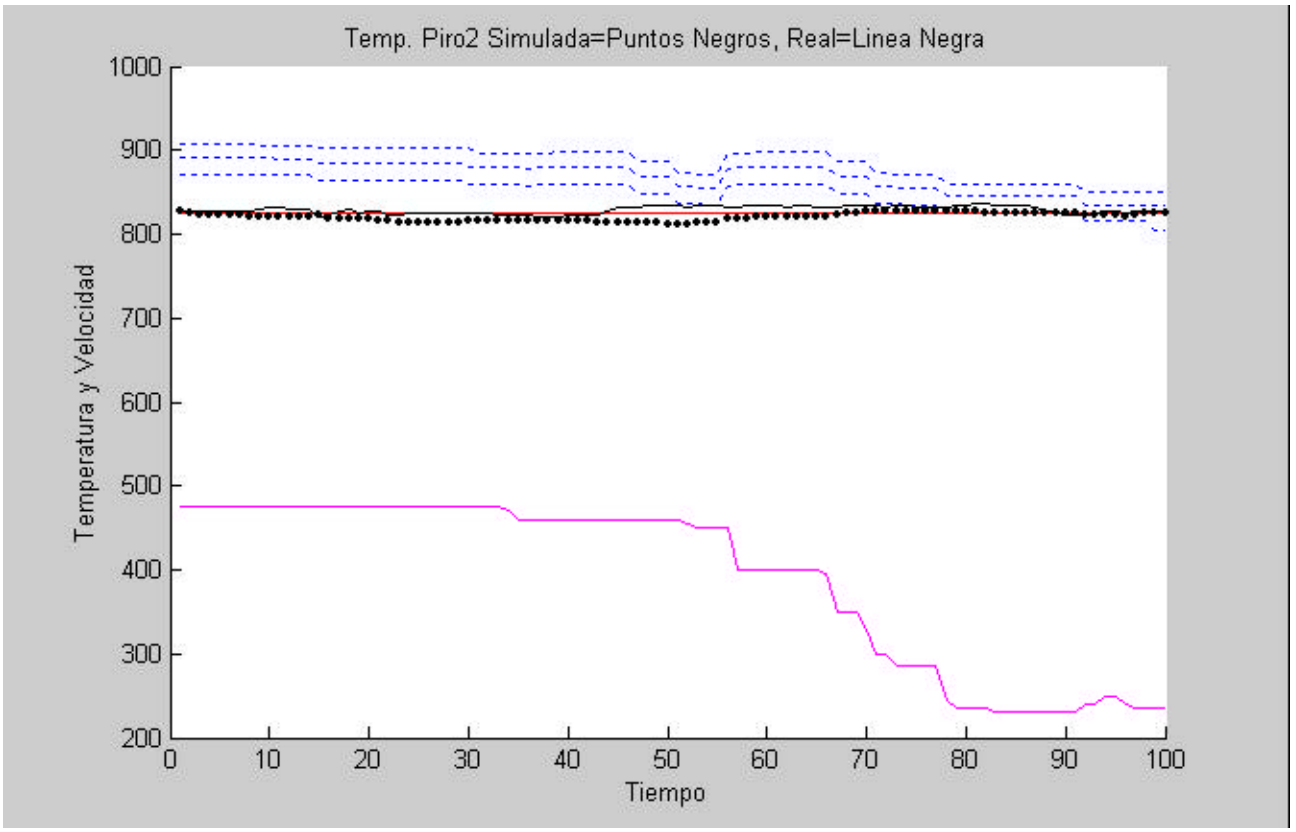


Figura 393. Resultados de la simulación con los nuevos datos: MEANERROR=9,11 ºC y MAXERROR=22 ºC.

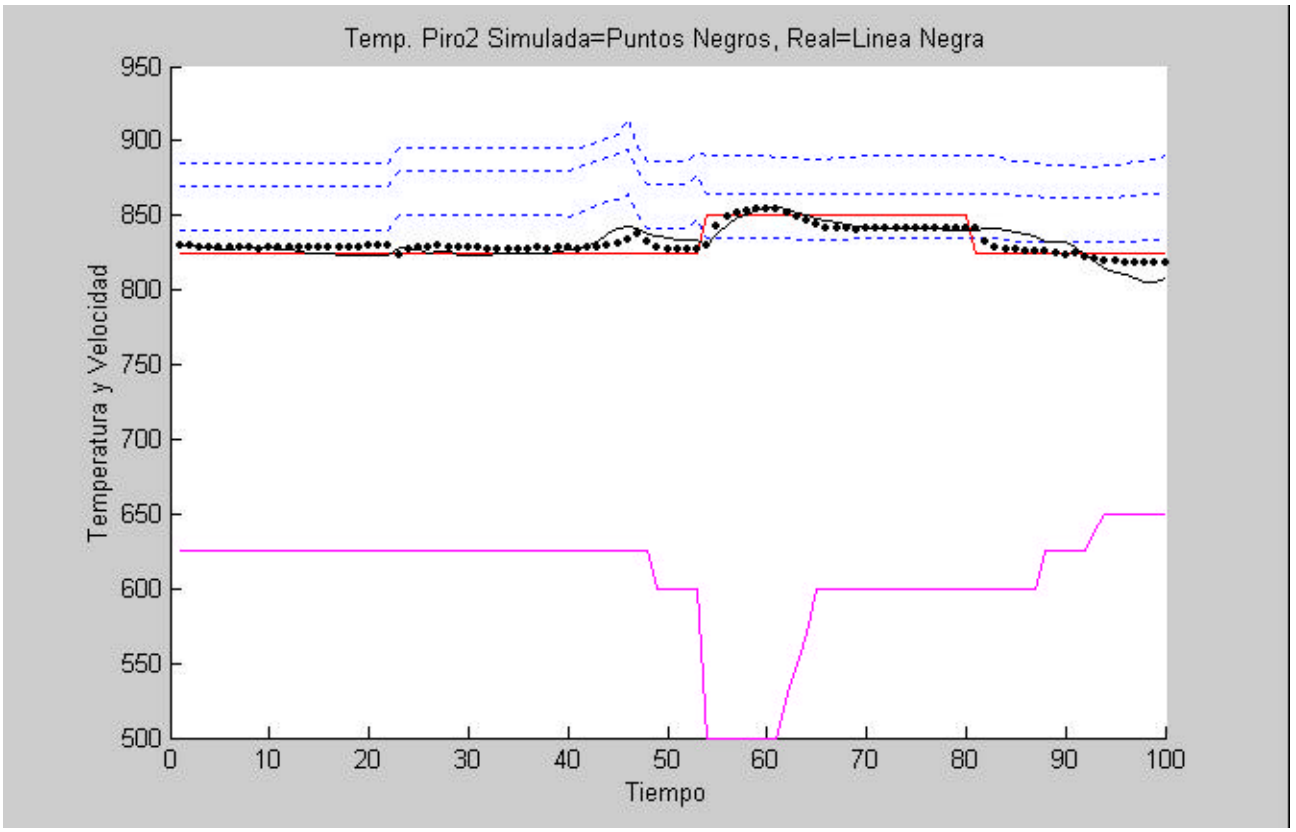


Figura 394. Resultados de la simulación con los nuevos datos: MEANERROR=4,7 ºC y MAXERROR=20 ºC.

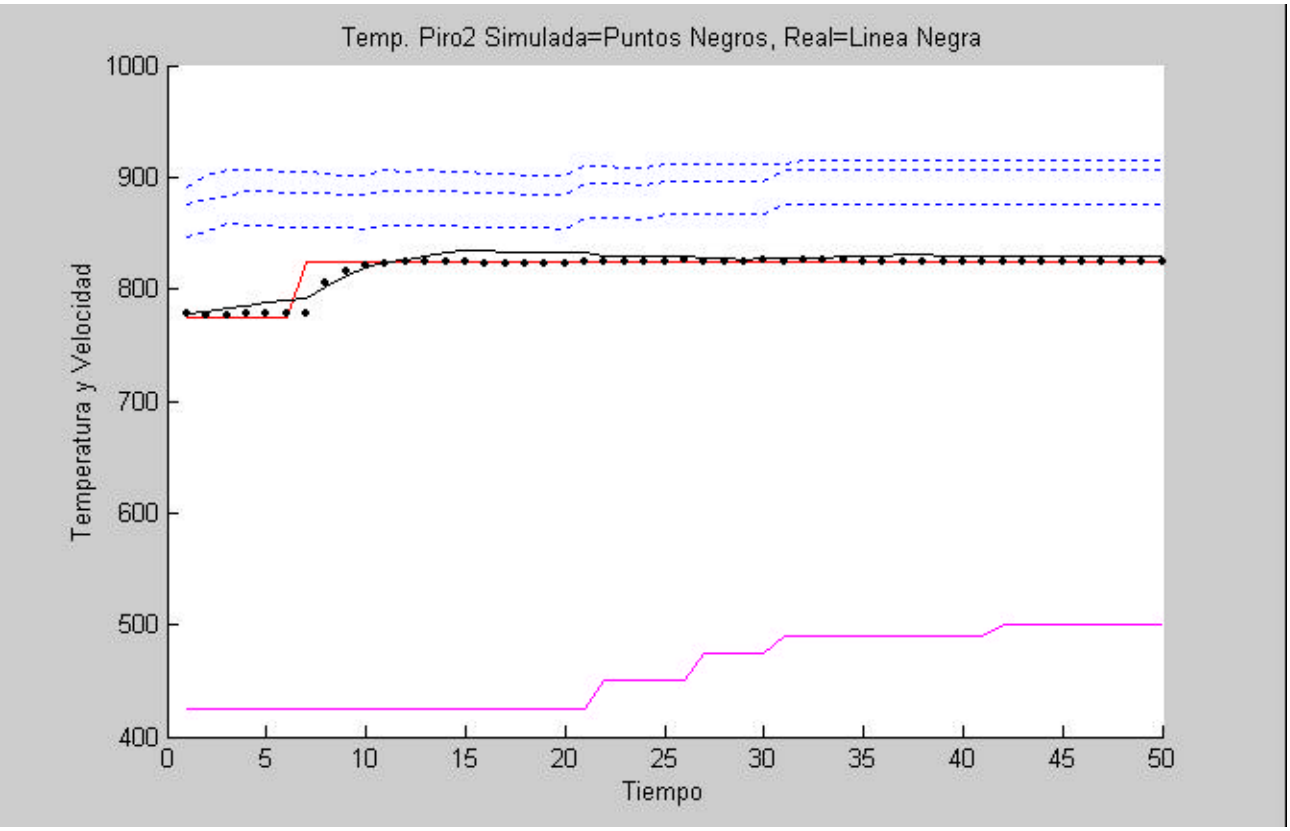


Figura 395. Resultados de la simulación con los nuevos datos: MEANERROR=5,5 ºC y MAXERROR=13 ºC.

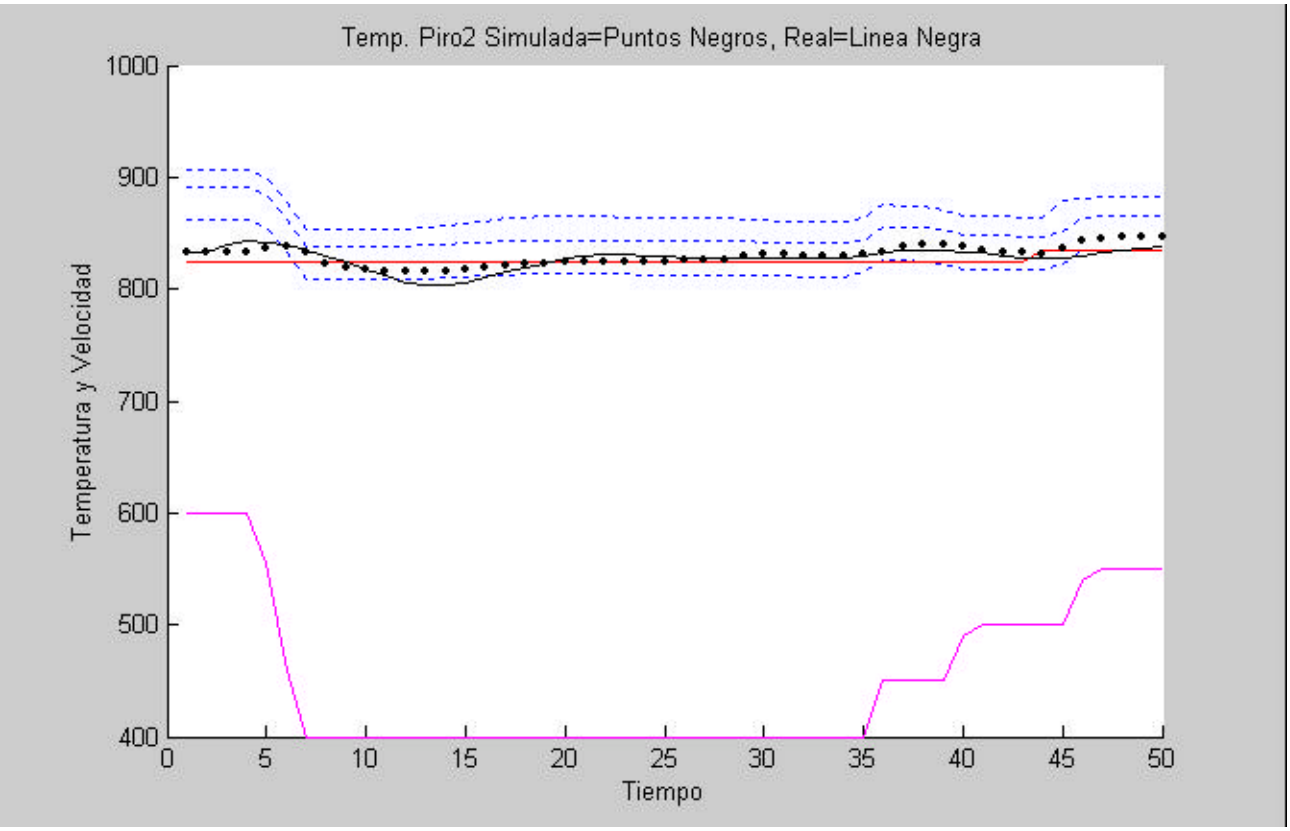


Figura 396. Resultados de la simulación con los nuevos datos: MEANERROR=5,14 ºC y MAXERROR=13 ºC.

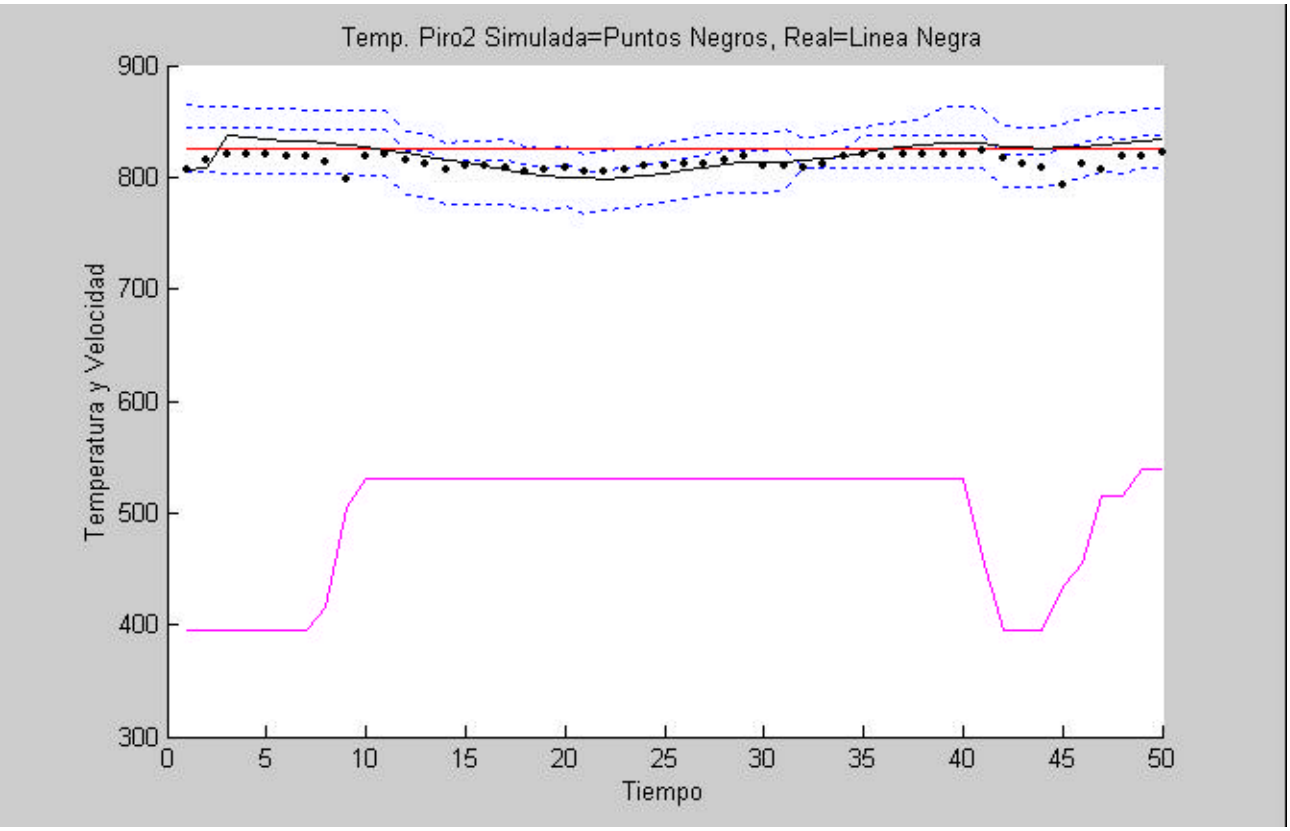


Figura 397. Resultados de la simulación con los nuevos datos: MEANERROR=9,11 ºC y MAXERROR=33 ºC.

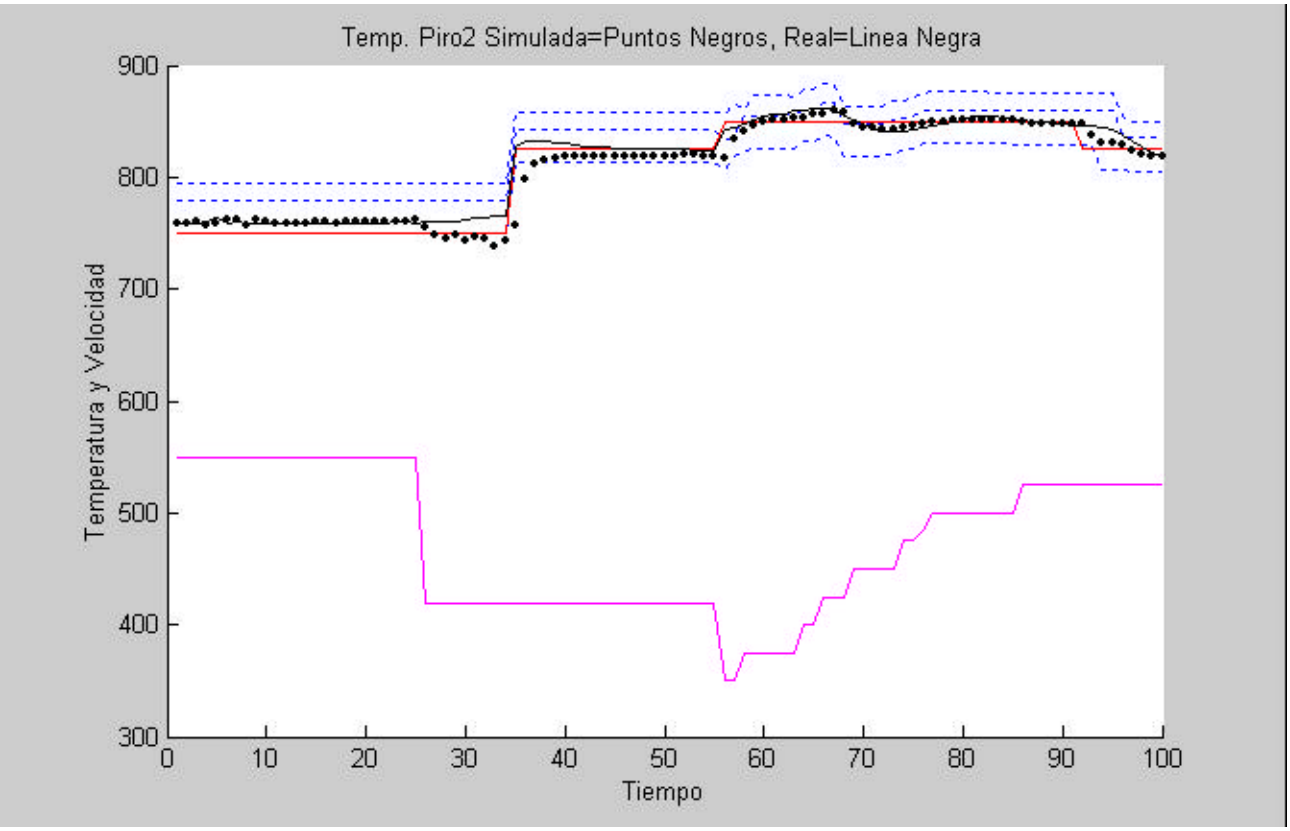


Figura 398. Resultados de la simulación con los nuevos datos: MEANERROR=6,65ºC y MAXERROR=71 ºC.

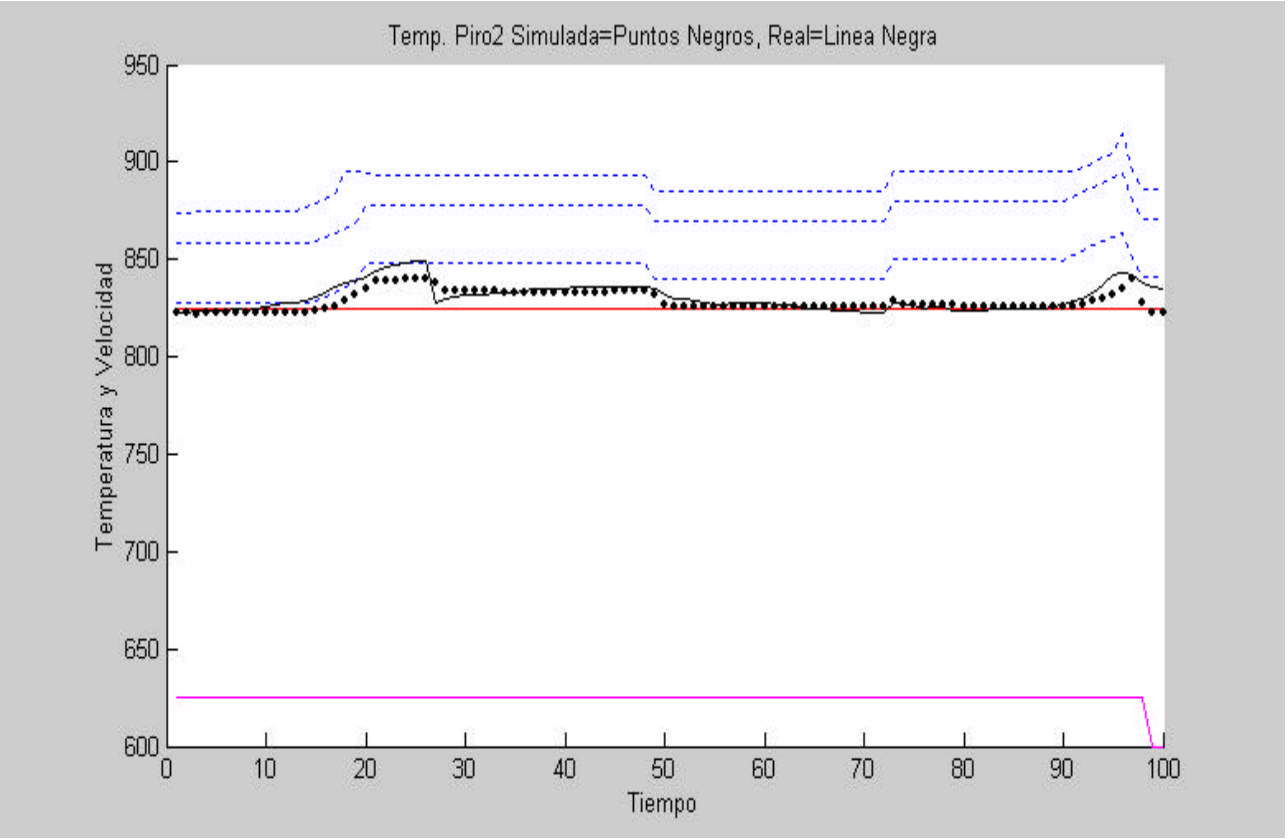


Figura 399. Resultados de la simulación con los nuevos datos: MEANERROR=5,6ºC y MAXERROR=14 ºC.

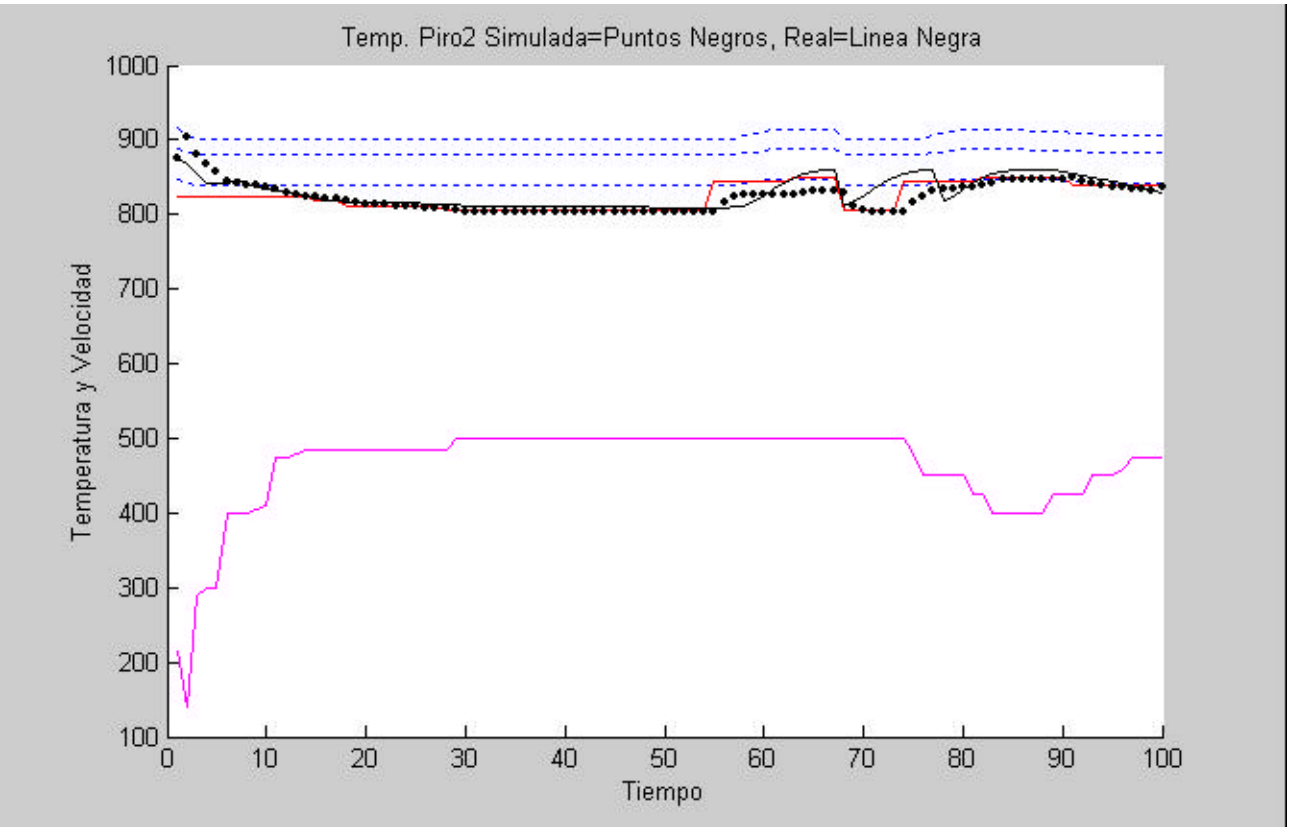


Figura 400. Resultados de la simulación con los nuevos datos: MEANERROR=10,9ºC y MAXERROR=52 ºC.

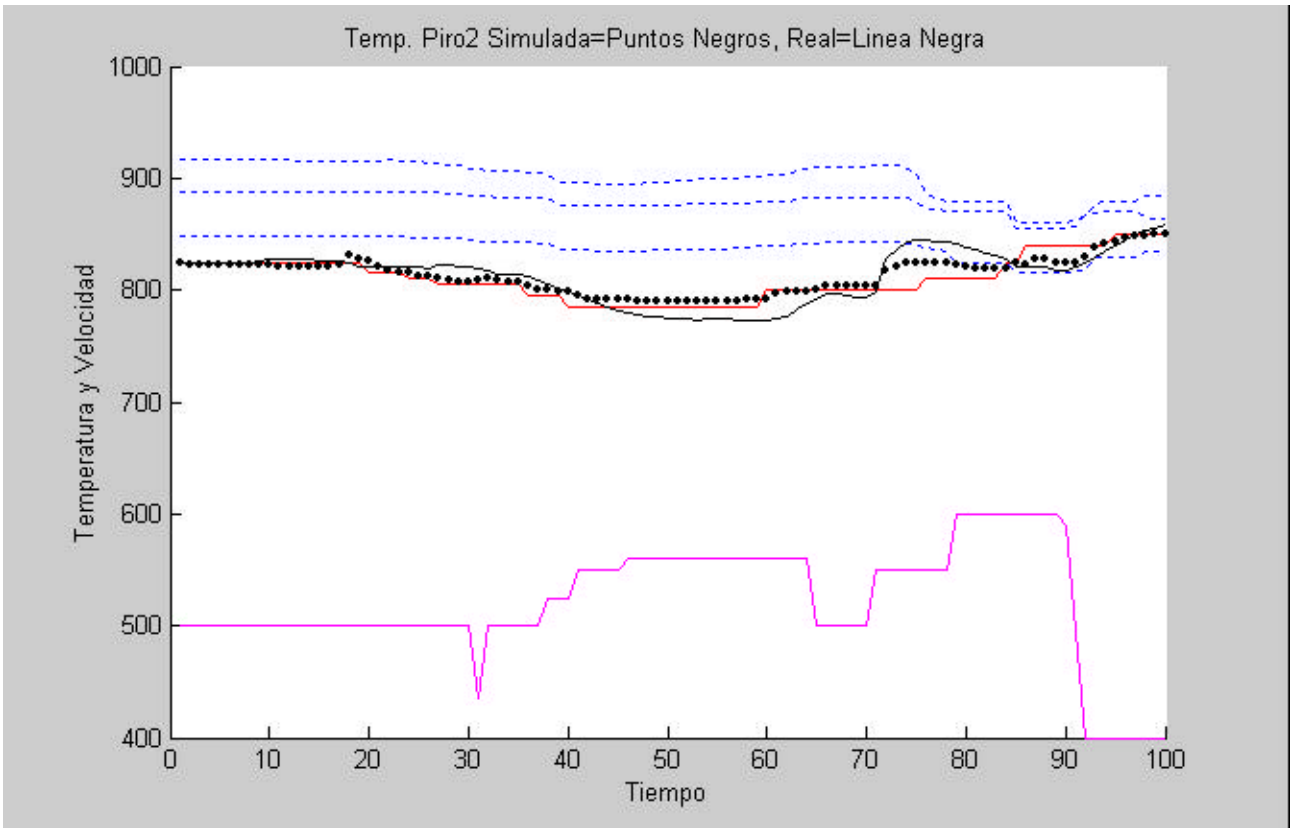


Figura 401. Resultados de la simulación con los nuevos datos: MEANERROR=8,9ºC y MAXERROR=22 ºC.

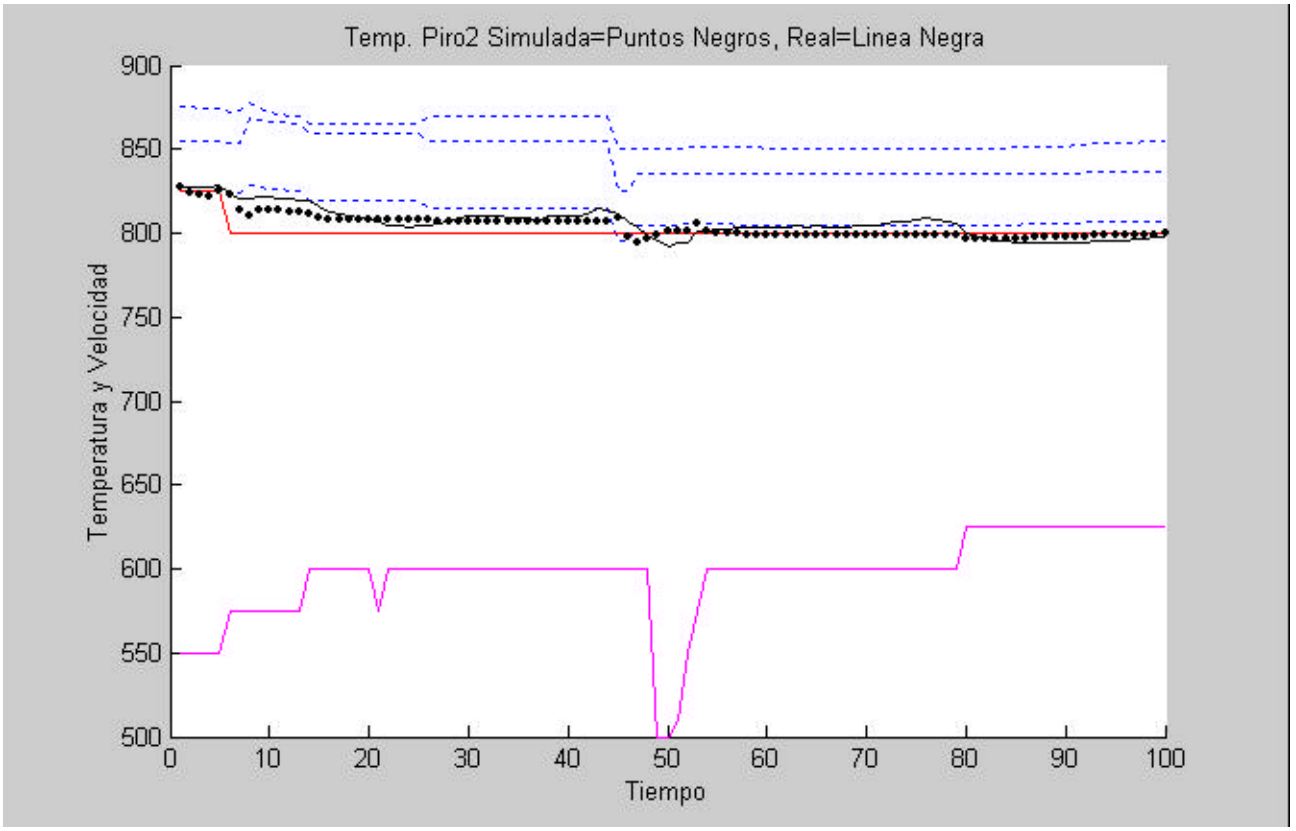


Figura 402. Resultados de la simulación con los nuevos datos: MEANERROR=4,4ºC y MAXERROR=11 ºC.

8.4 CONCLUSIONES

8.4.1.1 CONCLUSIONES DEL CLASIFICADOR DE BOBINAS

El proyector desarrollado para visualizar la composición del acero de cada una de las bobinas, **parece ser una excelente herramienta de predicción para:**

- Detectar cambios de tipos de acero que puedan dar problemas en las temperaturas reales de la banda, tal y como se explicó en el capítulo 6.

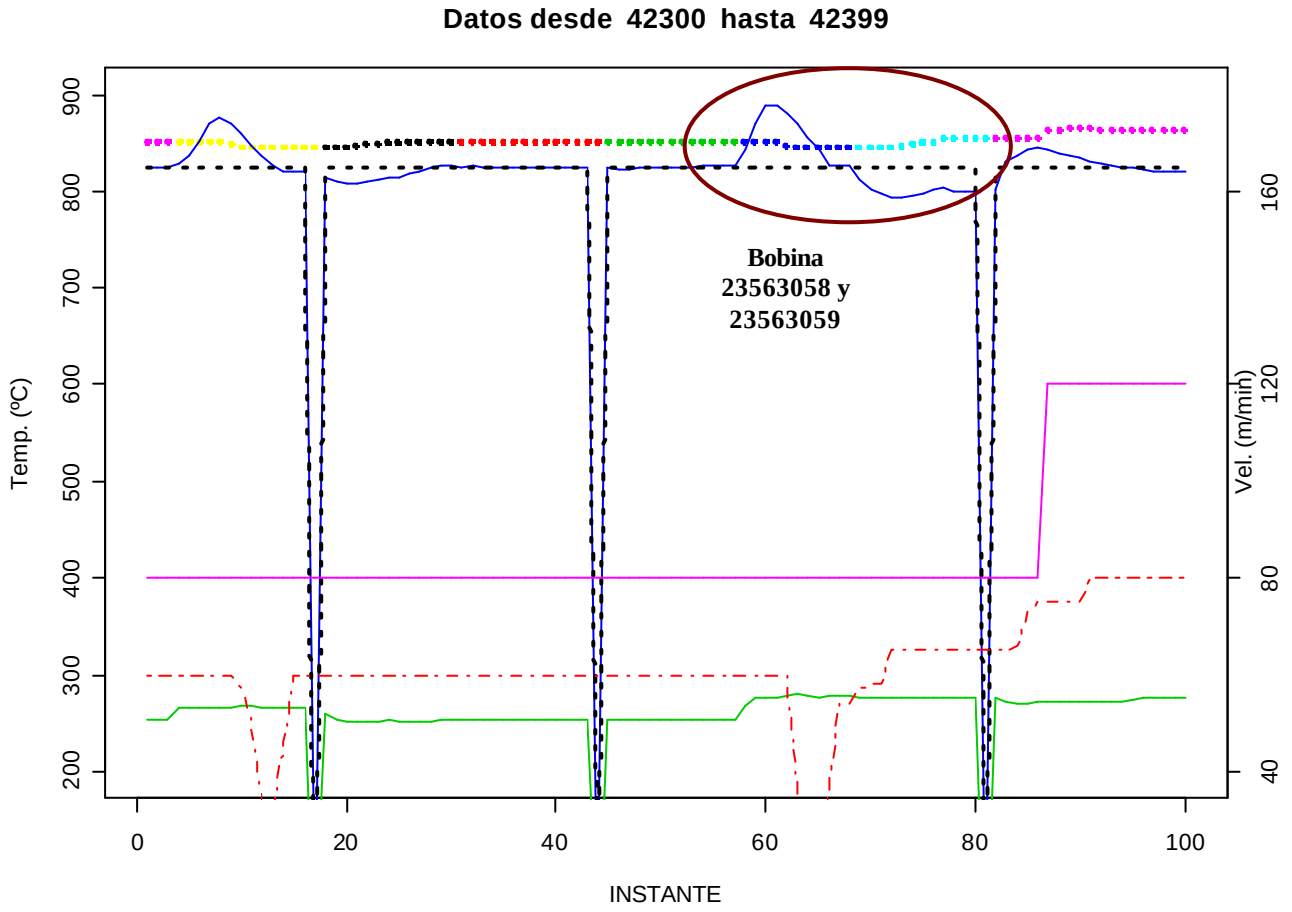


Figura 403. Ejemplo del tratamiento de una serie de bobinas y cómo el error producido en una de ellas es debido a una composición metalúrgica diferente. (bobinas de 23563052 a 23563060).

- Desarrollar un tipo de clasificación más real y práctico de los tipos de aceros, y detectar si la clasificación actual para cada bobina es correcta o no.

- Prever algunas posibles paradas o roturas de banda debido a cambios bruscos en la composición del acero.

Código de la Bobina	Horas de Parada	Distancia en el Proyector con la Siguiente Bobina	Puede ser Explicado con el Proyector
23243047	16	0,8	SI
23313018	5	0,4	SI
23313028	3	0,32	SI
23313038	7	0,9	SI
23323002	3	0,92	SI
23323006	11	0,40	SI
23363025	4	0,05	NO
23373014	4	0,02	NO
23393002	10	0,37	SI
23423036	9	0,09	NO
23443017	4	0,27	SI
23513001	17	0,77	SI
23583033	23	0,45	SI
23653018	3	0	NO

Tabla 72. Paradas explicadas o no.

Esta tercera característica, puede ayudar a reducir costos y problemas en la línea de galvanizado.

En la Tabla 72 se muestran los resultados de la evaluación de este clasificador con las paradas mayores de dos horas producidas durante un mes de trabajo, también se muestra la distancia normalizada de su proyección con respecto a la bobina anterior o posterior y si visualmente se explica la parada o no.

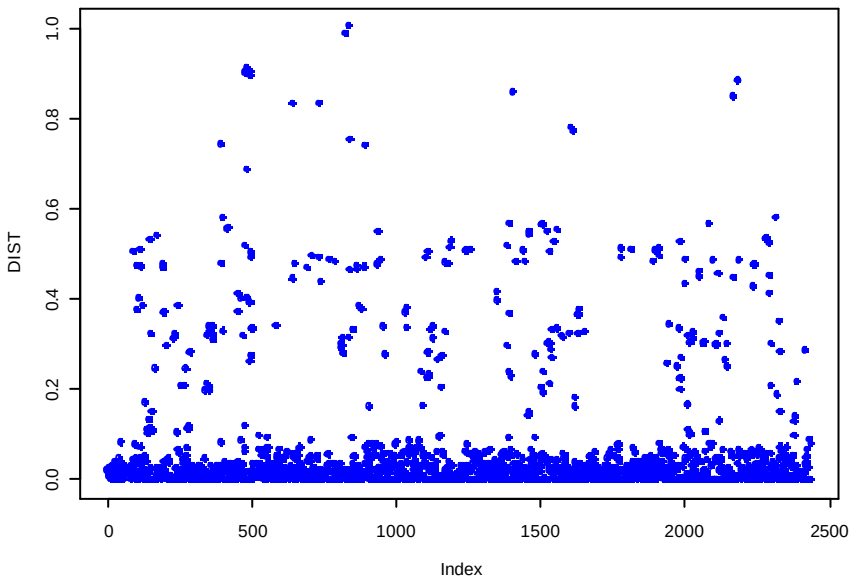


Figura 404. Distancias normalizadas de la proyección de bobinas consecutivas.

En esta tabla, se observa **que una buena parte de las paradas o roturas de la banda pueden ser explicadas y detectadas previamente con el uso de este clasificador ya que sus distancias son elevadas (mayores de 0,30 sobre 1).** Aun así, no siempre cuando las distancias entre bobinas son altas significa que se va a producir una rotura o parada.

Así, si visualizamos las distancias normalizadas de las proyección de bobinas consecutivas, podemos ver que existen un número elevado de ellas cuya distancia se mueve entre el rango 0,3 al 0,6. Lógicamente, todos estos cambios de bobinas de diferentes composiciones no han producido roturas en la banda.

De todas formas, la distancia normalizada de la proyección de bobinas consecutivas puede ser un factor muy importante a considerar para evitar cambios bruscos de aceros que puedan ser causa de roturas no deseadas y, debido a los enormes costes que suponen cada hora de parada, para producir un ahorro destacable en la producción.

Por ejemplo, en el caso estudiado, si considerando unas pérdidas aproximadas de 5.850€ por hora de parada[22] el ahorro según la base de históricos analizada puede alcanzar:

16+5+3+7+3+11+10+4+17+23=99 horas·5.850€=579.150€ (96,36 millones de pesetas en 35 días)

Un ahorro que se sumaría a la reducción de otros costes debidos a:

- Pérdida de calidad en las bobinas siguientes, hasta llegar al estado óptimo de tratamiento.
- Retrasos y pérdida de imagen frente al cliente.
- Pérdida de motivación, de confianza y necesidad de sobreesfuerzos en el personal.
- Etc.

[22] Estos datos corresponden, según fuentes de la empresa, al coste medio por hora considerando los gastos fijos más las pérdidas de producción.

8.4.1.2 CONCLUSIONES DE LA EVALUACIÓN DE LOS MODELOS DE CONSIGNA

Observando las figuras anteriores y viendo los resultados de los errores obtenidos, parece que las consignas pueden ser determinadas, con un cierto grado de error, a partir de las características de la banda y de la temperatura de entrada de la misma, **siempre y cuando sean bobinas del GRUPO-A**[23].

Aún así, los modelos de consigna **podrían ser mejorados sustancialmente si se utilizan históricos del proceso de varios años, en vez de un mes, ya que ayudarían a realizar una red mucho más precisa que abarque muchos más casos.**

Además, puede suceder, que para un mismo tipo de bobina existan diferentes combinaciones de consigna que puedan ser correctas. Esto podría ser abordado con nuevos modelos de redes neuronales que relacionen una de las variables buscada, por ejemplo la velocidad, a partir de las temperaturas de consigna estimadas con otro modelo, lo que permitiría globalizar mejor los comportamientos en modo manual y en modo automático.

8.4.1.3 CONCLUSIONES DE LA EVALUACIÓN DE LOS MODELOS DE COMPORTAMIENTO DINÁMICO DE LA BANDA

Los resultados obtenidos son muy buenos, ya que los errores varían poco según la serie de datos analizada; siendo cercanos al 5% cuando las variaciones no son muy bruscas y de un 10% cuando se producen variaciones elevadas.

Aún así, los modelos de redes neuronales, tanto para el comportamiento dinámico de la banda como para la obtención de las mejores consignas según las características de cada una de las bobinas, **pueden ser mejorados sustancialmente con la utilización de bases de datos más amplias y con la optimización de las redes neuronales utilizadas.**

[23] Ya que será necesario desarrollar un modelo para cada grupo.

CAPÍTULO 9

CONCLUSIONES, APORTACIONES Y LÍNEAS FUTURAS

9.1 CONCLUSIONES

En esta tesis se propone un enfoque nuevo, mediante el uso de herramientas de análisis multivariante y de Minería de Datos, para la planificación y control del horno de una línea de galvanizado de bobinas de acero.

Los primeros pasos realizados, consistieron en una prospección bastante amplia de las técnicas de Minería de Datos más actuales, **desarrollándose unos esquemas de clasificación que pueden resultar de gran ayuda para comprender el "estado del arte" del Data Mining.**

En este trabajo, se ha intentado demostrar que el uso de estas técnicas puede ayudar considerablemente a determinar las causas de los fallos y a comprender con más profundidad un proceso industrial. Los análisis efectuados han pretendido comprender:

- **Las variables más influyentes en el sistema, sus relaciones y dependencias.**
- **El comportamiento del sistema ante las variaciones de consignas.**
- **Gran parte de las causas de los fallos del proceso.**

Una vez analizado el sistema, se desarrolló un clasificador de bobinas según la composición de los aceros. Ésta **herramienta presenta unas posibilidades prometedoras para la predicción de roturas de banda o para las detección de otro tipo de problemas debidos a bobinas con aceros que no pertenecen a ninguna de las familias detectadas**. Estudiando los históricos correspondientes a 35 días del proceso, parece que el simple uso de este clasificador puede generar una reducción de costes considerable ya que, considerando unas pérdidas aproximadas de 5.850€ por hora de parada[24], el ahorro según la base de históricos analizada puede alcanzar:

16+5+3+7+3+11+10+4+17+23=99 horas·5.850€=579.150€ (96,36 millones de pesetas en 35 días)

[24] Estos datos corresponden, según fuentes de la empresa, al coste medio por hora considerando los gastos fijos más las pérdidas de producción.

Este indicador, junto con otros debidos a cambios de anchuras o espesores, **puede ser de mucha utilidad para la planificación de la lista de bobinas a procesar.**

También se ha creado un sensor-software que proyecta los puntos de operación del horno y que **puede complementarse con los programas de control para la planificación y generación de alarmas de los puntos de operación del horno.**

Por último, el trabajo final de esta Tesis se ha centrado en el desarrollo de una metodología que, mediante el uso de algoritmos genéticos y redes neuronales, **permite la optimización de las curvas de consigna entre transiciones, reduciendo la diferencia de temperatura esperada de la banda y la real.**

En los capítulos 6, 7 y 8, se ha desarrollado una serie de técnicas que aplicadas *off-line*, pueden servir de ayuda para planificar las mejores curvas de consigna que "suavicen" los comportamientos de la banda cuando en ésta existan bobinas con diferentes tipos de aceros, anchuras o espesores.

Para ello, se han creado y entrenado diversas redes neuronales que:

- Por un lado, "aprenden" las mejores temperaturas y velocidades de consigna tanto del "modo automático" como del "modo manual".
- Modelizan el comportamiento de la banda ante las diferentes variaciones de temperatura y velocidad.

Los resultados, han sido excelentes, aunque pueden ser aún ser mejorados si se utilizan bases de datos más completas (históricos de un año o más) y se usan otros tipos de redes o formas de entrenamiento de los modelos.

El uso de algoritmos genéticos ha sido muy satisfactorio, ya que la búsqueda de las mejores consignas, para cada transición, no pasaba de los 15 minutos.

9.2 APORTACIONES

Las aportaciones más importantes de esta tesis son:

- Se ha desarrollado una **nueva y actualizada clasificación de las técnicas más útiles de Minería de Datos** para el tratamiento de procesos industriales (ver capítulo 2). Este intenso trabajo, plasmado en más de 200 páginas, se ha apoyado en las últimas publicaciones y tesis doctorales, en el análisis de los artículos más actuales, así como en una intensa búsqueda en Internet y otras fuentes. Además de la clasificación efectuada, se ha desarrollado una librería de cálculo de la "dimensión fractal" que ha sido incluida en la herramienta de análisis estadístico R (de libre distribución) [RPR02]. Además, de todos los trabajos efectuados, se han presentado varias publicaciones [ORD00a][MAR02][PER01].
- Siguiendo las primeras fases de análisis y preparación de los datos de la metodología CRISP-DM, **se han aplicado diversas técnicas de minería de datos que han aportado el siguiente conocimiento sobre el proceso de galvanizado actual**:
  - Existían pequeños errores en los sistemas de adquisición o almacenamiento de los históricos del horno que fueron, más adelante, subsanados. **Se detectaron variables no fiables** como por ejemplo: los valores mínimos o máximos de temperaturas de consigna y reales de subzona, el espesor a la salida del horno, etc. También, se han confirmado o descartado suposiciones iniciales, como por ejemplo: que el modelo actual explica todos los casos, que el horno trabaja correctamente para cualquier dimensión de bobina, etc.
  - Las temperaturas de zona del horno siguen con fidelidad las temperaturas de consigna. **Se observa que el horno tiene capacidad para "llevar" a todas las bobinas a la temperatura de consigna buscada.** Aunque las variaciones de temperatura de la bobina a la entrada, son mantenidas, en cierto modo, a la salida. Los saltos térmicos entre unas zonas y otras, y las temperaturas de las zonas de la parte de calentamiento del horno, **parecen ser adecuados siempre que la variables mantenga un régimen permanente en cada bobina.**
  - Las variables más importantes son: temperaturas de consigna de las zonas 1, 3 y 5, temperatura de entrada y salida de la banda (pirómetros 1 y 2), temperatura de consigna de pirómetro 2, velocidad, características de cada bobina (anchura, espesor y tipo de acero) y modo de uso ("modo manual" o "modo automático").
  - Las variables estáticas como: temperaturas medias, velocidades medias, dimensiones de la banda, etc.; no presentan ninguna causalidad significativa con el "error final". **El estudio y control de las variables del horno debe orientarse hacia una análisis dinámico en modo continuo de las curvas**, evitando la focalización del mismo en las bobinas.

- **Las dimensiones de la banda NO presentan correlación lineal con el "error".** Las diferentes dimensiones (ESPESOR, ANCHURA y LONGITUD) son muy dependientes entre sí debido a que las bobinas proceden de pedidos de iguales dimensiones, por lo tanto, solo se usará una o dos de ellas. La velocidad de la banda, lógicamente, presenta una correlación significativa con las dimensiones de la misma pero no con el error.
- Prácticamente, tres de cada cuatro bobinas han sido tratadas en "modo manual", existiendo aceros que prácticamente no han sido tratados en "modo automático". En el "modo automático", se asegura un 98,3% de bobinas con errores BAJOS frente al 93,1% del "modo manual" de los diez aceros más comunes. Para todos los aceros, el porcentaje es parecido: 98,2% frente al 92,5% del "modo manual". **Comparando los errores entre los dos modos, parece que el número de errores en el "modo automático" es ligeramente menor que en el "modo manual"[25].**

	MODO MANUAL		MODO AUTOMATICO	
Tipo de Error	Núm.	En %	Núm.	En %.
BAJO <=20ºC	1024	93,1%	521	98,3%
MEDIO (>20ºC & <=50ºC)	52	4,7%	7	1,3%
ALTO (>50ºC)	24	2,2%	2	0,4%

Tabla 73. Número y porcentajes globales de tipos de error medio absoluto para los dos modos.

% ERRORABSMANUAL	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	4,3%	10,1%	4,5%	8,4%	4,4%	21,5%	1,5%	0,0%	5,7%	8,0%
BAJO	95,7%	89,9%	95,5%	91,6%	95,6%	78,5%	98,5%	100,0%	94,3%	92,0%

Tabla 74. Comparación con los tipos de error para el modo manual (en porcentajes relativos).

% ERRORABSAUTOMAT	B011F97	B012F53	B025F55	B100F55	B102G33	B102G55	B105F55	C107G55	C114G55	K011F57
MEDIO-ALTO	0,0%	1,7%	0,0%	1,1%	2,4%	0,0%	1,1%	0,0%	12,5%	0,0%
BAJO	100,0%	98,3%	100,0%	98,9%	97,6%	100,0%	98,9%	100,0%	87,5%	100,0%

Tabla 75. Comparación con los tipos de error para el modo automático (en porcentajes relativos).

	TIPO DE ERROR	
SUCESO	ALTOS (>30ºC)	BAJOS (<=30ºC)
Cambio de Acero	22,4%	20,7%
Cambio de Anchura	38,8%	22,9%
Cambio de Espesor	30,6%	19,3%

Tabla 76. Porcentajes de errores ALTOS y BAJOS para cambios de acero, anchura o espesor de bobinas.

[25] Aunque sería conveniente analizar cuándo se producen los cambios del "modo automático" al "modo manual" o viceversa y por qué, ya que estas suposiciones pueden ser equivocadas. Por ejemplo, puede suceder que el operario deba pasar al "modo manual" para resolver contingencias y solo cuando el horno está funcionando correctamente, lo vuelven a pasar a "modo automático", lo que perjudica claramente las estadísticas del "modo manual", ya que los errores aparecerán siempre cuando se resuelven las contingencias en este modo.

- Los errores mayores en la diferencia entre la temperatura de la banda real y la esperada a la salida de la banda son debidos a:
  - Bobinas con aceros de composición irregular.
  - Cambios bruscos en las temperaturas o velocidades de consigna debidos fundamentalmente a: transiciones entre bobinas con aceros de diferentes familias, cambios grandes en anchuras o espesores.
  - Un 66% de los errores ALTOS en "modo manual", NO se producen por cambios de anchura, de espesor o tipo de acero de la bobina (ver Figura 405). Esto **parece indicar que los errores altos en "el modo manual" pueden ser debidos a que, lógicamente, la reacción es más lenta en el manejo de las curvas de velocidades y temperatura de zonas de horno para reducir el error entre la medida de temperatura del pirómetro 2 y la de consigna cuando estos se producen de forma espontánea o, en algunas ocasiones, no existe supervisión durante un cierto tiempo.**

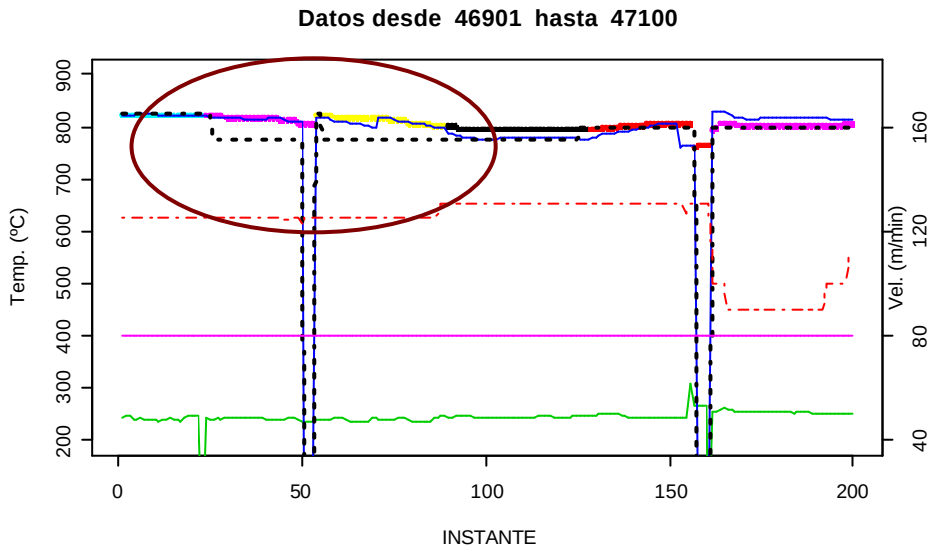


Figura 405. Falta de reacción en "modo manual" para dos bobinas consecutivas.

- Se ha desarrollado un clasificador de bobinas según la composición metalúrgica de los aceros, que **ha permitido clasificar las bobinas en familias y detectar aquellas que se escapan de unos márgenes preestablecidos**. Esto puede servir para detectar anomalías en la composición metalúrgica de la banda que puedan ser causa de:
  - Roturas en la banda.
  - Paradas imprevistas debidas a fallos de otro tipo.
  - Pérdidas del punto de operación óptimo.

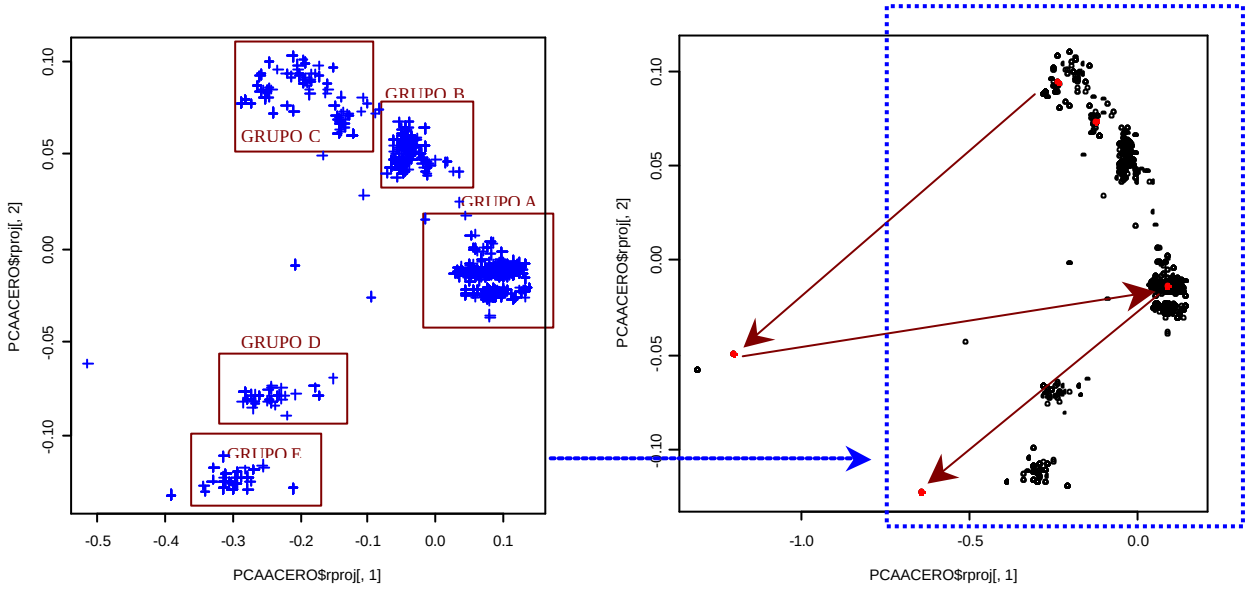


Figura 406. Detección previa de bobinas problemáticas, que pueden producir roturas de la banda.

Código de la Bobina	Horas de Parada	Distancia en el Proyector con la Siguiente Bobina	Puede ser Explicado con el Proyector
23243047	16	0,8	SI
23313018	5	0,4	SI
23313028	3	0,32	SI
23313038	7	0,9	SI
23323002	3	0,92	SI
23323006	11	0,40	SI
23363025	4	0,05	NO
23373014	4	0,02	NO
23393002	10	0,37	SI
23423036	9	0,09	NO
23443017	4	0,27	SI
23513001	17	0,77	SI
23583033	23	0,45	SI
23653018	3	0	NO

Tabla 77. Paradas explicadas o no.

El uso de esta herramienta, junto a otras, para planificar la lista de bobinas a tratar, **puede ayudar considerablemente a reducir el número de paradas y los costes que se derivan de las mismas.**

- Se ha desarrollado un proyector del punto de operación del horno que puede ayudar a detectar tendencias que se "salen" de la "zona normal de operación". **Este sensor-software, junto con el clasificador de bobinas según las características del acero, puede ayudar considerablemente cuando el operario deba cambiar en modo manual las consignas y necesite detectar las tendencias del punto de operación.**
- Por último**, se ha planteado una metodología, basada en algoritmos genéticos y redes neuronales, que permite simular y plantear con anterioridad las mejores consignas de velocidad y temperatura de las zonas del horno, de forma que se reduzca el error final entre la temperatura de la banda y la esperada.**

Todas estas aportaciones se implementan dentro de la última fase de la metodología CRISP-DM, tal y como se explica a continuación.

9.3 LÍNEAS DE FUTURO

9.3.1 APLICACIÓN PRÁCTICA

La última fase de la metodología del CRISP-DM plantea el modo de explotar los resultados.

A continuación, se describen las tareas propias de esta última fase:

- Generación del plan de explotación.
- Modo de vigilar y mantener el sistema de explotación.
- Desarrollo de los informes finales.

9.3.1.1 GENERACIÓN DEL PLAN DE EXPLOTACIÓN

Para poder aplicar todos los resultados obtenidos del DM, se desarrollará un plan que permita implantar en la empresa todas las estrategias desarrolladas.

Para ello, se plantearán los siguientes aspectos:

- **Selección de las mejoras a implantar.**
- **Modo de tratamiento y transformación de la información.**

Sistemas necesarios para la adquisición y transformación de la información: Para desarrollar con garantías los proyectores y modelos no lineales, será necesario una base de datos que abarque una gran cantidad de históricos del proceso, de un año o más de duración, y con la mayor variedad de bobinas posible

- **Forma de manejo de las herramientas de análisis y simulación.**
- **Desarrollo para implantar en la línea de producción de las siguientes herramientas software:**
  - **Organizador de bobinas, adaptado a los ya existentes que, a partir del clasificador de bobinas, la composición del acero y las características físicas de las bobinas**:
    - Detecte aquellas bobinas defectuosas y las elimine de la lista.
    - Optimice la lista de bobinas a tratar, según la composición del acero y dimensiones físicas de la bobina, para que la transición entre ellas sea óptima[26] (ver Figura 407).

[26] Una aproximación bastante interesante para generar listas de bobinas que reduzcan los cambios bruscos de espesores o anchuras de bobinas se desarrolla en [REN99]. En este caso, se hace uso de una matriz de penalización tal y como se realizan los problemas de optimización del viaje de un vendedor de una empresa, donde las ciudades son las bobinas y las distancias entre ellas son las "distancias" entre bobinas. Esta distancia, denominada coeficiente de penalización, se basa en un formula que depende de la diferencia de espesores y anchuras entre bobinas consecutivas. Ha

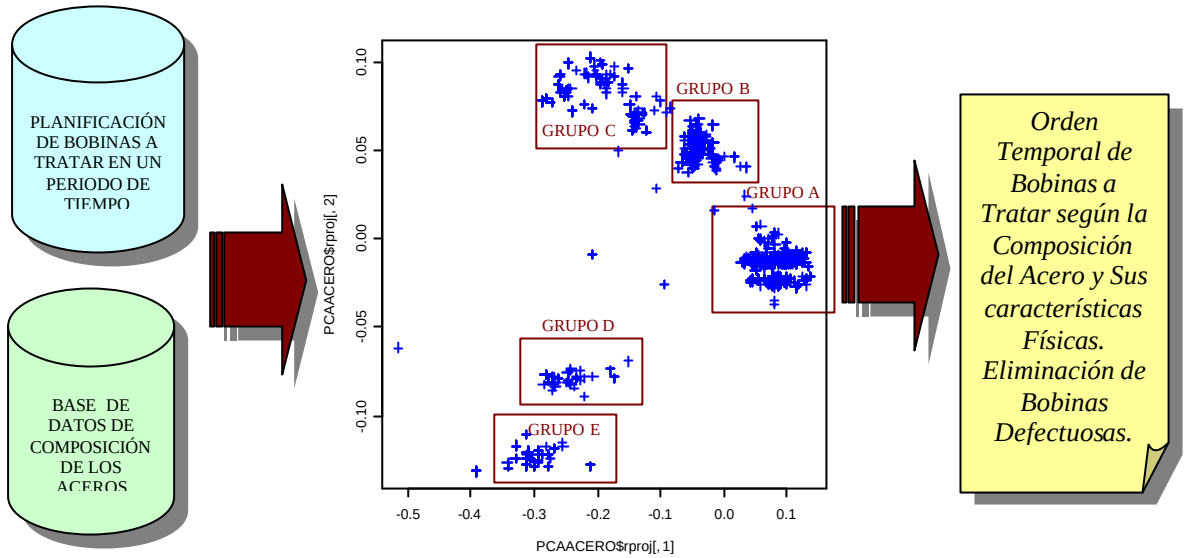


Figura 407. Herramienta de Detección de Bobinas Defectuosas y Optimización de la Lista de Bobinas a Tratar.

- **Generador OFF-LINE, mediante algoritmos genéticos u otras técnicas de control robustas, de curvas de consigna adaptadas a las transiciones (ver capítulo 6).** Se diseñará un software que, a partir de la lista de bobinas a tratar, sus características físicas y los modelos no lineales desarrollados para cada grupo de bobinas; diseñe previamente las curvas de consigna para que el error entre la temperatura de la banda real y esperada sea mínimo (ver Figura 408).

  Los pasos más importantes serán:

  - Generación y verificación de los modelos mediante redes neuronales de señales de consigna y comportamiento de la banda para cada grupo de bobinas según el proyector de aceros anteriormente descrito.
  - Detectar transiciones más problemáticas.
  - Optimizar las curvas de consigna, para cada transición, mediante la simulación del comportamiento de la banda.
  - Almacenamiento de las curvas de consigna mejores.

este coeficiente se le puede añadir un nuevo parámetro que indique la distancia entre puntos proyectados de la composición de los aceros, tal y como se ha explicado en el capítulo 8.

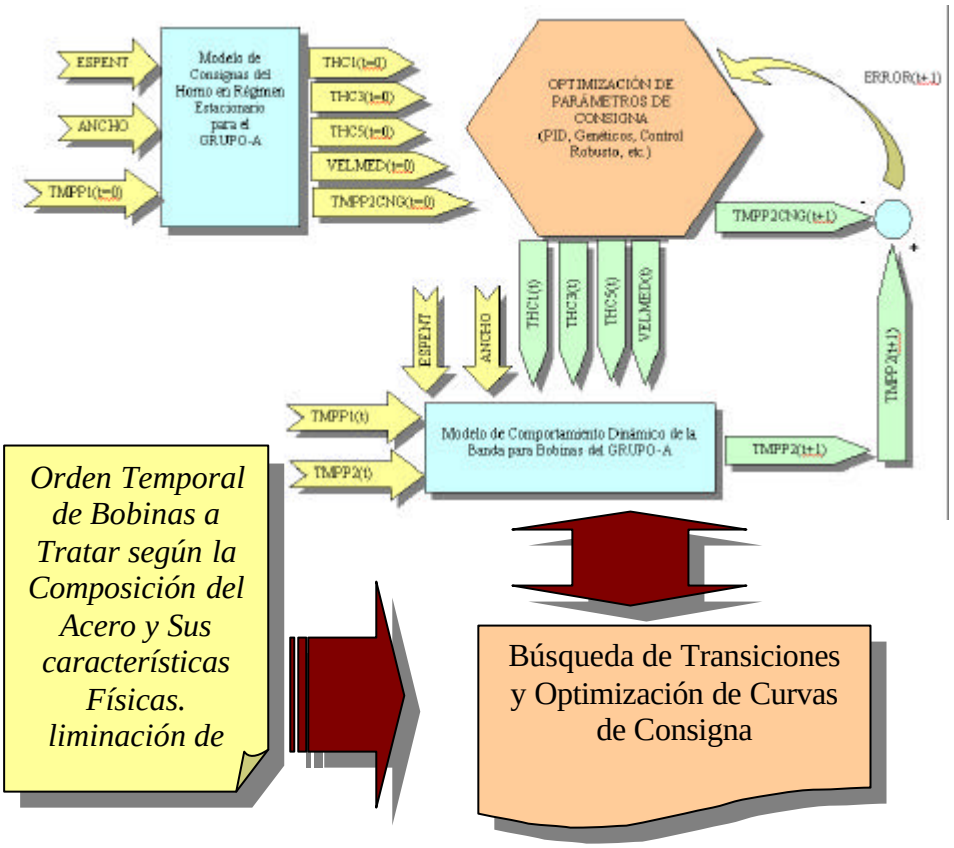


Figura 408. Sistema de optimización de curvas de consigna.

- **Se implementará una herramienta de visualización ON-LINE de los puntos de operación del horno (ver capítulos 4, 5, 6 y 7).** Mediante el uso de los proyectores de bobinas, del punto de operación y de las herramientas utilizadas en este trabajo, se planteará un software (ver Figura 409), adaptado a los ya existentes, que:
  - Muestre el comportamiento dinámico de las variables más importantes.
  - Visualice el punto de operación del horno y la tendencia del mismo.
  - Avise de las transiciones problemáticas entre bobinas.
  - Indique, el tipo de bobina que se está tratando en ese momento.

➢ **Uso y aplicación de las Herramientas Suministradas**:

- Manejo de las herramientas suministradas.
- Interpretación de los resultados.

➢ **Modo de uso y aplicación de los modelos:**

- Datos que necesitarán los modelos.

- Interpretación de los resultados.

➢ Aplicación en la mejora del proceso.

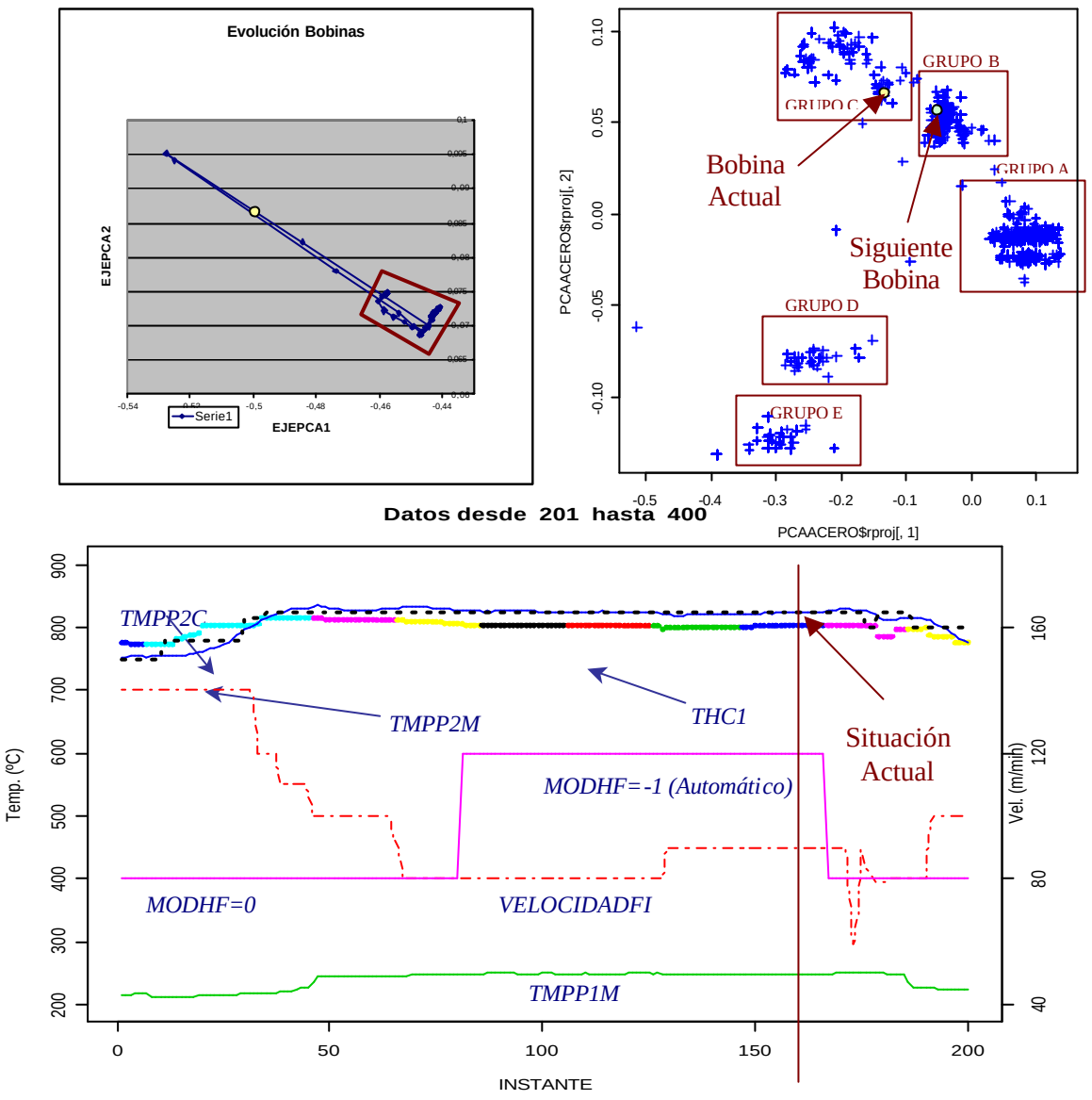


Figura 409. Ejemplo del uso de las técnicas de visualización ON-LINE.

9.3.1.2 PLAN DE MONITORIZACIÓN Y MANTENIMIENTO

Se describirán las tareas para:

- Comprobar la eficiencia de uso de las técnicas planteadas.
- Detectar posibles fallos no considerados.
- Modo de generación de informes periódicos.
- Futuras fuentes de ampliación.

9.3.1.3 GENERACIÓN DE LA DOCUMENTACIÓN FINAL Y REVISIÓN PERIÓDICA

Todo el proceso CRISP-DM, se describirá concienzudamente en una serie de documentos que facilite su consulta.

Se realizará una presentación de los pasos realizados, los resultados y conclusiones obtenidas, así como de las estrategias planteadas.

Toda esta documentación, se completará con informes periódicos donde se plasme la experiencia diaria, consideraciones correctas y equivocadas, mejoras realizadas, etc.

9.3.2 ÁMBITO CIENTÍFICO

A medida que los trabajos en esta tesis avanzaban, se abrían otras líneas investigación que podían ampliar o continuar los estudios realizados. De los resultados obtenidos y de la experiencia extraída se plantean las líneas futuras siguientes:

- **Desarrollo de herramientas automáticas de planificación de galvanizado de bobinas mediante proyectores multidimensionales**. Que se sirvan de los proyectores desarrollados en esta tesis u otros (*RADVIZ, PROJECT PURSUIT,* etc.) y de otras técnicas para la mejora de la planificación de la lista de bobinas a tratar, con el objetivo de reducir el número de roturas de banda y paradas, generar avisos, etc. Fundamentalmente deberán tender a reducir los cambios bruscos de espesores, anchuras y distancias de tipos de aceros.
- **Utilización de otras arquitecturas de redes neuronales (ART1, ART2, Fuzzy-ART, etc.) para la mejora de los modelos no lineales del comportamiento dinámico de la banda y de las curvas de consigna**. De forma que se mejore la planificación de las curvas de consigna y el comportamiento de la banda ante las mismas. Se buscarán modelos más robustos y fiables.
- **Utilización de técnicas de control robusto, para la mejora de las temperaturas y velocidades de consigna.** Se plantea el mismo esquema de optimización desarrollado en esta tesis, pero sustituyendo los algoritmos genéticos por otro tipo de técnicas más rápidas que incluso puedan ser aplicadas en tiempo real.
- **Desarrollo de sensores-software,** mediante el uso de herramientas de visualización multivariante y de técnicas de Minería de Datos, que permitan la monitorización, en tiempo real del punto de operación del sistema, la generación de alarmas, la detección de zonas de operación, el análisis de tendencias, etc.
- **Realización de generadores de informes automáticos o semi-automáticos** que, mediante algoritmos clasificadores, generadores de reglas u otras técnicas de minería de datos, extraigan conocimiento del sistema de forma que, generen periódicamente informes que aporten información sustancial para la mejora del proceso y la detección de anomalías o circunstancias adversas. Lógicamente, este aspecto deberá ser desarrollado en estrecha cooperación con el personal tecnológico y de producción de la línea.

BIBLIOGRAFÍA

[ABA01] De Abajo Martínez, Nicolás. "Optimización mediante data mining de modelos para el diagnóstico de calidad en hojalata". Tesis Doctoral. Universidad de Oviedo. Gijón, (2001).

[ACO00] Acosta, M. I.; Salazar, H Zuluaga, C."Curso de Redes Neuronales". Dirección Web: http://ohm.utp.edu.co/neuronales. Universidad Tecnológica de Pereira. Colombia. (2000).

[ADA01] Adamo, Jean-Marc. "Data Mining for Association Rules and Sequential Patterns. Sequential and Parallel Algorithms". Springer. New York, (2001).

[AGO99] AGOCG. Advisory Group on Computer Graphics "An Investigation of Methods for Visualising Highly Multivariate Datasets". Dirección Web: http://www.agocg.ac.uk/reports/visual/casestud/brunsdon/contents.htm (1999).

[AGR93] Agrawal. R. Imeilinski, T. Swami. A. "Mining association rules sets of items in large databases". Proceedings of ACM SIGMOD conference on management of data SIGMOD'93.(1993).

[AGR94] Agrawal, R.; Srikant, R. "Fast algorithms for mining association rules in large databases". Proc International Conference on Very Large Databases, pp. 478-499. Santiage, Chile: Morgan Kaufmann, Los Altos, CA. (1994).

[AHA92] Aha, D. "Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms". International Journal of Man Machine Studies.36(2). 267-287. (1992).

[AND88] Anderson A., Rosenfeld, E., "Neurocomputimg: Foundations of Research". MIT Press, Cambridge, MA.(1988).

[AND98] Andrássyová, E. Paralic, J. "Intelligent Knowledge Discovery" Dept. of Cybernetics and Artificial Intelligence (1998).

[AND99] Andrássyová, E. Paralic, J. "Knowledge Discovery in Databases: A Comparison of Different Views" Dept. of Cybernetics and Artificial Intelligence (1999).

[ANK96] Ankerst, M., Keim, D., and H.-P., K. "Circle segments: A technique for visually exploring lagre dimensional data sets". In Proceedings of the IEEE Visualization Conference. (1996).

[ATK00] Atkinson, A.; Marco, R. “Robust Diagnostic Regression Analysis”. New York: Springer-Verlag Series in Statistic, (2000).

[BAL97] Balcells, Josep; Romeral, José Luis. "Autómatas Programables". Marcombo. Barcelona, (1997)

[BAU89] Baum, E. B.; Haussler, D. "What size net gives valid generalization?". Neural Computation, 1, 151-160. (1989).

[BAU94] Baum, Von L. "LTV stell hot dip galvanizing line upgrade for exposed automotive products". Iron and Steel Engineer. 71, 32-37. (1994).

[BER00] Berry, Michael J.A.; Gordon Linoff. "Mastering Data Mining. The Art and Science of Customer Relationship Management". John Wiley & Sons, USA (2000).

[BER97] Berry, Michael J.A.; Gordon Linoff. "Data Mining Techniques For Marketing, Sales and Customer Support". Editorial Wiley, (1997).

[BIG96] Bigus Joseph P. "Data Mining with Neural Networks". Ed. McGraw Hill, (1996).

[BIS96] Bishop, Ch. M; Svensén, M.; Williams, C.K.I. "The Generative Topographics Mapping" Neural Computation 10. Pag 215-235 (1996)

[BOR96] Bortels, L.; Deconinck, J.; Van Den Bossche, B. "The Multi-Dimensional Upwinding Method (MDUM) as a new simulation tool for the analysis of multi-ion electrolytes controlled by diffusion, convection and migration". Part I. Steady-state analysis of a parallel plane flow channel, Electroanalysis Chemical, p. 404 (1996).

[BOS98] Boosley, K.M. "Neurofuzzy Modelling Approaches in System Identification". University of Southampton, (1998).

[BOX76] Box, G.E.P.; Jenkins, G.M. "Time Series Analysis: Forecasting and Control". 2nd. ed. San Francisco: Holden Day. (1976).

[BRA00] Brauner, Neima; Shacham, Mordechai. "Considering precision of data in reduction of dimensionality and PCA". Computers and Chemical Engineering. 24, 2603-2611. (2000).

[BRA92] Braun H., Riewdmiller M., “Rprop: a fast adaptative learning algorithm”, En Proc. of the Inst. Symposium on Computer and Information Science VII. (1992)

[BRA96] Brachman, R.;Anand T. "The process of Knoviledge discovery in databases: A human centered approach, Advances in Knowledge Discovery and Data Mining". AAAIIMIT Press. (1996)

[BRA98] Bradley, P.S. Fayyad U. M.. Mangasarian 0. L. "Mathematical programing technical report 98", Computer Sciences Department, University of Wisconsin, Madiso, Wi, Enero (1998).

[BRE84] Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. "Classification and regression trees". Belmon Ed. California (1984).

[CAB97] Cabena, Peter; Hadjinian, Pablo; Stadler, Rolf; Verhees, Jaap; Zanasi, Alessandro; "Discovering Data Mining. From Concept to Implementation". IBM. Prentice Hall. New Jersey, (1997).

[CAR87] Carpenter G. A., Grossberg S., "A massively parallel architecture for a self-organizing neural pattern machine", Computer Vision, Graphics, and Image Processing 37, 54-115.(1987)

[CAR88] Carpenter G. A., Grossberg S., "The ART of Adaptive Resonance Theory by a self-Organizing Neural Network", IEEE Computer, 21(3), 77-88. (1988).

[CAS01] Castejón, M.; Ordieres, J.B.; De Cos, F.J.; Mtnez de Pisón, F.J. "Control de Calidad. Metodología para el análisis previo a la modelización de datos en procesos industriales. Fundamentos teóricos y aplicaciones prácticas con R". Logroño: Universidad de La Rioja. Servicio de Publicaciones. (2001).

[CAS03] Castejón Limas, M.; Ordieres, J.; M. Martínez de Pisón, F.J.; Vergara, E.; "Outlier Detection and Data Cleanin in multivariate non-normal samples. The PAELLA Algorithm". (Pendiente de publicación por Data Mining and Knowledge Discovery. ED. Kluwer). (2003).

[CEN87] Cendrowska,J. "PRISM: An algorithm for inducting modular rules". International Journal of Man-Machine Studies, 27 (4):349-370. (1987).

[CHA01] Chang, G.; Healey, M.; McHugh, Jason; Wang, J. "Mining the World Wide Web. An information Search Approach". Kluwer academic Publishers. London, (2001).

[CHE02] Chen, Junghui; Liao, Chien-Mao. "Dynamic process fault monitoring based on neural network and PCA". Journal of Process Control. 12, 277-289. (2002)

[CHE99] Chen, V. C. P. "Application of MARS and Orthogonal Arrays to Inventory Forecasting Stochastic Dynamic Programs". Compuational Statistics and Data Analysis (1999).

[CHI03] Chiang, Leo H.; Braatz, Richard D. "Process monitoring using causal map and multivariate statistics: fault detection and identification". Chemometrics and Intelligent Laboratory Systems. 65, 159-178. (2003).

[COD93] Codd, E.F.; Codd, S.B.; Salley, C.T. "Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate". E.F. Codd&Associates (1993).

[COH95] Cohen, W. W. "Fast effective rule induction. In Proceedings of the 12th International Conference on Machine Learning". (pp. 115--123). Morgan Kaufmann. (1995).

[COH95a] Cohen W. W., Singer, Y. "A Simple, Fast, and Effective Rule Learner". (1995).

[COM99] Compton, P. Richards, D. "Extending Ripple-Down Rules". (1999)

[COT98] COTEC, Documentos sobre oportunidades tecnológicas., "Redes Neuronales", Diciembre 1998, 1ª Ed.(1998)

[CRA79] Craven, P.; Wahba, G. "Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of Generalized Cross-Validation". Numerische Mathematic, 31 (1979).

[CRE93] Creus, Antonio. "Instrumentación Industrial. 5ª Edición". Marcombo. Barcelona, (1993).

[CRI00] Chapman, P. Clinto, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. "CRISP-DM 1.0. Step-by-step data mining guide". Dirección Web: http://www.crisp-dm.org. (2002).

[CRI00a] Cristianini, N; Shawe-Taylor., J. "An introduction to support vector machines". Cambridge University Press. (2000).

[CSI85] CSI Planos, Dirección de Recursos Humanos,. "Línea de Galvanizado de Avilés, Descripción General de la Línea". Avilés (Asturias). (1985).

[CUA02] Cuadrado Vega, Abel Alberto. "Supervisión de Procesos Complejos mediante Técnicas de Data Mining con Incorporación de Conocimiento Previo". Tesis Doctoral. Universidad de Oviedo. Noviembre, (2.002).

[CUE02] Cuevas Cabrera, W.; Dinora Orantes Jiménez, S. "Soporte a la Toma de Decisiones para el Sistema MECA aplicando características OLAP". XI Congreso Internacional de Computación. Mexico D.F. (2002).

[DAE02] Daedalus. "Minería de Datos: Conceptos y Objetivos". Dirección Web: http://www.daedalus.es. (2002).

[DAS03] Daszykowski, M.; Walczak, B.;Massart, D.L. "Projection methods in chemistry". Chemometrics and Intelligent Laboratory Systems. Num, 1345 (2003).

[DAS03] Daszykowski, M.; Walczak, B.; Massart, D.L. "Projection methods in chemistry". Chemometrics and intelligent laboratory systems. (2003).

[DAV02] Yul Huh, Moon: Ryeol Song, Kwang.Dirección Web: http://stat.skku.ac.kr/~myhuh/software/DAVIS/DAVIS.htm. (2002).

[DAV91] Davis, L. "Handbook of Genetic Algorithms" Van Nostrand Reinhold, (1991).

[DEB78] DeBoor, C. "A practical guide to splines." Springer.Verlag. Nueva York (1978).

[DEL02] Dirección Web: http://www.gtic.ssr.upm.es/encuestas/delphi.htm (2002).

[DEM77] Dempster, A. P., Laird,N.M. And Rubin, D.B. "Maximum likelihood from incomplete data via the EM algorithm". Journal of the Royal Statistical Society, Series B, 39:1-38, (1977).

[DIA84] Diaconis, P. y Shahshahani, M. "On non linear functions of linear combinations." SIAM Journal on Scientific and Statistical Computation. 5, 175-191, (1984).

[DIS02] Kloesgen, W; Zytkowhttp, J. Dirección Web: http://orgwis.gmd.de/projects/explora/terms.html."Machine Discovery Terminology". (2002).

[DIX79] Dixon, J.K. "Pattern recognition with partly missing data". Rev: IEEE Transactions on Systems, Man ans Cybernetics, 9:617-621, (1979).

[DON83] Donoho, D.; Huber, P. “The notion of breakdown point, in A Festschrift for Erick Lehmann”. Wadsworth: Belmont C.A, (1983).

[DON85] Donoho, D.L. y Johnstone, I. "Projection based smoothing and duality with kernel methods." Departament of Statistic. Technical Report 238. Stanford University, (1985).

[DRA98] Draper, N.; Smith, H. "Applied regression analysis" John Wiley & Sons, Inc. NY (1998).

[DRE98] Drever International S.A. "Hot Dip Coating Line for ACERALIA, Spain. 030608222 Rev 1.0.". Documentación Técnica del Modelo Matemático. (1998).

[DUD73] Duda, R. 0., Hart P. E., "Pattern Classification and Scene analysis." New York. John Wiley & Sons,(1973).

[DVO91] Dvorak, D.; Kuipers, B. "Process monitoring and diagnosis: a model-based approach". Ref Citeseer IEEE expert (1991).

[EAN94] Eansor, Timothy. "DNN's continuos hot dip galvanizing line". Iron and Steel Engineer. 71, 38-40. (1994).

[EDW76] Edwards, W.J.; Carlton, A. J.; Harvey, G. F.; Evans, R. F. K.; McKerrow, P. J. "Coating mass control system design for a continuous galvanizing line". Automatica. Pergamon Press. Vol 12, 225-235. (1976).

[FAL88] Falhman S. E., "Faster learning variations on back-propagation: An empirical study", Conectionist Models Summer School. T.J.Sejnowski, G.E. Hinston y D.S. Touretzky (Eds.), San Mateo, CA, EEUU.(1988)

[FAL91a] Falhman S. E., Leberie C., "The Cascade Correlation learning architecture", Technical Report CMU CS 90 100, School of Computer Science, Carnegie Mellon University. (1991).

[FAL91b] Falhman S. E., "The Recurrent Cascade Correlation learning architecture", Technical Report CMU CS 91 100, School of Computer Science, Carnegie Mellon University. (1991)

[FAM97] Famili, A.; Shen , W.-M. "Data Preprocessing and Intelligent Data Analysis". Rev: Intelligent Data Analysis. Vol. 1, nº 1, pp 3-23 (1997).

[FAY02] Fayyad, Usama; Grinstein, Georges G.; Wierse, Andreas. "Information Visualization in Data Mining and Knowledger Discovery". Academic Press, UK, (2002).

[FAY96] Fayyad, U.; Haussler, D.; Stolors, P. "Mining Science Data". Comunications of ACM. Vol 39, Nº 11. (1996).

[FAY96a] Fayyad, UM; Simoudis, E. "Data Mining and Knowledge Discovery". Tutorial Notes at PADD'97 - First International Conf. Prac. App. KDD & Data Mining, London, (1997).

[FAY96b] Fayyad, UM; Piatetsky Shapiro, G; Smyth, P. "From Data Mining to Knowledge Discovery: an overview". Advances Discovery and Data Mining. AAAI Press/MIT Press, (1996), 1-36.

[FRA98] Frank, E; Witten. I. H. "Generating Accurate Rule Sets Without Global Optimization". In Shavlik, J., ed., Machine Learning: Proceedings of the Fifteenth International Conference, Morgan Kaufmann Publishers, San Francisco, CA. (1998)

[FRI81] Friedman, J.H., Wright, M.H. "A nested portioning procedure for numerical multiple integration." ACM Trans. math Software, Vol 7, 76-92, (1981).

[FRI91] Friedman, J. H. "Multivariate adaptive regression splines". The Annals of Statisties, Vol. 19, No 1. 1-141, (1991).

[FUK72] Fukunaga K, Mantock J.M. "A non parametric two-dimensional display for clasification", IEEE Trans. Pattern, Anal. Mach. Intell. PAMI-4, (1972).

[FUK75] Fukushima, K., "Cognitron: a self organizing multilayered neural network", Biological Cybernetics 20.(1975)

[FUK88] Fukushima, K., "Neocognitron: a hierarchical neural network capable of visual pattern recognition", Neural Networks 1, 119-130.(1988)

[GAI95] Gaines, B. R.; Compton, P. "Induction of ripple-down rules applied to modeling large data bases". Journal of Intelligent Information Systems. 5(3):211-228. (1995).

[GAR01] Garcke,J; Griebel, M. "Data mining with sparse grids using simplicial basis functions". Knowledge Discovery and Data Mining (2001)

[GAR02] Garcke, J. Griebel, M. "Classification With Sparse Grids Using Simplicial Basis Functions" (2002)

[GOE99] Goebel, Michael; Gruenwald, Le. "A survey of data mining and knowledge discovery software tools". ACM SIGKDD Explorations. (1999).

[GON94] González Rodríguez, Juan Antonio. "Modelización Matemática de un Horno de Recocido". Trabajo de Investigación. Oviedo. (1994).

[GON99] González Rodríguez, J.A., "Predicción de la Calidad de Bandas Laminadas en Caliente Mediante Modelos Inteligentes", Tesis Doctoral. Universidad de Oviedo, Marzo. (1999).

[GRO76] Grossberg S., "Adaptive pattern classification and universal recoding I II", Biological Cybernetics 20, 121-136. (1976).

[HAI99] Hair, J.F.; Anderson, R. E.; Tatham R.L.; Black, W.C. "Análisis Multivariante". 5ª Edición. Prentice Hall. Madrid, (1999).

[HAL65] Halmos, P. "Teoría intuitiva de los conjuntos". Ed. México. Ed. Continental, (1965).

[HAL86] Halsey C.T., Mogens H.J., Kandanoff L.P., Procaccia I., Shraiman B.I. "Fractal Measures and their singularities: The caracterization of strange sets". Physical Review vol 33, nº 2. (1986).

[HAN01] Hand, David; Mannila, Heikki; Smyth, Padhraic. "Principles of Data Mining". A Bradford Book. The MIT Press. London, (2001).

[HAN02] Hansen, James V.; Nelson, Ray D.; "Data mining of time series using stacked generalizers". Neurocomputing. Noviembre, (2002).

[HAS90] Hastie. T., Tibshirani. R. "Generalized Additive Models". Chapman and Hall, New York, (1990).

[HAY93] Hayes, Arthur R. "Modernization of California Steel's Nº. 1 galvanizing line". Iron and Steel Engineer. 70, 17-19. (1993).

[HAY99] Haykin, S. "Neural Networks. A Comprehensive Foundation." 2ª edición. Prentice-Hall, 1994, (1999)

[HEC87] Hecht-Nielsen, R. "Kolmogorov's mapping neural networks existence theorem". Proc. Int. Conf on Neual Networks, III, pp. 11-13, (1987)

[HEC90] Hecht-Nielsen, R. "Neurocomputing". Addison Wesley, (1990).

[HOL92] Holland, J.H. "Genetic Algorithms" Scientific American, 267, pp. 44-50, (1992).

[HOP82] Hopfield J. J. “Neural networks and physical systems with emergent collective computational abilities”, Proceedings of the National Academy of Sciences, USA, vol. 79, 2554-2558.(1982)

[HOP84] Hopfield J. J. “Neurons with graded response have collective computational properties like those of Two-State Neurons”, Proceedings of the National Academy of Sciences, USA, vol. 81, 3088-3092.(1984).

[HOS89][Hosmer, C.; Lemesshow, S. "Applied discriminant analysis". John Wiley & Sons, Inc. NY (1994).

[HUL00] Hulten, Geoff. Spencer, Laurice. Domingos, Pedro. "Mining Time-Changing Data Streams". ACM. 2000.

[INS87] Inselberg, A., Tuval, C. and Reif, M., "Convexity algorithms in parallel coodinates". Journal of the ACM 34: 765-801. (1987).

[ITI99] IT Innovation Centre. "CRITIKAL. European Project for Large Scale Data Mining". Dirección Web: http://www.attar.com/pages/critikal.htm. (1999).

[JAC95] Jacobs, O.L.R.; "Designing feedback controllers to regulate deposited mass in hot-dip galvanizing". Control Engineering Practice. Vol 3, Nº 11, 1529-1542. (1995).

[JAG93] Jagannathan, V. "Emerging technologies in the hot dip coating of automotive sheet steel". 45, 48-51. (1993)

[JON87] Jones, M. and Sibson, R. "What is projection pursuit (with discussions)". Journal of the Royal Statistical Society 150: 1-36.(1987)

[JON94] Jones, Dennis M. "Coating weigth control upgrade of U.S. Steel fairless hot dip galvanizing line". Iron and Steel Engineer. 71, 62-66. (1994).

[KAR00] Kargupta, H.; Chan, Philip. "Advances in Distributed and Parallel Knowledge Discovery". AAAI Press. California, (2000).

[KDN02] Dirección Web: http://www.kdnuggets.com. KDNuggets. Portal de Minería de Datos, (2002).

[KEI94][Keinbaum, D. "Logistic regression: A self-learning text" Springer-Verlag. NY (1994).

[KIM98] Kimball, Ralph; Reeves, Laura; Ross, Margy; Thornthwaite, Warren; "The Data Warehouse Lifecycle Toolkit". Wiley Computer Publishing.USA (1998).

[KIM98b] Kim, Young; Cheol Moon, Ki; Sam Kang, Byoung; Han, Chonghun; Chang ç, Kun Soo. "Application of neural network to the supervisory control of a reheating furnace in the steel industry". Control Engineering Practice. 6, 1009-1014. (1998).

[KOH77] Kohonen T., "Associative memory: a system theoretical approach", Spring Verlag, Berlin. (1977)

[KOH82] Kohonen T., "Self organizing formation of topologically correct feature maps", Biological Cybernetics, 43, 59-69.(1982)

[KOH88] Kohonen, T. Self-organized formation of topologically correct feature maps. Ed. Springer-Verlag.(1988)

[KRA91] Kramer, Mark. "Nonlinear Principial Component Analysis Using Autoassociative Neural Networks". AIChe Journal.Vol 37, Nº 2. Febrero. (1991)

[LI02] Li, R. F.; Wang, X. Z. "Dimension reduction of process dynamic trends using independent component analysis". Computers and Chemical Engineering. 26, 467-473. (2002).

[LIU01] Liu, Lon-Mu; Bhattacharyya, Siddhartha; "Data mining on times series: an illustration using fast-food restaurant franchise data". Computation Statistics & Data Analysis. (2001).

[LU97] Lu, Yong-Zai. Markward, S. W. "Development and Application of an Integrated Neural System for an HDCL". IEE Transactions on Neural Networks. Vol 8. Nº 6. (1997).

[MAN75] Mandami, E.H., Assilian, S. "An esperiment in lingüisitic, synthesis with a fuzzy logic controller". Int. J. Man-Machine Studies, 7(1), (1975).

[MAO02] Maojo, Victor. "Adquisición de Conocimientos". Departamento de Inteligencia Artificial. Universidad Politécnica de Madrid. (2002).

[MAR01] Martín del Brío, Bonifacio. Sanz Molina, Alfredo. "Redes Neuronales y Sistemas Borrosos. 2ª Edición ampliada y actualizada". Ra-Ma. Madrid, (2001).

[MAR02] Martínez de Pisón, F.J.; Pernía Espinoza, A.; Castejón Limas, M.; González Marcos, A. "Minería de Datos: Herramientas, Técnicas y Metodologías". Proceeding VI International Congress on Project Engineering. Barcelona, (2002)

[MAR95] Martin, B. "Instance-Based learning: Nearest neighbour with generalisation". MSc Thesis, Department of Computer Science. University of Waikato, New Zealand (1995).

[MAT00] Mathworks, Inc. " Neural Network Toolbox User's Guide". Matlab User Guide. Ver 6.0 R12, (2000.).

[MCC43] McCulloch, W. S., Pitts, W., "A logical calculus of the ideas immanent in nervous activity". Bulletin of Mathematical Biophysics 5,115-133. (1943).

[MEH96] Mehta, M.; Agrawal, R.; Rissanen, J. SLIQ: A fast scalbel classifier for data mining. Proceedings of the Fifth International Conference on Extending Database Technology, 1996.

[MIC86] Michalski, R.; Mozetic, I.; Hong, J.; Lavrac, N. "The Multi-Purpose incremental learning system AQ15 ans its testing application to three medical models". Proceedings of the AAAI-86. (1986).

[MIC98] Michalski, R; Bratko, I.; Kubat, M. "Machine Learning and Data Mining. Methods and Applications". John Wiley & Sons LTD. England, (1998).

[MIN54] Minsky M., "Neural Nets amd the Brain. Model problem". Doctoral Dissertation. Princeton University. Pricenton, N.J. (1954).

[MIN69] Minsky M., Papert, S., " Perceptrons. An introduction to computational Geometry." MIT Press. Cambridge, MA.(1969).

[MIS02] Misra, Manish; Henry Yue, H.; Qin, S. Joe.; Ling, Cheng. "Miltivariate process monitoring and fault diagnosis by multi-scale". Computer & Chemical Engineering. Elsevier. 26, 1281-1293. (2002).

[MOR00] Morales, E. "Descubrimiento de Conocimiento en Bases de Datos". Curso KDD On-Line. Dirección Web: http://dns1.mor.itesm.mx/~emorales/Cursos/KDD01/. (2000).

[NAU95] Nauck. D. Beyond "Neuro-Fuzzy. Perspectives and Directions". Technical University of Braunschweig, Germany, (1995).

[NET96] Neter, J.; Kutner, M.H. Nachtdheim, C.J. Wasserman, W. "Applied linear regression models". Richard D. Irwin, Inc. Chicago (1996).

[NIB89] Niblet, T.; Clark, P. "The CN2 induction algoirthm". Machine Learning, 1989.

[NIL65] Nilsson, N., "Learning Machines". Mc-Graw-Hill. New York. (1965).

[OGA93] Ogata Katsuhiko. "Ingeniería de Control Moderna. 2ª Edición". Prentice Hall, (1993).

[OPE01] OPEAL. "Tutorial sobre algoritmos genéticos". Direccion web:http://opeal.sourceforge.net/tindex.html. Universidad de Granada. (2001).

[ORD00] Ordieres, J.; Castejón, M.; De Cos, F.J.; Mtnez de Pisón, F.J. "Análisis de la Importancia del Acero en la Condiciones de Laminación en Caliente". XIV España: Congreso Nacional de Ingeniería Mecánica. (2000).

[ORD00a] Ordieres, J.;Martínez de Pisón, F.J.; De Cos, F.J.;Ortega, Fco. "La Dimensión Fractal como Medida de la Dimensión Intrínseca de Variedades Geométricas". XII Congreso Internacional de Ingeniería Gráfica. España.Valladolid, (2000).

[ORT95] Ortega Fernández, Francisco. "Técnicas de Inteligencia Artificial Aplicadas al Control de Calidad en Procesos Industriales". Tesis Doctoral. Universidad de Oviedo. (1995).

[PAR01] Parr Rud, Olivia. "Data Mining Cookbook". Wiley Computer Publishing. USA, (2001).

[PAR62] Parun.E. "On estimation of a probability density function and mode." Annual of Mathematics and Statistics. 33, 1065-1076, (1962).

[PEÑ97] Peña, Daniel. De River, Sánchez. "Estadística. Modelos y métodos. 2. Modelos lineales y series temporales". 2ª Edición. Alianza Universidad Textos.Madrid, (1989)

[PER01] Pernía, A.; Mtnez de Pisón F.J.; Ordieres J.B.; Castejón M.; De Cos F.J. "Gestión del Conocimiento y Minería de Datos". Actas del XVII Congreso Nacional de Ingeniería de Proyectos. Murcia, (2001).

[PER02] Pernía, A.; González, A.; Alba, F. "Medida de calidad en modelos de procesos industriales". Proceeding VI International Congress on Project Engineering. Barcelona, (2002)

[PIA91] Piatetski-Shapiro G.; Frawley W.J. "Knowledge Discovery in Databases". Ed. AAAI/MIT Press. (1991).

[POT96] Potts, M.; Broomhead, D.S.; Huke, J.P. "Applications of Radial Basis Function fitting to the analysis of dynamical systems" Aston University. DRA and RSRE. Malvern. UK (1996).

[PRE98] Prechelt, L. "Early Stopping. But When?" en "Neural Networks: Tricks of the Trade" Springer-Verlag, (1998.)

[PRU02] Prudsys. "XELOPES Library. Version 1.00". Dirección Web: http://www.prudsys.com/Produkte/Algorithmen/Xelopes/ (2002).

[PYL99] Pyle, Dorian. "Data Preparation For Data Mining". Morgan Kaufmann Publishers. San Francisco, California (1999).

[QUI86] Quinlan, J.R. "Induction of decision trees". Machine learning,1. (1986).

[QUI90] Quinlan, J.R. "Learning Logical Definitions from Relations". Machine Learning, 5:239-266. (1990).

[QUI93] Quinlan, J.R. "C4.5: Programs for machine learning". Morgan Kaufmann, San Francisco (1993).

[QUI93a] Quinlan, J.R. Cameron-Jones, R.M. "FOIL: A Midterm Report". Proceedings of the European Conference on Machine Learning, (1993).

[REN02] Rendueles Vigil, José Luis. "Optimización de Procesos Industriales Complejos Mediante Técnicas de Visualización Multidimensional. Aplicación a un Tren de Laminación en Frío." Tesis Doctoral. Universidad de Oviedo. Diciembre, (2002).

[REN99] Renn, DJ; Stott, KL; Vasko, FJ. "Penalty-based sequencing strategy implemented within a knowledge-based system". Journal of the Operational Research Society. 50, 205-210. (1999)

[ROD00] Rodriguez Montequín, Vicente. "Mejora de los Modelos de Temperatura, Fuerza, Par y Forma de un Tren de Laminación de Chapa Gruesa". Tesis Doctoral. Oviedo. Mayo, (2000).

[ROS59] Rosenblatt F., Principles of Neurodinamycs, Spartan Books, New York. (1959).

[ROU87] Rousseeuw, P; Leroy, A.; “Robust Regression and Outlier Detection”. New York: John Wiley & Sons. (1987).

[RPR02] Dirección Web: http://www.r-project.org. "The R Project for Statistical Computing" (2002).

[RUM85] Rumelhart D. E., Zipser D., "Feature discovery by competitive learning". Cognitive Science 9, 75-112. (1985).

[RUM86] Rumelhart, D. E., McClelland, J. L. "Parallel Distributive Processing: Explorations in the Microstructure of Cognition", Vol.1., MIT Press. (1986).

[SAL00] Salvador Figueras, M. "Análisis Discriminante. En línea". Direccion Web: http://ciberconta.unizar.es/LECCION/discri/INICIO.HTML. Univesidad de Zaragoza. (2000).

[SAM69] Sammon Jr., John W.; "A Nonlinear Mapping for Data Structure Analysis". IEEE Transaction on computers. Vol C-18, Nº 5. (1969).

[SAM93] Samways, Norman L.. "Hot dip galvanizing facilities dedicated at DNN, PRO-TEC and Wheeling-Nisshin". Iron and Steel Engineer. 70, 56-62. (1993).

[SAS01] SAS. "SEMMA. A Proven Data Mining Process". Dirección Web: http://www.sas.com/products/miner/semma.html. (2001).

[SCH89] Schaffer, J.D., Caruna, R.A;. Eshelman, L.J, y Das, R. "A study of control parameters affecting onlin'e performance of genetic algorithms for function optimization". Proceedings of the Third Intemational Conference on Genetic Algorithms, Morgan Kaufmann, (1989)..

[SCO99] Scott, C J and Al-Attar, A and Schneider, W and Nisbet, D and Barth, T and Schwarz, H. "CRITIKAL Final Report". Departament of ECS. University of Southampton, (1999).

[SEB01] Sebzalli, Y.M.; Wang, X.Z. "Knowledge discovery from process operational data using PCA and fuzzy clustering". Engineering Applications of Artificial Intelligence. Num 14, (2001).

[SIE00] Siebes, A. "Data Mining and Statistics". Cism Courses and Lectures, nº 408. International Centre for Mechanical Sciences. CISM. Pag. 1 a 38 (2000).

[SIL85] Silverman.B.W. "Some aspects of the spline smoothing approach to non-parametric regression curve fitting." Journal of Royal Statistical Society b. 47, 1-52 (1985).

[SMI97] Smith, Carlos A.; Corripio, Armando A."Principles and Practice of Automatic Process Control".2th Edition. John Wiley & Sons. (1997).

[SOD89] Soderstrom, Torsten; Stoica, Petre. "System Identification". Prentice Hall. UK, (1989).

[SON92] Sontag E. D., "Feedforward Nets for Interpolation and classification", Computer J, System Sciences 45, 20-48. (1992)

[SPS01] SPSS "Clementine 6.0: Users Guide" SPSS (2001).

[SPS02] Dirección Web: http://www.spss.com. "SPSS Home Page" (2002).

[STA01] Dirección Web: http//www.statsoft.com/textbook/stathome.html. "Libro electrónico sobre algoritmos de data mining" (2001).

[STE01] Steinberg, D.; Bernstein, B.; Colla, P.; Martin, K. "MARS User Guide" Saldford Systems San Diego (2001).

[STE97] Steel, N.C.; Godjevac, J. "Radial Basis Function Artificial Neural Network and Fuzzy Logic Control Theory and Applications Centre" Coventry University, EPFL. Microcomputing Laboratory, Suiza (1997).

[STO77] Stone,C.J. "Non parametric regression and its applications". Annual of Statistics. 5, 595-645 (1977)

[SUG88] Sugeno, M. Kang. G.T. "Structure identification of fuzzy model". Fuzzy sets and systems, (1988).

[SUM01] Sumathi, S.; Sivanandam, S. N.; Balachandar; "Design and development of self-organised neuronal network schemes as a data mining tool". Engineering. Intelligent Systems. Vol 9. Nº 2. Junio (2001).

[TAB96] Tabachnick, B.; Fidell, L. "Discriminant Fuction Analysis. Using Multivariate statistics. Capítulo 11". HarpingCollings College Publishers NY (1996).

[TAK85] Takagi, T.. Sugeno, M. "Fuzzy identification of systems and its applications to modelling and control". IEEE Trans. on Systemes, Man, and Cybernetics. (1985).

[TAK92] Takahashi, T.; Oikawa, Y.; Komori, T.; Ito, I.; Hashimoto, M. "Surface modification of stainless steel in coil by an in-line dry coating process". Surface and Coatings Technology, 51, 522-528. (1992).

[TAN95] Tan, Shufeng; Mavrovouniotis, Michael. "Reducing Data Dimensionality through Optimizing Neural Network Inputs". AIChe Journal.Vol 41, Nº 6. Junio. (1995)

[THU99] Thuraisingham, B. "Data Mining. Technologies, Techniques, Tools and Trends". Ed. CRC Press LLC, (1999).

[TOM02] Tomita, Rosana K.; Park, Song W.; Sotomayor, Oscar A. Z. "Analysis of activated sludge process using multivariate statistical tool- A PCA approach". Chemical Engineering Journal. 90, 283-290. (2002)

[TOW88] Townsend, Carl S. "Closed-loop control of coating weigth on a hot dip galvanizing line". Iron and Steel Engineer. 65, 44-47. (1988).

[VER99] Vergara González, Eliseo. "Modelo de Control Inteligente de Espesor de Recubrimiento en Galvanizado Contínuo por Inmersión". Tesis Doctoral. Universidad de Oviedo. Noviembre, (1999)

[VIT91] Vitale Dori, Edmundo. "Diseño de Filtros Pasivos, Activos y Digitales. Tomos I, II y III". Consejo de Publicaciones. Universidad de Los Andes. Mérida. Venezuela (1991).

[WAL99] Walpole, R; Myers, R; Myers, S. "Probabilidad y Estadística para Ingenieros.". 6ª Edición. Prentice Hall. (1999)

[WAN02] Wang, Haiqing; Song, Zhihuan; Wang, Hui. "Statistical process monitoring using improved PCA with optimized sensor locations". Journal of Process Control. Elsevier. 12, 735-744. (2002).

[WAN99] Wang, Xue Z. "Data Mining and Knowledge Discovery For Process Monitoring and Control". Advances in Industrial Control. Ed. Springer. London, (1999).

[WAT95] Watanabe. K. Hara, K. Koga. S. Tzafestas, S.G. "Fuzzy-neural controllers using meanvalue functional reasoning". Neurocomputing, (1995).

[WEK02] Dirección Web: http://www.cs.waikato.ac.nz/~ml/weka/. "Weka 3 -- Machine Learning Software in Java" (2002).

[WES98] Westphal, Christopher; Blaxton, Teresa. "Data Mining Solutions. Methods and Tools for Solving Real-World Problems". John Wiley & Sons. USA, (1998).

[WID60] Wildrow, B., Hoff, M. E. "Adaptive switching circuits", in 1960 IRE WESCON Convention Record, New York, 96-104.(1960).

[WIT00] Witten, Ian H.; Frank, Eibe. "Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations". Morgan Kaufmann Publishers. San Francisco, California (2000).

[XMD02] Dirección Web: http://davis.wpi.edu/~xmdv/. "XMDVTOOL Home Page" (2002).

[ZAD65] Zadeh, L.A. "Fuzzy Sets. Information and Control" 338-353, (1965)

[ZEL93] Zell, A. y otros. "SNNS, Stuttgart Neural Network Simulator. User Manual Version 3.0", Report, Nº 3/93. (1993)