



Controladores MPC y PID con autoajuste para un proceso de dinámica rápida a través de MATLAB[®]/Simulink[®] y OPC

Ramón Imad Verdés Kairuz, Ana Isabel González Santos

RESUMEN / ABSTRACT

En este trabajo se presenta el diseño de un controlador predictivo basado en modelo (MPC) para el control efectivo de la velocidad de un motor de inducción con rotor tipo jaula de ardilla ante una referencia de velocidad variable en el tiempo. Para el diseño del controlador y posteriormente el control de la velocidad del motor se emplea la herramienta MATLAB[®]/Simulink[®]. Este control se realiza a través de un servidor OPC (Ole for Process Control, en inglés) el cual funciona como pasarela entre dicha herramienta y el autómata programable encargado de enviar la señal de control al variador de velocidad que maneja el motor. El bus de campo empleado para las comunicaciones entre el ordenador y el autómata es Ethernet. Para llegar a conclusiones sobre el controlador diseñado este es comparado con un controlador PID auto-ajustado en el autómata OMRON[®].

Palabras claves: Control Predictivo, OPC, MATLAB[®]/Simulink[®]

This paper shows the design of a model based predictive controller to perform a speed control of an induction motor with squirrel cage rotor with a speed reference changing through the time. To achieve this goal the software MATLAB[®]/Simulink[®] were used. The control is performed employing an OPC server which acts like a gateway between MATLAB[®]/Simulink[®] and a Programmable Logic Controller (PLC). This PLC sends the control signal to a speed driver which handles the motor. The field bus employed for the communications between the Personal Computer (PC) and the PLC was Ethernet. The designed controller is compared with an auto-tuned PID controller from the PLC OMRON[®].

Key words: Predictive Control, OPC, MATLAB[®]/Simulink[®]

Speed control of and induction motor through MATLAB[®]/Simulink[®] and an OPC server

INTRODUCCIÓN

Los motores de inducción, en particular los motores de inducción con rotor tipo jaula de ardilla, (Fig. 1) han sido ampliamente empleados en aplicaciones industriales tales como vehículos híbridos, molinos de papel y textiles, en la robótica, y en la generación de energía eólica. La diversidad de aplicaciones son consecuencia de sus ventajas inherentes tales como su construcción simple, robustez, confiabilidad, bajo costo, y pocas necesidades de mantenimiento^{1 2 3}. Su campo de aplicación va desde potencias fraccionarias hasta varios cientos de kilovatios.



Fig. 1. Motor trifásico de inducción con rotor tipo jaula de ardilla.

El motor trifásico de inducción con rotor tipo jaula de ardilla es una máquina de corriente alterna, cuyo funcionamiento no presenta grandes dificultades siempre que se desee trabajar a velocidad constante. Una gran parte de los equipos utilizados en la industria moderna como los ya mencionados deben funcionar a velocidades variables. Para alcanzar un buen desempeño de estos motores se hace necesario que esta velocidad variable sea controlada de una manera adecuada. En otros trabajos realizados^{2 4 5 6 7} enfocan el control de la velocidad de un motor de inducción empleando lógica difusa y algunos combinan técnicas de control que incluyen un modelo difuso como elemento de predicción en el control predictivo^{8 9}. El objetivo de este trabajo es el control de un servosistema con este tipo de motor.

MATERIALES Y MÉTODOS

Entre los materiales empleados en esta investigación estuvieron una maqueta con PLC OMRON[®] como muestra la Fig. 2. El autómatas programable CJ1M-CPU21 es el equipo encargado de contener la programación para ejecutar el control de la velocidad del motor. El módulo de Ethernet ETN21 permite la implementación de una red donde se incorporan la PC (Personal Computer, en inglés) y el HMI (Human Machine Interface, en inglés) o interface de operador.



Fig. 2. Autómata programable OMRON[®]: Fuente, CPU y módulos de izquierda a derecha.

La red implementada en la investigación utiliza OPC lo cual es (Object Linking and Embedding for Process Control, en inglés). Entre los materiales se utilizó la caja de herramientas OPC en el entorno interactivo MATLAB[®]. Los métodos utilizados son los propios de la investigación científica y la ingeniería. En específico hay que mencionar que se utilizó la metodología de identificación de sistemas¹⁰ para la obtención del modelo del proceso que constituye el elemento clave de la técnica de control avanzado que se implementa.

OPC Y LA CAJA DE HERRAMIENTAS OPC DE MATLAB[®]

OPC es un estándar industrial que facilita la interoperabilidad de dispositivos de control de diferentes fabricantes. Es una especificación técnica no propietaria que define un conjunto de interfaces estándar basadas en las plataformas de Microsoft[®] OLE/COM/DCOM y la tecnología punto NET. La aplicación de las interfaces estándar de OPC garantiza la interoperabilidad entre aplicaciones de control/automatización y aplicaciones de oficina/negocio con una filosofía cliente servidor¹¹. El servidor OPC es la aplicación software la cual opera como la API (Application Programming Interface, en inglés) o como el convertidor de protocolos. El servidor OPC se conecta a dispositivos como los PLC (Programmable Logic Controller, en inglés), DCS (Distributed Control Systems, en inglés), RTU (Remote Terminal Unit, en inglés) o a fuentes de datos (bases de datos o interfaces de usuario) y traduce los datos en un formato basado en un estándar de OPC.

La caja de herramientas OPC extiende las prestaciones de MATLAB® y Simulink® con herramientas para la interacción con servidores OPC. Este conjunto de funciones permite la lectura, escritura y el registro de datos OPC que provienen de dispositivos que están comprendidos en el estándar de Acceso de Datos de la fundación OPC como los ya mencionados anteriormente. Este conjunto de funciones provee tres formas de implementar un cliente de Acceso de Datos OPC, una de ellas es emplear el conjunto de bloques de la librería de Simulink® para leer y escribir datos hacia y desde el servidor OPC mientras se simula un sistema.

La caja de herramientas OPC de Simulink® ofrece un bloque de configuración para especificar los servidores OPC empleados en la simulación, para definir el comportamiento para errores y eventos OPC y para establecer el comportamiento en tiempo real. Durante la simulación, el modelo se ejecuta en pseudo-tiempo real, haciendo coincidir el reloj del sistema tan cerca como sea posible del tiempo de la simulación del proceso disminuyendo automáticamente la simulación. Los parámetros del bloque pueden ser también configurados de forma tal que la simulación se ejecute más lentamente que el reloj del sistema. La Fig. 3 muestra la ventana de configuración OPC y la ventana de configuración del cliente se puede observar en la Fig. 4.

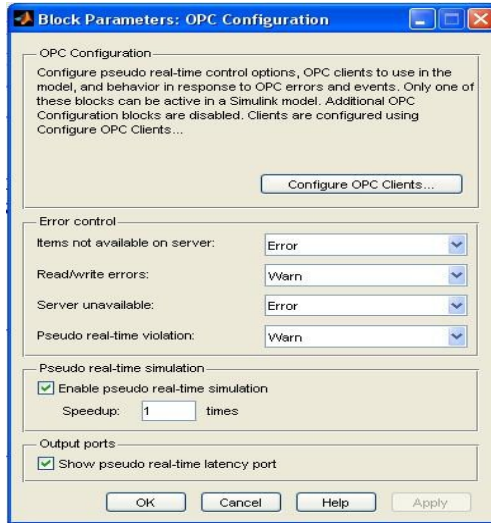


Fig.3. Configuración de OPC.

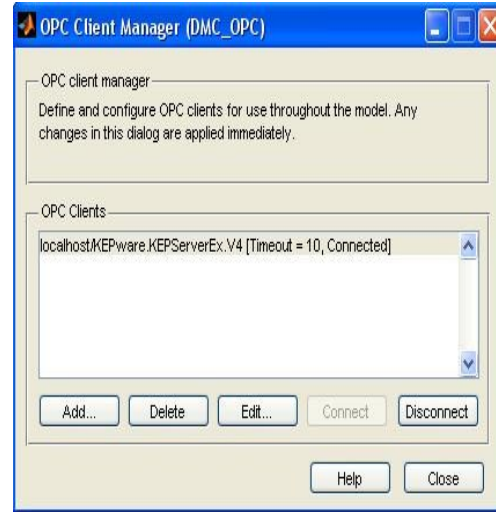


Fig. 4. Administrador de cliente OPC.

En la ventana de configuración de OPC, se puede habilitar la simulación en pseudo-tiempo real. Cuando está marcado “*Enable pseudo real-time simulation*” (como aparece en la Fig. 3), el tiempo de ejecución del modelo se iguala al reloj del sistema tanto como sea posible. El ajuste “*Speedup*” determina cuantas veces más rápido correrá la simulación con respecto al reloj del sistema. Por ejemplo si “*Speedup*” es ajustado a dos, esto significa que una simulación de diez segundos tomará solo cinco segundos en ejecutarse.

Es necesario destacar que los ajustes de control de tiempo real no garantizan un comportamiento de tiempo real. Si el modelo se ejecuta a un tiempo más lento que el tiempo real, ocurre un error de violación en la latencia de pseudo-tiempo real. Cuando se marca “*Show pseudo real time latency port*” se habilita la salida de la latencia del modelo, la cual aparece expresada en segundos como una salida del bloque de configuración de OPC. La latencia de pseudo-tiempo real es el tiempo empleado esperando por el reloj del sistema en cada paso de simulación. Si este valor es negativo, la simulación se ejecuta más lentamente que el tiempo real.

Una vez que ha sido creado un objeto grupo que contiene elementos individuales, se puede realizar la lectura desde o la escritura hacia elementos individuales o a todos los elementos en un grupo simultáneamente. En MATLAB®, las operaciones de lectura y escritura pueden ocurrir lo mismo sincrónica (la ejecución de MATLAB® se bloquea hasta que se completa la operación) que asincrónicamente (MATLAB® se puede seguir ejecutando mientras la operación está en progreso).

En la herramienta Simulink® de MATLAB®, los bloques contienen un administrador de clientes que hace posible especificar y administrar los servidores OPC, seleccionar elementos, y definir el tiempo de muestreo de los bloques. El bloque de lectura OPC mostrado en la Fig. 5 habilita la selección de elementos del servidor OPC y la lectura de datos en línea de la planta directamente hacia el modelo en Simulink®. El bloque de escritura OPC (Fig. 6) habilita la selección de elementos del modelo en Simulink® y permite escribir directamente datos de la planta a través del servidor OPC.

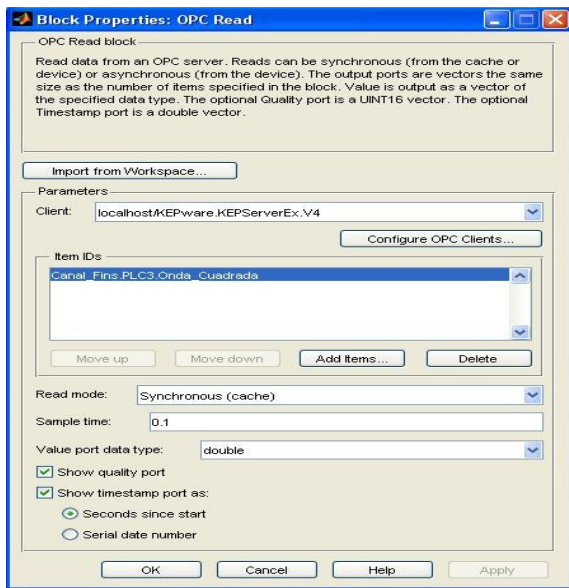


Fig. 5. Configuración del bloque de lectura OPC.

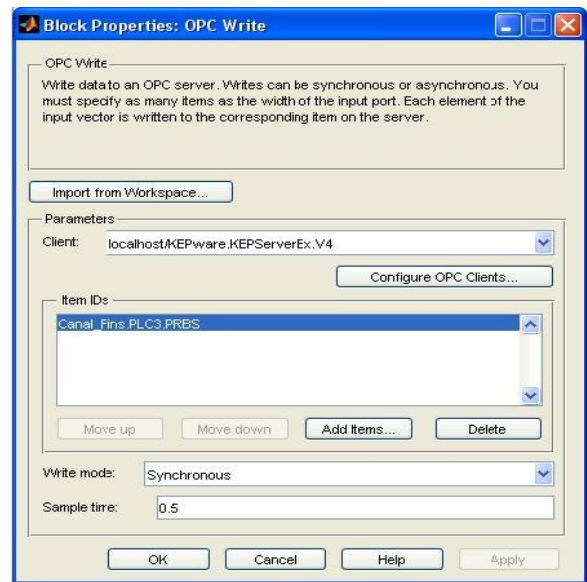


Fig. 6. Configuración del bloque de escritura OPC.

A la hora de implementar experimentos en tiempo real en Simulink® a través de la comunicación OPC puede ocurrir que el tiempo real de ejecución sea mayor que el tiempo de simulación establecido en la configuración de los parámetros Simulink®. En estos casos la latencia a cada paso es negativa. La latencia representa el margen dentro del tiempo de muestreo; una mayor latencia representa un mayor margen, por ejemplo, el tiempo de muestreo puede ser disminuido. Existen diversos factores que permiten tener un tiempo de ejecución cercano al valor de tiempo establecido en el esquema Simulink®. Un estudio detallado de este tema se puede encontrar en ¹². El empleo de OPC como pasarela dentro del ensayo de controladores aplicados a procesos reales es una tendencia actual y así lo evidencian los trabajos ^{13 14 15 16 17 18 19 20}.

MPC

El controlador predictivo basado en modelo (MPC en inglés) ^{21 22 23 24} basa su funcionamiento en una estrategia llamada horizonte deslizante, donde a cada instante de tiempo son calculadas las salidas futuras del proceso $\hat{y}(t+k|k)$ dentro de un horizonte de predicción finito N . Para esto se emplea un modelo matemático que describe la dinámica del proceso. En una segunda etapa la señal de control futura $u(t+k|k)$ es calculada optimizando cierto criterio en aras de mantener tanto como sea posible al proceso dentro de una trayectoria de referencia deseada, como aparece en la Fig. 7.

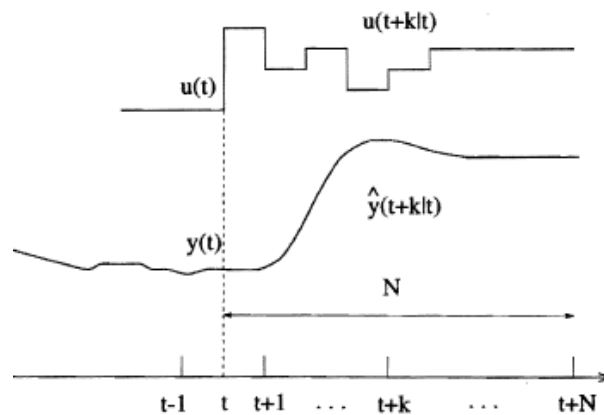


Fig. 7. Estrategia de control MPC.

La estrategia de control MPC empleada es la desarrollada dentro de la caja de herramientas MATLAB® R2013a la cual se define como establece ²⁵. El diagrama en bloques de la estructura del controlador MPC es la mostrada en la Fig. 8.

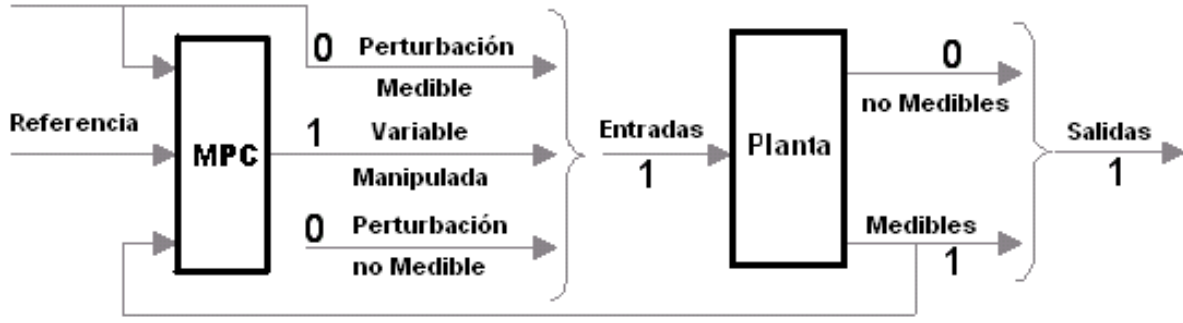


Fig. 8. Diagrama en bloques de la estructura del controlador predictivo basado en modelo.

En el esquema del controlador predictivo, el modelo de predicción del comportamiento futuro de la planta tiene como entradas las variables manipuladas, las perturbaciones medibles, y las perturbaciones no medibles. Las perturbaciones no medibles emplean un modelo que toma como entrada señales tipo ruido blanco.

El modelo de predicción del comportamiento de la planta asume el sistema como lineal, invariante en el tiempo y está descrito por las ecuaciones (1) donde $x(k)$ es el vector de estados n_x -dimensional de la planta, $u(k)$ es el vector n_u -dimensional de las variables manipuladas (MV), $v(k)$ es el vector n_v -dimensional de las perturbaciones medibles (MD), $d(k)$ es el vector n_d -dimensional de las perturbaciones no medibles (UD), $y_m(k)$ es el vector de las salidas medibles (MO), $y_u(k)$ es el vector de las salidas no medibles (UO) y $y(k)$ es el vector completo n_y -dimensional de todas las salidas medibles ($y_m(k)$) y no medibles ($y_u(k)$).

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \quad (1)$$

$$y_m(k) = C_m x(k) + D_{vm} v(k) + D_{dm} d(k)$$

$$y_u(k) = C_u x(k) + D_{vu} v(k) + D_{du} d(k) + D_{uu} u(k)$$

La perturbación no medible $d(k)$ es modelada como la salida del sistema lineal e invariante en el tiempo (2) y (3).

$$x_d(k+1) = \bar{A}x_d(k) + \bar{B}n_d(k) \quad (2)$$

$$d(k) = \bar{C}x_d(k) + \bar{D}n_d(k) \quad (3)$$

El sistema descrito mediante las ecuaciones (2) y (3) es afectado por el ruido aleatorio Gaussiano $n_d(k)$, que tiene media cero y matriz de covarianza unitaria. Otros modelos son empleados también dentro de la formalización de la técnica de control predictivo como pueden ser los modelos neuronales^{26 27} y los modelos difusos²⁸.

Asumiendo que los estimados de $x(k)$, $x_d(k)$ están disponibles en el instante k . La acción de control del controlador predictivo en el instante k se obtiene a partir de la solución de un problema de optimización que se plantea de la forma (4).

$$\Delta u(k|k), \dots, \Delta u(m-1+k|k), \varepsilon \left\{ \sum_{i=0}^{p-1} \left(\sum_{j=1}^{n_y} \omega_{i+1,j}^y (y_j(k+i+1|k) - r_j(k+i+1)) \right)^2 \right. \\ \left. + \sum_{j=1}^{n_u} \left| \omega_{i,j}^{\Delta u} \Delta u_j(k+i|k) \right|^2 \right. \\ \left. + \sum_{j=1}^{n_u} \left| \omega_{i,j}^u (u_j(k+i|k) - u_{j,target}(k+i)) \right|^2 + \rho_\varepsilon \varepsilon^2 \right\} \quad (4)$$

donde el subíndice “ j ” denota el j -ésimo componente de un vector, “ $(k+i|k)$ ” denota el valor predicho para el tiempo $k+i$ con la información disponible hasta k ; $r(k)$ es el valor actual de la referencia para la salida, donde la señal de control, sus incrementos y la salida están sujetas a las restricciones (5) con respecto a la secuencia de incrementos de la entrada $\{\Delta u(k|k), \dots, \Delta u(m-1+k|k)\}$ y a la variable de relajación ε , y se establece además $u(k) = u(k-1) + \Delta u(k|k)$, donde $\Delta u(k|k)$ es el primer elemento de la secuencia óptima.

$$\begin{aligned} u_{jmin}(i) - \varepsilon V_{jmin}^u(i) &\leq u_j(k+i|k) \leq u_{jmax}(i) + \varepsilon V_{jmax}^u(i) \\ \Delta u_{jmin}(i) - \varepsilon V_{jmin}^{\Delta u}(i) &\leq \Delta u_j(k+i|k) \leq \Delta u_{jmax}(i) + \varepsilon V_{jmax}^{\Delta u}(i) \end{aligned} \quad (5)$$

$i = 0, \dots, p-1$

$$\begin{aligned} y_{jmin}(i) - \varepsilon V_{jmin}^y(i) &\leq y_j(k+i+1|k) \leq y_{jmax}(i) + \varepsilon V_{jmax}^y(i) \\ \Delta u(k+h|k) &= 0, h = m, \dots, p-1 \end{aligned}$$

$$\varepsilon \geq 0$$

$\omega_{i,j}^{\Delta u}, \omega_{i,j}^u, \omega_{i,j}^y$, son pesos no negativos para las correspondientes variables. Mientras más pequeño sea ω , menos importancia tiene esa variable en el comportamiento del índice de performance del sistema. $u_{jmin}, u_{jmax}, \Delta u_{jmin}, \Delta u_{jmax}, y_{jmin}, y_{jmax}$ son las barreras superior/inferior de las correspondientes variables. En la ecuación (5), las restricciones a "u", "Δu" y "y" son relajadas introduciendo la variable de relajación $\varepsilon \geq 0$. El peso ρ_ε en la variable de relajación ε penaliza la violación de las restricciones. Mientras más grande ρ_ε con respecto a los pesos de las entradas y las salidas, más es penalizada la violación de las restricciones. Los vectores ECR (Equal Concern for the Relaxation, en inglés) $V_{min}^u, V_{max}^u, V_{min}^{\Delta u}, V_{max}^{\Delta u}, V_{min}^y, V_{max}^y$ tienen valores no negativos los cuales representan el interés por el relajamiento de la correspondiente restricción; mientras más grande V, más "suave" es la restricción. $V=0$ significa que la restricción es "dura" y que no puede ser violada. Por defecto todas las restricciones para las entradas son "duras" ($V_{min}^u = V_{max}^u = V_{min}^{\Delta u} = V_{max}^{\Delta u} = 0$) y todas las restricciones para las salidas son "suaves" ($V_{min}^y = V_{max}^y = 1$). Debido a que las restricciones "duras" de las salidas pueden causar que el problema de optimización no sea apropiado (por ejemplo, debido a perturbaciones no predichas, discrepancias del modelo, o simplemente debido a redondeo numérico), se produce un mensaje de alerta si V_{min}^y, V_{max}^y son menores que un valor pequeño dado y automáticamente se ajustan a ese valor. Por defecto se cumple (6).

$$\rho_\varepsilon = 10^5 \max\{\omega_{i,j}^{\Delta u}, \omega_{i,j}^u, \omega_{i,j}^y\} \quad (6)$$

El vector $u_{jtarget}(k+i)$ es una referencia para el vector de entradas. Se emplea típicamente si el número de entradas es mayor que el número de salidas, como una especie de referencia de menor prioridad.

El algoritmo implementado en la caja de herramientas de control predictivo basado en modelo emplea diferentes procedimientos dependiendo de la presencia de restricciones o no. Si todas las barreras son infinitas, entonces la variable de relajación ε se elimina, y el problema de optimización de las ecuaciones (4) y (6) tiene solución analítica. De lo contrario un método de solución de programación cuadrática (QP, en inglés) es empleado.

Debido a que las restricciones para las salidas son siempre "suaves", el problema QP nunca deja de tener solución. Si por razones numéricas el problema QP se vuelve inapropiado, la segunda muestra de la secuencia óptima anterior es aplicada, es decir, se cumple (7).

$$u(k) = u(k-1) + \Delta u(k|k-1) \quad (7)$$

Los parámetros definidos para el controlador diseñado son mostrados en la Tabla 1.

Tabla 1. Parámetros del objeto MPC en MATLAB®/Simulink®.

| Parámetro | Valor |
|--|---|
| Tiempo de muestreo (s) | 0.1 |
| Horizonte de predicción | 10 |
| Horizonte de control | 5-5 (El valor óptimo de la señal de control se calcula al principio y se mantiene hasta la mitad del horizonte de predicción en donde se calcula el segundo valor óptimo y se mantiene hasta el final del horizonte de predicción.) |
| Modelo del proceso | $G_M(s) = \frac{60}{0.0045s^2 + 0.1034s + 1}$ |
| Peso asociado a la variable manipulada | 0 |
| Peso asociado a la razón de cambio de la variable manipulada | 70 |
| Peso asociado a la variable de salida | 20 |

Restricciones:

$$0 \leq \text{Frecuencia(Hz)} \leq 50$$

$$-0.2 \leq \text{Razón de cambio de la Frecuencia} \leq 0.2$$

$$0 \leq \text{Velocidad} \leq 2980$$

Las restricciones expresadas en la ecuación (5) quedan para este caso de la siguiente manera:

$$0 - 0.75 * 0 \leq y_j(k+i | k) \leq 2980 + 0 * 0.75 * 0$$

$$0 - 0.75 * 0 \leq u_j(k+i | k) \leq 50 + 0.75 * 0$$

$$0.2 - 0.75 * 0.1 \leq \Delta u(k+i | k) \leq 0.2 + 0.75 * 0.1$$

$$i=0, \dots, 9$$

$$\varepsilon = 0.75$$

$$\Delta u(k+h | k) = 0, h = 2, \dots, 9$$

El modelo del proceso que muestra la Tabla 1, es un modelo en función de transferencia de segundo orden que fue obtenido y validado empleando la metodología de Lennart Ljung¹⁰ con la peculiaridad de que los estímulos utilizados en los experimentos fueron señales pseudo-aleatorias multinivel de tipos MLB (Multi Level Binay, en inglés) y QRB (Quadratic Residue Binary, en inglés)²⁹ en el rango de trabajo del motor. Se hicieron 10 réplicas de una longitud de 31 elementos entre 42 y 50 Hz para un rango de velocidades entre 2500 y 2980 rpm. Los juegos de datos obtenidos fueron pre-tratados y compartidos para la estimación de los parámetros del modelo y la validación del mismo³⁰.

RESULTADOS

En esta sección se realiza un estudio del control de la velocidad de un motor de inducción con rotor tipo jaula de ardilla mediante la ley de control predictivo MPC que se ejecuta desde MATLAB® en la PC (Personal Computer, en inglés) y a través de OPC-PLC se envía la señal de control al variador de velocidad el cual es el encargado de variar la relación tensión/frecuencia del motor y con esto la velocidad. El diagrama tecnológico de la maqueta empleada para las simulaciones se puede apreciar en la Fig. 9.

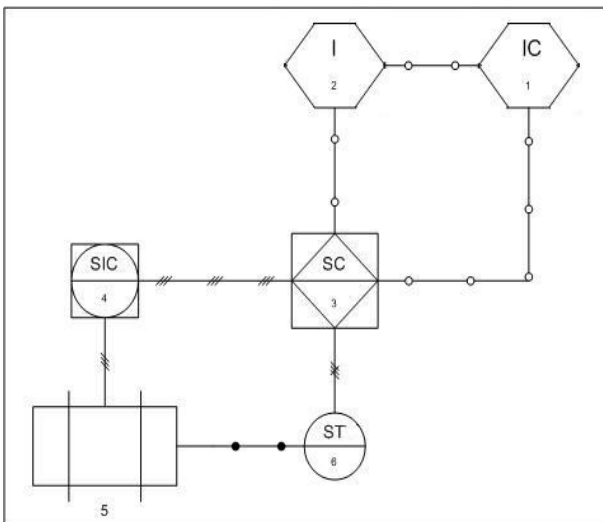


Fig. 9. PC-HMI-PLC-Variador-Motor

- 1-PC (Computadora de escritorio)
- 2-HMI (Interface hombre-máquina del fabricante OMRON® NS- NS8-TV01-V2)
- 3-PLC (Controlador Lógico Programable del fabricante OMRON® CJ1M CPU21)
- 4-Variador de velocidad (fabricante OMRON®, modelo VZAB0P2BAA)
- 5-Motor de inducción con rotor de ardilla (fabricante Kelvin®, modelo K90T2)
- 6-Encoder incremental (fabricante OMRON®, modelo E6B2-CWZ6C)

Simulación del control MPC en MATLAB®/Simulink®

Todas las simulaciones desarrolladas tienen una duración de 300 segundos, el diagrama en bloques del esquema de simulación desarrollado en Simulink® se muestra en la Fig. 10. En la Fig. 11 se muestra el comportamiento de la velocidad ante una referencia que varía desde 2500 r.p.m hasta 2800 r.p.m empleando el bloque MPC que brinda Simulink® y el modelo del proceso.

Para hacer más cercana la simulación a la realidad se adiciona un ruido con valores comprendidos entre las +/- 7 r.p.m. La señal de control para este caso tiene la forma que se aprecia en la Fig. 12.

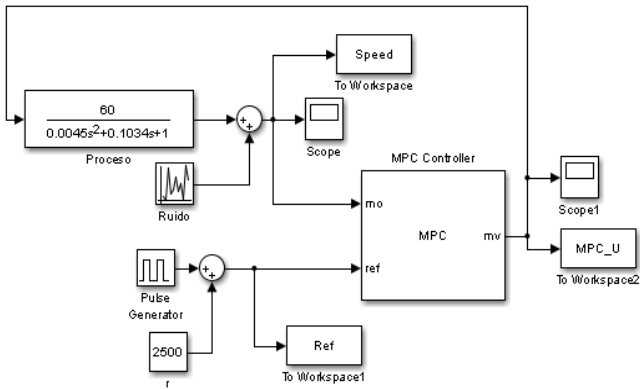


Fig. 10. Diagrama en bloques del controlador MPC simulado en MATLAB®/Simulink®.

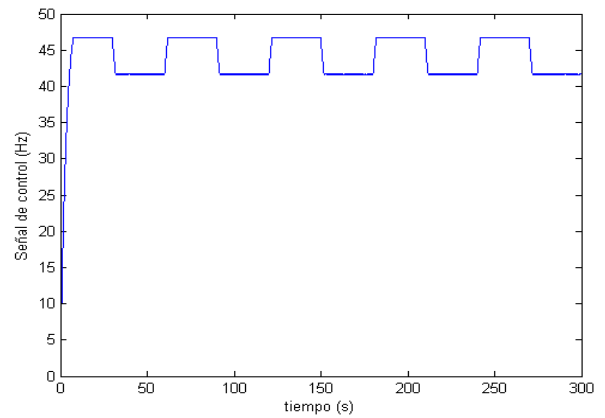
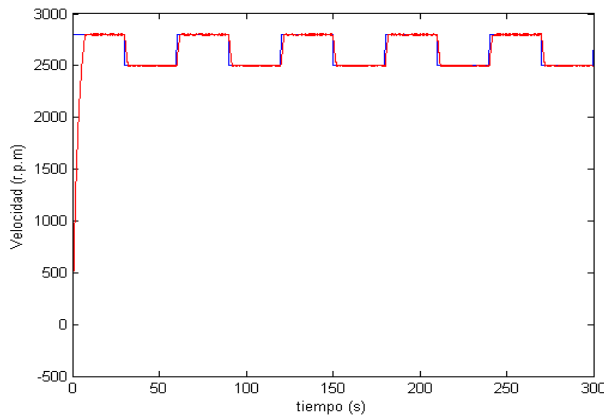


Fig. 11. Comportamiento de la velocidad mediante simulación en MATLAB®.

Fig. 12. Señal de control para simulación en MATLAB®.

Control MPC en tiempo real de la velocidad desde MATLAB®/Simulink®

La ley de control es calculada por el controlador predictivo MPC desde MATLAB®/Simulink® (Fig. 13) y mediante la pasarela OPC esta es escrita en la salida analógica del autómata programable conectada al variador de velocidad que controla al motor. La latencia durante este experimento se aprecia en la Fig. 14. En esta figura se corrobora la condición de tiempo real durante los experimentos.

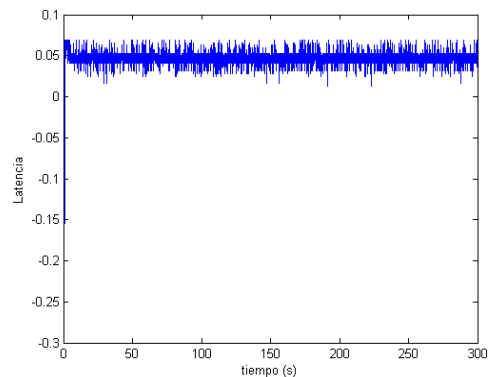
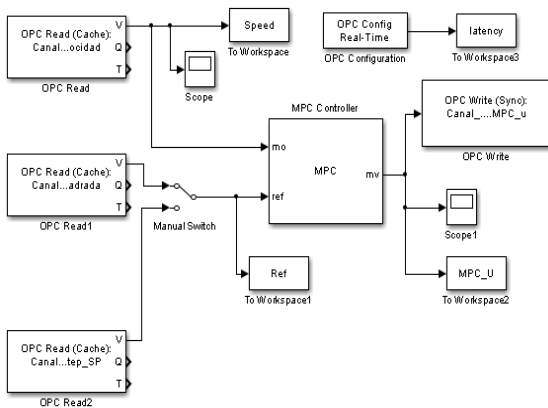


Fig. 13. Diagrama en bloques del sistema de control de la velocidad del motor desde MATLAB®/Simulink®.

Fig. 14. Latencia de los datos durante los experimentos.

En la Fig. 15 se muestra el comportamiento de la velocidad siguiendo una referencia que varía desde 2500 r.p.m hasta 2800 r.p.m. La señal de control óptima para este caso tiene la forma que se aprecia en la Fig. 16.

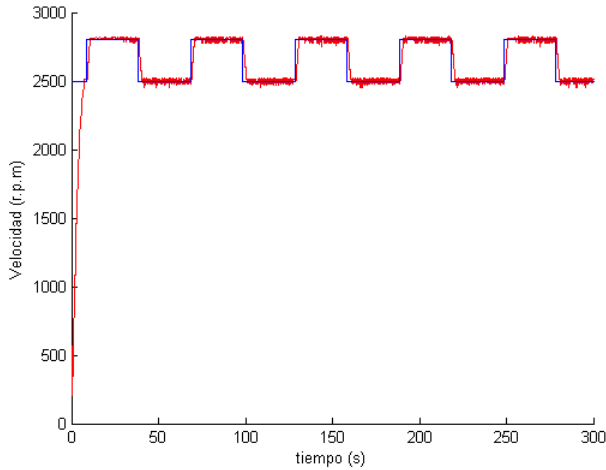


Fig. 15. Comportamiento de la velocidad ante una referencia variable.

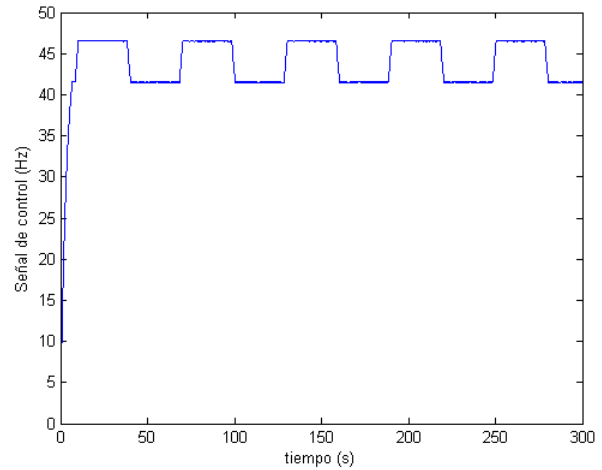


Fig. 16. Señal de control durante el seguimiento de una referencia variable.

Control de la velocidad mediante PID con autoajuste en autómatas programables

Para el caso del controlador PID, este es autoajustado en un experimento preliminar ante una señal paso escalón de amplitud 2500 r.p.m. Una vez el método de oscilaciones de ciclo límite implementado por este autómata determina y actualiza las nuevas constantes P, I y D del controlador, se cambia la referencia a la mostrada en la Fig. 17. La señal de control calculada por el controlador, ante esta referencia de velocidad, para el motor de inducción con rotor tipo jaula de ardilla se muestra en la Fig. 18.

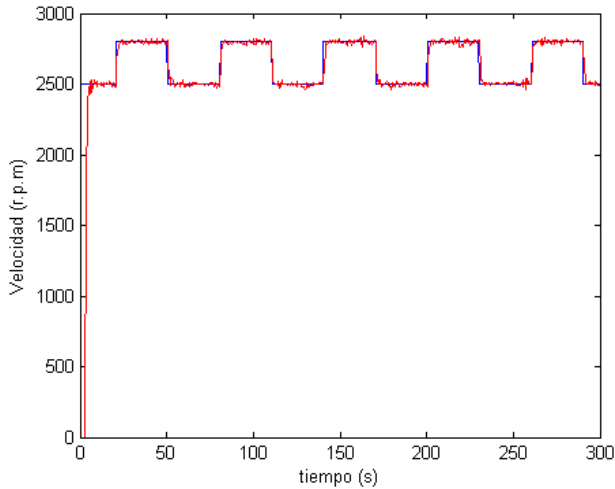


Fig. 17. Variación de la velocidad ante una referencia variable.

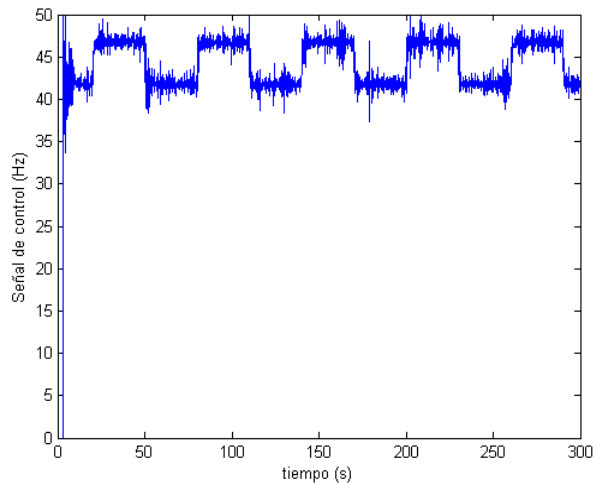


Fig. 18. Señal de control del PID auto-ajustado ante una referencia variable.

El experimento para el análisis del comportamiento del proceso bajo perturbaciones se desarrolla aplicando un freno manual al eje del motor como muestran las Fig. 19, Fig. 20, Fig. 21 y Fig. 22 para el caso del controlador MPC y las Fig. 23, Fig. 24, Fig. 25 y Fig. 26 para el caso del controlador PID auto-ajustado. Lo que se analiza en este caso es la capacidad del controlador de rechazar a estas perturbaciones y devolver al proceso a su condición de estado estacionario.

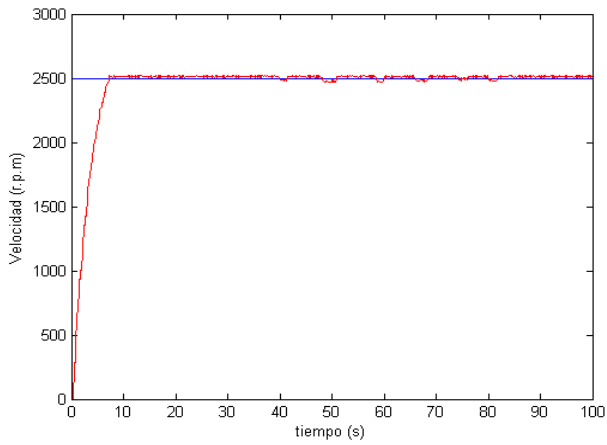


Fig. 19. Proceso perturbado aleatoriamente con controlador MPC.

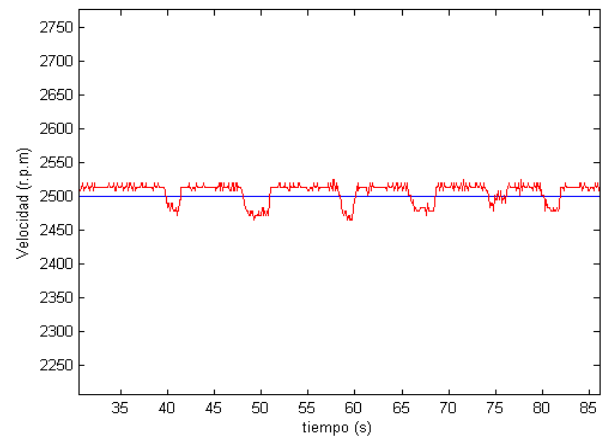


Fig. 20. Ampliación de la Fig. 19.

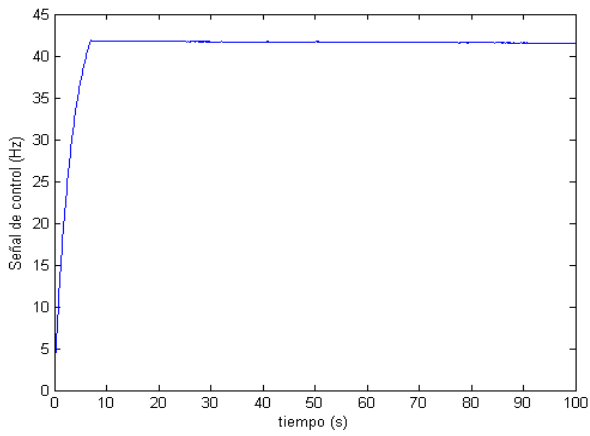


Fig. 21. Señal de control del MPC durante la perturbación.

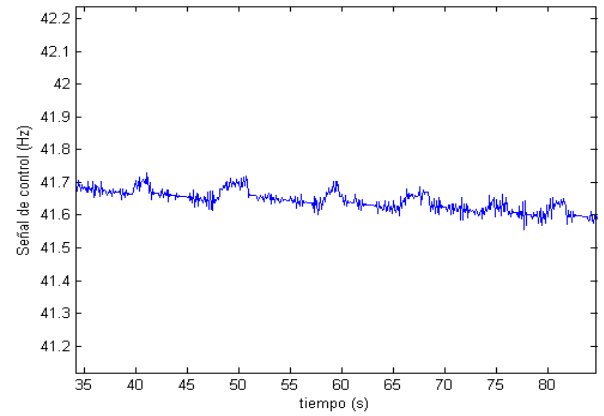


Fig. 22. Ampliación de la Fig. 21.

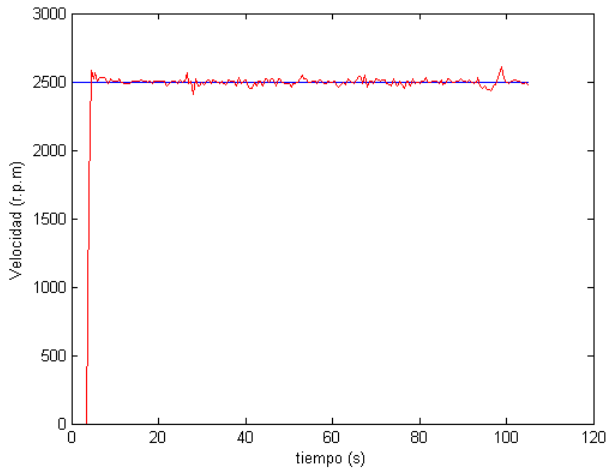


Fig. 23. Variación de la velocidad ante perturbaciones con control PID auto-ajustado.

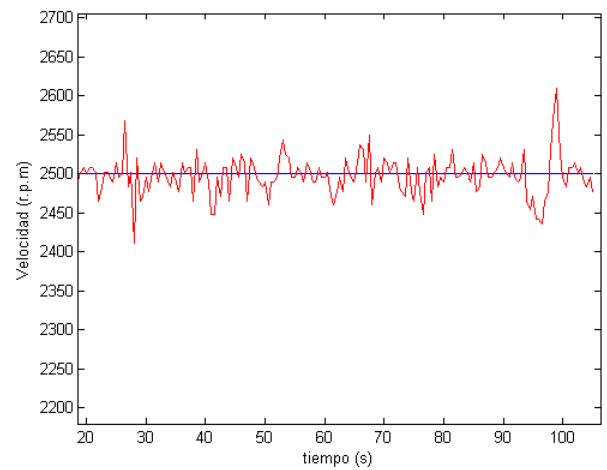


Fig. 24. Ampliación de la Fig. 23.

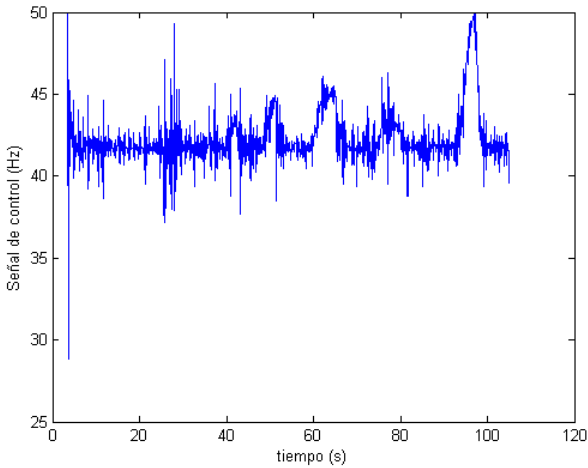


Fig. 25. Señal de control del PID auto-ajustado durante la perturbación.

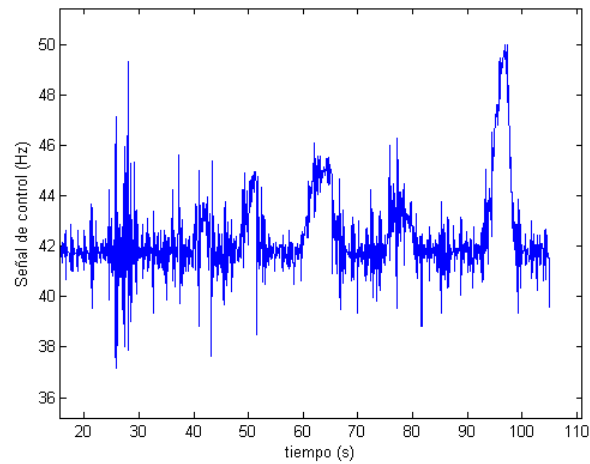


Fig. 26. Ampliación de la Fig. 25.

ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

Como parte de la investigación se realizó el siguiente experimento sometiendo al proceso real a tres valores constantes, durante un periodo de tiempo determinado, a la salida analógica que da la referencia al variador de velocidad con el objetivo de analizar la medición de velocidad a través del encoder incremental. Los resultados se muestran en la Fig. 27.

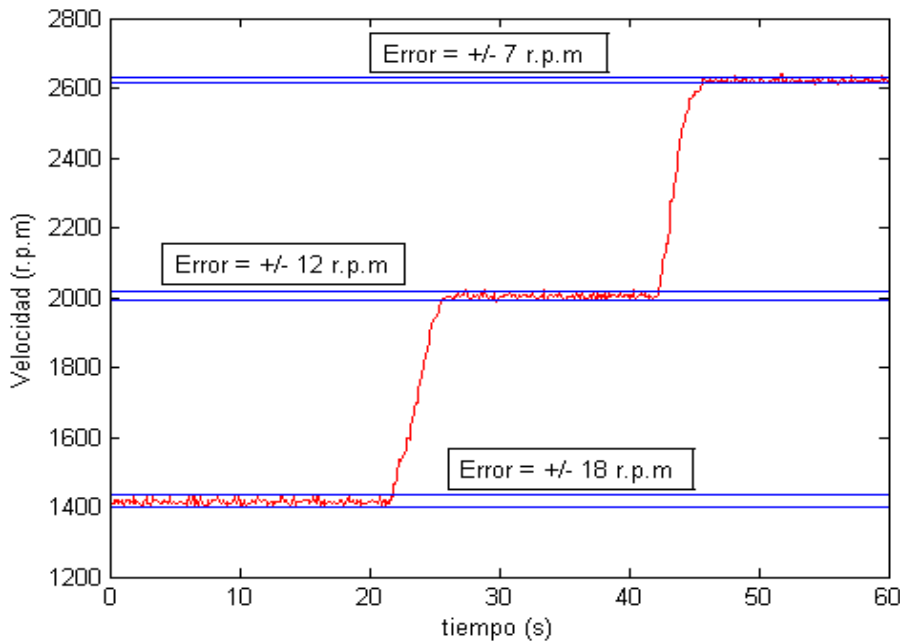


Fig. 27. Análisis de ruidos inherentes al canal de medición.

Como se puede apreciar en la Fig. 27 en los tres intervalos en los que se mantuvo constante el valor de la salida analógica (5, 7 y 9 V), la medición de la velocidad presenta un comportamiento oscilatorio dentro de una banda específica para cada caso. Estas oscilaciones tienden a disminuir a medida que la velocidad crece en valor absoluto. Sobre el intervalo de interés alrededor de las 2600 r.p.m estas oscilaciones tienen un valor de +/- 7 r.p.m aproximadamente.

Los valores dentro de los cuales oscila la velocidad para el seguimiento de una trayectoria variable en el caso del controlador MPC no supera las +/- 20 r.p.m (Fig. 28). Para el caso del controlador PID con auto-ajuste el rango dentro del cual oscilan los valores de velocidad para una referencia fija en 2800 r.p.m no superan las +/- 23 r.p.m y para el caso de la referencia fija en 2500 r.p.m no superan las +/- 30 r.p.m como se aprecia en la Fig. 30.

Como se vio la señal de medición de la velocidad proveniente del encoder incremental no está exenta de ruidos de baja intensidad, no es una señal plana, a pesar de que la señal de control se mantiene estable en un valor constante para cada caso analizado. Esto ciertamente tiene sus repercusiones en el lazo de control y se puede analizar como una perturbación no medible con comportamiento aleatorio que se adiciona a la variable medida del proceso.

Estas perturbaciones no medibles tienen gran repercusión en el controlador PID con autoajuste y así lo evidencia el comportamiento de la señal de control (Fig. 26), la cual presenta oscilaciones más marcadas que en el caso del algoritmo de control predictivo MPC. Para este último un papel importante lo juegan las restricciones para el cambio y la razón de cambio impuestas para la señal de control, una ventaja sin dudas de este controlador sobre el PID, el cual no admite restricciones ni restricciones en la razón de cambio para la señal de control. Otra desventaja de la naturaleza del controlador PID comparado con el controlador MPC es que al no tener restricciones para la señal de control como ya se mencionó, este alcanza la saturación ante cambios bruscos en la referencia como se aprecia en la Fig. 31 en los primeros instantes de tiempo. En este caso donde ocurre la saturación se puede entender que el proceso queda como si estuviera a lazo abierto ³¹, ya que, cualquier acción de control sobre el elemento de acción final no tendrá ningún efecto sobre el proceso dado que este está saturado y esto sin dudas es una gran desventaja con respecto a los resultados mostrados por el controlador MPC el cual presenta cambios suaves sin grandes oscilaciones (Fig. 29) en la señal de control.

A pesar de todo lo analizado hay que destacar que tanto el controlador MPC implementado desde MATLAB®/Simulink® como el algoritmo de control PID con auto-ajuste logran ciertamente el control de la velocidad y el seguimiento de la trayectoria establecida con oscilaciones alrededor del punto de consigna.

Por otra parte el controlador MPC logra devolver al sistema a su estado estacionario ante la influencia de perturbaciones externas de magnitud desconocida como se puede apreciar en las Fig. 19 y Fig. 21 sin grandes esfuerzos en la señal de control, esto se debe a la estrategia de ventana deslizante que implementa este controlador, otra de sus grandes ventajas con respecto al algoritmo de control tradicional PID.

Como se aprecia en la Fig. 29 existe una variación absoluta de 0.1 Hz en la señal de control lo que equivale a 6 r.p.m aproximadamente en valor de velocidad. Evidentemente existe un error de +/- 7 r.p.m que queda fuera de discusión y que se considera error de estado estacionario. Los parámetros de la respuesta temporal para el controlador MPC y el PID con auto-ajuste se pueden apreciar en la Tabla 2.

Tabla 2. Parámetros de la respuesta temporal.

| Controlador | Pico Máximo | Tiempo de subida | Tiempo de establecimiento |
|-------------|-------------|------------------|---------------------------|
| MPC | 0.6 % | 4.98 s | 5.92 s |
| PID | 23.3% | 0.7 s | 1.1 s |

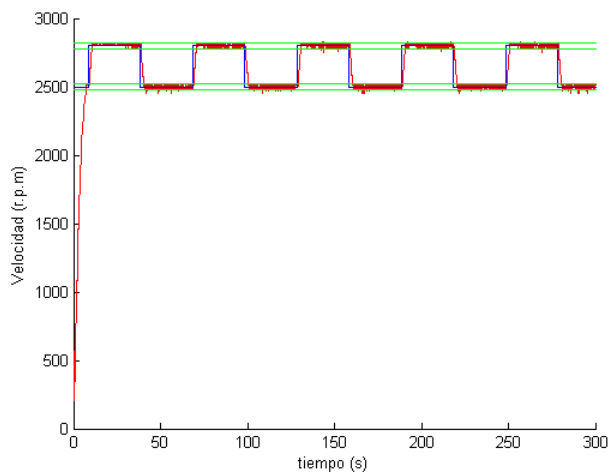


Fig. 28. Análisis del comportamiento de la velocidad ante una referencia variable para el controlador MPC.

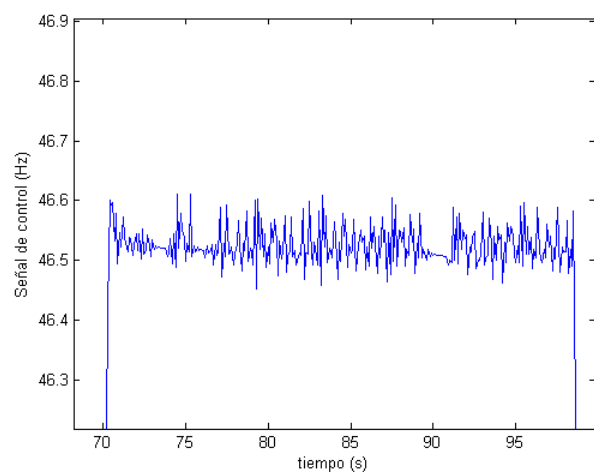


Fig. 29. Ampliación de la señal de control de la Fig. 16.

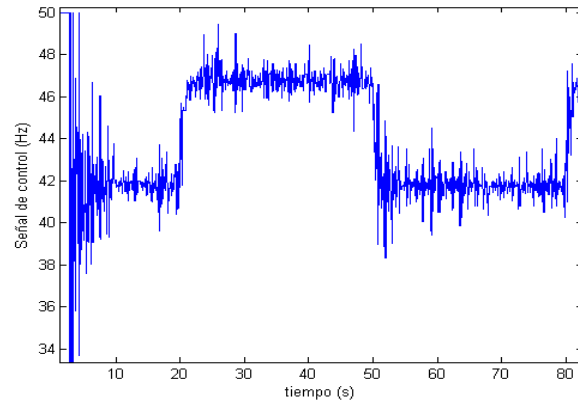
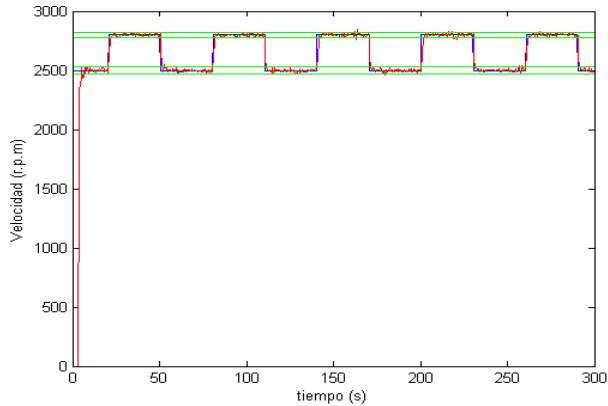


Fig. 30. Análisis del comportamiento de la velocidad ante una referencia variable para el controlador PID con auto-ajuste.

Fig. 31. Ampliación de la señal de control de la Fig. 18.

CONCLUSIONES

En primer lugar los resultados demuestran la posibilidad del control de la velocidad del motor de inducción con rotor tipo jaula de ardilla empleando MATLAB®/Simulink® y una pasarela OPC para la comunicación entre el proceso real y el ordenador donde están soportadas todas las aplicaciones. Esto evidencia ciertamente las potencialidades del empleo de OPC y las posibilidades que este brinda de realizar simulaciones de diferentes técnicas de control sobre procesos reales. Sin embargo no se debe perder de vista la posibilidad de que las simulaciones pierdan la propiedad de tiempo real, esto constituye sin dudas una limitación para los tiempos de muestreo mínimos permisibles por la caja de herramientas OPC de MATLAB®, los cuales en muchos casos están supeditados a las prestaciones del ordenador donde se llevan a cabo los experimentos.

En segundo lugar se evidencian las superioridades del controlador MPC sobre el controlador PID, las cuales están arraigadas a la propia concepción y formalización matemática de la técnica de control en cuestión. Es evidente que el controlador predictivo MPC resuelve en tiempo real un problema de optimización de un funcional y en este caso con restricciones, el cual, le permite a este controlador mejores desempeños. También se debe mencionar que la cantidad de parámetros que intervienen en el ajuste del controlador MPC son mayores que los que intervienen en el controlador PID por lo cual le siguen brindando al controlador PID su carácter atractivo e intuitivo y si a esto añadimos la capacidad de auto-ajuste que además tienen estos últimos en muchos de los autómatas programables que se encuentran en el mercado, lo vuelven, como ha quedado demostrado hasta la fecha, el algoritmo de control más difundido en las industrias a nivel mundial.

Por último, el empleo sin embargo de uno u otro algoritmo está supeditado al proceso en particular de que se trate, ya que, como se puede apreciar en la Tabla 2 el controlador MPC no presenta sobre-oscilaciones pero demora más en alcanzar el estado estacionario, sin embargo, el controlador PID responde más rápido pero a costa de una sobre-oscilación considerable.

REFERENCIAS

1. Kusagur, A., Fakirappa Kodad, S. y Ram, S. Modelling & Simulation of an ANFIS controller for an AC drive. *World Journal of Modelling and Simulation*, 2012, 8(1), 36-49.
2. Abdul Mannan, M., Islam, A., Nasir Uddin, M., Kamrul Hassan, M., *et al.* Fuzzy-Logic Based Speed Control of Induction Motor Considering Core Loss into Account. *Intelligent Control and Automation* 2012, vol. 3, pp. 229-235. Disponible en Internet: <<http://dx.doi.org/10.4236/ica.2012.33026>>.
3. Prasanna Mark, A., Vairamani, R. y Raj Irudayaraj, G. C. Mathematical Modeling and Analysis of Different Vector Controlled CSI Fed 3-Phase Induction Motor Drive. *Journal of Applied Mathematics* 2014, pp. 1-13. Disponible en Internet: <<http://dx.doi.org/10.1155/2014/623982>>.
4. Abdelkarim, E., Ahmed, M., Orabi, M. y Mutschler, P. Fuzzy Logic Speed Controller of 3-Phase Induction Motors for Efficiency Improvement. *Journal of Power Electronics* 2012, vol. 12, no. 2, pp. 305-316. Disponible en Internet: <<http://dx.doi.org/10.6113/JPE.2012.12.2.305>>.
5. Devi, K., Gautam, S. y Nagaria, D. Speed Control of 3-Phase Induction Motor Using Self-Tuning Fuzzy PID Controller and Conventional PID Controller. *International Journal of Information & Computation Technology*, 2014, 4(12), 1185-1193.
6. Kusagur, A., Kodad, S. F. y Sankar Ram, B. V. Modeling, Design & Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Speed Control of Induction Motor. *International Journal of Computer Applications*, 2010, 6(12), 29-44.
7. Tripura, P. y Kishore Babu, Y. S. Fuzzy Logic Speed Control of Three Phase Induction Motor Drive. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 2011, vol. 5, no. 12, pp. 1769-1773. Disponible en Internet: <<http://scholar.waset.org/1999.5/10485>>.

8. Thomsen, S., Hoffmann, N. y Wilhelm Fuchs, F. PI Control, PI-Based State Space Control, and Model-Based Predictive Control for Drive Systems With Elastically Coupled Loads—A Comparative Study. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 2011, vol. 58, no. 8, pp. 3647-3657. Disponible en Internet: <<http://ieeexplore.ieee.org/>>.
9. Guzman, H., Duran, M. J., Barrero, F., Bogado, B., et al. Speed Control of Five-Phase Induction Motors With Integrated Open-Phase Fault Operation Using Model-Based Predictive Current Control Techniques. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 2014, vol. 61, no. 9, pp. 4474-4484. Disponible en Internet: <<http://ieeexplore.ieee.org/>>.
10. Ljung, L. *System Identification. Theory for the user*. 2da ed. New Jersey: Prentice-Hall. Inc., 1999. 609 p. ISBN 0-13-656695-2.
11. Zamarreño, J. M. *Acceso a datos mediante OPC*. Santiago de Compostela, España: Andavira Editora, S.L., 2010. 82 p. ISBN 978-84-8408-565-2.
12. Mrosko, M. y Miklovičová, E. Real-time implementation of predictive control using programmable logic controllers. *INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT*, 2012, 6(1), 106-113.
13. Zamarreño, J. M., Fuente, M. J. y Acebes, L. F. Desarrollo de un controlador PID accesible como servidor OPC. En *Actas de XXXVI Jornadas de Automática*, Bilbao, 2015, Comité Español de Automática de la IFAC (CEA-IFAC) ed., p. 997-1000.
14. Neila, O., Roshany-Yamchi, S., Gnyszka, B. y Limón, D. Control Predictivo Generalizado en un PLC Siemens Simatic S7-1200. En *Actas de XXXVI Jornadas de Automática*, Bilbao, 2015 Comité Español de Automática de la IFAC (CEA-IFAC), p. 983-988.
15. Meng, Q., Liu, H., Zhou, S., Zhang, Y., et al. Hardware-in the-loop Simulation of Process Control Based on OPC Technology and MATLAB/SIMULINK. *Advanced Science and Technology Letters* 2014, vol. 53, no. 12, pp. 49-52. Disponible en Internet: <<http://dx.doi.org/10.14257/astl.2014.53.12>>. ISSN 2287-1233 ASTL.
16. Kosarac, A., Prodanovic, S. y Zeljkovic, M. Possibilities for interactive control of machine tools in the virtual reality enviroment. *Acta Technica Corviniensis-Bulletin of Engineering*, 2013, 6(4), 105-108.
17. Silvente, J., Monroy, I., Escudero, G., España, A., et al. A promising OPC-based computer system applied to fault diagnosis. En *Actas de 22nd European Symposium on Computer Aided Process Engineering*, London, 2012, Ian David Lockhart Bogle and Michael Fairweather ed. Elsevier B.V., p. 1-5.
18. Peniak, P., Franekova, M. y Luley, P. Possibilities of control and information systems integration within industrial applications area. *International Journal of Engineering*, 2012, 10(2), 173-176.
19. González, I., Calderón, A. J. y Calderón, M. Fuzzy control of an electrolyzer in a stand-alone renewable energy system. En *Actas de 10th Portuguese Conference on Automatic Control*, Portugal, 2012, p. 39-44.
20. Manoj, R. y Janaki, S. Fuzzy Adaptive PID for Flow Control System based on OPC. *IJCA Special Issue on "Computational Science - New Dimensions & Perspectives"*, 2011, 5-8.
21. Maciejowski, J. M. *Predictive Control with Constraints*. London: Prentice Hall Editor. An imprint of Pearson Education. Harlow, England, 2000. 331 p.
22. Richalet, J. y O'Donovan, D. *Predictive Functional Control. Principles and Industrial Applications*. London: Springer, 2009. 222 p. ISBN 978-1-84882-492-8.
23. Rossiter, J. A. *Model-Based Predictive Control. A Practical Approach*. editado por Robert H. Bishop. London: CRC Press LLC, 2005. 318 p. ISBN 0-8493-1291-4.
24. Camacho, E. F. y Bordons, C. *Model Predictive Control*. España: Springer, 1999. 280 p. ISBN 3-540-76241-8.
25. Bemporad, A., Morari, M. y Ricker, N. L. *Model Predictive Control Toolbox. Getting Started Guide*. En.: The Mathworks, Inc., 2015, p. 198.
26. Kassem, A. M. Neural predictive controller of a two-area load frequency control for interconnected power system. *Ain Shams Engineering Journal* 2010, vol. 1, pp. 49-58. Disponible en Internet: <<http://dx.doi.org/10.1016/j.asej.2010.09.006>>.
27. Pivonka, P. y Miksanek, V. Real-Time Communication between MATLAB/Simulink and PLC via Process Visualization Interface. En *Actas de 11th WSEAS International Conference on SYSTEMS*, Greece, 2007, p. 28-32.
28. Roubos, J. A., Molloy, S., Babuska, R. y Verbruggen, H. B. Fuzzy model-based predictive control using Takagi-Sugeno models. *International Journal of Approximate Reasoning* 1999, vol. 22, pp. 3-30.
29. Davies, W. D. T. *System Identification for Self-Adaptive Control*. London: John Wiley and Sons, 1970. ISBN 0 471 19885 4.
30. Vérdes Kairuz, R. I. Síntesis de controladores predictivos y PID con autoajuste para un proceso de dinámica rápida a través de MATLAB®/Simulink® y OPC. En opción al título de Master en Informática Industrial y Automática CUJAE, 2016.
31. Astrom, K. y Hagglund, T. *PID Controllers: Theory, Design and Tuning*. 2da ed. USA: Instrument Society of America, 1995. 343 p. ISBN 1-55617-516-7.

AUTORES

Ramón Imad Verdés Kairuz

Ingeniero en Automática, Departamento de Automática y Computación, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría, CUJAE, La Habana, Cuba. imad@electronica.cujae.edu.cu

Ana Isabel González Santos

Ingeniera en Máquinas Computadoras, Máster en Automática, Doctora en Ciencias Técnicas, Departamento de Automática y Computación, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría, CUJAE, La Habana, Cuba. anita@electronica.cujae.edu.cu