Vol. 18, n.° 2, pp. 147-158,2019
Revista UIS Ingenierías
Página de la revista: revistas.uis.edu.co/index.php/revistauisingenierias

Implementación de un sistema de control para regular la velocidad y posición de motores industriales utilizando el protocolo de comunicación OPC

Implementation of a control system to regulate the speed and position of industrials motors using the OPC communication protocol

Hernando González-Acevedo [1a], Ólmer Giovanny Villamizar-Galvis [1b]

[1]Grupo de investigación de Control y Mecatrónica, programa de Ingeniería Mecatrónica, Universidad Autónoma de Bucaramanga, Colombia. Correos electrónicos: [a] hgonzalez7@unab.edu.co, [b] ovillamizar10@unab.edu.co

Recibido: 20 julio, 2018. Aceptado: 13 enero, 2019. Versión final: 6 febrero, 2019.

Resumen

El articulo presenta la implementación del protocolo de comunicación OPC (*Object Linking and Embedding for Process Control*) entre el *software* matemático Matlab y la plataforma de programación gráfica LabView, para enlazarlos con dos controladores lógicos programables (PLC): Siemens S7-300 y Allen Bradley ControlLogix 5566, y una estación de trabajo Festo. En cada programa se implementaron tres técnicas de control (PID, lógica difusa y LQG), para regular la velocidad de un motor AC acoplado a un generador DC y controlar la posición de un servomotor industrial, marca Rockwell. Se analiza el tiempo de respuesta que se obtiene con cada enlace de comunicación, así como las ventajas de la comunicación OPC y de cada *software*.

Palabras clave: control PID; controlador difuso; control LQG; protocolo OPC.

Abstract

The article presents the implementation of the OPC communication protocol (Object Linking and Embedding for Process Control) between the mathematical software Matlab and the graphics programming platform LabView to link them with two Programmable Logic Cntrollers (PLC): Siemens S7-300 and Allen Bradley ControlLogix 5566 and to a workstation Festo. In each program, three control techniques (PID, fuzzy logic and LQG) were implemented to regulate the speed of an AC motor coupled with a DC generator and to control the position of an industrial servomotor, Rockwell. The response time obtained with each communication link was analyzed, as well as the advantages of OPC communication and each software.

Keywords: OPC protocol; PID controller; LQG controller; fuzzy controller.

1. Introducción

Al conjunto de métodos, sistemas y herramientas que posibilitan el intercambio de información entre diferentes componentes industriales se los conoce como protocolos de comunicaciones industrial. Entre estos protocolos, uno de los más conocidos es el OPC (*Object Linking and Embedding for Process Control*), el cual funciona como un servidor de enlace entre diferentes *softwares* para la transmisión de datos entre sí. La comunicación por OPC es una herramienta en la cual diferentes equipos con protocolos de comunicaciones distintos puedan realizar

ISSN impreso: 1657 - 4583. ISSN en línea: 2145 - 8456, **CC BY-ND 4.0**
H. González-Acevedo, O. Villamizar-Galvis, "Implementación de un sistema de control para regular la velocidad y posición de motores industriales utilizando el protocolo de comunicación OPC," *Rev. UIS Ing.*, vol. 18, no. 2, pp. 147-158, 2019. doi: 10.18273/revuin.v18n2-2019014

una transmisión de datos, como se observa en la figura 1. La comunicación OPC se realiza a través de una arquitectura cliente-servidor. El servidor OPC es la fuente de datos (como un dispositivo *hardware* a nivel de planta), y cualquier aplicación basada en OPC puede acceder a ese servidor para leer/escribir cualquier variable que ofrezca el servidor [1].

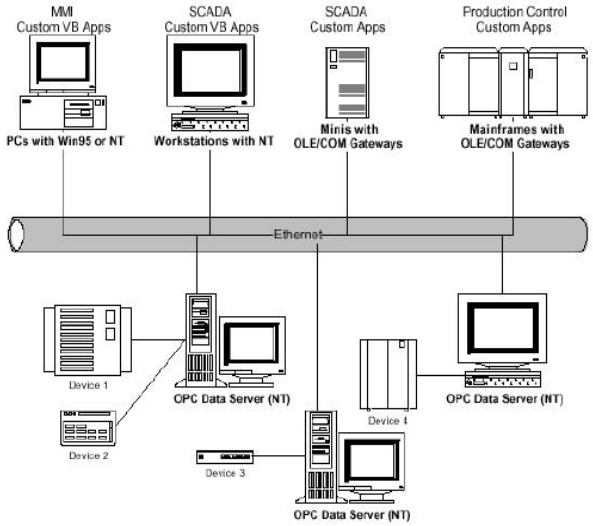


Figura 1. Red de comunicación OPC.

Una de las funciones de la comunicación por OPC actualmente es la solución al problema de interacción entre *softwares* y autómatas programables. En el estado del arte se han presentado trabajos, como el referenciado en [2], en el cual se realiza la implementación de un sistema de control en cascada y se valida los resultados de un controlador PID, utilizando dos métodos diferentes para la adquisición de datos (Sistema SCADA, PLC-OPC), comparando la facilidad en el manejo de datos que otorga cada dispositivo. En [3] se establece una comunicación entre el *software* Matlab y un PLC S7-300 para el monitoreo remoto de un tanque de medición de nivel; su enfoque principal es demostrar cómo se realizó detalladamente la conexión de estos dos sistemas, la configuración del *hardware* y del servidor OPC, y además dar un ejemplo de la programación. En [4] se observa una relación directa entre el *software* LabView y un PLC Allen Bradley Micrologix 1200C, comparando la ventaja de conectar el PLC a la red de área local y conectarlo por servidor OPC.

El artículo se distribuye de la siguiente forma: en la primera sección se describe la configuración de la comunicación OPC y cada dispositivo de la red; en la segunda sección, el modelo matemático y el diseño de cada uno de los controladores; y en la tercera sección, se establecen las interfaces HMI creadas y se compararan los resultados obtenidos aplicando cada estrategia de control en los dos *softwares* de programación, Matlab y Labview.

2. Comunicación OPC

Se reconoce la estación de trabajo (PC) como el dispositivo, el cual recibe los datos de los PLC por medio de un servidor OPC, para realizar de forma simultánea el control de velocidad de un motor AC y el control de posición de un servomotor. Además, la estación de trabajo se comunica con la MPS de Festo. En la figura 2 se detalla el esquema que se implementó para comunicar el computador con los dispositivos industriales. En los dos programas se encuentran las tres estrategias de control que se analizaron en el artículo, y el usuario puede seleccionar en tiempo real con el cual desea operar cada proceso. En este esquema se observan tres subgrupos, los cuales corresponden a los siguientes sistemas:

- Control de velocidad. En este caso se utiliza un PLC S7300 de la marca Siemens. Este PLC se encarga de comunicarse por medio de una red Profibus con el variador de frecuencia Micromaster 420, al cual se conecta un motor AC de un caballo de fuerza.

- Control de posición de un servomotor. El PLC ControlLogix 5566 de Allen Bradley se comunica vía *Ethernet* con el *driver* Kinetix 300, que regula la velocidad del servomotor.

- Estación FESTO. El PLC Festo se comunica vía *Ethernet* a través del servidor OPC con la estación de trabajo, la cual controla un módulo MPS (Handling) que se encarga de realizar una secuencia para transportar una pieza hasta un lugar de almacenamiento.

2.1. Configuración de la red

La comunicación de los equipos industriales se realiza por medio de una red Ethernet, topología tipo estrella [5]. Se estableció una dirección IP de red "Clase C", dadas las direcciones IP disponibles en la red en la cual se implementó el sistema bajo estudio. La red IP sobre la cual se direccionaron los equipos fue la dirección 192.168.124. En la tabla 1 se describen las direcciones IP de cada dispositivo.

2.2. Servidor OPC

El servidor KEPServerEX fue elegido por dos factores importantes: la velocidad de comunicación y la integración en un solo *software* de los diferentes

dispositivos soportados. El tiempo mínimo que se puede establecer para la comunicación usando este servidor OPC es de 0,15 segundos. Posterior a la configuración del servidor OPC, se configuran los clientes OPC, los cuales van ligados de la mano a los *softwares* de programación que se usarán para el desarrollo del proyecto.

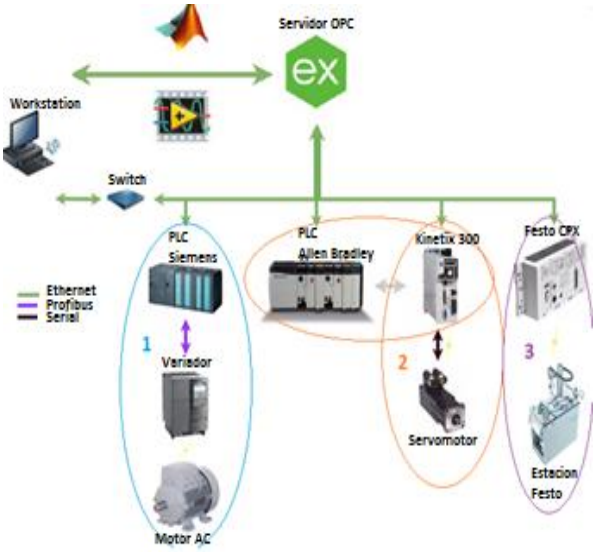


Figura 2. Esquema de comunicaciones detallada de la red industrial.

Tabla 1. Direccionamiento IP de los dispositivos.

EQUIPO	DIRECCIÓN IP
 Siemens S7 300	192.168.124.100
 AllenBradley ControlLogix 5566	192.168.124.80
 Driver Kinetix300	192.168.124.81
 Festo CPX	192.168.124.120

Los programas que se usan para el control de los procesos son Matlab y LabView; estos programas presentan herramientas internas, las cuales permiten configurarlos como clientes OPC, para poder realizar la transmisión de los datos entre cada *software* y el servidor.

3. Diseño del sistema de control

Para determinar el modelo matemático de los procesos sobre los cuales se va a implmentar una estrategia de control, se optó por realizar un modelo de caja negra, ya que este tipo de modelos describe fácilmente el comportamiento de las dos plantas. La primera planta consta de un variador de frecuencia Micromaster 420, un motor AC marca Siemens de 1 hp acoplado por un eje a un motor DC de 1/8 de hp que actuará como generador (figura 3). La función de transferencia que describe la dinámica de este sistema está dada por la ecuación (1); la variable manipuladora es la frecuencia que se programa al variador de frecuencia, y la variable por medir es la velocidad del eje, la cual se encuentra en un rango de cero hasta 1200 rpm.

$$\mathrm{G(s)} = \frac{29.8\ \mathrm{e}^{-0.2645\ \mathrm{s}}}{0.00315\ \mathrm{s\hat{}2} + 0.5481\ \mathrm{s} + 1} \qquad (1)$$

Se discretiza la función de transferencia $\mathrm{G(s)}$, de acuerdo con el periodo de muestreo, que en ese caso es la velocidad de transmisión de datos del servidor OPC, $\mathrm{T_m} = 0{,}15$ sg , y se obtiene la función de transferencia:

$$\mathrm{G(z)} = \mathrm{z}^{-2} * \frac{1.612\ \mathrm{z}^2 + 5.587\ \mathrm{z} + 0.0004981}{\mathrm{z}^2 - 0.7584\ \mathrm{z} + 4.638\mathrm{e}^{-12}} \qquad (2)$$

La segunda planta corresponde a una barra de acero acoplada a un servomotor industrial marca Rockwell; el *driver* que regula la velocidad del servomotor, Kinetix 300, tiene incorporado un controlador de velocidad. La dinámica del sistema está descrita por la función de transferencia de la ecuación (3). La variable manipuladora es la palabra de control que se le envía al *driver*, la cual va en un rango de -1 hasta 1, donde ± 1 representa la máxima velocidad en el sentido horario o antihorario, según corresponda. La barra se mueve en un rango de 0^{o} a 360^{0}.

$$G(s) = \frac{0.5552e^{(-0.337s)}}{0.4675\ \mathrm{s\hat{}2} + \mathrm{s}} \qquad (3)$$

Se discretiza la función de transferencia $G(s)$ con el periodo de muestreo de la comunicación OPC, $T_m = 0{,}15$ sg, y se obtiene la siguiente función de la planta.

$$G(z) = z^{-3} * \frac{0.007006\ z\hat{}2 + 0.01525\ z + 0.0006057}{z\hat{}2 - 1.726\ z + 0.7255} \qquad (4)$$

3.1. Control PID

El control PID se ha utilizado de manera exitosa en muchos sistemas de control industrial por más de medio siglo. El principio básico del esquema del control PID es que actúa sobre la variable que será manipulada, a través

de una apropiada combinación de tres acciones de control: acción de control proporcional, donde la acción de control es proporcional a la señal de error, la cual es la diferencia entre la entrada y la señal de realimentación; la acción de control integral, donde la acción de control es proporcional a la integral de la señal de error; y la acción de control derivativa, donde la acción de control es proporcional a la derivada de la señal de error [6]. La acción de control PID en controladores digitales se representa de la siguiente forma:

$$U_k = K\left[e_k + \frac{T_m}{T_i}\sum_{i=0}^{k-1} e_i + \frac{T_d}{T_m}\,(e_k - e_{k-1})\right] \quad (5)$$

donde e_k es la entrada al controlador (señal de error), T_m es el periodo de muestreo, K es la ganancia proporcional, T_i es el tiempo integral (o tiempo de reajuste) y T_d es el tiempo derivativo (o tiempo de adelanto). Aplicando la transformada Z se obtiene la función de transferencia del controlador.

$$C(z) = K_p + K_i\frac{1}{(1-z^{-1})} + K_d(1-z^{-1}) \quad (6)$$

$$K_P = K - \frac{KT}{2T_i} \quad (7)$$

$$K_I = \frac{KT}{T_i} \quad (8)$$

$$K_D = \frac{KT_d}{T} \quad (9)$$

Donde K_P es la ganancia proporcional, K_D es la ganancia derivativa y K_I es la ganancia integral. En un sistema, puede suceder que la variable de control alcance los límites prefijados del actuador. Cuando esto pasa, el bucle realimentado permanece en su límite independientemente de la salida del proceso. Si se usa un controlador con acción integral, el error continuará siendo integrado, y s eincrementará aún más su valor. Esto significa que el valor integral puede volverse muy grande y producirse el efecto llamado *windup* [7]. Para evitar que ocurra este fenómeno hay dos formas:

- Introducir limitadores en las variaciones de la referencia, de modo tal que la salida del controlador nunca alcance los límites del actuador. Esto a menudo produce límites en el funcionamiento del controlador y no evita el *windup* causado por las perturbaciones.

- Otra forma es el recálculo de la integral. Cuando la salida se satura, la integral se recalcula, de modo tal que su nuevo valor proporciona una salida en el límite de la saturación.

En la figura 3 se puede apreciar la estrategia implementada en Simulink para el controlador PID agregando la ganancia *anti-windup*, K_{aw}.

Para diseñar los controladores se utilizó la técnica de ubicación de polos y ceros [8], para ambos sistemas se decidió que la respuesta transitoria presentara un comportamiento sobreamortiguado. En la tabla 2 se muestran los criterios de diseño y las constantes del controlador PID.

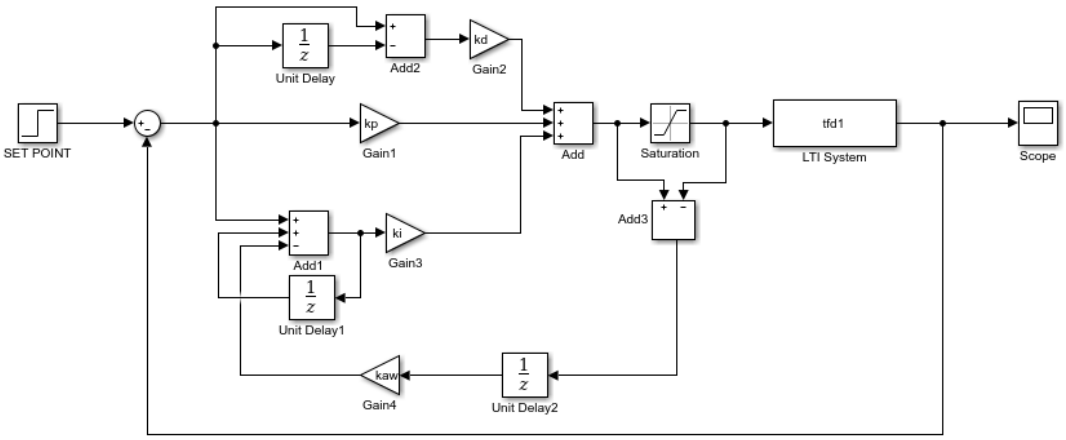


Figura 3. Diagrama de bloques controlador PID en Simulink.

3.2. Control LQG

El controlador LQG es la combinación de una ganancia de realimentación de estados y un estimador de estados tipo Kalman [9].

Para el diseño del controlador se debe representar el modelo lineal del sistema en espacio de estados, ecuaciones (10) y (11), donde G es la matriz de estados; H, la matriz de entrada; C, la matriz de salida; D, la matriz

de transmisión directa; x(k), el vector de estados; u(k), el vector de entrada al sistema; y(k), el vector de salida del sistema; w(k),el ruido presente en los actuadores, y v(k), el ruido captado por los sensores.

$$x(k+1) = Gx(k) + H(u(k) + w(k)) \quad (10)$$

$$y(k) = Cx(k) + D(u(k) + w(k))] + v(k) \quad (11)$$

Tabla 2. Parámetros de diseño y constantes del controlador PID.

Parámetro	Control de velocidad	Control de posición
Criterios de diseño	$Ts = 15\,sg$ $Mp = 0\%$	$Ts = 8\,sg$ $Mp = 0\%$
Constantes del controlador	$Kp = 0.00068$ $Kd = 0.00099375$ $Ki = 0.00099375$ $K_{aw} = 56$	$Kp = 3$ $Kd = 0.8$ $Ki = 0$

La ganancia de retroalimentación de estados se determina de forma que minimice la función dada por la ecuación (12). La solución se encuentra establecida por la ecuación de Ricatti.

$$J = \sum_{k=1}^{\infty}(x(k)^T\, Qx(k) + u(k)^T Ru(k)) \quad (12)$$

Para estimar los estados se utiliza un filtro Kalman, el cual asume la presencia de ruido gaussiano en la señal de control y la variable por medir en el proceso. Conocidas las varianzas del ruido de estas dos señales, Q_n y R_n, se determina un vector de ganancias L que minimiza el error en la estimación, x - $\hat{x}$, utilizando la ecuación de Ricatti. Matemáticamente, los estados se calculan a partir de la siguiente expresión:

$$\hat{x}(k+1) = G\hat{x}(k) + Hu(k) + L(y(k) - C\hat{x}(k)) \quad (13)$$

Para el diseño del sistema de control LQG, se definen las matrices de penalidad Q y R, y la matriz de covarianza asociada al ruido presente en la señal de control y del sensor, Q_n y R_n. En la figura 4 se observa el diagrama de bloques del sistema de control implementado en la herramienta Simulink de Matlab. En el esquema se ajusta una ganancia *anti-windup* K_{aw}, que permite reducir el efecto de la saturación del actuador, la constante integral K_i, la matriz que multiplica los estados K_{est} y el estimador de Kalman.

En la tabla 3 se presentan los matrices que se definen para el diseño del controlador LQG para los dos sistemas bajo estudio.

Tabla 3. Parámetros de diseño y representación matricial del controlador LQG.

Parámetro	Control de velocidad	Control de posición
Criterios de diseño	$Ts = 15\,sg$ $Mp = 0\%$	$Ts = 8\,sg$ $Mp = 0\%$
Matrices de penalidad	$Q = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$ $R = [15]$	$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.0020 \end{bmatrix}$ $R = [2]$
Matrices de covarianza	$Q_n = [0.2^2]$ $R_n = [1.9^2]$	$Q_n = [0.00001^2]$ $R_n = [0.001^2]$
Constantes del controlador	$K_e = [\,0.4200 \quad 0.8570]$ $K_i = [0.0240]$ $L = \begin{bmatrix} -0.0004 \\ 0.0094 \end{bmatrix}$ $K_{aw} = 20$	$K_e = [\,0.6234 \quad 1.2891]$ $K_i = [0.0301]$ $L = \begin{bmatrix} 0.0140 \\ 0.0610 \end{bmatrix}$ $K_{aw} = 18$

3.3. Controlador basado en lógica difusa

El control difuso es una alternativa para resolver complejas aplicaciones; propone un método para construir controles no lineales a través de la información heurística. Al desarrollar un controlador difuso es posible prescindir de la rigidez matemática y transmitir el raciocinio humano y convertirlo en un sistema. El control difuso se compone de los siguientes elementos: fusificación, base de conocimiento, inferencia y defusificación.

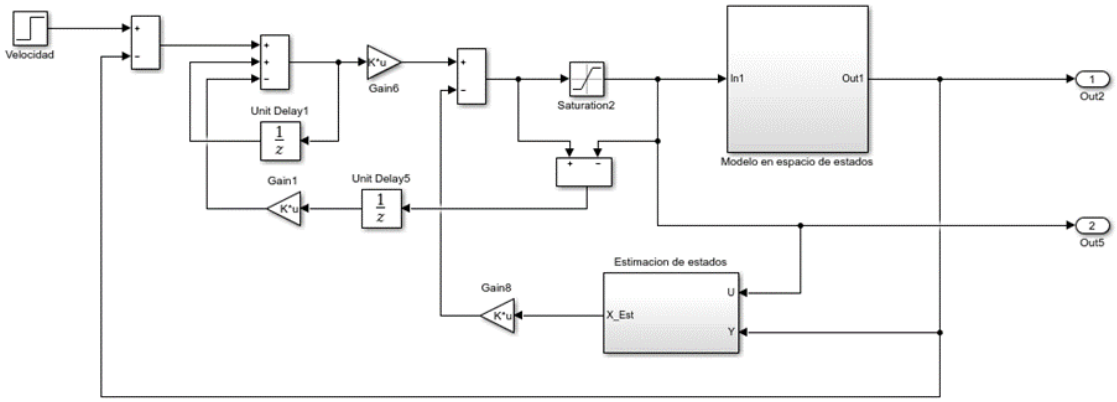


Figura 4. Diagrama de bloques del controlador LQG.

El proceso comienza con el fuzificador. La entrada de un sistema de lógica difusa normalmente es un valor numérico proveniente, por ejemplo, de un sensor; para que este valor pueda ser procesado por el sistema difuso, se hace necesario convertirlo a un "lenguaje" que el mecanismo de inferencia pueda procesar. Esta es la función del fuzificador, que toma los valores numéricos provenientes del exterior y los convierte en valores difusos que pueden ser procesados por el mecanismo de inferencia. Estos valores difusos son los niveles de pertenencia de las entradas a los diferentes conjuntos difusos en los cuales se ha dividido el universo de discurso de las diferentes variables. Luego pasa por el mecanismo de inferencia difusa. Teniendo los diferentes niveles de pertenencia arrojados por el fuzificador, los niveles deben ser procesados para generar una salida difusa. La tarea del sistema de inferencia es tomar los niveles de pertenencia y apoyado en la base de reglas generar la salida del sistema difuso. Finalmente, está la base de reglas difusas, las cuales se definen de acuerdo con la experiencia y presentan la sintaxis definida en la figura 5.

Para el diseño del controlador difuso, se implementó el diagrama de bloques de la figura 6, en el cual el error y la razón de cambio del error, previamente multiplicadas por dos ganancias de normalización (K_e y K_{RE}), son la entrada al sistema difuso. Si el proceso que se desea controlar no presenta integrador, la salida del sistema difuso se integra, multiplicando previamente por una ganancia K_I, esto con el objetivo de garantizar que la variable del proceso alcance el punto de consigna; por lo tanto, en el sistema de control de posición el integrador se anula [10]. En la figura 7 se observan las cinco funciones de pertenencia que se establecieron para las dos variables de entrada, el error y la razón de cambio del error, y la variable de salida.

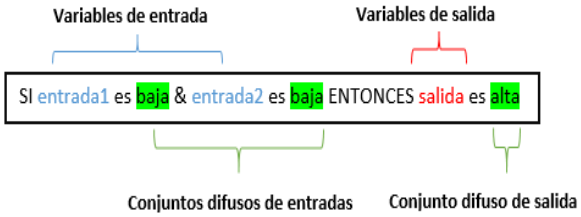


Figura 5. Lógica de condiciones de un sistema difuso tipo Mamdani.

El universo para cada variable se definió en un rango entre -1 a 1.

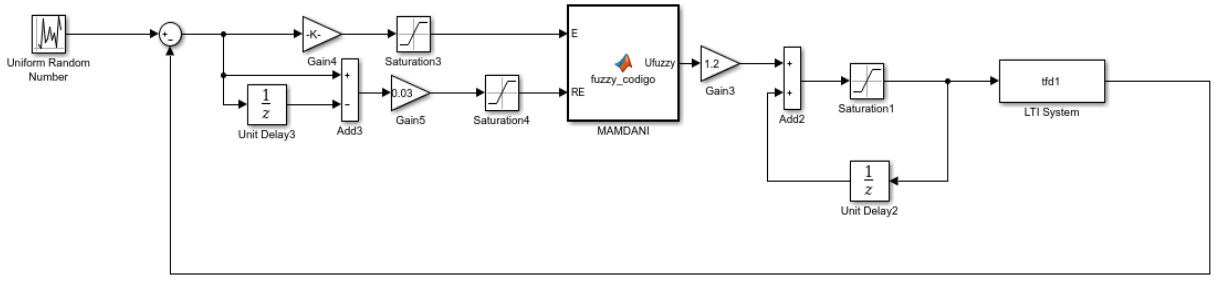
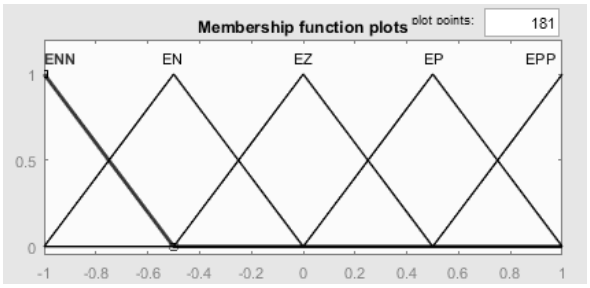
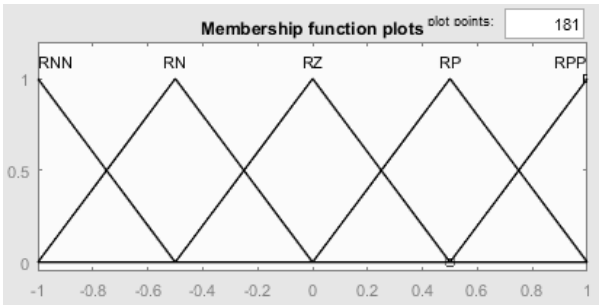


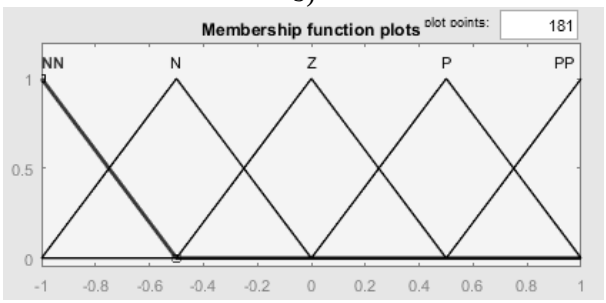
Figura 6. Diagrama de bloques del controlador difuso.



a)



b)



c)

Figura 7. Funciones de pertenencia. a) Variable error, b) variable razón de cambio del error y c) variable de salida del sistema difuso.

El diseño de las reglas es el resumen del sistema de control que ofrece el experto sobre el conocimiento de cómo controlar la planta, lo que afecta directamente la calidad del sistema. El sistema de reglas de control se diseñó con los siguientes principios de sintonización:

- Cuando $E = 0$ y $RE = 0$, la variable por controlar se encuentra sobre del punto de consigna, la acción de control no varía.

- Cuando $E > 0$ la variable a controlar se encuentra por debajo del punto de consigna. La señal de control depende de la razón de cambio del error; si es negativa significa que se acerca rápidamente a la señal de referencia; por tanto, la acción de control disminuye; si la razón de cambio es positiva, significa que la variable del proceso se aleja de la señal de referencia, por lo cual la señal de control debe aumentar.

- Cuando $E < 0$, la variable por controlar se encuentra por encima del punto de consigna. La señal de control depende de la razón de cambio del error. Si es negativa, significa que se aleja rápidamente de la señal de referencia, y, por tanto, la acción de control debe disminuir; si la razón de cambio es positiva, significa que la variable del proceso se acerca a la señal de referencia, por lo cual la señal de control debe disminuir.

En la tabla 4 se observa la base de reglas establecidas para el sistema de control difuso. En la tabla 5 se observan las

constantes definidas para el controlador difuso, para los dos procesos bajo estudio.

Tabla 4. Base de reglas del controlador difuso.

		E				
		ENN	**EN**	**EZ**	**EP**	**EPP**
RE	**RNN**	NN	NN	N	Z	P
	RN	NN	N	N	P	P
	RZ	NN	N	Z	P	PP
	RP	N	N	P	P	PP
	RPP	N	Z	P	PP	PP

4. Interfaz HMI

En la industria se observan diferentes métodos que le permiten a un operario manipular la planta de una forma sencilla, por medio de una interfaz gráfica, la cual puede ser propuesta, como en este caso, por algún *software* de programación, ya sea LabVIEW o Matlab. En este caso se vio la necesidad de crear una interfaz HMI para cada uno de los programas, que le permita al operario manipular y visualizar el comportamiento de los sistemas utilizando el protocolo de comunicación OPC [11] [12]. Para el *software* Matlab, se diseñó una interfaz que integra los controladores PID, LQG y lógica difusa de cada sistema, respectivamente, además del control de la MPS de Festo, y ofrece un panel en el cual se puede seleccionar el proceso, el controlador y el punto de consigna que se requiera (figura 8). En el sistema se visualiza en tiempo real el comportamiento de cada una de las variables y la señal de control.

Tabla 5. Ganancias del controlador difuso. K_e y K_{RE}.

Parámetro	Control de velocidad	Control de posición
Ganancias del controlador	$K_e = 0.8$ $K_{RE} = 0.4$ $K_I = 1.3$	$K_e = 0.07$ $K_{RE} = 0.1$ $K_I = 1.1$

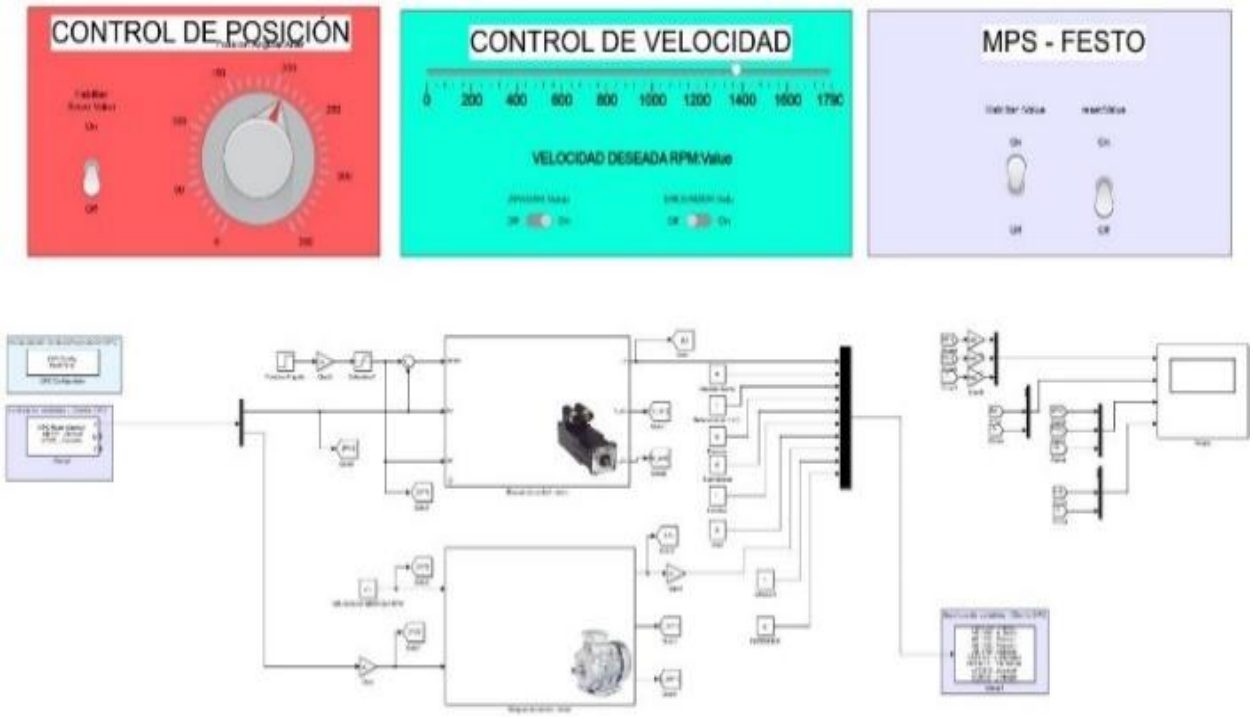


Figura 8. Interfaz HMI en Matlab.

LabVIEW tiene la ventaja de ofrecer una HMI más amigable para el usuario, a comparación de Matlab. En este *software* se integró la comunicación con los tres PLC utilizados (Siemens, AllenBradley y Festo); con este *software* se puede obtener mayor provecho en cuanto a la integración y visualización de procesos que trabajen con diferentes tecnologías (figura 9). La interfaz permite el control del proceso y monitorear las variables relevantes de cada sistema.

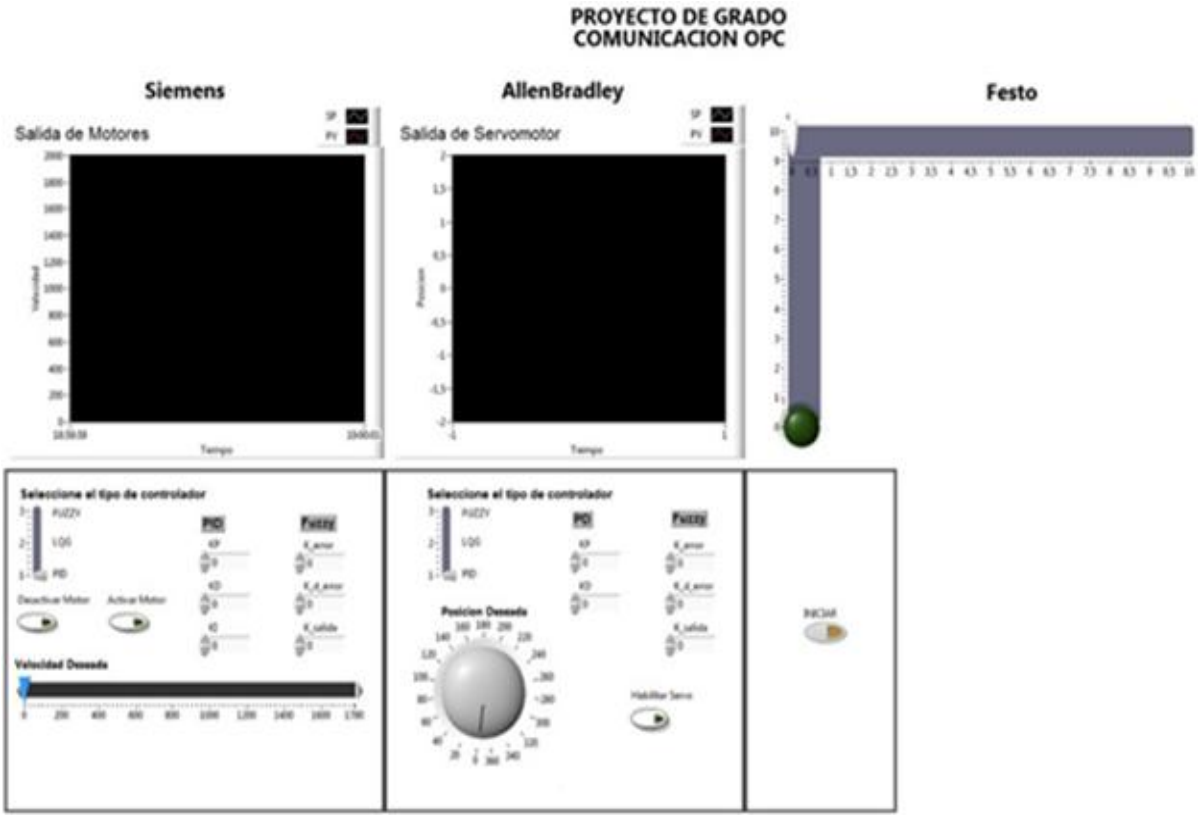


Figura 9. Interfaz HMI en LabView.

5. Resultados

Se expondrán los resultados de los controladores implementados para cada sistema respectivamente. En este caso se compararán las respuestas obtenidas del modelo implementado en cada *software* (Matlab y LabView) aplicando los tres controladores.

5.1. Control de velocidad

Se implementaron los tres controladores para el sistema de regulación de velocidad. En las figuras 10, 11 y 12 se observa la respuesta transitoria, dado un *setpoint* variable, para el controlador PID, LQG y lógica difusa, respectivamente.

En cada gráfica se superponen los resultados evaluados con la HMI desarrollada en Matlab y Labview. Se observa en los tres controladores que la respuesta transitoria evaluada con la HMI de LabView es lenta comparada con la HMI implementada en Matlab. Esto se debe a que en el *software* LabView no se garantiza el periodo de muestreo ($T_m = 0.15\ sg$), la respuesta no se afecta en cuanto a sobrepaso, pero sí se estabiliza en un tiempo diferente, mayor al de la implementación en Matlab. De otro lado, el sistema de control adecuado para regular la velocidad es el controlador PID, y presenta una respuesta sobreamortiguada y con un tiempo de establecimiento similar al establecido en la etapa de diseño, $Ts = 15$ sg.

El control LQG responde de forma adecuada, pero demora 22 sg en estabilizar. Se resalta que el modelo de caja negra es una buena aproximación de la dinámica del sistema. El controlador difuso responde de forma lenta, debido al integrador que se encuentra a la salida de la etapa de defusificación.

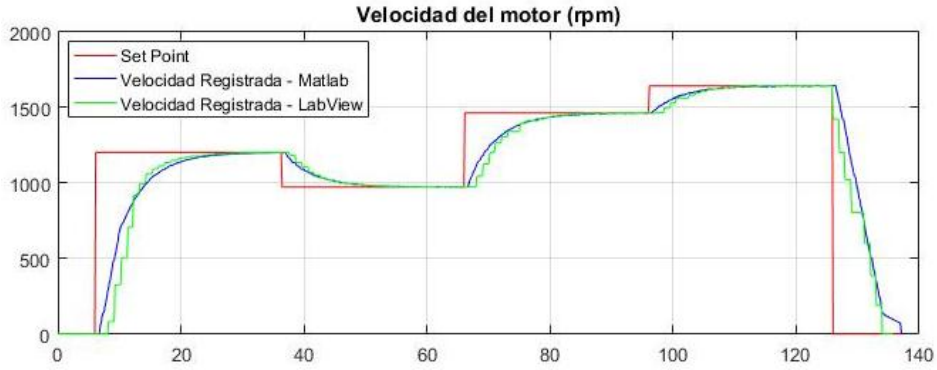


Figura 10. Control PID para regular la velocidad de un motor AC - Matlab y LabView.

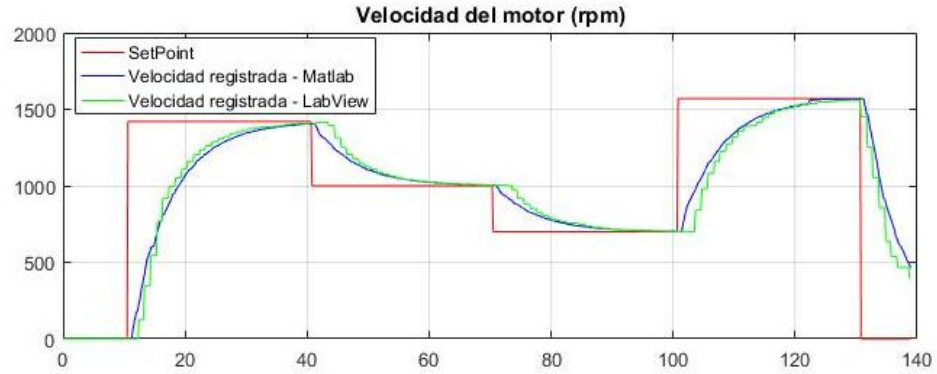


Figura 11. Control LQG para regular la velocidad de un motor AC - Matlab y LabView.

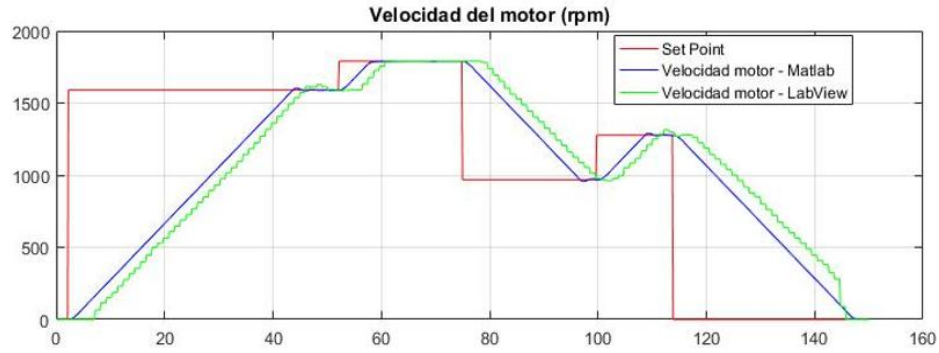


Figura 12. Control basado en lógica difusa para regular la velocidad de un motor AC - Matlab y LabView.

5.2. Control de posición

Se implementaron los tres controladores para el sistema de posición, y los resultados se observan en las siguientes imágenes: control PID en la figura 13, control LQG en la figura 14 y control basado en lógica difusa en la figura 15. Para el control de posición se observa que la respuesta en LabView, la señal correspondiente a la posición de la barra, presenta un retardo significativo entre el *software* Labview y el Matlab; y esto se debe a la gran cantidad de objetos gráficos ubicados en el *software* de Labview, que no garantizan el periodo de muestreo. Esto ocasiona que el sistema no sea capaz de controlar de forma eficaz la inercia de la barra y se presenten sobrepasos, o sea más lenta la estabilización de la planta. Para este sistema, el control basado en lógica difusa es el más adecuado; presenta un tiempo de establecimiento de 15 sg y un sobrepaso del 4 %.

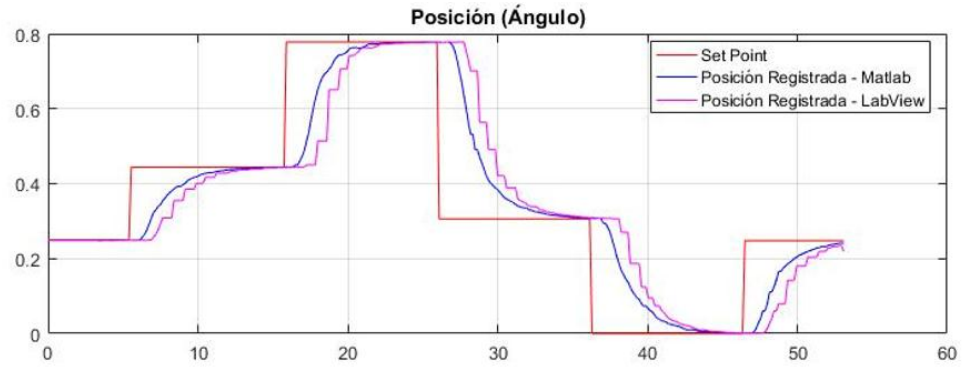


Figura 13. Control PID para regular la posición angular de un servomotor - Matlab y LabView.

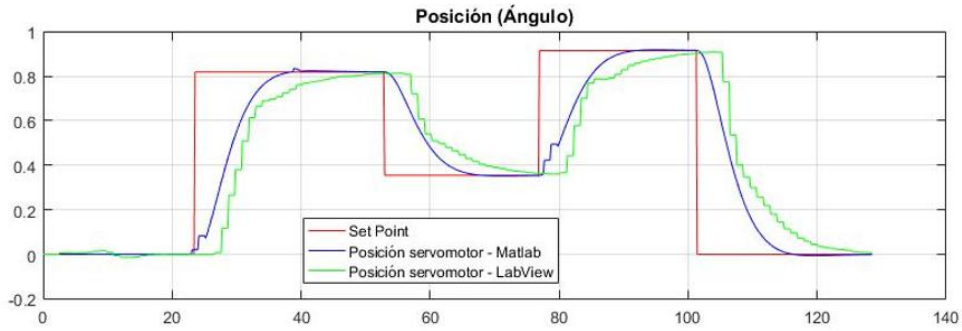


Figura 14. Control LQG para regular la posición angular de un servomotor - Matlab y LabView.

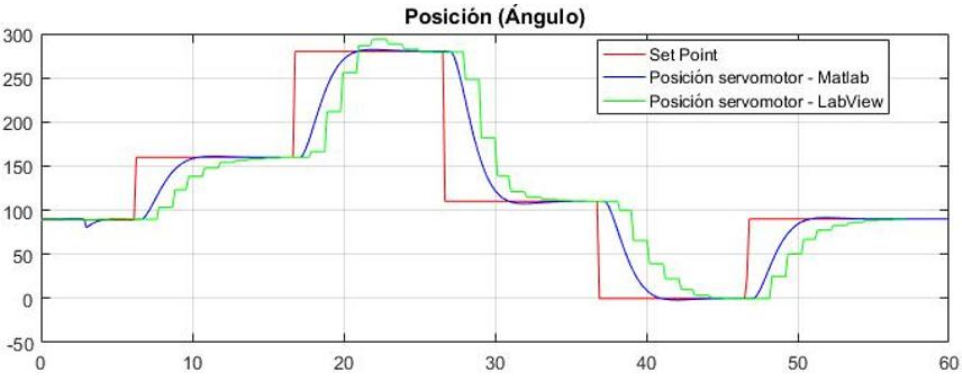


Figura 15. Control basado en lógica difusa para regular la posición angular de un servomotor - Matlab y LabView.

6. Conclusiones

El estándar de comunicación OPC es una herramienta útil, ya que permite integrar diferentes tipos de tecnologías (Siemens, AllenBradley y Festo) en una mimas interfaz HMI; se pueden controlar cantidades significativas de diferentes dispositivos conectados a una red, y además permiten ser trabajados por *softwares* como Matlab y LabView, los cuales son programas diseñados para cálculos o diseños más complejos que los que permite un PLC convencional, como lo son el control LQG y el control difuso.

Para periodos de muestreo menores a 0,5 sg, el *software* LabView no garantizaba una tasa de muestreo constante; por tanto, la respuesta del sistema era diferente a la diseñada, y esto se debe a que la programación de bloques de LabView realiza el barrido de toda la lógica y encuentra lazos que consumen más tiempo y no puede asegurar el periodo de muestreo. Para aplicaciones lentas,

controles de temperatura u otros, LabView es una herramienta útil, porque su ambiente de programación es amigable para el diseño de la interfaz HMI.

Referencias

[1] J. M. Zamarreño, *Acceso a datos mediante OPC*. Editorial Andavira, 2010.

[2] A. Lakshmi; B. Sangeetha; A. Naveenkumar; Balaji Ganesh; N. Bharathi, "Experimental Validation of PID Based Cascade Control System through SCADA-PLC-OPC Interface", en *International Conference on Computer Communication and Informatics*. 2012. doi: 10.1109/ICCCI.2012.6158893

[3] Z.Lieping; Z. Aiqun; Z.Yunsheng, "On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology", *Chinese Control Conference*, 2007. doi: 10.1109/CHICC.2006.4346760

[4] Tushar V. Bhaskarwar; Shripad S Giri; R. G. Jamakar, "Automation of shell and tube type heat exchanger with PLC and LabVIEW", *International Conference on Industrial Instrumentation and Control (ICIC*, 2015. doi: 10.1109/IIC.2015.7150859

[5] W. Stallings, *Comunicaciones y redes de computadores*. Editorial Prentice Hall, 2000.

[6] K. Ogata, *Sistemas de control en tiempo discreto*. Editorial Prentice Hall, 1996.

[7] Xin-lan Li; Jong-Gyu Park; Hwi-Beom Shin, "Comparation and Evaluation of Anti-Windup PI Controllers ", *Journal of Power Electronics*, vol. 11, no. 1, 2011.

[8] L. J. Marín, V. M. Alfaro, "Sintonización de controladores por ubicación de polos y ceros", *IEEE CONESCAPAN XXVI*, 2007.

[9] R. S. Burns, *Advanced Control Engineering*. Editorial Butterwoth Heinemann, 2001.

[10] P. Ponce, *Inteligencia artificial con aplicaciones a la ingeniería*, Editorial Marcombo S.A., 2011.

[11] "Matlab Toolbox OPC guide", 2017. [En línea]. Disponible en: https://www.mathworks.com/products/opc.html. [Accedido: 05-ene-2017]

[12] National Instruments, "Conecte LabVIEW a Cualquier PLC", 2017. [En línea]. Disponible en: http://www.ni.com/white-paper/7906/es/.[Accedido: 05-ene-2017].