

Diseño e Implementación de un Sistema de Control utilizando MATLAB enlazado a un Autómata Programable

Uribe A.¹, Gutiérrez W.²

auribec@sena.edu.co¹, william.gutierrez@sena.edu.co²

Centro de Electricidad y Automatización Industrial, SENA, Regional Valle, Santiago de Cali, Colombia

Resumen—En la actualidad, el estudio y análisis de los sistemas de control automático representan un campo de aplicación bastante extendido, esto debido a las imperantes necesidades por mejorar los procesos productivos, optimizar los recursos e incluso para predecir comportamientos, desde entornos educativos, investigativos o industriales, hasta entornos económicos, sociales y demás. En efecto, esto ha impulsado a los desarrolladores a buscar herramientas o alternativas que permitan aumentar las capacidades de cómputo al momento de desarrollar un sistema de control automático. Ahora bien, la instrumentación industrial desde hace muchos años viene comandada por los Autómatas Programables (PLC), dispositivos útiles e ideales para ejecutar las tareas de control de una planta, independientemente de la escala. El presente documento presenta el diseño e implementación de un controlador PID en una planta didáctica conformada con equipos industriales, haciendo uso de OPC y algunos conceptos de la teoría de control. Mediante la herramienta computacional MATLAB se realiza la identificación de la planta obteniendo su función de transferencia, y haciendo uso de algunas herramientas disponibles en este software, se determinan los parámetros de sintonía de un controlador tipo PID, para posteriormente sintonizar la estrategia de control implementada en un autómata programable de gama alta.

Index Terms—controlador PID, estándar OPC, función de transferencia.

I. INTRODUCCIÓN

En la actualidad, existen numerosas estrategias y maneras de realizar control sobre variables físicas, ejemplo de esto, tenemos controladores mecánicos como los vetustos trenes de la revolución industrial, que basaban el control del flujo del combustible mediante la manipulación de una válvula conectada a uno de los ejes del motor y mediante el efecto centrífugo, se realizaba el control en la velocidad; de otro lado, controladores eléctricos de dos posiciones como los que realizan los termostatos de los antiguos sistemas de aire acondicionado o neveras; y finalmente, controladores realimentados, cuyo propósito consiste en cerrar el lazo de control, obteniendo una lectura permanente de las variables controladas, y mediante la ejecución de un algoritmo o ecuación matemática, establecen una señal de control sobre actuadores.

Existen numerosas estrategias de control, sin embargo, en los entornos industriales, este tipo de controladores, denominados en algunas ocasiones como continuos, análogos,

o de lazo cerrado, usualmente son basados en estrategias como PID, el cual se calcula matemáticamente a través de la diferencia entre la consigna (SetPoint) y el valor medido de la variable, donde se tiene en cuenta la proporcionalidad, el error estacionario y el error transicional, esto traducido matemáticamente como un valor proporcional al error, su integral y su derivada.

Algunos PLC tienen ya incorporados módulos especiales como el PID y mediante interfaces análogas, ejecutan el control sobre actuadores, con base en la medición de variables físicas del proceso, a través de transmisores industriales. Ahora bien, para obtener una respuesta de control satisfactoria, el PID debe sintonizarse previamente, es decir, deben establecerse sus parámetros principales: constante proporcional, constante integral y constante derivativa (k_p , k_i , k_d). Estas dependen directamente del comportamiento de la planta y los requerimientos de control, como tiempos de estabilización, sobre impulso y demás. (Leonardo F. Lozano Valencia - 2014).

El proceso de sintonía es fundamental para el éxito de un buen controlador PID; existen diferentes maneras de realizarlo, bien sea analítica o matemáticamente o mediante ensayos empíricos. Ziegler-Nichols sugieren una metodología muy sencilla, la cual consiste en obtener la respuesta de la planta ante un escalón y a través de una serie de simples cálculos, se obtienen los parámetros del controlador. Este método puede ser funcional y arrojar en algunas ocasiones, buenos resultados, sin embargo, en todos los casos no es posible hacer ensayos, sobre todo cuando se trata de entornos industriales (Víctor M. Alfaro Ruíz – 2002).

Una solución eficaz a esto es pues, destinar la mirada a software tipo CAE -ingeniería asistida por computador- y existe una portentosa aplicación computacional que integra herramientas de cálculo y simulación, capaz de seducir al buen desarrollador mediante sus innumerables módulos funcionales. MATLAB es un software que, entre sus tantas funcionalidades, contiene módulos que podrían estimar la función de transferencia, el ADN de una planta; permitiría también calcular su respuesta ante diferentes entradas, y estimar las constantes de un controlador PID, dependiendo

de un par de parámetros.

Teniendo en cuenta lo anterior, surge la enorme cuestión: ¿cómo conectar esta herramienta a un autómata programable? Pues de esta manera ¿se imagina usted las incontables aplicaciones que podrían realizarse? Este documento presenta una solución y un ejemplo de aplicación. A continuación, se expone una manera de realizarlo a través del estándar OPC (Ole for Process Control).

II. METODOLOGÍA

OPC es el método de conectividad de datos basado en los estándares más populares del mundo. Es utilizado para responder a uno de los mayores retos de la industria de la automatización: cómo comunicar dispositivos, controladores y/o aplicaciones sin caer en los problemas habituales de las conexiones basadas en protocolos propietarios.

OPC no es un protocolo, sino más bien un estándar para la conectividad de datos que se basa en una serie de especificaciones OPC gestionadas por la OPC Foundation. Cualquier software que sea compatible con estas especificaciones OPC proporciona a usuarios e integradores conectividad abierta e independiente tanto del fabricante del dispositivo como del desarrollador de la aplicación Cliente. (Darek Kominek, P. Eng. Alberta, Canada – 2009).

La metodología de comunicación entre dos dispositivos a través de OPC se basa en la filosofía Cliente-Servidor. Las figuras 1 y 2 ilustran las capas de este concepto.

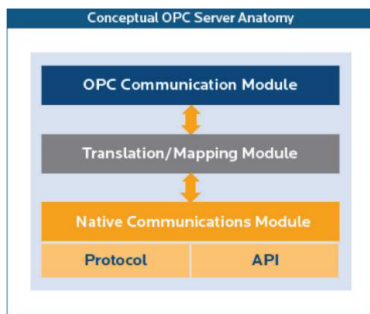


Figura 1. Anatomía conceptual de un Servidor OPC.

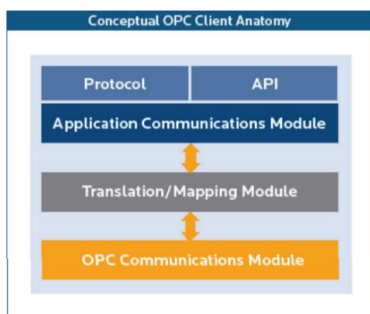


Figura 2. Anatomía conceptual de un Cliente OPC.

MATLAB a su vez, posee una ToolBox, dedicada al uso del estándar OPC, a través de la cual, una vez configurados los clientes/servidores OPC, y establecida la conexión, se hace posible controlar las variables o TAGS del PLC en todo momento. La gran ventaja de esto es que, bajo este estándar de conexión, no se aísla el hilo comunicación, permitiendo así el multi-enlace. Esto es útil cuando los dispositivos están dentro de una red de comunicación, sistema SCADA o sistema de control distribuido.

II-A. Conexión MatLab - Servidor OPC

Como se mencionó, la herramienta computacional MATLAB contiene una toolbox con funcionalidades de Cliente, la cual permite establecer conexión a uno o varios servidores OPC y mediante el uso de TAGS se puede interactuar con estos.

II-A1. Registro de OPC en MatLab: Para registrar OPC en MATLAB, se ingresa el siguiente comando:

```
>>opcregister(install)
```

Una vez registrado, se podrá hacer uso de la herramienta, a través de la cual será primordial tener en cuenta los nombres de los servidores OPC con los que se trabajará. Para el presente estudio, se trabaja con el módulo OPC del software RSLinx Classic Gateway v2.51, cuyo servidor OPC por defecto está denominado como: "RSLinx OPC Server".

A continuación, se ejecuta en el entorno de MATLAB, esta ToolBox, escribiendo el comando:

```
>>opctool()
```

Se despliega la siguiente ventana:

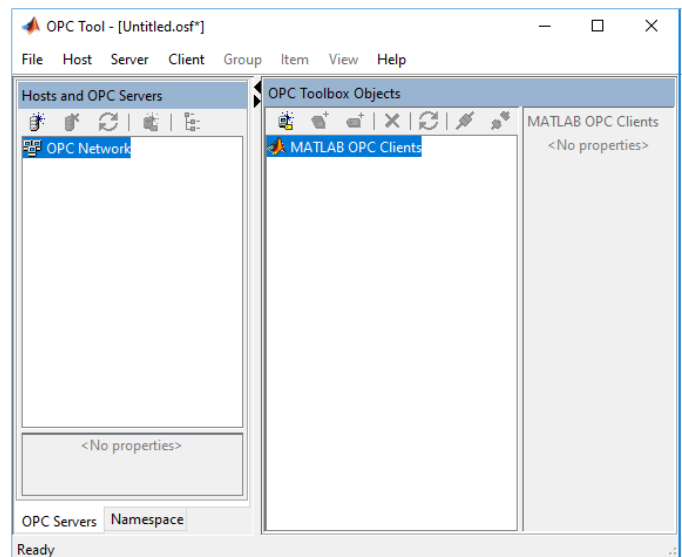


Figura 3. Ventana principal de la ToolBox: OPC Tool.

En dicha ventana se puede establecer conexión al servidor OPC, mediante la configuración del cliente y posteriormente,

a manera de prueba, se pueden leer o escribir TAGS, sin mayor inconveniente, sin embargo, el potencial de esta herramienta se puede obtener mediante el uso de scripts. A continuación, se ilustra un ejemplo de conexión, lectura y escritura a dos variables.

Código

```
canal=opcda('localhost','RSLinx OPC
Server');
connect(canal);
grp=addgroup(canal);
in=additem(grp,'[TOPICO]Entrada');
out=additem(grp,'[TOPICO]Salida');
lectura=read(in);
write(out,50);
```

Esta rutina muestra la configuración de dos TAGS en el entorno de MATLAB.

III. OBTENCIÓN DEL MODELO DE LA PLANTA

Se implementó el desarrollo mediante una planta didáctica (ver figura 4) de control de procesos que dispone de actuadores y transmisores que trabajan en el estándar 4-20mA.

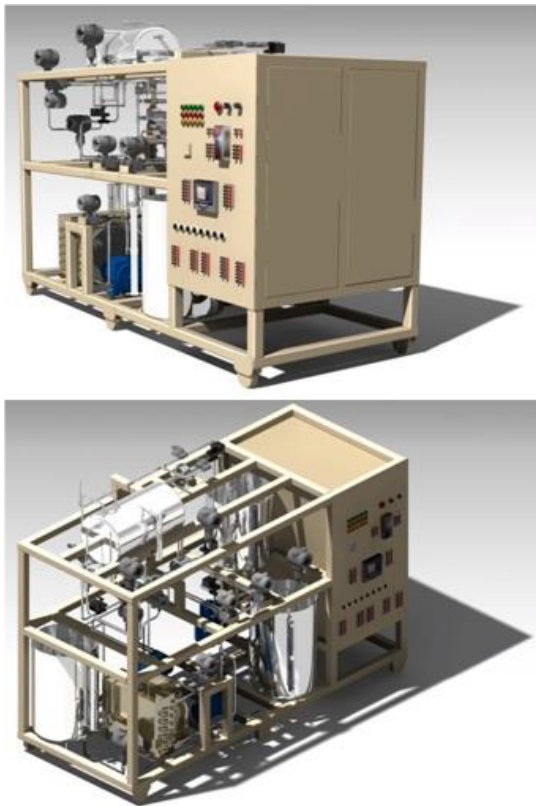


Figura 4. Planta didáctica remota para control de procesos.

Esta planta posee un proceso a lazo cerrado, donde se

emula mediante dos tanques, un procedimiento de mezcla o llenado. Para efectos didácticos, se trabaja con agua. (Victoria, C. Mantilla, W, y Gutiérrez W – 2005).

Los dispositivos a tener en cuenta son:

Transmisores:

FT100 (A) Transmisor de flujo

LT101 (A) Transmisor de nivel

Actuadores:

LCV100 (A) Válvula proporcional

LAV102 (A) Válvula proporcional

P100 (D) Bomba

A=Análogo D=Digital.

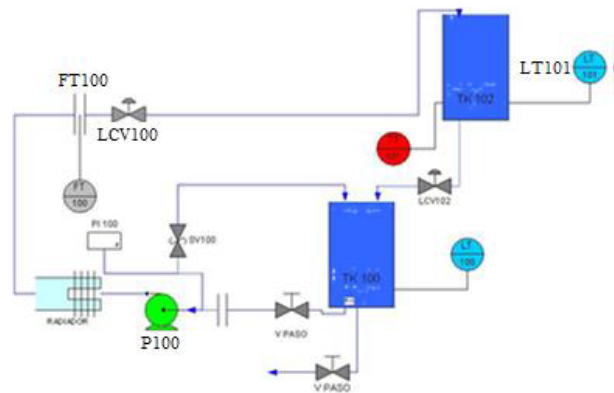


Figura 5. Diagrama P&ID de la planta.

Teniendo en cuenta lo anterior, se procede a realizar la programación del PLC. En la figura 6, se ilustran 4 líneas de programación mediante lenguaje Ladder. La primera línea corresponde a la activación de la bomba P100; la segunda corresponde a la lectura de los transmisores, la tercera a la escritura sobre los actuadores y la última, captura el valor del SetPoint y lo guarda sobre el bloque PID. Cabe destacar que la válvula LCV102 permite simular una carga al sistema, permitiendo generar perturbaciones y así determinar el comportamiento del sistema de control.

Es importante aclarar que los módulos análogos I/O, previamente han sido configurados para trabajar con valores escalados de 0% como valor mínimo, hasta 100% como valor máximo.

**El software de programación del PLC es RSLogix5000 (v16).*

La figura 7 ilustra la programación del bloque PID. Cabe destacar que el PLC que se está trabajando (AllenBradley ControlLogix L55) posee este módulo y viene pre-configurado, por tanto, es importante revisar las variables de control que, para este caso, son:

Process Variable: FT100.

Control Variable: LCV100.

En dicho bloque debe, además configurarse otros parámetros como escalas, saturaciones, tiempo de muestreo, tipo de control, entre otros.

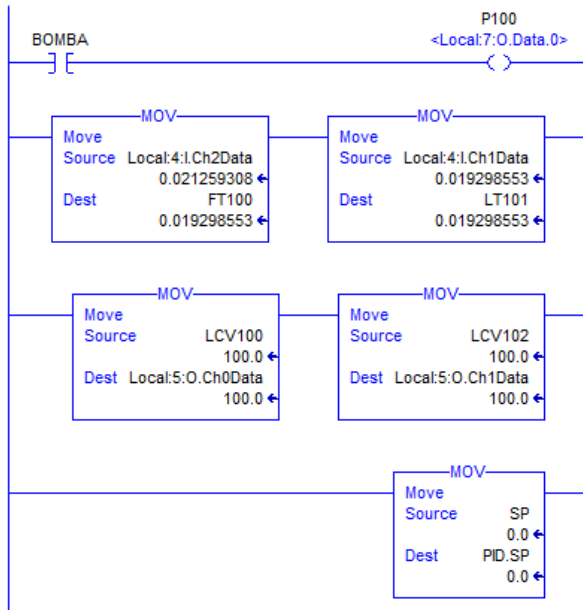


Figura 6. Programación Ladder del lazo principal.

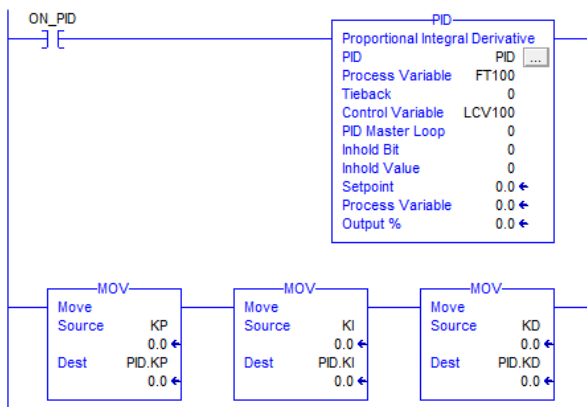


Figura 7. Programación Ladder del lazo PID.

El PID a trabajar es de tipo Independiente-Paralelo con un tiempo de muestreo a es de 100mS.

Una vez ejecutada la programación sobre el PLC, se configura sobre el software RSLinx (v2.51) el servidor OPC. Para esto, se crea el Tópico, que en el presente caso, se ha llamado PLC_REMOTO.

La figura 8 muestra cómo debería quedar establecido el tópic. Para llegar a él, dentro del software RSLinx se debe escoger la CPU, se oprime clic derecho y se selecciona la opción: *Configure new DDE/OPC Topic*.

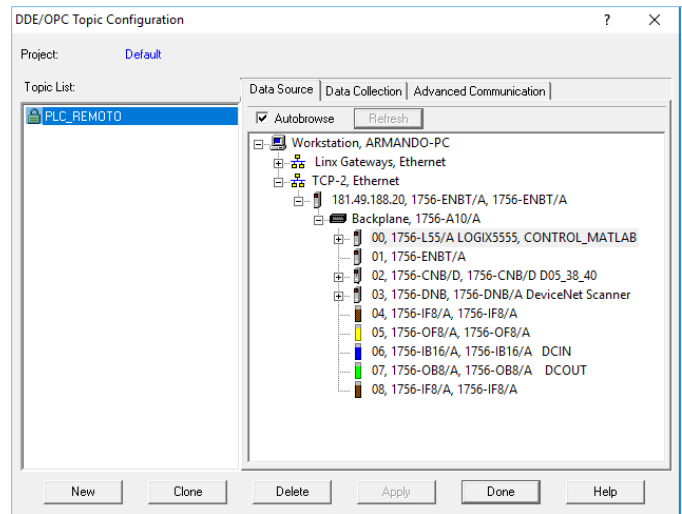


Figura 8. Configuración del Tópico OPC.

Para obtener el modelo de la planta, se aplica un escalón de la forma:

$$u(t) = \begin{cases} 0 & t < 0 \\ 50 & t \geq 0 \end{cases}$$

El tiempo de toma de datos se estableció a 50 segundos, espacio necesario para obtener un buen resultado.

Con todo lo anterior, se procede a enlazar el software MATLAB y se ejecuta una rutina que realiza la siguiente secuencia:

1. Configuración el objeto OPC.
2. Creación de los TAGS.
3. Se enciende la Bomba.
4. Se aplica un escalón sobre la válvula LCV100.
5. Se inicia la captura de los datos durante 50s.
6. Se apaga la Bomba.
7. Se cierra la válvula LCV100.

Con lo anterior, se grafican los datos obtenidos de la variación del flujo (ver figura 9).

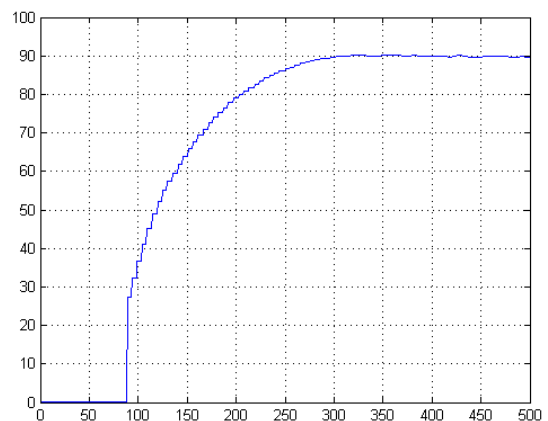


Figura 9. Respuesta al escalón.

Se observa que se han tomado 500 muestras, y esto debido a que el tiempo de captura de datos corresponde a 50 segundos, pero el muestreo es de 0.1 segundos, por tanto, por cada segundo, se obtienen 10 muestras. Otro factor importante al visualizar la respuesta, es que la planta presenta un retardo, y esto debe tenerse en cuenta al momento de realizar la identificación de la planta.

Finalmente, se estima la función de transferencia a través del comando: `tfest`. Este comando recibe como parámetros, la información obtenida, el número de polos, número de ceros y el retardo calculado. Ejemplo:

```

Código

data = iddata (FLUJO,ESCALON, muestreo );
polos=1;
ceros=0;
sistema=tfest (data , polos , ceros , retardo );
    
```

De esta manera, se obtiene que la función de transferencia estimada es:

$$\frac{0,2848}{s + 0,1567} e^{-8,9}$$

Fit to estimation data: 80.39 %

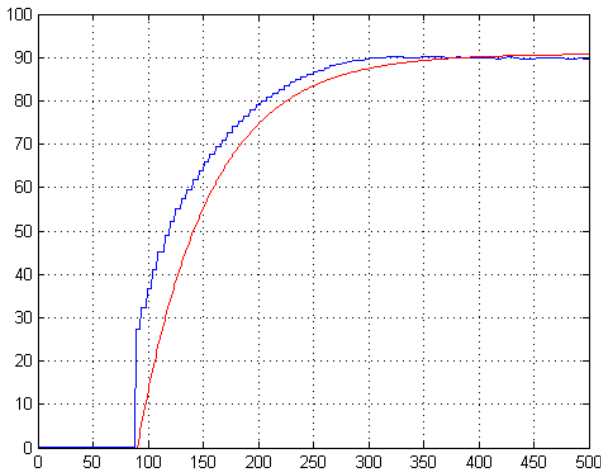


Figura 10. Comparación entre FDT estimada y respuesta real del sistema. (Azul: real – Rojo: estimada)

Se procede a realizar otra estimación, esta vez con 1 polo y 1 cero y se obtiene:

$$\frac{PV(s)}{MV(s)} = \frac{0,524s + 0,2848}{s + 0,1567} e^{-8,9}$$

Este último resultado se acerca mucho más a la respuesta real obtenida, con un ajuste de estimación de 95.67 %
Fit to estimation data: 95.67 %

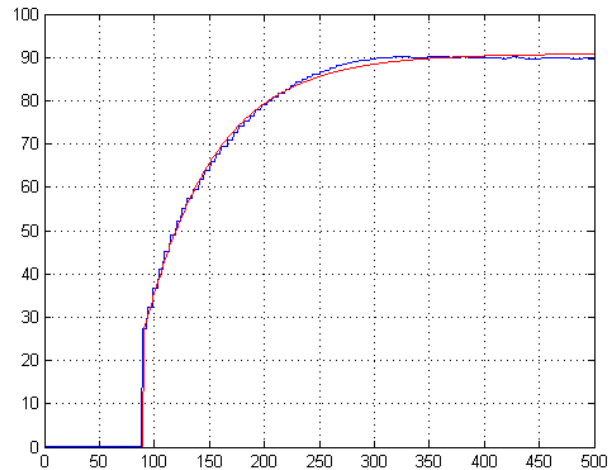


Figura 11. Comparación entre FDT estimada y respuesta real del sistema. (Azul: real – Rojo: estimada)

IV. DISEÑO DEL CONTROLADOR

Se toma como referencia el formato del controlador PID paralelo-independiente como lo ilustra la figura 12. Cabe destacar que el uso óptimo de este tipo de controladores está en las plantas cuyos modelos son lineales, o altamente linealizables.

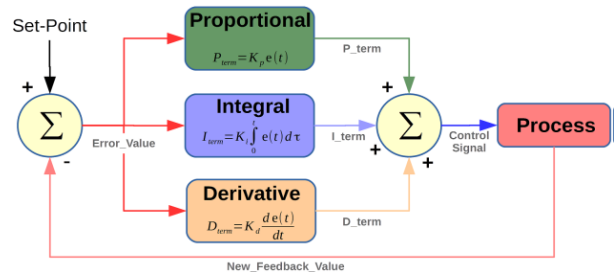


Figura 12. Anatomía del controlador PID.

La función de transferencia de este controlador es la siguiente:

$$\frac{PID(s)}{E(s)} = \frac{Kds + Kps + Ki}{s}$$

Tomando como referencia la ecuación 3, la función de transferencia del sistema resulta ser la realimentación de estas dos funciones, quedando de la siguiente manera:

$$\frac{F(s)}{U(s)} = \frac{\frac{0,524s+0,2848}{s+0,1567} e^{-8,9s} * \frac{Kds+Kps+Ki}{s}}{1 + \frac{0,524s+0,2848}{s+0,1567} e^{-8,9s} * \frac{Kds+Kps+Ki}{s}}$$

$$P = E(t) + 5 * 3 + 2$$

Siendo:

F(s) Salida del sistema = Flujo.

U(s) Entrada del sistema = SetPoint.

Este controlador se puede estimar con MATLAB a través de la herramienta PID Tuner (en algunas versiones, se puede encontrar como pidTuner o pidtool), de la siguiente manera:

```
>>[PID Info] = pidtune(sistema,'PID');
```

Esto arroja como resultado, la función de transferencia del PID con sus constantes K_p , K_i y K_d , debidamente sintonizadas. Es importante aclarar que los parámetros como tiempo de respuesta, frecuencia de cruce y margen de fase, pueden ser modificados y son visualizados de la propiedad *Info*.

Con lo anterior, se obtiene como resultado:

$K_p=0.3002$

$K_i=0.0382$

$K_d=0.3138$

Se simula el controlador y se grafica el resultado, ilustrado en la figura 13.

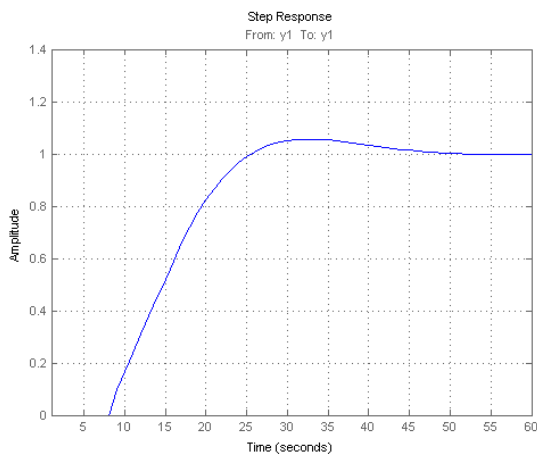


Figura 13. Respuesta del sistema controlado.

V. RESULTADOS

Se realizaron dos pruebas de validación; la primera a modo de simulación bajo el software MATLAB, con la cual se establecen diferentes frecuencias de corte como se ilustra en la figura 14.

Finalmente se selecciona la respuesta ilustrada en color verde (ver figura 14) cuyo valor w_c es 0.075 rad.

Para una frecuencia de corte de $w_c=0.075$ rad, se obtienen las siguientes constantes.

$$K_p = 0.34, K_i = 0.0401, K_d = 0.377$$

Estas a su vez, son enviadas desde MATLAB vía OPC al PLC (como se manifiesta en el apartado 2.1), y se realiza una segunda prueba, donde se valida el funcionamiento real de la planta, obteniendo un resultado como se observa en la figura 15, donde se ilustra en color rojo, el establecimiento del SetPoint; en fucsia el esfuerzo de control, es decir, la manipulación de la válvula LCV100; y en azul, la respuesta

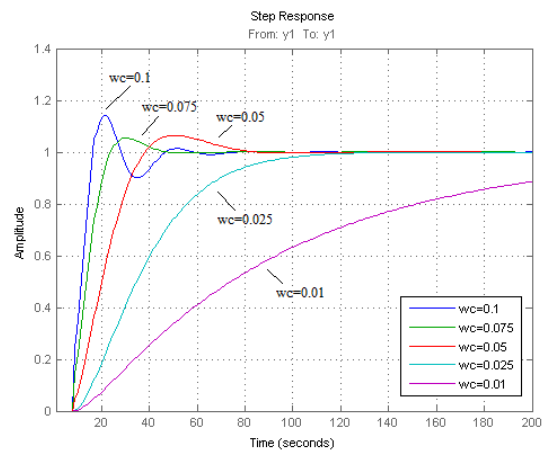


Figura 14. Respuesta del sistema controlado ante diferentes w_c .

de la planta o variable de proceso, es decir, el transmisor de flujo FT100.

Un detalle a considerar, es que el esfuerzo de control presenta cambios bruscos, y esto es debido al efecto de la acción derivativa, en el momento donde se aplica un escalón en el SetPoint y durante la etapa transicional.

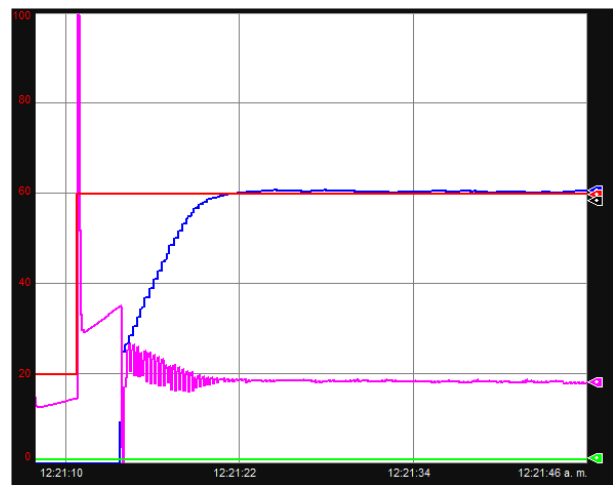


Figura 15. Respuesta de la planta.
SP=Rojo=50 % - PV=Azul - MV=Fucsia

En esta prueba, el escalón se considera para un valor inicial del 20 % y un valor final del 60 %.

Con lo anterior, se abre la puerta a la posibilidad de calcular controladores de tipo adaptativo, tomando como base, la idea de obtener al instante, datos del comportamiento de la planta, de tal modo que a medida que se capturan los datos, se podría aprovechar el potencial de MATLAB para calcular en tiempo real, los parámetros del controlador.

VI. CONCLUSIONES

Mediante el estándar OPC, se consiguió enlazar el software MATLAB a un autómatas programable, permitiendo enviar y

recibir datos en tiempo real, visualizados en el PLC a través del software de programación RSLogix5000.

Se logró obtener la función de transferencia de la planta, mediante la estimación a través de MATLAB tomando como referencia los datos obtenidos en tiempo real vía OPC.

Se desarrolló un script que permitió realizar el proceso de enlace al PLC, identificación del modelo de la planta, estimación del controlador PID, envío de las constantes proporcional, integral y derivativa al PLC y puesta en marcha de un ensayo. Todo esto sin ninguna intervención y de manera automática.

REFERENCIAS

- [1] William Gutiérrez, *Estudio de diferentes técnicas de inteligencia computacional para sintonía de acciones de control PID*, Tesis para optar al grado de Magister en Ingeniería con énfasis en Automática, Santiago de Cali, Colombia: Abril, 2009.
- [2] Miguel Mato, Alberto Herreros, *Aplicaciones en MATLAB y Simulink para el Modelado y control del movimiento de una Estacion ABB IRB-120*, Valladolid, España, 2014.
Disponible en: http://www.ja2014.upv.es/wp-content/uploads/papers/paper_32.pdf
- [3] César Reyes, *Control de la planta de los cuatro tanques mediante la realización de una pasarela matlab-http-opc*, Sevilla, España: 2006.
Disponible en: <https://www.researchgate.net/publication/228855217>
- [4] Leonardo F. Lozano Valencia, Alberto Herreros, *Diseño, Implementación y Validación de un Controlador PID Autosintonizado*, Pereira, Colombia, 2012.
Disponible en: <http://www.scielo.org.co/pdf/teclo/n28/n28a03.pdf>
- [5] César Victoria, William Mantilla y William Gutiérrez, *Diseño e implementación de una planta didáctica para la Formación Remota en Control de Procesos*, SENA, Regional Valle, Santiago de Cali, Colombia: 2005.
- [6] Víctor M. Alfaro Ruíz, *Métodos de sintonización de controladores PID que operan como reguladores*, San José, Costa Rica, 2002.
Disponible en: http://eie.ucr.ac.cr/uploads/file/documentos/pub_inv/articulos/valfaro02B.pdf