

IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL PARA REGULAR LA VELOCIDAD Y POSICIÓN DE MOTORES INDUSTRIALES UTILIZANDO EL PROTOCOLO DE COMUNICACIÓN OPC

PRESENTA: OLMER GIOVANNY VILLAMIZAR GALVIS

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA (UNAB)

FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL PARA REGULAR LA
VELOCIDAD Y POSICIÓN DE MOTORES INDUSTRIALES UTILIZANDO EL
PROTOCOLO DE COMUNICACIÓN OPC**

PRESENTA:

OLMER GIOVANNY VILLAMIZAR GALVIS

**TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARA OPTAR POR
EL TÍTULO DE:
INGENIERO MECATRÓNICO**

DIRECTOR:

Ms. HERNANDO GONZALEZ ACEVEDO

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA (UNAB)
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA
BUCARAMANGA, COLOMBIA.**

2017

NOTA DE ACEPTACIÓN

FIRMA DE DIRECTOR DE PROYECTO



FIRMA EVALUADORES





AGRADECIMIENTOS

A Dios primero que todo. A mi familia, en especial a mi madre por todo el esfuerzo, dedicación y apoyo fundamental no solo económico sino además espiritual y moral.

A todas las personas que me colaboraron en la realización de la tesis y me animaron a seguir adelante.

A mi director de proyecto, el cual estuvo para ayudarme en todo momento y además ser un excelente docente a lo largo de esta carrera.

TABLA DE CONTENIDO

INTRODUCCION.....	10
OBJETIVOS.....	11
OBJETIVO GENERAL.....	11
OBJETIVOS ESPECIFICOS.....	11
1. COMUNICACIÓN OPC.....	12
1.1. SERVIDORES OPC.....	14
1.2. ESTRUCTURA DE COMUNICACIÓN.....	15
1.3. DISEÑO DE LA RED DE COMUNICACIÓN	17
1.4. CONFIGURACIÓN DEL SERVIDOR OPC	19
1.5. CONFIGURACIÓN SOFTWARES DE PROGRAMACIÓN	22
1.5.1. MATLAB.....	22
1.5.2. LABVIEW.....	23
2. DISEÑO DE SISTEMAS DE CONTROL.....	24
2.1. ESTRATEGIAS DE CONTROL.....	24
2.1.1. CONTROL PID	24
2.1.2. CONTROL ÓPTIMO (LQG)	25
2.1.3. CONTROL FUZZY	27
2.2. MODELADO DINÁMICO DE LOS SISTEMAS	29
2.2.1. CONTROL DE VELOCIDAD DE UN MOTOR AC	29
2.2.1.1. MODELO DEL SISTEMA.....	29
2.2.1.2. DISEÑO DEL CONTROLADOR PID.....	37
2.2.1.3. DISEÑO DEL CONTROLADOR LQG	40
2.2.1.4. DISEÑO DEL CONTROLADOR FUZZY	45
2.2.2. CONTROL DE POSICIÓN.....	49
2.2.2.1. MODELO DEL SISTEMA.....	49
2.2.2.2. DISEÑO DEL CONTROLADOR PD.....	54
2.2.2.3. DISEÑO DEL CONTROLADOR LQG	56
2.2.2.4. DISEÑO DEL CONTROLADOR FUZZY	57
3. VALIDACIÓN EXPERIMENTAL	60

3.1.	INTERFACES HMI.....	60
3.1.1.	MATLAB.....	60
3.1.2.	LABVIEW.....	63
3.2.	VALIDACION DE LAS ESTRATEGIAS DE CONTROL.....	65
3.2.1.	CONTROL DE VELOCIDAD.....	65
3.2.1.1.	CONTROL PID	65
3.2.1.2.	CONTROL LQG	66
3.2.1.3.	CONTROL FUZZY.....	67
3.2.1.4.	INDICES DE ERROR	68
3.2.2.	CONTROL DE POSICIÓN.....	71
3.2.2.1.	CONTROL PD	71
3.2.2.2.	CONTROL LQG	72
3.2.2.3.	CONTROL FUZZY.....	73
3.2.2.4.	INDICES DE ERROR	74
	CONCLUSIONES Y RECOMENDACIONES.....	76
	BIBLIOGRAFIA.....	78
	ANEXO 1. CONFIGURACION SERVIDOR OPC – SIEMENS.....	79
	ANEXO 2. CONFIGURACION SERVIDOR OPC – ALLEN BRADLEY	85
	ANEXO 3 CONFIGURACION SERVIDOR OPC – FESTO	92
	ANEXO 4 CONFIGURACION DEL CLIENTE OPC – MATLAB.....	97
	ANEXO 5 CONFIGURACION DEL CLIENTE OPC – LABVIEW.....	100
	ANEXO 6 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 1	104
	ANEXO 7 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 2	106
	ANEXO 8 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 3	109
	ANEXO 9 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 4	112
	ANEXO 10 GUÍA LABORATORIO DE CONTROL # 1	115
	ANEXO 11 GUÍA LABORATORIO DE CONTROL # 2	122
	ANEXO 12 GUÍA LABORATORIO DE CONTROL # 3	127
	ANEXO 13 GUÍA LABORATORIO DE CONTROL # 4	133

LISTA DE FIGURAS

<i>Figura 1 Red heterogénea de sistemas comunicados por un cable Ethernet.</i>	12
<i>Figura 2 Estructura de la comunicación por OPC.</i>	13
<i>Figura 3 Ejemplo de comunicación OPC con un sistema SCADA.</i>	13
<i>Figura 4 Esquema de comunicaciones detalladas de la red.</i>	15
<i>Figura 5 Interface Software KEPServerEX.</i>	20
<i>Figura 6 Esquema de comunicación entre cliente/servidor detallada.</i>	21
<i>Figura 7 Bloques de lógica Fuzzy.</i>	27
<i>Figura 8 Reglas sistema Mamdani.</i>	28
<i>Figura 9 Diagrama de bloques de la estrategia de control.</i>	28
<i>Figura 10 Variador de frecuencia - motor AC.</i>	29
<i>Figura 11 Modelo del variador de frecuencia.</i>	30
<i>Figura 12 Señal de alimentación en componentes 'abc'.</i>	31
<i>Figura 13 Modelo caja blanca vs datos experimentales.</i>	32
<i>Figura 14 Linealidad del motor ante diferentes frecuencias de alimentación.</i>	33
<i>Figura 15 Velocidad del motor ante diferentes frecuencias de alimentación.</i>	34
<i>Figura 16 Toolbox Ident – Funciones con mejor best fit para representar la planta del sistema de control de velocidad.</i>	35
<i>Figura 17 Respuesta de la planta discreta ante una entrada escalón.</i>	36
<i>Figura 18 Modelo caja negra vs datos experimentales.</i>	36
<i>Figura 19 Diagrama de polos y ceros del controlador PID.</i>	37
<i>Figura 20 Diagrama de bloques controlador PID.</i>	38
<i>Figura 21 Respuesta de la planta con controlador PID.</i>	39
<i>Figura 22 Diagrama de bloques controlador LQG.</i>	40
<i>Figura 23 Subsistema correspondiente a la estimación de los estados.</i>	41
<i>Figura 24 Respuesta del sistema ante el controlador LQG.</i>	44
<i>Figura 25 Conjunto difuso correspondiente al error.</i>	45
<i>Figura 26 Conjunto difuso correspondiente a la derivada del error.</i>	45
<i>Figura 27 Conjunto difuso correspondiente a la salida del controlador.</i>	46
<i>Figura 28 Diagrama de bloques del controlador Fuzzy tipo Mamdani.</i>	47
<i>Figura 29 Superficie de control difuso.</i>	47
<i>Figura 30 Respuesta del sistema aplicando un controlador Fuzzy.</i>	48
<i>Figura 31 Modelo de barra rígida.</i>	49
<i>Figura 32 Modelo caja blanca vs datos experimentales.</i>	50
<i>Figura 33 Respuesta del sistema en lazo abierto para identificación.</i>	51
<i>Figura 34 Toolbox Ident – Funciones con mejor best fit para representar la planta del sistema de control de posición.</i>	52
<i>Figura 35 Comportamiento de la planta ante una entrada escalón.</i>	52
<i>Figura 36 Modelo caja negra vs datos experimentales.</i>	53
<i>Figura 37 Diagrama de bloques para implementar un control PD.</i>	54
<i>Figura 38 Diagrama polos y ceros control PD.</i>	55
<i>Figura 39 Respuesta del sistema con controlador PD.</i>	55

<i>Figura 40 Diagrama de bloques control LQG para planta con integrador.</i>	56
<i>Figura 41 Respuesta del comportamiento de la planta con controlador LQG.</i>	57
<i>Figura 42.-Diagrama de bloques del controlador Fuzzy tipo Mamdani para planta con integrador.</i>	58
<i>Figura 43 Superficie de control difuso</i>	58
<i>Figura 44 Respuesta del comportamiento de la planta con controlador Fuzzy.</i>	59
<i>Figura 45 Interfaz para control de velocidad del motor AC.</i>	61
<i>Figura 46 Interfaz para control de posición del servomotor industrial.</i>	61
<i>Figura 47 Interfaz HMI para control de Velocidad de un motor AC, Posición de un servomotor industrial y secuencia de MPS - Festo.</i>	63
<i>Figura 48 Datos experimentales vs datos teóricos implementando un control PID.</i>	65
<i>Figura 49 Datos experimentales vs datos teóricos implementando un control LQG.</i>	66
<i>Figura 50 Datos experimentales vs datos teóricos implementando un control Fuzzy.</i>	67
<i>Figura 51 Comparación de los controladores ante una misma entrada escalón.</i>	69
<i>Figura 52 Datos experimentales vs datos teóricos implementando un control PID.</i>	71
<i>Figura 53 Datos experimentales vs datos teóricos implementando un control LQG.</i>	72
<i>Figura 54 Datos experimentales vs datos teóricos implementando un control Fuzzy.</i>	73
<i>Figura 55 Comparación de los controladores ante una misma entrada escalón.</i>	74

LISTA DE TABLAS

<i>Tabla 1</i> Direccionamiento de red de los dispositivos	18
<i>Tabla 2</i> Comparación de servidores OPC.	19
<i>Tabla 3</i> Reglas establecidas para el controlador Fuzzy.	46
<i>Tabla 4</i> Índices de error - Control de velocidad.	69
<i>Tabla 5</i> Índices de error - Control de posición.	75

INTRODUCCION

En la actualidad las comunicaciones industriales son una base fundamental en las tareas de automatización en las compañías, incluso han ido siendo aplicadas para el control de procesos, se puede decir que la comunicación industrial es actualmente un elemento indispensable en la industria. Cuando se habla de comunicación industrial en la automatización de procesos se encuentra que hay diferentes maneras de realizar una comunicación dada la cantidad de protocolos de comunicación existentes, dado esto suelen suceder conflictos al poseer diferentes tipos de tecnologías, ya que estas suelen tener protocolos de comunicaciones diferentes.

Existe un estándar de comunicación llamado OPC (OLE 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 una transmisión de datos de forma más sencilla. Una de las funciones de la comunicación por OPC actualmente es la solución al problema de interacción entre softwares y autómatas programables.

El objetivo de este proyecto es implementar este tipo de comunicación (OPC) entre los softwares Matlab y LabView para comunicarlos con dos PLC (Siemens y Allen Bradley) que posee la UNAB, con el fin de aplicar diferentes controladores (PID, Fuzzy, LQG) a un variador de frecuencia (Siemens) acoplado a un generador y a un servomotor industrial (Rockwell) acoplado a una barra y de esta forma comparar y analizar el tiempo de respuesta que se obtenga de cada enlace de comunicación, así mismo analizar las ventajas de la comunicación OPC.

OBJETIVOS

OBJETIVO GENERAL

Implementar un sistema de control clásico y avanzado, para regular la velocidad y posición de motores industriales utilizando el protocolo de comunicación OPC.

OBJETIVOS ESPECIFICOS

- Establecer una comunicación OPC entre los softwares de programación, Matlab y LabView, y los controladores lógicos programables, marca Siemens, para regular la velocidad de un motor trifásico acoplado a un generador.
- Establecer una comunicación OPC entre los softwares de programación, Matlab y LabView, y el controlador lógico programable, marca Allen-Bradley, para realizar el control de posición de una barra acoplada a un servomotor industrial de marca Rockwell.
- Determinar el modelo de cada uno de los sistemas (Motor trifásico acoplado a un generador y un servomotor industrial acoplado a una barra).
- Diseñar los sistemas de control PID, Fuzzy y LQG para cada uno de los sistemas.

1. COMUNICACIÓN OPC

El estándar OPC (Object Linking and Embedding for Process Control) surgió a mediados de los años 90 como una tecnología para la comunicación del software SCADA y otros HMI con los dispositivos generales de control en el ámbito de la industria. OPC es un mecanismo estándar de comunicación, interconecta en forma libre numerosas fuentes de datos, donde se incluyen dispositivos de planta en la fábrica (PLC's, Variadores de Frecuencia, DCS, Drivers, etc), o una estación de datos en un cuarto de control. En la Figura 1 observamos el ejemplo de una red heterogénea que integra diferentes sistemas comunicados por OPC, en esta se puede analizar que no importa el tipo de sistema o dispositivo, se puede conectar a esta red con la ayuda de servidores OPC.

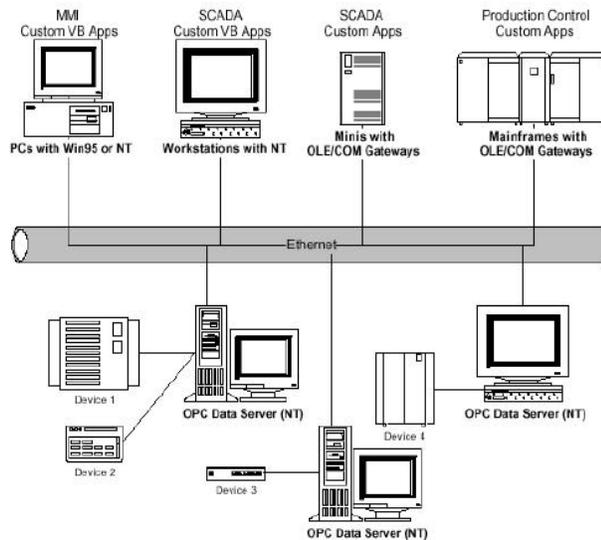


Figura 1 Red heterogénea de sistemas comunicados por un cable Ethernet. Fuente: Wikipedia, comunicación OPC

Podemos observar la estructura principal de la comunicación OPC en la figura 2, en esta se establece una línea de comunicación entre los proveedores de Hardware y

los diseñadores de Software, provee un mecanismo para registrar datos de una fuente de información y comunicarla a cualquier cliente, en forma transparente.

Un fabricante puede desarrollar un servidor que se comunique con la fuente de datos, y mantenga a su vez una forma para acceder a la fuente de datos y dispositivos de manera eficaz.

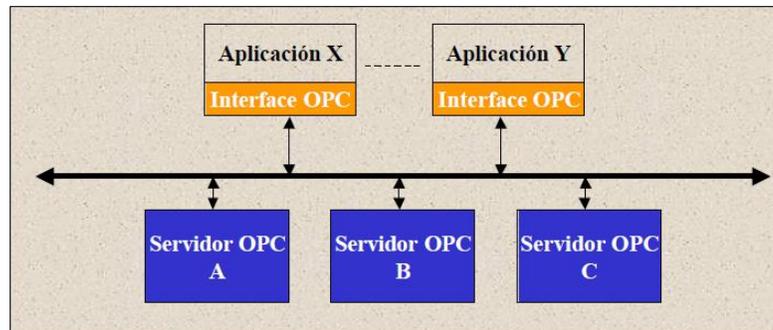


Figura 2 Estructura de la comunicación por OPC. Fuente: OPC: Un estándar en las redes industriales y buses de campo

OPC proporciona una interface entre el servidor y cualquier cliente, para así poder acceder a los dispositivos requeridos. En la Figura 3, podemos observar un ejemplo de una aplicación en la cual se integran por medio de OPC sistemas físicos externos al sistema principal el cual corresponde a un sistema SCADA.

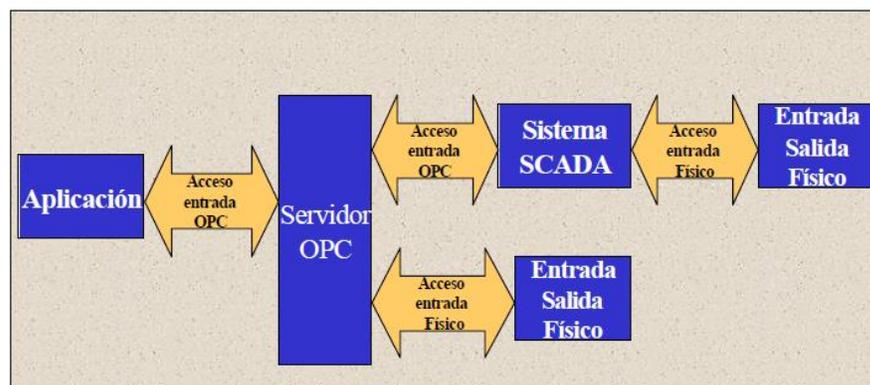


Figura 3 Ejemplo de comunicación OPC con un sistema SCADA. Fuente: OPC: Un estándar en las redes industriales y buses de campo

1.1. SERVIDORES OPC

El servidor OPC es la fuente de datos (como un dispositivo hardware a nivel de planta), cualquier aplicación basada en OPC puede acceder a dicho servidor para leer/escribir cualquier variable que ofrezca el servidor. Un dispositivo (cliente) puede conectarse por medio de una red a servidores OPC, de esta forma no existe restricción por cuanto a tener un Software cliente para un Software servidor, este es uno de los principales problemas que se tiene en la comunicación entre sistemas de diferentes fabricantes, los cuales suelen tener diferentes protocolos de comunicación o tener restricción de solo comunicarse con dispositivos de la misma marca. Los fabricantes, a su vez, proporcionan el código que identifica: Dispositivos, tipos de datos a los que cada servidor tiene acceso, valor de los datos, y detalles sobre cómo el servidor físicamente da acceso a los datos. Sin estos códigos Servidores y Clientes no podrían comunicarse y reconocerse como sistemas compatibles. También es posible que otros sistemas como lo son SCADA o DCS puedan comunicarse con un servidor OPC y llevar su información recopilada desde un banco de datos o dispositivos físicos como lo son del tipo PLC. Así de esta forma aplicaciones cliente OPC de otros fabricantes tendrán acceso a estos datos por medio del Servidor.

Existen diferentes tipos de servidores OPC, uno de estos es el servidor de acceso a datos OPC, este se compone de varios objetos: Servidor, Grupo e Ítem. La función de este servidor OPC, es mantener la información sobre sí mismo y hacer las veces de un “recipiente” unificando los datos en un Grupo. La función del Grupo OPC es mantener la información y proporcionar un mecanismo por contener y organizar lógicamente los Ítems.

Los Grupos OPC proveen a los clientes OPC, quienes ejecutan aplicaciones, una forma de organizar sus datos. Otro tipo de servidor es el servidor de alarmas y eventos. Estas interfaces proveen de mecanismos a los Clientes OPC, con los cuales pueden ser notificados de la ocurrencia de eventos y condiciones de alarmas

específicas. Estas también proporcionan servicios que les permiten a los Clientes OPC determinar eventos y condiciones necesarias para alarmas, eventos, y para obtener su estado actual, todo ello apoyado por un Servidor OPC.

1.2. ESTRUCTURA DE COMUNICACIÓN

Es necesario plantear un esquema inicial en el cual se determinen los equipos que van a pertenecer a esta red de comunicación la cual se creará para suplir la necesitada principal que es una transmisión bidireccional de datos por parte de la estación de trabajo con los dos controladores lógicos programables que se manejaran. Se reconoce la Estación de trabajo (PC) como el dispositivo el cual va a recibir los datos de los PLC's para así poder realizar ya sea el caso un control de velocidad o un control de posición o ambos simultáneamente.

Luego de realizar las configuraciones de red propias de cada dispositivo, se obtiene un esquema detallado como el que se observa en la figura 4.

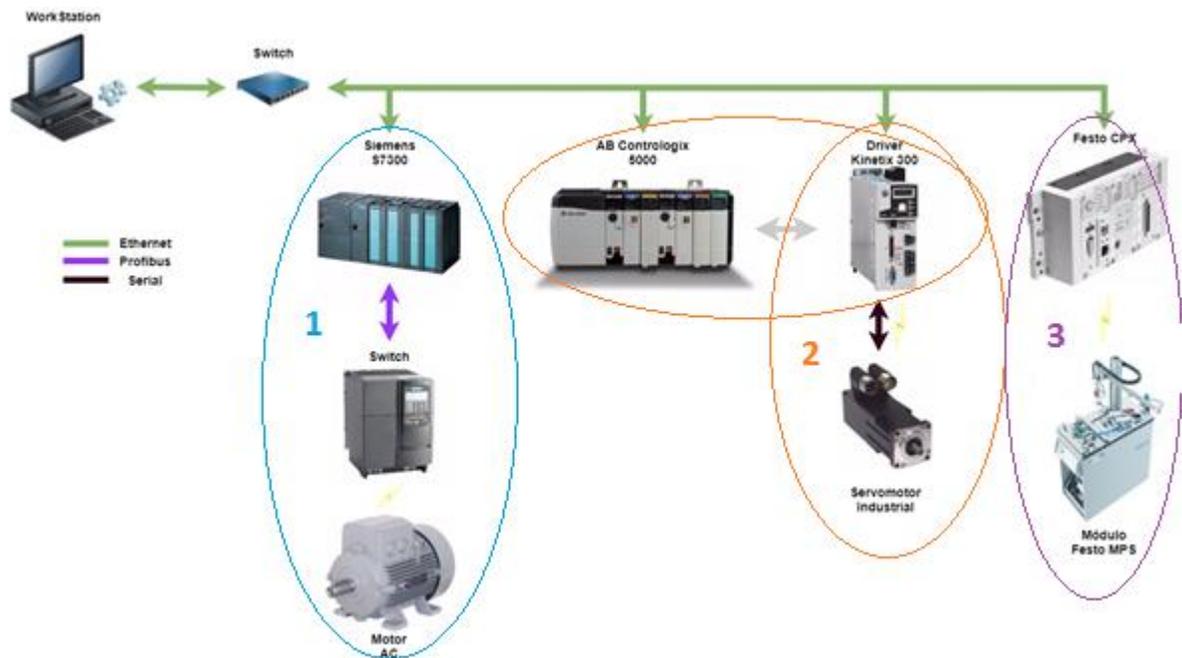


Figura 4 Esquema de comunicaciones detalladas de la red.

En este esquema se observan tres subgrupos los cuales corresponden a las dos variables a controlar planteadas en el proyecto y al módulo MPS agregado al sistema.

El primer subgrupo corresponde al objetivo de control de velocidad, en este caso se usa 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, este dispositivo tiene la función de realizar la variación de frecuencia de alimentación de un motor AC. Esta variación de frecuencia afecta la velocidad del motor, al tener la frecuencia máxima (60Hz) el motor por ende tiene su velocidad máxima (1780 rpm). En este caso se define entonces la velocidad del motor como la variable a controlar mediante la variación de frecuencia de alimentación.

El segundo subgrupo se establece con la finalidad de realizar el control de posición de un servomotor industrial usando para esto un PLC ControlLogix 1765 marca Allen Bradley. Este PLC se encarga de comunicarse vía Ethernet con un driver controlador Kinetix 300, este dispositivo se usa para realizar la configuración y el manejo del servomotor industrial. El Kinetix 300 permite establecer el servomotor en tres diferentes condiciones:

- Control de corriente
- Control de velocidad
- Control de posición

Para cumplir el objetivo del proyecto se estableció la condición de control de velocidad, y así, poder realizar desde los softwares de programación el respectivo controlador de posición. En este caso la variable de control es la posición del servomotor mediante la velocidad. El tercer subgrupo consiste en un módulo conocido como MPS conectado a un PLC Festo CPX, este módulo MPS consiste en proceso de identificación de una pieza, y una pinza la cual se encarga de dirigir este objeto identificado a un carril de almacenamiento. El PLC Festo se comunica vía Ethernet a través del servidor OPC en el cual ya están establecidas las variables para realizar el intercambio de datos.

1.3. DISEÑO DE LA RED DE COMUNICACIÓN

Para el diseño de la red de comunicación, se tuvo en cuenta algunas características de la red para poder realizar el direccionamiento. En este proyecto no se tienen requerimientos específicos sobre la cantidad de equipos o de redes necesarias y la comunicación se realiza por medio de una red Ethernet (topología de red tipo estrella) [6]. Se estableció una dirección IP de red "Clase C" dado que uno de los equipos usados (Driver Kinetix 300) tenía anteriormente establecida una dirección IP la cual requería de un software para poder modificarla, este software (BOOTP/DHCP) tenía dificultades al reconocer el driver por lo tanto se decidió dejar la dirección IP que poseía el equipo, **192.168.124.81** y se trabajó el proyecto alrededor de esta dirección de red.

Las direcciones IP Clase C normalmente se asignan a un tamaño muy pequeño red ya que sólo puede tener 254 hosts en una red, pero permite tener mayor cantidad de subredes. El primer octeto de IP de Clase C tiene sus primeros 3 bits a 110, es decir:

$$\begin{array}{r} 11000000 \text{ -- } 11011111 \\ 192 \text{ -- } 223 \end{array}$$

Las direcciones IP Clase C se encuentran ubicadas de 192.0.0.x a 192.255.255.x. La máscara de subred predeterminada de la Clase C es 255.255.255.x.

La red de Clase C posee:

- 2097152 - (2^{21}) direcciones de red.
- 254 - ($2^8 - 2$) las direcciones de host.

En este caso se establecieron las direcciones IP para los dispositivos de la forma que se muestra en la tabla 1.

Tabla 1 Direccionamiento de red 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 anteriores son los cuatro dispositivos que pertenecen a la red, en este caso se eligió trabajar con la subred **192.168.124** por que como se mencionaba anteriormente se encontraba establecida en uno de los equipos y no permitía ser modificada. Para cada tecnología se dejó un rango de 20 hosts disponibles dado que se requieran adicionar equipos de la misma tecnología poder encontrarlos organizados según su marca.

1.4. CONFIGURACIÓN DEL SERVIDOR OPC

En la selección del servidor OPC que se va a usar para la comunicación entre los dispositivos es indispensable conocer las características de cada tipo de servidor que ofrece el mercado. En este caso, en la tabla 2, realizaremos una comparación entre tres servidores OPC que se encontraron para descarga libre.

Tabla 2 Comparación de servidores OPC.

SERVIDOR	VENTAJAS	DESVENTAJAS
Matrikon OPC 	<ul style="list-style-type: none"> • Permite el uso del programa continuamente sin restricción de tiempo por sesión en su versión DEMO. • Permite comunicación con Matlab y LabView. • Permite trabajar con tiempos de comunicación no menores a 0.15 segundos. 	<ul style="list-style-type: none"> • Exige la descarga de un software para cada dispositivo que va a comunicar. • Versión DEMO limitada a 30 días de uso.
KEPServerEX 	<ul style="list-style-type: none"> • Posee alta variedad de dispositivos soportados. • Permite comunicación con Matlab y LabView. • Permite trabajar con tiempos de comunicación no menores a 0.10 segundos. 	<ul style="list-style-type: none"> • Versión DEMO que permite trabajar 2 horas continuas.
NI OPC Servers 	<ul style="list-style-type: none"> • Software ya instalado en los equipos de la UNAB. • Posee alta variedad de dispositivos soportados. • Enlace directo con LabView. • Permite trabajar con tiempos de comunicación no menores a 0.15 segundos. 	<ul style="list-style-type: none"> • Versión DEMO que permite trabajar 2 horas continuas. • No permite la comunicación con Matlab.

El servidor KEPServerEX fue elegido por dos factores importantes para el proyecto, la velocidad de comunicación y la integración en un solo software de los diferentes dispositivos soportados. El software KEPServerEX es un servidor OPC que ofrece una versión demo gratuita la cual soporta los dispositivos utilizados en este proyecto, con la limitante que solo funciona durante dos horas por sesión de manejo,

al finalizar estas dos horas se requiere reiniciar la estación de trabajo (se conoce que esta limitante es solo por ser versión DEMO). Este software comparado con otros servidores que se encuentran disponibles para descarga gratuita, permitió establecer una velocidad de transmisión de datos de 0.15 segundos (siendo la menor que se encontró para un software libre). Además de esto permite establecer el enlace de comunicación sobre un mismo software (a diferencia de otros softwares que son independientes para cada una de las máquinas que se van a trabajar). En la figura 5 se muestra la interfaz de trabajo del software KEPServerEX en la cual ya se encuentran creadas las dos comunicaciones entre los PLC y la estación de trabajo.

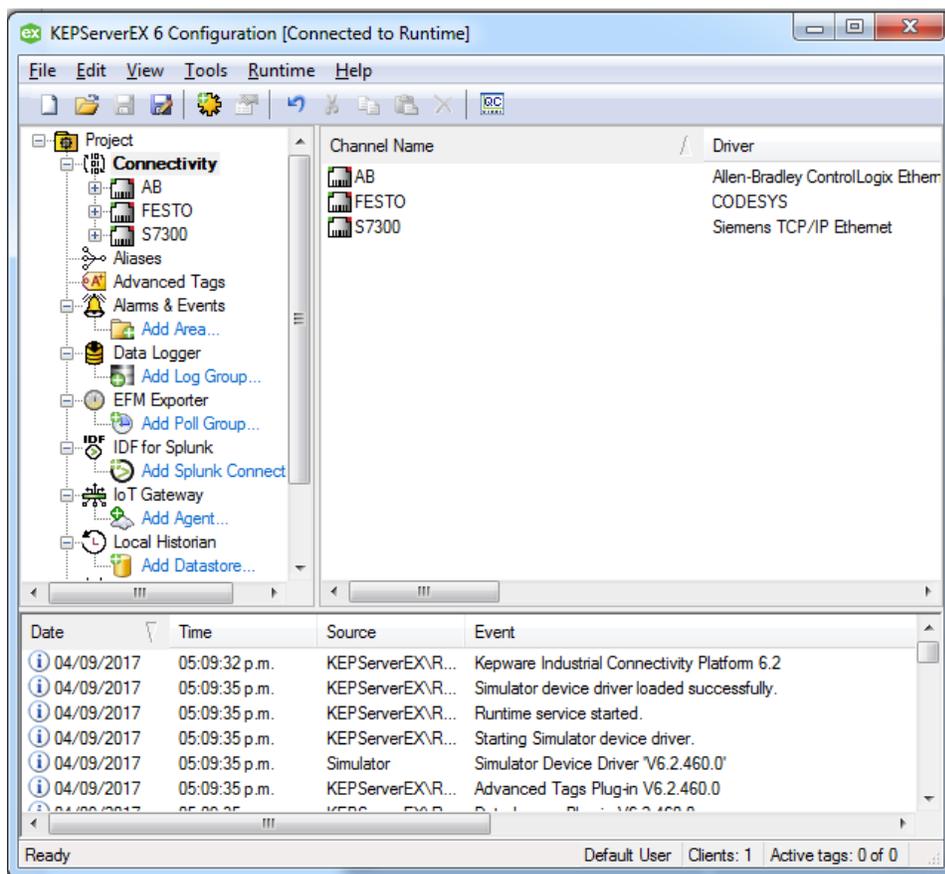


Figura 5 Interface Software KEPServerEX.

- La configuración detallada de la conectividad con el PLC Siemens se explica de forma detallada en el anexo 1.

- La configuración detallada de la conectividad con el PLC AllenBradley se explica detalladamente en el anexo 2
- La configuración detallada de la conectividad con el PLC CPX Festo se explica detalladamente en el anexo 3

Luego de tener configurado nuestro servidor OPC, se procede a la configuración de los clientes OPC, los cuales van ligados de la mano a los softwares de programación que se usaran para el desarrollo del proyecto. En la figura 6 se establece un esquema en el cual se detalla la forma de comunicar los datos de la red, explicando el proceso de conexión entre datos al servidor, y servidor a cliente. Los servidores OPC usados se encuentran en los siguientes enlaces web para descarga gratuita (versiones demo).

- Enlace de descarga del servidor OPC “KEPServerEX”:
<https://www.kepware.com/en-us/products/kepserverex> (visto en la web el día 30 de noviembre de 2017).
- Enlace de descarga del servidor OPC “Matrikon OPC servers”:
<http://www.matrikonopc.es/drivers/driver-types.aspx> (visto en la web el día 30 de noviembre de 2017).

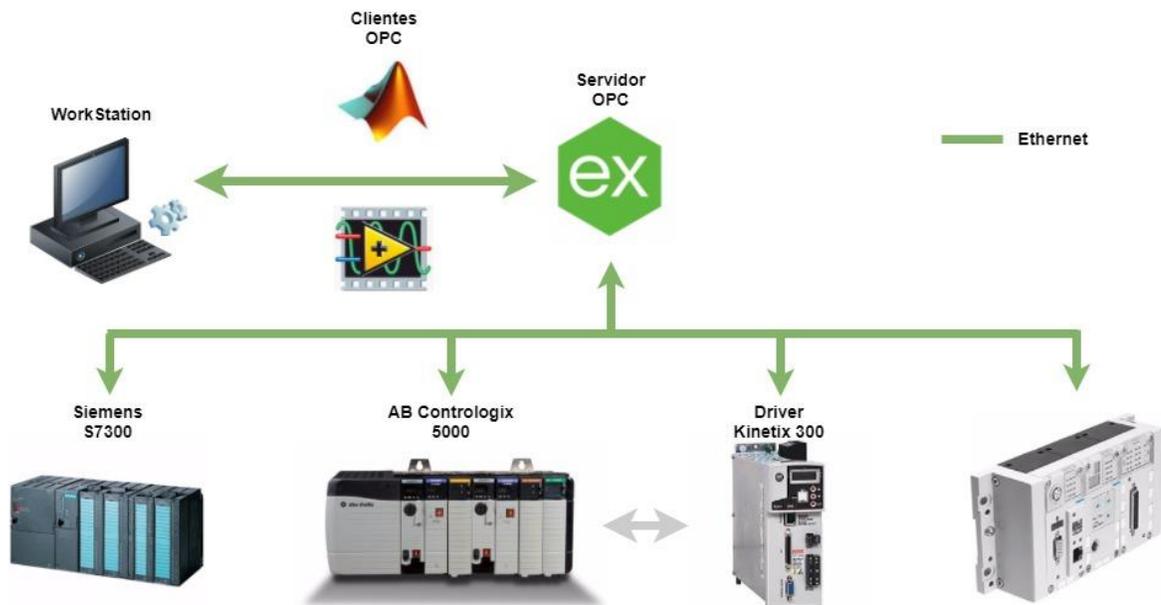


Figura 6 Esquema de comunicación entre cliente/servidor detallada.

1.5. CONFIGURACIÓN SOFTWARES DE PROGRAMACIÓN

1.5.1. MATLAB

Una de las herramientas usadas para el diseño de diferentes tipos de controladores es Matlab, esta herramienta desarrollada por MathWorks que brinda facilidad al momento de realizar cálculos matemáticos y además de esto permite trabajar con funciones de transferencia lo cual es ideal para realizar en este caso, los tres controladores propuestos para el desarrollo del proyecto (PID, óptimo, Fuzzy). Matlab, como muchos otros softwares de programación, permite realizar comunicación de datos por distintos métodos (comunicación serial, TCP/IP, OPC) usando herramientas licenciadas que ofrece este software. La relación que tiene el Cliente, en este caso Matlab, gracias a su Toolbox OPC, con el servidor, en este caso nuestro servidor seleccionado KEPServerEX, para la simulación de un proceso.

Para el desarrollo de este proyecto era requerida la herramienta “OPC TOOLBOX versión 4.0.3” la cual fue adquirida por la universidad. Esta Toolbox permite la adquisición directa de Matlab con el servidor OPC para el manejo de los datos en tiempo real [7]. En la figura 10 se muestra la interfaz la Toolbox OPC de Matlab usada para realizar la comunicación con el servidor OPC. En este caso se visualizan diferentes servidores OPC en el host local de la estación de trabajo.

- La configuración detallada de la comunicación entre Matlab y el servidor KEPServerEX se explica en el anexo 4.

1.5.2. LABVIEW

LabVIEW es un Entorno de desarrollo integrado especializado en informática industrial. Su particularidad es que se basa en el lenguaje “G” (G por Gráfico), creada por National Instruments. Permite el desarrollo de programas informáticos complejos facilitando al mismo tiempo la programación y en consecuencia disminuir los plazos de desarrollo.

El software NI LabVIEW puede comunicarse con cualquier controlador lógico programable de diferentes formas, una de estas es OPC, inicialmente se define el estándar para comunicar datos en tiempo real entre los dispositivos de control de una planta y las interfaces hombre-máquina. Los servidores OPC están disponibles virtualmente para todos los PLC's [8]. Una de las ventajas que ofrecía trabajar con el software LabVIEW era que poseía una versión demo de su propio servidor OPC (NI OPC Servers), lo cual brindaba la posibilidad de trabajarlo directamente con este servidor sin tener que instalar otro software en la Workstation, pero con fin de unificar todo, se trabajó con el mismo servidor OPC con el que se usó en el programa Matlab (KEPServerEX). En la figura 12 se muestra la interfaz que identifica los diferentes servidores OPC disponibles para configurar la comunicación con el software.

- La configuración detallada de la comunicación entre LabView y el servidor KEPServerEX se explica en el anexo 5.

2. DISEÑO DE SISTEMAS DE CONTROL

Como estrategias de control en este proyecto se propusieron tres opciones buscando analizar cuál es el mejor controlador para cada una de las plantas que se tiene, que en este caso corresponden a dos sistemas, el primer sistema para realizar el control de velocidad de un motor trifásico acoplado a un generador y un segundo sistema para realizar el control de posición de un servomotor industrial el cual tendrá acoplado una barra rígida. A continuación, se hace una introducción a cada una de las estrategias de control a aplicar y luego se explicará el desarrollo para el diseño de cada uno de los controladores para ambos sistemas.

2.1. ESTRATEGIAS DE CONTROL

2.1.1. CONTROL PID

El control PID ha sido usado 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 a ser 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) [8]. La acción de control PID en controladores digitales se representa de la forma:

$$U_k = K_p \left[e_k + \frac{T}{T_i} \sum_{i=0}^{k-1} e_i + \frac{T_d}{T} (e_k - e_{k-1}) \right] \quad (2.1)$$

Donde e_k es la entrada al controlador (señal de error), T es el periodo de muestreo, K_p 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.

$$C(z) = \left(\frac{(Kp+Ki+Kd)*z^2 - (Kp+2Kd)*z + Kd}{z(z-1)} \right) \quad (2.2)$$

Donde Kp es la ganancia proporcional, Kd es la ganancia derivativa y Ki es la ganancia integral.

2.1.2. CONTROL ÓPTIMO (LQG)

El control LQG (Linear Quadratic Gaussian) es una variante del control LQR (Linear Quadratic Regulator). La estrategia usada en este tipo de controlador consiste en considerar el sistema en estudio mediante su modelo matemático lineal y en utilizar modelos de ruido estocásticos (ruidos de medición y perturbaciones del sistema) con propiedades estadísticas conocidas [9]. La expresión del modelo lineal del sistema es escrita como:

$$x[k + 1] = Gx[k] + Hu[k] + Gw[k] \quad (2.3)$$

$$y[k] = Cx[k] + Hw[n] + v[n] \quad (2.4)$$

Donde las matrices G , H , C corresponden a las matrices de los sistemas de dimensiones apropiadas; x - son los estados del sistema; u - entradas del sistema; y - salida del sistema.

El propósito de la estrategia de control consiste en encontrar el vector óptimo de señales de control $u[k]$ tal que se minimice la función de coste:

$$J = \sum_{k=1}^{\infty} (x[k]^T Qx[k] + u[k]^T Ru[k] + 2x[k]^T Nu[k]) \quad (2.5)$$

Donde las matrices de sintonía Q y R son seleccionadas apropiadamente cumpliendo las condiciones de diseño $Q = Q^T \geq 0$ y $R = R^T > 0$. Las matrices Q y R son conocidas como matrices de penalización estado y entrada respectivamente, ellas se encargan de dar prioridad bien sea al seguimiento de una trayectoria por parte

de los estados o a la penalización de la energía de control requerida por el sistema para alcanzar los objetivos de control deseados.

La solución de problema de control LQG, es decir, hallar el $u[n]$ óptimo, consiste en la determinación de una ganancia estabilizante K , la cual se aplica al sistema mediante una ley de realimentación de estado expresada como:

$$u = -Kx[k] \quad (\text{Donde } K \text{ es la matriz de ganancia del controlador.}) \quad (2.6)$$

Por otro lado, a fin de mejorar la regulación del sistema, es posible agregar al esquema en lazo cerrado un estado dado por un integrador. Esto permite que el error de estado estacionario tienda a cero y, por lo tanto, que la salida de la planta siga más precisamente a la señal de referencia. Así, el diseño de la ley LQG implica la obtención de una ganancia K considerando la variable adicional relacionada con el integrador.

Con el fin de calcular la matriz K , es necesario disponer de la medición de todo el vector de estado, lo cual no es siempre posible. Por lo tanto, es factible diseñar un observador de estados a partir de las mediciones de las salidas de la planta. Este sistema auxiliar se encargará de estimar las trayectorias del sistema con un error y velocidad de convergencia dados mediante el parámetro de diseño K_e , el cual es conocido como la ganancia de estimación. Así pues, la ley de control es calculada como

$$u = -K\hat{x}[k] \quad (\text{Donde } \hat{x}[n] \text{ es el vector de estado estimado por el observador}) \quad (2.7)$$

Donde

$$\hat{x}[n + 1] = A\hat{x}[n - 1] + Bu[n] + L(y[n] - C\hat{x}[n - 1] - Du[n]) \quad (2.8)$$

Siendo L una matriz de respuesta resolviendo la ecuación discreta de Ricatti que da como resultado.

$$L = (APC^T + \bar{N})(CPC^T + \bar{R})^{-1} \quad (2.9)$$

2.1.3. CONTROL FUZZY

El control difuso es una alternativa para resolver complejas aplicaciones de control de una forma diferente, ya que 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 [10].

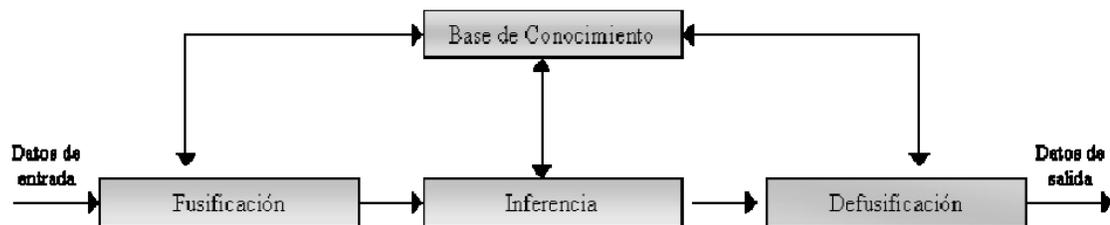


Figura 7 Bloques de lógica Fuzzy.

La figura 7 muestra el diagrama de bloques de un control difuso, el cual se compone de los siguientes elementos.

- Fusificación
- Base de conocimiento
- Inferencia
- Defusificación

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 los valores de entrada a los diferentes conjuntos difusos en los cuales se ha dividido el universo de discurso de las diferentes variables de entrada al sistema.

Luego pasa por el mecanismo de inferencia difusa. Teniendo los diferentes niveles de pertenencia arrojados por el fuzificador, los mismos 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 reglas de la base de conocimiento de un sistema Fuzzy - Mamdani corresponden a conjuntos difusos en las salidas, las cuales van a activarse según el nivel de pertenencia de la etapa de inferencia combinada con la etapa denominada base de conocimiento o base de reglas establecidas. Para establecer la base de reglase se sigue una condición básica en la cual se tienen dos entradas, y se establece una lógica de condiciones como se muestra en la figura 8

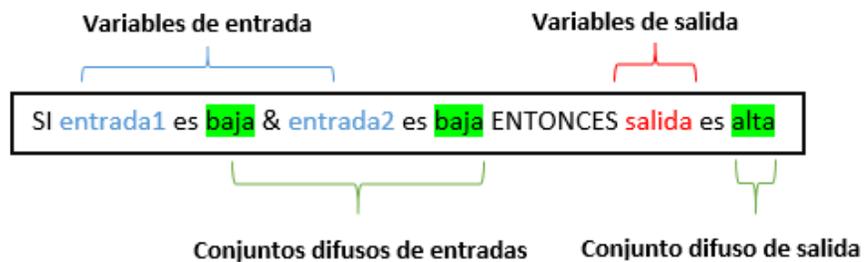


Figura 8 Reglas sistema Mamdani.

Como en este punto, el valor de salida es un valor que se encuentra en un lenguaje difuso, necesita una etapa de Defusificación para poder convertirse en un valor numérico interpretable. En la figura 9 se implementa un diagrama de bloques de un control difuso, en este, se emula al controlador PD, en este caso usando un integrador a la salida del controlador Fuzzy.

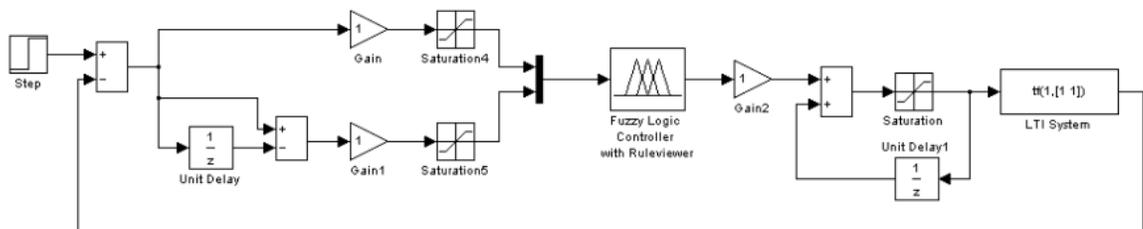


Figura 9 Diagrama de bloques de la estrategia de control.

2.2. MODELADO DINÁMICO DE LOS SISTEMAS

2.2.1. CONTROL DE VELOCIDAD DE UN MOTOR AC

2.2.1.1. MODELO DEL SISTEMA

Al momento de implementar los controladores es necesario tener un modelo que describa el comportamiento general de la planta, en este caso esta planta consta de un variador de frecuencia el cual alimenta un motor AC acoplado por un eje a un motor DC que actuara como generador.

Para esto es necesario encontrar un modelo adecuado que relacione la velocidad deseada en contra de la frecuencia que debe suministrar el variador. Se decidió realizar un modelo caja blanca y un modelo caja negra por separado como se muestra en la figura 10 con el fin de comparar cual se ajusta de mejor forma al comportamiento real de la planta.

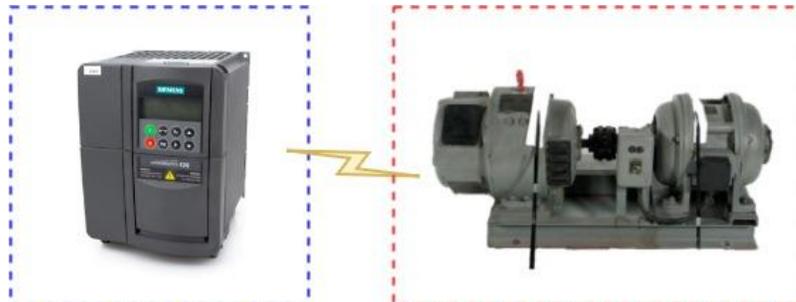


Figura 10 Variador de frecuencia - motor AC

Primero se realizó el modelo caja blanca para esto se debe tener conocer el comportamiento de un motor AC asíncrono de jaula de ardilla alimentado por un variador de frecuencia conectado a la red trifásica de 220V. El motor AC se hace arrancar con un voltaje dependiente de la frecuencia de alimentación en las terminales de línea, su estator desarrolla un par de arranque que hará que aumente su velocidad.

Los motores asíncronos también llamados de inducción son conocidos por ser robustos y de bajo costo, son los más empleados en la industria [9]. En este caso tenemos el modelo del variador de frecuencia el cual se puede realizar modificando el comportamiento de la señal de alimentación constante que suministra la red, el cual corresponde a un voltaje de 220V para este tipo de variadores. Lo que se hace es controlar mediante un valor de frecuencia estipulado el valor de alimentación real que se suministra al motor variando desde 0 a 220V nominales. Luego de esto se realiza la conversión del voltaje en componentes trifásicos 'abc' a su transformada 'dq0'. Este modelo se realizó de la forma que se muestra en la figura 11.

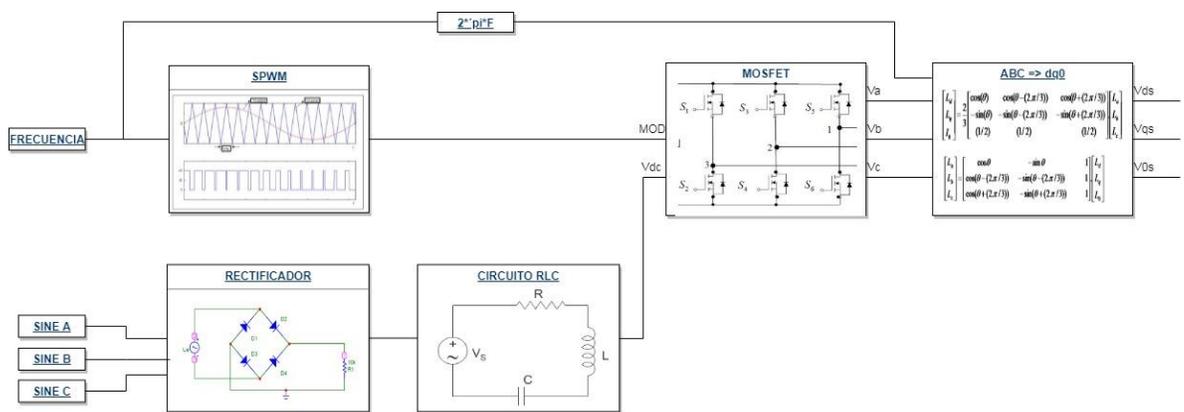


Figura 11 Modelo del variador de frecuencia

Inicialmente se genera una señal SPWM la cual es la encargada de activar el circuito mosfet a una frecuencia deseada (establecida por el usuario), esta es la parte donde se genera el cambio de frecuencia de alimentación al motor. A su vez se realiza el modelo de la señal de alimentación la cual consiste en una señal senoidal de valor 220 con tres fases las cuales pasan a través de un rectificador de onda completa para generar una señal de valor positivo y poderse modelar como un circuito RLC, la salida del circuito va a la entrada del bloque de mosfet este valor de voltaje Vdc recibe su activación gracias a la señal SPWM generada anteriormente, luego de esto, se tiene la salida de voltaje del circuito mosfet con las características mostradas en la figura 12 en las cuales se tienen las componentes 'abc' de la señal de alimentación.

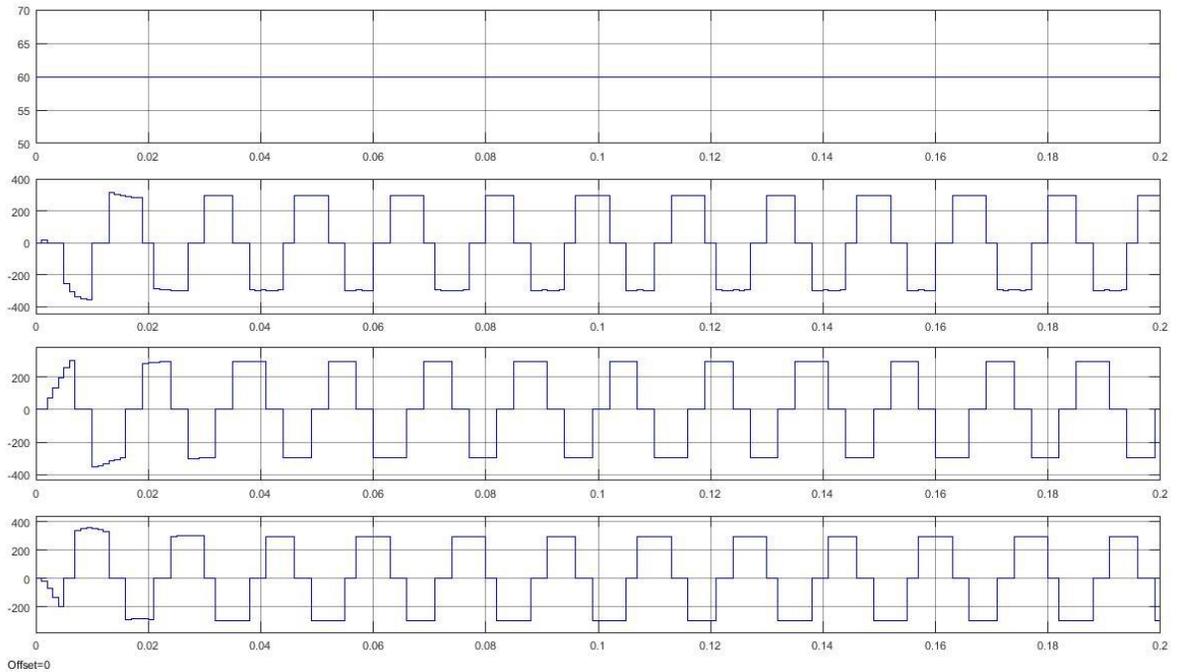


Figura 12 Señal de alimentación en componentes 'abc'

Luego de tener la señal en componentes 'abc' se procede a realizar la transformada de Park, la cual transformara la señal en componentes 'dq0', para realizar esta transformada se parte de la matriz de transformación.

$$\begin{bmatrix} V_d \\ V_q \\ V_0 \end{bmatrix} = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 1 \\ \cos(\omega t - \frac{2\pi}{3}) & -\sin(\omega t + \frac{2\pi}{3}) & 1 \\ \cos(\omega t + \frac{2\pi}{3}) & -\sin(\omega t + \frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.10)$$

Al obtener la señal de alimentación del motor en su transformada 'dq0', las ecuaciones que describen el comportamiento de un motor AC son:

$$V_{qs} = R_s i_{qs} + \frac{d}{dt} \phi_{qs} + \omega \phi_{ds} \quad (2.11)$$

$$V_{ds} = R_s i_{ds} + \frac{d}{dt} \phi_{qs} - \omega \phi_{qs} \quad (2.12)$$

$$V'_{qr} = R'_{r} i'_{qr} + \frac{d}{dt} \phi'_{qr} + (\omega - \omega_r) \phi'_{dr} \quad (2.13)$$

$$V'_{dr} = R'_{r} i'_{dr} + \frac{d}{dt} \phi'_{qr} - (\omega - \omega_r) \phi'_{qr} \quad (2.14)$$

$$T_e = 1.5p(\phi_{ds} i_{qs} - \phi_{qs} i_{ds}) \quad (2.15)$$

$$\frac{d}{dt} \omega_n = \frac{1}{2H} (T_e - F\omega_m - T_M) \quad (2.16)$$

$$\frac{d}{dt} \theta_m = \omega_m \quad (2.17)$$

Donde:

Parámetros del modelo	Definición
R_s, L_s	Resistencia del estator e Inductancia de fuga
L_m	Inductancia magnética
L_s	Inductancia total de estator
V_{qs}, i_{qs}	Eje q voltaje y corriente del estator
V_{ds}, i_{ds}	Eje d voltaje y corriente del estator
ϕ_{qs}, ϕ_{ds}	Flujos del estator en ejes d y q
ω_m	Velocidad angular del rotor
Θ_m	Posición angular del rotor
p	Numero de pares de polos
ω_r	Velocidad angular eléctrica ($\omega_m \times p$)
Θ_r	Posición angular eléctrica (rotor) ($\Theta_m \times p$)
T_e	Torque electromagnético
T_m	Torque mecánico del eje
J	Coefficiente de inercia de carga (rotor y carga).
H	Constante de inercia (rotor y carga).
F	Coefficiente de fricción viscosa (rotor y carga)

Las anteriores ecuaciones representan las componentes eléctricas y mecánicas para el modelado de un motor trifásico AC. Estas ecuaciones se implementaron junto con el modelo del variador en Simulink para comparar modelo caja blanca con los datos experimentales tomados figura 13.

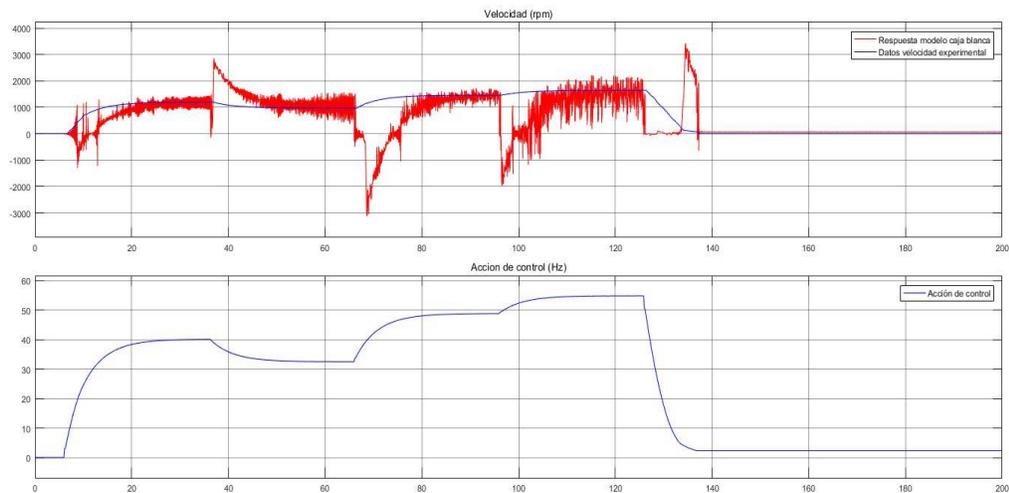


Figura 13 Modelo caja blanca vs datos experimentales

En el modelo caja blanca implementado se observó que la respuesta alcanza los mismos valores de velocidad comparado con la respuesta experimental, pero, este modelo en simulación posee un transitorio el cual demuestra inestabilidad en el sistema

Luego de esto, se planteó un modelo de caja negra con el fin de comparar las respuestas y elegir el modelo que más se adapte al sistema. Este tipo de modelos hace referencia a un cálculo de una función de transferencia que represente el sistema deseado, basándose en la entrada y la respuesta que tiene el sistema ante dicha entrada. Este método permite identificar claramente el comportamiento del sistema ante las entradas y salidas y estudiar las relaciones que existen entre ellas, permitiendo así maximizar la eficiencia del hallazgo los sistemas sin tener que introducirnos en los procesos matemáticos complejos de cada uno para poder encontrar una función que lo identifique. Inicialmente es necesario conocer el comportamiento del motor ante las variaciones de frecuencia y así determinar si posee un comportamiento lineal. En este caso en la figura 14 se somete el motor AC a variaciones de frecuencia iniciando desde los 5Hz hasta la frecuencia nominal de 60Hz.

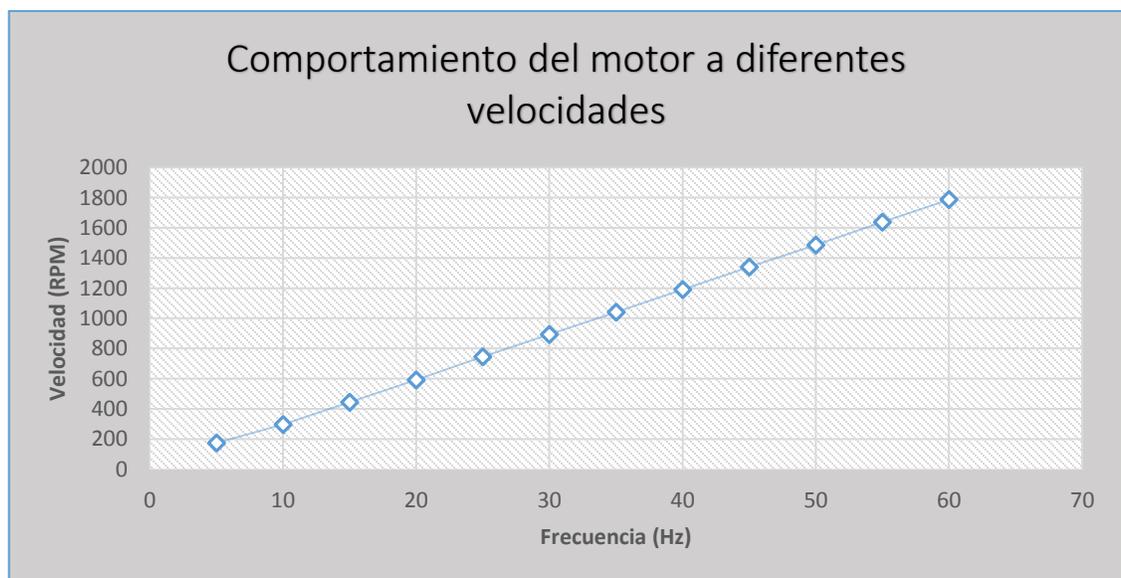


Figura 14 Linealidad del motor ante diferentes frecuencias de alimentación.

Luego de comprobar que el motor tiene un comportamiento lineal con respecto a los valores de frecuencia aplicada y con el fin de obtener un modelo que describa correctamente el comportamiento del sistema, se decidió encontrar un modelo caja negra que pudiese expresar el comportamiento de la planta relacionado con los cambios de frecuencia aplicadas al variador. En este caso esta planta consta de un motor AC acoplado por un eje a un motor DC que actuara como generador. Para esto es necesario encontrar un modelo adecuado que relacione la velocidad deseada en contra de la frecuencia que debe suministrar el variador. Para esto se decidió realizar un modelo caja blanca y un modelo caja negra con el fin de comparar cual se ajustaba de mejor forma al comportamiento real de la planta.

Se identificó el comportamiento del sistema respecto a la frecuencia de alimentación suministrada por el variador, esto con el fin de conocer si la velocidad de los motores era linealmente proporcional a los diferentes valores de frecuencia y se muestra el resultado en la figura 15.

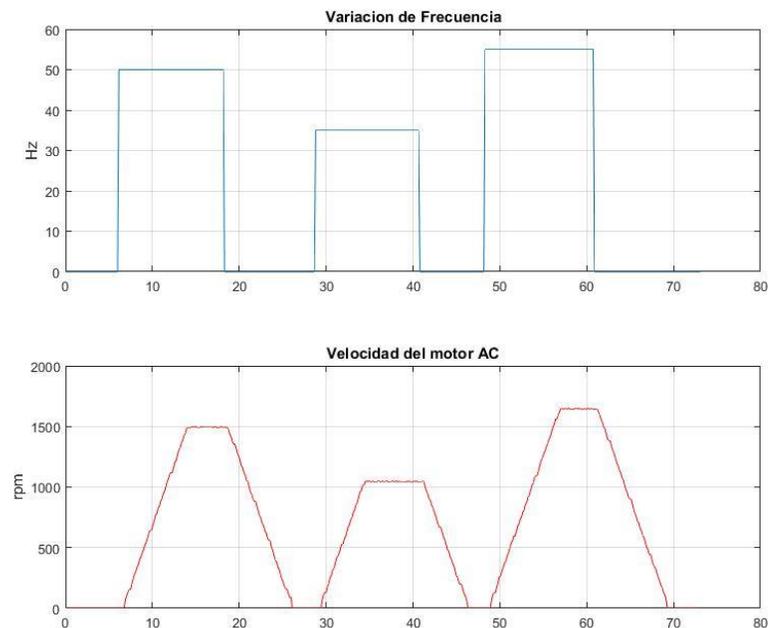


Figura 15 Velocidad del motor ante diferentes frecuencias de alimentación.

Por medio de la herramienta *ident*, de Matlab, se ingresaron estos datos obtenidos, y se usó como datos de validación otra respuesta con diferentes valores de frecuencia (figura 16). Luego de definir el tipo de función y comprobar que esta función poseía un comportamiento similar al de los datos de entrada evaluando su *best fit* se seleccionó la función de transferencia con el mejor *best fit* y se obtuvo el siguiente modelo del sistema,

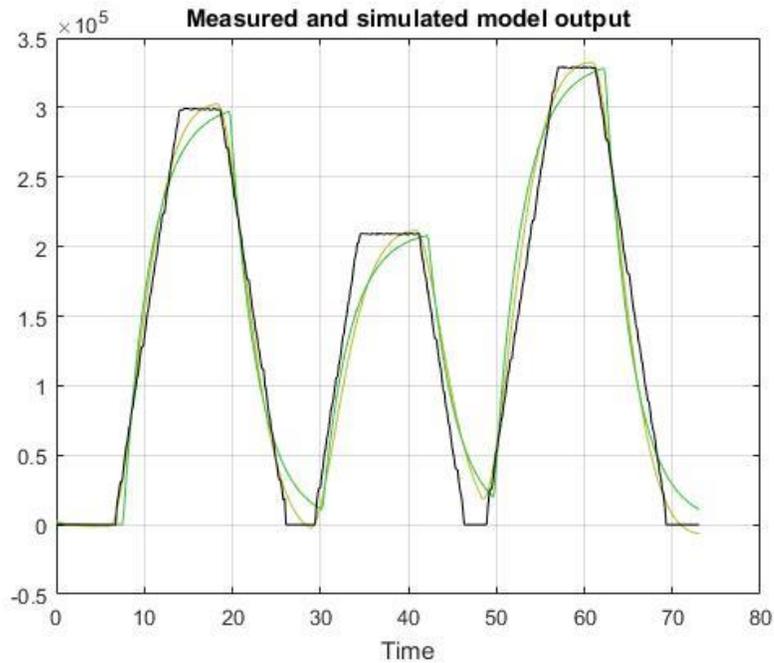


Figura 16 Toolbox Ident – Funciones con mejor best fit para representar la planta del sistema de control de velocidad.

$$G(s) = \exp(-0.264 * s) \frac{29.8}{0.00315 s^2 + 0.5481 s + 1} \quad (2.18)$$

Discretizada de acuerdo con el periodo de muestreo que en ese caso es la velocidad de transmisión de datos del servidor OPC ($T = 0.15 s$) se obtiene la función que identifica el sistema de forma discreta, la cual posee este comportamiento en lazo abierto ante una entrada escalón (Figura 17).

$$G(z) = z^{-2} * \frac{1.612 z^2 + 5.587 z + 0.0004981}{z^2 - 0.7584 z + 4.638e^{-12}} \quad (2.19)$$

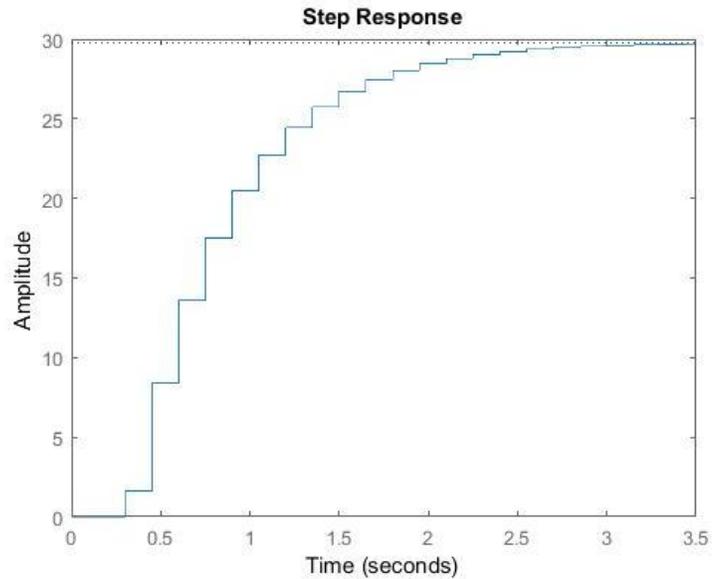


Figura 17 Respuesta de la planta discreta ante una entrada escalón.

Esta función se implementó en Simulink con el fin de comparar la respuesta de este modelo caja negra, con los datos tomados experimentalmente (figura 18)

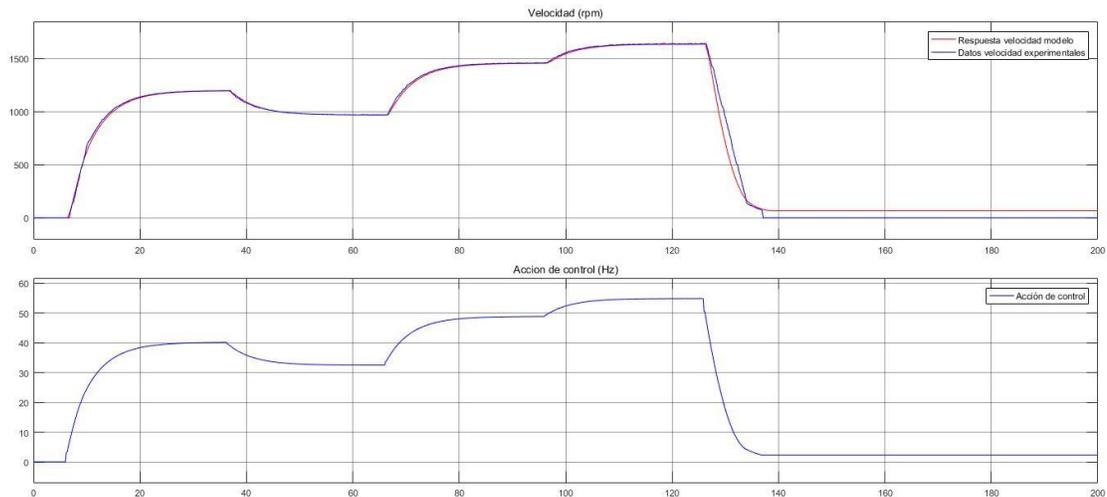


Figura 18 Modelo caja negra vs datos experimentales.

Luego de comparar las respuestas de los dos modelos se decidió trabajar con el modelo caja negra dado que presenta una mejor respuesta comparada con los datos experimentales tomados.

2.2.1.2. DISEÑO DEL CONTROLADOR PID

Para diseñar el controlador se utilizó la técnica de ubicación de polos en lazo cerrado en una posición específica, permitiendo variar parámetros como sobrepaso, tiempo de establecimiento y tiempo de levantamiento en la planta. Este tipo de diseño brinda una facilidad ya que conociendo las características del sistema podemos llegar a obtener una respuesta del sistema sin sobrepaso o con un sobrepaso, pero alcanzando el punto de consigna de manera más rápida. Con la herramienta *rltool*, de Matlab, se realizó el diseño del controlador según las condiciones requeridas, para este sistema se decidió que el controlador no permitiera el sobrepaso de la variable de proceso así fuese un poco más lento el tiempo de establecimiento, a este tipo de controlador se le denomina “sobre-amortiguado”. La ecuación característica del controlador fue la siguiente:

$$C(z) = \left(\frac{0.00365 (z-0.5)^2}{z(z-1)} \right) \quad (2.20)$$

Para comprobar la estabilidad del sistema se analiza el diagrama de polos y ceros, para determinar si el controlador es estable debe contener todos sus polos y ceros dentro del círculo unitario (esto si se trabaja en tiempo discreto), como podemos observar en la figura 19, el sistema posee todos los componentes dentro del círculo unitario lo cual garantiza su estabilidad.

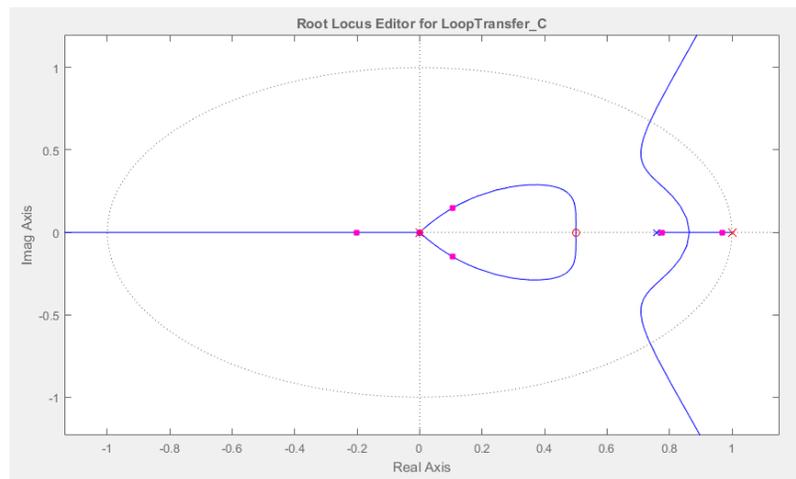


Figura 19 Diagrama de polos y ceros del controlador PID

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, incrementando aún más su valor. Esto significa que el valor integral puede volverse muy grande y producirse el efecto llamado “windup”.

Para evitar que ocurra este fenómeno hay dos formas:

- Introducir limitadores en las variaciones de la referencia 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 re-cálculo de la integral: cuando la salida se satura, la integral es recalculada tal que su nuevo valor proporciona una salida en el límite de la saturación.

En la figura 20 se puede apreciar la estrategia implementada en Simulink para el controlador PID agregando el bloque anti-windup y utilizando las siguientes ganancias del controlador.

$$K_p = 0.00068$$

$$K_i = 0.00099375$$

$$K_d = 0.00099375$$

$$K_{aw} = 56 \text{ (Ganancia anti windup).}$$

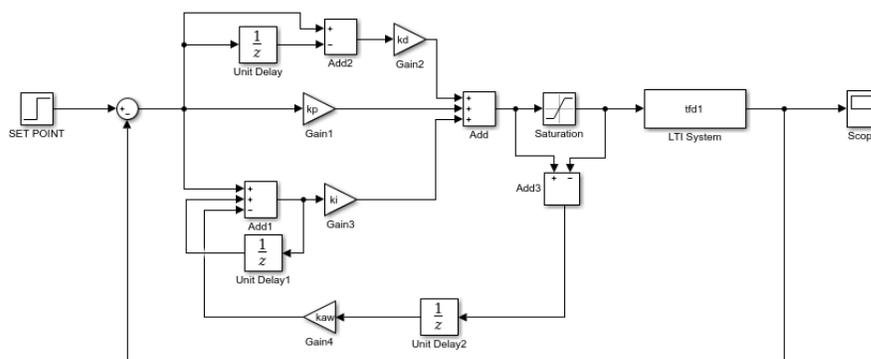


Figura 20 Diagrama de bloques controlador PID.

En la figura 21 se aprecia la respuesta del controlador frente a cambios en la señal de referencia, en este caso diferentes valores de velocidad establecidos, con intervalos de 22.5 segundos para asegurar el establecimiento de la variable del proceso (velocidad), en la parte inferior se observa el comportamiento en la acción de control referente a la frecuencia de alimentación suministrada por el variador de frecuencia. Se puede apreciar que en lazo cerrado con el controlador PID se tiene un tiempo de establecimiento aproximado de 18 segundos, con una respuesta sobre amortiguada.

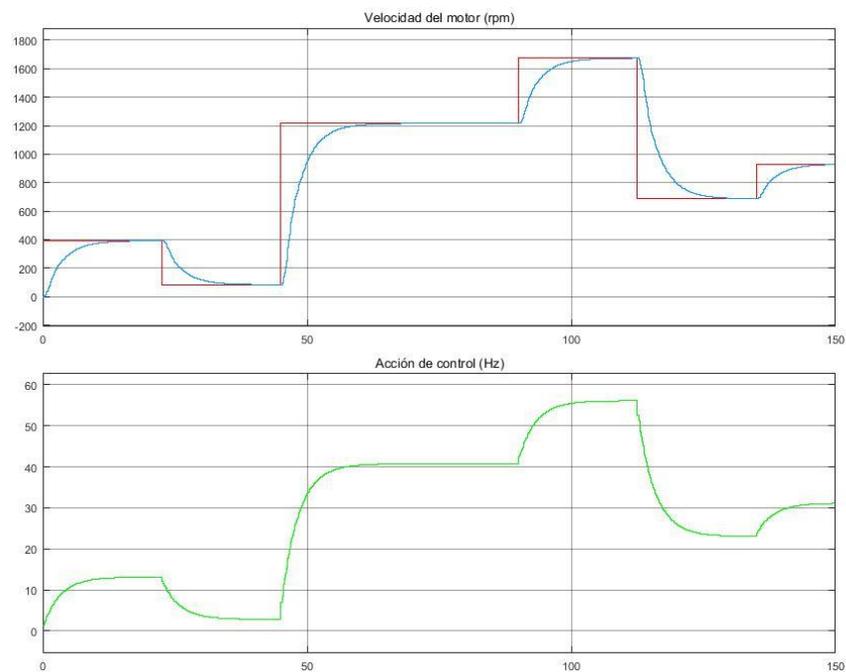


Figura 21 Respuesta de la planta con controlador PID.

2.2.1.3. DISEÑO DEL CONTROLADOR LQG

Para el diseño del controlador LQG, se tuvo en cuenta el modelo caja negra el cual es el que mejor se ajusta a la respuesta de la planta real. En el caso del controlador LQG se pretende variar los valores de las matrices Q y R , con el fin de obtener una respuesta óptima del sistema. La matriz R permite restringir el comportamiento de la acción de control, en este caso hay que recordar que la acción de control está dada por la frecuencia suministrada por el variador está en el rango de 0 a 60 Hz.

De la misma forma, la matriz Q permite restringir el comportamiento de la salida, modificando parámetros como tiempo de establecimiento, sobrepaso, tiempo de retardo, etc.

En la figura 22 se observa el sistema de bloques implementado en la herramienta Simulink de Matlab. En dicho esquema se observan parámetros como la ganancia anti-windup que permite reducir el comportamiento de la saturación del actuador, la constante integral K_r que penaliza la entrada, la constante de penalización de los estados K_Q y el modelo en tiempo discreto expresado en espacio de estados.

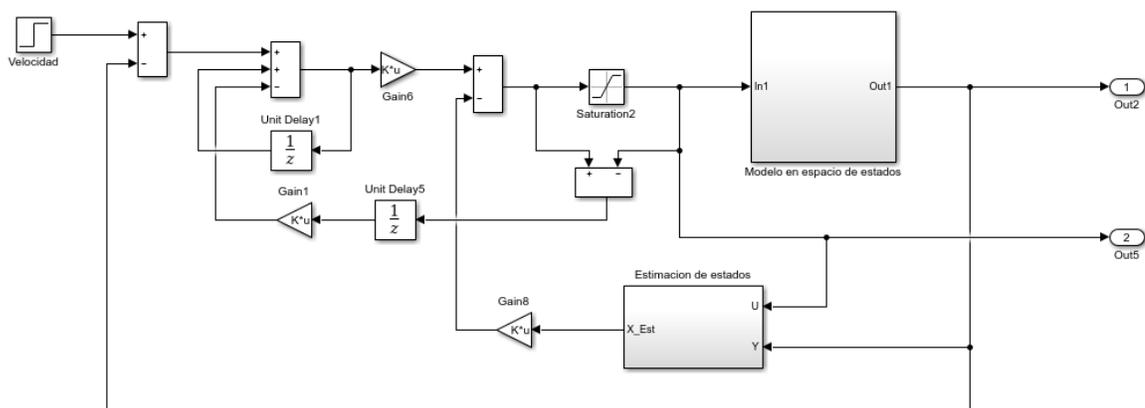


Figura 22 Diagrama de bloques controlador LQG.

La estimación de los estados se ha realizado mediante la implementación de un filtro de Kalman, este filtro se define:

$$X_{ob} = (Gl - K_e \cdot C)X_{ob_1} + Hl \cdot U + K_e \cdot PV \quad (2.21)$$

Donde:

- X_{ob} : Estados observados
- Gl : Matriz A del modelo lineal discreto
- Hl : Matriz B del modelo lineal discreto
- C : Matriz del modelo lineal discreto;
- X_{ob_1} : Valor anterior de la matriz X_{ob}
- U : Señal de control
- PV : Variable del proceso o señal de salida
- K_e : Ganancia de estimación.

En la parte inferior de la figura 23, se observa un subsistema, dicho bloque corresponde a la estimación de los estados adicionales del modelo. En la figura 21, se observa con mayor claridad el bloque correspondiente a la estimación de los estados.

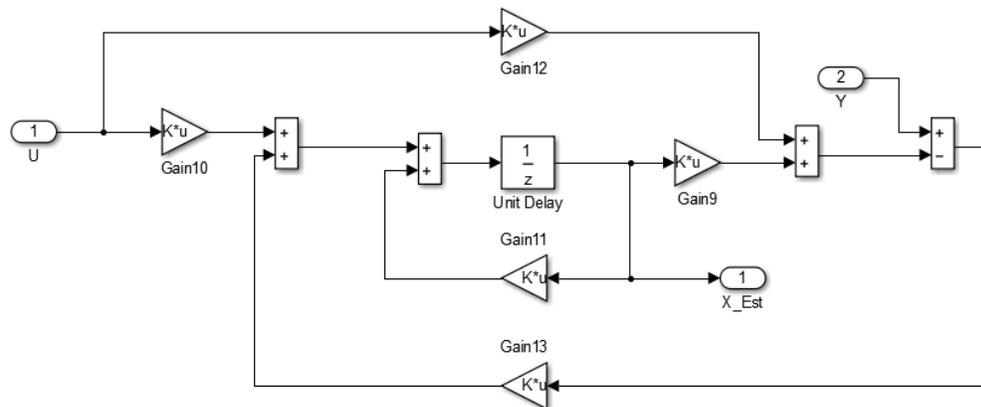


Figura 23 Subsistema correspondiente a la estimación de los estados.

Para el diseño del filtro de Kalman es necesario generar las matrices de covarianza del sistema, donde Q_n es la matriz de covarianza de la señal de salida y R_n es la matriz de covarianza de la señal de entrada. Para la obtención de los valores de las matrices Q_n y R_n se utilizan los datos de identificación de la señal de entrada y salida, de los cuales se generan los valores de varianza.

Para el diseño de seguimiento es necesario crear una matriz aumentada \hat{G} y \hat{H} , que reemplacen a las matrices G y H en tiempo discreto, referentes a las matrices A y B en tiempo continuo del modelo en espacio de estados. Dichas matrices se crean de la siguiente forma:

$$\hat{G} = \begin{bmatrix} G & Z(n, m) \\ -C \cdot G & I(i, j) \end{bmatrix}; \quad \text{En donde} \quad \begin{cases} G, \text{ Matriz referente a la matriz } A. \\ H, \text{ Matriz referente a la matriz } B. \\ Z, \text{ Matriz de ceros.} \\ n, \text{ Número de estados.} \\ m, i \text{ y } j, \text{ Número de integradores.} \\ I, \text{ Matriz de identidad.} \end{cases} \quad (2.22)$$

$$\hat{H} = \begin{bmatrix} H \\ -C \cdot H \end{bmatrix} \quad (2.23)$$

Una vez son creadas estas matrices se introducen en la función de Matlab *dlqr*, dicha función recibe las matrices aumentadas \hat{G} , \hat{H} , Q y R ; la función crea un controlador en lazo cerrado en forma de ganancias K , a partir de un sistema discreto en espacio de estados.

- $\hat{G} = \begin{bmatrix} -0.0082 & -0.0883 & 0 \\ 0.0713 & 0.7666 & 0 \\ -1.2316 & -132503 & 0 \end{bmatrix}$
- $\hat{H} = \begin{bmatrix} 0.1425 \\ 0.3765 \\ -6.8096 \end{bmatrix}$

Para el cálculo de las ganancias K_e que multiplican a los estados observados, se utiliza la función *Kalman* de Matlab. Dicha función recibe el modelo discreto en espacio de estados que representa a la planta, y las matrices Q_n y R_n correspondientes a la covarianza de la salida y la entrada respectivamente. Es

importante aclarar que la respuesta del observador debe ser más rápida que la del controlador, de esta forma se garantiza un comportamiento óptimo en la planta.

Las matrices de desviación estándar (elevadas al cuadrado para obtener la varianza) en el filtro Kalman son las siguientes:

- $Q_n = [0.2^2]$
- $R_n = [1.9^2]$

En el diseño del controlador LQG se obtuvieron los siguientes valores de las matrices Q y R , y las dos ganancias K_e y K_i .

- $Q = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$
- $R = [15]$.
- $K_e = [0.4200 \quad 0.8570]$.
- $K_i = [0.0240]$
- $L = \begin{bmatrix} -0.0004 \\ 0.0094 \end{bmatrix}$

Para comprobar la estabilidad del sistema se analizan los valores propios del sistema en lazo cerrado y se obtienen los siguientes valores.

$$[0.0000 \quad 0.9381 \quad 0.7612]$$

Estos valores corresponden a los valores propios del sistema en lazo cerrado y se encuentran adentro del círculo unitario garantizando la estabilidad del sistema. En la figura 24 se aprecia la respuesta del controlador frente a cambios en la señal de referencia, en este caso cambios de velocidades, en la parte inferior se observa el comportamiento en la acción de control referente a la frecuencia de alimentación suministrada por el variador.

Se puede apreciar que en lazo cerrado con el controlador LQG se tiene un tiempo de establecimiento de 13 segundos aproximadamente, sin sobrepaso.

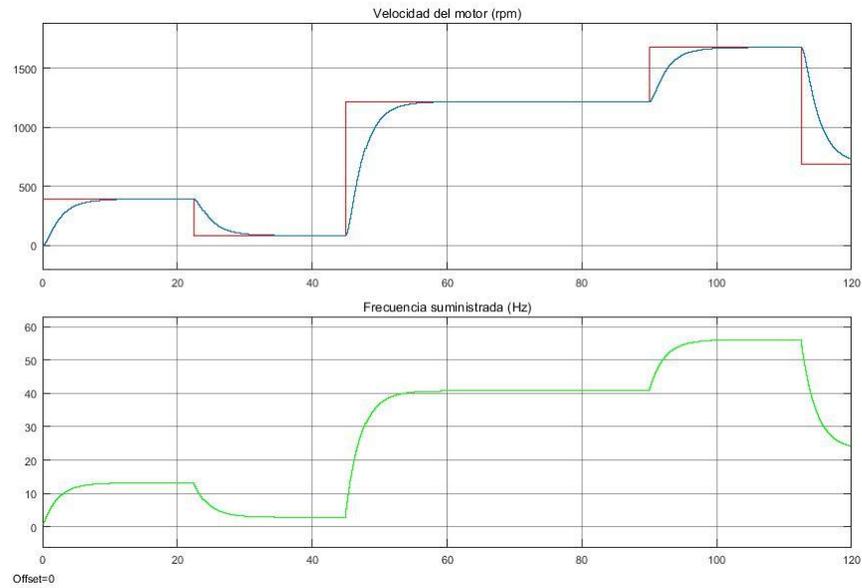


Figura 24 Respuesta del sistema ante el controlador LQG.

2.2.1.4. DISEÑO DEL CONTROLADOR FUZZY

Se diseñaron dos conjuntos de igual tamaño para trabajar el error y la razón de cambio del error como se puede observar en la figura 25 y figura 26.

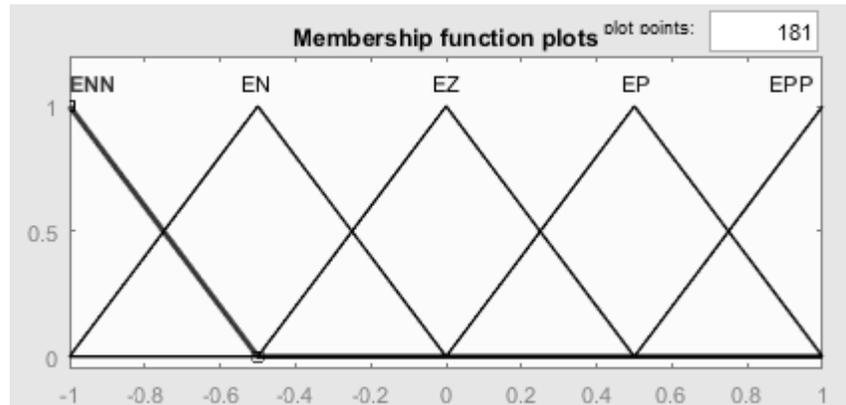


Figura 25 Conjunto difuso correspondiente al error.

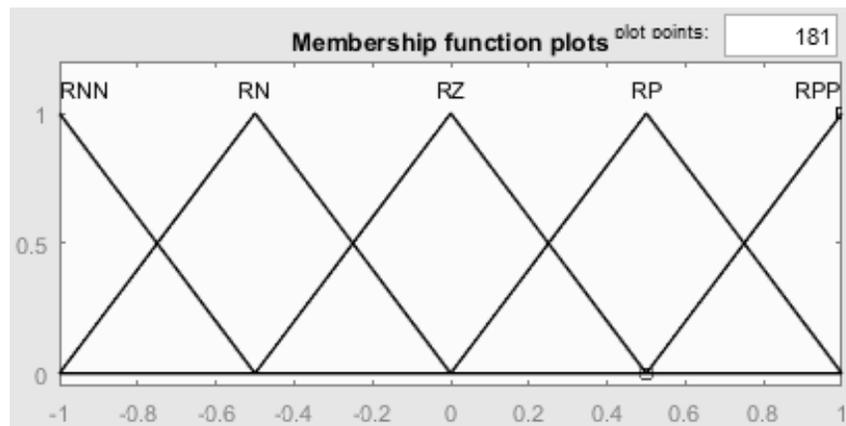


Figura 26 Conjunto difuso correspondiente a la derivada del error.

La base de reglas difusas establecidas para los conjuntos de las figuras anteriores que se aplican al error y la razón de cambio del error se puede observar en la tabla 2, donde se define la posición de los valores difusos para luego pasar a ser clasificado en las funciones de salida.

Tabla 3 Reglas establecidas para el controlador Fuzzy.

		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

El conjunto de salida para el controlador tipo Mamdani se estableció de la siguiente manera como se muestra en la figura 27.

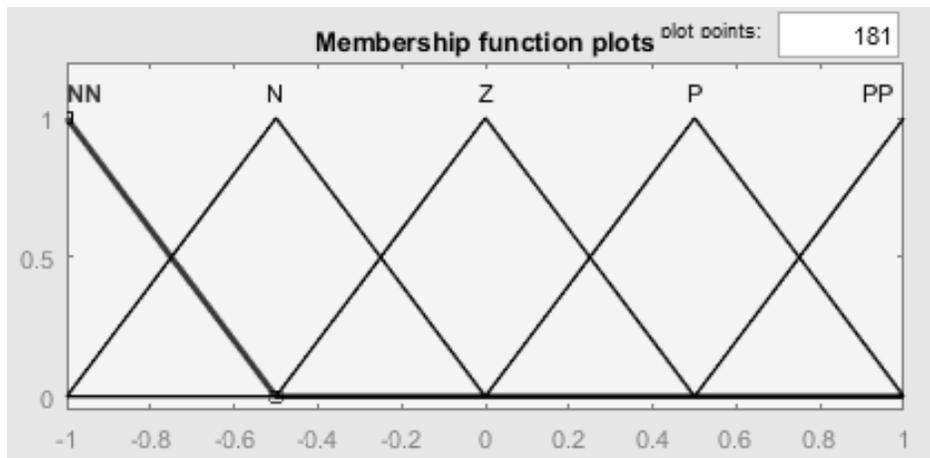


Figura 27 Conjunto difuso correspondiente a la salida del controlador.

El diagrama de bloques para simular el sistema de control Fuzzy tipo Mamdani está representado por una constante proporcional (K_F) y una constante derivativa (K_d), a la salida del controlador se escala con la constante K_{salida} y posterior a esto se integra, por último se tiene el saturador de la acción de control que se encuentra entre 0 y 60 ya que estos son los valores mínimo y máximo de frecuencia que suministra el variador de frecuencia. Para este diagrama de bloques se usó una función para reemplazar el bloque controlador Fuzzy ya que este no permite trabajar

en tiempo discreto. En la figura 28 se puede observar el diagrama de bloques del controlador Fuzzy.

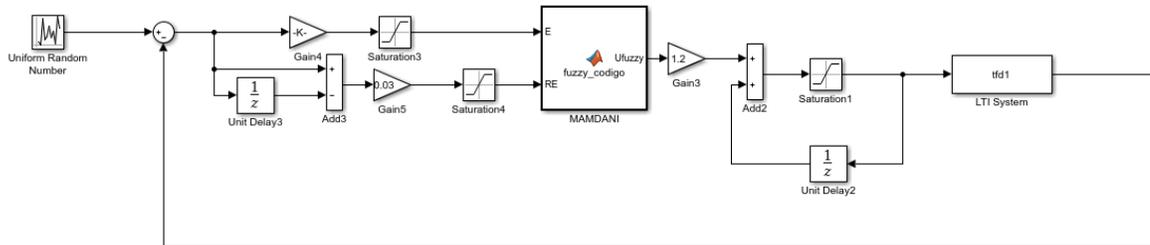


Figura 28 Diagrama de bloques del controlador Fuzzy tipo Mamdani.

Para determinar el comportamiento que adoptara el controlador Fuzzy se analiza la superficie de control que se rige por la base de reglas y la forma de los conjuntos difusos utilizados. En la figura 29 observamos la superficie del controlador Fuzzy aplicado para este sistema.

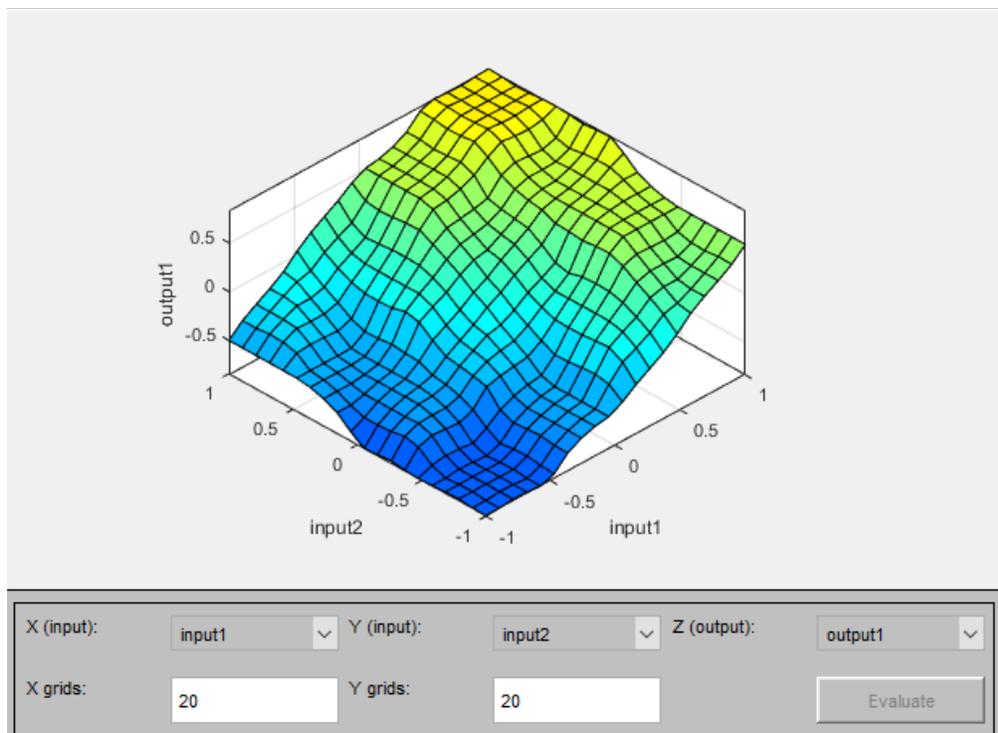


Figura 29 Superficie de control difuso

En la figura 30 se observa la respuesta del controlador Fuzzy frente a cambios de la señal de referencia, donde se tiene un tiempo de establecimiento de 28 segundos, con un pico de sobrepaso del 3.2% para el set point de velocidad. Las ganancias usadas para este controlador son las siguientes:

$$K_{error} = 0.07$$

$$K_{d_error} = 0.1$$

$$K_{salida} = 1.1$$

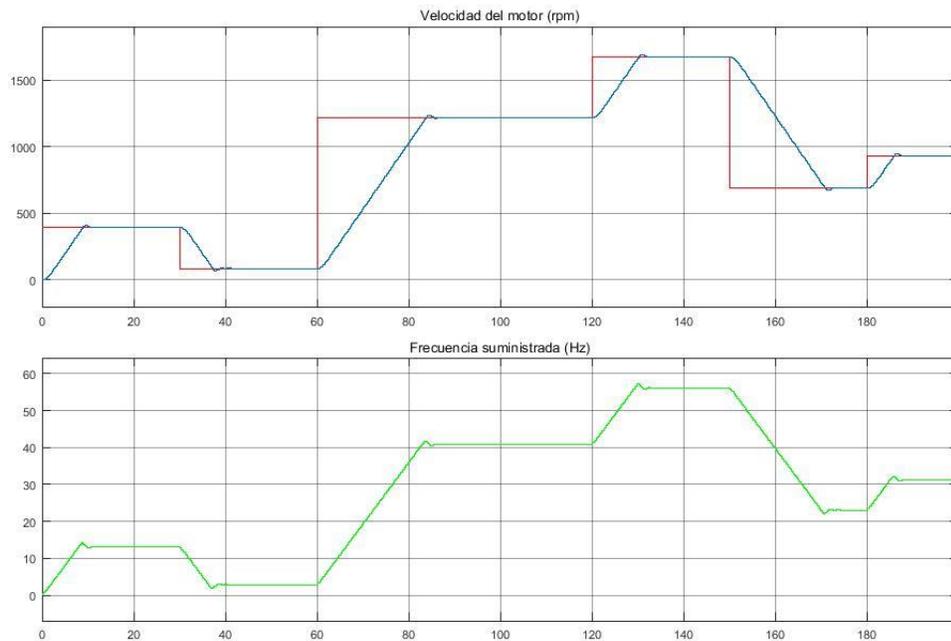


Figura 30 Respuesta del sistema aplicando un controlador Fuzzy.

2.2.2. CONTROL DE POSICIÓN

2.2.2.1. MODELO DEL SISTEMA

Para el modelado de la segunda planta, la cual corresponde a una barra de acero acoplada a un servomotor industrial marca Rockwell, es necesario adquirir la función de transferencia que identifica este sistema. Para esto se decidió realizar un modelo caja blanca y un modelo caja negra con el fin de comparar cual se ajustaba de mejor forma al comportamiento real de la planta.

Para el modelo de caja blanca del motor, se usó la ecuación que determina el modelo dinámico de un robot rígido [12]

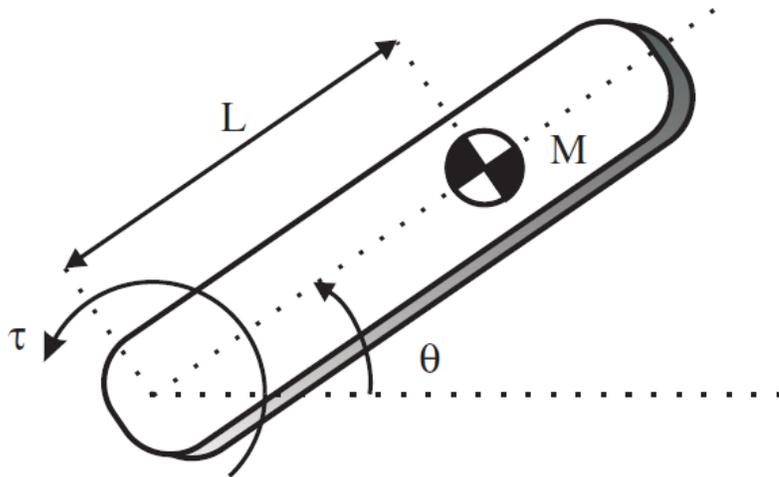


Figura 31 Modelo de barra rígida.

Este modelo se establece planteando el equilibrio de fuerzas con la segunda ley de Euler. La obtención del modelo dinámico de un mecanismo, y en particular de un robot, se basa fundamentalmente en el planteamiento del equilibrio de fuerzas establecido en la segunda ley de Newton, o su equivalente para movimientos de rotación, la denominada ley de Euler.

$$\sum T = \frac{d}{dt}(I\omega) = I\dot{\omega} + \omega \times (I\omega) \quad (2.24)$$

En el caso simple de un robot mono articular como el que se posee, el equilibrio de fuerza daría como resultado la ecuación:

$$\tau = ML^2\ddot{\theta} + MgL \cos \theta \quad (2.25)$$

Para poder representar esta ecuación como mi sistema se debe realizar la linealización y representarlo como la relación $\frac{\theta}{\omega}$ (ángulo/velocidad angular).

$$\frac{K}{\tau\omega*s+1} = ML^2\theta(s) s^2 + MgL \quad (2.26)$$

Se puede describir el sistema dinámico de una barra rígida acoplada a un servomotor industrial en un diagrama de bloques en Simulink. La ventaja de realizar este modelo en Simulink es que se pueden variar los parámetros desconocidos para poder encontrar las constantes desconocidas y se pueda identificar correctamente el sistema, además de esto, se comparó la respuesta del modelo caja blanca con los datos experimentales obtenidos en la figura 32.

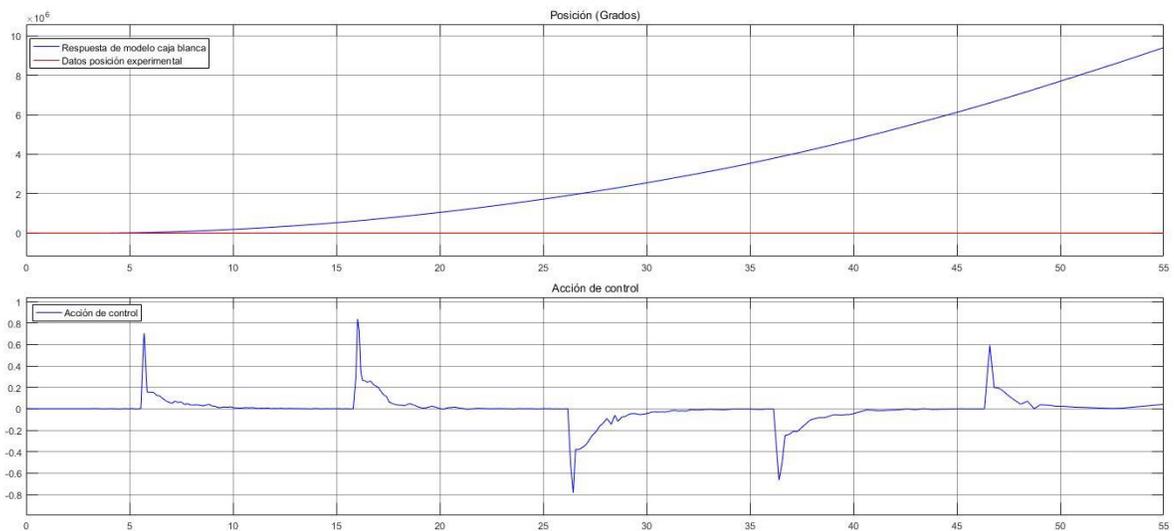


Figura 32 Modelo caja blanca vs datos experimentales.

En el modelado de este sistema no se logró obtener una respuesta adecuada para el sistema.

Para la identificación de caja negra, se parte de que en este caso el servomotor posee un controlador de velocidad incorporado en su configuración, lo cual hace que se pueda establecer una velocidad deseada al servomotor, y este responda alcanzando esta velocidad y variando la posición de forma constante a la relación de la velocidad de entrada, en este caso, se le adapto un control proporcional al sistema y se tomaron datos en lazo cerrado para poder realizar la identificación de la planta siendo un sistema con integrador propio. En la figura 33 tenemos la respuesta del sistema a diferentes puntos de consigna de posición.

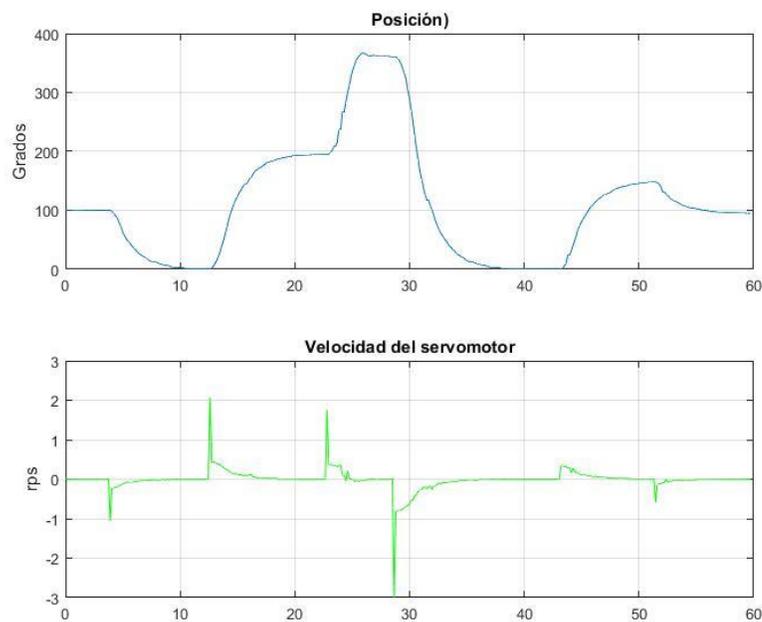


Figura 33 Respuesta del sistema en lazo abierto para identificación.

Por medio de la herramienta *ident*, de Matlab, se ingresaron estos datos obtenidos. Luego de definir el tipo de función y comprobar que esta función poseía un comportamiento similar al de los datos de entrada evaluando su *best fit*, se obtuvo el siguiente modelo del sistema

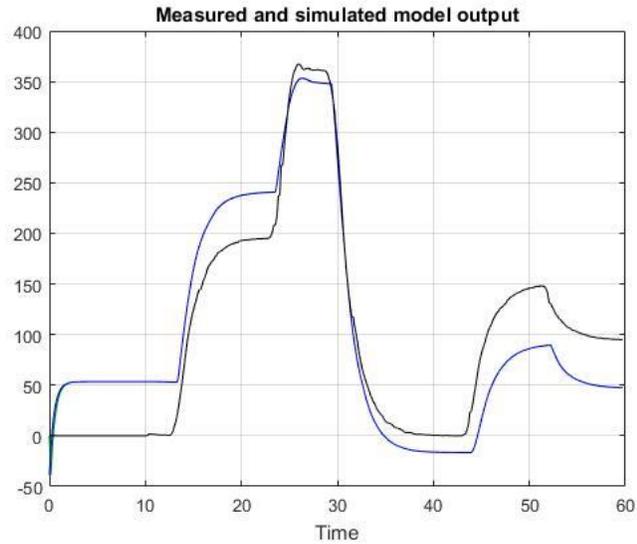


Figura 34 Toolbox Ident – Funciones con mejor best fit para representar la planta del sistema de control de posición.

$$G(s) = e^{(-0.337s)} * \frac{0.5552}{0.4675 s^2 + s} \quad (2.28)$$

Discretizada de acuerdo con el periodo de muestreo que brinda el servidor OPC ($T = 0.15 s$) se obtiene la función que identifica el sistema de forma discreta.

$$G(z) = z^{-3} * \frac{0.007006 z^2 + 0.01525 z + 0.0006057}{z^2 - 1.726 z + 0.7255} \quad (2.29)$$

En la figura 35 se observa el comportamiento en lazo abierto de la planta identificada.

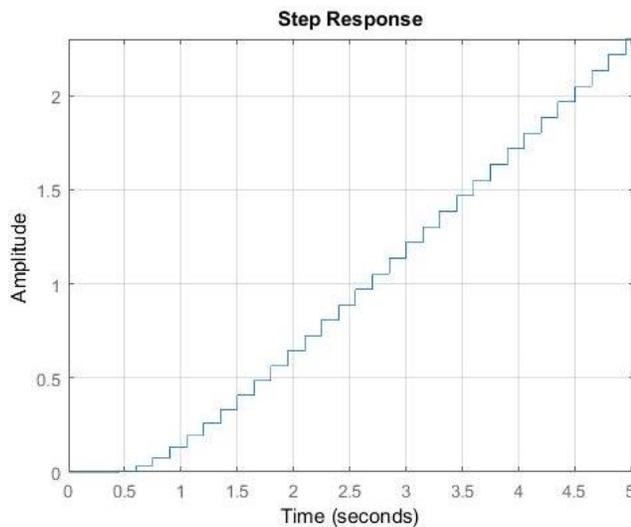


Figura 35 Comportamiento de la planta ante una entrada escalón.

Esta función se implementó en Simulink con el fin de comparar la respuesta de este modelo caja negra, con los datos tomados experimentalmente (figura 36)

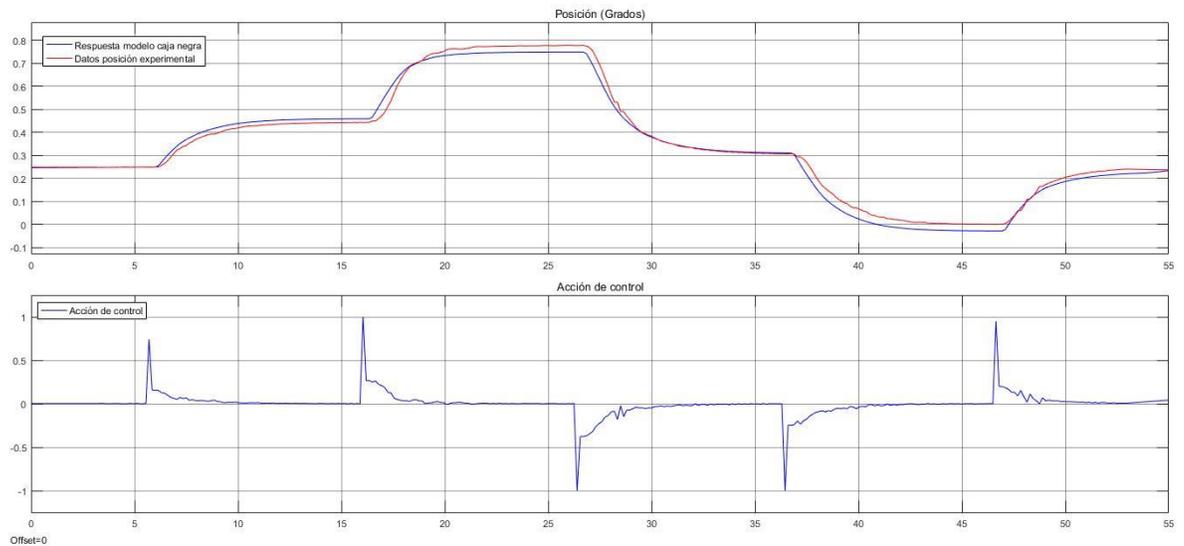


Figura 36 Modelo caja negra vs datos experimentales.

Luego de comparar las respuestas de los dos modelos se decidió trabajar con el modelo caja negra dado que presenta una mejor respuesta comparada con los datos experimentales tomados.

2.2.2.2. DISEÑO DEL CONTROLADOR PD

En este caso, se decidió aplicar un control PD ya que la planta de nuestro sistema posee integrador, lo que hace que en lazo cerrado alcance el punto de consigna, el control PD se utiliza para regular el tiempo de estabilización. Para diseñar el controlador se utilizó la técnica de ubicación de polos en lazo cerrado en una posición específica, permitiendo variar parámetros como sobrepaso, tiempo de establecimiento y tiempo de levantamiento en la planta. El diagrama de bloques usado para simular este controlador se observa en la figura 37, diagrama similar al del controlador PID, pero esta vez con la ganancia integral y la ganancia anti wind-up en "0".

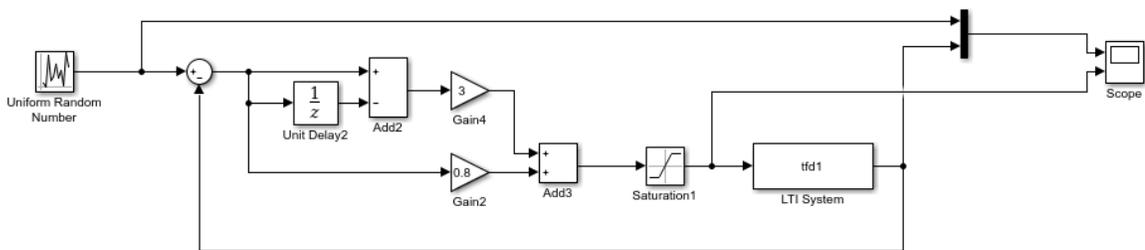


Figura 37 Diagrama de bloques para implementar un control PD.

En la figura 38 se aprecia la respuesta del controlador frente a cambios en la señal de referencia, en este caso, diferentes valores de posición aleatorios con intervalos 7.5 segundos para asegurar el establecimiento de la variable del proceso, en la parte inferior se observa el comportamiento en la acción de control referente a velocidad suministrada por el servomotor. Se puede apreciar que en lazo cerrado con el controlador PD se tiene un tiempo de establecimiento aproximado de 7 segundos, con una respuesta sobre amortiguada. La función del controlador PD es la siguiente.

$$C(z) = ((3 + 0.8) * (1 - z)^{-1}) \quad (2.30)$$

Las ganancias usadas para la implementación de este controlador son las siguientes:

$$Kp = 3$$

$$Kd = 0.8$$

Para comprobar la estabilidad del sistema se analiza el diagrama de polos y ceros, para determinar si el controlador es estable debe contener todos sus polos y ceros dentro del círculo unitario (esto si se trabaja en tiempo discreto), como podemos observar en la figura 38 el controlador posee todos los componentes dentro del círculo unitario.

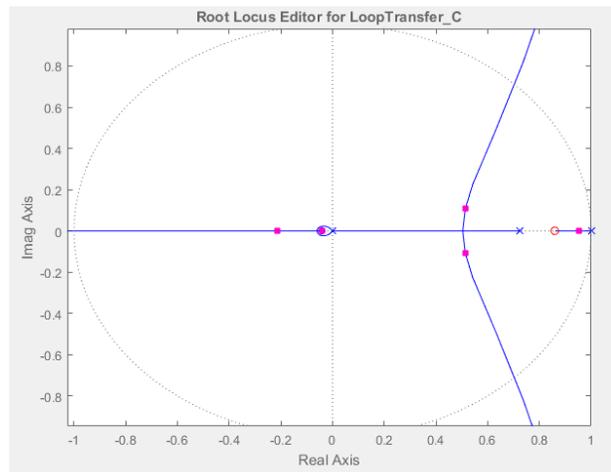


Figura 38 Diagrama polos y ceros control PD

Se obtiene la respuesta del modelo con control PD para el sistema de posición ante una entrada como se muestra en la figura 39.

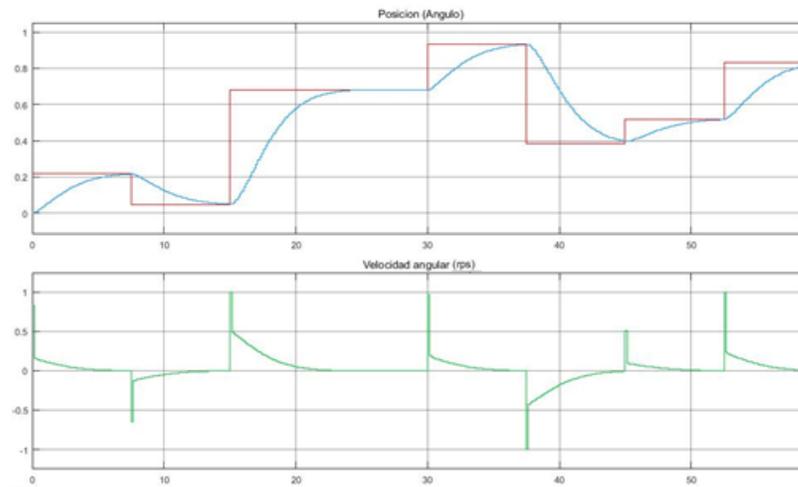


Figura 39 Respuesta del sistema con controlador PD.

la figura 41 se aprecia la respuesta del controlador frente a cambios en la señal de referencia, en este caso, diferentes valores de posición aleatorios con intervalos 12 segundos para asegurar el establecimiento de la variable del proceso, en la parte inferior se observa el comportamiento en la acción de control referente a velocidad suministrada por el servomotor. Se puede apreciar que en lazo cerrado con el controlador LQG se tiene un tiempo de establecimiento aproximado de 8 segundos, con una respuesta sub amortiguada.

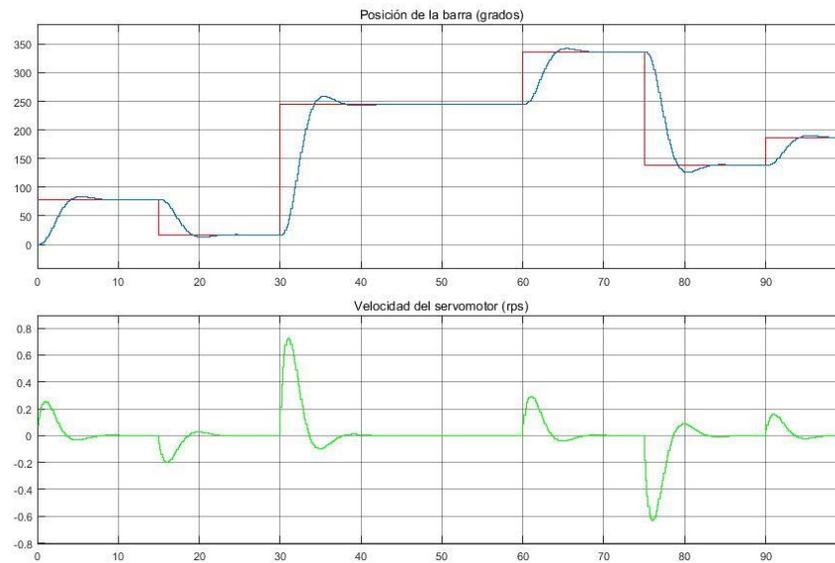


Figura 41 Respuesta del comportamiento de la planta con controlador LQG.

2.2.2.4. DISEÑO DEL CONTROLADOR FUZZY

El diagrama de bloques para simular el sistema de control Fuzzy tipo Mamdani está representado por una constante proporcional (K_F) y una constante derivativa (K_d), a la salida del controlador se escala con la constante (K_{salida}), por último se tiene el saturador de la acción de control que se encuentra entre -1 y 1 ya que estos son los valores mínimo y máximo de velocidad del servomotor establecidos para un buen manejo de la variable. Para este diagrama de bloques se usó una función para reemplazar el bloque controlador Fuzzy ya que este no permite trabajar en tiempo discreto. Se utilizó el mismo controlador Fuzzy diseñado anteriormente para el otro

sistema, el diagrama de bloques se muestra en la figura 42. Las ganancias usadas para este controlador son las siguientes:

$$K_{error} = 0.09$$

$$K_{d_error} = 1$$

$$K_{salida} = 2$$

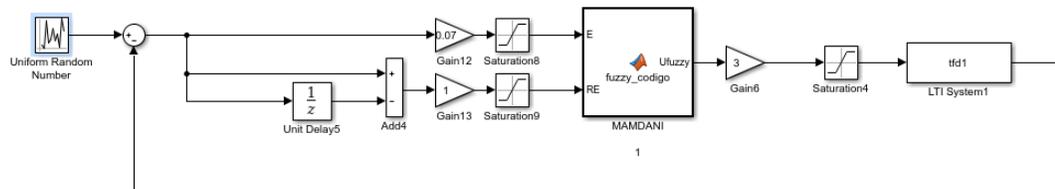


Figura 42. Diagrama de bloques del controlador Fuzzy tipo Mamdani para planta con integrador.

Para determinar el comportamiento que adoptara el controlador Fuzzy se analiza la superficie de control que se rige por la base de reglas y la forma de los conjuntos difusos utilizados. En la figura 43 observamos la superficie del controlador Fuzzy aplicado para este sistema. En este caso la acción de control actúa diferente ya que la planta tiene integrador propio por lo tanto la acción de control se verá con un comportamiento lineal.

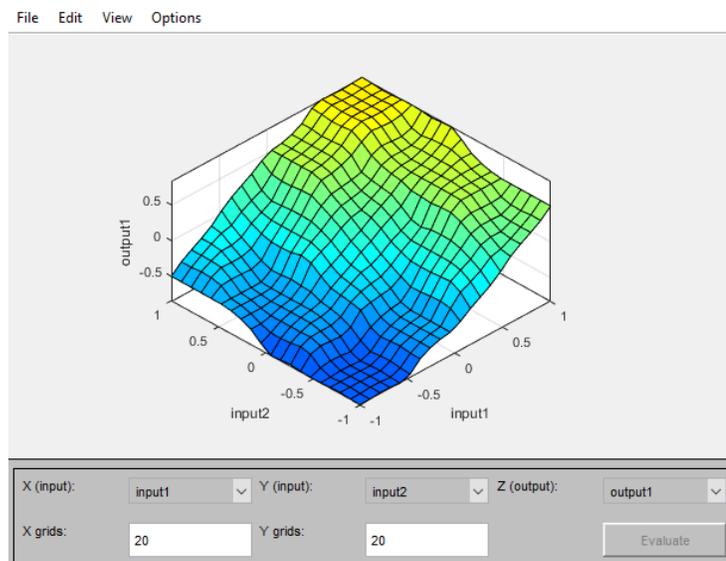


Figura 43 Superficie de control difuso

En la figura 44 se observa la respuesta del controlador Fuzzy frente a cambios de la señal de referencia, donde se tiene un tiempo de establecimiento de 9 segundos con una respuesta sobre amortiguada. Las ganancias usadas para este controlador son las siguientes:

$$K_{error} = 0.07$$

$$K_{d_error} = 0.1$$

$$K_{salida} = 1.1$$

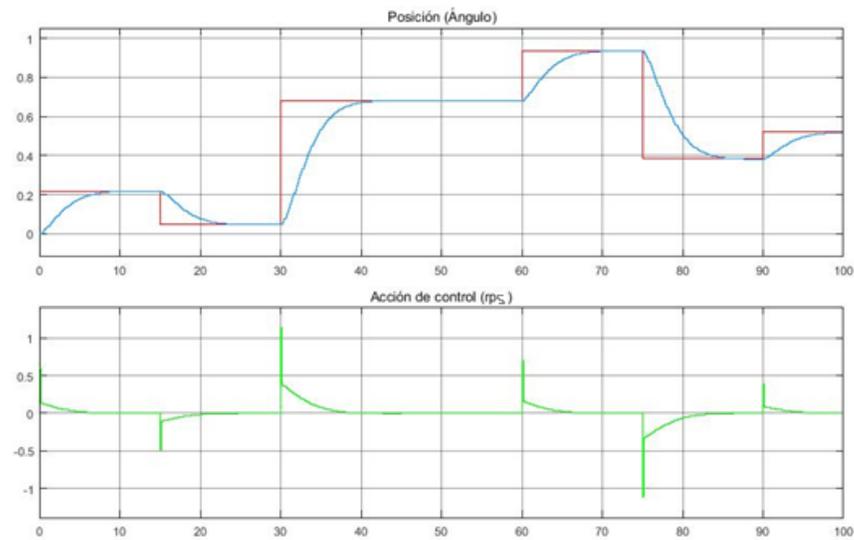


Figura 44 Respuesta del comportamiento de la planta con controlador Fuzzy.

3. VALIDACIÓN EXPERIMENTAL

En este capítulo se presentarán los resultados de la implementación de los controladores a cada planta, además de las interfaces HMI creadas para él la interacción del usuario y cada sistema, tanto en Matlab como en LabVIEW.

3.1. INTERFACES 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 brindada, como en este caso, por algún software de programación ya sea LabVIEW o Matlab. En el caso del proyecto se vio la necesidad de crear una interfaz HMI para cada uno de los softwares, que le permita al operario, sea estudiante o profesor, manipular y visualizar el comportamiento de los sistemas.

3.1.1. MATLAB

Como se mencionó anteriormente, para cada software se diseñaron diferentes HMI, en este caso, para el software Matlab, se diseñó una interfaz independiente para cada sistema. Cada interfaz integra los tres controladores diferentes, ofreciendo un panel en el cual se puede seleccionar el tipo de controlador que se desea aplicar y dando opción de modificar las ganancias de los controladores PID (o PD) y Fuzzy. En las figuras 45 y 46 se muestran las dos diferentes interfaces creadas para el sistema de velocidad y de posición respectivamente, estas interfaces se pueden operar de forma simultánea, aunque se encuentren en diferentes ventanas (las imágenes se encuentran en mayor tamaño en el anexo 6).

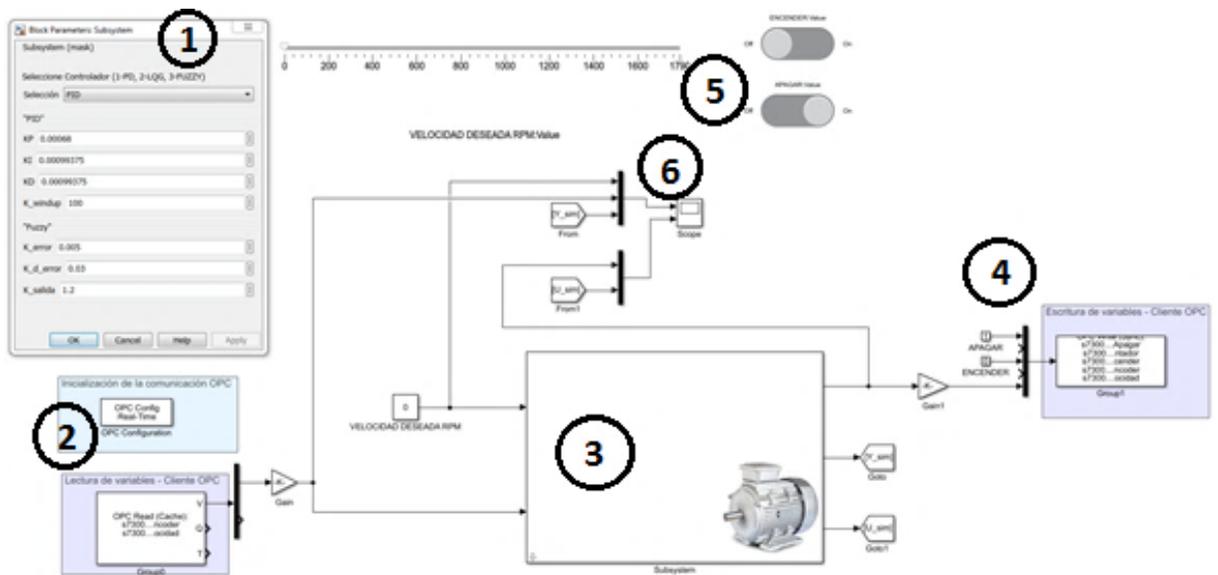


Figura 45 Interfaz para control de velocidad del motor AC.

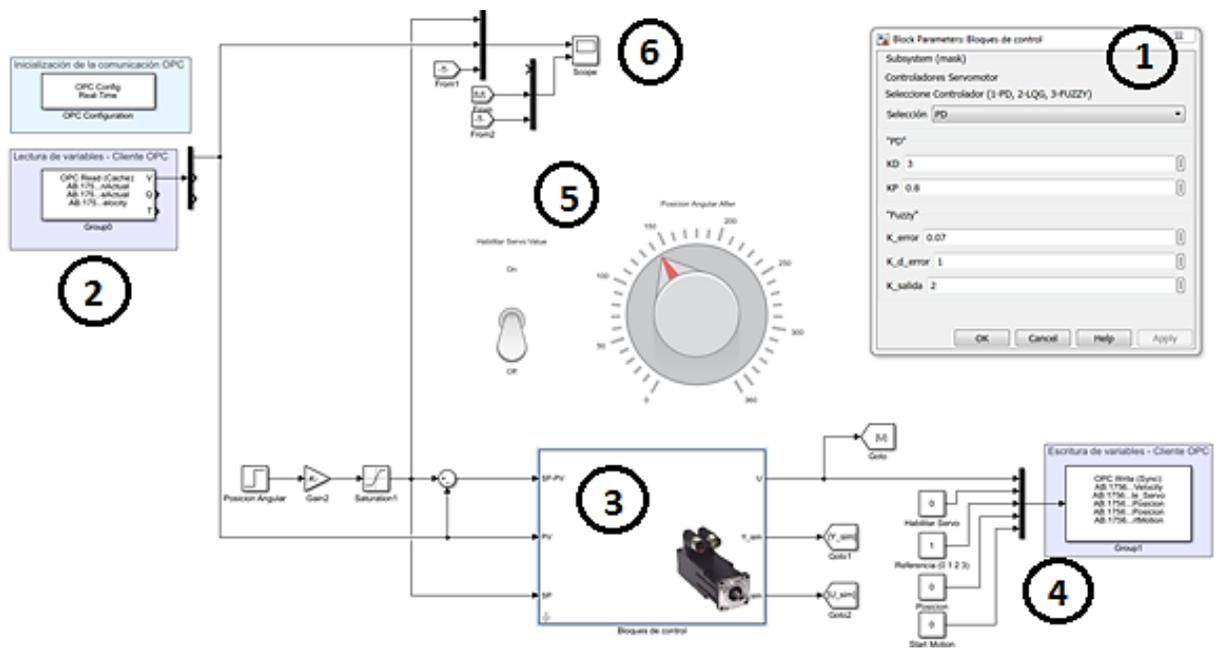


Figura 46 Interfaz para control de posición del servomotor industrial.

En las figuras anteriores se muestran las interfaces creadas, ambas interfaces tienen secciones que corresponden a lo enumerado a continuación:

Numero 1: Selección de controladores: En esta sección se elige el tipo de controlador, y las ganancias por si se desean modificar para ver cómo varía la respuesta del sistema.

Numero 2: Lectura de datos: En esta sección se reciben los datos enviados desde el servidor OPC, son los datos de los cuales se requiere lectura como lo son los encoder o incluso la corriente que se está suministrando por algún actuador.

Numero 3: Controlador: En esta sección se encuentran en un subsistema los tres controladores implementados y sus respectivas simulaciones. Esta es una de las ventajas que ofrece usar Matlab, ya que se pueden usar los bloques junto con la función de transferencia y obtener la simulación de la planta y compararla con los datos experimentales en tiempo real.

Numero 4: Envío de datos: En esta sección se envían los datos de escritura hacia el servidor OPC, son las variables del sistema (acción de control, encendido o apagado de motores o instrucciones de programación).

Numero 5: Variables de control: En esta sección se disponen herramientas para establecer las variables que el sistema va a tomar para iniciar el proceso, ya sea una instrucción de encendido o apagado e incluso el mismo valor de set point deseado.

Numero 6: Visualización de la respuesta: En esta sección se dispone un *Scope* o un display el cual va a ilustrar las variables que creamos necesarias, en este caso se desea mostrar el valor de consigna, el estado actual de la planta y de la simulación y sus acciones de control correspondientes.

3.1.2. LABVIEW

Este es el segundo software de programación utilizado, LabVIEW tiene la ventaja de ofrecer una interfaz HMI más amigable con el usuario a comparación de Matlab. En este software se integró la comunicación con los tres PLC utilizados (Siemens, AllenBradley y Festo) permitiéndonos tener un mayor dominio sobre los procesos utilizados, con este software se puede sacar mayor provecho en cuanto a la integración y visualización de procesos que trabajen con diferentes tecnologías. En la figura 47 se muestra el resultado de la interfaz que se usará en LabView (la imagen se encuentra en mayor tamaño en el anexo 6).

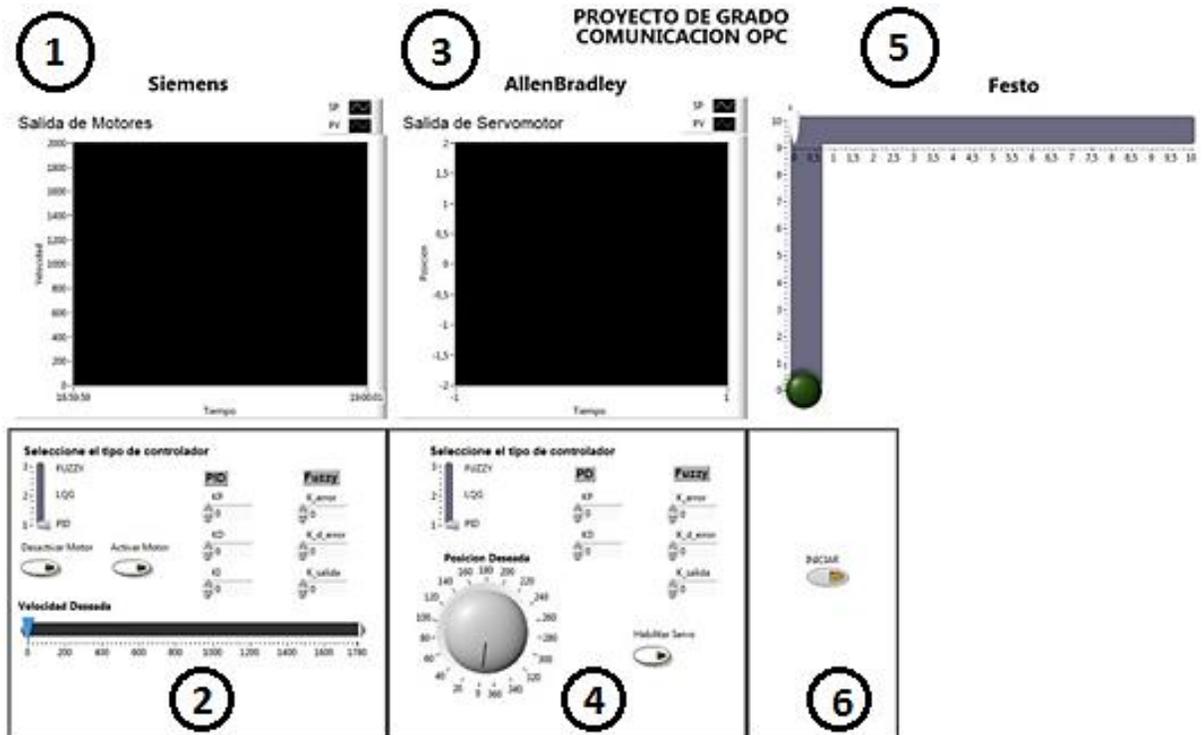


Figura 47 Interfaz HMI para control de Velocidad de un motor AC, Posición de un servomotor industrial y secuencia de MPS - Festo.

En esta interfaz diseñada en LabView, también se puede seleccionar el tipo de controlador y el punto de consigna al que se desea llegar para cada uno de los sistemas, como se mencionaba anteriormente, en esta interfaz se puede tener el

control de los tres sistemas al tiempo (algo que también se puede hacer en Matlab con la diferencia que este software no es tan amigable con el operario).

Para el sistema de MPS (Controlador Festo) se realiza un seguimiento del estado de la secuencia, se activa mediante la pulsación de un botón y el sistema espera a detectar una pieza, en ese momento comienza a realizar la secuencia la cual es graficada por la interfaz de LabView.

En la figura anterior (figura 47) se muestra la interfaz creada para LabView, se dividió en secciones que corresponden a lo enumerado a continuación:

Numero 1: Sección de visualización (Motor): En esta sección se visualiza la respuesta de velocidad del sistema (en este caso el motor AC) y su respectiva acción de control.

Numero 2: Sección de control (Motor): En esta sección se selecciona el controlador a usar, sus respectivas ganancias y el valor del punto de consigna al que se desea llegar (velocidad), también se puede encender o apagar el motor si es necesario.

Numero 3: Sección de visualización (servomotor): En esta sección se visualiza la respuesta de posición del sistema (en este caso del servomotor) y su respectiva acción de control.

Numero 4: Sección de control (servomotor): En esta sección se selecciona el controlador a usar, sus respectivas ganancias y el valor del punto de consigna al que se desea llegar (posición), también se puede encender o apagar el servomotor motor si es necesario.

Numero 5: Sección de visualización (MPS): En esta sección se visualiza el estado de la secuencia de la MPS (planta de secuencia de Festo).

Numero 6: Sección de control (MPS): En esta sección se activa o desactiva el inicio de la secuencia de la MPS.

3.2. VALIDACION DE LAS ESTRATEGIAS DE CONTROL

A continuación, se expondrán los resultados de los controladores implementados a cada sistema. En este caso se compararán las respuestas obtenidas del modelo simulándolo contra la respuesta de los datos experimentales en tiempo real.

3.2.1. CONTROL DE VELOCIDAD

3.2.1.1. CONTROL PID

Para implementar el controlador PID en el sistema, se usó el modelo obtenido en la sección 2.2.1.2 junto con las constantes K_P , K_I y K_D obtenidas en esta misma sección. Se evaluará inicialmente la respuesta real del controlador comparada con la respuesta de la simulación (figura 48).

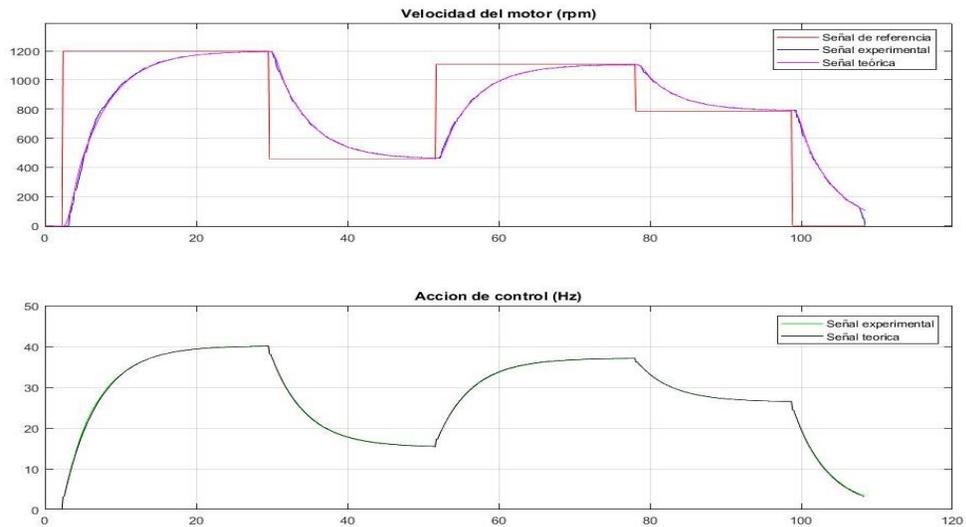


Figura 48 Datos experimentales vs datos teóricos implementando un control PID.

Con esta grafica se puede comprobar que la planta hallada hace un buen seguimiento de la planta, lo que quiere decir que el modelo que se usó inicialmente corresponde correctamente al comportamiento del sistema. Se tienen tiempos de establecimientos aproximados de 23 segundos, pero se garantiza que la señal no presente sobrepaso.

3.2.1.2. CONTROL LQG

Para implementar el controlador LQG, se usó el modelo previamente mencionado junto con los parámetros establecidos en la sección 2.2.1.3 además de las constantes K_{est} (ganancias que multiplican los estados), K_I (ganancia integral). En la figura 49 se muestra la respuesta de la planta comparada con la simulación.

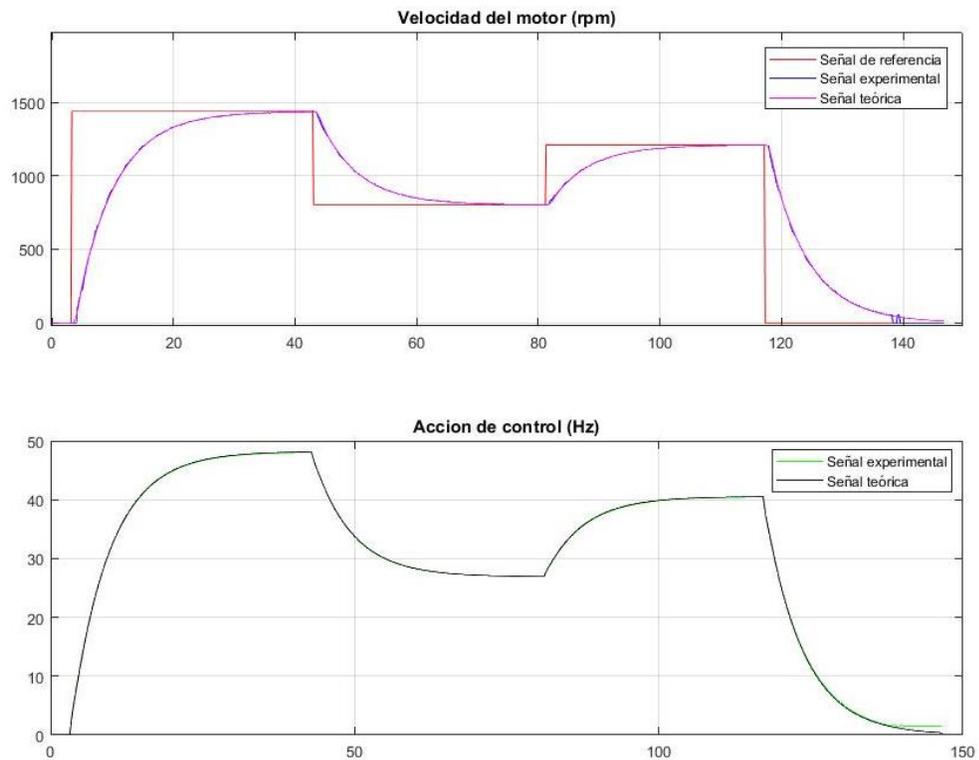


Figura 49 Datos experimentales vs datos teóricos implementando un control LQG.

Con la gráfica anterior se puede comprobar que la planta hallada hace un buen seguimiento de la planta real, lo que quiere decir que el modelo que se usó inicialmente corresponde correctamente al comportamiento del sistema. Se tienen tiempos de establecimientos aproximados de 28 segundos, pero se garantiza que la señal nunca exceda la señal de referencia.

3.2.1.3. CONTROL FUZZY

Para implementar el controlador Fuzzy, se usó el modelo de bloques previamente mencionado junto con los parámetros establecidos en la sección 2.2.1.4 además de las constantes K_{error} , K_{d_error} y K_{salida} obtenidas en esta misma sección. Se evaluará inicialmente la respuesta real del controlador comparada con la respuesta de la simulación (figura 50).

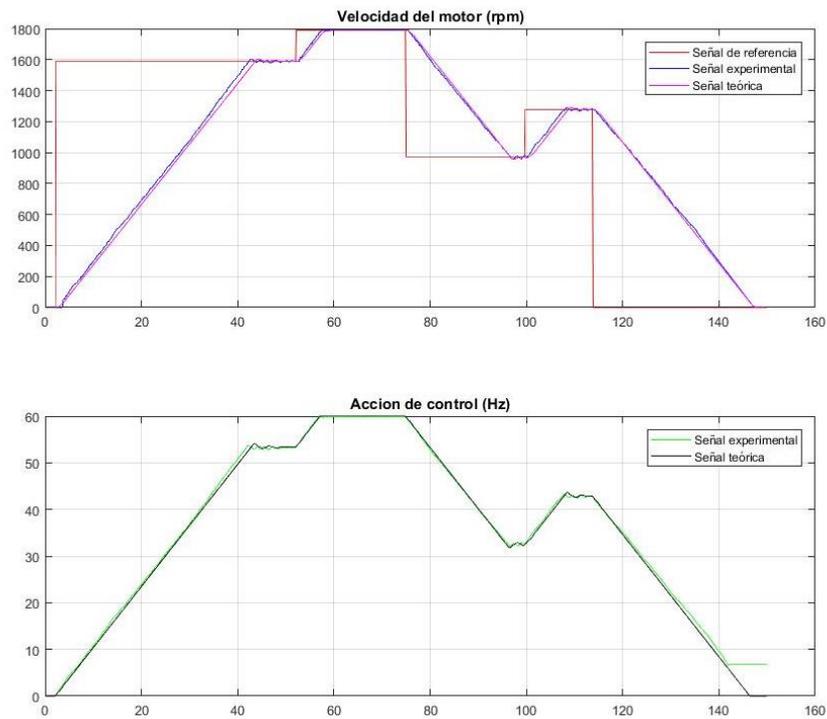


Figura 50 Datos experimentales vs datos teóricos implementando un control Fuzzy.

Con la gráfica anterior se puede comprobar que la planta hallada hace un buen seguimiento de la planta real, el controlador presenta una pequeña oscilación en el valor de consigna antes de lograr estabilizarse, se tienen tiempos de establecimientos aproximados de 40 segundos. Para este tipo de planta en la cual los controladores anteriores lograron tiempos de 25 segundos o menos podemos decir que no es un controlador adecuado para la aplicación.

3.2.1.4. INDICES DE ERROR

Los índices de comportamiento se definen como un valor que sirve como parámetro para evaluar la calidad de la respuesta de un sistema ante una entrada. Así, la selección del mejor controlador dependerá del índice de comportamiento que se obtenga, donde el mejor índice será el que tenga el mínimo rango de error. Para esta prueba se aplicaron cuatro índices de desempeño:

- Integral del error absoluto (IAE) que está dada por la ecuación en tiempo discreto (3.1):

$$IAE = Ts \sum_{k=0}^N |e(k)|$$

- Integral del error cuadrático (ISE) que está dado por la ecuación en tiempo discreto (3.2):

$$ISE = Ts \sum_{k=0}^N e(k)^2$$

- Integral del tiempo por el error absoluto (ITAE) que está dado por la ecuación en tiempo discreto (3.3):

$$ITAE = Ts \sum_{k=0}^N t(k)|e(k)|$$

- Integral del tiempo por el error cuadrático (ITSE) que está dado por la ecuación en tiempo discreto (3.4):

$$ITSE = Ts \sum_{k=0}^N t(k)e(k)^2$$

Para evaluar cada índice se sometió a la planta a un mismo valor de referencia con los tres diferentes controladores en un tiempo aproximadamente similar (figura 51), de esta forma, se puede analizar el resultado de los índices de mejor manera.

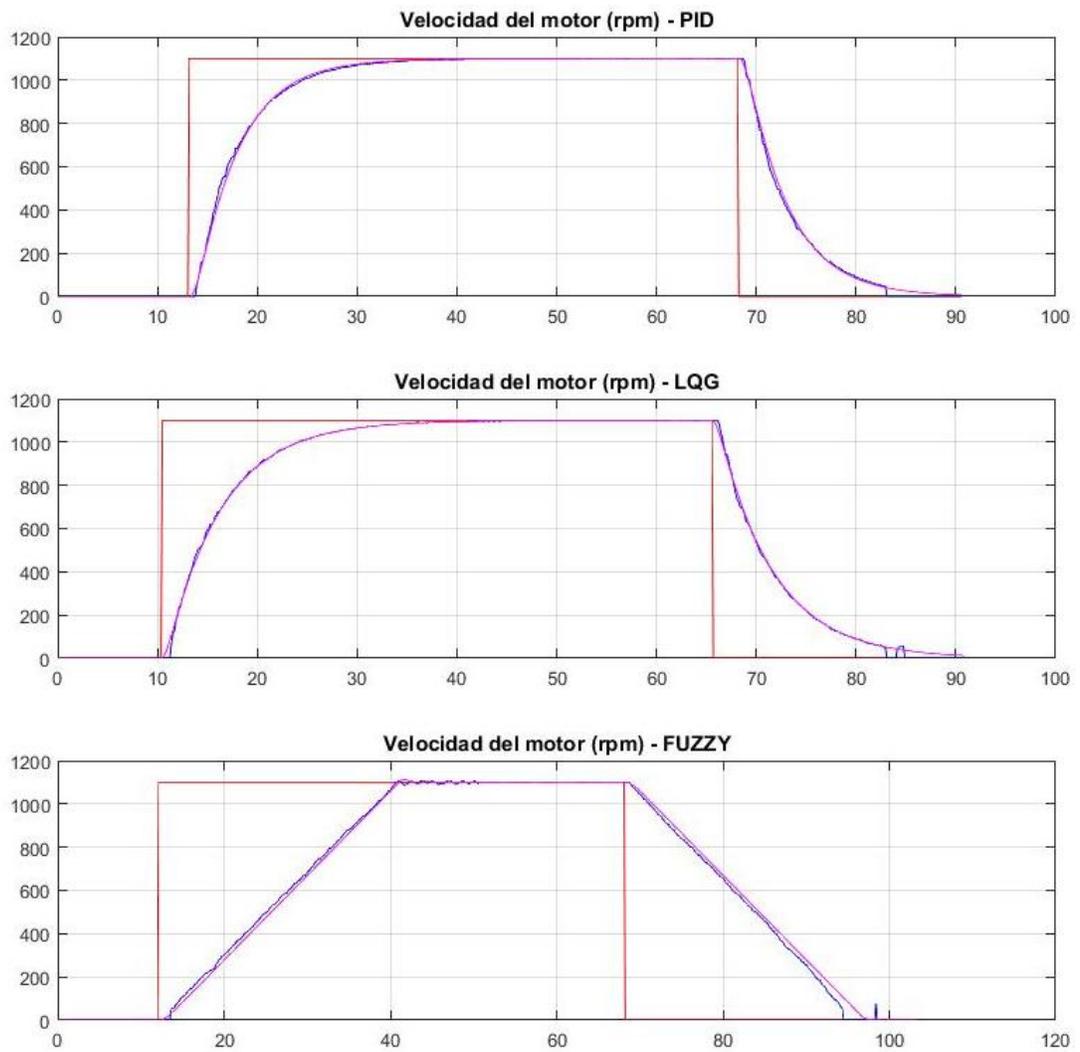


Figura 51 Comparación de los controladores ante una misma entrada escalón.

La tabla 4 presenta los resultados obtenidos para los índices previamente mencionados.

Tabla 4 Índices de error - Control de velocidad.

	IAE	ISE	ITAE	ITSE
PID	3.8221e+03	8.3496e+04	2.0985e+05	4.4346e+06
LQG	3.0342e+03	7.5396e+04	1.8619e+05	4.6036e+06
FUZZY	8.9975e+03	2.6042e+05	5.5172e+05	1.8721e+07

Según los datos mostrados en la tabla 4, el controlador PID resulta ser la mejor opción en la implementación de control para velocidad dado que el tiempo de estabilización se realiza en menor tiempo y tiene índices de error similares al control LQG, sin embargo, el controlador LQG presenta a su vez un comportamiento sobresaliente frente a las perturbaciones del sistema y los puntos de referencia en la etapa de seguimiento.

3.2.2. CONTROL DE POSICIÓN

3.2.2.1. CONTROL PD

Para implementar el controlador PD en el sistema de control de posición, se usó el modelo obtenido en la sección 2.2.2.2 junto con las constantes K_P , y K_D obtenidas en esta misma sección. Se evaluará inicialmente la respuesta real del controlador comparada con la respuesta de la simulación (figura 52), la idea es demostrar que el modelo hallado se asemeja a la respuesta real de la planta.

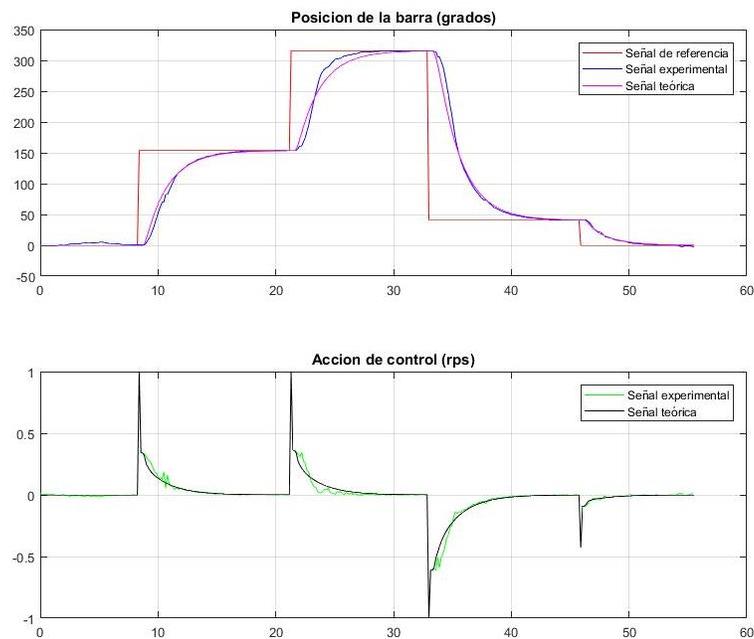


Figura 52 Datos experimentales vs datos teóricos implementando un control PID.

Con esta grafica se puede comprobar que la planta hallada hace un buen seguimiento de la planta, lo que quiere decir que el modelo que se usó inicialmente corresponde correctamente al comportamiento del sistema. Se tienen tiempos de establecimientos aproximados de 9.5 segundos, pero se garantiza que la señal nunca exceda la señal de referencia.

3.2.2.2. CONTROL LQG

Para implementar el controlador LQG, se usó el modelo previamente mencionado junto con los parámetros establecidos en la sección 2.2.2.3 además de las constantes K_{est} (ganancias que multiplican los estados), K_I (ganancia integral). En la figura 53 se muestra la respuesta de la planta comparada con la simulación.

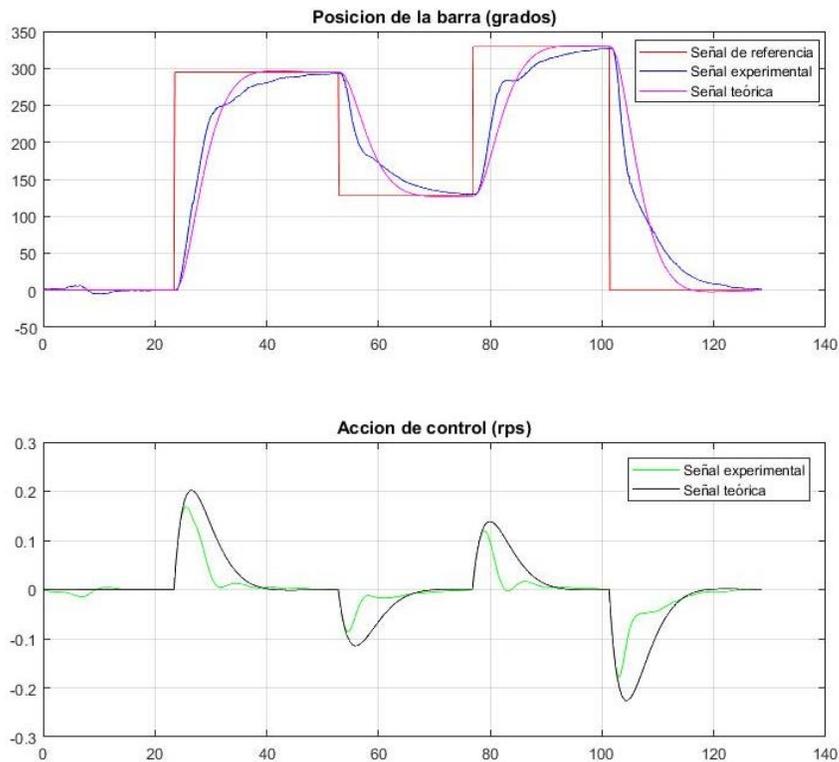


Figura 53 Datos experimentales vs datos teóricos implementando un control LQG.

Con la gráfica anterior se puede comprobar que el sistema no hace un seguimiento continuo, pero se establece en tiempos aproximados a los de simulación. Esto es común el diseño de controladores LQG ya que para el diseño de este controlador se debe tener una función de transferencia que identifique mejor el sistema. Se tienen tiempos de establecimientos aproximados de 20 segundos. Lo cual es lento comparado con los otros dos controladores utilizados (PD y Fuzzy).

3.2.2.3. CONTROL FUZZY

Para implementar el controlador Fuzzy, se usó el modelo de bloques previamente mencionado junto con los parámetros establecidos en la sección 2.2.2.4 además de las constantes K_{error} , K_{d_error} y K_{salida} obtenidas en esta misma sección. Se evaluará inicialmente la respuesta real del controlador comparada con la respuesta de la simulación (figura 54), la idea es demostrar que el modelo hallado se asemeja a la respuesta real de la planta.

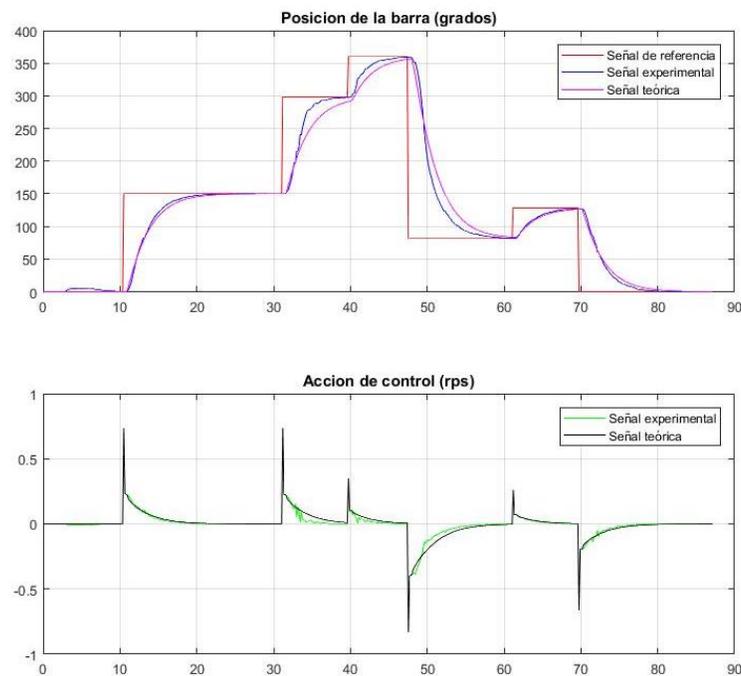


Figura 54 Datos experimentales vs datos teóricos implementando un control Fuzzy.

Dada la gráfica anterior se puede comprobar que la planta hallada hace un buen seguimiento de la planta real y se tienen tiempos de establecimientos aproximados de 10 segundos.

3.2.2.4. INDICES DE ERROR

Para evaluar cada índice de error se tomaron en cuenta las ecuaciones mencionadas en el numeral 3.2.1.4, se sometió a la planta a un mismo valor de referencia con los tres diferentes controladores en un tiempo aproximadamente similar (figura 55), de esta forma, se puede analizar el resultado de los índices de mejor manera.

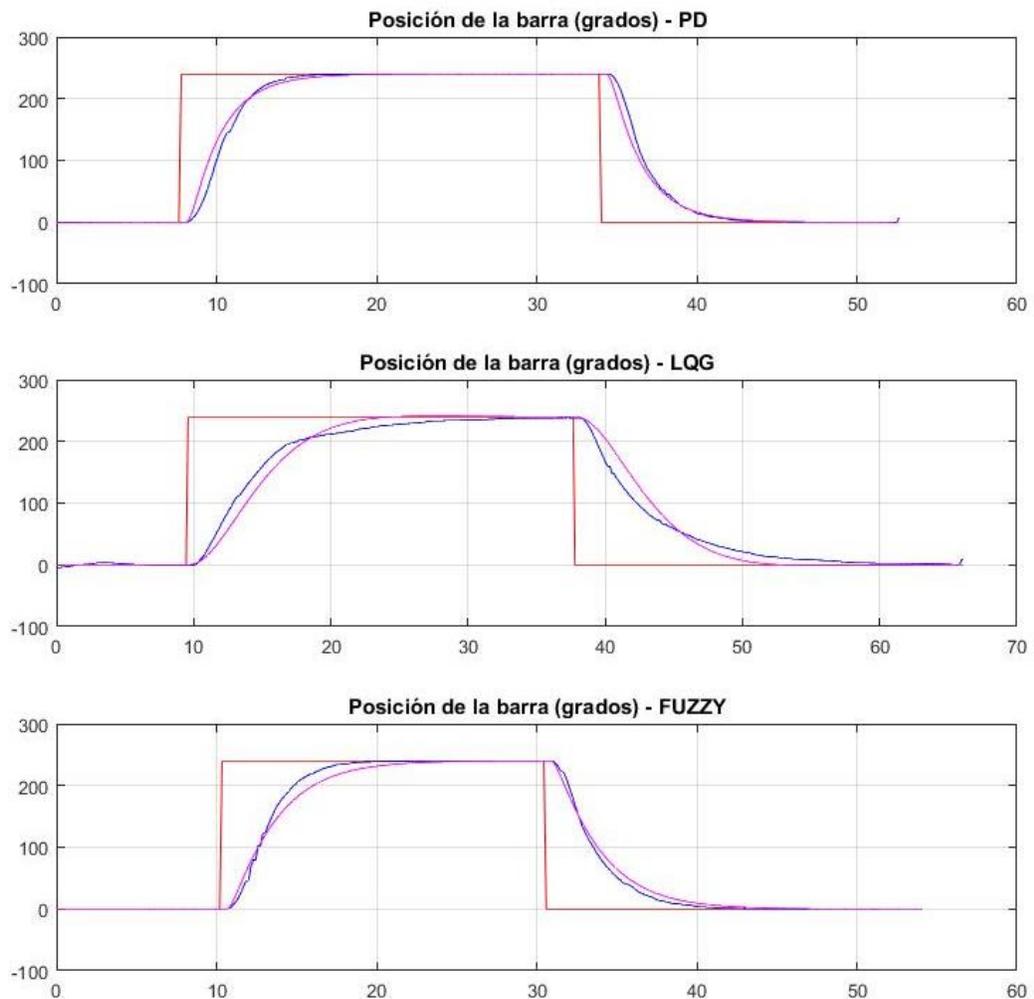


Figura 55 Comparación de los controladores ante una misma entrada escalón.

La tabla 5 presenta los resultados obtenidos para los índices previamente mencionados.

Tabla 5 Índices de error - Control de posición.

	IAE	ISE	ITAE	ITSE
PID	8.5842e+03	1.2858e+06	1.9689e+05	2.8290e+07
LQG	1.7602e+04	2.8669e+06	4.8151e+05	7.4792e+07
FUZZY	1.1273e+04	1.6408e+06	2.6143e+05	3.6017e+07

Según los datos mostrados en la tabla 4, el controlador PID resulta ser la mejor opción en la implementación del control de posición dado que el tiempo de estabilización se realiza en menor tiempo y tiene índices menores a los otros dos controladores, sin embargo, el controlador Fuzzy presenta a su vez un comportamiento sobresaliente en cuanto a los índices de error y su tiempo de estabilización.

CONCLUSIONES Y RECOMENDACIONES

- El estándar de comunicación OPC es una herramienta bastante útil ya que permite integrar diferentes tipos de tecnologías en una misma interfaz HMI, se pueden controlar cantidades enormes de diferentes dispositivos simplemente conectados a una red, además del hecho de permitir ser trabajados por softwares como lo son Matlab y LabView, que son programas diseñados para cálculos más complejos que desde un PLC convencional no se pueden realizar. Además de brindarnos diferentes tipos de programaciones diferentes, es un protocolo que integra y brinda la facilidad al usuario final.
- Cada uno de los softwares de comunicación ofrece una ventaja característica, en el caso de Matlab, obtenemos que los tiempos de comunicación no varían, independientemente de la cantidad de variables que se estén comunicando, o la complejidad de la programación de los sistemas de control, aun así, Matlab no ofrece la posibilidad que ofrece el software LabView, de poder integrar gráficamente en un panel de control toda la visualización y las instrucciones para cada sistema. Aun así, independiente de que cada software presente ventajas o desventajas, ambos son capaces de realizar la integración de diferentes tecnologías.
- Para cada sistema se seleccionó la mejor estrategia de control basado en los índices de error y en su tiempo de estabilización, para el control de velocidad se definió que la mejor estrategia de control es el control LQG, ya que realiza un mejor seguimiento de la variable del proceso simulada y se establece en un tiempo considerable dado el planteamiento del sistema. En el control de velocidad se eligió el controlador PD como la mejor estrategia ya que

presenta los menores índices de error y un tiempo de establecimiento mejor comparado con los otros controladores

- La eficacia de los controladores se vio afectada por el periodo de muestreo que se tuvo en el desarrollo del proyecto, en este caso tuvimos un periodo de muestreo de 0.15 segundos, esto dado a la limitante de trabajar con una versión gratuita de los servidores OPC, si se contara con una velocidad de comunicación mayor, los tiempos de establecimiento de los sistemas no serían tan altos. Aun así, la ventaja que ofrece esta comunicación es grande comparada con la facilidad de modificar y visualizar los controladores realizados.
- El software LabView fue dejado de lado en el momento de realizar la comparación de los controladores dado que este software no garantizaba el periodo de muestreo establecido (0.15 segundos) por lo tanto la respuesta del sistema era diferente a la diseñada, 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, aun así, se concluye que para aplicaciones lentas (controles de temperatura u otros), LabView es una herramienta útil dada su interfaz gráfica.

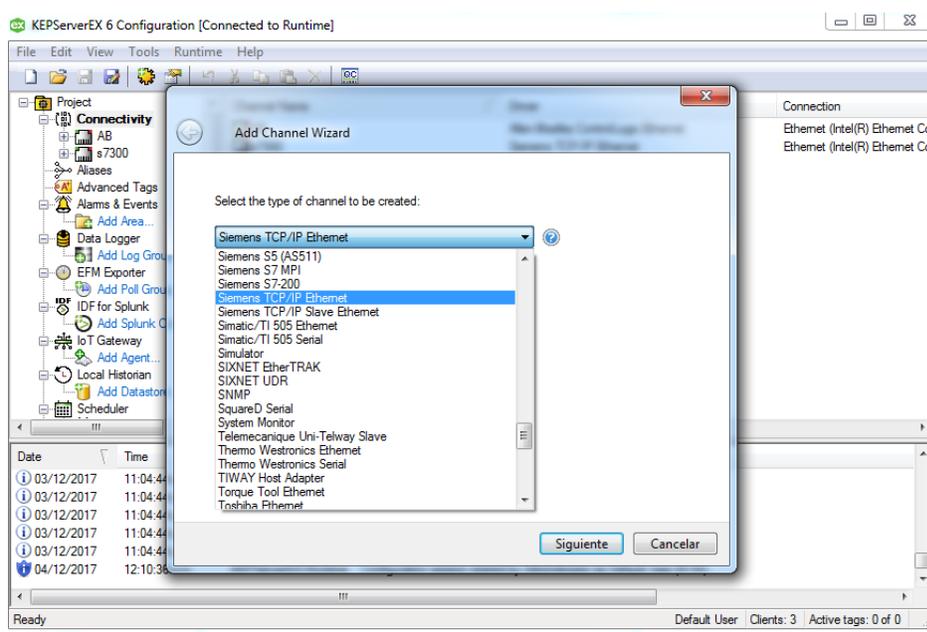
BIBLIOGRAFIA

- [1] Implementación de controladores en Matlab y plc con comunicación opc aplicado a plantas industriales. Manuel Manyari. Febrero 2, 2008
- [2] Centro integrado politécnico. OPC: UN ESTANDAR EN LAS REDES INDUSTRIALES Y BUSES DE CAMPO, Obtenido en <http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewintroduccionopcserver.pdf>, el día 8 de enero de 2017
- [3] Experimental Validation of PID Based Cascade Control System through SCADA-PLC-OPC Interface. A. Lakshmi, Sangeetha B, enero 10, 2012
- [4] On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology. Zhang Lieping, Zeng Aiqun. China. Julio 26, 2007
- [5] Automation of shell and tube type heat exchanger with PLC and LabVIEW. Tushar V, Shripad S, G. Jamakar. India. Mayo 28, 2015
- [6] Universidad del Cauca. Redes de comunicación industrial. Obtenido en: <ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Redes%20Industriales/Redes/Conferencias/Capitulo%201.pdf>, el día 17 de enero de 2018.
- [7] Matlab Toolbox OPC guide. Obtenido de <https://www.mathworks.com/products/opc.html>, el día 5 de enero de 2017.
- [8] National Instruments. Conecte LabVIEW a Cualquier PLC. Obtenido de <http://www.ni.com/white-paper/7906/es/>, el día 5 de enero de 2017.
- [9] Sistemas de control el tiempo discreto. Ogata K. Prentice Hall, 1996.
- [10] Model Predictive Control System Design and Implementation Using MATLAB. Wang, Liuping. Melbourne, Australia: Springer, 2009.
- [11] Diseño de controladores LQR/LQG para su aplicación en sistemas de pilas de combustible tipo PEM. Ali Niknezhadi, Cristian Kunsch, Carlos Ocampo-Martínez. 2010. Barcelona, España: s.n., 2010.
- [12] Fundamentos de robótica. Antonio Barrientos. McGrawHill, 2012.
- [13] Leonardo J. Marín, Víctor M. Alfaro. "Sintonización de controladores por ubicación de polos y ceros". IEEE CONESCAPAN XXVI, 2007.
- [14] X. Lan Li, J. Gyu y H. Shin. "Comparison and evaluation of anti-windup PI Controllers". Journal of Power Electronics, Volume 11 No 1. Enero 2011.

ANEXO 1. CONFIGURACION SERVIDOR OPC – SIEMENS

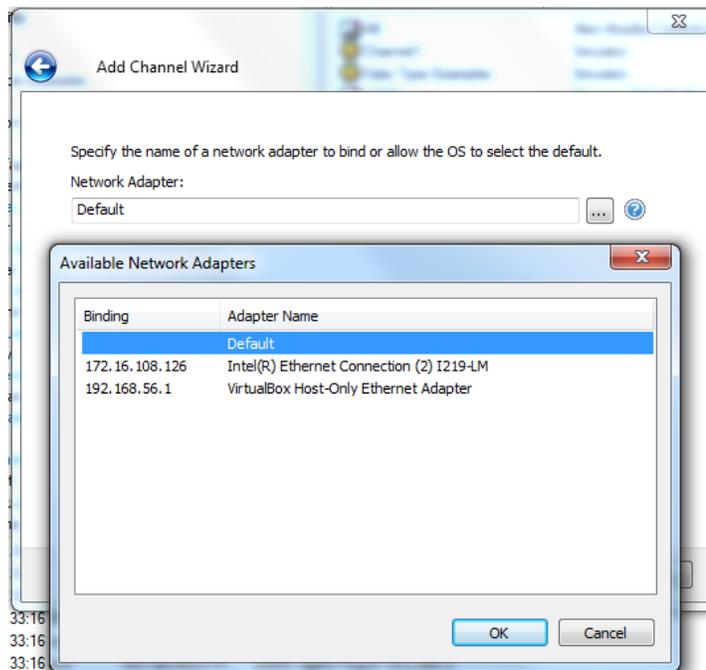
En el panel principal del servidor OPC KEPServerEX, dar clic derecho en Connectivity y seleccionar New Channel.

A continuación, aparecerá una ventana en la cual se selecciona el dispositivo con el cual se realizará el enlace de comunicación. En este caso ubicaremos la opción Siemens TCP/IP Ethernet, ya que en este caso la comunicación con el PLC Siemens se realizará vía Ethernet.

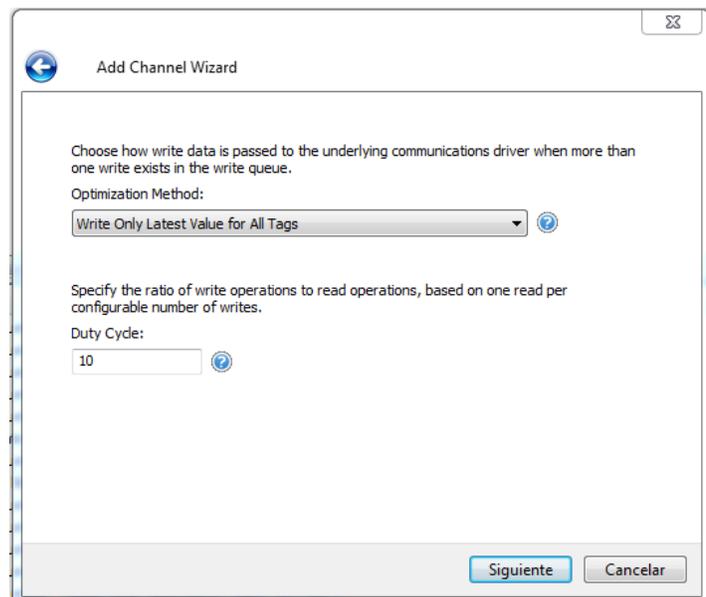


Luego de esto se establece un nombre al canal que trabajaremos.

Selección de la tarjeta de red a conectar, hay que asegurarse que la dirección de la tarjeta de red se encuentre en la misma red de nuestro dispositivo para que pueda reconocerlo.

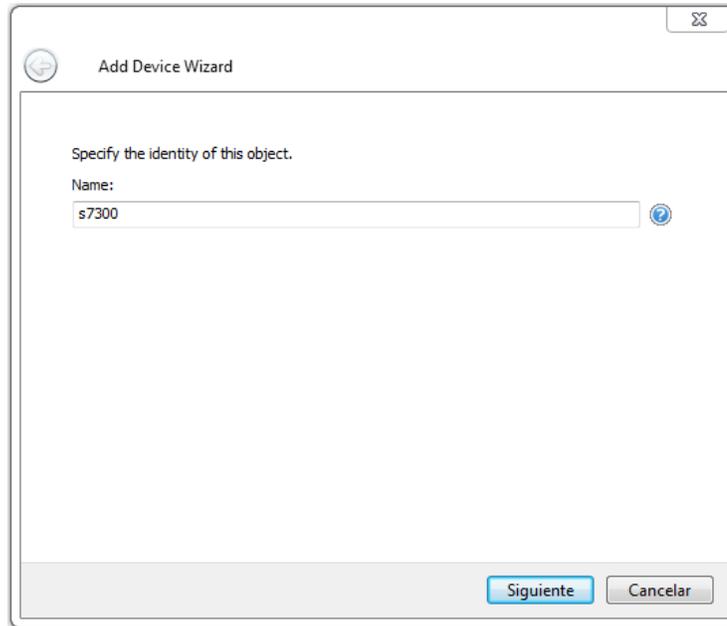


A continuación de esto se establece el tipo de lectura de datos y el radio de operación de escritura y lectura de datos, se deja por defecto en los valores establecidos

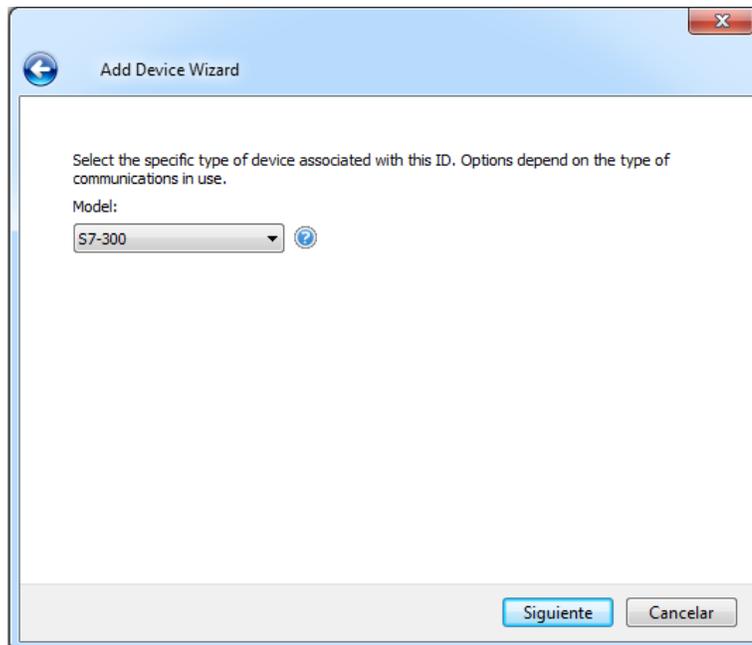


Se continúa hasta finalizar. Luego de finalizada la creación del canal, se procede a configurar el dispositivo con el cual se va a configurar.

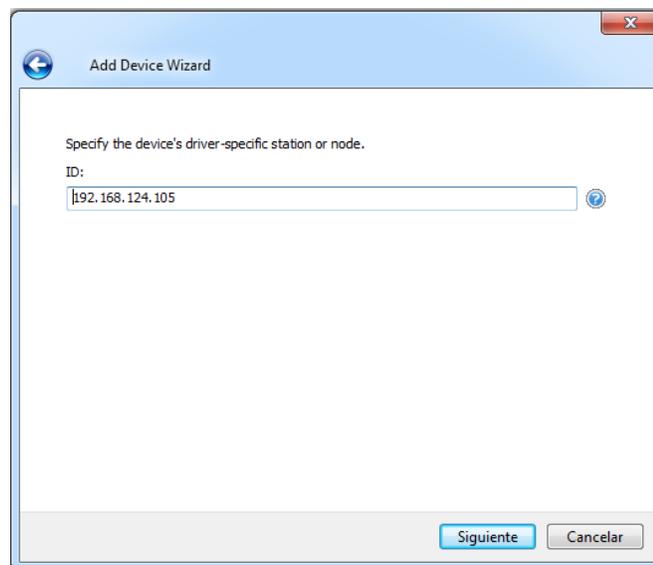
Se selecciona el canal, clic derecho y New Device (Nuevo dispositivo). Inmediatamente aparece una ventana donde inicialmente se solicita asignarle un nombre al dispositivo, en este caso se usó la referencia del PLC Siemens que se está trabajando.



Luego de esto se selecciona el tipo de controlador que es. En este caso se trabaja con un S7-300.

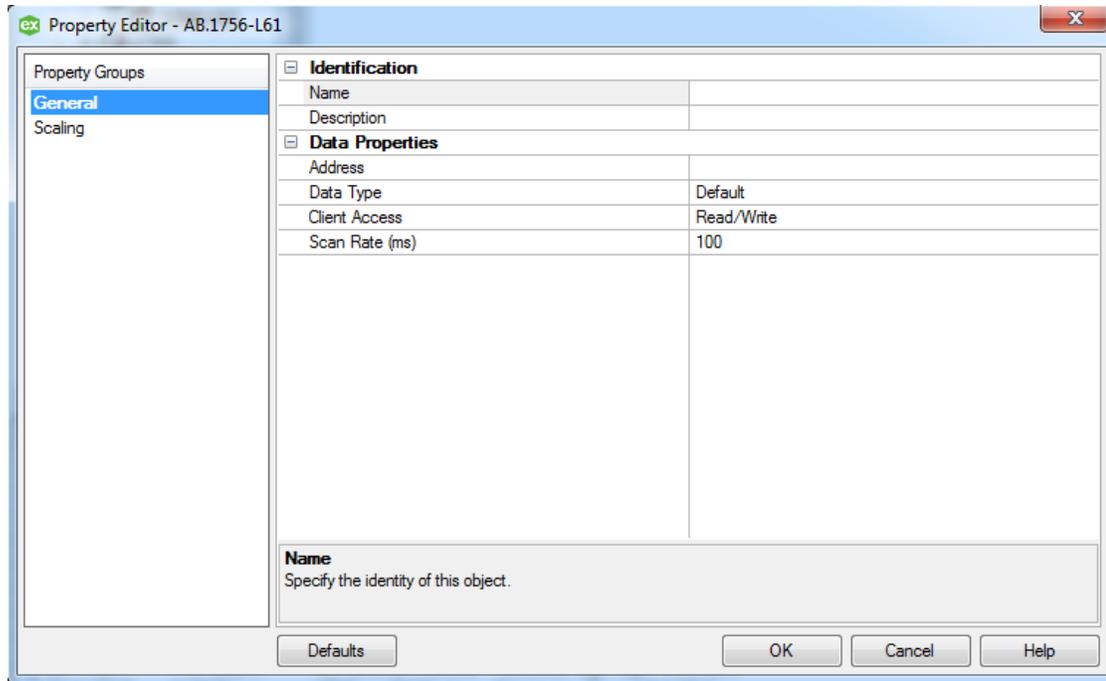


En la siguiente ventana se solicita la dirección IP que tiene configurada PLC. En este caso esta es la dirección IP que posee el PLC es “192.168.124.105”.



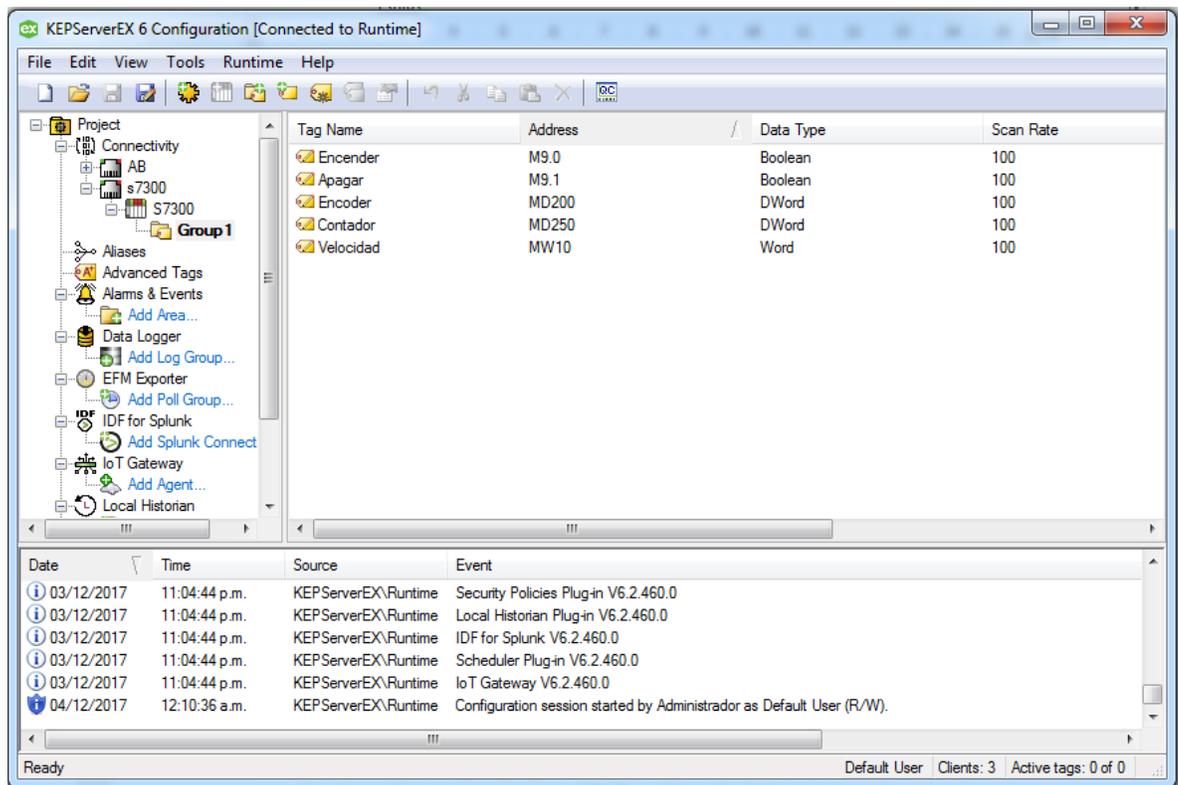
Luego de esto se continúa hasta finalizar dejando los valores por defecto de las demás características, en esta se habla del tipo de comunicación, del tiempo de detección de falla y demás situaciones. Luego de esto el dispositivo se ha creado y se continúa con la declaración de variables del servidor.

Para establecer que variables se van a leer del PLC es necesario crear en el servidor estas variables como TAGS (etiquetas), para esto se da clic derecho al dispositivo creado dentro del canal de comunicaciones, y se selecciona “New Tag” (nueva etiqueta).



En esta ventana se agregan las diferentes variables que se van a leer del PLC, cabe resaltar que las variables deben estar creadas en el programa del controlador lógico, se recomienda que sean variables que puedan identificarse fácilmente dado que la forma de declararlas en el servidor depende del tipo de variable que sea (booleana, flotante, double) y el nombre que posea en el programa del PLC.

Para esto esta ventana ofrece una ventana de ayuda en la cual se ejemplifica como se declara una variable según el PLC que se esté trabajando.

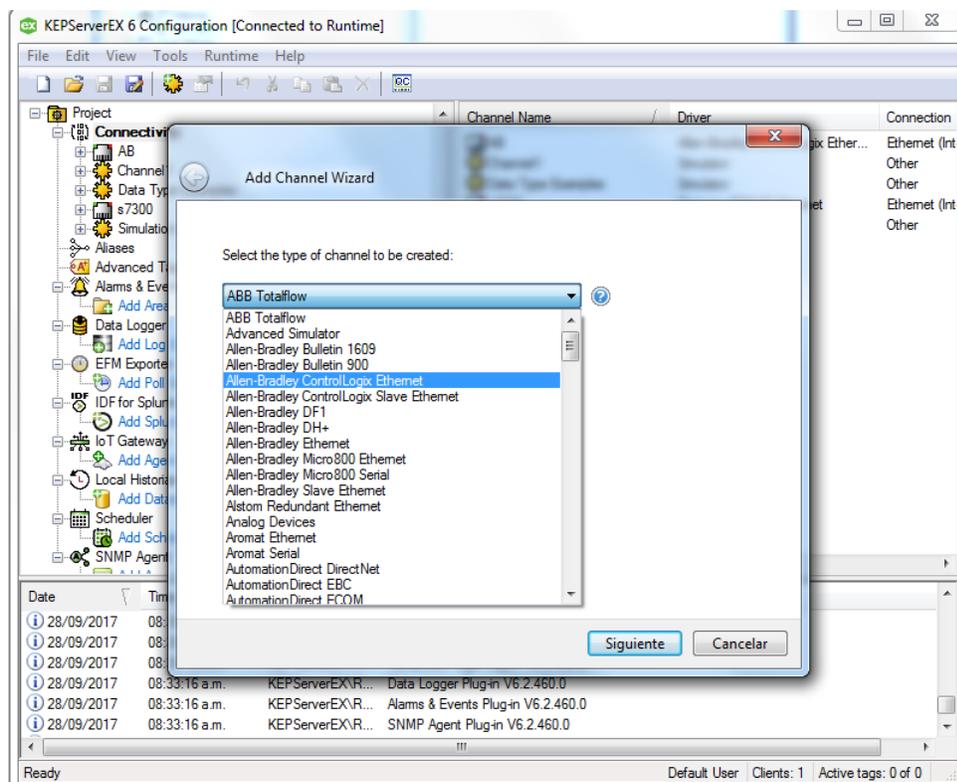


Al agregar las variables que se requieren el servidor tendrá una apariencia como esta, en la cual, los tags agregados aparecen mostrando el nombre con el que fue creado, el nombre que posee en el PLC y el tipo de dato que maneja.

ANEXO 2. CONFIGURACION SERVIDOR OPC – ALLEN BRADLEY

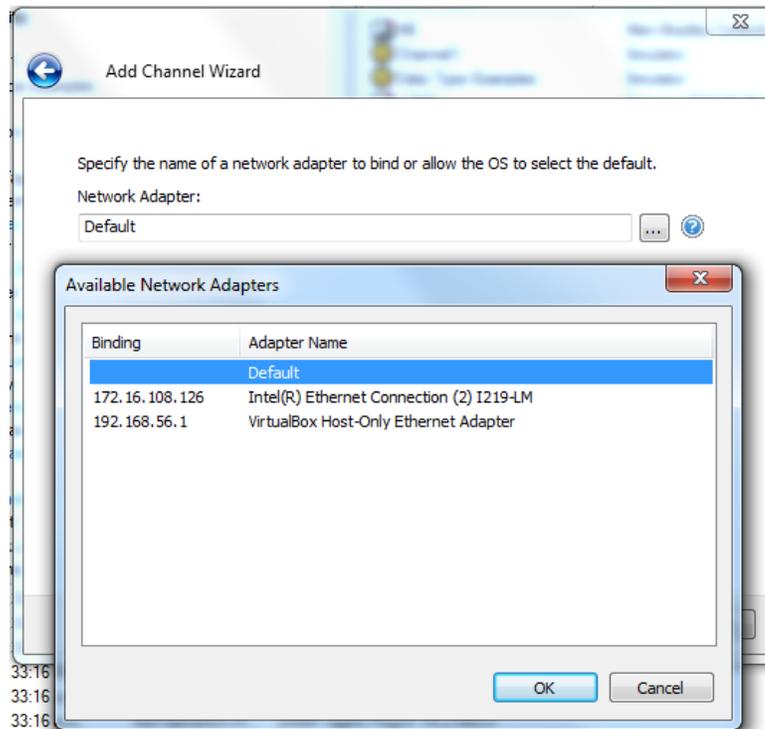
En el panel principal del servidor OPC KEPServerEX, dar clic derecho en Connectivity y seleccionar New Channel.

A continuación, aparecerá una ventana en la cual se selecciona el dispositivo con el cual se realizará el enlace de comunicación. En este caso ubicaremos el Allen-Bradley ControlLogix Ethernet, ya que en este caso la comunicación se realizará vía Ethernet.

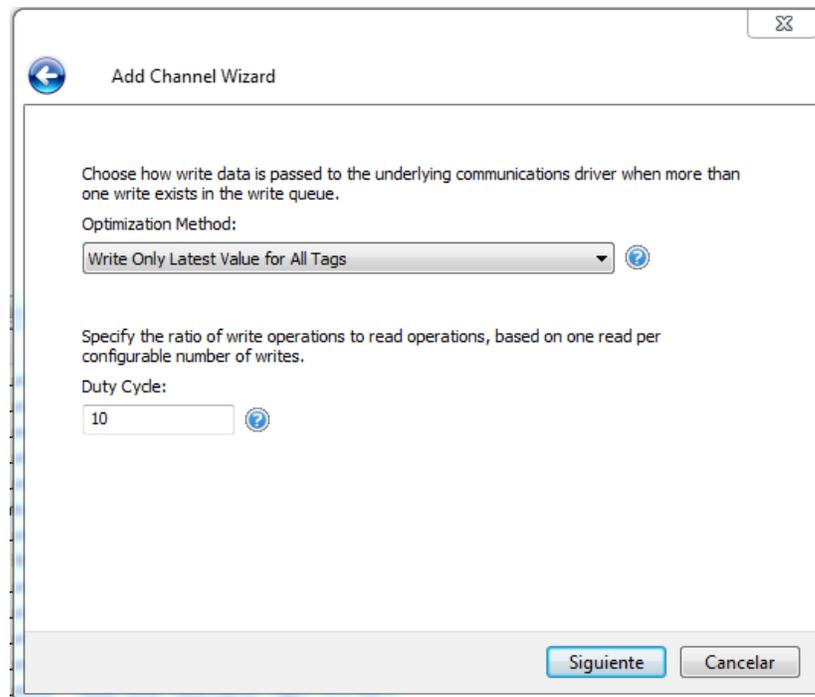


Luego de esto se establece un nombre al canal que trabajaremos.

Selección de la tarjeta de red a conectar, hay que asegurarse que la dirección de la tarjeta de red se encuentre en la misma red de nuestro dispositivo para que pueda reconocerlo.

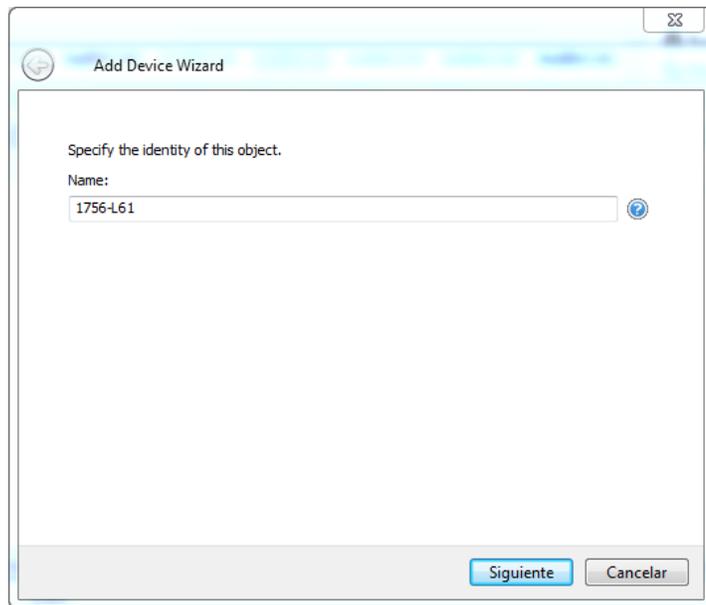


Luego de esto se establece el tipo de lectura de datos y el radio de operación de escritura y lectura de datos, se deja por defecto en los valores establecidos

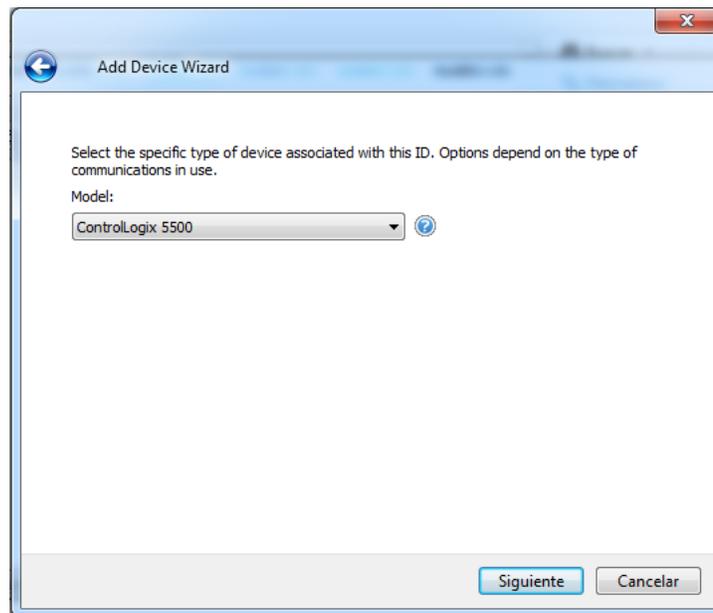


Se continúa hasta finalizar. Luego de finalizada la creación del canal, se procede a configurar el dispositivo con el cual se va a configurar.

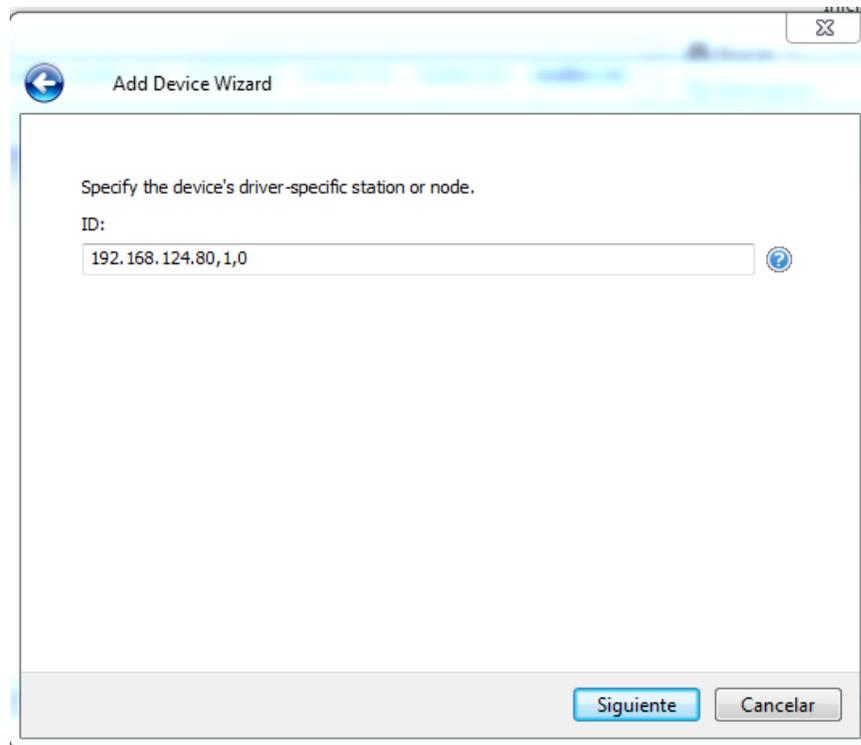
Se selecciona el canal, clic derecho y New Device (Nuevo dispositivo). Inmediatamente aparece una ventana donde inicialmente se solicita asignarle un nombre al dispositivo, en este caso se usó la referencia del PLC Allen Bradley que se está trabajando.



Luego de esto se selecciona el tipo de controlador que es. En este caso se trabaja con un ControlLogix 5500.

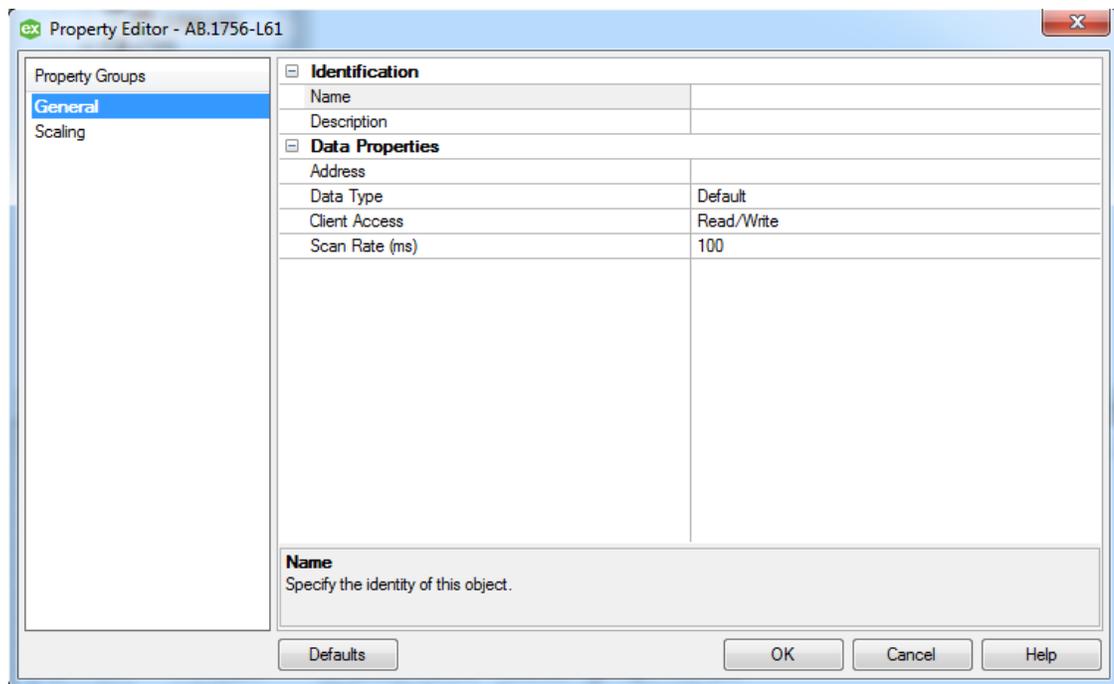


En la siguiente ventana se solicita algunas características del PLC como la dirección IP y la posición en el rack que se encuentra la CPU de este mismo. En este caso esta es la dirección IP que posee el PLC (192.168.124.80) y el puesto en el rack de la CPU es el slot "0"



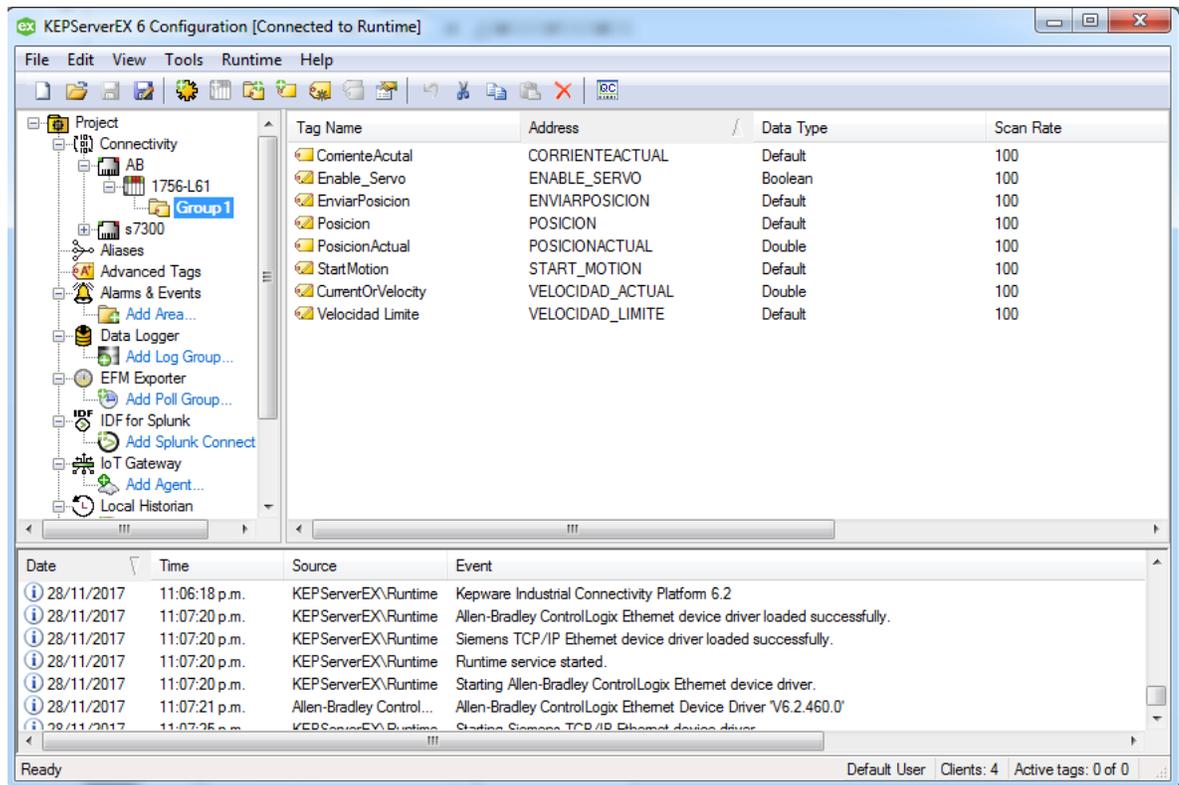
Luego de esto se continúa hasta finalizar dejando los valores por defecto de las demás características, en esta se habla del tipo de comunicación, del tiempo de detección de falla y demás situaciones. Luego de esto el dispositivo se ha creado y se continúa con la declaración de variables del servidor.

Para establecer que variables se van a leer del PLC es necesario crear en el servidor estas variables como TAGS (etiquetas), para esto se da clic derecho al dispositivo creado dentro del canal de comunicaciones, y se selecciona "New Tag" (nueva etiqueta).



En esta ventana se agregan las diferentes variables que se van a leer del PLC, cabe resaltar que las variables deben estar creadas en el programa del controlador lógico, se recomienda que sean variables que puedan identificarse fácilmente dado que la forma de declararlas en el servidor depende del tipo de variable que sea (booleana, flotante, double) y el nombre que posea en el programa del PLC.

Para esto esta ventana ofrece una ventana de ayuda en la cual se ejemplifica como se declara una variable según el PLC que se esté trabajando.

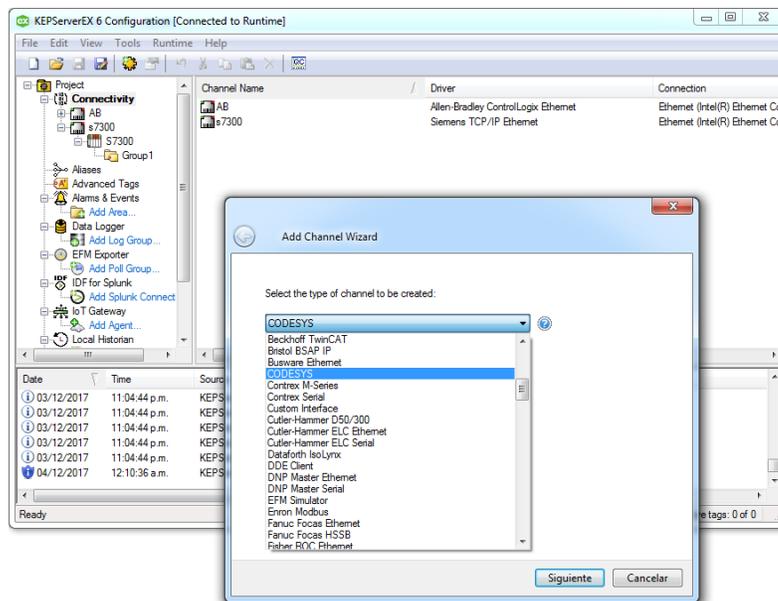


Al agregar las variables que se requieren el servidor tendrá una apariencia como esta, en la cual, los tags agregados aparecen mostrando el nombre con el que fue creado, el nombre que posee en el PLC y el tipo de dato que maneja.

ANEXO 3 CONFIGURACION SERVIDOR OPC – FESTO

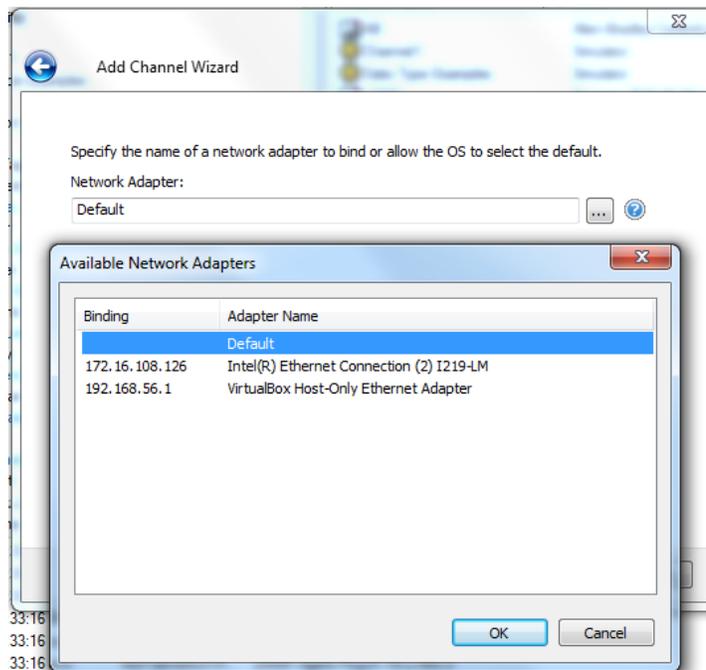
En el panel principal del servidor OPC KEPServerEX, dar clic derecho en Connectivity y seleccionar New Channel.

A continuación, aparecerá una ventana en la cual se selecciona el dispositivo con el cual se realizará el enlace de comunicación. En este caso ubicaremos la opción CODESYS, ya que en este caso la comunicación con el PLC Festo y se realizará vía Ethernet por medio del programa CODESYS, que es con el cual se programa este tipo de PLC.

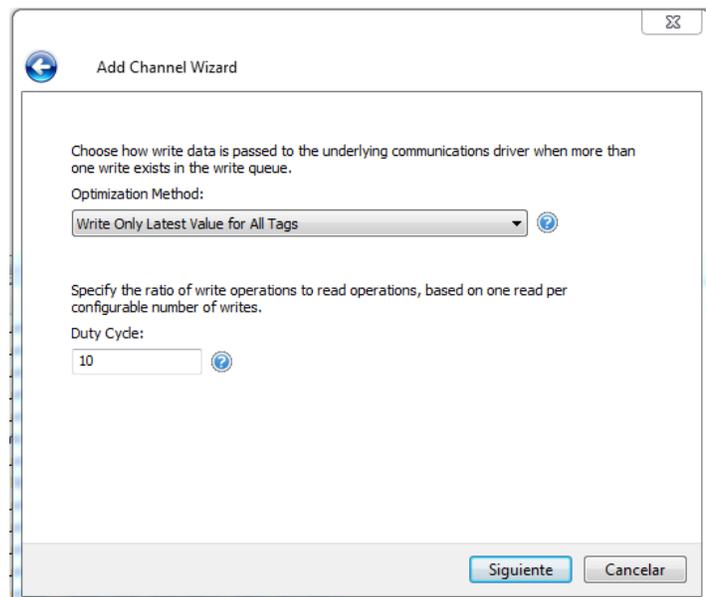


Luego de esto se establece un nombre al canal que trabajaremos.

Selección de la tarjeta de red a conectar, hay que asegurarse que la dirección de la tarjeta de red se encuentre en la misma red de nuestro dispositivo para que pueda reconocerlo.

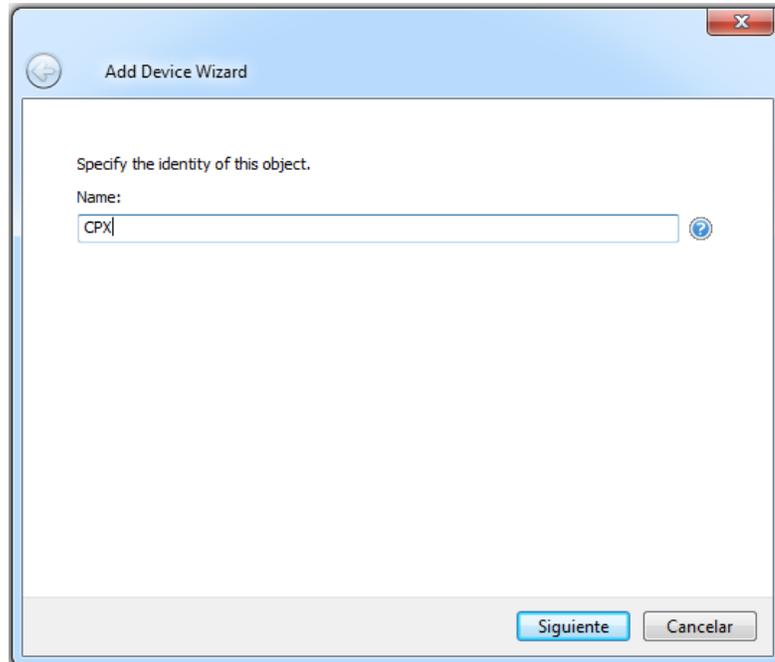


A continuación de esto se establece el tipo de lectura de datos y el radio de operación de escritura y lectura de datos, se deja por defecto en los valores establecidos

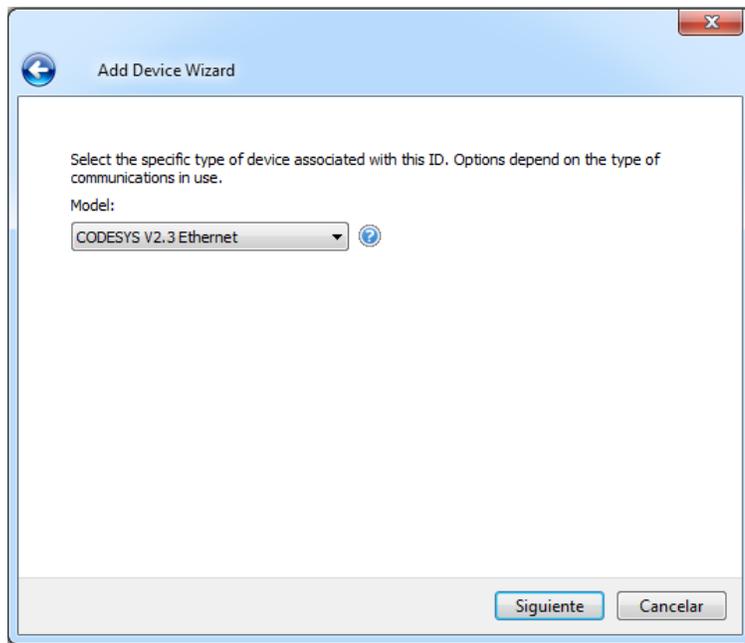


Se continúa hasta finalizar. Luego de finalizada la creación del canal, se procede a configurar el dispositivo con el cual se va a configurar.

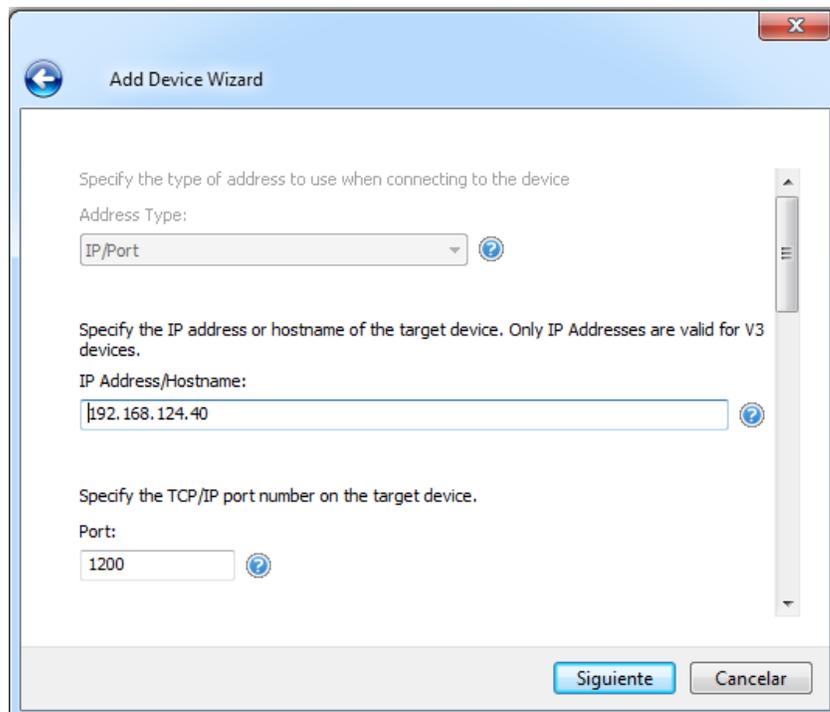
Se selecciona el canal, clic derecho y New Device (Nuevo dispositivo). Inmediatamente aparece una ventana donde inicialmente se solicita asignarle un nombre al dispositivo, en este caso se usó la referencia del PLC Festo que se está trabajando "CPX".



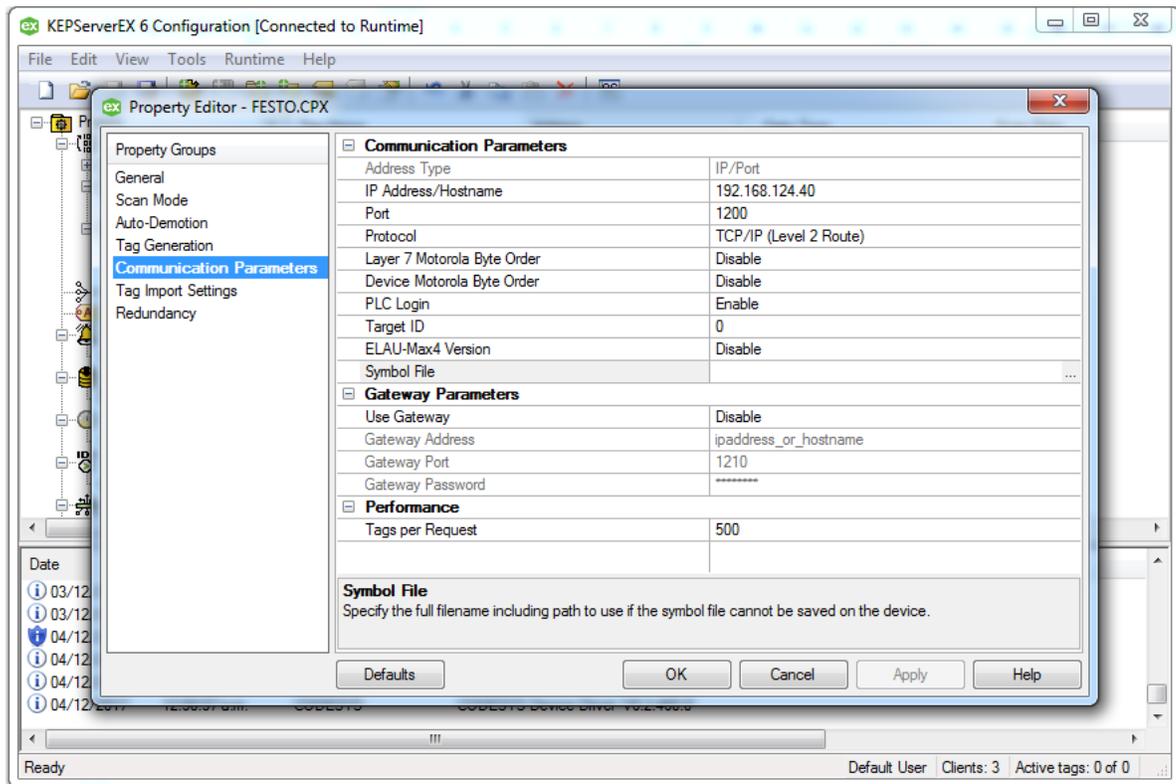
Luego de esto se selecciona la versión del software que se trabaja, en este caso la UNAB posee la versión 2.3 de CODESYS



En la siguiente ventana se solicita la dirección IP y el puerto de enlace que se configura en el programa CODESYS. En este caso esta es la dirección IP que posee el PLC es “192.168.124.40” y el puerto 1200.



Luego de esto se continúa hasta finalizar dejando los valores por defecto de las demás características, en esta se habla del tipo de comunicación, del tiempo de detección de falla y demás situaciones. Luego de esto el dispositivo se ha creado y se continúa con la declaración de variables del servidor.

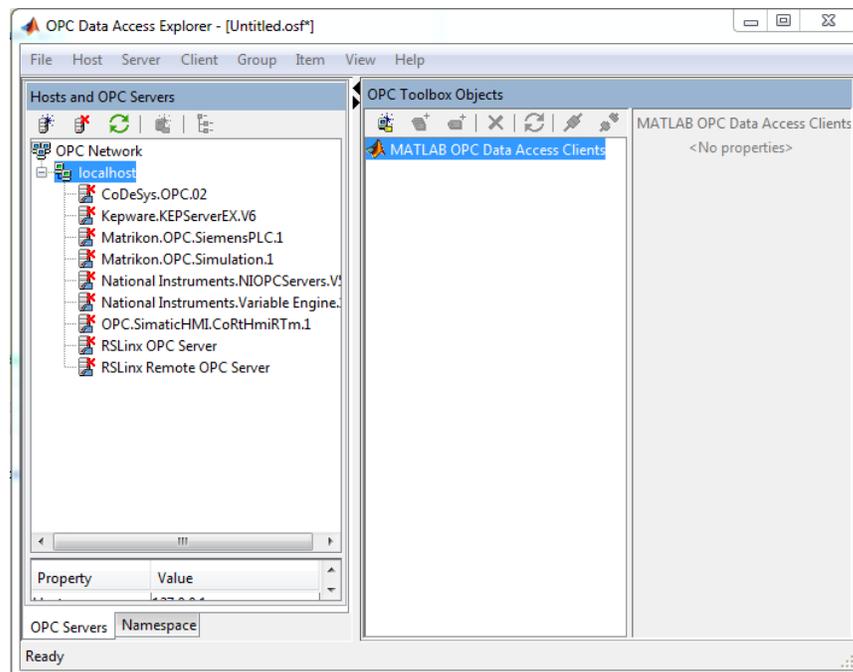


Para establecer las variables que se van a leer del PLC es necesario importar un archivo que genera el programa CODESYS el cual posee las variables compartidas para comunicación OPC, este archivo se importa ingresando a la configuración del dispositivo, en la sección de parámetros de comunicación (Communication Parameters), dando clic a la casilla de Symbol File, se agrega el archivo con extensión .SDB que genera el programa de CODESYS.

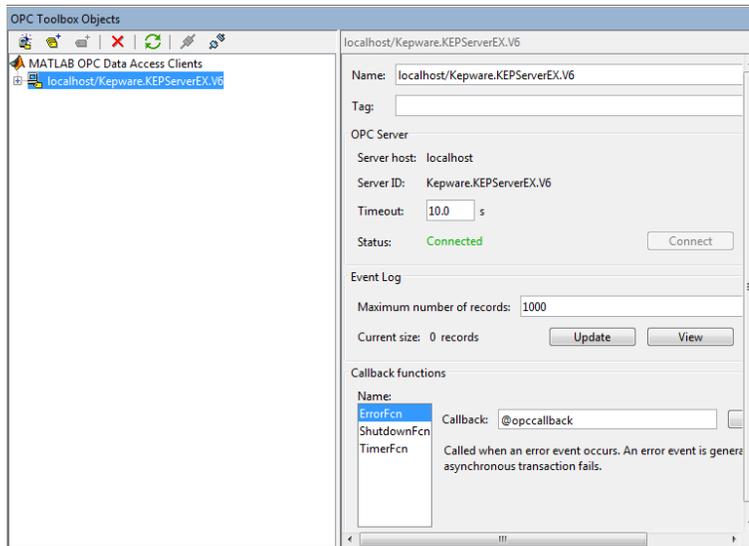
ANEXO 4 CONFIGURACION DEL CLIENTE OPC – MATLAB

La comunicación entre cliente-servidor vía Matlab se realiza a través de una Toolbox licenciada que ofrece este software “OPC TOOLBOX V 4.0.3”.

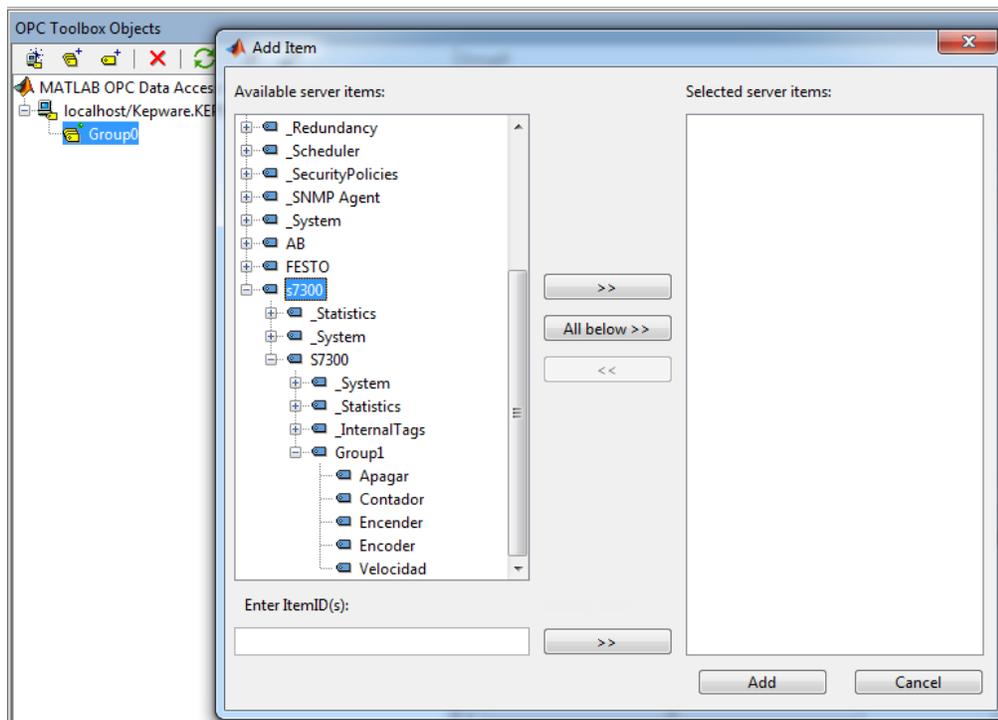
Para la configuración del cliente inicialmente se debe abrir la ventana del cliente OPC, en el command window de Matlab se escribe “opctool” e instantáneamente se despliega la ventana del servidor que es la siguiente.



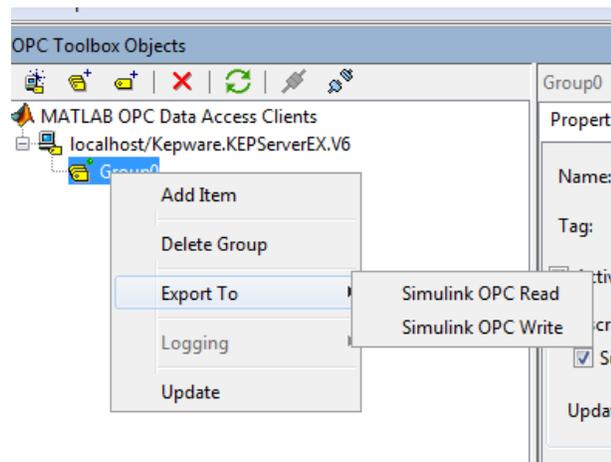
Esta ventana despliega todos los servidores OPC disponibles en el host local de la estación de trabajo (PC). Accedemos al servidor Kepware.KEPServerEX.V6.



Esta ventana muestra algunas de las características de la comunicación del servidor tales como el estado, o el tiempo de espera a respuesta máximo. Después de esto, damos clic derecho al servidor OPC y seleccionamos “Add group” en esta opción agregaremos las variables o “TAG’S” anteriormente creados en el servidor OPC, esto facilita mucho la conexión entre el cliente y servidor.



Como se ve en la imagen, se pueden seleccionar las variables anteriormente creadas en la configuración del servidor OPC, luego de agregar las variables deseadas damos aceptar y las variables se agregan al grupo de la comunicación y finaliza la configuración del cliente OPC, si se desea trabajar inmediatamente esta Toolbox ofrece la ventaja de exportar el cliente directamente a Matlab o a Simulink (lo cual es una ventaja ya que en este caso toda la configuración de los controladores se realizó en Simulink).

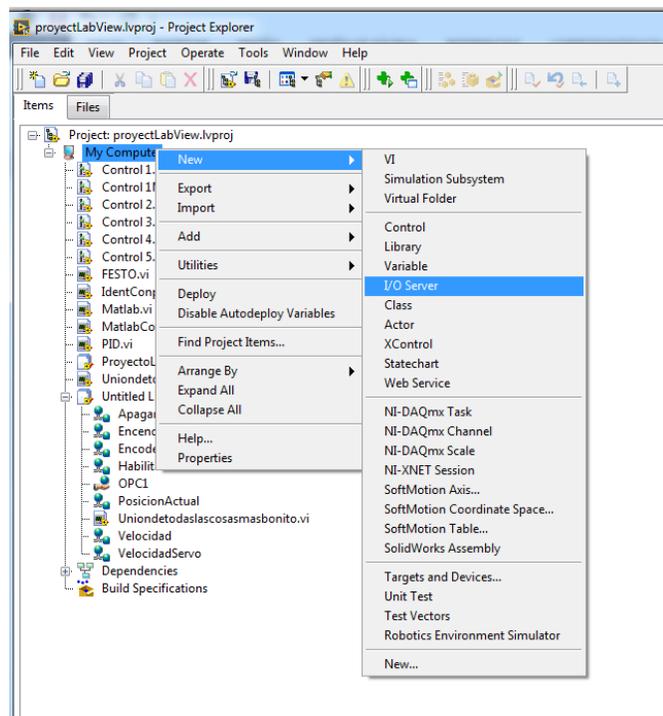


Se especifica qué tipo de bloque de comunicación creará, si lectura o escritura, se pueden exportar ambos bloques dependiendo de lo que requiera la aplicación.

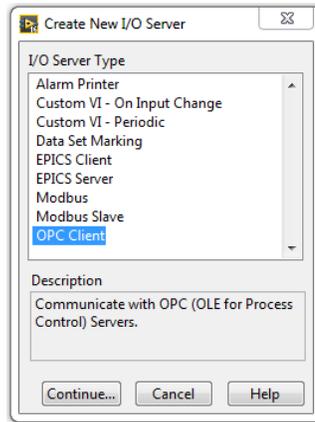
ANEXO 5 CONFIGURACION DEL CLIENTE OPC – LABVIEW

La comunicación entre cliente-servidor vía LabView se realiza a través de una licencia de comunicación OPC que posee la UNAB.

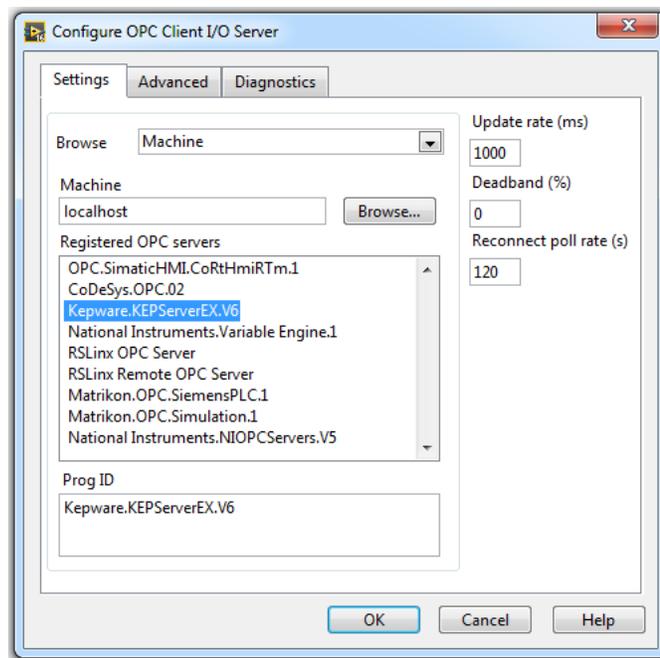
La configuración del cliente OPC de LabView se realiza en la ventana de proyectos. En esta ventana se crea un nuevo “I/O Server” o servidor de entradas y salidas.



Luego de seleccionar esta opción seleccionar esta opción, se escoge el tipo de servidor que se desea crear entre las diversas opciones que ofrece LabView.

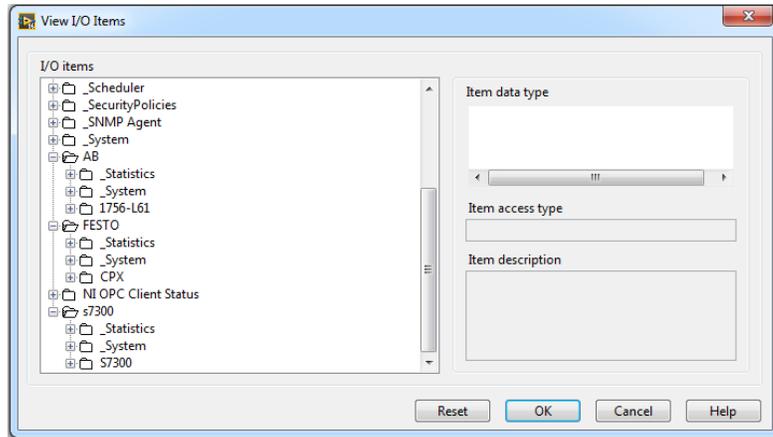


En este caso se selecciona Cliente OPC, al seleccionarlo se despliega una ventana en la cual se debe seleccionar el servidor con el cual se va a trabajar, en este caso elegimos el servidor Kepware.KEPServerEX.V6, que se encuentra en el localhost de nuestra estación de trabajo (PC).

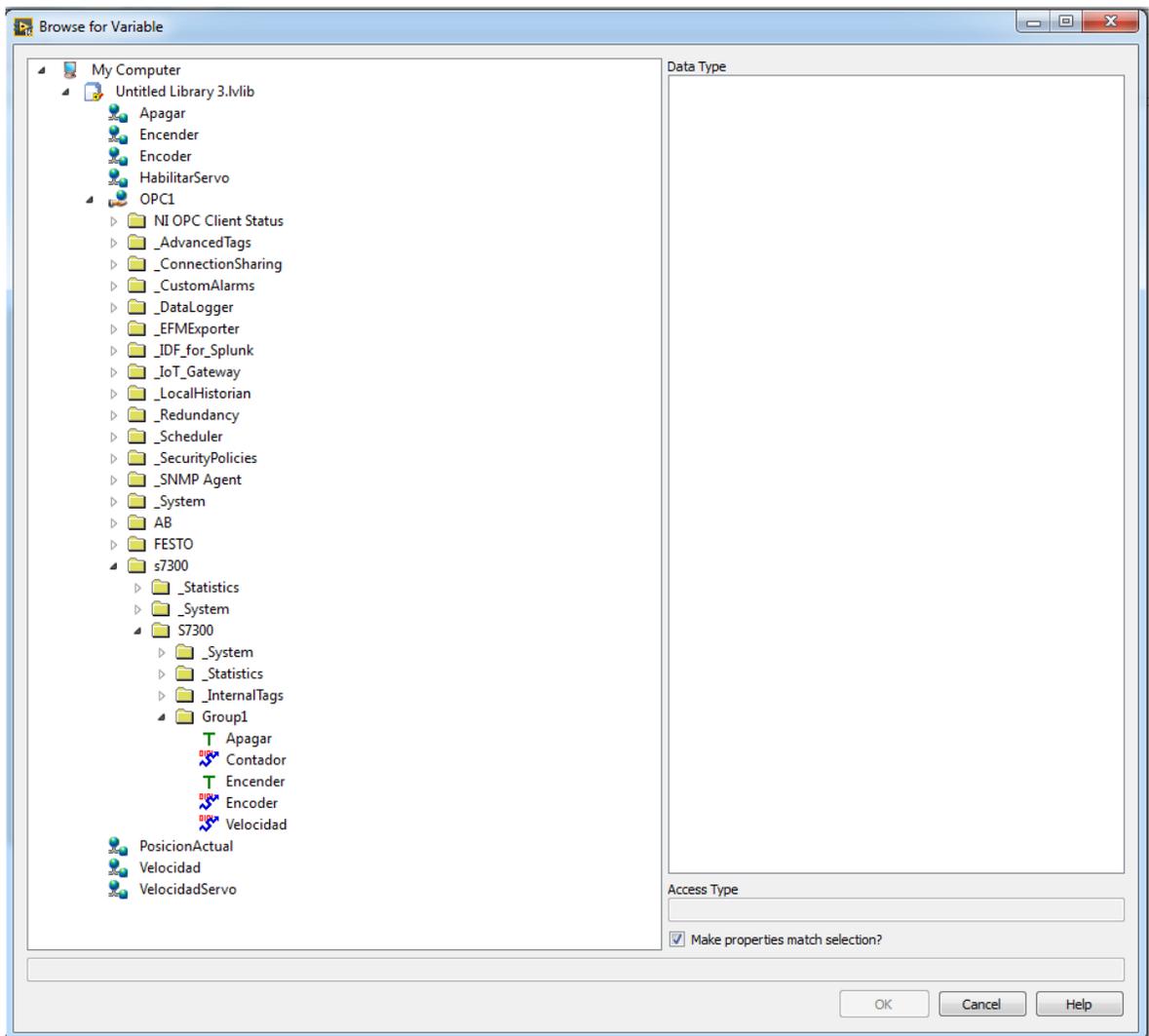


Al seleccionar este servidor se crea el enlace entre Cliente-Servidor, en el proyecto de LabView creado se agrega un archivo que contiene la comunicación entre este cliente y ese servidor específico, pueden existir comunicaciones con diferentes servidores en un mismo proyecto. Si damos clic derecho a este icono de

comunicación nos muestra que clientes están creados dentro del servidor y sus variables.



Para añadir las variables deseadas simplemente damos clic derecho en el proyecto, New, Variable. Esto abre una ventana emergente en la cual se puede acceder a los tipos de variables que se pueden crear, en este caso nos dirigimos al nodo de comunicaciones creado y se despliegan los canales creados en el servidor anteriormente, los cuales en su interior poseen las variables ya creadas con su tipo de dato.



Simply se selecciona y se agrega automáticamente al proyecto para ser usada en cualquier diagrama creado en LabView.

ANEXO 6 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 1

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB LABORATORIO COMUNICACIÓN OPC – PLC SIEMENS
<i>DOCENTE ENCARGADO</i>	

OBJETIVOS

- Conocer los alcances de la comunicación OPC aplicada a la integración de diferentes tecnologías.
- Establecer una comunicación OPC entre un PC y un PLC Siemens S7-300 por medio de Matlab y LabView.

TRABAJO PREVIO

- Realizar una investigación sobre comunicación OPC (¿Qué es?, ¿Cuál es su utilidad en la industria?, ¿Qué tipos servidores existen?)
- Características del PLC Siemens S7-300
- Funcionamiento Software Step 7 / Tia portal

PROCEDIMIENTO

1. Identificar los dispositivos con los cuales se trabajará (tener en cuenta que el PC debe tener un servidor OPC disponible, y el PLC debe tener un módulo Ethernet).
2. Crear el servidor OPC para comunicarse con un PLC siemens S7-300. (Revisar anexo 1)
3. Crear el Cliente OPC de Matlab y LabView. (Revisar anexo 4 y 5)

4. Luego de realizar la configuración de los clientes y servidores OPC iniciamos el desarrollo del laboratorio el cual consta de una lectura análoga por los canales análogos que nos ofrece el PLC siemens, en este caso se realizara una lectura análoga de una variación de voltaje entre 0 y 10 voltios obtenidos de un circuito con potenciómetro. En este caso al realizar la lectura análoga se deben visualizar los datos en los dos softwares con los cuales se puede establecer la comunicación OPC (Matlab y LabView).
5. Luego de realizar la lectura análoga y obtener el valor en alguno de los softwares de programación, se debe implantar un proceso en el cual se establecen rangos de activación para los valores de voltaje medidos como se ve a continuación.

RANGO DE VOLTAJE	ACTIVACION SALIDA DIGITAL
0v – 2.5v	Salida digital Q124.0
2.5v – 5v	Salida digital Q124.1
5v – 7.5 v	Salida digital Q124.2
7.5v – 10v	Salida digital Q124.3

Para las activaciones de las salidas digitales y para la lectura de la entrada análoga se deben crear las variables en el servidor OPC, y realizar la lectura y escritura de datos desde el software de programación vía cliente OPC.

CONCLUSIONES

Aspectos a tener en cuenta.

- Ventajas de la comunicación OPC con Siemens.
- Comparación entre el uso de ambos softwares (Matlab y LabView)

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTONOMA DE BUCARAMANGA - UNAB LABORATORIO COMUNICACIÓN OPC – PLC ALLEN BRADLEY DOCENTE ENCARGADO

ANEXO 7 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 2

OBJETIVOS

- Conocer los alcances de la comunicación OPC aplicada a la integración de diferentes tecnologías.
- Establecer una comunicación OPC entre un PC y un PLC ControlLogix 5560 por medio de Matlab y LabView.

TRABAJO PREVIO

- Realizar una investigación sobre comunicación OPC (¿Qué es?, ¿Cuál es su utilidad en la industria?, ¿Qué tipos servidores existen?)
- Características del PLC AllenBradley ControlLogix 5560
- Funcionamiento Software RSLOGIX 5000

PROCEDIMIENTO

1. Identificar los dispositivos con los cuales se trabajará (tener en cuenta que el PC debe tener un servidor OPC disponible, y el PLC).

2. Crear el servidor OPC para comunicarse con un PLC siemens S7-300. (Revisar anexo 2)

3. Crear el Cliente OPC de Matlab y LabView. (Revisar anexo 4 y 5)

4. Luego de realizar la configuración de los clientes y servidores OPC iniciamos el desarrollo del laboratorio el cual consta de una secuencia de pistones realizada con el PLC ControlLogix, en este caso se realizará la siguiente secuencia neumática.

A+ B+ B- C+ C- B+ B- A-

5. Para realizar esta secuencia se deben tener en cuenta la cantidad de salidas y entradas digitales necesarias para el buen desarrollo del ejercicio.

ENTRADA DIGITAL	DESCRIPCION
I124.0	Sensor A+
....
....
....

SALIDA DIGITAL	DESCRIPCION
....
....
....
....

Para las activaciones de las entradas y salidas digitales deben crear previamente las variables en el servidor OPC, y realizar la lectura y escritura de datos desde el software de programación vía cliente OPC.

6. La secuencia se debe poder visualizar en ambos softwares de programación (Matlab y LabView), para esto, debe existir una interfaz gráfica que represente cada pistón y realice la secuencia en tiempo real.

CONCLUSIONES

Aspectos a tener en cuenta.

- Ventajas de la comunicación OPC con AllenBradley
- Comparación entre el uso de ambos softwares (Matlab y LabView)

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB LABORATORIO COMUNICACIÓN OPC – PLC FESTO (CPX)

DOCENTE ENCARGADO

ANEXO 8 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 3

OBJETIVOS

- Conocer los alcances de la comunicación OPC aplicada a la integración de diferentes tecnologías.
- Establecer una comunicación OPC entre un PC y un PLC Festo CPX por medio de Matlab y LabView.

TRABAJO PREVIO

- Realizar una investigación sobre comunicación OPC (¿Qué es?, ¿Cuál es su utilidad en la industria?, ¿Qué tipos servidores existen?)
- Características del PLC Festo – CPX
- Funcionamiento Software CODESYS

PROCEDIMIENTO

1. Identificar los dispositivos con los cuales se trabajará (tener en cuenta que el PC debe tener un servidor OPC disponible, y el PLC).
2. Crear el servidor OPC para comunicarse con un PLC siemens S7-300. (Revisar anexo 3)
3. Crear el Cliente OPC de Matlab y LabView. (Revisar anexo 4 y 5)
4. Luego de realizar la configuración de los clientes y servidores OPC iniciamos el desarrollo del laboratorio el cual consta en hacer uso del módulo MPS de Festo que posee la UNAB. Este módulo consiste en una secuencia de almacenamiento de piezas diferenciándolas por color, en este caso, se trabajarán dos tipos de piezas, rojas y azules, las cuales el dispositivo almacenara de la siguiente forma.

COLOR DE LA PIEZA	CARRIL CORRESPONDIENTE
ROJO	Carril 1 (Más cercano)
AZUL	Carril 1 (Más lejano)

5. Para realizar este proceso se deben tener en cuenta la cantidad de salidas y entradas digitales del módulo CPX necesarias para el buen desarrollo del ejercicio.

ENTRADA DIGITAL	DESCRIPCION
DI0	Sensor 1 (Detección de pieza)
....
....
....

SALIDA DIGITAL	DESCRIPCION
DO0	Actuador 1 (Pistón 1 A+)
....
....
....

Para las activaciones de las entradas y salidas digitales deben crear previamente las variables en el servidor OPC, y realizar la lectura y escritura de datos desde el software de programación vía cliente OPC.

6. La secuencia se debe poder visualizar en ambos softwares de programación (Matlab y LabView), para esto, debe existir una interfaz gráfica que represente cada pistón y realice la secuencia en tiempo real, El proceso debe detenerse en el momento que el operario decida por medio de un botón de parada de emergencia.

CONCLUSIONES

Aspectos a tener en cuenta.

- Ventajas de la comunicación OPC con Festo
- Comparación entre el uso de ambos softwares (Matlab y LabView)

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB LABORATORIO COMUNICACIÓN OPC – RED DE COMUNICACIÓN

DOCENTE ENCARGADO

ANEXO 9 GUÍA LABORATORIO DE REDES DE COMUNICACIÓN INDUSTRIAL # 4

OBJETIVOS

- Conocer los alcances de la comunicación OPC aplicada a la integración de diferentes tecnologías.
- Establecer una red comunicación OPC entre un PC y diferentes controladores lógicos programables.

TRABAJO PREVIO

- Realizar una investigación sobre comunicación OPC (¿Qué es?, ¿Cuál es su utilidad en la industria?, ¿Qué tipos servidores existen?)
- Características del PLC Festo – CPX, PLC AllenBradley – ControlLogix 5560, PLC Siemens S7-300.
- Funcionamiento Software CODESYS, Step7/TIA PORTAL, RSLOGIX 5000.

PROCEDIMIENTO

1. Para el desarrollo de este laboratorio, se desea implementar una red en la cual existan al menos dos tipos de comunicaciones (Ethernet, Profibus, Serial, DeviceNet, Modbus, etc) y se integren los tres PLC disponibles en la UNAB. Para este laboratorio se debe plantear un proceso industrial, en el cual, se apliquen al menos tres diferentes tipos de actuadores (Motores, servomotores, pistones) los cuales deben ser controlados por un PLC diferente.
2. Identificar los dispositivos con los cuales se trabajará (tener en cuenta que el PC debe tener un servidor OPC disponible, y los PLC que se trabajaran).
3. Crear el servidor OPC para comunicarse con los dispositivos.
4. Crear el Cliente OPC de Matlab y LabView.
5. Luego de realizar la configuración de los clientes y servidores OPC iniciamos el desarrollo del laboratorio el cual consta en la creación de una red de comunicación. En este punto se describe la función de cada PLC y su proceso a realizar.

CONTROLADOR LOGICO PROGRAMABLE	FUNCION CORRESPONDIENTE	TIPOS DE COMUNICACIONES
SIEMENS		
ALLENBRADLEY		
FESTO		

6. Para realizar este proceso se deben tener en cuenta la cantidad de salidas y entradas digitales de cada PLC necesarias para el buen desarrollo del ejercicio.

PLC SIEMENS

TIPO DE ENTRADA/SALIDA	DESCRIPCION
ENTRADA DIGITAL – I124.0	Sensor 1 (Detección de pieza)
....
....
....

PLC ALLEN BRADLEY

TIPO DE ENTRADA/SALIDA	DESCRIPCION
SALIDA ANALOGA – O:0.0	Actuador 1 (Pistón 1 A+)
....
....
....

PLC FESTO

TIPO DE ENTRADA/SALIDA	DESCRIPCION
SALIDA DIGITAL – DO0	Actuador 1 (Pistón 1 A+)
....
....
....

Para las activaciones de las entradas y salidas digitales deben crear previamente

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB MODELADO DE SISTEMAS – CONTROL DE VELOCIDAD

las variables en el servidor OPC, y realizar la lectura y escritura de datos desde el software de programación vía cliente OPC.

7. El proceso se debe poder controlar desde ambos softwares de programación (Matlab y LabView), para esto, debe existir una interfaz gráfica que represente uno de los actuadores y debe existir un panel de control si es necesario. El proceso debe detenerse en el momento que el operario decida por medio de un botón de parada de emergencia.

CONCLUSIONES

Aspectos a tener en cuenta.

- Ventajas de la comunicación OPC
- Comparación entre el uso de ambos softwares (Matlab y LabView)

ANEXO 10 GUÍA LABORATORIO DE CONTROL # 1

TRABAJO PREVIO

1. Investigar:
 - Modelo caja blanca y caja negra de un sistema.
 - Funcionamiento de un motor AC – Jaula de ardilla.
 - Control de velocidad de un motor AC.

PROCEDIMIENTO

2. Hallar el modelo caja blanca del siguiente sistema el cual corresponde a un motor AC jaula de ardilla acopado a un motor DC que actúa como generador.



Fig. 1 Control de velocidad motor AC.

3. Hallar el modelo caja negra del sistema. Para esto aplique diferentes valores de frecuencia de alimentación a la entrada del motor AC y registre el valor de velocidad a la salida. Utilice la herramienta ident de Matlab para hallar la función de transferencia con los datos obtenidos en lazo abierto. Compare la respuesta del modelo caja blanca con el modelo caja negra y seleccione la función de transferencia que va a utilizar.

Control proporcional

4. Ajustar un controlador proporcional. Analizar el efecto de modificar la ganancia proporcional en la respuesta transitoria de la velocidad de salida de los motores. Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una entrada de referencia variable, para dos valores diferentes de ganancia proporcional.

Ganancia proporcional	Set point – Velocidad de motores	Acción de control

Observaciones:		

- Para la variable del proceso (PV) determinar el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Sobrepaso	Tiempo de establecimiento

- ¿Qué valor de la ganancia proporcional estabiliza el sistema? Graficar la respuesta transitoria de la velocidad de los motores y la acción de control para este valor.

Ganancia proporcional	Set point - Temperatura al interior del horno	Acción de control

Control proporcional - integral

5. Fijar un valor de ganancia proporcional y analizar el efecto de modificar el tiempo integral (T_i) en la respuesta transitoria de la velocidad ($T_d = 0$). Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una

entrada de referencia variable, para dos valores diferentes de T_i . El valor de banda proporcional y la constante de tiempo integral (T_i) seleccionados deben garantizar la estabilidad del sistema. Con el objetivo de comparar los resultados, la señal de referencia se debe mantener para los dos valores de constantes predeterminadas.

Ganancia proporcional	Tiempo integral (T_i)	Set point – Velocidad de motores	Acción de control
Observaciones:			

- Para la variable del proceso (PV) determinar el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Tiempo integral (T_i)	Sobrepaso	Tiempo de establecimiento

- ¿Qué valor de la constante de tiempo integral (T_i) inestabiliza el sistema? Graficar la respuesta transitoria de la temperatura al interior del horno y la acción de control para este valor.

Tiempo integral (T_i)	Set point – Velocidad de motores	Acción de control

Control proporcional - derivativo

6. Fijar un valor de ganancia proporcional y analizar el efecto de modificar el tiempo derivativo (T_d) en la respuesta transitoria de la velocidad de los motores ($T_i = 0$). Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una entrada de referencia variable, para dos valores diferentes de T_d . El valor de ganancia proporcional y la constante de tiempo derivativo (T_d) seleccionados deben garantizar la estabilidad del sistema. Con el objetivo de comparar los resultados, la señal de referencia se debe mantener para los dos valores de constantes predeterminadas.

Ganancia proporcional	Tiempo derivativo (T_d)	Set point – Velocidad de motores	Acción de control
Observaciones:			

- Para la variable del proceso (PV) determine el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Tiempo derivativo (T_d)	Sobrepaso	Tiempo de establecimiento

- ¿Qué valor de la constante de tiempo derivativo (T_d) inestabiliza el sistema? Graficar la respuesta transitoria de la temperatura al interior del horno y la acción de control para este valor.

Tiempo derivativo (T_d)	Set point – Velocidad de motores	Acción de control

Control proporcional – integral – derivativo

7. Fijar un valor de ganancia proporcional y analizar el efecto de modificar el tiempo integral (T_i) y el tiempo derivativo (T_d). Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una entrada de referencia variable, para dos valores diferentes de T_i y T_d . Los valores deben garantizar la estabilidad del sistema. Con el objetivo de comparar los resultados, la señal de referencia se debe mantener para los dos valores de constantes predeterminadas.

Ganancia proporcional	Tiempo integral (T_i)	Tiempo derivativo (T_d)	Set point – Velocidad de motores	Acción de control
Observaciones:				

- Para la variable del proceso (PV) determinar el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Tiempo integral (T_i)	Tiempo derivativo (T_d)	Sobrepaso	Tiempo de establecimiento

--	--	--	--	--

Conclusiones.

8. Entre las conclusiones resalte los siguientes puntos:

- Selección el mejor controlador. ¿Por qué?
- Efecto en la respuesta transitoria al modificar las constantes de los diferentes controladores.
- Características de los controladores industriales

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB MODELADO DE SISTEMAS Y CONTROL PID

ANEXO 11 GUÍA LABORATORIO DE CONTROL # 2

TRABAJO PREVIO

1. Investigar:
 - Modelo caja blanca y caja negra de un sistema.
 - Funcionamiento servomotor industrial Rockwell.
 - Control de posición de un servomotor.
 - Diseño de controladores para sistemas con integrador.

PROCEDIMIENTO

2. Hallar el modelo caja blanca del siguiente sistema el cual corresponde a una barra rígida acoplada a un servomotor industrial, este tipo de sistemas se modela como un robot rígido con un grado de libertad.

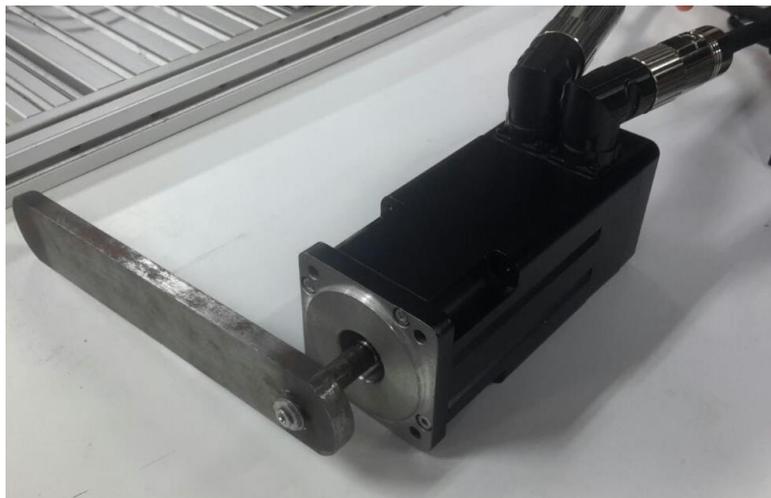


Fig. 1 Control de velocidad motor AC.

- Hallar el modelo caja negra del sistema. Para esto aplique diferentes valores de velocidad al servomotor industrial y registre el cambio de posición a la salida. Utilice la herramienta ident de Matlab para hallar la función de transferencia con los datos obtenidos en lazo abierto. Compare la respuesta del modelo caja blanca con el modelo caja negra y seleccione la función de transferencia que va a utilizar para el diseño de los controladores.

Control proporcional

- Ajustar un controlador proporcional. Analizar el efecto de modificar la ganancia proporcional en la respuesta transitoria de la posición de la barra acoplada al servomotor. Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una entrada de referencia variable, para dos valores diferentes de ganancia proporcional.

Ganancia proporcional	Set point – Posición de la barra	Acción de control
Observaciones:		

- Para la variable del proceso (PV) determinar el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Sobrepaso	Tiempo de establecimiento

- ¿Qué valor de la ganancia proporcional in estabiliza el sistema? Graficar la respuesta transitoria de la posición de la barra y la acción de control para este valor.

Ganancia proporcional	Set point – Posición de la barra	Acción de control

Control proporcional - derivativo

5. Fijar un valor de ganancia proporcional y analizar el efecto de modificar el tiempo derivativo (T_d) en la respuesta transitoria posición de la barra acoplada al servomotor ($T_i = 0$). Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una entrada de referencia variable, para dos valores diferentes de T_d . El valor de ganancia proporcional y la constante de tiempo derivativo (T_d) seleccionados deben garantizar la estabilidad del sistema. Con el objetivo de comparar los resultados, la señal de referencia se debe mantener para los dos valores de constantes predeterminadas.

6.

Ganancia proporcional	Tiempo derivativo (T_d)	Set point – Posición de la barra	Acción de control
Observaciones:			

- Para la variable del proceso (PV) determine el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Tiempo derivativo (T_d)	Sobrepaso	Tiempo de establecimiento

--	--	--	--

- ¿Qué valor de la constante de tiempo derivativo (T_d) inestabiliza el sistema? Graficar la respuesta transitoria de la temperatura al interior del horno y la acción de control para este valor.

Tiempo derivativo (T_d)	Set point – Posición de la barra	Acción de control

Control proporcional – integral – derivativo

7. Fijar un valor de ganancia proporcional y analizar el efecto de modificar el tiempo integral (T_i) y el tiempo derivativo (T_d). Registrar la respuesta transitoria de la variable del proceso y la acción de control, dada una entrada de referencia variable, para dos valores diferentes de T_i y T_d . Los valores deben garantizar la estabilidad del sistema. Con el objetivo de comparar los resultados, la señal de referencia se debe mantener para los dos valores de constantes predeterminadas.

Ganancia proporcional	Tiempo integral (T_i)	Tiempo derivativo (T_d)	Set point – Velocidad de motores	Acción de control
Observaciones:				

- Para la variable del proceso (PV) determinar el sobrepaso y el tiempo de establecimiento.

Ganancia proporcional	Tiempo integral (T_i)	Tiempo derivativo (T_d)	Sobrepaso	Tiempo de establecimiento

Conclusiones.

8. Entre las conclusiones resalte los siguientes puntos:
- Selección el mejor controlador. ¿Por qué?
 - Que efecto que sucede en la respuesta transitoria al modificar las constantes de los diferentes controladores.
 - Características de los controladores industriales
 - Que sucede al agregar un control integral a una planta que posee integrador.

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB CONTROL ÓPTIMO Y FUZZY – CONTROL DE VELOCIDAD

ANEXO 12 GUÍA LABORATORIO DE CONTROL # 3

TRABAJO PREVIO

1. Investigar:
 - Modelo caja blanca y caja negra de un sistema.
 - Funcionamiento de un motor AC – Jaula de ardilla.
 - Control de velocidad de un motor AC.

PROCEDIMIENTO

2. Hallar el modelo caja blanca del siguiente sistema el cual corresponde a un motor AC jaula de ardilla acoplado a un motor DC que actúa como generador.

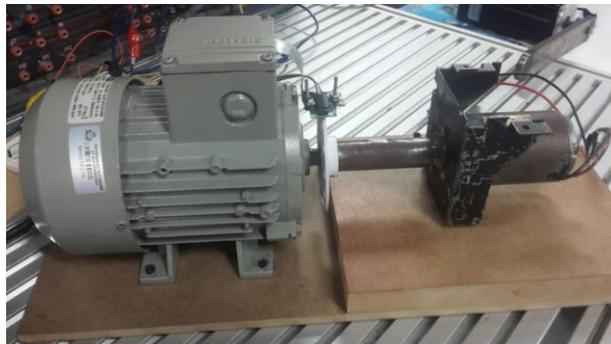


Fig. 1 Planta de motor AC acoplado a un generador.

- Hallar el modelo caja negra del sistema. Para esto aplique diferentes valores de frecuencia de alimentación a la entrada del motor AC y registre el valor de velocidad a la salida. Utilice la herramienta *ident* de Matlab para hallar la función de transferencia con los datos obtenidos en lazo abierto, seleccione la respuesta que posea un mejor *best fit* y que tenga un comportamiento similar al observado en los datos experimentales. Compare la respuesta del modelo caja blanca con el modelo caja negra y seleccione la función de transferencia que va a utilizar.

CONTROL LQG

- Diseñar un sistema digital de seguimiento utilizando la técnica de control óptimo; para estimar los estados diseñar un observador completo utilizando la técnica de ubicación de polos.

Control Óptimo		
Matriz Q		
Matriz R		
Vector de ganancias		
Polos en lazo cerrado		
Ganancia antiwindup K_a		
Diseño del observador		
Matriz observabilidad (V)		
Rango de la matriz de observabilidad V		
Criterios de diseño	Sobrepaso	
	Tiempo de establecimiento	
Polos del observador en continua s_i		
Polos del observador en discreto $z_i = e^{Ts_i}$		
Matriz de ganancia L		

- Diseñar un sistema discreto de seguimiento utilizando la técnica de control óptimo; para estimar los estados diseñar un filtro Kalman.

Control Óptimo	
Matriz Q	
Matriz R	
Vector de ganancias	
Polos en lazo cerrado	

Ganancia antiwindup K_a	
Filtro Kalman	
Matriz de covarianza Q_n	
Matriz de covarianza R_n	
Matriz de ganancia L	
Polos del observador	

6. Discretizar las dos estrategias de control LQG e implementarlas en una unidad de procesamiento digital (diagrama de bloques de la figura 2). Determinar la matriz de ganancia antiwindup adecuada para el sistema.

Control LQG	
Representación en espacio de estados	
Matriz de ganancia antiwindup	

7. Programar el código para implementar el control LQG en una unidad de procesamiento digital. (guiado por esta estructura de programación)

Código de un control LQG de un grado de libertad

Inicialización de variables de almacenamiento	$X_{ob_1} = 0$ Estados del observador $V_{k_1} = 0$ Integral del error $D_u = 0$ Memoria actuador saturado
Inicialización de constantes	K_i Ganancia del integrador K Matriz de ganancias que multiplica a los estados K_a Ganancia antiwindup G Matriz de estados H Matriz de entrada C Matriz de salida L Matriz de ganancia del observador
Programa principal	$SP =$ Valor del setpoint $PV =$ Valor de la variable del proceso $en = SP - PV$ $V_k = en + V_{k_1} - K_a * D_u$ $un = K_i * V_k - K * X_{ob_1}$ if $un > -100$ & $un < 100$ $U = un$ if $un > 100$ $U = 100$ if $un < -100$ $U = -100$ Actualizar U

	<pre> % Calculo del observador Xob = (G - L*C)*Xob_1 + H*U + L*PV Vk_1 = Vk Du = un - U Xob_1 = Xob </pre>
--	---

- Bloques de Matlab de un controlador LQG con observador.

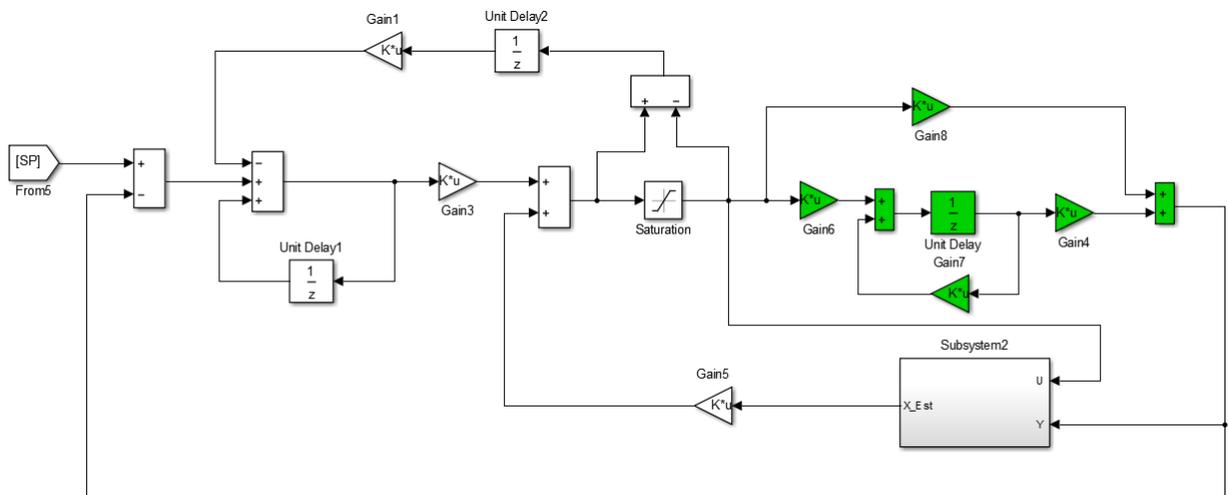


Fig. 2 Controlador LQG de un grado de libertad

8. Evaluar la respuesta transitoria de la variable del proceso y la acción de control del sistema en lazo cerrado ante una entrada pulso de magnitud igual a valor de velocidad que se encuentre en el rango de acción del motor. Para la variable del proceso (PV) determinar el sobrepaso, el tiempo de establecimiento y el valor en estado estable.

	Controlador LQG
Set point – Variable del proceso	
Acción de control	

CONTROL FUZZY

9. Establezca el tamaño de los conjuntos difusos para dos entradas, la entrada "Error" la cual será la diferencia entre el set point y la variable del proceso y la entrada "D_error" la cual corresponde a la diferencia entre el "Error" y el valor anterior del error, ver figura 3. Inserte la gráfica de los conjuntos difusos que seleccionó.

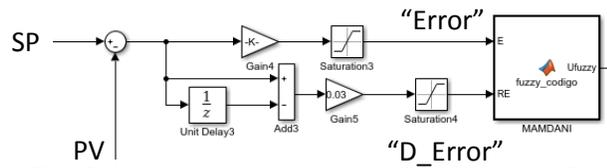


Fig. 3. Variables de entrada al controlador Fuzzy

Control Fuzzy	
Conjunto difuso "Error"	
Conjunto difuso "D_Error"	

10. Definir el tipo de control Fuzzy (Mamdani o Sugeno) y establecer el conjunto difuso de salida "U". Inserte la gráfica del conjunto difuso seleccionado.

Tipo de controlador Fuzzy	
Conjunto difuso de salida "U"	

11. Programe la base de reglas para los conjuntos difusos, defina la posición de los valores difusos para ser clasificado en la función de salida.

- Reglas establecidas.

- Superficie del controlador.

12. Implemente el controlador difuso para analizar la respuesta ante diferentes valores de ganancia a las entradas del controlador, las cuales van a afectar el peso de cada conjunto difuso. Aplique diferentes valores de ganancia y analice el efecto que tiene en el controlador. Registre al menos 3 valores de ganancias. (En la figura 4 se observa el diagrama de bloques para implementar el control Fuzzy.

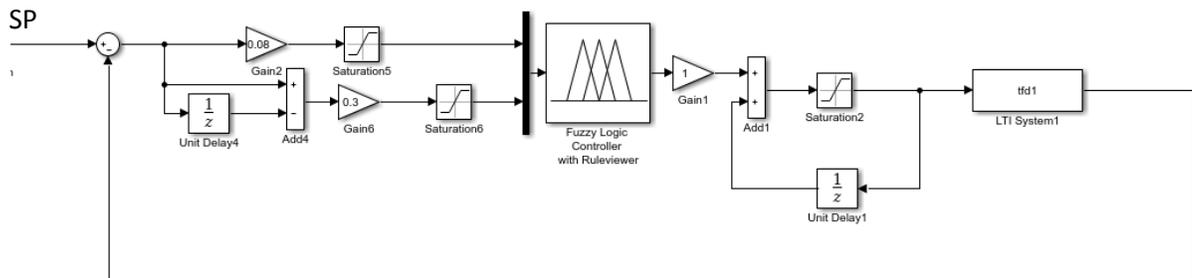


Fig. 4. Diagrama de bloques para control Fuzzy.

Ganancia "K_Error"	Ganancia "K_D_Error"	Ganancia a la salida del controlador "K_Salida"	Set point – Velocidad del motor	Acción de control

Conclusiones.

Entre las conclusiones resalte los siguientes puntos:

	PROGRAMA DE INGENIERÍA MECATRÓNICA
	UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB CONTROL ÓPTIMO Y FUZZY – CONTROL DE POSICIÓN

- Selección el mejor controlador. ¿Por qué?
- Utilidad del control LQG y Fuzzy en la industrial
- Efecto en la respuesta transitoria al modificar las ganancias en el controlador Fuzzy de los diferentes controladores.

ANEXO 13 GUÍA LABORATORIO DE CONTROL # 4

TRABAJO PREVIO

1. Investigar:
 - Modelo caja blanca y caja negra de un sistema.
 - Funcionamiento servomotor industrial Rockwell.
 - Control de posición de un servomotor.
 - Diseño de controladores para sistemas con integrador.

2. Hallar el modelo caja blanca del siguiente sistema el cual corresponde a una barra rígida acoplada a un servomotor industrial, este tipo de sistemas se modela como un robot rígido con un grado de libertad.

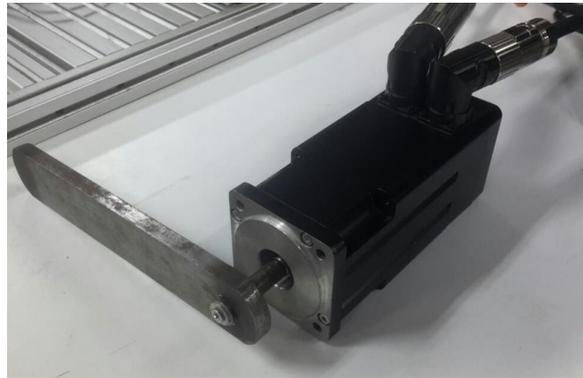


Fig. 1 Control de velocidad motor AC.

3. Hallar el modelo caja negra del sistema. Para esto aplique diferentes valores de velocidad al servomotor industrial y registre el cambio de posición a la salida. Utilice la herramienta ident de Matlab para hallar la función de transferencia con los datos obtenidos en lazo abierto. Compare la respuesta del modelo caja blanca con el modelo caja negra y seleccione la función de transferencia que va a utilizar para el diseño de los controladores.

CONTROL LQG

4. Diseñar un sistema digital de seguimiento utilizando la técnica de control óptimo; para estimar los estados diseñar un observador completo utilizando la técnica de ubicación de polos.

Control Óptimo		
Matriz Q		
Matriz R		
Vector de ganancias		
Polos en lazo cerrado		
Ganancia antiwindup K_a		
Diseño del observador		
Matriz observabilidad (V)		
Rango de la matriz de observabilidad V		
Criterios de diseño	Sobrepaso	
	Tiempo de establecimiento	
Polos del observador en continua s_i		
Polos del observador en discreto $z_i = e^{Ts_i}$		

Matriz de ganancia L	
------------------------	--

5. Diseñar un sistema discreto de seguimiento utilizando la técnica de control óptimo; para estimar los estados diseñar un filtro Kalman.

Control Óptimo	
Matriz Q	
Matriz R	
Vector de ganancias	
Polos en lazo cerrado	
Ganancia antiwindup K_a	
Filtro Kalman	
Matriz de covarianza Q_n	
Matriz de covarianza R_n	
Matriz de ganancia L	
Polos del observador	

6. Discretizar las dos estrategias de control LQG e implementarlas en una unidad de procesamiento digital (diagrama de bloques de la figura 2). Determinar la matriz de ganancia antiwindup adecuada para el sistema.

Control LQG	
Representación en espacio de estados	
Matriz de ganancia antiwindup	

7. Programar el código para implementar el control LQG en una unidad de procesamiento digital. (guiado por esta estructura de programación)

Código de un control LQG de un grado de libertad

Inicialización de variables de almacenamiento	$X_{ob_1} = 0$ Estados del observador $V_{k_1} = 0$ Integral del error $D_u = 0$ Memoria actuador saturado
Inicialización de constantes	K_i Ganancia del integrador K Matriz de ganancias que multiplica a los estados K_a Ganancia antiwindup G Matriz de estados H Matriz de entrada C Matriz de salida L Matriz de ganancia del observador

<p>Programa principal</p>	<pre> SP = Valor del setpoint PV = Valor de la variable del proceso en = SP - PV Vk = en + Vk_1 - Ka*Du un = Ki*Vk - K*Xob_1 if un > -100 & un<100 U = un If un > 100 U = 100 If un < -100 U = -100 Actualizar U % Calculo del observador Xob = (G - L*C)*Xob_1 + H*U + L*PV Vk_1 = Vk Du = un - U Xob_1 = Xob </pre>
---------------------------	--

- Bloques de Matlab de un controlador LQG con observador.

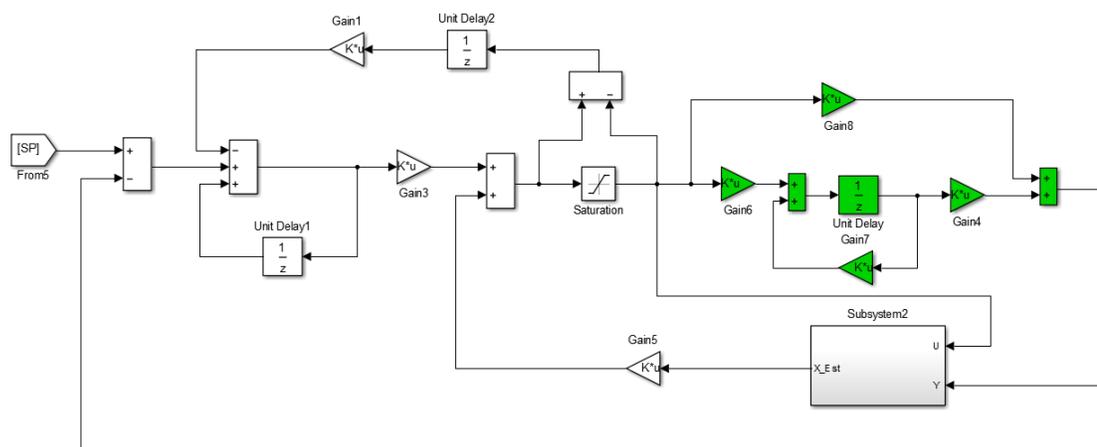


Fig. 2 Controlador LQG de un grado de libertad

8. Evaluar la respuesta transitoria de la variable del proceso y la acción de control del sistema en lazo cerrado ante una entrada pulso de magnitud igual a valor de velocidad que se encuentre en el rango de acción del motor. Para la variable del proceso (PV) determinar el sobrepaso, el tiempo de establecimiento y el valor en estado estable.

Controlador LQG	
Setpoint – Variable del proceso	
Acción de control	

CONTROL FUZZY

9. Establezca el tamaño de los conjuntos difusos para dos entradas, la entrada “Error” la cual será la diferencia entre el set point y la variable del proceso y la entrada “D_error” la cual corresponde a la diferencia entre el “Error” y el valor anterior del error, ver figura 3. Inserte la gráfica de los conjuntos difusos que seleccionó.

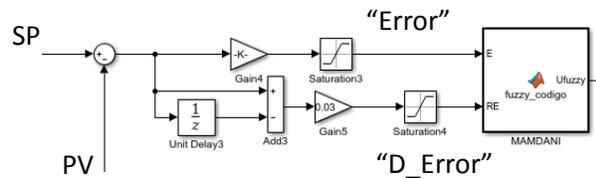


Fig. 3. Variables de entrada al controlador Fuzzy

Control Fuzzy	
Conjunto difuso “Error”	
Conjunto difuso “D_Error”	

10. Definir el tipo de control Fuzzy (Mamdani o Sugeno) y establecer el conjunto difuso de salida “U”. Inserte la gráfica del conjunto difuso seleccionado.

Tipo de controlador Fuzzy	
Conjunto difuso de salida “U”	

11. Programe la base de reglas para los conjuntos difusos, defina la posición de los valores difusos para ser clasificado en la función de salida.

- Reglas establecidas.

- Superficie del controlador.

12. Implemente el controlador difuso para analizar la respuesta ante diferentes valores de ganancia a las entradas del controlador, las cuales van a afectar el peso de cada conjunto difuso. Aplique diferentes valores de ganancia y analice el efecto que tiene en el controlador. Registre al menos 3 valores de ganancias. (En la figura 4 se observa el diagrama de bloques para implementar el control Fuzzy.

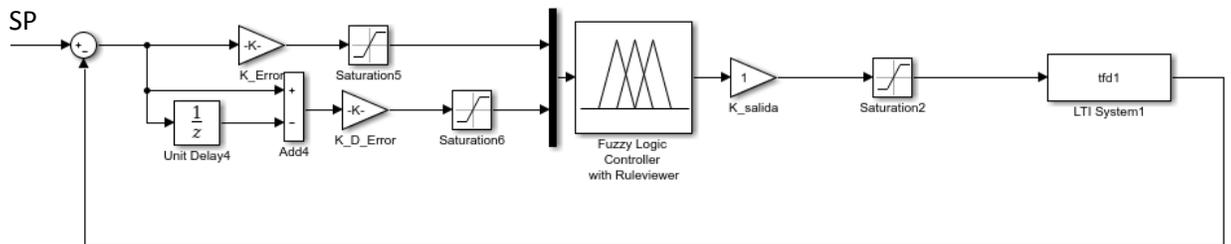


Fig. 4. Diagrama de bloques para control Fuzzy.

		Ganancia a la salida del	Set point – Velocidad del motor	Acción de control
--	--	---------------------------------	--	--------------------------

Ganancia "K_Error"	Ganancia "K_D_Error"	controlador "K_Salida"		

Conclusiones.

Entre las conclusiones resalte los siguientes puntos:

- Selección el mejor controlador. ¿Por qué?
- Comportamiento de estos controladores a una planta con integrador
- Utilidad del control LQG y Fuzzy en la industrial
- Efecto en la respuesta transitoria al modificar las ganancias en el controlador Fuzzy de los diferentes controladores.