

# **Sistema de control de temperatura basado en arquitectura ANFIS.**



**Daniela Acosta Bermúdez**  
**Pablo Cesar Gutiérrez Trujillo**

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Electrónica, Instrumentación y Control**  
**Ingeniería en Automática Industrial**

Popayán, Abril de 2013

# **Sistema de control de temperatura basado en arquitectura ANFIS.**



Documento Final de Trabajo de Grado para optar al título de  
Ingeniero en Automática Industrial

**Daniela Acosta Bermúdez**  
**Pablo Cesar Gutiérrez Trujillo**

Director: Ing. Elena Muñoz España

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Electrónica, Instrumentación y Control**  
**Ingeniería en Automática Industrial**

Popayán, Abril de 2013

## CONTENIDO

INTRODUCCIÓN .....	1
1. CONCEPTOS BÁSICOS .....	3
1.1. Control Clásico.....	3
1.1.1. Estrategias de Control. ....	3
1.1.2. Técnicas de Control. ....	4
1.2. Control Inteligente.....	5
1.2.1. Sistemas Difusos. ....	6
1.2.2. Control PD+I Difuso. ....	8
1.2.3. Redes Neuronales Artificiales.....	9
1.2.4. Sistemas Neuro-Difusos. ....	10
1.2.5. ANFIS (Adaptive-Networks based Fuzzy Inference Systems). ....	11
2. DISEÑO DE LOS CONTROLADORES .....	15
2.1. Descripción de la Planta de Temperatura del LCP. ....	15
2.2. Diseño de los Controladores PID.....	17
2.2.1. Identificación de la planta .....	17
2.2.2. Sintonización de controlador PID serie industrial.....	22
2.3. Controlador PD+I Difuso.....	25
2.3.1. Sintonización del controlador PID paralelo .....	26
2.3.2. Diseño del controlador PD+I Difuso.....	29
2.4. Diseño del Controlador ANFIS.....	35
2.4.1. Modelo de la acción integral del PID por ANFIS.....	37
2.4.2. Modelo de la acción derivativa del PID por ANFIS .....	44
2.4.3. Esquema del Controlador basado en ANFIS para la planta de temperatura. ....	47
2.5. Resultados del Diseño de los controladores.....	53

3.	IMPLEMENTACIÓN DE LOS CONTROLADORES EN PLANTA. ....	56
3.1.	Aspectos básicos para la implementación. ....	56
3.1.1.	Tarjeta Arduino Mega 2560 R3. ....	56
3.1.2.	Simulink Support Package for Arduino Hardware .....	57
3.2.	Implementación de los Controladores en la Planta de Control de Temperatura del LCP. ....	59
3.2.1.	Descripción del cableado de la planta para la adecuación del nuevo escenario de control. ....	59
3.2.2.	Diagrama de bloques para la programación de la tarjeta Arduino. ....	61
3.2.3.	Implementación del Controlador PID Serie Industrial. ....	65
3.2.4.	Implementación del Controlador PD+I Difuso. ....	67
3.2.5.	Implementación de los Modelos de Acción Integral y Derivativa obtenidos por ANFIS. ....	69
3.2.6.	Resultados de la implementación. ....	72
4.	GUÍA DE LABORATORIO .....	76
4.1.	Objetivos .....	76
4.2.	Requerimientos .....	76
4.3.	Procedimientos Generales. ....	77
4.3.1.	Instalación Simulink Support Package for Arduino Hardware .....	77
4.3.2.	Encendido de la Planta de Temperatura de LCP. ....	78
4.3.3.	Apagado de la Planta. ....	81
4.3.4.	Programar la Tarjeta Arduino Mega 2560 R3. ....	82
4.4.	Identificación de la Planta por el método de curva de reacción. ...	84
4.5.	Sintonización e Implementación de un Controlador PD+I Difuso. .	87
4.6.	Diseño e Implementación de un controlador basado en ANFIS, a partir de un PID serie industrial. ....	92
4.6.1.	Sintonización de un PID serie industrial .....	92
4.6.2.	Diseño del controlador basado en arquitectura ANFIS. ....	95
	CONCLUSIONES.....	105
	BIBLIOGRAFÍA .....	107

## INDICE DE FIGURAS

Figura 1.1 Estrategia de control en lazo cerrado	4
Figura 1.2 Diagrama de Bloques de un –Sistema de Inferencia Difuso	6
Figura 1.3 Esquema de un controlador PD+I Difuso, (Jantzen, 2002)	8
Figura 1.4 Red Neuronal Artificial.	10
Figura 1.5 Arquitectura ANFIS para dos entradas. (Jang, 1993)	13
Figura 2.1 Planta de control de temperatura LCP (Fuente Propia).	15
Figura 2.2 Panel de control planta de temperatura (Laboratorio, 2010).	16
Figura 2.3 Señal de voltaje: (a) con periodos de 2ms de PWM. (b) con área aproximada que pasa del PWM	18
Figura 2.4 Salida de voltaje con kT aproximado de paso del PWM	18
Figura 2.5 Respuesta de la planta de temperatura a una variación del 30 al 70% del voltaje	20
Figura 2.6 Implementación del FOPDT en Matlab Simulink	21
Figura 2.7 Comparación entre la respuesta del FOPDT y la curva de reacción experimental de la planta.	22
Figura 2.8 Diseño del controlador PID serie industrial	23
Figura 2.9 Esfuerzo de control del controlador PID serie industrial	24
Figura 2.10 Respuesta del sistema bajo el controlador PID a diferentes <i>Set Point</i> .	24
Figura 2.11 Diseño del controlador PID serie industrial con presencia de disturbio.	25
Figura 2.12 Respuesta del sistema bajo el controlador PID ante un aumento de 3°C al segundo 1500.	25
Figura 2.13 Diseño del controlador PID	27
Figura 2.14 Respuesta del controlador PID de la planta de temperatura a diferentes <i>Set Point</i> .	27
Figura 2.15 Esquema PID con disturbio	28
Figura 2.16 Respuesta del controlador ante un disturbio aplicado en el segundo 1500 .	28
Figura 2.17 Conjunto difuso de entrada “Error”. Toolbox fuzzy	30
Figura 2.18 Conjunto difuso de entrada “CambioError”. Toolbox fuzzy	30
Figura 2.19 Conjunto difuso de salida. “Salida” Toolbox fuzzy.	31
Figura 2.20 Diseño controlador PD+I Difuso.	32
Figura 2.21 Esfuerzo de control PD+I Difuso.	33
Figura 2.22 Respuesta de la planta bajo el controlador PD+I difuso a diferentes <i>set point</i> .	33
Figura 2.23 Diseño del controlador PD+I con presencia de disturbio	34
Figura 2.24 Respuesta de la planta bajo el controlador PD+I ante un aumento de 3°C al segundo 1500.	34

Figura 2.25 Señal de esfuerzo de control del PID paralelo y del PID serie industrial.	36
Figura 2.26 Datos de la señal de error del sistema.	37
Figura 2.27 Datos salida del integrador	38
Figura 2.28 Toma de datos acción integral.	38
Figura 2.29 Esquema modelo ANFIS acción Integral	39
Figura 2.30 Parámetros del sistema difuso para el modelo ANFIS.	40
Figura 2.31 Parámetros del entrenamiento del FIS para el modelo ANFIS.	40
Figura 2.32 Estructura del modelo ANFIS para la acción integral.	41
Figura 2.33 Resultado del modelo ANFIS, datos de entrenamiento.	42
Figura 2.34 Resultado del modelo ANFIS, datos de testeo.	42
Figura 2.35 Resultado del modelo ANFIS, datos de chequeo	43
Figura 2.36 Toma de datos acción derivativa	44
Figura 2.37 Datos temperatura salida	44
Figura 2.38 Datos salida derivador.	45
Figura 2.39 Esquema modelo ANFIS acción derivativa.	45
Figura 2.40 Estructura del Modelo ANFIS para la acción integral.	46
Figura 2.41 Resultado del modelo ANFIS, datos de entrenamiento	46
Figura 2.42 Resultado del modelo ANFIS, datos de testeo	47
Figura 2.43 Resultado del modelo ANFIS, datos de chequeo	47
Figura 2.44 Diagrama de bloques del sistema modelando por ANFIS la acción derivativa.	48
Figura 2.45 Respuesta del sistema modelando la acción derivativa por ANFIS.	48
Figura 2.46 Esfuerzo de controlador modelando la acción derivativa por ANFIS	49
Figura 2.47 Esquema del controlador con el modelado de la acción derivativa por ANFIS	49
Figura 2.48 Respuesta del sistema al controlador basado en ANFIS para la planta de temperatura.	50
Figura 2.49 Diagrama de bloques de los modelos de la acción integral y derivativa por ANFIS	50
Figura 2.50 Respuesta del sistema bajo el controlador basado en ANFIS para la planta de temperatura.	51
Figura 2.51 Esfuerzo de control del controlador modelado por ANFIS.	51
Figura 2.52 Esquema controlador basado en ANFIS para la planta de temperatura.	52
Figura 2.53 Respuesta del sistema al controlador basado en ANFIS para la planta de temperatura.	52
Figura 2.54 Respuesta controlador basado en ANFIS para la planta de temperatura.	54
Figura 2.55 Comparación entre los esfuerzos del control aplicado a la planta.	55

Figura 3.1 Tarjeta Arduino Mega 2560	56
Figura 3.2 Panel de Control de la planta de control de temperatura LCP actualizado (Fuente Propia).	60
Figura 3.3 Diagrama de lazo para la Tarjeta Arduino (Fuente Propia).	61
Figura 3.4 Diagrama de bloques genérico para la programación de la tarjeta Arduino, para la implementación en la planta de control de temperatura del LCP.	62
Figura 3.5 Configuración para la lectura del puerto analógico para la Tarjeta Arduino Mega, a través de Matlab Simulink.	63
Figura 3.6 Configuración del puerto de salida PWM para la Tarjeta Arduino Mega, a través de Matlab Simulink.	64
Figura 3.7 Configuración para la transmisión serial por la Tarjeta Arduino Mega, a través de Matlab Simulink.	64
Figura 3.8 Configuración para la recepción serial de la Tarjeta Arduino Mega, a través de Matlab Simulink.	65
Figura 3.9 Diagrama en Matlab-Simulink para la implementación de un controlador PID Serie Industrial.	65
Figura 3.10 Respuesta del sistema bajo la implementación del controlador PID serie industrial	66
Figura 3.11 Respuesta del controlador ante presencia de disturbios externos.	66
Figura 3.12 Diagrama en Matlab-Simulink para la implementación de un controlador PD+I Difuso	67
Figura 3.13 Comportamiento del sistema bajo la Implementación de un controlador PD+I Difuso.	68
Figura 3.14 Comportamiento del controlador PD+I Difuso, ante la presencia de perturbaciones externas.	69
Figura 3.15 Esquema en Simulink para la implementación del modelo obtenido por ANFIS de la acción derivativa del controlador PID serie industrial.	69
Figura 3.16 Respuesta del sistema bajo la implementación del modelo de acción derivativa por ANFIS.	70
Figura 3.17 Comportamiento del modelo obtenido por ANFIS más la acción integral, ante la presencia de una perturbación.	70
Figura 3.18 Esquema en Simulink para la implementación de los modelos obtenidos por ANFIS de la acción integral y derivativa del controlador PID serie industrial.	71
Figura 3.19 Respuesta del sistema bajo la implementación de los modelos de acción integral y derivativa obtenidos por ANFIS.	71
Figura 3.20 Comportamiento de los modelos obtenidos por ANFIS, ante la presencia de una perturbación.	72
Figura 4.1 Tragetinstaller	77

Figura 4.2 Opción de instalación <i>Internet</i>	77
Figura 4.3 Elegir como <i>hardware</i> la tarjeta arduino Mega 2560	78
Figura 4.4 Instalar el paquete.	78
Figura 4.5 Posición de las perillas para el encendido de la planta.	79
Figura 4.6 Breaker HS7	79
Figura 4.7 Fuente de 24 Voltios	79
Figura 4.8 <i>Switch</i> HS5	80
Figura 4.9 <i>Switch</i> HS6	80
Figura 4.10 Pulsador para encender el ventilador.	80
Figura 4.11 Interruptor para el encendido y apagado de las resistencias.	81
Figura 4.12 Pulsador HS4, Apaga el ventilador de la planta.	81
Figura 4.13 Detener la interfaz de <i>LabVIEW</i> .	81
Figura 4.14 Ubicación de la opción Tools	82
Figura 4.15 Opción Run on Target Hardware	82
Figura 4.16 Opción Prepare toRun	83
Figura 4.17 Ventana de configuración, para la comunicación HW-SW	83
Figura 4.18 Opción RUN	83
Figura 4.19 Operación Exitosa.	84
Figura 4.20 Establecer una carpeta de trabajo en Matlab.	84
Figura 4.21 Archivos de la carpeta de trabajo.	85
Figura 4.22 Modelo en Simulink para la identificación de la planta	85
Figura 4.23 Interfaz para Identificación de la planta	86
Figura 4.24 Extraer Imagen desde la interfaz	86
Figura 4.25 Ventana que permite elegir la ubicación donde se guarda la gráfica.	87
Figura 4.26 Establecer carpeta PDIDifuso como carpeta de trabajo en Matlab	88
Figura 4.27 Modelo en Simulink para el controlador PD+I Difuso	89
Figura 4.28 Archivo a descargar para la implementación del Controlador PD+I Difuso en la Planta	89
Figura 4.29 Interfaz para el Control de la Planta.	90
Figura 4.30 Extraer Imagen desde la interfaz	90
Figura 4.31 Ventana que permite elegir la ubicación donde se guarda la gráfica.	91
Figura 4.32 Disturbio instalado en la planta de temperatura del LCP.	91
Figura 4.33 <i>Switch</i> S7	91
Figura 4.34 Establecer Controlador basado en ANFIS como Carpeta de trabajo en Matlab	92
Figura 4.35 Archivo PIDIndustrialSerie.	93
Figura 4.36 Interfaz para Control de la Planta de Temperatura LCP	94
Figura 4.37 Extraer Imagen desde la interfaz	94
Figura 4.38 Ventana que permite elegir la ubicación donde se guarda la gráfica.	95
Figura 4.39 Archivo "TomadDatos.mdl"	96

Figura 4.40 Configuración Scope	96
Figura 4.41 Configuración Scope1	96
Figura 4.42 ArchivosDatos_derivada y Datos _integral en el workspace de Matlab	97
Figura 4.43 Comandos para adquirir datos del scope de Simulink	97
Figura 4.44 Comando para guardar las matrices	97
Figura 4.45 GUI ANFISedit	98
Figura 4.46 Carga de datos de entrenamiento	99
Figura 4.47 Carga de datos de testeo	99
Figura 4.48 Carga de datos de chequeo	99
Figura 4.49 Parámetros del Sistema Difuso para el modelo ANFIS de la acción integral.	100
Figura 4.50 Parámetros del entrenamiento del FIS para el modelo ANFIS de la acción integral	100
Figura 4.51 GUI de ANFIS durante el entrenamiento del sistema difuso	101
Figura 4.52 Comportamiento del modelo realizado por ANFIS frente a los 3 conjuntos de datos.	101
Figura 4.53 Ventana Generar FIS modelo ANFIS Acción derivativa	102
Figura 4.54 Archivo ANFIS_simulacion.mdl	103
Figura 4.55 Archivo para descargar en arduino	103

## INDICE DE TABLAS

Tabla 2.1 Parámetros IAE .....	23
Tabla 2.2 Comparación entre los diferentes controladores diseñados en Matlab Simulink .....	54
Tabla 3.1 Características Técnicas de la Tarjeta Arduino Mega 2560 .....	57
Tabla 3.2 Bloques que conforman el paquete de soporte para Arduino de Simulink .....	57
Tabla 3.3 Listado de Tags para el nuevo escenario de Control .....	60
Tabla 3.4 Comparación entre los diferentes controladores implementados en planta. .....	75
Tabla 4.1 Registro de los valores de $dy$ .....	87
Tabla 4.2 Registro de Tiempos. ....	87
Tabla 4.3 Parámetros IAE .....	93

## INTRODUCCIÓN

Con la evolución de los sistemas de producción, se ve con frecuencia la implementación de metodologías inteligentes para el control de las variables de un proceso, dado que éstas cumplen con características propias, que permiten al usuario obtener mejores resultados que con técnicas clásicas son más difíciles o imposibles de alcanzar.

El control inteligente es una representación del mundo, limitada al proceso y basada en un conocimiento sobre él y las operaciones que se desean realizar, que permite implementar sistemas de control eficaces en una gran variedad de situaciones [Sanz 1990], haciendo uso de diferentes técnicas tomadas de la inteligencia artificial, como lo son las redes neuronales, los algoritmos genéticos y la lógica difusa entre otros.

El programa de Ingeniería en Automática Industrial de la Universidad del Cauca, cuenta con un Laboratorio de Control de Procesos (LCP) en el cual existen diferentes sistemas físicos que emulan el comportamiento de variables como temperatura, presión o nivel en un tanque o en una tubería entre otras, que le ayudan al estudiante a familiarizarse con el ambiente industrial además de aplicar la teoría aprendida en las diferentes materias de la carrera; estos sistemas cuentan con diferentes escenarios de automatización en los cuales se han realizado controladores clásicos, específicamente control tipo PID implementados en PLC (controladores lógicos programables) en su gran mayoría.

El propósito de este trabajo es llevar a cabo el diseño e implementación de un controlador basado en sistemas inteligentes, para la planta de control de temperatura del laboratorio, con el fin de mostrar de manera práctica las ventajas y desventajas que este tipo de control presenta tanto en simulación como en implementación al momento de controlar sistemas. De igual forma darle a conocer a los estudiantes otra opción diferente al controlador convencional.

Para el desarrollo del proyecto se diseñaron tres controladores de temperatura, un PID serie industrial, un PD+I difuso y un controlador basado en la técnica neuro-difusa ANFIS, los cuales son implementados en la Planta de Control de Temperatura del LCP, con el fin de realizar una comparación entre las diferentes técnicas, utilizando una tarjeta Mega 2560 R3 de Arduino, que es programada desde el software Matlab – Simulink, permitiendo realizar el código para la

implementación de manera intuitiva y sencilla, lo que facilita el trabajo para personas no expertas en programación clásica.

Para el diseño del controlador basado en la técnicas neuro-difusas se escogió una red adaptativa de un sistema de inferencia difusa, ANFIS de sus siglas en inglés (Adaptitive-Networks based Fuzzy Inference Systems) (Jang, 1993), la cual consiste en mapear sobre una red un sistema de inferencia difusa tipo Takagi-Sugeno de primer orden (Takagi & Sugeno, 1985).

ANFIS es una de las técnicas neuro-difusas más conocidas y empleadas en el área del control y modelado de sistemas. Este trabajo cuenta no solo con la implementación de los controladores en planta, a través de un nuevo escenario de control, sino que también deja como aporte a la carrera una guía donde se describe el procedimiento para la obtención de los resultados, por medio de la cual los estudiantes podrán realizar un trabajo práctico sobre un sistema real, como apoyo a sus clases teóricas de control inteligente.

El trabajo está organizado en cuatro capítulos. En el primer capítulo se realiza una revisión del estado actual de las técnicas de control dentro de las que se enmarca el trabajo realizado.

En el segundo capítulo se diseñan y prueban en simulación tres controladores, un controlador PID serie industrial, un controlador PD+I difuso y los modelos de las acciones derivativa e integral del controlador PID serie industrial obtenido por ANFIS.

En el tercer capítulo se implementan los controladores diseñados anteriormente sobre la planta de temperatura del LCP, utilizando una tarjeta Arduino Mega 2560 como hardware, el software Matlab Simulink como interfaz de programación de la tarjeta y *LabVIEW* para el diseño de la interfaz hombre máquina.

Por último en el cuarto capítulo se encuentran las guías para el laboratorio, con las cuales los estudiantes pueden realizar prácticas para la materia control inteligente.

## 1. CONCEPTOS BÁSICOS

En este capítulo se encuentra una breve descripción del estado actual de las técnicas a utilizar en el desarrollo del proyecto, tales como el esquema y estrategia de control, que se utilizarán en el desarrollo e implementación de los controladores, así como de la teoría necesaria para el progreso del mismo.

### 1.1. Control Clásico.

El control clásico de procesos consiste en la utilización de diferentes estrategias y métodos de control, que mantienen las variables de los sistemas en el valor deseado por el usuario.

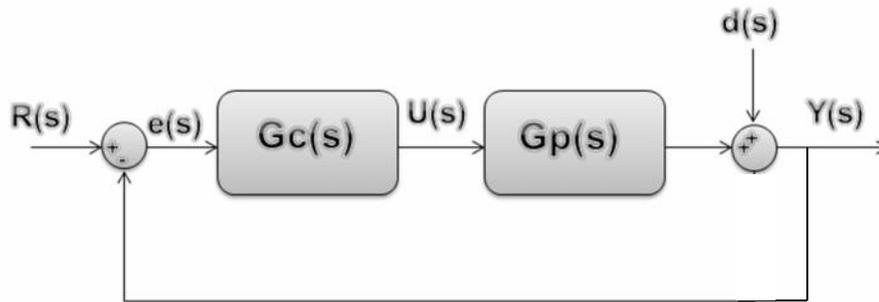
#### 1.1.1. Estrategias de Control.

Dada la gran complejidad de los procesos productivos, no es suficiente contar solo con un buen método de control, sino que debe complementarse con una estrategia de control adecuada según sea el caso. Dichas estrategias consisten en diferentes configuraciones, que proveen al sistema de mayor robustez, resistencia a perturbaciones externas, mejor desempeño, entre otros.

La estrategia de control por realimentación (*Feedback*), es la estrategia más conocida, utiliza la realimentación para que la salida de un sistema dinámico siga o tome un valor deseado pese a las perturbaciones externas o cambios de los parámetros internos del proceso.

En la estrategia de control realimentado ilustrada en la figura 1.1, una perturbación externa  $D(s)$ , altera su estado, la perturbación debe de propagarse por todo el sistema para que se pueda ver en la salida  $Y(s)$ , una vez la salida del sistema cambia, la señal de error  $E(s)$  cambia, dado que existe una diferencia con la referencia del sistema  $R(s)$ , el controlador detecta el cambio en la señal de error y entiende que debe de compensarlo aumentando o disminuyendo la señal de esfuerzo de control  $U(s)$  según sea el caso, permitiendo que el sistema regrese al valor deseado. Dentro de las ventajas de la estrategia de control realimentado, esta su sencilla implementación, y su capacidad de compensar la mayoría de perturbaciones, sin embargo se debe tener en cuenta que para corregir la

perturbación, ésta primero debe propagarse por todo el sistema, convirtiendo este suceso como su principal desventaja (Smith & Corripio, 1991).



**Figura 1.1** Estrategia de control en lazo cerrado

Es importante resaltar que existen diferentes estrategias aparte de la explicada anteriormente, si se requiere conocer a fondo sobre este tema remitirse al libro de Control Automático de Procesos, Teoría y Práctica (Smith & Corripio, 1991).

### 1.1.2. Técnicas de Control.

En el transcurso del tiempo se han creado diferentes tipos de técnicas con las cuales se puede realizar control sobre algún sistema, entre ellas la más común en la industria es el Control Tipo PID (Proporcional Integral Derivativo).

El controlador PID es un sistema que posee una componente proporcional al error, logrando que dicho error disminuya, sin lograr desaparecerlo; la acción integral da una salida al controlador que es proporcional al error acumulado, lo que ayuda a que el sistema reduzca el error a cero, aunque se debe tener en cuenta que la acción integral, agrega un poco de inestabilidad al sistema, y por último se tiene la acción de control derivativa, la cual permite obtener un controlador de alta sensibilidad, es decir que responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande, esto lo logra a través de la predicción del error futuro, consiguiendo de esta manera reducir las oscilaciones que se pueden presentar por la adición de la acción integral (Mazzones, 2002).

Sin embargo no existe una sola representación de los controladores PID, en (Rojas, 2007) se resumen diferentes representaciones alternativas. La teoría define principalmente la notación denominada ideal, en la cual el tiempo integral no influye

en la parte derivativa, al igual que el tiempo derivativo no influye en la parte integral, dicha cualidad ha llevado a que esta notación se conozca cómo no interactuante. Su representación en el dominio de la frecuencia se puede ver a continuación.

$$U(s) = K \left( 1 + \frac{1}{T_i s} + T_d s \right) e(s) \quad (1.1)$$

Donde  $k$  es la ganancia proporcional,  $T_i$  el tiempo integral y  $T_d$  el tiempo derivativo.

La configuración en serie de un controlador PID, es denominada interactuante, este tipo de controladores son especialmente escogidos, por su facilidad al ser sintonizados, son representados en el dominio de la frecuencia de la siguiente manera.

$$U(s) = K \left( 1 + \frac{1}{T_i s} \right) (1 + T_d s) e(s) \quad (1.2)$$

Sin embargo existen más configuraciones (Alfaro, 1993), (Alfaro, 2002), (Alfaro, 2002), entre ellas la ecuación del PID paralelo con limitación en la derivada que está dada por el valor de  $N$ , este valor se encuentra dentro de un rango de 5 a 30 (ecuación 1.3), dado que puede presentar dificultades si existe ruido de medición de alta frecuencia como se ve en (Amestegui Moreno, 2001) y otra denominada PID serie industrial (ecuación 1.4), en el cual la acción derivativa se efectúa sobre la señal de salida del sistema.

$$U(s) = K \left( 1 + \frac{1}{T_s i} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right) e(s) \quad (1.3)$$

$$U(s) = K \left( 1 + \frac{1}{T_s i} \right) \left( R(s) - \left( \frac{1 + T_d s}{1 + N T_d s} \right) Y(s) \right) \quad (1.4)$$

## 1.2. Control Inteligente.

En la actualidad, los avances en el estilo de vida del hombre, ha llevado a que cada día se exija más de los productos que se consumen a diario, llevando así a que los procesos productivos también evolucionen y aumenten su complejidad con el paso del tiempo. Desarrollo que provocó que las técnicas de control clásico no fueran suficientes para la exigencia actual, dando como resultado el surgimiento de diferentes técnicas avanzadas o inteligentes.

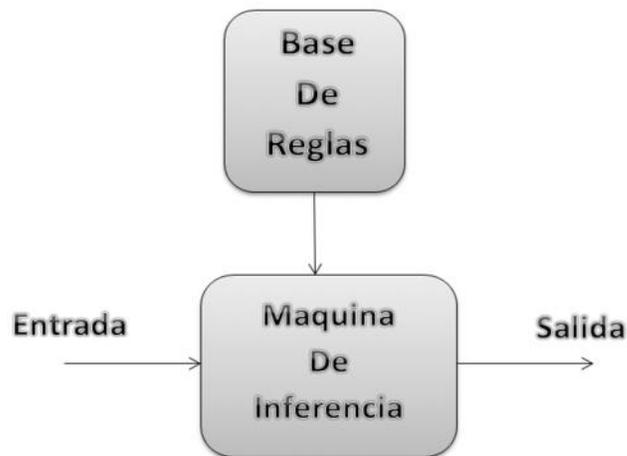
Estas técnicas tienen como objetivo resolver problemas de no linealidades, o de toma de decisiones entre otras, que no son factibles de resolver por procedimientos clásicos; en el control de procesos complejos se debe tener en cuenta, la posibilidad de realizar acciones de control en múltiples circunstancias, lo que vuelve idóneo la utilización de técnicas de inteligencia artificial para resolver dicha clase de problemas, con dos enfoques centrales, ayudar al diseño y construcción de técnicas de control convencionales y al control propiamente dicho (Sanzs, 1990).

### 1.2.1. Sistemas Difusos.

Los sistemas difusos intentan emular el esquema de razonamiento humano, bajo un marco de procesamiento numérico de la información, propio de los sistemas basados en el conocimiento. Esta técnica combina la teoría de los conjuntos difusos y la lógica difusa (Zadeh, 1965), logrando representar comportamientos complejos, que en su mayoría presentan no linealidades.

En la figura 1.2, se muestra la estructura básica de un sistema de inferencia difusa, con sus dos bloques fundamentales, el primero denominado base de reglas, el cual sintetiza y estructura los enunciados que constituyen el conocimiento que se posee sobre el problema en cuestión en forma de condiciones de tipo:

“Si A (Situación), Entonces B (Condición)”



**Figura 1.2** Diagrama de Bloques de un –Sistema de Inferencia Difuso

Este tipo de sentencias expresan el conocimiento cualitativo además de establecer una relación causa efecto, altamente adecuada para tareas de control.

El segundo bloque fundamental se denomina máquina de inferencia, el cual formaliza la capacidad de generalización de un experto humano, combinando la información almacenada en la base de reglas.

Para que los sistemas difusos puedan ser utilizados en tareas de control es necesario añadir dos bloques más, llamados *fuzzificador* que convierten valores reales en conjuntos difusos y *defuzzificador*, que convierte la salida del sistema difuso en valores reales.

Existen tres tipos fundamentales de sistemas de inferencia difusa, que principalmente son diferenciables por la forma como expresan el consecuente de la regla difusa que se ven en (Navas, 2000):

- Tipo 1: El caso más general, emplea reglas que son enunciadas en términos de variables lingüísticas, para la entrada como para la salida.

$$\text{Si } X_i \text{ es } A_i, \dots Y X_n \text{ es } B_i \text{ Entonces } Y = C_i$$

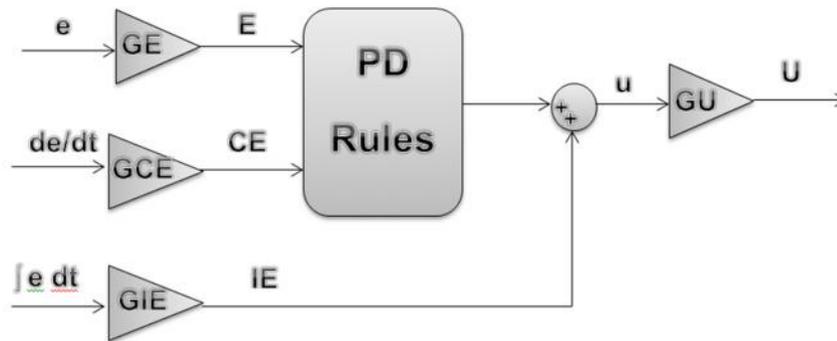
La salida difusa de cada regla se obtiene cambiando su función de pertenencia de salida, ya sea empleando un operador de mínimo (Método Mandani (Mamdani & Assilian, 1975)) o un operador producto.

- Tipo 2 (Modelo de Tsukamoto, referencia) (Tsukamoto, 1979): Emplea el mismo tipo de reglas que el anterior, diferenciándose en que la función de pertenencia de la salida  $C_i$  es una función monótona creciente (siempre crece), la salida final del sistema es la suma ponderada de cada salida de la regla, dada por el peso de la salida resultante del antecedente y por la función monótona creciente del consecuente.
- Tipo 3 (Modelo de Takagi – Sugeno,) (Takagi & Sugeno, 1985): Emplea reglas enunciadas en términos lingüísticos para las entradas, y el consecuente de cada regla se obtiene como una combinación lineal de los valores de entradas más un término constante.

$$\text{Si } X_i \text{ es } A_i, \dots Y X_n \text{ es } B_i \text{ Entonces } Y = f_i(X_1 \dots X_n)$$

### 1.2.2. Control PD+I Difuso.

Un controlador PD+I difuso actúa igual que un controlador del tipo PID clásico sobre una estrategia realimentada, la diferencia radica en que la señal de esfuerzo de controlador difuso es el resultado de una estrategia de control basado en reglas difusas.



**Figura 1.3** Esquema de un controlador PD+I Difuso, (Jantzen, 2002)

En la figura 1.3, se muestra el esquema que representa un controlador PD+I difuso, el cual funciona con tres entradas, el error, la integral del error y el cambio en el error, el sistema de inferencia difuso, solo trabaja con dos de las tres entradas, dado que es muy complicado su diseño si se tiene en cuenta la acción de los tres parámetros (Jantzen, 2002), por lo que se plantea la utilización de un sistema que representa un controlador PD difuso, al cual se le suma a su salida el valor de la integral del error, para mejorar el comportamiento del controlador, lo que simplifica el diseño del mismo. La ley de control está dada por.

$$U(n) = GE * GU \left[ e(n) + \frac{GCE}{GE} * e(n) + \frac{GIE}{GE} \sum_{j=1}^n e(j) * Ts \right] \quad (1,5)$$

Donde los valores de las constantes se obtiene de la sintonización de un controlador PID clásico y están dados por (Jantzen, 2002).

$$GE * GU = Kp \quad (1,6)$$

$$\frac{GCE}{GE} = Td \quad (1,7)$$

$$\frac{GIE}{GE} = \frac{1}{Ti} \quad (1,8)$$

### 1.2.3. Redes Neuronales Artificiales.

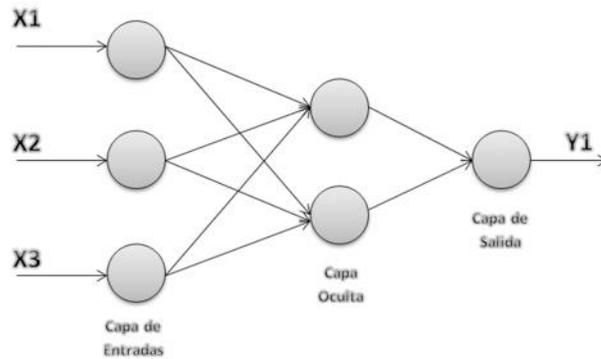
Las redes neuronales artificiales nacen de la necesidad de tener métodos de aprendizaje automático, éstas son inspiradas en el funcionamiento del cerebro humano, básicamente la idea es modelar artificialmente el comportamiento que tiene las redes neuronales biológicas.

Las redes neuronales artificiales ilustradas en la figura 1.4 (ANN de sus siglas en inglés), son un conjunto de procesadores simples, con capacidad de memoria limitada, donde cada procesador tiene una función fija, y se conectan entre sí para lograr una mayor interacción, convirtiendo la salida de uno en la entrada del otro. Las ANN presentan la capacidad de adaptar su arquitectura y parámetros, para llevar a cabo funciones de minimización de error entre otros, basándose en muestras representativas (datos entrada y salida). Las ANN se caracterizan básicamente por.

- ✓ Estar formada por un conjunto de unidades (Procesadores) con capacidad de procesamientos limitada.
- ✓ Dichos procesadores están unidos por enlaces ponderados.
- ✓ Posee parámetros adaptables (que deben ser ajustados), para satisfacer los requerimientos de diseño.

Se debe resaltar que la propiedad más importante de esta clase de sistemas es su capacidad de aprender a partir de parámetros de entrenamiento, en pocas palabras “Encontrar un modelo que se ajuste a los datos” (Salas, 2004).

Las ANN pueden ser entrenadas de manera supervisada o no supervisada, la diferencia entre las técnicas radica en que en el entrenamiento supervisado, se entregan datos de entrada y salida, y lo que se busca es que la red aprenda que ante determinado dato de entra, debe entregar determinado dato de salida, mientras que en el no supervisado solo se entregan las entradas y el algoritmo de aprendizaje debe encontrar una estructura o configuración en dichos datos de entrada (Salas, 2004).



**Figura 1.4** Red Neuronal Artificial.

#### 1.2.4. Sistemas Neuro-Difusos.

Los sistemas neuro-difusos nacen de la combinación de dos técnicas ya existentes, las redes neuronales artificiales y los sistemas difusos, con el fin de potenciar las ventajas que cada una de éstas individualmente y obviar las desventajas propias de los sistemas por separado.

Las principales características de las RNA son, su gran capacidad de aprendizaje y adaptación, sin embargo la forma como éstas procesan la información no es muy fácil de expresar en lenguaje natural, “Se dice que las redes neuronales no son transparentes al conocimiento estructurado” [Navas, 2000].

Por otra parte los sistemas difusos tienen una gran capacidad para agregar información suministrada por expertos a su estructura, resultando de gran utilidad para automatizar sistemas que los operarios controlan con facilidad, no obstante, su poca capacidad de adaptación es un inconveniente notorio (Navas, 2000).

Existen capacidades comunes entre ambas tecnologías, que los sistemas neuro-difusos heredan, entre ellos se encuentran mapeado no lineal, tolerancia a fallos, operación en tiempo real entre otros, además de esto, la fusión da nuevas habilidades que potencializan su utilidad, tales como “La transparencia propia de los sistemas difusos que mejoran la comprensión de la red neuronal” y “La capacidad de aprender y adaptarse de las redes neuronales que aportan un mecanismo de sintonización automática, y adaptabilidad al sistema difuso” (Navas, 2000).

### **1.2.5. ANFIS (Adaptive-Networks based Fuzzy Inference Systems).**

La red adaptativa de un sistema de inferencia difusa ANFIS, permite llevar el conocimiento adquirido en forma de datos (Entradas - Salidas), a un sistema de inferencia difusa tipo Takagi – Sugeno de primer orden. Esto quiere decir que ANFIS permite obtener un sistema difuso, sin la necesidad de un experto para el diseño del mismo.

El aprendizaje de la red es híbrido, posee dos parámetros que deben ser ajustados, antecedentes y consecuentes; utiliza la propagación hacia atrás de los errores para determinar los parámetros correspondientes a los antecedentes, mientras que los consecuentes se estiman utilizando el método de mínimos cuadrados.

ANFIS es clasificado como un sistema neuro-difuso, dado que estos sistemas son definidos como aquellos en los que se utiliza una red neuronal como herramienta para el diseño de sistemas difusos.

La arquitectura ANFIS es multipropósito, de tal manera que sus aplicaciones en campo son diversas, en ingeniería civil se realizó una implementación para predecir la mediana y el valor máximo, de la velocidad del aire dentro de un edificio, con el fin de investigar el uso potencial de la ventilación natural en un nuevo modelo de edificación (Ayata, Cam, & Yildis, 2007); en el área financiera es ampliamente aplicado, dada su capacidad predictiva, por ejemplo se utilizó un sistema basado en arquitectura ANFIS complementado con sistemas de aprendizaje por refuerzo para predecir los cambios en las tendencias del comercio, y de esta manera tomar una decisión de inversión (Tan, Quek , & Cheng, 2011) o en el pronóstico de un título de renta variable como se muestra en (Mendoza & Mazo , 2009) o para el pronóstico del precio de contratación de mercados de energía para el mercado colombiano (Arias & Ribon, 2009); en Colombia se han realizado varias implementaciones basadas en ésta arquitectura, entre ellas se encuentra el caso del modelo que se realizó para prever la generación mensual ideal de un agente con una planta hidráulica en el mercado eléctrico colombiano (Moreno , 2009).

En el área de control industrial, existen múltiples aplicaciones entre ellas se encuentra el control de un horno básico de oxígeno (BOF) utilizado en el segundo periodo de la fabricación de acero, el cual debe calcular los valores de las variables de control, cantidad de oxígeno y la necesidad de refrigerante (Han & Zhao, 2011). Otro tipo de aplicación se realizó en la Universidad de Firat (Turquía), donde se diseñó un HVAC (calefacción, ventilación y aire acondicionado), que busca controlar la temperatura y la humedad del interior del sistema, mediante la variación en la

apertura de dos compuertas las cuales son controladas con un PID. Los parámetros del PID se obtuvieron mediante el uso de un algoritmo difuso (Soyguder & Karakose, 2008), los cuales posteriormente se utilizaron para formar dos modelos ANFIS, que optimizaron el proceso (Soyguder & Alli, 2009).

Los sistemas de control basados en arquitectura ANFIS, son utilizados con gran frecuencia en ambientes industriales para aplicaciones de control de temperatura, un ejemplo es el realizado en un reactor tipo *batch*. La temperatura del reactor se controla mediante la manipulación de la temperatura del agua de refrigeración; el enfoque del diseño utiliza un controlador PI y un sistema ANFIS (Alippor, Zeinali, & Sadoghi, 2009). En la industria alimentaria, como la de los pollos de engorde en Indonesia, se utilizan sistemas de control para regular los parámetros ambientales como la temperatura, la humedad, la intensidad de la luz, y el nivel de contenido de amoníaco, los cuales son importantes y muy relevantes para mejorar la producción, se diseña un sistema de control ANFIS para obtener la misma temperatura durante los tres periodos del proceso (inicio, crecimiento y finalización) (Alimuddin, Seminar, Made Subrata, Nomura, & Sumiati, 2012).

Dentro de las diferentes aplicaciones que tiene la arquitectura ANFIS en el control de temperatura, se encuentran aquellas en las que se debe realizar un sistema híbrido entre diferentes tecnologías para alcanzar un mejor resultado, a continuación se citan algunos ejemplos como el que se realizó para obtener cristales de MgO (Óxido de magnesio) de alta pureza y de gran tamaño, donde se requiere que la distribución de la temperatura sea uniforme dentro de un horno, además que la temperatura de procesamiento sea controlada con precisión. Debido a la complejidad en la producción de cristales de MgO y la dificultad para medir la temperatura dentro del horno, se utilizó el método de elementos finitos (FEM), para estudiar la distribución de temperatura del interior y para regular la posición de los electrodos se realizó un control basado en ANFIS, dando como resultados una mejora en cuanto a la calidad y la cantidad de la producción (Li & Wang, 2011); o la aplicación en la que se implementó un control basado en ANFIS haciendo un híbrido con algoritmos genéticos para un sistema de extrusión de plástico, dado que el sistema es no lineal, y presenta una gran cantidad de disturbios, además de una pésima aproximación lineal al proceso, lo que conlleva a que la implementación de la estructura híbrida sea la más adecuada para el problema (Ravi, Sudha, & Balakrishnan, 2011).

En (Salunkhe, 2010) se comparan diferentes técnicas de control que utilizan sistemas de inferencia difusa y sistemas híbridos como ANFIS, esta comparación se enmarca sobre el control de temperatura de una cámara de agua, obteniendo como resultado que las arquitecturas multi-sistemas son más adecuadas para procesos

no lineales, recomendando entre ellas la arquitectura ANFIS. A continuación se citan ejemplos de otras comparaciones aplicadas al control de temperatura, entre ellos se encuentra la aplicación en la que se propone un controlador ANFIS de cinco capas para un calentador de agua doméstico, demostrando a través de una comparación con un controlador PID clásico que la propuesta híbrida es mucho mejor y genera resultados de mayor calidad (Tavoosi, Alaei, & Jahani, 2011); o el sistema de control ANFIS que se propone para controlar la temperatura de un sistema térmico, los resultados experimentales se compararon con un controlador PID convencional y un controlador de red neuronal. Todos los controladores son probados en diferentes condiciones de operación, mostrando un mejor rendimiento a la sintonización ANFIS (Mote & Lokhande, 2012).

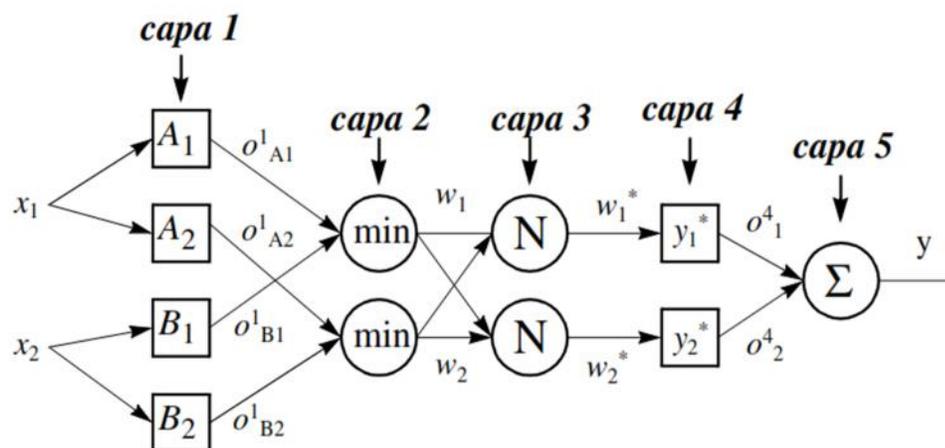
### 1.2.5.1. Arquitectura ANFIS.

A continuación se describirá la arquitectura de ANFIS (Jang, 1993) para un sistema que posee dos entradas y una salida que se muestra en la figura 1.5, que corresponden a las siguientes reglas difusas.

*Si  $X_1$  es  $A_1$  Y  $X_2$  es  $B_1$ , ENTONCES  $y_1$*

*Si  $X_1$  es  $A_2$  Y  $X_2$  es  $B_2$ , ENTONCES  $y_2$*

ANFIS consta de 5 capas, cuyas funciones se asocian a cada nodo de la siguiente forma.



**Figura 1.5** Arquitectura ANFIS para dos entradas. (Jang, 1993)

- Capa 1: Los nodos de esta capa tienen asociada la función de pertenencia correspondiente a la etiqueta lingüística de cada nodo.

$$o_{Ai}^1 = U_{Ai}(X_1) ; o_{Bi}^1 = U_{Bi}(X_2) ; i = 1,2 \quad (1,8)$$

- Capa 2: La función de este nodo es fija y consiste en realizar la T-norma entre los grados de pertenencia dada por la capa 1, dando el peso de la regla en el proceso de obtención de consecuente.

$$w_i = U_{Ai}(X_1) \times U_{Bi}(X_2), \quad i = 1,2 \quad (1,9)$$

- Capa 3: Esta capa tiene asociada la siguiente función:

$$w_i^* = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \quad (1,10)$$

Es decir, da el peso de la regla normalizada con respecto al peso de todas las reglas.

- Capa 4: Multiplica el peso normalizado del antecedente de la regla, por el valor del *singleton* asociado al consecuente.

$$o_i^4 = w_i^* y_i^*, \quad i = 1,2 \quad (1,11)$$

- Capa 5: Contiene un solo nodo que agrega los *singletons* ponderados para obtener la salida del sistema.

Anteriormente se afirmó que ANFIS permite obtener un sistema de inferencia difusa sin la necesidad de un experto que lo diseñe, la arquitectura evidencia dicha información dado que cada una de las capas que pertenecen a la red cumplen una función propia de un sistema difuso, se debe tener en cuenta que la red tiene dos capas cuyos nodos son adaptables, la capa 1 que hace referencia a la función de pertenencia de un conjunto difuso de la variable antecedente y la capa 4 que ajusta el valor de las constantes del consecuente, identificados en la figura 1.5 con los nodos de tipo cuadrado, y que al momento de terminar el entrenamiento, han tomado el valor óptimo, lo cual equivale al trabajo realizado por el experto en el diseño de un sistema difuso, que es la persona encargada de ajustar dichos parámetros, basado en su conocimiento.

## 2. DISEÑO DE LOS CONTROLADORES

En este capítulo se presenta el diseño de los controladores implementados en la planta, controlador PID serie industrial, controlador PD+I Difuso y el modelo del controlador basado en ANFIS.

Inicialmente se muestra una descripción general de la planta, además de los dispositivos que hacen parte del sistema, después se realiza la sintonización de los controladores PID, el diseño del controlador PD+I Difuso y finalmente el modelo del controlador basado en ANFIS. La prueba del diseño de los controladores se implementa en la herramienta de Matlab, Simulink, donde se observa el comportamiento del sistema con la implementación de cada uno de los controladores frente a una señal de referencia y ante un disturbio, al igual que la señal del esfuerzo de control realizada por cada controlador.

### 2.1. Descripción de la Planta de Temperatura del LCP.

En la planta de temperatura se pueden distinguir dos espacios: campo y control. El primer espacio es el lugar en el que se encuentra la planta térmica junto con los sensores y resistencias de calefacción donde se desarrolla el proceso físico, ver figura 2.1. El segundo de ellos es el llamado panel de control o cableado, donde se implementan los dos escenarios de automatización de la planta: basado en PC y basado en PLC ver figura 2.2 (Laboratorio, 2010).



**Figura 2.1** Planta de control de temperatura LCP (Fuente Propia).

La figura 2.1 muestra la planta de temperatura que consiste en un tubo cilíndrico unido a una cámara donde ingresa aire a temperatura ambiente. Al interior de la cámara se encuentra un ventilador, el cual conduce el flujo de aire aspirado a una zona de calentamiento con dos resistencias de calefacción de 1100w conectadas en serie y alimentadas por una fuente de 220v, en esta zona el aire aumenta su temperatura y sale despedido por el tubo cilíndrico a una determinada temperatura,

donde se encuentran ubicados dos sensores, el primero cerca a la conexión del tubo con la cámara y el segundo a la salida de aire del tubo. La variable controlada es la temperatura en el interior del cilindro y se propone tomar como variable manipulada en este proceso el voltaje que llega a las resistencias, dado que el actuador de la planta permite el paso o no de voltaje hacia el sistema, voltaje que se manipula desde el controlador. En la planta de temperatura, la presencia de disturbios son generados modificando manualmente la posición de una compuerta en la toma de aire de la cámara.



**Figura 2.2** Panel de control planta de temperatura (Laboratorio, 2010).

En el panel de control o cableado (figura 2.2) se reciben y entregan las señales utilizadas en campo, en él se encuentran.

- Fuentes de 110VA.C, 220V A.C y 24V D.C
- Borneras que distribuyen los diferentes voltajes requeridos por los instrumentos
- Un PLC Micrologix 1500 de Allen Bradley que se encarga del control en el escenario PLC
- Llaves selectores encargadas del escenario que se requiere utilizar y del funcionamiento manual o remoto del ventilador
- Un botón de encendido y apagado del ventilador
- Luces piloto que indican el estado en el que se encuentra la planta
- Relés electromecánicos y térmicos que tienen entre sus funciones principales distribuir las señales hacia los dos escenarios de automatización según la posición de la llave selectora y proteger al ventilador de cualquier sobrecarga.
- Dos transmisores indicadores de temperatura que transmiten la señal de los sensores de temperatura PT1000 al PLC o al PC dependiendo del escenario de automatización que se esté utilizando

## 2.2. Diseño de los Controladores PID

Para el diseño de los controladores PID se realiza como primera instancia la identificación de la planta de temperatura, con el fin de obtener un modelo matemático de la planta que simule el comportamiento dinámico de la misma, para posteriormente diseñar los controladores PID paralelo y PID serie industrial, que serán la base para el diseño del controlador PD+I Difuso y del controlador modelado por ANFIS, respectivamente.

### 2.2.1. Identificación de la planta

Para el diseño del controlador PID de la planta de temperatura se realizó como primera instancia la identificación de la misma con el método de la curva de reacción (Nichols & Ziegler, 1942). Según las características de la planta se obtuvo un modelo de primer orden más tiempo muerto (FOPDT).

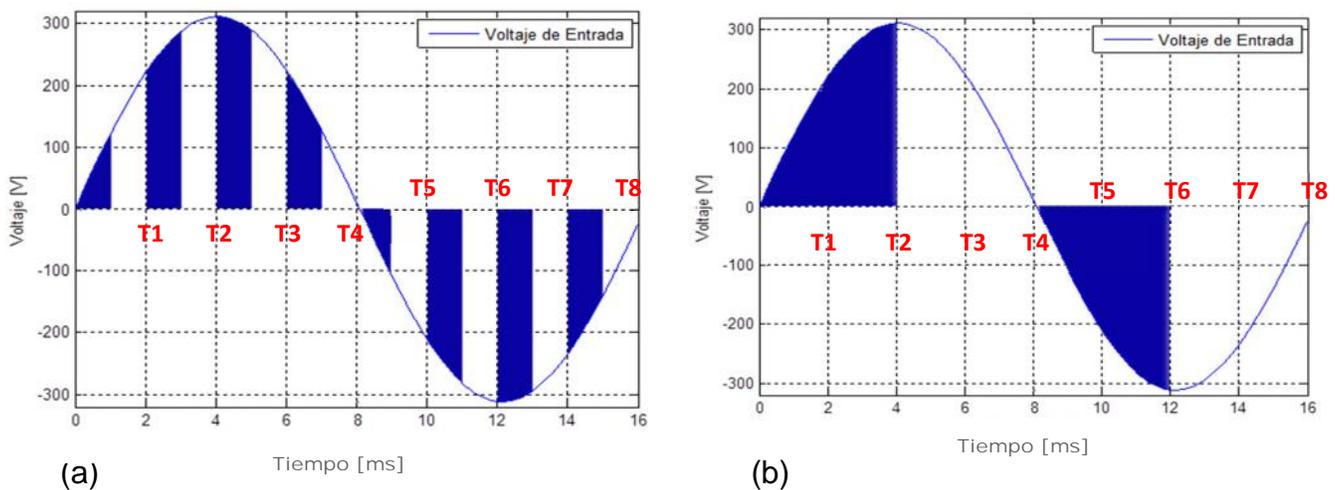
Los tres parámetros a ser ajustados en este modelo son: la ganancia del proceso, la constante de tiempo del proceso y el tiempo muerto, estos se observan en la ecuación 2.1, donde  $Y(s)$  es la transformada de Laplace de la variable controlada la cual toma un valor entre 24 a 75 °C aproximadamente y  $U(s)$  es la transformada de la variable manipulada la cual varía de 0 a 220 v.  $Kp$  es la ganancia de la planta en estado estacionario,  $\tau$  la constante de tiempo y  $t_m$  el tiempo muerto o retardo.

$$\frac{Y(s)}{U(s)} = \frac{Kp e^{-t_m s}}{\tau s + 1} \quad (2.1)$$

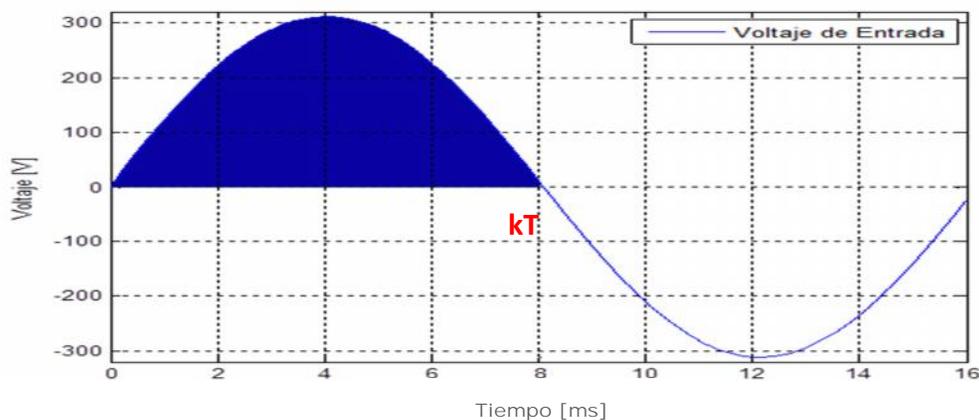
Con el fin de obtener los parámetros del FOPDT, se aplica experimentalmente a la planta en físico un porcentaje de 30% del voltaje que llega a las resistencias, se espera alrededor de doce minutos a que se estabilice en un valor determinado de temperatura, en este caso de 31°C aproximadamente, después se aumenta el porcentaje a 70% del voltaje y cuando se estabiliza nuevamente se toma el valor de temperatura el cual es de aproximadamente 50°C, se tiene en cuenta que el máximo voltaje que puede llegar a las resistencias es 220  $V_{a.c}$  por lo tanto el 30% del voltaje es 66 $V_{a.c}$  y el 70% es de 154 $V_{a.c}$ , el comportamiento de la planta es el mostrado en la figura 2.3, el cual es fundamental al momento de determinar los valores de los parámetros del modelo. Para obtener los parámetros se utilizó el método general de los dos puntos de Smith, (Smith C. , 1972).

Para realizar la identificación de la planta, se cuenta con una tarjeta Arduino Mega 2560 R3, de la cual para obtener los datos experimentales se hace uso de las

entradas analógicas con el fin de leer la señal que proviene de los sensores y la salida del PWM para controlar el voltaje que llega a la planta, por esto se debe tener en cuenta que el periodo de la señal de PWM que tiene la tarjeta es de aproximadamente 2ms (Arduino, Sitio web Arduino) y que el voltaje que ingresa a la planta es A.C con un periodo de 16 ms, por tal motivo para obtener la ecuación que relaciona la salida de la tarjeta Arduino (PWM) con la variable manipulada (voltaje) se considera que en un periodo de la señal manipulada hay aproximadamente 8 periodos de la señal de PWM generada por la tarjeta (Figura 2.3), por consecuente se establece que la sumatoria del voltaje que deja pasar los 8 ciclos es equivalente a la sumatoria desde 0 hasta  $kT$  donde  $k$  es el ciclo activo del PWM y  $T$  es el periodo de la onda manipulada de la señal de voltaje A.C (Figura 2.4).



**Figura 2.3** Señal de voltaje: (a) con periodos de 2ms de PWM. (b) con área aproximada que pasa del PWM



**Figura 2.4** Salida de voltaje con  $kT$  aproximado de paso del PWM

Para obtener la relación se aplica la siguiente ecuación.

$$V_E^2 = \int_0^T (Vm \text{ sen } (wt))^2 dt \quad (2.1)$$

Donde :

$V_E$  = Voltaje eficaz de la señal

$Vm$  = Voltaje máximo ( $\approx 311 V_{a.c}$ )

$T$  = el periodo de la señal

A partir de la ecuación se obtiene la relación que determina la cantidad de voltaje que llega a la planta es función de la señal de PWM es:

$$V_E^2 = \int_0^{kT} (Vm \text{ sen } (wt))^2 dt$$

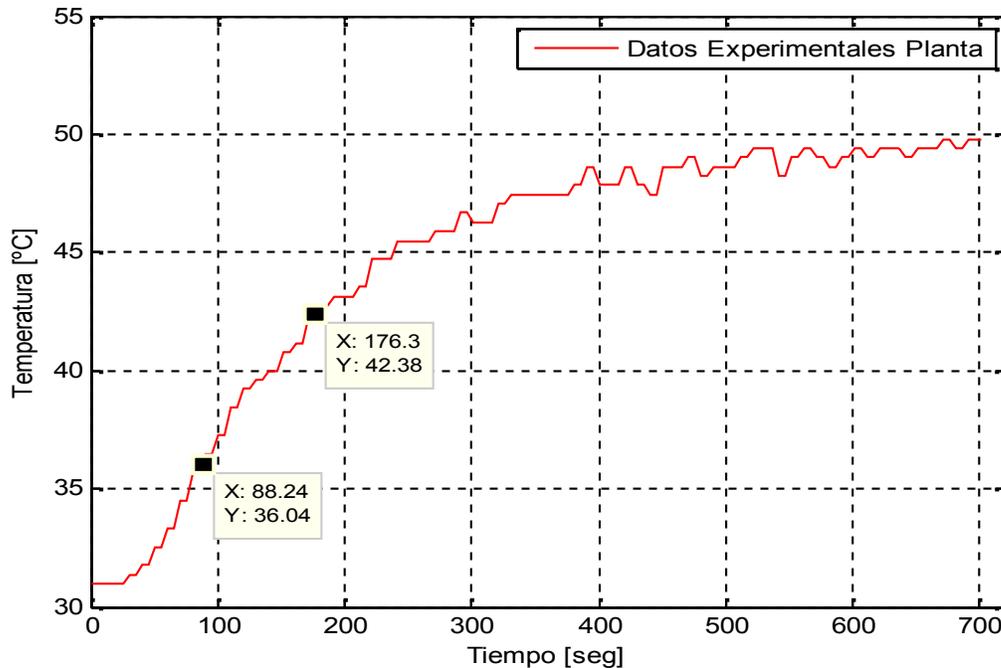
$$V_E^2 = \frac{Vm^2}{T} \left[ \frac{kT}{2} - \frac{\text{Sen } (4\pi k)}{4w} \right]$$

$$0 \leq k \leq 1$$

$$V_E = \frac{Vm}{\sqrt{2}} * \sqrt{k}$$

$$V_E = 220V_{a.c} * \sqrt{k}$$

A partir de esta ecuación se determina que el 30% del voltaje que llega a la planta es de  $66 V_{a.c}$  equivalente a un 9% de PWM y que el 70% del voltaje es de  $154 V_{a.c}$  que equivale a 49% de PWM (Figura 2.5).



**Figura 2.5** Respuesta de la planta de temperatura a una variación del 30 al 70% del voltaje

La ganancia del proceso  $k_p$  es el cambio total en la salida ( $\Delta y = y_f - y_i$ ) dividido por el cambio en la entrada ( $\Delta u = u_f - u_i$ ). Un cambio del 30 al 70 por ciento del voltaje que llega a las resistencias equivale a un cambio de 66 a 154 voltios respectivamente, dando como valor para  $k_p$  aproximadamente  $0,2967 \text{ } ^\circ\text{C/V}$

$$K_p = \frac{\Delta y}{\Delta u} \quad (2.2)$$

$$K_p = \frac{(49.08 - 30.98)^\circ\text{C}}{(154 - 66) V_{a.c}} \approx 0,2056$$

Después de obtener  $k_p$  se procede a hallar los valores de  $\tau$  y  $t_m$  de la siguiente forma.

Los instantes seleccionados por Smith fueron los tiempos requeridos para que la respuesta alcance el 28.3% ( $t_{28}$ ) y el 63.2% ( $t_{63}$ ) del valor final (figura 2.5), y corresponden a,

$$t_{28} = 88.24s$$

$$t_{63} = 176.03s$$

Después de obtener los tiempos, según Smith se resuelven las siguientes ecuaciones para hallar los parámetros  $\tau$  y  $t_m$ ,

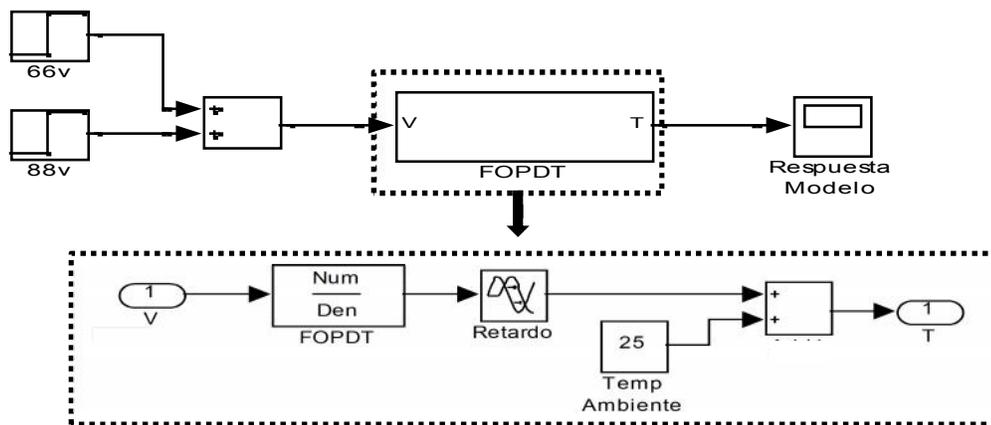
$$\tau = 0.9102(t_{63} - t_{28}) = 79.90s \quad (2.3)$$

$$t_m = 1,262t_{28} - 0,262t_{63} = 65.23s \quad (2.4)$$

Por lo tanto el modelo de primer orden más tiempo muerto queda de la siguiente forma:

$$\frac{Y(s)}{U(s)} = \frac{0.2056 e^{-65.23s}}{79.90s + 1}$$

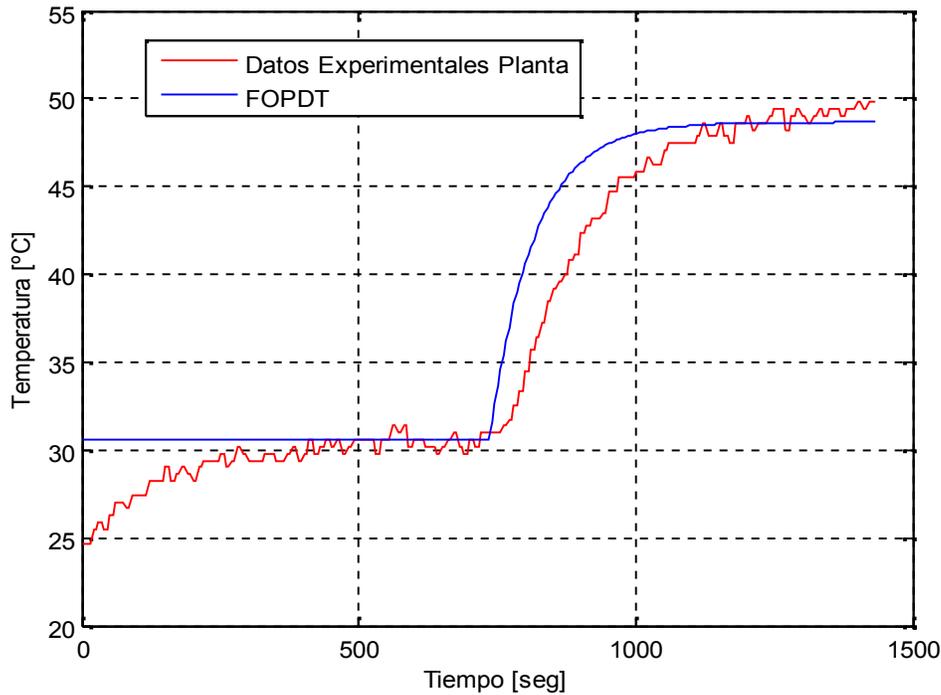
La implementación del modelo de la planta se realizó con la ayuda del editor de Matlab y Simulink (Figura 2.6), donde al bloque que representa el modelo de la planta (FOPDT) se aplica un cambio de voltaje de 66 a 154  $V_{a.c}$  (30% a 70%) con la ayuda del bloque "step" el cual inicialmente genera una señal tipo escalón que simula los 66 v y después es necesario sumarle otra señal del mismo tipo pero de 88v para poder realizar el cambio de 66 $V_{a.c}$  a 154 $V_{a.c}$ . Se tiene en cuenta que al FOPDT se le debe adicionar la temperatura ambiente que en este caso es de aproximadamente 25°C.



**Figura 2.6** Implementación del FOPDT en Matlab Simulink

Al obtener el modelo de forma matemática, se compara su respuesta, con la curva de reacción obtenida experimentalmente, dando como resultado la figura 2.7, en la cual se puede notar que las dos señales tienen un comportamiento similar sin embargo, la temperatura en la que se estabilizan cuando se le aplica el 70% del

voltaje, tiene una diferencia de aproximadamente 3 °C, diferencia que puede ser generada principalmente por la temperatura ambiente con la que se tomaron los datos experimentales, ya que puede haber sido superior o inferior a 25°C, condición que para el sistema implementado en la planta de temperatura, es muy difícil de corregir.



**Figura 2.7** Comparación entre la respuesta del FOPDT y la curva de reacción experimental de la planta.

### 2.2.2. Sintonización de controlador PID serie industrial

Al tener el modelo de la planta, se procede a sintonizar las constantes del PID. Se utilizará un PID serie industrial el cual se rige por la ecuación 1.4.

Como método de sintonización se utilizó el de Kaya y Sheib (Alfaro, 2002), el cual se preside por las siguientes ecuaciones.

$$K_c K_p = a \left( \frac{tm}{\tau} \right)^b \quad (2.5)$$

$$\frac{T_i}{\tau} = \frac{1}{c} \left( \frac{tm}{\tau} \right)^{-d} \quad (2.6)$$

$$\frac{T_d}{\tau} = e \left( \frac{tm}{\tau} \right)^f \quad (2.7)$$

Los valores de  $a$  a  $f$  que se utilizaron son los del criterio de desempeño *integral del error absoluto* (IAE) (tabla 2.1).

**Tabla 2.1** Parámetros IAE

IAE	0,98089	0,76167	0,91032	-1,05211	0,59974	0,89819
-----	---------	---------	---------	----------	---------	---------

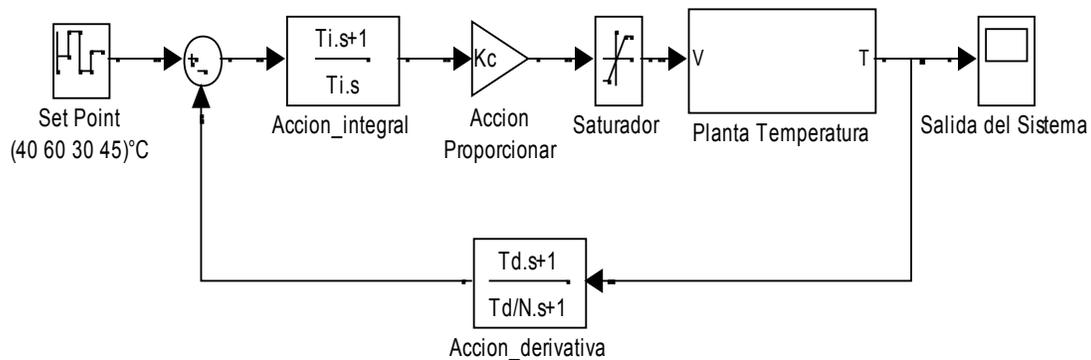
Partiendo de los valores obtenidos en la identificación de la planta y de los parámetros para el IAE se obtienen los siguientes valores para las variables del PID serie industrial, de igual forma se requiere el valor de  $N$  (ver sección 1.1.2) el cual se maneja en un rango de 5 a 30, para este caso se elige un  $N$  de 20.

$$K_c = 3,9471$$

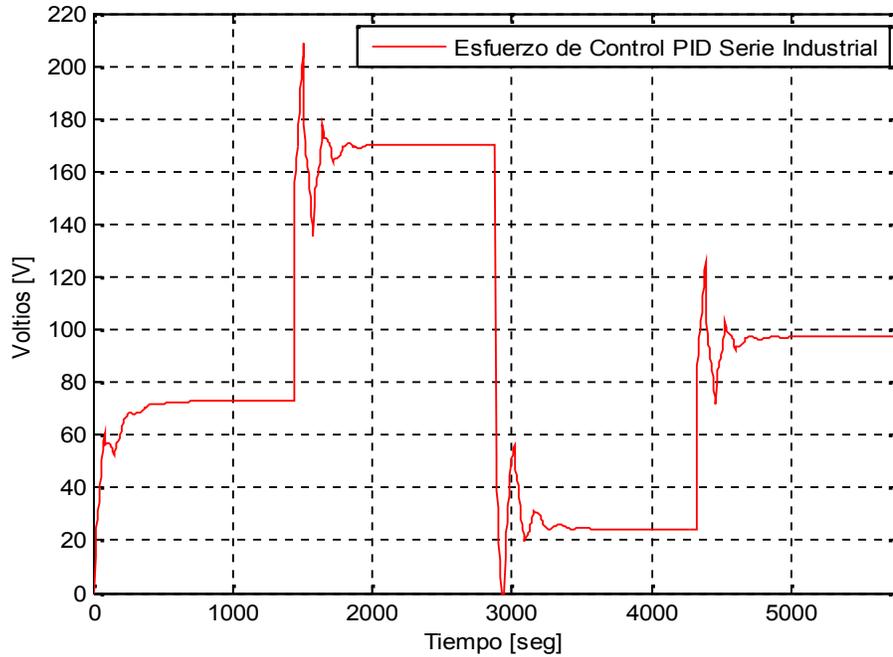
$$T_i = 49,1971$$

$$T_d = 90.5143$$

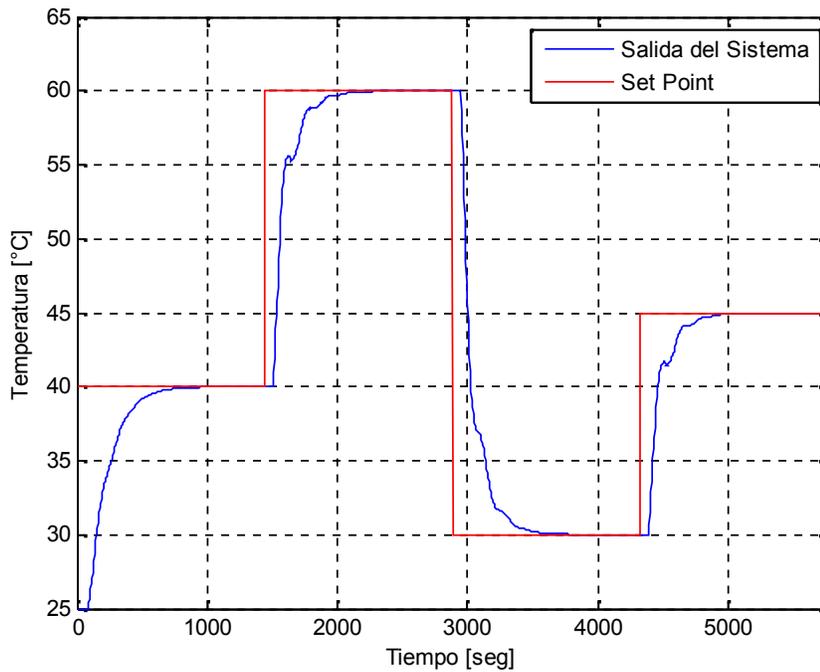
La prueba de la sintonización del PID, se realiza con la ayuda de la herramienta de Simulink-Matlab, dando como resultado el esquema de la figura 2.8, además en la figura 2.9 se gráfica la señal del esfuerzo del controlador, donde se puede observar que no sobrepasa el voltaje máximo de la planta, además al presentar picos altos permite que se acerque rápidamente a el valor deseado; por otra parte en la figura 2.10 se observa la respuesta de la planta bajo el control del PID serie industrial.



**Figura 2.8** Diseño del controlador PID serie industrial



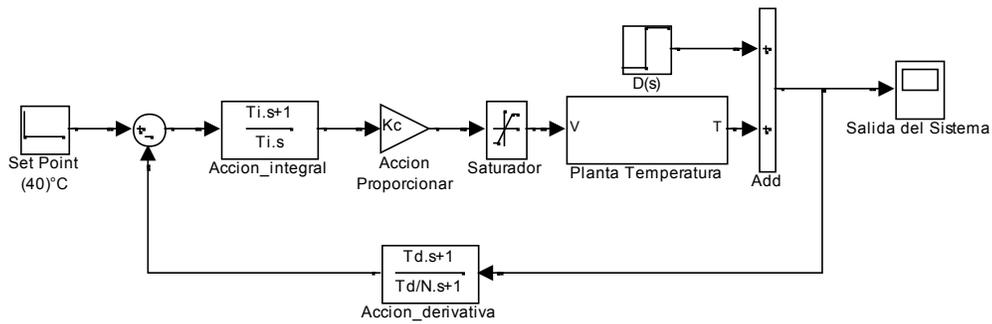
**Figura 2.9** Esfuerzo de control del controlador PID serie industrial



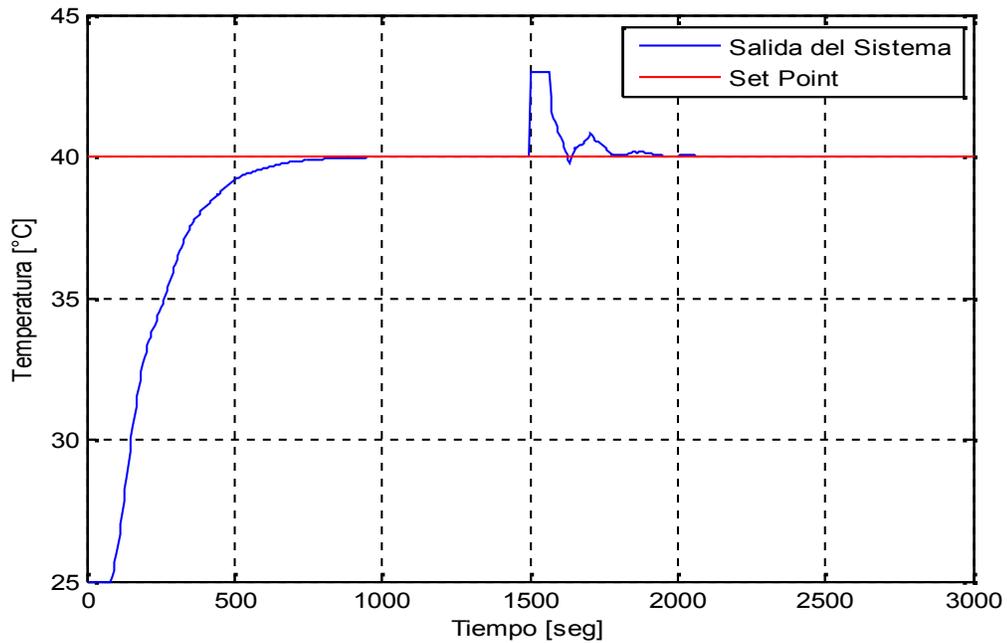
**Figura 2.10** Respuesta del sistema bajo el controlador PID a diferentes *Set Point*.

Para observar la respuesta del controlador ante un cambio inesperado de temperatura de 3°C en el interior del cilindro se realizó el esquema de la figura 2.11,

como muestra la figura 2.12 el controlador responde satisfactoriamente ante la presencia de un disturbio.



**Figura 2.11** Diseño del controlador PID serie industrial con presencia de disturbio.



**Figura 2.12** Respuesta del sistema bajo el controlador PID ante un aumento de 3°C al segundo 1500.

### 2.3. Controlador PD+I Difuso.

Se realiza el diseño e implementación en la planta de temperatura de un controlador PD+I Difuso con el fin de comparar la respuesta que tiene el sistema con el

controlador modelado por ANFIS. Para esto se debe realizar inicialmente la sintonización de un controlador PID paralelo, por ser la base de los controladores PD+I Difusos.

### 2.3.1. Sintonización del controlador PID paralelo

Al tener el modelo de la planta, se procede a sintonizar las constantes del PID. Se utilizará un PID paralelo con restricción en la derivada el cual se rige por la ecuación 1.3.

Las ecuaciones para encontrar los parámetros  $K_p, T_i, T_d$ , de Ziegler y Nichols son las siguientes (Nichols & Ziegler, 1942).

$$K_c = 1.2 \frac{\tau}{K_p tm} \quad (2.8)$$

$$T_i = 2 tm \quad (2.9)$$

$$T_d = 0.5 tm \quad (2.10)$$

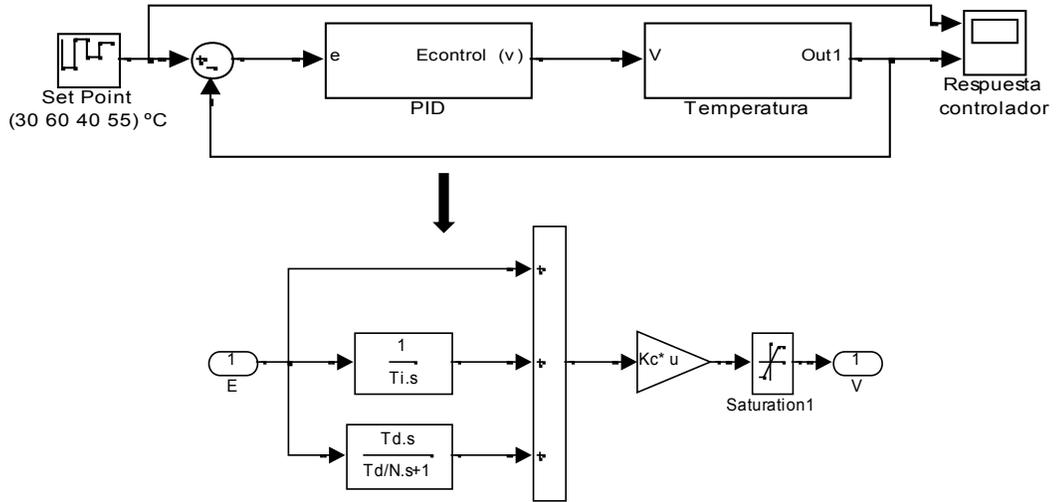
Reemplazando los valores de  $K_p, tm, \tau$ , se obtienen los siguientes parámetros de sintonización, además se debe tener en cuenta que como se realizará un PID paralelo con restricción en la derivada se escoge un valor de  $N$  de 5, como se explica en la sección 1.1.2.

$$K_c = 3.5541$$

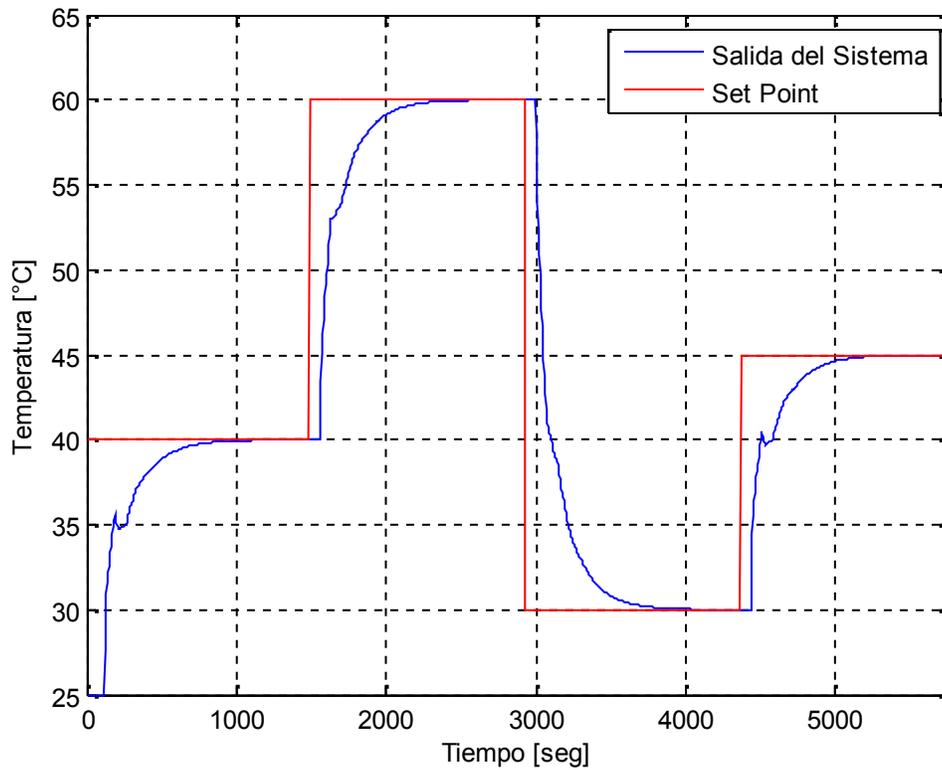
$$T_i = 130.46$$

$$T_d = 32.6150$$

La prueba de la sintonización del PID, se realiza con la ayuda de la herramienta de Simulink-Matlab, se establece un tiempo de estabilización de 24 minutos para cada cambio en la señal de referencia dando como resultado el esquema de la figura 2.13 y la respuesta del sistema al controlador de la figura 2.14.

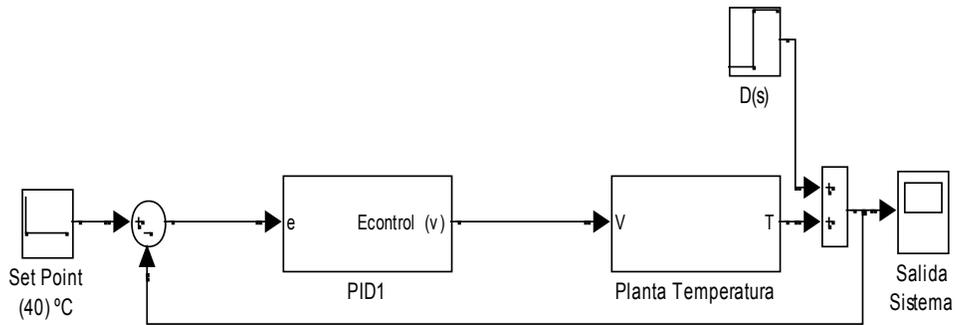


**Figura 2.13** Diseño del controlador PID

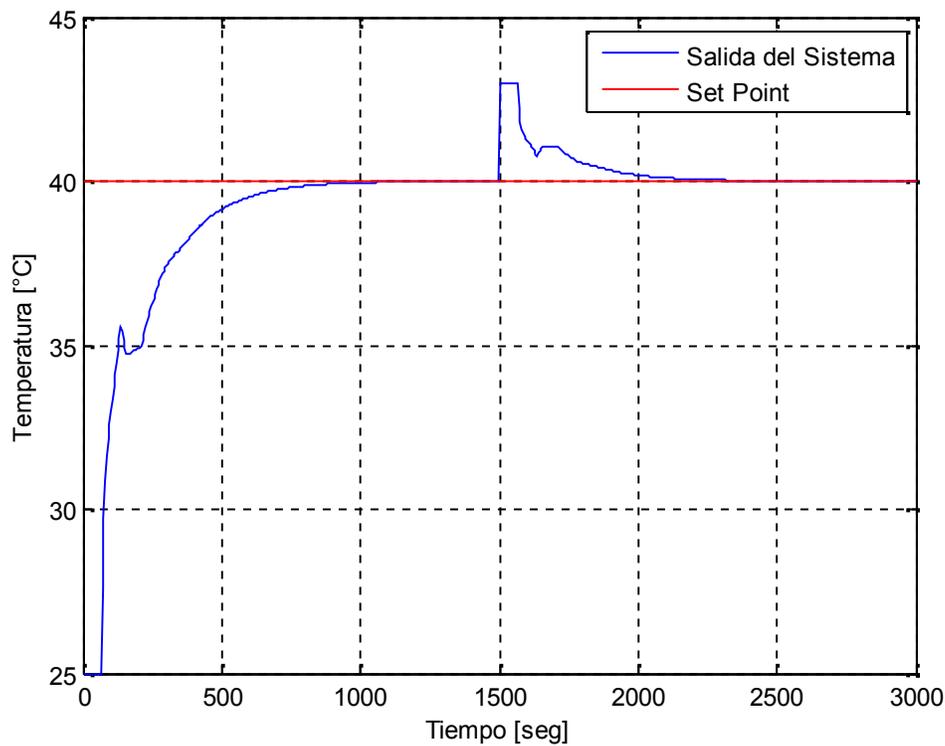


**Figura 2.14** Respuesta del controlador PID de la planta de temperatura a diferentes *Set Point*.

Se realiza el esquema de la figura 2.15, para observar el comportamiento que tiene el controlador, al momento de presentarse un aumento de 3°C en la temperatura en el interior del cilindro, como muestra la figura 2.16 el controlador responde satisfactoriamente ante la presencia de un disturbio.



**Figura 2.15** Esquema PID con disturbio



**Figura 2.16** Respuesta del controlador ante un disturbio aplicado en el segundo 1500 .

### 2.3.2 Diseño del controlador PD+I Difuso.

Para realizar el diseño del controlador PD+I difuso explicado en la sección 1.2.2 de este documento, es necesario sintonizar algunos parámetros al igual que en las técnicas clásicas, sin embargo para hallar estos valores, se parte del  $K_p, T_i, T_d$  del PID paralelo con restricción en la derivada sintonizado anteriormente, las constantes a sintonizar son  $GE, GU, GCE, GIE$ .

$$GE * GU = K_p \quad (2.11)$$

$$\frac{GCE}{GE} = T_d \quad (2.12)$$

$$\frac{GIE}{GE} = \frac{1}{T_i} \quad (2.13)$$

Se establece que GE tiene un valor de 1, con el fin de no afectar la escalización de los demás valores.

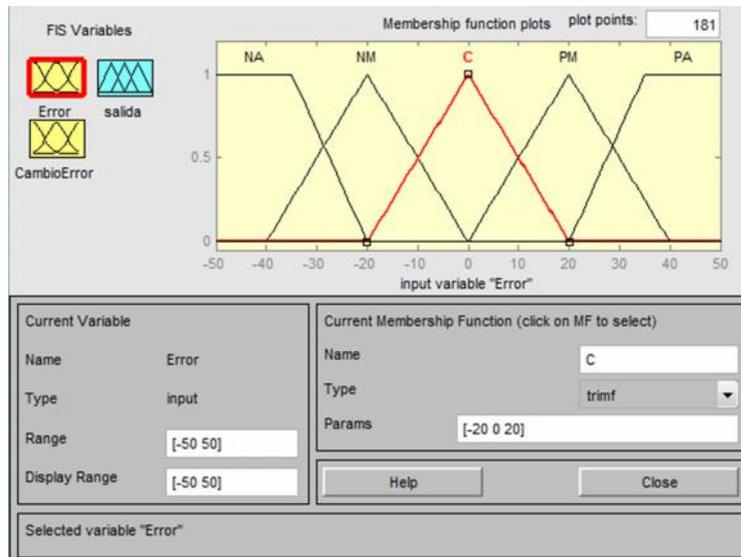
$$GU = \frac{K_p}{GE} = 3.5541$$

$$GCE = T_d * GE = 32.6150$$

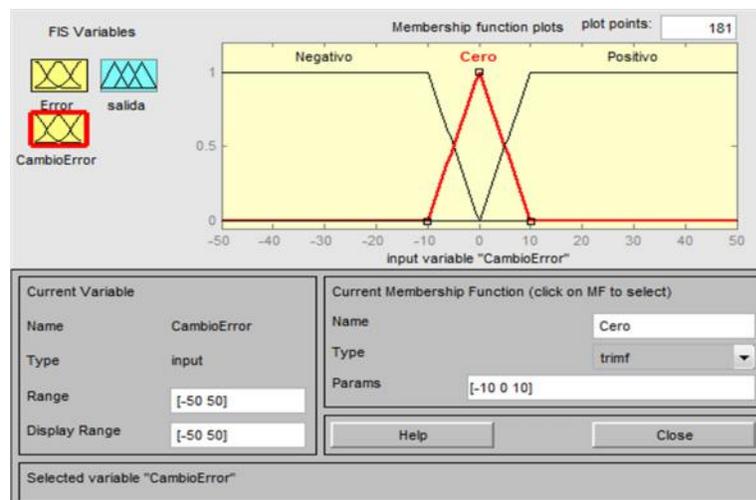
$$GIE = \frac{GE}{T_i} = 0.0077$$

Además de sintonizar los parámetros, se debe realizar un controlador difuso tipo PD, para esto se hace uso del toolbox de sistemas difusos de Matlab.

Donde se establece que los conjuntos de entrada al controlador difuso son: el Error (Figura 2.17), en el que se especifican cinco funciones de membresía (MFs): NA (Negativo Alto), NM (Negativo Medio), C (Cero), PM (Positivo Medio), PA (Positivo Alto).y el conjunto CambioError (Figura 2.18), el cual cuenta con tres funciones de membresía: Negativo, Cero y Positivo.

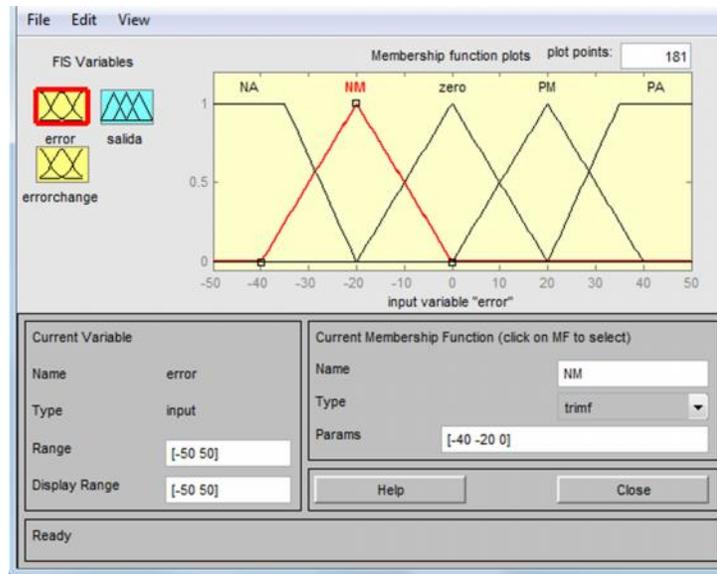


**Figura 2.17** Conjunto difuso de entrada “Error”. Toolbox fuzzy



**Figura 2.18** Conjunto difuso de entrada “CambioError”. Toolbox fuzzy

Igualmente se establece un conjunto de salida del controlador (Figura 2.19), que cuenta con cinco MFs: NA (Negativo Alto), NM (Negativo Medio), C (Cero), PM (Positivo Medio), PA (Positivo Alto).

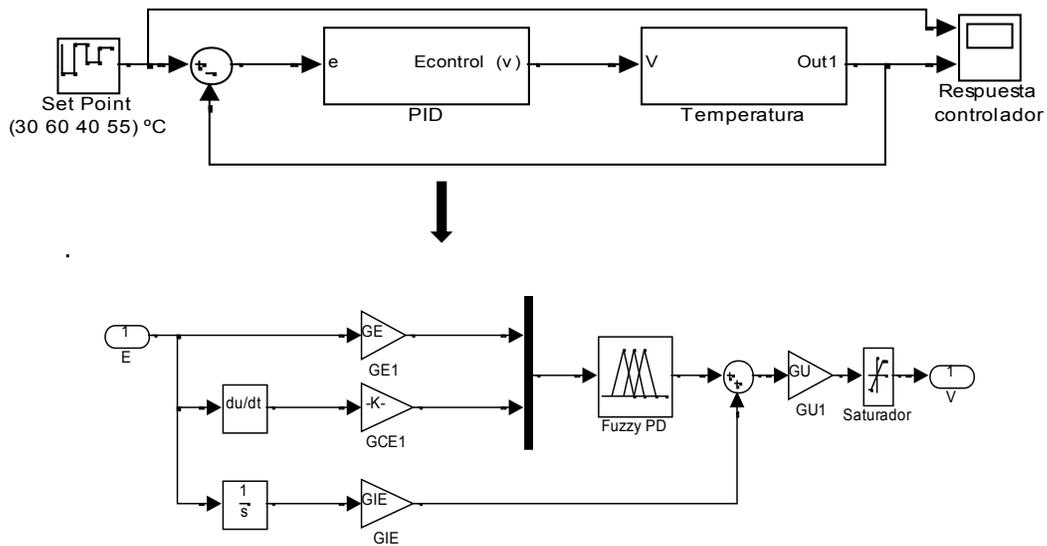


**Figura 2.19** Conjunto difuso de salida. “Salida” Toolbox fuzzy.

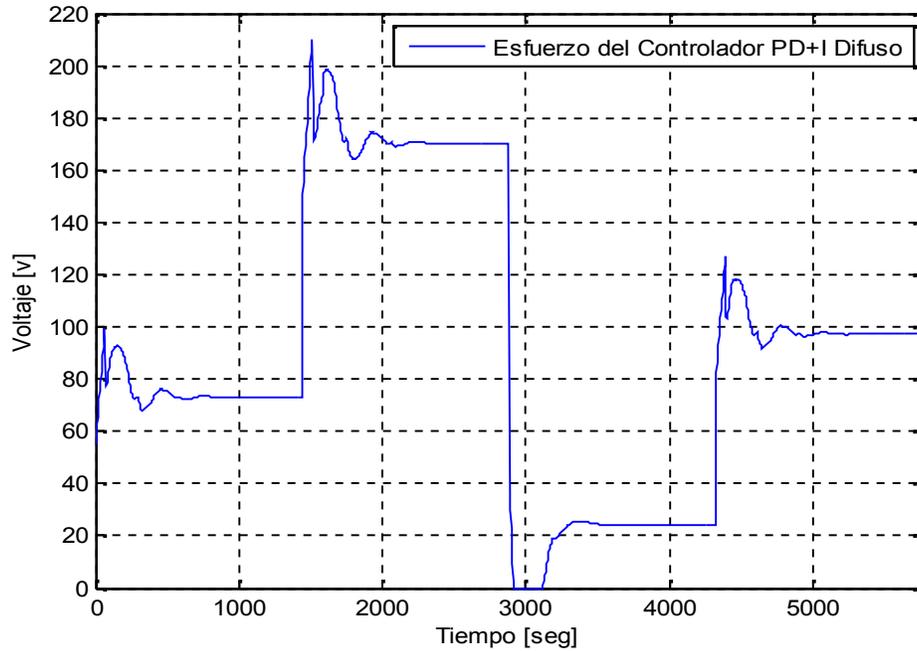
Para el control del sistema difuso, se realizaron trece reglas tipo Mandani en las que se tiene en cuenta todas las posibles variaciones que pueda presentar el sistema (Jantzen, 2002).

1. Si error es NA y CambioError is Negativo entonces salida es NA
2. Si error es NA y CambioError is Cero entonces salida es NA
3. Si error es NA y CambioError is Positivo entonces salida es NA
4. Si error es NM y CambioError is Negative entonces salida es NM
5. Si error es NM y CambioError is Cero entonces salida es NM
6. Si error es NM y CambioError is Positivo entonces salida es NM
7. Si error es C y CambioError is Negative entonces salida es C
8. Si error es C y CambioError is Cero entonces salida es C
9. Si error es C y CambioError is Positivo entonces salida es C
10. Si error es PM y CambioError is Positivo entonces salida es PM
11. Si error es PM y CambioError is Cero entonces salida es PM
12. Si error es PA y CambioError is Positivo entonces salida es PA
13. Si error es PA y CambioError is Cero entonces salida es PA

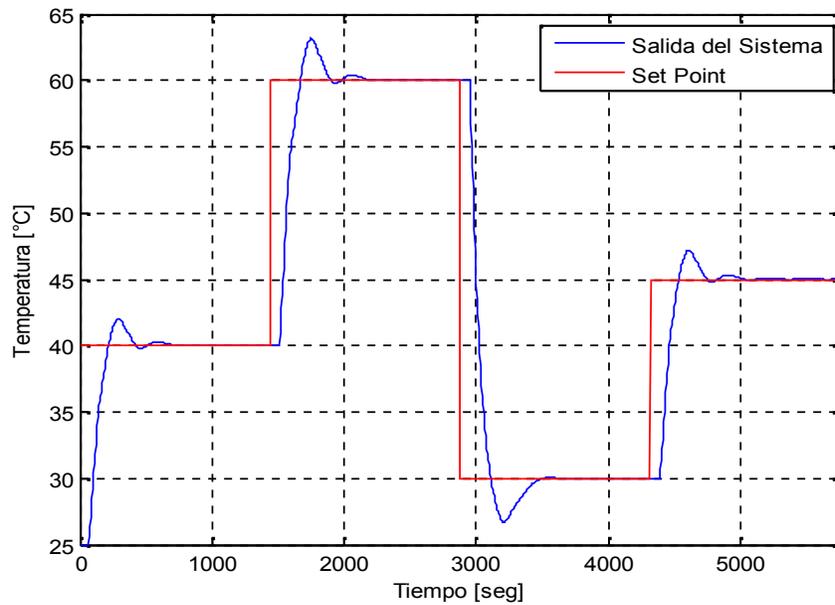
Después de obtener el sistema difuso con la ayuda de la interfaz de Fuzzy en Matlab, se diseña el control PD+I difuso en la interfaz Simulink de Matlab (Figura 2.20), teniendo en cuenta los parámetros definidos anteriormente y un tiempo de estabilización de 24 minutos para cada *Set Point*, da como respuesta del sistema la gráfica de la figura 2.22. Además se adquiere la señal del esfuerzo del controlador, la cual no sobrepasa el valor máximo de voltaje para la planta y posee picos que permite llegar al valor deseado de una forma rápida (figura 2.21).



**Figura 2.20** Diseño controlador PD+I Difuso.



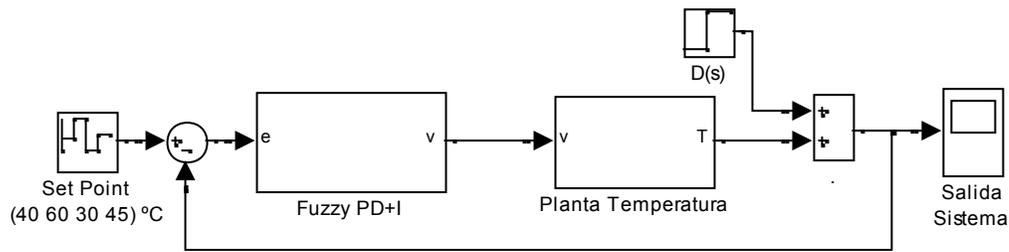
**Figura 2.21** Esfuerzo de control PD+I Difuso.



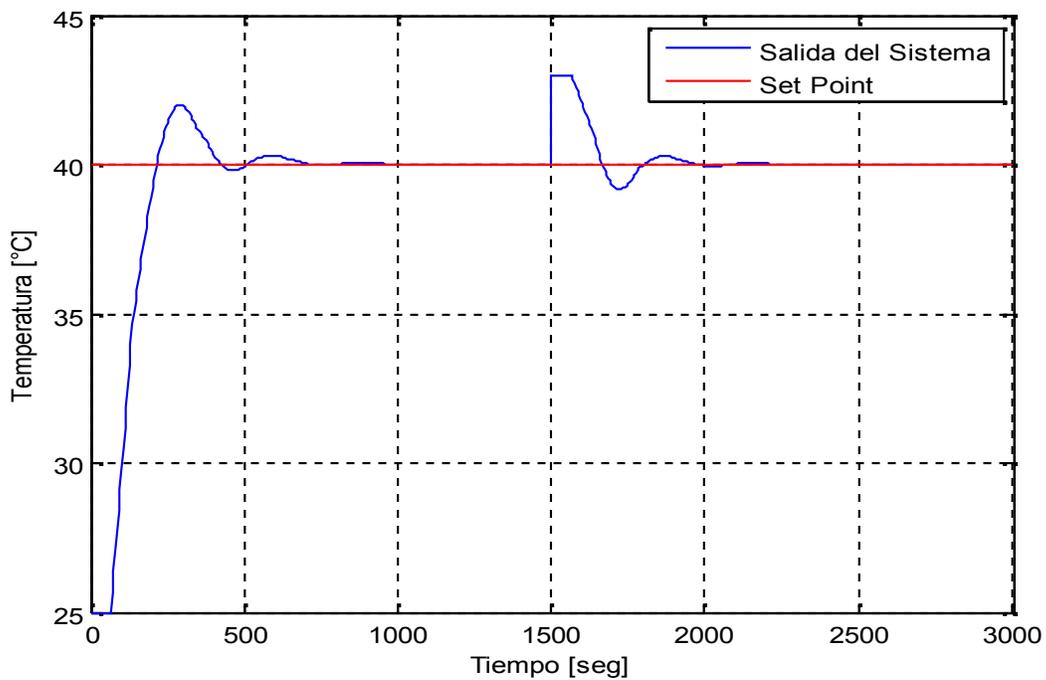
**Figura 2.22** Respuesta de la planta bajo el controlador PD+I difuso a diferentes *set point*.

Para observar la respuesta del controlador ante un cambio inesperado de temperatura de 3°C en el interior del cilindro se realizó el esquema de la figura 2.23,

como muestra la figura 2.24 el controlador responde ante la presencia de un disturbio, aunque presenta oscilaciones alrededor del valor de referencia.



**Figura 2.23** Diseño del controlador PD+I con presencia de disturbio



**Figura 2.24** Respuesta de la planta bajo el controlador PD+I ante un aumento de 3°C al segundo 1500.

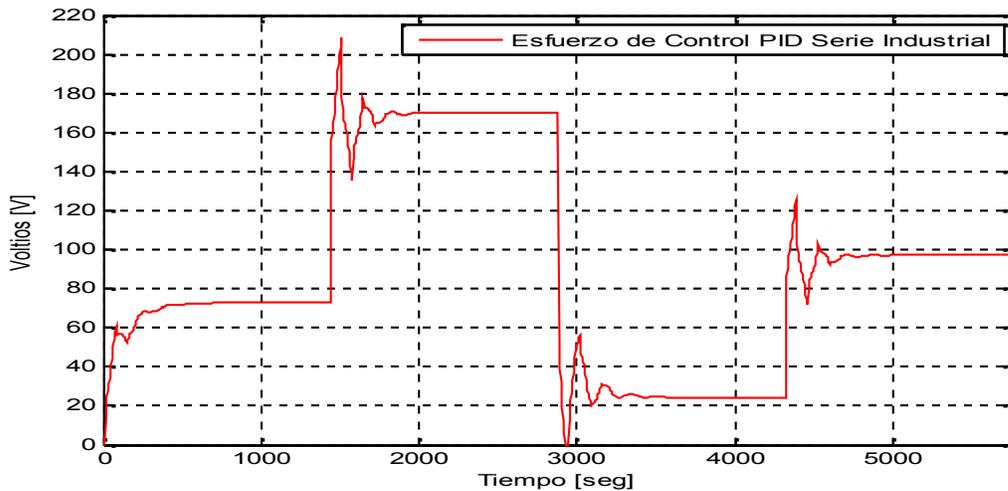
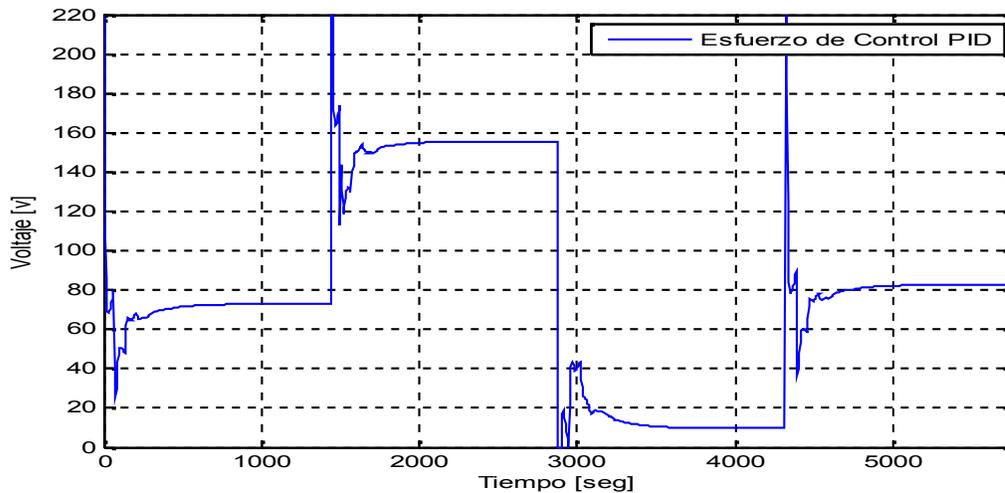
## 2.4 Diseño del Controlador ANFIS.

Como ya se explicó en el capítulo uno, ANFIS permite obtener un sistema de inferencia difusa, a partir de datos de entrada y salida, sin la necesidad del diseño por un experto, por tal motivo se conoce como un aproximador universal, hecho que al momento de realizar un sistema difuso es de gran importancia debido a que no se necesita suficiente conocimiento sobre el sistema, modelo o función de la que se requiere obtener el modelo.

Sin embargo, hay que tener en cuenta que ANFIS modela relaciones algebraicas, por eso es necesario introducirle suficiente dinámica a través de los datos a utilizar para obtener un sistema adecuado. Además, se requiere buscar la mejor combinación de entradas, con el fin de que ANFIS pueda obtener la mayor información posible sobre lo que desea modelar. Generalmente se utilizan los datos de la salida retardada como parte de la entrada del modelo, para que ANFIS pueda determinar si la acción que está realizando es la adecuada.

En este trabajo se utiliza ANFIS, con el propósito de ver su funcionalidad como controlador para la planta de temperatura. Inicialmente, se decidió tomar los datos de entrada y salida del controlador PID con limitación en la derivada existente para las prácticas de laboratorio con el fin de realizar el modelo por ANFIS.

Con el controlador del PID modelado por ANFIS obtenido, el sistema lograba llegar a un *Set Point* deseado, sin embargo cuando se presentaba un disturbio en la planta, el controlador modelado por ANFIS, no actuaba de forma adecuada, permitiendo la propagación del disturbio en el sistema. Al momento de analizar porque no compensa disturbios se observan cambios bruscos y picos muy altos en la señal del esfuerzo de control, lo cual dificulta el modelamiento por ANFIS. Como medida para mejorar el modelado por ANFIS se realizaron otras combinaciones con los datos de entrada sin embargo se siguieron obteniendo los mismos resultados. Por tal motivo se estudia el comportamiento de un PID serie Industrial (Figura 2.8), en el cual la acción derivativa se realiza sobre la señal de salida, haciendo que la señal del esfuerzo de control se suavice notoriamente al compararla con el esfuerzo de control del PID en paralelo (figura 2.25).



**Figura 2.25** Señal de esfuerzo de control del PID paralelo y del PID serie industrial.

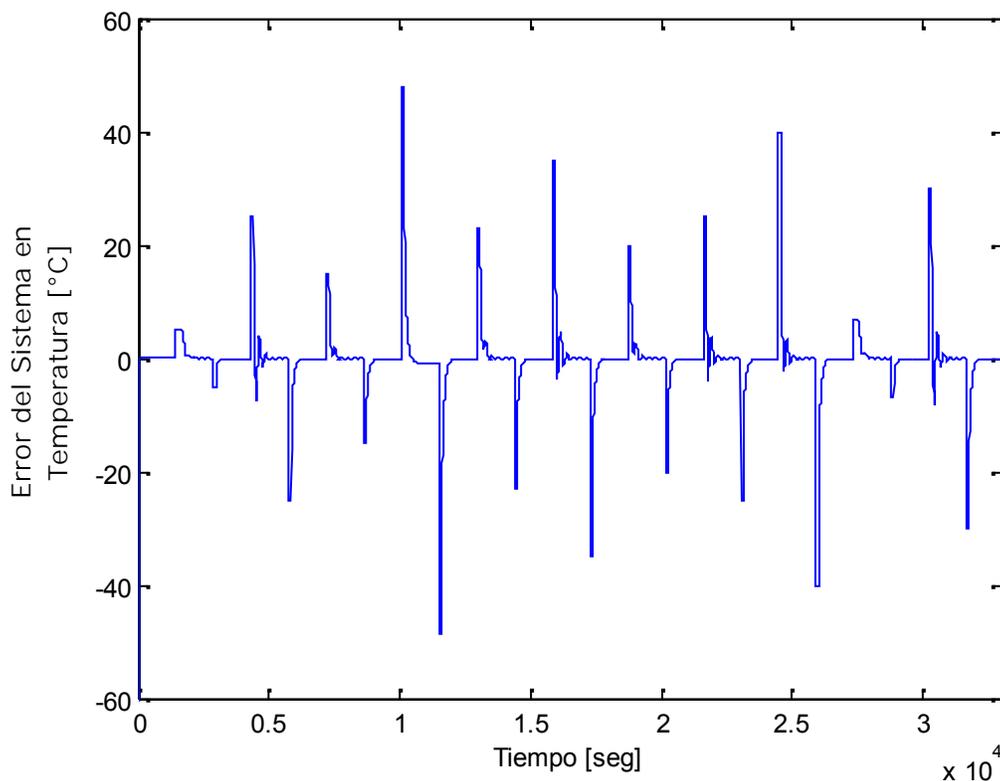
Teniendo en cuenta el esfuerzo de control del PID serie Industrial, se decidió realizar un nuevo modelado por ANFIS, sin embargo al tener la acción derivativa sobre la salida del sistema, no se puede realizar un solo modelo de ANFIS para el controlador, por lo cual se proponen dos esquemas, en el primero se realiza el modelado de la acción derivativa por ANFIS y se verifica su respuesta y en el segundo esquema se modela independientemente el comportamiento de la acción derivativa y la acción integral por ANFIS y de igual forma se analiza su respuesta.

A continuación se describe el diseño de los modelos ANFIS a realizar, se explicará sus diseños por separado y finalmente se probaran en conjunto los modelos ANFIS

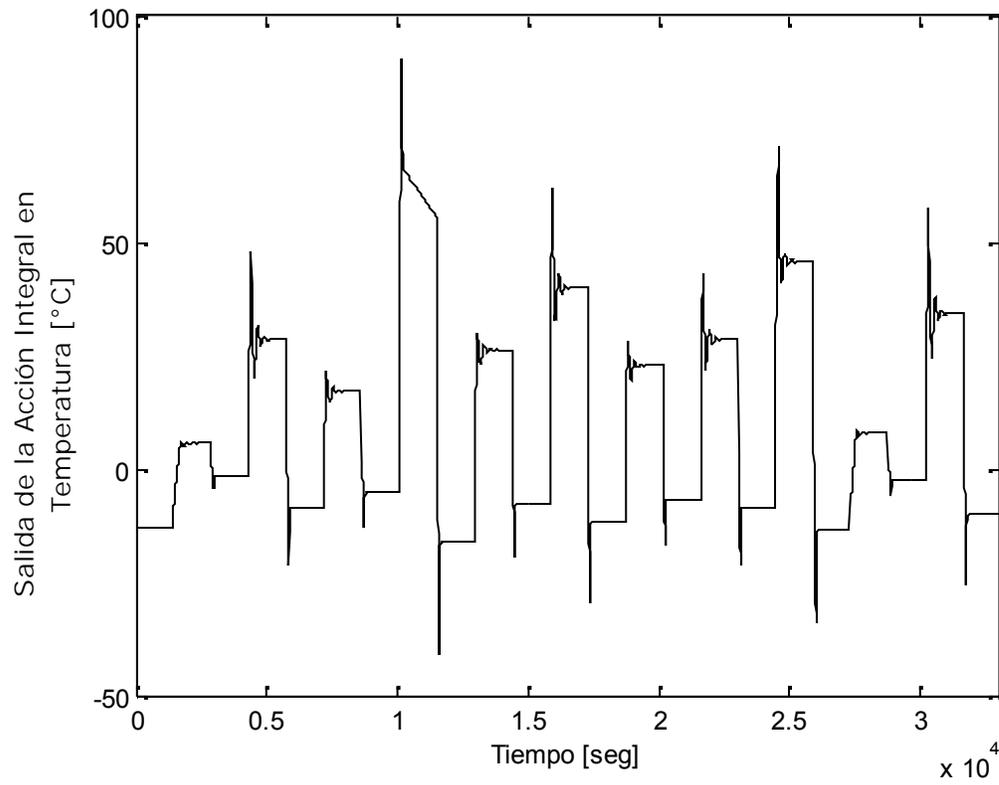
basado en un PID serie industrial, igualmente se analizará su señal de esfuerzo de control, el comportamiento frente a diferentes cambios en el *Set Point* y a la presencia inesperada de un disturbio.

#### 2.4.1 Modelo de la acción integral del PID por ANFIS.

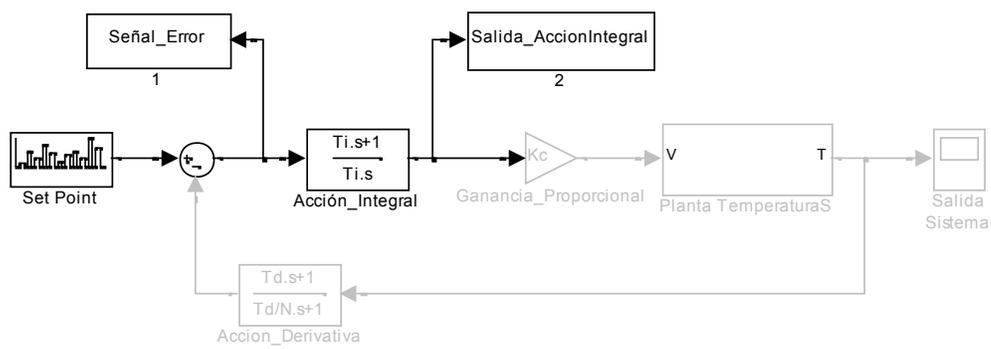
Para obtener el modelo ANFIS de la acción integral, se obtienen los datos de entrada y salida del bloque integrador, mostrados en las figuras 2.26 y 2.27 respectivamente. Aproximadamente se adquirieron 6600 datos tanto de entrada como de salida, muestreados cada 5 seg, para obtener los datos se aplicó al sistema, diferentes señales tipo escalón, que simulan la temperatura de referencia para la planta de temperatura (figura 2.28). Es importante tener en cuenta que las señales tipo escalón deben abarcar todos los posibles cambios de temperatura a los que se requiera llegar, entre ellos la temperatura ambiente, debido a que es la temperatura más baja que se puede obtener.



**Figura 2.26** Datos de la señal de error del sistema.



**Figura 2.27** Datos salida del integrador



**Figura 2.28** Toma de datos acción integral.

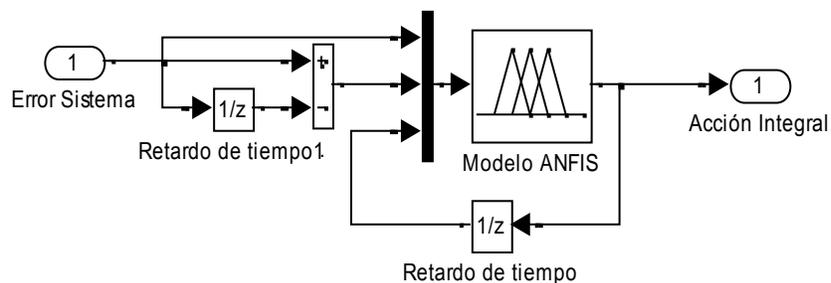
Los datos obtenidos se dividen en 3 conjuntos:

Conjunto de entrenamiento: Los datos de entrenamiento tienen una gran importancia al momento de generar un modelo ANFIS, por tal motivo se utilizan el 50% o más de los datos, “ya que son usados por el algoritmo de aprendizaje para ajustar los parámetros de la red. El objetivo del entrenamiento de la red es minimizar el error de predicción sobre este conjunto de entrenamiento” (Aros & Mardones, 2010). En esta ocasión se utilizaron 70% de los datos para el conjunto de entrenamiento.

Conjunto de testeo o control: Se hizo uso del 15% de los datos para este conjunto. Este grupo de datos no hace parte del entrenamiento, sin embargo tiene como función principal evitar un sobre entrenamiento, debido a que al momento de entrenar este grupo el error de entrenamiento aumenta deteniendo así el entrenamiento del modelo.

Conjunto de Chequeo o Validación: El 15% de los datos restantes son utilizados en este conjunto, en el cual se valida si el entrenamiento de los datos del modelo ha sido el adecuado.

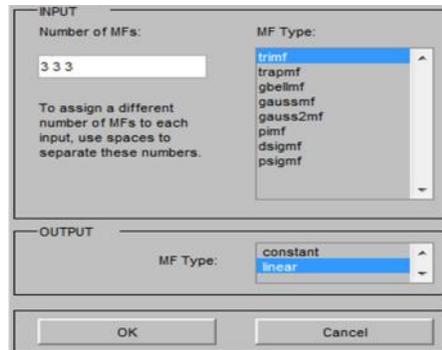
Se realizó un modelo ANFIS con tres entradas y una salida, a partir de los datos del PID. Las tres entradas se obtuvieron probando diferentes configuraciones teniendo en cuenta las características de la planta de temperatura, obteniéndose mejores resultados con la siguiente combinación, *el error del sistema; la razón de cambio del error del sistema y la acción integral con un retardo igual al tiempo de muestreo*, de igual forma como salida la acción integral (figura 2.29).



**Figura 2.29** Esquema modelo ANFIS acción Integral

Matlab cuenta con una GUI para ANFIS en el *toolbox fuzzy logic* donde se cargan los conjuntos de datos organizados en tres matrices, entrenamiento, chequeo y testeo con el fin de entrenar el ANFIS, además se requiere realizar la siguiente configuración

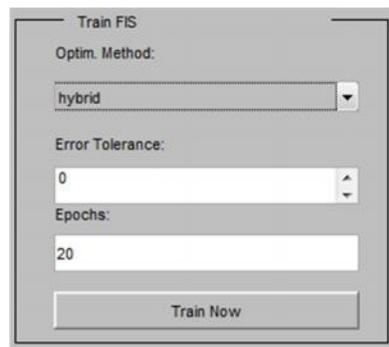
Para la configuración del sistema difuso en el editor de ANFIS se establecieron 3 MFs triangulares por el bajo costo computacional que tiene del tipo Takagi – Sugeno de primer orden (Figura 2.30).



**Figura 2.30** Parámetros del sistema difuso para el modelo ANFIS.

De igual forma existen dos métodos de entrenamiento “Propagación hacia atrás” e “Híbrido”, para obtener un modelo por ANFIS se determinó como “Híbrido” el método de entrenamiento, debido a que utiliza la propagación hacia atrás de los errores para determinar los parámetros correspondientes a los antecedentes, mientras que los consecuentes se estiman utilizando el método de mínimos cuadrados.

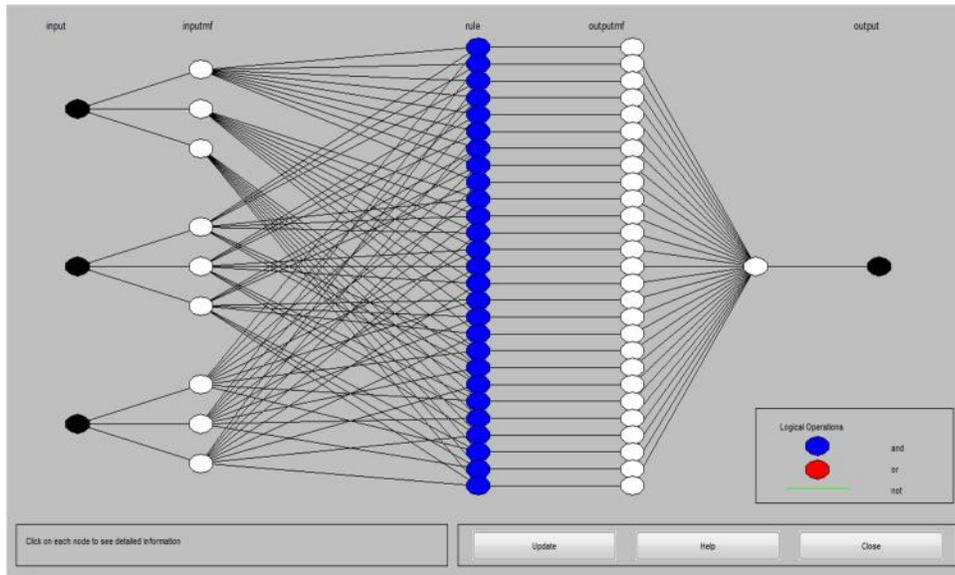
Por otra parte se realizaron 20 iteraciones para el entrenamiento del sistema difuso. La cantidad de interacciones varía dependiendo del sistema a modelar (figura 2.31).



**Figura 2.31** Parámetros del entrenamiento del FIS para el modelo ANFIS.

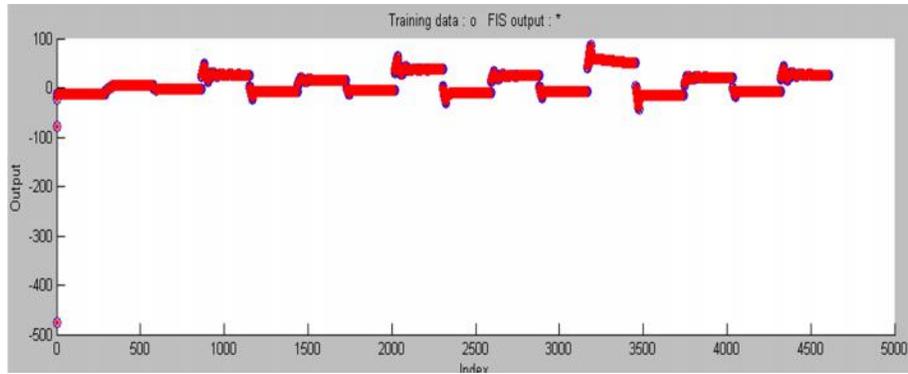
Después de configurar los parámetros del modelo ANFIS, se obtuvo la estructura de la figura 2.32, que consta de 78 nodos, y 27 reglas difusas, donde *input* hace referencia a las tres entradas que se definieron para el modelo ANFIS de la acción integral, los nodos pertenecientes a la primera capa *inputmf* realizan la acción de la **capa 1** de la arquitectura ANFIS, la cual consiste en asignar el valor de pertenencia a las MFs; en los nodos de la capa *rule* se realizan las actividades de las **capas 2** y

3 de ANFIS en las que se forman los parámetros del antecedente de la regla y se define su peso; los nodos de la capa *outputmf* hacen referencia a la función de la **capa 4** donde se definen los parámetros del consecuente de las reglas y finalmente en el nodo *output* se realiza la suma ponderada de las salidas de la regla para obtener la salida del sistema que equivale a la función de la **capa 5**.



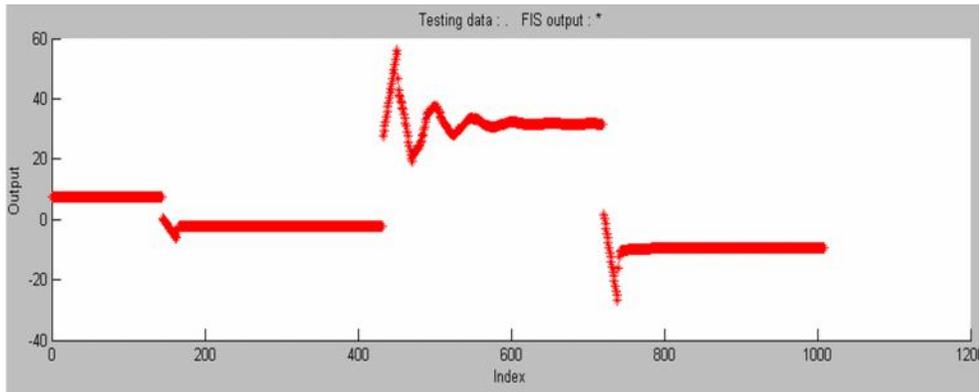
**Figura 2.32** Estructura del modelo ANFIS para la acción integral.

En la figura 2.33, se muestra el comportamiento de los datos de salida del modelo ANFIS (rojo) frente a los datos de salida del conjunto de entrenamiento (azul), previamente cargados en el editor de ANFIS, es decir por las tres entradas el editor de GUI gráfica su respectivo valor de salida del conjunto de entrenamiento y el valor de salida del modelo ANFIS, que al compararlos se puede evidenciar que los datos rojos son muy similares a los datos azules, teniendo en cuenta que el conjunto de entrenamiento es de gran relevancia al momento de modelar por ANFIS, con lo que se puede decir que el modelo aproximado a partir de los datos de entrenamiento es bueno y puede ser implementado como la acción integral del PID.

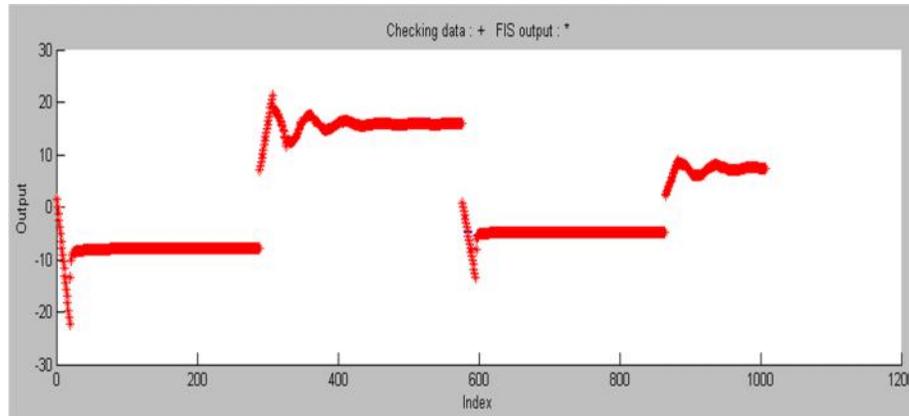


**Figura 2.33** Resultado del modelo ANFIS, datos de entrenamiento.

Por otra parte en la figura 2.34, se muestra la comparación entre los datos de salida del modelo ANFIS (rojo) y el conjunto de testeo (azul) y la figura 2.35, la comparación entre los datos de salida del modelo ANFIS y el conjunto de chequeo, se comprueba una vez más que el modelo obtenido por ANFIS a partir de los datos ingresados al sistema puede ser considerado como bueno al no observarse ningún dato azul, lo que quiere decir que los datos de salida del modelo ANFIS, con respecto a los datos de chequeo y testeo respectivamente son muy similares, sin embargo sólo con la gráfica que muestra el editor no se puede definir qué tan cerca están los datos rojos de los datos azules, porque no se puede obtener una vista más detallada de los datos.



**Figura 2.34** Resultado del modelo ANFIS, datos de testeo.



**Figura 2.35** Resultado del modelo ANFIS, datos de chequeo

Sin embargo, aunque el editor de ANFIS en Matlab muestra un error, de que tan dispersos están los datos del modelo obtenido por ANFIS con respecto a los conjuntos de entrenamiento se decide validar el ANFIS utilizando los métodos estadísticos  $RMS$  (2.14)(Error de mínimos cuadrados), el cual evalúa la diferencia entre un estimador y el valor real que se quiere calcular (Skoog, Crouch, & Holler, 2008).y el  $R^2$  (2.15)(coeficiente de correlación múltiple) que es una medida descriptiva que estima el ajuste del modelo a los datos, ya que mide la capacidad predictiva del modelo ajustado (Sabadía, 1995),

$$RMS = \sqrt{\frac{\sum_{n-1}^n (y_{pre,m} - t_{mea,n})^2}{n}} \quad (2.14)$$

$$RMS = 1,1492$$

$$R^2 = 1 - \frac{\sum_{n-1}^n (y_{pre,m} - t_{mea,n})^2}{\sum_{m-1}^n (t_{mea,n})^2} \quad (2.15)$$

$$R^2 = 0,9987$$

Al tener un  $RMS$  de 1,1492, se puede concluir que el error de los datos estimados comparados con los datos de referencia es bastante pequeño, considerando que el  $RMS$  ideal es de 0 y un  $R^2$  de 99.87%, es un ajuste óptimo, si se tiene en cuenta que el ajuste ideal es del 100%. Por lo tanto, el modelo ANFIS obtenido se puede calificar como un buen estimador de los datos de la acción integral.

## 2.4.2 Modelo de la acción derivativa del PID por ANFIS

El procedimiento para hallar el modelo ANFIS de la acción derivativa del PID serie industrial tiene una gran similitud al realizado en la sección 2.4.1 para la acción integral, ya que cuenta con la misma cantidad de datos tomados, pero en este caso, a la entrada y salida de la acción derivativa (figura 2.36), de igual forma, se dividen en tres grupos, entrenamiento, testeo y chequeo, los cuales son adquiridos al aplicar al sistema diferentes señales tipo escalón que simulan la temperatura de referencia. Los datos obtenidos de la temperatura de salida del sistema se muestran en la figura 2.37 y los datos de la salida de la acción derivativa se encuentran en la figura 2.38

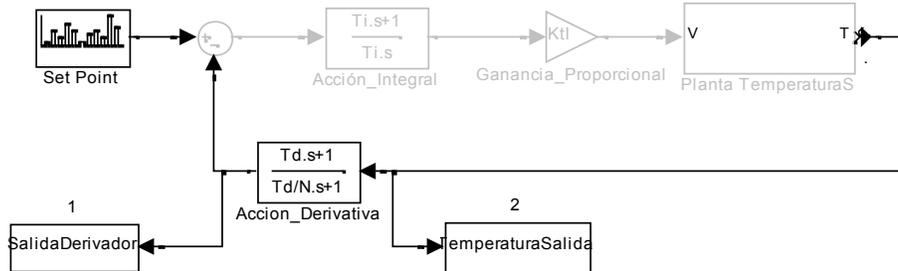


Figura 2.36 Toma de datos acción derivativa

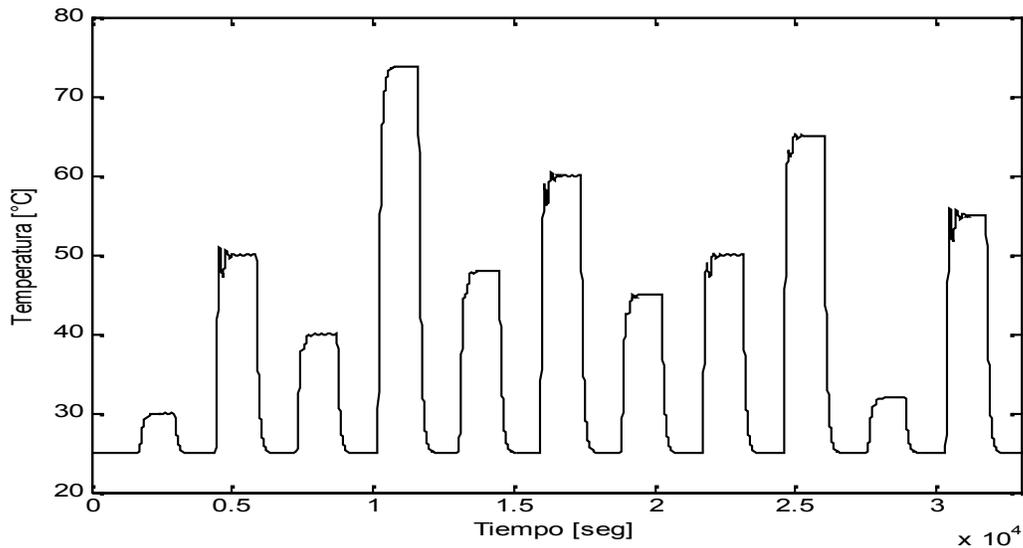
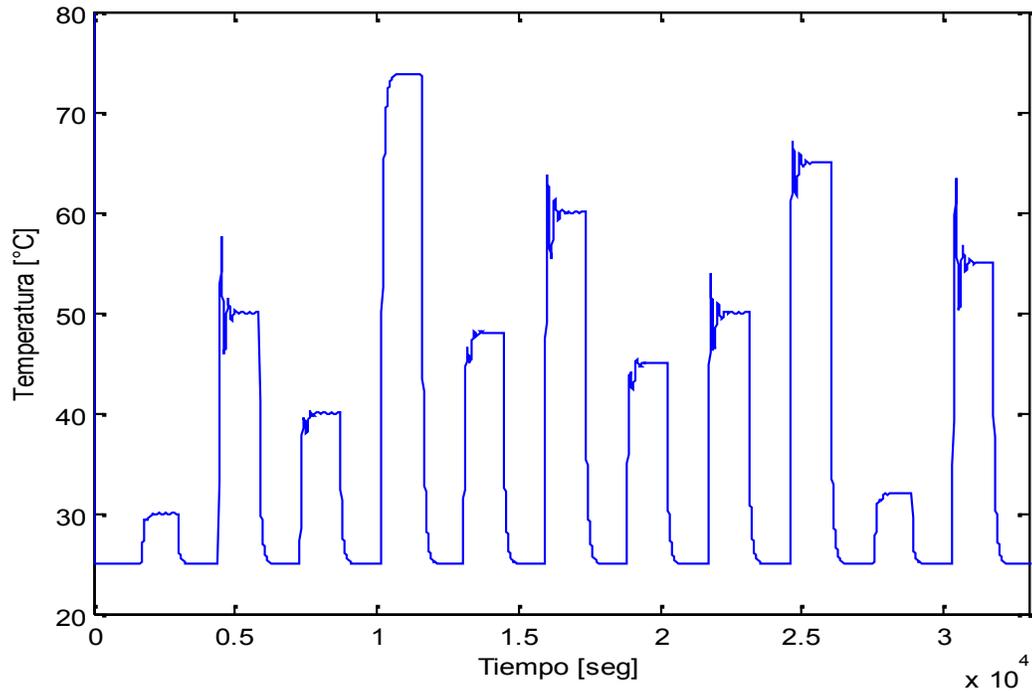
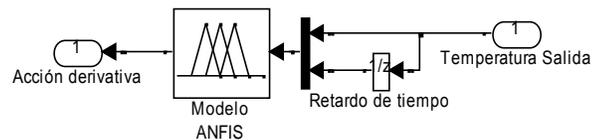


Figura 2.37 Datos temperatura salida



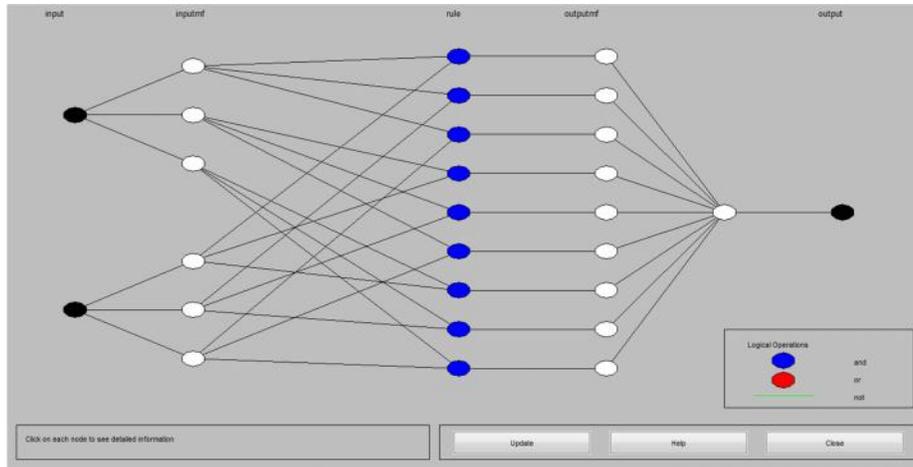
**Figura 2.38** Datos salida derivador.

Para realizar el modelo ANFIS de la acción derivativa se determinaron *la temperatura de salida del sistema y la temperatura de salida con un retardo del tiempo de muestreo* como entradas y la acción derivativa como salida (figura 2.39), después de probar diferentes configuraciones teniendo en cuenta las características de la planta de temperatura.



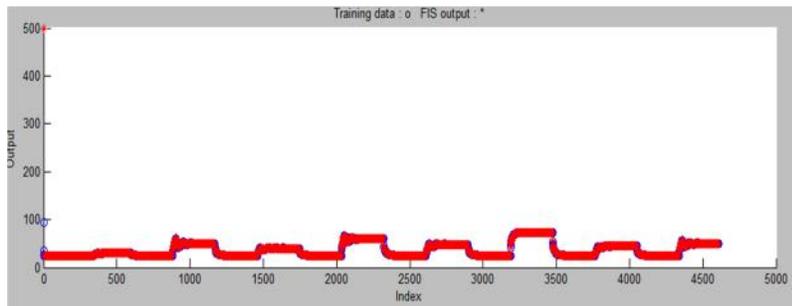
**Figura 2.39** Esquema modelo ANFIS acción derivativa.

La configuración en el ANFISedit de Matlab se realiza de igual forma que para el modelo de la acción integral, dando como resultado la estructura de la figura 2.40, que cuenta con 35 nodos, y 9 reglas difusas.



**Figura 2.40** Estructura del Modelo ANFIS para la acción integral.

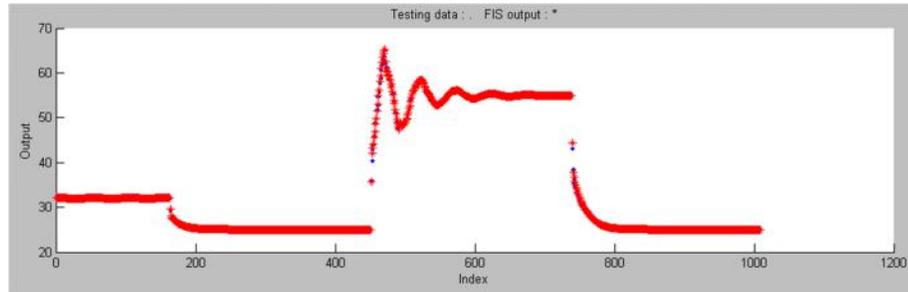
De la misma forma como se explicó para el modelado de la acción integral, en la figura 2.41 se muestra la distancia entre los datos de salida del modelo ANFIS (rojo) y los datos de salida del conjunto de entrenamiento (azul), al compararlos se puede evidenciar que los datos rojos presentan una gran similitud a los datos azules, teniendo en cuenta que el conjunto de entrenamiento es de gran relevancia al momento de modelar por ANFIS, permitiendo decir que el modelo aproximado a partir de los datos de entrenamiento es bueno y puede ser implementado como la acción derivativa del PID



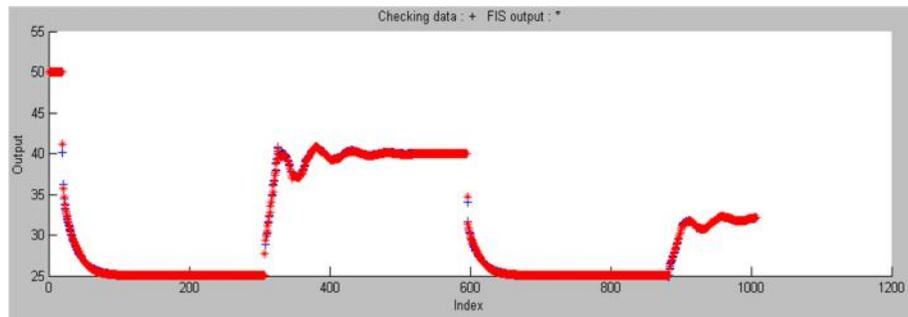
**Figura 2.41** Resultado del modelo ANFIS, datos de entrenamiento

Además en la figura 2.42 se muestra la dispersión de los datos de salida del modelo ANFIS (rojo) y los datos de salida del conjunto de testeo (azul) y en la figura 2.43, la distancia entre los datos de salida del modelo ANFIS y los datos de salida del conjunto de chequeo, verificando una vez más que el modelo obtenido por ANFIS a partir de los datos ingresados al sistema puede ser considerado como bueno al no observarse ningún dato azul, lo que indica que los datos de salida del modelo

ANFIS, con los datos de chequeo y los datos de testeo respectivamente, son muy similares, sin embargo igual que con la acción integral no se puede concluir que tan cerca están, con solo observar la gráfica.



**Figura 2.42** Resultado del modelo ANFIS, datos de testeo



**Figura 2.43** Resultado del modelo ANFIS, datos de chequeo

Para validar el ajuste del modelo ANFIS de la acción derivativa, con los datos ingresados al editor de ANFIS, se utilizaron los métodos estadísticos  $RMS$  (2.14) y  $R^2$  (2.15), obteniendo los siguientes resultados.

$$RMS \approx 1,0043$$

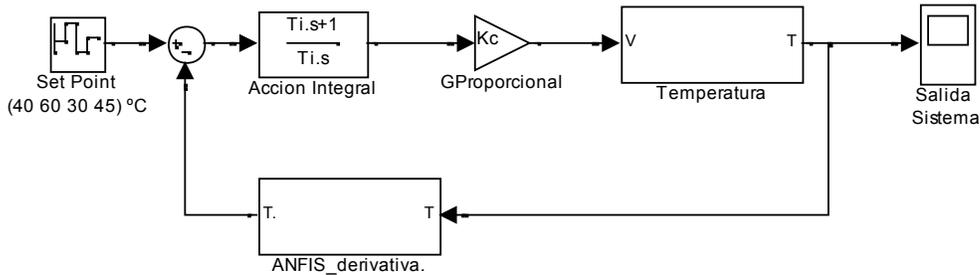
$$R^2 = 0,9993$$

Al igual que en la validación del modelo de la acción Integral, se obtuvieron, valores pequeños de  $RMS$  y  $R^2$ , lo que certifica que el modelo ANFIS si realiza buena aproximación del comportamiento de la acción derivativa.

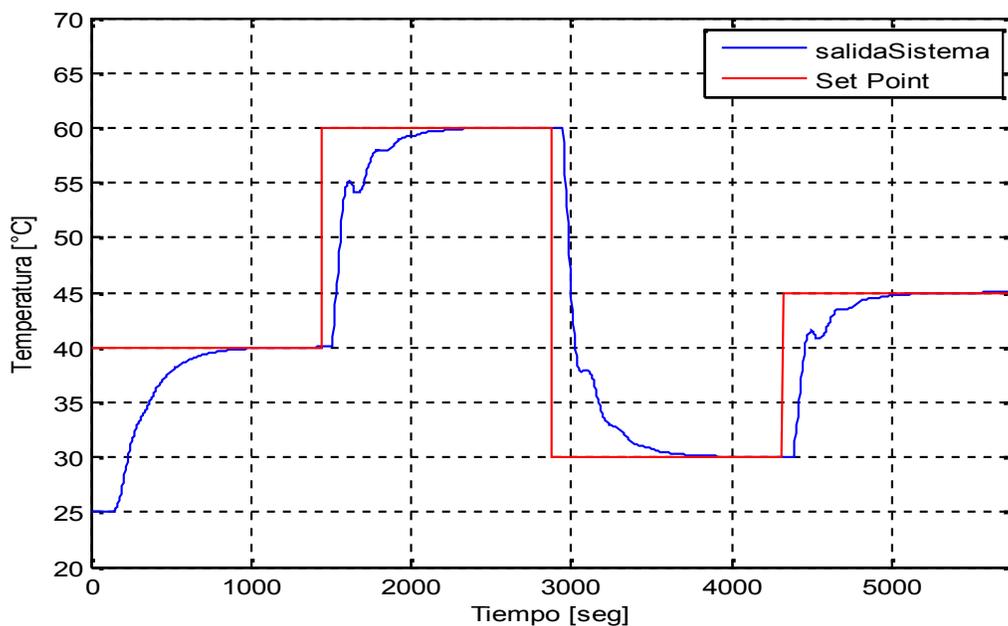
### 2.4.3 Esquema del Controlador basado en ANFIS para la planta de temperatura.

### 2.4.3.1 Esquema del Controlador basado en ANFIS modelando la acción derivativa.

Al obtener el modelo por ANFIS de la acción derivativa, se prueba su funcionamiento en la estructura del PID Serie Industrial (Figura 2.44), para observar el comportamiento del sistema con el controlador (Figura 2.45).

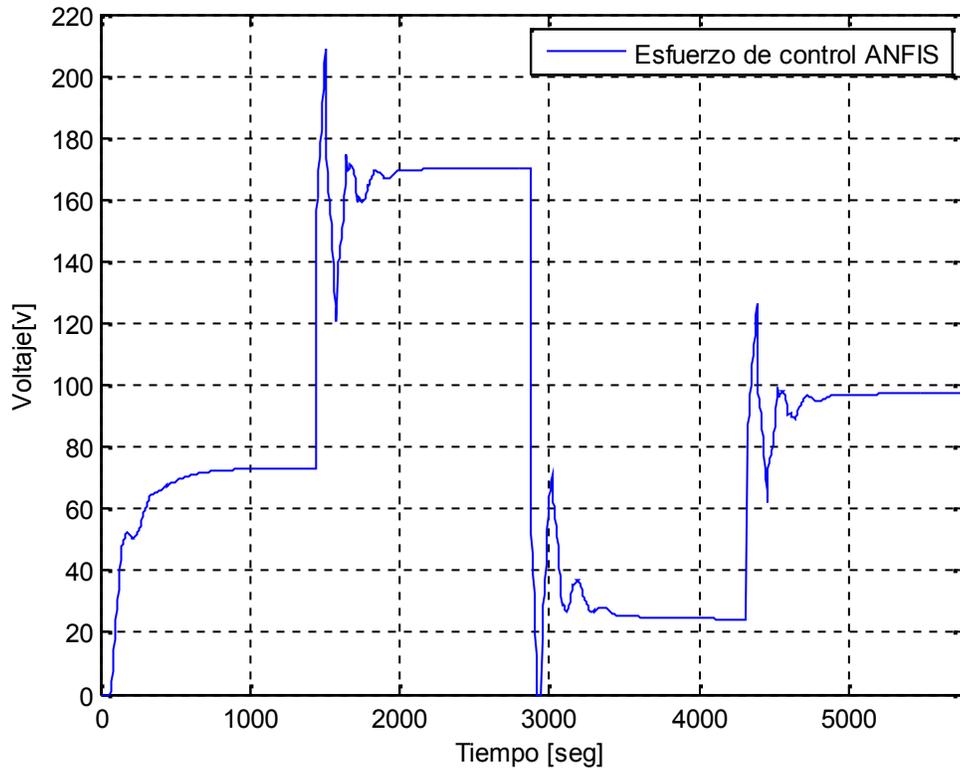


**Figura 2.44** Diagrama de bloques del sistema modelando por ANFIS la acción derivativa.



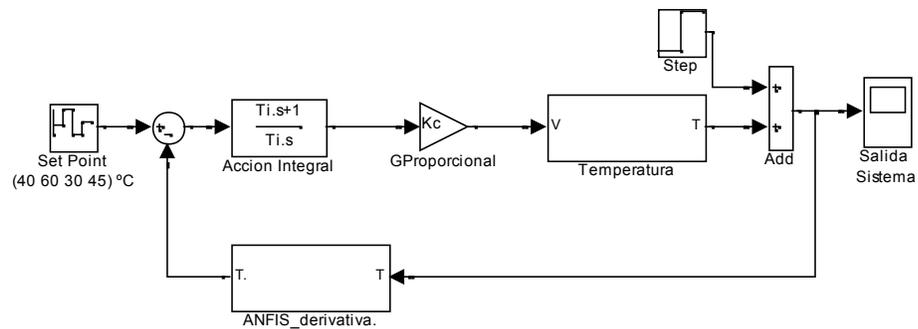
**Figura 2.45** Respuesta del sistema modelando la acción derivativa por ANFIS.

En la figura 2.46 se grafica la señal del esfuerzo de control, en la que se evidencia que la señal no sale del valor máximo de voltaje, además que al presentar picos altos al iniciar cada cambio de la señal de referencia permite que el sistema se acerca con más rapidez al valor deseado.

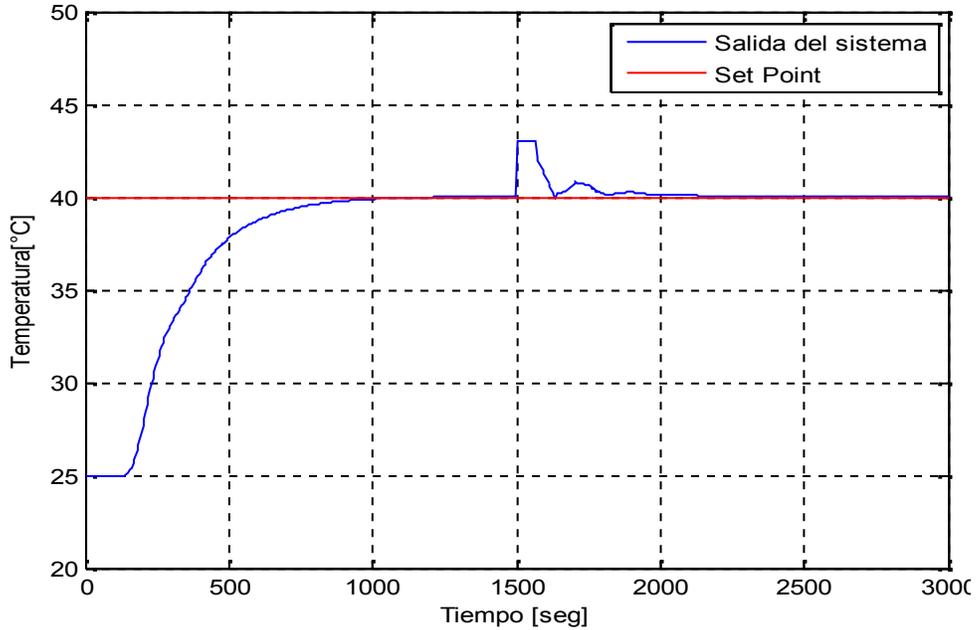


**Figura 2.46** Esfuerzo de controlador modelando la acción derivativa por ANFIS

Se verificó la respuesta del controlador ante un cambio inesperado en la temperatura, utilizando como esquema el mostrado en la figura 2.47, dando como resultado la figura 2.48.



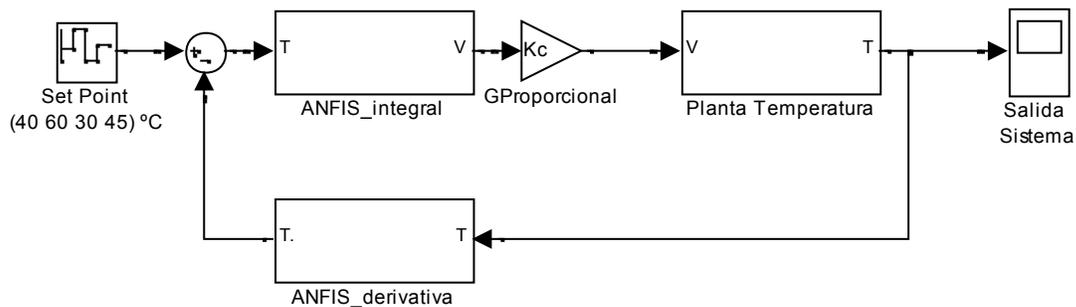
**Figura 2.47** Esquema del controlador con el modelado de la acción derivativa por ANFIS



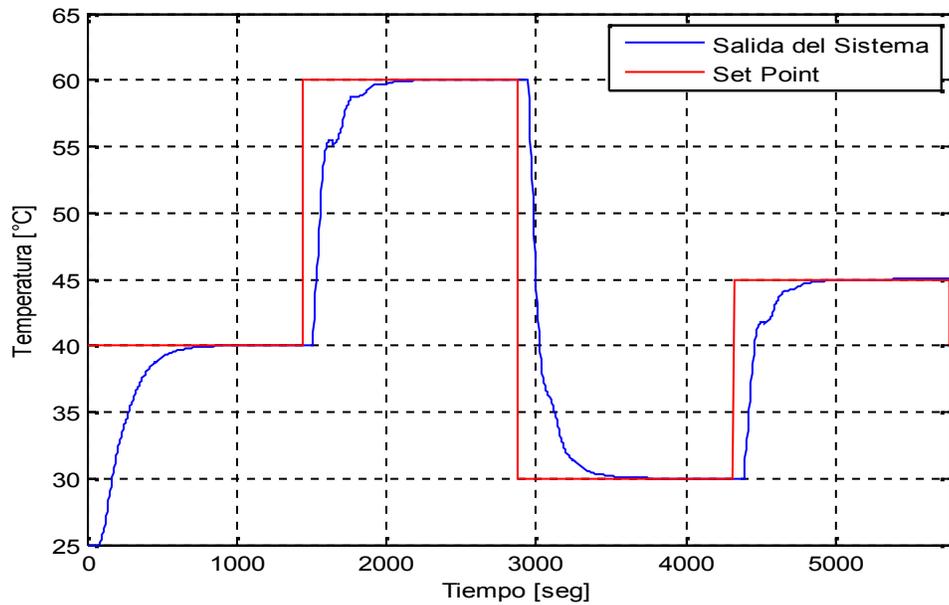
**Figura 2.48** Respuesta del sistema al controlador basado en ANFIS para la planta de temperatura.

### 2.4.3.2 Esquema del Controlador basado en ANFIS modelando la acción derivativa e integral.

Después de obtener el modelo por ANFIS de la acción integral y derivativa, se prueba su funcionamiento en conjunto (Figura 2.49), con el fin de observar, el comportamiento que tiene el sistema con el controlador basado en ANFIS (Figura 2.50).

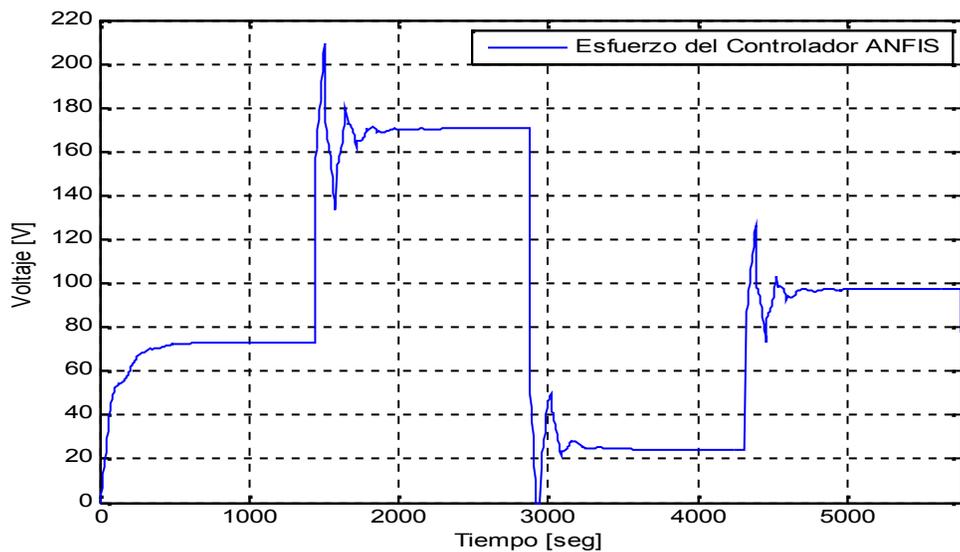


**Figura 2.49** Diagrama de bloques de los modelos de la acción integral y derivativa por ANFIS



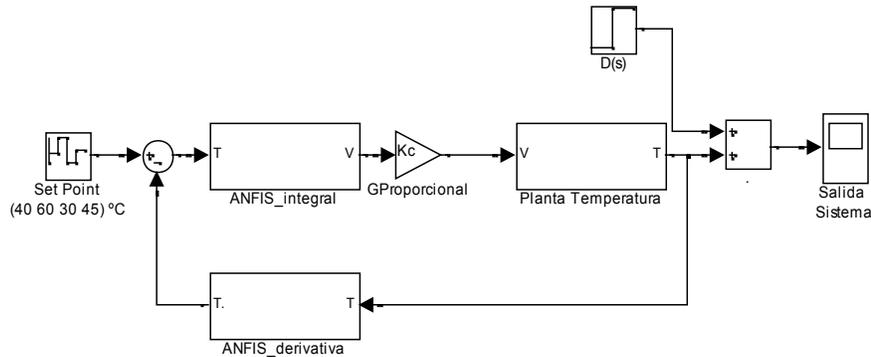
**Figura 2.50** Respuesta del sistema bajo el controlador basado en ANFIS para la planta de temperatura.

En la figura 2.51 se grafica la señal del esfuerzo de control, en la que se evidencia que la señal no sale del valor máximo de voltaje establecido para la planta, además de que al presentar picos tan altos permite que el sistema se acerca con más rapidez al valor deseado.

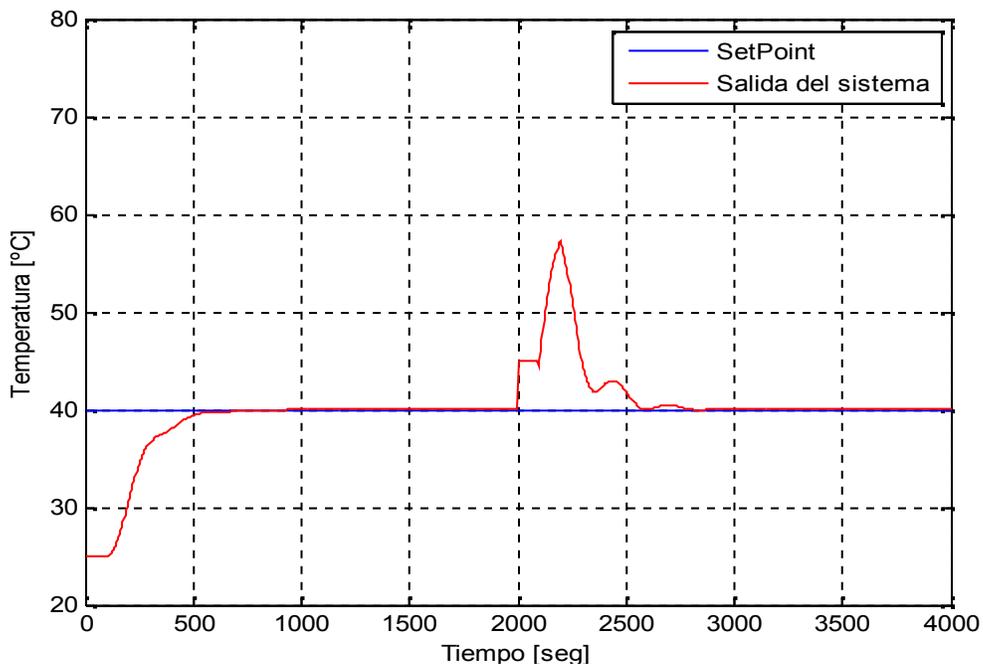


**Figura 2.51** Esfuerzo de control del controlador modelado por ANFIS.

Igualmente se verificó la respuesta del controlador ante un cambio inesperado en la temperatura, utilizando como esquema el mostrado en la figura 2.52, dando como resultado la figura 2.53, en la que se evidencia el buen comportamiento del controlador ante un disturbio.



**Figura 2.52** Esquema controlador basado en ANFIS para la planta de temperatura.



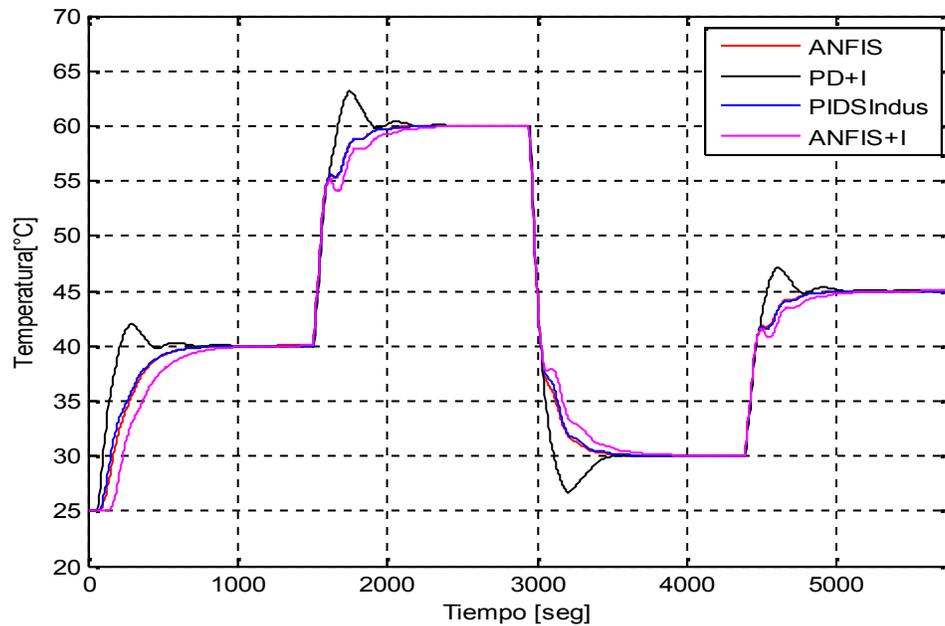
**Figura 2.53** Respuesta del sistema al controlador basado en ANFIS para la planta de temperatura.

## 2.5 Resultados del Diseño de los controladores.

Cada controlador tiene un efecto propio sobre la planta, en la figura 2.49, se ilustra la respuesta del sistema ante cambios en el *Set Point* definidos en 40, 60, 30 y 45°C, bajo la implementación de cada uno de los controladores previamente diseñados.

Se puede notar que la salida del sistema, con el controlador PD+I difuso presenta sobre impulsos y es lenta con respecto a las otras, aunque es necesario tener en cuenta que éste parte, de un controlador PID paralelo como se explicó en la sección 2.3, donde fue sintonizado por la técnica de Z y N, la cual se caracteriza por presentar grandes sobre impulsos, sin embargo como es general en los PID se puede obtener diferentes resultados según sea la técnica de sintonización utilizada.

Aunque es natural que la respuesta que presenta el sistema con el PID serie industrial, sea muy similar a la obtenida con los controladores modelados por ANFIS, dado que son formados a partir del primero, el controlador que tiene modelado por ANFIS solo la acción derivativa, presenta un mayor tiempo de estabilización al compararlo con la respuesta del sistema ante el PID Serie Industrial y el compuesto por los dos modelos ANFIS, por otra parte es importante resaltar que el sistema de control en el que se modeló la acción derivativa e integral sobresale por tener un menor tiempo de estabilización, y menor cantidad de oscilaciones alrededor de la referencia.



**Figura 2.54** Respuesta controlador basado en ANFIS para la planta de temperatura.

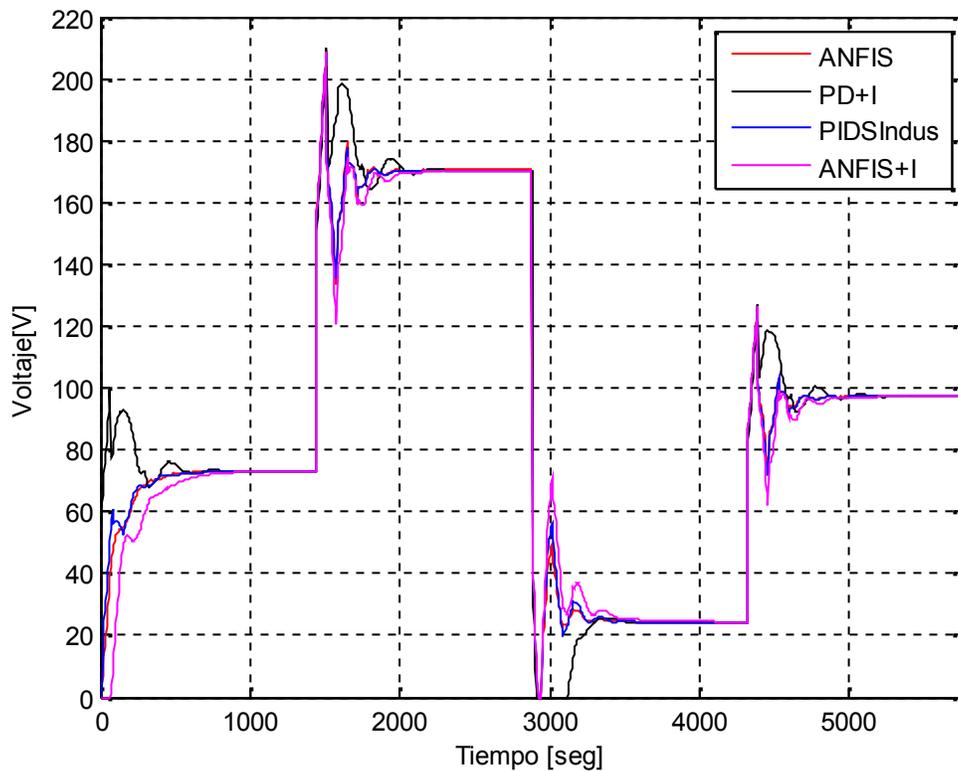
Con el fin de tener una mejor apreciación de lo anteriormente dicho, se construyó la tabla 2.2 en la cual se presenta una comparación entre los cuatro controladores, basada en 4 ítems básicos, que son el tiempo de estabilización del sistema, la presencia de sobre impulsos en la señal de salida, la compensación de disturbios externos y por último el tiempo que demora el controlador en corregir el disturbio.

Se evidencia que la respuesta menos satisfactoria es la presentada por el sistema bajo un controlador PD+I Difuso, al tener sobre impulso y tarda más tiempo que los otros controladores en compensar disturbios, igualmente se puede concluir que el PID serie Industrial y el modelado por ANFIS poseen características favorables al momento de analizar los resultados de los parámetros de la tabla 2.2.

Tabla 2.2 Comparación entre los diferentes controladores diseñados en Matlab Simulink

Controlador	Tiempo aprox Estabilización (s)	Máximo Sobre Impulso (°C)	Compensación a Disturbios	Tiempo aprox. de Compensación (s)
PID Serie Industrial	800	No presenta	Si	700
PD+I Difuso	900	No mayor a 2	Si	1000
Modelado ANFIS	900	No presenta	Si	500
Modelado ANFIS+I	1200	No presenta	si	700

Teniendo en cuenta que la salida del sistema depende de la señal de control proporcionada por los controladores, se realizó un análisis comparativo entre dichas señales para cada uno de los 4 controladores diseñados, en la figura 2.55 se ilustran las cuatro señales de control, para cambios en la referencia de 40, 60, 30 y 45 °C, en la gráfica se puede notar que el esfuerzo de control generado por el controlador ANFIS (Modelado por ANFIS acción derivativa e integral), va delante y posee unos picos más pronunciados, lo que genera que dicho controlador sea el más rápido de los cuatro como ya se había dicho, adicionándole una ventaja al momento de compensar disturbios de igual manera se observa que la señal generada por el PD+I difuso, presenta cambios bruscos, los cuales pueden afectar al actuador de la planta.



**Figura 2.55** Comparación entre los esfuerzos del control aplicado a la planta.

### 3. IMPLEMENTACIÓN DE LOS CONTROLADORES EN PLANTA.

Este capítulo presenta la descripción de los aspectos básicos para la implementación de los controladores diseñados en el capítulo anterior, entre los que se encuentran la modificación del cableado de la planta para la adecuación del nuevo escenario de control, la descripción del paquete de soporte para Arduino de Simulink y los esquemas en Simulink necesarios para la implementación de los controladores en la planta de temperatura del LCP.

#### 3.1. Aspectos básicos para la implementación.

La sección 3.1 trata los aspectos más generales para la implementación de un controlador sobre la planta de temperatura del LCP, se realiza la descripción de la tarjeta Arduino Mega 2560 R3, hardware escogido para implementar el controlador; se muestra el cableado necesario para el nuevo escenario de control, y se realiza una descripción del paquete perteneciente a Simulink, utilizado como interfaz de programación para la tarjeta Arduino.

##### 3.1.1. Tarjeta Arduino Mega 2560 R3.

La tarjeta Arduino Mega 2560 ilustrada en la figura 3.1, es una plataforma de código abierto fundamentada en prototipos de electrónica flexible, basada en un micro controlador ATmega 2560, posee 54 pines de E/S digitales, de los cuales 14 pueden ser usados como salidas de PWM, 16 entradas analógicas, 4 puertos para comunicación seriales por hardware, un cristal oscilador de 16 MHz y una conexión USB, entre otras (Arduino, Sitio web Arduino).

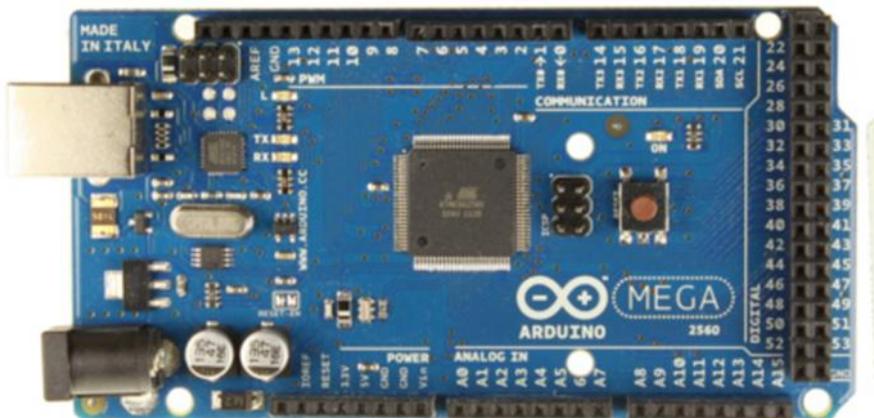


Figura 3.1 Tarjeta Arduino Mega 2560

Las características técnicas de la tarjeta se encuentran resumidas en la tabla 3.1.

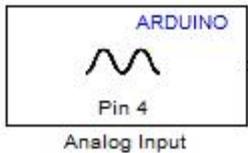
**Tabla 3.1** Características Técnicas de la Tarjeta Arduino Mega 2560

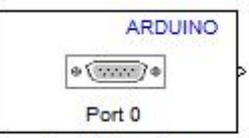
Características técnicas de la tarjeta Arduino Mega 2560 R3	
<b>Micro controlador</b>	ATmega 2560
<b>Voltaje de operación</b>	5 V
<b>Intervalo recomendado para el voltaje de entrada</b>	7-12 V
<b>Límites del voltaje de entrada</b>	6-20 V
<b>Pines de E/S digitales</b>	54 (de los cuales 14 pueden ser usadas como salida de PWM)
<b>Pines de entradas analógicas</b>	16
<b>Corriente DC por los pines E/S</b>	40 mA
<b>Corriente DC por el pin de 3.3V</b>	50 mA
<b>Memoria flash</b>	256 KB de los cuales usa 8 KB para el gestor de arranque (Bootloader)
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Velocidad de reloj</b>	16 MHz

### 3.1.2. Simulink Support Package for Arduino Hardware

En la versión 2012.a de Matlab, Simulink incorporó un paquete de soporte que permite la creación de prototipos y modelos que se pueden ejecutar, ya sea en una tarjeta Arduino UNO o una tarjeta Arduino Mega 2560. Básicamente le permite al usuario diseñar en la interfaz de Simulink, algoritmos de control, aplicaciones en robótica, entre otras, para posteriormente implementarlas en dichas tarjetas, sin la utilización de programación manual. En la tabla 3.2 se observa una breve descripción de cada uno de los elementos que componen el paquete.

**Tabla 3.2** Bloques que conforman el paquete de soporte para Arduino de Simulink

Bloque	Gráfica	Descripción
<b>Entrada Analógica</b>		Mide el valor de voltaje en un pin deseado, la resolución de dichos pines es de 10 bits, es decir que su salida es un valor entre 0 y 1023.
<b>PWM</b>		Cambia el ciclo de trabajo de una onda cuadrada para la salida de un pin digital, la resolución de dichos pines es de 8 bits, por lo que el valor que llega a este bloque debe estar entre 0 y 255.

<b>Entrada Digital</b>		<p>Lee el valor digital que entrega cualquiera de los pines de entrada digital de la tarjeta.</p>
<b>Salida Digital</b>		<p>Establece un valor lógico en alguno de los pines de salida digital de la tarjeta.</p>
<b>Recepción Serial</b>		<p>Recibe un byte de datos por un período de muestreo del búfer de recepción del puerto serie especificado.</p>
<b>Transmisión Serial</b>		<p>Envía datos almacenados en el búfer del puerto serie especificado</p>
<b>Lector Servo Estándar</b>		<p>Mide el ángulo de un eje del servo motor estándar en grados, de 0 a 180.</p>
<b>Escritor Servo Estándar</b>		<p>Establece la posición del eje de un motor servo estándar, de 0 a 180 grados, para girar el eje del servo, enviar valores de 0 a 180 a la entrada del bloque.</p> <p>Envío de fuera de alcance valores, tales como -5 o 200, a la entrada del bloque tiene el mismo efecto que el envío de los valores de entrada máximos o mínimos.</p>
<b>Escritor Servo Continuo</b>		<p>Establece la dirección y la velocidad de rotación de un motor servo continuo, envío de -90 a la entrada del bloque produce la tasa máxima de rotación en una dirección, envío de 90 a la entrada del bloque produce la tasa máxima de rotación en la dirección opuesta, Envío de 0 a la entrada del bloque para el motor servo.</p>

Envío de fuera de alcance valores, tales como -95 o 200, a la entrada del bloque tiene el mismo efecto que el envío de los valores de entrada máximos o mínimos.

Las características de algunos motores provocar que continúe girando cuando el valor del bloque de entrada es 0. En ese caso, usted puede experimentar hasta encontrar un valor de desplazamiento que detiene el motor.

### **3.2. Implementación de los Controladores en la Planta de Control de Temperatura del LCP.**

Para la implementación de los controladores en planta, se utilizó una tarjeta Arduino mega 2560 como solución hardware; Matlab-Simulink como programador de la tarjeta y el software *LabVIEW* para la interfaz hombre máquina, además se realiza la descripción del cableado necesario para el nuevo escenario de automatización en la planta de temperatura, basado en la tarjeta Arduino Mega 2560 R3, así como una descripción de la implementación de los controladores ya diseñados.

#### **3.2.1. Descripción del cableado de la planta para la adecuación del nuevo escenario de control.**

Inicialmente se adecua el cableado del panel de control de la planta de temperatura, con el fin de obtener un nuevo escenario de automatización, que no afecte el ya implementado, además se etiqueta nuevamente la planta de acuerdo a las recomendaciones de S.5 de ISA y se reubican algunos dispositivos con el fin de adecuar en el panel de control los dispositivos necesarios para el escenario de automatización (ver figura 3.2).

En la figura 3.3 se ilustra el diagrama de lazo de las conexiones realizadas para el nuevo escenario de control. La llave selectora del escenario de automatización, identificada en el panel de control como *HS2*, polariza el relé *HY1*, con el fin de cambiar el rumbo de la señal de PWM, que llega al relé de potencia *HY4*; como el de la señal que proviene del sensor cercano, por medio del transmisor indicador de temperatura *TIT2*.

Por otra parte se tiene en cuenta que la señal procedente de los transmisores es entregada en valores de 4 a 20 mA, y que las entradas analógicas de la tarjeta reciben un rango de voltaje que va de 0 a 5 voltios, por lo tanto se realiza un circuito de adecuación, que contiene una resistencia de 250  $\Omega$ , la cual convierte la señal de corriente a voltaje entre 1 y 5 voltios.

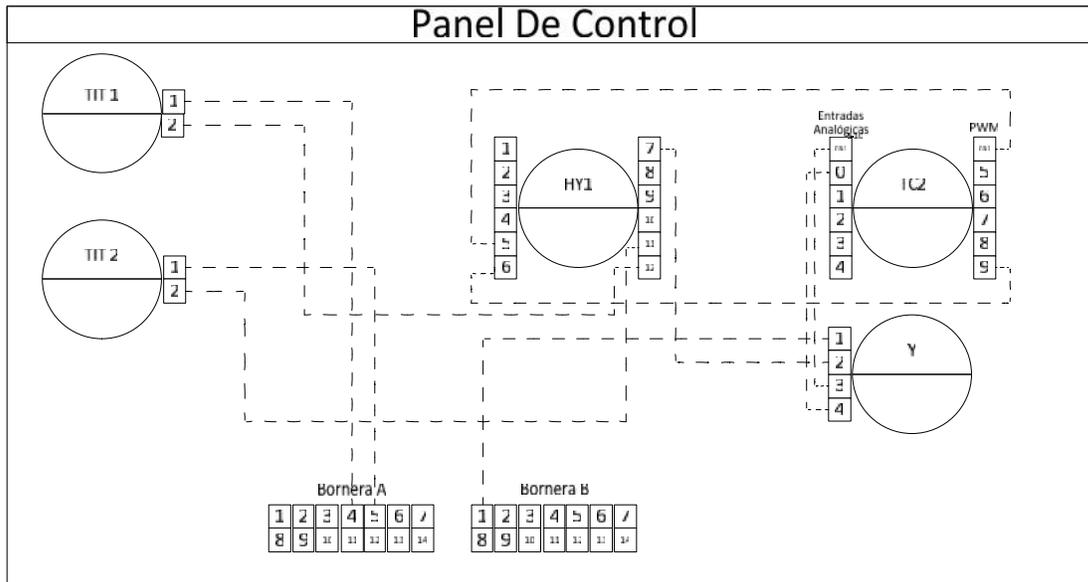
En la tarjeta Arduino se escogió el pin 9 como salida de PWM y el pin A0 como entrada analógica, en la tabla 3.3 se puede ver la lista de tags incluida en el nuevo escenario de control, y en la figura 3.3 se muestra la imagen del panel de control, luego de la implementación del nuevo escenario de control.

**Tabla 3.3** Listado de Tags para el nuevo escenario de Control

Tag	Descripción
TIT1	Transmisor indicador de temperatura 1
TIT1	Transmisor indicador de temperatura 2
HY1	Relé manual 1
TC2	Controlador de Temperatura 2
Y	Conversor de 4-20 mA a 1-5 v



**Figura 3.2** Panel de Control de la planta de control de temperatura LCP actualizado (Fuente Propia).

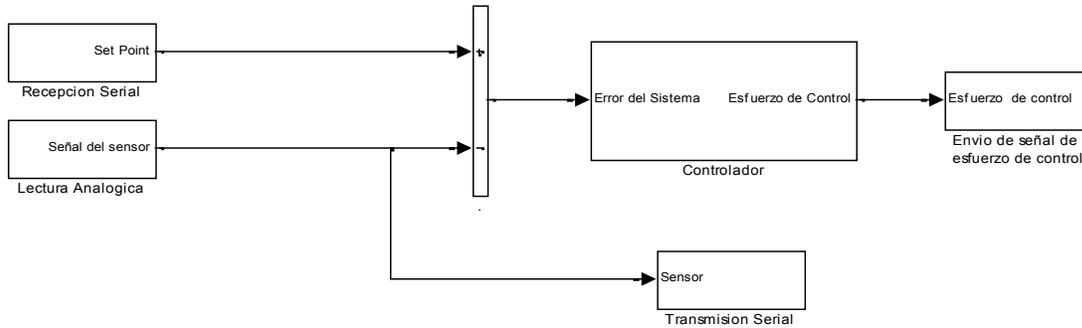


**Figura 3.3** Diagrama de lazo para la Tarjeta Arduino (Fuente Propia).

### 3.2.2. Diagrama de bloques para la programación de la tarjeta Arduino.

Matlab Simulink incorporó un paquete que permite codificar una tarjeta Arduino sin la necesidad de utilizar código manual (Revisar sección 3.1.2 de este documento), a continuación se describe cómo utilizar dicho paquete para programar la tarjeta Arduino Mega 25 60 R3, que se encuentra instalada en la planta de control de temperatura del LCP.

Como ya se dijo con anterioridad en la tarjeta Arduino se escogió el pin 9 como salida de PWM, el pin A0 como entrada analógica y además el puerto 0 para la comunicación serial; una vez se tengan definidos los puertos, se realiza el esquema necesario en Simulink para la programación de la tarjeta; como el fin de la implementación es probar los controladores ya diseñados, se debe seguir una estructura general para la creación de los esquemas, que se puede ver en la figura 3.4, en la cual se distinguen 5 subsistemas fundamentales, los cuales son, lectura analógica, transmisión de datos por el puerto serial, recepción de datos por el puerto serial, controlador, y envío de esfuerzo de control.



**Figura 3.4** Diagrama de bloques genérico para la programación de la tarjeta Arduino, para la implementación en la planta de control de temperatura del LCP.

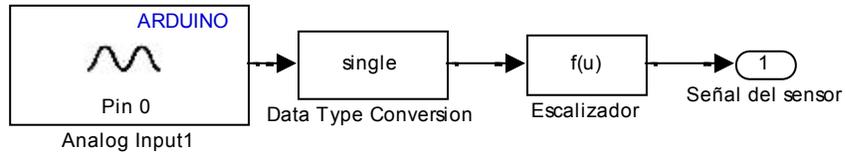
### 3.2.2.1. Subsistema de lectura analógica.

En la figura 3.5 se muestran los bloques necesarios para la configuración de la lectura del puerto analógico y la respectiva escalización de los datos obtenidos.

El primer bloque (*Analogal Input*) es el encargado de realizar la lectura directa del puerto, en él se define el número del puerto leído (A0 para el caso de estudio) y el tiempo de muestreo (5 segundos); este puerto entrega valores entre 0 y 1023, dado que su resolución es de 10 bits, por lo tanto se realizó una conversión a un tipo de dato doble (utilizando el segundo bloque *Data Type Conversion*), que permite realizar la operación de escalizado (se sugiere tipo doble o simple), que es llevada a cabo por el tercer bloque (Escalizador), a través de la implementación de la ecuación 3.1, que convierte los datos leídos en bits a un valor de temperatura; donde  $Temp$  hace referencia a la temperatura de salida y  $u(1)$  al valor digital leído por el puerto de entrada analógica.

La salida de este subsistema va hacia el subsistema del controlador, y hace referencia a la señal que cierra el lazo de control (señal de realimentación).

$$Temp = 0.122775 * u(1) - 27.558825 \quad (3.1)$$



**Figura 3.5** Configuración para la lectura del puerto analógico para la Tarjeta Arduino Mega, a través de Matlab Simulink.

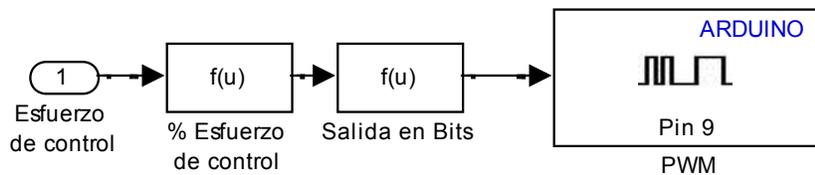
### 3.2.2.2. Subsistema de envío de señal de esfuerzo de control.

Para lograr que la señal de esfuerzo de control tenga un efecto sobre la planta, se realiza la configuración de uno de los puertos de salida PWM con los que cuenta la tarjeta.

La figura 3.6, ilustra la concatenación de los tres bloques necesarios para que la señal de esfuerzo de control se convierta en una salida de PWM. El primer bloque (% Esfuerzo de Control) implementa la ecuación 3.2, donde la variable  $PWM$  es la salida y  $u(2)$  es la señal correspondiente al esfuerzo de control, esta implementación tiene como fin convertir la salida de voltaje del controlador en un porcentaje de este que lo represente. A continuación el segundo bloque (Salida en Bits), efectúa la ecuación 3.3, donde  $Bits$  respresenta la salida y  $u(3)$  el porcentaje del voltaje que representa el esfuerzo de control, este bloque entrega el esfuerzo de control en bits de 0 a 255, debido a la resolución del puerto PWM (que es de 8 bits), y que es entregada al último bloque (PWM), el cual manipula el ciclo de trabajo de una onda cuadrada dependiendo del valor que se le entregue.

$$PWM = \left( \frac{u(2)}{220} \right)^2 \quad (3.2)$$

$$Bits = (u(3) * 255) \quad (3.3)$$



**Figura 3.6** Configuración del puerto de salida PWM para la Tarjeta Arduino Mega, a través de Matlab Simulink.

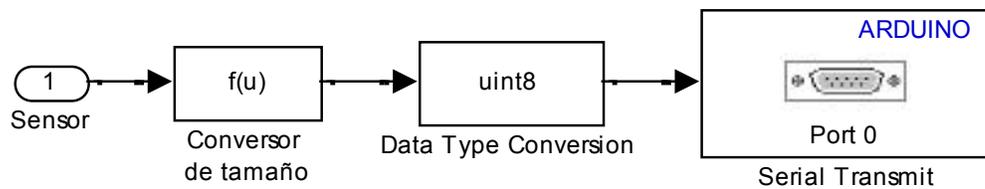
### 3.2.2.3. Subsistemas de recepción y transmisión serial.

Para lograr una buena visualización del comportamiento del sistema, se diseñó una interfaz hombre máquina que permite al usuario definir el *Set Point* de la planta, y ver la conducta que el sistema presenta ante los diferentes cambios. La tarjeta arduino cuenta con 4 puertos seriales, a continuación se describe la configuración del puerto que permite la comunicación entre la tarjeta y la interfaz.

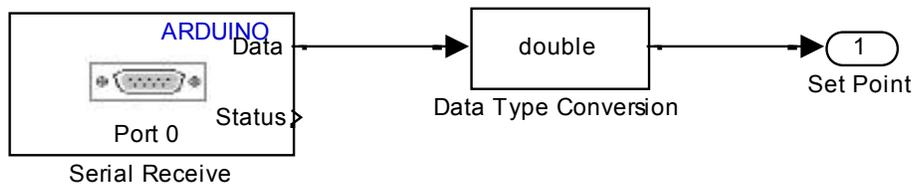
Los 4 puertos seriales que posee la tarjeta Arduino Mega, tienen una resolución de 8 bits, por lo que se realizó una conversión de los datos para la transmisión serial (ver figura 3.7), implementada a través del primer bloque (Conversor de Tamaño), y que responde a la ecuación 3.4, donde  $Bits2$  es la salida y  $u(4)$  el dato que se desea transmitir, en valores que van de 0 a 255, es importante resaltar que ese dato es de tipo doble, así el segundo bloque (*Data Type Conversion*) lo convierte a tipo uint8, para poder ser transmitido por el último bloque (*Serial Transmit*), al cual se le configura el puerto por el cual se realiza la conexión.

$$Bits2 = \frac{u(4) * 255}{100} \quad (3.4)$$

En la figura 3.8, se ilustra la recepción de datos por el puerto serial, esa comunicación es simple para el caso de estudio, un dato de 0 a 255 ingresa por el bloque (*Serial Receive*) de tipo uint8, el cual posteriormente es convertido a tipo double por el bloque (*Data Type Conversion*).



**Figura 3.7** Configuración para la transmisión serial por la Tarjeta Arduino Mega, a través de Matlab Simulink.



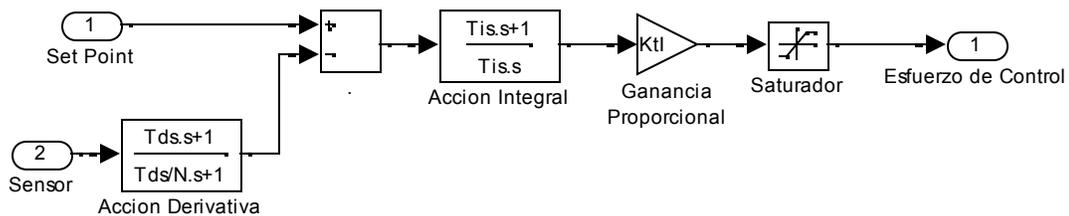
**Figura 3.8** Configuración para la recepción serial de la Tarjeta Arduino Mega, a través de Matlab Simulink.

### 3.2.2.4. Subsistema de control.

El subsistema de control, depende del controlador a implementar, por lo que posteriormente en este documento se explicara cómo hacer la implementación para cada uno.

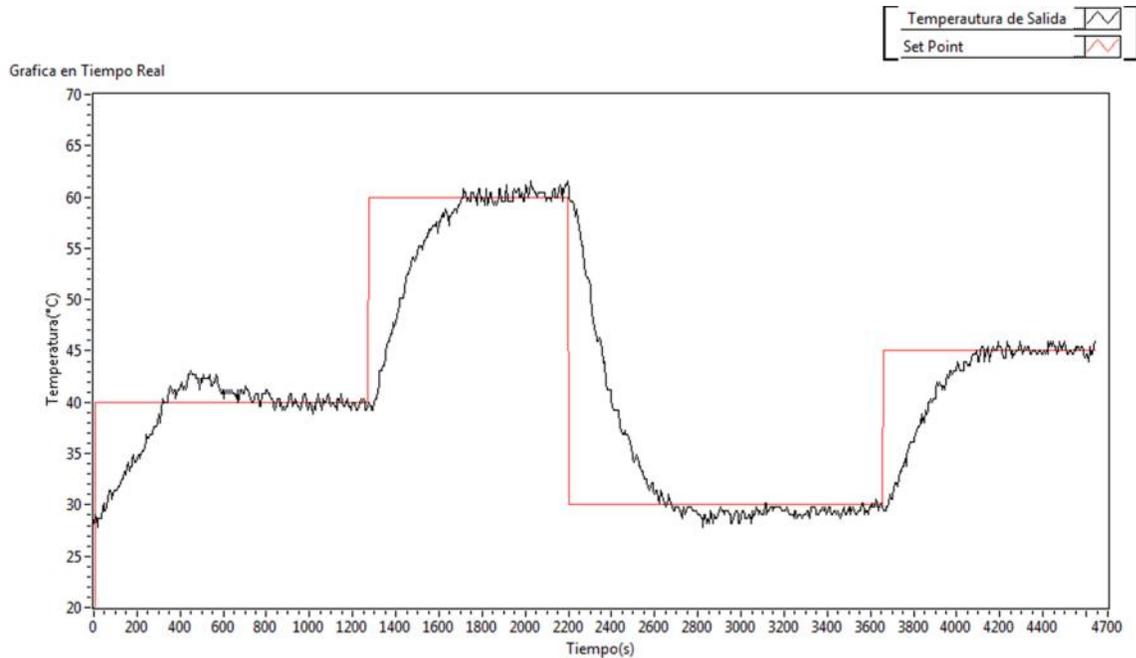
### 3.2.3. Implementación del Controlador PID Serie Industrial.

El esquema en Simulink para la implementación del controlador PID Industrial sintonizado en la sección 2.2.2, se muestra en la figura 3.9.



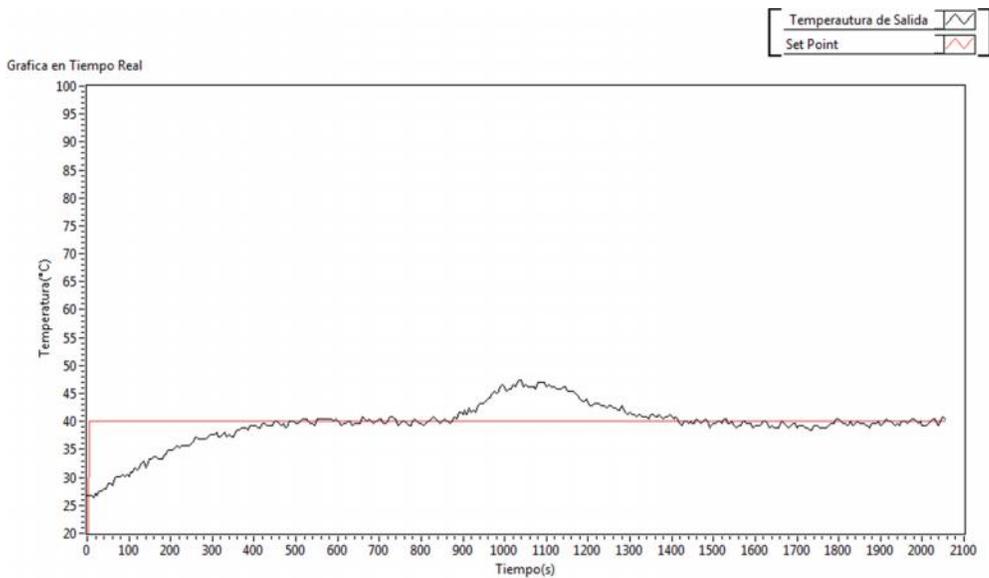
**Figura 3.9** Diagrama en Matlab-Simulink para la implementación de un controlador PID Serie Industrial.

En la gráfica 3.10, se ilustra la respuesta del sistema ante la implementación de dicho controlador, se realizaron 4 cambios a la referencia 40, 60, 30 y 45 °C respectivamente, durante aproximadamente 70 minutos, donde se puede ver que el sistema presenta sobre impulsos, no mayores a 5 °C, y un tiempo de estabilización de 1000 segundos aproximadamente.



**Figura 3.10** Respuesta del sistema bajo la implementación del controlador PID serie industrial

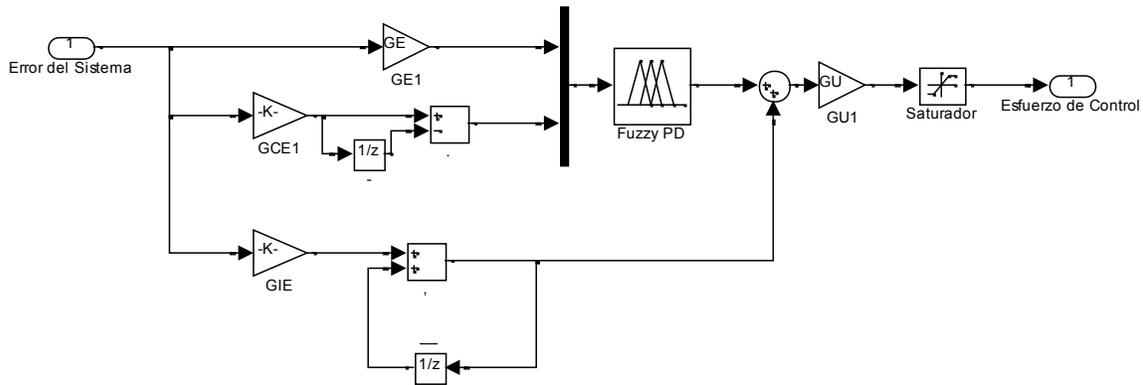
En la figura 3.11, muestra la respuesta del controlador ante disturbios, primero se estabiliza la planta en una temperatura de 40°C y aproximadamente al segundo 800 de simulación, sucede un disturbio que aumenta la temperatura aproximadamente 10°C, a lo que el controlador responde compensándolo en aproximadamente 400 segundos.



**Figura 3.11** Respuesta del controlador ante presencia de disturbios externos.

### 3.2.4. Implementación del Controlador PD+I Difuso.

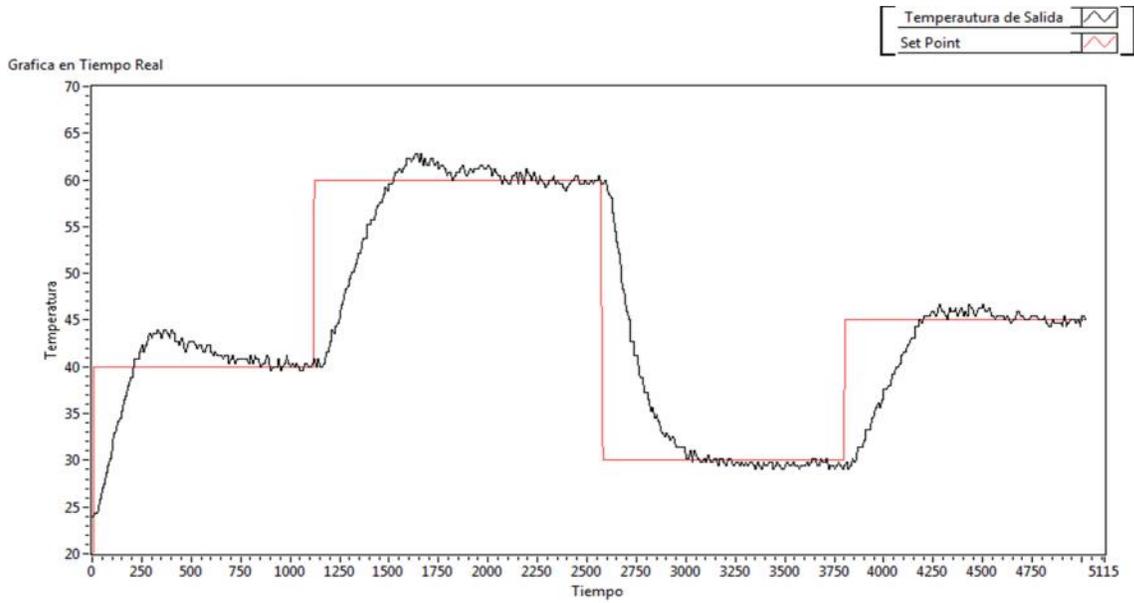
El esquema implementado en Simulink para el controlador PD+I difuso diseñado en la sesión 2.3.2, es ilustrado en la figura 3.12, que se muestra a continuación.



**Figura 3.12** Diagrama en Matlab-Simulink para la implementación de un controlador PD+I Difuso

Para la implementación del controlador en planta se cambió el bloque derivador, por una resta entre el valor actual del error y el pasado, y el bloque integrador por la suma entre el valor actual y los pasados del error, dado que por capacidad de memoria no se pueden descargar los dos bloques utilizados en el diseño sobre la tarjeta.

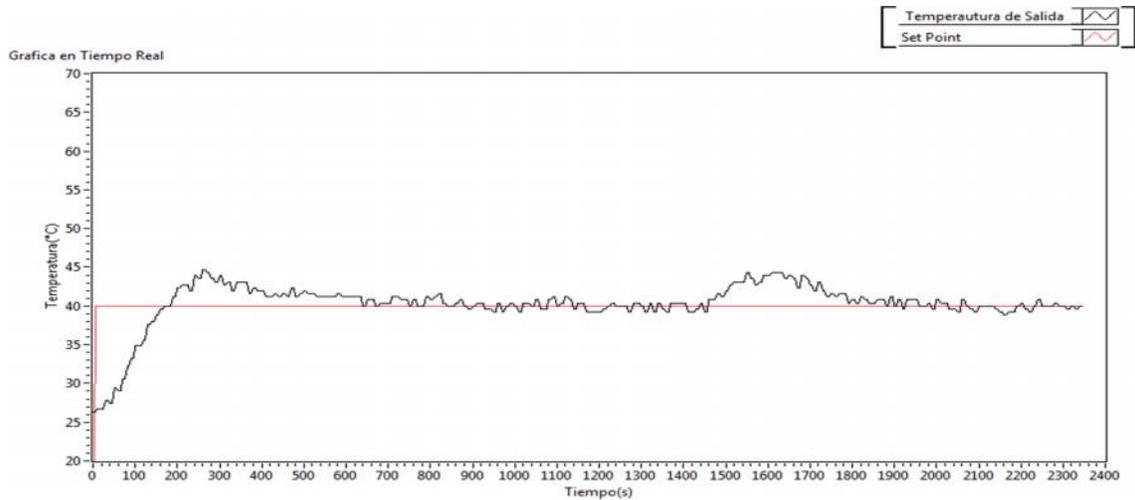
En la figura 3.13, se muestra el comportamiento que tiene el sistema ante la implementación del controlador PD+I difuso.



**Figura 3.13** Comportamiento del sistema bajo la Implementación de un controlador PD+I Difuso.

En la gráfica se observa la respuesta del sistema ante la implementación del controlador PD+I difuso diseñado anteriormente, ante cuatro cambios en la referencia 40, 60, 30 y 45 °C respectivamente, y se muestra que el sistema presenta sobre impulsos no mayores a 5 grados, y un tiempo de estabilización de aproximadamente 1000 segundos.

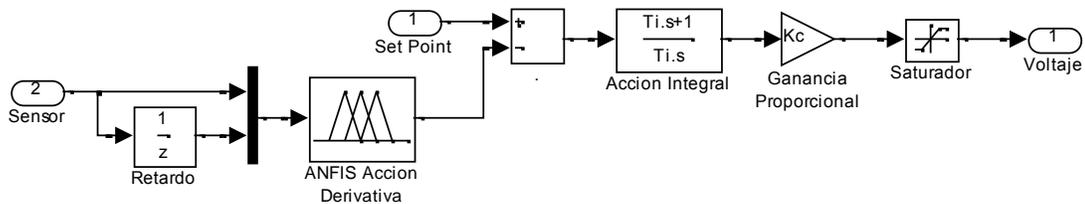
En la figura 3.13, se muestra la respuesta del sistema ante la implementación de un controlador PD+I difuso, con la presencia de disturbios externos a la planta, primero se estabilizó la planta en una temperatura de 40°C y aproximadamente a los 1500 segundos sucede una perturbación externa que aumenta la temperatura alrededor de 5 °C, la cual es compensada por el controlador en aproximadamente 400 segundos.



**Figura 3.14** Comportamiento del controlador PD+I Difuso, ante la presencia de perturbaciones externas.

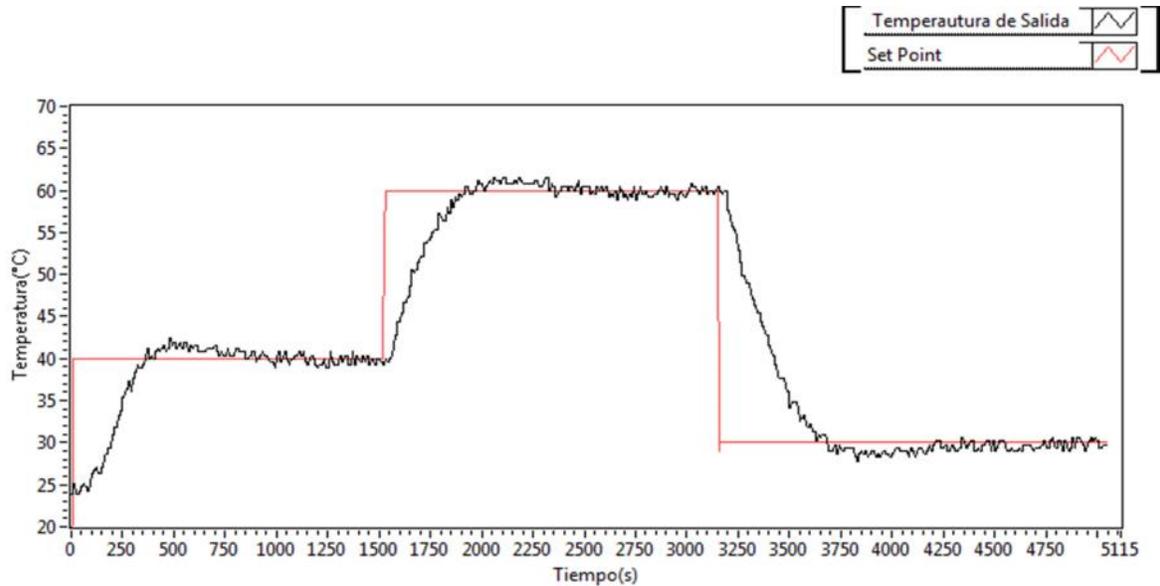
### 3.2.5. Implementación del Modelo de Acción Derivativa obtenido por ANFIS.

En la figura 3.15, se muestra el esquema utilizado para programar la tarjeta arduino desde Simulink.



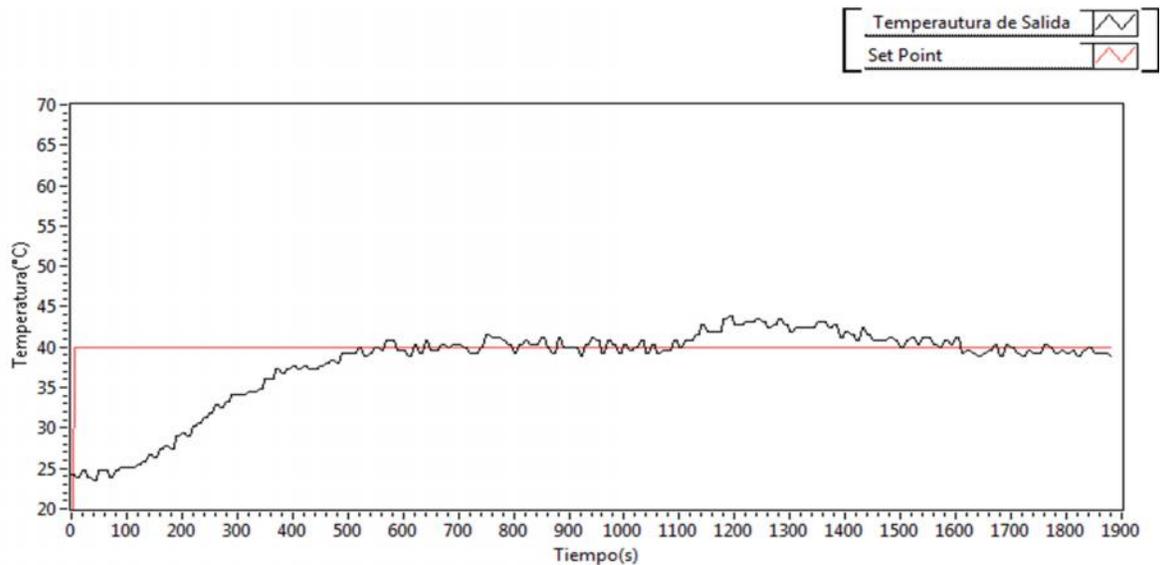
**Figura 3.15** Esquema en Simulink para la implementación del modelo obtenido por ANFIS de la acción derivativa del controlador PID serie industrial.

En la figura 3.16, se muestra la respuesta del sistema bajo la implementación del modelo de acción derivativa obtenido por ANFIS, al sistema se le realizaron cuatro cambios en la referencia 40, 60 y 30 °C respectivamente, en los cuales se puede observar que presenta sobre impulsos y un tiempo de estabilización de 1500 segundos aproximadamente.



**Figura 3.16** Respuesta del sistema bajo la implementación del modelo de acción derivativa por ANFIS.

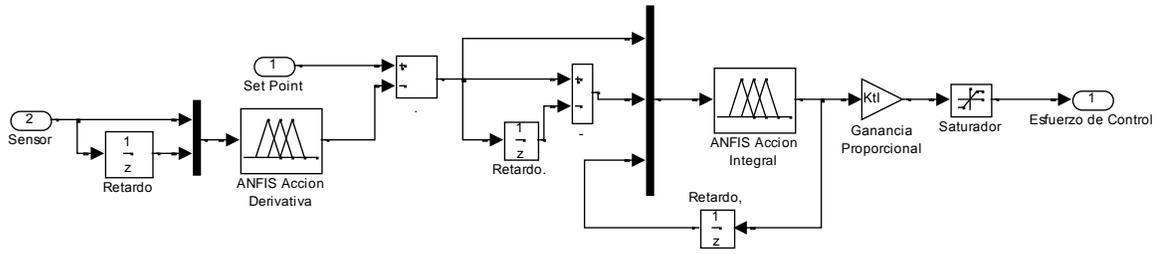
En la figura 3.17, se ilustra la respuesta del controlador compuesto por el modelo ANFIS y la acción integral ante la presencia de disturbios externos a la planta, se puede notar que primero se estabilizó la planta en una temperatura de 40°C y aproximadamente en el segundo 1500 se introduce un disturbio, el cual provoca que la temperatura del sistema aumente, a lo cual el controlador reacciona compensando el disturbio en aproximadamente 300 segundos.



**Figura 3.17** Comportamiento del modelo obtenido por ANFIS más la acción integral, ante la presencia de una perturbación.

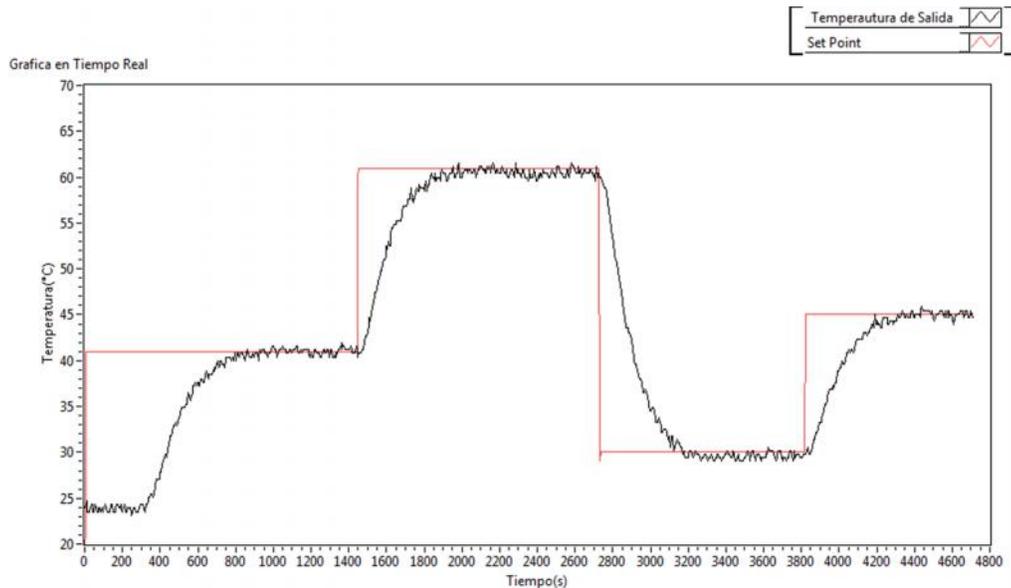
### 3.2.6. Implementación de los Modelos de Acción Integral y Derivativa obtenidos por ANFIS.

En la figura 3.18, se ilustra el esquema en Simulink necesario para la implementación de los modelos obtenidos por ANFIS en la tarjeta.



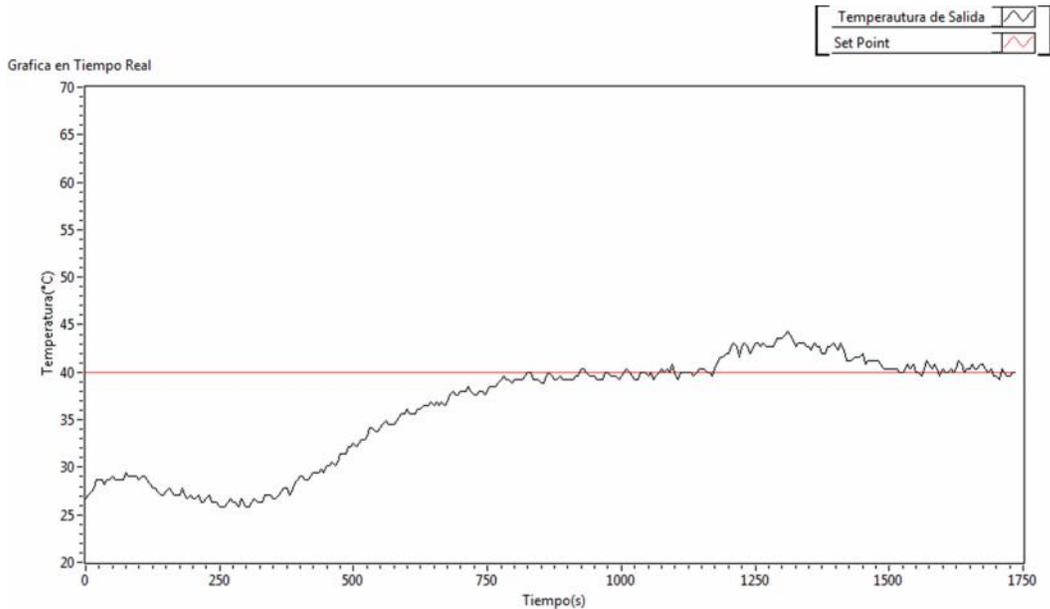
**Figura 3.18** Esquema en Simulink para la implementación de los modelos obtenidos por ANFIS de la acción integral y derivativa del controlador PID serie industrial.

En la figura 3.19, se muestra la respuesta del sistema bajo la implementación de los modelos de acción integral y derivativa obtenidos por ANFIS, al sistema se le realizaron cuatro cambios en la referencia 40, 60, 30 y 45 °C respectivamente, en los cuales se puede observar que el sistema no presenta sobre impulsos y un tiempo de estabilización de 800 segundos aproximadamente.



**Figura 3.19** Respuesta del sistema bajo la implementación de los modelos de acción integral y derivativa obtenidos por ANFIS.

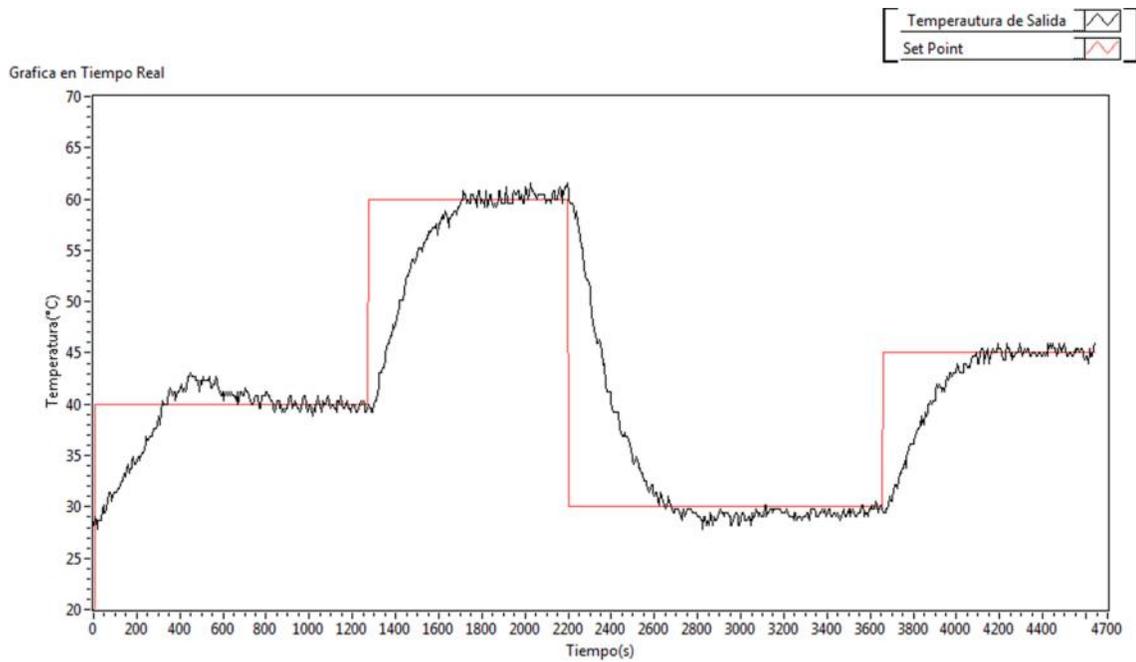
En la figura 3.20, se ilustra la respuesta del controlador compuesto por los modelos ANFIS ante la presencia de disturbios externos a la planta, se puede notar que primero se estabilizó la planta en una temperatura de 40°C y aproximadamente en el segundo 1200 se introduce un disturbio, el cual provoca que la temperatura del sistema aumente, a lo cual el controlador reacciona compensando el disturbio en aproximadamente 300 segundos.



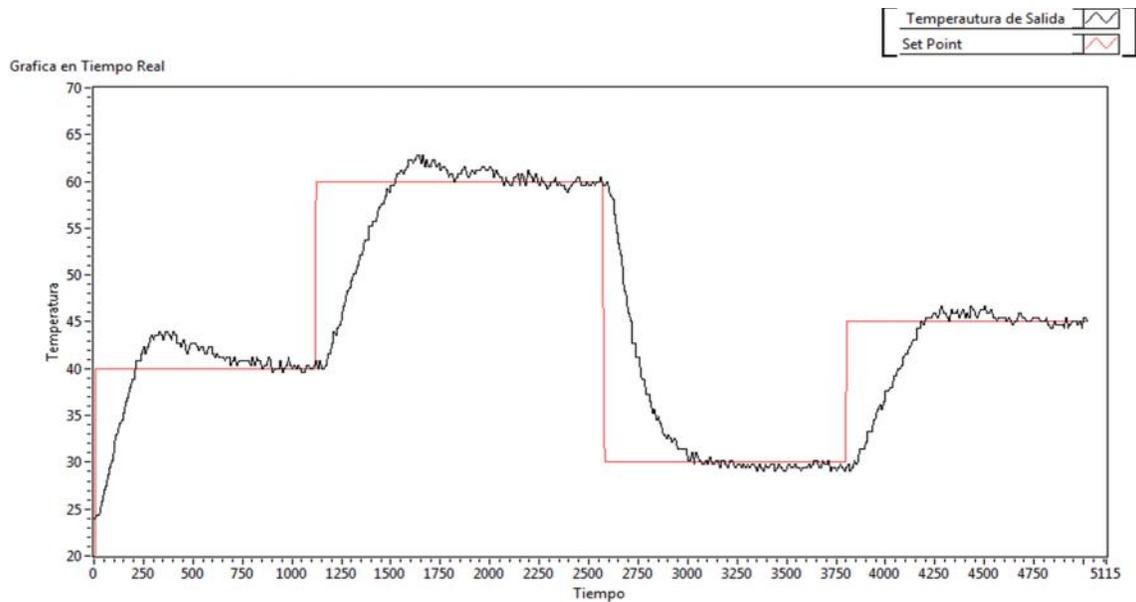
**Figura 3.20** Comportamiento de los modelos obtenidos por ANFIS, ante la presencia de una perturbación.

### 3.2.7. Resultados de la implementación.

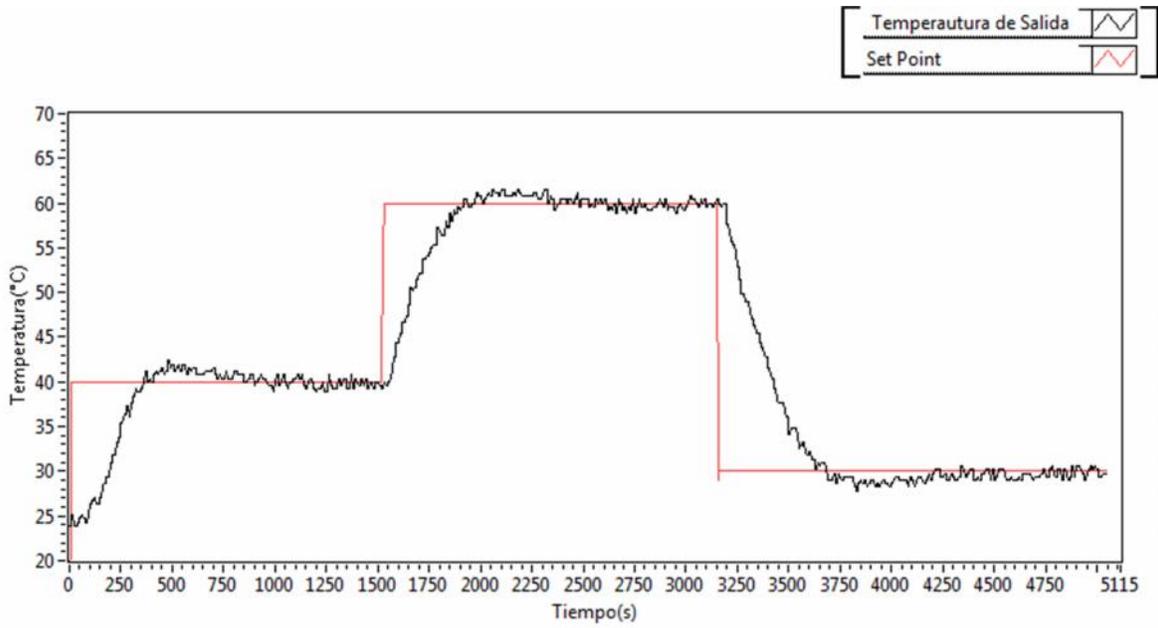
Con el fin de realizar un análisis y comparación de los datos obtenidos luego de la implementación de los controladores en la planta de temperatura del LCP, a continuación se muestran de nuevo las gráficas 3.10, 3.13 y 3.16, que muestran las respuestas del sistema ante la implementación de los controladores PID serie Industrial, PD+I Difuso y los modelos de la acción integral y derivativa obtenidos por ANFIS respectivamente.



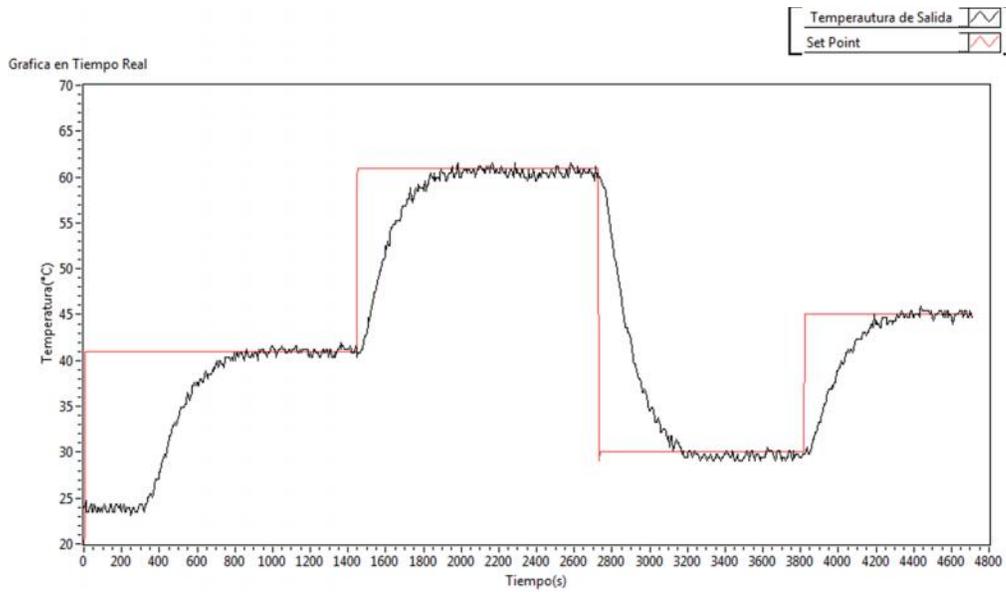
**Figura 3.10** Respuesta del sistema bajo la implementación del controlador PID serie industrial



**Figura 3.13** Comportamiento del sistema bajo la Implementación de un controlador PD+I Difuso.



**Figura 3.16** Respuesta del sistema bajo la implementación del modelo de acción derivativa obtenido por ANFIS.



**Figura 3.19** Respuesta del sistema bajo la implementación de los modelos de acción integral y derivativa obtenidos por ANFIS.

En la tabla 3.4, se comparan los diferentes controladores, identificando las diferencias que existen entre estos, basándose en 4 ítems específicos, que son el tiempo de estabilización del sistema, bajo la implementación de cada controlador, el valor del máximo sobre impulso presentado por el mismo, si el controlador está en la capacidad de compensar disturbios, y el tiempo que se demora en compensarlos.

**Tabla 3.4** Comparación entre los diferentes controladores implementados en planta.

Controlador	Tiempo de Estabilización (s)	Máximo Sobre Impulso (°C)	Compensación a Disturbios	Tiempo de Compensación (s)
PID Serie Industrial	1000	No mayor a 5	Si	400
PD+I Difuso	1000	No mayor a 4	Si	400
Modelado por ANFIS + I	1500	No mayor a 4	Si	600
Modelado por ANFIS	800	No presenta	Si	300

De acuerdo a los datos obtenidos de la implementación de los controladores sobre la planta, se puede deducir que el controlador PD+I difuso es el que presenta menores ventajas, dado que tiene un tiempo de estabilización mayor y presenta sobre impulsos de alrededor de 5 °C en la mayoría de cambios de *Set Point* realizados, mientras que el controlador PID serie industrial, solo presenta sobre un sobre impulso en el primer cambio presentado y los modelos obtenidos por ANFIS no los presentan, se tiene en cuenta también que el controlador formado por los modelos obtenidos por ANFIS es el mayor velocidad, aunque su diferencia no es muy considerable, comparándola con el PID serie industrial.

## 4. GUÍA DE LABORATORIO

A continuación se presenta una guía procedimental, en la cual se describen detalladamente los pasos que se deben seguir en la planta de control de temperatura del LCP, para diseñar e implementar un controlador PID serie industrial, un controlador PD+I difuso y un controlador basado en modelos ANFIS.

### 4.1. Objetivos

#### General

Diseñar e implementar sistemas de control de temperatura para la planta del LCP, basados en técnicas clásicas e inteligentes, que complementen el desarrollo de las clases teóricas del curso de control inteligente.

#### Específicos

1. Diseñar e implementar un sistema de control de temperatura PD+I basado en técnicas de lógica difusa.
2. Diseñar e implementar un sistema de control de temperatura, bajo la técnica de control PID serie Industrial.
3. Modelar las acciones integral y derivativa del control PID serie Industrial con ANFIS, e implementar el resultado en la planta.

### 4.2. Requerimientos

- Matlab - Simulink R 2012 a o versiones recientes.
- Simulink Support Package for Arduino Hardware
- Tarjeta Arduino Mega 2560 R3.
- National Instruments *LabVIEW* 2010.
- Planta de control de Temperatura LCP Universidad del Cauca.

### 4.3. Procedimientos Generales.

A continuación se explican los procedimientos generales para el encendido y apagado de la planta, aparte también se realiza una explicación detallada de como programar la tarjeta Arduino desde Simulink, al igual de una guía para la instalación del paquete de soporte para Arduino de Simulink.

#### 4.3.1. Instalación Simulink Support Package for Arduino Hardware

- Abrir Matlab.
- En la ventana de comandos escribir *targetinstaller*, y ejecutar oprimiendo *Enter* como se ve en la figura 4.1.



Figura 4.1 Tragetinstaller

- En la ventana desplegada elegir la opción *Internet* y dar clic en *next*, como se ve en la figura 4.2.

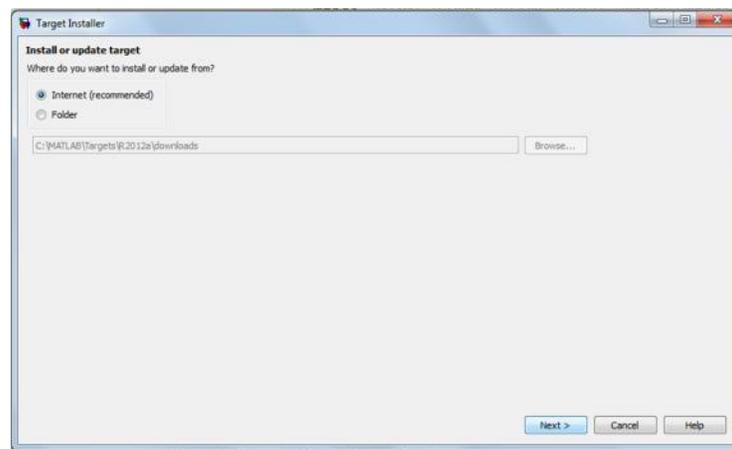
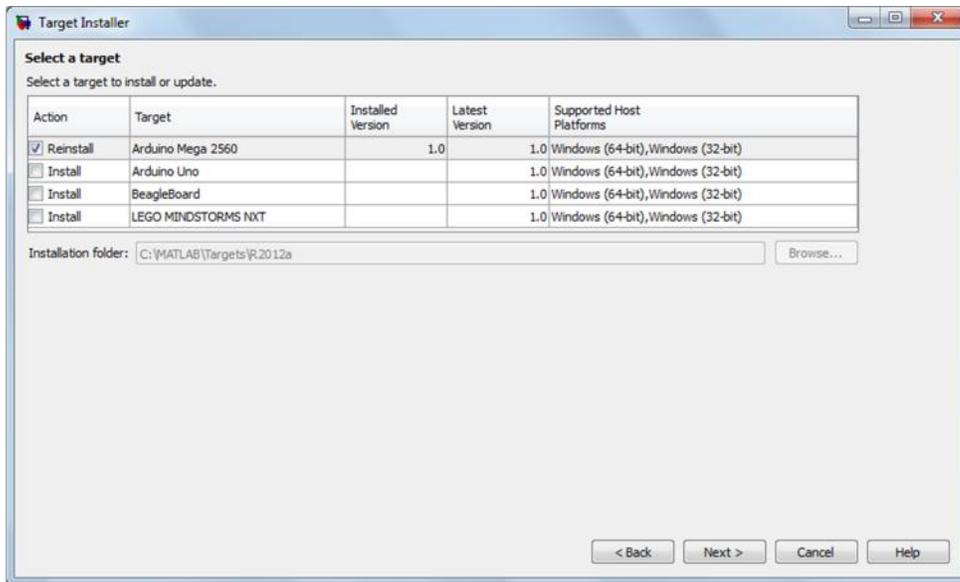


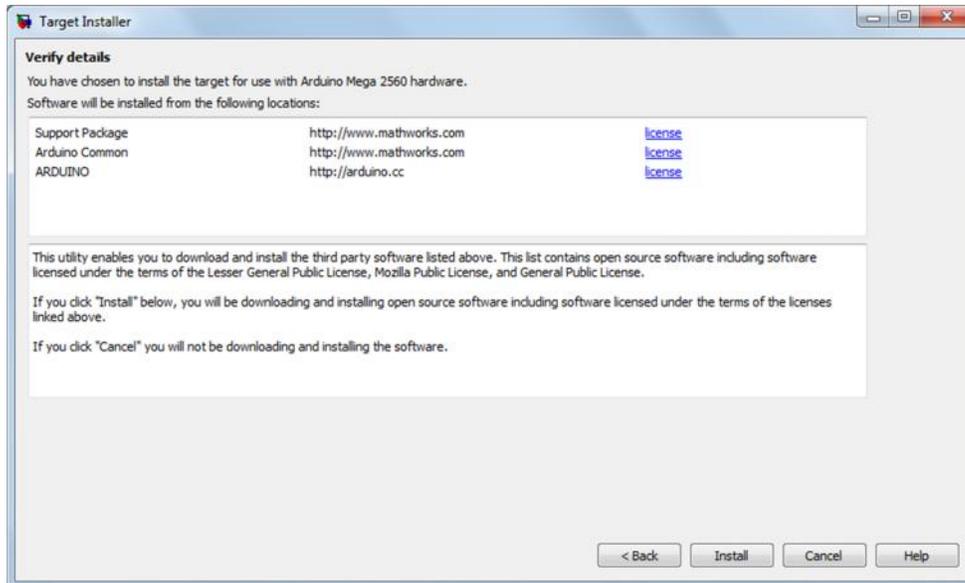
Figura 4.2 Opción de instalación *Internet*

- En la ventana desplegada elegir la opción Arduino Mega 2560 como se ve en la figura 4.3, y dar clic en *next*.



**Figura 4.3** Elegir como *hardware* la tarjeta arduino Mega 2560

- En la ventana desplegada dar clic en *Install*, como se ve en la figura 4.4.



**Figura 4.4** Instalar el paquete.

- Con esto se termina la instalación.

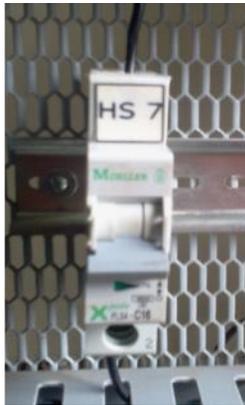
#### 4.3.2. Encendido de la Planta de Temperatura de LCP.

- Encienda el PC de la estación 6.
- Ponga las perillas HS1 y HS2 en posición *Manual* y *Arduino* respectivamente, como se ve en la figura 4.5.



**Figura 4.5** Posición de las perillas para el encendido de la planta.

- Verifique que el panel de control esté conectado a la fuente de 110 V.
- Encienda el *breaker* HS7 que alimenta a la planta con 110v (figura 4.6.).



**Figura 4.6** Breaker HS7

- Verifique que la fuente de 24 voltios ubicada en la estación 3 Planta de Nivel, este encendida.(figura 4.7)



**Figura 4.7** Fuente de 24 Voltios

- Oprima en ON el *Switch* HS5, que alimenta el panel con 24 voltios (figura 4.8).



**Figura 4.8** *Switch* HS5

- Verifique que el cable USB de la tarjeta Arduino esté conectado al PC.
- Encienda el *Switch* HS6, que alimenta a las resistencias con 220 v, ver figura 4.9.



**Figura 4.9** *Switch*HS6

- Encienda el ventilador, por medio del pulsador HS3, ilustrado en la figura 4.10.



**Figura 4.10** Pulsador para encender el ventilador.

### 4.3.3. Apagado de la Planta.

- Apague las resistencias, esto se hace oprimiendo el pulsador de color rojo que se encuentra en el *Switch* HS6, ver figura 4.11.



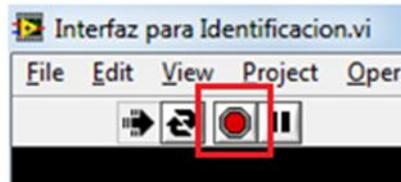
**Figura 4.11** Interruptor para el encendido y apagado de las resistencias.

- Ahora apague el ventilador, para esto se debe oprimir el pulsador HS4, que se encuentra en el panel de control de la planta, ver imagen 4.12.



**Figura 4.12** Pulsador HS4, Apaga el ventilador de la planta.

- Detenga la interfaz de usuario en *LabVIEW*, para esto se pulsa el botón de *Stop*, ver figura 4.13.

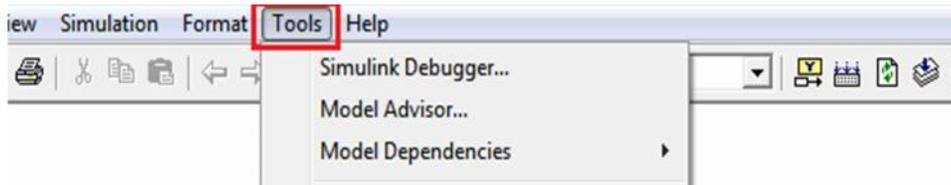


**Figura 4.13** Detener la interfaz de *LabVIEW*.

- Una vez detenida la interfaz de usuario, apague las fuentes de 110v y 24v.

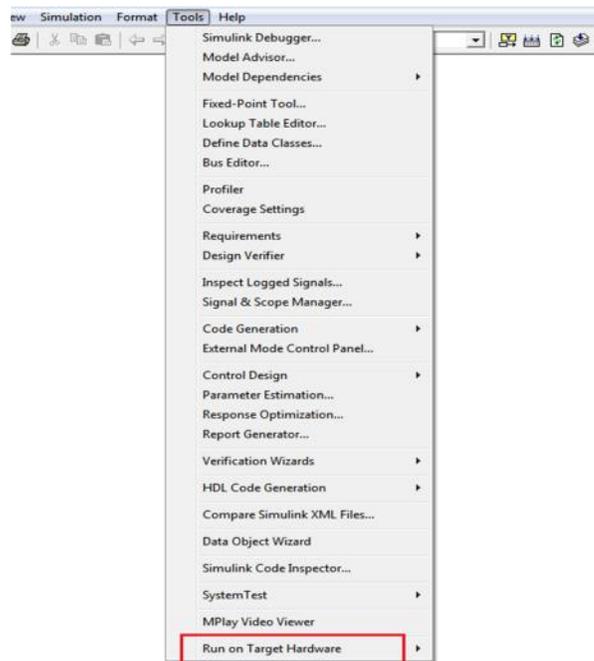
#### 4.3.4. Programar la Tarjeta Arduino Mega 2560 R3.

- Desde el archivo en Simulink que desee descargar, debe ir a la opción “Tools”, en la barra de herramientas de Simulink, ver figura 4.14.



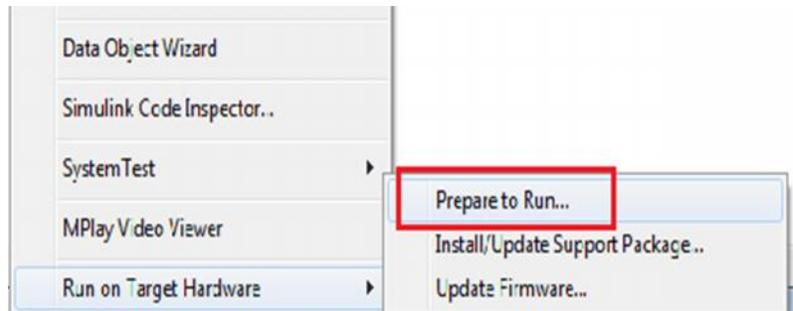
**Figura 4.14** Ubicación de la opción Tools

- Dar clic en la opción “Run on Target Hardware”, figura 4.10.



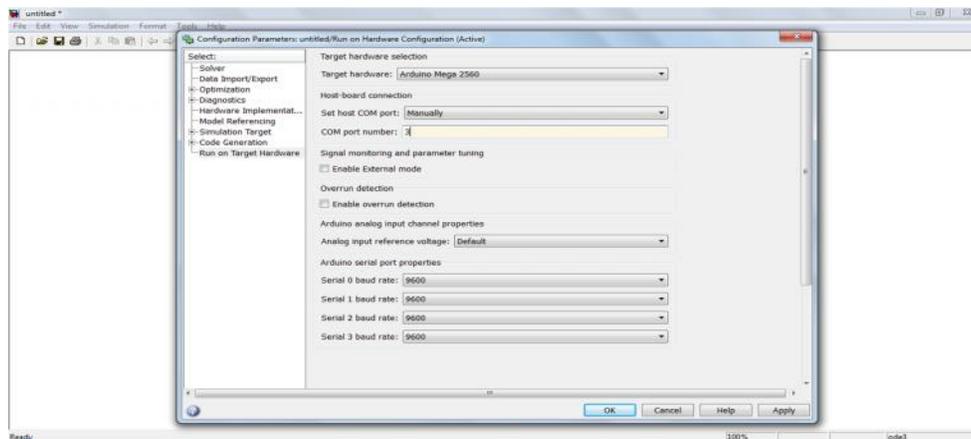
**Figura 4.15** Opción Run on Target Hardware

- Elegir la opción “Prepare to Run”, este paso debe ejecutarse solo una vez, figura 4.16.



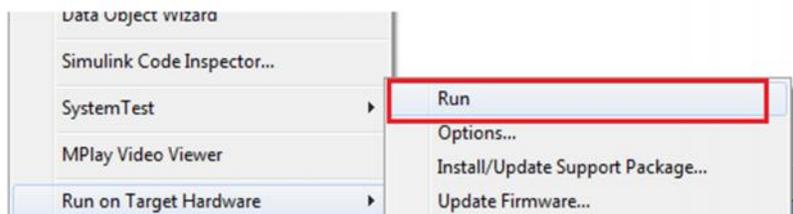
**Figura 4.16** Opción Prepare toRun

- En la ventana desplegada elegir en la opción “*Target Hardware*” la opción *Arduino Mega 2560*; en “*Set host COM port*” la opción *Manually* y en la opción “*COM port number*” el puerto COM en el cual se encuentra la tarjeta arduino, por ultimo dar clic en *Apply* y luego OK, este paso debe ejecutarse solo una vez, figura 4.17.



**Figura 4.17** Ventana de configuración, para la comunicación HW-SW

- Regresar a la opción “*Run on Target Hardware*”, y dar clic en *Run*, figura 4.18.



**Figura 4.18** Opción RUN

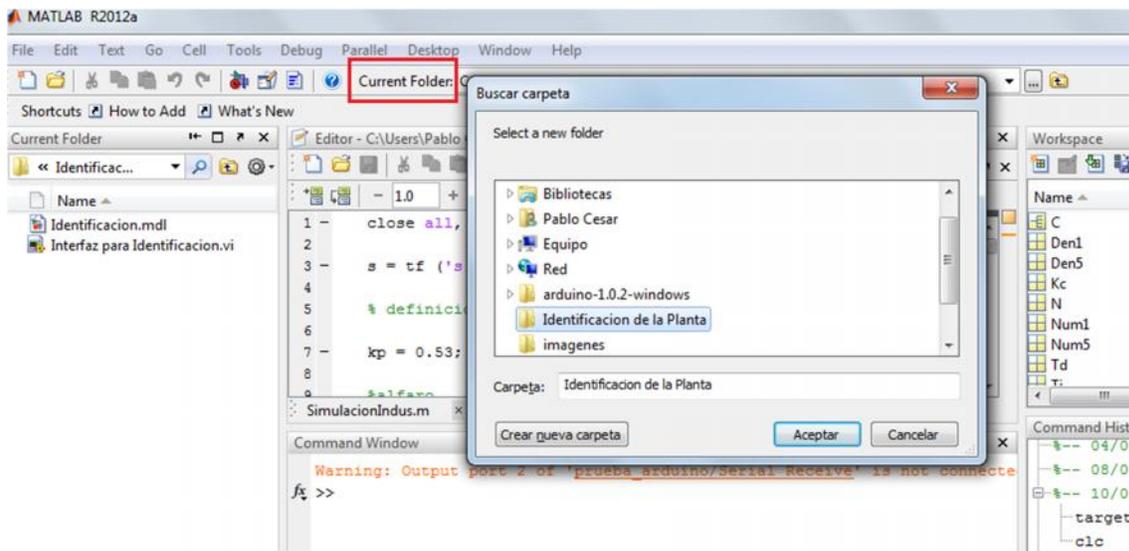
- Esperar que se descargue el código en la tarjeta, al finalizar, aparecerá un mensaje que indica que la operación fue exitosa, en la esquina inferior izquierda de la ventana en Simulink, como se muestra en la figura 4.19.



**Figura 4.19** Operación Exitosa.

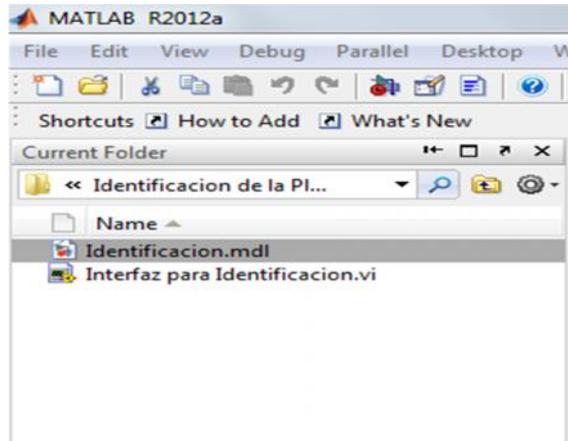
#### 4.4. Identificación de la Planta por el método de curva de reacción.

- Inicie Matlab.
- Establezca como carpeta de trabajo “Identificación de la planta”, para esto ir a *Current Folder*, dar clic en el icono , y buscar su ubicación, la carpeta debe encontrarse en el escritorio del computador, ver figura 4.20.



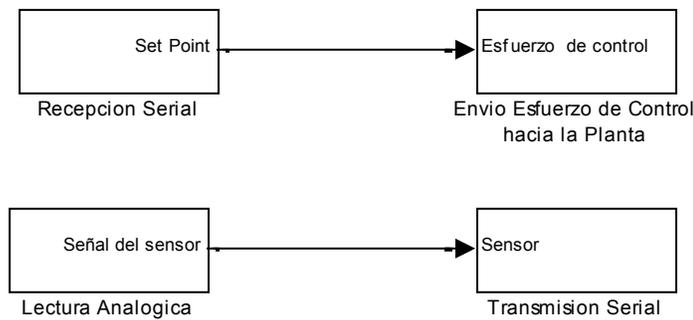
**Figura 4.20** Establecer una carpeta de trabajo en Matlab.

- Abrir el archivo “Identificacion.mdl”, ver figura 4.21.



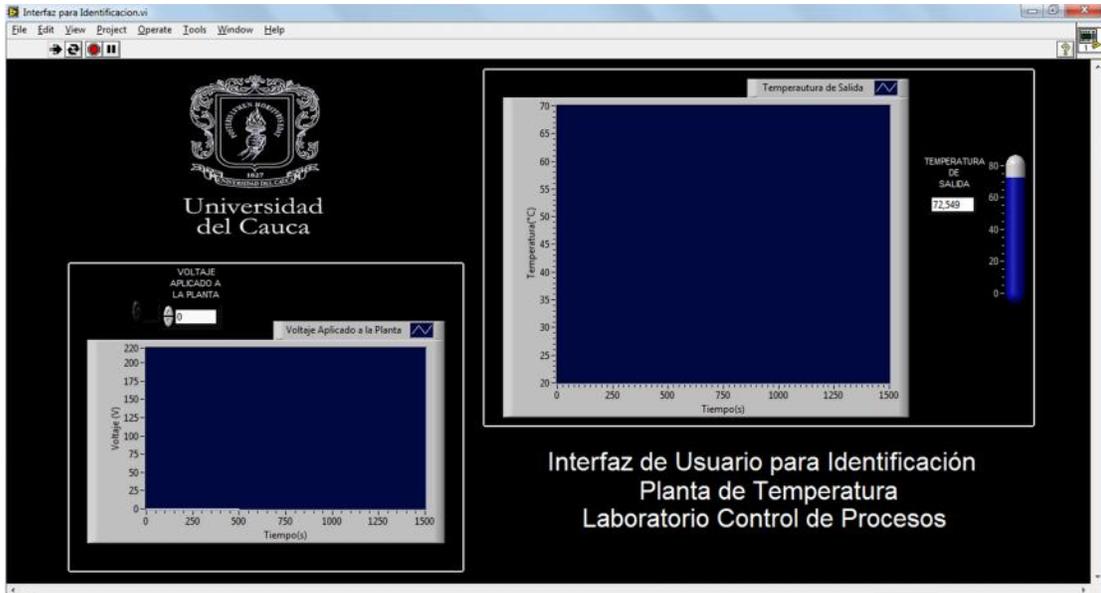
**Figura 4.21** Archivos de la carpeta de trabajo.

- Descargar dicho archivo (Ver figura 4.22) en la tarjeta Arduino, para esto realizar primero los pasos indicados en la sesión 4.3.3.



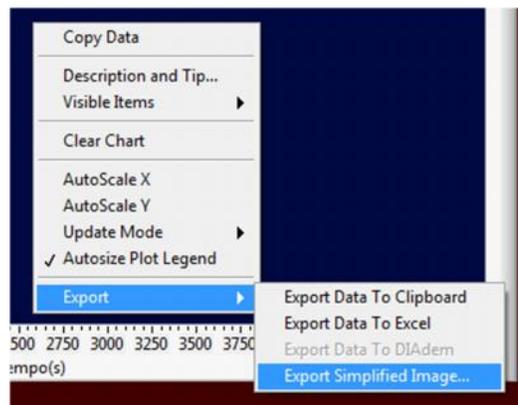
**Figura 4.22** Modelo en Simulink para la identificación de la planta

- Una vez programada la tarjeta con el código necesario para la identificación de la planta, se debe abrir y ejecutar el archivo “Interfaz para Identificación.vi”, que se encuentra en la carpeta “Identificación de la planta” ubicada en el escritorio del computador de la estación 6, ver figura 4.23.



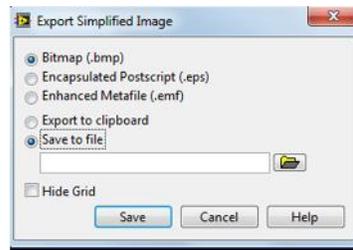
**Figura 4.23** Interfaz para Identificación de la planta

- Para obtener la curva de reacción, aplique un voltaje de  $66V_{a.c}$  (Colocando dicho valor en el espacio destinado para él en la interfaz), equivalente a un 30% de la señal total que ingresa a la planta, y espere de 12 a 15 minutos a que la temperatura estabilice alrededor de  $30^{\circ}\text{C}$ .
- Una vez estabilizada la temperatura, aplique un voltaje de  $154 V_{a.c}$ , equivalente a un 70% de la señal total que ingresa a la planta, y esperar de 12 a 15 minutos para que la temperatura estabilice alrededor de  $50^{\circ}\text{C}$ .
- Guardar la imagen del comportamiento de la planta (Temperatura de Salida), dando clic derecho sobre la gráfica, ir a la opción "Export" y luego a la opción "Export Simplified Image", ver figura 4.24.



**Figura 4.24** Extraer Imagen desde la interfaz

- En la ventana que se despliega, se debe elegir la opción “Save to File”, y la ubicación en la que se desea guardar la imagen obtenida, y el nombre, ver figura 4.25.



**Figura 4.25** Ventana que permite elegir la ubicación donde se guarda la gráfica.

- Con la gráfica que corresponde a la respuesta de la planta obtenía anteriormente, se procede a llenar las siguientes tablas.

**Tabla 4.1** Registro de los valores de dy

Variación	Variable Controlada	Variable Manipulada
dy		

**Tabla 4.2** Registro de Tiempos.

Método	T28	T63
Smith		

- Una vez obtenidos los parámetros se resuelven las siguientes ecuaciones, con el fin de obtener las constantes del modelo de primer orden más tiempo muerto que representa el comportamiento dinámico de la planta.

$$Kp = \frac{\Delta y}{\Delta u}$$

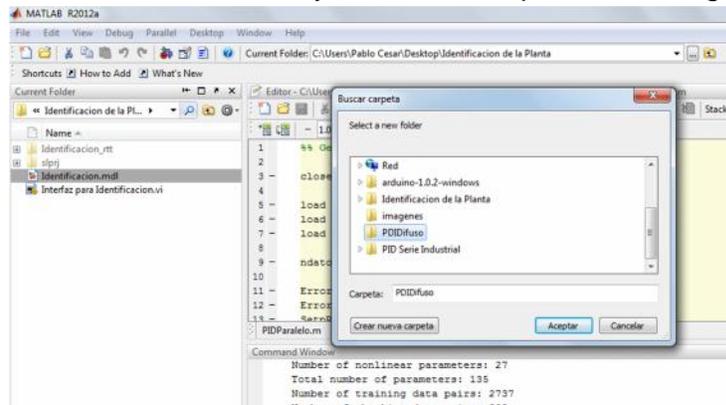
$$\tau = 0.9102(t_{63} - t_{28})$$

$$t_m = 1,262t_{28} - 0,262t_{63}$$

- Con esto se da por finalizada la identificación de la planta, si se desea apagar la planta, remitirse a la sección 4.3.2. de esta guía.

#### 4.5. Sintonización e Implementación de un Controlador PD+I Difuso.

- Primero se debe realizar la sintonización de un controlador PID paralelo clásico, por lo que se parte del modelo FOPDT, si no se tiene, remítase a la sección 4.3.4 de esta guía.
- Inicie Matlab.
- Establecer como carpeta de trabajo en Matlab, la carpeta “PDIDifuso”, que está ubicada en el escritorio del computador de la planta, para esto dirigirse a Current Folder, dar clic en el icono , y buscar la carpeta, ver imagen 4.26.



**Figura 4.26** Establecer carpeta PDIDifuso como carpeta de trabajo en Matlab

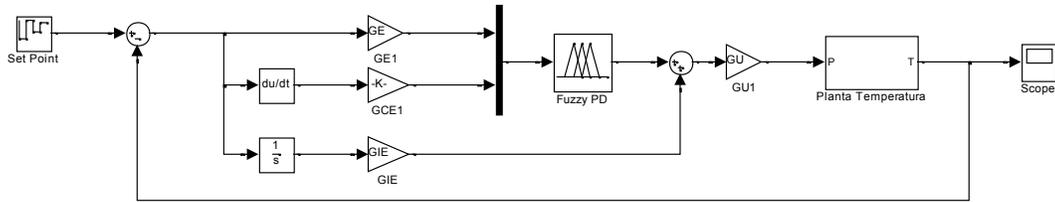
- Se sintonizan las constante del controlador, para esto se utilizan las siguientes ecuaciones, ejecutando el archivo “PIDParalelo.m” que se encuentra en la carpeta “PDIDifuso”.

$$K_c = 1.2 \frac{\tau}{K_p t_m}$$

$$T_i = 2 t_m$$

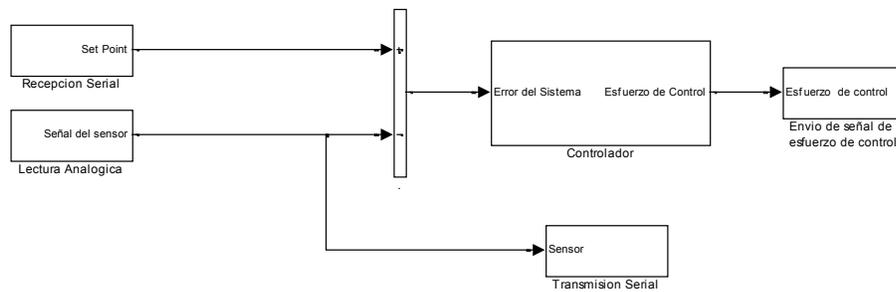
$$T_d = 0.5 t_m$$

- Una vez obtenidas las constantes del controlador PID paralelo clásico, se deben sintonizar las constantes para el controlador difuso, y cargar el controlador PD difuso, para esto abrir el archivo “PDIDifuso.m”, completar la información requerida y ejecutarlo.
- Ahora se debe probar el controlador en simulación, para esto abrir y ejecutar el archivo “PDIdifuso1.mdl”, ver figura 4.27.



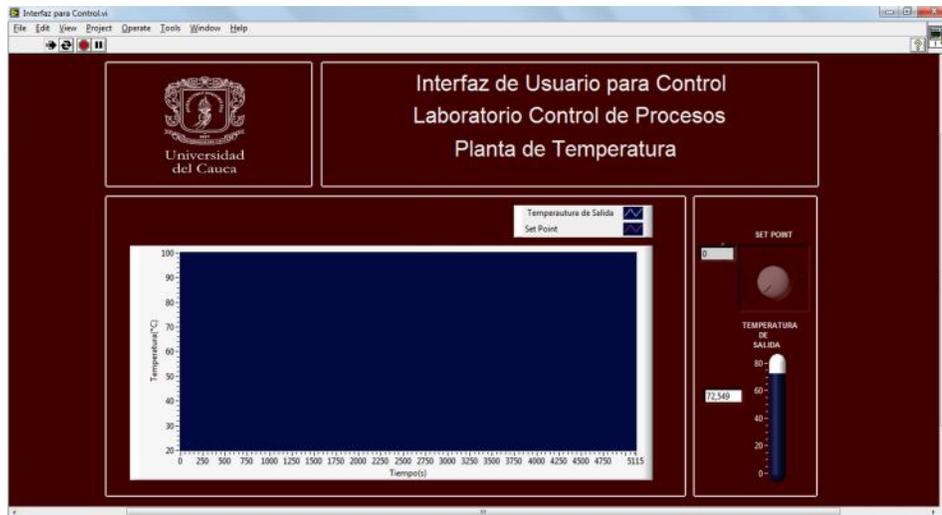
**Figura 4.27** Modelo en Simulink para el controlador PD+I Difuso

- Ahora se implementa el controlador en la planta, para esto se enciende la planta, como se explica en la sección 4.3.1. de esta guía.
- Una vez encendida la planta se descarga el controlador en la tarjeta, el archivo a descargar se llama “PDImplementacion.mdl” (ver figura 4.28.), para realizar la descarga, remitirse a la sección 4.3.3. de esta guía.



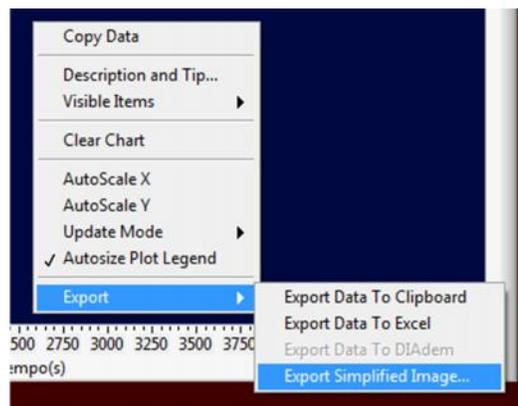
**Figura 4.28** Archivo a descargar para la implementación del Controlador PD+I Difuso en la Planta

- Una vez programada la tarjeta, se debe abrir y ejecutar la interfaz de usuario “InterfazParaControl”, ver figura 4.29.



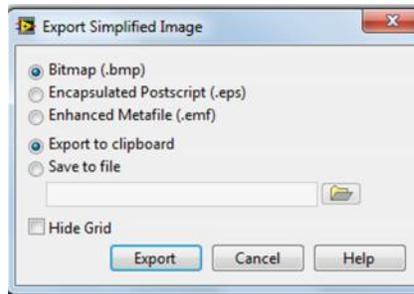
**Figura 4.29** Interfaz para el Control de la Planta.

- Desde la interfaz se puede modificar la temperatura de referencia para la planta, se deben realizar 4 cambios de *Set Point* establecidos en 40, 60, 30 y 45 °C respectivamente, debe esperar de 12 a 15 minutos entre cada cambio de referencia para que la planta alcance a llegar a la temperatura deseada.
- Guardar la imagen obtenida del comportamiento de la planta, dando clic derecho sobre la gráfica, ir a la opción “*Export*” y luego a la opción “*Export Simplified Image*”, ver figura 4.30.



**Figura 4.30** Extraer Imagen desde la interfaz

- En la ventana que se despliega, se debe elegir la opción “*Save to file*”, y la ubicación en la que se desea guardar la imagen obtenida, ver figura 4.31.



**Figura 4.31** Ventana que permite elegir la ubicación donde se guarda la gráfica.

Después de observar el comportamiento del sistema a diferentes cambios en el *Set Point*, se procede a comprobar el comportamiento de la planta ante un disturbio (Figura 4.32).



**Figura 4.32** Disturbio instalado en la planta de temperatura del LCP.

- Verificar que la planta este estable en un valor de temperatura.
- Oprimir ON en el *Switch S7* (Figura 4.33), el cual activa el disturbio y obstruye la salida de flujo del cilindro, con el fin de aumentar la temperatura en el interior del mismo.



**Figura 4.33** *Switch S7*

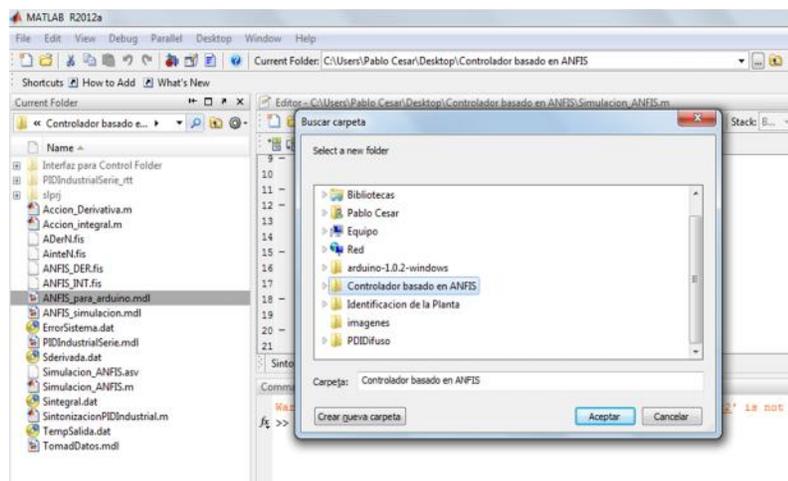
- Esperar a que se genere el disturbio y que posteriormente sea controlado por el sistema de control.
- De ser necesario se guarda la señal como se realizó anteriormente para el comportamiento de la planta ante diferentes *Set Point*.
- Apagar el disturbio oprimiendo OFF en el *Switch S7* (Figura 4.33).
- Apagar la planta, para esto remitirse a la sección 4.3.2. de esta guía.
- Analizar el comportamiento del sistema, con la presencia del controlador PD+I difuso.

#### 4.6. Diseño e Implementación de un controlador basado en ANFIS, a partir de un PID serie industrial.

##### 4.6.1. Sintonización de un PID serie industrial

Para la sintonización del PID serie industrial, se parte del modelo de primer orden más tiempo muerto que representa el comportamiento dinámico de la planta obtenido en la sección 4.3.4.

- Inicie Matlab.
- Establecer como carpeta de trabajo “Controlador basado en ANFIS”, la cual se encuentra ubicada en el escritorio del computador de la planta, para esto dirigirse a *Current Folder*, dar clic en el icono  y buscar la carpeta.



**Figura 4.34** Establecer Controlador basado en ANFIS como Carpeta de trabajo en Matlab

- Se sintonizan las constante del controlador, para esto se hace uso de las siguientes ecuaciones, ejecutando el archivo “SintonizacionPIDIndustrial.m” que se encuentra en la carpeta “Controlador basado en ANFIS”.

$$K_c K_p = a \left( \frac{tm}{\tau} \right)^b$$

$$\frac{T_i}{\tau} = \frac{1}{c} \left( \frac{tm}{\tau} \right)^{-d}$$

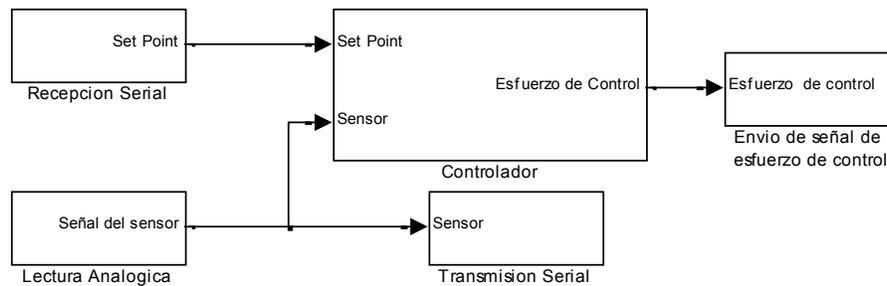
$$\frac{T_d}{\tau} = e \left( \frac{tm}{\tau} \right)^f$$

Los valores de las constantes que se utilizaron son las del criterio de desempeño *integral del error absoluto* (IAE) (tabla 4.3).

**Tabla 4.3** Parámetros IAE

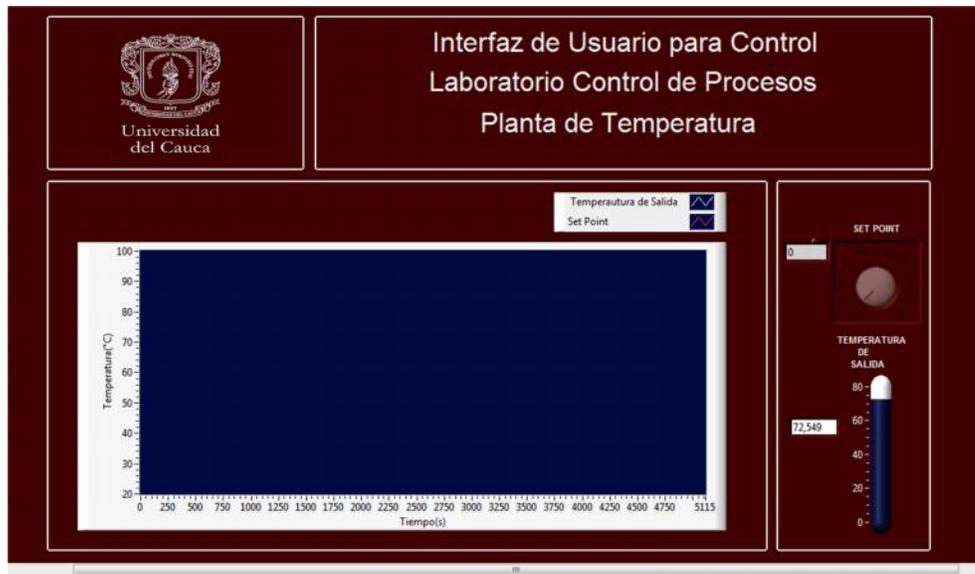
<b>IAE</b>	0,98089	0,76167	0,91032	-1,05211	0,59974	0,89819
------------	---------	---------	---------	----------	---------	---------

- Una vez obtenidos los parámetros se debe abrir el archivo “PIDIndustrial.mdl”, ver figura 4.35.



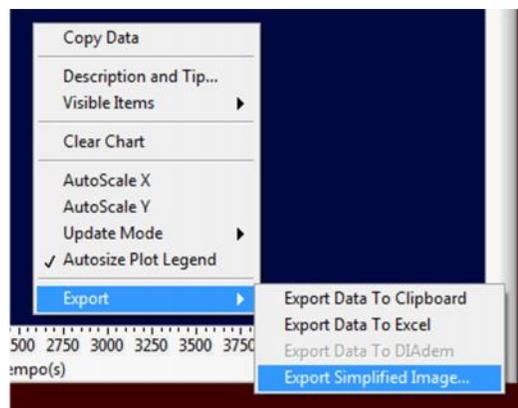
**Figura 4.35** Archivo PIDIndustrialSerie.

- Descargar el Archivo en la tarjeta, como se explica en la sección 4.3.3
- Una vez descargado el archivo sobre la tarjeta, se debe ejecutar el archivo “Interfaz paraControl.vi”, que se encuentra dentro de la carpeta “Controlador basado en ANFIS”, ver figura 4.36



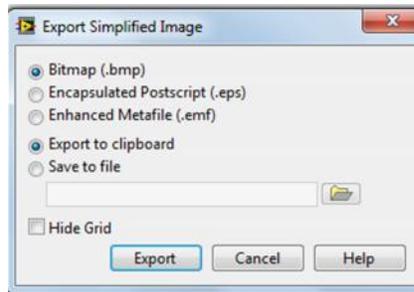
**Figura 4.36** Interfaz para Control de la Planta de Temperatura LCP

- Desde la interfaz se puede modificar la temperatura de referencia para la planta, se deben realizar 4 cambios de *Set Point* establecidos en 40, 60, 30 y 45 °C respectivamente, debe esperar de 12 a 15 minutos entre cada cambio de referencia para que la planta alcance a llegar a la temperatura deseada.
- Guardar la imagen obtenida del comportamiento de la planta, dando clic derecho sobre la gráfica, ir a la opción “*Export*” y luego a la opción “*ExportSimplifiedImage*”, ver figura 4.37.



**Figura 4.37** Extraer Imagen desde la interfaz

- En la ventana que se despliega, se debe elegir la opción “*Save to file*”, y la ubicación en la que se desea guardar la imagen obtenida, ver figura 4.38.



**Figura 4.38** Ventana que permite elegir la ubicación donde se guarda la gráfica.

Después de observar el comportamiento del sistema a diferentes cambios en el *Set Point*, se procede a comprobar el comportamiento de la planta ante un disturbio (Figura 4.32).

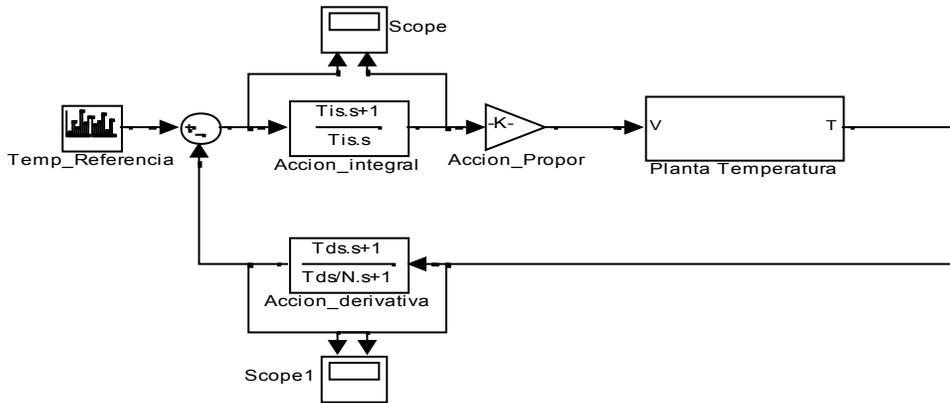
- Verificar que la planta este estable en un valor de temperatura.
  - Oprimir ON en el *Switch S7* (Figura 4.33), el cual activa el disturbio y obstruye la salida de flujo del cilindro, con el fin de aumentar la temperatura en el interior del mismo.
  - Esperar a que se genere el disturbio y que posteriormente sea controlado por el sistema de control.
  - De ser necesario se guarda la señal como se realizó anteriormente para el comportamiento de la planta ante diferentes *Set Point*.
  - Apagar el disturbio oprimiendo OFF en el *Switch S7* (Figura 4.33).
- 
- Apagar la planta, para esto remitirse a la sección 4.3.2. de esta guía.
  - Analizar el comportamiento del sistema, con la presencia del controlador PID serie industrial.

#### **4.6.2. Diseño del controlador basado en arquitectura ANFIS.**

Para el diseño del controlador basado en ANFIS, se parte del controlador PID serie Industrial, realizada en la sección 4.3.6.1 de esta guía.

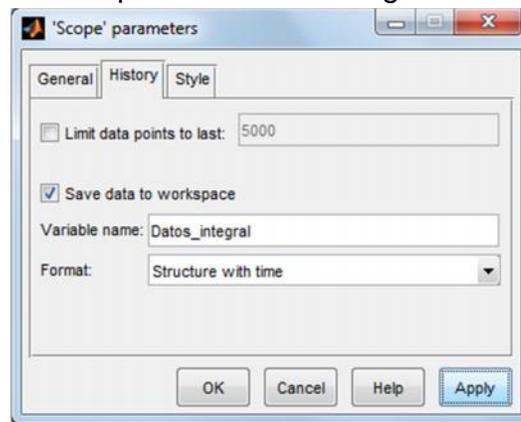
Procedimiento.

- Se continúa trabajando en la carpeta “Controlador basado en ANFIS”.
- ✓ Ubicar y abrir el archivo “SintonizacionPIDIndustrial.m”.
- ✓ Ejecutar el archivo “SintonizacionPIDIndustrial.m”
- ✓ Abrir el archivo “TomadDatos.mdl”, en el cual se realizará la toma de datos de la entrada y salida de la acción derivativa y la acción integral, con el fin de realizar posteriormente sus respectivos modelos ANFIS



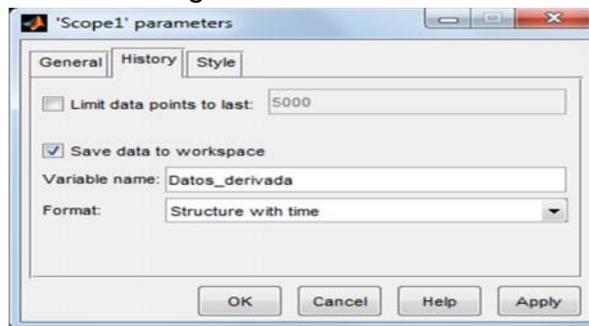
**Figura 4.39** Archivo “TomadDatos.mdl”

✓ En el bloque “scope”, dar clic en la opción *parameters* (🔧) >> *History* y verificar que se encuentren los parámetros de la figura 4.40.



**Figura 4.40** Configuración Scope

✓ De igual forma se verifica en el bloque “scope1”, opción *parameters* (🔧) >> *History*, los parámetros de la figura 4.41



**Figura 4.41** Configuración Scope1

- ✓ Establecer el tiempo de simulación en 33120 seg
- ✓ Ejecutar el archivo "TomadDatos.mdl".
- ✓ En el *workspace* de Matlab verificar que se encuentren los archivos Datos\_integral y Datos\_derivada.

Name	Value	Min	Max
Datos_derivada	<1x1 struct>		
Datos_integral	<1x1 struct>		
Den	[87.3800 1]	1	87.3800
Kc	3.9471	3.9471	3.9471
N	20	20	20
Num	0.2216	0.2216	0.2216
R	0.0500	0.0500	0.0500
Td	49.1961	49.1961	49.1961
Ti	90.5143	90.5143	90.5143
Tm	91.8490	91.8490	91.8490

**Figura 4.42** Archivos Datos\_derivada y Datos \_integral en el workspace de Matlab

- ✓ En la ventana de comandos escribir las líneas de código de la figura 4.43 con el fin de definir cada una de las matrices de los datos obtenidas del archivo "TomadDatos.mdl"

```

>> ErrorSistema = Datos_integral.signals(1,1).values;
>> Sintegral = Datos_integral.signals(1,2).values;
>> TempSalida = Datos_derivada.signals(1,1).values;
>> Sderivada = Datos_integral.signals(1,2).values;
fx >>

```

**Figura 4.43** Comandos para adquirir datos del scope de Simulink

- ✓ Posteriormente se guardan las matrices como archivos *.dat*, con los comandos de la figura 4.44

```

>> save ErrorSistema.dat ErrorSistema -ascii -tabs
>> save Sintegral.dat Sintegral -ascii -tabs
>> save TempSalida.dat TempSalida -ascii -tabs
>> save Sderivada.dat Sderivada -ascii -tabs
fx >> |

```

**Figura 4.44** Comando para guardar las matrices

## Modelado de la acción Integral por ANFIS

Para modelar la acción integral mediante ANFIS es necesario organizar los datos obtenidos del PID SERIE INDUSTRIAL en tres conjuntos indispensables, el conjunto de entrenamiento, el de chequeo y el de testeo, para esto se realiza el siguiente procedimiento.

- Abrir el archivo "Accion\_integral.m" y verificar que se encuentren las líneas de código que cargan los archivos "ErrorSistema.dat" y "Sintegral.dat"
- Ejecutar el archivo "Accion\_integral.m"
- Después en la ventana de comandos escribir "ANFISedit" y dar "Enter"
- Se generará la GUI del editor de ANFIS de la figura 4.45

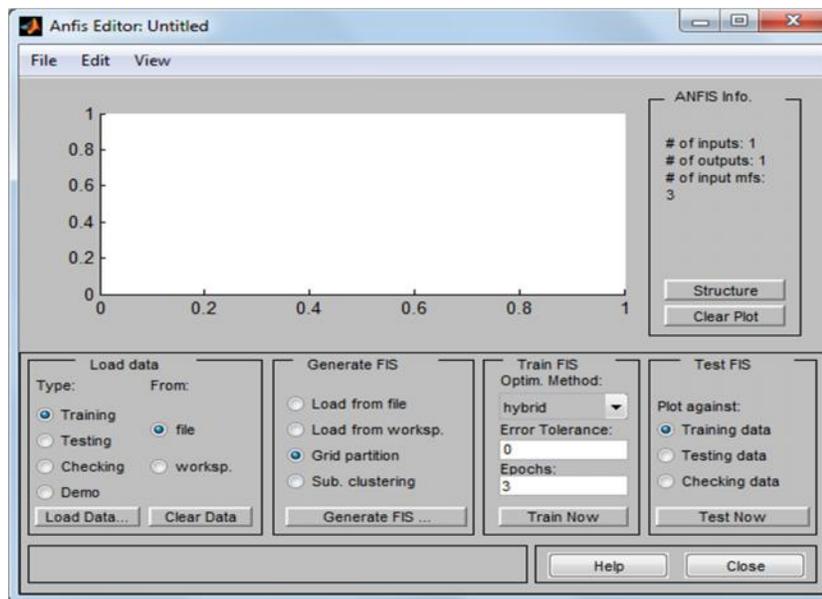
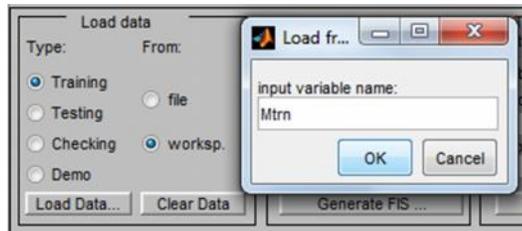


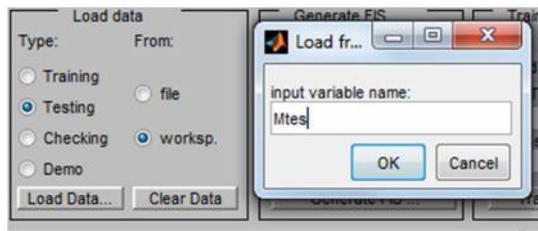
Figura 4.45 GUI ANFISedit

- **Configuración del ANFISedit para el modelo de la acción integral.**
    - ✓ Carga de datos en el editor
- En "load data" se cargan los tres conjuntos que hacen parte del entrenamiento, de la siguiente forma.
- Para cargar los datos de entrenamiento en "load data" escoger *Training* y *worksp.* y dar clic en *load data*, para ejecutar la ventana donde se escribe la matriz a utilizar, en este caso *Mtr* como muestra la figura 4.46, y finalmente dar *ok*.



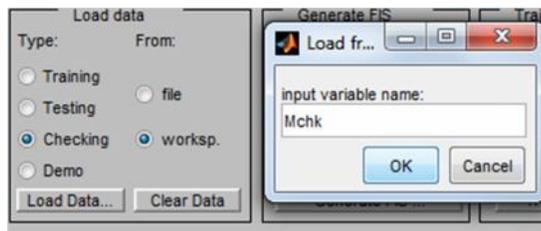
**Figura 4.46** Carga de datos de entrenamiento

- Para los datos de testeo, escoger *Testing* y *worksp.* y dar clic en load data, después escribir *Mtr* para llamar a la matriz de testeo y finalmente *OK*, como muestra la figura 4.47.



**Figura 4.47** Carga de datos de testeo

- Para los datos de chequeo, escoger *checking* y *worksp.* y dar clic en load data donde se escribe *Mchk* para llamar a la matriz de chequeo y finalmente dar *OK* como muestra la figura 4.48.



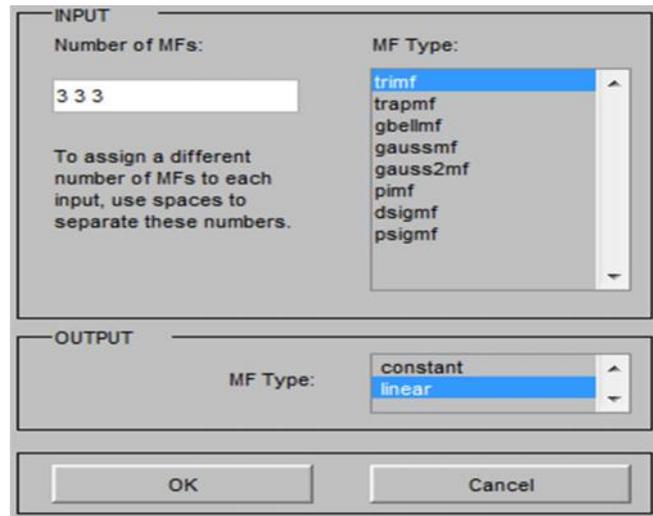
**Figura 4.48** Carga de datos de chequeo

- ✓ Generación del sistema difuso

Para crear el sistema difuso del modelo de la acción integral se verifica que en “*generate FIS*” esté seleccionada la opción “*GridPartition*”, posteriormente se da clic en “*Generate Fis*”, en la ventana que se genera se realiza la siguiente configuración (ver figura 4.49)

- Para los parámetros de entrada, se establece 3 como el número de MFs, para cada una de las entradas y triangular (trimf) como el tipo de MF.

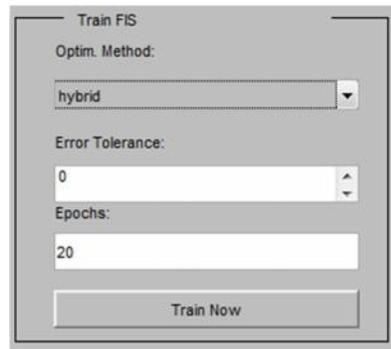
- Para los parámetros de salida se escoge lineal.
- Finalmente se da clic en *OK* aceptando los parámetros configurados.



**Figura 4.49** Parámetros del Sistema Difuso para el modelo ANFIS de la acción integral.

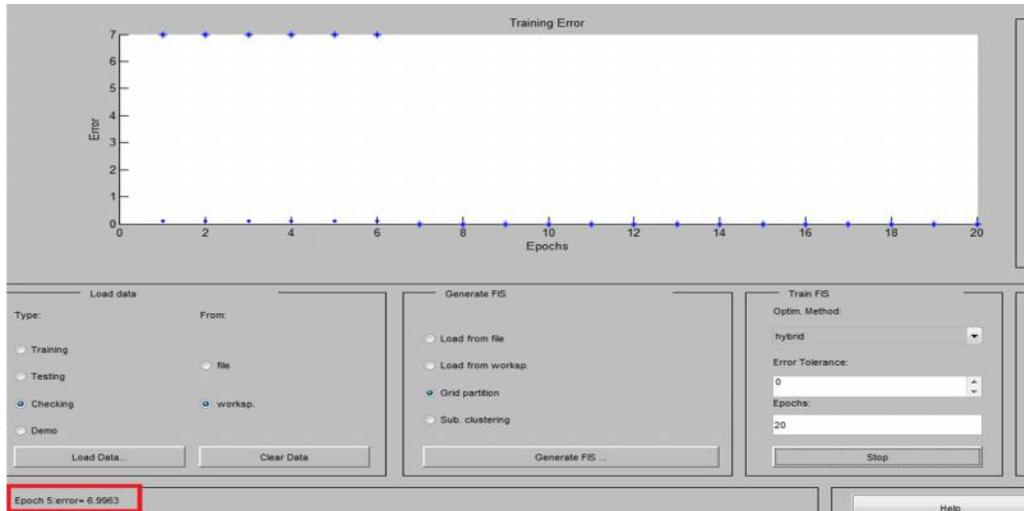
- ✓ Entrenamiento del sistema difuso

Para el entrenamiento del FIS se escoge como método de entrenamiento “hibrido” y 20 como el número de épocas de entrenamiento, para iniciar el entrenamiento dar clic en “*train now*”(ver figura 4.50).



**Figura 4.50** Parámetros del entrenamiento del FIS para el modelo ANFIS de la acción integral

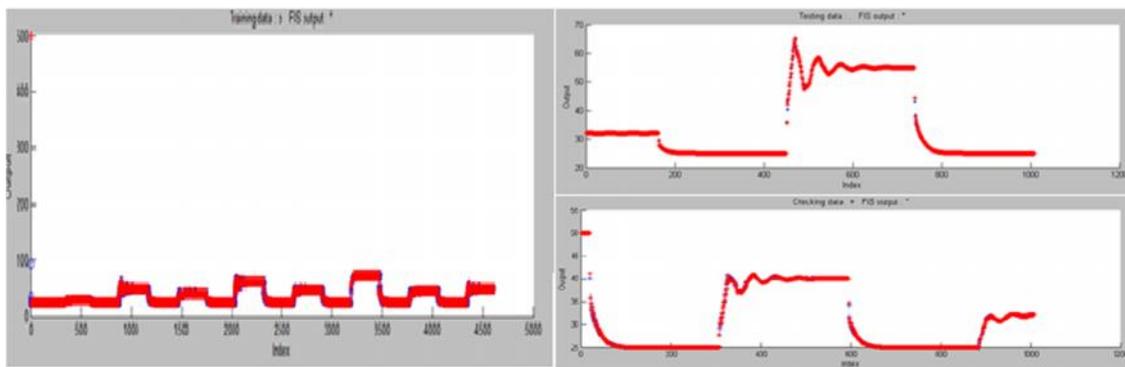
En la GUI de ANFIS se observará el comportamiento de la salida del modelo que se está generando con respecto a los datos ingresados, de igual forma al lado derecho aparecerá el error del entrenamiento en cada época, (figura 4.51).



**Figura 4.51** GUI de ANFIS durante el entrenamiento del sistema difuso

✓ Respuesta del ANFIS

Cuando se termina el entrenamiento se activa la opción "test now" con la cual se verifica el comportamiento del sistema difuso entrenado con respecto a los datos de entrenamiento, testeo o chequeo. Para determinar que el modelo es el apropiado la respuesta del sistema frente a los conjuntos de datos deben de ser muy similares como se muestra en la figura 4.52



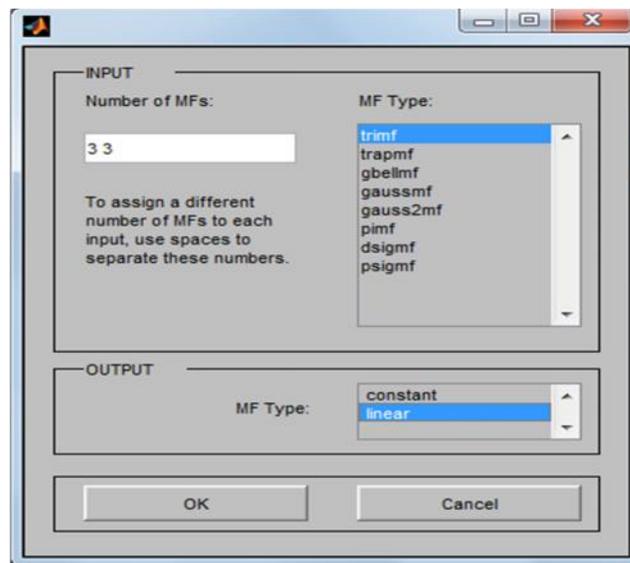
**Figura 4.52** Comportamiento del modelo realizado por ANFIS frente a los 3 conjuntos de datos.

Finalmente se guarda el sistema difuso obtenido con el nombre **ANFIS\_INT**, para esto se da clic en *File>>export>>To file* en la carpeta "**controlador basado en ANFIS**", finalmente verificar que el archivo se encuentra en el *current folder* de Matlab.

## Modelado de la acción derivativa por ANFIS

- Abrir el archivo "Accion\_derivativa.m" y verificar que se encuentren las líneas de código que cargan los archivos "TempSalida.dat" y "Sderivada.dat".
  - Ejecutar el archivo "Accion\_derivativa.m"
  - Después en la ventana de comandos escribir "ANFISedit" y dar "Enter"
- ✓ Configuración del ANFISedit para el modelo de la acción derivativa.

La configuración de los parámetros en el editor ANFIS para el modelo de la acción derivativa se realiza de igual forma que para la acción integral. Sin embargo es importante tener en cuenta que el conjunto de datos de entrada en esta ocasión son dos por lo tanto la ventana de "Generar Fis", se configura de la siguiente forma.

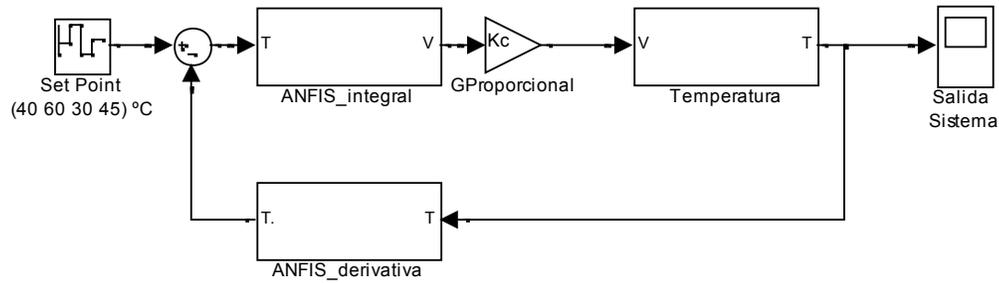


**Figura 4.53** Ventana Generar FIS modelo ANFIS Acción derivativa

Al obtener el modelo de la acción derivativa, se guarda con el nombre de **ANFIS\_DER** dando clic en, en la carpeta "**controlador basado en ANFIS**", finalmente verificar que el archivo se encuentra en el *current folder* de Matlab

Después de obtener los modelos de la acción integral y derivativa, se prueba su funcionamiento en simulación, para esto se abre el archivo "Simulacion\_ANFIS.m" donde se requiere copiar el valor del parámetro  $K_c$  hallado en el PID serie industrial y comprobar que los nombres de los modelos ANFIS sean los adecuados.

Finalmente se ejecuta el archivo "ANFIS\_simulacion.mdl", en el "Salida Sistema" se verifica la respuesta del controlador.

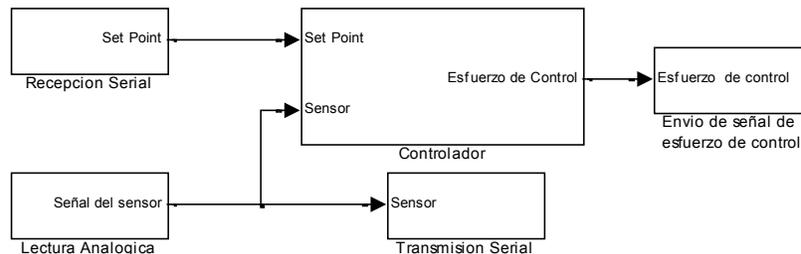


**Figura 4.54** Archivo ANFIS\_simulacion.mdl

### Implementación en la planta de temperatura del LCP

Cuando se tenga una buena respuesta del controlador en simulación se realiza el siguiente procedimiento para implementarlo en la planta de temperatura.

- Encender la planta, como se muestra en la sección 4.3.1.
- abrir el archivo ANFIS\_para\_arduino.mdl, y descargarlo en la tarjeta como se explicó en la sección 4.3.3



**Figura 4.55** Archivo para descargar en arduino

- Una vez programada la tarjeta, se debe abrir y ejecutar la interfaz de usuario “InterfazParaControl”, que está ubicada en la carpeta “Controlador basado en ANFIS”, ver figura 4.36.
- Desde la interfaz se puede modificar la temperatura de referencia para la planta, se deben realizar 4 cambios de *Set Point* establecidos en 40, 60, 30 y 45 °C respectivamente, debe esperar de 12 a 15 minutos entre cada cambio de referencia para que la planta alcance a llegar a la temperatura deseada.

- Guardar la imagen obtenida del comportamiento de la planta, dando clic derecho sobre la gráfica, ir a la opción “*Export*” y luego a la opción “*ExportSimplifiedImage*”, ver figura 4.37.

En la ventana que se despliega, se debe elegir la opción “*Save to file*”, y la ubicación en la que se desea guardar la imagen obtenida, ver figura 4.38.

Después de observar el comportamiento del sistema a diferentes cambios en el *Set Point*, se procede a comprobar el comportamiento de la planta ante un disturbio (Figura 4.32).

- Verificar que la planta este estable en un valor de temperatura.
- Oprimir ON en el *Switch S7* (Figura 4.33), el cual activa el disturbio y obstruye la salida de flujo del cilindro, con el fin de aumentar la temperatura en el interior del mismo.
- Esperar a que se genere el disturbio y que posteriormente sea controlado por el sistema de control.
- De ser necesario se guarda la señal como se realizó anteriormente para el comportamiento de la planta ante diferentes *Set Point*.
- Apagar el disturbio oprimiendo OFF en el *Switch S7* (Figura 4.33).
- Apagar la planta, para esto remitirse a la sección 4.3.2 de esta guía.
- Analizar el comportamiento del sistema, con la presencia del controlador modelado por ANFIS.

## CONCLUSIONES

- La lentitud en la dinámica de la planta, se convierte en una desventaja al momento de realizar aplicaciones sobre ella, como toma de datos, implementación de diferentes métodos de control, entre otras.
- El utilizar Simulink como interfaz de programación para la tarjeta, facilitó el desarrollo del trabajo, debido al conocimiento previamente adquirido y a su fácil manejo comparado con otros lenguajes de programación.
- Dentro de las tarjetas desarrolladas por Arduino hasta el momento, la tarjeta mega 2560 R3 es la de mayor capacidad, sin embargo, la implementación en la cual se utiliza como dispositivo de control es limitada, debido a su baja capacidad de memoria interna, la baja resolución de los módulos analógicos-digitales y la necesidad de hardware externo para la mayoría de puertos seriales.
- Matlab R2012.a es la primera versión que trae la herramienta para la programación de Arduino por medio de Simulink, por lo que no cuenta con una buena herramienta para graficar las señales capturadas en tiempo real, por lo que se hace uso de *LabVIEW* para realizar la interfaz hombre .
- Para el modelo del controlador por ANFIS se utilizaron los datos de un PID serie industrial, dado que el esfuerzo de control que produce es más suave que el de los controladores PID paralelos, lo que facilita el entrenamiento del sistema.
- Para el entrenamiento de un sistema ANFIS se debe tener en cuenta que los datos utilizados, tengan la mayor cantidad de información posible, además ANFIS sólo puede modelar relaciones algebraicas, de modo que es necesario introducir dinámica mediante bloques externos.
- El controlador modelado por ANFIS no presentó grandes ventajas al compararlo con un PID serie industrial, sin embargo presenta un esfuerzo de control rápido, lo cual se ve reflejado en un mejor tiempo de respuesta, y una compensación rápida frente a los disturbios. Además, abre un espacio para el estudio y aplicación de técnicas inteligente en el programa de Ingeniería en .Automática Industrial.



## BIBLIOGRAFÍA

- Alfaro, V. (1993). ¿Son todos los controladores PID iguales? *Ingenieria, Revista de la Universidad de Costa Rica Vol3*.
- Alfaro, V. (2001). Identificación de procesos sobreamortiguados. *Ingenieria*.
- Alfaro, V. (2002). Ecuaciones para Controladores PID Universales. *Ingenieria*.
- Alfaro, V. (2002). Metodos de Sintonizacion de Controladores PID que Operan como Reguladores. *Ingenieria*.
- Alimuddin, Seminar, K., Made Subrata, I., Nomura, N., & Sumiati. (2012). Temperature Control System in Closed House for Broilers Based on ANFIS. *University of Sultan Ageng Tirtayasa*.
- Alippor, M., Zeinali, M., & Sadoghi, H. (2009). Fuzzy Temperature Control in a Batch Polymerization Reactor Using ANFIS Method. *International Journal of Engineering and Technology*.
- Amestegui Moreno, M. (2001). Apuntes de Control PID. *Universidad Mayor de San Andres*.
- Arduino. (s.f.). *Arduino*. Recuperado el 13 de Septiembre de 2012, de <http://arduino.cc/en/Main/arduinoBoardMega2560>
- Arduino. (s.f.). Arduino Mega 2560 R3. <http://arduino.cc/en/Main/arduinoBoardMega2560>.
- Arduino. (s.f.). *Sitio web Arduino*. Recuperado el 13 de Septiembre de 2012, de <http://arduino.cc/en/Main/arduinoBoardMega2560>
- Arias, J., & Ribon, S. (2009). Pornostico de Precio de Contrator de la Energia en el Largo Plazo a Través del Metodo ANFIS. *Universidad NAcional de Colombia*.
- Aros, N., & Mardones, J. (2010). 1er. Congreso Iberoamericano de Ingenieria de Proyectos. *Control Predictivo no lineal de un horno rotatorio para la produccion de carbon activo mediante ANFIS*. Antofagasta.
- Ayata, T., Cam, E., & Yildis, O. (2007). Adaptative Neuro-Fuzzy Inference Systems (ANFIS) Application to investigate potential use of natural ventilation in new building designs in Turkey *Energy Conversion and Management* 48.

- Han, M., & Zhao, Y. (2011). Dynamic Control Model of BOF Steelmaking Process base on ANFIS and Robust Relevance Vector. *Machine Expert systems with applications* 38.
- Jang, R. (1993). ANFIS Adaptive Network Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, Vol 23 No 3.
- Jang, R. (1993). ANFIS Adaptive Network Based Fuzzy Inference System . *IEEE Transactions on Systems, Man and Cybernetics*.
- Jantzen, J. (2002). *Foundations of Fuzzy Control*. Technical University of Denmark.
- Laboratorio, P. a. (2010). Reconocimiento de la planta de Temperatura. Popayán, Cauca, Colombia.
- Li, T., & Wang , N. (2011). Temperature Field Analysis and Process Control Strategies for MgO Single Crystal Production Using Adaptive Neuro – Fuzzy Inference System. *The Open Materials Science Journal*.
- Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studics*, Vol 7, Issue 1, 1-13.
- Mazzones, V. (2002). Controladores PID. *Universidad Nacional de Quilmes*.
- Mendoza, J., & Mazo , A. (2009). Analisis del Modelo ANFIS en el pronóstico de un Titulo de Renta Variable. *Universidad Nacional de Colombia* .
- Moreno , J. (2009). Hydraulic Plant Generation Forecasting in Colombia Power Market using ANFIS Energy Economics 31.
- Mote, T., & Lokhande, S. (2012). Temperature Control System Using ANFIS. *International Journal of Soft Computing and Engineering*.
- Navas, R. (2000). *Diseños y Aplicacion de Controladores Neuro-Difusos Integrados en Modo Mixto de Alta Complejidad*. Malaga: Universidad de Malaga.
- Nichols, N., & Zigler, J. (1942). Optimum Settings for Automatic Controls. *ASME transactions*, vol 64, 759-768.
- Ravi, S., Sudha, M., & Balakrishnan, P. (2011). Design of Intelligent Self-Tuning GA ANFIS Temperature Controller for Plastic Extrusion System.
- Rojas, L. F. (2007). Controladores PID Comerciales. *Universidad de Costa Rica, Facultad de Ingenieria, Escuela de Ingenieria Electrica*.

- Sabadía, A. V. (1995). En *Estadística Descriptiva e Inferencial* (pág. 474). La Mancha: Compobell, S.L.Murcia.
- Salas, R. (2004). *Redes Neuronales Artificiales*. Universidad de Valparaiso.
- Salunkhe, A. (2010). Compound – Fuzzy Inference System for Temperature Controller. *International Journal of Electronics Engineering*.
- Sanzs, R. (1990). *Arquitectura de Control Inteligente de Procesos*. The Autonomus Systems Laboratory.
- Skoog, D., Crouch, S., & Holler, J. (2008). En *Principios de Analisis instrumental* (pág. 986). México D.C: Brooks cole.
- Smith, C. (1972). En *Digital Computer Process*. Scranton: International Textbook.
- Smith, C., & Corripio, A. (1991). *Control Automatico de Procesos Teoria y Practica*. USA.
- Soyguder, S., & Alli, H. (2009). An Expert System for the Humidity and Temperature Control in HVAC system using ANFIS and Optimization with Fuzzy Modeling Approach.
- Soyguder, S., & Karakose, M. (2008). Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system. *Expert. Systems with applications* 36.
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems man and cybernetics*, Vol 15, No 1, 116-132.
- Tan, Z., Quek, T., & Cheng, P. (2011). Stock Trading with Cycles A Financial Application of ANFIS and Reinforcement Learning Expert Systems with Applications 38.
- Tavoosi, J., Alaei, M., & Jahani, B. (2011). Temperature Control of Water Bath by using Neuro – Fuzzy Controller.
- Tsukamoto, Y. (1979). An approach to fuzzy reasoning method. *Advances in fuzzy set theory an applications*, 137-149.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, Vol 8, Issue 3, 338-353.