



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Laboratorio de Sistemas Inteligentes para la Automatización

Simulación de un sistema distribuido de supervisión y control con base en un sistema cliente servidor TCP/IP.

TESIS

QUE PARA OBTENER EL GRADO DE:

**MAESTRÍA EN CIENCIAS EN INGENIERÍA DE
CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES**

P R E S E N T A:

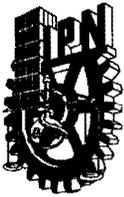
Ing. Jesús Gómez Sánchez

Directores de tesis:
Dr. José de Jesús Medel Juárez
Dr. Luis Pastor Sánchez Fernández



Mexico, D.F.

Mayo 2014



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 6 del mes de junio de 2014 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Simulación de un sistema distribuido de supervisión y control con base en un sistema cliente servidor TCP/IP"

Presentada por el alumno(a):

Gómez

Apellido paterno

Sánchez

Apellido materno

Jesús

Nombre(s)

Con registro:

B	1	2	1	0	2	5
---	---	---	---	---	---	---

aspirante de: **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. José de Jesús Medel Juárez

Dr. Luis Pastor Sánchez Fernández

Dr. Sergio Suárez Guerra

Dr. Oleksiy Pogrebnyak

Dr. José Juan Carbajal Hernández

M. en C. Romeo Urbieta Parrazales

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Raúl Alfonso Villa Vargas



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F el día 6 del mes Junio del año 2014, el (la) que suscribe Ing. Jesús Gómez Sánchez alumno (a) del Programa de MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES con número de registro B121025, adscrito a Laboratorio de Sistemas Inteligentes para la Automatización, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Luis Pastor Sánchez Fernández y Dr. José de Jesús Medel Juárez y cede los derechos del trabajo intitulado *Simulación de un sistema de supervisión y control con base en un sistema cliente servidor TCP/IP*, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección jesus.g.1983@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Jesús Gómez Sánchez
Nombre y firma



Resumen

En los procesos industriales de fabricación en diferentes sectores de producción como papelería, químico, cementero, agua, petróleo, siderurgia, electricidad, etc., es necesario realizar el control de los procesos en forma continua y también en forma remota. Actualmente las necesidades de supervisión y control de algunos de estos procesos de producción en México, son cubiertas por empresas como Honeywell, Yokogawa, Emerson, etc. estas son solo algunas de las empresas que proveen sistemas de control distribuido para cubrir las necesidades de supervisión y control en los diferentes procesos de producción en México; es por esta razón que se requiere de opciones tecnológicas nacionales que puedan cubrir estas necesidades al día de hoy. En este trabajo se presenta un sistema de supervisión y control, implementando algoritmos de control con Lógica Difusa, así como técnicas de Control Clásico. El sistema de supervisión y control está constituido por una computadora servidor que es la encargada de monitorear en forma continua lo que sucede en el proceso ó los procesos que se ejecutan en los clientes, siendo el servidor el centro de control del sistema.

La comunicación entre el servidor y los clientes se hace mediante el protocolo TCP/IP. En cada cliente se simula un proceso, y el servidor pide datos a los clientes, procesa esta información y también modifica los parámetros de control que los clientes están ejecutando. De esta forma se tiene un sistema de supervisión y control que usa una arquitectura cliente servidor TCP/IP para comunicar los diversos nodos con el centro de control. Se simuló un proceso de nivel de dos tanques que están interconectados y se desarrolló un controlador tipo Mamdani con dos entradas y una salida para controlar el nivel en el tanque 2 regulando el flujo en la entrada del tanque 1, se simuló la respuesta del sistema para funciones de membresía triangulares y funciones de membresía del tipo Gaussiana, se compara las respuestas de ambos controladores y se integran en un sistema cliente servidor que se comunica con el protocolo TCP/IP, en donde el servidor pide y envía información a cada uno de los clientes, de esta forma se logra tener un sistema que monitorea el proceso de forma continua y también puede controlar los parámetros de éste enviando información de que valores son los que tienen en el proceso.





Abstract

In the industrial processes of production in different sectors of production such as paper, cement, water, oil, steel, electricity, etc. It is necessary to carry out the control of processes on a continuous and remote basis. Currently the requirements for supervision and control of some of these processes of production in Mexico, are covered by companies such as Honeywell, Yokogawa, Emerson, etc. These are just some of the companies that provide distributed control systems to meet the needs of supervision and control in the different sectors of production in Mexico; that national technological options that can meet these needs are required the day of today.

This thesis presents a system for monitoring and controlling, and has implemented control algorithms with fuzzy logic as well as classic control techniques. The control system is composed by a computer server that is responsible for monitoring continuously what happens in the process(es) that are running on the client, being the server the system control center.

The communication between the server and clients is done by the TCP/IP protocol, each client is simulating a process, and the server requests data to clients, this information is processed and it also modifies the control of parameters that the clients are running. In this way you have a monitoring and control system that uses a client server TCP/IP architecture to communicate various nodes with the control center. A simulated two tanks level process that are interconnected, and are developing a Mamdani controller type with two inputs and one output to control the tank level 2 by regulating the flow in the entry of the tank 1, was simulated the system response for membership functions to triangular and membership functions of Gaussian type, it compares the responses of both controllers and are integrated into a client server system that communicates with the TCP/IP protocol, where the server requests and sends out information on a continuous basis to each of the clients, and also is able to control the parameters of the process by sending information to each client whose values must have in the process.





Agradecimientos

A Dios.
por darme la oportunidad de vivir y poder cumplir un objetivo en la vida.

A mi familia.
por estar conmigo en las buenas y en las malas de forma incondicional.

Al Instituto Politécnico Nacional
por regalarme toda una formación profesional.

A mis asesores
por regalarme su tiempo, experiencia y conocimiento en el desarrollo de este trabajo.





Índice general

Resumen	I
Abstract	II
Agradecimientos	III
Lista de figuras	VII
Lista de tablas	X
Glosario de siglas y términos.	XI
1. INTRODUCCIÓN	1
1.1. Motivación y antecedentes	1
1.2. Planteamiento del problema	4
1.3. Justificación	4
1.4. Hipótesis	4
1.5. Objetivos	5
1.5.1. Objetivo general	5
1.5.2. Objetivos específicos	5
1.6. Alcance	5
1.7. Metodología y desarrollo de la investigación	5
1.8. Contribuciones del trabajo	6
1.9. Estructura de la tesis	6
2. ESTADO DEL ARTE	8
2.1. Desarrollo de los sistemas de control	8
2.2. Red industrial	9
2.2.1. Niveles de una Red Industrial	9
2.2.2. Buses de campo	10
2.2.3. Sistemas de Control Distribuido	12
2.2.4. Sistemas SCADA	13
2.2.5. Diferencias entre SCD y SCADA.	14
2.3. Algoritmos de control.	15
2.4. Resumen del capítulo.	17





3. MARCO TEÓRICO	18
3.1. Tipo de controladores.	18
3.1.1. Control clásico.	18
3.1.2. Sistemas de control en lazo abierto.	18
3.1.3. Sistemas de control en lazo cerrado.	19
3.1.4. Respuesta típica de un sistema de control	19
3.1.5. Esquema de control PID.	21
3.1.6. Método gráfico de Ziegler and Nichols	22
3.2. CONTROL DIFUSO	24
3.2.1. Lógica difusa.	24
3.2.2. Teoría de conjuntos difusos.	24
3.2.3. El controlador difuso.	26
3.2.4. Fusificación.	27
3.2.5. Defusificación.	28
3.3. REDES DE COMPUTADORAS Y CONECTIVIDAD	31
3.3.1. Red de computadoras	31
3.3.2. Métodos de transmisión en redes LAN.	33
3.3.3. Capas de Protocolos y sus Servicios	33
3.4. PROTOCOLO TCP/IP.	35
3.5. Resumen del capítulo.	36
4. ARQUITECTURA Y FUNCIONALIDAD DEL SISTEMA DIS- TRIBUIDO.	37
4.1. Funciones básicas del sistema	37
4.2. Principales componentes	37
4.3. Funcionalidad del sistema	38
4.3.1. Funcionamiento del servidor	39
4.3.2. Funcionamiento del cliente.	40
4.4. Restricciones temporales.	42
4.5. Resumen del capítulo.	43
5. CONTROLADOR DIFUSO PARA UN PROCESO DE NIVEL SI- MULADO (PNS).	47
5.1. El problema de dos tanques interconectados.	47
5.2. Lazo de control.	49
5.3. Controlador difuso.	50
5.4. Simulación.	56
5.5. Resumen del capítulo.	58
6. INTEGRAR EL CONTROL DIFUSO DEL PNS EN UN SISTE- MA CLIENTE SERVIDOR DE VARIOS NODOS.	59
6.1. Descripción del sistema de supervisión y control.	59
6.1.1. Requerimientos del servidor.	60
6.1.2. Requerimientos del cliente.	60
6.2. Desarrollo del sistema cliente servidor.	60





6.3. Interfaz del proceso de nivel.	61
6.4. Interfaz de supervisión y control.	63
6.5. Resumen del capítulo.	66
7. CONTROLADOR PID AUTOAJUSTABLE PARA UN PROCE-	
SO SIMULADO (PS).	67
7.1. Controlador difuso como sintonizador de controlador PID.	67
7.2. Sintonización del controlador PID	68
7.3. Esquema de control	71
7.4. Desarrollo del controlador	71
7.5. Simulación PID autoajustable	75
7.6. Resumen del capítulo.	75
8. EVALUACIÓN DE RESULTADOS.	77
8.1. Diagrama básico de control	77
8.2. Pruebas realizadas	78
8.2.1. Prueba de lazo abierto	78
8.2.2. Prueba del lazo cerrado de control.	79
8.2.3. Prueba de los diferentes algoritmos de control.	79
8.3. Comparación con el Estado del Arte.	81
Conclusiones y trabajo futuro.	85
Conclusiones.	85
Trabajo futuro.	87
A.	88
A.1. Función de transferencia PNS	88
A.2. Reglas de Inferencia PNS	89
A.3. Superficies de control para PID autoajustable	90
A.4. Características de Buses y Protocolos	95
A.4.1. Estructura de un Segmento TCP.	96
A.4.2. Estructura del Datagrama IP	97
A.5. Resultados de la simulación.	99
Referencias Bibliográficas	108



Índice de figuras

2.1. Niveles de una Red Industrial	10
2.2. Es un esquema cortesía del proveedor SMAR [1].	13
2.3. Esquema cortesía del proveedor ABB [2].	14
3.1. Sistema de control en lazo abierto	19
3.2. Sistema de control en lazo cerrado	19
3.3. Respuesta al escalón unitario	20
3.4. Método gráfico Ziegler and Nichols	23
3.5. Función de pertenencia [3].	25
3.6. Función de pertenencia triangular y trapezoidal	26
3.7. Controlador difuso de dos entradas y una salida	27
3.8. Centro del área	29
3.9. Bisección del área de tres conjuntos triangulares	30
3.10. Medio de los máximos	31
3.11. Red de Computadoras	32
3.12. Red de área local (LAN,Local Area Network)[4].	32
3.13. Red de área extensa (WAN,Wide Area Network)	32
4.1. Arquitectura Cliente Servidor TCP/IP.	38
4.2. Diagrama de flujo servidor.	40
4.3. Diagrama de flujo identificación de las conexiones.	41
4.4. Diagrama de flujo escribir de servidor a cliente.	42
4.5. Diagrama de flujo leer desde servidor a cliente.	43
4.6. Diagrama de flujo cliente.	44
4.7. Diagrama de flujo lectura TCP/IP cliente.	45
4.8. Diagrama de flujo escritura TCP/IP cliente.	46
5.1. Sistema de dos tanques interconectados.	48
5.2. Lazo cerrado de control (FLC).	50
5.3. Velocidad de cambio de nivel de líquido en el tanque 2.	51
5.4. Error (Diferencia entre la referencia y el valor real del nivel de líquido).	51
5.5. Incremento de la señal de control U.	52
5.6. Simulación del controlador (FLC).	53
5.7. Velocidad de cambio de nivel de líquido en el tanque 2 (Gaussiana).	54





5.8. Error (Diferencia entre la referencia y el valor real del nivel de líquido)(Gaussiana).	54
5.10. Simulación del controlador (FLC)(Gaussiana).	55
5.9. Incremento de la señal de control U (Gaussiana).	55
5.11. Respuesta al escalón (Minimum) FLC.	57
5.12. Respuesta al escalón (Product T-norm) FLC.	58
6.1. Interfaz proceso de nivel cliente.	62
6.2. Interfaz proceso de nivel cliente "gráfica de parámetros en tiempo real".	62
6.3. Interfaz proceso de nivel cliente "control de parámetros del controlador PID".	63
6.4. Interfaz proceso de nivel cliente "selector de tipo de controlador". . .	63
6.5. Interfaz proceso de nivel cliente "control de la referencia".	63
6.6. Interfaz de supervisión y control servidor.	64
6.7. Interfaz de supervisión y control servidor "Nodos conectados".	65
6.8. Interfaz de supervisión y control servidor "valor de referencia propuesto".	65
6.9. Interfaz de supervisión y control servidor "valor de referencia actual en el cliente".	65
6.10. Interfaz de supervisión y control servidor "recepción Rx y transmisión Tx".	66
7.1. Método gráfico Ziegler and Nichols	70
7.2. Esquema de control para el ajuste de ganancias del PID.	72
7.3. Error (diferencia entre la referencia y el valor real del nivel de líquido en tanque 2)	72
7.4. Velocidad de cambio del nivel de líquido en tanque 2.	73
7.5. Ganancia proporcional.	73
7.6. Tiempo integral.	74
7.7. Tiempo derivativo.	74
7.8. Respuesta al escalón PID autoajutable	76
8.1. Diagrama de bloques de un sistema de control [5].	77
8.2. Resultados de simulación en lazo abierto del modelo en S y en Z respectivamente.	78
8.3. Resultados de simulación en lazo cerrado de control en S y en Z respectivamente.	79
8.4. Resultados de la simulación.	80
8.5. Resultados de la simulación del controlador PID y el controlador fuzzy PID [14].	82
8.6. Resultados de la simulación del controlador PID y un controlador fuzzy PD+PID [47].	83
8.7. Resultados de la simulación del controlador PID y un controlador fuzzy PID [51].	83





8.8. Resultados de la simulación del controlador PID, un controlador FLC y un controlador STFPID [25].	84
A.1. Superficie de control para la ganancia proporcional K_p	92
A.2. Superficie de control para el tiempo integral T_i	93
A.3. Superficie de control para el tiempo derivativo T_d	93
A.4. Estructura de un Segmento TCP.	96
A.5. Estructura de un Datagrama IP.	97



Índice de Tablas

2.1. Características de los sistemas SCADA y los sistemas DCS	15
3.1. Reglas de sintonización de Ziegler and Nichols basadas en la respuesta al escalón unitario de la planta.	23
3.2. La pila de protocolos TCP/IP	34
5.1. Motor de inferencias del controlador (FLC).	52
6.1. Variables de proceso a supervisar.	60
7.1. Reglas de sintonización de Ziegler and Nichols basadas en la respuesta al escalón unitario de la planta.	71
7.2. Motor de inferencias del controlador (FLC) para el autoajuste del PID.	75
8.1. Resultados	80
8.2. Resultados del estado del arte	84
A.1. Características de algunos Buses y Protocolos	95
A.2. Resultados de la simulación	108





Glosario de siglas y términos.

CONJUNTO es una colección de elementos, cada elemento tiene características que lo relaciona con todos los elementos del conjunto pero cada elemento del conjunto es irrepetible.

CONJUNTO DIFUSO es un conjunto en donde cada uno de los elementos tiene un grado de pertenencia asociado, dicho grado de pertenencia toma un valor en el rango de $[0, 1]$, esto quiere decir que cada elemento tiene una etiqueta que indica el grado en el que el elemento pertenece al conjunto.

DEFUSIFICAR establecer una relación entre valores difusos dentro de un universo de discurso y valores no difusos.

FLC del inglés Fuzzy Logic Controller hace referencia a un controlador cuya construcción está basada en lógica difusa.

FUNCIÓN es una regla de correspondencia que asocia a los elementos de dos conjuntos, en la cual a cada elemento del primer conjunto se le asocia un solo elemento del segundo conjunto.

FUNCIÓN DE MEMBRESÍA es una regla de correspondencia que asocia los elementos de un conjunto con un conjunto de grados de pertenencia, en la cual a cada elemento del conjunto se le asocia uno y solo un elemento del conjunto de grados de pertenencia.

FUSIFICAR establecer una relación entre puntos de entrada no difusos al sistema y sus correspondientes valores difusos dentro del universo de discurso.

IHM Interfaz Hombre Maquina y también se suele encontrar como HMI por sus siglas en inglés (Human Machine Interface) y se usa para referirse a la interacción entre humanos y maquinas.





LAZO ABIERTO se refiere a que en un sistema de control la salida de la planta es medida para compararse con el valor de referencia.

LAZO CERRADO se refiere a que en un sistema de control el valor de salida de la planta es medido para compararse con la referencia y obtener un valor de error.

MOTOR DE INFERENCIAS se llama motor de inferencias al sistema que interpreta las reglas if-then de una base de reglas, con el fin de obtener los valores de salida a partir de los actuales valores de las variables lingüísticas de entrada al sistema.

MUESTREO técnica para la selección de un valor a partir de una señal continua.

PC del inglés Personal Computer, que significa computadora personal, y se usa como término genérico para referirse a un sistema equipado para cumplir tareas de la informática moderna.

PLC son las siglas utilizadas para referirse a un controlador lógico programable, también hace referencia a las siglas en inglés programing logic controller.

PNS en este trabajo se adoptaron estas siglas para referirse a un proceso de nivel simulado.

PROTOCOLO DE COMUNICACIÓN es un conjunto de reglas y normas que permiten que dos o más entidades de un sistema se comuniquen entre ellos, para transmitir información por medio de cualquier tipo de variación de un magnitud física.

RT del inglés Real Time ó tiempo real significa que el correcto funcionamiento de un sistema no sólo depende del resultado lógico que devuelve la computadora sino que también depende del tiempo en el que se da este resultado, es decir que la reacción a los eventos externos debe ocurrir durante su evolución.

RTU del inglés Remote Terminal Unit, que significa unidad de transmisión remota y es usado para referirse a un dispositivo capas de información de un proceso y enviarla a un sitio remoto donde se procese, generalmente este sitio es una sala de control donde se visualiza la información enviada por el UTR.



Rx el termino Rx hace referencia a la recepción en telecomunicaciones.

SCADA del inglés Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos) es un sistema el cual permite supervisar y controlar procesos industriales a distancia; también provee de toda la información que se genera en dichos procesos.

SCD por sus siglas hace referencia a un Sistema de Control Distribuido, también es conocido por sus siglas en inglés DCS (Distributed Control System), es un sistema de control aplicado a procesos industriales en las grandes industrias.

Tarjeta I/O éste término suele usarse para referirse a un dispositivo electrónico el cual permite la comunicación entre un sistema de procesamiento de información, como un ordenador y el mundo exterior.

Tx el termino Tx hace referencia a transmisión en telecomunicaciones.

VARIABLE LINGÜÍSTICA se le denomina variable lingüística a aquella que puede tomar por valor palabras del lenguaje natural tales como: mucho, poco, positivo, negativo etc., estas palabras desempeñan el papel de etiqueta en un conjunto difuso.



Capítulo 1

INTRODUCCIÓN

1.1. Motivación y antecedentes

Los conceptos de automatización tienen sus orígenes en la Revolución Industrial. Los elementos mediante los cuales se llevan a cabo las decisiones de control eran elementos mecánicos y electromecánicos con el advenimiento de la electricidad (motores, relés, temporizadores, contadores). Esto tenía el problema que los tableros de control llegaron a tomar grandes tamaños según se lograron automatizaciones más complejas.

En los años 50 del siglo pasado con la aparición de la electrónica comienzan a utilizarse los semiconductores con lo que se reduce el tamaño de los armarios eléctricos y se reduce el número de averías por desgaste de componentes. Aunque esto resultaba más amigable que los tableros eléctricos del principio, presentaba un problema de falta de flexibilidad, ya que un sistema de control solo sirve para una aplicación específica y no es reutilizable. Debido a esto y a la creciente demanda industrial del momento en el año de 1968 las compañías Ford y General Motors plantean las especificaciones que debe cumplir un controlador electrónico programable para ser realmente útil en la industria.

Es así que Bedford Associates desarrolló un prototipo de controlador industrial, que puede ser considerado el primer PLC (Controlador Lógico Programable, por sus siglas en inglés) de la historia, cumpliendo con las características que reclamaba la industria:

- Reutilizable.
- Adaptado a entornos agresivos (industria).
- Fácilmente programable por técnicos eléctricos.
- Implementado con electrónica del estado sólido (semiconductores).

Los primeros PLC se utilizaron para controlar procesos secuenciales (cadenas de montaje, transporte, etc.). El problema que presentaba era que su memoria era cableada, Por lo que la utilización es posible pero costosa.





A principios de 1970 aparece el microprocesador y con los primeros ordenadores digitales, se empezaron a implementar memorias con semiconductores, eliminando las memorias cableadas y logrando una mayor flexibilidad por la facilidad de programación. Pero por entonces todavía la tecnología de los microprocesadores no era utilizable en la industria por falta de robustez, dificultad de conexión a equipos mecánicos y dificultad de programación.

A mediados de los 70 del siglo pasado los autómatas incorporaron el microprocesador y las memorias semiconductoras, lo que permitió programar sin recablear (aumentando la flexibilidad buscada). Además permitió realizar cálculos matemáticos y comunicar con un ordenador central (ordenador encargado de controlar la planta enviando órdenes a los autómatas que gobiernan cada proceso). Junto con esto aparecen los primeros DCS (Sistema de control distribuido, por sus siglas en inglés). Que eran controladores lógicos al igual que los PLC solo que estos en el principio fueron usados para el control de variables analógicas.

De esta manera los DCS trabajaban a la par de sistemas separados de PLC para control discreto de variables on/off.

Hacia los finales de los años 70s aparecen mejoras en los autómatas dándole a estos:

- Mayor memoria.
- Capacidad de gobernar bucles de control.
- Más tipos de E/S (conexión más flexible de sensores/actuadores).
- Lenguajes de programación más potentes.
- Comunicaciones más potentes.

En los años 80s se continúa con las mejoras de los DCS siendo algunas de estas:

- Mayor velocidad de proceso.
- Dimensiones más reducidas.
- Técnicas de control más complejas (PID autoajustables, PID predictivos, fuzzy).
- Múltiples lenguajes (contactos, lista de instrucciones, GRAFCET, etc.).

En los años 90s cuando los sistemas de control basado en PC (computadora personal, por sus siglas en inglés) hicieron su aparición en la escena de la automatización industrial, los partidarios de estos sistemas más poderosos y abiertos afirmaban que dichos sistemas podrían llegar a suplantar a los controladores lógicos programables (PLC) y hasta los Sistemas de Control Distribuido (DCS) en numerosas aplicaciones. La PC tuvo mucho que ofrecer, pero no suplantaría las plataformas ya probadas de control industrial, no por lo menos en la forma de una PC de escritorio. La PC era la mejor opción a la hora de integrar funcionalidad avanzada, como puede ser conectividad de base de datos, integración, control analógico, simulación basada en WEB y comunicación de datos con terceros. El problema con el control basado en



PC ha sido siempre su etapa de potencia. Las PC que operan en OS (Sistemas Operativos) multiusuarios con hardware común tienen la desventaja de no contar con operaciones dedicadas y por prioridades; sino de acuerdo a su diseño original. El resultado de todas estas innovaciones fue la aparición de controladores híbridos que permitieron el manejo de variables analógicas y digitales, en conjunto con características del procesador de punto flotante para cálculos complejos, servicios de WEB interactivo embebido que facilita las tareas de control y monitoreo, Flash compacto removible para la recolección y registro de datos, puertos seriales múltiples y conexión mediante buses de campo para la comunicación con terceros.

Desde los primeros sistemas de control digital que se implantaron desde 1959 se puso de manifiesto que la aplicación de la tecnología en el control de los procesos industriales tendría un gran impacto en temas que por mucho no eran tomados en cuenta como factores importantes en la productividad.

El primer sistema de control digital en la refinería de la TEXACO ubicada en el puerto Arthur, en Texas. Este sistema se llamó RW-300 y fue diseñado e implementado por la compañía Ramo-Wooldridge.

Desde la implantación del primer sistema de control digital hasta la aparición del primer sistema de control distribuido tuvieron que pasar 16 años (1974). Muchos adelantos tecnológicos tuvieron que darse para que esto fuera posible, desde luego que estos adelantos tecnológicos se debieron a las teorías e investigaciones de muchos científicos, no solo en el campo de la electrónica, sino también en las matemáticas aplicadas a la resolución de problemas comunes.

Hoy en día los sistemas de control distribuidos operan en RT (tiempo real) y cuentan con sistemas de autosintonía, control adaptivo y predictivo, y lo más importante es que los datos de producción ya pueden ser leídos en línea por los sistemas de administración de la producción con lo cual se optimiza la producción eliminando tiempos muertos y sobreproducción. Otra ventaja importante es que las personas que deben conocer la información la pueden tener a la mano en el momento que la requieran en la pantalla de sus PCs en cualquier lugar que se encuentren.

En 1975, casi al mismo tiempo, se introdujeron los primeros sistemas de control distribuido por las compañías Honeywell y Yokogawa.

En 1980 varias compañías entraron al mercado de los DCS, con lo cual empezó la competencia tecnológica y mercadológica.

Junto con el desarrollo de la electrónica (hardware), también llegó el desarrollo en el campo de la programación (software) lo que produjo que una de las primeras ventajas de los controles distribuidos fuera la inclusión de los bloques de control embebidos dentro de los controladores, lo cual permitió que la programación de las estrategias de control fueran más fáciles de configurar. Después de esto vinieron los bloques de configuración funcional que volvieron la programación de los controladores mucho más amigables. Sin embargo, todos estos adelantos tecnológicos en el campo de la programación todavía tenían que esperar la evolución tecnológica de las interfaces de operación de los sistemas de control distribuido. También como parte del desarrollo de la industria de sistemas de control de procesos y la incipiente necesidad de centralizar la información o comunicar varios controladores o instrumentos



de medición de variables a los sistemas de control empezaron a surgir algunos protocolos de comunicación que fueron del tipo propietario; es decir, cada fabricante diseñaba y promocionaba sus propios protocolos de comunicación los cuales no eran compatibles con los protocolos de otro fabricante. Esto trajo como consecuencia que se crearan mercados de clientes cautivos.

El desarrollo del DCS va paralelo al de las comunicaciones, cada vez es más necesario disponer de dispositivos inteligentes para realizar el control o la supervisión remota. Los sistemas o redes de comunicación empleados en entornos industriales se encuentran sometidos a una problemática específica que condicionan enormemente su diseño y las diferencias de las redes de datos o redes de oficina.

El desarrollo de los microprocesadores, microcontroladores y los controladores lógicos programables (PLC) dio lugar a la aparición del control distribuido. En este tipo de esquema un microprocesador o un PLC permitió controlar una o más variables del sistema realizando un control directo de las mismas. Estos equipos de control local se comunican con otros elementos de su nivel y con el nivel superior de supervisión.

1.2. Planteamiento del problema

El día de hoy existe una carencia en el país de desarrollos nacionales, en la programación de sistemas distribuidos de medición y control que utilicen plataformas y protocolos estandarizados.

1.3. Justificación

En general, las industrias utilizan sistemas SCADA (Supervisory Control And Data Adquisition), los cuales no se programan solo se configuran, siendo de costos elevados y manteniendo la dependencia con el proveedor. Su costo depende de las prestaciones contratadas (número de variables que permiten ser medidas, supervisadas y controladas). si se desea ampliar hay que pagar el costo que requiere dicha ampliación, puesto que no se tiene acceso al código fuente del sistema.

El sistema cliente/servidor TCP/IP que se pretende realizar no tiene restricciones críticas de tiempo, porque los clientes trabajan de forma autónoma. no obstante se consideran tiempos de transmisión recepción aceptables entre 1 y 5 minutos, si los tiempos de transmisión recepción son menores de un minuto se considera que la comunicación se realiza de forma óptima.

1.4. Hipótesis

Es posible desarrollar una estrategia de control en un entorno distribuido, en el cual se definen los nodos supervisores y locales, así como una metodología para la obtención de los parámetros de los controles a implementar y variables del proceso. mediante un sistema cliente/servidor TCP/IP, en una red local Ethernet, que satisfaga las restricciones típicas de tiempo que permitan una operación exitosa.





1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un sistema de supervisión y control, que integre estrategias de control automático en un ambiente distribuido mediante un protocolo de comunicación estándar.

1.5.2. Objetivos específicos

- Desarrollar una estrategia de control en un entorno distribuido, en el que se definen los nodos supervisores y locales.
- Especificar una metodología para la obtención de los controladores a implementar.
- Implementar el control difuso de la planta mediante una simulación dinámica.
- Programar un algoritmo de autoajuste para un controlador PID.
- Integrar los algoritmos de control en una arquitectura cliente servidor de múltiples nodos.

1.6. Alcance

Al no contar con una planta física en el centro, se trabajó en forma simulada.

1.7. Metodología y desarrollo de la investigación

El presente trabajo de tesis se realizó en 5 etapas las cuales comprenden:

- Etapa conceptual: en esta etapa se planteó cual es el objetivo de interés, cual es la problemática o los aspectos que interesa estudiar en el trabajo de investigación.
- Etapa de planificación: en esta etapa se planeó como dar respuesta a la hipótesis y concretar los objetivos.
- Etapa de investigación: es en esta etapa cuando se recabó información teórica que diera respuesta a lo planteado en la etapa conceptual y de planificación.
- Etapa empírica: es en esta etapa donde se implementó lo que se planeó tomando como base los conceptos teóricos que se recabaron en la etapa de investigación.
- Etapa de interpretación: en esta etapa se resumen los resultados obtenidos de la etapa empírica.



1.8. Contribuciones del trabajo

- Programación y simulación de un sistema distribuido de supervisión y control con base en un sistema cliente/servidor TCP/IP.
- El diseño del sistema hace posible que los nodos y el centro de control intercambien información cumpliendo con las restricciones de tiempo impuestas y en caso de fallas en la conectividad, los nodos operarían de forma autónoma.
- Comparación del desempeño de los controladores difuso y PID autoajutable para el proceso de nivel simulado.

1.9. Estructura de la tesis

El resultado del presente trabajo se ha estructurado en 8 capítulos, adicionalmente se incluyen 2 anexos para consultas de algunos contenidos de la tesis.

la estructura es la siguiente:

- Capítulo I.- Introducción: contiene las bases bajo las cuales se origina la idea de llevar a cabo este trabajo.
- Capítulo II.- Estado del arte: en este capítulo se hace un breve estudio de la evolución de los sistemas de supervisión y control, así como del estado actual de las redes industriales y los principales protocolos de comunicación.
- Capítulo III.- Marco teórico: contiene las bases teóricas necesarias para poder desarrollar un sistema de supervisor y control; estas bases son: control automático, control fuzzy y redes de computadoras y conectividad.
- Capítulo IV.- Arquitectura y funcionalidad del sistema distribuido: Describe las principales partes y funciones que se requiere para que el sistema pueda ser útil a las necesidades de operación.
- Capítulo V.- Controlador difuso para un proceso de nivel simulado: Se describe la metodología bajo la cual se diseñó el controlador difuso para el proceso de nivel simulado.
- Capítulo VI.- Integrar el control difuso del PNS en un sistema cliente servidor de varios nodos: Con base en los requerimientos del sistema, se describe el funcionamiento de la interfaz en los nodos cliente y servidor.
- Capítulo VII.- Controlador PID autoajutable para un proceso simulado (PS): Se describe la metodología bajo la cual se diseñó el controlador PID autoajutable para el proceso simulado.
- Capítulo VIII.- Evaluar resultados: Se evalúa la respuesta del sistema con cada uno de los controladores diseñados en este trabajo.



- Capítulo IX.- Conclusiones y trabajo futuro: de acuerdo con lo planteado en los objetivos se evalúa qué objetivos se cumplieron, que se deduce del trabajo y que trabajos se harán en un futuro.



Capítulo 2

ESTADO DEL ARTE

En este capítulo se hace un breve estudio de los sistemas de supervisión y control; en la sección 2.1 se describe un poco de la historia y desarrollo de los sistemas de supervisión y control, la sección 2.2 se describe los niveles de una red industrial, así como los protocolos de comunicación en cada uno de los niveles de la red industrial y las diferencias entre SCD y SCADA; por último la sección 2.3 hace un estudio de los principales algoritmos de control que se aplican en este trabajo.

2.1. Desarrollo de los sistemas de control

El control automático ha evolucionado a través del tiempo y no es muy fácil precisar la fecha exacta del primer sistema de control en lazo cerrado, con la aparición de las primeras computadoras digitales, comenzaron los desarrollos en el campo del control de procesos industriales en forma digital. Desde los primeros sistemas de control que se implantaron en el año de 1959 se puso de manifiesto que la aplicación de la tecnología en el control de los procesos industriales tendría un gran impacto en la productividad. El primer sistema de control digital serio fue implementado en 1959 en la refinería de la Texaco ubicada en puerto Arthur, Texas este sistema se llamó RW-300 y fue diseñado e implementado por la compañía Ramo-Wooldrige [6]. Desde la implantación del primer sistema de control digital hasta la aparición del primer sistema de control distribuido tuvieron que pasar 16 años. fue que muchos adelantos científicos y tecnológicos tuvieron que darse para que este hecho fuera posible, desde luego que este adelanto se debió a las investigaciones de muchos científicos, no solo en el campo de la electrónica, sino también en el campo de las matemáticas aplicadas a la resolución de problemas comunes. Ejemplo de esto es el desarrollo de la Teoría del control digital.

En 1975, se introdujeron los primeros Sistemas de Control Distribuido por las compañías Honeywell quien introdujo el TDC-2000 [7] y Yokogawa quien introdujo el CENTUM [8]. En 1980 el mercado de los sistemas de control digital se expandió, con lo cual empezó la competencia tecnológica y mercadológica en este campo; las compañías que entraron en esta competencia fueron Bailey quien introdujo el NETWORK 90, Fischer and Porter quien introdujo DCI-4000.



Como parte del desarrollo de los sistemas de control y la necesidad de comunicar varios controladores o instrumentos de variables a los sistemas de control, empezaron a surgir algunos protocolos de comunicación, Estos protocolos de comunicación fueron por supuesto del tipo propietario, es decir que cada fabricante diseñaba y promocionaba sus protocolos de comunicación, los cuales no eran compatibles con los protocolos de otro fabricante, Esto trajo como consecuencia que se crearan mercados de clientes cautivos o dependientes de una sola tecnología.

Con la proliferación de fabricantes y diferentes protocolos de comunicación, vino la necesidad de normalizar dichos protocolos: la consecuencia de ésto es que los fabricantes empezaran a crear protocolos abiertos de comunicación, o bien que los fabricantes abrieran sus protocolos para que cualquiera pudiera comunicarse con sus equipos. Actualmente los Sistemas de Control Distribuido hacen uso de todos los avances tecnológicos en todas las ramas de la ingeniería.

2.2. Red industrial

2.2.1. Niveles de una Red Industrial

- **Nivel de gestión:** es el nivel más elevado y se encarga de gestionar e integrar los niveles siguientes, las maquinas aquí conectadas suelen ser estaciones de trabajo que hacen de puente entre el proceso productivo y el área de gestión, se emplea una red tipo LAN o WAN.
- **Nivel de control:** enlaza distintas zonas de trabajo con autómatas de gama alta y PC dedicados a diseño, programación, etc.
- **Nivel de campo y procesos:** integra pequeños automatismos (autómatas compactos, multiplexores de E/S, Controladores PID, etc.).
- **Nivel de E/S:** es el nivel más próximo al proceso. Está compuesto por sensores y actuadores, de diferentes especificaciones.

La Fig. 2.1 muestra en forma esquemática los niveles de una red industrial.

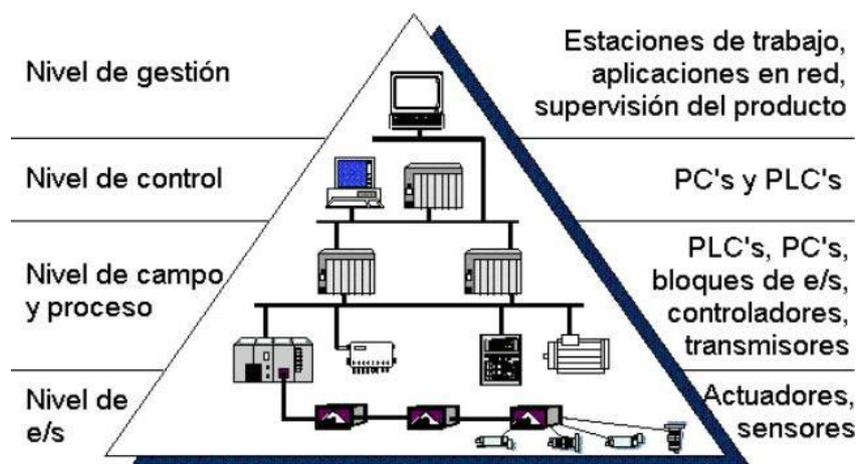


Figura 2.1: Niveles de una Red Industrial

La integración de los diferentes niveles da lugar a la estructura de redes industriales, las cuales es posible agrupar en tres categorías:

- Buses de campo
- Redes LAN
- Redes LAN-WAN

Los buses de datos que permiten la integración de equipos para la medición y control de variables de proceso, reciben la denominación genérica de buses de campo, estos buses de campo son un sistema de transmisión de información que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales. El objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional lazo de corriente de 4-20mA o 0 a 10 VDC, según corresponda. Generalmente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conecta dispositivos de campo como PLC's, transductores, actuadores, sensores y equipos de supervisión. Varios grupos han intentado generar e imponer una norma que permita la integración de equipos de distintos proveedores, no obstante, hasta ahora no existe un bus de campo universal.

2.2.2. Buses de campo

Los buses de campo con mayor presencia en el área del control y la automatización de procesos son:

- HART
- Profibus
- Fieldbus Foundation



HART

El protocolo HART (High way-Addressable-Remote-Transducer) agrupa la información digital sobre la señal analógica típica de 4 a 20 mA DC. La señal digital usa dos frecuencias individuales de 1200 y 2200 Hz, que representan los dígitos 1 y 0 respectivamente y que en conjunto forman una onda sinusoidal que se superpone al lazo de corriente de 4-20mA.

PROFIBUS

(Process Field Bus) Norma internacional de bus de campo de alta velocidad para control de procesos normalizada en Europa por EN50170.

Existen tres perfiles:

- Profibus DP (Decentralized Periphery). Orientado a sensores/actuadores enlazados a procesadores (PLC's) o terminales.
- Profibus PA (Process Automation). Para control de proceso, cumple normas especiales de seguridad para la industria química (IEC 1 1 15 8-2, seguridad intrínseca).
- Profibus FMS (Fieldbus Message Specification). Para comunicación entre células de proceso o equipos de automatización.

FOUNDATION FIELBUS

Es un protocolo de comunicación digital para redes industriales, específicamente utilizado en aplicaciones de control distribuido. Puede comunicar grandes volúmenes de información, ideal para comunicaciones con varios lazos complejos de control de procesos y automatización. Está orientado principalmente a la interconexión de dispositivos en industrias de proceso continuo. Los dispositivos de campo son alimentados a través del bus Fieldbus cuando la potencia requerida para el funcionamiento lo permite.

Otros protocolos ampliamente usados aunque de menor alcance son:

- Modbus
- DeviceNet

MODBUS

Modbus es un protocolo de transmisión para sistemas de control y supervisión de procesos (SCADA) con control centralizado, puede comunicarse con una o varias estaciones remotas (RTU) con la finalidad de obtener datos de campo para la supervisión y control de un proceso. La interfaz de capa física puede estar configurada en: RS-232, RS-422, RS-485.

En Modbus los datos pueden intercambiarse en dos modos de transmisión:





- Modo RTU
- Modo ASCII

DEVICENET

Red de bajo nivel adecuada para conectar dispositivos simples como sensores fotoeléctricos, sensores magnéticos, pulsadores, etc. y dispositivos de alto nivel (PLC, controladores, computadores, IHM, entre otros). Provee información adicional sobre el estado de la red, cuyos datos serán desplegados en la interfaz del usuario.

En la Tabla A.1 que se localiza en el APENDICE A muestra en resumen las redes industriales más usadas y sus características [9].

2.2.3. Sistemas de Control Distribuido

Un Sistema de Control Distribuido o SCD, más conocido por sus siglas en inglés DCS (Distributed Control System), es un sistema de control aplicado a procesos industriales complejos en las grandes industrias como petroquímicas, papeleras, metalúrgicas, centrales de generación, plantas de tratamiento de aguas o la industria farmacéutica.

Los SCD están constituidos principalmente por cuatro elementos, como se puede observar en la Fig. 2.2:

- Tarjetas de entradas y salidas
- Procesadores de control
- Interfaz Hombre-Máquina (IHM)
- Red de comunicaciones

El procesador de control generalmente se encarga de ejecutar todos y cada uno de los algoritmos y del procesamiento de datos es el corazón del SCD. Este dispositivo procesa las señales que provienen de las tarjetas de entradas para enviar estos datos a la IHM de las estaciones cliente del sistema de control. Es importante mencionar que la comunicación entre las tarjetas I/O's, los procesadores, y las estaciones de operación (parte de la IHM) es llevada a cabo mediante una red que comúnmente es llamada network (red de trabajo).

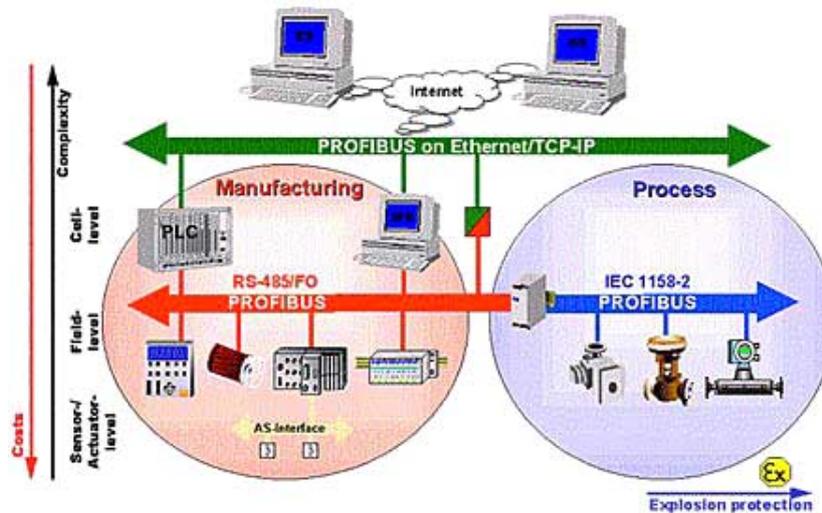


Figura 2.2: Es un esquema cortesía del proveedor SMAR [1].

2.2.4. Sistemas SCADA

SCADA proviene de las siglas Supervisory Control and Data Acquisition (Supervisión, Control y adquisición de datos). Los sistemas SCADA son aplicaciones de software diseñadas con el objetivo de controlar y supervisar procesos a distancia, se basan en la adquisición de datos de procesos remotos.

Este tipo de sistemas son diseñados para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (Actuadores y sensores), y controlando el proceso de forma automática desde una computadora, además envía la información generada en el proceso productivo a diversos usuarios. Las tareas de supervisión y control generalmente están más relacionadas con el software SCADA, en el, el operador puede visualizar en la pantalla del computador cada una de las estaciones remotas que conforman el sistema. Todo esto se ejecuta en tiempo real, y están diseñados para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos.

Los sistemas SCADA están constituidos por tres elementos, como se muestra en la Fig. 2.3:

- Múltiples unidades de terminal remota (también conocida como UTR, RTU o estaciones externas).
- Estación maestra y computador con IHM.
- Infraestructura de comunicación.

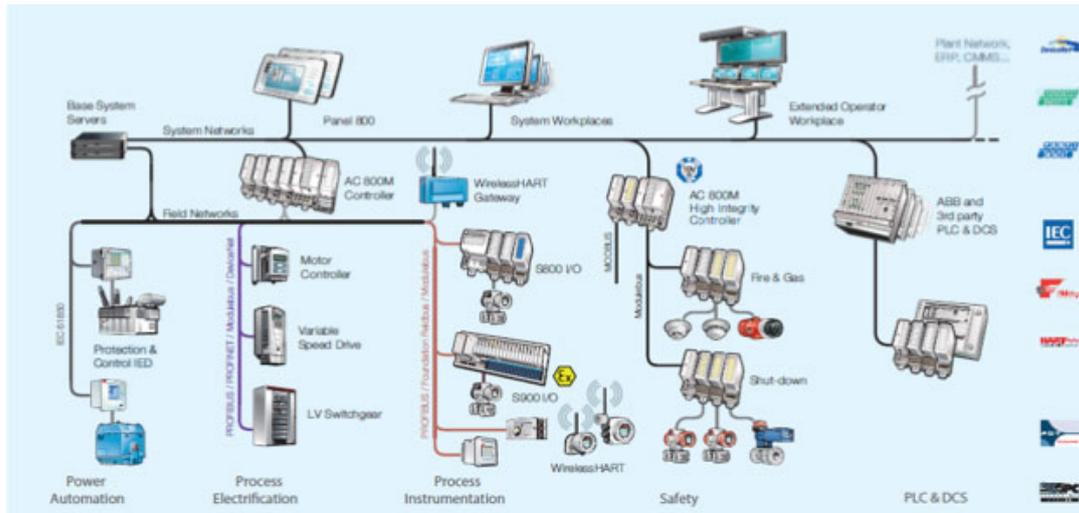


Figura 2.3: Esquema cortesía del proveedor ABB [2].

2.2.5. Diferencias entre SCD y SCADA.

Los sistemas SCADA se utilizan en el control de oleoductos, sistemas de transmisión de energía eléctrica, yacimientos de gas y petróleo, redes de distribución de gas natural, generación energética (convencional y nuclear). No todos los sistemas SCADA están limitados a procesos industriales sino que también se ha extendido su uso a instalaciones experimentales como la fusión nuclear donde la alta capacidad de gestionar un número elevado de E/S. los convierte ideales para este tipo de procesos. Sistemas similares a los SCADA son vistos rutinariamente en fábricas, plantas de tratamiento. Etc. Estos son llamados a menudo como Sistemas de Control Distribuido (DCS, Distributed Control System). tienen funciones similares a los sistemas SCADA. Pero las unidades de colección o de control de datos de campo se establecen generalmente dentro de un área confinada. Las comunicaciones pueden ser vía una red de área local (LAN), y serán normalmente confiables y de alta velocidad. Un sistema DCS emplea generalmente cantidades significativas de control a lazo cerrado. A diferencia de un sistema DCS un sistema SCADA generalmente el lazo de control es cerrado por el operador. Los sistemas DCS se caracterizan por realizar las acciones de control en forma automática, hoy en día es fácil hallar un sistema SCADA realizando labores de control automático en cualquiera de sus niveles, aunque su labor principal sea de supervisión y control por parte del operador.

En la Tabla 2.1 se muestra un cuadro comparativo de las principales características de los sistemas SCADA y los sistemas DCS .



Tabla 2.1: Características de los sistemas SCADA y los sistemas DCS

ASPECTO	SCADA	DCS
TIPO DE ARQUITECTURA	CENTRALIZADA	DISTRIBUIDA
TIPO DE CONTROL PREDOMINANTE	SUPERVISORIO: Lazos de control cerrados por el operador. Control secuencial y regulatorio.	REGULATORIO: Lazos de control cerrados automáticamente por el sistema. Control secuencial o algoritmos avanzados
TIPOS DE VARIABLES	DESACOPLADAS	ACOPLADAS
ÁREA DE ACCIÓN	Áreas geográficamente distribuidas.	Área de la planta.
UNIDADES DE ADQUISICIÓN DE DATOS Y CONTROL	Remotas, PLC's.	Controladores de lazo, PLC's.
MEDIOS DE COMUNICACIÓN	Radio, Satélite, Líneas telefónicas, Conexión directa, LAN, WAN.	Redes de área local, conexión directa

2.3. Algoritmos de control.

Actualmente el esquema de control PID es el más usado en la industria y en la academia debido a su estructura simple, fiabilidad, practicidad y fácil implementación [10], no obstante, existen también técnicas con lógica difusa que hacen que el control sea factible con plantas altamente no lineales o que no se tiene acceso a su modelo matemático.

En la actualidad existen diversos estudios en los cuales se presentan mejoras a el algoritmo de control PID tradicional, usando para esto técnicas de inteligencia artificial como la lógica difusa y las redes neuronales [11], los controladores basados en lógica difusa y redes neuronales han demostrado ser altamente efectivos en plantas complejas y no lineales [12][13][14]. También un campo abierto a la investigación actualmente es la mejoría de los algoritmos de sintonización de controladores PID, que tienen por objetivo mejorar la respuesta del controlador PID.

Los algoritmos de control con lógica difusa han demostrado su eficiencia en aplicaciones en donde es posible el control de plantas que incluyen no linealidades, así como la programación de los algoritmos en diversos lenguajes de programación y dispositivos electrónicos como FPGA [15], Microcontroladores [16] y computadoras personales [17].

Un problema en el diseño de controladores con lógica difusa tipo Mamdani es que el controlador se diseña para una referencia dada y si esta referencia cambia entonces



se tiene que replantear o ajustar las reglas difusas, por esta razón se propone que el seguimiento de la referencia se ajuste por un incremento en la señal de control, en [18] se hace una comparación entre un controlador PID clásico, y un controlador fuzzy incremental, mostrando que para el diseño de un controlador fuzzy incremental únicamente es necesario tener información de las entradas y salidas del sistema, y no de la dinámica de la planta o un modelo matemático, los resultados de la simulación muestran que este tipo de controlador tiene mejor rendimiento que un controlador clásico.

El principal problema en los controladores PID es la correcta elección de las ganancias del controlador, en este campo existen actualmente en la literatura métodos de sintonización, entre los más conocidos están los métodos de Ziegler and Nichols, en este trabajo se presenta el método gráfico, el cual se aplica a plantas cuya respuesta al escalón en lazo abierto tiene una respuesta en forma de s , en [19] se propone un método modificado de Ziegler and Nichols para encontrar las ganancias de un controlador PID y se compara la respuesta del controlador con las ganancias obtenidas con el método modificado y el método clásico, observando una mejoría en la respuesta del PID. En el trabajo [20] se presenta un algoritmo evolutivo para la selección de las ganancias del controlador PID, este algoritmo propone establecer un vector para cada parámetro del controlador, tomando como base los resultados de aplicar el método de Ziegler and Nichols y a partir de estos resultados proponer dicho vector, obteniendo como resultados una mejoría en la respuesta del controlador comparando con el controlador de ganancias fijas. y en el trabajo [21] se presenta un método que mejora el ya propuesto, la idea principal de este algoritmo resulta de generar aleatoriamente vectores de ajuste para los parámetros del PID y como resultado se mejora aún más la respuesta del controlador PID clásico. En el campo del ajuste de las ganancias de un controlador PID la lógica difusa permite generar un controlador el cual tiene una etapa de autoajuste, este tipo de controlador tiene una etapa de control en la cual se encuentra el algoritmo PID clásico y una etapa de autoajuste en la cual se encuentra un controlador con lógica difusa tipo Mamdani, el controlador difuso es el encargado de ajustar las ganancias del controlador PID conforme la dinámica del sistema evoluciona [22][23][24][25][26]; este tipo de ajuste se realiza debido a que un controlador PID es sintonizado con los métodos clásicos y las ganancias del controlador quedan fijas para las condiciones de la planta en el momento en el que se sintonizó el controlador, pero si las condiciones de la planta cambian entonces estos valores del controlador ya no darán una respuesta favorable es por esto que el diseño de un controlador con una etapa de autoajuste no únicamente mejora la respuesta del sistema sino que hace al controlador más tolerante a las variaciones en la dinámica de la planta, logrando con esto una mayor robustez del controlador [27]; una variante es establecer un controlador difuso incremental que ajuste las ganancias del controlador, esto quiere decir que un controlador tipo Mamdani regula la variación de cada uno de los parámetros de un controlador PID partiendo de las ganancias de Ziegler and Nichols para dicho controlador, a esta variante del algoritmo de autoajuste se le llama algoritmo de autoajuste con autoaprendizaje [28][29]



2.4. Resumen del capítulo.

En este capítulo se presentó un estudio de la situación que vive en la actualidad los sistemas de control distribuido (SCD), los sistemas SCADA y los algoritmos de control.

En la sección 2.1 se describió como los sistemas de control distribuido iniciaron, donde se implementó el primer sistema distribuido de control y como han evolucionado estos sistemas hasta nuestros días.

En la sección 2.2 se describió como está constituida una red industrial en sus diferentes niveles, así como los protocolos que se maneja en cada nivel de la red industrial, también se mencionó las principales diferencias entre los sistemas SCD y los sistemas SCADA.

Y finalmente la sección 2.3 mencionó algunos de los diferentes algoritmos de control y las técnicas que se usan para desarrollarlos.

Capítulo 3

MARCO TEÓRICO

En el presente capítulo se abordan los principales conceptos que sirven como base para el desarrollo de este trabajo, en la sección 3.1 se aborda de forma breve el control clásico, en la sección 3.2 el control difuso, la sección 3.3 las redes de computadoras y conectividad y por último se aborda de forma particular el protocolo TCP/IP.

3.1. Tipo de controladores.

3.1.1. Control clásico.

El control automático hoy en día juega un papel importante en la industria, ya que con la práctica de las técnicas de control automático se a logrado obtener un mejor desempeño de los sistemas. La manera en la cual un controlador automático produce una señal se le llama acción de control [30]. En este tipo de sistemas la variable controlada juega un papel muy importante, ya que esta es la variable que se quiere llevar hasta un valor de referencia, y es esta misma la que se monitorea en todo momento, comúnmente esta es la señal de salida del sistema. Otro aspecto que se trata en este tipo de sistemas es la variable manipulada, que es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada, comúnmente esta señal se denomina señal de control. la planta es cualquier objeto físico que se desea controlar, este tipo de objetos pueden ser: un dispositivo mecánico, eléctrico, biológico, térmico, electromecánico, etc. Una señal que tiende a afectar adversamente el valor de salida o variable controlada de un sistema se le conoce como perturbación, si es que esta se genera dentro del sistema se le conoce como perturbación interna, si se produce fuera del sistema se le conoce como perturbación externa y es una entrada al sistema.

3.1.2. Sistemas de control en lazo abierto.

Un sistema de control en lazo abierto es aquel en el cual la señal o variable manipulada entra a la planta y produce una señal o variable controlada, la cual no se usa para comparar el valor de referencia y la variable controlada y por lo tanto no



se produce un error o diferencia entre la señal de referencia y la variable controlada. Esto se muestra en la Fig. 3.1.

En este tipo de sistemas la variable controlada no tiene ningún efecto sobre la acción de control, esto quiere decir que la variable controlada no se mide, de esta manera a cada entrada de referencia le corresponde una y solo una acción de control, dado que el controlador tiene únicamente una señal o un valor fijo en la entrada de referencia.

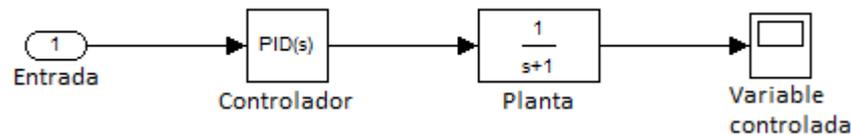


Figura 3.1: Sistema de control en lazo abierto

3.1.3. Sistemas de control en lazo cerrado.

Un sistema de control en lazo cerrado o realimentado, a diferencia del control en lazo abierto, la variable controlada se está midiendo con la finalidad de comparar este valor con un valor de referencia, esa comparación arrojará un valor el cual es la referencia que el controlador toma como entrada para producir la señal manipulada [30]. En un sentido estricto la diferencia entre la señal de referencia y la señal de control es llevada hasta un valor de cero o muy aproximada a cero, de esta manera se consigue que la señal o variable controlada converja a la señal o variable de referencia. En la Fig. 3.2. Se muestra este tipo de sistema.

En este tipo de sistema la entrada de referencia se compara con la variable controlada, de esta manera el controlador tiene una señal de referencia que varía, de esta manera el controlador ajusta valores de la señal de control o manipulada, con el objetivo de hacer que la señal controlada sea igual a la señal de referencia.



Figura 3.2: Sistema de control en lazo cerrado

3.1.4. Respuesta típica de un sistema de control

La respuesta típica de un sistema de control se conoce como respuesta transitoria, y es aquella porción que tiende a una constante cuando el tiempo crece en forma



indefinida. esta respuesta es importante porque tanto la amplitud como la duración de dicha respuesta debe mantenerse dentro de los límites tolerables o prescritos. para un sistema de control frecuentemente la respuesta transitoria se realiza mediante una función escalón unitario como entrada, esta respuesta se conoce como respuesta al escalón unitario [30][31][32]. La Fig.3.3 ilustra una respuesta típica al escalón unitario de un sistema de control, el criterio de desempeño comúnmente utilizado para la caracterización de los sistemas de control en el tiempo se define como:

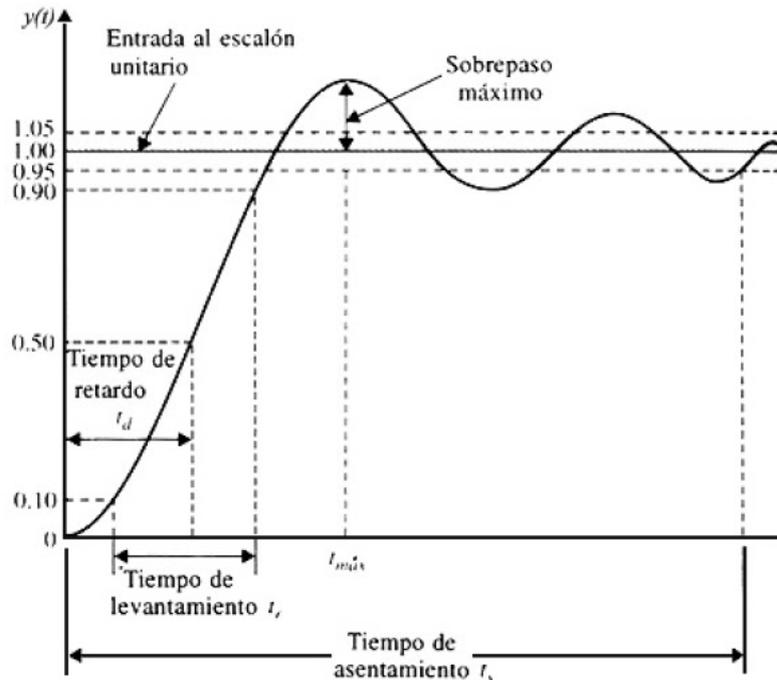


Figura 3.3: Respuesta al escalón unitario

■ **Sobrepaso máximo.**

Asume que $y(t)$ es la respuesta al escalón unitario, y que $y_{máx}$ es el valor máximo de $y(t)$ y y_{ss} es el valor en estado estable de $y(t)$ y $y_{máx} \geq y_{ss}$. el sobrepaso máximo se define como:

$$\text{Sobrepaso máximo} = y_{máx} - y_{ss} \tag{3.1}$$

El sobrepaso máximo se presenta como un porcentaje del valor final de la respuesta; como:

$$\text{Porcentaje máximo de sobrepaso} = \frac{\text{Sobrepaso máximo}}{y_{ss}} * 100 \tag{3.2}$$

■ **Tiempo de retardo.**

El tiempo de retardo se representa por T_d y se define como el tiempo que se





requiere para que la respuesta al escalón alcance el 50 % de su valor final, esto se puede ver la Fig.3.3.

■ **Tiempo de levantamiento.**

El tiempo de levantamiento se representa por t_t y es el tiempo que se requiere para que la respuesta al escalón se eleve del 10 al 90 % de su valor final, como se muestra en la Fig.3.3 Una medida alternativa es representar este tiempo como el recíproco de la pendiente de la respuesta al escalón en el instante en el que la respuesta es igual al 50 % de su valor final.

■ **Tiempo de asentamiento.**

Se representa por t_s y se define como el tiempo que se requiere para que la respuesta al escalón disminuya y permanezca dentro de un porcentaje específico de su valor final. una cifra de uso frecuente es 5 %.

3.1.5. Esquema de control PID.

El esquema de control PID ha sido usado de manera exitosa en sistemas de control industrial por más de 50 años. El principio básico de este esquema de control es que actúa sobre la variable a ser manipulada a través de una apropiada combinación de sus tres acciones de control (acción proporcional, acción integral y acción derivativa). En situaciones donde muchos procesos se controlan directamente mediante una sola computadora digital, la mayoría de los lazos se pueden manipular mediante esquemas de control PID. El esquema de control PID en controladores analógicos está dado por:

$$m(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (3.3)$$

Donde $e(t)$ es la entrada al controlador (señal de error), $m(t)$ es la salida del controlador (señal manipulada), K es la ganancia proporcional, T_i es el tiempo integral y T_d es el tiempo derivativo.

Para obtener la función de transferencia pulso del controlador PID digital, se puede discretizar (3.3), al aproximar el término integral mediante la sumatoria trapezoidal y el término derivativo mediante la diferencia de dos puntos [33] [5], se obtiene.

$$m(KT) = K \left\{ e(KT) + \frac{T}{T_i} \sum_{h=1}^K \frac{e((h-1)T) + e(hT)}{2} + \frac{T_d}{T} [e(KT) - e((K-1)T)] \right\} \quad (3.4)$$

Se define

$$\frac{e((h-1)T) + e(hT)}{2} = f(hT), f(0) = 0$$

Entonces





$$\sum_{h=1}^K \frac{e((h-1)T) + e(hT)}{2} = \sum_{h=1}^K f(hT)$$

Al tomar la transformada z de esta última ecuación, se obtiene

$$Z \left[\sum_{h=1}^K \frac{e((h-1)T) + e(hT)}{2} \right] = Z \left[\sum_{h=1}^K f(hT) \right] = \frac{1}{1-z^{-1}} [F(z) - f(0)] = \frac{1}{1-z^{-1}} F(z)$$

Nótese que

$$F(z) = Z[f(hT)] = \frac{1+z^{-1}}{2} E(z)$$

Por lo tanto.

$$Z \left[\sum_{h=1}^K \frac{e((h-1)T) + e(hT)}{2} \right] = \frac{1+z^{-1}}{2(1-z^{-1})} E(z)$$

Entonces la transformada Z de (3.4) da como resultado

$$M(z) = K \left[1 + \frac{T}{2T_i} \frac{1+z^{-1}}{1-z^{-1}} + \frac{T_d}{T} (1-z^{-1}) \right] E(z)$$

Esta última ecuación se puede reescribir como sigue:

$$M(z) = K \left[1 - \frac{T}{2T_i} \frac{1+z^{-1}}{1-z^{-1}} + \frac{T_d}{T} (1-z^{-1}) \right] E(z) = \left[K_p + \frac{K_i}{1-z^{-1}} + K_d(1-z^{-1}) \right] E(z)$$

La función de transferencia pulso para el controlador digital se convierte en

$$G_D(z) = \frac{M(z)}{E(z)} = K_p + \frac{K_i}{1-z^{-1}} + K_d(1-z^{-1}) \quad (3.5)$$

3.1.6. Método gráfico de Ziegler and Nichols

El proceso de seleccionar adecuadamente los parámetros que cumplan con las especificaciones de comportamiento dadas se conoce comúnmente como sintonía de controladores; existen diversos métodos de sintonía de controladores, entre los más importantes se encuentra el método propuesto por Ziegler and Nichols en 1941, en el cual se sugieren reglas para la sintonía de controladores PID, esto significa dar valores (K_p , T_i y T_d) basándose en la respuesta escalón experimentales o en el valor de K_p que produce estabilidad marginal cuando solo se usa la acción de control proporcional, en este trabajo se aplica el método de respuesta de la planta al escalón; este método es muy conveniente cuando no se conoce el modelo matemático de la planta, no obstante, estas reglas se pueden aplicar también en el diseño de sistemas con modelos matemáticos conocidos. En este método se obtiene la respuesta de la planta a un escalón unitario, si la planta no tiene integradores o polos complejos



conjugados, entonces la curva de respuesta al escalón tendrá una forma de S, este método se aplica si la respuesta del sistema tiene la forma de S [30] [31], las curvas se pueden generar con una simulación dinámica de la planta.

La curva se caracteriza porque a partir de esta se pueden obtener dos constantes, la constante de retardo (L) y la constante de tiempo (T); las constantes se determinan dibujando una línea tangente en el punto de inflexión de la curva y determinando el punto de intersección de la recta tangente con el eje del tiempo y con la línea $c(t) = K$ tal como se muestra en la Fig.3.4.

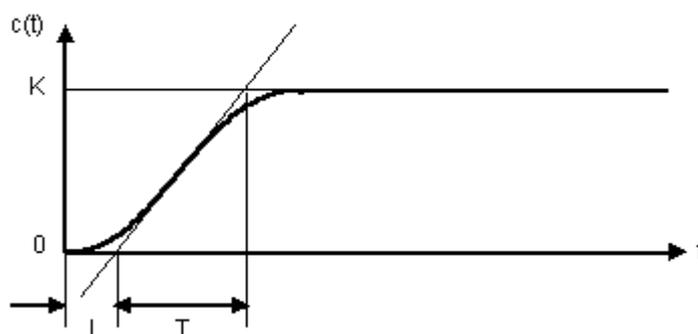


Figura 3.4: Método gráfico Ziegler and Nichols

Ziegler and Nichols proponen los valores para los parámetros del controlador (K_p, T_i, T_d) de acuerdo a la Tabla 3.1; es importante resaltar que el controlador PID sintonizado con este método tiene la forma de (3.6).

Tabla 3.1: Reglas de sintonización de Ziegler and Nichols basadas en la respuesta al escalón unitario de la planta.

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 * \frac{T}{L}$	$\frac{L}{3}$	0
PID	$1,2 * \frac{T}{L}$	$2 * L$	$0,5 * L$

$$G_c(s) = 1,2 \frac{T}{L} \left(1 + \frac{1}{2Ls} + 0,5Ls \right) \quad (3.6)$$



3.2. CONTROL DIFUSO

3.2.1. Lógica difusa.

La lógica difusa es una alternativa a la lógica clásica que pretende introducir un grado de verdad en las cosas que evalúa. En el mundo en el que vivimos existe mucho conocimiento impreciso por naturaleza. El razonamiento humano con frecuencia actúa con este tipo de información, la lógica difusa fue diseñada precisamente para imitar la forma en la cual el ser humano razona, ya que este no lo hace de forma de valores certeros si no con valores imprecisos incluyendo los certeros[34].

La lógica difusa se inició en 1965 por Lofti A. Zadeh, surgió como una herramienta importante para el control de sistemas y procesos industriales complejos , sistemas de diagnóstico y otros sistemas expertos.

La lógica difusa a diferencia de la lógica convencional permite trabajar con información que no es exacta para poder definir evaluaciones convencionales, al contrario de la lógica convencional que permite trabajar con valores precisos y bien definidos [34].

¿Cuándo aplicar la lógica difusa? La lógica difusa se puede aplicar en procesos demasiado complejos, cuando no existe un modelo de solución simple o un modelo matemático preciso. Es muy útil también cuando se necesita utilizar el conocimiento de un experto que utiliza conceptos ambiguos o imprecisos [3] [35]. De la misma manera también se puede aplicar cuando ciertos parámetros de un sistema no se conocen y no pueden medirse de forma confiable. Es importante mencionar que no es recomendable utilizar la lógica difusa cuando algún modelo matemático ya soluciona eficientemente el problema ejemplo de esto son sistemas lineales e invariantes en el tiempo.

3.2.2. Teoría de conjuntos difusos.

La lógica difusa permite tratar con información que no es exacta o con un alto grado de precisión a diferencia de la lógica convencional la cual trabaja con información precisa. El problema principal surge de la poca capacidad de expresión de la lógica clásica [36].

Conjunto clásico.

Los conjuntos clásicos surgen a partir de la necesidad del ser humano de clasificar objetos así como conceptos, estos conjuntos se establecen como un conjunto bien definido de elementos, y entonces se puede evaluar si los objetos pertenecen o no pertenecen al conjunto. Y únicamente se tienen estos valores de pertenencia de 1 si el objeto pertenece al conjunto o 0 si el elemento no pertenece al conjunto [37]. De esta manera un conjunto clásico se puede definir con una función de pertenencia



mostrada en (3.7).

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \text{ no pertenece a } A \\ 1 & \text{si } x \text{ pertenece a } A \end{cases} \quad (3.7)$$

Conjuntos difusos.

La necesidad de trabajar con conjuntos difusos surge del hecho de que en el lenguaje natural del ser humano existen conceptos que no tienen límites claros, y los conjuntos difusos permiten expresar de una forma matemática este tipo de razonamiento. Un conjunto difuso se encuentra asociado por un valor lingüístico que está definido por una palabra o etiqueta lingüística. En los conjuntos difusos la función de pertenencia puede tomar valores del intervalo entre 0 y 1, y la transición entre 0 y 1 es gradual y no cambia de manera instantánea como pasa con los conjuntos clásicos [37]. Un conjunto difuso en un universo de discurso puede definirse como lo muestra (3.8).

$$A = \mu_A(x)\{(x, \mu_A(x))\}_{x \in U} \quad (3.8)$$

Dónde:

$\mu_A(x)$ es la función de pertenencia de la variable x .

U es el universo en discurso.

Cuanto más cerca este la pertenencia del conjunto A al valor de 1, mayor será la pertenencia de la variable x al conjunto A . En la Fig. 3.5 se muestra un ejemplo de tres conjuntos difusos.

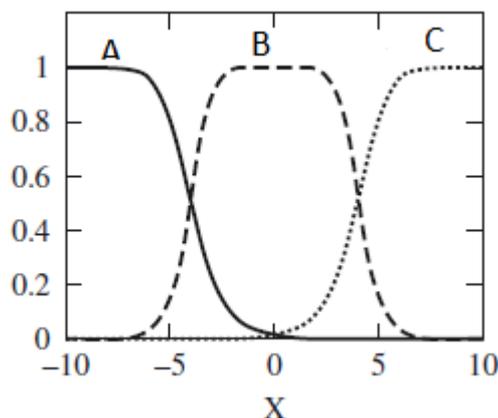


Figura 3.5: Función de pertenencia [3].

Aun cuando cualquier función puede ser válida para definir un conjunto difuso, existen ciertas funciones que son más comúnmente utilizadas por su simplicidad



matemática, entre estas se encuentran las funciones de tipo triangular, Trapezoidal, Gaussiana, etc.

En la Fig. 3.6 se muestra una función de pertenencia del tipo triangular y una función de pertenencia del tipo trapezoidal:

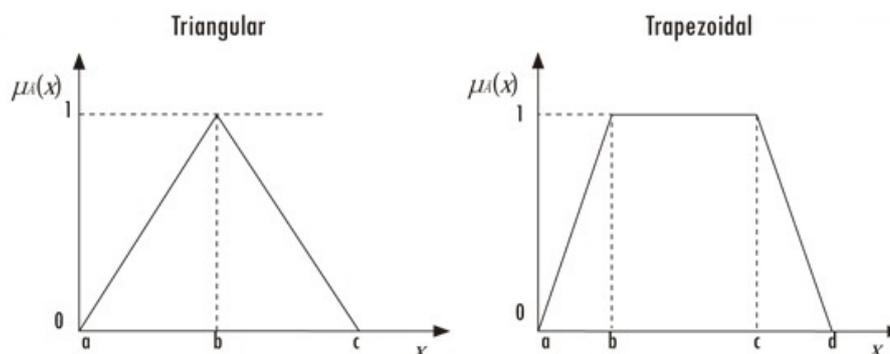


Figura 3.6: Función de pertenencia triangular y trapezoidal

La función de pertenencia triangular queda definida por (3.9).

$$\mu_A(x) = \begin{cases} 0 & \text{Si } x \leq a \\ \frac{x-a}{b-a} & \text{Si } a < x \leq b \\ \frac{c-x}{c-b} & \text{Si } b < x < c \\ 0 & \text{Si } x \geq c \end{cases} \quad (3.9)$$

La función de pertenencia trapezoidal queda definida por (3.10).

$$\mu_A(x) = \begin{cases} 0 & \text{Si } x \leq a \\ \frac{x-a}{b-a} & \text{Si } a < x \leq b \\ 1 & \text{Si } b < x \leq c \\ \frac{d-x}{d-c} & \text{Si } c < x < d \\ 0 & \text{Si } x \geq d \end{cases} \quad (3.10)$$

3.2.3. El controlador difuso.

En un sistema controlado por lógica difusa, el valor de entrada es convertido en un valor difuso mediante un proceso llamado Fusificación. A continuación ese valor es introducido a un proceso también lógico difuso llamado Mecanismo de inferencia, en donde es evaluado mediante un conjunto de reglas, provenientes generalmente del conocimiento de un experto o elaboradas de manera empírica; este proceso produce un valor de salida también difuso, el cual es transformado a una salida real,



a través de un proceso llamado Defusificación, el valor que se obtiene de la salida del proceso de Defusificación es utilizado como una señal de control, que se puede ocupar como entrada a una planta o sistema, los cambios en el sistema son medidos y retroalimentados al sistema y el proceso se repite nuevamente [38][39].

El control difuso puede aplicarse a sistemas cuyos modelos matemáticos son muy sencillos así como en sistemas cuyos modelos matemáticos son muy complejos, e inclusive si no se cuenta con un modelo matemático del sistema o es muy difícil de modelar matemáticamente el sistema.

La estructura básica de un controlador difuso se muestra en la Fig. 3.7.

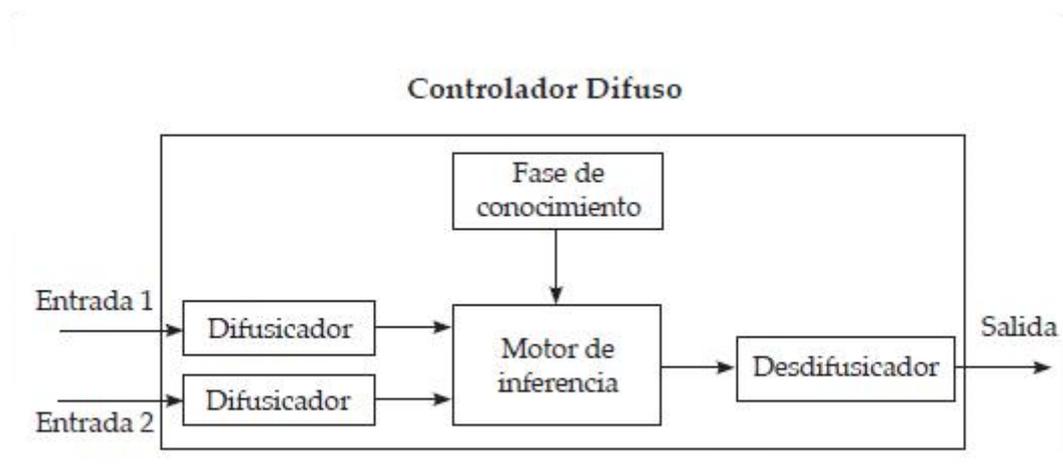


Figura 3.7: Controlador difuso de dos entradas y una salida

3.2.4. Fusificación.

Tiene como objetivo convertir valores crisp o valores reales en valores difusos. En la fusificación se asignan grados de pertenencia a cada una de las variables de entrada con relación a los conjuntos difusos previamente definidos ajustando un rango de valores de las variables de entrada, tomando como base un conocimiento experto [40].

Se pueden utilizar diversas estrategias de fusificación, a continuación se mostrarán la fusificación Triangular y Gaussiana:

Fusificador triangular.

Mapea el elemento x que pertenece a un universo de discurso U a un conjunto difuso A en U , el cual tiene la función de pertenencia que se muestra en (3.11).

$$\mu_A(x) = \begin{cases} 1 - \frac{|x - \tilde{x}|}{b} & \text{Si } |x - \tilde{x}| \leq b \\ 0 & \text{Si } |x - \tilde{x}| > b \end{cases} \quad (3.11)$$



Fusificador Gaussiano.

Mapea el elemento x que pertenece a un universo de discurso U a un conjunto difuso A en U , el cual tiene la función de pertenencia (3.12).

$$\mu_A(x) = \exp\left(-\frac{(x - \tilde{x})^2}{\sigma}\right) \quad (3.12)$$

Base de conocimiento.

La base de conocimiento contiene el conocimiento asociado con el dominio de la aplicación y los objetivos del control. En esta etapa es donde se deben definir las reglas lingüísticas de control que realizarán la toma de decisiones que decidirán la forma en la que debe actuar el sistema.

Inferencia.

La inferencia relaciona los conjuntos difusos de entrada y salida, para representar las reglas que definirán el sistema, en esta etapa se utiliza la información de la base de conocimiento para generar reglas mediante el uso de condiciones, por ejemplo si caso 1 y caso 2, entonces acción 1 [40].

3.2.5. Defusificación.

La defusificación realiza el proceso de adecuar los valores difusos generados en la inferencia en valores crisp, que posteriormente se utilizaran en el proceso de control, en la defusificación se utilizan métodos matemáticos simples como el método del centro del área, Bisección del área y Medio de los máximos [40].

Centro del área.

Genera el centro de gravedad de la posible distribución del conjunto difuso. En el caso de un universo discreto, este método se define en la Fig. 3.8.

$$z_0 = \frac{\sum_{j=1}^n \mu_A(z_j) z_j}{\sum_{j=1}^n \mu_A(z_j)} \quad (3.13)$$

Y en el caso de un universo continuo se define en (3.14).

$$z_0 = \frac{\int \mu_A(z) z dz}{\int \mu_A(z) dz} \quad (3.14)$$

En la Fig. 3.8 se presenta el centro del área, es importante notar que en esta figura no se muestran los conjuntos difusos de forma completa ya que se supone que las reglas de inferencia disparan algunos de estos conjuntos, y entonces se muestra únicamente la sección del área que activaran estos conjuntos según las reglas de



inferencia, el valor de z que se proyecta sobre el eje de las abscisas es el valor certero que toma este conjunto de áreas, y es esta la salida del controlador difuso:

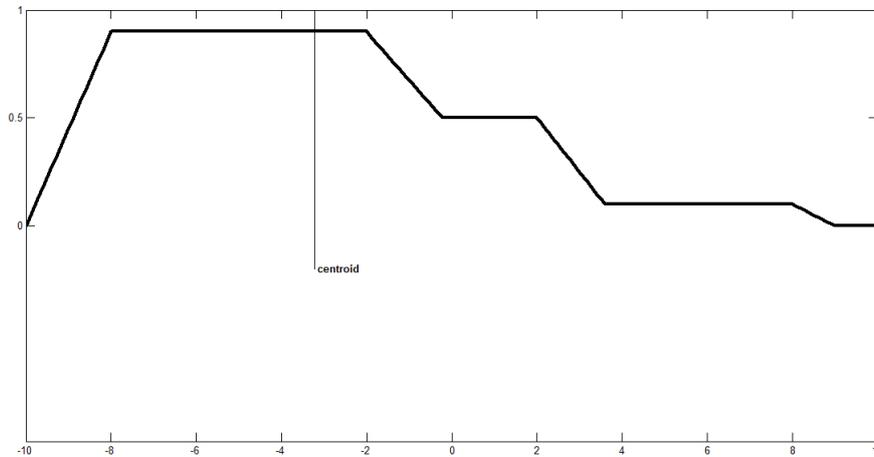


Figura 3.8: Centro del área

Bisección del área.

Este método parte el área en dos regiones con la misma área. El método de la bisección del área satisface (3.15).

$$\int_{\alpha}^{z_0} \mu_A(z) dz = \int_{z_0}^{\beta} \mu_A(z) dz \quad (3.15)$$

Dónde:

$$\alpha = \min \{z | z \in W\}$$

$$\beta = \max \{z | z \in W\}$$

En la Fig. 3.9 se muestra la bisección del área de tres conjuntos triangulares, nótese que el valor de 39.9 es el valor certero que se tiene como resultado de la defusificación:

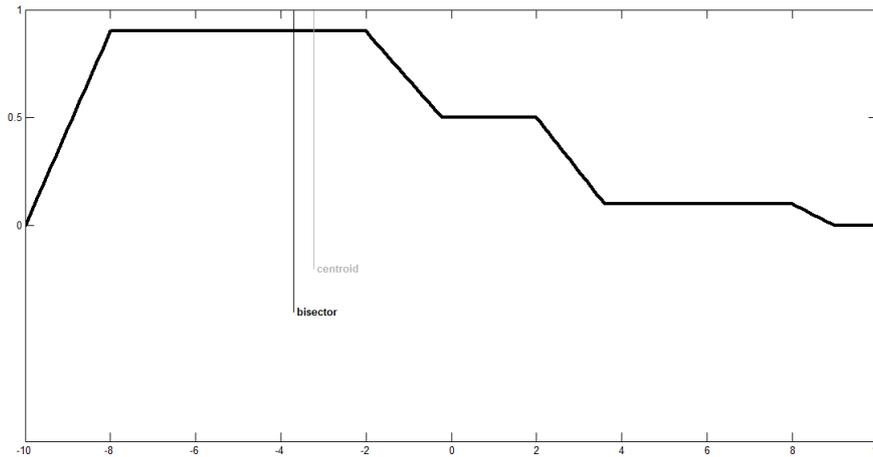


Figura 3.9: Bisección del área de tres conjuntos triangulares

Medio de los máximos.

Es el resultado del promedio de maximizar z donde la función de membresía alcanza un máximo.

En el caso de un universo continuo se tiene que el medio de los máximos satisface (3.16).

$$z_0 = \frac{\int_{z'}^{z''} z dz}{\int_{z'}^{z''} dz} \quad (3.16)$$

Dónde:

z' Es el valor mínimo proyectado sobre el eje de las abscisas de los valores máximos de la función de pertenencia.

z'' Es el valor máximo proyectado sobre el eje de las abscisas de los valores máximos de la función de pertenencia.

En el caso de un universo discreto se tiene que el medio de los máximos satisface (3.17).

$$z_0 = \sum_{j=1}^k \frac{z_j}{k} \quad (3.17)$$

En la Fig. 3.10 se muestra el medio de los máximos.

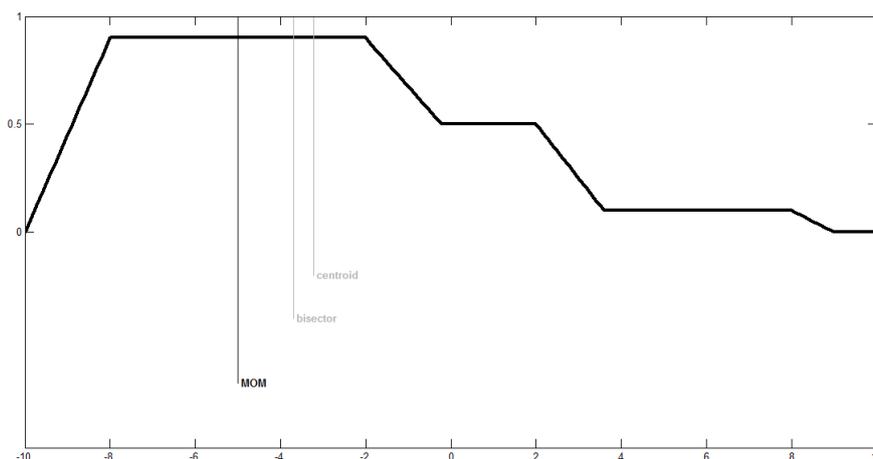


Figura 3.10: Medio de los máximos

3.3. REDES DE COMPUTADORAS Y CONECTIVIDAD

3.3.1. Red de computadoras

Una red de computadoras Fig. 3.11, es también llamada red de ordenadores o red informática, es un conjunto de equipos conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios. En sus orígenes las redes de computadores se concebían como más de una computadora y estas conectadas entre sí, estas redes de computadoras podían únicamente consultar información dentro de su área de trabajo. De tal manera que existían muchas redes de computadoras independientes entre sí, y cada una de ellas contenía sus propios recursos. La tendencia era unir estas redes de computadoras independientes, para generar cada vez una red más grande, ya que cada red de computadoras compartía algunos recursos o algún tipo de información. En sus orígenes nadie pensó en establecer un protocolo estándar de comunicación de datos. Cada red de computadoras debía tener las computadoras lo más similares posible, para que los programas generados para compartir información, pudieran trasladarse de forma transparente de una computadora a otra con el mínimo de modificaciones. Por esta razón las grandes compañías generaban su propia tecnología de comunicaciones en red, exclusiva para el tipo de computadoras que ellas mismas producían. Se establecía inclusive protocolos de comunicación que se encontraban optimizados para la arquitectura de la red y la tecnología propia de cada compañía, es por esta razón que existían diversos protocolos de comunicación así como diversas maneras de conexión de computadoras [41]. En términos muy generales internet se define también como una red de redes de computadoras, compartiendo muchos tipos de información, recursos y servicios.

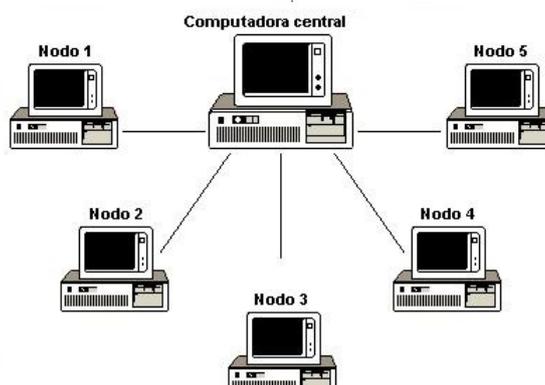


Figura 3.11: Red de Computadoras

REDES LAN Una red LAN Fig. 3.12, es una red de alta velocidad tolerante a fallas, que comunica computadoras en un área relativamente pequeña. Típicamente conecta estaciones de trabajo, computadoras personales y otros dispositivos [4].

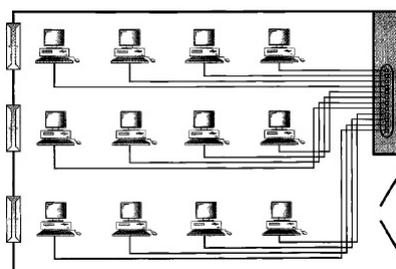


Figura 3.12: Red de área local (LAN,Local Area Network)[4].

REDES WAN Una red WAN Fig. 3.13, es una red de alta velocidad (banda ancha) que da cobertura en un área geográfica extensa, una red WAN es también una colección de LAN's dispersas en un área extensa interconectadas entre sí, esta área puede llegar a extenderse hacia un país, un continente o el mundo entero [4].

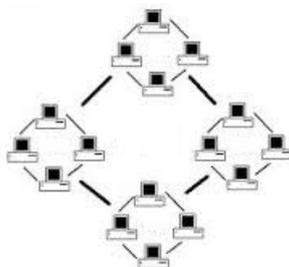


Figura 3.13: Red de área extensa (WAN,Wide Area Network)

Las Redes de Computadoras ofrecen a los usuarios de computadoras acceso a



dispositivos compartidos y aplicaciones, intercambio de archivos y comunicación entre computadoras vía correo electrónico local y otras aplicaciones [41].

3.3.2. Métodos de transmisión en redes LAN.

La transmisión de datos LAN caen dentro de tres clasificaciones: Unicast, Multicast y Broadcast, en cada uno de los tipos de transmisión un paquete es enviado a uno o varios nodos.

Unicast: Un paquete es enviado desde un único emisor hacia un único receptor a través de la red, primero el emisor dirige el paquete usando la dirección del receptor, el paquete es entonces enviado sobre la red, y finalmente la red pasa el paquete a su destino o receptor.

Multicast: Un paquete es enviado desde un único emisor hacia un grupo de destinatarios específicos de la red. Primero el emisor direcciona el paquete, usando la dirección de multicast. El paquete es enviado entonces sobre la red, la cual hace copias del paquete y las envía a cada receptor que forma parte de la red multicast.

Broadcast: Un paquete es enviado desde un único emisor hacia todas las estaciones o nodos en la red, en este tipo de transmisión el emisor direcciona el paquete usando la dirección broadcast. El paquete es entonces enviado sobre la red, la cual hace copias del paquete y envía una copia a cada estación sobre la red.

3.3.3. Capas de Protocolos y sus Servicios

Una arquitectura en capas nos permite estudiar y analizar una parte bien específica y muy bien definida de un sistema más grande y complejo. Para proporcionar una estructura al diseño de protocolos de red, los diseñadores de redes organizan los protocolos en capas. De esta manera cada protocolo pertenece a una de las capas, y es de especial interés el servicio que ofrece una capa a la capa superior, dicho en otras palabras, las capas proporcionan una forma estructurada de estudiar y organizar los componentes del sistema. Cuando los protocolos de las distintas capas se toman en conjunto se habla de pila de protocolos. La pila de protocolos de internet consta de cinco capas: capa física, capa de enlace, capa de red, capa de transporte y capa de aplicación. Como se muestra en la Tabla 3.2.

CAPA DE APLICACIÓN.

La capa de aplicación es donde se ubican las aplicaciones de red y los protocolos de red. La capa de aplicaciones de red incluye protocolos como HTTP (que permite la solicitud y transferencia de datos web), SMTP (que permite la transferencia de mensajes de correo electrónico) y FTP (que permite la transferencia de archivos entre dos sistemas terminales).





Tabla 3.2: La pila de protocolos TCP/IP

CAPA DE APLICACIÓN
CAPA DE TRANSPORTE
CAPA DE RED
CAPA DE ENLACE
CAPA FÍSICA

CAPA DE TRANSPORTE.

La capa de transporte en internet es la etapa en donde se transporta los mensajes de la capa de aplicación entre los puntos terminales de la aplicación. Existen dos protocolos de transporte llamados TCP y UDP, pudiendo cada uno de estos transportar mensajes de la capa de aplicación. TCP por su parte ofrece un servicio orientado a la conexión. Este servicio proporciona un suministro garantizado de los mensajes de la capa de aplicación al destino y un mecanismo de control de flujo (es decir, adaptación de las velocidades del emisor y el receptor). Por su parte el protocolo UDP proporciona un servicio sin conexión, de esta forma UDP es un servicio básico que no ofrece ninguna fiabilidad ni control de flujo. Es importante mencionar que a los paquetes de la capa de transporte se les denomina segmentos.

CAPA DE RED.

La capa de red es la responsable de trasladar los paquetes de la capa de red (conocidos como datagramas), de un host a otro. El protocolo de transporte (TCP o UDP) de internet de un host origen pasa un segmento de la capa de transporte y una dirección de destino a la capa de red y finalmente la capa de red proporciona el servicio de suministrar el segmento a la capa de transporte del host destino. La capa de red de internet incluye al conocido protocolo IP, que es el encargado de definir los campos del datagrama, así como la forma en que actúan los sistemas terminales y los ruteadores sobre estos campos.

CAPA DE ENLACE.

La capa de red se encarga de encaminar un datagrama a través de una serie de ruteadores entre el transmisor y el emisor, la capa de red confía en los servicios de la capa de enlace la cual es encargada de trasladar un paquete de un nodo (host o ruteador) al siguiente nodo de la ruta. En concreto en cada nodo la capa de red pasa el datagrama a la capa de enlace, que entrega el datagrama al siguiente nodo existente a lo largo de la ruta, y una vez estando el paquete en el siguiente nodo la capa de enlace pasa el datagrama a la capa de red. Los servicios proporcionados por la capa de enlace dependen del protocolo de la capa de enlace concreto que se emplee en el enlace, entre los protocolos de la capa de enlace se incluyen Ethernet, WiFi y el protocolo punto a punto (PPP, point to point protocol). Puesto que normalmente los datagramas atraviesan varios enlaces para viajar del emisor al receptor un datagrama



puede ser manipulado por diferentes protocolos de la capa de enlace a través de los distintos enlaces disponibles a lo largo de la ruta. Los paquetes de la capa de enlace se denominan tramas.

CAPA FÍSICA.

La capa física es la etapa en donde se mueven los bits individuales de la trama de un nodo al siguiente. Los protocolos de esta capa son dependientes del enlace, por lo tanto, dependen del medio de transmisión del enlace (cable de cobre de par trenzado o fibra óptica monomodo), en cada caso los bits se desplazan en forma diferente a través del enlace.

3.4. PROTOCOLO TCP/IP.

El crecimiento de los usuarios de las redes de computadoras y la necesidad de intercambio de información entre estos usuarios, obligó a establecer un estándar de comunicaciones que fuese independiente de la tecnología de interconexión, tipo de computadoras, velocidad de conexión o sistema operativo. El estándar de comunicación entre sistemas de computadoras ha sido y será durante muchos años el conjunto de protocolos de pila TCP/IP, se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP). TCP/IP fue desarrollado y demostrado por primera vez en 1972 por el Departamento de Defensa de los Estados Unidos, ejecutándolo en ARPANET, una red de área extensa de dicho departamento. Por su parte TCP se dice que está orientado a la conexión porque antes de que un proceso de la capa de aplicación pueda comenzar a enviar datos a otro los dos procesos deben primero establecer una conexión entre ellos, es decir tienen que enviarse ciertos mensajes preliminares para definir los parámetros de la transferencia de datos que van a llevar a cabo a continuación. el protocolo TCP se ejecuta únicamente en los sistemas terminales y no en los elementos intermedios de la red (ruteadores y switches de la capa de enlace), por lo tanto los elementos intermedios de la red no mantienen el estado de la conexión TCP. Una conexión TCP es una conexión punto a punto, es decir, entre un único emisor y un único receptor, por esta razón la transferencia de información desde un emisor a muchos receptores en una única operación no es posible. el protocolo IP por su parte es un protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red de la pila de protocolos de internet, y su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo no orientado a la conexión, esto quiere decir que la entrega de paquetes se hará lo mejor posible, pero garantizado poco. Por ejemplo al no garantizar nada sobre la recepción del paquete este podría llegar dañado, en otro orden con respecto a otros paquetes, duplicado o simple y sencillamente no llegar. Si es necesaria fiabilidad en la recepción de paquetes esta es proporcionada por el protocolo de transporte TCP. Las cabeceras IP contienen las direcciones de las máquinas de origen y destino, estas



direcciones son usadas por los enrutadores para decidir el tramo de red por el que se enviarán los paquetes [41] [42]. si se desea ver la estructura de un segmento TCP y la estructura de un datagrama IP vease el Anexo A.

3.5. Resumen del capítulo.

En este capítulo se presentó los fundamentos teóricos bajo los cuales se fundamenta el desarrollo de este trabajo de tesis. La sección 3.1 presentó un estudio básico de la teoría de control clásica, en donde se aborda que es un sistema de lazo abierto, un sistema de lazo cerrado, cual es la respuesta típica de un sistema de control a lazo cerrado y sus características, se presentó el esquema de control PID y un método implantado por Ziegler and Nichols para sintonizar controladores PID.

La sección 3.2 se mencionó que es un controlador difuso, que es fusificación y que es defusificación.

En las secciones 3.3 y 3.4 se presentó un estudio sobre las redes de computadoras y las capas de protocolos, así como un estudio en especial al protocolo TCP/IP.

Capítulo 4

ARQUITECTURA Y FUNCIONALIDAD DEL SISTEMA DISTRIBUIDO.

Dentro de este capítulo se describe las principales partes y funciones que se requieren para que el sistema de supervisión y control pueda ser útil a las necesidades de operación, y las características mínimas que debe cumplir el sistema de control para una operación eficiente. la sección 4.1 explica en forma breve las principales funciones del sistema de supervisión y control; la sección 4.2 contiene una explicación de los componentes del sistema y el tipo de comunicación entre los mismos, la sección 4.3 explica el funcionamiento de los componentes del sistema y finalmente la sección 4.4 explica las restricciones de tiempo para la transmisión y recepción de datos entre los elementos del sistema.

4.1. Funciones básicas del sistema

las principales funciones de este sistema de supervisión y control son:

- Recabar, almacenar y mostrar información en forma continua y confiable, correspondiente a las variables del proceso de nivel.
- Ejecutar acciones de control iniciadas por el operador, tales como: variar el valor de la referencia, cambiar los parámetros del controlador PID e inclusive cambiar el tipo de controlador que opera sobre la planta.

4.2. Principales componentes

Los principales componentes del sistema de supervisión y control son las computadoras, que actúan como clientes o estaciones de operación y un servidor el cual tiene la capacidad de comunicarse con los clientes. Se trabaja con tres clientes los



cuales están conectados a un servidor, cabe señalar que el servidor puede atender hasta n clientes. El establecimiento de cada una de las conexiones se lleva a cabo usando el protocolo TCP/IP véase Cap.3.

La Fig. 4.1 muestra el servidor con tres clientes conectados, en donde el sentido de las flechas representan el flujo de información en forma bidireccional, es decir, que los clientes pueden enviar y recibir datos al servidor, así como el servidor puede enviar y recibir datos a los múltiples clientes.

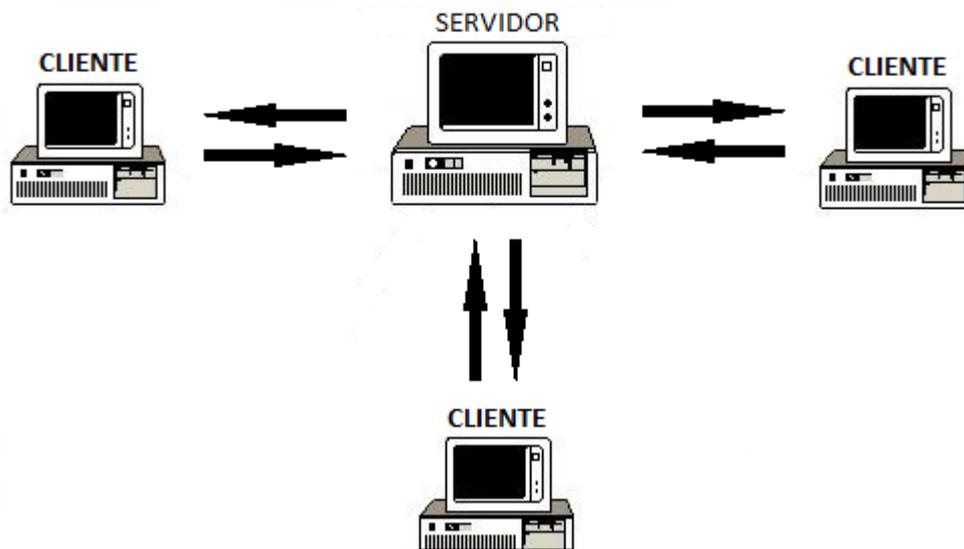


Figura 4.1: Arquitectura Cliente Servidor TCP/IP.

4.3. Funcionalidad del sistema

Se estableció una arquitectura con un servidor y múltiples clientes en donde el servidor tiene la capacidad de comunicarse con cualquiera de los clientes, y cada cliente se comunica con el servidor. la comunicación es en forma bidireccional, esto quiere decir que el servidor puede enviar información a cada cliente y también puede recibir información del proceso que se está simulando en cada uno de los clientes. así mismo cada cliente puede enviar información al servidor y también puede recibir información del servidor.

Establecimiento de la conexión

El cliente solicita una conexión con el servidor enviando un mensaje con una clave de autenticidad, el servidor una vez que recibe este mensaje verifica la clave, si esta clave coincide con la clave que tiene el servidor como clave de autenticidad, entonces se establece la conexión entre el cliente y el servidor.



el servidor puede establecer una conexión con cada cliente y puede soportar hasta n clientes, y cada cliente únicamente se puede comunicar con el servidor, en otras palabras, la comunicación entre clientes no está permitida. para establecer esta conexión se usó el protocolo de comunicación TCP/IP.

Intercambio de información

Una vez que se ha establecido la conexión, entonces se tiene la capacidad de intercambiar información entre el servidor y el cliente. El proceso de intercambio de información se hace a petición del servidor, ya que el servidor pregunta por variables que se están ejecutando en el cliente, como ejemplo de esto es la referencia actual del PNS (Proceso de Nivel Simulado), es decir, que el servidor envía un mensaje al cliente en el cual se pide que envíe información de la referencia actual en el PNS. O en su defecto también se puede mandar a modificar el valor actual de la referencia por un valor propuesto por el servidor, es decir que el servidor envía un mensaje al cliente en el cual se le pide que modifique el valor actual de la referencia. De esta forma se consigue supervisar y controlar el valor de la referencia.

4.3.1. Funcionamiento del servidor

El principal funcionamiento del servidor es recabar información como: el valor actual de la referencia, los parámetros del controlador etc. de cada uno de los clientes y tiene la capacidad de solicitar al cliente cambiar estos parámetros.

En la Fig. 4.2 se muestra los bloques que integran el diagrama de flujo principal del funcionamiento del servidor. como se puede ver, es necesario inicializar las variables del servidor, así como identificar las conexiones, una vez que se establece la conexión con un cliente entonces se puede leer o escribir desde el servidor a cualquiera de los clientes conectados al sistema. Es importante observar que el servidor inicializa las variables del sistema en una sola ocasión, y posteriormente se queda a la espera de que exista una conexión con un cliente para que la lectura y escritura mediante TCP/IP se pueda llevar a cabo según lo disponga el operador.

Identificación de conexiones.

Para la identificación de las conexiones el servidor está escuchando continuamente en espera de que un cliente solicite una conexión, una vez que el cliente solicita una conexión el servidor verifica que el cliente tenga la clave de identificación, con esto se asegura que el cliente que se conecte al sistema sea realmente un computador que pertenece al sistema. Si el cliente que se conecta no pertenece al sistema el servidor cierra la conexión de forma inmediata ó si existe una falla en la conexión el servidor también cierra la conexión, si el cliente tiene la clave de identificación y además no tiene error en la comunicación el servidor identifica la conexión; el diagrama de flujo que representa este proceso se muestra en la Fig. 4.3.

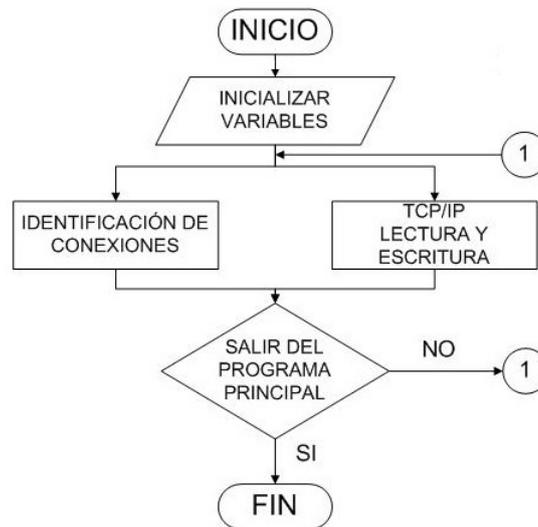


Figura 4.2: Diagrama de flujo servidor.

TCP/IP lectura y escritura.

En esta parte del programa, el servidor envía información de ejecución (Tx) al cliente o en su defecto se envía que información es la que el servidor espera recibir del cliente (Rx), para este caso se envía o se recibe información a solicitud del servidor. Si el cliente al que se está enviando información no se encuentra conectado y se intenta enviar información, entonces aparecerá en la pantalla un mensaje de aviso diciendo que el cliente no se encuentra conectado. en la Fig. 4.4 se muestra un diagrama de flujo de la acción de escribir información (Tx) tomando como emisor el servidor y como receptor el cliente.

Para recibir información, el servidor verifica un arreglo con la identificación de las conexiones, así se comprueba si el nodo está conectado, si el nodo se encuentra conectado entonces se extrae la información que está enviando; si se desconectó entonces envía un mensaje el cual contiene la información del estado de la conexión, si el servidor lee esta información, entonces termina la conexión modificando el arreglo de las conexiones; si no se encuentra una conexión entonces el servidor no hace nada de tal forma que únicamente se lee la información de los clientes conectados. El diagrama de flujo que se muestra en la Fig.4.5 muestra como el servidor lee la información (Rx) que los clientes envían mediante el protocolo TCP/IP.

4.3.2. Funcionamiento del cliente.

En el cliente se simula el proceso de nivel, por esta razón es aquí donde se recolecta la información de las variables del proceso, y se reporta al servidor en forma constante cuando este último las demande; también se programan los algoritmos de control para el mismo proceso y se modifican los parámetros de estos controladores

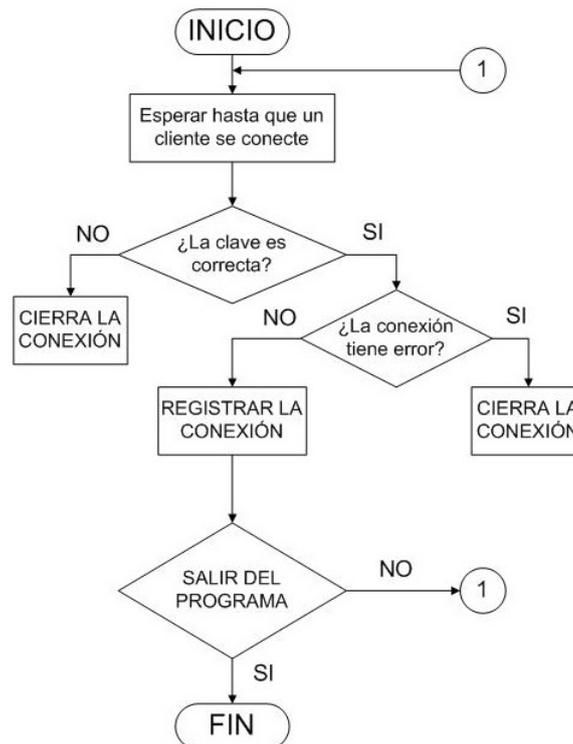


Figura 4.3: Diagrama de flujo identificación de las conexiones.

bajo mandato del servidor, de tal manera que en cada nodo del sistema se ejecuta el proceso de nivel simulado con los algoritmos de control, pero es necesario leer y escribir información al servidor; la información que lee el cliente mediante el protocolo de comunicación TCP/IP es información que contiene comandos de ejecución que afecta a la dinámica del sistema de nivel simulado, o en su defecto contiene la información de los parámetros que el servidor solicita para su base de datos. en la Fig. 4.6 se muestra la estructura de funcionamiento del cliente mediante un diagrama de flujo.

TCP/IP lectura.

El cliente está a la espera de recibir mensajes que el servidor envíe, y queda en espera de forma indefinida; una vez que se lee un mensaje es necesario distinguir el tipo de mensaje que se está recibiendo, si el mensaje es un mensaje de ejecución de instrucciones de control la información se envía a una base de datos, y esta a su vez modifica los valores indicados del proceso de nivel, si la información que se lee es una solicitud de información, estos parámetros le indican al cliente que información es la que se debe enviar al servidor. En la Fig. 4.7 se muestra un diagrama de flujo que indica el proceso de recepción de datos mediante el protocolo TCP/IP en el cliente.

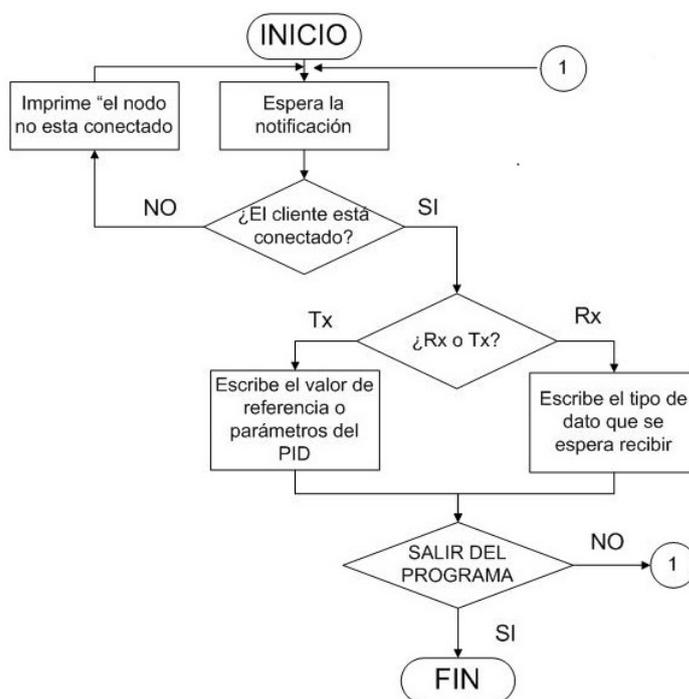


Figura 4.4: Diagrama de flujo escribir de servidor a cliente.

TCP/IP escritura.

El cliente envía mensajes al servidor en forma constante a través de la red, la información que se envía es tomada de la base de datos, y esta información es la que el servidor solicitó previamente; cuando se cierra la conexión el cliente deja de enviar la información para enviar un mensaje, el cual le indica al servidor que la conexión se ha cerrado. En la Fig. 4.8 se puede observar el diagrama de flujo que corresponde al proceso de enviar información al servidor por medio de una conexión TCP/IP.

4.4. Restricciones temporales.

Dado que el protocolo de comunicación sobre el cual se trabajó es el protocolo TCP/IP, los algoritmos de transmisión y recepción de datos entre el servidor y los múltiples nodos que se pueden conectar con el servidor, se diseñaron pensando en que la red de comunicación no puede ser inundada con información, dado a que esta podría llegar a su destino en un tiempo no exitoso. Considerando que un tiempo entre 1 y 5 minutos es un tiempo exitoso, un tiempo menor que 1 minuto se considera como un tiempo óptimo y un tiempo mayor a 5 minutos se considera como un tiempo no exitoso. Es por esta razón que el sistema se diseñó para que el cliente pudiera trabajar sin conexión al servidor.

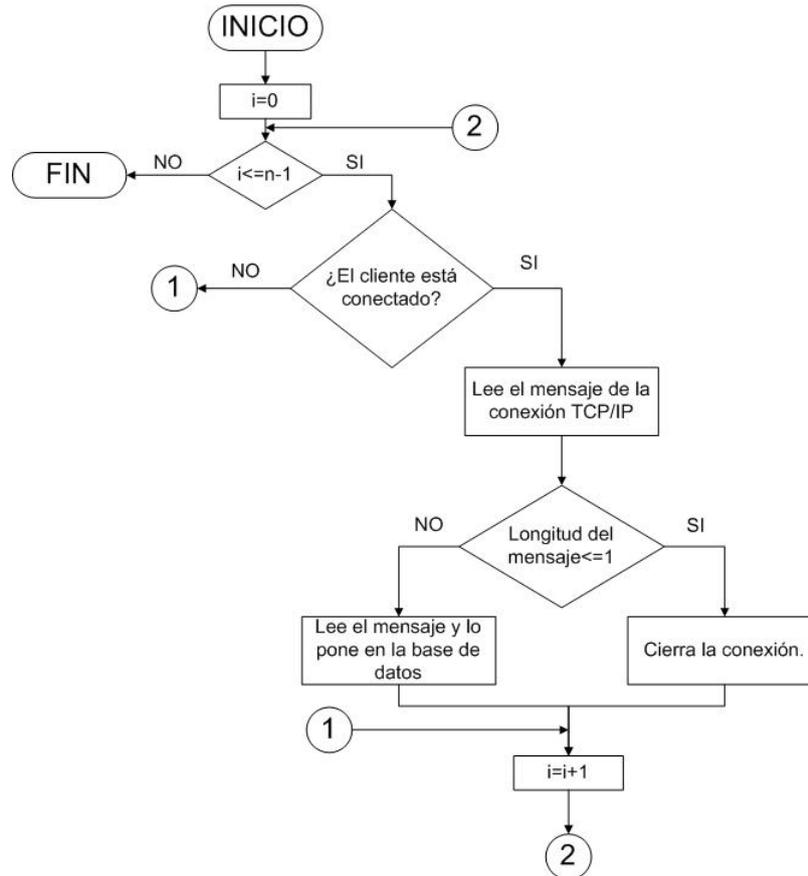


Figura 4.5: Diagrama de flujo leer desde servidor a cliente.

4.5. Resumen del capítulo.

En este capítulo se presentó una explicación de la manera en como el sistema opera. La sección 4.1 presentó en forma resumida las principales funciones del sistema, la sección 4.2 presenta cuales son los principales componentes del sistema, así como un diagrama en el que muestra de forma explícita los principales componentes y la forma en la cual se comunican estos, la sección 4.3 presentó de forma explícita cual es la funcionalidad del sistema, como se establece la comunicación entre un cliente y el servidor, y como es el flujo de datos entre estos elementos, la sección 4.4 explicó cuales son las restricciones de tiempo del sistema.

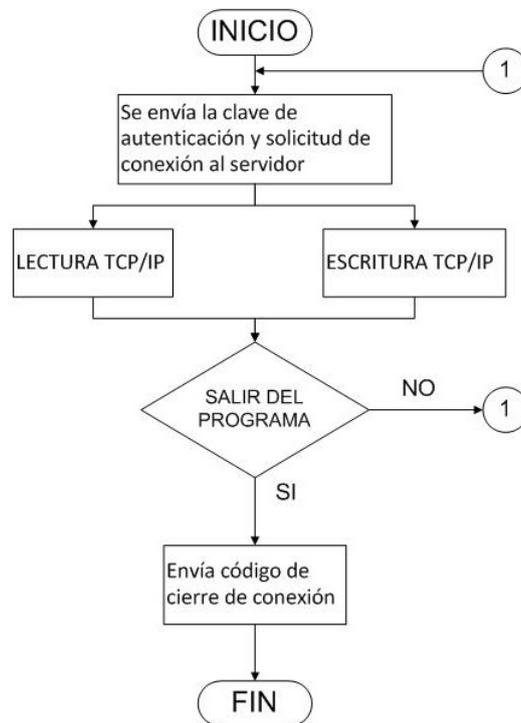


Figura 4.6: Diagrama de flujo cliente.

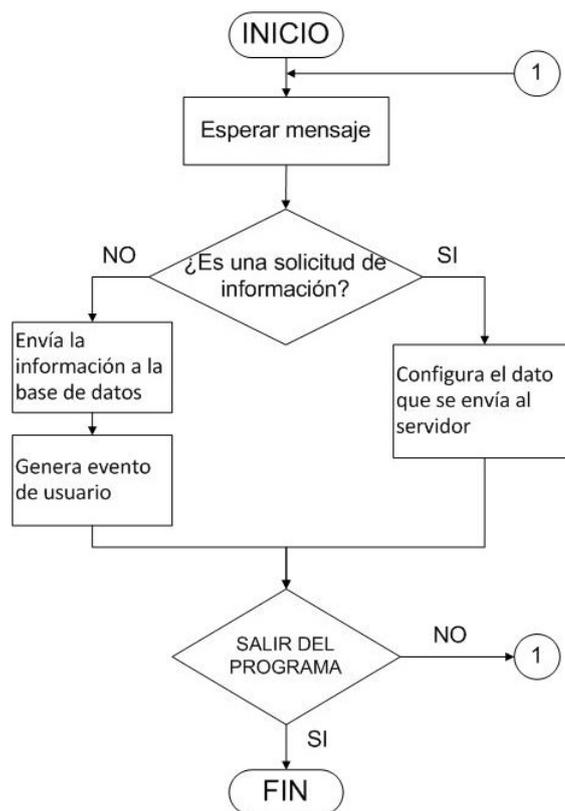


Figura 4.7: Diagrama de flujo lectura TCP/IP cliente.

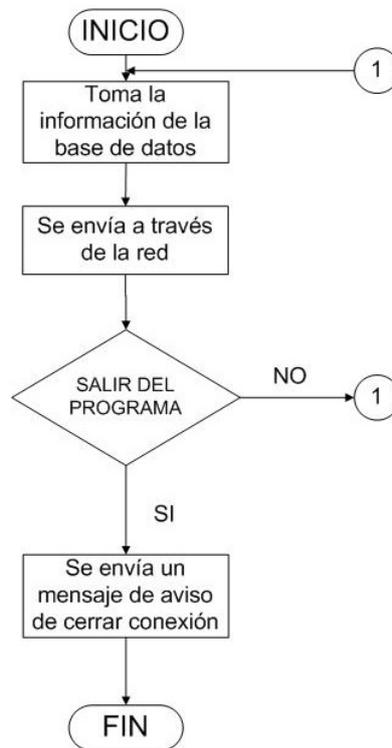


Figura 4.8: Diagrama de flujo escritura TCP/IP cliente.

Capítulo 5

CONTROLADOR DIFUSO PARA UN PROCESO DE NIVEL SIMULADO (PNS).

En este capítulo se desarrolla dos controladores difusos para el proceso de nivel simulado, cada controlador está formado con dos entradas y una salida, cada una de las entradas se compone de 5 conjuntos difusos y una salida con 7 conjuntos difusos. La sección 5.1 explica cuál es el modelo matemático del sistema de los tanque interconectados; la sección 5.2 contiene cual es el lazo de control que se simuló; la sección 5.3 explica el diseño del controlador difuso y finalmente la sección 5.4 contiene la simulación de los controladores que se diseñaron.

5.1. El problema de dos tanques interconectados.

Se tiene dos tanques en serie conectados en línea. El objetivo de este experimento es controlar el nivel en el tanque 2, regulando el caudal a la entrada del tanque 1, se diseñó un controlador con lógica difusa, el cual permite aumentar o disminuir el punto de referencia, y la variable controlada seguirá el punto de referencia, hasta llevar el error (la diferencia entre la altura del líquido deseada y la altura real) a un valor de cero, esto quiere decir que la altura real alcanza la altura deseada y se mantiene estable en ese valor.

En la Fig. 5.1 se muestra un esquema del sistema con dos tanques interconectados en línea.

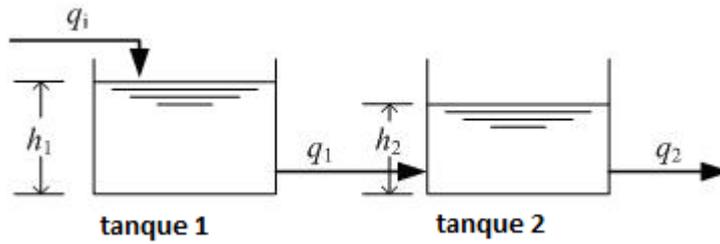


Figura 5.1: Sistema de dos tanques interconectados.

Donde:

q_i es el caudal que entra al tanque 1 y es el caudal a manipular o variable manipulada.

q_1 es el caudal de salida del tanque 1 y caudal de entrada al tanque 2.

q_2 es el caudal de salida del tanque 2.

h_1 es la altura del líquido en el tanque 1.

h_2 es la altura del líquido en el tanque 2.

Este sistema se puede modelar matemáticamente por medio de ecuaciones diferenciales, y estas son (5.1) y (5.2) como se puede ver en [30][31].

$$A_{T_1} * \frac{dh_1}{dt} = q_i - q_1 \quad (5.1)$$

$$A_{T_2} * \frac{dh_2}{dt} = q_1 - q_2 \quad (5.2)$$

estas ecuaciones se pueden expresar como se muestra en (5.3) y (5.4), para una planta no lineal:

$$\frac{dh_1}{dt} = \frac{q_i}{A_{T_1}} - \frac{A_{o_1}}{A_{T_1}} * \sqrt{2g(h_1 - h_2)} \quad (5.3)$$

$$\frac{dh_2}{dt} = \frac{A_{o_1}}{A_{T_2}} * \sqrt{2g(h_1 - h_2)} - \frac{A_{o_2}}{A_{T_2}} * \sqrt{2gh_2} \quad (5.4)$$

Un modelo lineal de la planta se propone en [30] donde el valor de resistencia de las válvulas de flujo es una constante, permitiendo linealizar alrededor de un punto de operación, suponiendo que se tienen pequeñas variaciones de flujo. De esta manera se puede tener el modelo lineal de la planta como se muestra en (5.5) y (5.6).

$$A_{T_1} * \frac{dh_1}{dt} = q_i - \frac{h_1 - h_2}{R_1} \quad (5.5)$$



$$A_{T_2} * \frac{dh_2}{dt} = \frac{h_1 - h_2}{R_1} - \frac{h_2}{R_2} \quad (5.6)$$

Esta planta tiene un modelo lineal así como un modelo no lineal; de cualquier manera el controlador difuso que se implementó en esta tesis no toma ninguno de estos dos modelos para su diseño, ya que se estableció un controlador difuso tipo Mamdani con dos entradas correspondientes al error y a la velocidad de cambio del nivel del líquido en el tanque 2, y a la salida del controlador se tiene un incremento en la señal de control o variable manipulada, teniendo finalmente un controlador PD incremental como se plantea en [43]. El control difuso plantea una alternativa para controlar plantas las cuales carecen de un modelo matemático, o son plantas altamente no lineales; razón por la cual se implementó un control con lógica difusa para poder estabilizar la planta en el punto de referencia o valor deseado [44].

5.2. Lazo de control.

Se realizó un controlador con lógica difusa tipo Mamdani, con dos entradas y una salida. En una entrada se tiene el error, que es la diferencia que existe entre el valor deseado o referencia y el valor real de la planta en un momento dado. En la otra entrada se tiene el valor de la velocidad con la que cambia el nivel del líquido en el tanque 2. Y a la salida se tiene el valor del incremento con la que se aumentara el valor de la variable de control. Dicho en otras palabras se tiene un controlador PD incremental como se plantea en [43].

La arquitectura del lazo de control que se implementó se muestra en la Fig. 5.2. Donde se puede notar que la salida del controlador se retroalimenta y se suma con el incremento de la señal de control que sale del controlador con lógica difusa. De esta manera la entrada a la planta es la señal de control en el tiempo anterior más el incremento de la señal.

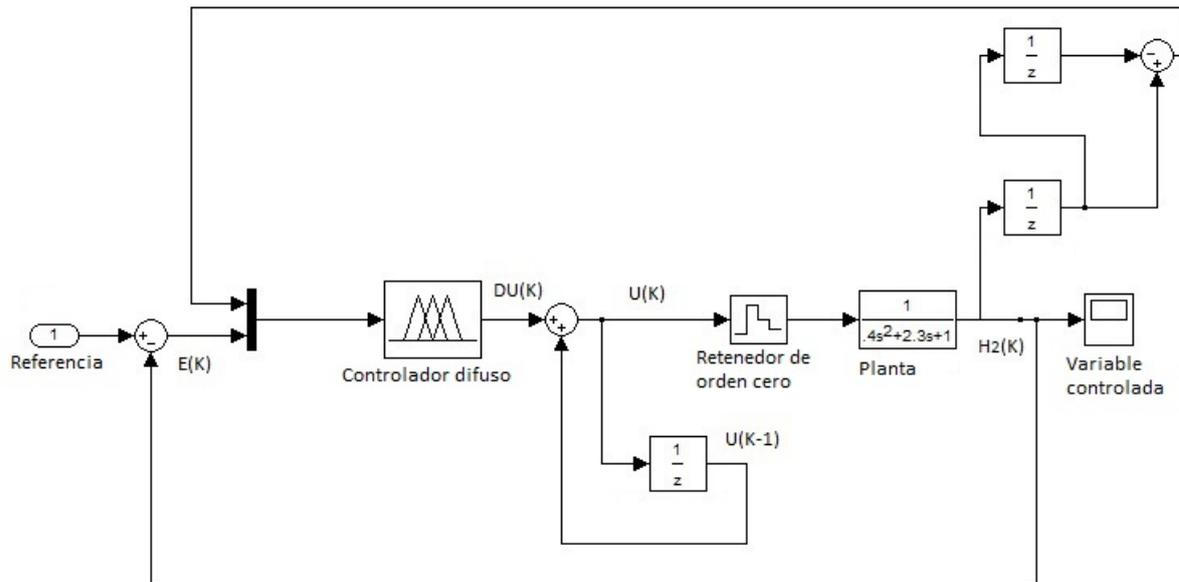


Figura 5.2: Lazo cerrado de control (FLC).

Donde:

- REF es la referencia a la cual se quiere llevar el nivel en el tanque 2.
- $E(K)$ es la diferencia entre la REF y el valor real del nivel de líquido $H2(K)$ en un instante de tiempo determinado.
- $DU(K)$ es el incremento que se añade a un valor anterior de la señal de control $U(K-1)$ generando así el nuevo valor de la señal de control $U(K)$.
- $U(K)$ es la señal de control que entra a la planta.
- $H2(K)$ es el nivel de líquido en el tanque 2.

Este lazo de control adapta una porción de flujo a la entrada del tanque 1 con la finalidad de aumentar en esas proporciones el valor que tiene en ese momento al flujo de entrada hasta lograr adaptar el flujo a la entrada del tanque 1 en un valor tal que el nivel en el tanque 2 se mantenga estable en el valor de referencia. En la literatura regularmente a este tipo de esquema de control se le denomina control incremental ver [43], ya que este control varía en pequeños incrementos la variable de control, hasta ajustarla al valor de referencia y llevar el error a cero.

5.3. Controlador difuso.

El controlador difuso que se estableció para el proceso de nivel simulado (PNS), está basado en un lenguaje natural que se asemeja al pensamiento humano, esto



quiere decir que las variables de entrada al controlador que en este caso es el error y la velocidad con que cambia la altura en el tanque 2, se dividieron en 5 conjuntos difusos, los cuales se identifican por variables lingüísticas y manejan rangos; del rango total que puede ocurrir con la variable de entrada. Este mismo principio se ocupa con la variable de salida del controlador, que es el flujo de entrada al tanque 1, básicamente este principio provee una captura efectiva de lo que ocurre en el mundo real, y esta es la parte esencial de un FLC (Fuzzy Logic Controller) [45] [46].

La variable de entrada correspondiente a la velocidad con la que cambia el nivel del líquido en el tanque 2, se dividió en 5 conjuntos difusos, que se enumeran de izquierda a derecha como NL, NS, Z, PS, PL, respectivamente. Esta división se muestra en la Fig. 5.3.

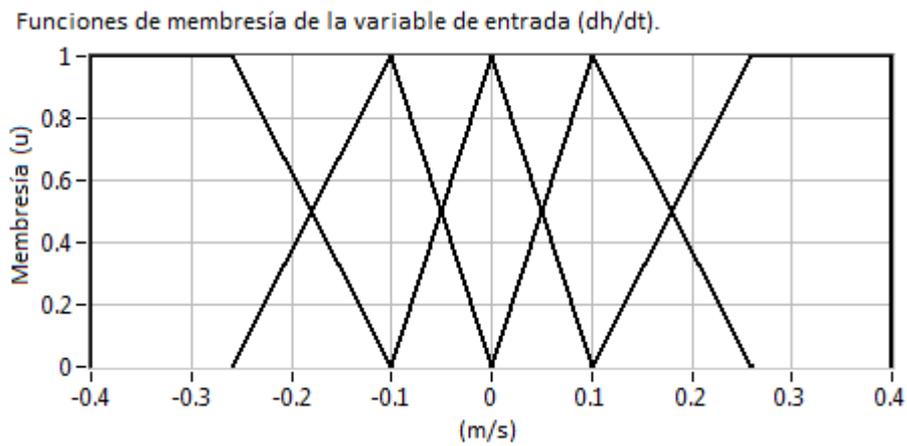


Figura 5.3: Velocidad de cambio de nivel de líquido en el tanque 2.

La variable de entrada correspondiente al error o la diferencia que existe entre la referencia y el nivel que tiene el tanque 2, se dividió en 5 conjuntos difusos, que se enumeran de izquierda a derecha como NL, NS, Z, PS, PL, respectivamente. Esta división se muestra en la Fig. 5.4.

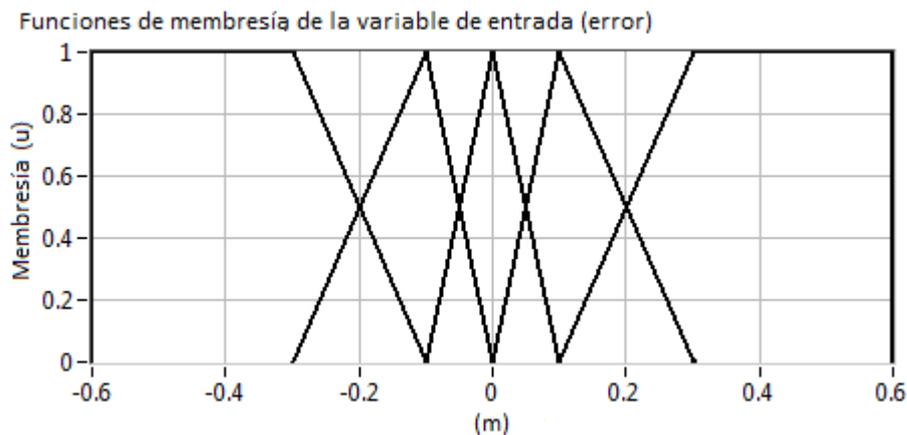


Figura 5.4: Error (Diferencia entre la referencia y el valor real del nivel de líquido).



La variable de salida del controlador, que es la correspondiente al incremento que se dará a la señal de control se dividió en 7 conjuntos difusos, los cuales se enumeran de izquierda a derecha como NL, NM, NS, Z, PS, PM, PL, respectivamente. Esta división se muestra en la Fig. 5.5.

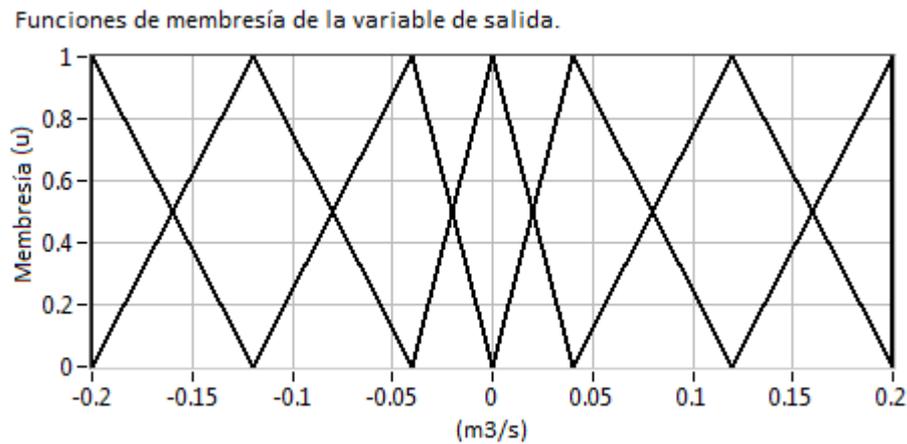


Figura 5.5: Incremento de la señal de control U.

El motor de inferencias de este controlador depende de todas las posibles combinaciones que se pueden dar con las diferentes variables lingüísticas del rango de valores que puede tomar las variables de entrada; con base a estas combinaciones se crea el motor de inferencias (Tabla 5.1). Este motor de inferencias se puede ver como un arreglo matricial, con todas las posibles combinaciones. En las filas y columnas se ponen las particiones o variables lingüísticas y cada celda de la matriz resultante es la parte consecuente del motor de inferencias (véase el capítulo 3).

Tabla 5.1: Motor de inferencias del controlador (FLC).

dh/dt	NL	NS	Z	PS	PL
e	NL	NS	Z	PS	PL
NL	Z	NS	NL	NL	NL
NS	PS	Z	NS	NM	NL
Z	PM	PS	Z	NS	NM
PS	PL	PM	PS	Z	NS
PL	PL	PL	PL	PS	Z

En la Fig. 5.6, se muestra una simulación del comportamiento que tiene el controlador FLC con respecto a toda la gama de valores y posibles combinaciones de valores que se pueden dar, o los eventos que podrían ocurrir y la respuesta que este controlador tendrá para dichos eventos. Estos eventos se conocen como reglas de inferencia, en la Tabla (5.1) se muestra en forma breve un resumen de estas reglas



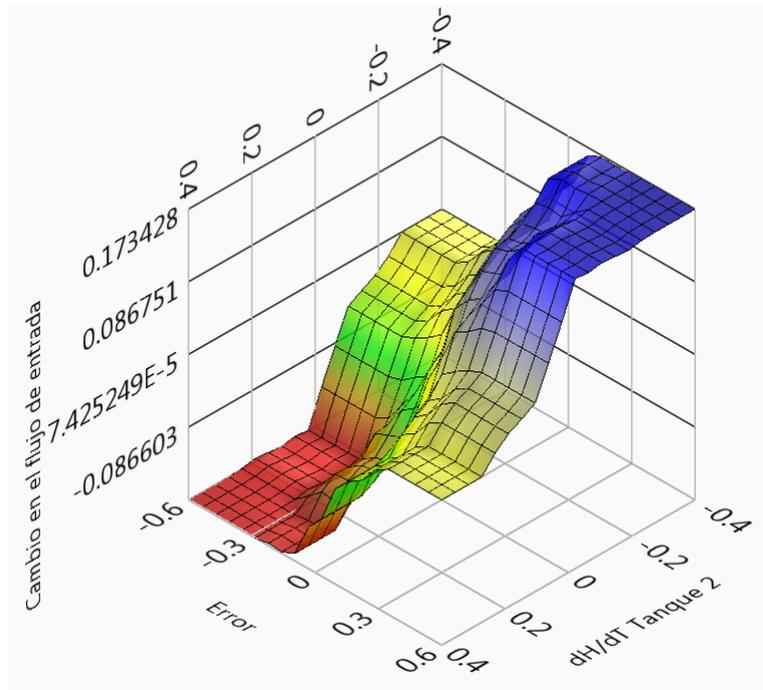


Figura 5.6: Simulación del controlador (FLC).

de inferencia. Se anexa en forma desglosada este conjunto de reglas de inferencia en el Anexo A.

Se realizó también una modificación a los conjuntos que se mencionan anteriormente, esta modificación consistió en generar conjuntos difusos tipo Gaussiano, sustituyendo a los triangulares y trapezoidales, así como en la variable de salida cambiar los conjuntos triangulares por Gaussianos y aumentar el valor de la variable de salida al doble de lo que se tiene con los conjuntos triangulares. La variable de entrada correspondiente a la velocidad con la que cambia el nivel del líquido en el tanque 2. Se dividió en 5 conjuntos difusos, que se enumeran de izquierda a derecha como NL, NS, Z, PS, PL, respectivamente. Esta división se muestra en la Fig. 5.7.

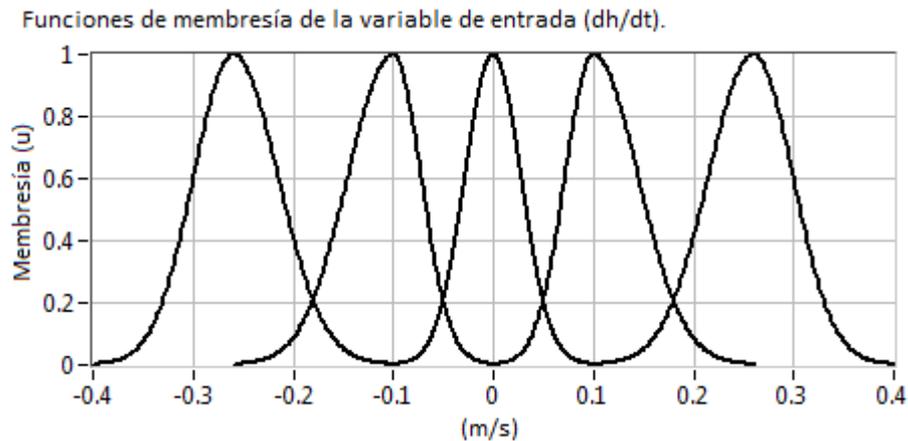


Figura 5.7: Velocidad de cambio de nivel de líquido en el tanque 2 (Gaussiana).

La variable de entrada correspondiente al error o la diferencia que existe entre la referencia y el nivel que tiene el tanque 2, se dividió en 5 conjuntos difusos, que se enumeran de izquierda a derecha como NL, NS, Z, PS, PL, Respectivamente. Esta división se muestra en la Fig. 5.8.

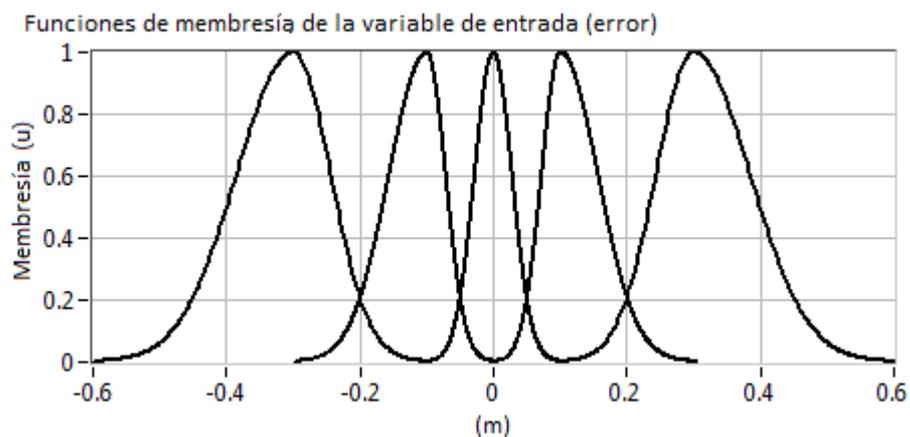


Figura 5.8: Error (Diferencia entre la referencia y el valor real del nivel de líquido)(Gaussiana).

La variable de salida del controlador, que es la correspondiente al incremento que se dará a la señal de control se dividió en 7 conjuntos difusos y se incrementaron al doble, Los cuales se enumeran de izquierda a derecha como NL, NM, NS, Z, PS, PM, PL, respectivamente. Esta división se muestra en la Fig. 5.9.

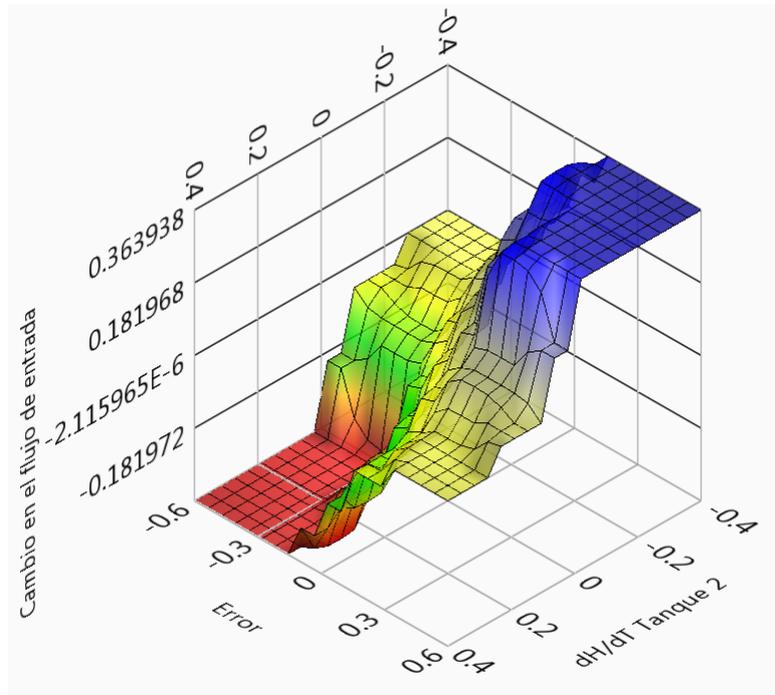


Figura 5.10: Simulación del controlador (FLC)(Gaussiana).

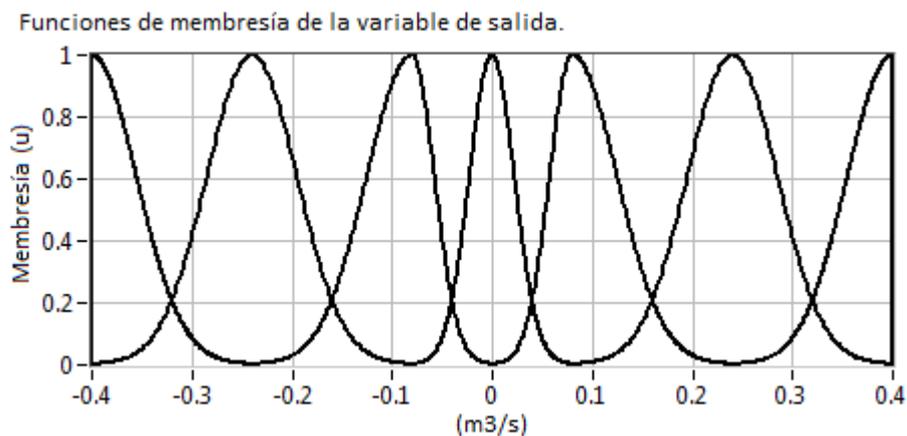


Figura 5.9: Incremento de la señal de control U (Gaussiana).

En la Fig. 5.10, se muestra una simulación del comportamiento que tiene el controlador FLC con funciones de pertenencia del tipo Gaussiana, con respecto a toda la gama de valores y posibles combinaciones de valores que se pueden dar, o los eventos que podrían ocurrir, y la respuesta que este controlador tendrá para dichos eventos. Estos eventos se conocen como reglas de inferencia, en la Tabla 5.1 se muestra en forma breve un resumen de estas reglas de inferencia. Se anexa en forma desglosada este conjunto de reglas de inferencia en el Anexo A.



5.4. Simulación.

Para hacer la simulación de la planta con el controlador difuso fue necesario tener una función de transferencia, ya que a pesar de que esta función de transferencia no se utilizó para el diseño del controlador difuso FLC, si se ocupó para poner el modelo de la planta en diferencias finitas o ecuaciones de diferencias y programar mediante relaciones algebraicas el modelo de la planta. A partir del modelo lineal de la planta (5.5) y (5.6) se obtiene una función de transferencia que relaciona el flujo de entrada q_i , con la salida de la planta como h_2 . Véase la Fig. 5.1.

De esta manera la función de transferencia para el Proceso de Nivel de Líquido de dos tanques interactuantes se muestra en (5.7). Para ver el análisis completo y la deducción de la función de transferencia véase el Anexo A.

$$\frac{H_2(S)}{Q_i(S)} = \frac{R_2}{R_1 C_1 R_2 C_2 S^2 + (R_1 C_1 + R_2 C_2 + R_2 C_1) S + 1} \quad (5.7)$$

En este caso particular se trabaja con los siguientes parámetros:

- $A_1 = 2m$
- $A_2 = 1m$
- $R_1 = 0,4 \frac{seg}{m^2}$
- $R_2 = 0,5 \frac{seg}{m^2}$

Finalmente se tiene la siguiente función de transferencia:

$$\frac{H_2(S)}{Q_i(S)} = \frac{0,5}{0,4S^2 + 2,3S + 1} \quad (5.8)$$

Con el objetivo de poder simular esta planta en un software de computadora, fue necesario convertir esta función de transferencia en el dominio de Laplace a su equivalente en el dominio discreto o en términos de Z, Mediante el uso de la transformada Z, así como agregar un retenedor de orden cero para recuperar la señal. De esta forma la función de transferencia pulso del sistema se muestra en (5.9).

$$\frac{H_2(Z)}{Q_i(Z)} = \frac{0,00142221Z^{-1} + 0,00129229Z^{-2}}{1 - 1,74471Z^{-1} + 0,750137Z^{-2}} \quad (5.9)$$

De la función de transferencia que se muestra en (5.9) se puede obtener el nivel del tanque 2 en función de la entrada de flujo en el tanque 1 como se muestra en (5.10).

$$H_2(K) = 0,00142221Q_i(K-1) + 0,00129229Q_i(K-2) + 1,74471H_2(K-1) - 0,750137Q_i(K-2) \quad (5.10)$$

Este sistema se simuló con un tiempo de muestreo de 50 ms, con dos entradas fusificadas como se muestra en la Fig. 5.3 y Fig. 5.4, como método de defusificación





se implementó el mínimo y el centro del área, obteniéndose una respuesta como la que se muestra en la Fig. 5.11. Observe que al sistema le costara aproximadamente 5 segundos para llevar el valor del nivel en el tanque 2 al valor de la referencia.

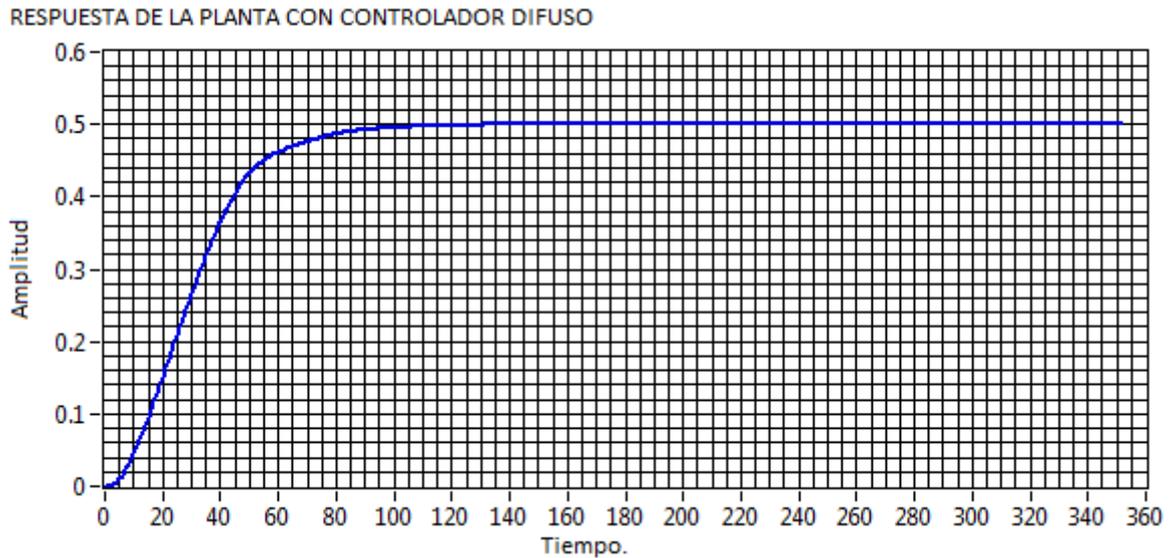


Figura 5.11: Respuesta al escalón (Minimum) FLC.

Este sistema se simuló con un tiempo de muestreo de 50 ms, con dos entradas fusificadas como se muestra en la Fig. 5.7 y la Fig. 5.8, como método de defusificación se implementó el Producto T-norma y el centro del área, obteniéndose una respuesta como la que se muestra en la Fig. 5.12. Observe que al sistema le costara aproximadamente 5 segundos para llevar el valor del nivel en el tanque 2 al valor de la referencia.

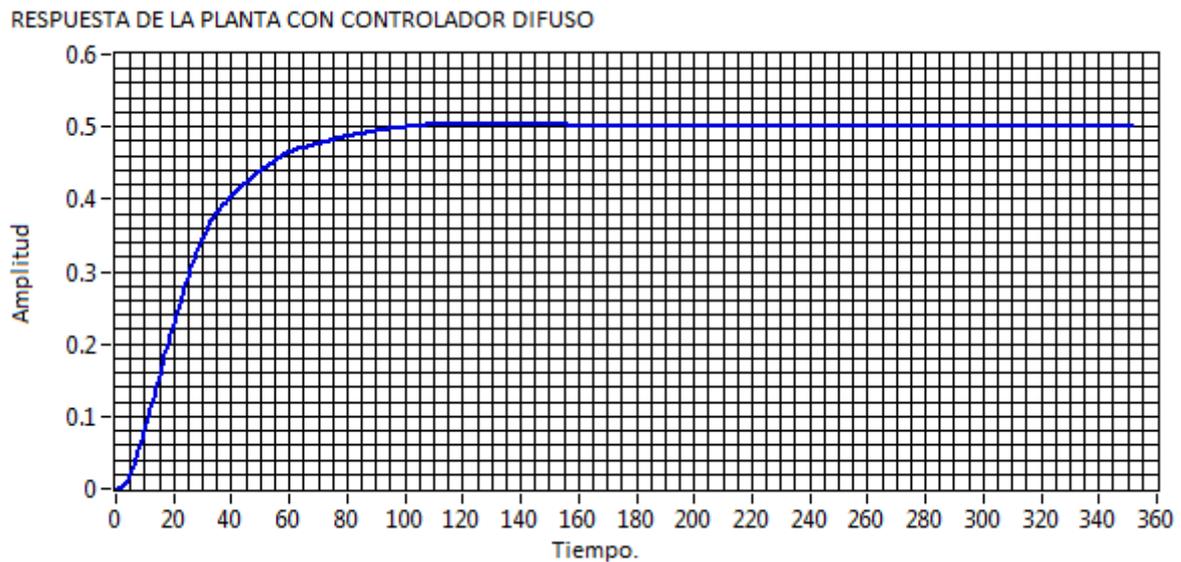


Figura 5.12: Respuesta al escalón (Product T-norm) FLC.

5.5. Resumen del capítulo.

En el presente capítulo se presentó el modelo matemático de la planta a controlar, así como el lazo de control que se implementó en este trabajo. La sección 5.1 presentó la dinámica de la planta expresada en un modelo matemático, la sección 5.2 explicó cual es el lazo de control que se implementó y mencionó de forma importante porque se implementó este lazo de control, es importante resaltar que la necesidad de implementar un controlador incremental fue porque las reglas difusas cambiaban cuando cambiaba la referencia, la sección 5.3 explica la metodología que se siguió para el diseño del controlador y finalmente la sección 5.4 presenta una simulación del lazo de control.

Capítulo 6

INTEGRAR EL CONTROL DIFUSO DEL PNS EN UN SISTEMA CLIENTE SERVIDOR DE VARIOS NODOS.

En este capítulo se integra el controlador difuso para el proceso de nivel en una arquitectura cliente servidor de varios nodos con comunicación bidireccional. en la sección 6.1 se describe algunos de los requerimientos para que el sistema de supervisión y control opere de forma eficiente; en la sección 6.2 se describe como se desarrolló el sistema y también se menciona la plataforma de programación que se eligió para este sistema; la sección 6.3 describe como está conformada la interfaz que se ejecuta en el cliente y la sección 6.4 describe como está conformada la interfaz que se ejecuta en el servidor del sistema.

6.1. Descripción del sistema de supervisión y control.

De acuerdo a la arquitectura y funcionalidad del sistema (véase Cap.4) el sistema de supervisión y control debe estar basado en una arquitectura cliente servidor con comunicación bidireccional y manejar el protocolo TCP/IP para establecer la conexión entre el servidor y los múltiples clientes que pueden conectarse al sistema. Para el monitoreo y control de las variables de proceso es necesario programar una interfaz humano-maquina(HMI) con la capacidad de monitorear y procesar los datos recibidos, almacenados y transmitidos por el proceso que se ejecuta en el cliente y soportar el almacenamiento de datos históricos para los transitorios de las variables del proceso de nivel que se muestra en la Tabla 6.1.



Tabla 6.1: Variables de proceso a supervisar.

Variables monitoreadas	
Identificación	Descripción
h_1	altura del líquido en el tanque 1
h_2	altura del líquido en el tanque 2
q_i	caudal de entrada al tanque 1
ref	altura del líquido deseada en el tanque 2

6.1.1. Requerimientos del servidor.

el servidor debe contar con un programa el cual le permita monitorear y procesar los datos recibidos, así como también le permita enviar comandos de ejecución de órdenes de control a los diferentes clientes que se encuentran conectados al sistema. La interfaz debe realizar las siguientes funciones:

- mantener una comunicación bidireccional con todos los nodos (clientes) que se encuentren conectados al sistema de supervisión y control.
- poder cambiar los parámetros de los controladores, así como la referencia del proceso que se ejecuta en los clientes
- desplegar gráficos con el histórico de los transitorios de las variables del proceso.

6.1.2. Requerimientos del cliente.

El cliente debe contar con un programa el cual le permita poder simular el proceso de nivel, y también debe contener los algoritmos de control para este; el programa debe ser capaz de enviar datos así como también ejecutar acciones de control que el servidor solicita.

la interfaz debe realizar las siguientes funciones:

- mantener una comunicación bidireccional con el servidor del sistema de supervisión y control.
- ejecutar las acciones de control que el servidor solicita.
- almacenar y enviar el histórico de los transitorios de las variables del proceso.

6.2. Desarrollo del sistema cliente servidor.

El controlador difuso para el proceso de nivel se implementó en un sistema cliente servidor, para lo cual se diseñó una base de datos para el cliente y una para el servidor, las cuales permiten que el contenido de la información que llega a través



del protocolo TCP/IP sea depositado en estas bases de datos. Una vez que se diseñó el controlador difuso véase Cap.5 se estableció el sistema de comunicación entre el servidor y múltiples clientes, esto se logró a partir de una estructura básica cliente servidor en la cual el servidor envía información a múltiples clientes y los clientes solo reciben la información que es enviada del servidor. La modificación que se realizó sobre esta estructura fue hacer que el servidor enviara y recibiera información de los clientes, así como el cliente envía y recibe información que proviene del servidor, así como seleccionar el tipo de información que se enviara o recibirá a cada cliente. dicho en otras palabras cada cliente puede estar manejando diferente información. En el diseño de la estación de supervisión y control (servidor) se tienen dos principales aplicaciones a realizar, la primera de ellas corresponde a identificar que clientes están conectados y cuantos están conectados con una señal visual de un led, correspondiendo uno a cada cliente, cuando el led se enciende significa que el cliente correspondiente esta conectado, y si el led está apagado esto significa que el cliente correspondiente está desconectado, la segunda de ellas corresponde a leer/escribir datos a los clientes conectados, eligiendo que parámetros se desea leer/escribir y el cliente con el cual se establecerá el flujo de información.

Para el monitoreo y control de la planta se requirió de una programación la cual proporcionara una interfaz gráfica y amigable al usuario, para esto se utilizó LabVIEW®, que es un programa de cómputo para realizar tareas de adquisición, análisis y presentación de datos. Para integrar el controlador difuso de la planta en una arquitectura cliente/servidor de varios nodos se diseñó una estación la cual estuviera ejecutando el proceso de nivel, y una estación la cual estuviera monitoreando y controlando el proceso que se ejecuta en el cliente. Adicionalmente se utilizaron algunas herramientas que permiten el desarrollo de la estación de monitoreo y control (servidor) y la comunicación.

- LabVIEW® PID control toolkit.- esta herramienta permite implementar algoritmos de control, combina las funciones de control clásico y lógica difusa.
- LabVIEW® Data communication toolkit.- esta herramienta permite la implementación de la comunicación con el protocolo TCP/IP entre dos pc's.

El ambiente jerárquico o la arquitectura bajo la cual se diseñó el sistema de supervisión y control se puede ver en la Fig.4.1 en la cual se puede observar que se tiene un flujo de información bidireccional, donde el servidor solicita a los clientes la información que tienen ejecutando y/o la ejecución de acciones en el proceso que se está ejecutando en cada cliente.

6.3. Interfaz del proceso de nivel.

Se desarrolló una interfaz de la simulación del proceso de nivel de dos tanques interactuantes, usando el software de programación gráfica LabVIEW®, para lo cual se tomó la función de transferencia en el dominio de S de la planta como se plantea en el Cap.5 y en el Aexo A. y se transformó a su equivalente en el dominio Z





con la finalidad de poder conocer el modelo discreto de la planta y programar en LabVIEW® dicha función de transferencia. El objetivo de esta interfaz es poder simular en el cliente el proceso de nivel y poder ver de una forma gráfica lo que sucede con la información que el servidor envía o recibe del cliente.

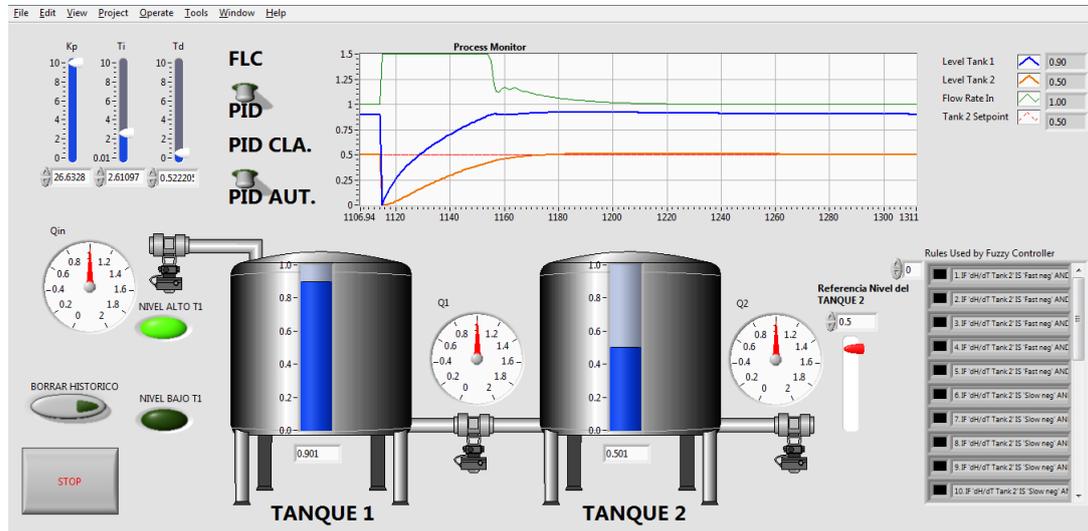


Figura 6.1: Interfaz proceso de nivel cliente.

En la Fig. 6.1 se muestra la interfaz que se ejecuta en el cliente simulando el proceso de nivel de tanques interactuantes.

Gráfica de parámetros en tiempo real.

La gráfica de parámetros en tiempo real es la pantalla en la cual el operador puede ver el estado del proceso de nivel en tiempo real, en ella se puede visualizar el estado de las siguientes variables (h_1, h_2, q_i, ref). para que el operador tenga acceso a este tipo de información, la gráfica de parámetros se incluye en la interfaz que se ejecuta en el cliente, en la Fig. 6.2 se muestra esta gráfica.

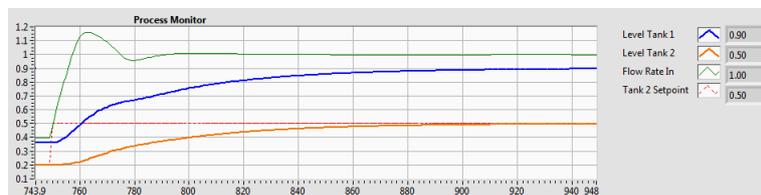


Figura 6.2: Interfaz proceso de nivel cliente "gráfica de parámetros en tiempo real".

Ajuste de parámetros del controlador PID.

Se incluye en la interfaz del cliente tres controles que permiten ajustar los parámetros del controlador PID en tiempo real; los parámetros de ajuste son: T_i



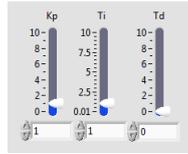


Figura 6.3: Interfaz proceso de nivel cliente "control de parámetros del controlador PID".

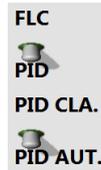


Figura 6.4: Interfaz proceso de nivel cliente "selector de tipo de controlador".

tiempo integral, T_d tiempo derivativo, K_p ganancia proporcional; en la Fig. 6.3 se muestra los controles de ajuste para los parámetros del controlador PID.

Selector del tipo de controlador.

Se incluye un selector del tipo de controlador el cual permite seleccionar el tipo de controlador que estará controlando la planta, una vez que se selecciona el tipo de controlador, en caso de seleccionar el controlador PID se tiene la opción de tener el controlador PID clásico o PID autoajutable; el controlador se puede cambiar en tiempo de ejecución, este selector se muestra en la Fig. 6.4 y se incluye en la interfaz del cliente Fig.6.1.

Referencia nivel del líquido en el tanque 2.

Se incluye un control, el cual le permite al usuario poder cambiar el valor de la referencia en tiempo de ejecución. este control se incluye en la interfaz del cliente Fig.6.1 y se muestra en la Fig.6.5.

6.4. Interfaz de supervisión y control.

Se desarrolló una interfaz en el servidor la cual permite al usuario supervisar y controlar algunos parámetros del proceso de nivel que se ejecuta en el cliente, así como también permite ver que clientes son los que se encuentran conectados,



Figura 6.5: Interfaz proceso de nivel cliente "control de la referencia".



ya que se hace un monitoreo constante de que clientes tienen una comunicación abierta con el servidor, de esta forma si falla la conexión con alguno de los clientes el servidor se percata de que cliente fue el que se desconectó. También el servidor permite configurar si se desea enviar o recibir datos así como de que cliente se desea que sea el flujo de información. de esta forma se puede leer que valor de referencia tiene el cliente a petición del servidor así como cambiar este valor de referencia por algún valor que el servidor proponga. Es así como se supervisa y se controla el valor de la variable de referencia.

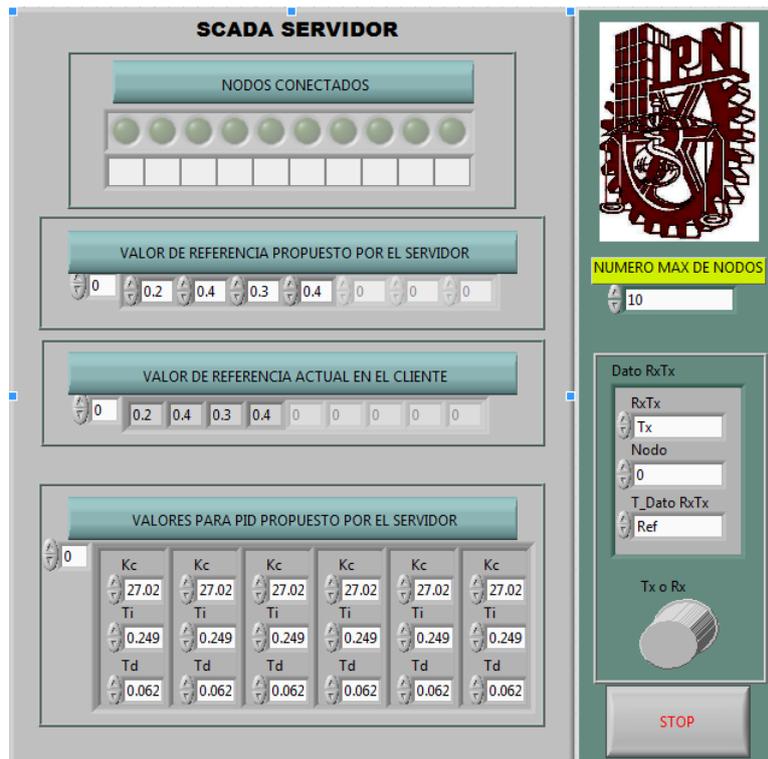


Figura 6.6: Interfaz de supervisión y control servidor.

La Fig. 6.6 muestra la interfaz que se ejecuta en el servidor y permite supervisar y controlar algunos parámetros del proceso de nivel que se ejecuta en el cliente, se puede configurar el número de clientes que el servidor aceptara que se conecten, también se puede observar el número de clientes conectados y desconectados.

Nodos conectados.

Se incluye en el servidor un arreglo el cual permite que se supervise en forma constante la cantidad de clientes que están conectados al sistema, si el cliente se encuentra conectado se enciende el led correspondiente al número de cliente y si no se encuentra conectado el led se apaga. este arreglo se incluye en la interfaz del servidor Fig. 6.6 y se muestra en la fig.6.7.



Figura 6.7: Interfaz de supervisión y control servidor "Nodos conectados".

Valor de la referencia propuesto por el servidor.

Es un arreglo el cual se muestra en la interfaz del servidor Fig.6.6, y permite guardar en este arreglo el valor de la referencia para cada uno de los clientes que se conectan al sistema; este valor de referencia es un valor que el servidor propone para cada uno de los clientes, y no necesariamente es el valor de referencia que se ejecuta en el cliente, este arreglo se muestra en la Fig.6.8.

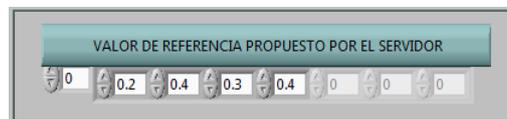


Figura 6.8: Interfaz de supervisión y control servidor "valor de referencia propuesto".

Valor de referencia actual en el cliente.

Este arreglo permite visualizar en la interfaz del servidor el valor actual de la referencia en cada uno de los clientes, este valor no es necesariamente el mismo que el valor propuesto, ya que el valor puede ser cambiado directamente por el cliente. de tal forma que este arreglo es únicamente el monitoreo de la variable de referencia en cada uno de los clientes conectados, este arreglo se muestra en la Fig. 6.9 y se incluye en la interfaz del servidor Fig. 6.6.

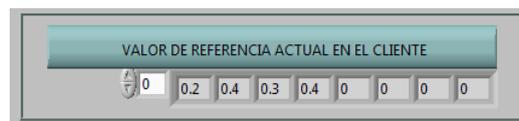


Figura 6.9: Interfaz de supervisión y control servidor "valor de referencia actual en el cliente".

Recepción (Rx) y Transmisión (Tx).

En la interfaz de servidor Fig. 6.6 se incluye una sección en la cual se puede configurar el número máximo de clientes que se conectarán al sistema; un arreglo el cual permite configurar una transmisión o recepción de información, el nodo al que se enviara o recibirá la información, así como el tipo de información que se enviara



o recibirá.

los campos que integran esta sección son:

- RxTx: este campo permite configurar la acción que se ejecutara, si se selecciona Rx la acción que se ejecutara es una recepción de información, si se selecciona Tx la acción que se ejecutara sera una transmisión de información.
- Nodo: en este campo se selecciona el cliente al cual se enviara o recibirá información.
- T Dato RxTx: este campo permite seleccionar el tipo de dato que se requiere enviar o recibir, y este dato puede ser referencia, parámetros del PID, Histórico de los transitorios de las variables de proceso.

Para que la configuración de recepción o de transmisión se ejecute es necesario presionar el botón Tx o Rx. en la Fig. 6.10 se muestra esta sección.

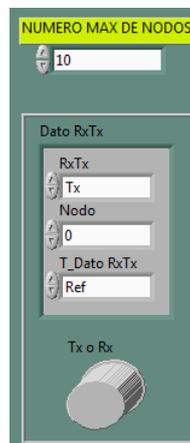


Figura 6.10: Interfaz de supervisión y control servidor "recepción Rx y transmisión Tx".

6.5. Resumen del capítulo.

En este capítulo se presentó los requerimientos o características de funcionamiento que debe cubrir el cliente y el servidor, así como las funciones que tiene la interfaz hombre maquina en el cliente y servidor. La sección 6.1 se proporcionó los requerimientos del cliente y el servidor, con la finalidad de cubrir de la mejor manera la funcionalidad del sistema, la sección 6.2 presentó el desarrollo del sistema y las herramientas que se ocuparon para esto, la sección 6.3 y 6.4 explica cómo está conformada la interfaz hombre máquina y cuáles son las funciones de esta.

Capítulo 7

CONTROLADOR PID AUTOAJUSTABLE PARA UN PROCESO SIMULADO (PS).

En este capítulo se explica el desarrollo de un controlador con lógica difusa el cual ajusta en forma automática las ganancias de un controlador PID clásico con la finalidad de optimizar la respuesta de dicho controlador. en la sección 7.1 se presenta una explicación de la metodología de diseño de un controlador PID autoajustable, en la sección 7.2 se presenta el código en MATLAB[®] para el método gráfico de Ziegler and Nichols, la sección 7.3 explica el esquema de control que se desarrolló, en la sección 7.4 se desarrolla el controlador y en la sección 7.5 se presentan los resultados de la simulación.

7.1. Controlador difuso como sintonizador de controlador PID.

Como es conocido un controlador difuso tipo Mamdani tiene la ventaja de que se puede desarrollar a partir del conocimiento de un experto, esto quiere decir que para plantear la base de conocimiento de este controlador se parte de la experiencia con el proceso a controlar (véase capítulo 3). es posible que la experiencia del experto sea tomado de una ley de control tan popular como el control PID; este tipo de controlador se usa para llevar la salida de una planta al valor de referencia, para lograr esto es necesario tener una retroalimentación la cual proporcione información de la diferencia entre la referencia y el valor de salida de la planta (error). las características de la respuesta de la planta dependen fundamentalmente del ajuste de los parámetros del controlador[47], dichos parámetros son tres y son conocidos como la ganancia proporcional, la ganancia integral y la ganancia derivativa. el ajuste de estos parámetros afecta directamente a los términos proporcional, integral y derivativo respectivamente. El termino esencial es el termino proporcional, en el cual se multiplica la ganancia proporcional por el error, de tal forma que si el error



decrece el termino proporcional lo hace también, en ocasiones es imposible que el sistema llegue al valor deseado con solo el termino proporcional, es entonces que el termino integral toma importancia, ya que esta acción integra el error durante cierto tiempo y la suma a la acción proporcional, de este modo el error en estado estable se elimina; por último la acción derivativa es alimentado por la derivada o cambio en el error, de este modo el control puede predecir la tendencia del sistema para anticiparse y eliminar el sobretiro del sistema. los tres términos dependen de las ganancias para poder actuar en forma apropiada [30][31][48][32]. El método de Ziegler and Nichols propuesto en 1942 [49] sigue siendo hasta estos días un método muy efectivo para calcular las ganancias apropiadas para un controlador PID [50], de esta forma el controlador difuso toma como punto de partida las ganancias que arrojan dicho método para poder establecer los rangos de las variables lingüísticas. es así como un controlador difuso puede ajustar en forma automática las ganancias del controlador para mejorar las características de la respuesta de la planta [51].

7.2. Sintonización del controlador PID por el método de Ziegler and Nichols.

Primero se realiza un ensayo introduciendo una señal escalón en la planta en lazo abierto, dicha respuesta muestra una forma de s, a esta respuesta se calcula la pendiente o derivada en el punto de inflexión; de esta forma es posible trazar una recta tangente a la curva en dicho punto; esta recta es proyectada sobre el eje x y también se proyecta una recta paralela al eje y, que corta el eje x e intercepta a la recta tangente en el punto máximo de la respuesta de la planta; así es posible conocer los parámetros T y L de Ziegler and Nichols.

En este trabajo se programó en MATLAB[®] el método gráfico de Ziegler and Nichols (ver Cap.3) para el proceso de nivel simulado que se plantea en el capítulo 5 cuya función de transferencia se muestra en (5.7), a continuación se muestra los pasos a seguir para desarrollar el método en forma adecuada.

1. INICIO
2. Obtener la respuesta de la planta a una entrada escalón unitario.
3. Dibujar una recta tangente a la curva en el punto de inflexión.
4. Determinar el punto de intersección con el eje del tiempo y también la intersección con la una recta paralela al eje de tiempo y que corta al eje y en el punto máximo de la respuesta de la planta a una entrada escalón unitario.
5. Calcular los coeficientes de Ziegler and Nichols L y T.
6. Calcular las ganancias del controlador propuestas por Ziegler and Nichols como se muestra en la Tabla 3.1.





7. FIN

A continuación se muestra el código del programa en MATLAB®.

```

1  %Método de Ziegler and Nichols
2
3  clear all;
4  % 1.-INICIO
5  C1=2;
6  C2=1;
7  R1=0.4;
8  R2=0.5;
9  s=tf('s');
10 tanques=R2/((C1*R1*C2*R2)*s^2+(C1*R1+R2*C1+R2*C2)*s+1)
11 pp=pole(tanques); %polos del sistema
12 dt=0.05;
13 t=0:dt:28;
14 % 2.- Obtener la respuesta de la planta a una entrada escalón
    unitario.
15 y=step(tanques,t); %respuesta al escalón unitario
16 dy=diff(y)/dt; %derivada
17 [m p]=max(dy) %punto de inflexión
18 % 3.- Dibujar una recta tangente a la curva en el punto de
    inflexión.
19 for i=1:length(t)
20     yt(i)=m*(t(i)-t(p))+y(p);
21     if yt(i)>max(y)
22         yt(i)=max(y);
23     end
24 end
25 % 4.- Determinar el punto de intersección con el eje del tiempo
    y
26 % también la intersección con la una recta paralela al eje de
    tiempo
27 % y que corta al eje y en el punto máximo de la respuesta de la
28 % planta a una entrada escalón unitario.
29 xc=(-1*(y(p)/m))+t(p); %se calcula el corte con x cuando y=0
30 xc2=((max(y)-y(p))/m)+t(p); %se calcula el corte cuando la recta
    alcanza y max
31 yc=m*(xc-t(p))+y(p);
32 yc2=m*(xc2-t(p))+y(p);
33 % 5.- Calcular los coeficientes de Ziegler and Nichols L y T.
34 L=xc; %coeficientes de Zeigler
35 T=xc2; %coeficiente de Ziegler
36
37 % 6.- Calcular las ganancias del controlador propuestas por
    Ziegler and Nichols
38 Kp=1.2*(T/L)

```





```

39Ti=2*L
40Td=.5*L
41% 7.-FIN
42plot(t,y)
43hold on
44plot(t,yt,'k')
45hold on
46plot(xc,yc,'rx')
47hold on
48plot(xc2,yc2,'rx')
49hold off
50grid on
51title('Respuesta al escalón')
52xlabel('tiempo (ciclos x 50ms)')
53ylabel('Nivel de líquido en tanque 2')

```

La gráfica del programa se muestra en la Fig.7.1 , en donde se puede observar la respuesta del sistema a un escalón unitario, así como la recta tangente en el punto de inflexión y una recta paralela al eje x que toca a la respuesta del sistema en el punto máximo e intersecta a la recta tangente. Los parámetros T y L de Ziegler and Nichols se obtienen de medir el corte con el eje x así como el punto proyectado sobre este mismo eje en donde se intersecta la recta con la tangente.

Los resultados de la simulación dan valores para las constantes de tiempo de retardo

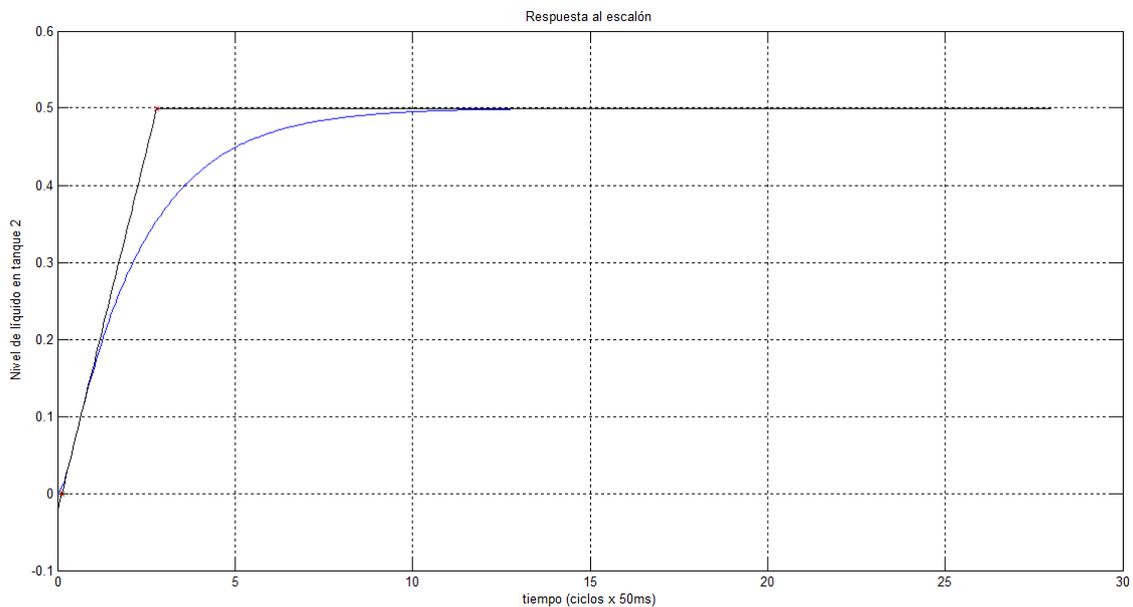


Figura 7.1: Método gráfico Ziegler and Nichols

L y constante de tiempo T aplicando el método de Ziegler and Nichols dichos valores son $L = 0,1245$ y $T = 2,8039$ respectivamente. tomando como base las reglas de





Ziegler and Nichols (véase Tabla 3.1 en el Anexo A) se tienen los valores propuestos para la sintonía de controlador PID, estos valores se muestran en la Tabla 7.1.

Tabla 7.1: Reglas de sintonización de Ziegler and Nichols basadas en la respuesta al escalón unitario de la planta.

Tipo de controlador	Kp	Ti	Td
P	22.5191	∞	0
PI	20.2672	0.0415	0
PID	27.0230	0.2490	0.0623

7.3. Esquema de control

se desarrolló un esquema de control en el cual un controlador PID es el encargado de regular la entrada en el tanque 1 para poder llevar el nivel del tanque 2 al valor de la referencia y con esto poder establecer el error en un valor de cero. este controlador se sintonizó usando el método de Ziegler and Nichols (véase Anexo A), con el objetivo de mejorar la respuesta de dicho controlador se ajusta en forma dinámica el valor de las ganancias del controlador PID usando para esto un controlador difuso. este controlador toma como punto de partida las ganancias del método de Ziegler and Nichols y a partir de estas ganancias con base a la respuesta del sistema se ajustan estas ganancias.

En la Fig. 7.2 se puede observar el esquema de control que se usó para el ajuste de las ganancias del controlador PID, es importante mencionar que en este esquema se dibuja tres controladores difusos cada uno de los cuales está encargado de regular el valor de ganancia asociado para cada uno de los parámetros del controlador PID, se debe entender que en este esquema cada uno de los controladores difusos no es precisamente un controlador, es la base de conocimientos asociada al parámetro de ajuste del PID ya que se desarrolló un controlador con dos entradas y tres salidas.

7.4. Desarrollo del controlador

El controlador difuso que ajusta las ganancias del controlador PID tiene como entradas el error y la velocidad de cambio del nivel de líquido en el tanque 2. el rango de estas dos variables se estableció mediante simulaciones del comportamiento de la planta con el controlador PID que tiene como ganancias los valores obtenidos aplicando el método gráfico de Ziegler and Nichols. dichas variables de entrada se dividieron en 7 conjuntos difusos. la Fig. 7.3 muestra el rango de valores válidos para el error, así como los conjuntos difusos que pertenecen a esta variable. la Fig. 7.4 muestra el rango de valores válidos para la velocidad de cambio en el nivel del líquido en el tanque 2, así como los conjuntos difusos que pertenecen a esta variable.

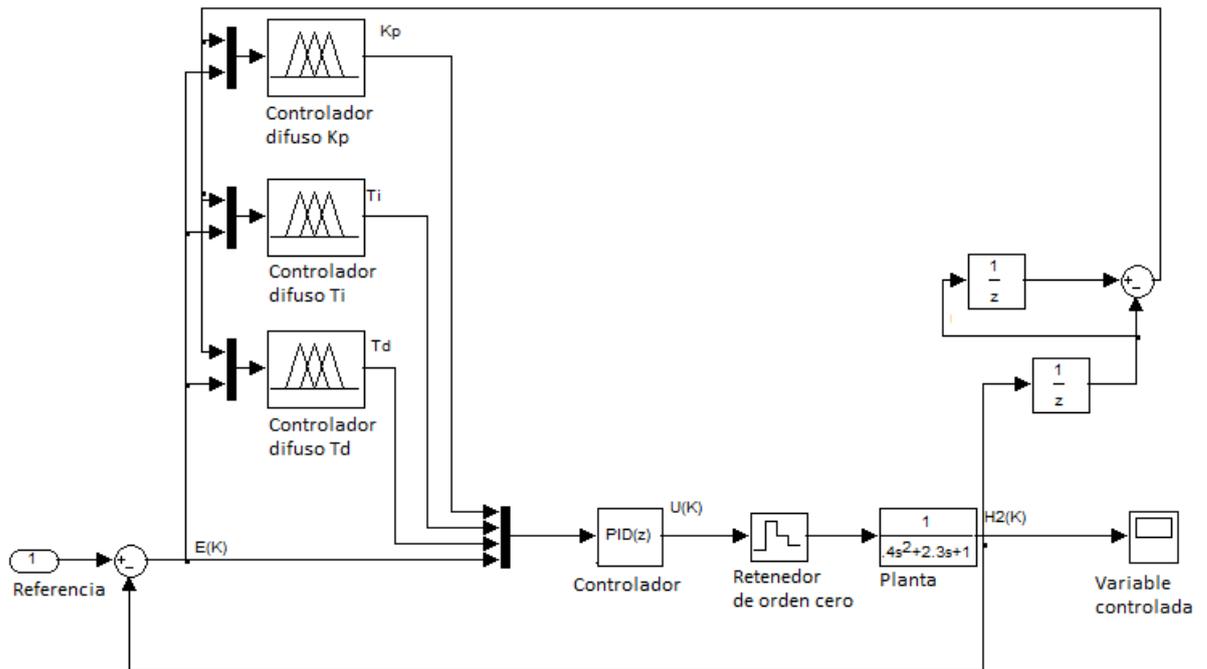


Figura 7.2: Esquema de control para el ajuste de ganancias del PID.

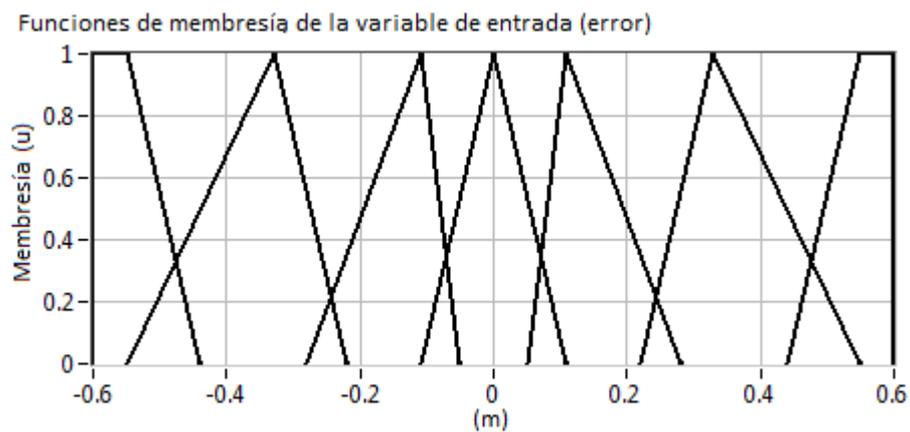


Figura 7.3: Error (diferencia entre la referencia y el valor real del nivel de líquido en tanque 2)

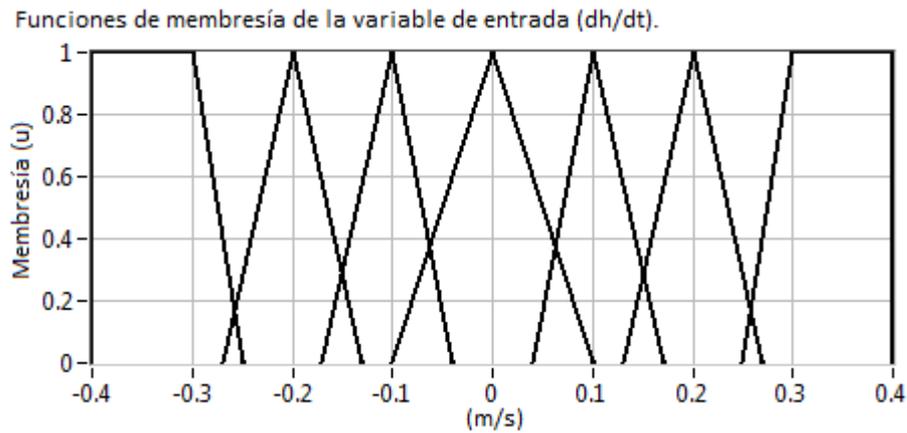


Figura 7.4: Velocidad de cambio del nivel de líquido en tanque 2.

El controlador tiene tres variables de salida correspondientes a K_p , T_i y T_d respectivamente como ya se ha mencionado el rango de los valores que estas variables ocupan se calcula a partir de las ganancias que arroja el método de Ziegler and Nichols. una vez que se tienen estos valores y mediante el uso de LabVIEW® y PIDtool de MATLAB® se ajustan y se proponen valores máximo y mínimo para cada una de las ganancias del controlador; de tal forma que el controlador estará ajustando en forma dinámica las ganancias del controlador PID alrededor de las ganancias de Ziegler and Nichols.

A continuación se mostrara el rango de valores en el cual se estará moviendo los valores de las ganancias para el controlador PID, dichos valores son la salida del controlador difuso.

La Fig. 7.5 muestra el rango de valores alrededor del cual el controlador estará ajustando los valores para la ganancia proporcional del controlador PID.

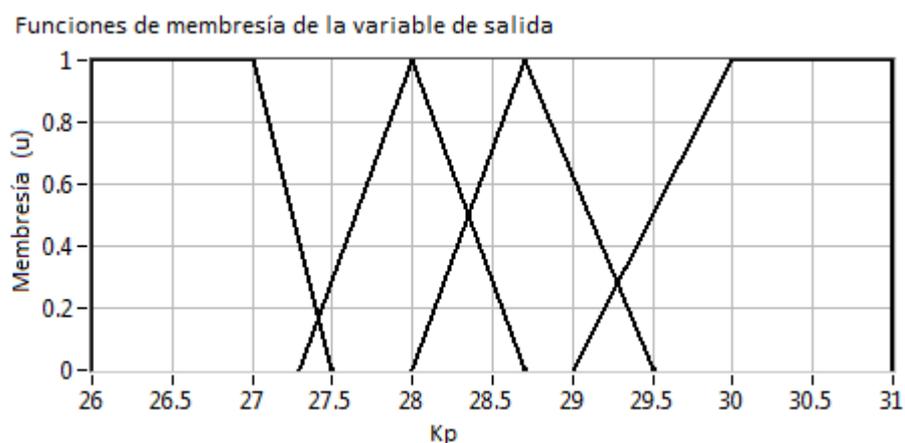


Figura 7.5: Ganancia proporcional.



La Fig. 7.6 muestra el rango de valores alrededor del cual el controlador estará ajustando los valores para la ganancia integral del controlador PID.

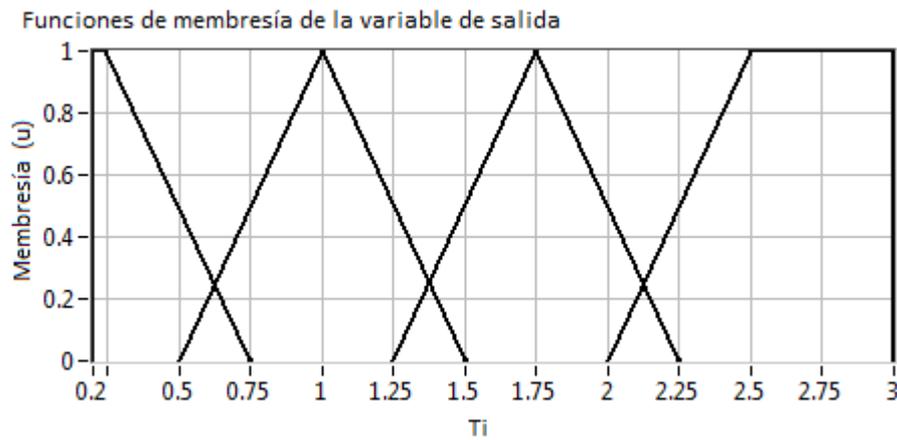


Figura 7.6: Tiempo integral.

La Fig. 7.7 muestra el rango de valores alrededor del cual el controlador estará ajustando los valores para la ganancia derivativa del controlador PID.

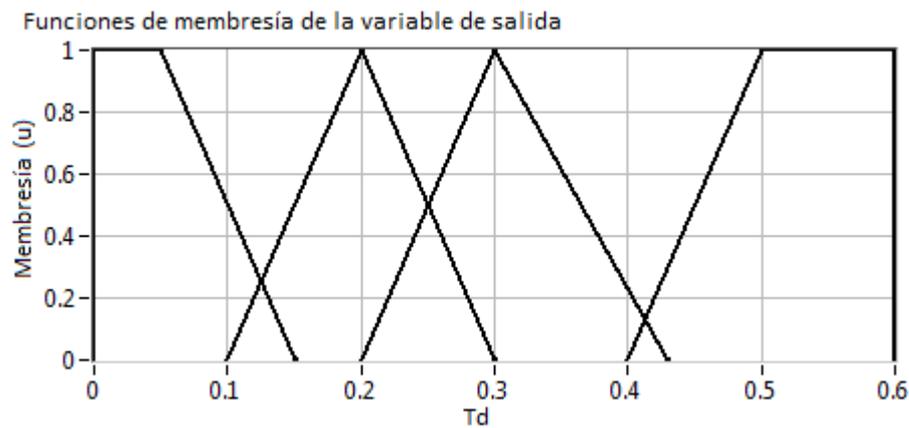


Figura 7.7: Tiempo derivativo.

Para establecer el motor de inferencias del controlador se tomo como base de conocimiento el siguiente razonamiento:

- si el error se incrementa; el valor de K_p se incrementa y K_i disminuye, esto último con el fin de evitar una acumulación de la acción integral que lleve a un sobreimpulso.
- si el error se decrementa, el valor de K_p se decrementa y K_i aumenta, esto último, con el fin de que la acción integral ayude a corregir el error a medida que el tiempo transcurre.



- si la velocidad de cambio del nivel de líquido en el tanque 2 incrementa; el valor de K_d decremента.
- si la velocidad de cambio del nivel del nivel de líquido en el tanque 2 decremента; el valor de K_d incrementa con la finalidad de restarle oscilaciones al sistema.

En la Tabla 7.2 se puede observar el motor de inferencias del controlador difuso. en las filas y columnas se muestra la parte antecedente y en el interior de la matriz se colocan tres valores lingüísticos como la parte consecuente en cada celda, correspondientes a el valor de K_p, T_i y T_d respectivamente.

Tabla 7.2: Motor de inferencias del controlador (FLC) para el autoajuste del PID.

e dh/dt	NL	NM	NS	Z	PS	PM	PL
NL	PL,PL,Z	PM,PM,Z	PS,PS,Z	Z,Z,Z	PS,PS,Z	PM,PM,Z	PL,PL,Z
NM	PL,PL,PS	PM,PM,PS	PS,PS,PS	Z,Z,PS	PS,PS,PS	PM,PM,PS	PL,PL,PS
NS	PL,PL,PM	PM,PM,PM	PS,PS,PM	Z,Z,PM	PS,PS,PM	PM,PM,PM	PL,PL,PM
Z	PL,PL,PL	PM,PM,PL	PS,PS,PL	Z,Z,PL	PS,PS,PL	PM,PM,PL	PL,PL,PL
PS	PL,PL,PM	PM,PM,PM	PS,PS,PM	Z,Z,PM	PS,PS,PM	PM,PM,PM	PL,PL,PM
PM	PL,PL,PS	PM,PM,PS	PS,PS,PS	Z,Z,PS	PS,PS,PS	PM,PM,PS	PL,PL,PS
PL	PL,PL,Z	PM,PM,Z	PS,PS,Z	Z,Z,Z	PS,PS,Z	PM,PM,Z	PL,PL,Z

En el Anexo A se muestra las superficies de control que el motor de inferencias genera sobre las salidas del controlador (K_p, T_i, T_d); así como las reglas de inferencia de este controlador.

7.5. Simulación PID autoajustable

A continuación se mostrara el rango de valores en el cual se estará moviendo los valores de las Se simuló el comportamiento de la planta con el controlador PID autoajustable con lógica difusa siguiendo el mismo procedimiento que en el Cap.5. los resultados de esta simulación se muestran en la Fig. 7.8 en donde el sistema se simuló con un tiempo de muestreo de 50 ms, con dos entradas fusificadas como se muestra en las Fig. 7.3 y 7.4 como método de defusificación se implementó el producto y el centro del área.

7.6. Resumen del capítulo.

En este capítulo se presentó el diseño de un controlador PID autoajustable con lógica difusa, para lo cual fue necesario encontrar las ganancias del controlador aplicando el método gráfico de Ziegler and Nichols, una vez que se encontraron las ganancias del controlador a partir de estas se propone el rango de las variables del





RESPUESTA DE LA PLANTA CON CONTROLADOR PID AUTOAJUSTABLE

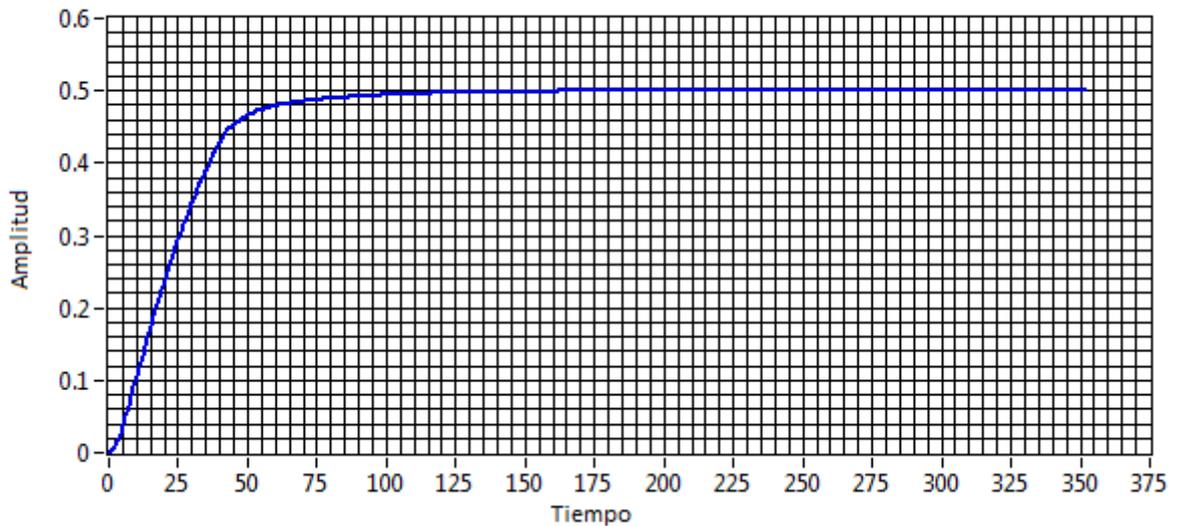


Figura 7.8: Respuesta al escalón PID autoajustable

controlador difuso. En la sección 7.1 se explicó la filosofía bajo la cual se implementó el controlador difuso como sintonizador de un controlador PID, la sección 7.2 mostró como se implementó el método de Ziegler and Nichols en MATLAB®, la sección 7.3 explicó el lazo de control que se programó, la sección 7.4 explicó la metodología para la obtención del controlador PID autoajustable con lógica difusa y por último la sección 7.5 mostró una simulación del controlador con la planta.

Capítulo 8

EVALUACIÓN DE RESULTADOS.

En este capítulo se presentan las pruebas realizadas y los resultados obtenidos del desarrollo de cada uno de los controladores y su rendimiento con la planta.

8.1. Diagrama básico de control

Partiendo de la función de transferencia de la planta expresada como la transformada de Laplace de la salida del sistema con respecto a la transformada de Laplace de la entrada del sistema, se propone una estructura en la cual sea posible poder programar y simular el comportamiento de la planta con un controlador digital, para lo cual es necesario transformar la planta del dominio de Laplace al dominio de Z y agregar un retenedor de orden cero como se plantea en la Fig. 8.1.

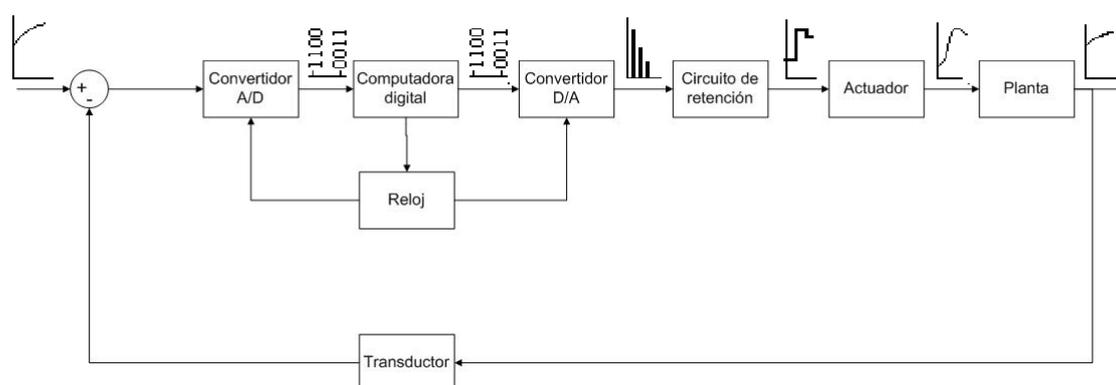


Figura 8.1: Diagrama de bloques de un sistema de control [5].



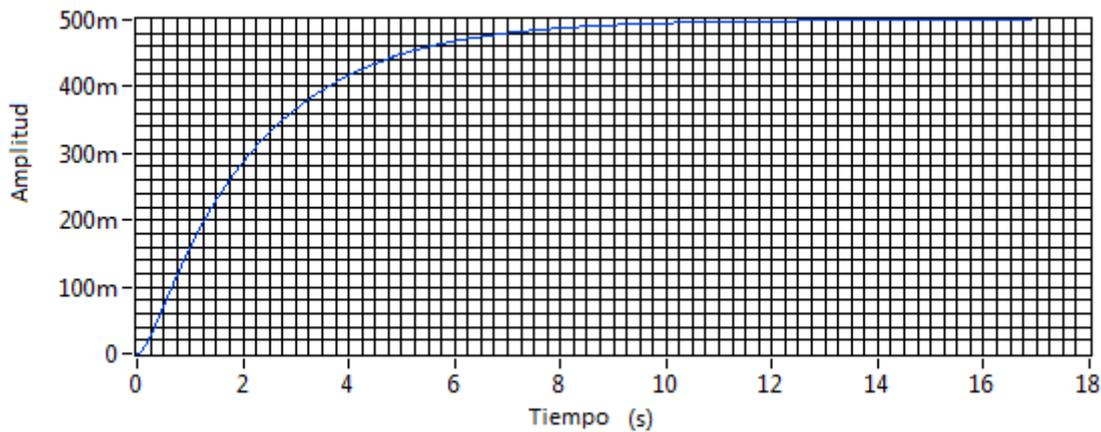
8.2. Pruebas realizadas

En esta sección se explica cada una de las pruebas aplicadas en este trabajo.

8.2.1. Prueba de lazo abierto

Con base en [5] se decidió programar en labVIEW® una función de transferencia pulso con un retenedor de orden cero para la planta y con esto poder simular el rendimiento de la planta con cada uno de los algoritmos de control propuestos en este trabajo, y como primera prueba se verifica que la respuesta en lazo abierto a una entrada escalón de la función de transferencia en el dominio de S es equivalente a la respuesta de pulsos de la función de transferencia equivalente en Z con un retenedor de orden cero. los resultados de la simulación se muestran en la Fig. 8.2. donde se puede observar que las funciones de transferencia son equivalentes.

RESPUESTA AL ESCALON EN TIEMPO CONTINUO



RESPUESTA AL ESCALON TIEMPO DISCRETO

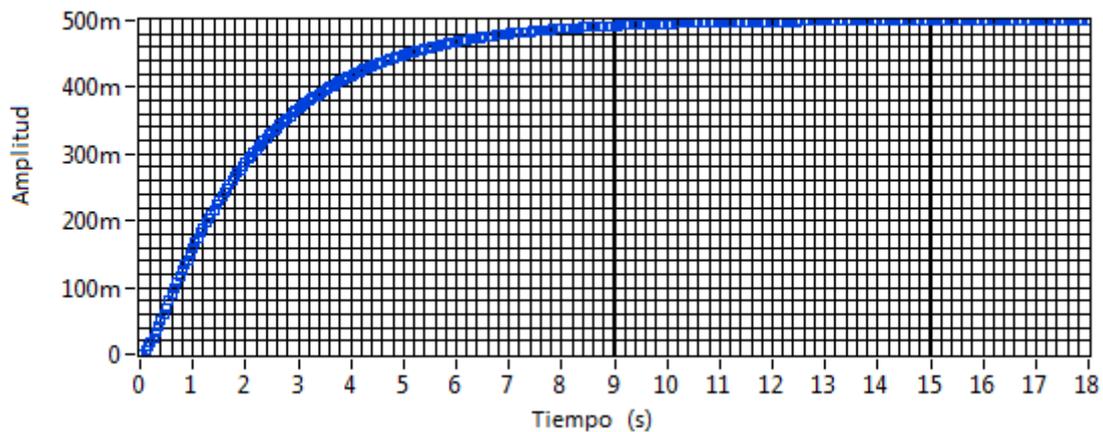


Figura 8.2: Resultados de simulación en lazo abierto del modelo en S y en Z respectivamente.



8.2.2. Prueba del lazo cerrado de control.

En esta prueba se simuló la respuesta del sistema en lazo cerrado con un controlador PID, para lo cual se consigue la función de transferencia en el dominio de Laplace y posteriormente se consigue la transformada Z de esta función con un retenedor de orden cero y se comprueba que la respuesta del sistema es equivalente, como se muestra en la Fig. 8.3.

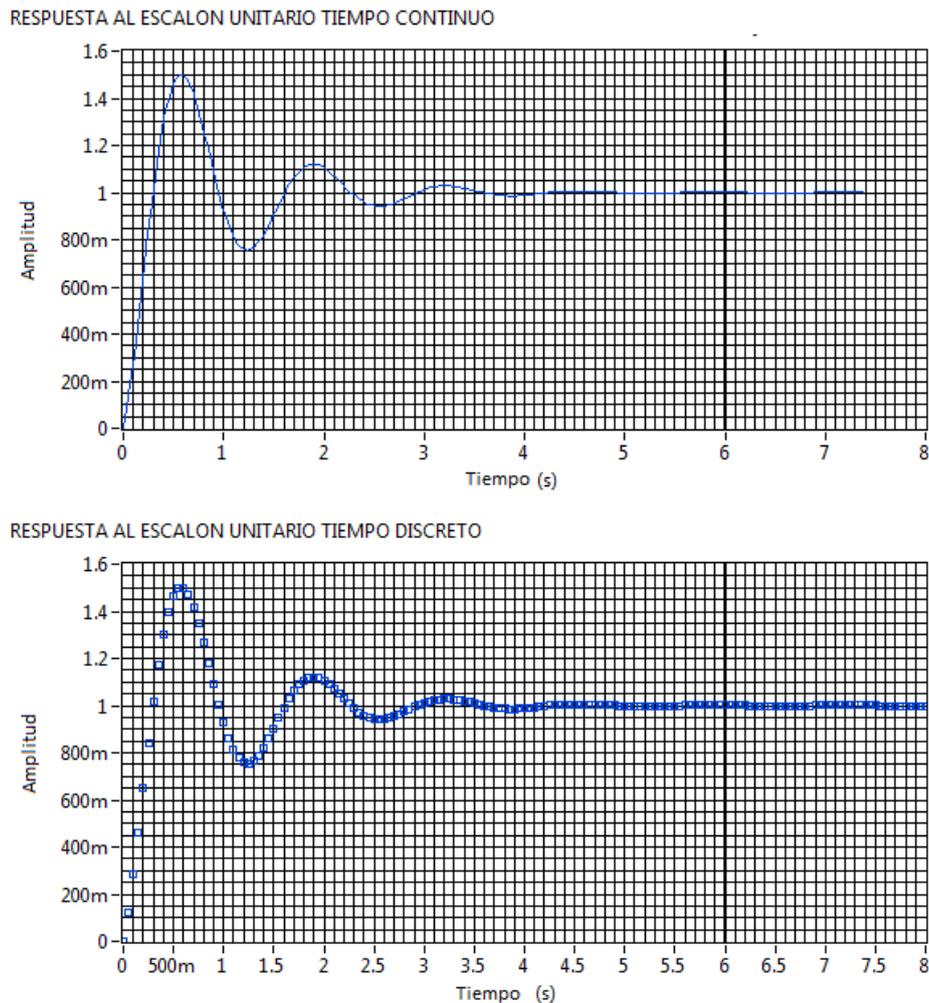


Figura 8.3: Resultados de simulación en lazo cerrado de control en S y en Z respectivamente.

8.2.3. Prueba de los diferentes algoritmos de control.

Se realizaron 20 simulaciones del lazo de control con los diferentes algoritmos (controlador difuso, controlador PID clásico y controlador PID autoajustable con lógica difusa) sobre la planta simulada, y se eligió los resultados que más veces se repitieron, estos resultados se muestran en la Tabla A.2 ubicada en el Anexo





A, en esta tabla se plasmaron los resultados a una entrada escalón unitario con una magnitud de 0.5. de tal manera que se muestra el nivel que alcanza el líquido contenido en el tanque 2 en el tiempo t y con los diferentes controladores.

Los resultados que se mostraron en la Tabla A.2 se muestran en forma gráfica en la Fig. 8.4 ; en la cual se muestran los resultados comparando los diferentes controladores. con color azul se muestra la gráfica con los resultados de la planta con el controlador difuso tipo Mamdani, con color rojo se muestra los resultados de la planta con el controlador PID sintonizado por el método de Ziegler and Nichols y con color verde se muestra los resultados de la planta con el controlador PID autoajustable con lógica difusa.

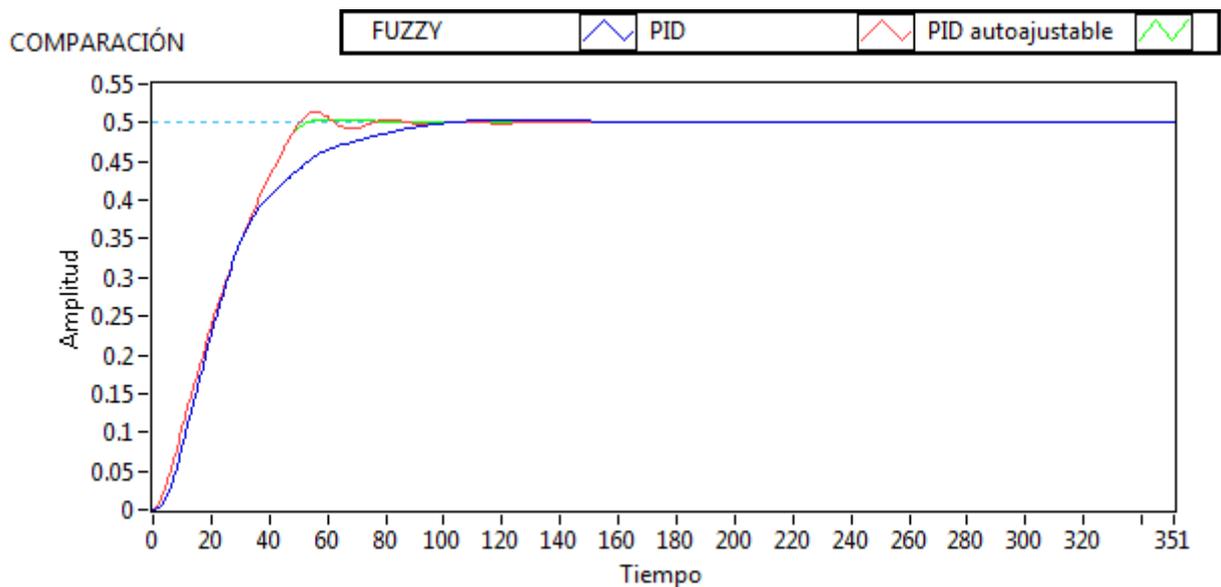


Figura 8.4: Resultados de la simulación.

Tabla 8.1: Resultados

CONTROLADOR	S.M %	T.R	T.L	T.A
FUZZY	0.83 %	1100ms	2300ms	4250ms
PID	2.80 %	1050ms	1850ms	2950ms
PID autoajustable	0.89 %	1050ms	1850ms	2550ms

en la Tabla 8.1 se muestra el sobrepaso máximo (S.M), el tiempo de retardo (T.R), el tiempo de levantamiento (T.L) y el tiempo de asentamiento (T.A) para cada uno de los controladores. donde se puede observar que el controlador PID autoajustable es el que mejor rendimiento tiene para el sistema de los tanques interactuantes ya que llega a la región de estabilidad en un tiempo menor que el controlador difuso y el controlador PID clásico. también muestra un sobreimpulso



muy pequeño en comparación con el controlador PID clásico, aunque en comparación con el controlador difuso el sobreimpulso es mayor.

8.3. Comparación con el Estado del Arte.

Se realizó una comparación con el estado del arte, y se consideraron trabajos los cuales implementaron algoritmos similares a los de este trabajo de tesis, que son un controlador PID clásico, un controlador FLC y un controlador PID autoajustable con lógica difusa. Es importante mencionar que la planta a controlar en ningún caso es la misma que en este trabajo de tesis ya que en [13][14][47] y [51] a pesar de tratarse del control de nivel en un sistema de tanques no se trata de la misma planta que en este trabajo de tesis ya que cambian los parámetros y la dinámica de las plantas a controlar, y en el caso de [16] y [25] se trata de un motor DC y un motor de inducción respectivamente.

En [13] se controló una planta no lineal con un controlador PID convencional y un controlador difuso (FLC). La planta se linealizó usando una herramienta de linealización de MATLAB[®], una vez que se obtuvo el modelo lineal de la planta se propuso un controlador PID convencional. También se diseñó un controlador difuso para la planta no lineal, y se demostró que el controlador difuso tiene un mejor rendimiento que un controlador PID clásico siempre y cuando la planta sea una planta no lineal. Sin embargo a pesar de que el controlador difuso tiene un mejor comportamiento, no se logró eliminar el error en estado estacionario. En este trabajo de tesis se elimina el error en estado estacionario porque a diferencia de [13] se programó un controlador difuso incremental con la finalidad de ajustar a la referencia y eliminar el error en estado estacionario. En [14] se presentó un controlador PID clásico y un controlador PID difuso para controlar una planta de dos tanques interactuantes cuyo modelo matemático es lineal, se presentó una comparación entre los dos controladores y se observa que el controlador PID difuso tiene una mejor respuesta, el controlador PID difuso es un algoritmo el cual encuentra las ganancias del controlador PID en forma dinámica al igual que en el trabajo de esta tesis, y los resultados de la comparación son parecidos a los de esta tesis ya que se logra mejorar la respuesta del sistema con el controlador PID difuso, como se muestra en la Fig.8.5

En [47] se hace una comparación entre un controlador PID clásico y un controlador PID difuso para controlar el nivel de un tanque, se observa que no se especifica el tipo de tanque y la función de transferencia se obtiene con MATLAB[®], pero los algoritmos son parecidos a los presentados en este trabajo de tesis, en la Fig.8.6 se presentan los resultados obtenidos en [47] y como se puede observar se mejora la respuesta de la planta con un controlador PID difuso, estos resultados son similares a los obtenidos en este trabajo de tesis ya que se logró mejorar la respuesta del sistema con un controlador PID difuso.

En [51] se implementó un controlador PID clásico sintonizado con el método de Ziegler and Nichols y un controlador PID autoajustable con lógica difusa en el cual se encontraron las ganancias de Ziegler and Nichols para el PID y unas ganancias optimizadas con una herramienta de MATLAB[®] y de esta forma se establecieron

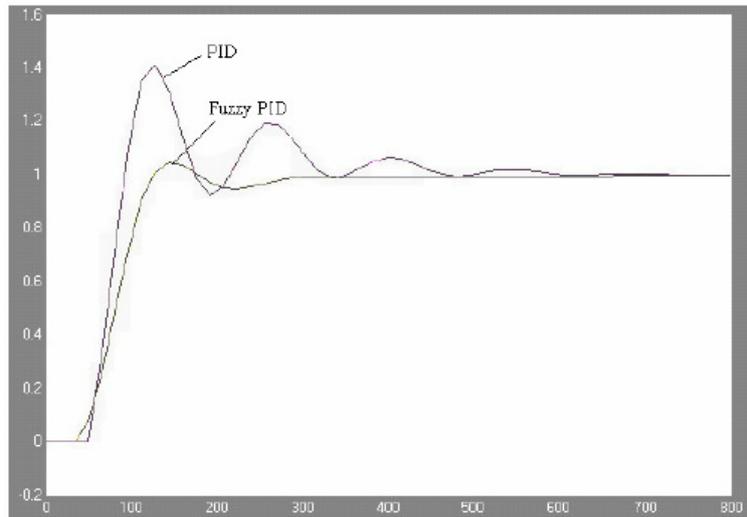


Figura 8.5: Resultados de la simulación del controlador PID y el controlador fuzzy PID [14].

los rangos para las ganancias del controlador PID. La planta es un sistema de dos tanques interactuantes al igual que en el trabajo de esta tesis, pero con dimensiones distintas. En la Fig.8.7 se muestra una comparación de la respuesta de la planta con cada uno de los controladores, y se puede observar que el controlador PID es más efectivo que el controlador PID difuso. Es importante observar que estos resultados contrastan con los obtenidos en este trabajo de tesis, ya que en el trabajo de tesis el controlador PID difuso es el más eficiente.

En [16] se presentó un lazo de control, el cual se compone de un motor DC, un módulo IPM para manejar el motor, un tacómetro hecho con un encoder es usado para retroalimentar la velocidad a la que está girando la flecha del motor y un DSPF2812 se usa para generar pulsos PWM para alimentar el motor. El algoritmo que se usa en este artículo es parecido al usado en este trabajo de tesis (controlador difuso incremental) pero la planta a diferencia de la tesis es un motor DC. En la Tabla 8.2 se puede observar los resultados de la simulación. Comparando con el trabajo de tesis se puede observar que la respuesta es mucho más rápida que la mostrada en la tesis, ya que la dinámica de la planta que se muestra en el artículo es totalmente distinta a la del trabajo de tesis, aunque el diseño del controlador mantiene la idea básica de los incrementos en la señal de control. En [25] se hace una comparación de tres algoritmos de control FLC, PID y STFPID (que es un algoritmo PID autoajustable con lógica difusa) y la planta a controlar es un motor de inducción, en donde la variable a controlar es la velocidad de la flecha en rad/seg. Al igual que en la tesis se muestra una comparación entre los tres algoritmos, pero en [25] se puede observar que el controlador STFPID tiene un sobrepico mayor al PID y al FLC como se muestra en la Fig.8.8.

En la Tabla 8.2 se muestran los resultados obtenidos en el estado del arte, así como los resultados obtenidos en este trabajo de tesis.

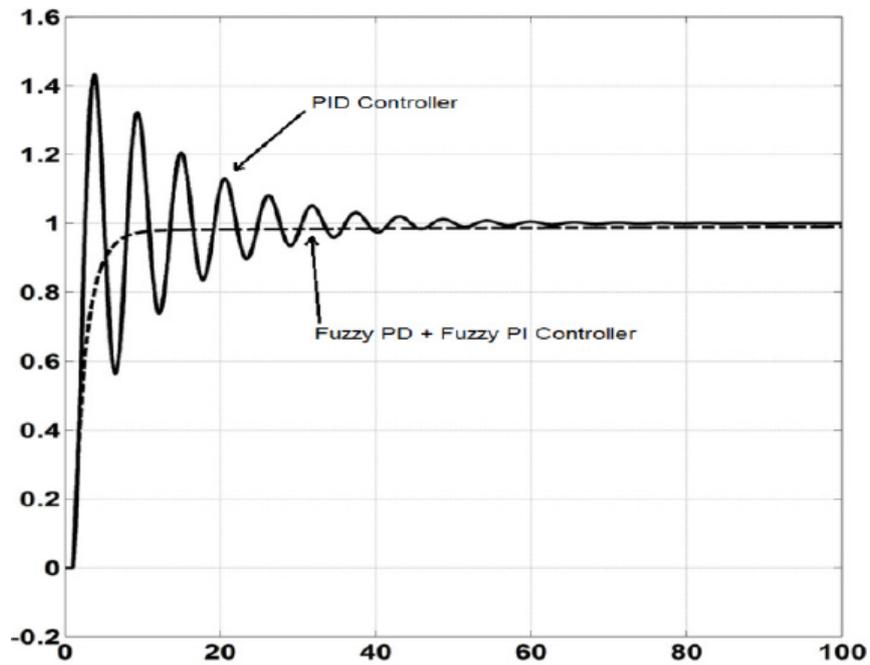


Figura 8.6: Resultados de la simulación del controlador PID y un controlador fuzzy PD+PID [47].

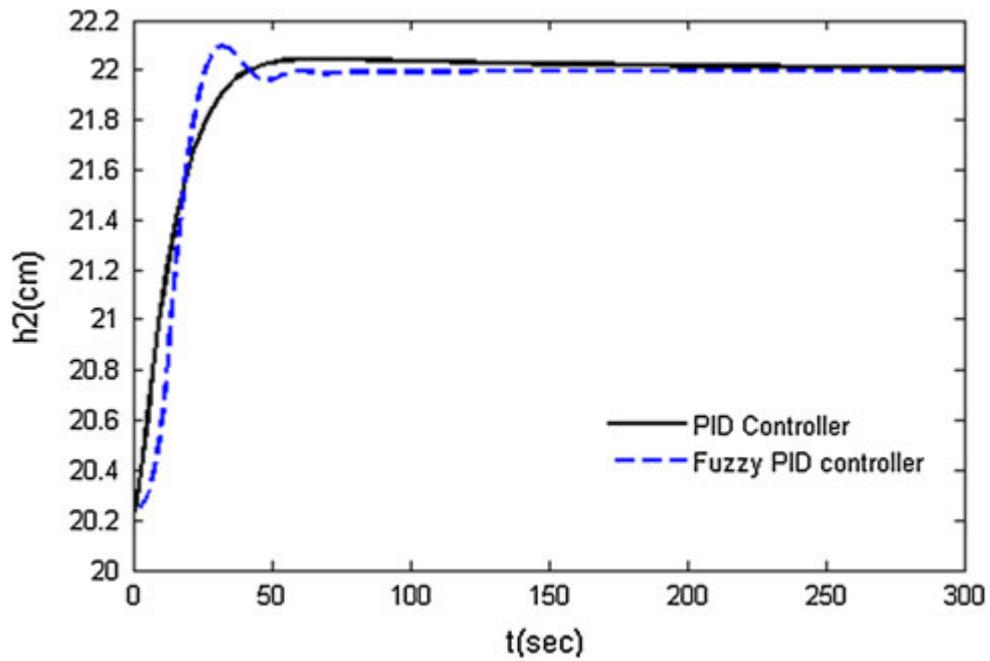


Figura 8.7: Resultados de la simulación del controlador PID y un controlador fuzzy PID [51].

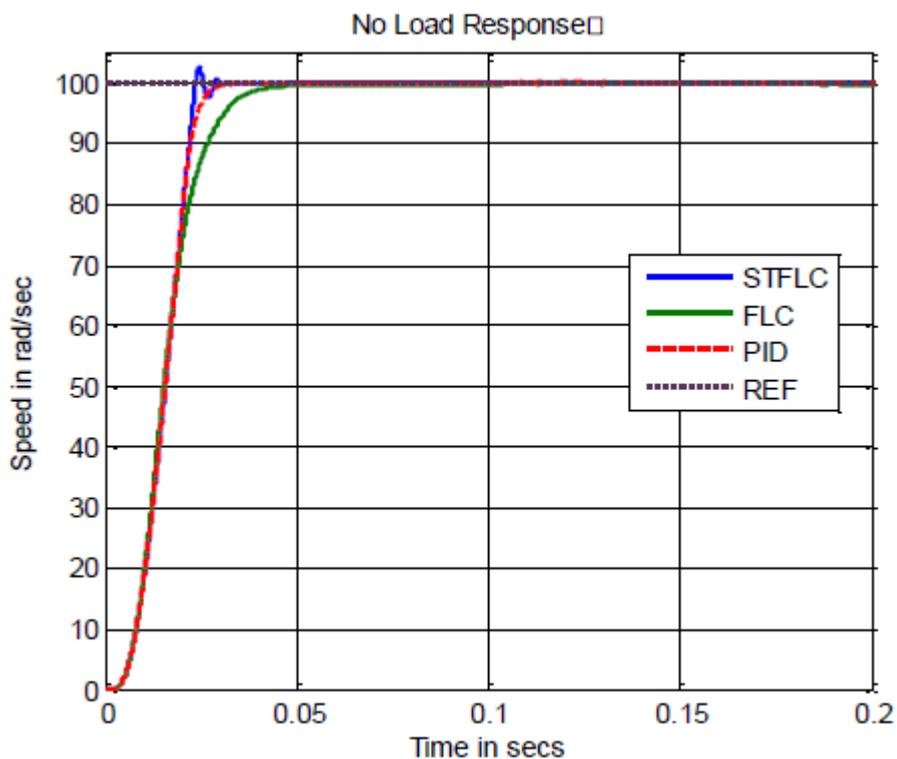


Figura 8.8: Resultados de la simulación del controlador PID, un controlador FLC y un controlador STFPID [25].

Tabla 8.2: Resultados del estado del arte

FUENTE	CONTROLADOR	TIEMPO DE ASENT.	SOBREPASO MÁXIMO
13	PID	5520s	N.A
16	FLC	0.7s	0.30 %
47	PID	67s	43.40 %
	FUZZY PD+PID	23s	0.72 %
25	FLC	0.06s	-0.0763
	PID	0.05s	-0.1344
	STFPID	0.03s	2.43
TESIS	FLC	4.250s	0.83 %
	PID	2.950s	2.80 %
	PID AUTOAJUSTABLE	2.550s	0.89 %

CONCLUSIONES Y TRABAJO FUTURO

Conclusiones.

Es posible desarrollar un sistema de control en forma distribuida con diversas herramientas tales como LabVIEW[®] y MATLAB[®], en los cuales se puede desarrollar algoritmos nuevos en diversas áreas de la ciencia y la tecnología, así como también se pueden usar algoritmos previamente diseñados y programados por el mismo proveedor de estas herramientas, este trabajo demuestra que es posible la programación y el uso de herramientas alternativas que pueden ser opciones para cubrir las necesidades de supervisión y control en los sectores productivos del país. cubriendo también los requisitos de seguridad en el envío y recepción de información así como las restricciones de tiempo que se debe cubrir con respecto al tiempo que tarda en llegar un paquete desde que el emisor lo envía hasta que el receptor recibe la información.

Con base en el desempeño de los diferentes algoritmos de control, se puede concluir que el mejor rendimiento sobre el control de la planta lo tiene el controlador PID autoajustable con lógica difusa ya que su tiempo de asentamiento es menor en comparación con un controlador PID y un controlador difuso, a pesar también de que el sobrepaso máximo es mayor que el controlador difuso (ver Tabla 8.1). es posible concluir esto ya que en la simulación se observa que el controlador PID en algunas ocasiones puede llegar a derramar el líquido contenido en el tanque 1. aunque el controlador difuso no derrama el líquido en el tanque 1 el tiempo de asentamiento es mucho mayor que el controlador PID autoajustable, es por estas razones que se puede concluir que el mejor rendimiento sobre la planta lo tiene el controlador PID autoajustable.

En este trabajo de tesis se cumplieron los siguientes objetivos:

- Se desarrolló una estrategia de control en un entorno distribuido, en el que se definieron los nodos supervisores y locales.
Se partió de la estructura básica de un sistema cliente servidor TCP/IP, el cual establece la comunicación entre una computadora servidor y una computadora cliente. Este sistema se modificó de tal manera que el servidor fuera capaz de establecer la comunicación con n clientes, y de esta manera poder establecer un sistema cliente servidor TCP/IP con múltiples clientes (ver Cap.4).



- Se implementó el modelo dentro de una simulación dinámica.
Se buscó una función de transferencia de la planta que describiera la dinámica de la misma en el dominio de Laplace, una vez obtenida la función de transferencia, se obtuvo la función de transferencia pulso para la planta con un retenedor de orden cero y se implementó la función de transferencia en el lenguaje de programación gráfica LabVIEW® (ver Cap.5).
- Se programó un controlador PID con autoajuste con un controlador difuso, que controla las ganancias del PID en forma automática tomando la respuesta de la planta como entrada al controlador difuso.
Se encontró las ganancias de un controlador PID mediante el método gráfico de Ziegler and Nichols, y a partir de estas ganancias se encuentra el rango de las variables de salida para un controlador difuso el cual controlara en forma automática y conforme la dinámica de planta evoluciona las ganancias de un controlador PID para la planta (ver Cap.7).
- Se integró un sistema cliente servidor de múltiples nodos con los diferentes algoritmos de control (ver Cap.6).

Se cumplió la hipótesis la cual fue verificada, pues se desarrolló un sistema cliente/servidor con todas las funciones previstas y los tiempos de comunicación cumplieron las restricciones impuestas.

Las aportaciones logradas en este trabajo, pueden ser útiles en el desarrollo e implementación de sistemas reales en la industria, sustentadas con base en los desarrollos teóricos y prácticos del presente trabajo.

Conclusiones de los resultados obtenidos.

En este trabajo de tesis se programaron tres algoritmos distintos de control para la planta de dos tanques interactuantes cuyos resultados se pueden observar en la Tabla 8.1, de estos resultados es posible concluir que la mejor respuesta es la que se muestra con el controlador PID autoajutable ya que a pesar de que tiene un sobrepaso máximo de 0.89 %, y el controlador FLC tiene un sobrepaso máximo de 0.83 %, el tiempo de asentamiento es de 2550 ms y este tiempo es menor que el tiempo de asentamiento con el controlador FLC. Si se compara con el controlador PID clásico se puede observar que la diferencia que existe en el tiempo de asentamiento entre ambos controladores es de 400ms ya que el PID clásico le toma 2950ms contra 2550ms del controlador PID autoajutable, pero se puede observar que el PID autoajutable elimina el sobrepaso máximo ya que se observa un 0.83 % del PID autoajutable contra un 2.80 % del PID clásico.

Los resultados obtenidos en el estado del arte se resumen en la Tabla 8.2 y se puede observar que estos varían demasiado ya que las plantas a controlar son absolutamente distintas, pero los algoritmos aplicados mantienen la idea básica en cuanto a



mejorar la respuesta de un controlador PID clásico buscando algunas otras alternativas con lógica difusa principalmente.

En [47] se logró reducir el tiempo de asentamiento de 67s a 23s, y el sobrepaso máximo de 43.40 % a 0.74 %, en el caso de [25] se logra reducir el tiempo de asentamiento ya que pasó de 0.05s a 0.03s, aunque es bueno observar que el sobrepaso máximo aumento demasiado ya que paso de un valor de -0.1344 a 2.43, en el caso de este trabajo de tesis se mejoró la respuesta ya que paso de un tiempo de asentamiento de 2950ms a 2550ms y un sobrepaso máximo de 2.80 % a 0.895 %. este trabajo de tesis no solo mejoró la respuesta del sistema con un controlador PID autoajustable sino que también se programó un sistema de comunicación entre un servidor y múltiples clientes, en donde la información a comunicar son parámetros de control del sistema de tanques interactuantes.

Trabajo futuro.

Como trabajo futuro se puede destacar como objetivo principal la implementación del sistema de control distribuido en una planta de producción en donde sea necesario el monitoreo y el control de procesos. para lo cual se pueden tener los siguientes objetivos parciales:

- Establecer un controlador PID autoajustable con lógica difusa, el cual modifique las ganancias del controlador PID con un algoritmo PD incremental, y comparar las respuestas del sistema con lo ya establecido en la tesis.
- Establecer una red neuronal artificial, la cual modifique los parámetros del controlador PID y comparar el desempeño del controlador con respecto a los ya establecidos.
- Establecer un controlador neurodifuso para la planta y comparar el desempeño del controlador con respecto a los ya establecidos.
- Hacer pruebas en donde los algoritmos de control se ejecuten en la computadora y los datos de entrada y salida al controlador se obtengan mediante una tarjeta de adquisición de datos. comparar los resultados con los del presente trabajo.

Anexo A

A.1. Función de transferencia PNS

A partir de (5.5) y (5.6) que corresponden al modelo lineal de la planta de tanques interactuantes y considerando que $q_1 = \frac{h_1 - h_2}{R_1}$ y que $q_2 = \frac{h_2}{R_2}$ (Véase Cap. 5) se tiene que para el tanque 1 la ecuación del balance de masa se muestra en (A.1).

$$C_1 * \frac{dh_1}{dt} = q_i - \frac{h_1 - h_2}{R_1} \quad (\text{A.1})$$

Para el tanque 2 el balance de masa queda descrito por (A.2).

$$C_2 * \frac{dh_2}{dt} = \frac{h_1 - h_2}{R_1} - \frac{h_2}{R_2} \quad (\text{A.2})$$

Para encontrar una función de transferencia que relacione como entrada al sistema q_i , que es el flujo que entra al tanque 1. Y como salida del sistema el nivel del líquido en el tanque 2 h_2 . primero se despeja de (A.1) y (A.2) a h_1 , Obteniéndose de este despeje (A.3) y (A.4).

$$C_1 \frac{dh_1}{dt} + \frac{h_1}{R_1} = q_i + \frac{h_2}{R_1} \quad (\text{A.3})$$

$$h_1 = R_1 C_2 \frac{dh_2}{dt} + h_2 + \frac{R_1}{R_2} h_2 \quad (\text{A.4})$$

Ahora sustituyendo (A.4) en (A.3) se tiene que:

$$C_1 R_1 C_2 \frac{d^2 h_2}{dt^2} + C_1 \frac{dh_2}{dt} + \frac{C_1 R_1}{R_2} \frac{dh_2}{dt} + C_2 \frac{dh_2}{dt} + \frac{h_2}{R_1} + \frac{h_2}{R_2} = q_i + \frac{h_2}{R_1}$$

Agrupando términos se tiene que:

$$C_1 R_1 C_2 \frac{d^2 h_2}{dt^2} + (C_1 + \frac{C_1 R_1}{R_2} + C_2) \frac{dh_2}{dt} + \frac{h_2}{R_2} = q_i$$

Multiplicando toda la ecuación por R_2 se tiene que:

$$R_2 C_2 R_1 C_1 \frac{d^2 h_2}{dt^2} + (R_2 C_1 + C_1 R_1 + R_2 C_2) \frac{dh_2}{dt} + h_2 = R_2 q_i \quad (\text{A.5})$$



Aplicando transformada de Laplace a (A.5) se tiene que:

$$[R_2C_2R_1C_1S^2 + (R_2C_1 + C_1R_1 + R_2C_2)S + 1]H_2(S) = R_2Q_i(S)$$

Y finalmente despejando se tiene la función de transferencia que relaciona $\frac{H_2(S)}{Q_i(S)}$.

$$\frac{H_2(S)}{Q_i(S)} = \frac{R_2}{R_2C_2R_1C_1S^2 + (R_2C_1 + C_1R_1 + R_2C_2)S + 1} \quad (\text{A.6})$$

A.2. Reglas de Inferencia PNS

A continuación se muestran las reglas de inferencia que se implementaron en el controlador difuso para el proceso de nivel simulado.

- if dh/dt is NL and e is NL then DU(k) is Z
- if dh/dt is NL and e is NS then DU(k) is PS
- if dh/dt is NL and e is Z then DU(k) is PM
- if dh/dt is NL and e is PS then DU(k) is PL
- if dh/dt is NL and e is PL then DU(k) is PL
- if dh/dt is NS and e is NL then DU(k) is NS
- if dh/dt is NS and e is NS then DU(k) is Z
- if dh/dt is NS and e is Z then DU(k) is PS
- if dh/dt is NS and e is PS then DU(k) is PM
- if dh/dt is NS and e is PL then DU(k) is PL
- if dh/dt is Z and e is NL then DU(k) is NL
- if dh/dt is Z and e is NS then DU(k) is NS
- if dh/dt is Z and e is Z then DU(k) is Z
- if dh/dt is Z and e is PS then DU(k) is PS
- if dh/dt is Z and e is PL then DU(k) is PL
- if dh/dt is PS and e is NL then DU(k) is NL
- if dh/dt is PS and e is NS then DU(k) is NM
- if dh/dt is PS and e is Z then DU(k) is NS
- if dh/dt is PS and e is PS then DU(k) is Z



- if dh/dt is PS and e is PL then $DU(k)$ is PS
- if dh/dt is PL and e is NL then $DU(k)$ is NL
- if dh/dt is PL and e is NS then $DU(k)$ is NL
- if dh/dt is PL and e is Z then $DU(k)$ is NM
- if dh/dt is PL and e is PS then $DU(k)$ is NS
- if dh/dt is PL and e is PL then $DU(k)$ is Z

A.3. Superficies de control y reglas de inferencia para PID autoajutable.

En esta sección se presentan las reglas difusas del motor de inferencias del controlador difuso, así como también se muestra las superficies de control para cada uno de los parámetros (K_p , T_i , T_d) del controlador PID autoajutable que se plantea en el Cap.7

A continuación se muestran las reglas de inferencia que se implementaron en el controlador difuso para el autoajuste del controlador PID.

- if error is NL and dh/dt is NL then K_p is PL also T_i is PL also T_d is Z
- if error is NL and dh/dt is NM then K_p is PL also T_i is PL also T_d is PS
- if error is NL and dh/dt is NS then K_p is PL also T_i is PL also T_d is PM
- if error is NL and dh/dt is Z then K_p is PL also T_i is PL also T_d is PL
- if error is NL and dh/dt is PS then K_p is PL also T_i is PL also T_d is PM
- if error is NL and dh/dt is PM then K_p is PL also T_i is PL also T_d is PS
- if error is NL and dh/dt is PL then K_p is PL also T_i is PL also T_d is Z
- if error is NM and dh/dt is NL then K_p is PM also T_i is PM also T_d is Z
- if error is NM and dh/dt is NM then K_p is PM also T_i is PM also T_d is PS
- if error is NM and dh/dt is NS then K_p is PM also T_i is PM also T_d is PM
- if error is NM and dh/dt is Z then K_p is PM also T_i is PM also T_d is PL
- if error is NM and dh/dt is PS then K_p is PM also T_i is PM also T_d is PM
- if error is NM and dh/dt is PM then K_p is PM also T_i is PM also T_d is PS
- if error is NM and dh/dt is PL then K_p is PM also T_i is PM also T_d is Z



- if error is NS and dh/dt is NL then K_p is PS also T_i is PS also T_d is Z
- if error is NS and dh/dt is NM then K_p is PS also T_i is PS also T_d is PS
- if error is NS and dh/dt is NS then K_p is PS also T_i is PS also T_d is PM
- if error is NS and dh/dt is Z then K_p is PS also T_i is PS also T_d is PL
- if error is NS and dh/dt is PS then K_p is PS also T_i is PS also T_d is PM
- if error is NS and dh/dt is PM then K_p is PS also T_i is PS also T_d is PS
- if error is NS and dh/dt is PL then K_p is PS also T_i is PS also T_d is Z
- if error is Z and dh/dt is NL then K_p is Z also T_i is Z also T_d is Z
- if error is Z and dh/dt is NM then K_p is Z also T_i is Z also T_d is PS
- if error is Z and dh/dt is NS then K_p is Z also T_i is Z also T_d is PM
- if error is Z and dh/dt is Z then K_p is Z also T_i is Z also T_d is PL
- if error is Z and dh/dt is PS then K_p is Z also T_i is Z also T_d is PM
- if error is Z and dh/dt is PM then K_p is Z also T_i is Z also T_d is PS
- if error is Z and dh/dt is PL then K_p is Z also T_i is Z also T_d is Z
- if error is PS and dh/dt is NL then K_p is PS also T_i is PS also T_d is Z
- if error is PS and dh/dt is NM then K_p is PS also T_i is PS also T_d is PS
- if error is PS and dh/dt is NS then K_p is PS also T_i is PS also T_d is PM
- if error is PS and dh/dt is Z then K_p is PS also T_i is PS also T_d is PL
- if error is PS and dh/dt is PS then K_p is PS also T_i is PS also T_d is PM
- if error is PS and dh/dt is PM then K_p is PS also T_i is PS also T_d is PS
- if error is PS and dh/dt is PL then K_p is PS also T_i is PS also T_d is Z
- if error is PM and dh/dt is NL then K_p is PM also T_i is PM also T_d is Z
- if error is PM and dh/dt is NM then K_p is PM also T_i is PM also T_d is PS
- if error is PM and dh/dt is NS then K_p is PM also T_i is PM also T_d is PM
- if error is PM and dh/dt is Z then K_p is PM also T_i is PM also T_d is PL
- if error is PM and dh/dt is PS then K_p is PM also T_i is PM also T_d is PM
- if error is PM and dh/dt is PM then K_p is PM also T_i is PM also T_d is PS



- if error is PM and dh/dt is PL then Kp is PM also Ti is PM also Td is Z
- if error is PL and dh/dt is NL then Kp is PL also Ti is PL also Td is Z
- if error is PL and dh/dt is NM then Kp is PL also Ti is PL also Td is PS
- if error is PL and dh/dt is NS then Kp is PL also Ti is PL also Td is PM
- if error is PL and dh/dt is Z then Kp is PL also Ti is PL also Td is PL
- if error is PL and dh/dt is PS then Kp is PL also Ti is PL also Td is PM
- if error is PL and dh/dt is PM then Kp is PL also Ti is PL also Td is PS
- if error is PL and dh/dt is PL then Kp is PL also Ti is PL also Td is Z

En la Fig. A.1 se muestra la superficie de control correspondiente a la ganancia proporcional K_p del controlador PID.

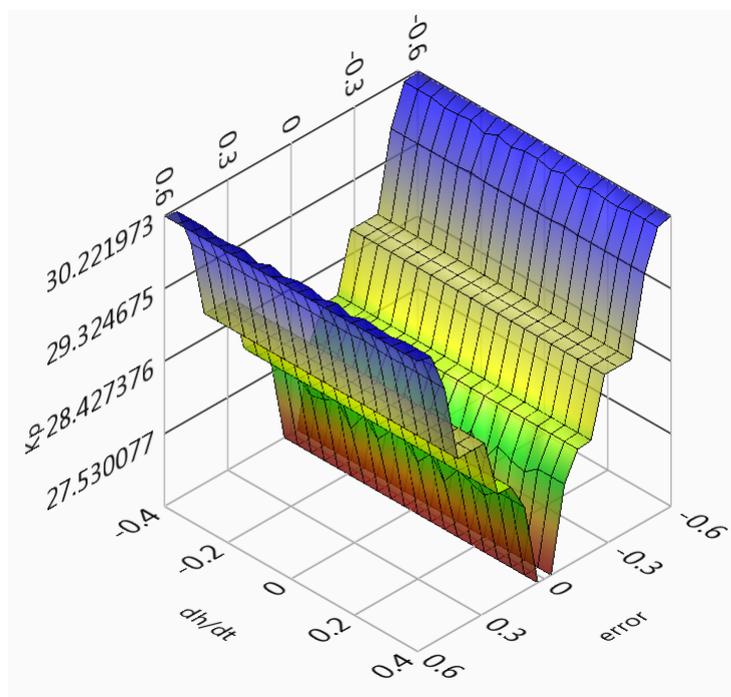


Figura A.1: Superficie de control para la ganancia proporcional K_p .

En la Fig. A.2 se muestra la superficie de control correspondiente al tiempo integral T_i del controlador PID.

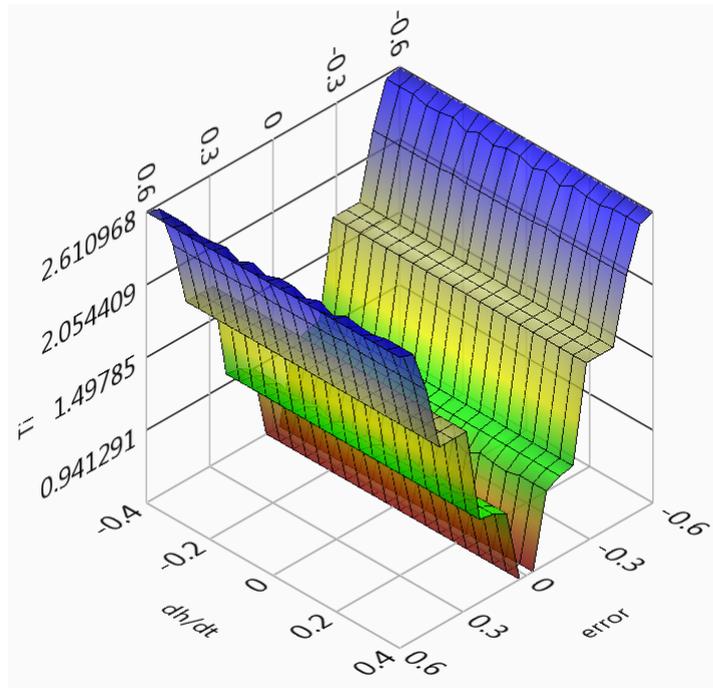


Figura A.2: Superficie de control para el tiempo integral T_i .

En la Fig. A.3 se muestra la superficie de control correspondiente al tiempo derivativo T_d del controlador PID.

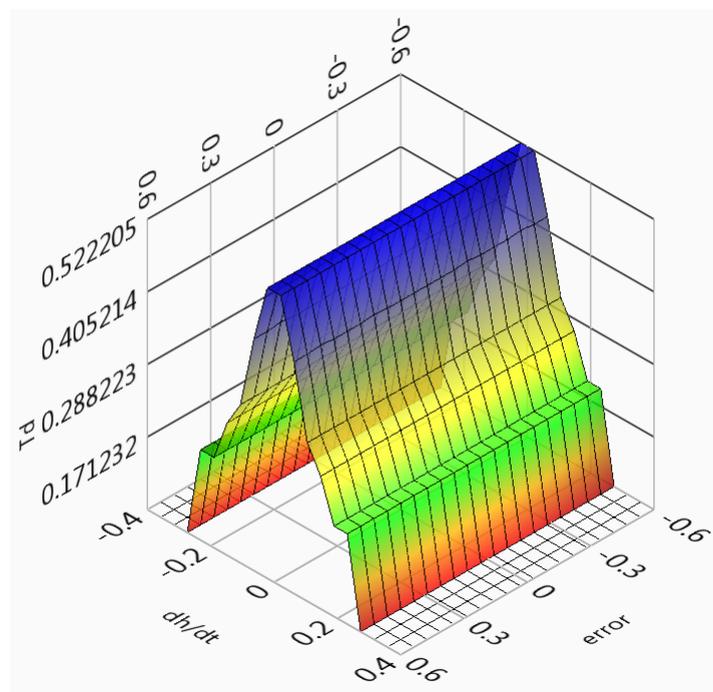


Figura A.3: Superficie de control para el tiempo derivativo T_d .





A.4. Características de Buses y Protocolos

Tabla A.1: Características de algunos Buses y Protocolos

Comparación de características entre algunos Buses y Protocolos						
Nombre	Topología	Soporte	Max Dis-positivos	Rate transm. bps	Distancia máx. Km	Comunicación
Profibus DP	Línea, Estrella y Anillo	Par trenzado y Fibra Óptica	127/segm	Hasta 1.5M y 12M	0.1 segm. 24 Fibra	Master/Slave peer to peer
Profibus PA	Línea, Estrella y Anillo	Par Trenzado y Fibra Óptica	14400/segm	31.5K	0.1 segm 24 Fibra	Master/Slave peer to peer
Profibus FMS		Par Trenzado y Fibra Óptica	127/segm	500K		Master/Slave peer to peer
Foundation Fieldbus HMS	Estrella	Par Trenzado y Fibra Óptica	240p/segm 32.768sist.	100M	0.1par 2Fibra	Single/Multi master
Foundation Fieldbus H1	Estrella o Bus	Par Trenzado, Fibra Óptica, Coaxial, Radio	240p/segm 32,768sist.	31.25K	1.9 cable	Single/Multi master
DeviveNet	Troncal, Puntual, c/bifurcación	Par Trenzado y Fibra Óptica	2048 nodos	500K	0.5 6c/repetid	Master/Slave, Multi-master, peer to peer
Ethernet Industrial	Bus, Estrella, Malla, Cadena	Coaxial, Par Trenzado y Fibra Óptica	400p/segm	10, 100M	0.150 mono c/switch	Master/Slave peer to peer
HART		Par Trenzado	15p/segm	1.2K		95 Master/Slave



A.4.1. Estructura de un Segmento TCP.

La estructura de un segmento TCP es la que se muestra en la Fig.A.4.

Puerto TCP Origen				Puerto TCP Destino			
Número de secuencia							
Número de ACK							
long CAB	Reservado	U R G	A C K	P S H	R S T	S Y N	F I N
Suma de comprobación				Puntero urgente			
Opciones							
Datos							

Figura A.4: Estructura de un Segmento TCP.

Los campos **Puerto TCP Origen** y **Puerto TCP Destino** identifican la conexión a la que pertenece el segmento.

El campo **Número de secuencia** indica el orden dentro de la transmisión del primer byte de datos contenido en el segmento. Junto al tamaño total del paquete IP, que permite conocer cuántos bytes de datos se reciben, es posible determinar cuál será el número de secuencia del siguiente segmento.

El campo **Número de ACK** o Número de reconocimiento, indica cual es el byte que el emisor del segmento espera recibir como número de secuencia. Esto implica que todos los bytes transmitidos en segmentos anteriores han llegado de forma satisfactoria. Solo es significativo cuando el bit ACK del campo **Código** está en 1.

El campo **Long CAB**, o longitud de cabecera, indica cuantas palabras de 32 bits (4 bytes) componen la cabecera TCP.

El campo **Código** consta de seis bits, teniendo cada uno de ellos un significado diferente del resto.

- El bit **ACK** se usa para indicar que el valor transportado en el campo de reconocimiento es válido, es decir, que el segmento contiene un reconocimiento para un segmento que ha sido recibido correctamente.
- Los bits **RST**, **SYN** y **FIN** se utilizan para el establecimiento y cierre de conexiones.
- La activación del bit **PSH** indica que el receptor deberá pasar los datos a la capa superior de forma inmediata.
- El bit **URG** indica que hay datos en este segmento que la entidad de la capa superior del lado emisor ha marcado como 'urgentes'.



El campo **Ventana** o ventana indica el tamaño de la ventana de recepción del emisor del segmento, es decir, el espacio en memoria disponible en el receptor, este campo limita la ventana de transmisión del receptor del segmento, limitando el número de bytes que puede transmitir sin recibir reconocimientos.

El campo **Suma de comprobación** contiene un código detector de errores.

El campo **Puntero urgente** o puntero a datos urgentes permite, junto al bit URG, la transmisión de datos urgentes no sujetos al control de flujo.

Finalmente, pueden existir una o varias **Opciones** que permiten gestionar aspectos como el tamaño máximo de segmento, los reconocimientos selectivos, etc.

A.4.2. Estructura del Datagrama IP

La cabecera de internet contiene además de las direcciones origen y destino, una serie de parámetros que son esenciales para la operación de los ruteadores, el formato de un datagrama se muestra en la Fig. A.5.

0	4	8	16	19	24	31
VERSIÓN	CAB	TIPO SERVICIO		LONGITUD TOTAL		
IDENTIFICACIÓN			BANDERA	DESPLAZAMIENTO		
TIEMPO DE VIDA		PROTOCOLO	SUMA DE COMPROBACIÓN			
DIRECCIÓN IP FUENTE						
DIRECCIÓN IP DESTINO						
OPCIONES IP						
DATOS						

Figura A.5: Estructura de un Datagrama IP.

- El campo **VERSIÓN** o número de versión, contiene la información referente a la versión del protocolo IP del datagrama, a partir de esta información el ruteador sabe o determina como se interpreta el resto de la información contenida en el datagrama IP, las distintas versiones de IP usan un formato distinto para los datagramas. Las versiones que actualmente se manejan son IPv4 e IPv6.
- El campo **CAB** o longitud de cabecera, con una longitud de 4 bits, contiene la información de donde comienzan realmente los datos, ya que debido al campo de opciones un datagrama de IPv4 puede ser de longitud variable. La mayoría de los datagramas IP no contienen opciones por lo tanto un datagrama típico tiene una cabecera de 20 bytes.



- El campo **TIPO SERVICIO** o tipo de servicio, ésta información se incluye en la cabecera de los datagramas IPv4 con una longitud de 8 bits, y sirve para diferenciar entre distinto tipo de datagramas IP, por ejemplo resulta de utilidad poder diferenciar entre tráfico de tiempo real y tráfico que no es de tiempo real.
- El campo **LONGITUD TOTAL** o longitud total del datagrama, sirve para especificar la longitud total del datagrama (cabecera más datos), y tiene una longitud de 16 bits, por esta razón el tamaño máximo teórico de un datagrama IP es de 65535 bytes, no obstante, estos datagramas rara vez tienen una longitud mayor que 1500 bytes.
- Los campos **IDENTIFICACIÓN**, **BANDERA** y **DESPLAZAMIENTO** o identificador, indicadores y desplazamiento de fragmentación, estos tres campos contienen la información que tiene que ver con la fragmentación IP (fragmentación: algunos protocolos pueden transportar datagramas grandes mientras que otros protocolos transportan datagramas pequeños, en otras palabras dependiendo el protocolo la longitud del datagrama será variable, la cantidad máxima de datos que una trama de la capa de enlace puede transportar se le llama Unidad Máxima de Transmisión (MTU, Maximum Transfer Unit) puesto que cada datagrama IP se encapsula dentro de una trama de la capa de enlace para ir de un ruteador al siguiente, la MTU del protocolo de la capa de enlace impone un límite estricto a la longitud de un datagrama IP, de esta manera si se recibe un datagrama IP con una longitud mayor a la MTU del protocolo de la capa de enlace, es necesario dividir el datagrama IP de forma tal que sea posible enviarlo dentro de la trama de la capa de enlace).
- El campo **TIEMPO DE VIDA**, con una longitud de 8 bits, se incluye dentro de la cabecera IP con la finalidad de que el datagrama no permanezca circulando en la red de forma infinita, este campo se disminuye en una unidad cada que un ruteador procesa el datagrama IP, de tal manera que si este campo tiene un valor de cero el datagrama IP se descarta.
- El campo **PROTOCOLO**, éste campo se incluye dentro de la cabecera IP con una longitud de 8 bits y se usa únicamente cuando el datagrama IP alcanza su destino final, con el propósito de indicar el protocolo específico de la capa de transporte al que se pasaran los datos contenidos en el datagrama IP. Por ejemplo, un valor de 6 indica que los datos se pasaran a TCP, mientras que un valor de 17 indica que los datos se pasaran a UDP.
- El campo **SUMA DE COMPROBACIÓN** o suma de comprobación de cabecera, éste campo se incluye dentro de la cabecera IP con una longitud de 16 bits, y esta suma de comprobación ayuda a los ruteadores a detectar errores de bit en un datagrama recibido, un ruteador calcula la suma de comprobación para cada datagrama recibido y calcula la condición de error comparando si la suma de comprobación calculada corresponde a la suma de comprobación



que se incluye en el campo **SUMA DE COMPROBACIÓN**, es decir, si la diferencia entre la suma de comprobación calculada y la suma de comprobación que se incluye dentro de la cabecera del datagrama IP es diferente de cero, en este caso el datagrama se descarta debido a dicha condición de error.

- Los campos **DIRECCIÓN IP FUENTE** y **DIRECCIÓN IP DESTINO** o dirección IP origen y dirección IP destino, cuando un emisor crea un datagrama inserta dentro de la cabecera una dirección de origen y una dirección de destino, con la finalidad de dar información a los ruteadores de quien envía el mensaje y a donde tiene que llegar.
- El campo **OPCIONES IP** u opciones de IP, este campo se incluye dentro de la cabecera IP de forma opcional, y cuando este campo se incluye dentro de la cabecera IP permite que esta se amplíe. La existencia de opciones dentro de la cabecera IP complica un poco las cosas debido a que no se sabe con anticipación donde comienzan los datos, la consecuencia de esto es que el tiempo que un ruteador emplea en procesar un datagrama es variable. Por esta razón en la versión IPv6 se eliminaron las opciones.
- El campo **DATOS**, se incluye dentro del datagrama IP, y es la razón de ser del datagrama IP, porque contiene la información que será entregada a la capa de transporte en el destino final.

A.5. Resultados de la simulación.

En esta tabla se muestran los resultados de la simulación, se puede observar los resultados en cada instante de muestreo para cada algoritmo de control, que en este caso son: FUZZY que corresponde al controlador difuso incremental, PID que corresponde al controlador PID clásico y PID autotuning que corresponde al controlador PID autoajutable con lógica difusa. (Véase Cap.8).

Time	FUZZY	PID	PID autotuning
0	0	0	0
1	0.00050067	0.0021333	0.00213331
2	0.00232981	0.0077938	0.00779377
3	0.00606826	0.0160693	0.0160693
4	0.0119339	0.0262617	0.0262617
5	0.0198477	0.0378366	0.0378366
6	0.0295891	0.0503857	0.0503857
7	0.0408633	0.0635976	0.0635976
8	0.05339	0.077235	0.077235
9	0.0668806	0.0911174	0.0911174
10	0.0810433	0.105108	0.105108
11	0.0956362	0.119105	0.119105

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
12	0.110473	0.13303	0.13303
13	0.125411	0.146825	0.146825
14	0.140346	0.160448	0.160448
15	0.155196	0.173868	0.173868
16	0.169902	0.187064	0.187064
17	0.184421	0.200018	0.200018
18	0.19872	0.212722	0.212722
19	0.212777	0.22517	0.22517
20	0.226576	0.237356	0.237356
21	0.240106	0.249282	0.249282
22	0.253357	0.260946	0.260946
23	0.266323	0.272352	0.272352
24	0.278991	0.283501	0.283501
25	0.291338	0.294398	0.294398
26	0.303321	0.305046	0.305046
27	0.314869	0.31545	0.31545
28	0.325885	0.325614	0.325614
29	0.336261	0.335543	0.335543
30	0.345895	0.345242	0.345242
31	0.354715	0.354716	0.354716
32	0.362686	0.363969	0.363969
33	0.369829	0.373007	0.373007
34	0.376233	0.381834	0.381834
35	0.382018	0.390455	0.390455
36	0.387301	0.398875	0.398875
37	0.392177	0.407098	0.407098
38	0.396721	0.415128	0.415128
39	0.400993	0.422971	0.422971
40	0.405043	0.430631	0.430631
41	0.408907	0.438111	0.438111
42	0.412615	0.445417	0.445417
43	0.416194	0.452552	0.452552
44	0.419667	0.459519	0.459519
45	0.423057	0.466324	0.466162
46	0.426382	0.472969	0.472208
47	0.429661	0.479459	0.477513
48	0.432906	0.485797	0.482071
49	0.436125	0.491904	0.485947
50	0.439319	0.49759	0.489231
51	0.442479	0.502647	0.492007
52	0.445589	0.506896	0.49434
53	0.448623	0.510204	0.496293

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
54	0.451547	0.512495	0.497956
55	0.454324	0.513747	0.499393
56	0.456913	0.514	0.500626
57	0.459282	0.513341	0.50167
58	0.461411	0.511903	0.502525
59	0.463299	0.509851	0.503195
60	0.464969	0.507368	0.5037
61	0.466452	0.504649	0.50406
62	0.46779	0.50188	0.504292
63	0.46902	0.499237	0.504411
64	0.470174	0.496869	0.504438
65	0.471281	0.494897	0.50439
66	0.472357	0.493404	0.504287
67	0.473418	0.49244	0.504145
68	0.47447	0.492016	0.503979
69	0.47552	0.492111	0.503802
70	0.476567	0.492672	0.503624
71	0.477613	0.493625	0.503451
72	0.478656	0.494877	0.503289
73	0.479692	0.496328	0.503139
74	0.48072	0.49787	0.503004
75	0.481736	0.499405	0.502883
76	0.482736	0.500838	0.502775
77	0.483719	0.502094	0.502679
78	0.484681	0.503112	0.502593
79	0.485621	0.503851	0.502515
80	0.486536	0.504293	0.502444
81	0.487426	0.504436	0.502377
82	0.488289	0.504299	0.502314
83	0.489124	0.503914	0.502253
84	0.489932	0.503328	0.502194
85	0.490713	0.502593	0.502136
86	0.491466	0.501766	0.502078
87	0.492191	0.500906	0.502021
88	0.49289	0.500068	0.501965
89	0.493563	0.499301	0.501909
90	0.49421	0.498643	0.501854
91	0.494833	0.498126	0.5018
92	0.495431	0.497768	0.501747
93	0.496007	0.497576	0.501695
94	0.49656	0.497548	0.501644
95	0.497092	0.49767	0.501595

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
96	0.497604	0.497922	0.501547
97	0.498096	0.498277	0.5015
98	0.498569	0.498704	0.501455
99	0.499024	0.499172	0.501412
100	0.499462	0.499648	0.501369
101	0.499883	0.500103	0.501329
102	0.500288	0.500511	0.501289
103	0.500678	0.500853	0.501251
104	0.501051	0.501112	0.501214
105	0.501406	0.501282	0.501178
106	0.501741	0.501359	0.501143
107	0.502055	0.501347	0.50111
108	0.502347	0.501255	0.501077
109	0.502617	0.501095	0.501045
110	0.502864	0.500884	0.501014
111	0.503089	0.500638	0.500984
112	0.503291	0.500375	0.500955
113	0.50347	0.500113	0.500927
114	0.503626	0.499867	0.500899
115	0.50376	0.499652	0.500873
116	0.503872	0.499476	0.500847
117	0.503963	0.499348	0.500822
118	0.504033	0.49927	0.500797
119	0.504083	0.499243	0.500774
120	0.504114	0.499265	0.500751
121	0.504127	0.499328	0.500728
122	0.504122	0.499427	0.500707
123	0.5041	0.499552	0.500686
124	0.504065	0.499692	0.500666
125	0.504019	0.499839	0.500646
126	0.503965	0.499982	0.500627
127	0.503904	0.500114	0.500608
128	0.503839	0.500227	0.50059
129	0.50377	0.500316	0.500572
130	0.503699	0.500379	0.500555
131	0.503626	0.500413	0.500539
132	0.503552	0.500419	0.500523
133	0.503477	0.500399	0.500507
134	0.503402	0.500357	0.500492
135	0.503326	0.500297	0.500478
136	0.503251	0.500225	0.500463
137	0.503176	0.500145	0.50045

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
138	0.503101	0.500064	0.500436
139	0.503027	0.499986	0.500423
140	0.502954	0.499916	0.500411
141	0.502881	0.499857	0.500399
142	0.502809	0.499812	0.500387
143	0.502738	0.499782	0.500375
144	0.502667	0.499768	0.500364
145	0.502598	0.49977	0.500353
146	0.502529	0.499785	0.500343
147	0.502461	0.499812	0.500333
148	0.502394	0.499847	0.500323
149	0.502328	0.499889	0.500313
150	0.502263	0.499934	0.500304
151	0.502199	0.499979	0.500295
152	0.502136	0.500021	0.500286
153	0.502073	0.500058	0.500278
154	0.502012	0.500088	0.500269
155	0.501951	0.50011	0.500261
156	0.501892	0.500124	0.500254
157	0.501833	0.500129	0.500246
158	0.501776	0.500126	0.500239
159	0.501719	0.500115	0.500232
160	0.501663	0.500099	0.500225
161	0.501608	0.500078	0.500218
162	0.501554	0.500054	0.500212
163	0.501501	0.500029	0.500205
164	0.501449	0.500004	0.500199
165	0.501398	0.499982	0.500193
166	0.501348	0.499962	0.500188
167	0.501299	0.499947	0.500182
168	0.50125	0.499936	0.500177
169	0.501203	0.49993	0.500171
170	0.501156	0.499929	0.500166
171	0.50111	0.499932	0.500161
172	0.501065	0.499939	0.500157
173	0.501021	0.499949	0.500152
174	0.500978	0.499961	0.500147
175	0.500936	0.499975	0.500143
176	0.500894	0.499989	0.500139
177	0.500853	0.500002	0.500135
178	0.500814	0.500014	0.500131
179	0.500774	0.500024	0.500127

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
180	0.500736	0.500032	0.500123
181	0.500699	0.500037	0.500119
182	0.500662	0.500039	0.500116
183	0.500626	0.500039	0.500112
184	0.500591	0.500037	0.500109
185	0.500557	0.500032	0.500106
186	0.500523	0.500026	0.500103
187	0.50049	0.500019	0.5001
188	0.500458	0.500012	0.500097
189	0.500427	0.500004	0.500094
190	0.500396	0.499997	0.500091
191	0.500366	0.499999	0.500088
192	0.500337	0.499985	0.500086
193	0.500308	0.499981	0.500083
194	0.50028	0.499979	0.500081
195	0.500253	0.499978	0.500078
196	0.500226	0.499978	0.500076
197	0.5002	0.49998	0.500074
198	0.500175	0.499983	0.500072
199	0.50015	0.499987	0.500069
200	0.500126	0.499991	0.500067
201	0.500102	0.499995	0.500065
202	0.500079	0.499999	0.500063
203	0.500057	0.500003	0.500062
204	0.500035	0.500006	0.50006
205	0.500014	0.500009	0.500058
206	0.499993	0.500011	0.500056
207	0.499973	0.500012	0.500055
208	0.499955	0.500012	0.500053
209	0.499942	0.500012	0.500051
210	0.499935	0.500011	0.50005
211	0.499937	0.500009	0.500048
212	0.499947	0.500007	0.500047
213	0.499965	0.500004	0.500046
214	0.49999	0.500002	0.500044
215	0.500019	0.5	0.500043
216	0.50005	0.499998	0.500042
217	0.500083	0.499996	0.50004
218	0.500112	0.499995	0.500039
219	0.500136	0.499994	0.500038
220	0.500154	0.499993	0.500037
221	0.500162	0.499993	0.500036

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
222	0.500161	0.499994	0.500035
223	0.500151	0.499994	0.500034
224	0.500131	0.499995	0.500033
225	0.500104	0.499997	0.500032
226	0.500073	0.499998	0.500031
227	0.500038	0.499999	0.50003
228	0.500001	0.5	0.500029
229	0.499962	0.500002	0.500028
230	0.499923	0.500002	0.500027
231	0.499885	0.500003	0.500026
232	0.49985	0.500004	0.500026
233	0.499822	0.500004	0.500025
234	0.499802	0.500004	0.500024
235	0.499792	0.500003	0.500023
236	0.499791	0.500003	0.500023
237	0.499801	0.500002	0.500022
238	0.499821	0.500002	0.500021
239	0.499851	0.500001	0.500021
240	0.499889	0.5	0.50002
241	0.499931	0.5	0.50002
242	0.499976	0.499999	0.500019
243	0.500024	0.499998	0.500018
244	0.500073	0.499998	0.500018
245	0.500121	0.499998	0.500017
246	0.500165	0.499998	0.500017
247	0.500203	0.499998	0.500016
248	0.500234	0.499998	0.500016
249	0.500254	0.499998	0.500015
250	0.500265	0.499999	0.500015
251	0.500265	0.499999	0.500015
252	0.500254	0.5	0.500014
253	0.500233	0.5	0.500014
254	0.500205	0.5	0.500013
255	0.500172	0.500001	0.500013
256	0.500135	0.500001	0.500012
257	0.500095	0.500001	0.500012
258	0.500054	0.500001	0.500012
259	0.500011	0.500001	0.500011
260	0.499969	0.500001	0.500011
261	0.499926	0.500001	0.500011
262	0.499885	0.500001	0.50001
263	0.499848	0.500001	0.50001

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
264	0.499818	0.5	0.50001
265	0.499797	0.5	0.50001
266	0.499785	0.5	0.500009
267	0.499783	0.5	0.500009
268	0.499792	0.5	0.500009
269	0.499812	0.499999	0.500008
270	0.499841	0.499999	0.500008
271	0.499878	0.499999	0.500008
272	0.499919	0.499999	0.500008
273	0.499964	0.499999	0.500007
274	0.500011	0.499999	0.500007
275	0.50006	0.5	0.500007
276	0.500108	0.5	0.500007
277	0.500152	0.5	0.500007
278	0.50019	0.5	0.500006
279	0.500219	0.5	0.500006
280	0.50024	0.5	0.500006
281	0.50025	0.5	0.500006
282	0.50025	0.5	0.500006
283	0.500239	0.5	0.500006
284	0.500218	0.5	0.500005
285	0.50019	0.5	0.500005
286	0.500156	0.5	0.500005
287	0.500119	0.5	0.500005
288	0.500079	0.5	0.500005
289	0.500038	0.5	0.500005
290	0.499996	0.5	0.500004
291	0.499953	0.5	0.500004
292	0.499911	0.5	0.500004
293	0.499874	0.5	0.500004
294	0.499844	0.5	0.500004
295	0.499821	0.5	0.500004
296	0.499808	0.5	0.500004
297	0.499806	0.5	0.500004
298	0.499813	0.5	0.500004
299	0.499832	0.5	0.500003
300	0.49986	0.5	0.500003
301	0.499896	0.5	0.500003
302	0.499936	0.5	0.500003
303	0.49998	0.5	0.500003
304	0.500026	0.5	0.500003
305	0.500074	0.5	0.500003

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
306	0.50012	0.5	0.500003
307	0.500163	0.5	0.500003
308	0.5002	0.5	0.500003
309	0.500228	0.5	0.500003
310	0.500248	0.5	0.500002
311	0.500257	0.5	0.500002
312	0.500256	0.5	0.500002
313	0.500244	0.5	0.500002
314	0.500222	0.5	0.500002
315	0.500193	0.5	0.500002
316	0.500158	0.5	0.500002
317	0.50012	0.5	0.500002
318	0.50008	0.5	0.500002
319	0.500038	0.5	0.500002
320	0.499994	0.5	0.500002
321	0.499951	0.5	0.500002
322	0.499909	0.5	0.500002
323	0.499871	0.5	0.500002
324	0.49984	0.5	0.500002
325	0.499817	0.5	0.500002
326	0.499803	0.5	0.500002
327	0.499799	0.5	0.500001
328	0.499806	0.5	0.500001
329	0.499825	0.5	0.500001
330	0.499852	0.5	0.500001
331	0.499887	0.5	0.500001
332	0.499927	0.5	0.500001
333	0.49997	0.5	0.500001
334	0.500016	0.5	0.500001
335	0.500063	0.5	0.500001
336	0.500109	0.5	0.500001
337	0.500152	0.5	0.500001
338	0.500188	0.5	0.500001
339	0.500216	0.5	0.500001
340	0.500235	0.5	0.500001
341	0.500244	0.5	0.500001
342	0.500243	0.5	0.500001
343	0.50023	0.5	0.500001
344	0.500208	0.5	0.500001
345	0.500179	0.5	0.500001
346	0.500144	0.5	0.500001
347	0.500106	0.5	0.500001

Sigue en la página siguiente.





Time	FUZZY	PID	PID autotuning
348	0.500065	0.5	0.500001
349	0.500023	0.5	0.500001
350	0.499979	0.5	0.500001
351	0.499935	0.5	0.500001

Tabla A.2: Resultados de la simulación



Referencias Bibliográficas

- [1] SMAR. Fabricante de buses de campo y redes industriales. <http://www.smar.com/espanol/profibus.asp>, 2014.
- [2] ABB. Empresa proveedora de sistemas scada. <http://www05.abb.com>, 2014.
- [3] Timothy J Ross. *Fuzzy logic with engineering applications*. John Wiley & Sons, 2009.
- [4] A.B. Forouzan. *Data Communications & Networking (sie)*. McGraw-Hill Forouzan networking series. McGraw-Hill Education (India) Pvt Limited, 2006.
- [5] Katsuhiko Ogata. *Sistemas de control en tiempo discreto*. Pearson educación, 1996.
- [6] Thomas M Stout and Theodore J Williams. Pioneering work in the field of computer process control. *Annals of the History of Computing, IEEE*, 17(1):6–18, 1995.
- [7] Honeywell. Distributed control systems (dcs) maintenance, support and upgrade services from honeywel. <https://www.honeywellprocess.com/en-US/explore/services/distributed-control-systems/pages/default.aspx>, 2014.
- [8] Yokogawa. Corporate history. <http://www.yokogawa.com/pr/corporate/pr-corp-history-en.htm>, <http://www.yokogawa.com/dcs/centumvp/overview/dcs-vp-0201en.htm>, 2014.
- [9] Empresa de Ingeniería ELECTROMATICA. Generalidades sobre protocolos de comunicación industrial. <http://www.electromatica.cl/protocolos.html>, 2014.
- [10] A.G. Alexandrov and M.V. Palenov. Adaptive pid controllers: State of the art and development prospects. *Automation and Remote Control*, 75(2):188–199, 2014.
- [11] D. Pelusi. Genetic-neuro-fuzzy controllers for second order control systems. In *Fifth UKSim European Symposium on Computer Modeling and Simulation (EMS), 2011*, pages 12–17, Nov 2011.





- [12] M.A. Azman, A.A.M. Faudzi, M.O. Elnimair, O.F. Hikmat, K. Osman, and Chai Chang Kai. P-adaptive neuro-fuzzy and pd-fuzzy controller design for position control of a modified single acting pneumatic cylinder. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2013, pages 176–181, July 2013.
- [13] Muhammad Saleheen Aftab and Muhammad Bilal Kadri. Design of fuzzy logic based level controller for surge tank system. In *3rd International Conference on Computer, Control & Communication (IC4)*, 2013, pages 1–4. IEEE, 2013.
- [14] Yan Zhao. Research on application of fuzzy pid controller in two-container water tank system control. In *International Conference on Machine Vision and Human-Machine Interface (MVHI)*, 2010, pages 679–682. IEEE, 2010.
- [15] Hsiao-Ping Wang. Design of fast fuzzy controller and its application on position control of dc motor. In *International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 2011, pages 4902–4905. IEEE, 2011.
- [16] Abdallah A Ahmed, Yuanqing Xia, and Bo Liu. Design and simulation of fuzzy controller based (ipm) converter fed dc motors. In *International Conference on Control (CONTROL)*, 2012 UKACC, pages 217–222. IEEE, 2012.
- [17] P Thepsatorn, A Numsomran, V Tipsuwanporn, and T Teanthong. Dc motor speed control using fuzzy logic based on labview. In *International Joint Conference SICE-ICASE*, 2006., pages 3617–3620. IEEE, 2006.
- [18] Qingchun Li and Deyao Shen. A new incremental fuzzy pd+ fuzzy id fuzzy controller. In *ICIE'09. WASE International Conference on Information Engineering*, 2009., volume 1, pages 615–619. IEEE, 2009.
- [19] PM Meshram and Rohit G Kanojiya. Tuning of pid controller using ziegler-nichols method for speed control of dc motor. In *International Conference on Advances in Engineering, Science and Management (ICAESM)*, 2012, pages 117–122. IEEE, 2012.
- [20] Jian Lin, Jiakai Huang, and Gui Chen. Design of de-tuning pid controller for linear servo system. In *IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, 2012, pages 437–440, Oct 2012.
- [21] K. ChandraSekhar and K. Vaisakh. An intelligent technique based on code algorithm for determination of optimum gain values of pid controller in an agc system. In *International Conference on Intelligent Systems and Signal Processing (ISSP)*, 2013, pages 34–41, March 2013.
- [22] Hyeong-Pyo Hong, Suk-Joon Park, Sang-Joon Han, Kyeong-Young Cho, Young-Chul Lim, Jong-Kun Park, and Tae-Gon Kim. A design of auto-tuning pid controller using fuzzy logic. In *Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control.*, pages 971–976 vol.2, Nov 1992.



- [23] T. Sangyam, P. Laohapiengsak, W. Chongcharoen, and I. Nilkhamhang. Path tracking of uav using self-tuning pid controller based on fuzzy logic. In *Proceedings of SICE Annual Conference 2010*,, pages 1265–1269, Aug 2010.
- [24] N. Wahid and N. Hassan. Self-tuning fuzzy pid controller design for aircraft pitch control. In *Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS), 2012*, pages 19–24, Feb 2012.
- [25] R. Arun Kumar and J.L. Febin Daya. A novel self - tuning fuzzy based pid controller for speed control of induction motor drive. In *International Conference on Control Communication and Computing (ICCC), 2013*, pages 62–67, Dec 2013.
- [26] RituRani (Maity) and RajaniK. Mudi. Fuzzy self-tuning of conventional pid controller for high-order processes. In Suresh Chandra Satapathy, Siba K Ud-gata, and Bhabendra Narayan Biswal, editors, *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013*, volume 247 of *Advances in Intelligent Systems and Computing*, pages 41–48. Springer International Publishing, 2014.
- [27] N. Kanagaraj, P. Sivashanmugam, and S. Paramasivam. Fuzzy logic based supervisory control scheme for real-time pressure regulation. *Control and Intelligent Systems*, 37(3):160 – 169, 2009.
- [28] Chaohua Ao. and Jianchao Bi. Parameter auto-tuning method based on self-learning algorithm. *Journal of Computers*, 7(9):2168 – 2175, 2012.
- [29] Thanana Nuchkrua and Thananchai Leephakpreeda. Fuzzy self-tuning {PID} control of hydrogen-driven pneumatic artificial muscle actuator. *Journal of Bionic Engineering*, 10(3):329 – 340, 2013.
- [30] Katsuhiko Ogata. *Ingeniería de control moderna*. Prentice-Hall Pearson Education, S.A, 2003.
- [31] Benjamin C Kuo. *Sistemas de control automático*. Pearson Educación, 1996.
- [32] Richard C Dorf, Robert H Bishop, Sebastián Dormido Canto, Raquel Dormido Canto, and Sebastián Dormido. *Sistemas de control moderno*. Pearson Prentice Hall, 2005.
- [33] Madan Gopal. *Digital control engineering*. New Age International, 1988.
- [34] Lotfi A Zadeh. Fuzzy logic= computing with words. *Fuzzy Systems, IEEE Transactions on*, 4(2):103–111, 1996.
- [35] SangChul Ahn, Yong Ho Kim, and Wook Hyun Kwon. Design of a fuzzy logic controller module for a loop controller. In *International Conference on Industrial Technology, 1996.(ICIT'96), Proceedings of The IEEE*, pages 598–602. IEEE, 1996.



- [36] Jan Jantzen. *Foundations of Fuzzy Control: A Practical Approach*. John Wiley & Sons, 2013.
- [37] George J Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice Hall New Jersey, 1995.
- [38] Zdenko Kovacic and Stjepan Bogdan. *Fuzzy controller design: theory and applications*. CRC press, 2005.
- [39] Chuen-Chien Lee. Fuzzy logic in control systems: fuzzy logic controller. i. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):404–418, 1990.
- [40] Kevin M Passino, Stephen Yurkovich, and Michael Reinfrank. *Fuzzy control*, volume 42. Citeseer, 1998.
- [41] James F Kurose and Keith W Ross. *Computer networking*. Pearson Education, 2012.
- [42] Andrew S Tanenbaum. *Redes de computadoras*. Pearson educación, 2003.
- [43] John H Lilly. *Fuzzy control and identification*. John Wiley & Sons, 2010.
- [44] Michail Petrov, Ivan Ganchev, and Albena Taneva. Fuzzy pid control of nonlinear plants. In *First International IEEE Symposium Intelligent Systems, 2002. Proceedings. 2002*, volume 1, pages 30–35. IEEE, 2002.
- [45] Kwang Hyung Lee. *First course on fuzzy theory and applications*, volume 27. Springer, 2006.
- [46] Hongying Wang and Wei Zhou. Design and research on fuzzy controller in digital speed control system. In David Jin and Sally Lin, editors, *Advances in Mechanical and Electronic Engineering*, volume 177 of *Lecture Notes in Electrical Engineering*, pages 171–176. Springer Berlin Heidelberg, 2012.
- [47] DavoodMohammadi Souran, SeyedHamidreza Abbasi, and Faridoon Shabania. Comparative study between tanks water level control using pid and fuzzy logic controller. In Valentina Emilia Balas, János Fodor, Annamária R. Várkonyi-Kóczy, Jozsef Dombi, and Lakhmi C. Jain, editors, *Soft Computing Applications*, volume 195 of *Advances in Intelligent Systems and Computing*, pages 141–153. Springer Berlin Heidelberg, 2013.
- [48] Aidan O’Dwyer. Pi and pid controller tuning rules: an overview and personal perspective. 2006.
- [49] JG Ziegler and NB Nichols. Optimum settings for automatic controllers. *trans. ASME*, 64(11), 1942.



- [50] H.B. Kazemian. Intelligent fuzzy pid controller. In Robert Lowen and Alain Verschoren, editors, *Foundations of Generic Optimization*, volume 24 of *Mathematical Modelling: Theory and Applications*, pages 241–260. Springer Netherlands, 2008.
- [51] Morteza Esfandyari, MohammadAli Fanaei, and Hadi Zohreie. Adaptive fuzzy tuning of pid controllers. *Neural Computing and Applications*, 23(1):19–28, 2013.