

**CONTROL DE POSICION UTILIZANDO LOGICA DIFUSA EN TIEMPO REAL DE
UN SISTEMA NO LINEAL**

ENRIQUE ALFREDO DIAZGRANADOS GARRIDO

JHON JAIRO MONTALVO RESTREPO

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

CARTAGENA DE INDIAS, D.T. Y C.

JUNIO DE 2005

**CONTROL DE POSICION UTILIZANDO LOGICA DIFUSA EN TIEMPO REAL DE
UN SISTEMA NO LINEAL**

ENRIQUE ALFREDO DIAZGRANADOS GARRIDO

JHON JAIRO MONTALVO RESTREPO

**Trabajo de grado presentado como requisito para optar al título
de Ingenieros Electrónicos.**

**Director
JUAN CARLOS MARTÍNEZ SANTOS
Ingeniero Electrónico**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
CARTAGENA DE INDIAS, D.T. Y C.**

JUNIO 2005

Cartagena, 8 de Junio de 2005

Señores

COMITÉ DE PROYECTOS DE GRADO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

La ciudad.

Distinguidos señores:

Por medio de la presente me permito comunicarles que la **tesis** titulada **"CONTROL DE POSICION UTILIZANDO LOGICA DIFUSA EN TIEMPO REAL DE UN SISTEMA NO LINEAL"** ha sido desarrollada de acuerdo con los objetivos establecidos.

Como director considero que el trabajo es satisfactorio y cumple con todos los requerimientos necesarios para ser presentado por sus autores para su evaluación. Por tal motivo hago entrega formal del proyecto.

Cordialmente,

Ing. JUAN CARLOS MARTINEZ

Director del proyecto

Cartagena, 8 de Junio de 2005

Señores

COMITÉ DE PROYECTOS DE GRADO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

La ciudad.

Distinguidos señores:

La presente tiene como objetivo presentar a su consideración para estudio y aprobación, la **tesis** titulada "**CONTROL DE POSICION UTILIZANDO LOGICA DIFUSA EN TIEMPO REAL DE UN SISTEMA NO LINEAL**", como requisito para optar el título de Ingeniero Electrónico.

Cordialmente,

Enrique Alfredo DiazGranados Garrido

Jhon Jairo Montalvo Restrepo

Cartagena, 5 de Julio de 2005

Señores

BLIBLIOTECA LUIS ENRIQUE BORJA BARON

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

La ciudad.

Distinguidos señores:

Nosotros, Enrique Alfredo DiazGranados Garrido, identificado con C.C. 9.100.863 de Cartagena y Jhon Jairo Montalvo Restrepo identificado con C.C. 73.181.344 de Cartagena, autorizamos la puesta en linea de la tesis de grado titulada **"CONTROL DE POSICION UTILIZANDO LOGICA DIFUSA EN TIEMPO REAL DE UN SISTEMA NO LINEAL"**, en la pagina web de la biblioteca de la Universidad Tecnológica de Bolívar.

Cordialmente,

Enrique Alfredo DiazGranados Garrido

Jhon Jairo Montalvo Restrepo

Nota de aceptación

Presidente del Jurado

Jurado

Jurado

Ciudad y fecha (día, mes, año): _____

Dedicatoria

Dedico esta tesis de grado a toda mi familia que en todo momento me dieron su inmenso apoyo y enseñanzas, especialmente a mis padres y hermanas que aunque no lo crean siempre los llevo presentes en mi corazón. Esto es un pedacito más de mi proyecto de vida que lo dedico también a mi abuela Lina que sin ella mucho de esto no hubiese pasado.

Enrique Alfredo Díaz Granados Garrido

Dedicatoria

*A mi familia por su apoyo y perseverancia.
A mi abuelo; hombre recto y lleno de
sabiduría, por sus valiosos consejos y
enseñanzas.
A mi pueblo **Guaranda**.*

*“Si lo puedes imaginar
lo puedes lograr”. Einstein...*

Jhon Jairo Montalvo Restrepo

CONTENIDO

	Pág.
INTRODUCCIÓN	17
OBJETIVOS	19
1. ¿PORQUE EL SISTEMA ES NO LINEAL?	20
1.1. INTRODUCCION A LOS SISTEMAS DE CONTROL NO LINEALES	20
1.2. PLANTEAMIENTO DEL PROBLEMA	22
1.3. DESCRIPCION DEL PROBLEMA	22
1.4. LEYES QUE RIGEN EL COMPORTAMIENTO DE LA ESFERA	23
2. DISEÑO Y MONTAJE DE LA PLANTA	27
2.1. MOTOR DC LINEAL	28
2.2. SENSOR INFRARROJO	29
2.3. TUBO PARA TUNEL DE VIENTO	30
2.4. ESFERA (BOLITA DE PING PONG)	30
2.5. ACTUADOR O ETAPA DE POTENCIA	31
2.6. TARJETA DE ADQUISICION DE DATOS	32
2.7. MONTAJE COMPLETO	33
3. MATLAB COMO PLATAFORMA PARA EL CONTROL EN TIEMPO REAL	34
3.1. TOOLBOX DE FUZZY LOGIC	34
3.1.1. CONSTRUCCIÓN DEL FIS	34
3.1.2. EDITOR DE FIS	34

3.1.3.	EDITOR DE LAS FUNCIONES DE MEMBRESÍA	34
3.1.4.	EDITOR DE REGLAS	34
3.1.5.	VISOR DE REGLAS	35
3.1.6.	VISOR DE SUPERFICIE	35
3.2.	REAL-TIME WORKSHOP	35
3.2.1.	ACCESO DEL RTW AL VISUAL C++ 6.0	36
3.2.2.	RELACIÓN DEL RTW CON OTROS ELEMENTOS	36
3.2.3.	PLATAFORMAS QUE TRABAJAN EN MODO EXTERNO	37
3.2.4.	ESTRUCTURA DE PROGRAMACIÓN	37
3.3.	REAL-TIME WINDOWS TARGET	38
3.3.1.	INSTALACIÓN DE VISUAL C++ DESDE LA PANTALLA PRINCIPAL DE MATLAB	38
3.3.2.	KERNEL	39
3.3.3.	ELEMENTOS BASICOS DEL RTWT	40
4.	CONTROL EN TIEMPO REAL UTILIZANDO LOGICA DIFUSA	41
4.1.	DISEÑO DEL CONTROLADOR CON EL TOOLBOX DE FUZZY LOGIC	41
4.1.1.	CONJUNTOS DIFUSOS	41
4.1.2.	FUNCIONES DE INCLUSIÓN	43
4.1.3.	VARIABLE LINGÜÍSTICA	44
4.1.4.	PARTICIONES DIFUSAS	44
4.1.5.	BASE DE REGLAS	46
4.2.	FUNCIONAMIENTO EN TIEMPO REAL	49
4.2.1.	MONTAJE COMPLETO EN SIMULINK	49
4.2.2.	RUN, START Y TOMA DE DATOS	51

4.2.3.	GRAFICAS Y ANALISIS DE RESULTADOS	53
5.	CONTROL EN TIEMPO REAL UTILIZANDO PID	54
5.1.	CONTROL PID	54
5.2.	DISEÑO DEL CONTROLADOR PID	54
5.3.	METODOS DE SINTONIZACION	55
5.3.1.	MÉTODO DEL RELAY	55
5.3.1.1.	IMPLEMENTACIÓN EN TIEMPO REAL	55
5.3.1.2.	PARÁMETROS DEL RELAY	56
5.3.1.3.	PROCEDIMIENTO DE SINTONIZACIÓN	57
5.3.2.	MÉTODO DE RESPUESTA EN FRECUENCIA DE ZIEGLER– NICHOLS	59
5.3.2.1.	PROCEDIMIENTO DE SINTONIZACIÓN	60
5.3.3.	COMPARACIÓN DE LOS MÉTODOS DE SINTONIZACIÓN	61
5.4.	OBTENCION DE LOS PARAMETROS PID	62
5.4.1.	PARÁMETROS DEL CONTROLADOR	63
5.5.	FUNCIONAMIENTO EN TIEMPO REAL	64
5.5.1.	MONTAJE COMPLETO EN SIMULINK	64
5.5.2.	GRAFICAS Y ANALISIS DE LOS RESULTADOS	65
6.	ANALISIS COMPARATIVO DE LOS RESULTADOS	68
6.1.	RESPUESTA DESCENDENTE.	69
6.2.	RESPUESTA ASCENDENTE.	72
	CONCLUSIONES Y RECOMENDACIONES	74
	BIBLIOGRAFÍA	76
	ANEXOS	

LISTA DE FIGURAS

		Pag.
Figura 1.	Comportamiento de una esfera en el interior de un gas.	23
Figura 2.	Leyes que actúan sobre la esfera.	23
Figura 3.	Movimiento de la esfera en el eje horizontal.	24
Figura 4.	Fuerzas resistivas del movimiento.	25
Figura 5.	Cambio de posición de la esfera	25
Figura 6.	Curva característica del motor	27
Figura 7.	Motor DC con aspas.	29
Figura 8.	Sensor SHARP Modelo GP2Y0A02YK.	29
Figura 9.	Tubo de acrílico	30
Figura 10.	Esfera de Ping Pong	31
Figura 11.	Circuito Actuador	32
Figura 12.	Tarjeta de adquisición de datos NI 6024E	33
Figura 13.	Montaje completo de la planta	33
Figura 14a.	Editor de las funciones de membresía	35
Figura 14b.	Editor de FIS	35
Figura 14c.	Editor de regla	35
Figura 14d.	Visor de regla	35

Figura 14e.	Visor de superficie	35
Figura 15.	Relación del RTW con otros elementos	37
Figura 16.	Instalación del compilador de lenguaje C	39
Figura 17.	Elementos básicos del RTWT	40
Figura 18.	Variables de entrada y salida en el FIS editor de MATLAB	42
Figura 19.	Función de pertenencia tipo trapezoidal	43
Figura 20.	Función de pertenencia tipo triangular	44
Figura 21.	Particiones y funciones de inclusión de <i>ERROR</i> en el funtion editor de MATLAB	45
Figura 22.	Particiones y funciones de inclusión de <i>DERROR</i> en el funtion editor de MATLAB.	45
Figura 23.	Particiones y funciones de inclusión de <i>Vsal</i> en el funtion editor de MATLAB	46
Figura 24.	Bases de reglas para el control de posición en el Ruler Editor de MATLAB	47
Figura 25.	Características de nuestro controlador en el Rule visor de MATLAB	48
Figura 26.	Superficie de los conjuntos en base a las reglas en el Rule visor de MATLAB	48
Figura 27.	Montaje completo de la planta con Fuzzy en un modelo de Simulink	49
Figura 28.	Parámetros del bloque de entrada y salida análogas	50
Figura 29.	Linealización y conversión de la información del sensor desde Simulink	50
Figura 30.	Configuración de los parámetros de simulación	51
Figura 31.	Configuración de los parámetros de del External mode	52
Figura 32.	Respuesta del sistema con el controlador Fuzzy	53

Figura 33.	Diagrama de bloque del controlador PID	54
Figura 34.	Bloque del Relay	55
Figura 35.	Bloque de los parámetros del relay	56
Figura 36.	Diagrama de bloque para la conexión del relay	57
Figura 37.	Respuestas del sistema para calcular parámetros del relay	58
Figura 38.	Respuesta del sistema ante un valor K_u conocido	60
Figura 39.	Respuesta del sistema con método de Ziegler-Nichols	61
Figura 40.	Respuesta del sistema con método del Relay	62
Figura 41.	Bloque del controlador PID	63
Figura 42.	Parámetros del bloque de PID	63
Figura 43.	Montaje completo de la planta con PID en un modelo de Simulink	64
Figura 44.	Respuesta del sistema para Set Point de 100 cm.	65
Figura 45.	Respuesta del sistema para Set Point de 20 cm.	66
Figura 46.	Respuesta del sistema para un cambio de referencia de 0 a 50 cm.	66
Figura 47.	Respuesta del sistema para Set Point 80 cm.	67
Figura 48.	Respuesta descendente de los controladores Fuzzy y PID	69
Figura 49.	Respuesta ascendente de los controladores Fuzzy y PID	72

LISTA DE TABLAS

		Pag.
Tabla 1.	Base de reglas para el control de posición	47
Tabla 2.	Calculo de los parámetros PID	59
Tabla 3.	Comparación entre Fuzzy y PID de bajada	71
Tabla 4.	Comparación entre Fuzzy y PID de subida	73

LISTA DE ANEXOS

		Pag.
Anexo A.	Programa para calcular la ecuación del sensor.	78
Anexo B	Diseño de la etapa de potencia	83
Anexo B.	Hoja característica del sensor SHARP modelo GP2Y0A02YK.	85
Anexo C.	Hoja característica del amplificador de potencia TDA2040A	87
Anexo D	Hoja característica del amplificador operacional LM358	102

INTRODUCCION

En algunas aplicaciones de control automático, no se puede recurrir a los conceptos tradicionales cuando se manejan procesos muy complejos que presentan alinealidades, perturbaciones e inestabilidades, convirtiéndose en un dolor de cabeza para muchos investigadores. Estos procesos reales se vuelven incontrolables debido a que los eventos futuros son difíciles de predecir y manejar. Lo anterior ha generado problemas y pérdidas a muchas personas o empresas por desconocer el comportamiento de un sistema no lineal, dificultando el desarrollo de investigaciones relacionadas con el tema por falta de conocimiento sobre sistemas complejos.

Este trabajo de grado pretende mostrar cómo, una técnica de control automático inteligente, más precisamente la **Lógica Difusa**, es más óptima que la técnica de control tradicional **PID** para controlar un sistema no lineal. El estudio se realiza sobre un prototipo en el que se quiere controlar, en tiempo real, la posición de una esfera dentro de un tubo transparente, manipulando de forma automatizada un flujo de aire. Para conseguir tal propósito fué necesario recurrir a temas propios de control automático y electrónica, como señales y sistemas, modelamiento matemático, identificación de sistemas, circuitos, control digital, software, programación y en general los temas concernientes al diseño de controladores en tiempo real.

El estudio, diseño y desarrollo de esta aplicación se realizó con controladores tipo Difuso o “fuzzy logic” y tipo *PID* (*proporcional, integral, derivativo*) implementados en un computador, utilizando como plataforma de trabajo el software MatLab versión 6.5. Se plantean además los procedimientos para la utilización de este software en tiempo real.

Por último, después de analizar las respuestas observadas en ambos tipos de controladores ante cambios de referencias y realizar las debidas comparaciones, se pudo llegar a conclusiones y a establecer recomendaciones para el manejo de procesos con cada tipo de controlador.

OBJETIVOS

Objetivo General

Controlar un sistema no lineal en tiempo real, aplicado al control de altura de una esfera suspendida en un flujo de aire utilizando lógica difusa, de manera que el sistema de control funcione adecuadamente a pesar de variaciones no lineales, en el modelo del sistema.

Objetivos Específicos

- Determinar los problemas para controlar un sistema no lineal, en general, y de control en tiempo real de la altura de una esfera suspendida en un flujo de aire, en particular.
- Diseñar y construir un controlador con Lógica Difusa y un controlador PID para un sistema no lineal, en tiempo real.
- Comparar el comportamiento de un sistema no lineal controlado con PID, con el comportamiento de uno igual, pero controlado con Lógica Difusa.
- Identificar los factores que hacen que el control de un sistema no lineal, sea poco óptimo con los métodos tradicionales.
- Documentar las prácticas, métodos, modelos, resultados, análisis y diseño.
- Analizar los resultados obtenidos en este proyecto para establecer conclusiones y recomendaciones.

1. ¿PORQUE EL SISTEMA ES NO LINEAL?

En este capítulo, se trata de explicar los sistemas no lineales con una introducción acerca de los mismos; y luego en base a la naturaleza del fenómeno que se quiere controlar, se hace el planteamiento del problema, la descripción del mismo y las leyes que lo rigen.

1.1. INTRODUCCION A LOS SISTEMAS DE CONTROL NO LINEALES

El uso moderno de la palabra *sistema* tiene muchos significados. Así que comencemos por definir lo que queremos decir cuando la utilicemos en esta tesis; primero, en forma abstracta y luego, de manera más específica, en relación con la literatura científica.

Definición 1: Un *sistema* es un conjunto, arreglo o colección de cosas unidas o relacionadas de tal manera que forman una entidad o un todo.

Definición 2: Un *sistema* es un ordenamiento de componentes físicos, unidos o relacionados de tal manera que forman y/o interactúan como una unidad completa.

Definamos dos términos: *entrada* y *salida*, los cuales nos ayudarán a identificar, delinear o definir un sistema.

La entrada: Es el estímulo, la excitación o el mandato aplicado a un sistema, generalmente desde una fuente externa de energía, usualmente para producir una respuesta específica del sistema.

La salida: Es la respuesta real que se obtiene de una planta o sistema. Puede ser o no igual a la respuesta implícita especificada en la entrada.

Las entradas y salidas pueden tener muchas formas diferentes. Las entradas, por ejemplo, pueden ser variables físicas o cantidades más abstractas, tales como valores de referencia, de ajuste o deseados para la salida del sistema.

Los sistemas pueden tener más de una entrada o de una salida. A menudo, todas las entradas y salidas están bien definidas en la descripción del sistema, pero algunas veces no, por presentar alinealidades, como en el caso que se trata en esta tesis.

La linealidad se plantea como una propiedad de los *sistemas generales*, con una variable independiente, el tiempo t . Definidos anteriormente, los conceptos de sistema, entrada y salida, la linealidad se puede definir como:

Linealidad: “Si todas las condiciones iniciales en un sistema son cero, esto es, si el sistema está completamente en reposo, entonces el sistema es lineal si tiene la siguiente propiedad:

Si una entrada $u_1(t)$ produce una salida $y_1(t)$ y una entrada $u_2(t)$ produce una salida $y_2(t)$ entonces $c_1u_1(t) + c_2u_2(t)$ produce una salida $c_1y_1(t) + c_2y_2(t)$ para todos los pares de entrada de constantes c_1 y c_2 .

Cualquier sistema que no satisfaga la anterior definición de linealidad es no lineal.

La mayor dificultad con los sistemas no lineales es que las soluciones matemáticas o de forma cerrada solo son posibles para muy pocos casos especiales, y estos por lo general, no son de interés práctico en el análisis o el diseño de control.

1.2. PLANTEAMIENTO DEL PROBLEMA

¿Qué pasa cuando los procesos dependen de muchos factores, y presentan perturbaciones y variables de naturaleza aleatoria? Controlar un sistema no lineal no es tarea fácil, sobre todo cuando es difícil obtener un modelo matemático capaz de linealizarlo suficientemente como para que pueda ser controlado por algoritmos convencionales como es el caso de un PID.

Como el proceso es completamente no lineal y no se dispone de un modelo óptimo que tenga en cuenta todas las perturbaciones y variables que intervienen en el mismo, el control mediante un algoritmo PID resulta insuficiente. Por ello es necesario el uso de otra técnica de control más novedosa y eficiente. Los controles basados en métodos de Inteligencia artificial tienen gran aplicación en la actualidad, destacándose entre ellos las Redes Neuronales y la *Lógica Difusa*.

1.3. DESCRIPCION DEL PROBLEMA

La naturaleza de esta planta o sistema se puede explicar basándose en el modelo de un cuerpo que se desplaza en el interior de un gas (ver Figura 1). Partiendo de dicho modelo se tiene que, en los puntos A y C, la velocidad del flujo de aire es cero mientras que en los puntos B y D las líneas de flujo se estrechan y por tanto la velocidad del aire es mayor que la de la corriente de flujo no perturbada.

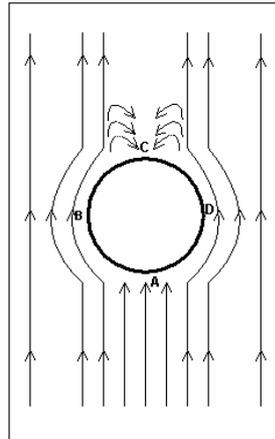


Figura 1: Comportamiento de una esfera en el interior de un gas.

1.4. LEYES QUE RIGEN EL COMPORTAMIENTO DE LA ESFERA¹

Sobre la esfera actúan cuatro fuerzas fundamentales (ver Figura 2): la fuerza de fricción debida a la Ley de Stokes, la de gravedad hacia abajo, así como las fuerzas de empuje y de sustentación hacia arriba.

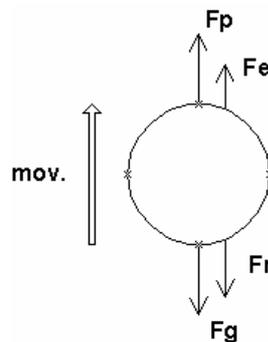


Figura 2: Leyes que actúan sobre la esfera.

Donde F_g es la fuerza de gravedad, F_p la fuerza de la columna de aire sobre la esfera, F_e la fuerza de empuje y F_r la fuerza de resistencia del aire.

¹ Tomado del Paper “Control de posición de un Sistema no Lineal”
http://csd.newcastle.edu.au/control/SpanishPages/clase_slides.com

Aplicándose la segunda Ley de Newton a este sistema se tiene:

$$\sum Fy = ma \quad (1)$$

$$Fp - Fr = ma + Fg - Fe \quad (2)$$

Sustituyendo en la ecuación (2):

$$\frac{1}{3} \frac{\pi r^2 V_0 M_0 \bar{v}^2}{N^2} - 6\pi\eta r v = ma - m_g g + mg \quad (3)$$

Donde m_g es la masa del gas desplazado por la esfera: $m_g = \rho_{aire} V$, donde ρ es la densidad del aire.

De la fórmula anterior se deduce que la posición de la esfera depende de la velocidad de la columna de aire. Esta velocidad es suministrada al sistema a través de un motor de corriente directa.

Pero, el movimiento de la esfera en el eje X es descontrolado, debido a que al pegarse o acercarse la esfera a uno de los lados del tubo (ver Figura 3) el lado más próximo, punto B, recibe menos presión que el más alejado, punto D.

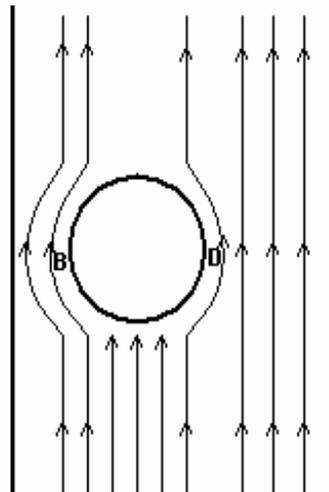


Figura 3: Movimiento de la esfera en el eje horizontal.

Esta variación en ambos lados de la esfera hace que en los puntos B y D surjan fuerzas resistivas de diferente magnitud (ver Figura 4)

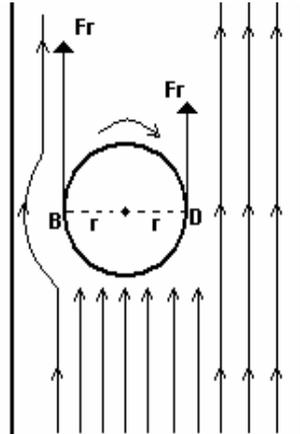


Figura 4: Fuerzas resistivas del movimiento.

Al ocurrir esto se produce un par de fuerzas desbalanceadas que provocan un torque, capaz de hacerla girar con una velocidad ω_r según la ecuación:

$$\sum M = I\omega_r \quad (4)$$

$M_B - M_D = I\omega_r$. Siendo $M_B = Fr_B * r$, $M_D = Fr_D * r$ e I el momento de inercia de la esfera.

De (4) se deduce que la masa de la esfera es inversamente proporcional a la velocidad de rotación. Al girar, parte de la energía potencial gravitatoria se convierte en energía cinética de rotación y otra parte en calor producto de la fricción, por lo que la esfera “cae” a una nueva posición (ver Figura 5).

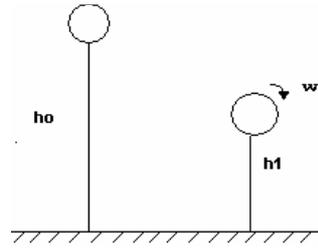


Figura 5: Cambio de posición de la esfera

Aplicando la ley de conservación de la energía:

$$mgh_0 = mgh_1 + \frac{1}{2}I\omega^2 + Q \quad (5)$$

Cuando la esfera se encuentra girando es muy difícil elevarla debido a que un pequeño aumento de la velocidad del motor provoca un incremento de la velocidad de rotación. Por esto, cuando el proceso cae en esta fase, es necesario suministrarle un mayor voltaje al motor para lograr más velocidad del flujo de aire y de esta forma suplir el déficit de energía producto de la rotación.

Como se puede apreciar, el movimiento de la esfera es bastante complejo y en el mismo concurren a la vez diferentes factores. Un modelo exacto de este proceso es muy difícil de obtener y nunca abarcaría todas las variables que intervienen en el mismo, ni sus comportamientos, muchas veces aleatorios.

Lo que se pretende en esta tesis es demostrar que utilizando la técnica de control *fuzzy logic*, se obtienen resultados más óptimos (en cuanto al tiempo de respuesta, tiempo de establecimiento, oscilaciones, etc.) que utilizando la técnica convencional *PID*, para el control satisfactorio de un sistema no lineal, y en tiempo real (ambas técnicas) utilizando, en este caso, como plataforma el software MATLAB.

2. DISEÑO Y MONTAJE DE LA PLANTA

En este capítulo se describe y se muestra cada uno de los componentes de la planta, así como el diseño y montaje completo de la misma al final del mismo.

El prototipo de planta para controlar, en tiempo real, la posición de una esfera (bola de ping pong) con un flujo de aire, consiste en dos partes diferentes pero que siempre están ligadas la una con la otra. Una es el “cuerpo” de la planta y la otra es el “cerebro” que sería la parte de control:

- El cuerpo de la planta consta de un tubo de acrílico transparente abierto por ambos extremos para dejar que el flujo de aire circule libremente; en el extremo inferior del mismo se encuentra un motor DC, con aspas para suministrar el flujo que mueve o impulsa la esfera a lo largo del tubo y en el extremo superior se tiene un sensor infrarrojo que proporciona, en todo momento, la posición de la esfera también a lo largo del tubo.
- Para la parte del control automático se tiene que la salida de información, sobre la altura de la esfera, suministrada por el sensor, es una entrada para el computador, más la requerida por el usuario final que es quien desea que la esfera alcance una altura específica (*set point*, en base al cual se realiza el control). Los datos anteriores son procesados por el computador, para realizar la(s) debida(s) acción(es) de control y producir luego una salida desde este último hacia el motor. La comunicación con el sensor y con el motor se realiza por medio de una tarjeta de adquisición de datos, conectada a la *mother board* del computador. Sobre la plataforma del *software* MATLAB, el usuario

introduce el *set point* deseado y se ejecutan los controladores de LÓGICA DIFUSA y PID en tiempo real.

2.1. MOTOR DC LINEAL

El motor utilizado en el sistema es un motor que tiene la característica de ser lineal, tal como se aprecia en su curva característica (Ver Figura 6).

La ecuación del motor es:

$$M = 192.5765 \cdot V + 597 \quad (6)$$

Donde: M es el número de Rpm, V es voltaje aplicado al motor.

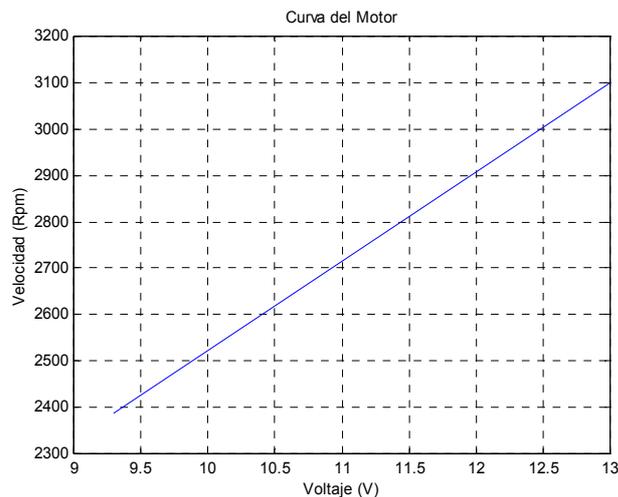


Figura 6: Curva característica del motor

Esta gráfica muestra que el motor es lineal, ya que las revoluciones son directamente proporcionales al voltaje aplicado, es decir a mayor voltaje, mayor revoluciones por minuto y a menor voltaje aplicado menor Rpm.

El rango de operación del motor está entre 9.3V y 13V. Con el voltaje de 9.3V el motor gira, pero no produce la suficiente presión de aire para que este impulse la esfera (Ver Figura 7).

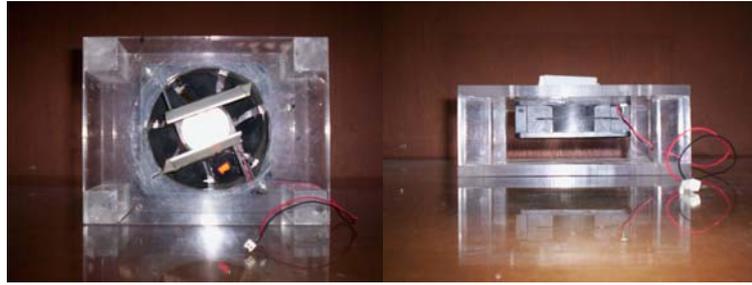


Figura 7: Motor DC con aspas.

2.2. SENSOR INFRARROJO

Se adquirió y trabajó con el sensor infrarrojo para medición de distancias variables marca SHARP Modelo GP2Y0A02YK (Ver Figura 8), el cual posee la característica más importante para esta aplicación como es la de detectar donde está la esfera en todo momento y a todo lo largo del tubo.



Figura 8: Sensor SHARP Modelo GP2Y0A02YK.

Rangos de salida: Mínimo 20 cm. y 0,45 voltios, hasta máximo 150 cm. y 2,5 voltios aproximadamente. La alimentación recomendada es entre 4,5V y 5,5V DC. Su ubicación es en la parte superior del tubo, y para poder censar correctamente la posición variable de cualquier objeto, en este caso la esfera, debe tener una distancia mínima de 20 cm., según recomendaciones del fabricante, por lo cual se adecuó un tope en la parte superior-interna del tubo a aproximadamente 22 cm. para dar un margen de seguridad en las medidas. Las características específicas y el funcionamiento del sensor se describen en el Anexo A.

2.3. TUBO PARA TUNEL DE VIENTO

El tubo está fabricado en acrílico transparente, para facilitar la observación del comportamiento de la esfera y el respectivo control de su posición. La longitud del tubo es de 125cm., con un diámetro interno de 4,8cm. y externo de 5,2cm. Sus superficies tanto externa como interna son completamente lisas, para que la esfera y el flujo aire puedan circular con facilidad y así reducir la fricción.

En la parte superior del tubo está sostenido el sensor y en la parte inferior el motor-ventilador y la esfera en la parte interna. (Ver Figura 9).

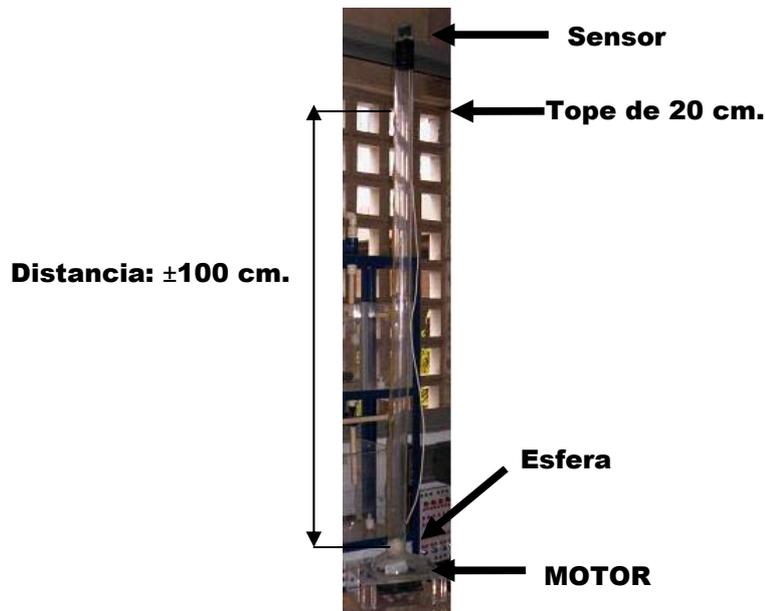


Figura 9: Tubo de acrílico.

2.4. ESFERA (BOLITA DE PING PONG)

Para demostrar esta aplicación, de control de posición, se escogió una esfera (bolita de ping pong), que está hecha de una capa delgada de plástico, para no afectar el uso que se le da a la misma ya que como sabemos, se utiliza en una

disciplina deportiva específica de agilidad física y mental. Su superficie al igual que la del tubo es totalmente lisa, tiene 4.5cm de diámetro y es de color blanco. Con esta esfera, se quiere alcanzar una altura deseada, dependiendo de la referencia que se le dé al sistema (Ver Figura 10.).



Figura 10: Esfera de ping pong.

2.5. ACTUADOR O ETAPA DE POTENCIA

La etapa de potencia de la planta convierte el voltaje (4 - 7V DC) suministrado por el controlador (Fuzzy o PID), a través de la tarjeta de adquisición de datos. Esta potencia suministrada no es suficiente para que el motor trabaje y por lo tanto se debe amplificar la misma con el siguiente diseño de actuador.

CONDICIONES DE DISEÑO: La tarjeta de adquisición de datos conectada al computador proporciona un máximo de 10V DC y un mínimo de 0V. Por seguridad de la tarjeta misma y, para no sobrepasarnos en corriente, trabajamos con un voltaje máximo de 7V DC.

El motor que se alimenta tiene las siguientes características:

$V_{max}=13V$ DC $I_n=500mA$, donde: V_{max} , es el voltaje DC máximo que debe ser suministrado al motor, I_n , es corriente nominal para que el motor opere.

Con esta condición se tiene que la resistencia del motor es:

$$R_m = \frac{V_{\max}}{I_n} = \frac{13V}{500mA} \Rightarrow R_m = 26\Omega \quad (7)$$

La Potencia del motor es: $P_m = V_{\max} * I_n = 6.5W$ (8)

Con lo anterior se establecen las premisas de diseño (maximas) para la realización de los cálculos para el actuador, los cuales son:

$V_{in} = 7V$, $V_{out} = 13V$, $R_L = 26\Omega$, $I_L = 500mA$

El circuito general diseñado es (Ver Figura 11), el diseño y sus especificaciones se encuentra en Anexo B.

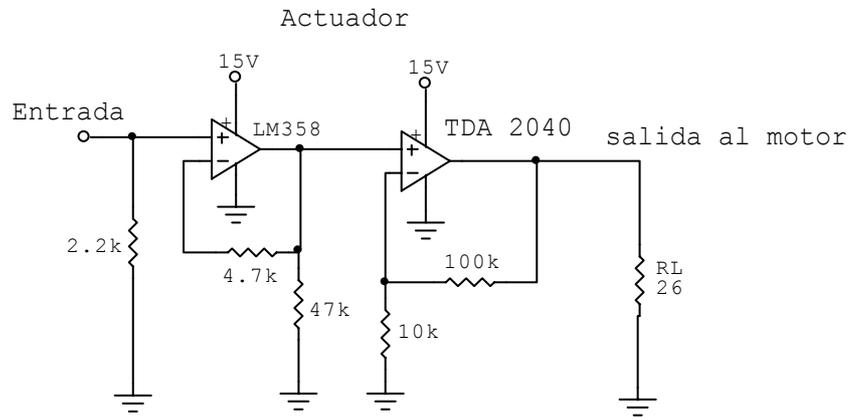


Figura 11: Circuito Actuador.

2.6. TARJETA DE ADQUISICION DE DATOS

Las tarjetas DAQ son tarjetas insertables que permiten la entrada y salida de datos del computador a otros aparatos, donde se conectan sensores y actuadores, para interactuar con el mundo real. Los datos que entran y salen pueden ser señales digitales o análogas o simplemente conteos de ocurrencias digitales, tanto de entradas como de salida. Para esta aplicación usamos la tarjeta de adquisición de datos NI 6024E de la *National Instruments* (ver Figura 12).



Figura 12: Tarjeta de adquisición de datos NI 6024E.

Las principales características de la tarjeta son: 16 entradas analógicas y 2 salidas analógicas, de las cuales solo usaremos una de cada una, es decir una entrada y una salida. Los rangos de voltaje entrada y salida, son de hasta ± 10 V.

Para más información consultar el PDF DAQ 6023E/6024E/6025E User Manual.

2.7. MONTAJE COMPLETO

Después de hacer una breve descripción de los elementos que integran la planta, a continuación se muestra en la figura, la interconexión de los mismos y la forma como el usuario interactúa en tiempo real con la misma. (Ver Figura 13)

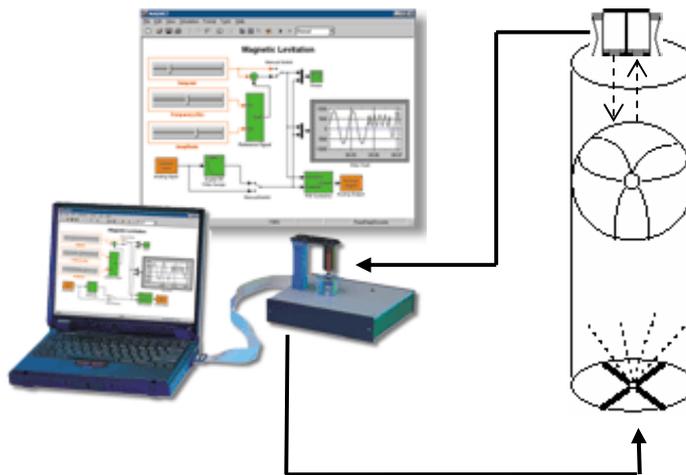


Figura 13: Montaje completo de la planta.

3. MATLAB COMO PLATAFORMA PARA EL CONTROL EN TIEMPO REAL

Este capítulo trata de la forma como se utiliza MATLAB 6.5, sus toolboxes y el Simulink, como plataformas para el montaje de las plantas con fuzzy logic y PID en tiempo real.

3.1. TOOLBOX DE FUZZY LOGIC

Usos del toolbox de *Fuzzy Logic*:

- Desarrollar y analizar sistemas de inferencia difusa.
- Proporcionar un bloque con el controlador difuso que se pueda utilizar en Simulink para modelar y para simular un sistema de control por lógica difusa.
- Desde Simulink, se puede generar un código C, para el uso de los sistemas que incluyen lógica difusa.

3.1.1. Construcción de FIS (Sistema de inferencia difusa):

El FIS interpreta los valores en la entrada y, basado en reglas definidas por el usuario, asigna valores a la salida. Usando los editores del GUI en el toolbox de *Fuzzy Logic*.

3.1.1.1. Editor de FIS: Información de carácter general que muestra sistema de inferencia difusa (Ver Figura 14a).

3.1.1.2. Editor de las funciones de membresía: Permite mostrar y corregir las funciones de membresía asociadas a las variables de la entrada y de salida del FIS (Ver Figura 14b).

3.1.1.3. Editor de reglas: Permite ver y corregir reglas difusas (Ver Figura 14c).

3.1.1.4. Visor de reglas: Permite ver el comportamiento detallado de un FIS para ayudar a diagnosticar el comportamiento de reglas específicas o a estudiar el efecto de cambiar variables de la entrada (Ver Figura 14d).

3.1.1.5. Visor de superficie: Genera una superficie tridimensional a partir de las variables de entrada y salida del FIS (Ver Figura 14e).

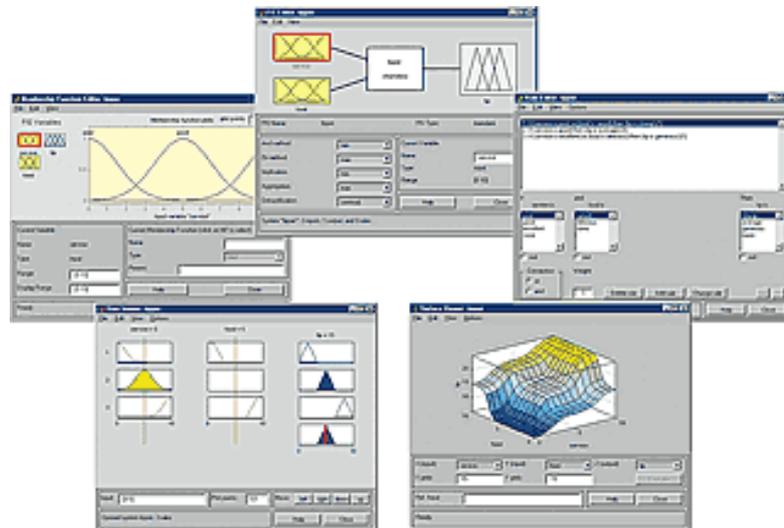


Figura 14: a) El editor de las funciones de membresía (izquierda superior). b) Editor de FIS (centro). c) Editor de regla (derecha superior). d) Visor la regla (izquierda inferior). e) Visor de Superficie (derecha inferior).

3.2. REAL-TIME WORKSHOP

El RTW (*Real-Time Workshop*) y el RTWT (*Real-Time Windows Target*) son herramientas de MATLAB que facilitan el control en tiempo real desde SIMULINK, acortando considerablemente el ciclo de desarrollo del software de control (prototipado rápido), permitiendo concretarse en el diseño.

El RTW permite generar código o un programa ejecutable (.exe, .dll) desde un diagrama de SIMULINK a diferentes plataformas: DOS, TORNADO, xPC, RTWT y otras.

En la mayoría de los casos, el diagrama de simulación de SIMULINK funciona en modo normal, como la interfaz de usuario (GUI). La diferencia consiste en que para poder trabajar en tiempo real se debe ejecutar SIMULINK en modo externo. SIMULINK funciona como la interfaz de usuario para monitoreo y ajuste de parámetros.

Para poder usar el RTW se necesita un compilador soportado para MATLAB como Visual C++ que es el escogido para trabajar en esta aplicación.

3.2.1. Acceso del RTW al Visual C++ 6.0:

Soporta sistemas de tiempo continuo, sistemas de tiempo discreto, sistemas híbridos, sistemas MF. Convierte automáticamente los modelos de SIMULINK a código en C.

En general, el RTW genera código para una plataforma diferente (target, sistema objetivo) a aquella en que corre el MATLAB (host, sistema anfitrión).

Existen procedimientos para la generación de código, creación del ejecutable y transferencia de éste al target (ejemplo: xPC, Tornado). Al trabajar en modo externo es posible: monitorear las variables y ajustar los parámetros, mientras el programa se ejecuta en el target (se tiene una comunicación tipo cliente-servidor vía TCP/IP, RS232 o Memoria Compartida).

3.2.2. Relación del RTW con otros elementos (Ver Figura 15):

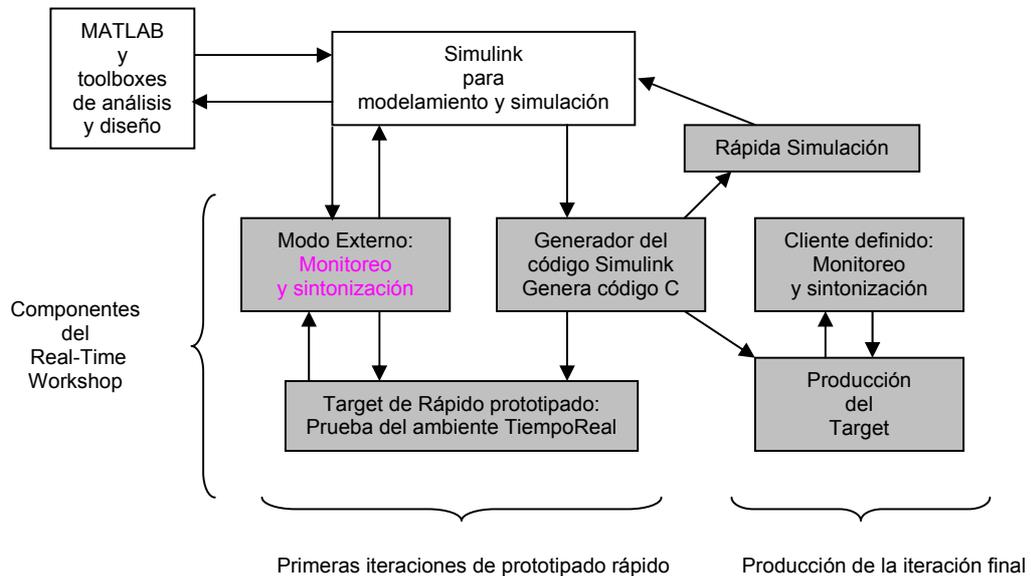


Figura 15: Relación del RTW con otros elementos.

3.2.3. Plataformas que trabajan en modo externo:

RTWT (Windows como host y target), Tornado (VxWorks), xPC, Real-Time Linux Target (Linux como host y target).

SIMULINK junto con el RTW es lenguaje de muy alto nivel (VHLL); los bloques de SIMULINK son los elementos básicos del lenguaje gráfico; el generador de código se incluye en el RTW y es un compilador gráfico de última generación basado en diagramas de bloques, a partir de los cuales se crea un código en C en forma rápida y correcta, altamente legible y personalizado.

3.2.4. Estructura de programación:

El RTW utiliza el lenguaje TLC (Target Language Compiler) para obtener el código a partir del diagrama de simulación; TLC utiliza dos tipos de archivos (extensión .TLC), los cuales pueden personalizarse:

System target file. Describe cómo se genera el código para un determinado target.

Block target file. Define el código para un modelo determinado.

El RTW utiliza plantillas .TMF para la construcción de ejecutables, dependiendo del target. Una rutina make_rtw genera un archivo model.rtw y de ahí el model.mk a partir de la plantilla .TMF. Del model.MK se genera el ejecutable.

3.3. REAL-TIME WINDOWS TARGET

RTWT es un sistema auto-dirigido (*self-targeting*), donde el host y el target están en el mismo computador; sistemas operativos soportados: Win 95, 98, NT, Win ME y Win XP.

Productos requeridos: MATLAB, SIMULINK, RTW y un compilador de C (en esta aplicación utilizamos Microsoft Visual C++ 6.0).

3.3.1. INSTALACIÓN DE VISUAL C++ DESDE LA PANTALLA PRINCIPAL DE MATLAB

>> *mex -setup*, comando para la escogencia del compilador a ser usado en los procesos en tiempo real (Ver Figura 16).

Previamente se debe haber instalado en Windows, algún compilador de los que están en la lista de la grafica.

Se recomienda que el computador tenga RAM de 32 megas o más.

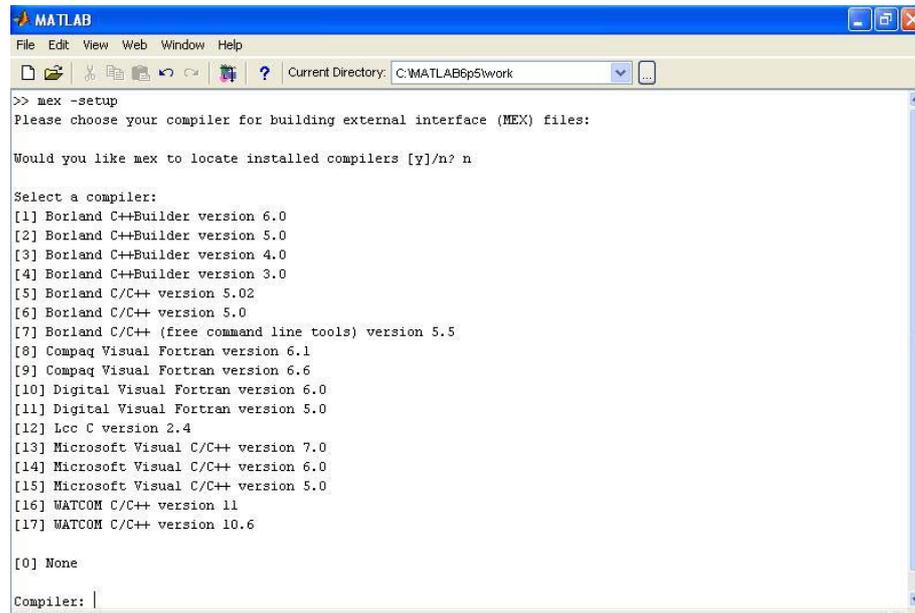


Figura 16: Instalación del compilador de lenguaje C.

3.3.2. KERNEL

El RTWT utiliza un pequeño núcleo (kernel) de tiempo real para asegurar que las aplicaciones realmente se ejecuten en tiempo real (arranca cada vez que se inicia Windows). El núcleo tiene mayor prioridad que Windows, porque esto es lo que permite darle a las aplicaciones de tiempo real la mayor prioridad posible. En Windows el Kernel es un VxD(controlador de dispositivo virtual).

Instalación del kernel desde la pantalla principal de MATLAB:

>>*rtwintgt -install*; este comando es indispensable para realizar la aplicaciones de tiempo real, porque es el que conecta, en pocas palabras, a MATLAB con la tarjeta de adquisición de datos.

3.3.3. ELEMENTOS BASICOS DEL RTWT (Ver Figura 17):

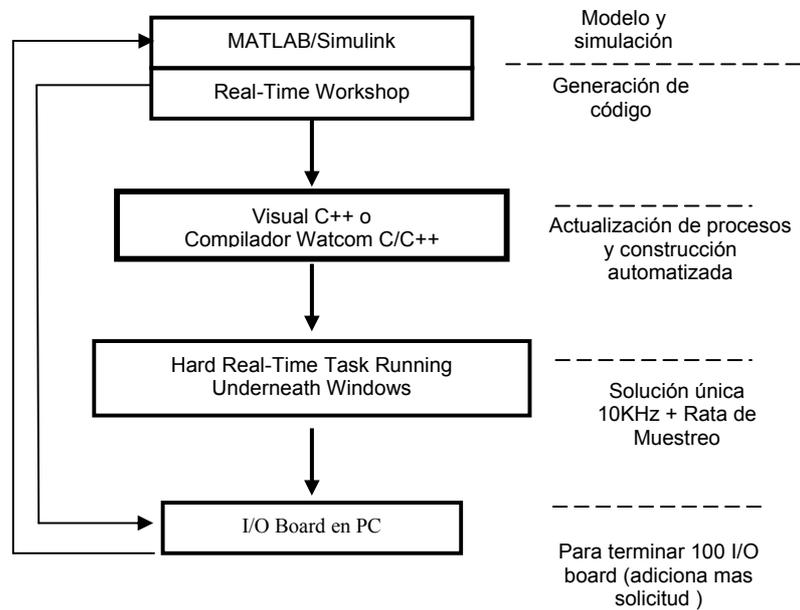


Figura 17: Elementos básicos del RTWT.

4. CONTROL EN TIEMPO REAL UTILIZANDO LOGICA DIFUSA

En este capítulo se muestra cómo se aplica al sistema no lineal de esta tesis, en base a la teoría de la lógica difusa para controlar la posición de una esfera, en tiempo real, con MATLAB 6.5.

El modelo de control difuso consiste en la implementación en el FIS editor de MATLAB de un controlador Fuzzy-PD tipo Mandani.

4.1. DISEÑO DEL CONTROLADOR CON EL TOOLBOX DE FUZZY LOGIC

Se expone de un modo más o menos formal los conceptos básicos relacionados con la lógica y sistemas difusos, y, paralelamente, se aborda el diseño del controlador utilizando el toolbox de fuzzy logic.

4.1.1. CONJUNTOS DIFUSOS

En los conjuntos clásicos algo está incluido completamente en él o no lo está en absoluto. Esta situación puede describirse asignando un 1 a todos los elementos incluidos en el conjunto, y un 0 a los no incluidos. A la función que asigna estos valores se denomina función de inclusión o pertenencia (*membership function*). Los conjuntos difusos permiten describir el grado de pertenencia o inclusión de un objeto (o el valor de una variable) al concepto dado por la etiqueta que le da nombre, asignando un número real entre 0 y 1.

Para este controlador se escogen 2 conjuntos difusos o variables de entrada, el error *ERROR* y la derivada del error *DERROR*; y uno de salida, el voltaje *Vsal* (Ver Figura 18).

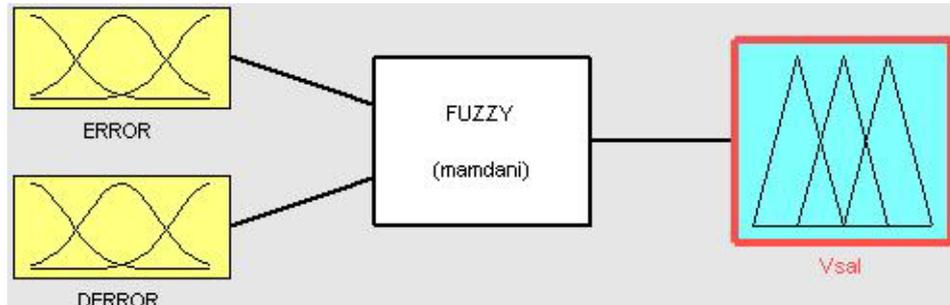


Figura 18: Variables de entrada y salida en el FIS editor de MATLAB.

- Como *ERROR* se toma la diferencia existente entre la referencia o *set point* $r(t)$, suministrada por el usuario, y la altura de la esfera suministrada por el sensor en un instante dado $y(t)$:

$$e(t) = r(t) - y(t)$$

De $e(t)$ se observa que si, por ejemplo, la pelota se encuentra a una altura de 30cm y se quiere llevar a una de 40cm el error sería de 10cm. El error puede ir desde -100cm hasta 100cm.

- La segunda entrada al controlador *DERROR* no es más que la diferencia entre el error actual $e(t)$ y el error en el intervalo de muestreo anterior $e(t-1)$. Esta entrada da una medida de cuán rápida es la respuesta del sistema. La ecuación para esta entrada es:

$$\Delta e(t) = e(t) - e(t-1) \quad \text{La derivada de error puede ir de -1cm hasta 1cm.}$$

- La variable de salida *Vsal* es el voltaje que se suministra al motor DC, como resultado de las operaciones que realiza el controlador FUZZY con las variables de entrada. Voltaje puede ir de 4v a 7v.

4.1.2. FUNCIONES DE INCLUSIÓN

La función de inclusión o pertenencia (*membership function*) de un conjunto difuso consiste en un conjunto de pares ordenados $F = \{(u, u_F(u)) \mid u \in U\}$ si la variable es discreta, o una función continua si no lo es. El valor de $u_F(u)$ indica el grado en que el valor u de la variable U está incluida en el concepto representado por la etiqueta F . Se utilizan convencionalmente ciertas familias de formas estándar, por coincidir con el significado lingüístico de las etiquetas más utilizadas. Las más frecuentes son la función de tipo trapezoidal, singleton, triangular, S, exponencial y tipo π .

Para los conjuntos difusos de esta aplicación se tuvieron en cuenta la de tipo Trapezoidal y la de tipo Triangular, las cuales permiten definir un conjunto difuso con pocos datos y calcular su valor de pertenencia con pocos cálculos.

- **Función tipo Trapezoidal:**

Se define por cuatro puntos a, b, c, d . (Ver Figura). Esta función es 0 para valores menores de a y mayores de d , vale 1 entre b y c , y toma valores en $[0,1]$ entre a y b , y entre c y d (ver gráfica 29a).

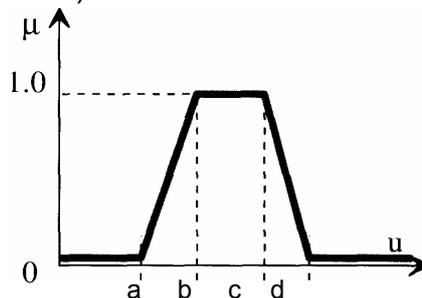


Figura 19. Función de pertenencia tipo trapezoidal

Esta función resulta adecuada para modelar propiedades que comprenden un rango de valores (*adulto, normal, adecuada...*).

- **Función tipo T (triangular):**

Esta función es adecuada para modelar propiedades con un valor de inclusión distinto de 0 para un rango de valores estrecho en torno a un punto b . (Ver Figura 29b)

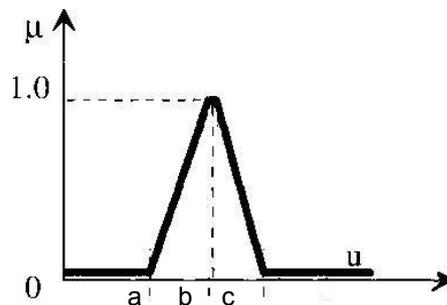


Figura 20. Función de pertenencia tipo triangular.

4.1.3. VARIABLE LINGÜÍSTICA

Se denomina variable lingüística a aquella que puede tomar por valor términos del lenguaje natural, como *mucho, poco, positivo, negativo*, etc., palabras que pueden desempeñar el papel de etiquetas en un conjunto difuso. Aunque el objetivo principal de este concepto es expresar de manera formal el hecho de que pueden asignarse como valor de una variable, palabras tomadas del lenguaje natural, no obstante, a una variable lingüística podrán asignarse también valores numéricos.

4.1.4. PARTICIONES DIFUSAS

Se conoce por partición a un subconjunto de los conjuntos difusos que se han definido para la variable X . Una partición de X es uno de los subconjuntos que pueden formarse con los elementos (términos) de $T(X)$. Así, por ejemplo para una variable llamada *ESTATURA*, las posibles particiones serían las correspondientes

con tres posibles subconjuntos difusos, cada uno identificado por una etiqueta, *Bajo, Medio, Alto*.

Para nuestra aplicación se opta por 5 particiones para los conjuntos difusos *ERROR* y *Vsal*, y para *DERROR* 3 particiones, con lo cual la complejidad para el controlador no es excesiva, y se le logra suficiente precisión.

Para *ERROR* se escogen 5 funciones inclusión, 2 de tipo trapezoidal con variables lingüísticas llamadas *alto(-)* y *alto(+)*; y 3 de tipo triangular llamadas *bajo(-)*, *medio* y *bajo(+)* (Ver Figura 21).

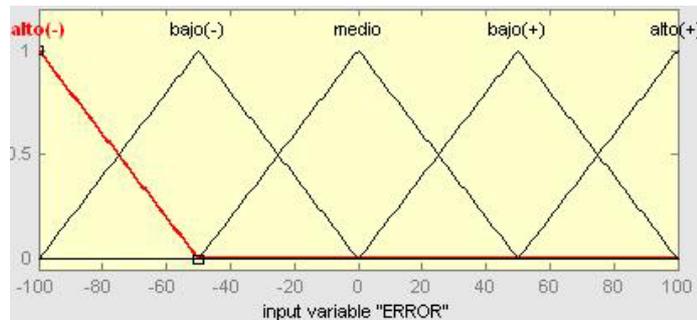


Figura 21: Particiones y funciones de inclusión de *ERROR* en el funtion editor de MATLAB
Para *DERROR* se escogen 3 funciones de inclusión, 2 de tipo trapezoidal con variables lingüísticas llamadas *negativa* y *positiva*; y 1 de tipo triangular llamada *media* (Ver Figura 22).

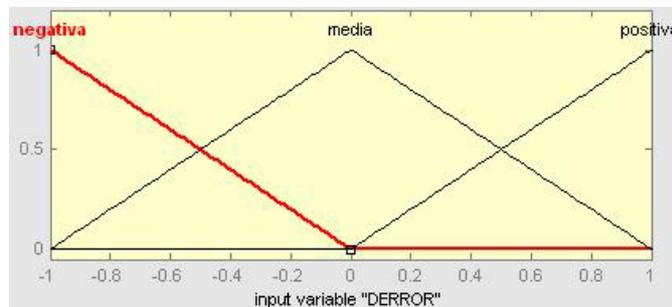


Figura 22: Particiones y funciones de inclusión de *DERROR* en el funtion editor de MATLAB.

Para el *Vsal* se escogen 5 funciones de inclusión, 2 de tipo trapezoidal con variables lingüísticas llamadas *bajo* y *alto*; y 3 de tipo triangular llamadas *mediobajo*, *medio* y *medioalto* (Ver Figura 23).

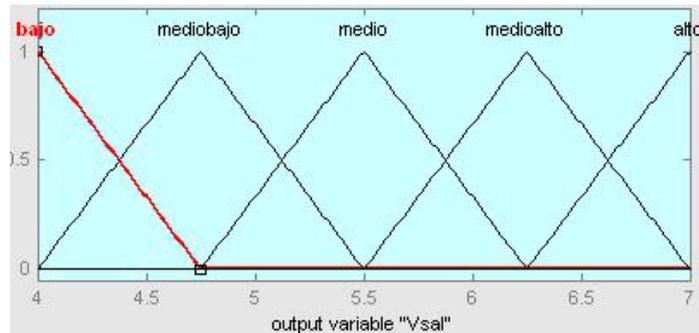


Figura 23: Particiones y funciones de inclusión de *Vsal* en funtion editor de MATLAB.

4.1.5. BASE DE REGLAS

Las reglas difusas combinan uno o más conjuntos difusos de entrada, llamados *antecedentes* o *premisas*, y les asocian un conjunto difuso de salida, llamado *consecuente* o *consecuencia*. Los conjuntos difusos de la premisa se asocian mediante conjuntivas lógicas como *y*, *o*, etc. Una regla típica, de tipo IF-THEN, para un sistema cualquiera de control sería "Si error es positivo_pequeño y derivada_de_error es negativo_pequeño Entonces acción es positiva_pequeña".

Las reglas difusas permiten expresar el conocimiento que se dispone sobre la relación entre antecedentes y consecuentes. Para expresar este conocimiento de forma completa, normalmente, se precisa de varias reglas, que se agrupan formando lo que se conoce como una *base de reglas*, es decir, el conjunto de reglas que expresan las relaciones conocidas entre antecedentes y consecuentes.

Para combinar el ERROR y DERROR con Vsal se utiliza la regla típica, de tipo IF-THEN, en donde la base de reglas que rigen la planta se muestra en una tabla de reglas (Ver Tabla 1) y en el ruler editor de MatLab (Ver figura 24).

		ERROR				
		alto(-))	bajo(-)	medio	bajo(+)	alto(+)
DERROR	negativa	bajo	mediobajo	medio	medioalto	alto
	media	bajo	mediobajo	medio	medioalto	alto
	positiva	bajo	mediobajo	medio	medioalto	alto

Tabla 1: Base de reglas para el control de posición

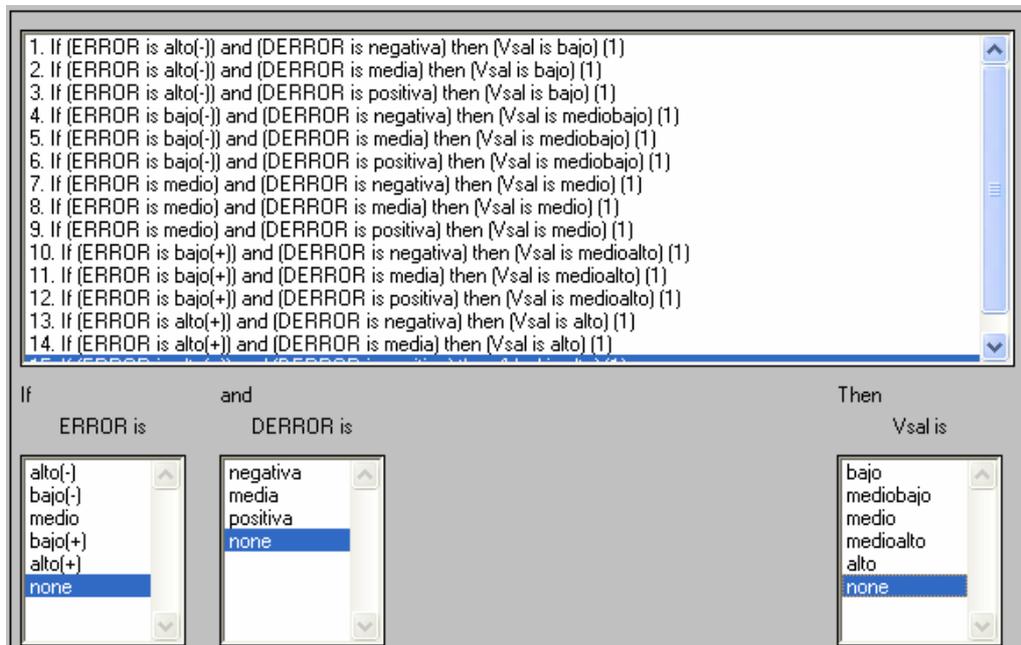


Figura 24: Bases de reglas para el control de posición en el Ruler Editor de MATLAB

Luego en el visor de reglas (*rule visor*) de MATLAB se pueden apreciar los conjuntos difusos de entrada (*ERROR*, *DERROR*) y de salida (*Vsal*), las funciones de inclusión y las particiones; además para la parte de **desfuzificación** se utiliza el método de centro de área (Ver Grafica 25).

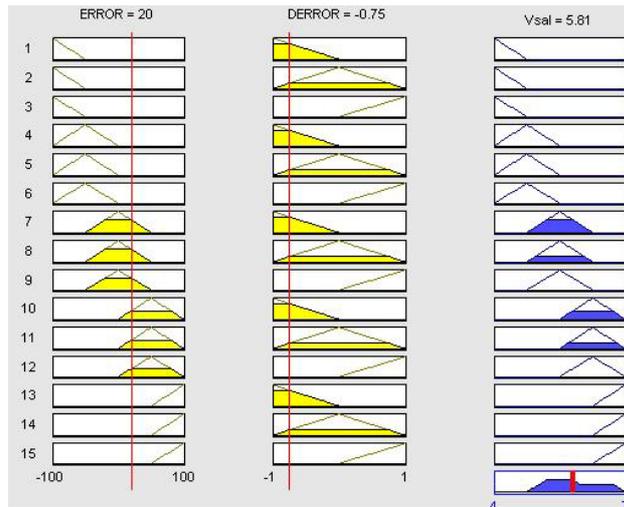


Figura 25: Características de nuestro controlador en el Rule visor de MATLAB.

Y en el visor de superficies (*surface visor*) de MatLab se genera la siguiente superficie de control (Ver Figura 26):

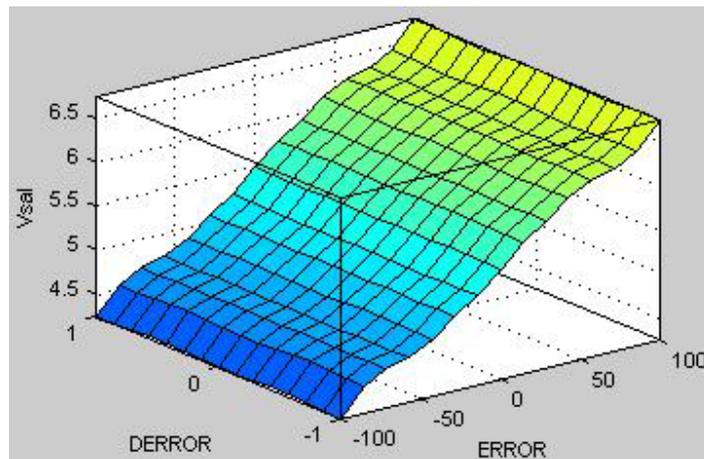


Figura 26: Superficie de los conjuntos en base a las reglas en el Rule visor de MATLAB.

4.2. FUNCIONAMIENTO EN TIEMPO REAL

Ahora se procede a explicar, de manera clara, los pasos para el funcionamiento del control de posición con lógica difusa, utilizando Simulink. Para luego finalmente correr el programa de control de posición de la esfera, obtener las gráficas y analizar los resultados.

4.2.1. MONTAJE COMPLETO EN SIMULINK

Los pasos para montar toda la planta en Simulink son los siguientes:

- De la librería de Simulink montar en el modelo los siguientes bloques: La entrada y salida análoga, el controlador difuso, la barra para el *set point*, *displays*, sumadores, multiplexores, una caja para subsistema, el derivador, un *To Workspace* para la toma de datos, unir todos los bloques como muestra la siguiente figura. (Ver Figura 27).

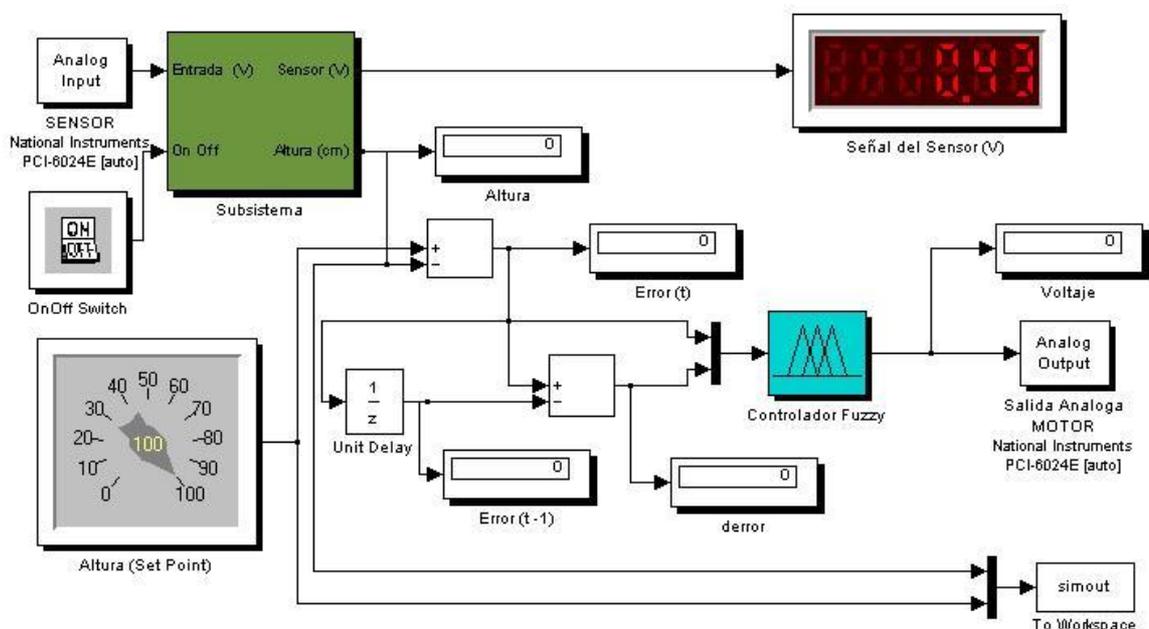


Figura 27: Montaje completo de la planta con Fuzzy en un modelo de Simulink.

- Se establecen los parámetros de la tarjeta de adquisición de datos, se instala, se le da el tiempo de muestreo y se habilitan los canales. (Ver Figura 28).

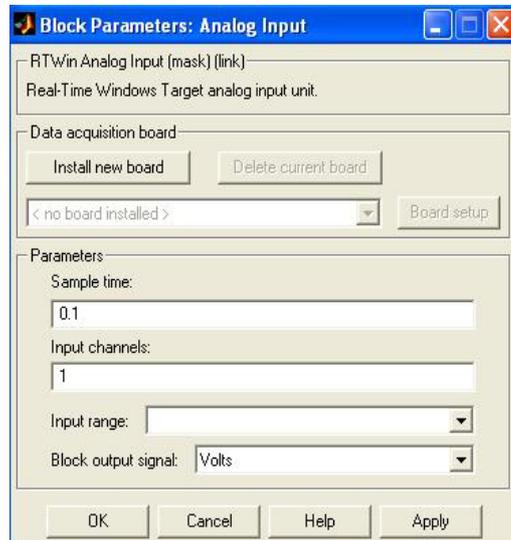


Figura 28: Parámetros del bloque de entrada y salida análogas.

- Luego, en el bloque *subsistema* se hace la linealización de la curva del sensor y se convierte la información que proporciona este último, de Voltaje, a Altura en centímetros (Ver Figura 29). El programa para calcular la ecuación de aproximación del sensor, y visualizar las diferentes curvas, para realizar las comparaciones entre ellas se encuentra en el Anexo B.

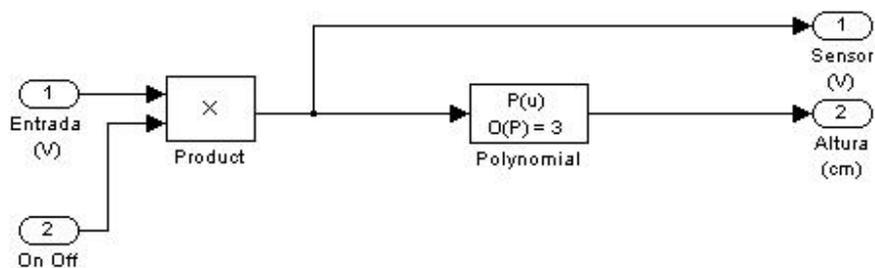


Figura 29: Linealización y conversión de la información del sensor desde Simulink.

4.2.2. RUN, START Y TOMA DE DATOS

Después de tener el montaje completo en Simulink, se procede a parametrizar la simulación en tiempo real, así:

- Se trabaja el simulink en modo externo (*mode external*).
- Luego en la barra superior de Simulink, en *Simulation*, se ingresa a *Simulation Parameters.....*, donde se configuran parámetros para el *Solver* como tiempo de simulación, formato del *Solver*; y enseguida a los parámetros del *Real Time WorkShop* (Ver Figura 30). Aquí se categoriza el *Real Time WorkShop*, primero configurando el *Target*. Luego de configurar el archivo del *Target* que en este caso será *rtwin.tlc*, se coloca en la categoría *Real Time Windows Target build options*, a fin de que la configuración de simulación quede lista para la compilación.

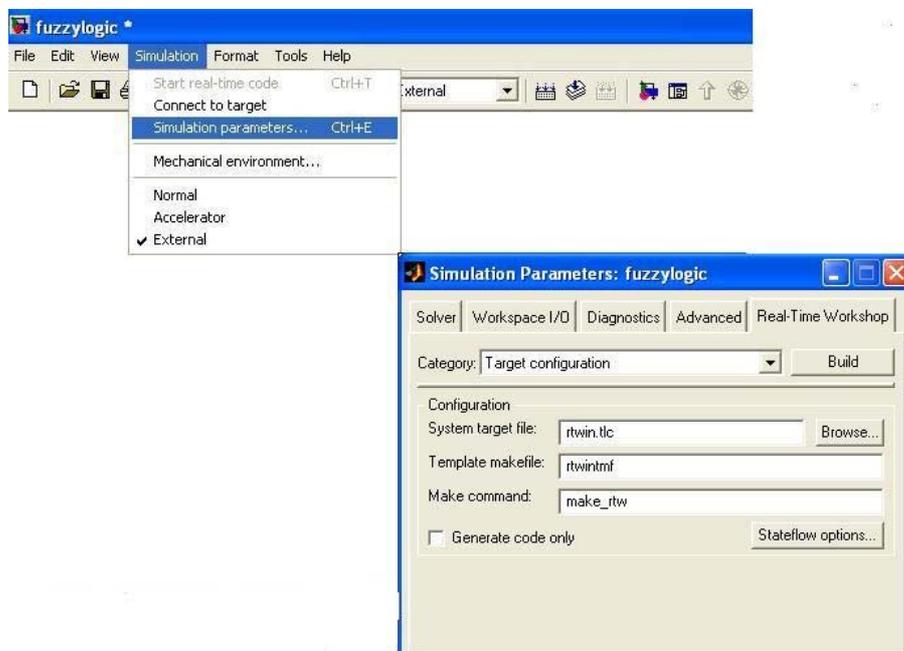


Figura 30: Configuración de los parámetros de simulación.

- Después en el *Tools* y *External mode control panel*...., se ingresaron las opciones deseadas en este caso el *rtwinext* (Ver Figura 31).

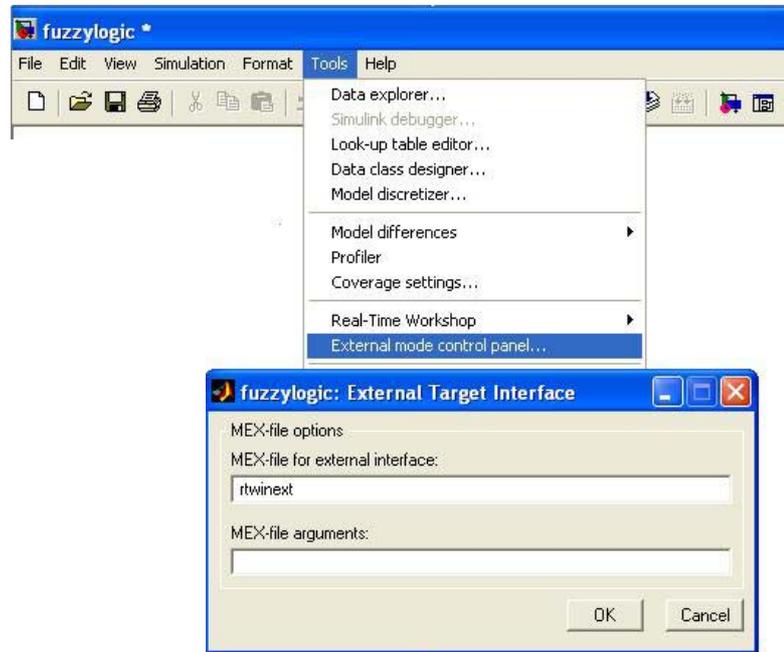


Figura 31: Configuración de los parámetros del *External mode*.

- Después se construyó el ejecutable (*build*) a partir del modelo, el cual se deberá reconstruir si se cambia la estructura del modelo.
- Luego se transfirió el ejecutable al *target*.
- Se conectó al *target* (*Connect to target* en el menú *Simulation*): Esto permite la conexión con el programa *target* desde el *host*.
- Al correr el modelo de la planta en Simulink (*Start*), el usuario en tiempo real ingresa la altura o *set point* que desee y aprecia el comportamiento físico del sistema.

4.2.3. GRAFICAS Y ANALISIS DE RESULTADOS

Para que el usuario pueda apreciar la gráfica con las respuestas de la planta a los cambios de *Set Point*, debe primero parar el modelo, ir a la pantalla principal de MatLab y escribir el comando `plot(simout)` y entonces le aparecerá la misma.

Lo siguiente es un ejemplo de la obtención de una gráfica del control *Fuzzy* a cambios de *Set Point* luego de haber corrido y parado la planta (Ver Figura 32).

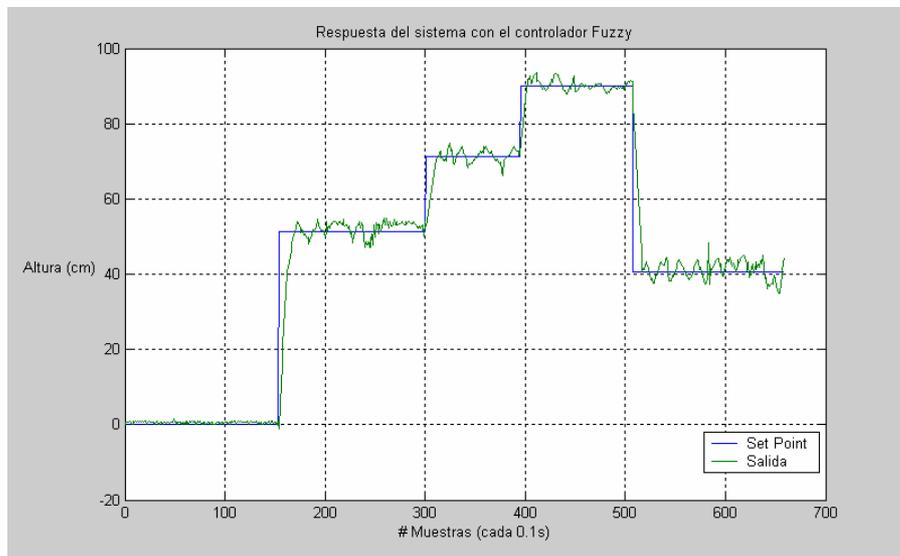


Figura 32: Respuesta del sistema con el controlador Fuzzy

En esta grafica se puede apreciar las respuestas de la planta a los cambios de *Set Point* de 0cm (apagado) a 50 cm., luego a 70cm., a 90 cm, y finalmente a 40 cm.

La planta alcanza los cambios de *set point* a 1 ó 2 segundos, dependiendo de la distancia (*ERROR* ±) y las oscilaciones alrededor de la misma son del 2 al 5 %.

Por lo tanto se puede considerar que el controlador *fuzzy* responde de manera satisfactoria, nunca perfecta, a pesar que el movimiento oscilatorio de la esfera es sumamente “caprichoso” debido a las perturbaciones y alinealidades inherentes al sistema.

5. CONTROL EN TIEMPO REAL UTILIZANDO PID

El propósito del controlador PID es suprimir los efectos de interrupciones en variables del proceso y forzar a una variable a seguir un punto de referencia deseado.

El controlador PID mide la salida del proceso y calcula la diferencia (error) entre lo que se ha medido y el punto de referencia. Si existe un error, el controlador ajusta su salida para alterar el proceso con el fin de acercarlo al punto de referencia, disminuyendo así el error. Cada vez que se calcule un error, el controlador debe decidir cuánto alterará el proceso.

5.1. CONTROL PID

El control PID consiste en tres efectos básicos que incluyen las siguientes acciones de control: Efecto Proporcional (P), Efecto Integral (I) y Efecto Diferencial (D). La mayoría de los lazos de control se pueden manipular mediante esquemas de este tipo de control (Ver figura 33).

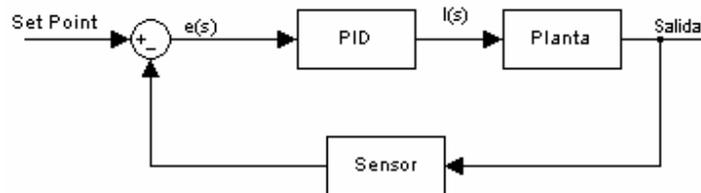


Figura 33: Diagrama de bloque del controlador PID

5.2. DISEÑO DEL CONTROLADOR PID

Debido a que se desconoce el modelo de la planta, las constantes del controlador PID matemáticamente no se puede hallar; dada esta dificultad, se crea la necesidad de calcular estos valores constantes por métodos de sintonización.

5.3. METODOS DE SINTONIZACION

La sintonización del controlador consiste en asignar valores a las constantes (K_p , T_i y T_d) de forma que el lazo de control gobernado por el controlador responda adecuadamente a las perturbaciones externas y que consiga un funcionamiento adecuado del sistema (en términos de tiempo de respuesta aceptable, pico de sobreoscilación aceptable, etc.).

Debido a su difundido uso en la práctica, presentamos a continuación 2 métodos de ajuste empírico de controladores PID.

1. Método de Relay.
2. El método de respuesta en frecuencia de Ziegler-Nichols.

5.3.1. MÉTODO DEL RELAY

Este método de auto-sintonización, fué propuesto y utilizado por Åstrom y Hägglund en (1984); este es una versión automatizada del segundo método experimental de Ziegler –Nichols.

5.3.1.1. IMPLEMENTACIÓN EN TIEMPO REAL

En Simulink Library Browser, encontramos el bloque Relay (Ver Figura 34).

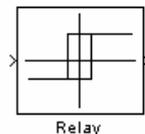


Figura 34: Bloque del Relay.

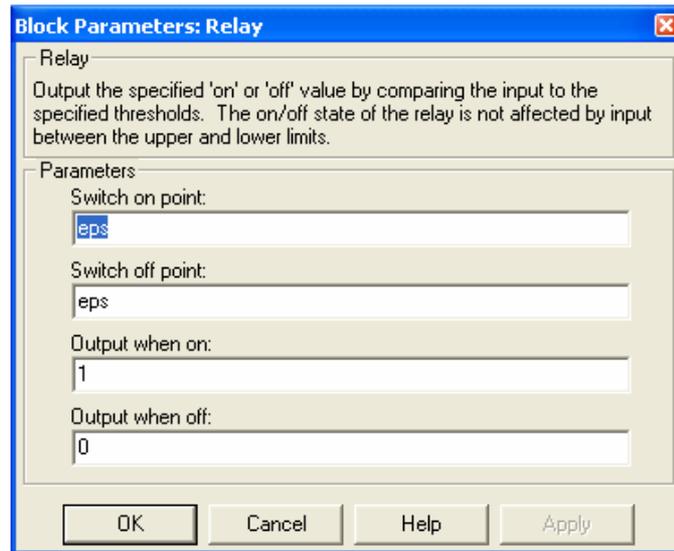


Figura 35: Bloque de los parámetros del relay

Este bloque contiene los siguientes parámetros, los cuales se pueden modificar de acuerdo al procedimiento de sintonización. (Ver figura 35)

5.3.1.2. PARÁMETROS DEL RELAY

Switch on point: Por configuración en este parámetro aparece eps, este valor se puede dejar así o cambiarlo. En este caso no se le hizo ningún cambio. Este parámetro indica el punto de switcheo en encendido, es decir cuando el *relay* se enciende.

Switch off point: Este es igual que el anterior solo que indica el punto de Switcheo en apagado.

Output when on: Es la salida máxima que suministra el Relay cuando se activa. Con este parámetro se comienza a variar, visualizando si la esfera alcanza o sobrepasa la referencia deseada, para encontrar el punto máximo donde la esfera

(pelota) alcance o sobrepase el *set-point* y de esta manera se consigue el rango máximo donde la esfera llega para la referencia deseada.

Output when off: Es la salida mínima cuando el relay se desactiva. Igual que el anterior, se va variando y se observa el comportamiento de la misma. El rango en el que la esfera oscile lo menor posible para un set-point deseado, permite establecer los parámetros de sintonización del relay.

5.3.1.3. PROCEDIMIENTO DE SINTONIZACIÓN.

El Relay va conectado en el sistema de acuerdo como lo muestra el diagrama de bloque de la figura 36.

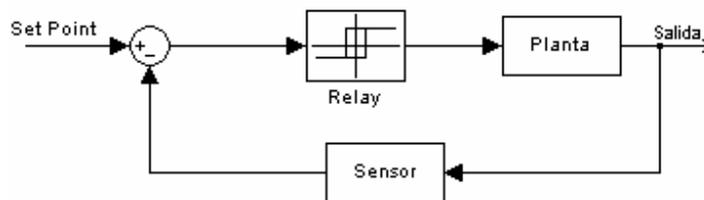


Figura 36: Diagrama de bloque para la conexión del relay

1. Se usa una entrada constante, es decir u_0 , y se determina el estado correspondiente de salida, es decir y_0 ; de forma general, se somete el sistema a una entrada constante y se observa como es la respuesta.
2. Con prueba y error se comienza a cambiar los parámetros del Relay. Para ir mejorando la estabilidad del sistema.

3. De este modo, la salida se debe colocar bajo un ciclo límite, tal como muestra la figura 37. Después de una serie de ajustes, se pudo lograr las siguientes respuestas del sistema, con las cuales se calculan las variables deseadas.

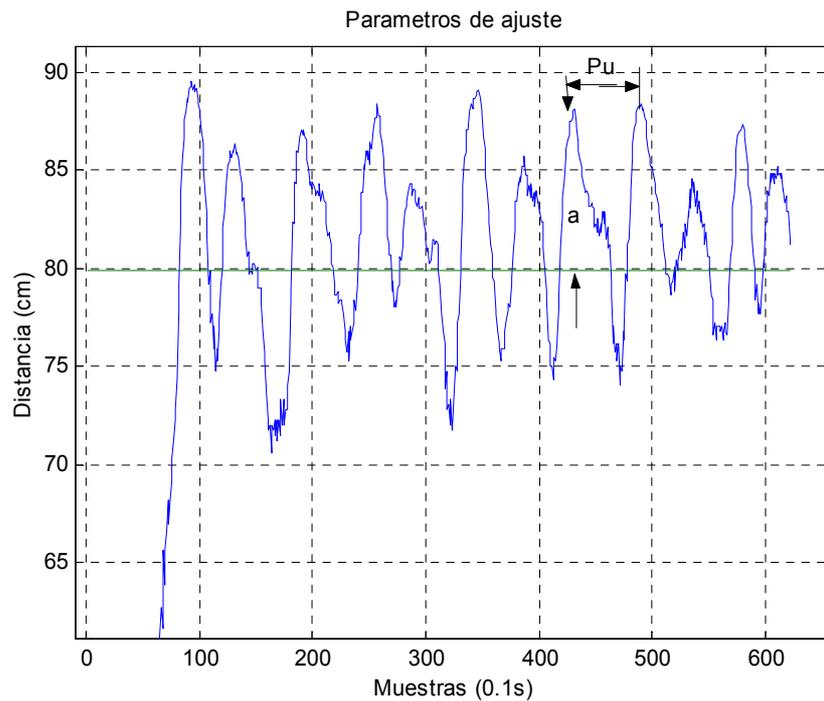


Figura 37: Respuestas del sistema para calcular parámetros del relay

De la figura anterior (ver figura 37), se mide la amplitud del ciclo de salida, a . Usando a y h , y el último aumento de ganancia, K_u . Se realiza la aproximación por medio de la siguiente ecuación:

$$K_u = \frac{4 * h}{\pi * a}$$

La gráfica anterior muestra que: $a = 7.8$, $P_u = 0.51s$ y el rango de variación del relay está en: Output when on $= 6.15$ y Output when off $= 5.8$, con esto se calcula, $h = 0.17$.

Remplazando en la ecuación se tiene:

$$K_u = \frac{4 * 0.17}{\pi * 7.8} = 0.027$$

4. Usando como aproximación la última ganancia K_u y el último período P_u (el período del ciclo límite de la gráfica anterior), se aplica la tabla 2 de cálculos de aproximación del método de Ziegler –Nichols, los cuales se estuvieron siguientes parámetros. $K_p=0.0162$ $T_i=0.25$ $T_d=0.061$

Estos valores obtenidos son una aproximación para sintonizar el controlador PID; estos pueden cambiar a valores muy diferentes a los obtenidos. Lo que se busca con los métodos de sintonización, es que se tenga una idea de donde se pueda comenzar a sintonizar el controlador.

5.3.2. MÉTODO DE RESPUESTA EN FRECUENCIA DE ZIEGLER–NICHOLS

Este método propone unas reglas para determinar los valores de ganancia proporcional K_p , del tiempo integral T_i y del tiempo derivativo T_d , con base en las características de respuesta transitoria de una planta específica.

Tipo de controlador	Constante Proporcional (K_p)	Tiempo de Integración (T_i)	Tiempo Derivativo (T_d)
P	$0.5K_u$	∞	0
PI	$0.45K_u$	$0.8P_u$	0
PID	$0.6K_u$	$0.5P_u$	$0.12P_u$

Tabla 2: Cálculos de los parámetros del PID

5.3.2.1. PROCEDIMIENTO DE SINTONIZACIÓN.

1. En primer lugar es necesario hacer el experimento siguiente, que consiste en alcanzar la frontera de estabilidad con una realimentación proporcional, estableciendo $T_i = \infty$ y $T_d = 0$.
2. Se varía muy lentamente la ganancia K desde 0 hasta un valor crítico en el que el proceso exhiba oscilaciones sostenidas.
3. Se miden los parámetros K_u y P_u llamados “ganancia última” y “período último”. Ganancia última y período último, son la ganancia (K) a la que empieza a oscilar el sistema.

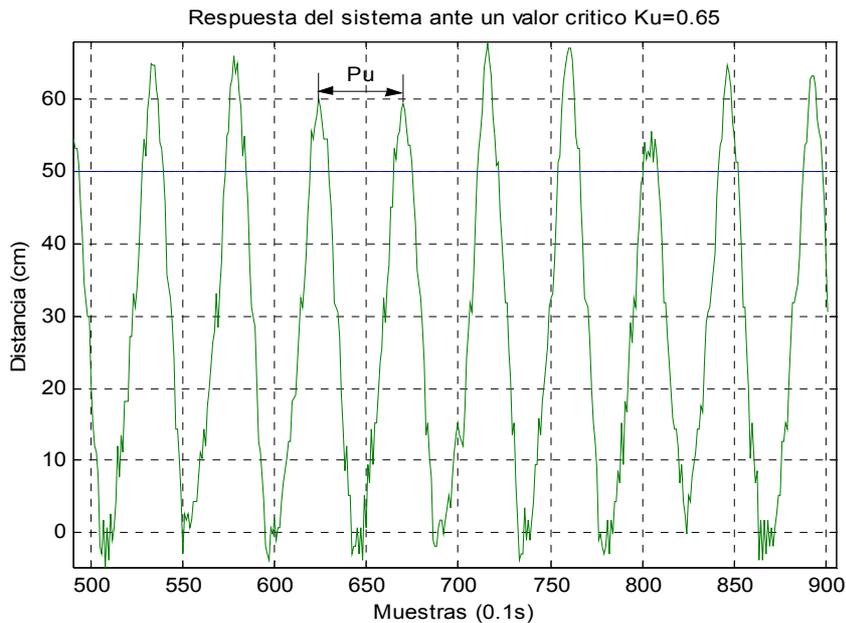


Figura 38: Respuesta del sistema ante un valor K_u conocido.

4. Una vez calculados estos parámetros, se puede hallar los valores del controlador, remitiéndose a la tabla 2.

Con la Figura 38 se tiene que: $K_u = 0.65$ y $P_u = 0.48s$

Calculando los valores para el controlador PID se halla que:

$$K_p=0.39 \quad T_i=0.24 \quad T_d=0.057$$

5.3.3. COMPARACIÓN DE LOS MÉTODOS DE SINTONIZACIÓN.

Con el método de sintonización Ziegler-Nichols para el controlador PID, se obtuvo la siguiente respuesta del sistema en Tiempo Real. Ver Figura 39.

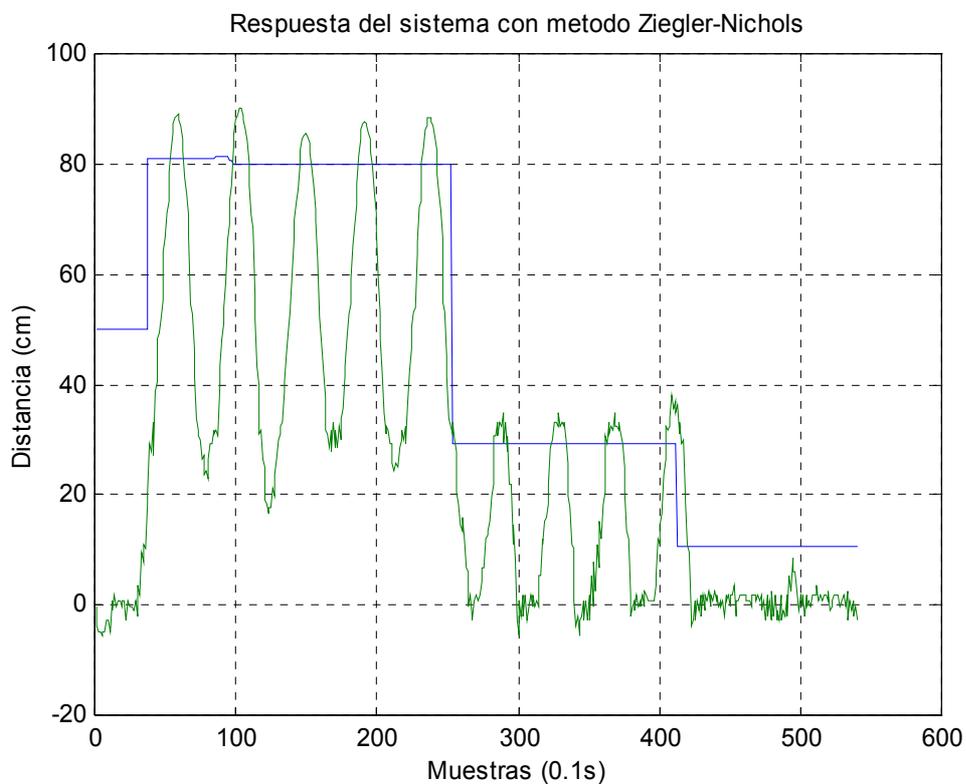


Figura 39: Respuesta del sistema con método de Ziegler-Nichols

Como se observa en la gráfica anterior, el sistema se vuelve bastante oscilatorio ante diferentes cambios de Set-point, pues aunque el sistema alcanza la referencia, es demasiado inestable.

La Figura 40, muestra la respuesta del sistema, con los parámetros calculados por el método de sintonización del Relay.

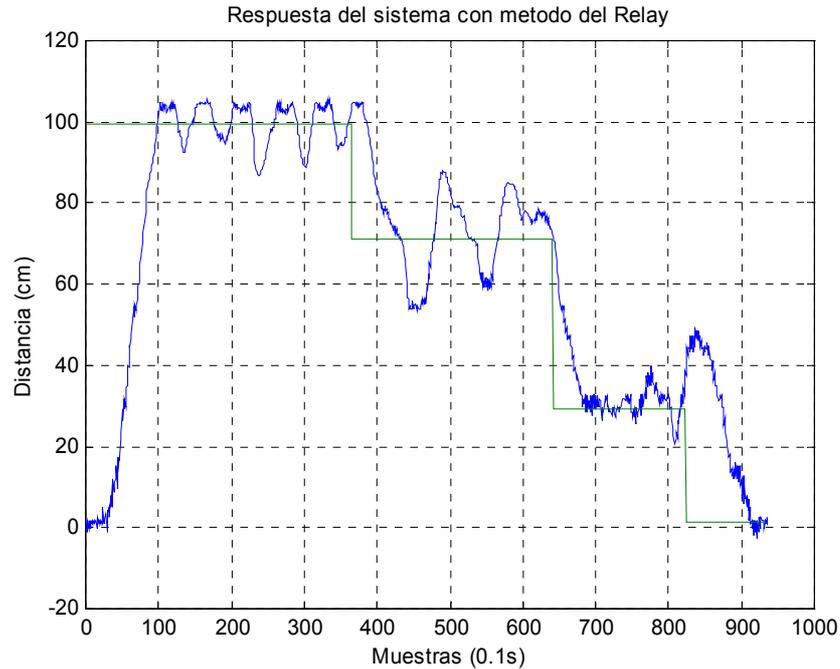


Figura 40: Respuesta del sistema con el metodo del Relay

Como se puede observar, el sistema presenta menos oscilaciones, e intenta alcanzar el punto de referencia de forma leve, siendo más estable ante los diferentes cambios de *Set-point*.

Comparando las respuestas obtenidas de los dos métodos utilizados para sintonizar el controlador PID, se concluye que es conveniente y más eficaz utilizar el método del Relay para sintonizar el controlador PID.

5.4. OBTENCION DE LOS PARAMETROS PID.

Con el método de sintonización del Relay se encontraron los siguientes parámetros: $K_p=0.0162$ $T_i=0.25$ $T_d=0.061$

En Simulink, se escoge el diagrama de bloque del controlador PID y se inserta en la hoja de trabajo de Simulink. Simulink ofrece una variedad de estos bloques de controladores PID, que se pueden ejecutar en "Real Time" (Ver figura 41).

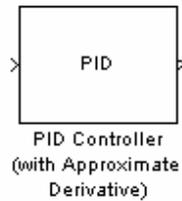


Figura 41: Bloque del controlador PID

Se escogió el bloque PID Controller (with Aproxímate Derivative), ya que este presenta la ventaja de que tiene aproximación derivativa y como se sabe, la parte derivativa da más estabilidad al sistema.

5.4.1. PARÁMETROS DEL CONTROLADOR

Este controlador PID presenta los siguientes parámetros tal como se muestra en la figura 42:

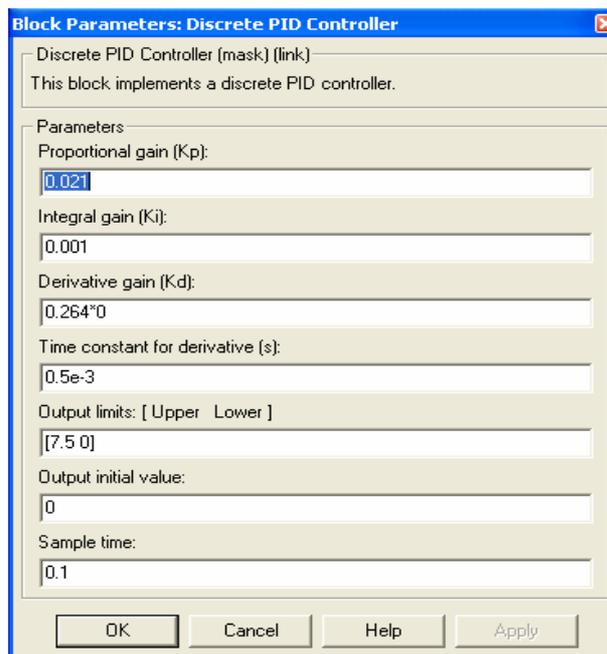


Figura 42: Parámetros del bloque de PID

Los valores obtenidos para el controlador PID son reemplazados en los parámetros de este, y se comienza con el ajuste para la sintonización del controlador PID.

Por prueba y error; se lograron conseguir los siguientes valores para el controlador

$$K_p=0.021 \quad T_i=0.001 \quad T_d=0.0001$$

5.5. FUNCIONAMIENTO EN TIEMPO REAL

Ahora se procede a explicar, de manera clara, los pasos para el funcionamiento del control de posición con un controlador PID, utilizando Simulink. Luego se correr el programa cómo se hizo en la sección 4.2.2.

5.5.1. MONTAJE COMPLETO EN SIMULINK

Los pasos para montar toda la planta en Simulink son los siguientes:

- De la librería de simulink, montar en el modelo los siguientes bloques: La entrada y salida análoga, el controlador PID, la barra para el *set point*, *displays*, sumadores, multiplexores, una caja para subsistema, un *To Workspace* para la toma de datos, unir todos los bloques para formar el siguiente diagrama. (Ver Figura 43).

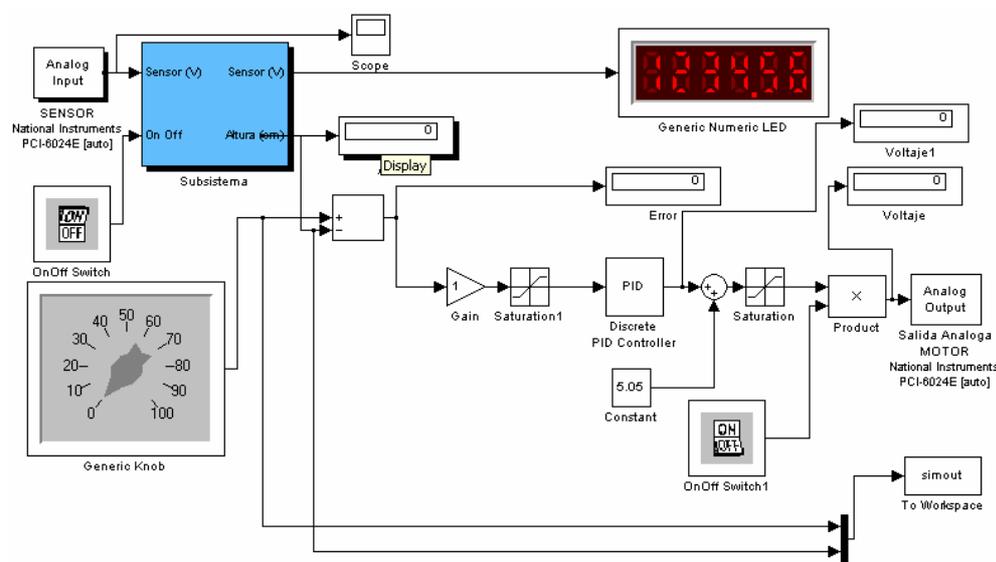


Figura 43: Montaje completo de la planta con PID en un modelo de Simulink.

5.5.2. GRAFICAS Y ANALISIS DE LOS RESULTADOS

Con las siguientes gráficas se puede observar las respuestas obtenidas del sistema en Tiempo Real, con el controlador PID. Estas gráficas, muestran cómo es el comportamiento del sistema, para diferentes **Set-points**.

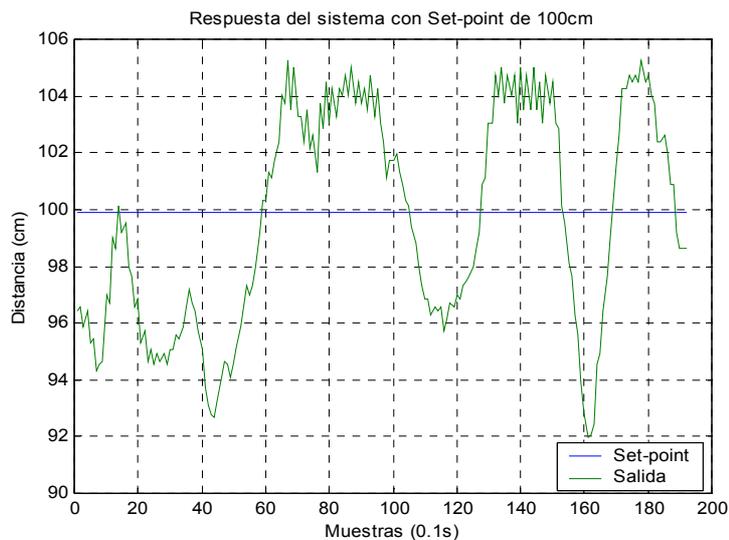


Figura 44: Respuesta del sistema para set-point de 100cm

La figura 44 muestra el comportamiento oscilatorio del sistema, aunque el error es del 10% aprox., el mismo es aceptable para el sistema y sobre todo si es controlado con PID. También se aprecia que a los 16seg., la esfera cae súbitamente a 12cm por debajo de la referencia, esto pudo ser producido por turbulencia, en un tiempo determinado.

La figura 45 muestra la respuesta del sistema para un *set-point* de 20cm. Las oscilaciones están dentro de un rango de 18cm, y el error del 18% aprox., que en comparación con la respuesta del sistema para un *set-point* de 100cm, presentando mayor error en el rango de oscilaciones, provocado por la

inestabilidad del sistema a esta referencia. El sistema responde al cambio de *set-point* a los 3.08seg de realizado.

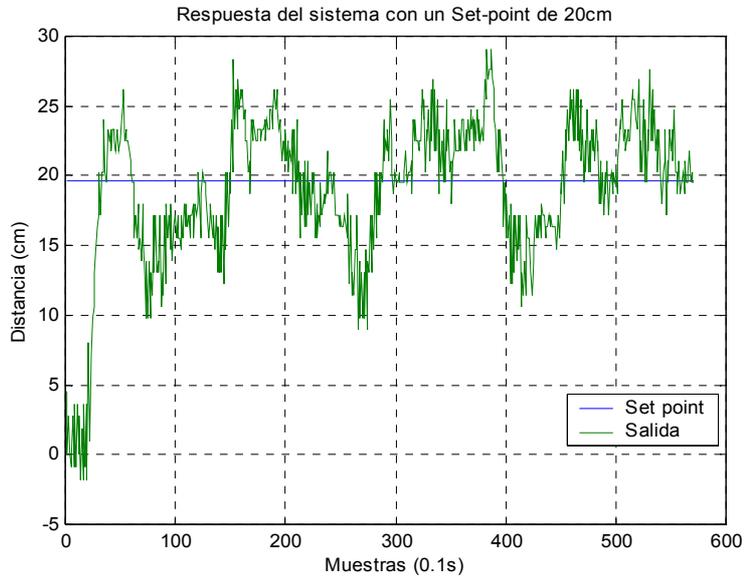


Figura 45: Respuesta del sistema para un set-point de 20 cm

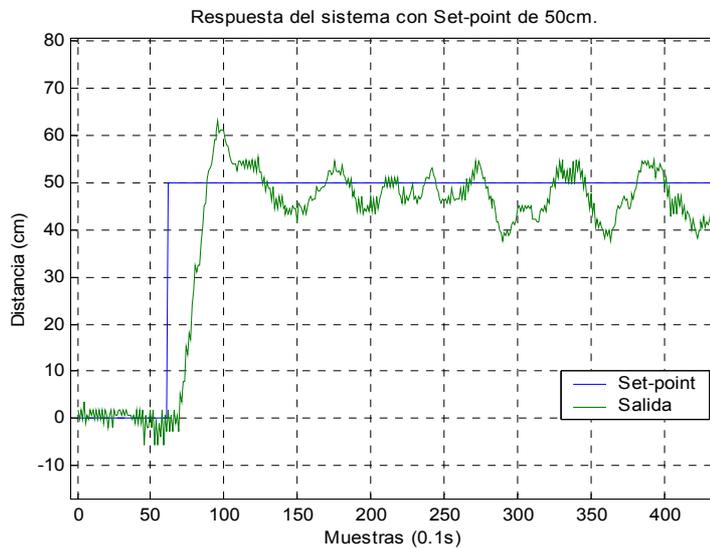


Figura 46: Respuesta del sistema con cambio de referencia de 0-50cm

La figura 46 muestra el sistema con un *set-point* a 50cm. Se observa un sobrepaso máximo de 11cm con respecto a la referencia, las oscilaciones están

en un rango promedio de 15cm, el *set-point* es alcanzado a los 2.2seg, lo que demuestra que el tiempo de respuesta es relativamente bueno, para esta referencia. Pero nos muestra que para esta referencia necesita mayor tiempo de establecimiento que las referencias mostradas anteriormente.

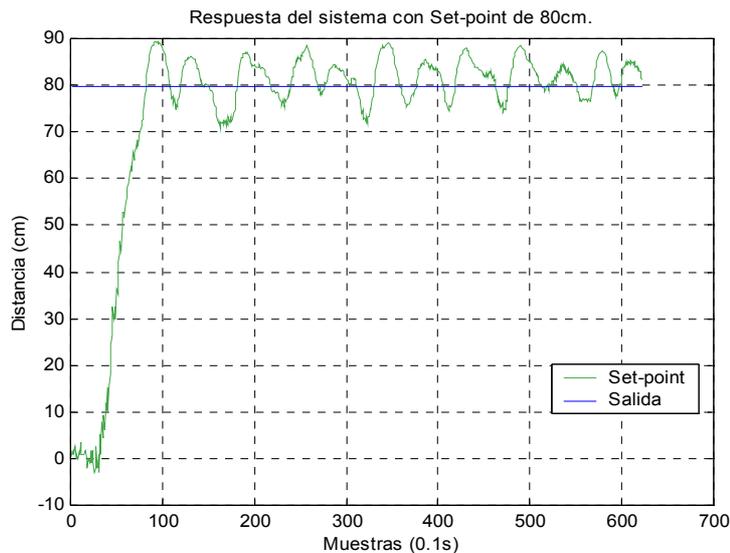


Figura 47: Respuesta del sistema para set-point 80cm

En la figura 47 se observa que para este *Set-point* a 80 cms., el sobrepaso máximo es prácticamente uniforme, porque todas las oscilaciones, por encima de la referencia, llegan casi a 90cm, es decir tienen un sobrepaso máximo de 10cm, presentándose un error del 16%. También se observa que la referencia es alcanzada a los 2.07seg de ser encendido el sistema, siendo la respuesta más rápida que la obtenida con el set-point de 20cm (Ver Figura 45), mostrando mayor eficiencia en este punto de referencia.

6. ANALISIS COMPARATIVO DE LOS RESULTADOS

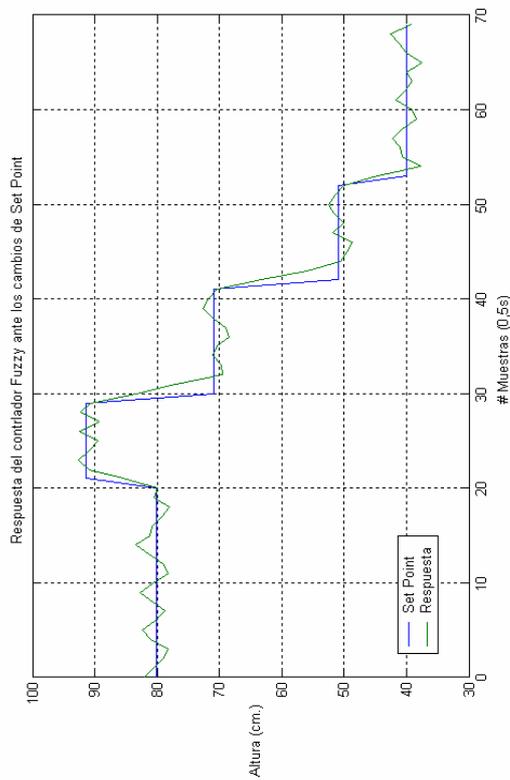
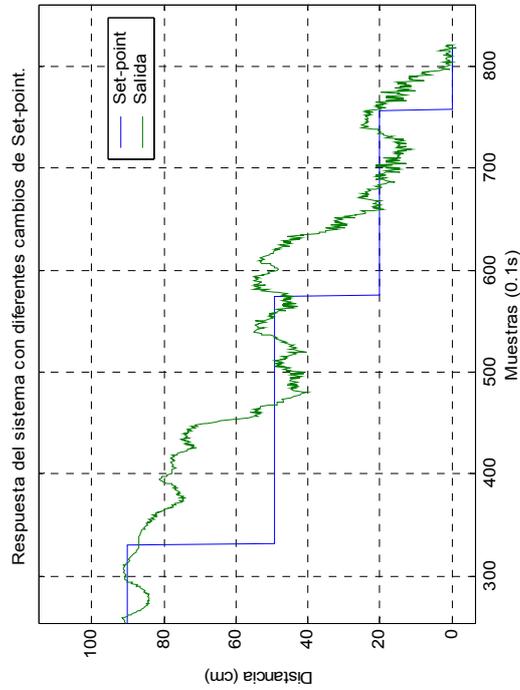
En este capítulo final se hace una explicación comparativa en base a las gráficas de respuesta a cambios de *Set Point* del sistema, no lineal, con controlador Fuzzy y controlador PID.

Lo que se pretende con este análisis comparativo no es demostrar la superioridad de un tipo de controlador sobre otro, debido a que mucho depende de la aplicación y las especificaciones de funcionamiento del sistema de control, y además, debe quedar perfectamente claro que estas nuevas técnicas no vienen a suplantar a las más tradicionales como el PID, sino más bien a completarlas en aquellos problemas donde menos eficacia proporcionan.

Se ha dado paso a estrategias con controles no convencionales capaces de adaptarse a medios ruidosos, perturbados y donde puedan existir variaciones en los parámetros de la planta. El control “fuzzy” o difuso, ofrece una nueva alternativa en la estrategia de control, donde la decisión o acción final de control esta fundamentada en la intuición, conocimiento y experiencia humana, permitiendo obtener controles semi-inteligentes.

Actualmente existe una extensa información, investigación y aplicaciones del control difuso. Así mismo en el ámbito académico se están planteando reformas en los contenidos de las materias de control introduciéndose nuevas estrategias denominadas control “inteligente”.

6.1 RESPUESTA DESCENDENTE DE LOS CONTROLADORES FUZZY Y PID.



Grafica 48: Respuesta descendente de los controladores Fuzzy y PID

La **Figura 48** es de gran importancia, porque muestra el comportamiento del sistema con controladores PID y Fuzzy para referencias menores que la referencia inmediatamente anterior y visualiza la respuesta del sistema cuando este cae.

Se puede apreciar que el controlador PID tiene un “dudoso” desempeño al seguir la referencia por que el tiempo de respuesta es de entre 14 seg. a 4 seg. dependiendo la altura del set point, y se infiere que entre más cerca esté el set-point del punto inicial, menos lento responde el sistema.

El Fuzzy tiene un excelente desempeño al seguir la referencia, ya que el tiempo de respuesta es de entre 1 a 3 seg. máximo, dependiendo la altura del set point y no difiere mucho de la distancia a que se encuentre este para tratar de reducir el *ERROR* a 0 lo más rápido posible.

La diferencia entre PID y Fuzzy para este comportamiento es que los controladores proporcionales y derivativos del PID son muy sensibles a señales con ruido y es necesario un ajuste muy preciso para hacerlos funcionar adecuadamente. También, la acción proporcional tiene una posibilidad de introducir oscilaciones cuando existe un compromiso con el tiempo y esto hace que se retrase la respuesta en tiempo de la misma.

El PID es un controlador lineal, esto es, las ecuaciones del PID asumen valores lineales. Sin embargo cuando la dinámica del proceso no es lineal, hay que tener especial cuidado para compensar la no linealidad del sistema, y el controlador PID por si mismo no es suficiente para lograr esta compensación. La esencia del control difuso es explotada en este tipo de hechos conocidos porque los sistemas que usan control difuso tienen las mismas entradas que el PID (error y su

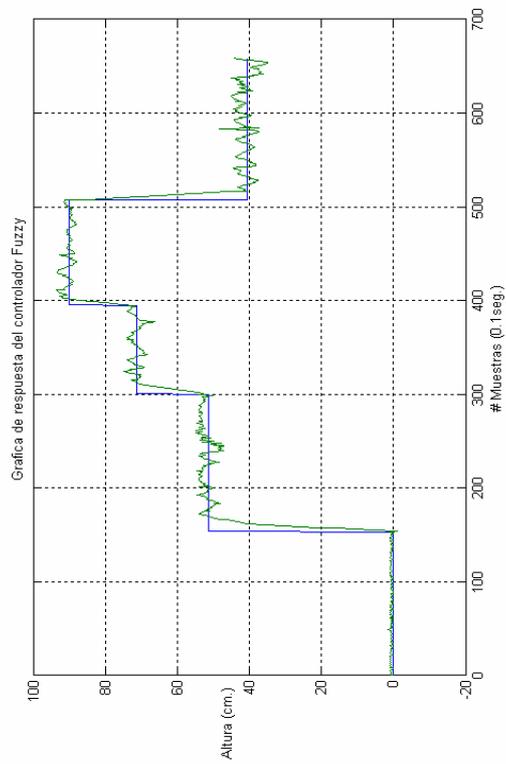
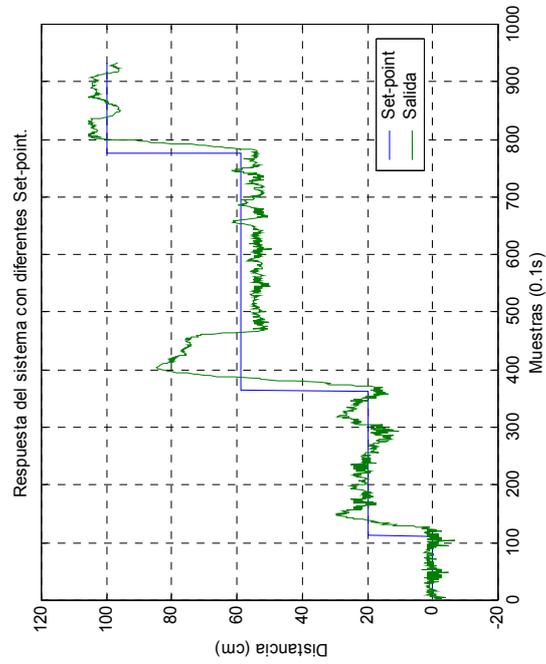
derivada), pero procesan la no linealidad del sistema. De aquí que el controlador difuso puede ser usado como un controlador PID no lineal. En la misma grafica, se puede apreciar perfectamente que la respuesta del sistema tanto con controlador difuso es más rápida e intente estabilizarse también más rápido que el PID.

Controlador \ parámetro	% de Error	Tiempo promedio en llegar a la referencia
PID BAJADA	15.13%	De 4 a 14 seg
FUZZY BAJADA	4.26%	De 1 a 3 seg

Tabla 3: Comparación entre Fuzzy y PID de bajada

La tabla 3 anterior muestra una comparación entre los diferentes controladores en forma ascendente que fueron implementados en la planta, esta tabla compara el porcentaje de error promedio a través del tiempo y el tiempo de establecimiento de cada uno de los controladores.

6.2 RESPUESTA ASCENDENTE DE LOS CONTROLADORES FUZZY Y PID.



Grafica 49: Respuesta ascendente de los controladores Fuzzy y PID

De igual forma la **Figura 49** es muy interesante, porque se logra observar el comportamiento del sistema cuando se cambia el set-point de una referencia menor a otra inmediatamente mayor para el sistema con controladores PID y Fuzzy. Acá se observa que el PID trabaja de mejor forma a como lo hizo de bajada pero no alcanza el set-point para 60cm, aunque la planta intenta alcanzarla; logrando llegar un poco a la referencia pero cayendo inmediatamente. El Fuzzy al igual que la forma en que baja se comporta de manera eficiente llegando a la referencia rápidamente e intenta estabilizarse de igual. Las respuestas dadas por el controlador difuso son, en esta aplicación, mejores que la del controlador PID, como se muestra en la tabla 4. El sistema es muy difícil de controlar, producto de la gran cantidad de alinealidades y puntos de bifurcación que presenta.

Controlador \ parámetro	% de Error	Tiempo promedio en llegar a la referencia
PID SUBIDA	12.13%	De 2 a 6 seg.
FUZZY SUBIDA	5.44%	De 1 a 4 seg.

Tabla 4: Comparación entre Fuzzy y PID de subida

El control PID da una respuesta satisfactoria, siempre y cuando no tenga que llevar el sistema de alturas superiores a inferiores. Cuando ocurre esto se hacen más evidentes las turbulencias y las alinealidades del proceso, el cuerpo comienza a rotar sobre su eje y se vuelve por consiguiente inestable, lo cual es muy difícil de manejar con el control PID.

CONCLUSIONES Y RECOMENDACIONES

En el presente trabajo de investigación se diseñó y monto un controlador basado en lógica difusa, para controlar la posición en tiempo real de un sistema no lineal bajo la plataforma del software MatLab. El objetivo era comparar la respuesta de una planta de tal característica con un controlador fuzzy, frente a la respuesta de la misma planta pero con un controlador convencional PID.

El desarrollo del controlador difuso fue relativamente sencillo, dado que se contó con una metodología definida que permitió identificar las partes que constituyen la solución del problema a controlar. Así mismo se contó con uno de los softwares más adecuado para desarrollar estrategias de control difuso como lo es MATLAB y sus Toolboxes de Fuzzy Logic y Simulink.

Para implementar la lógica difusa en esta planta, se realizaron las debidas maniobras con, los conjuntos difusos de entrada y salida, la base de reglas y la adecuación del control en tiempo real para los requerimientos del sistema, obteniendo resultados muy satisfactorios que hacen de la lógica difusa una técnica a tener en cuenta para aplicaciones de este tipo en particular y del control inteligente en general.

Para el controlador PID, se tuvo que ajustar de forma distinta ya que en la simulación se utilizó el método del relay y el método de respuesta en frecuencia de Ziegler-Nichols, debido a que la planta presenta un comportamiento no-lineal y el PID convencional utilizado es lineal y los valores de los parámetros del controlador

obtenidos para el modelo dieron una aproximación no tan satisfactoria respecto a los valores finales requeridos en el set point como parte del objetivo de una estrategia de control. La estructura PID ha mostrado empíricamente ofrecer suficiente flexibilidad para dar excelentes resultados en muchas aplicaciones, pero para este caso se presentó una excepción.

Con este trabajo se demuestra, de una manera clara y con un proceso fácil de implementar, la alta efectividad y/o eficacia del control difuso para suplir las no linealidades de estos sistemas y hacer que su comportamiento sea bastante o casi lineal, frente a algunas limitaciones de los controladores clásicos PID que hacen que sean menos óptimos para esta clase de sistemas.

Dentro de las nuevas ramas del control inteligente, los métodos difusos son los más fáciles de implementar y de comprender, no requiriendo sofisticados algoritmos de programación y con resultados excelentes.

BIBLIOGRAFIA

- Grantham y Vicent: “Sistema de Control Moderno, Análisis y Diseño”, Editorial Limusa, México 1998.
- Benjamín C. Kuo: “Automatic Control System”. Parte 7. Editorial Prentice-Hall, New Jersey 1995.
- Dale E. Seborg; Thomas F. Edgar; Duncan A. Mellichamp: “Process Dynamics and Control”. Second edition. Editorial John Wiley & Sons, EEUU 2004.
- Eronini-Umez-Eronini: “Dinamica de Sistemas y Control”. Editorial Thomson, 2001
- Jesé E. Rodríguez Avila: “Introducción a la Ingeniería del Control Automático”. Editorial McGraw-Hill, Mexico 1998.
- B. Wayne Bequette: “Process Dynamics. Modeling, Analysis and Simulation”. Editorial Prentice Hall International Series, New Jersey 1998
- Katsuhiko Ogata: “Ingeniería de Control Moderno”. Tercera edición. Editorial Prentice Hill, México 1998.
- Distefano, Stubberud y Williams: “Retroalimentación y Sistemas de Control”. Segunda edición. Editorial McGraw-Hill, Santa fe de Bogotá 1992.
- Smith, Carlos A. Corripio. “Control Automático de Procesos. Teoría y Práctica”. Limusa Noriega Editores. 1996
- Franklin, Gene. Powell, David. Emami-Naeine, Abbas. Control de Sistemas Dinámicos con Retroalimentación. Addison-Wesley Iberoamericana. 1991.

- Bonifacio Martín del Brio, Alfredo Sanz Molina. “Redes Neuronales y Sistemas Difusos”. Segunda edición. Editorial Alfaomega, México 2002.

Sitios Web

<http://www.fuzzytech.com>

<http://www.ortech-engr.com/fuzzy/tilshell.html>

<http://www.mathworks.com/products/fuzzylogic/>

<http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repositori/ai/areas/fuzzy/0.html>

<http://www.controlautomatico.net>

http://csd.newcastle.edu.au/control/SpanishPages/clase_slides_download/C07.pdf

http://www.ece.osu.edu/~passino/lab_pres_Spanish.pdf

<http://www.engin.umich.edu/group/ctm/examples/motor2/PID2>.

http://www.brewerscience.com/cee/otherprods/cee_pid.

<http://www.jashaw.com/pid/description>

http://www.aromat.com/acsd/product/plc/fp_sigma/pid/sigma_pid.

<http://www.jashaw.com/pid/tutorial/pid4>

<http://lorien.ncl.ac.uk/ming/pid/PID.pdf>

<http://computing.breinestorm.net/fuzzy%2Bpid%2Bcontrollers%2Bcontroller>

Papers

- Control de un sistema no lineal, en la pagina http://csd.newcastle.edu.au/control/SpanishPages/clase_slides.com
- Real Time WorkShop, Carlos Mario Velez, Universidad EAFIT.

ANEXO A

ECUACION DEL SENSOR

Para encontrar la ecuación del sensor se tomaron los diferentes datos del sensor, en escala de 1 para tener los datos de 0-100cm con mayor precisión. Con estos valores del sensor se obtiene la curva de la misma, graficando estos valores en MatLab. También se calcula la ecuación de aproximación del sensor, y se pueden visualizar las diferentes curvas, para realizar las comparaciones entre ellas, para eso se tiene el siguiente programa:

```
Clc
clear all
%****CONTROL DE POSICION DE UN SISTEMA NO LINEAL CON LOGICA DIFUSA****
%****ENRIQUE DIAZGRANADO, JHON JAIRO MONTALVO
%****PROGRAMA PARA CALCULAR LA ECUACION DEL SENSOR****
v=[0.41 0.44 0.47 0.48 0.49 0.496 0.496 0.5 0.51 0.523 0.527 0.53 0.535 0.566 0.571 0.58 0.586
0.587 0.587 0.588 0.588 0.588 0.588 0.59 0.592 0.6 0.62 0.629 0.65 0.657 0.672 0.68 0.684 0.7
0.71 0.726 0.729 0.73 0.73 0.73 0.73 0.73 0.74 0.767 0.792 0.8 0.8 0.8 0.8 0.81 0.82 0.82 0.83
0.85 0.86 0.87 0.88 0.89 0.9 0.93 0.96 0.99 1.01 1.04 1.07 1.09 1.092 1.11 1.13 1.16 1.18 1.21
1.24 1.27 1.3 1.33 1.39 1.44 1.47 1.5 1.53 1.57 1.61 1.65 1.69 1.73 1.77 1.83 1.88 1.95 2.02 2.07
2.11 2.16 2.21 2.23 2.24 2.25 2.24 2.22 2.2 2.18 2.18 2.22 2.25 2.35];%Valores de voltaje
suministrados por el sensor
d=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105];%Distancias de alcance del sensor
figure,plot(d,v)%Grafica de Distancia Vs Voltaje del sensor (Cada voltaje le corresponde
una respectiva distancia)
title('Gráfica de sensor con valores Prácticos.')
```

xlabel('Distancia (cm)')

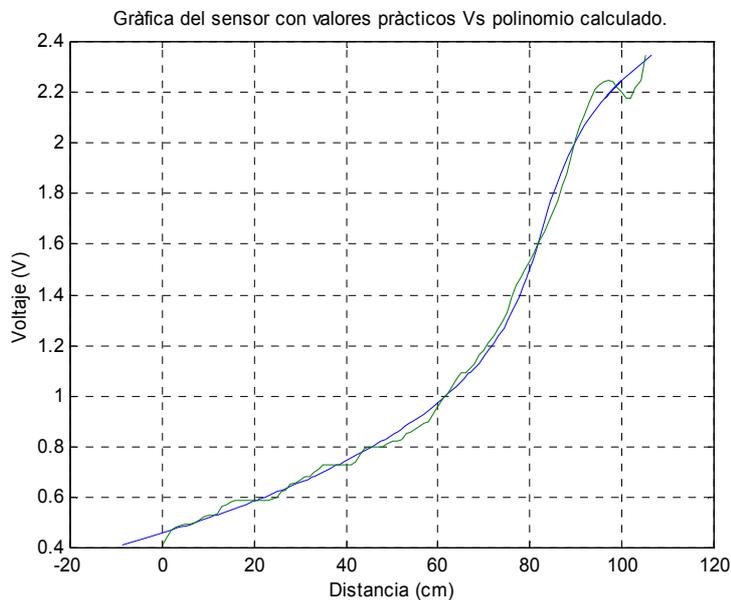
ylabel('Voltaje (V)')

grid

pause

[p,t]=polyfit(v,d,3)%Comando para realizar el calculo de la ecuación del polinomio

```
a=v.^3*p(1)+v.^2*p(2)+v*p(3)+p(4)%sintaxis para el calculo del polinomio
figure,plot(a,v)%Grafica del polinomio calculado
title('Gràfica del polinomio calculado para el sensor.')
xlabel('Distancia (cm)')
ylabel('Voltaje (V)')
grid
pause
figure,plot(a,v,d,v)%Grafica del sensor con los datos practicos y con el polinomio calculado.
title('Gràfica del sensor con valores pràcticos Vs polinomio calculado.')
xlabel('Distancia (cm)')
ylabel('Voltaje (V)')
grid
pause
a1=36.1198*v.^3-183.2488*v.^2+319.3958*v-105.3565%Ecuaciòn del polinomio calculado
ajustado.
figure,plot(a1,v,a,v)%Grafica del polinomio ajustado y con el polinomio calculado.
title('Gràfica del sensor con polinomio ajustado Vs polinomio calculado.')
xlabel('Distancia (cm)')
ylabel('Voltaje (V)')
grid
pause
figure,plot(a1,v,d,v)%Grafica del sensor con los datos practicos y con el polinomio ajustado.
title('Gràfica del sensor con valores pràcticos Vs polinomio ajustado.')
xlabel('Distancia (cm)')
ylabel('Voltaje (V)')
grid
pause
```



Gràfica : ecuaciòn de Sensor Vs ecuaciòn de aproximaciòn.

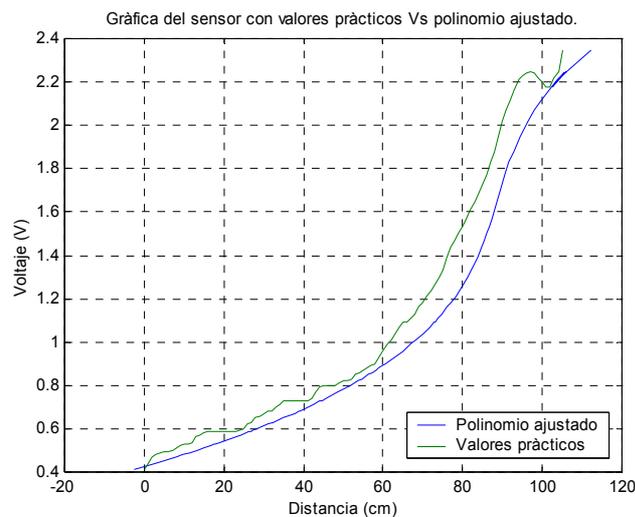
La gráfica nos muestra que el polinomio calculado es muy parecido a la grafica del sensor original, dicho polinomio es el siguiente:

$$d = 36.9198V^3 - 183.2488V^2 + 319.3958V - 111.4251$$

Donde : d se expresa en distancia (cm) y V se expresa en voltaje (V)

Como se observa en la grafica, esta no parte desde cero y por lo tanto esta presenta un error a la hora, de controlar sistema en Tiempo Real, para esto se realiza unos ajustes en el polinomio para que no se presente demasiado error en el sistema.

Después de ajustar la ecuación a los requerimientos, se tiene la grafica:



Gráfica: Curva del sensor Vs curva de aproximación

Ahora se obtiene la ecuación del sensor ajustada, con la que se trabaja en el sistema en Real Time.

$$d = 36.1198V^3 - 183.2488V^2 + 319.3958V - 105.3565$$

Donde : d se expresa en distancia (cm) y V se expresa en voltaje (V)

ANEXO B

Diseño del Actuador

Circuito a Diseñar (Ver Figura):

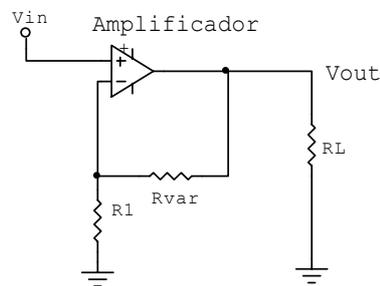


Figura 14: Circuito amplificador

Calculando se tiene:

$$\frac{V_{in}}{R1} + \frac{V_{in} - V_{out}}{R_{var}} = 0$$

Como se tiene una resistencia variable (R_{var}), se escoge un valor para $R1=10K \Omega$

y se calcula R_{var} , despejando de la ecuación:

$$R_{var} = -\frac{V_{in}}{R1}(V_{in} - V_{out}), \quad \text{reemplazando los valores queda que:}$$

$$R_{var} = -\frac{7V}{26\Omega}(7V - 13V) = 8.57K\Omega$$

Esta resistencia R_{var} se ajusta para obtener el valor deseado de voltaje, entonces

$R_{var}=9.7K \Omega$

La ganancia del circuito es: $H = \frac{V_{out}}{V_{in}} = 1.8 \approx 2$

Se escogió el amplificador de potencia TDA2040A, que debe ser alimentado con 15V DC y que cumple con estas especificaciones de diseño (Ver Anexo B).

También se tiene una etapa de un circuito en configuración seguidor de tensión con resistencia de compensación (Ver Figura 15), las premisas de diseño para este son:

Ganancia unitaria $H=1$ $H = \frac{V_{out}}{V_{in}}, \Rightarrow V_{out} = V_{in}$

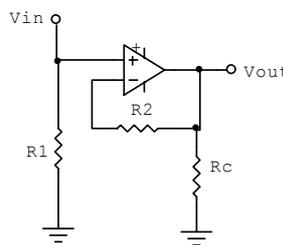


Figura 15: Amplificador seguidor.

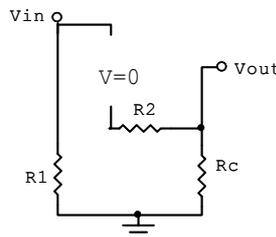


Figura 16: Circuito Equivalente.

$$V_{in} + \frac{V_{in} * R2}{R1} - V_{out} = 0, \quad \text{pero como } H = -\frac{V_{out}}{V_{in}}, \quad \text{Reemplazando en la ecuacion queda :}$$

$$V_{in} + \frac{V_{in} * R2}{R1} - (-H * V_{in}) = 0, \quad \Rightarrow \quad 2V_{in} + \frac{V_{in} * R2}{R1} = 0$$

Por lo que se tiene que:

$R2 = 2R1$, asumiendo $R1=2.2K\Omega$, entonces $R2=4.4K\Omega$, se escoge $R2=4.7K\Omega$, para compensación del circuito se escoge $R3=10R2=47K\Omega$.

El amplificador Operacional utilizado es el Lm358 (Ver Anexo C), debido a que este puede ser alimentado con 15V, porque la misma fuente de alimentación del TDA 2040 nos sirve para alimentarlo.