



## **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**



**“DISEÑO DE UN ALGORITMO DE CONTROL INTELIGENTE UTILIZANDO LÓGICA DIFUSA COMPARADO CON UN ALGORITMO PID, PARA SU IMPLEMENTACIÓN EN UN SISTEMA MULTIVARIABLE NO LINEAL BOLA Y PLATO”**

### **TRABAJO DE TITULACIÓN**

Previo a la obtención del Título de:

**MAGISTER EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL**

Presentado por:

Ing. Victor Manuel Huilcapi Subía

Guayaquil – Ecuador

2015

## **AGRADECIMIENTO**

Agradezco a Dios Padre Celestial por otorgarme la vida y la salud para poder culminar con éxito esta etapa de formación académica, a mi esposa Elizabeth, a mi hija Doménica por sus comprensiones, e impulso para ser mejor en todos los aspectos de mi vida, a mis padres quienes siempre me ofrecen sabios consejos, a mis hermanos y amigos, a mi tutor el Ph.D. Wilton Agila quien siempre estuvo presto a brindarme su asesoraría, así como también a las autoridades de la Universidad Politécnica Salesiana, quienes me han apoyado económicamente para cursar este programa de posgrado, así como me han permitido el uso de su infraestructura tecnológica.

## **DEDICATORIA**

Dedico este trabajo, fruto de mucho esfuerzo, sacrificio, y amor, a mi esposa Elizabeth, y a mi hija Doménica, así como también a mis padres María Rosario y Victor Manuel

## TRIBUNAL DE SUSTENTACIÓN

---

**MSc. Sara Ríos O**

**SUB-DECANA DE LA FIEC**

---

**PhD. Wilton Agila**

**DIRECTOR**

---

**MSc. Efrén Herrera**

**VOCAL PRINCIPAL**

---

**MSc. Carlos Valdivieso**

**VOCAL SUPLENTE**

## **DECLARACIÓN EXPRESA**

La responsabilidad del contenido de este Trabajo de titulación me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral.

(Reglamento de Graduación de la ESPOL).

---

Ing. Victor Huilcapi Subía

## RESUMEN

En este trabajo de titulación se presenta el diseño e implementación de un algoritmo de control inteligente basado en lógica difusa para hacer frente a la inestabilidad y complejidad en el modelado matemático que presenta la plataforma electromecánica del sistema bola y plato 33-240. El algoritmo diseñado garantiza el control de estabilización de los ángulos de inclinación alfa y beta de la plataforma electromecánica, a través de la regulación de los niveles de voltaje y corriente aplicados a los inductores que a su vez realizan la levitación de la plataforma.

El algoritmo de control difuso es validado a través de dos escenarios, en modo manual mediante la aplicación de una señal de referencia tipo escalera en donde se emula la posición de la bola sobre la plataforma en los dos ejes de inclinación de manera independiente, y, en modo automático, mediante una onda cuadrada periódica que permite analizar la robustez del controlador al estabilizar la bola en una determinada posición sobre la plataforma.

Previo al diseño del controlador difuso se hizo necesario desarrollar un algoritmo que permita caracterizar las variables de voltaje aplicado a las bobinas y los ángulos de inclinación que experimenta la plataforma. Para evaluar la eficiencia del algoritmo de control difuso con otro tipo de controlador, se establecieron comparaciones de rendimiento con algoritmos de control convencional PID y PD desarrollados en este trabajo.

La identificación matemática de la plataforma electromecánica del sistema bola y plato 33-240 se realiza empleando los principales modelos de identificación paramétrica, como son: ARX, ARMAX, y BJ. Las comparaciones de rendimiento establecidas para los modelos de identificación empleados, confirman que la identificación paramétrica ARX brinda en todas las pruebas realizadas un modelo satisfactorio para la plataforma.

Todos los algoritmos citados anteriormente e implementados en LabView, permiten la exportación de los datos de las variables medidas hacia la herramienta matemática MATLAB, el mismo que permite realizar los análisis pertinentes para establecer conclusiones y recomendaciones del presente trabajo.

Finalmente, se proponen tres prácticas a realizar por los estudiantes de pre- o postgrado, las mismas que involucran a cada uno de los algoritmos implementados en el presente trabajo.

**Palabras claves:** algoritmo de control difuso, PID, PD, identificación matemática, caracterización del sistema.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIA .....	III
TRIBUNAL DE SUSTENTACIÓN .....	IV
DECLARACIÓN EXPRESA .....	V
RESUMEN .....	VI
ÍNDICE GENERAL.....	VIII
ABREVIATURAS Y SIMBOLOGÍA .....	XII
ÍNDICE DE FIGURAS.....	XVI
ÍNDICE DE TABLAS .....	XXV
INTRODUCCIÓN.....	XXVIII
CAPÍTULO I.....	1
ANÁLISIS DEL PROBLEMA.....	1
1.1 Planteamiento del problema .....	1
1.2 Solución Propuesta.....	1
1.3 Delimitación del problema.....	2
1.4 Objetivos.....	3
1.4.1 Objetivo General .....	3
1.4.2 Objetivos Específicos .....	3
1.5 Metodología .....	4
CAPÍTULO 2.....	5
MARCO TEÓRICO .....	5



2.1	Sistemas en lazo abierto y cerrado.....	5
2.2	Identificación de sistemas en lazo cerrado .....	6
2.2.1	Consideraciones para la identificación en lazo cerrado .....	7
2.2.2	Método de identificación.....	8
2.2.3	Proceso de identificación en lazo cerrado.....	9
2.3	Sistemas de Control PID.....	10
2.3.1	Estructura de un PID .....	11
2.3.2	Acciones de control de un PID .....	11
2.3.3	Ajuste de acciones de control de un PID.....	14
2.4	Sistemas de control difuso.....	18
2.4.1	Funciones de membresía.....	20
2.4.2	Interfaz de fusificación.....	24
2.4.3	Base de conocimientos .....	24
2.4.4	Lógica de decisiones.....	25
2.4.5	Interfaz de defusificación.....	25
2.5	Sistemas de control digital .....	26
2.5.1	Muestreador y retenedor (S&H) .....	27
2.5.2	Convertidor analógico-digital (A/D).....	27
2.5.3	Convertidor digital-analógico (D/A).....	28
2.5.4	Planta.....	28
2.6	Transductor.....	28
2.7	LabView, herramientas de diseño de controladores .....	29

2.7.1	Diseño de un controlador PID .....	29
2.7.2	Diseño de un controlador difuso.....	31
CAPÍTULO 3.....		36
DESCRIPCIÓN DEL ESCENARIO .....		36
3.1	Sistema Bola y Plato 33-240.....	36
3.2	Modelo no lineal del sistema Bola y Plato 33-240 .....	38
3.3	Linealización del sistema bola y plato 33-240.....	43
CAPÍTULO 4.....		45
DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS.....		45
4.1	Algoritmo de caracterización de la planta .....	45
4.2	Algoritmo de control difuso .....	51
4.3	Algoritmo de Control PID .....	66
4.4	Algoritmo de Identificación.....	69
CAPÍTULO 5.....		78
ANÁLISIS Y VALIDACIÓN DE RESULTADOS.....		78
5.1	Prueba de funcionamiento del algoritmo de control difuso .....	78
5.1.1	Pruebas en modo manual .....	78
5.1.2	Pruebas en modo automático .....	86
5.2	Prueba de funcionamiento del algoritmo de control PID.....	94
5.2.1	Pruebas en modo manual .....	94
5.2.2	Pruebas en modo automático .....	101
5.3	Análisis comparativo de algoritmos de control.....	108

5.3.1	Análisis para el ángulo de inclinación alfa.....	109
5.3.2	Análisis para el ángulo de inclinación beta.....	114
5.4	Análisis comparativo de modelos de identificación.....	118
	CONCLUSIONES .....	122
	RECOMENDACIONES.....	124
	BIBLIOGRAFÍA.....	125
	ANEXOS.....	126

## ABREVIATURAS Y SIMBOLOGÍA

$u(t)$	Acción de control
A/D	Analógico digital
$\alpha(t), \beta(t)$	Ángulos de inclinación del plato
ARX	Auto-Regresive with external input
	Autoregressive Moving Average with
ARMAX	external input
BJ	Box Jenkins
PC	Computador personal
$K_p$	Constante proporcional
C	Controlador
g	Constante gravitacional
$k_c$	Constante
$k$	Constante
D/A	Digital analógico
T	Energía cinética
W	Energía potencial
$e(t)$	Error en estado estable
F	Fuerzas externas
GLA	Ganancia en lazo abierto
GLC	Ganancia en lazo cerrado

$K_c$	Ganancia crítica
Hz	Hertz
VI	Instrumento virtual
IA	Inteligencia artificial
J	Joule
Kg	Kilogramo
L	Lagrangiano
FL	Lógica difusa
$m_p$	Masa del plato
m	Masa de la bola
m	Metro
m/s	Metros sobre segundos
MCDG	Método de centro de gravedad
m/s <sup>2</sup>	Metro sobre segundos cuadrados
LS	Mínimos cuadrados
$I_b$	Momento de inercia de la bola
$I_p$	Momento de inercia del plato
MIMO	Múltiples entradas múltiples salidas
Nm	Newton-metro
$P_c$	Período crítico
x	Posición de la bola en dirección x

$y$	Posición de la bola en dirección $y$
PID	Proporcional + Integral + Derivativo
PD	Proporcional+ derivativo
PCI	Peripheral Component Interconnect
P	Proporcional
rad	Radianes
rad/s	Radianes sobre segundos
$r(t)$	Ruido
S&H	Sample and Hold
PRBS	Secuencia binaria pseudo aleatoria
$r(t)$	Señal de perturbación
$m(t)$	Señal de entrada
SP	Set point
Seg	Segundos
B&P	Sistema Bola y Plato
SISO	Simple input simple output
$T_i$	Tiempo integral
$T_d$	Tiempo derivativo
$\tau_1$	Torque 1
$\tau_2$	Torque 2

UPS	Universidad Politécnica Salesiana
	Velocidad angular del plato en
$\dot{\alpha}$	dirección x
	Velocidad angular del plato con
$\dot{\beta}$	respecto al eje y
$\dot{x}$	Velocidad de la bola en dirección x
$\dot{y}$	Velocidad de la bola en dirección y
ZN	Ziegler y Nichols

## ÍNDICE DE FIGURAS

Figura 2.1 Diagrama en bloques de un sistema de control en lazo abierto y lazo cerrado [2].....	6
Figura 2.2 Identificación de un sistema en lazo cerrado, [9] .....	9
Figura 2.3 Lazo de control PID de un sistema SISO. [2] .....	11
Figura 2.4 Diagrama en bloques de un sistema de control en lazo abierto y lazo cerrado [3].....	16
Figura 2.5 Respuesta al escalón de la planta, curva de reacción. [3] .....	17
Figura 2.6 Diagrama de bloques para desarrollar la metodología de lógica difusa [4].....	20
Figura 2.7 Función de saturación [4] .....	21
Figura 2.8 Función hombro [4] .....	22
Figura 2.9 Función triangular. [4].....	23
Figura 2.10 Función trapecio. [4].....	23
Figura 2.11 Función sigmoideal. [4] .....	24
Figura 2.12 Lógica de decisiones de un controlador difuso. [4].....	25
Figura 2.13 Diagrama en bloques de un sistema de control digital [5] .....	26
Figura 2.14 Ruta para el acceso a las herramientas de control, diseño, y simulación de sistemas. [7] .....	30
Figura 2.15 Esquemas PID de Labview, [7] .....	30
Figura 2.16 Descripción y programación de un PID de Labview, [7].....	31
Figura 2.17 Ruta para el ingreso al diseñador de sistemas difusos. [7] ....	32



Figura 2.18 Diseñador de sistemas difusos. [7].....	32
Figura 2.19 Diseño de funciones de membresía. [7].....	33
Figura 2.20 Creación de reglas de un sistema difuso. [7] .....	34
Figura 2.21 Superficie de control de un sistema difuso. [7].....	34
Figura 2.22 Controlador difuso diseñado en LabView. [7].....	35
Figura 3.1 Diagrama de bloques del sistema bola y plato [1].....	36
Figura 3.2 Sistema de control bola y plato. [9] .....	38
Figura 3.3 Diagrama vectorial del sistema bola y plato. [9] .....	39
Figura 3.4 Diagrama en bloques del modelo del sistema bola y plato [9] .	43
Figura 3.5 Diagrama en bloques del modelo del sistema bola y plato [9].	45
Figura 4.1 Diagrama de bloques del algoritmo de caracterización de la planta .....	46
Figura 4.2 Modo visual del algoritmo para caracterizar Voltaje Vs Ángulo de la planta.....	47
Figura 4.3 Etapa principal del algoritmo de caracterización de la planta...	49
Figura 4.4 Función de transferencia Ángulo Vs Voltaje del plato .....	49
Figura 4.5 Gráficas de ángulo y voltaje en función del tiempo .....	50
Figura 4.6 Programa en Matlab para generar la función de transferencia del ángulo alfa Vs voltaje .....	51
Figura 4.7 Diagrama de bloques del algoritmo de control difuso para la planta .....	52
Figura 4.8 Modo visual del algoritmo de control difuso de la planta.....	53

Figura 4.9 Programación gráfica de las funciones de membresía de alfa y beta .....	54
Figura 4.10 Ruta para el acceso al diseñador de sistemas difusos de LabView .....	54
Figura 4.11 Funciones de membresía del ángulo alfa y voltaje para el algoritmo de control difuso .....	55
Figura 4.12 Asistente para creación y edición de funciones de membresía de ángulo alfa para el algoritmo de control difuso propuesto .....	57
Figura 4.13 Asistente para creación y edición de funciones de membresía de voltaje para el algoritmo de control difuso. ....	57
Figura 4.14 Reglas de inferencia para el algoritmo de control difuso.....	59
Figura 4.15 Superficie de control del algoritmo difuso Ángulo alfa Vs voltaje .....	60
Figura 4.16 Funciones de membresía del ángulo beta y voltaje para el algoritmo de control difuso .....	61
Figura 4.17 Programación para el desplazamiento de las funciones de membresía del algoritmo de control difuso diseñado. ....	64
Figura 4.18 Programación para seleccionar el modo de operación del algoritmo de control.....	65
Figura 4.19 Diagrama de bloques del algoritmo PID para la planta .....	66
Figura 4.20 Modo visual del algoritmo de control PID de la planta.....	67
Figura 4.21 Programación para guardar datos de las variables del PID ...	67

Figura 4.22 Programación del PID para el control de ángulo .....	68
Figura 4.23 Diagrama de bloques del algoritmo de identificación de la planta en lazo cerrado.....	69
Figura 4.24 Modo visual del algoritmo de identificación de la planta.....	71
Figura 4.25 Respuesta de la planta ante una entrada PRBS.....	71
Figura 4.26 Primera etapa del algoritmo para la identificación de la planta ..	73
Figura 4.27 Segunda etapa del algoritmo para la identificación de la planta ..	74
Figura 4.28 Tercera etapa del algoritmo para la identificación de la planta ..	75
Figura 4.29 Primer escenario de identificación de la planta .....	76
Figura 5.1 Respuesta del ángulo alfa ante entradas tipo paso, algoritmo difuso, modo manual .....	79
Figura 5.2 Respuesta del ángulo beta para fijarse en cero grados, algoritmo difuso, modo manual .....	80
Figura 5.3 Esfuerzo de control para el ángulo alfa, algoritmo difuso, modo manual .....	81
Figura 5.4 Esfuerzo de control para el ángulo beta, algoritmo difuso, modo manual .....	81
Figura 5.5 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador difuso, modo manual .....	82

Figura 5.6 Respuesta del ángulo beta ante entradas tipo paso, algoritmo difuso, modo manual .....	83
Figura 5.7 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo difuso, modo manual .....	84
Figura 5.8 Esfuerzo de control para el ángulo beta variable, algoritmo difuso, modo manual .....	85
Figura 5.9 Esfuerzo de control para el ángulo alfa fijo, algoritmo difuso, modo manual.....	85
Figura 5.10 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador difuso, modo manual.....	86
Figura 5.11 Respuesta del sistema ante una entrada cuadrada, ángulo alfa, algoritmo difuso, modo automático.....	87
Figura 5.12 Respuesta del ángulo beta para fijarse en cero grados, algoritmo difuso, modo automático.....	88
Figura 5.13 Esfuerzo de control ángulo alfa variable, algoritmo difuso, modo automático.....	89
Figura 5.14 Esfuerzo de control ángulo beta fijo, algoritmo difuso, modo automático.....	89
Figura 5.15 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador difuso, modo automático .....	90
Figura 5.17 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo difuso, modo automático.....	92

Figura 5.18 Esfuerzo de control ángulo beta variable, algoritmo difuso, modo automático.....	92
Figura 5.19 Esfuerzo de control ángulo alfa fijo, algoritmo difuso, modo automático.....	93
Figura 5.20 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador difuso, modo automático .....	93
Figura 5.21 Respuesta del ángulo alfa ante entradas tipo paso, algoritmo PID, modo manual.....	94
Figura 5.22 Respuesta del ángulo beta para fijarse en cero grados, algoritmo PID, modo manual.....	95
Figura 5.23 Esfuerzo de control para el ángulo alfa variable, algoritmo PID, modo manual.....	96
Figura 5.24 Esfuerzo de control para el ángulo beta fijo, algoritmo PID, modo manual.....	96
Figura 5.25 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador PID, modo manual.....	97
Figura 5.26 Respuesta del ángulo beta ante entradas tipo paso, algoritmo PID, modo manual.....	98
Figura 5.27 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo PID, modo manual.....	99
Figura 5.28 Esfuerzo de control para el ángulo alfa, algoritmo PID, modo manual .....	100

Figura 5.29 Esfuerzo de control para el ángulo beta, algoritmo PID, modo manual .....	100
Figura 5.30 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador PID, modo manual .....	101
Figura 5.31 Respuesta del sistema ante una entrada cuadrada, ángulo alfa, algoritmo PID.....	102
Figura 5.32 Respuesta del ángulo beta para fijarse en cero grados, algoritmo PID, modo automático .....	103
Figura 5.33 Esfuerzo de control para el ángulo alfa, algoritmo PID, modo automático.....	104
Figura 5.34 Esfuerzo de control para el ángulo beta, algoritmo PID, modo automático.....	104
Figura 5.35 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador PID, modo automático.....	105
Figura 5.36 Respuesta del sistema ante una entrada cuadrada, ángulo beta, algoritmo PID.....	106
Figura 5.37 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo PID, modo automático .....	107
Figura 5.38 Esfuerzo de control para el ángulo alfa, algoritmo PID, modo automático.....	107
Figura 5.39 Esfuerzo de control para el ángulo beta, algoritmo PID, modo automático.....	108

Figura 5.40 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador PID, modo automático.....	108
Figura 5.41 Comparación de rendimiento de algoritmos de control difuso, PD, PID para el control del ángulo de inclinación alfa.....	109
Figura 5.42 Comparación de rendimiento de algoritmos de control difuso, PD, y PID para el seguimiento de la referencia cero grados del ángulo beta .....	111
Figura 5.43 Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo alfa variable .....	112
Figura 5.44 Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo beta fijo.....	112
Figura 5.45 Programación en Matlab para comparar rendimientos de algoritmos difuso, PID, PD, ángulo alfa.....	114
Figura 5.46 Comparación de rendimiento de algoritmos de control difuso, PD, PID para el control del ángulo de inclinación beta.....	114
Figura 5.47 Comparación de rendimiento de algoritmos de control difuso, PD, y PID para el seguimiento de la referencia cero grados del ángulo beta .....	115
Figura 5.48 Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo beta variable.....	116
Figura 5.49 Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo alfa fijo .....	117

Figura 5.50 Programación en Matlab para comparar rendimientos de algoritmos difuso, PID, PD, ángulo alfa .....	118
Figura 5.51 Identificación de la planta, primer modelo ARX.....	118
Figura 5.52 Identificación de la planta, segundo modelo ARX .....	119
Figura 5.53 Identificación de la planta, tercer modelo ARX.....	119
Figura 5.54 Identificación de la planta, cuarto modelo ARX.....	119
Figura 5.55 Identificación de la planta, quinto modelo ARX .....	120
Figura 5.56 Respuesta de los modelos identificados a una entrada sinusoidal .....	120
Figura 5.57 Programación en Matlab para el cálculo del error de aproximación de los modelos ARX identificados de la planta .....	121



## ÍNDICE DE TABLAS

Tabla 2.1 Parámetros de ajuste, método de oscilación [3].....	15
Tabla 2.2 Parámetros de ajuste, método de la curva de reacción [7].....	17
Tabla 3.1 Parámetros del modelo del plato del sistema. [9].....	40
Tabla 4.1 Parámetros de las funciones de membresía del controlador difuso para el ángulo alfa .....	56
Tabla 4.2 Parámetros de las funciones de membresía del controlador difuso para el voltaje aplicado a los inductores del sistema .....	58
Tabla 4.3 Reglas de inferencia del algoritmo de control difuso para el ángulo alfa.....	59
Tabla 4.4 Parámetros de las funciones de membresía del controlador difuso para el ángulo beta .....	61
Tabla 4.5 Parámetros de las funciones de membresía del controlador difuso para el voltaje aplicado a los inductores del sistema .....	62
Tabla 4.6 Reglas de inferencia del algoritmo de control difuso para el ángulo beta.....	63
Tabla 5.1 Características del ángulo alfa ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual.....	79
Tabla 5.2 Características del ángulo beta para una referencia fija de cero grados, algoritmo de control difuso, modo manual.....	80
Tabla 5.3 Características del ángulo beta ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual.....	83

Tabla 5.4 Características del ángulo beta para una referencia fija de cero grados, algoritmo de control difuso, modo manual.....	84
Tabla 5.5 Características del ángulo alfa para una referencia de entrada tipo onda cuadrada, algoritmo de control difuso.....	87
Tabla 5.6 Características del ángulo beta para una referencia de entrada tipo onda cuadrada, algoritmo de control difuso, ángulo beta variable.....	91
Tabla 5.7 Características del ángulo alfa ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual.....	94
Tabla 5.8 Características del ángulo beta para una referencia fija de cero grados, algoritmo PID, modo manual.....	95
Tabla 5.9 Características del ángulo beta ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual.....	98
Tabla 5.10 Características del ángulo alfa para una referencia fija de cero grados, algoritmo PID, modo manual.....	99
Tabla 5.11 Características del ángulo alfa para una referencia de entrada tipo onda cuadrada, algoritmo de control PID.....	102
Tabla 5.12 Características del ángulo beta para una referencia fija de cero grados, algoritmo PID, modo automático.....	103
Tabla 5.13 Características del ángulo beta para una referencia de entrada tipo onda cuadrada, algoritmo de control PID.....	106
Tabla 5.14 Características del ángulo alfa para una referencia fija de cero grados, algoritmo PID, modo automático.....	107

Tabla 5.15 Características del rendimiento de los algoritmos difuso, PD, PID, ángulo alfa.....	110
Tabla 5.16 Características del rendimiento de los algoritmos difuso, PD, PID, ángulo beta.....	114
Tabla 5.17 Errores relativos y RMS de los modelos identificados de la planta .....	120

## INTRODUCCIÓN

En teoría de control, los sistemas o plantas físicas de experimentación refuerzan el aprendizaje teórico a los docentes y estudiantes. En este contexto la Universidad Politécnica Salesiana (UPS) adquirió varias de estas plantas para el dictado y experimentación de la teoría de control clásico. Sin embargo, el desarrollo tecnológico actual hace necesario la implementación de algoritmos de control moderno, que permitan ampliar los conocimientos y experiencias por parte de la comunidad académica de la UPS sede Guayaquil.

En este sentido, se hace uso del sistema Bola y Plato 33-240 para la implementación de algoritmos de control moderno. La planta Bola y Plato 33-240 es un sistema mecatrónico fabricado por la empresa Feedback que permite el estudio, análisis e implementación de teorías de control avanzado.

El sistema está formado principalmente por cuatro inductores que actúan como pilares los cuales generan el campo electromagnético que permite la levitación de la plataforma electromecánica del sistema; una plataforma móvil con anillos de aluminio en sus extremos y que encajan en cada inductor; dos potenciómetros sujetos sobre la plataforma que actúan como sensores para indicar el ángulo de inclinación de la misma sobre dos ejes transversalmente opuestos; y, una cámara que permite ubicar una bola en el plato. El sistema se conecta a un computador a través de la tarjeta PCI 6251 [1].

En el primer capítulo se analiza el problema de investigación a resolver en todas sus dimensiones. Los principales fundamentos teóricos concernientes al desarrollo de sistemas de control automático son explicados en el capítulo dos los cuales son necesarios para enfrentar y buscar la solución óptima al problema de investigación planteado.

El capítulo tres describe de manera detallada las partes del sistema no lineal bola y plato que serán la base para el desarrollo del trabajo propuesto.

El diseño e implementación de los algoritmos de caracterización, control basado en lógica difusa, e identificación, son presentados en el capítulo cuatro, los cuales son necesarios para la estabilización la plataforma del sistema mecatrónico, y corresponde al núcleo de la investigación

En el capítulo cinco se establecen comparaciones de eficiencia de los algoritmos desarrollados, en este capítulo se hace hincapié en el controlador difuso versus un controlador PID y PD aplicados al sistema bola y plato. Así también se analiza los modelos identificados para la planta. Estos análisis permiten establecer conclusiones y recomendaciones necesarias que son presentadas posterior a este capítulo.

Finalmente, en los anexos se desarrolla una guía de prácticas didácticas concernientes al desarrollo e implementación de los algoritmos para la plataforma electromecánica bola y plato 33-240.

# **CAPÍTULO 1**

## **ANÁLISIS DEL PROBLEMA**

### **1.1 Planteamiento del problema**

En la Universidad Politécnica Salesiana (UPS) con sede en Guayaquil existe un deficiente estudio experimental por parte de docentes y estudiantes de la facultad de Ingenierías en el área de control inteligente en especial en aplicaciones de lógica difusa. A esto se suma el hecho que la UPS adquirió varias plantas para la experimentación de la teoría de control clásica, no existiendo desarrollo de algoritmos de control modernos para la experimentación por parte de la comunidad académica de la sede Guayaquil.

### **1.2 Solución Propuesta**

Para lograr la solución al problema de investigación se propone el diseño de un algoritmo de control moderno usando lógica difusa que

será aplicado al sistema no lineal bola y plato 33-240 que dispone la Universidad Politécnica Salesiana sede Guayaquil, y a su vez analizar su eficiencia al compararlo con un algoritmo proporcional, integral, y derivativo (PID) convencional.

### **1.3 Delimitación del problema**

En la Universidad Politécnica Salesiana con sede en Guayaquil ubicada en las calles robles 107 y Chambers, en el laboratorio de control automático existe un deficiente estudio práctico en el área de control inteligente usando lógica difusa por parte de estudiantes y docentes de ingenierías. Actualmente la experimentación es realizada utilizando solamente simuladores como Matlab y Labview. Por esta razón que surge la necesidad de fortalecer los conocimientos prácticos en los estudiantes de ingenierías de la UPS, a través del diseño e implementación de un algoritmo de control difuso para el módulo didáctico bola y plato 33-240 que dispone la institución.

La investigación es relevante e innovadora ya que se propone diseñar un algoritmo de control difuso para estabilizar la plataforma o plato del sistema en mención, analizar su robustez frente a perturbaciones en el sistema, y comparar su funcionamiento frente a un algoritmo de control clásico PID. Los beneficiarios directos de esta investigación serán los

estudiantes y docentes de las carreras de ingenierías que dispone la UPS.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Diseñar un algoritmo de control utilizando lógica difusa para implementarlo en un sistema multivariable no lineal “bola y plato” y realizar un análisis comparativo de eficiencia con un algoritmo PID convencional.

### **1.4.2 Objetivos Específicos**

1. Elaborar un algoritmo de control usando la herramienta Fuzzy Logic de LabView para un sistema electromagnético bola y plato.
2. Establecer un análisis comparativo entre el controlador diseñado a través de lógica difusa y un controlador PID convencional.
3. Realizar la identificación matemática del sistema Bola y Plato empleando la herramienta system identification de Labview.
4. Utilizar Matlab para efectuar simulaciones del sistema bola y plato y comparar con los resultados obtenidos con Labview.



## 1.5 Metodología

Para el desarrollo y ejecución del tema propuesto se plantea realizar en su análisis teórico los siguientes métodos:

- **Análisis- Síntesis:** Permitirá descomponer el sistema en sus partes básicas a ser estudiadas, para luego caracterizar el sistema.
- **Hipotético-Deductivo:** Debido a que se está empleando la estructura de investigación siguiente: Planteamiento del problema, solución, delimitación etc.

Y en su fase experimental se empleará:

- **Observación:** Permitirá analizar la respuesta y comportamiento temporal del sistema en función de los algoritmos aplicados.
- **Medición:** Porque se tomará información de datos numéricos a través de técnicas, instrumentos y condiciones iniciales.
- **Experimentación:** Se evaluará el desempeño del sistema ante los algoritmos de control aplicados a través de la medición de sus variables medibles que posea el mismo.

## **CAPÍTULO 2**

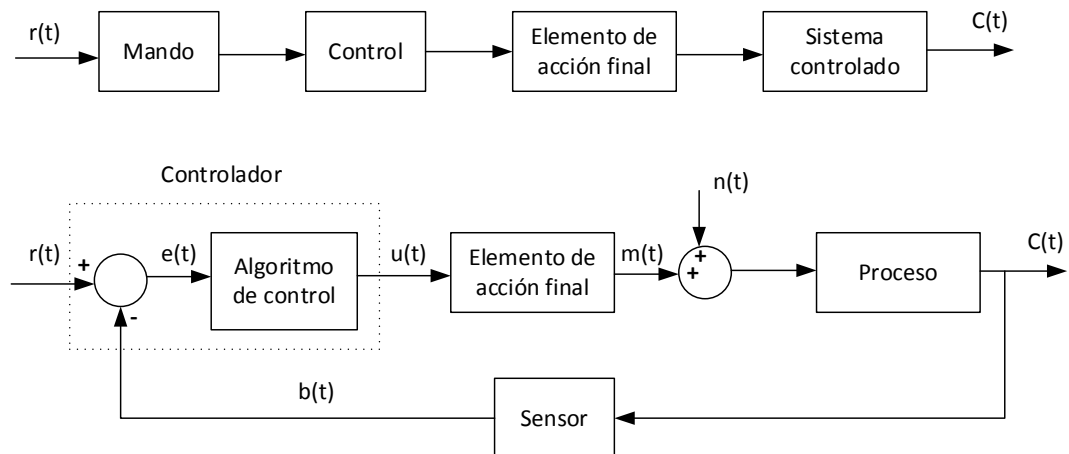
### **MARCO TEÓRICO**

#### **2.1 Sistemas en lazo abierto y cerrado**

En todo sistema de control es fundamental establecer las definiciones de lazo abierto y lazo cerrado, así como sus diferencias.

En [2] se establece que un sistema de control realimentado establece una relación entre la salida y la entrada de referencia, comparandolas y usando la diferencia como medio de control, a este sistema se lo conoce también como sistema de control en lazo cerrado. Los sistemas de control en lazo abierto son aquellos en los cuales la salida no afecta la acción de control, en otras palabras en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada de referencia.

A continuación se describe gráficamente en diagramas de bloques un sistema de control en lazo abierto versus uno en lazo cerrado [2]:



**Figura 2.1 Diagrama en bloques de un sistema de control en lazo abierto y lazo cerrado [2]**

Las señales escritas en la figura 2.1 se describen a continuación:

**$r(t)$** : Señal de referencia, valor deseado, punto de consigna, o set point

**$e(t)$** : Señal de error ;  $e(t) = r(t) - b(t)$

**$u(t)$** : Señal de control

**$m(t)$** : variable manipulada

**$c(t)$** : variable controlada

**$b(t)$** : variable medida

**$n(t)$** : Señal de perturbación

## 2.2 Identificación de sistemas en lazo cerrado

La identificación de un sistema consiste en la determinación de un modelo matemático que represente la dinámica del mismo, a partir de las medidas de las señales de entrada y salida.

En varios casos de la vida real es imposible remover la realimentación de un lazo de control, o se tienen procesos inestables en lazo abierto, aquí se hace necesario identificar el sistema en lazo cerrado. [8]

### **2.2.1 Consideraciones para la identificación en lazo cerrado**

De acuerdo a [9] se debe tener en consideración los siguientes puntos cuando se realiza un proceso de identificación de un sistema de control:

#### **a) Estabilidad de un sistema**

Cuando se desea identificar una planta inestable como lo es el sistema bola y plato en estudio, la identificación debe hacerse trabajando con un controlador, por el contrario si la planta es estable no es necesario el controlador, y la identificación es mucho más sencilla.

#### **b) Selección de la estructura**

Un aspecto muy importante en todo proceso de identificación de sistemas lineales, sean estos continuos o discretos es la elección de la orden de numerador y el denominador de la función de transferencia que lo representa.

### **c) Tiempo de muestreo**

Una correcta selección del tiempo de muestreo dará como resultado un proceso de identificación del sistema óptimo. El tiempo de muestreo no debe ser demasiado pequeño ya que podría influir en la calidad de la identificación debido al error de cuantificación introducido por los convertidores analógicos digitales del sistema de control.

### **d) Señal de excitación**

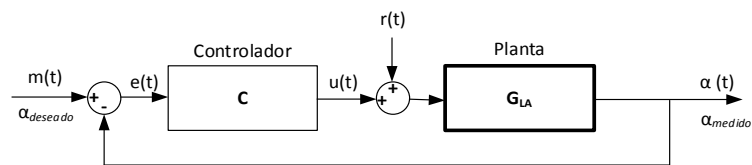
Para los modelos lineales la elección de la señal de excitación es simple, se podría identificar un sistema lineal con una señal de ruido blanco, no obstante en plantas complejas, se debe establecer otro tipo de señales de excitación, entre ellas se podría utilizar una secuencia binaria pseudo aleatoria (PRBS), o una señal que sea la sumaria de varias señales sinusoidales.

## **2.2.2 Método de identificación**

Para realizar la identificación de un sistema usualmente son usados dos métodos, el de mínimos cuadrados (LS), y el método de la variable instrumental. En el presente proyecto de titulación se utilizará el LS, y se lo implementará usando LabView.

### 2.2.3 Proceso de identificación en lazo cerrado

En el siguiente esquema se realizará el análisis matemático del proceso de identificación de un sistema en lazo cerrado.



**Figura 2.2 Identificación de un sistema en lazo cerrado, [9]**

Para identificar un sistema en lazo cerrado se debe establecer un controlador (C), el mismo que regule la respuesta de la planta, y no permite que mientras es excitada por una señal externa pierda estabilidad. Debido a que este controlador puede introducir una correlación entre el ruido de salida y la señal de control, el modelo identificado podría no ser tan preciso, para eliminar esta correlación se introduce una señal de excitación adicional  $r(t)$ , la cual se suma a la señal de control  $u(t)$ . [9].

Para realizar el análisis matemático, se procede a encontrar la función de transferencia en lazo cerrado que la llamaremos  $G_{LC}$ , la cual representa la relación entre  $\alpha(t)$  y  $r(t)$ , para lo cual  $m(t)$  se hace igual a cero. A continuación se detalla:

$$\alpha(t) = G_{LA} [u(t) + r(t)] \quad (2.1)$$

$$u(t) = C.e(t) = -\alpha(t).C \quad (2.2)$$

$$\alpha(t) = G_{LA} [-\alpha(t).C + r(t)] \quad (2.3)$$

$$\alpha(t).(1 + G_{LA}.C) = G_{LA}.r(t) \quad (2.4)$$

$$G_{LC} = \frac{\alpha(t)}{r(t)} = \frac{G_{LA}}{1 + G_{LA}.C} \quad (2.5)$$

Ahora se puede despejar la función de transferencia de la planta en lazo abierto, en términos de su función de transferencia de lazo cerrado, a continuación se muestra:

$$G_{LC} (1 + G_{LA}.C) = G_{LA} \quad (2.6)$$

$$G_{LC} = G_{LA} (1 - G_{LC}.C) \quad (2.7)$$

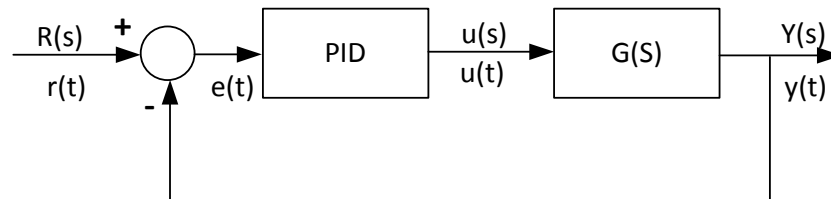
$$\frac{\alpha(t)}{u(t)} = G_{LA} = \frac{G_{LC}}{1 - G_{LC}.C} \quad (2.8)$$

### 2.3 Sistemas de Control PID

A continuación se analizarán algunas familias de controladores Proporcionales, integrales, y derivativos (PID) que nos permitirán tener una visión completa de la temática a abordar.

### 2.3.1 Estructura de un PID

Considérese el siguiente lazo de control para un sistema de una sola entrada y una sola salida (SISO), y de un grado de libertad:



**Figura 2.3 Lazo de control PID de un sistema SISO. [2]**

Un controlador PID, incluye tres acciones: proporcional (P), integral (I), y derivativa (D), así se pueden generar los siguientes controladores: P, PI, PD, y PID. [2]. A continuación se detalla cada una de las acciones:

### 2.3.2 Acciones de control de un PID

#### a) Acción de control proporcional (P)

En este caso la salida del controlador  $u(t)$  es proporcional al error, es

$$\text{decir: } u(t) = K_p \cdot e(t) \quad (2.9)$$

Dicha acción vista en el dominio de Laplace, es decir su función de transferencia es:

$$C_p(s) = K_p \quad (2.10)$$

Donde  $K_p$  es una ganancia proporcional ajustable, este tipo de controlador puede controlar cualquier planta estable.



### b) Acción de control integral (I)

La salida del controlador  $u(t)$  es proporcional al error acumulado, es decir de alguna manera este tipo de control es un poco lento, se lo describe como a continuación se detalla [2]:

$$u(t) = K_i * \int_0^t e(\tau) d\tau \quad (2.11)$$

Su función de transferencia es:

$$C_p(s) = \frac{K_i}{s} \quad (2.12)$$

Este tipo de control dada una referencia constante, y en presencia de perturbaciones permite mantener el error en régimen permanente igual a cero.

### c) Acción de control proporcional- integral (PI)

En este caso se combina la acción proporcional e integral, la ecuación que gobierna este controlador se detalla a continuación:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (2.13)$$

Donde  $T_i$  se denomina tiempo integral, y es justamente el parámetro que ajusta la acción integral. La función de transferencia es como se detalla a continuación:

$$C_{PI}(s) = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (2.14)$$

La acción de control PI es suficiente para muchos procesos cuya dinámica es predominantemente de primer orden. [2].

#### **d) Acción de control proporcional- derivativa (PD)**

La acción de control proporcional derivativa se define por la ecuación:

$$u(t) = K_p e(t) + K_p T_d \cdot \frac{d}{dt} e(t) \quad (2.15)$$

Donde el parámetro  $T_d$  es una constante que se denomina tiempo derivativo. La acción derivativa tiene el objetivo de prevenir, es decir se adelanta al error y lo trata de compensar, esto hace más rápida la acción de control. Es de tener en consideración que la acción de control derivativa no se utiliza por si sola. La función de transferencia es:

$$C_{PD}(s) = K_p + s \cdot K_p T_d \quad (2.16)$$

Se debe resaltar que cuando una acción de control derivativa se agrega a un controlador proporcional, esto permite obtener un control de alta sensibilidad, ya que maneja la velocidad con que cambia el error, y trata de corregirlo antes de que sea demasiado grande y vuelva inestable la planta. [2].

### e) Acción de control proporcional-integral- derivativa (PID)

La acción de control PID agrupa las ventajas de los controladores que se analizaron de manera independiente. La ecuación de un controlador PID es como a continuación se describe:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \cdot \frac{d}{dt} e(t) \quad (2.17)$$

Aplicando la transformada de Laplace a la ecuación anterior obtenemos su función de transferencia [2]:

$$C_{PID}(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2.18)$$

### 2.3.3 Ajuste de acciones de control de un PID

Una vez que se haya seleccionado la estructura del controlador, el siguiente paso es su sintonización, es decir la calibración de sus ganancias proporcional, integral, y derivativas. Para este efecto existen diferentes metodologías, una de las más usadas lo constituye el método de Ziegler y Nichols (ZN). A continuación se detalla esta metodología:

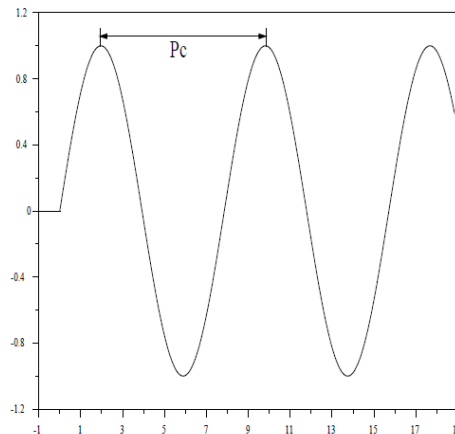
#### a) Método de oscilación

El método de oscilación solamente poder ser utilizado en plantas estables en lazo abierto, para lo cual se deben efectuar los siguientes pasos [3]:

1. Se debe utilizar solo un control proporcional, empezando con un valor de ganancia pequeño, se incrementa la ganancia progresivamente hasta que el lazo comience a oscilar las mismas que deben ser observadas a la salida del controlador.
2. Cuando se obtengan oscilaciones de amplitud constante, se registra la ganancia del controlador  $K_p = K_c$ , a esta ganancia se le llama crítica
3. Se ajusta los parámetros del controlador de acuerdo a la siguiente tabla:

**Tabla 2.1 Parámetros de ajuste, método de oscilación [3]**

	$K_p$	$T_i$	$T_d$
P	$0.505 K_c$		
PI	$0.45 K_c$	$P_c/1.2$	
PID	$0.60 K_c$	$0.5 P_c$	$P_c/8$

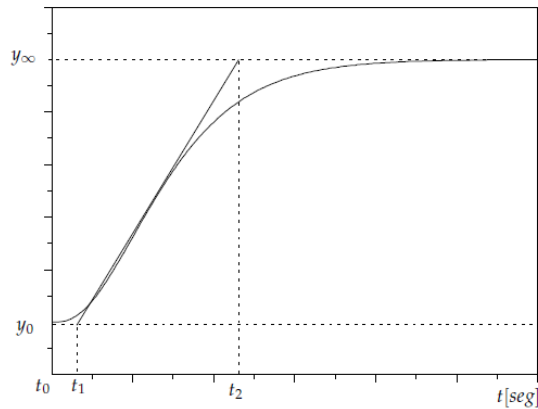


**Figura 2.4 Diagrama en bloques de un sistema de control en lazo abierto y lazo cerrado [3]**

**b) Método basado en la curva de reacción**

Para aplicar este método se deben aplicar los siguientes pasos [3]:

1. Con la planta en lazo abierto, se lleva a esta planta a un punto de operación normal. Por ejemplo la planta se estabiliza en  $y(t) = y_0$  para una entrada constante  $u(t) = u_0$ .
2. En el instante  $t_0$  se aplica un cambio en el escalón de entrada desde  $u_0$  hasta  $u_\infty$ . Esto debería ser en un rango del 10% al 20% del rango completo.
3. Se debe registrar la salida hasta que se estabilice en el nuevo punto de operación, si la curva que se obtiene es parecida a la descrita a continuación, esta curva se llama curva de reacción del proceso.



**Figura 2.5 Respuesta al escalón de la planta, curva de reacción. [3]**

El objetivo de diseño es alcanzar un amortiguamiento de tal forma que exista una relación de atenuación de 4:1 para el primer y segundo pico de la respuesta, cuando se aplica un escalón a la planta. [3]

Los valores de los parámetros del controlador se calculan de acuerdo

a

$$K_0 = \frac{y_\infty - y_0}{y_\infty - u_0} ; \tau_0 = t_1 - t_0 \quad ; \nu_0 = t_2 - t_1 \quad (2.19)$$

**Tabla 2.2 Parámetros de ajuste, método de la curva de reacción [7]**

	$K_p$	$T_i$	$T_d$
P	$\frac{\nu_0}{K_0 \tau_0}$		
PI	$\frac{0.9 \nu_0}{K_0 \tau_0}$	$3\tau_0$	

PID	$\frac{1.2v_0}{K_0\tau_0}$	$2\tau_0$	$0.5\tau_0$
-----	----------------------------	-----------	-------------

## 2.4 Sistemas de control difuso

Es muy provechoso conocer brevemente como se ha venido desarrollando la teoría de control inteligente, y sus aplicaciones en el campo de la lógica difusa.

Pedro Ponce Cruz en su libro de inteligencia artificial resume de muy buena manera así:

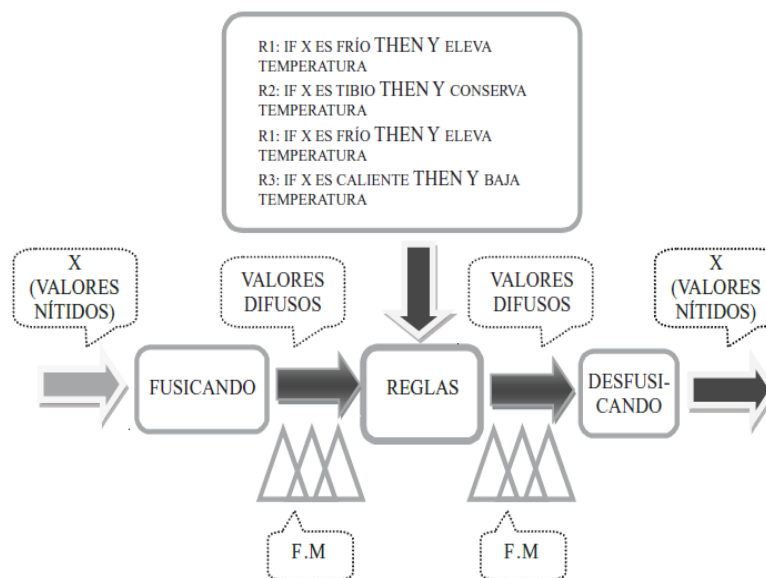
La lógica difusa es una de las ramas de la inteligencia artificial (IA) que le permite a una computadora analizar información del mundo real en una escala entre lo falso y lo verdadero. En 1965 Lotfi A. Zadeh, en aquel entonces director del Departamento de Ingeniería Eléctrica de la Universidad de California en Berkeley, publicó Fuzzy Sets, este artículo describe las matemáticas de los conjuntos difusos y por extensión de la lógica difusa, y este trabajo le dio nombre a su campo. La lógica difusa se aplicó a mediados de la década de 1970 por Ebrahim H. Mamdani en el Queen Mary Collage en Londres. Mamdani diseñó un controlador difuso para un motor a vapor. Desde entonces el término lógica difusa es sinónimo de cualquier sistema matemático o computacional que razona con lógica difusa. La lógica difusa hoy en día es muy común y se halla en diferentes sectores de la tecnología, tanto en la electrónica

como el control, las matemáticas, la robótica, etc. El objetivo principal de la lógica difusa es crear un sistema basado en el comportamiento y pensamiento humanos. La clave de la lógica difusa se basa en la experiencia. El sistema toma el banco de conocimiento del experto, ya sea de mecánica, construcción, fotografía, computación, etc., y con él crea sus reglas para desarrollar una propuesta, [4],[5].

Hay que mencionar que un sistema que trabaja con lógica difusa está compuesto de tres etapas bien definidas, en la primera etapa se toma los valores nítidos del mundo real y se asigna valores difusos en un rango de pertenencia entre cero y uno, es decir variables que tienen un determinado grado de incertidumbre metalingüístico, a este proceso se lo conoce como fusificación. En la segunda etapa se establecen el conjunto de reglas lingüísticas, estas reglas se depuran con la experimentación del sistema. El grado de pertenencia de cada variable lingüística es evaluado sobre un conjunto de reglas de inferencia. La tercera y última etapa toma los valores difusos del fusificador y los convierte en valores nítidos, este proceso es conocido como defusificación, así el sistema interpretará las órdenes y realizará las acciones que el experto haya ideado.

A continuación se muestra un diagrama de lo explicado respecto a un sistema que trabaja con lógica difusa [4]:





**Figura 2.6 Diagrama de bloques para desarrollar la metodología de lógica difusa [4]**

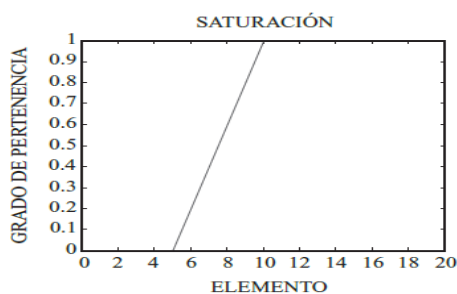
### 2.4.1 Funciones de membresía

Es muy importante la asignación de funciones de membresía en la representación de los grados de pertenencia de cada uno de los elementos que conforman un conjunto difuso.

Por lo tanto en este apartado trataremos las funciones de membresía convencionales que nos permitirá realizar la asociación de los valores nítidos a valores difusos. Es importante conocer que el grado de pertenencia de una función de membresía es un valor en el intervalo de cero a uno. A continuación se describen las funciones de membresías más utilizadas en los sistemas de lógica difusa de acuerdo al texto de inteligencia artificial de Pedro Ponce Cruz

### a) Función de saturación

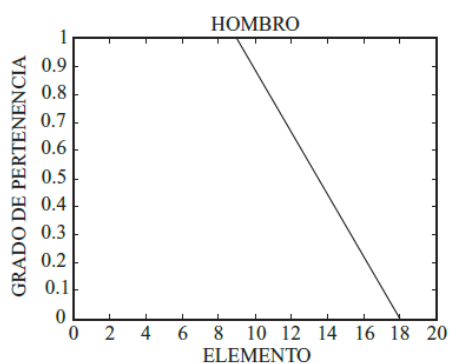
Es una función que tiene asignado el valor de cero hasta un determinado punto, y luego empieza a crecer con pendiente positiva hasta alcanzar el máximo valor en su grado de pertenencia que es la unidad, y de allí en adelante permanece constante. A continuación se muestra [5]:



**Figura 2.7 Función de saturación [4]**

### b) Función de hombro

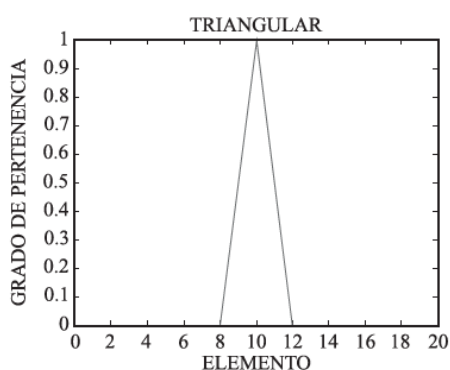
Se puede decir que la función hombro es lo inverso de la función saturación, ya que inicia con un grado de pertenencia igual a la unidad hasta un determinado valor y posteriormente descendiendo con pendiente negativa hasta llegar al valor de cero. [4]. A continuación se muestra:



**Figura 2.8 Función hombro [4]**

### c) Función triangular

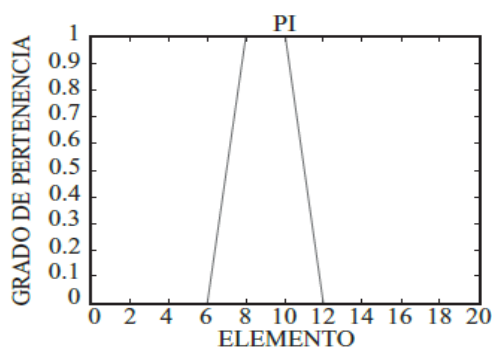
Esta función de acuerdo a su nombre tiene forma de un triángulo, y está formada por una parte con pendiente positiva hasta alcanzar un grado de pertenencia igual a la unidad, y posterior desciende con una pendiente negativa hasta alcanzar un grado de pertenencia de cero, se puede tratar el valor central del triángulo como el valor óptimo, y una vez que el elemento se aleja de este disminuye su grado de pertenencia. [4]. A continuación se muestra el gráfico:



**Figura 2.9 Función triangular. [4]**

**d) Función trapecio**

Se puede decir que la función trapecio es la generalización de la función triangular, en este caso la función de membresía presenta un intervalo de valores para el cual el grado de pertenencia es unitario, y no solo en un valor, el uso de la misma dependerá del fenómeno que se está intentando representar, pero se puede apreciar que existe un intervalo de valores óptimos, y luego de este intervalo, los valores de pertenencia empiezan a disminuir. [4]

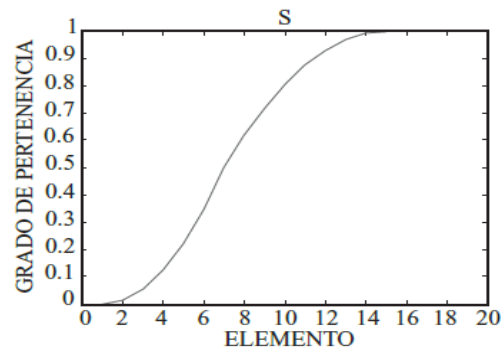


**Figura 2.10 Función trapecio. [4]**

**e) Función sigmoideal**

La función sigmoideal es también conocida como "S" por su forma es muy parecida a la función saturación, con la diferencia que la subida lo hace a través de una curva de segundo orden y no de una recta. Esta

curva de segundo orden cambia su concavidad en un determinado punto y posterior alcanza la saturación.



**Figura 2.11 Función sigmoideal. [4]**

#### **2.4.2 Interfaz de fusificación**

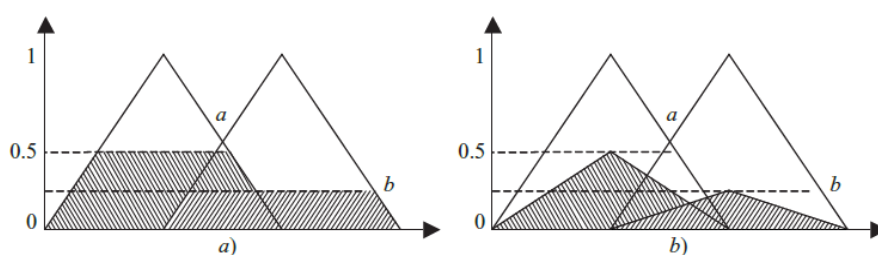
Esta interfaz es encargada de asignar a cada variable de entrada su valor de pertenencia a cada conjunto difuso del universo de discurso, es decir convierte los datos de entrada reales en valores lingüísticos. [4]

#### **2.4.3 Base de conocimientos**

En la base de conocimientos está contenida toda la información de la aplicación que se desea controlar, así también las metas del controlador, en nuestro caso del sistema bola y plato de la empresa Feedback. La base de datos se basa en un conjunto de reglas lingüísticas para controlar cada una de las variables del proceso. [4]

#### 2.4.4 Lógica de decisiones

El núcleo de un controlador difuso, está fundamentado en la lógica que utiliza el mismo para tomar decisiones. El resultado de utilizar la lógica de decisiones da lugar a conjuntos difusos recortados (a), o conjuntos escalados (b) como a continuación se muestra. [4]



**Figura 2.12 Lógica de decisiones de un controlador difuso. [4]**

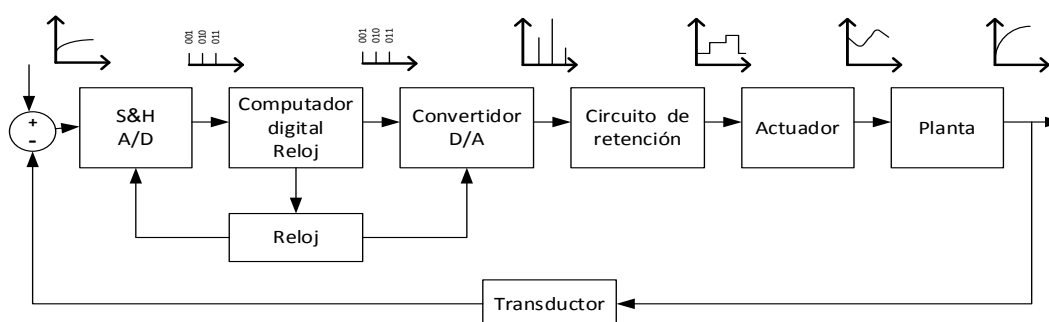
#### 2.4.5 Interfaz de defusificación

La interfaz de defusificación convierte el rango de las variables de salida a valores concretos, es decir es la herramienta para obtener la acción de control nítida a partir de la acción de control difusa. La defusificación es un proceso más complejo que la fusificación ya que existen varias formas de realizarlo, es decir hay varios métodos matemáticos para representar las conclusiones a partir de los conjuntos difusos, entre los más relevantes se tiene: el método del centro de gravedad, el método del promedio de los centros, el método de máximo, si el lector desearía profundizar los mismos, se sugiere

analizarlos en el libro de inteligencia artificial con aplicaciones a la ingeniería de Pedro Ponce cruz. [4]

## 2.5 Sistemas de control digital

A continuación se muestra un sistema de control digital en diagrama de bloques, el cual ha sido tomado de [6]:



**Figura 2.13 Diagrama en bloques de un sistema de control digital [5]**

Para comprender de manera integral como funciona un sistema de control en forma digital, [6] dice:

La operación del controlador se maneja por el reloj. En dicho sistema de control digital, en algunos puntos del sistema pasan señales de amplitud variables, ya sea en tiempo continuo o en tiempo discreto, mientras que en otros pasan señales codificadas en forma numérica.

La salida de la planta es una señal en tiempo continuo. La señal de error se convierte en forma digital mediante el circuito de muestreo y retención y el convertidor analógico-digital. La computadora digital

procesa las secuencias de números por medio de un algoritmo y produce nuevas secuencias de números. El reloj en tiempo real de la computadora sincroniza los eventos. La salida del circuito de retención, una señal en tiempo continuo se alimenta a la planta ya sea de manera directa o a través de un actuador para controlar su dinámica. El circuito de muestreo y retención (S&H) y el convertidor analógico-digital (A/D) convierten la señal en tiempo continuo a una secuencia de palabras binarias, dicho proceso de conversión A/D se conoce como codificación. [6]

A continuación se hace una breve descripción de cada una de las partes que conforman el sistema de control digital mostrado anteriormente:

### **2.5.1 Muestreador y retenedor (S&H)**

Es la descripción que se le otorga al amplificador de muestreo y retención, el objetivo de este circuito es recibir una señal analógica y mantenerla dicha señal en un valor constante durante un intervalo de tiempo especificado. [6]

### **2.5.2 Convertidor analógico-digital (A/D)**

Este circuito se conoce también como codificador, y su función es la conversión de un señal analógica en digital, la cual esta codificada numéricamente, se debe tener en consideración que al digitalizar una



señal analógica se hace una aproximación de la misma ya que se asigna un conjunto finito de números, a este proceso de aproximación se le denomina cuantificación. [6]

### **2.5.3 Convertidor digital-analógico (D/A)**

Este circuito se conoce también como decodificador, y realiza el proceso inverso del (A/D), es decir recibe la señal digital y la convierte nuevamente en analógica. [6]

### **2.5.4 Planta**

Se refiere al objeto tangible a ser controlado, por ejemplo en el presente trabajo de titulación la planta es el sistema mecatrónico bola y plato 33-240 de la empresa Feedback. Se debe mencionar que una de las partes más complejas de diseñar un algoritmo de control, es justamente encontrar el modelo matemático que describe el funcionamiento y operación de la planta. [6]

## **2.6 Transductor**

Se refiere al dispositivo que se encarga de convertir una señal de entrada, en otra de naturaleza diferente. Se debe tener en cuenta que existen transductores analógicos y digitales. [6]

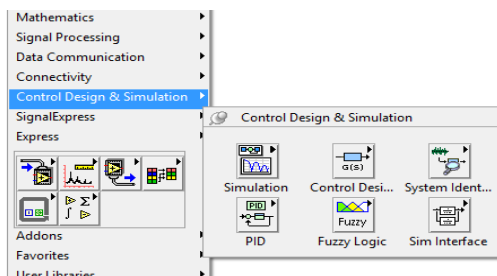
## **2.7 LabView, herramientas de diseño de controladores**

De acuerdo a lo que se indica en [1] el entorno de programación utilizado en la plataforma bola y plato diseñada por la empresa Feedback es LabView, el cual presenta varias herramientas para el diseño de sistemas de control utilizando lógica difusa, así como herramientas para el diseño de sistemas de control en tiempo continuo y discreto. Algo importante de LabView es que tiene un conjunto de herramientas para trabajar en la adquisición de datos en tiempo real. A continuación se realizará una breve descripción de los principales bloques utilizados para desarrollar la solución del presente proyecto de titulación.

### **2.7.1 Diseño de un controlador PID**

Para el diseño de un algoritmo de control clásico sea este analógico o digital PID, o utilizando variables de estado, o cualquier otro dentro de la teoría de control clásico, LabView tiene desarrollado un extenso conjunto de herramientas , el cual se encuentra dentro de la paleta : control, diseño y simulación, aquí se tiene herramientas de simulación equivalentes a las utilizadas en Matlab, así también está desarrollado todo un conjunto de herramientas para el diseño de controladores antes mencionados y muchos más, también dentro de la misma paleta de herramientas se tiene diseñados instrumentos para la identificación

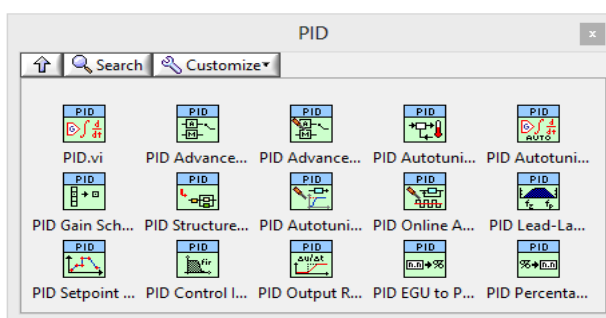
de sistemas. A continuación se muestra como acceder a este “toolkit” [7]:



**Figura 2.14 Ruta para el acceso a las herramientas de control, diseño, y simulación de sistemas. [7]**

En este apartado vamos a describir las principales herramientas para el diseño de un controlador PID, ya que en el presente trabajo de titulación se realizará una comparación entre un algoritmo de control inteligente diseñado con lógica difusa y un PID convencional.

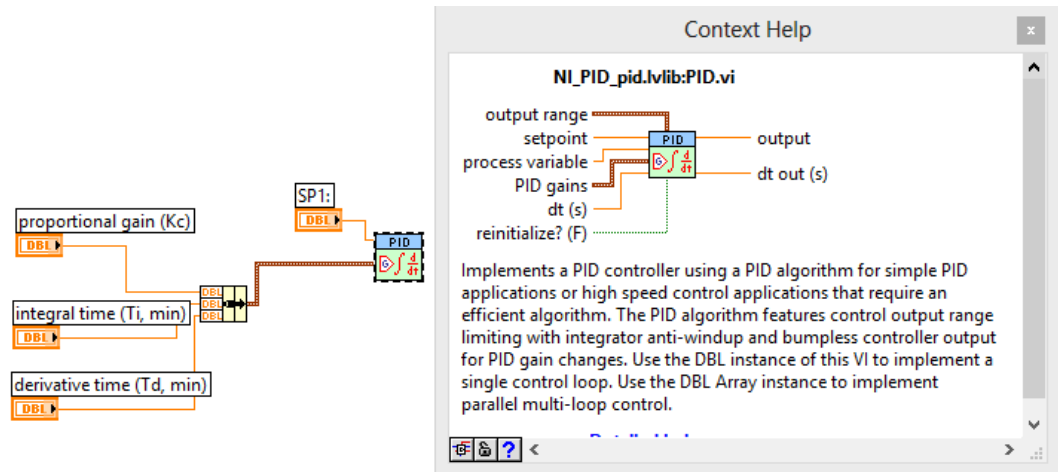
Los esquemas PID desarrollados en LabView se muestran a continuación:



**Figura 2.15 Esquemas PID de Labview, [7]**

Como se puede observar dentro de los esquemas PID, se tiene: el esquema PID normal, avanzado, y uno auto sintonizable, el más

usado por sus facilidades de implementación es el primero, como a continuación se detalla:

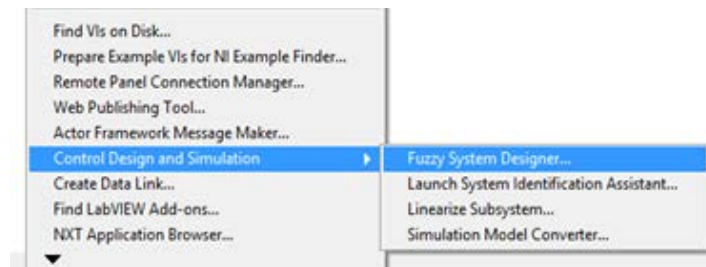


**Figura 2.16 Descripción y programación de un PID de Labview, [7]**

En la programación se muestra un PID con sus ganancias proporcional, integral, y derivativa. En el entorno gráfico se podrá modificar estas ganancias como se muestra en la siguiente figura:

### 2.7.2 Diseño de un controlador difuso

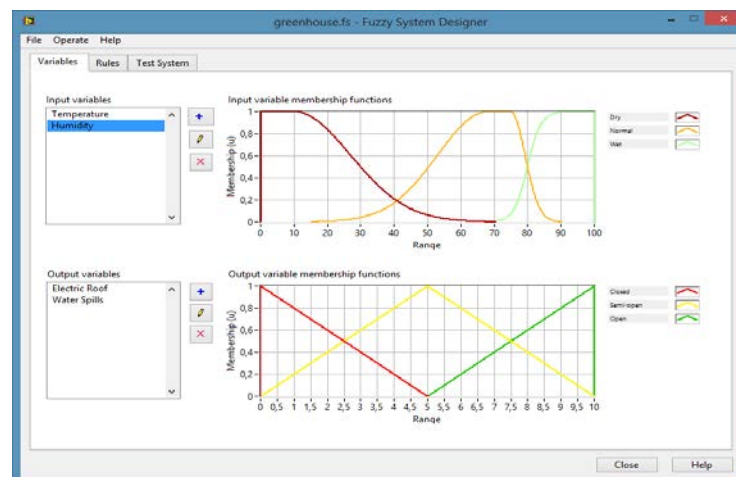
Lo primero que se debe realizar es ingresar a las herramientas de control, diseño y simulación, para posterior ingresar al diseñador de sistemas difusos, como se muestra en la siguiente figura [7]:



**Figura 2.17 Ruta para el ingreso al diseñador de sistemas difusos. [7]**

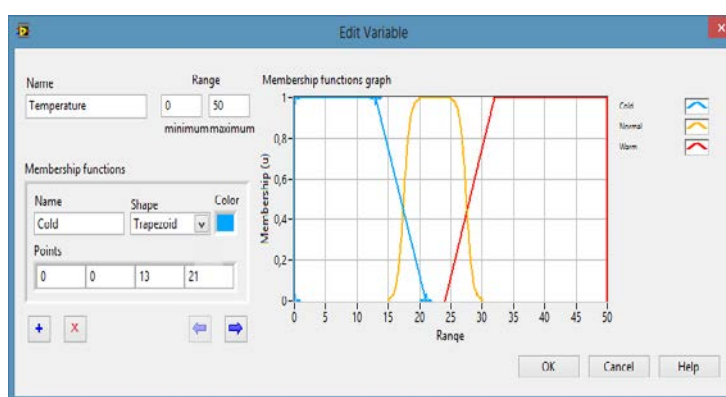
Una vez que el usuario ha ingresado al diseñador de sistemas difusos se despliega una pantalla gráfica, en donde se debe crear las variables de entrada, de salida, las reglas, y se puede hacer una comprobación o test del controlador diseñado.

El entorno gráfico del diseñador de sistemas difusos tiene dos pantallas en una de ellas se observará como varían las funciones de membresía de entrada, y en la otra las funciones de membresía de salida. A continuación se muestra la gráfica descrita:



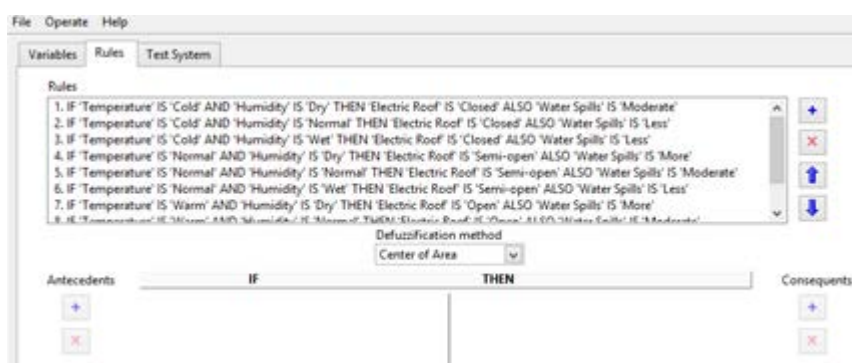
**Figura 2.18 Diseñador de sistemas difusos. [7]**

Para crear las funciones de membresía de entrada y salida se debe ingresar en el icono que tiene el signo de suma, y se despliega una pantalla gráfica que nos permite ir creando tales funciones estudiadas anteriormente, es decir, triangulares, trapezoidales, tipo sigma, etc., así como también esta herramientas nos brinda una opción libre para crear alguna función de membresía especial definida por el programador.



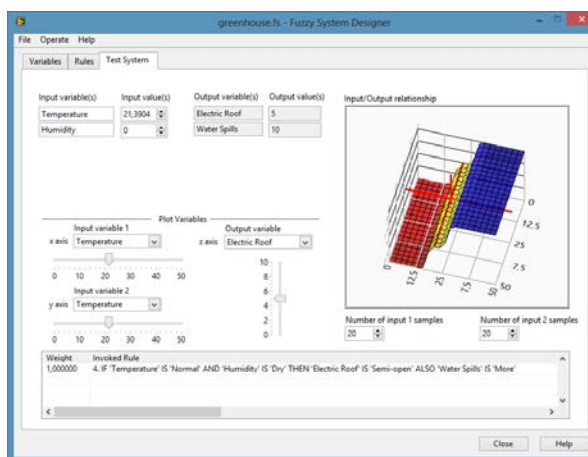
**Figura 2.19 Diseño de funciones de membresía. [7]**

Posteriormente a definir y crear las funciones de membresía de entrada y salida, se deben crear las reglas de asociación difusa, como se estudió en un apartado anterior, este conjunto de reglas son tan óptimas como el diseñador conozca y haya experimentado con el proceso o sistema a controlar. Así también se elige el método de defusificación, la herramienta por defecto utiliza el método del centro de área. A continuación se muestra [7]:



**Figura 2.20 Creación de reglas de un sistema difuso. [7]**

Una vez realizada todas las acciones descritas anteriormente, el controlador difuso está diseñado en su totalidad, ahora restaría hacer una comprobación del sistema ingresando a la opción “test” y observando la superficie de control como se muestra en la siguiente figura:



**Figura 2.21 Superficie de control de un sistema difuso. [7]**

Ahora que se ha diseñado el controlador difuso, se lo puede exportar a la aplicación desarrollada en LabView para el control de la planta en

estudio. A continuación se muestra a manera de ejemplo una aplicación de lógica difusa diseñada en LabView para un sistema de dos entradas y dos salidas en el cual se observa claramente la etapa de creación y carga de las funciones de membresía, la etapa de fusificación y la defusificación:

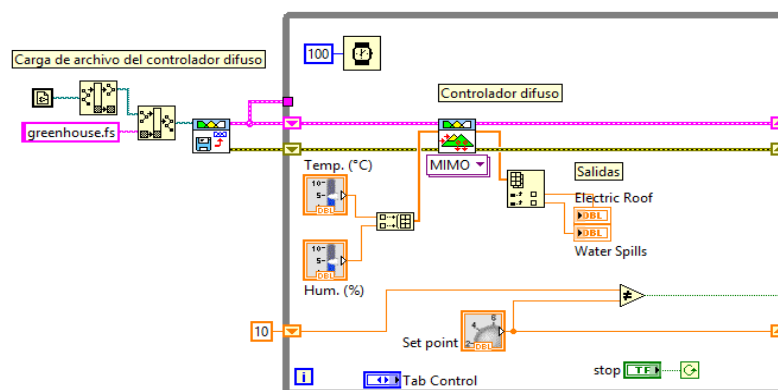


Figura 2.22 Controlador difuso diseñado en LabView. [7]



## CAPÍTULO 3

### DESCRIPCIÓN DEL ESCENARIO

#### 3.1 Sistema Bola y Plato 33-240

Ahora vamos aterrizar el esquema general de un sistema de control digital, justamente al sistema de estudio del presente trabajo de titulación. A continuación se muestra el esquema [1]:

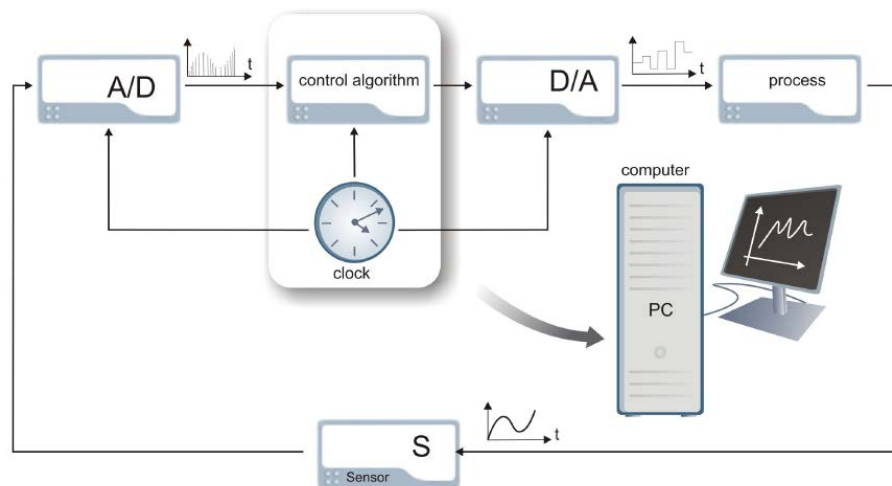


Figura 3.1 Diagrama de bloques del sistema bola y plato [1]

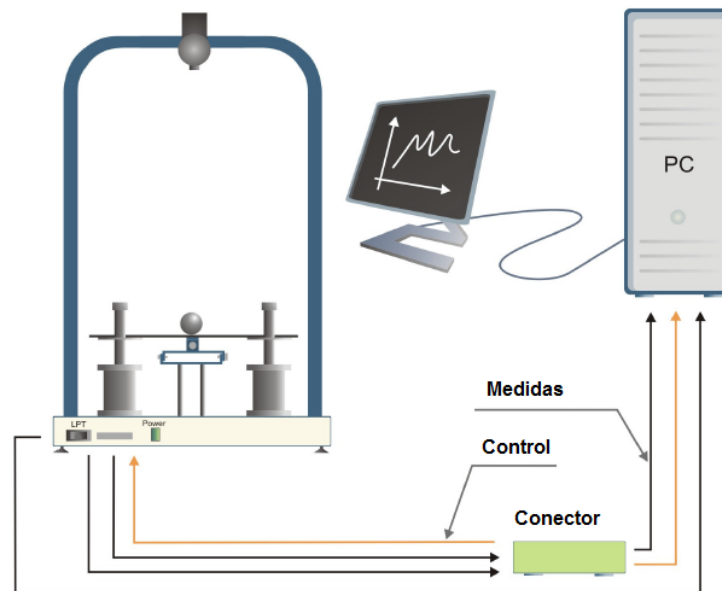
El sistema en forma general consiste de cuatro elementos: un computador con un algoritmo de control digital, los convertidores A/D y D/A, el proceso a controlar, y los sensores.

El prototipo mencionado trabaja con el sistema de programación gráfica LabView, y sus herramientas desarrolladas para aplicaciones en tiempo real.

Entre las principales partes del sistema B&P está la base formada por cuatro inductores en los cuales circula corriente eléctrica y estos a su vez generan un campo electromagnético en cada inductor, que permite la levitación de un plato. Este plato está montado sobre un sistema mecánico con potenciómetros instalados en ejes ortogonales, los cuales son encargados de medir los ángulos de inclinación de la tabla, y esta medida es enviada al computador a través de la tarjeta de adquisición de datos NI PCI/6251 (Peripheral Component Interconnect) [1] y [9].

El plato que forma el sistema descrito es de bajo peso, de material plástico, y de forma cuadrada, en sus extremos presenta cuatro anillos de aluminio, en donde por el efecto de corrientes de Eddy se induce el campo electromagnético para que este levite. El sistema bola y plato también tiene una unidad de potencia para el manejo de los niveles de corriente de cada inductor, y una cámara digital montada sobre un brazo que permite la ubicación de la bola en el plato. [9]

A continuación se muestra gráficamente la plataforma descrita anteriormente:



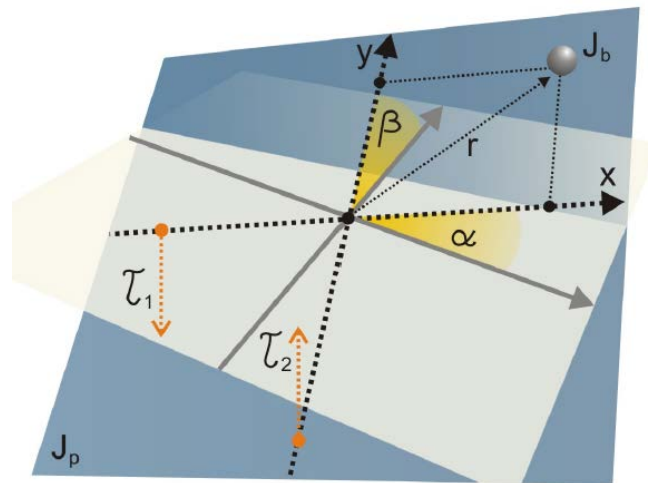
**Figura 3.2 Sistema de control bola y plato. [9]**

### **3.2 Modelo no lineal del sistema Bola y Plato 33-240**

Para poder diseñar de manera óptima algoritmos de control sobre la plataforma bola y plato 33-240, así como realizar procesos de identificación del sistema, es conveniente estudiar el modelo matemático del sistema para comprender de mejor manera las variables que determinan su funcionamiento.

Para este efecto se utilizará los modelos matemáticos propuestos por la empresa Feedback.

Para analizar el modelo matemático del sistema bola y plato 33-240 se utilizó el bosquejo del mismo de manera vectorial que nos proporciona la empresa Feedback, como se muestra a continuación [9]:



**Figura 3.3 Diagrama vectorial del sistema bola y plato. [9]**

A continuación se muestra la modelación matemática realizada por Feedback Instruments limited [9].

La metodología para modelar el sistema ha considerado el balance de energía cinética,  $T$ , y de energía potencial  $W$ , así se obtienen las siguientes ecuaciones:

$$T = \frac{1}{2} \left[ m(\dot{x}^2 + \dot{y}^2) + I_p(\dot{\alpha}^2 + \dot{\beta}^2) + I_b \left( \frac{\dot{x}^2 + \dot{y}^2}{r^2} \right) + I_b(\dot{\alpha}^2 + \dot{\beta}^2) + m(x\dot{\alpha} + y\dot{\beta}) \right] \quad (3.1)$$

Donde:

$$W = m.g.x.\text{sen}(\alpha) + m.g.y.\text{sen}(\beta) \quad (3.2)$$

Teniendo en consideración los pares de fuerzas externos  $\tau_1$  y  $\tau_2$  que generan los inductores del sistema, los cuales están en función de la corriente  $i$ , y los ángulos  $\alpha$  y  $\beta$  se obtiene:

$$\tau_1 = 0.2k_c \frac{(i_1 + 2.2)^2}{|\alpha - 0.17|} - 0.2k_c \frac{(i_1 - 2.2)^2}{|\alpha + 0.17|}; |\alpha| < 0.17 \quad (3.3)$$

$$\tau_2 = 0.2k_c \frac{(i_2 + 2.2)^2}{|\beta - 0.17|} - 0.2k_c \frac{(i_2 - 2.2)^2}{|\beta + 0.17|}; |\beta| < 0.17 \quad (3.4)$$

Los parámetros referidos en cada una de las ecuaciones se detallan en la siguiente tabla:

**Tabla 3.1 Parámetros del modelo del plato del sistema. [9]**

Parámetros	Valores
K- energía cinética	[J]
W- energía potencial	[J]
m- masa de la bola	[Kg]
x- posición de la bola en dirección x	[m]
y- posición de la bola en dirección y	[m]
$\dot{x}$ - velocidad de la bola en dirección x	[m/s]
$\dot{y}$ - velocidad de la bola en dirección y	[m/s]
$\alpha$ - ángulo del plato respecto al eje x	[rad]
$\beta$ - ángulo del plato respecto al eje y	[rad]

$\dot{\alpha}$ - velocidad angular del plato con respecto al eje x	[rad/s]
$\dot{\beta}$ - velocidad angular del plato con respecto al eje y	[rad/s]
g – constante gravitacional	9.81 [m/s <sup>2</sup> ]
$I_b$ - Momento de inercia de la bola	4.32*10 <sup>-6</sup> [Kg*m <sup>2</sup> ]
$I_p$ - Momento de inercia del plato	0.133 [Kg*m <sup>2</sup> ]
$\tau_1$ - torque 1	[Nm]
$\tau_2$ - torque 2	[Nm]
k - constante	0.0025 [Nm.deg/Kg.A <sup>2</sup> ]
$m_p$ - masa del plato	0.2 [Kg]
$k_c$ - constante	0.0125 [Nm.deg/A <sup>2</sup> ]

Con la energía cinética y potencial definida se puede ahora definir el Lagrangiano,  $L$  de acuerdo a la siguiente ecuación:

$$L = T - W \quad (3.5)$$

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = F \quad (3.6)$$

Donde  $q = [x, y, \alpha, \beta]$  y  $F$  denotan todas las fuerzas externas y torques.

Así también se han calculado las derivadas parciales:

$$\frac{\partial L}{\partial x} = m(x\dot{\alpha} + y\dot{\beta})\dot{\alpha} - m.g.\text{sen}(\alpha) \quad (3.7)$$

$$\frac{\partial L}{\partial y} = m(x\dot{\alpha} + y\dot{\beta})\dot{\beta} - m.g.\text{sen}(\beta) \quad (3.8)$$

$$\frac{\partial L}{\partial \alpha} = -m.g.x \cos(\alpha) \quad (3.9)$$

$$\frac{\partial L}{\partial \beta} = -m.g.y.\cos(\beta) \quad (3.10)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = \frac{d}{dt} \left( m\dot{x} + \frac{I_b \dot{x}}{r^2} \right) = \ddot{x} \left( m + \frac{I_b}{r^2} \right) \quad (3.11)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{y}} = \frac{d}{dt} \left( m\dot{y} + \frac{I_b \dot{y}}{r^2} \right) = \ddot{y} \left( m + \frac{I_b}{r^2} \right) \quad (3.12)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}} = \frac{d}{dt} \left[ \dot{\alpha} (I_p + I_b) + mx(x\dot{\alpha} + \dot{\beta}y) \right] = \ddot{\alpha} (I_p + I_b) + m(\ddot{\alpha}x^2 + \ddot{\beta}xy + \dot{\beta}x\dot{y} + \dot{\beta}\dot{x}y) \quad (3.13)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\beta}} = \frac{d}{dt} \left[ \dot{\beta} (I_p + I_b) + my(x\dot{\alpha} + \dot{\beta}y) \right] = \ddot{\beta} (I_p + I_b) + m(\ddot{\beta}y^2 + \ddot{\alpha}xy + \dot{\alpha}x\dot{y} + \dot{\alpha}\dot{x}y) \quad (3.14)$$

De la ecuación (3.6) y (3.7) combinándolas se obtiene:

$$m.g.\text{sen}(\alpha) + \ddot{x} \left( m + \frac{I_b}{r^2} \right) - m(x\dot{\alpha} + y\dot{\beta})\dot{\alpha} = 0 \quad (3.15)$$

$$m.g.\text{sen}(\beta) + \ddot{y} \left( m + \frac{I_b}{r^2} \right) - m(x\dot{\alpha} + y\dot{\beta})\dot{\beta} = 0 \quad (3.16)$$

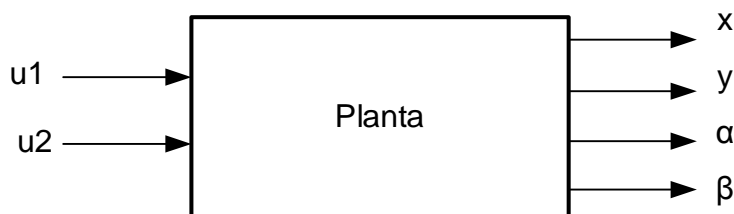
$$mg \cdot x \cdot \cos(\alpha) + \ddot{\alpha}(I_p + I_b) + m(\ddot{\alpha}x^2 + \ddot{\beta}xy + \dot{\beta}x\dot{y} + \dot{\beta}\dot{x}y) = \tau_1$$

(3.17)

$$mg \cdot y \cdot \cos(\beta) + \ddot{\beta}(I_p + I_b) + m(\ddot{\beta}y^2 + \ddot{\alpha}xy + \dot{\alpha}x\dot{y} + \dot{\alpha}\dot{x}y) = \tau_2$$

(3.18)

Desde la ecuación (3.15) a la ecuación (3.18) y la ecuación (3.3) y (3.4) representan el modelo no lineal del sistema, y este puede ser descrito de acuerdo a la siguiente figura:



**Figura 3.4 Diagrama en bloques del modelo del sistema bola y plato [9]**

### 3.3 Linealización del sistema bola y plato 33-240

A continuación se detallará la simplificación del modelo no lineal del sistema bola y plato presentado anteriormente, considerando que la masa de la bola es muy pequeña con relación a la dinámica de la tabla, así también la aceleración alcanzada por la pelota es muy pequeña, por lo cual el acoplamiento cruzado entre las dos direcciones perpendiculares puede despreciarse [9]. De la simplificación de las ecuaciones que describen el modelo matemático del sistema bola y plato se obtiene:



$$mg.\text{sen}(\alpha) + \ddot{x} \left( m + \frac{I_b}{r^2} \right) = 0 \quad (3.19)$$

$$mg.\text{sen}(\beta) + \ddot{y} \left( m + \frac{I_b}{r^2} \right) = 0 \quad (3.20)$$

$$mg.x.\cos(\alpha) + \ddot{\alpha} (I_p + I_b) = \tau_1 \quad (3.21)$$

$$mg.y.\cos(\beta) + \ddot{\beta} (I_p + I_b) = \tau_2 \quad (3.22)$$

Hay que resaltar que la linealización del sistema es con respecto a las variables  $i$  y  $\alpha$  o  $\beta$ . De las ecuaciones se nota que cuando  $i=0$  y  $\alpha=0$  o  $\beta=0$ , se obtiene el punto de equilibrio del sistema [10].

$$m\Delta\ddot{\alpha} = \frac{i_{10}+2.2}{|\alpha_0-0.17|} * \Delta i_1 - 2k_c \frac{i_{10}-2.2}{|\alpha_0+0.17|} * \Delta i_1 - k_c \frac{(i_{10}-2.2)^2}{(\alpha_0+0.17)^2} * \Delta\alpha + k_c \frac{(i_{10}+2.2)^2}{(\alpha_0-0.17)^2} * \Delta\alpha \quad (3.23)$$

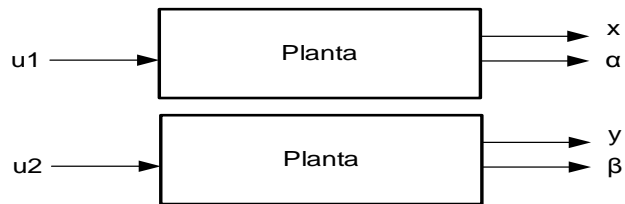
$$m\Delta\ddot{\beta} = \frac{i_{20}+2.2}{|\beta_0-0.17|} * \Delta i_2 - 2k_c \frac{i_{20}-2.2}{|\beta_0+0.17|} * \Delta i_2 - k_c \frac{(i_{20}-2.2)^2}{(\beta_0+0.17)^2} * \Delta\beta + k_c \frac{(i_{20}+2.2)^2}{(\beta_0-0.17)^2} * \Delta\beta \quad (3.24)$$

Reemplazando los puntos de equilibrio, las ecuaciones (3.23) y (3.24) se simplifican a:

$$s^2\beta(s) = 4k \frac{2.2}{0.17} * i_2(s) \quad (3.25)$$

$$s^2\alpha(s) = 4k \frac{2.2}{0.17} * i_1(s) \quad (3.26)$$

Ahora el sistema bola y plato podría ser analizado como dos sistemas desacoplados con una sola entrada y una sola salida (SISO) como se muestra en la figura siguiente:



**Figura 3.5 Diagrama en bloques del modelo del sistema bola y plato [9].**

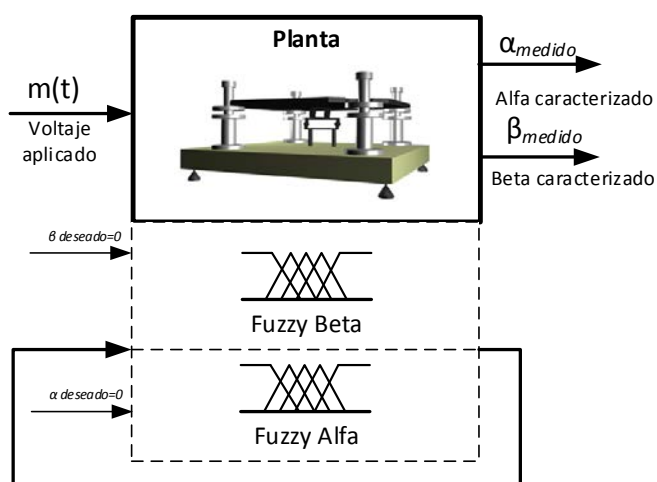
## **CAPÍTULO 4**

### **DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS**

#### **4.1 Algoritmo de caracterización de la planta**

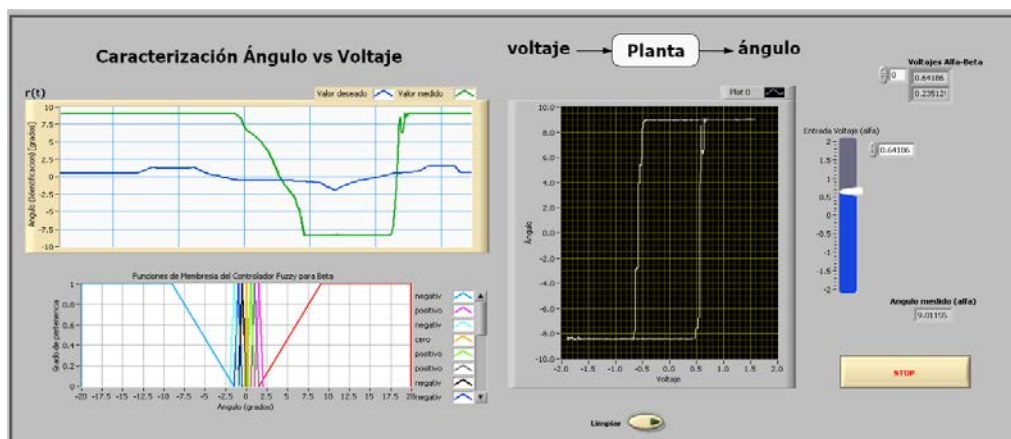
Para diseñar el algoritmo de control inteligente usando lógica difusa para el control de la planta que se muestra a continuación, primeramente fue necesario caracterizar las variables de voltaje y ángulos que manejan dicho algoritmo difuso, es decir es necesario conocer como varían los ángulos de inclinación alfa y beta al levitar el plato del sistema en estudio en función de un voltaje aplicado a cada

uno de los cuatro inductores que generan el campo electromagnético e inducen las corrientes de Eddy en cada anillo de aluminio que se encuentra ubicado en las esquinas de la base del plato. A continuación se muestra el diagrama de bloques del algoritmo de caracterización de la planta.



**Figura 4.1 Diagrama de bloques del algoritmo de caracterización de la planta**

A continuación se detalla la caracterización efectuada para el ángulo alfa, y debido a que el sistema tiene simetría ortogonal, se puede asumir con una buena aproximación que las variaciones del ángulo beta son muy aproximadas. No obstante el algoritmo permite caracterizar los dos ángulos de inclinación de la planta.



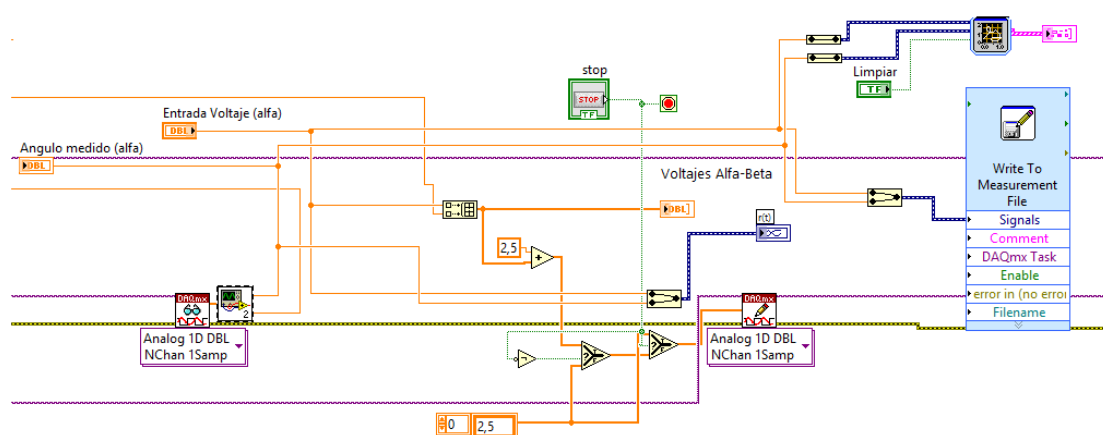
**Figura 4.2 Modo visual del algoritmo para caracterizar Voltaje Vs Ángulo de la planta**

Para poder realizar la caracterización de alfa, el ángulo beta debe estar fijo en posición de cero grados de manera que mantenga el plato lo más horizontal posible, y así alfa pueda variar en todo su rango permitido. Para esto inicialmente se experimentó con una sujeción manual del ángulo beta, y posteriormente cuando se diseñó el algoritmo difuso para el control de la plataforma en estudio se retroalimentó el algoritmo de caracterización incluyendo una forma automática de fijar el ángulo beta a la referencia de cero grados.

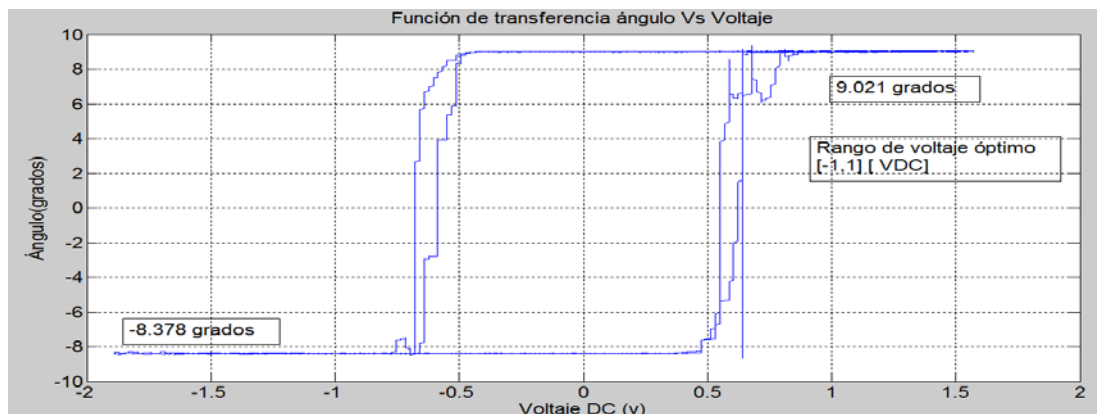
Posterior se estableció una escala de variación de voltaje de los inductores entre -2 y 2 voltios, aunque de la experimentación se apreció que en los límites de voltaje aplicado, los inductores empezaban a vibrar por el campo electromagnético generado, lo cual generó una nueva interrogante como es conocer el nivel de voltaje óptimo a aplicarse en los inductores para lograr que el ángulo de inclinación de la tabla haga

una excursión total dentro de los límites permitidos por el dimensionamiento físico de la planta.

Para solventar esta duda se adquirió los datos de las variables de voltaje y ángulo alfa de la planta y se exportó los datos al software Matlab para graficar la función de transferencia de la variación del ángulo alfa en función del voltaje aplicado a los inductores. A continuación se muestra la parte principal del algoritmo de caracterización de la planta realizado en Labview, y el código de Matlab que permite exportar la información y representarla gráficamente.



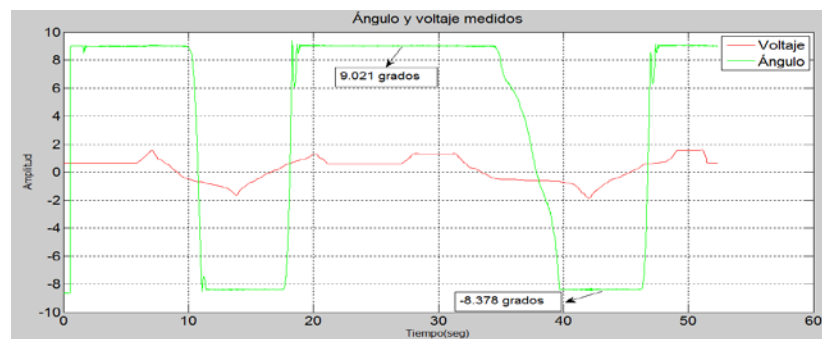
**Figura 4.3 Etapa principal del algoritmo de caracterización de la planta**



**Figura 4.4 Función de transferencia Ángulo Vs Voltaje del plato**

Como se puede observar se generó un lazo de histéresis, en donde se observa que el ángulo de inclinación alfa se encuentra aproximadamente entre -8.378 y 9.021 grados. Así también se observa en el gráfico que el ancho del lazo de histéresis es menor al rango comprendido entre -1 y 1 voltios de corriente directa (DC) el cual fue aplicado a los inductores, esto nos conduce a la conclusión que la zona segura de voltaje a ser aplicados a los inductores debe estar comprendida en este dominio, ya que posterior a estos voltajes los inductores empiezan a vibrar por la fuerza generada por el campo electromagnético y esto podría destruir la etapa de potencia que controlar la corriente que circula en cada inductor, y causar inestabilidades en el sistema.

Así también como se dispone de los datos de ángulo y voltaje, se graficó en Matlab cada uno de ellos en función del tiempo, se puede observar que la salida del ángulo alfa se corresponde a lo que indicada la función de transferencia anteriormente descrita, es decir excursiones entre -8.378 y 9.021 grados, ya que posterior se observa una saturación del ángulo medido.



**Figura 4.5 Gráficas de ángulo y voltaje en función del tiempo**

El algoritmo difuso en esta etapa no se lo describirá en detalle, en lo posterior se lo realizará. El objetivo del mismo es fijar el ángulo beta a cero grados, y mantenerlo estable en esa posición, para que la medida del ángulo alfa sea lo más precisa posible y viceversa.

A continuación se describe el programa en Matlab con el cual se elaboraron las gráficas que caracterizaron el ángulo alfa y voltaje del plato en estudio. Es importante mencionar que se hicieron diversas pruebas con el algoritmo diseñado en Labview para llegar a obtener conclusiones en cuanto al rango seguro de voltaje a aplicarse a los

inductores del sistema, así como el rango de variación del ángulo de inclinación alfa del plato.

```
DATOS=xlsread('DATOS1.XLSX');
t=DATOS(:,1);
voltaje=DATOS(:,2);
angulo=DATOS(:,4);

%Caracterizacion
figure(1)
plot(t,voltaje,'r',t,angulo,'g'),title('Angulo y voltaje
medidos'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('voltaje','angulo')

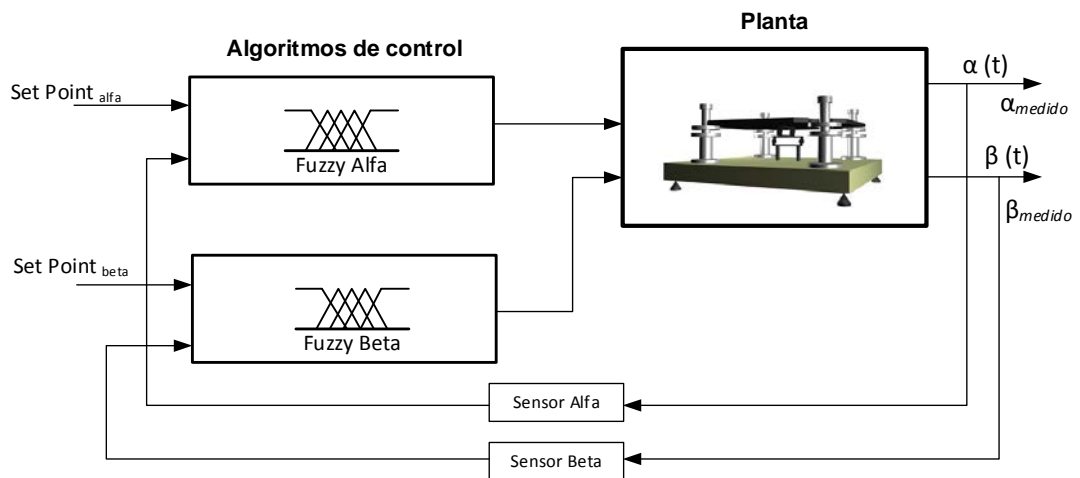
figure(2)
plot(voltaje,angulo),title('Funcion de transferencia'),grid,
xlabel('Voltaje DC (v)'),ylabel('Ángulo(grados)')
```

**Figura 4.6 Programa en Matlab para generar la función de transferencia del ángulo alfa Vs voltaje**

## 4.2 Algoritmo de control difuso

Una vez que se conoce como se comporta los ángulos de inclinación alfa y beta de la planta en función del voltaje aplicado a los inductores, es posible diseñar el algoritmo difuso para el control de la misma de forma más precisa y segura. El algoritmo de control diseñado funciona tanto para el ángulo de inclinación alfa como para el beta, y en modo manual y automático. A continuación se muestra el diagrama de bloques del algoritmo diseñado.

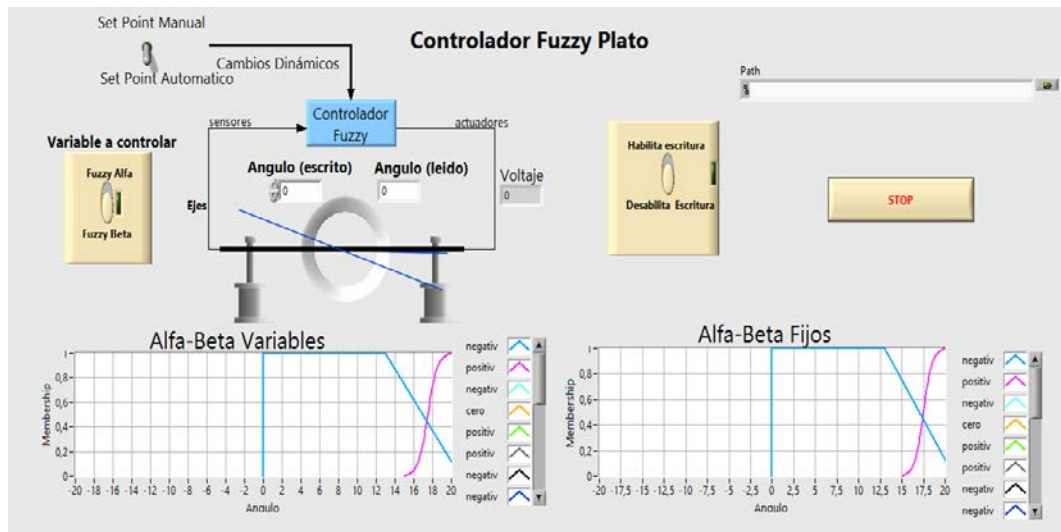




**Figura 4.7 Diagrama de bloques del algoritmo de control difuso para la planta**

En el Modo de operación manual el usuario establece la señal de referencia a manera de escalones y se puede observar como la planta responde a este estímulo hasta estabilizarse, de igual forma en modo automático, la diferencia es que la señal de referencia es un estímulo pulsante a manera de onda cuadrada.

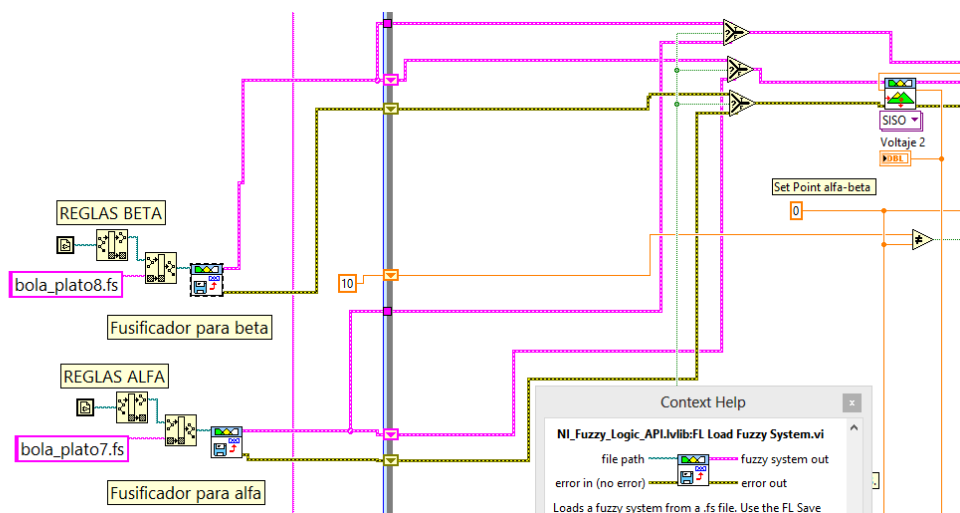
A continuación se muestra el entorno gráfico de la programación del algoritmo difuso y se analizará en detalle cada una de las partes constitutivas más importantes.



**Figura 4.8 Modo visual del algoritmo de control difuso de la planta**

A continuación se describe la programación gráfica del algoritmo difuso para el control de la planta en estudio.

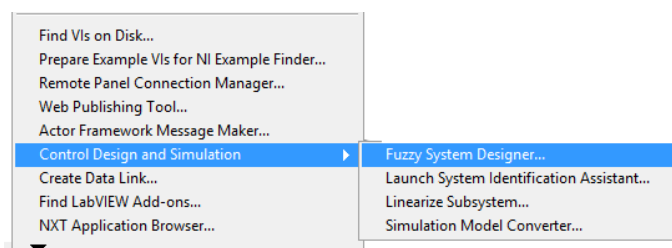
Primeramente se crearon las funciones de membresía, estas convierten los valores nítidos de las variables de ángulo y voltaje en valores difusos, a este proceso se le llama fusificación. A continuación se muestra la programación que se creó en Labview, el cual de manera equivalente a Matlab tiene desarrollado una herramienta que permite crear, editar, y guardar las funciones de membresía para su posterior exportación.



**Figura 4.9 Programación gráfica de las funciones de membresía de alfa y beta**

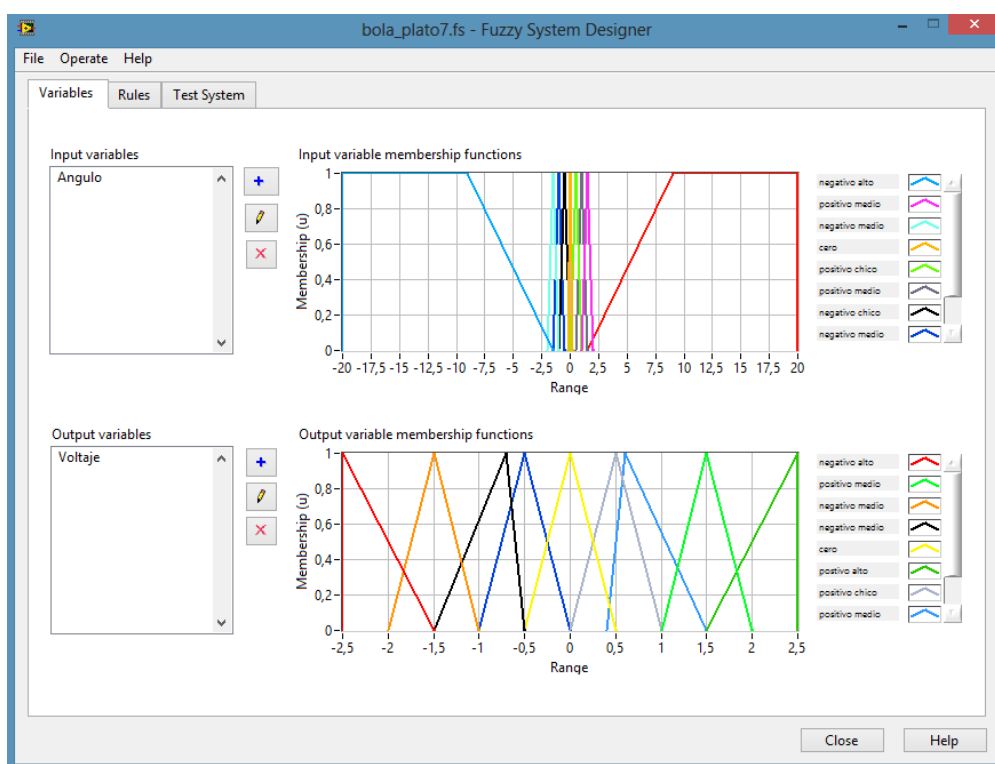
En este caso la familia de funciones de membresía para el ángulo beta se llaman “bola\_plato8.fs” y para alfa “bola\_plato7.fs”, estas se cargan en los fusificadores para cada una de las variables respectivamente.

Par crear las funciones de membresía del algoritmo difuso Labview tiene diseñada una herramientas denominada “diseñador de sistemas difusos”. A Continuación se muestra:



**Figura 4.10 Ruta para el acceso al diseñador de sistemas difusos de LabView**

Ahora el asistente está listo para empezar la creación de las funciones de membresía para el algoritmo de control difuso. A continuación se muestra el entorno gráfico de las funciones de membresía creadas para las variables alfa, y el voltaje aplicado a los inductores para la levitación del plato del sistema.



**Figura 4.11 Funciones de membresía del ángulo alfa y voltaje para el algoritmo de control difuso**

Para la variable de entrada ángulo alfa se crearon nueve funciones de membresía, siete de ellas triangulares, y dos trapezoidales. Las funciones trapezoidales que se observan en los extremos de la gráfica dan el levantamiento inicial del plato a la posición horizontal, es decir a

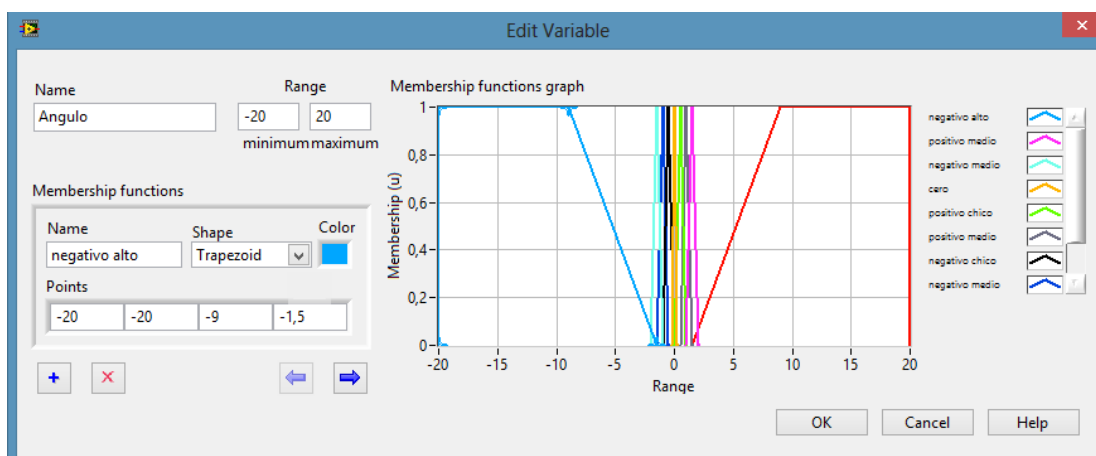
la referencia de cero grados, y las siete funciones que se concentran en el centro son las que hacen que el controlador siga la referencia fijada. A continuación se detalla las características de cada una de las funciones de membresía para el ángulo alfa.

**Tabla 4.1 Parámetros de las funciones de membresía del controlador difuso para el ángulo alfa**

<b>Función de membresía</b>	<b>Forma</b>	<b>Puntos</b>
Negativo alto	Trapezoidal	[-20;-20;-9;-1.5]
Positivo medio	Triangular	[1;1.5;2]
Negativo medio alto	Triangular	[-2;-1.5;-1]
Cero	Triangular	[-0.25;0;0.25]
Positivo Chico	Triangular	[0;0.5;1]
Positivo medio	Triangular	[0.5;1;1.5]
Negativo chico	Triangular	[-1;-0.5;0]
Negativo medio	Triangular	[-1.5;-1;-0.5]
Positivo alto	Triangular	[1.5 9 20 20]

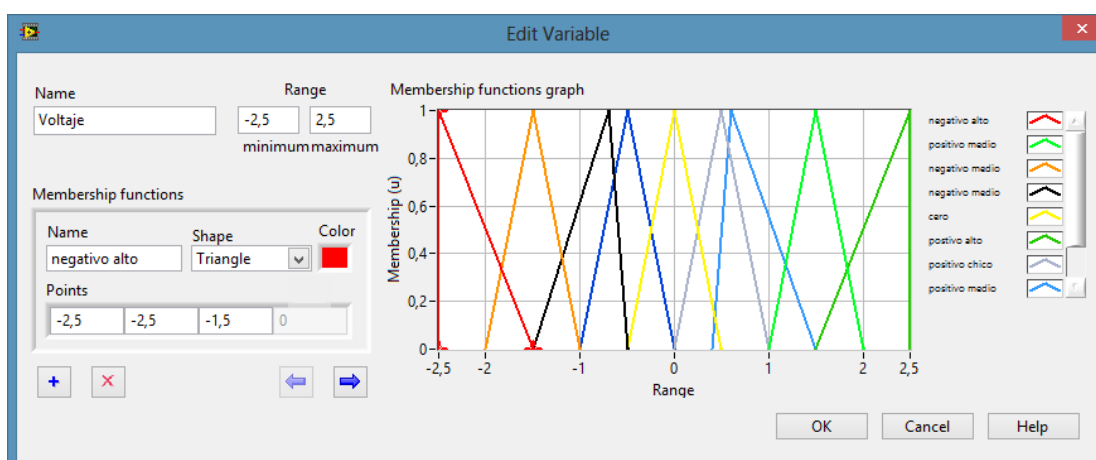
Luego de varias experimentaciones en crear diferentes familias de funciones de membresía se concluyó que mientras más finos sean los triángulos las variaciones de ángulo son más suaves y por consiguiente el seguimiento de la referencia es más preciso.

A continuación se muestra el asistente en Labview en donde se creó y editó las funciones de membresía del algoritmo de control diseñado



**Figura 4.12 Asistente para creación y edición de funciones de membresía de ángulo alfa para el algoritmo de control difuso propuesto**

De manera equivalente se crearon las funciones de membresía para la variable de salida voltaje como a continuación se muestra:



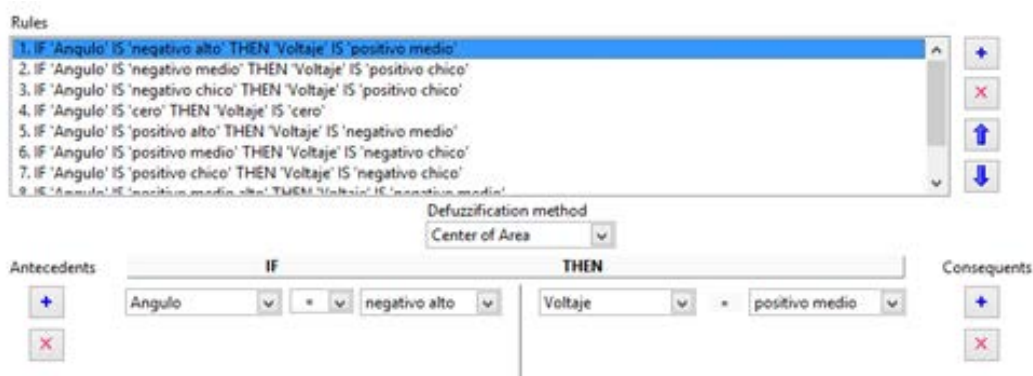
**Figura 4.13 Asistente para creación y edición de funciones de membresía de voltaje para el algoritmo de control difuso.**

A continuación se detalla las características de cada una de las funciones de membresía para el voltaje aplicado a cada inductor.

**Tabla 4.2 Parámetros de las funciones de membresía del controlador difuso para el voltaje aplicado a los inductores del sistema**

<b>Función de membresía</b>	<b>Forma</b>	<b>Puntos</b>
Negativo alto	Triangular	[-2.5;-2.5;-1.5]
Positivo medio	Triangular	[1;1.5;2]
Negativo medio alto	Triangular	[-2;-1.5;-1]
Negativo medio	Triangular	[-1.5;-0.7;-0.5]
Cero	Triangular	[-0.5;0;0.5]
Positivo alto	Triangular	[1.5;2.5;2.5]
Positivo chico	Triangular	[0;0.5;1]
Positivo medio	Triangular	[0.4;0.6;1.5]
Negativo chico	Triangular	[-1;-0.5;0]

Una vez establecidas cada una de las familias de funciones de membresías con sus variables lingüísticas, la siguiente parte en el diseño del algoritmo de control inteligente consiste en crear las reglas de inferencia que interpreten las variables lingüísticas del fusificador. En esta etapa es muy importante el conocimiento que tenga el experto del proceso o planta a controlar para generar e inferir las acciones de control adecuadas que estabilicen el sistema, en nuestro caso el sistema bola y plato de la Universidad Politécnica Salesiana. A continuación se detalla el conjunto de reglas creadas luego de una exhaustiva experimentación con el sistema mencionado anteriormente:



**Figura 4.14 Reglas de inferencia para el algoritmo de control difuso.**

Para poder visualizar de manera completa todas las reglas creadas para el algoritmo de control inteligente, estas se las describe a continuación:

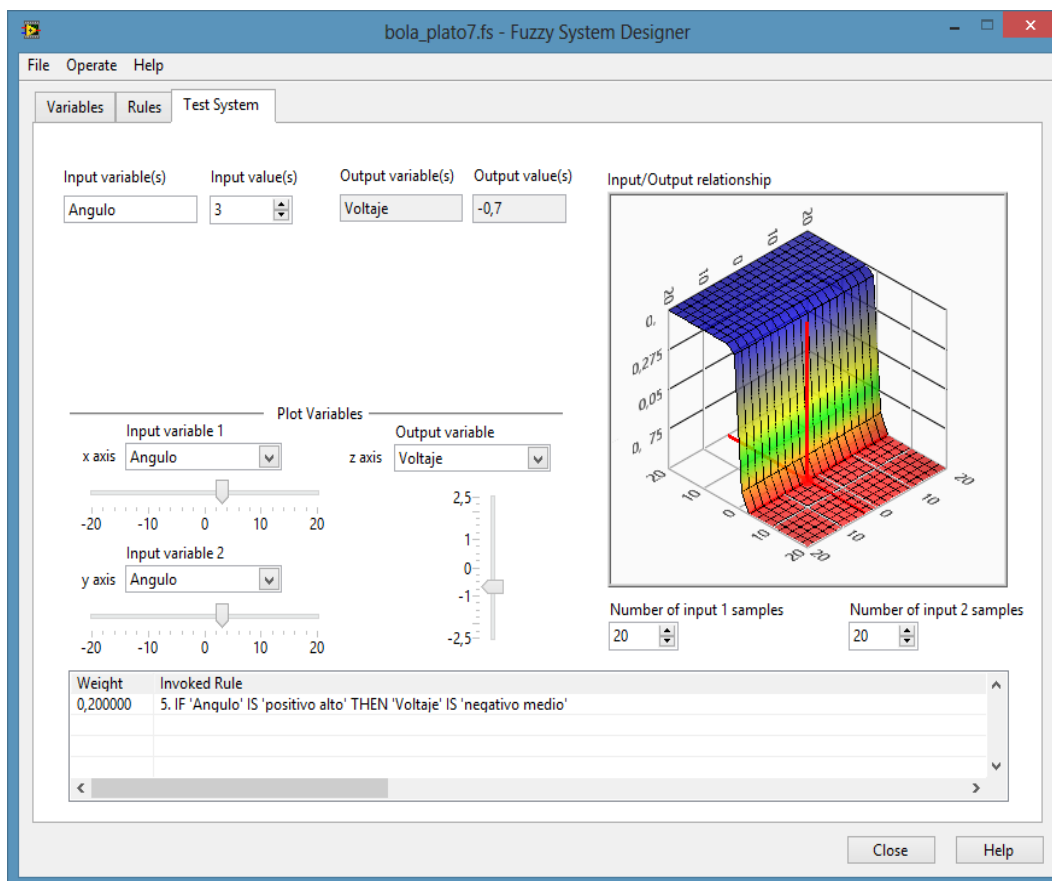
**Tabla 4.3 Reglas de inferencia del algoritmo de control difuso para el ángulo alfa**

1. IF 'Angulo' IS 'negativo alto' THEN 'Voltaje' IS 'positivo medio'
2. IF 'Angulo' IS 'negativo medio' THEN 'Voltaje' IS 'positivo chico'
3. IF 'Angulo' IS 'negativo chico' THEN 'Voltaje' IS 'positivo chico'
4. IF 'Angulo' IS 'cero' THEN 'Voltaje' IS 'cero'
5. IF 'Angulo' IS 'positivo alto' THEN 'Voltaje' IS 'negativo medio'
6. IF 'Angulo' IS 'positivo medio' THEN 'Voltaje' IS 'negativo chico'
7. IF 'Angulo' IS 'positivo chico' THEN 'Voltaje' IS 'negativo chico'
8. IF 'Angulo' IS 'positivo medio alto' THEN 'Voltaje' IS 'negativo medio'
9. IF 'Angulo' IS 'negativo medio alto' THEN 'Voltaje' IS 'positivo medio'

Ahora a continuación se muestra la superficie de control que se generó al crear el algoritmo de control difuso para el sistema en análisis, en esta interface gráfica se puede observar cómo se genera la acción de control de la salida de voltaje, cuando la entrada de ángulo varía dentro



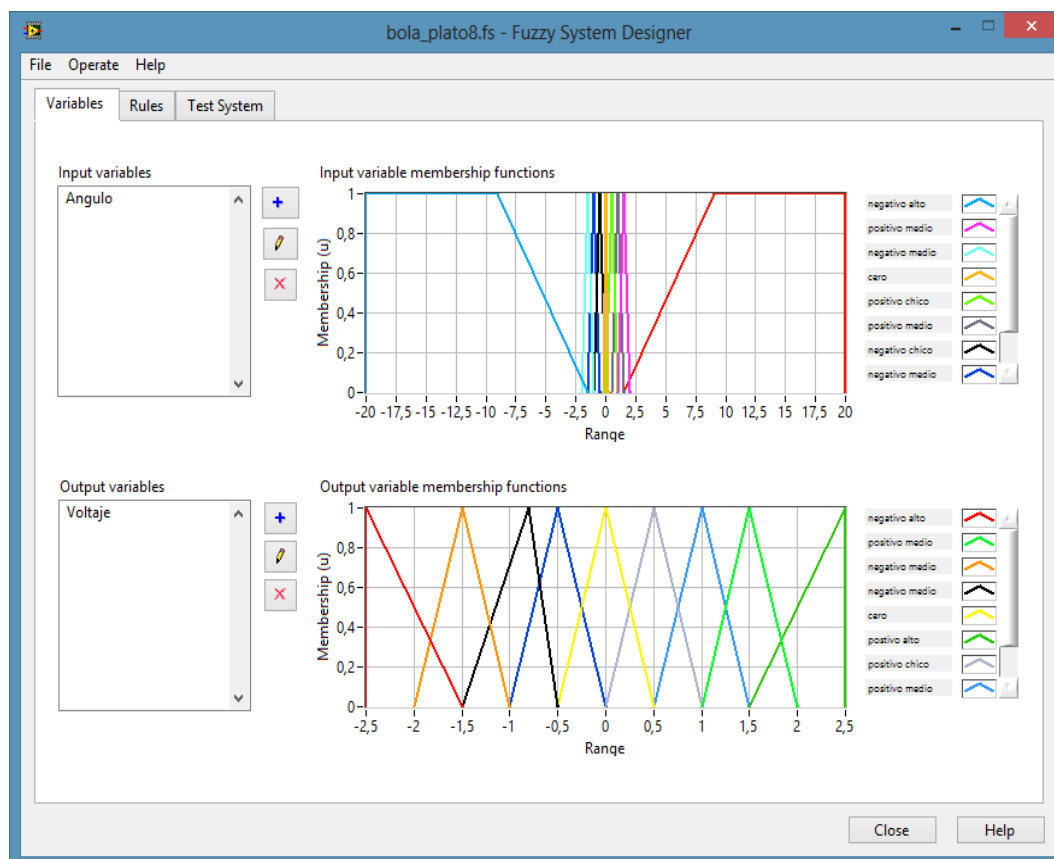
de los límites permitidos por las dimensiones físicas de la planta, para que se realice un seguimiento de la referencia fijada.



**Figura 4.15 Superficie de control del algoritmo difuso Ángulo alfa Vs voltaje**

Para el ángulo beta, la lógica de control es equivalente ya que el sistema bola y plato tiene simetría ortogonal, es decir las direcciones de los ángulos alfa y beta forman 90 grados, por lo cual conociendo el funcionamiento de uno de ellos el otro resulta ser equivalente. El análisis para el ángulo beta se muestra de manera simplificada.

A continuación se muestra las funciones de membresía del ángulo beta y del voltaje aplicado a los inductores:



**Figura 4.16 Funciones de membresía del ángulo beta y voltaje para el algoritmo de control difuso**

Ahora se describen las funciones de membresía para el ángulo beta y el voltaje aplicado a los inductores:

**Tabla 4.4 Parámetros de las funciones de membresía del controlador difuso para el ángulo beta**

Función de membresía	Forma	Puntos
Negativo alto	Trapezoidal	[-20;-20;-9;-1.5]
Positivo medio	Triangular	[1;1.5;2]

Negativo medio alto	Triangular	[-2;-1.5;-1]
Cero	Triangular	[-0.25;0;0.25]
Positivo Chico	Triangular	[0;0.5;1]
Positivo medio	Triangular	[0.5;1;1.5]
Negativo chico	Triangular	[-1;-0.5;0]
Negativo medio	Triangular	[-1.5;-1;-0.5]
Positivo alto	Triangular	[1.5; 9; 20; 20]

**Tabla 4.5 Parámetros de las funciones de membresía del controlador difuso para el voltaje aplicado a los inductores del sistema**

<b>Función de membresía</b>	<b>Forma</b>	<b>Puntos</b>
Negativo alto	Triangular	[-2.5;-2.5;-1.5]
Positivo medio	Triangular	[1;1.5;2]
Negativo medio alto	Triangular	[-2;-1.5;-1]
Negativo medio	Triangular	[-1.5;-0.8;-0.5]
Cero	Triangular	[-0.5;0;0.5]
Positivo alto	Triangular	[1.5;2.5;2.5]
Positivo chico	Triangular	[0;0.5;1]
Positivo medio	Triangular	[0.5;1;1.5]
Negativo chico	Triangular	[-1;-0.5;0]

Las reglas de inferencia del algoritmo difuso para el control del ángulo beta se muestra a continuación:

**Tabla 4.6 Reglas de inferencia del algoritmo de control difuso para el ángulo beta**

1. IF 'Angulo' IS 'negativo alto' THEN 'Voltaje' IS 'positivo medio'
2. IF 'Angulo' IS 'negativo medio' THEN 'Voltaje' IS 'positivo medio'
3. IF 'Angulo' IS 'negativo chico' THEN 'Voltaje' IS 'positivo chico'
4. IF 'Angulo' IS 'cero' THEN 'Voltaje' IS 'cero'
5. IF 'Angulo' IS 'positivo alto' THEN 'Voltaje' IS 'negativo medio'
6. IF 'Angulo' IS 'positivo medio' THEN 'Voltaje' IS 'negativo medio'
7. IF 'Angulo' IS 'positivo chico' THEN 'Voltaje' IS 'negativo chico'
8. IF 'Angulo' IS 'positivo medio alto' THEN 'Voltaje' IS 'negativo medio'
9. IF 'Angulo' IS 'negativo medio alto' THEN 'Voltaje' IS 'positivo medio'

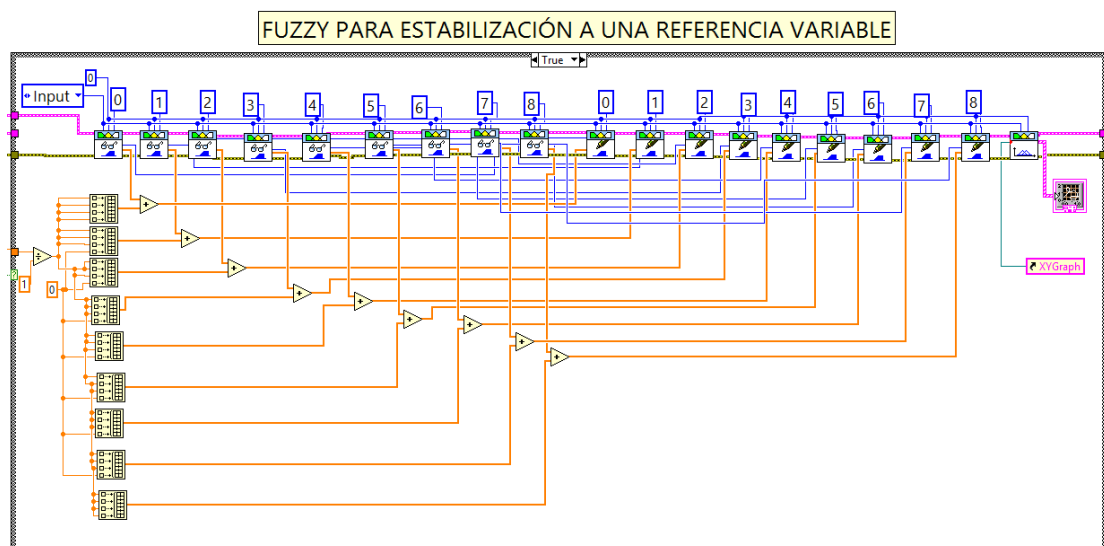
La superficie de control para el ángulo beta es equivalente que la obtenida para el ángulo alfa.

El método de defusificación elegido para poder enviar valores reales nítidos a la planta se conoce como centro de área o de gravedad (MCDG) y se fundamenta en cortar las funciones de membresía dependiendo del grado de pertenencia fijado formando áreas adyacentes las cuales se superponen y al final se calcula el centroide, este es el valor nítido que el defusificador envía a la planta.

A continuación se explica algunos detalles del algoritmo de control difuso diseñado que se consideran de importancia. Cuando se controla el ángulo de inclinación alfa, el ángulo beta debe quedar fijo en la posición de cero grados y viceversa, para esto fue necesario construir

dos algoritmos difusos, uno de ellos para el manejo dinámico de la variable de la referencia de entrada y otro fijo para el ángulo pivote.

Para lograr que el algoritmo de control difuso dinámico realice un seguimiento a la referencia, se hizo que las funciones de membresía se desplacen en tiempo real en un rango de valores adecuados, para el caso de los ángulos alfa y beta este rango se definió entre -20 y 20 grados de manera experimental. A continuación se muestra la programación en Labview que permite realizar este propósito.



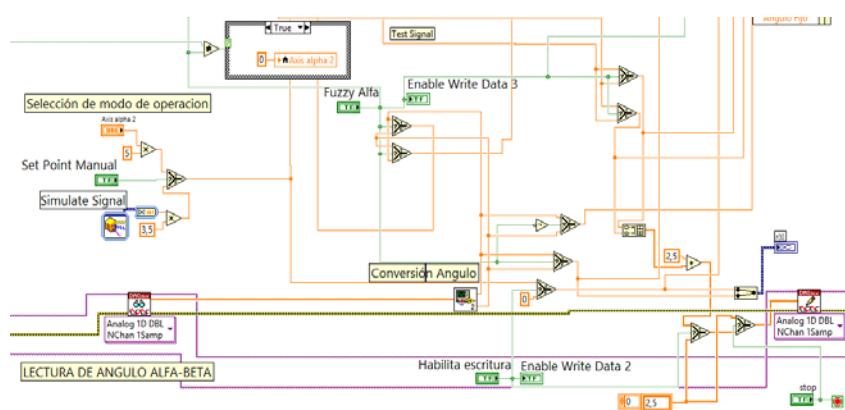
**Figura 4.17 Programación para el desplazamiento de las funciones de membresía del algoritmo de control difuso diseñado.**

Esta misma metodología se usó para el ángulo fijo que hace de pivote, la diferencia es que se fijó la referencia en cero.

Así también el algoritmo de control está diseñado para que trabaje frente a referencias manuales y automáticas, esto fue necesario y útil, ya que primeramente se verificó el funcionamiento del algoritmo difuso de manera manual, para ajustar las funciones de membresía hasta encontrar la mejor de las respuestas en cuanto a minimizar el error, y encontrar un tiempo de estabilización del sistema adecuado.

El algoritmo también permite seleccionar que ángulo controlar alfa o beta, al hacer la selección se establece el ángulo de inclinación inicial del plato en cero grados para que la referencia pueda hacer una excursión hasta sus máximos valores permitidos sean estos positivos o negativos.

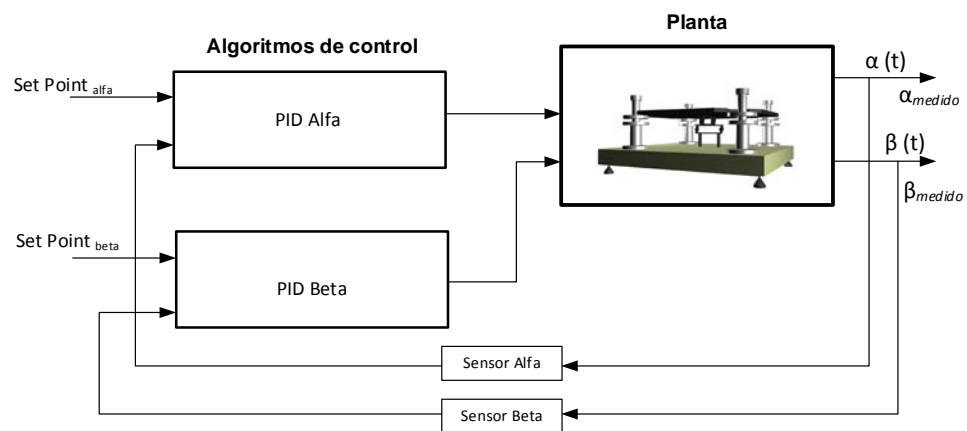
A continuación se muestra la programación del Labview para lograr este objetivo.



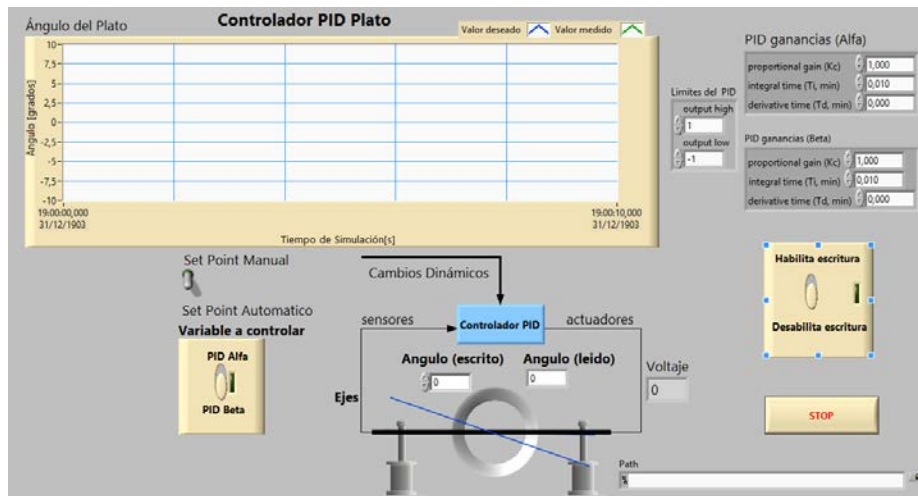
**Figura 4.18 Programación para seleccionar el modo de operación del algoritmo de control**

### 4.3 Algoritmo de Control PID

A continuación se muestra el diagrama de bloques del algoritmo PID para el control de la plataforma del sistema bola y plato, se debe mencionar que la planta de la empresa Feedback tiene un algoritmo proporcional derivativo (PD) de fábrica, el cual es muy cerrado, y no se ajusta a los requerimientos de diseño requeridos para establecer comparativas de rendimiento, es por esta razón que se diseñó un nuevo algoritmo para el control PID de la planta, tomando como referencia los parámetros proporcional  $K$ , el tiempo integral  $T_i$  y el Tiempo derivativo  $T_d$  dados de fábrica.

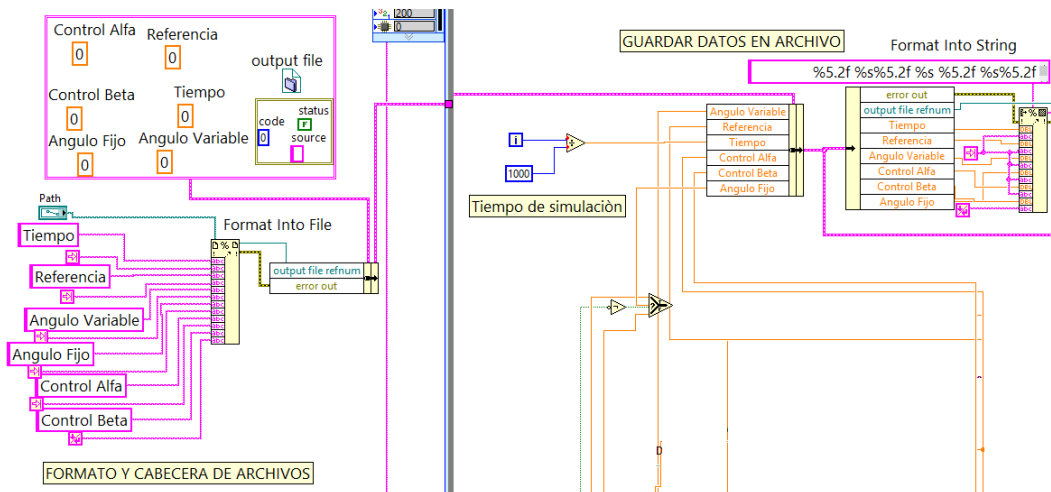


**Figura 4.19 Diagrama de bloques del algoritmo PID para la planta**



**Figura 4.20** Modo visual del algoritmo de control PID de la planta

A continuación se detalla la programación del algoritmo de control PID, identificando sus partes constitutivas:



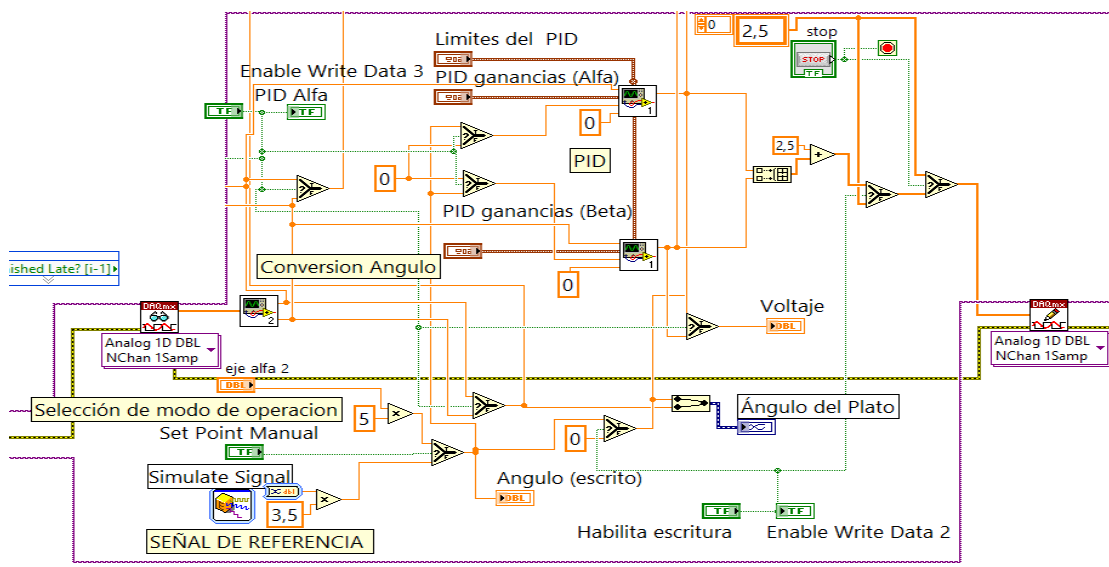
**Figura 4.21** Programación para guardar datos de las variables del PID

Esta parte de la programación del algoritmo PID es muy importante ya que permite guardar los datos de las variables, tiempo, referencia, ángulo variable, ángulo fijo, esfuerzo de control del ángulo alfa, y el del



ángulo beta, todos estos datos luego son exportados y analizados en Matlab, para establecer comparaciones de rendimiento de los algoritmos PD, PID y difuso.

Ahora se muestra la programación concerniente al algoritmo PID, en donde se observa dos controles tanto para el ángulo alfa como para el beta, así también al algoritmo permite el funcionamiento en modo manual y automático. Para establecer comparaciones de rendimiento entre los algoritmos diseñados se usó como señal de excitación una señal cuadrado de amplitud unitaria, y de frecuencia 0,125 Hz. A continuación se muestra:

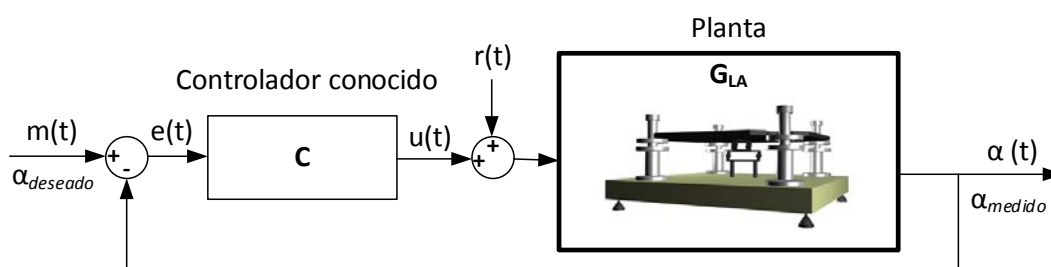


**Figura 4.22 Programación del PID para el control de ángulo**

La señal de referencia puede ser variable, es cuestión de manejar la constante de 3,5 que se fijó para las pruebas. En el próximo capítulo se establecerán comparaciones entre los datos obtenidos.

#### 4.4 Algoritmo de Identificación

Debido a la alta no linealidad del sistema bola y plato, y debido a que su funcionamiento en lazo abierto presenta una alta inestabilidad, el sistema propuesto se identificó de manera indirecta, es decir en lazo cerrado. A continuación se describe gráficamente el proceso realizado.



**Figura 4.23 Diagrama de bloques del algoritmo de identificación de la planta en lazo cerrado**

Es necesario establecer o conocer el controlador  $C$ , ya que si no existiera el mismo, a lo que se aplica voltaje a los inductores que hacen levitar el plato, el mismo se golpea bruscamente por su inestabilidad, pudiendo dañarse el equipo en el corto tiempo.

Debido a que este controlador puede introducir una correlación entre el ruido de salida y la señal de control, el modelo identificado podría no ser tan preciso, para eliminar esta correlación se introduce una señal de

excitación adicional  $r(t)$ , que trabaja como una perturbación externa y se suma a la señal de control  $u(t)$ .

Para poder tener éxito en este tipo de identificación, de acuerdo al **teorema de identificación en lazo cerrado** la señal  $m(t)$  debe ser una excitación persistente, y para lograr esto se debe anular esta entrada, y trabajar solamente con la señal de excitación  $r(t)$ .

El análisis matemático de este proceso de identificación en lazo cerrado se explicó en el capítulo 2. Es válido mencionar que se ha utilizado el controlador PD que el fabricante del módulo facilitó en el manual de usuario del equipo, ya que si no se conoce el controlador es imposible realizar la identificación en lazo cerrado siguiendo esta metodología.

Algo que también hay que mencionar es que la calidad del proceso de identificación en lazo cerrado depende de que tan óptimo sea este controlador, y de elegir una adecuada señal de excitación  $r(t)$ .

A continuación se explica en detalle el algoritmo realizado en Labview para realizar la identificación de la planta anteriormente descrita:

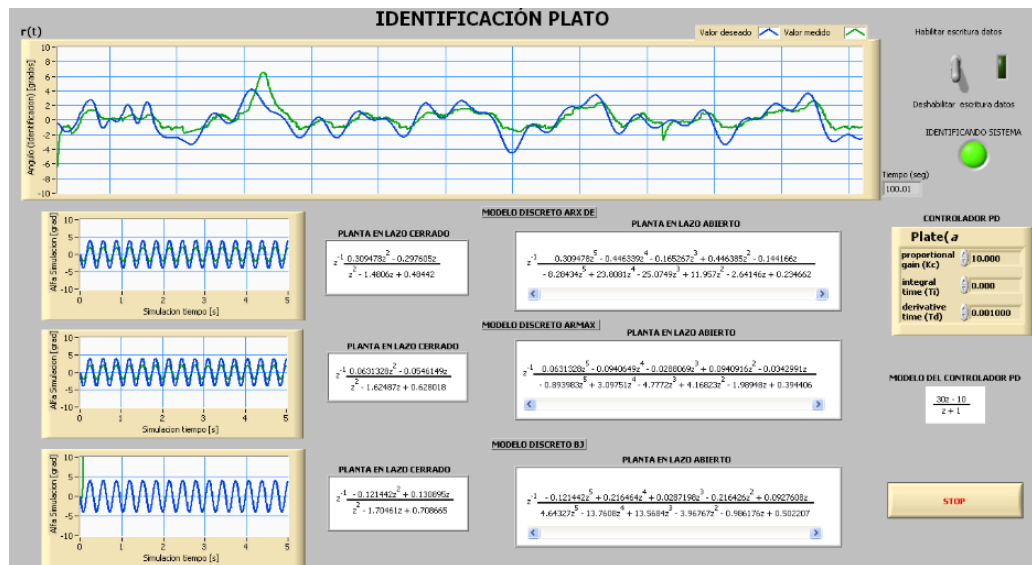


Figura 4.24 Modo visual del algoritmo de identificación de la planta

Inicialmente se intentó identificar el sistema utilizando una señal binaria pseudorandómica pero no se logró el objetivo debido a la alta inestabilidad de la planta, la salida de ángulo del sistema se saturó como se muestra a continuación:

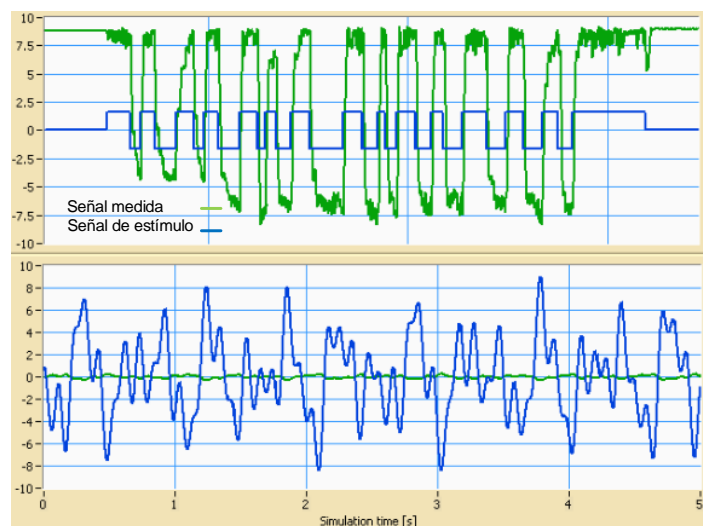
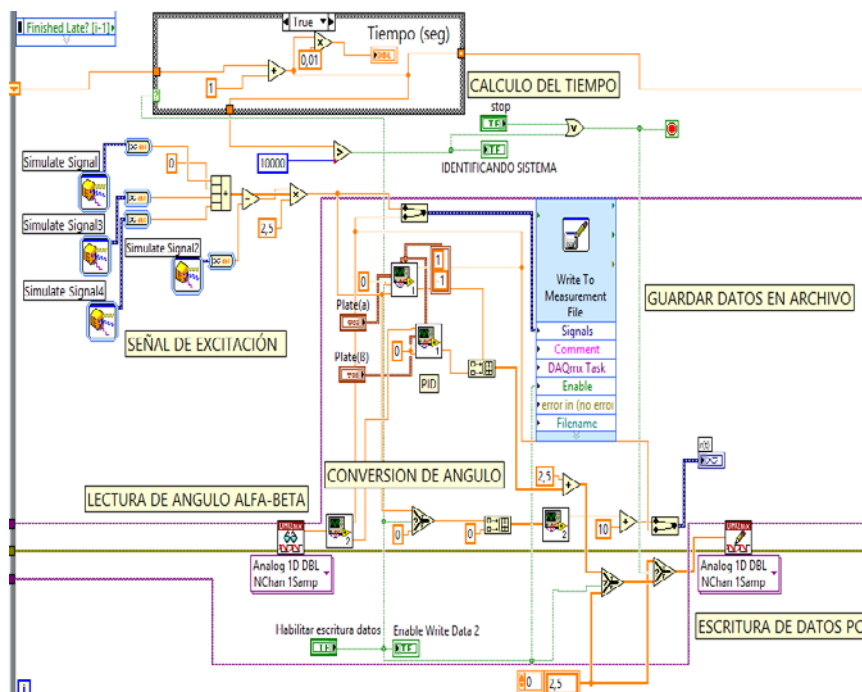


Figura 4.25 Respuesta de la planta ante una entrada PRBS

Se realizaron varias pruebas para intentar realizar la identificación en lazo cerrado de la planta utilizando una señal de excitación PRBS pero en ninguno de los casos se obtuvo resultados óptimos, esto conllevó al cambio de señal de excitación, debido a la alta inestabilidad del sistema se decidió usar una señal que varíe más suave en el tiempo, después de varias pruebas se configuró una señal que es la suma de varias señales sinusoidales con amplitudes y frecuencias pequeñas para evitar que el plato se llegue a golpear, y evitar también la saturación de la salida.

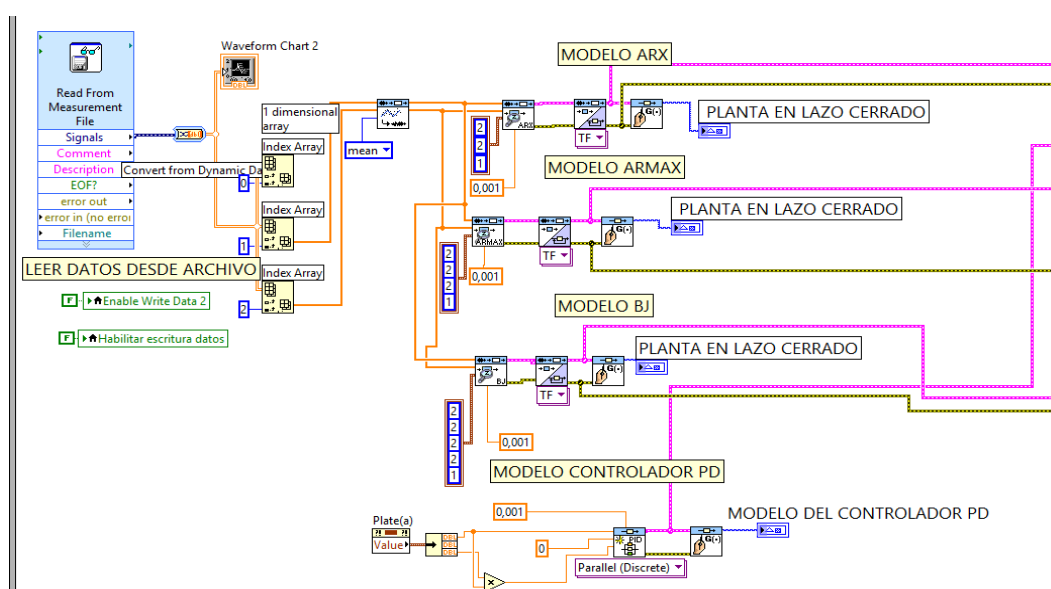
A continuación se muestra la programación efectuada en Labview para la identificación del sistema en estudio:



**Figura 4.26 Primera etapa del algoritmo para la identificación de la planta**

En esta primera etapa se programó la señal de excitación, como la suma de cuatro señales sinusoidales y se las multiplicó por un escalar, el cual luego de varias pruebas se lo fijó en 2,5 el valor del escalar que sirve para variar la amplitud de la señal de excitación. Así también se programó el tiempo de identificación en aproximadamente 100 segundos, y se registró la lectura de los ángulos alfa y beta para que estos datos sean usados en la etapa siguiente del programa de identificación de la planta.

A continuación se muestra la siguiente etapa del programa de identificación de la planta:



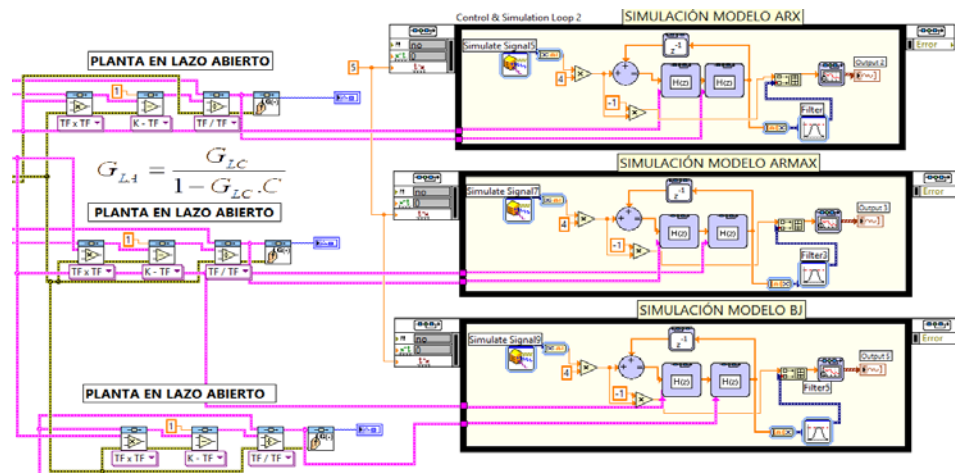
### **Figura 4.27 Segunda etapa del algoritmo para la identificación de la planta**

En esta etapa del algoritmo se leen los datos registrados de ángulo de salida y se los separa a través de arreglos, para posterior proceder a remover tendencias como la teoría de identificación lo requiere, para este caso se removió la media. Una vez que los datos están filtrados pasan a cada instrumento virtual (VI) que obtienen los modelos matemáticos más usados, en este caso se usó los VI de identificación paramétrica para los modelos auto regresivo con entrada externa (ARX), auto regresivo de media móvil con entrada externa (ARMAX), y Box Jenkins (BJ).

Con estos tres modelos podemos seleccionar el mejor, es decir el que tenga una mayor aproximación con un menor porcentaje de error, lo cual se explicará en detalle en el capítulo de validación de resultados.

Hasta este punto se tienen los modelos matemáticos de la planta en lazo cerrado, como se explicó anteriormente esta es función de la planta en lazo abierto, y del controlador, el cual también en esta parte tiene un instrumento virtual aplicado para obtener su función de transferencia. Ahora es necesario despejar el modelo de la planta en alzo abierto, que es el interés de aplicar la metodología de identificación matemática indirecta.

A continuación se muestra la programación efectuada para lograr este objetivo:



**Figura 4.28 Tercera etapa del algoritmo para la identificación de la planta**

A continuación se muestra la primera ejecución del algoritmo elaborado para la identificación matemática de la planta, los datos obtenidos serán analizados en más detalle en el capítulo de validación de resultados, donde se mostrará la simulación del proceso usando simulink, y los porcentajes de aproximación en más escenarios.



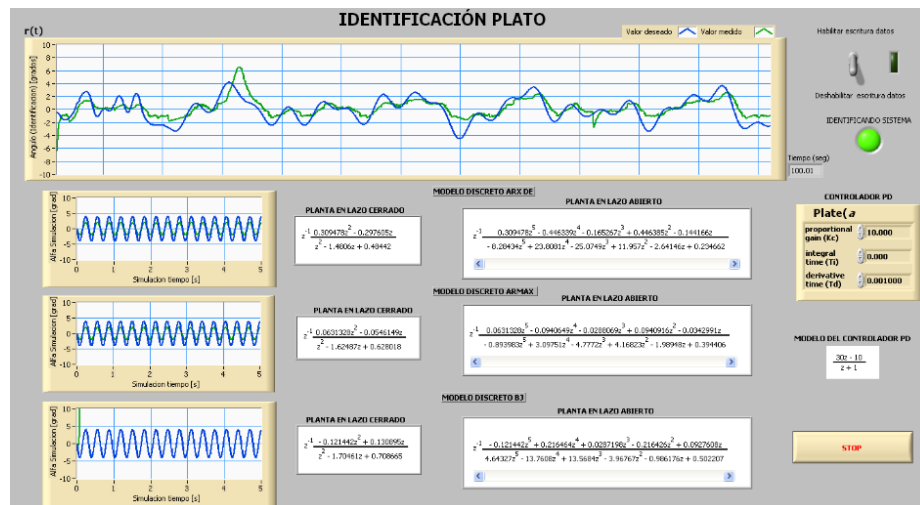


Figura 4.29 Primer escenario de identificación de la planta

De esta primera ejecución se puede observar en las pantallas pequeñas de la izquierda que los modelos ARX y ARMAX son los que presentan un menor porcentaje de error para el seguimiento de una señal sinusoidal, mientras que el BJ no ejecutó una correcta identificación, ya que la traza de color verde se dispara.

En diagrama de bloques el sistema identificado quedaría representado así:

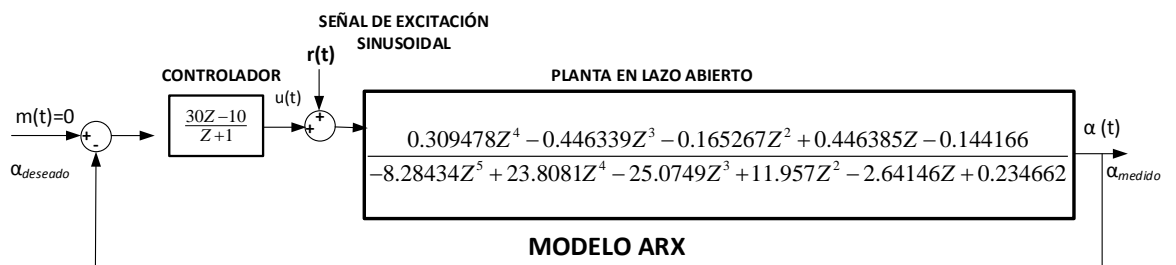
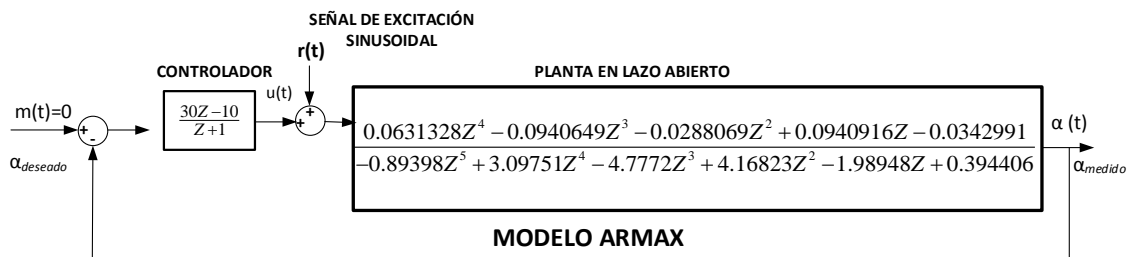


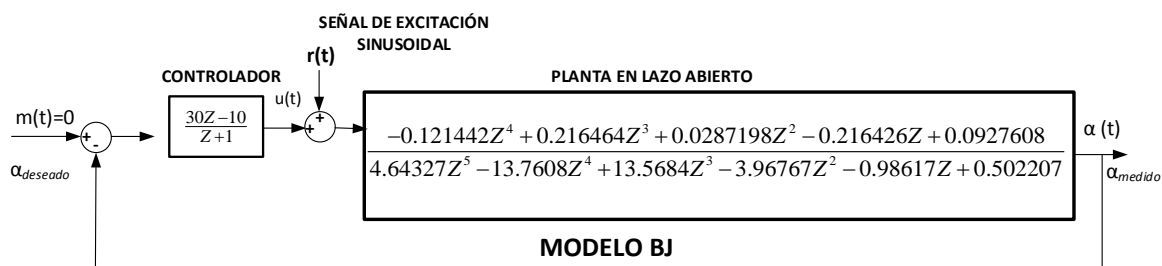
Figura 4.30 Identificación de la planta con el modelo ARX

El modelo ARMAX también resulto ser adecuado para realizar la identificación de la planta, ya que frente a una referencia se observó un seguimiento adecuado. A continuación se muestra cómo quedaría el sistema identificado:



**Figura 4.31 Identificación de la planta con el modelo ARMAX**

Para finalizar esta primera ejecución de identificación de la planta, se muestra también el modelo BJ que no fue satisfactorio.



**Figura 4.32 Identificación de la planta con el modelo BJ**

## **CAPÍTULO 5**

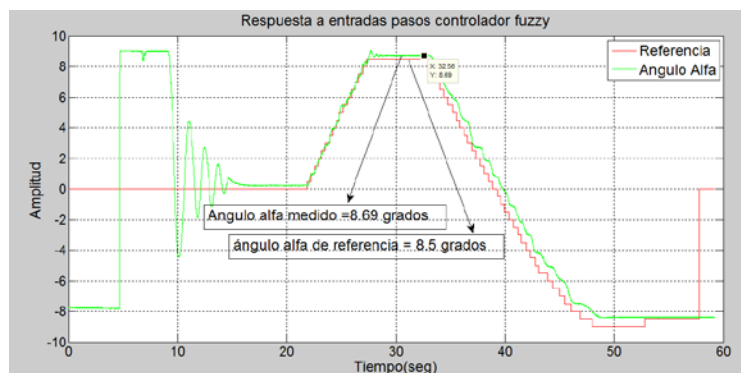
### **ANÁLISIS Y VALIDACIÓN DE RESULTADOS**

#### **5.1 Prueba de funcionamiento del algoritmo de control difuso**

##### **5.1.1 Pruebas en modo manual**

A continuación se muestra la salida del ángulo alfa medido versus el fijado como referencia, con la entrada en modo manual, se hizo una

prueba ascendente y descendente, para este primer análisis alfa es variable y beta fijo.



**Figura 5.1 Respuesta del ángulo alfa ante entradas tipo paso, algoritmo difuso, modo manual**

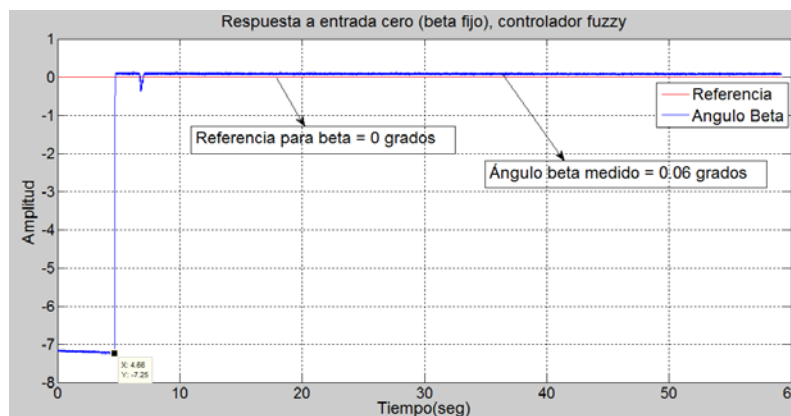
Así también se muestra cuantitativamente que tan óptimo es el seguimiento de la referencia.

**Tabla 5.1 Características del ángulo alfa ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual**

<b>Controlador difuso-ángulo alfa variable</b>	
Ángulo alfa de referencia (grados)	8.5
Ángulo alfa medido (grados)	8.69
% de error de aproximación	2.18%
Tiempo de aproximación (seg)	2.64

Como se describió anteriormente el plato levita por el campo electromagnético que generan 4 inductores los cuales funcionan en parejas, para este caso en que alfa es variable, el ángulo beta debe

estar fijo en posición horizontal, es decir en la referencia de cero grados, como se muestra a continuación:



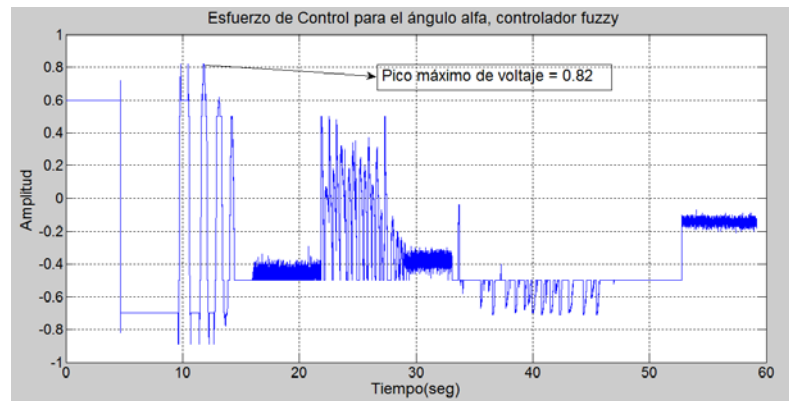
**Figura 5.2 Respuesta del ángulo beta para fijarse en cero grados, algoritmo difuso, modo manual**

Así también se muestra cuantitativamente que tan óptimo es el seguimiento de la referencia fija del ángulo beta.

**Tabla 5.2 Características del ángulo beta para una referencia fija de cero grados, algoritmo de control difuso, modo manual**

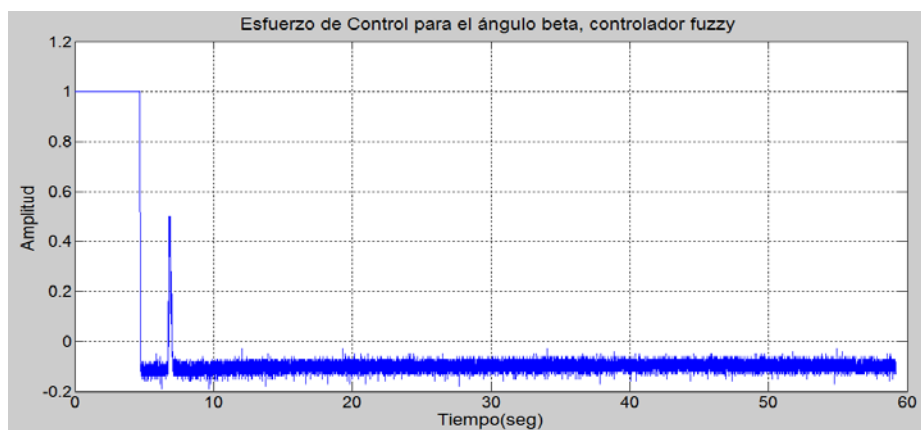
Controlador difuso-ángulo beta fijo	
Ángulo beta de referencia (grados)	0
Ángulo beta medido (grados)	0.06
Error absoluto de aproximación	0.06
Tiempo de aproximación (seg)	4.62

A continuación se muestra también el esfuerzo de control para el ángulo alfa al seguir la referencia fijada.



**Figura 5.3 Esfuerzo de control para el ángulo alfa, algoritmo difuso, modo manual**

También se muestra el esfuerzo de control para el ángulo beta que debe seguir la referencia fijada en cero grados, para que de esta forma el ángulo alfa varíe en todo su rango permitido por las dimensiones del equipo.



**Figura 5.4 Esfuerzo de control para el ángulo beta, algoritmo difuso, modo manual**

A continuación se muestra la programación efectuada en Matlab para

```

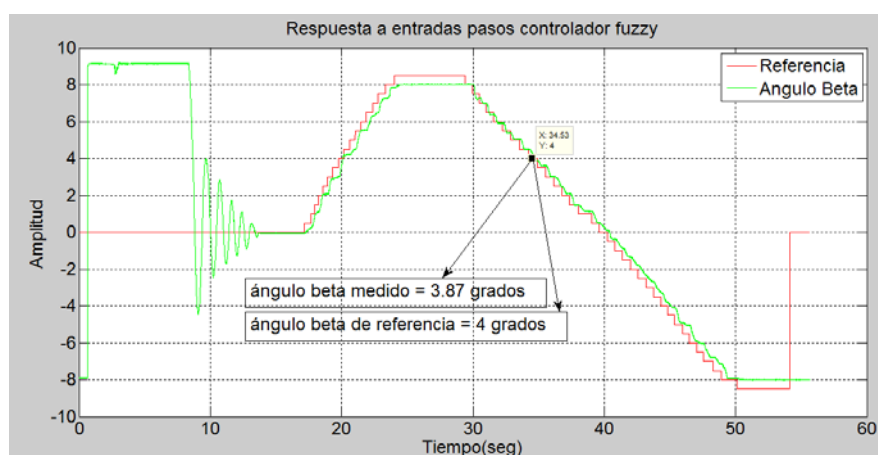
DATOS=xlsread('Dato_fuzzy_manual_alfa.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entradas
pasos controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Alfa')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Beta')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control fuzzy Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')

```

obtener las gráficas anteriormente descritas:

**Figura 5.5 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador difuso, modo manual**

Ahora se muestra el análisis cuando el ángulo beta es variable, y el ángulo alfa es fijo, a continuación se muestra la respuesta del ángulo beta frente a una secuencia tipo paso en la entrada, se establecerá que tan óptimo es el seguimiento de la referencia, así como que tiempo le toma al sistema para llegar a esta. Para lograr este objetivo se ha calculado el porcentaje de error de aproximación, así como el tiempo de aproximación en seguir la referencia fijada.



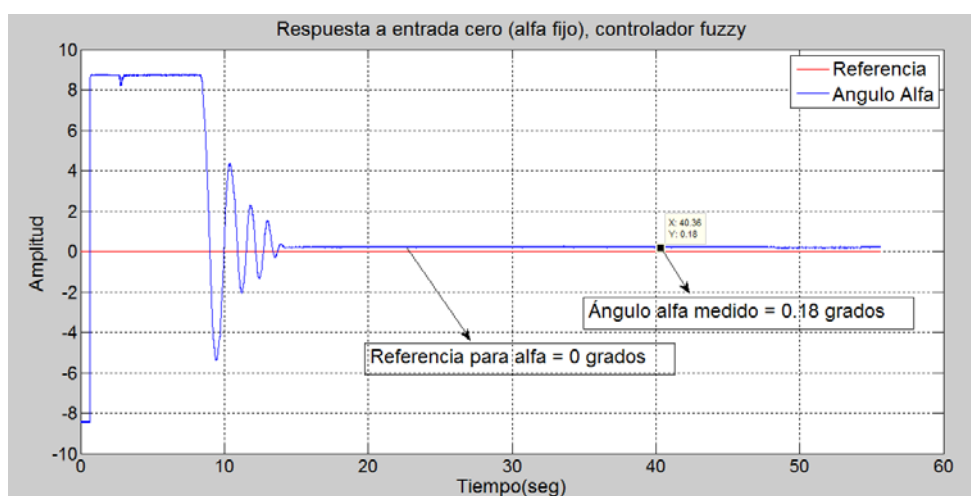
**Figura 5.6 Respuesta del ángulo beta ante entradas tipo paso, algoritmo difuso, modo manual**

**Tabla 5.3 Características del ángulo beta ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual**

<b>Controlador difuso-ángulo beta variable</b>	
Ángulo beta de referencia (grados)	4
Ángulo beta medido (grados)	3.87
% de error de aproximación	3.25%
Tiempo de aproximación (seg)	0.6

En este caso como el ángulo beta es variable, el ángulo alfa debe estar fijo referenciado a cero grados como se muestra a continuación:





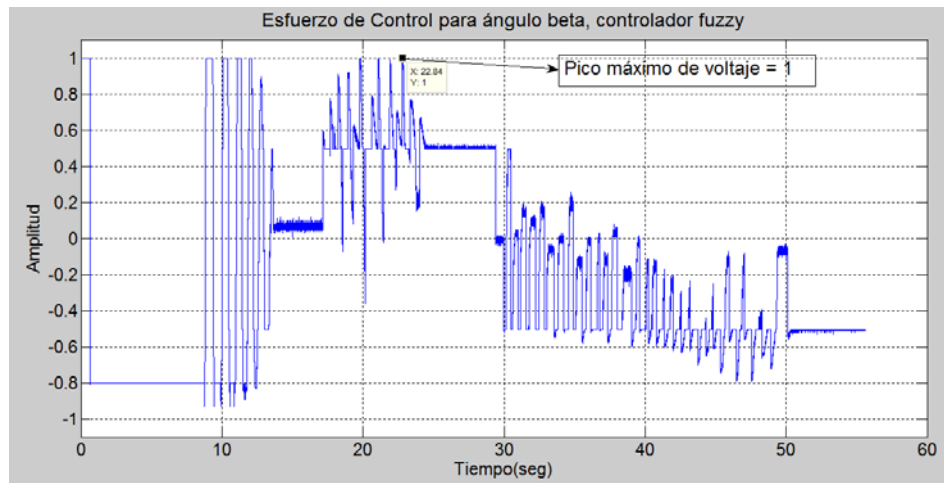
**Figura 5.7 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo difuso, modo manual**

Así también se muestra cuantitativamente que tan óptimo es el seguimiento de la referencia fija del ángulo alfa.

**Tabla 5.4 Características del ángulo beta para una referencia fija de cero grados, algoritmo de control difuso, modo manual**

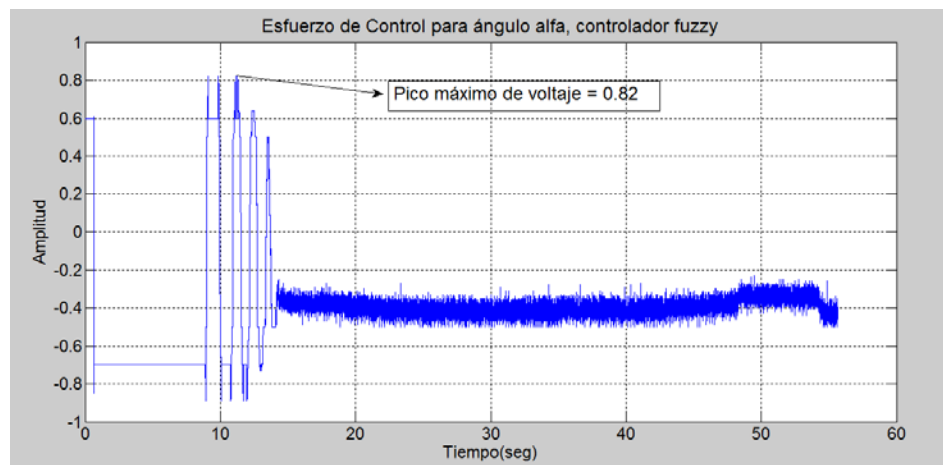
<b>Controlador difuso-ángulo alfa fijo</b>	
Ángulo alfa de referencia (grados)	0
Ángulo alfa medido (grados)	0.18
Error absoluto de aproximación	0.18
Tiempo de aproximación (seg)	14.13

A continuación se muestra también el esfuerzo de control del ángulo beta para seguir la referencia establecida en forma manual.



**Figura 5.8 Esfuerzo de control para el ángulo beta variable, algoritmo difuso, modo manual**

También se muestra el esfuerzo de control para el ángulo alfa, el cual debe seguir la referencia fijada en cero grados.



**Figura 5.9 Esfuerzo de control para el ángulo alfa fijo, algoritmo difuso, modo manual**

A continuación se muestra la programación efectuada en Matlab para graficar las variables descritas anteriormente:

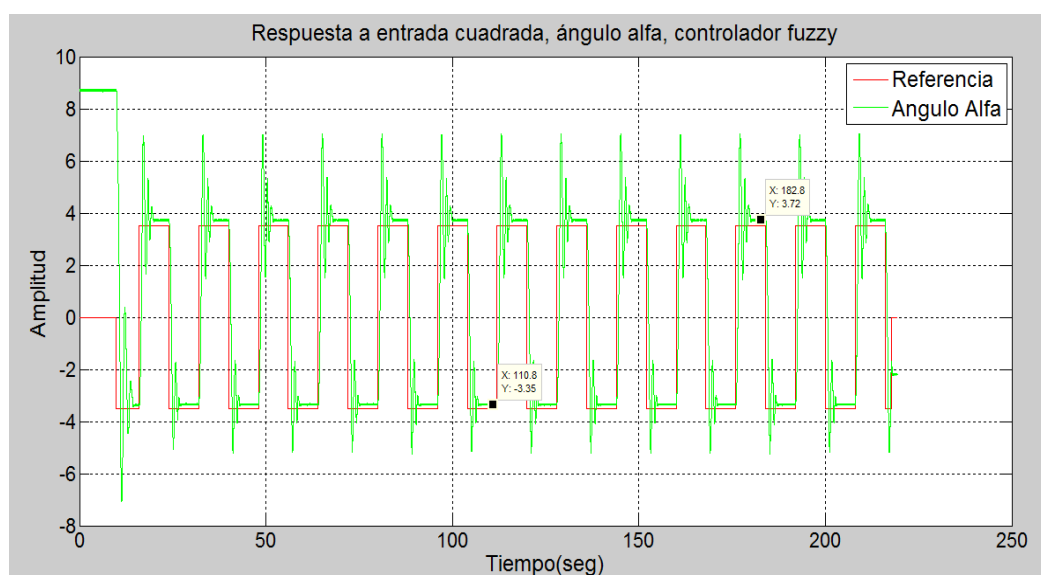
```
DATOS=xlsread('Dato_fuzzy_manual_Beta.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);

figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entradas
pasos controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angulo
Beta')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angulo
Alfa')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control fuzzy Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
```

**Figura 5.10 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador difuso, modo manual**

### 5.1.2 Pruebas en modo automático

Para realizar las pruebas de funcionamiento del algoritmo de control inteligente en modo automático se utilizó como señal de referencia una onda cuadrada de amplitud +/- 3.5, y se analizó la información en Matlab como se muestra a continuación para el ángulo alfa:



**Figura 5.11 Respuesta del sistema ante una entrada cuadrada, ángulo alfa, algoritmo difuso, modo automático**

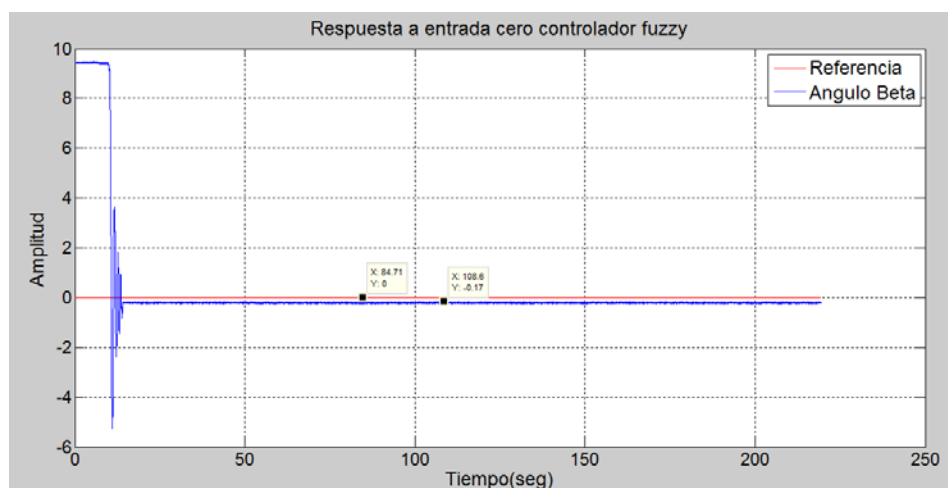
Así también se muestra cuantitativamente que tan óptimo es el seguimiento de la referencia fijada para el ángulo alfa. Aquí luego de varias ejecuciones del algoritmo se llegó a obtener los datos más consensuados para obtener un error y tiempo de aproximación coherentes.

**Tabla 5.5 Características del ángulo alfa para una referencia de entrada tipo onda cuadrada, algoritmo de control difuso**

<b>Controlador difuso-ángulo alfa variable</b>	
Ángulo alfa de referencia positivo (grados)	3.5
Ángulo alfa de referencia negativo (grados)	-3.5
Ángulo alfa positivo medido (grados)	3.72

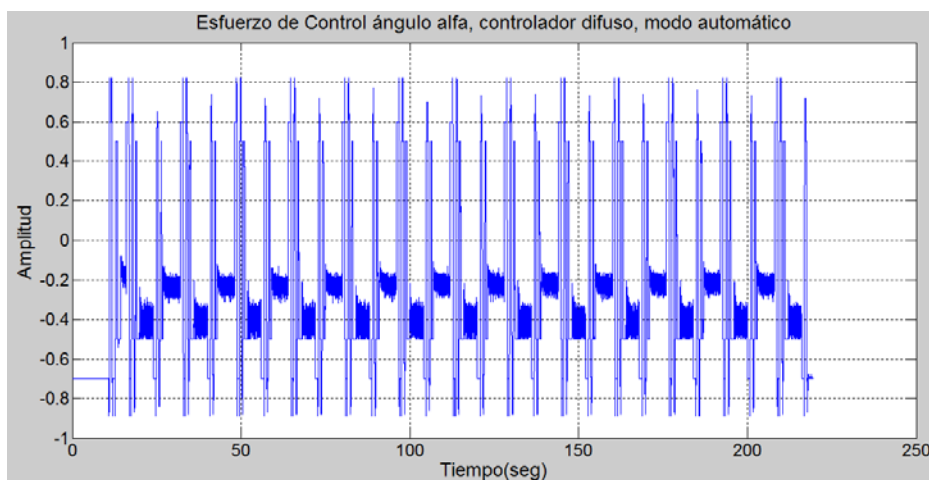
Ángulo alfa negativo medido (grados)	-3.35
Error absoluto de aproximación ciclo positivo	0.22
Error absoluto de aproximación ciclo negativo	0.15
Tiempo de aproximación (segundos)	5.3

En este caso debido a que alfa es el ángulo variable, entonces beta debe quedar fijo en cero grados. A continuación se muestra el seguimiento del ángulo beta:



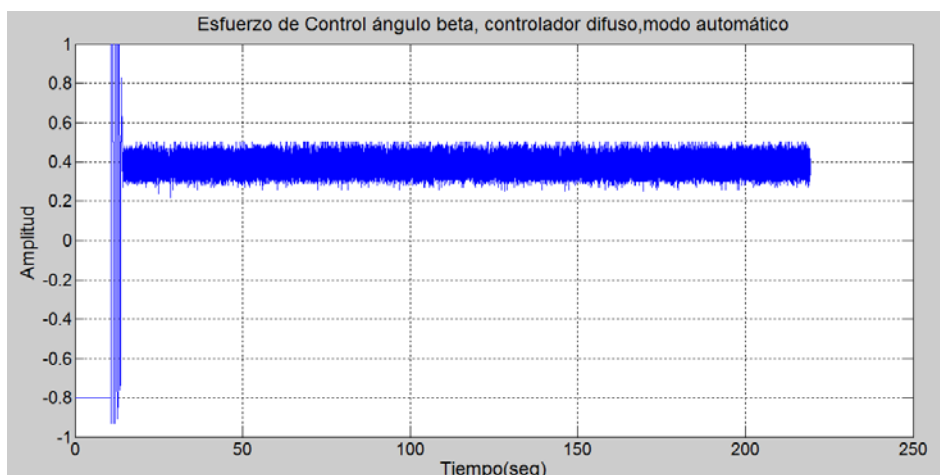
**Figura 5.12 Respuesta del ángulo beta para fijarse en cero grados, algoritmo difuso, modo automático**

Así también se muestra el esfuerzo de control del ángulo alfa para seguir la referencia:



**Figura 5.13 Esfuerzo de control ángulo alfa variable, algoritmo difuso, modo automático**

Ahora se muestra el esfuerzo de control del ángulo beta que está fijo en cero grados.



**Figura 5.14 Esfuerzo de control ángulo beta fijo, algoritmo difuso, modo automático**

La programación en Matlab que permitió exportar los datos desde Labview y graficarlos se detalla a continuación:

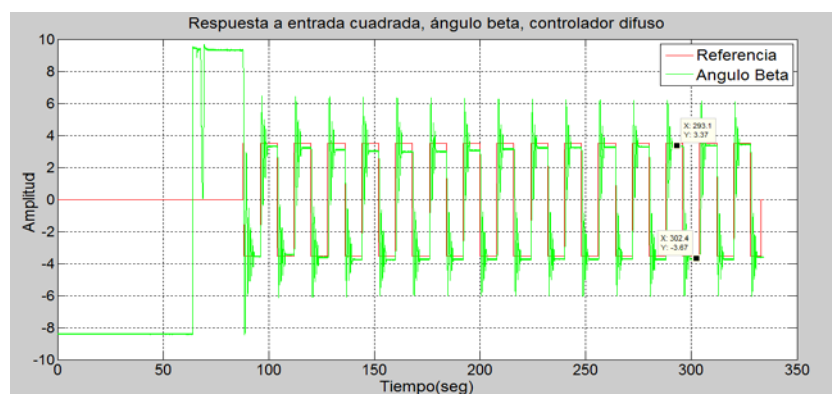
```

DATOS=xlsread('Dato_fuzzy_completo_Alfa.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entrada
cuadrada controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angulo
Alfa')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angulo
Beta')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control fuzzy Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')

```

**Figura 5.15 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador difuso, modo automático**

A continuación se muestra ahora el análisis del algoritmo difuso funcionando en modo automático para el ángulo beta.



**Figura 5.16 Respuesta del sistema ante una entrada cuadrada, ángulo beta variable, algoritmo difuso, modo automático**

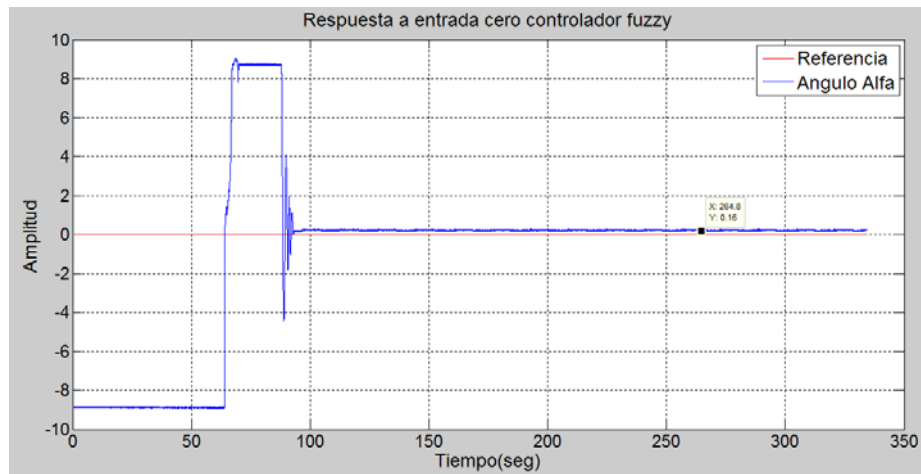
Así también se muestra cuantitativamente que tan óptimo es el seguimiento de la referencia fijada para el ángulo beta. Aquí luego de varias ejecuciones del algoritmo se llegó a obtener los datos más consensuados para obtener un error y tiempo de aproximación coherentes.

**Tabla 5.6 Características del ángulo beta para una referencia de entrada tipo onda cuadrada, algoritmo de control difuso, ángulo beta variable**

<b>Controlador difuso-ángulo beta variable</b>	
Ángulo beta de referencia positivo (grados)	3.5
Ángulo beta de referencia negativo (grados)	-3.5
Ángulo beta positivo medido (grados)	3.37
Ángulo beta negativo medido (grados)	-3.67
Error absoluto de aproximación ciclo positivo	0.13
Error absoluto de aproximación ciclo negativo	0.17
Tiempo de aproximación (segundos)	5.1

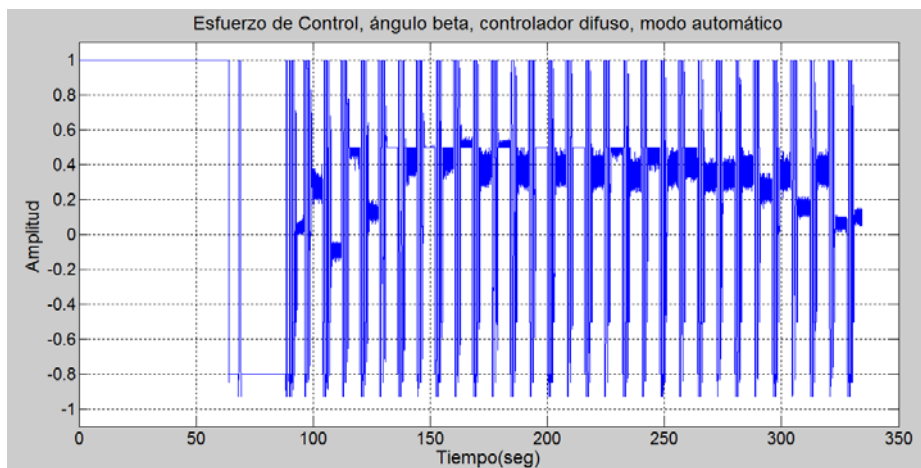
Como el ángulo beta es variable, alfa debe estar fijo en cero grados. A continuación se muestra el detalle:



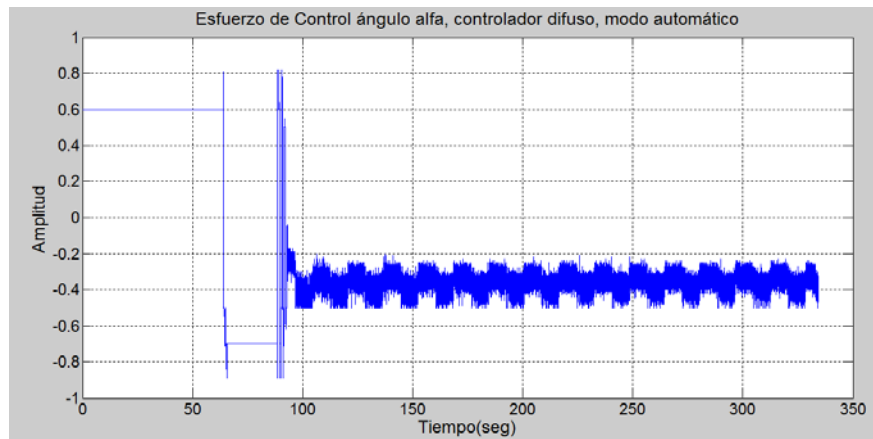


**Figura 5.17 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo difuso, modo automático**

Ahora se muestran los esfuerzos de control para los ángulos alfa y beta, es muy importante conocer esta característica ya que un excesivo esfuerzo de control traería un desgaste del sistema, y una posible reducción de la vida útil del equipo.



**Figura 5.18 Esfuerzo de control ángulo beta variable, algoritmo difuso, modo automático**



**Figura 5.19 Esfuerzo de control ángulo alfa fijo, algoritmo difuso, modo automático**

La programación para obtener las gráficas se muestra a continuación:

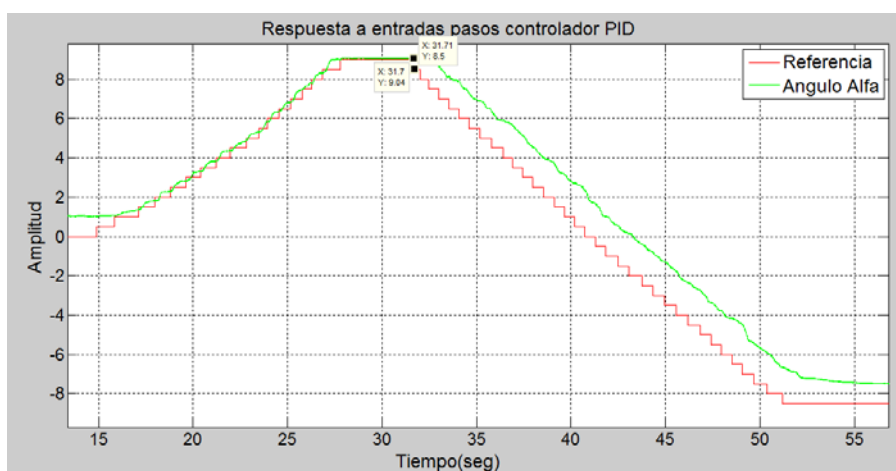
```
DATOS=xlsread('Dato_fuzzy_completo_Beta.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entrada
cuadrada controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Beta')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador fuzzy'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Alfa')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control fuzzy Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
```

**Figura 5.20 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador difuso, modo automático**

## 5.2 Prueba de funcionamiento del algoritmo de control PID

### 5.2.1 Pruebas en modo manual

A continuación se mostrarán las pruebas de funcionamiento que se efectuaron en algoritmo de control PID para el ángulo alfa variable y beta fijo.



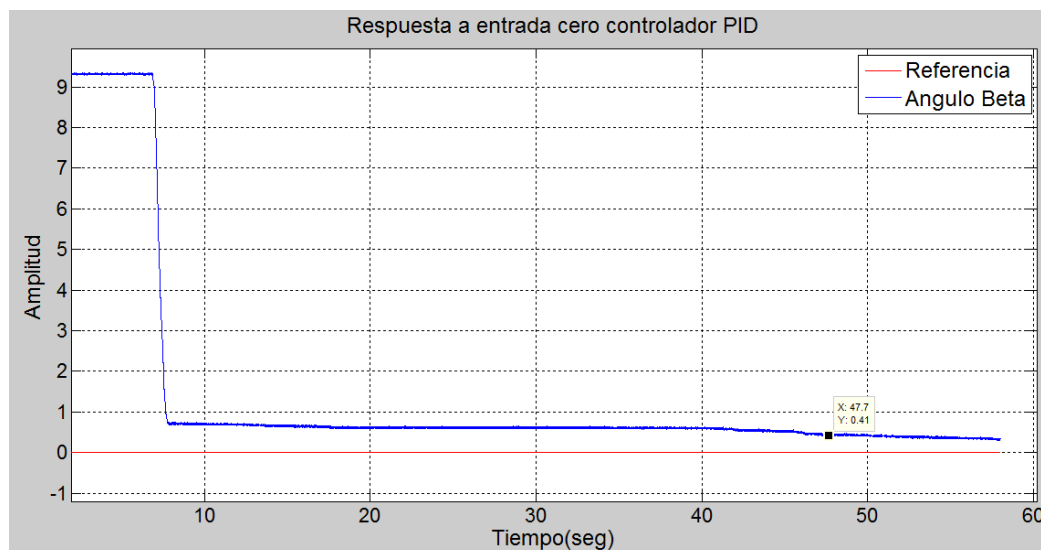
**Figura 5.21** Respuesta del ángulo alfa ante entradas tipo paso, algoritmo PID, modo manual

A continuación se muestran las principales características de esta prueba:

**Tabla 5.7** Características del ángulo alfa ante una secuencia de pasos en la entrada, algoritmo de control PID, modo manual

Controlador PID-ángulo alfa variable	
Ángulo alfa de referencia (grados)	8.5
Ángulo alfa medido (grados)	9.04
% de error de aproximación	5.97%

Como el ángulo alfa es variable, entonces beta dese ser fijo, a continuación se muestra:



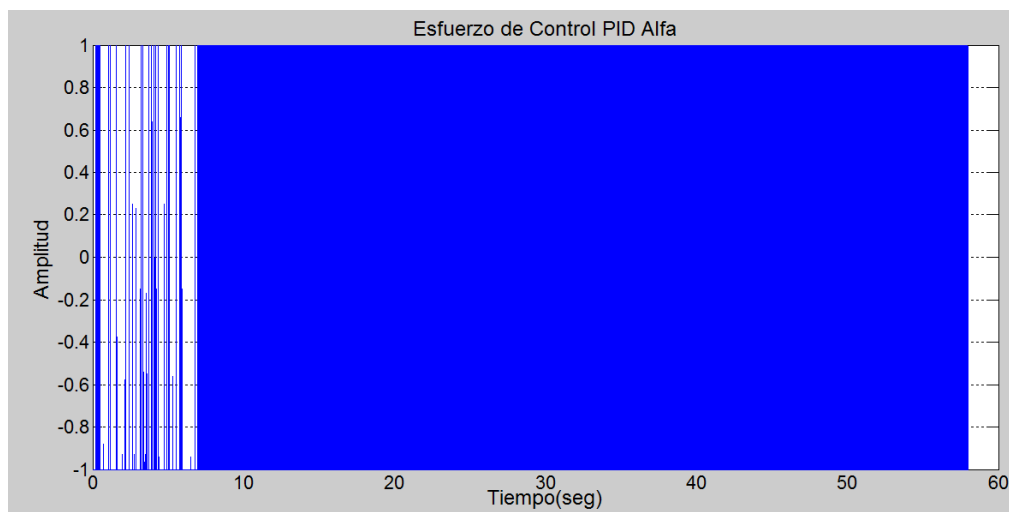
**Figura 5.22 Respuesta del ángulo beta para fijarse en cero grados, algoritmo PID, modo manual**

A continuación se muestran las características del ángulo beta para el seguimiento de la referencia de cero grados

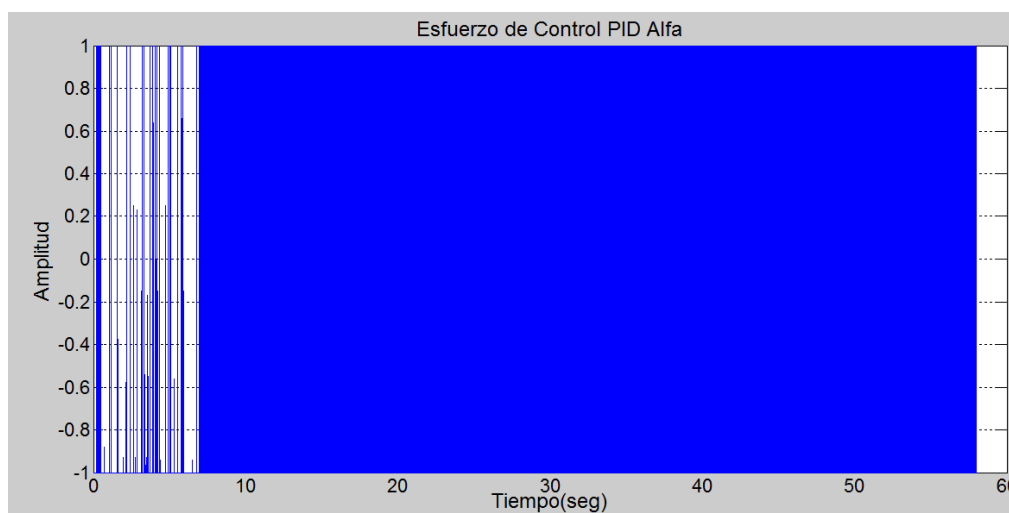
**Tabla 5.8 Características del ángulo beta para una referencia fija de cero grados, algoritmo PID, modo manual**

<b>Controlador PID-ángulo beta fijo</b>	
Ángulo beta de referencia (grados)	0
Ángulo beta medido (grados)	0.41
Error absoluto de aproximación	0.41
Tiempo de aproximación (seg)	47.7

Ahora se muestran los esfuerzos de control tanto para el ángulo alfa como para el ángulo beta.



**Figura 5.23 Esfuerzo de control para el ángulo alfa variable, algoritmo PID, modo manual**



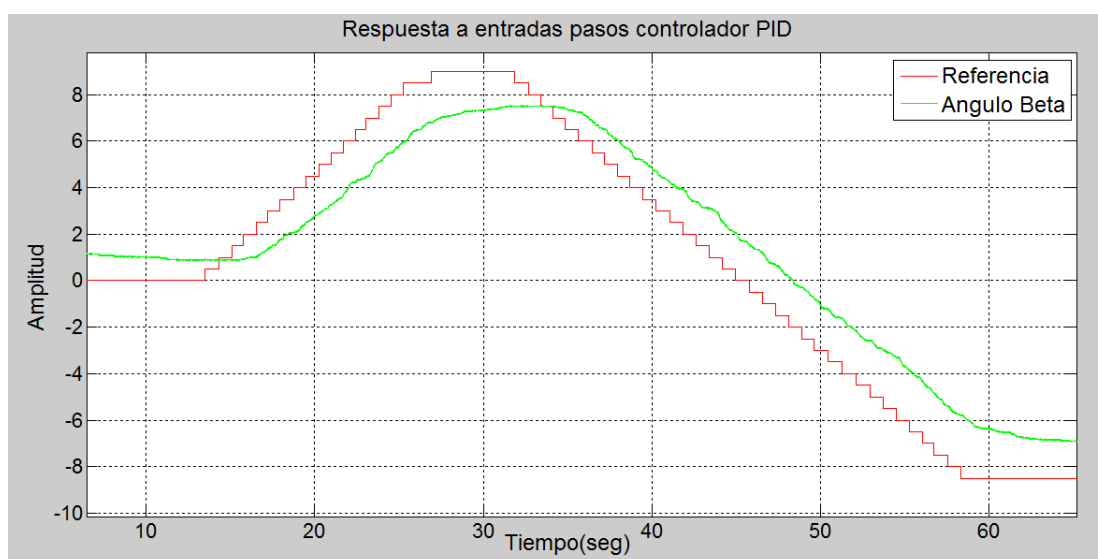
**Figura 5.24 Esfuerzo de control para el ángulo beta fijo, algoritmo PID, modo manual**

A continuación se muestra la programación efectuada en Matlab para exportar los datos desde Labview y obtener las gráficas anteriormente mostradas.

```
DATOS=xlsread('Dato_PID_completo_Alfa.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entrada
cuadrada controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Alfa')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Beta')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control PID Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control PID Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
```

**Figura 5.25 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador PID, modo manual**

Ahora se mostrarán las pruebas efectuadas al algoritmo PID en modo manual para el ángulo beta variable y alfa fijo.



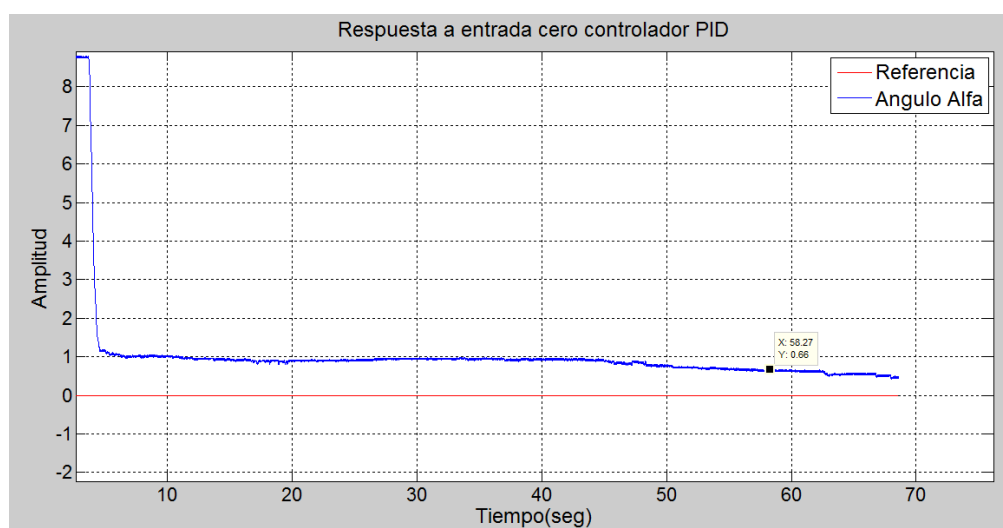
**Figura 5.26 Respuesta del ángulo beta ante entradas tipo paso, algoritmo PID, modo manual**

A continuación se muestran las principales características de esta prueba:

**Tabla 5.9 Características del ángulo beta ante una secuencia de pasos en la entrada, algoritmo de control difuso, modo manual**

<b>Controlador difuso-ángulo beta variable</b>	
Ángulo beta de referencia (grados)	9
Ángulo beta medido (grados)	7.35
% de error de aproximación	18.33%
Tiempo de aproximación (seg)	2.93

Ahora como beta es variable, el ángulo alfa debe seguir la referencia de cero grados. A continuación se muestra:



**Figura 5.27 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo PID, modo manual**

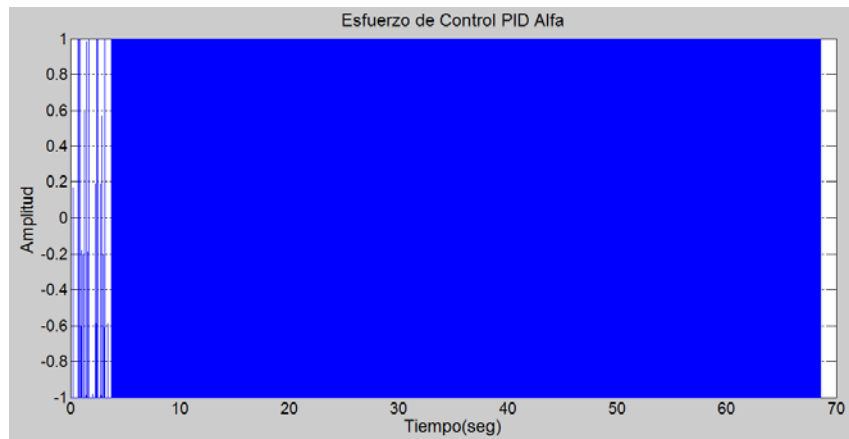
A continuación se muestran las características del ángulo alfa para el seguimiento de la referencia de cero grados

**Tabla 5.10 Características del ángulo alfa para una referencia fija de cero grados, algoritmo PID, modo manual**

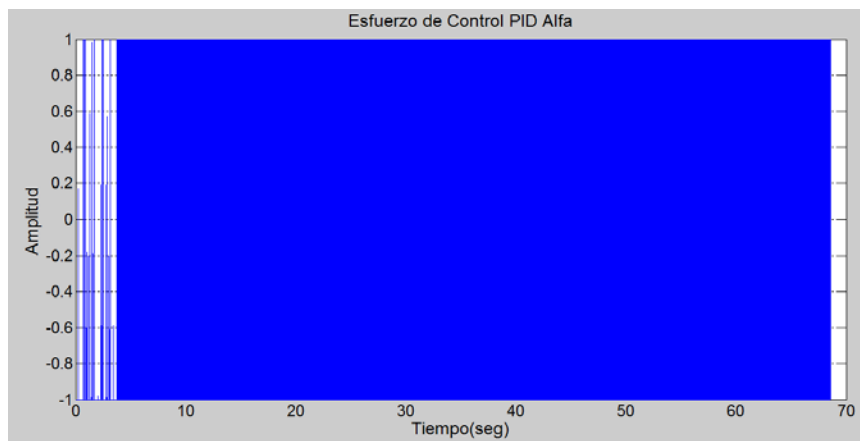
<b>Controlador difuso-ángulo alfa fijo</b>	
Ángulo alfa de referencia (grados)	0
Ángulo alfa medido (grados)	0.66
Error absoluto de aproximación	0.66
Tiempo de aproximación (seg)	58.27

Para finalizar esta prueba se muestran los esfuerzos de control tanto para el ángulo alfa como para el ángulo beta:





**Figura 5.28 Esfuerzo de control para el ángulo alfa, algoritmo PID, modo manual**



**Figura 5.29 Esfuerzo de control para el ángulo beta, algoritmo PID, modo manual**

A continuación se muestra la programación efectuada en Matlab para desarrollo de las gráficas mostradas anteriormente.

```

DATOS=xlsread('Dato_PID_completo_Beta.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);

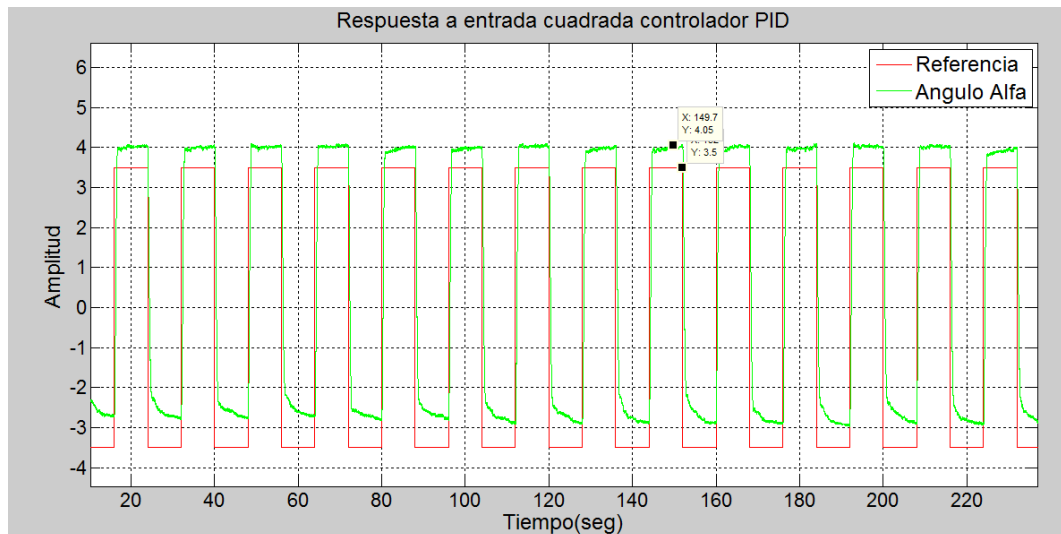
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entrada
cuadrada controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Beta')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Alfa')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control PID Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control PID Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')

```

**Figura 5.30 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador PID, modo manual**

## 5.2.2 Pruebas en modo automático

Para efectuar las pruebas de funcionamiento del algoritmo PID en modo automático se utilizó como señal de referencia una onda cuadrada de 3.5 unidades de amplitud, se obtuvieron los siguientes resultados:



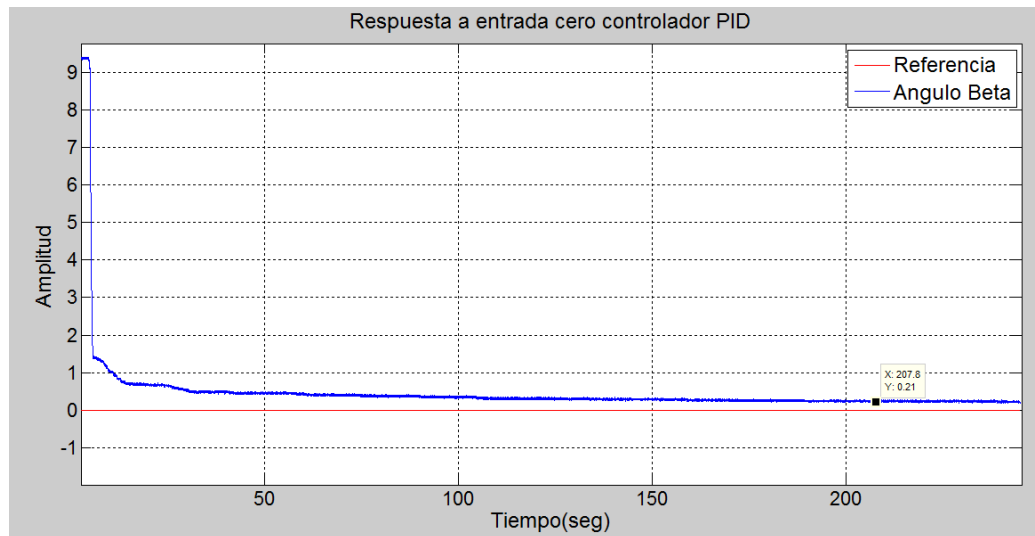
**Figura 5.31 Respuesta del sistema ante una entrada cuadrada, ángulo alfa, algoritmo PID**

Las principales características de esta prueba se muestran a continuación:

**Tabla 5.11 Características del ángulo alfa para una referencia de entrada tipo onda cuadrada, algoritmo de control PID**

Controlador PID-ángulo alfa variable	
Ángulo alfa de referencia (grados)	3.5
Ángulo alfa medido (grados)	4.05
Error absoluto de aproximación	13.58
Tiempo de aproximación (seg)	2.3

Ahora se muestra el seguimiento del ángulo beta a la referencia de cero grados, ya que alfa está variando.

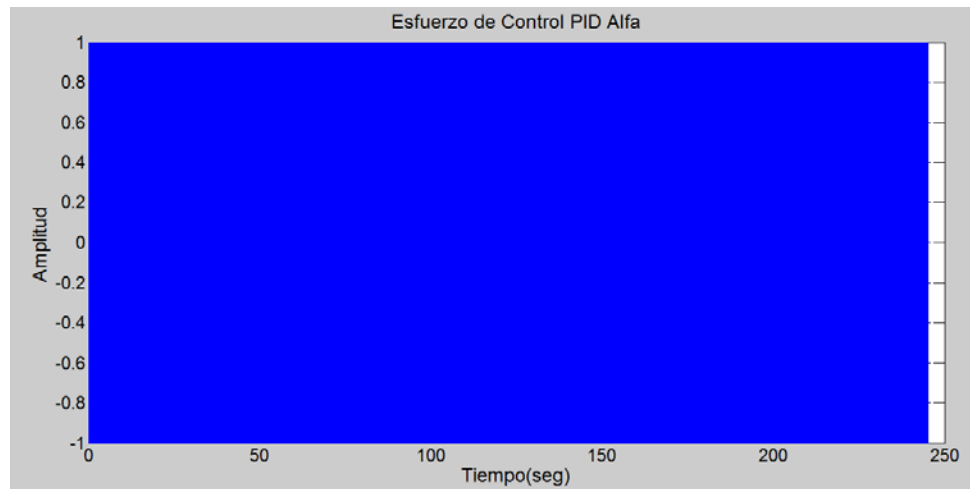


**Figura 5.32 Respuesta del ángulo beta para fijarse en cero grados, algoritmo PID, modo automático**

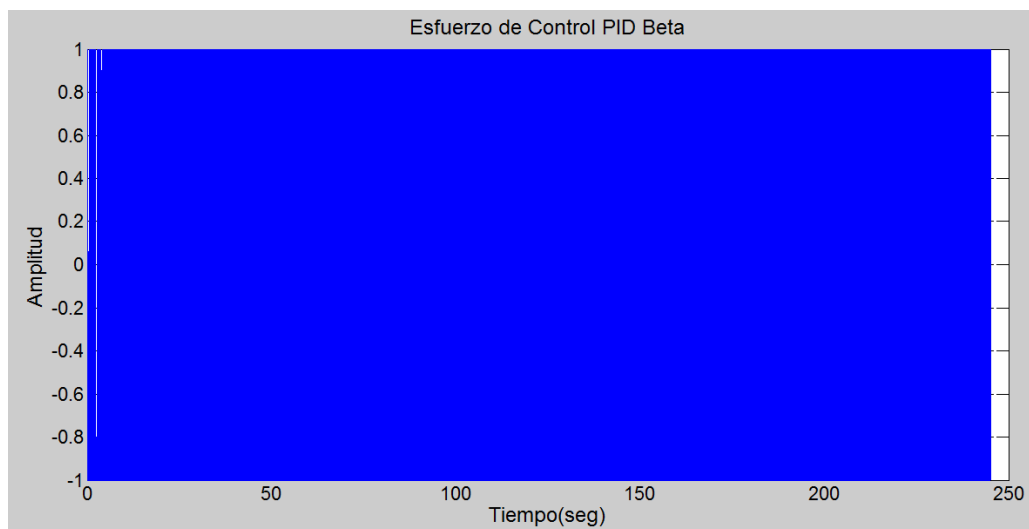
**Tabla 5.12 Características del ángulo beta para una referencia fija de cero grados, algoritmo PID, modo automático**

Controlador difuso-ángulo beta fijo	
Ángulo beta de referencia (grados)	0
Ángulo beta medido (grados)	0.21
Error absoluto de aproximación	0.21
Tiempo de aproximación (seg)	207.8

Ahora se muestran los esfuerzos de control tanto para el ángulo alfa como para el beta.



**Figura 5.33 Esfuerzo de control para el ángulo alfa, algoritmo PID, modo automático**



**Figura 5.34 Esfuerzo de control para el ángulo beta, algoritmo PID, modo automático**

La programación efectuada en Matlab, para el desarrollo de las gráficas mostradas se indica a continuación:

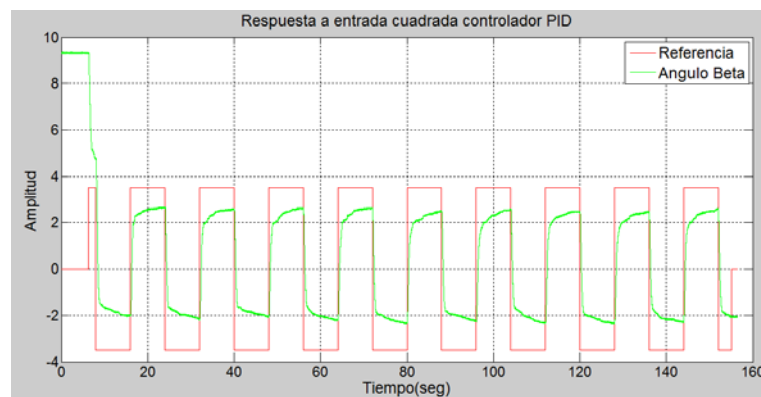
```

DATOS=xlsread('Dato_PID_completo_Alfa.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entrada
cuadrada controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Alfa')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Beta')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control PID Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control PID Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')

```

**Figura 5.35 Programación en Matlab para la graficación de variables que manejan en ángulo alfa, controlador PID, modo automático**

De modo equivalente se muestra las respuestas del algoritmo de control PID en modo automático para el ángulo beta.



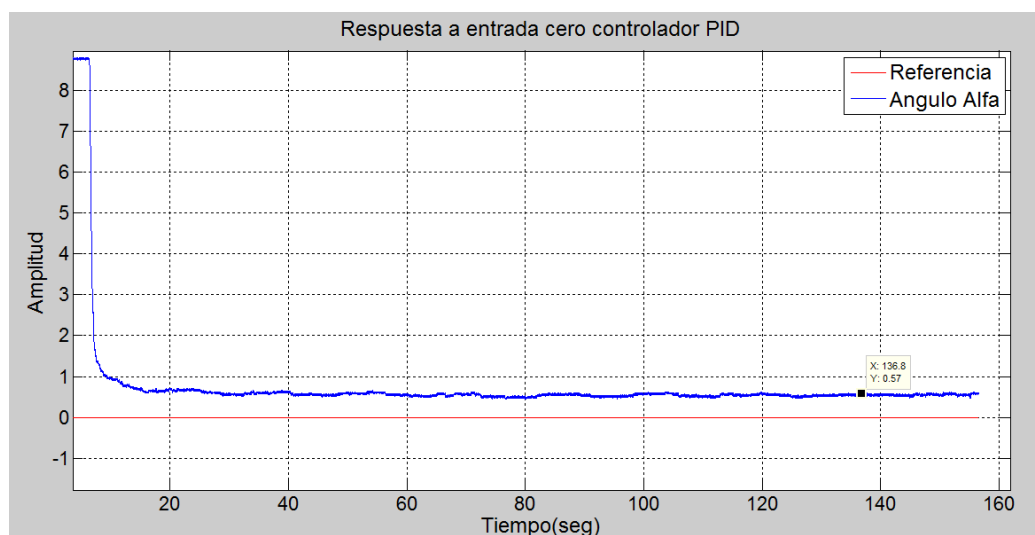
**Figura 5.36 Respuesta del sistema ante una entrada cuadrada, ángulo beta, algoritmo PID**

Las principales características de esta prueba se muestran a continuación:

**Tabla 5.13 Características del ángulo beta para una referencia de entrada tipo onda cuadrada, algoritmo de control PID**

Controlador PID-ángulo beta variable	
Ángulo beta de referencia (grados)	3.5
Ángulo beta medido (grados)	2.5
Error absoluto de aproximación	28.57
Tiempo de aproximación (seg)	7.73

Ahora se muestra como el ángulo alfa sigue la referencia de cero grados, ya que en este caso beta deber ser variable:

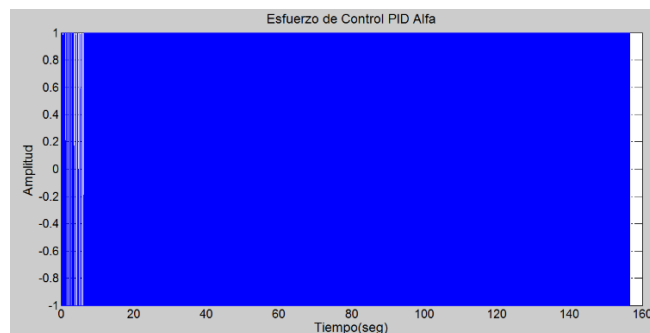


**Figura 5.37 Respuesta del ángulo alfa para fijarse en cero grados, algoritmo PID, modo automático**

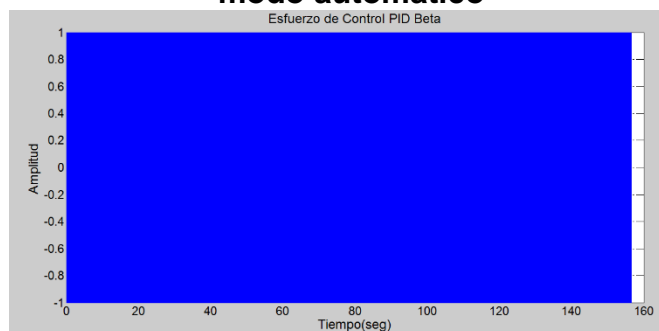
**Tabla 5.14 Características del ángulo alfa para una referencia fija de cero grados, algoritmo PID, modo automático**

<b>Controlador difuso-ángulo alfa fijo</b>	
Ángulo alfa de referencia (grados)	0
Ángulo alfa medido (grados)	0.57
Error absoluto de aproximación	0.57
Tiempo de aproximación (seg)	136.8

Ahora se muestran los esfuerzos de control tanto para el ángulo alfa como para el beta.



**Figura 5.38 Esfuerzo de control para el ángulo alfa, algoritmo PID, modo automático**





### Figura 5.39 Esfuerzo de control para el ángulo beta, algoritmo PID, modo automático

La programación efectuada en Matlab para el análisis mostrado se detalla a continuación:

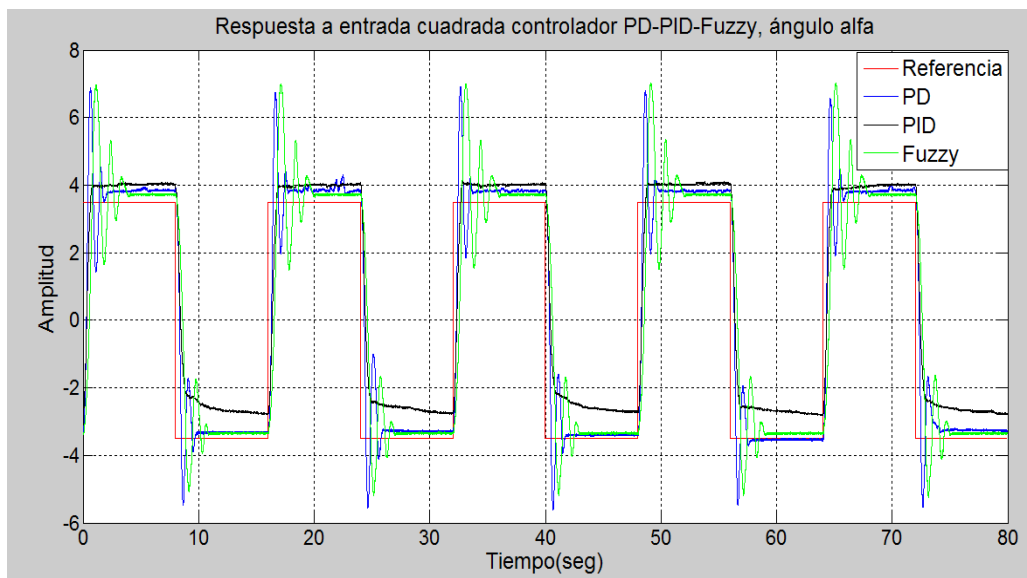
```
DATOS=xlsread('Dato_PID_completo_Beta.XLSX');
t=DATOS(:,1);
ref=DATOS(:,2);
ref2=zeros(length(ref),1);
angulo_variable=DATOS(:,3);
angulo_fijo=DATOS(:,4);
control_alfa=DATOS(:,5);
control_beta=DATOS(:,6);
figure(1)
plot(t,ref,'r',t,angulo_variable,'g'),title('Respuesta a entrada
cuadrada controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Beta')
figure(2)
plot(t,ref2,'r',t,angulo_fijo,'b'),title('Respuesta a entrada cero
controlador PID'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','Angu
lo Alfa')
figure(3)
plot(t,control_alfa),title('Esfuerzo de Control PID Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
figure(4)
plot(t,control_beta),title('Esfuerzo de Control PID Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud')
```

### Figura 5.40 Programación en Matlab para la graficación de variables que manejan en ángulo beta, controlador PID, modo automático

## 5.3 Análisis comparativo de algoritmos de control

En este apartado se muestran la comparación en el rendimiento y eficiencia entre los algoritmos de control difuso, PID, y PD.

### 5.3.1 Análisis para el ángulo de inclinación alfa



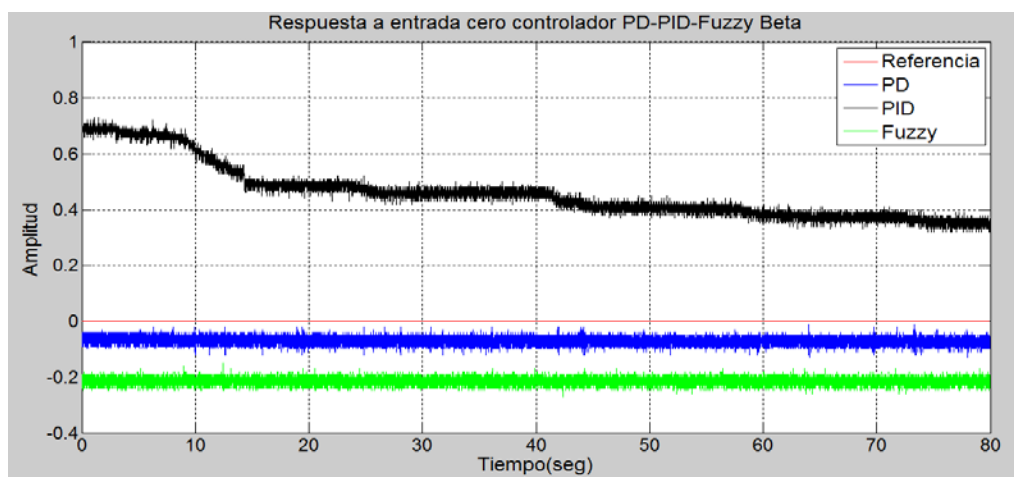
**Figura 5.41 Comparación de rendimiento de algoritmos de control difuso, PD, PID para el control del ángulo de inclinación alfa**

De la gráfica se observa que el algoritmo de control inteligente usando lógica difusa presenta una mejor eficiencia para el seguimiento de la señal de referencia programada, es decir el error de aproximación para el algoritmo difuso es más pequeño que en los otros casos, el tiempo de aproximación en los tres casos es muy similar. A continuación se detallan estas características:

**Tabla 5.15 Características del rendimiento de los algoritmos difuso, PD, PID, ángulo alfa**

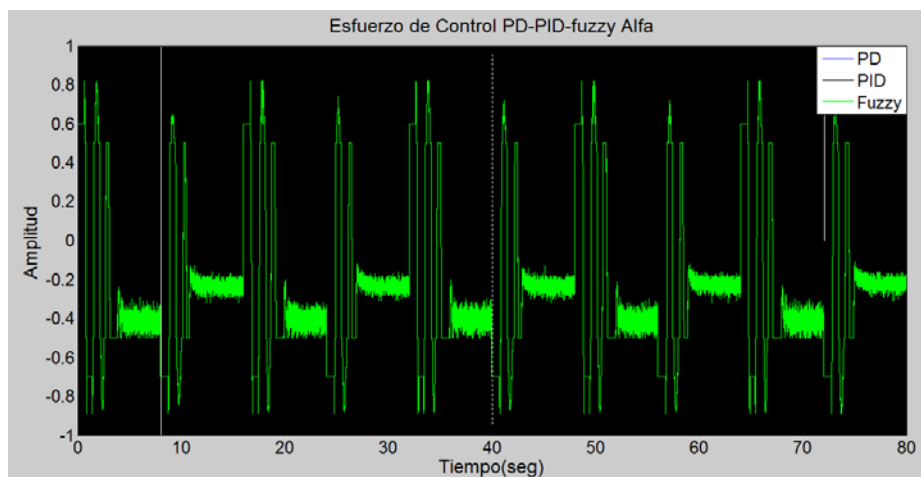
Características de rendimiento	Algoritmo	Algoritmo	Algoritmo
	Difuso	PID	PD
Ángulo alfa de referencia (grados)	3.5	3.5	3.5
Ángulo alfa medido (grados)	3.69	4.06	3.88
Error absoluto de aproximación	5.14%	13.79%	9.79%
Tiempo de aproximación (seg)	6.43	6.43	6.43

Para esta prueba el ángulo de inclinación alfa es variable, por lo cual beta debe estar fijo en cero grados. A continuación se muestra el rendimiento de cada uno de los algoritmos diseñados para establecer el ángulo beta a la referencia de cero grados.

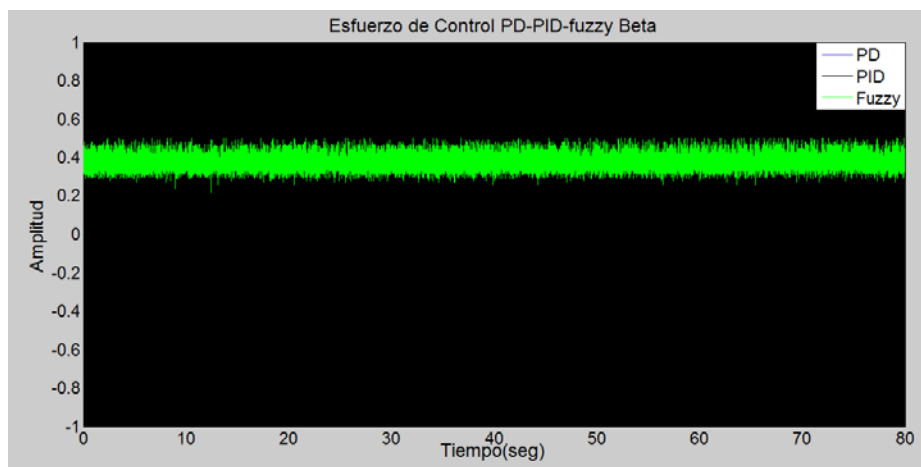


**Figura 5.42 Comparación de rendimiento de algoritmos de control difuso, PD, y PID para el seguimiento de la referencia cero grados del ángulo beta**

Así también en esta prueba se realizó una comparación de los esfuerzos de control para cada uno de los algoritmos anteriormente diseñados para los ángulos alfa y beta.



**Figura 5.43** Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo alfa variable



**Figura 5.44** Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo beta fijo

A continuación se muestra la programación en Matlab para la exportación de los datos desde Labview y el diseño de las gráficas mostradas anteriormente.

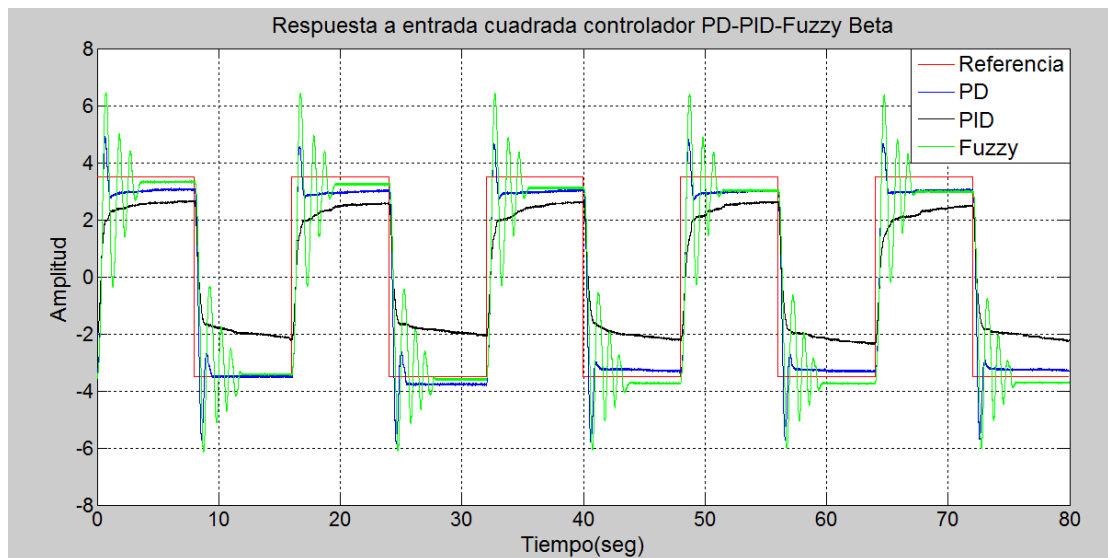
```

DATOS_FUZZY=xlsread('Dato_fuzzy_Alfa.XLSX');
DATOS_PD=xlsread('Dato_PD_Alfa.XLSX');
DATOS_PID=xlsread('Dato_PID_Alfa.XLSX');
tf=DATOS_FUZZY(:,1);
reff=DATOS_FUZZY(:,2);
ref2f=zeros(length(reff),1);
angulo_variablef=DATOS_FUZZY(:,3);
angulo_fijof=DATOS_FUZZY(:,4);
control_alfaf=DATOS_FUZZY(:,5);
control_betaf=DATOS_FUZZY(:,6);
tpd=DATOS_PD(:,1);
refpd=DATOS_PD(:,2);
ref2pd=zeros(length(reff),1);
angulo_variablepd=DATOS_PD(:,3);
angulo_fijopd=DATOS_PD(:,4);
control_alfapd=DATOS_PD(:,5);
control_betapd=DATOS_PD(:,6);
tpid=DATOS_PID(:,1);
refpid=DATOS_PID(:,2);
ref2pid=zeros(length(reff),1);
angulo_variablepid=DATOS_PID(:,3);
angulo_fijopid=DATOS_PID(:,4);
control_alfapid=DATOS_PID(:,5);
control_betapid=DATOS_PID(:,6);
figure(1)
plot(tf,reff,'r',tpd,angulo_variablepd,'b',tpid,angulo_variablepid,
'k',tf,angulo_variablef,'g'),title('Respuesta a entrada cuadrada
controlador PD-PID-Fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','PD',
'PID','Fuzzy')
figure(2)
plot(tf,ref2f,'r',tpd,angulo_fijopd,'b',tpid,angulo_fijopid,'k',tf,
angulo_fijof,'g'),title('Respuesta a entrada cero controlador PD-
PID-Fuzzy Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','PD',
'PID','Fuzzy')
figure(3)
plot(tpd,control_alfapd,'b',tpid,control_alfapid,'k',tf,control_al
faf,'g'),title('Esfuerzo de Control PD-PID-fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('PD','PID','Fuzzy')

```

**Figura 5.45 Programación en Matlab para comparar rendimientos de algoritmos difuso, PID, PD, ángulo alfa**

### 5.3.2 Análisis para el ángulo de inclinación beta



**Figura 5.46 Comparación de rendimiento de algoritmos de control difuso, PD, PID para el control del ángulo de inclinación beta**

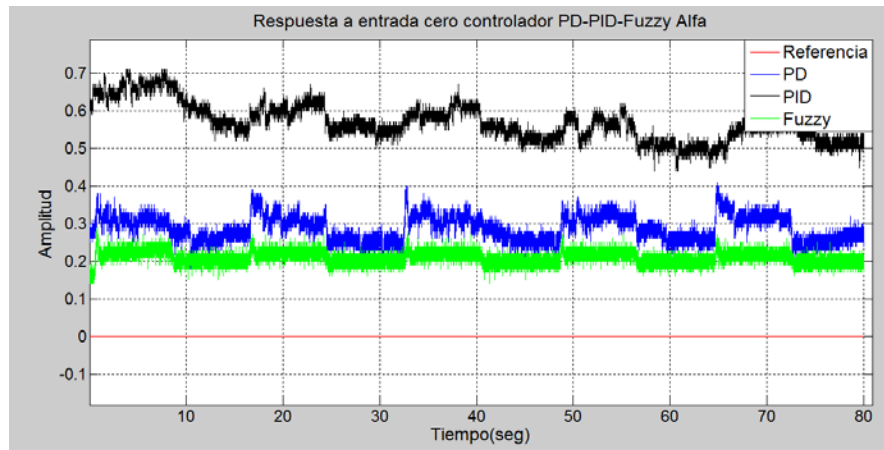
De manera equivalente a como se mostró el rendimiento de los algoritmos de control diseñados para el ángulo alfa, ahora se lo hace para beta.

**Tabla 5.16 Características del rendimiento de los algoritmos difuso, PD, PID, ángulo beta**

Características de rendimiento	Algoritmo	Algoritmo	Algoritmo
	Difuso	PID	PD
Ángulo alfa de referencia (grados)	3.5	3.5	3.5
Ángulo alfa medido (grados)	3.29	2.55	3.01
Error absoluto de aproximación	6	27.14	14

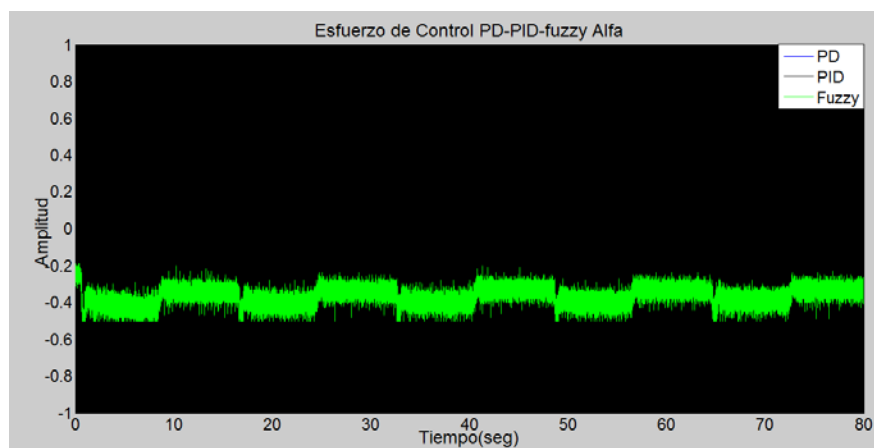
Tiempo de aproximación (seg)	5.78	5.78	5.78
------------------------------	------	------	------

En esta prueba como el ángulo beta es variable, alfa debe estar fijo en la referencia de cero grados. A continuación se muestra:



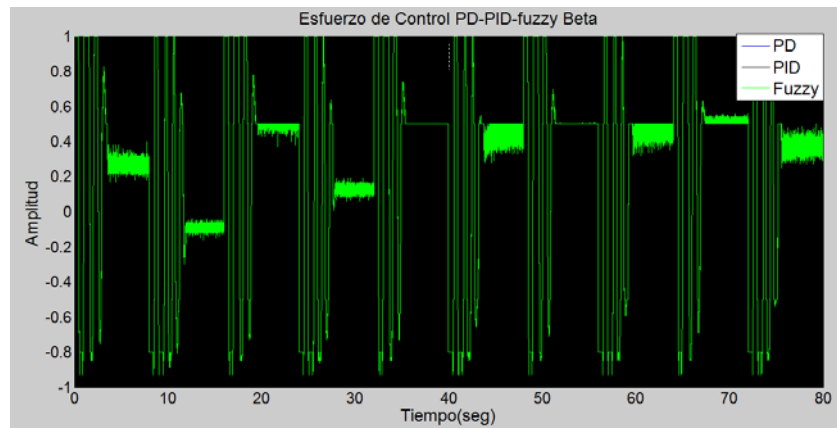
**Figura 5.47 Comparación de rendimiento de algoritmos de control difuso, PD, y PID para el seguimiento de la referencia cero grados del ángulo beta**

Como parte final de esta prueba se muestran los esfuerzos de control para el ángulo alfa, como para el ángulo beta.





**Figura 5.48 Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo beta variable**



**Figura 5.49 Comparación de rendimiento de algoritmos de control difuso, PD, y PID, esfuerzo de control ángulo alfa fijo**

Un resumen de la programación en Matlab, también se la muestra a

```
DATOS_FUZZY=xlsread('Dato_fuzzy_Beta.XLSX');
DATOS_PD=xlsread('Dato_PD_Beta.XLSX');
DATOS_PID=xlsread('Dato_PID_Beta.XLSX');

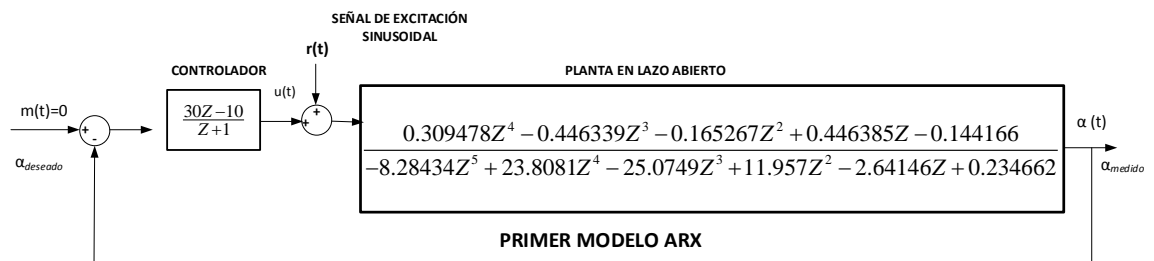
figure(1)
plot(tf,ref,'r',tpd,angulo_variablepd,'b',tpid,angulo_variablepid,
'k',tf,angulo_variablef,'g'),title('Respuesta a entrada cuadrada
controlador PD-PID-Fuzzy Beta'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','PD',
'PID','Fuzzy')
figure(2)
plot(tf,ref2f,'r',tpd,angulo_fijopd,'b',tpid,angulo_fijopid,'k',tf,
angulo_fijof,'g'),title('Respuesta a entrada cero controlador PD-
PID-Fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('Referencia','PD',
'PID','Fuzzy')
figure(3)
plot(tpd,control_alfapd,'b',tpid,control_alfapid,'k',tf,control_al
faf,'g'),title('Esfuerzo de Control PD-PID-fuzzy Alfa'),grid,
xlabel('Tiempo(seg)'),ylabel('Amplitud'),legend('PD','PID','Fuzzy'
)
figure(4)
plot(tpd,control_betapd,'b',tpid,control_betapid,'k',tf,control_be
taf,'g'),title('Esfuerzo de Control PD-PID-fuzzy Beta'),grid,
```

continuación:

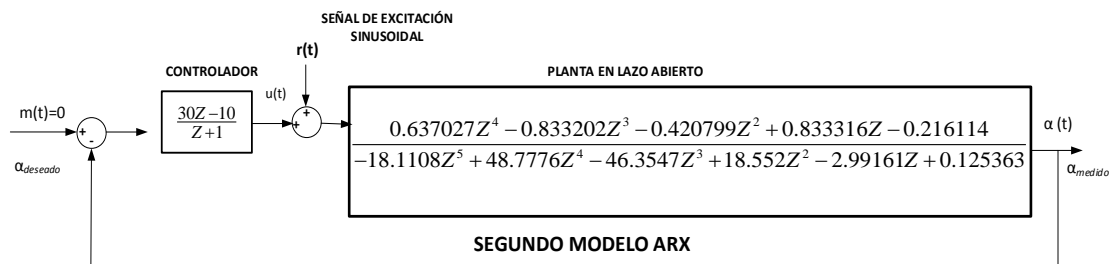
**Figura 5.50 Programación en Matlab para comparar rendimientos de algoritmos difuso, PID, PD, ángulo alfa**

#### 5.4 Análisis comparativo de modelos de identificación

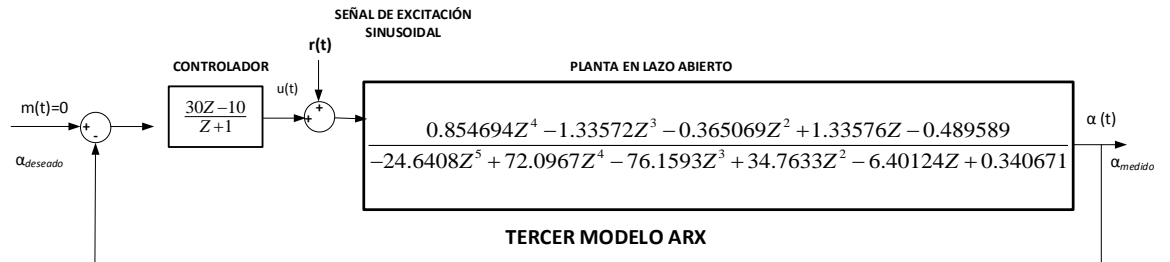
Para validar los modelos matemáticos obtenidos de la identificación de planta, se realizaron cinco ejecuciones del algoritmo de identificación de lazo cerrado, y se observó que el modelo ARX en todos los casos brindó una respuesta coherente, mientras que los otros modelos en algunas de las cinco ocasiones no ofrecieron una respuesta satisfactoria, es por esta razón se decidió validar los modelos ARX como se muestra a continuación:



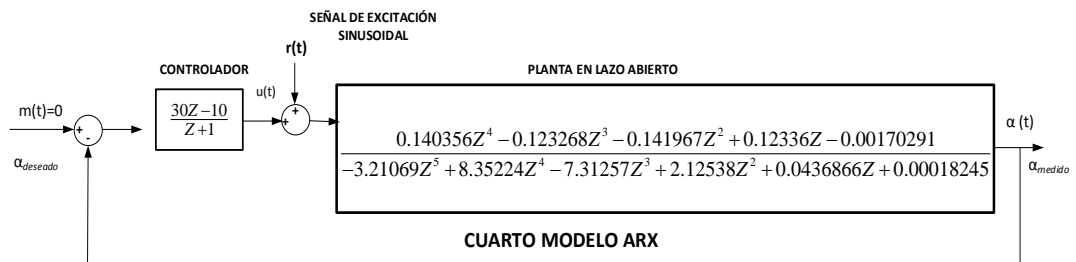
**Figura 5.51 Identificación de la planta, primer modelo ARX**



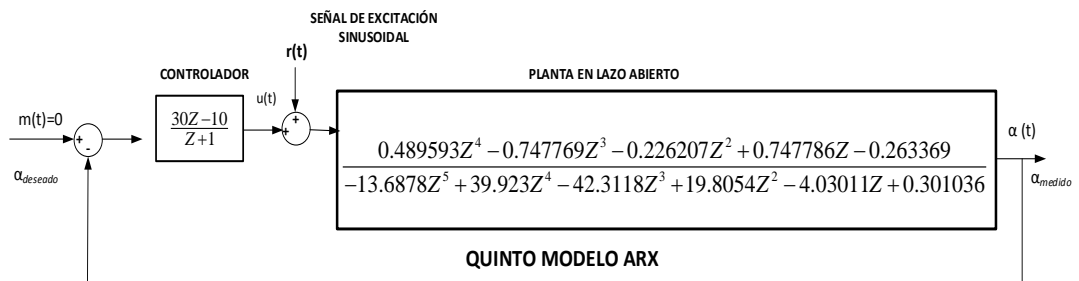
**Figura 5.52 Identificación de la planta, segundo modelo ARX**



**Figura 5.53 Identificación de la planta, tercer modelo ARX**

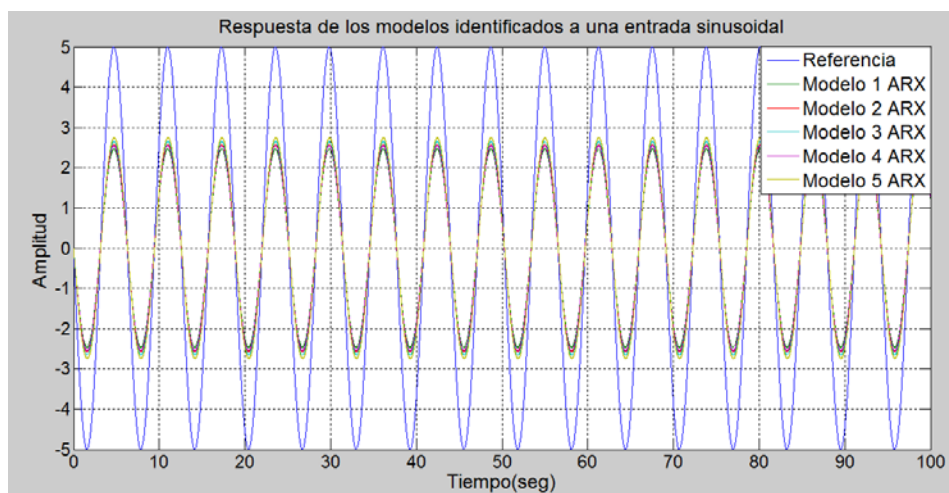


**Figura 5.54 Identificación de la planta, cuarto modelo ARX**



**Figura 5.55 Identificación de la planta, quinto modelo ARX**

Posterior se exportaron los datos de los modelos identificados a Matlab y se obtuvieron los siguientes resultados:



**Figura 5.56 Respuesta de los modelos identificados a una entrada sinusoidal**

Así también se realizó un algoritmo en Matlab que permite medir cuantitativamente el valor cuadrático medio del error (RMS), así como también su porcentaje. A continuación se muestran los datos obtenidos:

**Tabla 5.17 Errores relativos y RMS de los modelos identificados de la planta**

Modelo identificado	Error RMS	Error relativo
Modelo 1	1.7973	50.7244%
Modelo 2	1.7194	48.5270%
Modelo 3	1.6633	46.9439%
Modelo 4	1.7379	49.0494%

Modelo 5	1.5974	45.0832%
----------	--------	----------

La programación efectuada en Matlab se muestra a continuación:

```

%MODEL1
%CALCULO DEL ERROR RMS
error_RMS1=sqrt(mean((MODEL1(:,3)-MODEL1(:,2)).^2))

%CALCULO DEL ERROR RELATIVO RMS
Relative_error1=(error_RMS1/sqrt(mean(MODEL1(:,3).^2)))*100

%MODEL2
%CALCULO DEL ERROR RMS
error_RMS2=sqrt(mean((MODEL2(:,3)-MODEL2(:,2)).^2))

%CALCULO DEL ERROR RELATIVO RMS
Relative_error2=(error_RMS2/sqrt(mean(MODEL2(:,3).^2)))*100

%MODEL3
%CALCULO DEL ERROR RMS
error_RMS3=sqrt(mean((MODEL3(:,3)-MODEL3(:,2)).^2))

%CALCULO DEL ERROR RELATIVO RMS
Relative_error3=(error_RMS3/sqrt(mean(MODEL3(:,3).^2)))*100

%MODEL4
%CALCULO DEL ERROR RMS
error_RMS4=sqrt(mean((MODEL4(:,3)-MODEL4(:,2)).^2))

%CALCULO DEL ERROR RELATIVO RMS
Relative_error4=(error_RMS4/sqrt(mean(MODEL4(:,3).^2)))*100

%MODEL5
%CALCULO DEL ERROR RMS
error_RMS5=sqrt(mean((MODEL5(:,3)-MODEL5(:,2)).^2))

%CALCULO DEL ERROR RELATIVO RMS
Relative_error5=(error_RMS5/sqrt(mean(MODEL5(:,3).^2)))*100

```

**Figura 5.57 Programación en Matlab para el cálculo del error de aproximación de los modelos ARX identificados de la planta**

## CONCLUSIONES

1. El algoritmo de control difuso implementado para la plataforma electromecánica del sistema bola y plato, presenta menores errores de aproximación a la referencia fijada para los ángulos de inclinación alfa y beta que los algoritmos clásicos PID, y PD, esto confirma la eficiencia del control difuso ante sistemas no lineales de difícil modelamiento analítico, como lo es el sistema bola y plato
2. Los sobrepicos y tiempos de estabilización de las salidas de los algoritmos de control Difuso, PID, y PD son muy aproximados, siendo el esfuerzo de control del algoritmo difuso mucho más óptimo, lo que se traduce en una mejor utilización de la energía eléctrica y por tanto se prolongue el tiempo de vida útil del equipo.
3. La eficiencia de control del algoritmo difuso para el seguimiento de referencias es dependiente del dominio de las funciones de pertenencia fijado por el lazo de histéresis caracterizado para la señal de voltaje aplicado a los inductores que permiten levitar la plataforma.
4. Debido a la alta inestabilidad en lazo abierto que presenta el sistema electromecánico bola y plato, su identificación se realizó en lazo cerrado, en este caso la calidad del modelo identificado depende del

controlador externo utilizado, el cual debe evitar la saturación de la salida del sistema frente a la perturbación aplicada a la planta.

5. El modelo de identificación paramétrica ARX presenta en todas las pruebas realizadas un modelo satisfactorio para la plataforma electromecánica del sistema bola y plato 33-240.
6. El sistema de algoritmos implementado para la caracterización, identificación, y estabilización de los ángulos de inclinación de la plataforma electromecánico, permiten además generar diversas prácticas tipo educativo para las asignaturas de control automático, como las propuestas en los anexos uno, dos, y tres.



## RECOMENDACIONES

1. Para evitar daños irreversibles en el sistema bola y plato 33-240, se recomienda tener en consideración los valores presentados en la curva de histéresis voltaje-ángulo. Valores fuera de esta curva generan elevados voltajes aplicados a los inductores de levitación de la plataforma causando con ello excesivos golpes de la misma.
2. Por el exceso de calor generado en los inductores que emiten el campo electromagnético utilizado para levitar la plataforma se recomienda trabajar en ambientes climatizados.
3. Se recomienda analizar nuevos métodos de identificación matemática para el sistema bola y plato, ya que fue todo un reto realizar la identificación en lazo cerrado del mismo, debido a su alta inestabilidad que presenta.
4. Se recomienda generar trabajos futuros que incluyan en los algoritmos de control diseñados datos obtenidos de la posición de la bola a través de la cámara incorporada en el sistema.

## BIBLIOGRAFÍA

- [1] Feedback Instruments Ltd., «Manual:33-240IC-LabView Ed01 032011,» de Ball and plate Installation&Commissioning for labview, Park Road, Crowborough, East Sussex, TN6 2QR, UK, Feedback Instruments Ltd., 2013, p. 33.
- [2] K. Ogata, Ingeniería de control moderna, vol. 3, Naucalpan de Juarez: PEARSON, 1998.
- [3] V. Mazzone, «<http://iaci.unq.edu.ar/caut1>,» Universidad Nacional de Quilmes, Marzo 2002. [En línea]. Available: <http://iaci.unq.edu.ar/caut1>. [Último acceso: viernes abril 2015].
- [4] P. Ponce Cruz, Inteligencia artificial, Mexico: Alfaomega, 2010.
- [5] Misa Llorca, PH.D., «Control inteligente,» Maestría en automatización y control industrial MACI, Guayaquil, 2013.
- [6] K. Ogata, Sistemas de control en tiempo discreto, Naulcapan de Juarez: Prentice Hall Hispanoamericana S.A, 1996, p. 6.
- [7] National Instruments Corporation., «PID and Fuzzy Logic Toolkit User Manual,» National Instruments Corporation., Austin, Texas 78759-3504 USA, 2009.
- [8] C. R.-R. E. F.-M. D. Bravo-Montenegro, «Identificación en lazo cerrado de sistemas dinámicos,» *Revista Colombiana de física*, vol. 45, nº 1, pp. 45-52, 2013.
- [9] Feedback Instruments Ltd., «Manual:33-240-LabView Ed01032011,» de *Ball and Plate Control Experiments*, Park Road, Crowborough, East Sussex, TN6 2QR,UK, Feedback Instruments Ltd., 2013, p. 37.
- [10] H. S. Ramirez, Control de sistemas no lineales, linealización aproximada, linealización extendida, Mexico: Pearson Education, 2005, pp. 29-37.

# ANEXOS

## LABORATORIO DE CONTROL AUTOMÁTICO

### PRÁCTICA # 1

**TEMA:** *Caracterización del sistema bola y plato 33-240*

#### 1. Objetivo general

Analizar el funcionamiento del sistema bola y plato 33-240 y caracterizar las variables de ángulo y voltaje de su plataforma.

#### 2. Objetivos específicos

2.1. Elaborar la función de transferencia ángulo alfa versus voltaje para establecer los rangos de variaciones permitidas

2.2. Elaborar la función de transferencia ángulo beta versus voltaje para establecer los rangos de variaciones permitidas

2.3. Exportar los datos de las variables medidas a Matlab y realizar comparaciones entre las funciones de transferencias generadas

2.4. Realizar un gráfico que muestre como varia los ángulos alfa y beta en función del tiempo

### **3. Fundamentos Teóricos**

El sistema en forma general consiste de cuatro elementos: un computador con un algoritmo de control digital, los convertidores A/D y D/A, el proceso a controlar, y el sensor.

El prototipo mencionado trabaja con el sistema de programación gráfica LabView, y sus herramientas desarrolladas para aplicaciones en tiempo real.

Entre las principales partes del sistema B&P está la base formada por cuatro inductores en los cuales circula corriente eléctrica y esta a su vez genera un campo electromagnético en cada inductor, que permite la levitación de un plato. Este plato está montado sobre un sistema mecánico con potenciómetros instalados en ejes ortogonales, los cuales son encargados de medir los ángulos de inclinación de la tabla, y esta medida es enviada al computador a través de la tarjeta de adquisición de datos NI PCI/6251 (Peripheral Component Interconnect) [1] y [2].

El plato que forma el sistema descrito es de bajo peso, de material plástico, y de forma cuadrada, en sus extremos presenta cuatro anillos de aluminio, en donde por el efecto de corrientes de Eddy se induce el campo electromagnético para que este levite.

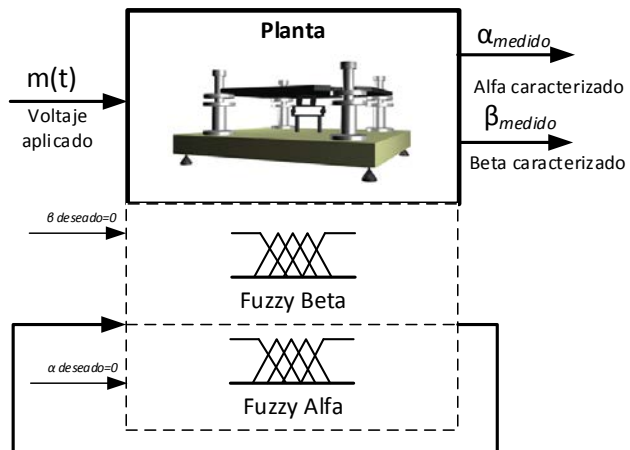


Figura 1. Diagrama de bloques del algoritmo de caracterización

#### 4. Materiales y equipos utilizados

Equipo/Material	cantidad
Módulo bola y plato 33-240	1
Tarjeta NI PCI 6251	1
NI cable- SHC-68-68-EPM	1
Programa LabView	1
Computador	1

#### 5. Procedimiento

5.1 Una vez encendido el computador, abra el instrumento virtual (VI) denominado "Caracterización de la planta"

5.2 Fije un rango de voltaje entre -2 y 2 voltios en el instrumento llamado "entrada de voltaje alfa", y fije su valor inicial en cero voltios.

5.3 Cargue las funciones de membresía en el programa para la caracterización de la planta, usando el archivo denominado “bola y plato 7”.

5.4 Enciende el sistema bola y plato 33-240.

5.5 Ejecute el algoritmo de caracterización de la planta

5.6 Mueva la referencia de voltaje aplicado a la planta, desde cero hasta valores positivos, y viceversa, esto permitirá observar cómo se va graficando la función de transferencia ángulo Vs voltaje.

5.7 Repita el paso 5.6, hasta obtener el mejor gráfico posible de la función de transferencia voltaje Vs ángulo, y exporte los datos de Labview hacia Matlab, realice las gráficas respectivas y establezca comparaciones de funcionamiento del sistema.

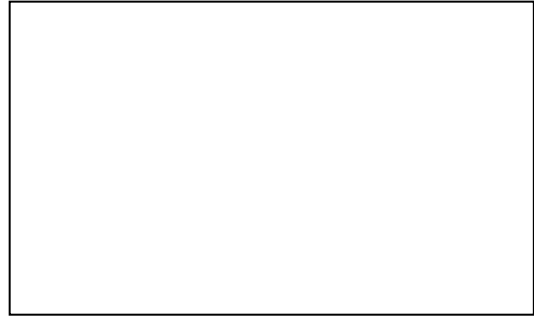
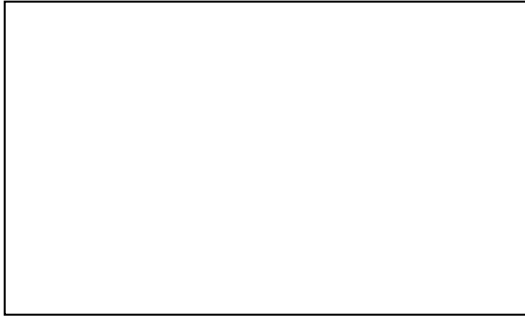
5.8 Repita los pasos del 5.1 hasta el 5.7 para el ángulo de inclinación beta.

## 6. Datos y cálculos

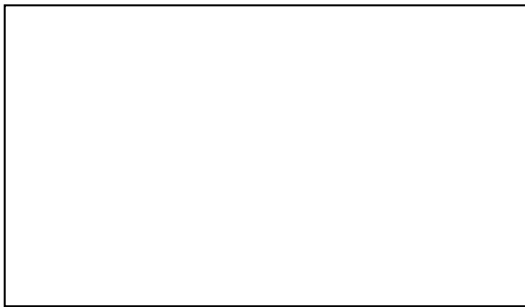
6.1 Llene la siguiente tabla con los datos de la experimentación realizada

Voltaje aplicado	Ángulo alfa medido	Ángulo beta medido

6.2 Grafique de forma manual la función de transferencia ángulo alfa Vs voltaje, y también ángulo beta Vs voltaje.



6.3 Grafique usando Matlab la función de transferencia ángulo alfa Vs voltaje, y también ángulo beta Vs voltaje.



## 7. Conclusiones y recomendaciones

---

---

---

---

# LABORATORIO DE CONTROL AUTOMÁTICO

## PRÁCTICA # 2

**TEMA:** *Diseño y experimentación con un algoritmo de control inteligente basado en lógica difusa para la plataforma bola y plato 33-240*

### 1. Objetivo general

Analizar el funcionamiento del sistema bola y plato 33-240 bajo el control de un algoritmo inteligente basado en lógica difusa, y comparar su rendimiento con un algoritmo convencional PID.

### 2. Objetivos específicos

2.1 Analizar el algoritmo de control difuso diseñado para el control de la plataforma del sistema B&P 33-240 en modo de operación manual y automático.

2.2 Establecer comparaciones de rendimiento en el control de la plataforma del sistema B&P 33-240 del algoritmo difuso versus un algoritmo PID.

2.3 Experimentar con diferentes familias de funciones de membresía generadas para el algoritmo de control inteligente basado en lógica difusa, y analizar su comportamiento.

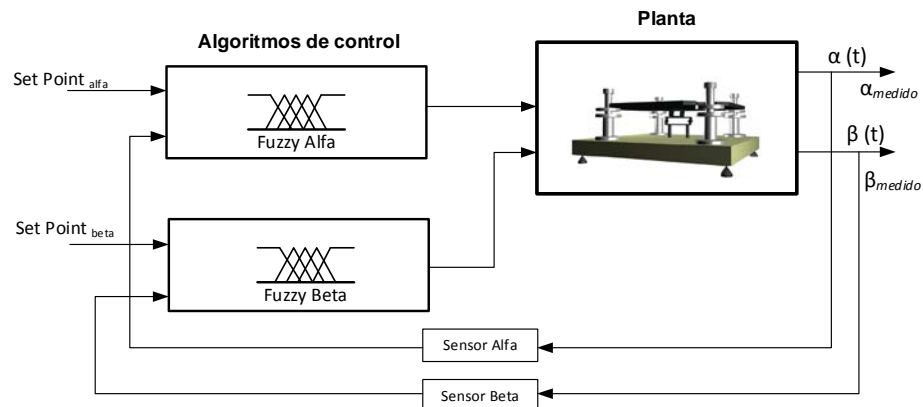


2.4 Exportar los datos de las variables medidas y contraladas a Matlab y realizar la graficación de las mismas

### **3. Fundamentos Teóricos**

Una vez que se ha caracterizado la planta y se conoce el comportamiento de los ángulos de inclinación alfa y beta en función del voltaje aplicado a los inductores, es posible diseñar el algoritmo difuso para el control de la misma de forma más precisa y segura.

El algoritmo de control diseñado funciona tanto para el ángulo de inclinación alfa como para el beta, y en modo manual y automático. En el Modo de operación manual el usuario establece la señal de referencia a manera de escalones y se puede observar como la planta responde a este estímulo hasta estabilizarse, de igual forma en modo automático, la diferencia es que la señal de referencia es un estímulo pulsante a manera de onda cuadrada. A continuación se muestra en entorno gráfico del mismo:



**Figura 2. Diagrama de bloques del algoritmo de control difuso para la planta**

#### 4. Materiales y equipos utilizados

Equipo/Material	cantidad
Módulo bola y plato 33-240	1
Tarjeta NI PCI 6251	1
NI cable- SHC-68-68-EPM	1
Programa LabView	1
Computador	1

#### 5. Procedimiento

5.1 Una vez encendido el computador, ejecute el instrumento virtual (VI) denominado "Controlador Fuzzy Completo Final alfa-Beta"

5.2 Establezca la ruta en donde se van a guardar los datos de las variables medidas

5.3 Cargue las funciones de membresía en el programa para el algoritmo de control difuso, usando el archivo denominado “bola y plato 7” para el ángulo alfa y “bola y plato 8” para el ángulo beta.

5.4 Seleccione el modo de operación del algoritmo, dispone de modo manual y automático.

5.5 Ejecute el algoritmo de identificación

5.6 Seleccione la variable a controlar, dispone de “fuzzy ángulo alfa” y “fuzzy ángulo beta”.

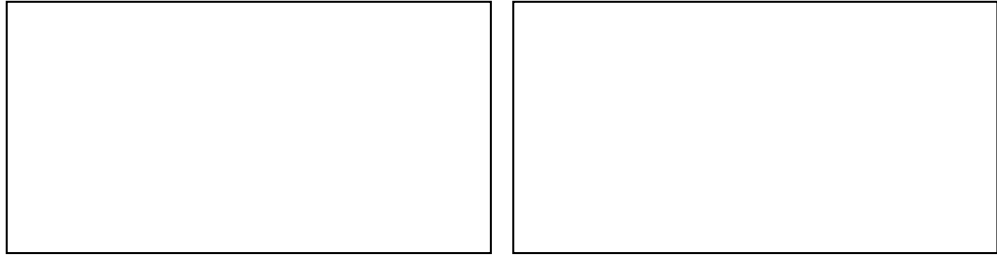
5.7 Registre los datos de los ángulos escritos y ángulos leídos, así como de la señal de voltaje.

## 6. Datos y cálculos

6.1 Llene la siguiente tabla con los datos obtenidos de la experimentación realizada.

Voltaje aplicado	Ángulo alfa escrito	Ángulo alfa leído	% de error	Ángulo beta escrito	Ángulo beta leído	% de error

6.2 Grafique la familia de funciones de membresía que mejor resultados ofrece en cuanto al seguimiento de referencias, y escriba las reglas de inferencia del controlador difuso.



6.3 Grafique usando Matlab las siguientes señales:

6.3.1 Seguimiento del ángulo alfa a la referencia en modo manual

6.3.2 Seguimiento del ángulo beta a la referencia en modo manual

6.3.3 Seguimiento del ángulo alfa a la referencia en modo automático

6.3.4 Seguimiento del ángulo beta a la referencia en modo automático

6.3.5 Esfuerzos de control para los ángulos alfa y beta

6.3.6 Comparaciones de seguimiento de referencia en modo automático para los ángulos alfa y beta de los algoritmos difuso y PID.

## 7. Conclusiones y recomendaciones

---

---

---

---

ANEXO 2. Guía de práctica, diseño de algoritmo difuso, practica 2

## LABORATORIO DE CONTROL AUTOMÁTICO

### PRÁCTICA # 3

**TEMA:** *Identificación matemática en lazo cerrado de la plataforma bola y plato 33-240*

#### 1. Objetivo general

Realizar la identificación matemática de la plataforma bola y plato 33-240, usando los modelos paramétricos ARX, ARMAX, y BJ

#### 2. Objetivos específicos

2.1 Obtener el modelo matemático identificado en lazo cerrado de la plataforma bola y plato 33-240

2.2 Obtener el modelo matemático identificado en lazo abierto de la plataforma bola y plato 33-240

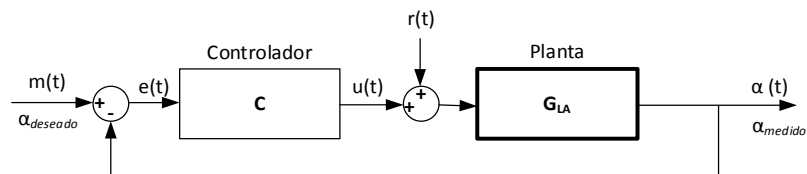
2.3 Experimentar con señales de excitación PBRS y sinusoidales en la identificación de la planta.

2.4 Comparar el rendimiento de los modelos identificados

2.5 Exportar los datos de los modelos identificados a Matlab y calcular los porcentajes de error en la aproximación ante una señal sinusoidal.

### 3. Fundamentos Teóricos

En el siguiente esquema se realizará el análisis matemático del proceso de identificación en lazo cerrado para un sistema inestable.



**Figura 3. Esquema de identificación en lazo cerrado**

Para identificar un sistema en lazo cerrado se debe establecer un controlador (C), el mismo que regule la respuesta de la planta, y no permite que mientras es excitada por una señal externa pierda estabilidad. Debido a que este controlador puede introducir una correlación entre el ruido de salida y la señal de control, el modelo

identificado podría no ser tan preciso, para eliminar esta correlación se introduce una señal de excitación adicional  $r(t)$ , la cual se suma a la señal de control  $u(t)$ . [2].

El análisis matemático, se desarrolla a manera de resumen a continuación:

$\alpha(t) = G_{LA} [u(t) + r(t)]$	$G_{LC} = \frac{\alpha(t)}{r(t)} = \frac{G_{LA}}{1 + G_{LA} \cdot C}$
$u(t) = C \cdot e(t) = -\alpha(t) \cdot C$	$G_{LC} (1 + G_{LA} \cdot C) = G_{LA}$
$\alpha(t) = G_{LA} [-\alpha(t) \cdot C + r(t)]$	$G_{LC} = G_{LA} (1 - G_{LC} \cdot C)$
$\alpha(t) \cdot (1 + G_{LA} \cdot C) = G_{LA} \cdot r(t)$	$\frac{\alpha(t)}{u(t)} = G_{LA} = \frac{G_{LC}}{1 - G_{LC} \cdot C}$

#### 4. Materiales y equipos utilizados

Equipo/Material	cantidad
Módulo bola y plato 33-240	1
Tarjeta NI PCI 6251	1
NI cable- SHC-68-68-EPM	1
Programa LabView	1
Computador	1

## 5. Procedimiento

5.1 Una vez encendido el computador, abra el instrumento virtual (VI) denominado "identificación plato"

5.2 Ajuste las constantes del controlador PD en los siguientes valores  $K_c=10$  y  $T_d= 0.001$

5.3 Habilite la escritura de datos en el instrumento virtual de identificación

5.4 Ejecute el VI de identificación del plato y analice cada uno de los modelos obtenidos

5.5 Registre los datos de los modelos identificados de la planta, y obtenga los porcentajes de error de cada uno de ellos.

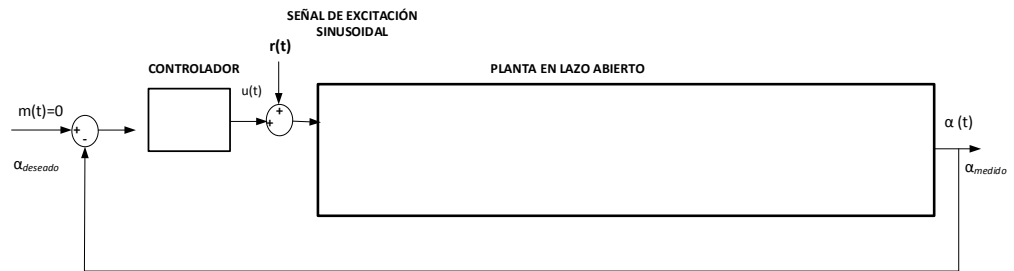
## 6. Datos y cálculos

6.1 Llene la siguiente tabla con los datos obtenidos de la experimentación realizada.

Modelo identificado	Modelo de la planta en lazo cerrado	Modelo de la planta en lazo abierto	Modelo del controlador	Porcentaje de error
ARX				
ARMAX				
BJ				



6.2 Escriba en el siguiente diagrama de bloques el mejor modelo identificado luego de realizar un mínimo de 5 ejecuciones del algoritmo de identificación del plato del sistema B&P.



6.3 Grafique usando Matlab los modelos identificados, y compare eficiencia.

## 7. Conclusiones y recomendaciones

---



---



---



---

ANEXO 3. Guía de práctica, identificación de la planta, práctica 3



## NI PCI-6251

### 16 Bits, 1 MS/s (multicanal), 1.25 MS/s (monocanal), 16 Entradas Analógicas

- National Instruments recomienda usar el dispositivo PCIe-6351 de la Serie X para todas las aplicaciones nuevas.
- 16 entradas analógicas, 1.25 MS/s monocanal, 1.25 MS/s multicanal, resolución de 16 bits,  $\pm 10$  V
- Certificado de calibración trazable expedido por el NIST y más de 70 opciones de acondicionamiento de señales
- DIO correlacionadas (8 líneas sincronizadas, 10 MHz), disparo analógico y digital
- La tecnología de calibración NI-MCal proporciona una mayor precisión
- Software controlador NI-DAQmx y software interactivo NI LabVIEW SignalExpress LE para registro de datos



## Información General

La NI PCI-6251 es una tarjeta DAQ multifunción de alta velocidad de la Serie M optimizada para una precisión superior a velocidades de muestreo más altas. Para mayor precisión en las medidas, considere usar los dispositivos de alta precisión de la Serie M con un convertidor analógico-digital que proporciona 4 veces más resolución.

Las tarjetas de alta velocidad de la Serie M ofrecen características avanzadas como el controlador de sistema NI-STC 2, el amplificador programable NI-PGIA 2, la tecnología de calibración NI-MCal para mejorar el rendimiento y la precisión. Tienen un amplificador NI-PGIA 2 integrado diseñado para tiempos de asentamiento rápido a altas velocidades de muestreo, garantizando precisión de 16 bits, incluso cuando mide todos los canales a máxima velocidad. Para aprender más sobre las tecnologías de la Serie M, especificaciones de dispositivos e información sobre cables y accesorios recomendados, por favor consulte la hoja de especificaciones.

#### Software Controlador

Los dispositivos de la Serie M funcionan con múltiples SOs usando tres opciones de software controlador: NI-DAQmx, NI-DAQmx Base y el Hardware de Medida DDK. Busque la información en la sección de Recursos para aprender más sobre el software controlador o para descargar un controlador. Los dispositivos de la Serie M no son compatibles con el controlador Tradicional NI-DAQ (Legado).

#### Software de Aplicación

Con NI LabVIEW usted puede crear aplicaciones DAQ personalizadas con la facilidad de la programación gráfica de más de 600 funciones de análisis y herramientas de programación avanzadas. Los sistemas de desarrollo Completo y Profesional de LabVIEW incluyen LabVIEW SignalExpress para registro de datos interactivo. Los dispositivos DAQ de la Serie M son compatibles con las siguientes versiones (o posteriores) de software de aplicación de NI - LabVIEW 7.x, LabWindows™/CVI 7.x, o Measurement Studio 7.x; LabVIEW SignalExpress 1.x; o LabVIEW con el Módulo LabVIEW Real-Time 7.1. Estos dispositivos también son compatibles con Visual Studio .NET, ANSI C/C++ y Visual Basic 6.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

## ANEXO 4. Manual de Tarjeta NI PCI-6251

Last Revised: 2014-11-06 07:14:50.0

## DAQ Accessories Selection Guide



Shielded and unshielded cables available in many lengths  
Custom cable connector kits and mounting accessories  
Direct BNC connectivity for most products



Shielded, unshielded, and DIN mountable terminal blocks  
Custom connectivity

### Overview

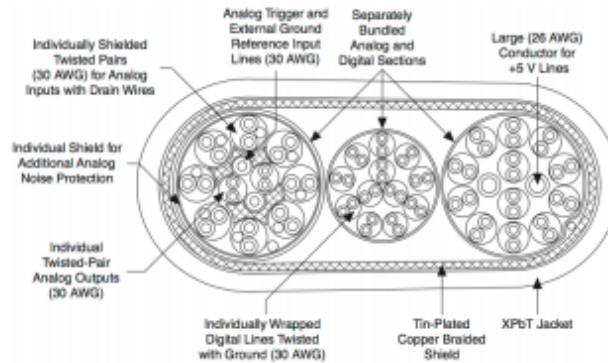
National Instruments accessories offer the flexibility you need in creating your measurement system. From shielded cables and terminal blocks to RTSI and ribbon cables, NI provides a variety of connectivity solutions including screw terminals and BNC connectors. NI shielded connector block (SCB) products feature rugged, very low-noise signal termination, and the NI terminal block (TB) and connector block (CB) families offer unshielded, low-cost alternatives.

[Back to Top](#)

### Application and Technology

#### NI Cable Design Advantages

National Instruments provides many different high-performance shielded cables that have been designed for specific NI multifunction DAQ devices. Each type of cable features improved signal integrity using the following technologies:



## ANEXO 5. NI CABLE-SHC-68-68-EPM