



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

TRABAJO DE TITULACIÓN:

“ANÁLISIS COMPARATIVO ENTRE CONTROLADORES BASADOS EN LÓGICA DIFUSA Y CONTROL PID CLÁSICO, APLICADOS A SISTEMAS DE CONTROL DE VELOCIDAD DE MOTORES DC SIN ESCOBILLAS.”

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

MAGISTER EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

PRESENTADO POR:

RICHARD MICHAEL SÁNCHEZ ROSADO

GUAYAQUIL – ECUADOR

2015 - 2016

AGRADECIMIENTO

En primer lugar a Dios por guiarme en la vida como un Padre que cuida de su hijo predilecto.

A mis padres José y Betty que han sido pilares importantes en mi vida. A mis hermanos que han sido testigos del esfuerzo y sacrificio realizado.

A mis profesores y amigos que gracias a su apoyo y consejos han hecho posible concretar este trabajo. De manera especial al MSc. Alberto Larco, por sus valiosos consejos y ayuda incondicional en este trabajo. No existe mejor palabra que decir "Gracias".

DEDICATORIA

A mis padres y maestros.

TRIBUNAL DE SUSTENTACIÓN

M.Sc. Sara Judith Ríos.

Presidente del tribunal.

M.Sc. Alberto Larco Gómez.

Director del Trabajo de titulación.

M.Sc. Holger Cevallos U.

Vocal principal

DECLARACIÓN EXPRESA

La responsabilidad del contenido de este trabajo de titulación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

(REGLAMENTO DE GRADUACIÓN DE LA ESPOL)

RICHARD MICHAEL SÁNCHEZ ROSADO

RESUMEN

El presente trabajo tiene como objetivo el desarrollo de un controlador basado en lógica difusa y un controlador clásico PID, para manipular la velocidad de un motor DC sin escobillas (BLDC) y determinar qué controlador presenta una mejor respuesta dinámica ante cambios repentinos de la referencia de velocidad o la carga aplicada al eje del motor.

Se desarrollará la teoría del control de lógica difusa y la de control clásico para controlar la velocidad de un motor DC sin escobillas. Las pruebas experimentales serán realizadas en el sistema de desarrollo para el control de motores (DMD2000) mediante procesadores digitales de señal (DSP), disponible en el Laboratorio de Electrónica de Potencia de la FIEC-ESPOL.

La simulación de los sistemas de control será realizada en MATLAB-Simulink, mientras que en la implementación usaremos el software VisSim/ECD, que es una herramienta de programación gráfica, que se enlaza con el software Code Composer Studio (CCS) para generar el código ensamblador final del DSP TMS320F2812 de Texas Instruments embebido en la tarjeta de desarrollo eZdspF2812 de Spectrum Digital.

ÍNDICE GENERAL

RESUMEN	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS.....	XIII
SIMBOLOGÍA	XXII
INTRODUCCIÓN	XXVII
CAPÍTULO 1	1
1. PLANTEAMIENTO DEL PROBLEMA.	1
1.1. Introducción.	1
1.2. Antecedentes.....	1
1.3. Descripción del problema.	3
1.4. Justificación.	4
1.5. Objetivo general.....	5
1.6. Objetivos específicos.	5
1.7. Propuesta de solución.	6
1.8. Metodología.	6
CAPÍTULO 2.....	8
2. ESTADO DEL ARTE.	8
2.1. Introducción.	8

2.2.	Estado del conocimiento.....	9
2.3.	Control Inteligente basado en lógica difusa.	11
2.3.1.	Introducción a lógica difusa.	12
2.3.2.	Conjuntos difusos.	14
2.3.3.	Operaciones sobre conjuntos difusos.....	16
2.3.4.	Controlador difuso.	18
2.3.5.	Sistema difuso Takagi - Sugeno.....	28
2.4.	Control Clásico PID.....	29
2.4.1.	Acción Proporcional.....	30
2.4.2.	Acción Proporcional-Integral.....	30
2.4.3.	Acción Proporcional-Derivativa.....	31
2.4.4.	Acción Proporcional-Integral-Derivativa.....	32
2.4.5.	Métodos de ajuste del controlador PID.....	35
2.5.	Motor DC sin escobillas BLDC.....	43
2.5.1.	Estructura física.....	44
2.5.2.	Principio de funcionamiento.....	49
2.5.3.	Modelo Matemático del motor.....	55
CAPÍTULO 3.....		59
3.	DISEÑO DE CONTROLADORES.	59

3.1.	Introducción.	59
3.2.	Descripción del sistema.	59
3.3.	Diseño del controlador difuso.....	61
3.3.1.	Entradas y salidas del controlador.....	63
3.3.2.	Definición de conjuntos difusos.	65
3.3.3.	Reglas difusas del sistema.	68
3.3.4.	Defusificación.	70
3.4.	Diseño del controlador PID.	72
3.4.1.	Método de Ziegler y Nichols en lazo abierto.	76
3.4.2.	Método de Ziegler y Nichols en lazo cerrado.....	78
CAPÍTULO 4.....		80
4.	IMPLEMENTACIÓN DE LA SOLUCIÓN.	80
4.1.	Introducción.	80
4.2.	Diagrama de bloques del sistema de desarrollo experimental DMD2000.	80
4.2.1.	Módulo de Potencia.....	82
4.2.2.	Tarjetas de procesamiento de señales.	83
4.2.3.	Tarjeta de desarrollo de Spectrum Digital (eZdspF2812). 85	
4.3.	Características del Motor DC sin escobillas a utilizar.	86

4.4.	Detalle del software de programación, VisSim.....	89
4.5.	Diseño de pruebas.....	90
4.5.1.	Control de velocidad en lazo abierto.....	91
4.5.2.	Control de velocidad en lazo cerrado con lógica difusa. 106	
4.5.3.	Control de velocidad en lazo cerrado con controlador PID. 123	
CAPÍTULO 5.....		129
5.	SIMULACIONES Y PRUEBAS EXPERIMENTALES.	129
5.1.	Introducción.	129
5.2.	Resultados de la Simulación.....	129
5.3.	Resultados de la Implementación.	137
5.4.	Análisis comparativo de los controladores.	141
CONCLUSIONES		145
BIBLIOGRAFÍA.....		147
6.	ANEXOS.....	152
6.1.	ANEXO A: Diagrama interno del controlador Difuso.....	153
6.2.	ANEXO B: Freno Magnético DMD-2000B	156

ÍNDICE DE TABLAS

Tabla 2.1 Operaciones sobre conjuntos difusos	18
Tabla 2.2 Reglas prácticas para la sintonía manual de controladores PID... 36	
Tabla 2.3 Parámetros del PID: K_p , T_i y T_d en lazo abierto con método Z-N. 40	
Tabla 2.4 Parámetros del PID: K_p , K_i y K_d en lazo abierto con método Z-N 41	
Tabla 2.5 Parámetros del PID: K_p , T_i y T_d en lazo cerrado con método Z-N.	42
Tabla 2.6 Parámetros del PID: K_p , K_i y K_d en lazo cerrado con método Z-N.	42
Tabla 2.7 Secuencia de conmutación.....	53
Tabla 3.1 Tabla de reglas para el controlador difuso.....	68
Tabla 3.2 Tabla del decodificador de los sensores de efecto Hall.	75
Tabla 3.3 Parámetros del PID: K_p , T_i y T_d en lazo abierto con método Z-N. 78	
Tabla 3.4 Valores del controlador PI por el método de Ziegler-Nichols	79
Tabla 4.1 Características del Motor BLDC usado en las pruebas.....	87
Tabla 4.2 Acción del PWM para cada sector	100

Tabla 4.3 Tabla de reglas para la implementación del controlador difuso. . 115

Tabla 5.1 Resultado de los controladores implementados..... 144

ÍNDICE DE FIGURAS

Figura 2-1 Grafica de lógica difusa y lógica clásica.	14
Figura 2-2 Elementos principales de los conjuntos difusos.....	15
Figura 2-3 Tipos de conjuntos difusos	16
Figura 2-4 Conjuntos difusos Intersección.....	16
Figura 2-5 Conjuntos difusos Unión.....	17
Figura 2-6 Conjuntos difusos negación.....	17
Figura 2-7 Controlador difuso	19
Figura 2-8 Controlador difuso, conocimiento del experto.....	20
Figura 2-9 Regla de control.....	21
Figura 2-10 Funciones de membresía.	22
Figura 2-11 Superficie de salida de dos controladores.	22
Figura 2-12 Función de membresía	23
Figura 2-13 Operador OR aplicado a dos entradas	25
Figura 2-14 Resultado de la implicación	25

Figura 2-15 Resultado de la agregación.	26
Figura 2-16 Resultado de la defusificación	28
Figura 2-17 Regla de control Takagi-Sugeno	29
Figura 2-18 Diagrama de bloques de un proceso con control de realimentación.....	29
Figura 2-19 Diagrama de bloques de un controlador PID paralelo.	33
Figura 2-20 Diagrama de bloques de un controlador PID Serie.....	34
Figura 2-21 Método de Ziegler-Nichols en lazo abierto	38
Figura 2-22 Curva de respuesta y obtención de tiempos para Z & N	39
Figura 2-23 Lazo cerrado de control	42
Figura 2-24 Sistema de control típico para un motor BLDC.....	44
Figura 2-25 Forma de onda de las corrientes trifásicas del motor BLDC.....	44
Figura 2-26 BLDC con rotor Interno (izquierda) o rotor externo (derecha)....	45
Figura 2-27 Estructura física de un motor BLDC	45
Figura 2-28 Bobinas del estator.	46
Figura 2-29 Rotor de imanes permanentes.....	47

Figura 2-30 Sensores de efecto Hall.....	49
Figura 2-31 Fuerza contra-electromotriz (E_a , E_b , E_c) y corriente de fase (I_a , I_b , I_c) respecto a la posición del rotor (A, B, C).....	51
Figura 2-32 Voltajes de Fase y Señal de sensores de efecto Hall.....	52
Figura 2-33 Inversor trifásico.	52
Figura 2-34 Conmutación trapezoidal (seis-pasos) con sensores de efecto Hall. (a) Modo II (Q1,Q4 ON), (b) Modo III (Q1,Q6 ON).....	54
Figura 3-1 Partes del equipo educativo DMD2000.	60
Figura 3-2 Diagrama de bloques de un FLC para el control de un motor BLDC.	63
Figura 3-3 Ventana FIS Editor: nueva variable de entrada.	64
Figura 3-4 Ventana FIS Editor: Asignación de nombres a las entradas.....	65
Figura 3-5 Ventana FIS Editor: Funciones de membresía.	66
Figura 3-6 Ventana FIS Editor: Rango de la variable de entrada error.	67
Figura 3-7 Ventana FIS Editor: Rango de la variable de salida dtorque.	67
Figura 3-8 Ventana para editar las reglas del controlador difuso.....	69
Figura 3-9 Superficie creada por las reglas del controlador.....	70

Figura 3-10 Ventana donde se muestra el valor de salida del controlador difuso	71
Figura 3-11 Controlador PI de velocidad de un Motor BLDC	72
Figura 3-12 Sistema de control de un motor BLDC.	73
Figura 3-13 Controlador PI de velocidad.....	74
Figura 3-14 Controlador de Corriente.	75
Figura 3-15 Bloque Decoder.....	76
Figura 3-16 Curva de respuesta y obtención de tiempos para Z & N	77
Figura 3-17 Respuesta oscilatoria en lazo cerrado.....	79
Figura 4-1 Diagrama de bloques de la plataforma experimental DMD2000..	81
Figura 4-2 Módulo de potencia de la plataforma experimental DMD2000	83
Figura 4-3 Tarjeta de interface.....	84
Figura 4-4 Tarjeta de procesamiento central.	85
Figura 4-5 Tarjeta de desarrollo eZdspF2812.....	86
Figura 4-6 Datos de placa del motor BLDC.	88
Figura 4-7 Sistema de pruebas DMD2000.....	88

Figura 4-8 Entorno gráfico de VisSim.	90
Figura 4-9 Sistema de control de velocidad en lazo abierto de un motor BLDC.	92
Figura 4-10 Diagrama de bloques “2BLDCOpenU.vsm”	93
Figura 4-11 Diagramas internos del Bloque compuesto “BLDC Inverter”. ...	94
Figura 4-12 Bloque compuesto “Position & Speed Sensor”	94
Figura 4-13 Bloque compuesto $d\theta/dt$	96
Figura 4-14 Proceso de cálculo de velocidad.	97
Figura 4-15 Bloque compuesto “Current Sensor”	98
Figura 4-16 Bloque compuesto “Current Sector”	98
Figura 4-17 Bloque compuesto “BLDC Driver”.....	99
Figura 4-18 Bloque compuesto “PWM Action Control”	99
Figura 4-19 Relación entre las variables Sector y Modo.....	101
Figura 4-20 Selección del Bloque “BLDC Inverter”	101
Figura 4-21 Cuadro de dialogo Code Generation Properties.	102

Figura 4-22 Mensajes de compilación y enlace en la ventana de comandos de Windows.	103
Figura 4-23 Ventana F281X DSP Interface Properties.	104
Figura 4-24 Reemplazo del bloque compuesto con el bloque DSPinterface.	105
Figura 4-25 Ventana Simulation Properties.	105
Figura 4-26 Referencia y velocidad actual en lazo abierto.....	106
Figura 4-27 Sistema de control difuso de velocidad para un motor BLDC..	107
Figura 4-28 Diagrama de bloques del controlador Fuzzy para un motor BLDC.	108
Figura 4-29 Diagrama de bloques del controlador Fuzzy para un motor BLDC.	109
Figura 4-30 Diagrama interno del bloque “Controlador”.....	109
Figura 4-31 Diagrama interno del bloque “Controlador FUZZY”.	110
Figura 4-32 Funciones de membresía del controlador difuso.	111
Figura 4-33 Fuzificación de la entrada 1, Función de membresía Grande Negativo (GN).	112

Figura 4-34 Fuzificación de la entrada 1, Función de membresía Negativo (N).....	113
Figura 4-35 Modelo del algoritmo de inferencia.	114
Figura 4-36 Diagrama interno del bloque “Reglas Fuzzy MIN”.	116
Figura 4-37 Diagrama interno del bloque “MIN 1 - 5”.....	117
Figura 4-38 Diagrama interno del bloque “Reglas Fuzzy MAX”.....	117
Figura 4-39 Funciones de Membresía de la salida.	118
Figura 4-40 Diagrama interno del bloque “Desfuzificación”.	119
Figura 4-41 Diagrama interno del bloque “Integrador”.....	119
Figura 4-42 Parámetros del bloque “TransferFunction (filter)”.	120
Figura 4-43 Diagrama de bloques del archivo “5BLDCSpeedFUZZYD.vsm”	121
Figura 4-44 Parámetros del bloque Select Source.	122
Figura 4-45 Respuesta del Controlador PID ante una entrada de velocidad.	122
Figura 4-46 Sistema de control PI de velocidad para un motor BLDC.	123

Figura 4-47 Diagrama de bloques del controlador PID para un motor BLDC.	124
Figura 4-48 Bloque compuesto "Control de velocidad PID"	125
Figura 4-49 Parámetros del bloque "PID Regulator"	126
Figura 4-50 Diagrama de bloques del archivo "4BLDCSpeedPIDD.vsm" ...	127
Figura 4-51 Parámetros del bloque Select Source.	128
Figura 4-52 Respuesta del Controlador PID ante una entrada de velocidad.	128
Figura 5-1 Diagrama de bloques del controlador PI en la simulación.	130
Figura 5-2 Señales de corriente, velocidad y torque del controlador PI, con variación de velocidad y torque de carga constante.	131
Figura 5-3 Respuesta de velocidad, Controlador PI, ante cambio en la referencia 250 a 500RPM.	132
Figura 5-4 Respuesta del sistema, Controlador PI, ante un cambio repentino de torque de carga.....	133
Figura 5-5 Diagrama de bloques del controlador Difuso en la simulación. .	134
Figura 5-6 Controlador Difuso de velocidad.....	134

Figura 5-7 Señales de corriente, velocidad y torque del controlador Difuso, con variación de velocidad y torque de carga constante.	135
Figura 5-8 Respuesta de velocidad, Controlador Difuso, ante cambio en la referencia de 250 a 500RPM.	136
Figura 5-9 Respuesta del sistema, Controlador Difuso, ante un cambio repentino de torque de carga.	137
Figura 5-10 Respuesta del sistema, Controlador PI, implementación. Cambio en referencia de velocidad.	138
Figura 5-11 Respuesta del sistema, Controlador PI, implementación. Perturbación de la carga.	139
Figura 5-12 Respuesta del sistema, Controlador Difuso, implementación. Cambio en referencia de velocidad.	140
Figura 5-13 Respuesta del sistema, Controlador Difuso, implementación. Perturbación de la carga.	141
Figura 5-14 Comparación de resultados de la simulación.	142
Figura 5-15 Comparación de resultados de la implementación; Arriba: PID; Abajo: Fuzzy	143

SIMBOLOGÍA

A	Amperio
AC	Corriente alterna
ADC	Convertidor analógico digital.
BLDCM	Motor DC sin escobillas, de sus siglas en inglés Brushless DC Motor.
CCS	Code Composer Studio.
Cnt	Salida de conteo en bloque Quadrature Encoder
DAC	Convertidor Digital Analógico.
DC	Corriente Directa.
DCM	Motor DC
dec	Decremento.
DSP	Procesador Digital de Señales.
EVM	Administrador de eventos.
Fc	Frecuencia de la portadora.
FEM	Fuerza electro motriz.

FOC	Control orientado de campo
Freq	Frecuencia.
FPSS	Control Lógico Fuzzy para un estabilizador de sistema de potencia, Fuzzy logic based power system stabilizer.
HP	Caballo de fuerza.
Hz	Unidad de medida de la frecuencia.
IA	Inteligencia Artificial.
I/O	Entrada-salida.
IGBT's	Transistor bipolar de compuerta aislada de sus siglas en inglés Insulated Gate Bipolar Transistor.
IM	Motor de inducción
inc	Incremento.
JTAG	Acrónimo para Joint Test Action Group.
K	Factor de multiplicación por mil.
Kd	Constante derivativa.
Ki	Constante integral.

KhZ	Mil Hertz.
Kp	Constante proporcional.
KW	Mil Vatios.
K Ω	Mil ohmios.
K/cm ²	Kilogramos por centímetro cuadrado.
LC	Filtro inductivo capacitivo.
LED	Diodo emisor de luz.
LED OH	Diodo emisor de luz que señala sobre-temperatura.
mA	Miliamperios.
Mag	Magnitud.
MHz	Mega Hertz.
MIT	Massachusetts Institute of Technology.
MIPS	Millones de instrucciones por segundo.
mVp-p	Mili-Voltios pico a pico.
NAND	Compuerta lógica que realiza la operación de producto lógico

	negado.
N-cm	Newton por centímetro.
ohm	Ohmio.
OPAMP's	Amplificadores operacionales.
P	Proporcional
PD	Proporcional-Derivativo
PI	Proporcional-Integral.
PID	Proporcional, Integral y Derivativo
PIM	Modulo de potencia integrado.
PLC	Controlador Lógico Programable.
PMDCM	Motor DC de imanes permanentes.
PMSM	Motor sincrónico de imanes permanentes.
ppr	Pulsos por revolución.
PWM	Modulación de ancho de pulso
RMS	Valor eficaz o valor cuadrático medio.
rpm	Revoluciones por minuto.

RTW	Plataforma de trabajo en tiempo real.
Sec	Segundos.
THD	Distorsión armónica total.
V	Voltios.
Vrms	Voltaje eficaz.
VSI	Inversor con fuente de voltaje.
W	Vatios.
Wm	Velocidad angular del motor.

INTRODUCCIÓN

En los últimos años, el uso de la lógica difusa, en sistemas de control ha ido ganando gran popularidad, especialmente en Japón. Ya desde mediados de la década de 1970, científicos japoneses han sido fundamentales en la transformación de la teoría de la lógica difusa en una realización tecnológica.

[1]

El motor de corriente continua sin escobillas (BLDCM) está recibiendo una gran atención para aplicaciones industriales debido a su alta densidad de par, alta eficiencia y pequeño tamaño [2]. Las ventajas del motor BLDC han contribuido a su amplia difusión y uso en sistemas de control de velocidad.

En el presente trabajo se desarrolla la teoría de lógica difusa en controladores difusos, aplicada a accionamientos de velocidad variable para motores DC sin escobillas, y se realiza una comparación entre esta lógica de control inteligente y los métodos de control clásico mediante controladores PID.

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA.

1.1. Introducción.

En el presente capítulo se exponen algunos antecedentes respecto al trabajo de titulación, así como el problema de la investigación junto con su hipótesis; se desarrollan aspectos tales como: Objetivo general y objetivos específicos, la propuesta de solución y la metodología a utilizar.

1.2. Antecedentes.

A mediados de la década del 60 el Profesor Lotfi Zadeh, de la Universidad de California en Berkeley, introdujo la teoría de los

conjuntos difusos, donde la idea de pertenencia de un elemento a un determinado conjunto no era ya pertenece o no pertenece, sino que pertenece en un cierto grado. Las funciones que vinculan al elemento con su grado de pertenencia, se llaman funciones de membresía.

Basándose en esta idea, se construyó nuevamente toda la teoría de conjuntos, redefiniendo: inclusión, unión, operadores lógicos (AND, OR), otras operaciones lógicas y sus propiedades. [3]

Un paso importante para el desarrollo de la lógica difusa fue que a principios de la década de los setenta se crearon varios grupos de investigación en diferentes universidades japonesas que hicieron grandes contribuciones sobre las aplicaciones que podía tener este tipo de lógica. De esta forma se consiguió crear el primer controlador difuso para una máquina de vapor y crear un controlador de inyección de químico en depuradoras de agua. [4]

Una de las aplicaciones más famosas de la lógica difusa es la del sistema de Metro de Sendai, Japón. Este control de la línea Nanboku, desarrollada por Hitachi, utiliza un controlador difuso para dirigir el tren durante todo el día. Esto ha hecho que el movimiento del metro sea uno de los más confortables en el mundo, añadiendo un aumento en la eficiencia, así como mejora en los tiempos de parada. Este metro se puso en funcionamiento en 1988. [5]

Desde la década de los 80, se desarrollaron nuevos conceptos en el diseño de motores de imanes permanentes sin escobillas (BLDC). Los BLDC son categorizados como motores AC de tipo sincrónicos con imanes permanentes en su rotor, con una fuerza-contraelectromotriz (FEM) de forma trapezoidal y la forma de su corriente es casi rectangular [3]. Los motores BLDC están siendo rápidamente populares en industrias como tracción eléctrica, automóviles, aviones, equipamiento militar, medicina, equipos de automatización industrial e instrumentación debido a su alta eficiencia, alto factor de potencia, operación silenciosa, estructura compacta, confiabilidad y bajo mantenimiento. [6]

1.3. Descripción del problema.

En la actualidad cerca del 80% de los controladores son PI (proporcional-integral) debido a que ellos son fáciles de entender e implementar [7]. Muchos controladores de velocidad están implementados con controladores PI. Por otro lado está la lógica difusa que puede ser considerada como una teoría matemática que combina lógica de múltiples valores, teoría de probabilidad e inteligencia artificial para simular el comportamiento humano en la solución de varios problemas. Ha sido reportado [8] que los controladores difusos son más robustos, a

cambios en los parámetros de la planta, que los clásicos PI y que tienen una mejor capacidad de rechazo al ruido.

Debido a que nuestro sistema a controlar es un motor DC sin escobillas (BLDC), cuyo modelo es no lineal y las variaciones en la consigna de velocidad o en la carga acoplada a su eje, pueden generar perturbaciones en la lógica de control. Nace la inquietud si será factible realizar un controlador difuso a un motor BLDC y que presente una mejor respuesta dinámica que un controlador PID convencional.

1.4. Justificación.

El estudio y aplicación de controladores con lógica difusa está en aumento y es cada vez más apremiante implementar dichos controles para poder entenderlos mejor y familiarizarnos con ellos. El control inteligente está abarcando cada vez con más fuerza el manejo de sistemas, emulando el actuar del ser humano.

Gracias a que en el laboratorio de Electrónica de Potencia de la FIEC se encuentra un equipo que puede manejar Motores DC sin escobillas se implementará un controlador PID y otro con Lógica difusa para comparar sus resultados.

1.5. Objetivo general.

Realizar un análisis comparativo de la respuesta obtenida entre un controlador inteligente con lógica difusa y un controlador clásico PID, en el control de velocidad de un Motor de imanes permanentes sin escobillas (BLDC).

1.6. Objetivos específicos.

1. Estudiar la teoría de control inteligente, específicamente el control difuso, junto con el control clásico representado por el controlador PID.
2. Desarrollar un controlador difuso el cual parta del conocimiento del sistema, un motor DC sin escobillas, identificando sus entradas-salidas y creando las reglas de inferencia para un control óptimo.
3. Desarrollar un controlador PID para el control de velocidad de un motor DC sin escobillas.
4. Estudiar la estructura interna del sistema de desarrollo para el control de motores (DMD2000) mediante procesadores digitales de señal (DSP), disponible en el Laboratorio de Electrónica de Potencia de la FIEC-ESPOL.

5. Generar los programas necesarios en VisSim para embeberlos en el procesador de señales digitales (DSP) del equipo experimental y poder controlar la velocidad de motores DC sin escobillas.
6. Validar los algoritmos desarrollados en el equipo DMD-2000, y comparar el control inteligente de lógica difusa y el control clásico PID.

1.7. Propuesta de solución.

El presente trabajo desarrollará una comparación entre los resultados obtenidos de un controlador inteligente con lógica difusa y un controlador clásico PID, en el control de velocidad de un motor DC sin escobillas. Estos controladores serán desarrollados en un simulador y posteriormente implementados en un equipo experimental para su análisis.

1.8. Metodología.

Para el desarrollo de este trabajo de titulación se hará uso del método experimental, el cual constituye el método más importante y complejo de la investigación empírica, pues por lo general incluye la observación y medición, que se utilizan como procedimientos auxiliares, incorporándole nuevos elementos. [9]

El método experimental será:

Indirecto, ya que se realizarán simulaciones sobre el modelo del motor BLDC.

Directo, debido a que se experimentará con la planta real a través de un equipo didáctico.

El Experimento es un método de investigación de enorme valor para confirmar o refutar hipótesis y teorías así como para formular explicaciones y leyes. Esto lo convierte en el método por excelencia en nuestro campo de investigación.

CAPÍTULO 2

2. ESTADO DEL ARTE.

2.1. Introducción.

En este capítulo se detalla el Estado del Arte y se presentan investigaciones que se han dado en los últimos años referentes al tema, se muestran algunas herramientas utilizadas para la solución del problema; además se describe la teoría del control difuso y la teoría del control clásico, así como el principio de operación del motor sincrónico de imanes permanentes con Fuerza contra-electromotriz (FEM) trapezoidal, comúnmente denominado Motor DC sin escobillas (BLDC), usado extensamente en aplicaciones industriales de sistemas de control de posición (servomotores).

2.2. Estado del conocimiento.

En los últimos años ha aumentado el interés en el estudio de la lógica difusa, tanto es así, que esta técnica de control ha sido aplicada en diversas áreas del conocimiento, que en ocasiones no están relacionadas con el control automático de procesos. Se pueden citar los siguientes ejemplos:

K. Tomsovic, M.Y. Chow (2000), describieron un extenso tutorial de aplicaciones de lógica difusa en sistemas de potencia, donde uno de los casos de estudio fue el control por lógica difusa de un estabilizador de sistema de potencia (FPSS).

González A. y López J. (2010), implementaron un sistema informático interactivo para el control de nivel de líquido en tanques, mediante el uso de sensores y lógica difusa, con el fin de monitorear, controlar y calibrar el nivel de tanques en el campo industrial, añadiéndole al sistema una interfaz de visualización en 3D.

D.A. Tibaduiza, I. Amaya (2011), presentaron en la revista de investigación chilena SciELO, la Implementación de un control difuso para el control cinemático directo en un robot manipulador. Donde usan la lógica difusa como herramienta de control de posición para cada una

de las articulaciones de un robot tipo PUMA. Adicionalmente mostraron el desarrollo y la simulación del controlador usando el toolbox Fuzzy de MATLAB, así como la descripción de una implementación realizada en un Controlador Lógico Programable (PLC).

Dávila Montenegro (2013), diseñó y construyó un controlador difuso de lazo cerrado de temperatura mediante el microcontrolador 18F452 para los laboratorios de teoría de control de la Universidad Politécnica Salesiana de Quito. Con una interfaz de control en LabView de National Instruments.

Así también los estudios en el área de los motores de imanes permanentes en especial los BLDC se evidencian en las siguientes investigaciones o trabajos.

Shiyoung Lee, Ph.D. (2009) desarrolla el tema, A comparison study of the commutation methods for the three phase permanent magnet brushless DC motor, donde compara tres estrategias de conmutación: la trapezoidal (seis-pasos), la sinusoidal y la de control orientado de campo FOC. Donde los resultados obtenidos indican que la técnica de conmutación FOC tiene una mejor respuesta.

Shane W. Colton (2010), en su tesis de maestría en el MIT, presentó el tema Design and prototyping methods for brushless motor and motor control, donde expone los principios de operación del BLDC, su

estructura interna en detalle, la característica torque-velocidad, hasta los métodos de control modernos como el control orientado de campo FOC.

Patrick Fisher (2014), de la universidad de CQUniversity de Australia, con su trabajo High Performance Brushless DC Motor Control, en el cual usa las técnicas avanzadas para control de motores: control de campo orientado (FOC) y Control directo de torque (DTC), para controlar los motores BLDC y comparar su desempeño.

2.3. Control Inteligente basado en lógica difusa.

Debido a que en muchos casos los sistemas son complejos, no lineales de orden elevado, con parámetros internos que varían en el tiempo, dependientes del entorno, no definidos completamente en forma matemática, etc. No es posible alcanzar el control de estos sistemas con la teoría de control clásico, es así como nace la intención de aplicar al control las técnicas de Inteligencia Artificial (IA) [10] a este tipo de control se lo denomina Control Inteligente, dentro de los cuales tenemos: Lógica difusa, Redes neuronales, Algoritmos genéticos, sistemas expertos.

La *Lógica Difusa* se origina con el deseo de emular la forma del razonamiento humano, que es aproximado, no cuantitativo y lingüístico. Esto se debe a que el mundo en que vivimos no es binario. Existen

muchos estados entre joven y viejo, bueno y malo, bajo y alto, negro y blanco, etc. [11]

Las *Redes Neuronales artificiales* debido a su constitución y a sus fundamentos presentan un gran número de características similares a las del cerebro. Por ejemplo son capaces de aprender de la experiencia, de generalizar de casos anteriores a casos nuevos, de abstraer características esenciales de entradas que presentan información relevante, etc.

Un *algoritmo genético* es un método de optimización que manipula una cadena de números de forma similar a como los cromosomas son cambiados en la evolución biológica.

Un *sistema experto* es un programa de computación que utiliza una base de conocimiento de la experiencia humana para resolver problemas, o para esclarecer incertidumbre donde normalmente, uno o más expertos humanos deberían ser consultados. El sistema experto comprende dos componentes esenciales: Una base de conocimiento y un motor de inferencia. [11]

2.3.1. Introducción a lógica difusa.

Para ilustrar el concepto de la lógica difusa y los conjuntos difusos podemos tomar como ejemplo del conjunto de “los hombres altos”.

Según la lógica clásica, al conjunto de hombres altos solo pertenecen los que miden más de una determinada altura y esa altura límite es 1.80 metros, así un hombre es considerado alto cuando mide por ejemplo 1.81 metros y uno bajo cuando mide 1.79 metros. Esto no parece una razón muy lógica para catalogar a un hombre de alto o bajo ya que por ejemplo en el caso expuesto la altura de uno a otro solo se diferencia en 2 centímetros. Ahí, en casos como este donde no es fácil catalogar algo, se introduce la lógica difusa. Según esta lógica, el conjunto de "hombres altos" es un conjunto que no tiene una frontera clara que indique que perteneces a ese grupo o no. El evaluar si un hombre es alto o bajo, se hace mediante una función que define la transición entre alto a bajo y para ello asigna a las distintas alturas un valor entre 0 y 1. Según sea este valor se considera que se pertenece al conjunto o no. Aplicando esto al caso anterior, un hombre que mida 1.79 metros se puede decir que pertenece al conjunto de hombres altos con un grado de 0.75 y el hombre que medía 1.81 metros pertenece al conjunto de hombres altos con un grado de 0.8. Si representamos esto en una gráfica se obtendrá que la transición entre alto o bajo con la lógica difusa es una curva con cambios no abruptos mientras que con la lógica clásica, el paso de alto a bajo o viceversa es brusco. [4]

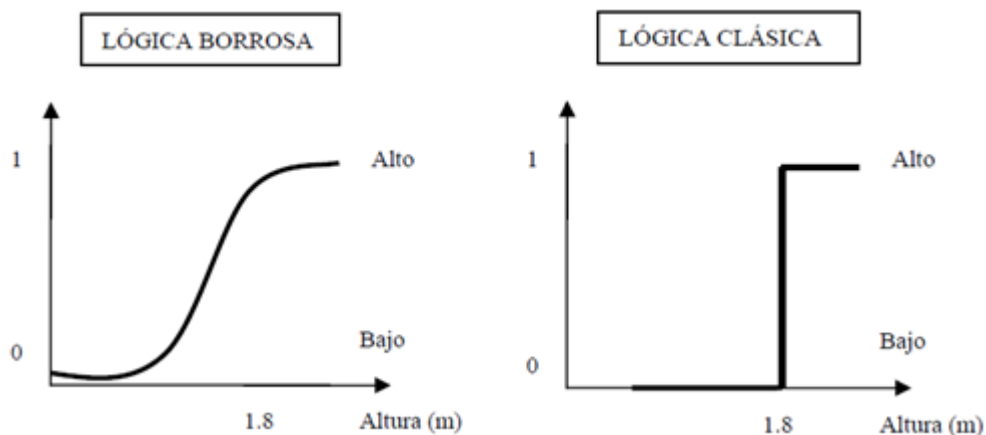


Figura 2-1 Grafica de lógica difusa y lógica clásica.

2.3.2. Conjuntos difusos.

Un conjunto difuso a diferencia de un conjunto clásico es un tipo especial de conjunto que permite la pertenencia parcial de sus elementos. En un conjunto clásico, sus elementos pertenecen o no pertenecen, mientras que en un conjunto difuso un elemento de su universo puede pertenecer a uno o más conjuntos con distintos grados de pertenencia.

El **universo de discurso** determina el rango de todos los posibles valores para una entrada a un sistema difuso.

Las **etiquetas** son las diferentes clasificaciones que se efectúan sobre la variable lingüística. Cada etiqueta tendrá un conjunto difuso asociado.

La **función de pertenencia o membresía $U(x)$** es una relación que asocia cada elemento en un conjunto con su grado de pertenencia (un número entre 0 y 1). Que puede ser expresada como un grupo de valores discretos o como una función continua.

Soporte, proporciona el rango de definición de la función de pertenencia. Es una forma de limitar el universo de discurso para cada etiqueta.

Corte - α . Es el subconjunto que pertenece al conjunto difuso con un grado de pertenencia mayor o igual que α . El valor α es llamado nivel- α .

Punto de cruce. Es el elemento del universo U , tal que su valor de membresía es 0.5.

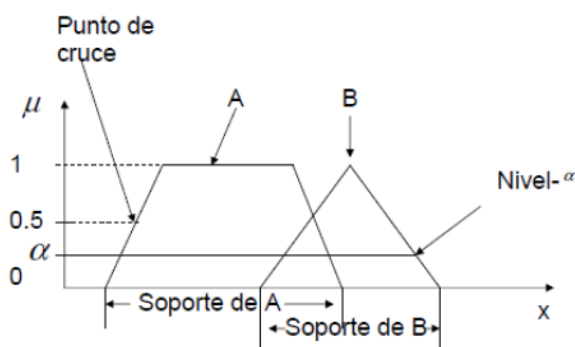


Figura 2-2 Elementos principales de los conjuntos difusos.

2.3.3. Operaciones sobre conjuntos difusos.

Existen distintos tipos de conjuntos difusos entre los más usados tenemos: Triangular, Trapezoidal, Gaussiana, Sigmoidal, Singleton.

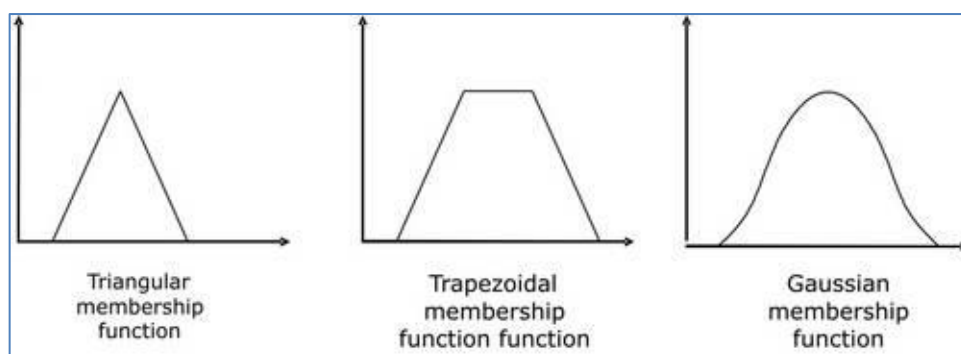


Figura 2-3 Tipos de conjuntos difusos

Así como en la lógica clásica, en la lógica difusa podemos definir operaciones entre conjuntos difusos.

La **intersección** (AND) de dos conjuntos difusos A y B, es un conjunto difuso C que se obtiene aplicando la operación T - norma.

$$C = A \cap B ; \quad (2.1)$$

$$\mu_C(x) = T(\mu_A(x), \mu_B(x)), \quad \forall x \in U \quad (2.2)$$

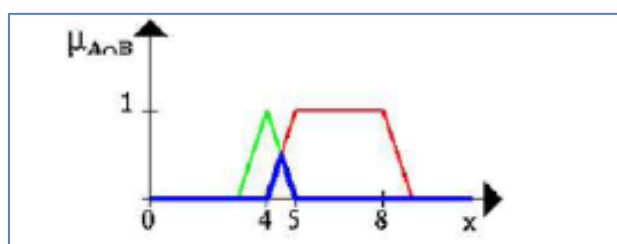


Figura 2-4 Conjuntos difusos Intersección.

La **unión** (OR) de dos conjuntos difusos A y B, es un conjunto difuso C que se obtiene aplicando la operación T – Conorma.

$$C = A \cup B ; \quad (2.3)$$

$$\mu_C(x) = T(\mu_A(x), \mu_B(x)), \quad \forall x \in U \quad (2.4)$$



Figura 2-5 Conjuntos difusos Unión.

El **complemento** de un conjunto difuso A, denotado por $\neg A$ o NOT A se define como:

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \quad \forall x \in U \quad (2.5)$$

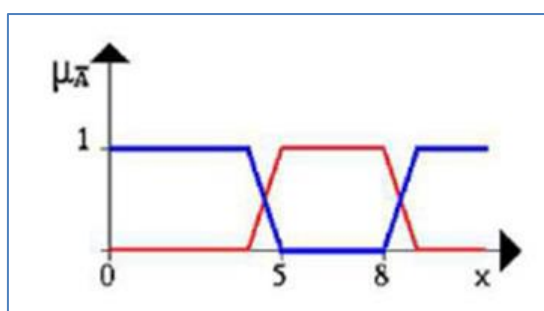


Figura 2-6 Conjuntos difusos negación.

El **complemento relativo** del conjunto difuso A con respecto al conjunto difuso B, es

$$\mu_{B-A}(x) = \mu_B(x) - \mu_A(x), \text{ tal que } \mu_B(x) \geq \mu_A(x) \quad (2.6)$$

En la tabla 2.1 se resumen las operaciones entre conjuntos difusos.

Tabla 2.1 Operaciones sobre conjuntos difusos

Intersección (t-normas)	Unión (t-conormas)
Mínimo $MIN(\mu_A(x), \mu_B(x))$	Máximo $MAX(\mu_A(x), \mu_B(x))$
Producto algebraico $\mu_A(x)\mu_B(x)$	Suma algebraica $\mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$
Producto drástico $MIN(\mu_A(x), \mu_B(x))$ si $MAX(\mu_A(x), \mu_B(x)) = 1, 0$ en otro caso.	Suma drástica $MAX(\mu_A(x), \mu_B(x))$ si $MIN(\mu_A(x), \mu_B(x)) = 0, 1$ en otro caso.
AND de Lukasiewiez $MAX(0, \mu_A(x) + \mu_B(x) - 1)$	OR de Lukasiewiez $MIN(1, \mu_A(x) + \mu_B(x))$

Tabla 2.1 Operaciones sobre conjuntos difusos

Tabla 2.1 Operaciones sobre conjuntos difusos

2.3.4. Controlador difuso.

Un sistema de control basado en lógica difusa está compuesto por los siguientes bloques: una base de reglas, un mecanismo de inferencia, fusificación, y defusificación. [12], En la figura 2.7 se muestra un controlador difuso embebido en un sistema de control de lazo cerrado.

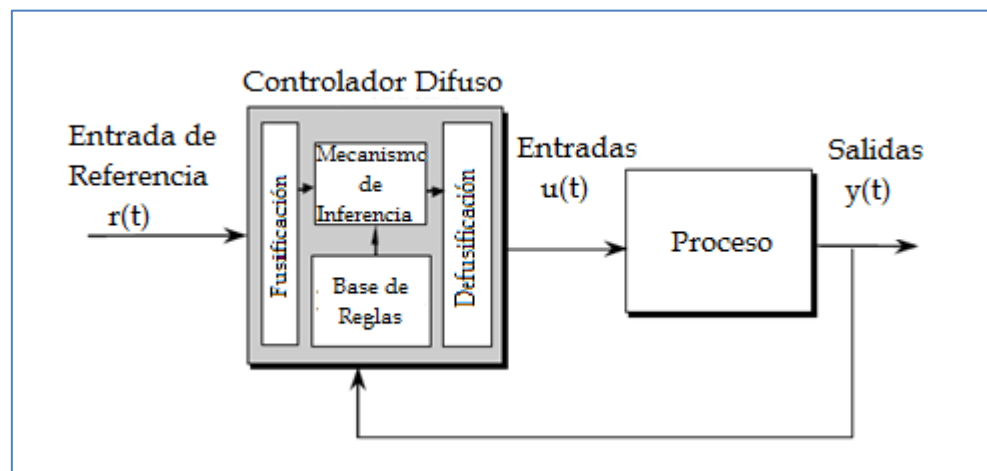


Figura 2-7 Controlador difuso

1. *Base de reglas* (un conjunto de reglas Si-ENTONCES), la cual contiene una cuantificación de la lógica difusa a partir de la descripción lingüística de los expertos del sistema para alcanzar un buen control.
2. Un *mecanismo de inferencia* (también llamado “inferencia de ingeniería” o “módulo de inferencia difuso”), el cual emula la decisión de los expertos para tomar una buena decisión, basados

en la interpretación y aplicando el conocimiento que tienen para controlar la planta de la mejor forma.

3. Una interfaz de *fusificación*, la cual convierte las entradas del controlador en información que el mecanismo de inferencia puede fácilmente usar para activar o aplicar una regla.

4. Una interfaz de *defusificación*, la cual convierte las conclusiones del mecanismo de inferencia en entradas reales para el proceso.

2.3.4.1. Base de reglas.

Para determinar las reglas debemos realizar un trabajo previo que comprende los siguientes pasos: Elección de entradas/salidas del controlador difuso, determinar las variables y valores, determinar el conjunto de reglas que gobernarán nuestro sistema y además definir las funciones de membresía.

Primeramente, la persona experta del sistema debe indicarnos que variable va a considerar como entrada o entradas a nuestro sistema difuso; así también, se debe determinar que variable de salida de nuestro controlador actuará eficazmente sobre la entrada de nuestra planta con el objetivo de realizar un mejor control.

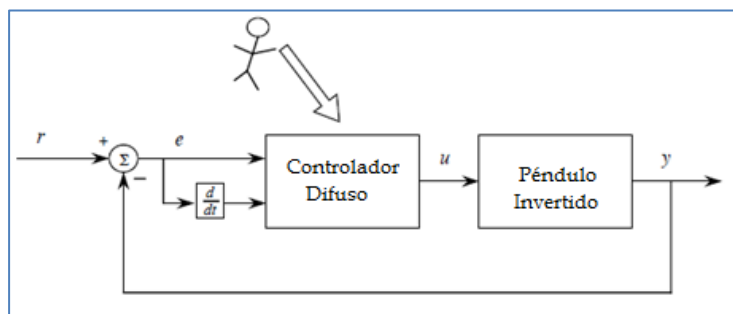


Figura 2-8 Controlador difuso, conocimiento del experto

La descripción lingüística dada por el experto puede ser interpretada, en dos secciones: *las variables lingüísticas* y *los valores lingüísticos*. Como ejemplo tenemos que una variable lingüística puede ser la temperatura, mientras que un valor lingüístico puede ser una de las siguientes: muy_baja, baja, templada, alta o muy_alta.

Las variables y valores lingüísticos proveen un lenguaje al experto para expresar su idea en el control del sistema.

EL siguiente paso es plantear *las reglas de control*, estas usarán la cuantificación lingüística que se realizó previamente, por ejemplo una regla puede ser: Si temperatura es muy baja ENTONCES encender calefactor.

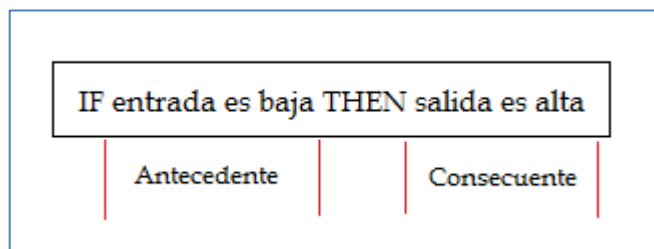


Figura 2-9 Regla de control

Debido a que existe un finito número de variables (entradas) y valores lingüísticos, entonces existirá un valor finito de reglas posibles. En el caso de 2 entradas y 5 valores lingüísticos existirá hasta $5^2=25$ reglas posibles.

Luego debemos definir gráficamente las *funciones de membresía*, con lo que se puede cuantificar el significado de los valores lingüísticos determinados por el experto.

Figura 2.10.

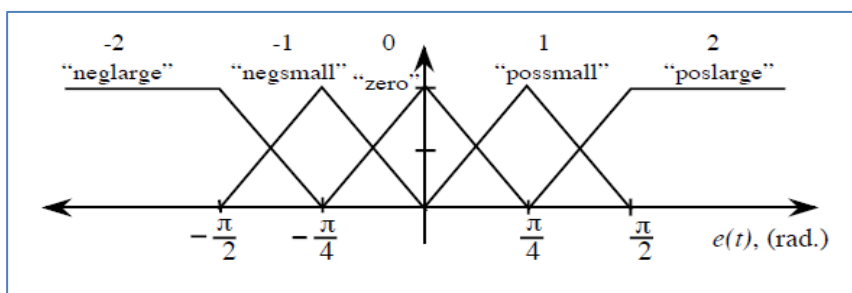


Figura 2-10 Funciones de membresía.

La creación de las reglas formará una *superficie de salida* que condensa el actuar de las reglas en el controlador. A

continuación se muestran superficies de salida para dos controladores.

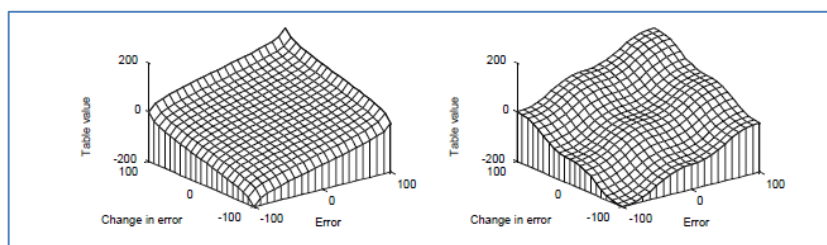


Figura 2-11 Superficie de salida de dos controladores.

2.3.4.2 Interfaz de Fusificación.

El proceso de fusificación es aquel mediante el cual partiendo de un valor de la entrada, se puede determinar un valor numérico de la función de membresía para dicha entrada. Por ejemplo, la función de membresía mostrada en la figura 2.12, puede cuantificar de forma continua si los valores de temperatura T , pertenecen al conjunto de valores que son “baja” (ej: $T=2,5^{\circ}\text{C}$) y además cuantificar el grado de pertenencia de la sentencia lingüística “temperatura es baja” (ej: $u=0,5$). Otro ejemplo sería si $T=5^{\circ}\text{C}$ entonces $u=1$

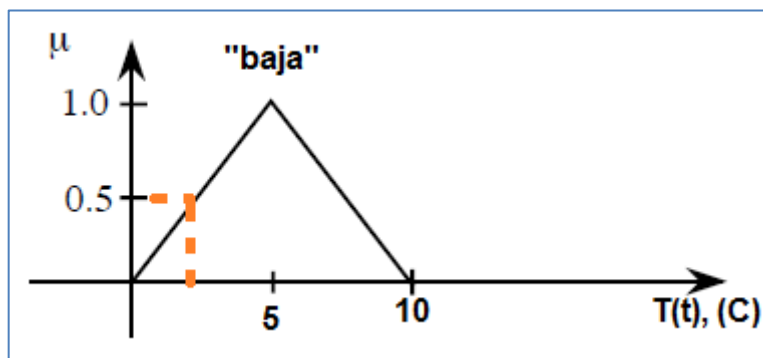


Figura 2-12 Función de membresía

2.3.4.3 Mecanismo de inferencia.

EL mecanismo de inferencia emula la decisión de los expertos para tomar una buena acción de control, basados en la interpretación de las entradas y aplicando el conocimiento que tienen para controlar el sistema en la forma más óptima. El método de inferencia Mamdani expuesto a continuación es uno de los más usados [13], en donde la entrada del mecanismo de inferencia es el dato obtenido en el proceso de fusificación y la salida es un conjunto difuso.

Generalmente el mecanismo de inferencia envuelve dos pasos [12]:

- 1) Implicación.
- 2) Agregación.

En la *implicación*, el antecedente de cada regla es comparado con la entrada del controlador para determinar que reglas corresponden con la situación actual. Si el antecedente de una regla tiene más de una entrada, se debe aplicar un operador difuso para obtener un número que represente el resultado del antecedente para esa regla [13]. Los operadores difusos más usados son **AND** ($\text{Min}(u_a(x), u_b(x))$) y **OR** ($\text{Max}(u_a(x), u_b(x))$). En la Figura 2.13 se muestra la *aplicación del operador OR en dos entradas*.

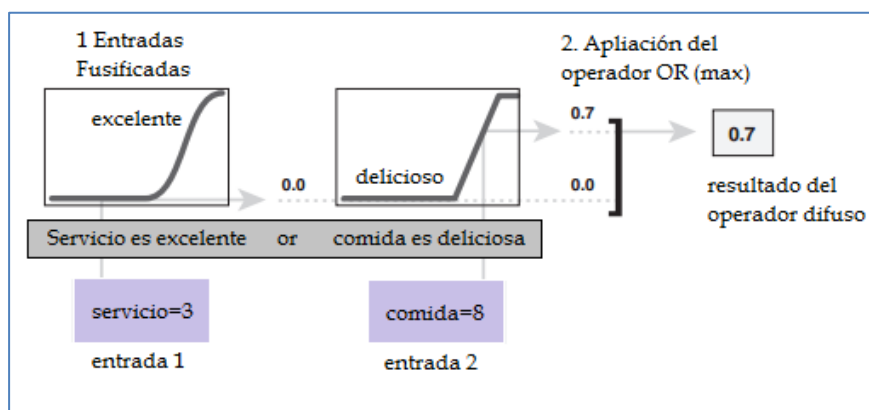


Figura 2-13 Operador OR aplicado a dos entradas

Finalmente el número que resulta del antecedente da como resultado una salida que es un conjunto difuso. Este método de implicación debe ser implementado a cada regla.

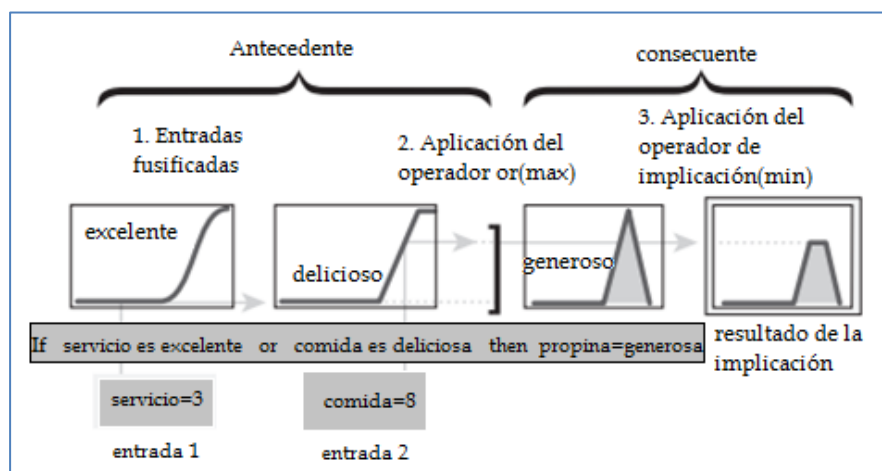


Figura 2-14 Resultado de la implicación

El segundo paso la *agregación*, es el proceso mediante el cual cada conjunto difuso, que representa la salida de cada regla, es combinado dentro de un único conjunto difuso. [13]. La entrada del proceso de agregación es la lista de funciones truncadas que retorna el proceso de implicación por cada regla. Mientras que la salida de este proceso es un conjunto difuso por cada variable de salida. El método más usado para la agregación es el Máximo. En la figura 2.15 observamos cómo se desarrolla dicho método.

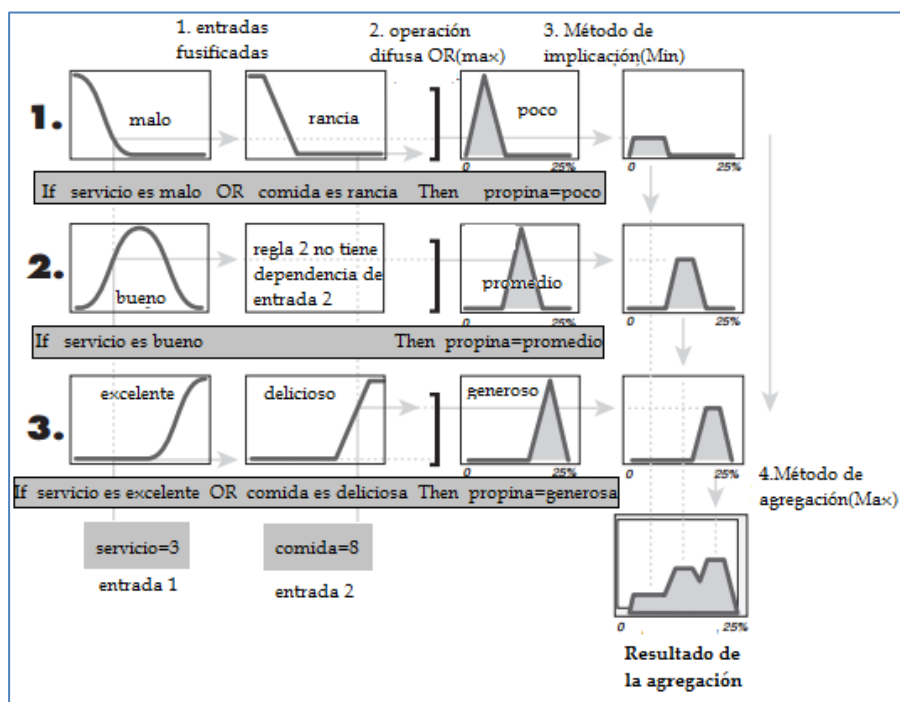


Figura 2-15 Resultado de la agregación.

2.3.4.4 Interfaz de defusificación.

Los métodos más comunes para combinar reglas difusas producen un conjunto difuso. Sin embargo, en los procesos reales es necesaria una salida puntual del controlador. Debido a esto el proceso de defusificación, es el que produce un número que refleja de mejor forma el conjunto difuso que resultó del mecanismo de inferencia. [14] Existen varias técnicas de defusificación, entre las

más usadas tenemos [15] Centro de Gravedad (COG), Método del promedio de los centros y método del máximo.

En el desarrollo de nuestro trabajo utilizaremos el Método del centro de gravedad o método del centroide, que es el más usado en aplicaciones de ingeniería, el cual consiste en calcular el centro del área del conjunto difuso resultante. [11] Este método, puede implicar gran cantidad de cálculo ya que debe calcular el área de figuras eventualmente irregulares.

$$y^* = \frac{\int_v u(z)z dz}{\int_v u(z)dz} \quad (2.7)$$

Que en su forma simplificada, trabajando con valores discretos quedaría:

$$y^* = \frac{\sum_{j=1}^F u(y_j) \cdot y_j}{\sum_{j=1}^F u(y_j)} \quad (2.8)$$

Donde F es el número de elementos discretos del conjunto de salida y y_j es la posición.

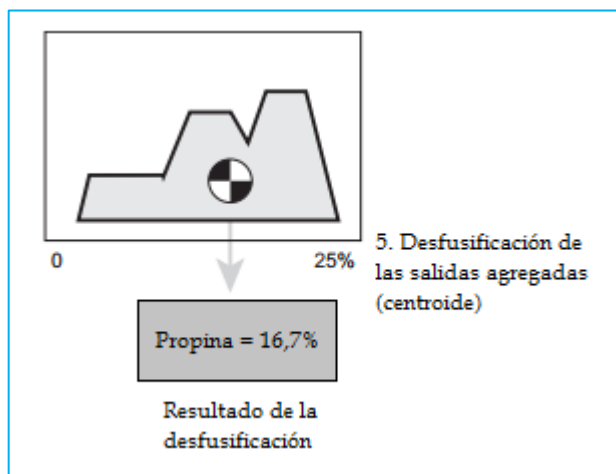


Figura 2-16 Resultado de la defusificación

2.3.5. Sistema difuso Takagi - Sugeno.

El método de inferencia difuso Takagi-Sugeno-Kang, introducido en 1985 [16] es similar al método de inferencia Mamdani en varios aspectos: es necesario el proceso de fusificación de las entradas y la aplicación del operador difuso al antecedente. Sin embargo, la principal diferencia es que en el método Sugeno la función de membresía de salida es una constante o una función lineal [13] por lo que ya no es necesaria la etapa de defusificación, la estructura se muestra como sigue.

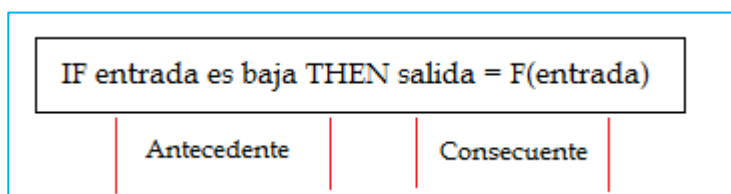


Figura 2-17 Regla de control Takagi-Sugeno

2.4. Control Clásico PID.

Un diagrama de bloques de un controlador clásico se muestra en la figura 2.18. En el diagrama de bloques podemos encontrar el proceso o planta cuya entrada es la variable manipulada y la salida es la variable del proceso, este bloque está precedido por el controlador, que tiene como entrada el error entre el valor de referencia y la variable del proceso multiplicada por una ganancia.

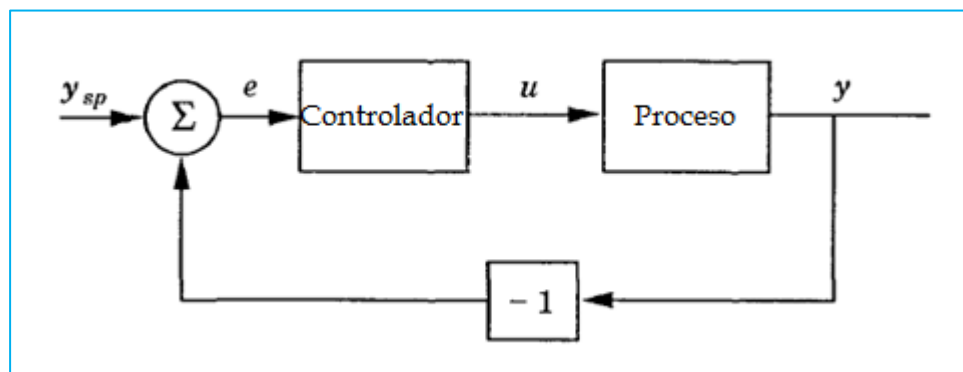


Figura 2-18 Diagrama de bloques de un proceso con control de realimentación

Entre los algoritmos de control más empleados tenemos [17].

ON-OFF

Proporcional (P)

Integral (I)

Proporcional-Integral (PI)

Proporcional-Derivativo (PD)

Proporcional-Integral-Derivativo (PID)

2.4.1. Acción Proporcional.

En el caso de control proporcional puro, la ley de control es

$$u(t) = K_c e(t) + U_o \quad (2.9)$$

La acción de control es simplemente proporcional al error, más una ganancia U_o que es el valor del controlador cuando el tiempo es cero.

2.4.2. Acción Proporcional-Integral.

La principal función de la acción integral es asegurar que la salida del proceso sea igual a la referencia en estado estable. [17]. Si solo es considerada la acción proporcional, entonces es normal que aparezca un error de estado estable distinto de cero.

La acción proporcional-integral PI puede ser representada de la siguiente forma:

$$u(t) = K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt \right] + U_o \quad (2.10)$$

Donde, K_c es la constante proporcional; $e(t)$ es el error; τ_I es el tiempo integral y U_o es el valor del controlador cuando el tiempo es cero.

2.4.3. Acción Proporcional-Derivativa.

EL objetivo de la acción derivativa es mejorar la estabilidad en lazo cerrado. . [17] La acción de un controlador PD puede ser interpretada como si el control se hace proporcional a la predicción de la salida del proceso, donde la predicción es hecha extrapolando el error por la tangente de la curva del error. La estructura de un controlador PD se muestra a continuación.

$$u(t) = K_c \left[e(t) + \tau_D \frac{de(t)}{dt} \right] + U_o \quad (2.11)$$

Donde, K_c es la constante proporcional; $e(t)$ es el error; τ_D es el tiempo derivativo y U_o es el valor del controlador cuando el tiempo es cero.

2.4.4. Acción Proporcional-Integral-Derivativa.

El controlador PID combina en un único controlador la mejor característica de estabilidad del controlador PD, con la ausencia de error en estado estacionario del controlador PI.

$$u(t) = K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_D \frac{de(t)}{dt} \right] + U_o \quad (2.12)$$

Donde, K_c es la constante proporcional; $e(t)$ es el error; τ_I es el tiempo integral; τ_D es el tiempo derivativo y U_o es el valor del controlador cuando el tiempo es cero.

Los controladores PID ideales están caracterizados por tener una función temporal que relaciona la señal de control $u(t)$ con el error, de la forma:

$$u(t) = u(0) + K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(t) dt + K_c \tau_D \frac{de(t)}{dt} \quad (2.13)$$

La función de transferencia es:

$$G_C(s) = K_c \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (2.14)$$

Esta clase de controlador es conocido como controlador PID ideal, debido a que corresponde a una función de transferencia de un sistema no-causal, y por lo tanto, no puede ser implementado con elementos reales (es físicamente irrealizable). La relación entre error y señal de control se puede describir en diagramas de bloques como:

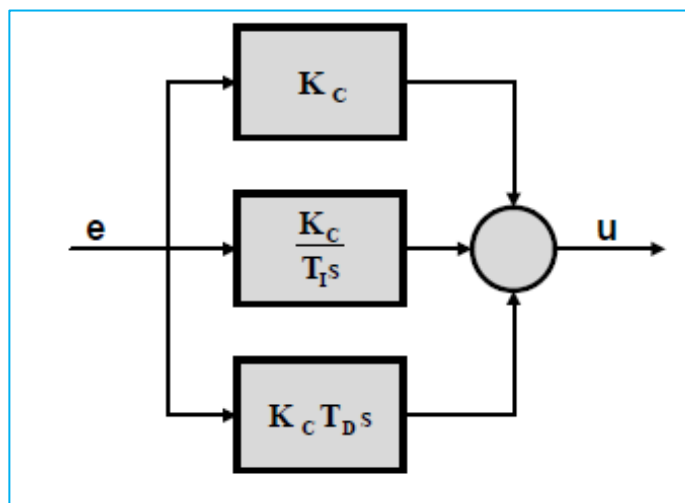


Figura 2-19 Diagrama de bloques de un controlador PID paralelo.

El efecto proporcional, integral y derivativo es aplicado en forma paralela. Por esta razón a este tipo de controladores se los denomina PID tipo “paralelo” o PID “no interactivo” (ya que las acciones no interactúan entre sí).

Los controladores PID-Paralelo que ofrecen las firmas comerciales habitualmente tienen la función de transferencia [18]

$$G_C(s) = K_C \left(1 + \frac{1}{T_I s} + \frac{T_D s}{\alpha T_D s + 1} \right) \quad (2.15)$$

Esta corresponde a un sistema causal. Los fabricantes asignan al coeficiente α valores entre 0.05 y 0.1 (y generalmente fijo).

Debe tenerse presente que este algoritmo es el que se usa casi siempre en los textos de control automático para explicar la combinación de las tres acciones de control, aunque muchos de

los controladores comerciales tienen algoritmos PID interactivo, como el controlador PID “serie”.

En los controladores PID 'serie' o 'interactivos', se aplica primeramente la acción derivativa y luego las acciones proporcional e integral [18] como se muestra en el diagrama de bloques a continuación.

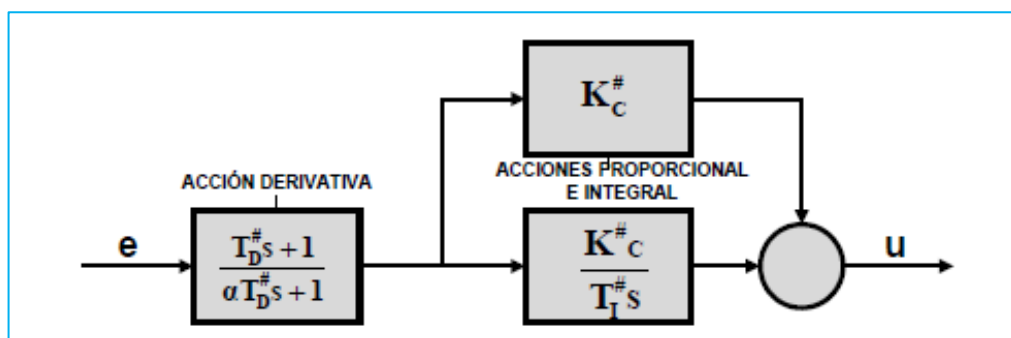


Figura 2-20 Diagrama de bloques de un controlador PID Serie.

Que presenta la función de transferencia

$$G_C(s) = K_C^\# \left(\frac{T_I^\# s + 1}{T_I^\# s} \right) \left(\frac{T_D^\# s + 1}{\alpha T_D^\# s + 1} \right) \quad (2.16)$$

El signo # significa que corresponde a una ganancia, tiempo integral y tiempo derivativo, no en el sentido tradicional (algoritmo

paralelo), sino en la configuración serie. Como en el caso anterior, α es una constante que varía según el fabricante, pero que está comprendida entre 0.05 y 0.1.

Debe aclararse que la respuesta con ambos tipos de controladores dan respuestas similares.

2.4.5. Métodos de ajuste del controlador PID.

Sintonizar (ajustar) un controlador PID significa establecer el valor que deben tener los parámetros de Ganancia (Banda Proporcional), Tiempo Integral y Tiempo derivativo, para que el sistema presente una respuesta satisfactoria. Existen varias formas para sintonizar un controlador PID entre los cuales tenemos: ajuste manual, método de Ziegler-Nichols, método de ajuste analítico, método de ubicación de polos, entre otros. [17]

A continuación presentaremos algunos de estos métodos de ajuste.

Sintonización Manual

La sintonización manual está basada en reglas prácticas usadas por ingenieros de procesos. La sintonización (tuning) es un compromiso entre obtener una reacción rápida y que el sistema sea estable. Tabla 2.2.

Tabla 2.2 Reglas prácticas para la sintonía manual de controladores PID.

Acción	Tiempo Subida	Sobrepico	Estabilidad
Incremento K_p	rápido	incrementa	empeora
Incremento T_d	lento	decrementa	mejora
Incremento $1/T_i$	rápido	incrementa	empeora

El procedimiento se detalla a continuación: [17]

Remueva toda acción integral y derivativa, configurando $1/\tau_i=0$ y $\tau_d=0$.

1. Ajuste la ganancia proporcional K_p hasta obtener la respuesta deseada, ignorando cualquier diferencia que exista entre la salida y la referencia.
2. Incremente la ganancia proporcional K_p un poco más y ajuste la ganancia derivativa τ_d para reducir el sobre pico.
3. Ajuste la ganancia integral $1/\tau_i$ para obtener que el valor de salida sea igual al de referencia y el error de estado estable sea cero
4. Repetir hasta que la ganancia proporcional sea tan grande como sea posible.

En el procedimiento se sugiere ajustar la ganancia derivativa antes que la ganancia integral, sin embargo en la práctica la secuencia debe ser invertida. La ventaja del ajuste manual es que el ingeniero de procesos puede sintonizar el controlador, en línea, desarrollando un conocimiento del comportamiento del sistema. Una desventaja es que puede tomar bastante tiempo desarrollar este conocimiento del sistema y saber si la configuración final fue la óptima.

Dos métodos clásicos para determinar los parámetros de controladores PID fueron presentados por Ziegler y Nichols en 1942. Estos métodos todavía son extensamente usados en su forma original o con alguna variación. [17]

Método de Ziegler-Nichols en lazo abierto

En este método las características estáticas y dinámicas de la planta (Elemento Final de Control + Proceso + Transmisor) se obtienen de un ensayo en lazo abierto, generalmente la respuesta a un escalón.

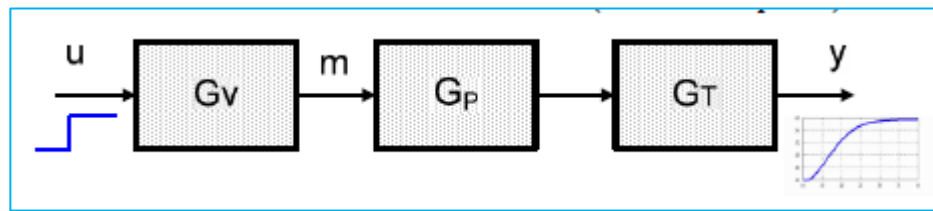


Figura 2-21 Método de Ziegler-Nichols en lazo abierto

Este método también es conocido como Método de la respuesta al Paso, está basado en el registro de la respuesta del sistema en lazo abierto, el cual es caracterizado por los parámetros τ , L y K . dando como resultado la obtención de una modelo simplificado de la planta a controlar como un sistema de primer orden más un tiempo muerto: [18]

$$G(s) = G_V(s)G_P(s)G_T(s) = \frac{Ke^{-Ls}}{\tau s + 1}, \text{ donde } t > 0 \quad (2.17)$$

Este modelo puede ser obtenido experimentalmente de la siguiente forma:

1. Con la planta en lazo abierto, llevar al sistema a un punto de operación de estado estable. Donde tenemos la salida $y(t) = y_0$ para una entrada constante $u(t) = u_0$.
2. En el instante inicial t_0 , aplicar una señal escalón en la entrada, desde u_0 a u_1 (esto debería ser en un rango del 10 al 20% del rango completo).

3. Registrar la salida hasta que se estabilice en el nuevo punto de operación. Supongamos que la curva que se obtiene es la que se muestra en la Figura 2.22.

Podemos calcular los parámetros del sistema de la siguiente forma:

$$K = \frac{u1-u0}{y1-y0}; \quad L = t1 - t0; \quad \tau = t2 - t1; \quad (2.18)$$

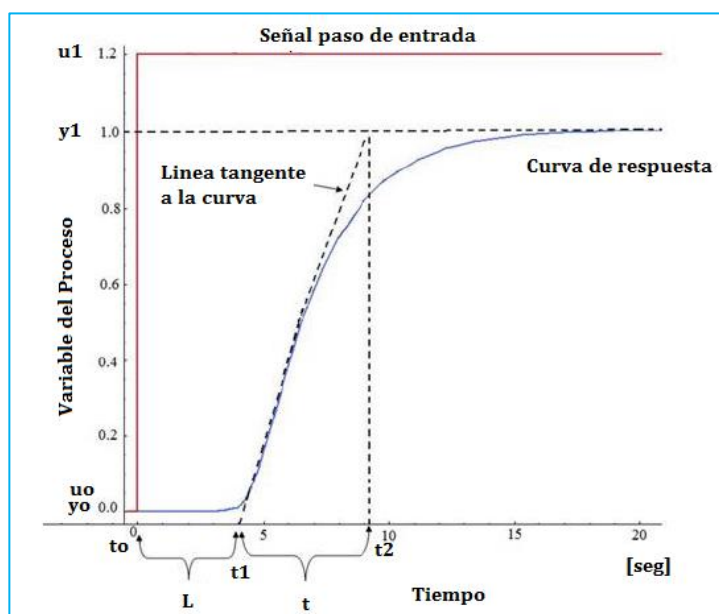


Figura 2-22 Curva de respuesta y obtención de tiempos para Z & N

A partir de estos valores obtendremos la constante $K0$.

$$K0 = \frac{\tau}{K*L} \quad (2.19)$$

Mediante la constante $K0$ se pueden calcular los parámetros del controlador PID por la siguiente tabla:

Tabla 2.3 Parámetros del PID: K_p , T_i y T_d en lazo abierto con método Z-N

Controlador	K_p	T_i	T_d
P	K_0	-	-
PI	$0,9 \cdot K_0$	$3,3 \cdot L$	-
PID	$1,2 \cdot K_0$	$2 \cdot L$	$L/2$

La constante K_p corresponde a la ganancia proporcional, τ_i es la constante de tiempo integral y τ_d es la constante de tiempo derivativa. En el caso de tener el controlador PID configurado con la ganancia integral K_i y derivativa K_d en vez de los tiempos τ_i y τ_d , hay que tener en cuenta las siguientes relaciones: [17]

$$K_i = \frac{K_p}{T_i} \quad (2.20)$$

$$K_d = K_p \cdot T_d \quad (2.21)$$

Con lo cual la tabla de valores para ajustar el controlador PID será la siguiente:

Tabla 2.4 Parámetros del PID: K_p , K_i y K_d en lazo abierto con método Z-N

Controlador	K_p	K_i	K_d
P	K_0	-	-

PI	$0,9 \cdot K_0$	$0,27 \cdot K_0/L$	-
PID	$1,2 \cdot K_0$	$0,6 \cdot K_0/L$	$0,6 \cdot K_0 \cdot L$

Método de Ziegler-Nichols en lazo cerrado

Este método se basa en un lazo cerrado de control, figura 2.23, donde el controlador PID solo tiene ganancia proporcional K_c , mientras que el tiempo integral $\tau_i = \infty$ y el tiempo derivativo $\tau_d = 0$. Aumentando la ganancia K_c el sistema empezará a oscilar, en este caso se obtiene K_u que es la "Ganancia Última" y el periodo de las oscilaciones τ_u denominado "Periodo último", con esto podemos establecer las ganancias del controlador PID. [18]

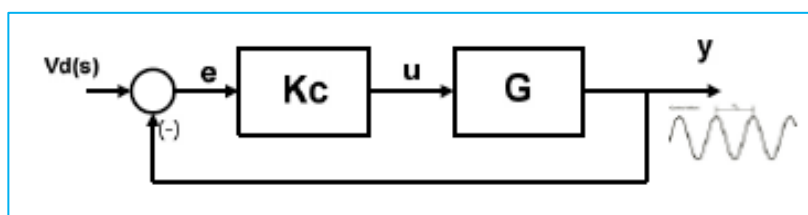


Figura 2-23 Lazo cerrado de control

Tabla 2.5 Parámetros del PID: K_p , T_i y T_d en lazo cerrado con método Z-N.

Controlador	K_p	T_i	T_d
P	$K_u/2$	-	-

PI	$K_u/2,2$	$\tau_u/1,2$	-
PID	$K_u/1,7$	$\tau_u/2$	$\tau_u/8$

En términos de K_i y K_d quedarían:

Tabla 2.6 Parámetros del PID: K_p , K_i y K_d en lazo cerrado con método Z-N.

Controlador	K_p	K_i	K_d
P	$K_u/2$	-	-
PI	$K_u/2,2$	$0,54 * K_u / \tau_u$	-
PID	$K_u/1,7$	$1,18 * K_u / \tau_u$	$0,074 * K_u * \tau_u$

2.5. Motor DC sin escobillas BLDC.

El motor DC sin escobillas BLDC, es un tipo de motor sincrónico de imanes permanentes, consta con imanes permanentes en el rotor y bobinados en el estator, su fuerza contra-electromotriz FEM es de forma trapezoidal. El motor BLDC es alimentado con una fuente DC fija que debe ser conmutada; la secuencia de conmutación es determinada con la posición del rotor. La corriente de fase del motor BLDC, que es típicamente de forma rectangular, es sincronizada con la FEM para

producir un torque constante a cualquier velocidad. El conmutador mecánico de los motores DC con escobillas es reemplazado por conmutadores (switches) electrónicos, los cuales suministran corriente a los bobinados del motor como una función de la posición del rotor. Este tipo de motores es llamado Motor DC sin escobillas, debido a que su desempeño es similar al tradicional motor DC con escobillas o con carbones. [19]

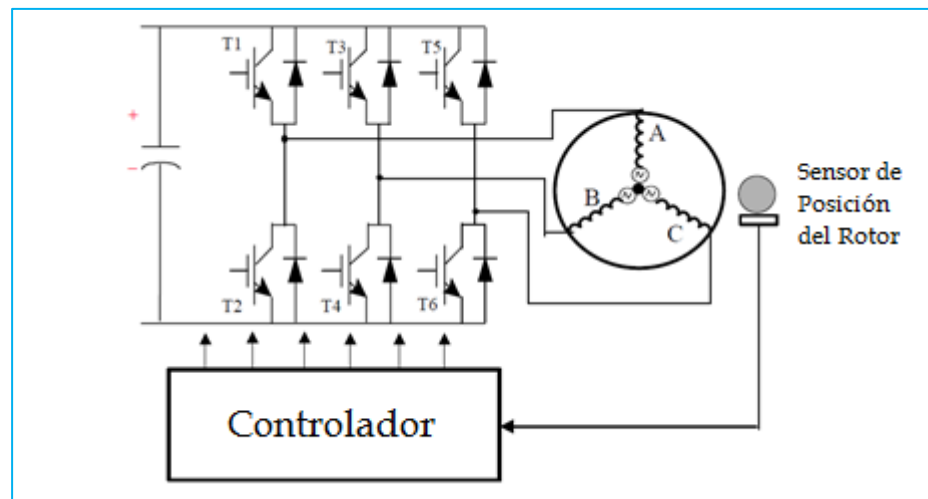


Figura 2-24 Sistema de control típico para un motor BLDC

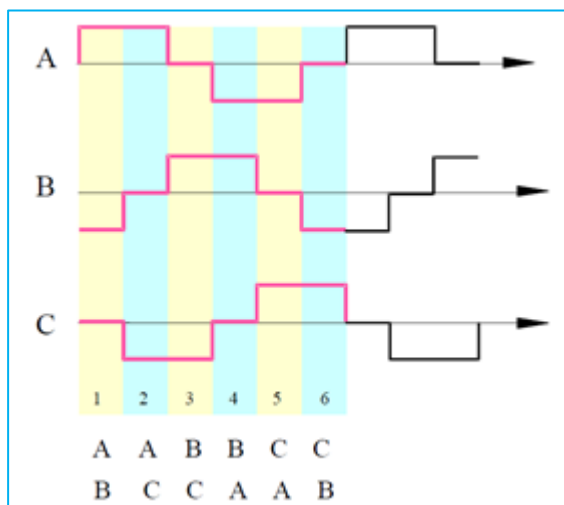


Figura 2-25 Forma de onda de las corrientes trifásicas del motor BLDC

2.5.1. Estructura física.

El BLDC es un tipo de motor síncrono. Esto significa que el campo magnético generado por el estator y el campo magnético generado por el rotor, rotan a la misma frecuencia. Los motores BLDC no experimentan el “slip” o deslizamiento que es normalmente visto en los motores de inducción. [8] En el estator se aloja el bobinado y en el rotor están los imanes. Según su construcción existen dos tipos de motores BLDC [20] pueden ser con rotor interno o con rotor externo, figura 2.26.

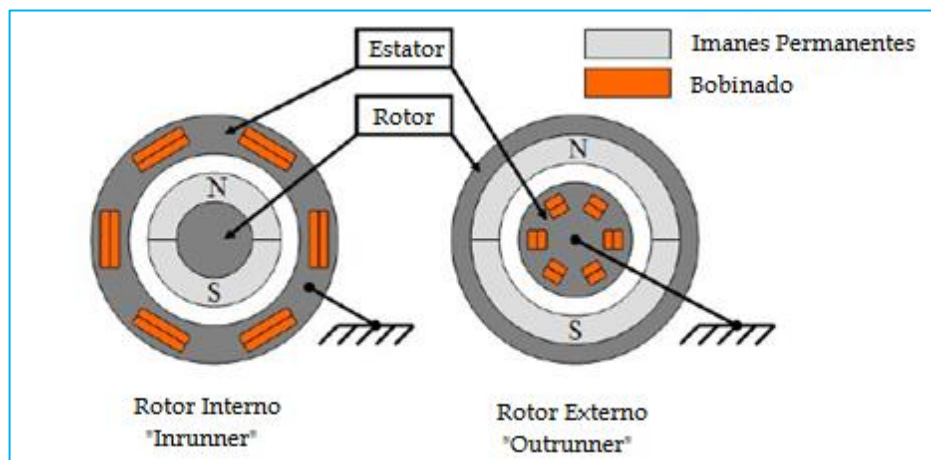


Figura 2-26 BLDC con rotor Interno (izquierda) o rotor externo (derecha).

En nuestro trabajo usaremos un motor BLDC con rotor interno, en la figura 2.27 se muestra la estructura física de dicho motor.

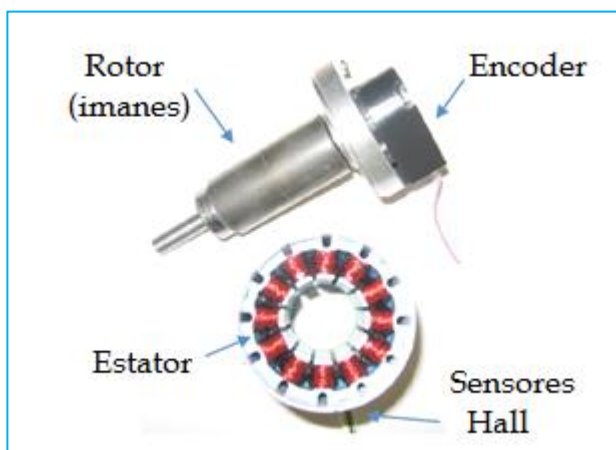


Figura 2-27 Estructura física de un motor BLDC

El Estator

El estator de un motor BLDC consta de laminaciones de acero apiladas, con devanados colocados en el interior de sus ranuras.

La mayoría de los motores BLDC tienen tres devanados conectados en estrella en el estator. Cada uno de estos devanados está construido con numerosas bobinas interconectadas para formar un devanado. Una o más bobinas se colocan en las ranuras que están interconectadas para hacer un bobinado. Cada uno de estos devanados se distribuye en la periferia del estator para formar un número de polos siempre par, como se muestra en la figura 2.28

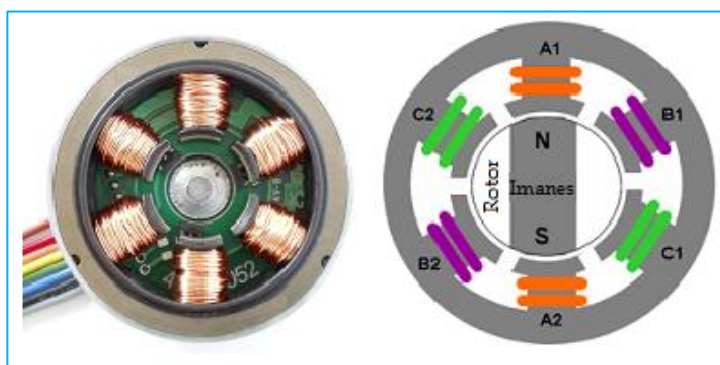


Figura 2-28 Bobinas del estator.

El Rotor

El rotor está hecho de imanes permanentes y puede variar de dos a ocho pares de polos, (Norte y Sur). Basándose en la densidad de campo magnético requerido en el rotor, se elige el material magnético adecuado que tendrá el rotor [8]. Usualmente los imanes de ferrita se utilizan para fabricar imanes permanentes. Actualmente, los imanes creados a partir de

aleación de tierras raras, como el imán de Neodimio y el imán de Samario-Cobalto, están ganando popularidad. En la figura 2.29 se muestran imágenes del rotor.

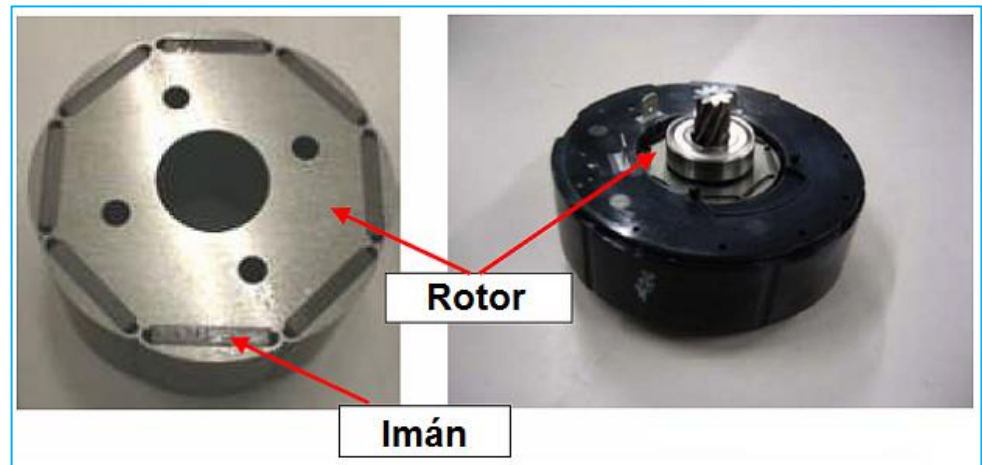


Figura 2-29 Rotor de imanes permanentes.

Sensores de Efecto Hall

La conmutación de un motor BLDC es controlada electrónicamente. Para girar el motor BLDC, los devanados del estator deben ser energizados en una secuencia específica. Es importante conocer la posición del rotor con el fin de entender qué bobinado debe ser energizado siguiendo la secuencia correcta. La posición del rotor se detecta mediante sensores de efecto Hall integrados en el estator, o acoplados en el eje del rotor. La mayoría de los motores BLDC tienen tres sensores Hall

integrados en el estator en el extremo del eje que no maneja la carga, cada ocasión que los polos magnéticos del rotor pasan cerca de los sensores Hall, se genera una señal alta o baja, indicando el polo Norte o Sur del rotor. [8] En base de la combinación de estas tres señales de los sensores Hall, se puede determinar la secuencia exacta de conmutación. Figura 2.30.

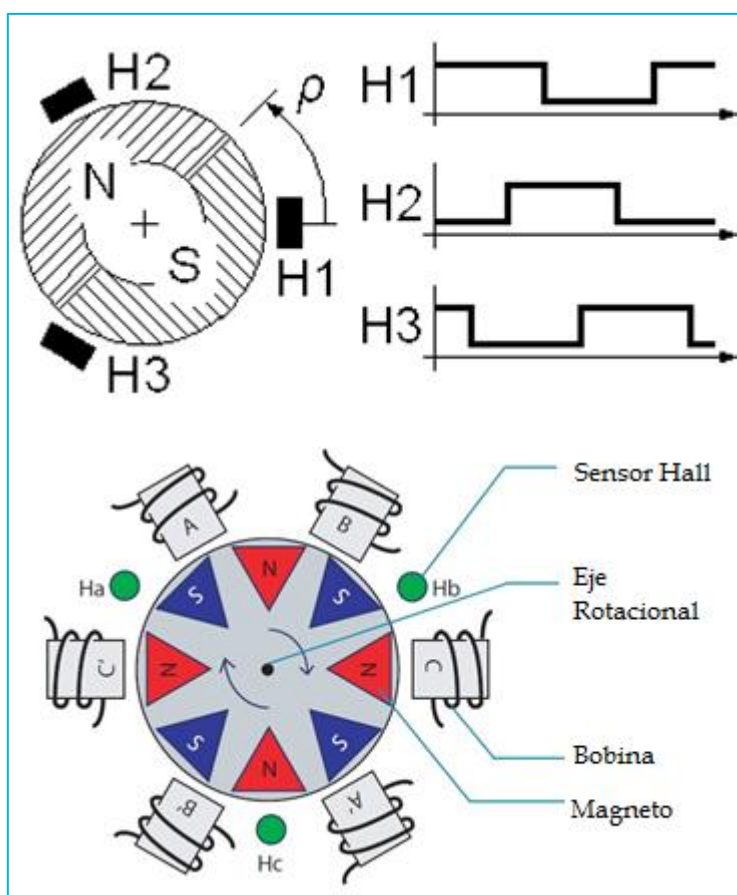


Figura 2-30 Sensores de efecto Hall.

2.5.2. Principio de funcionamiento.

Cada secuencia de conmutación tiene uno de los devanados conectados al mayor potencial (positivo), el segundo devanado está conectado al de menor potencial (negativo) y el tercer devanado está sin energizar. El torque es producido debido a la interacción entre el campo magnético generado por las bobinas del estator y los imanes permanentes del rotor. Idealmente, el torque máximo se produce cuando estos dos campos están a 90° entre sí. Con el objetivo de mantener el motor rotando, el campo magnético producido por los devanados debe cambiar de posición, ya que el rotor se mueve para alinearse con el campo del estator [21].

Cada 60 grados eléctricos de rotación, uno de los sensores Hall cambia su estado. Son necesarios seis pasos para completar un ciclo eléctrico. La corriente de fase debe ser actualizada cada 60 grados eléctricos. Sin embargo, en ocasiones un ciclo eléctrico no corresponde a una revolución mecánica completa del rotor. El número de ciclos eléctricos está determinado por el número de pares de polos del rotor. Los seis modos de funcionamiento se muestran en la figura 2.31 usando la estrategia de conmutación de 120 grados con realimentación de sensores de efecto Hall.

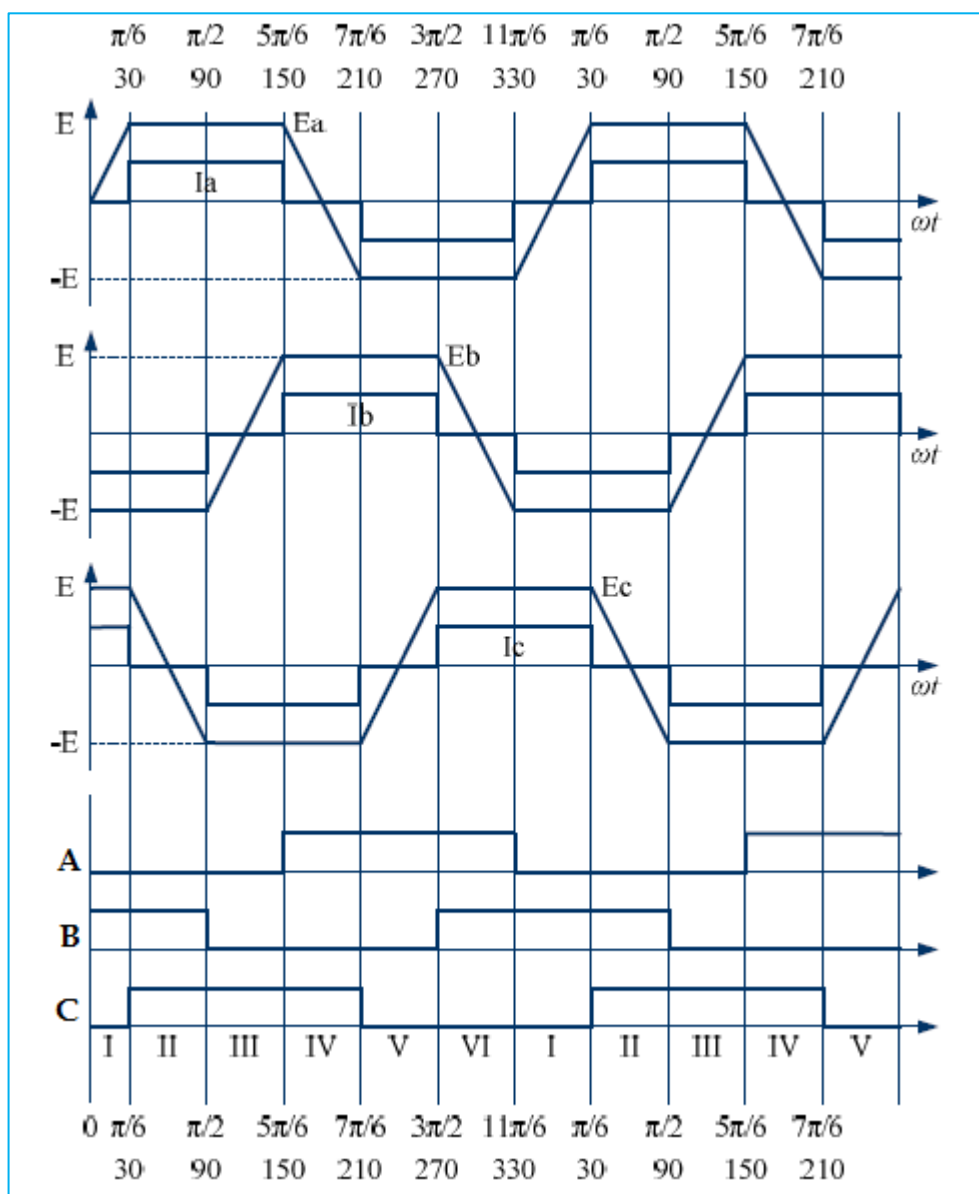


Figura 2-31 Fuerza contra-electromotriz (E_a, E_b, E_c) y corriente de fase (I_a, I_b, I_c) respecto a la posición del rotor (A, B, C).

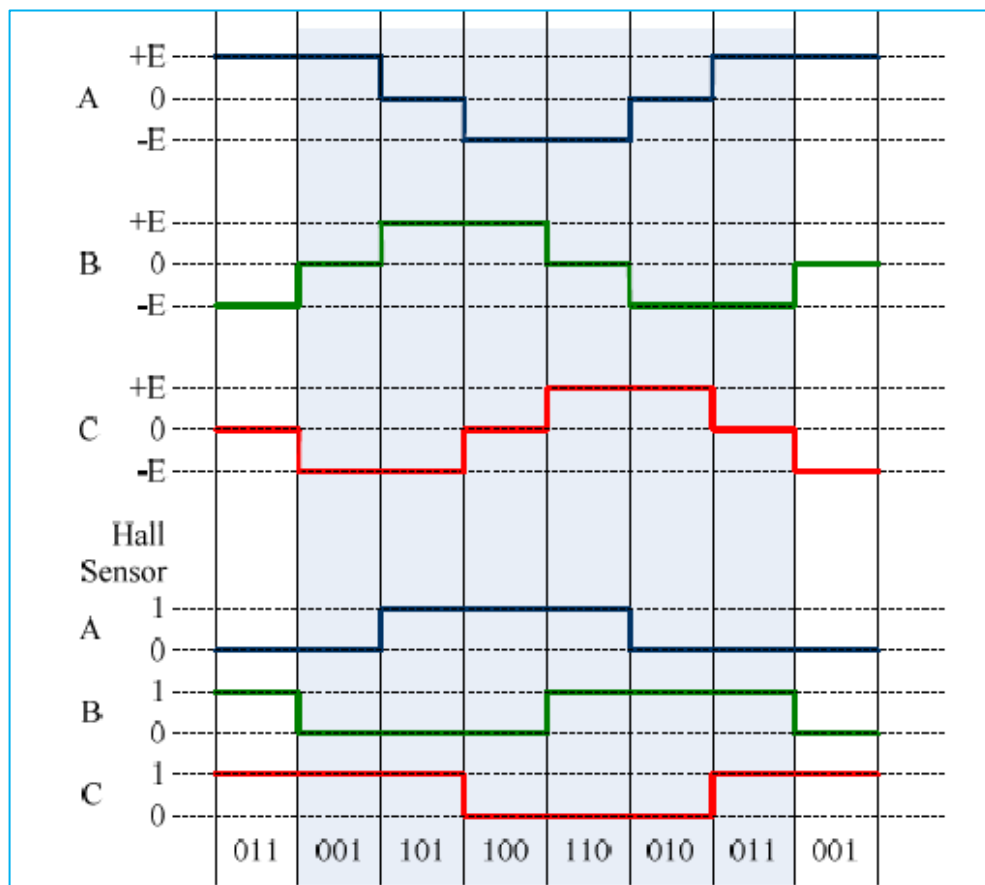


Figura 2-32 Voltajes de Fase y Señal de sensores de efecto Hall.

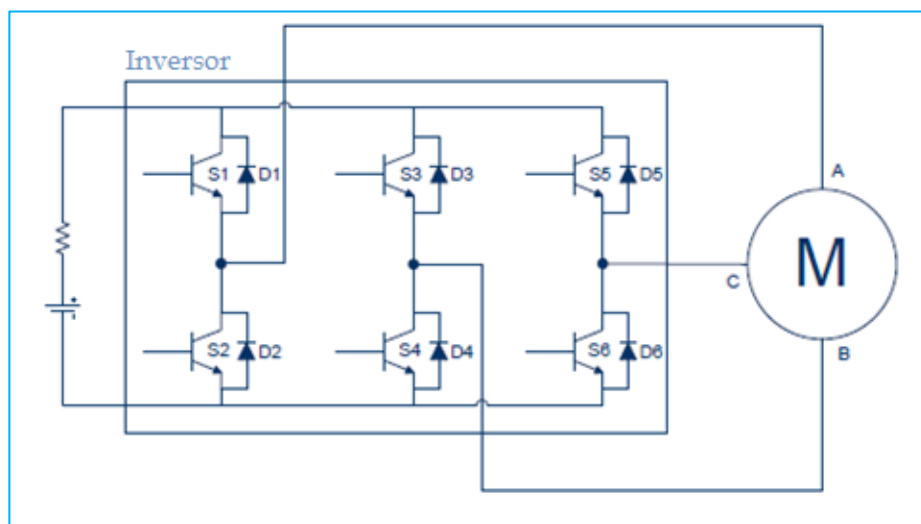


Figura 2-33 Inversor trifásico.

Un inversor trifásico, figura 2.33, es usado para controlar el motor BLDC. Hay seis interruptores y estos deben cambiar en base a las señales de los sensores Hall. La técnica de modulación de ancho de pulso PWM, se utiliza para encender o apagar los interruptores. La secuencia de conmutación se muestra en la tabla 2.7.

Tabla 2.7 Secuencia de conmutación.

Modos	Fases Activas	Fase apagada	IGBT's activos
Modo I ($0^\circ < \theta < 30^\circ$)	B, C	A	Q4, Q5
Modo II ($30^\circ < \theta < 90^\circ$)	A, B	C	Q1, Q4
Modo III ($90^\circ < \theta < 150^\circ$)	A, C	B	Q1, Q6
Modo IV ($150^\circ < \theta < 210^\circ$)	B, C	A	Q3, Q6
Modo V ($210^\circ < \theta < 270^\circ$)	A, B	C	Q2, Q3
Modo VI ($270^\circ < \theta < 330^\circ$)	A, C	B	Q2, Q5

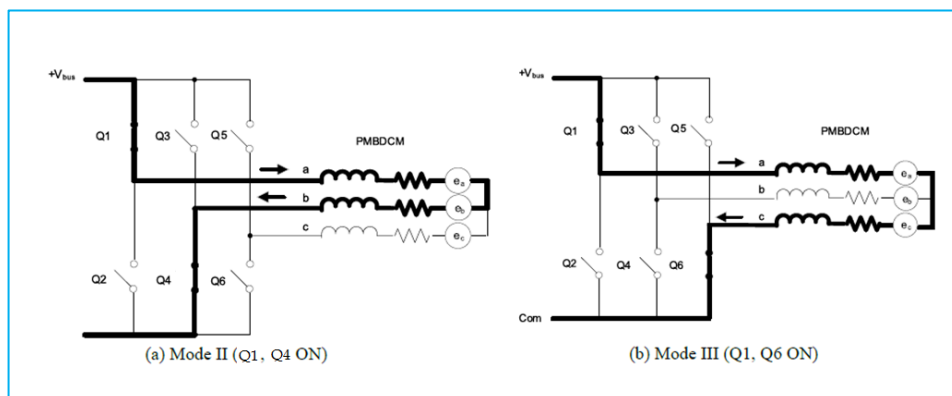


Figura 2-34 Conmutación trapezoidal (seis-pasos) con sensores de efecto Hall. (a) Modo II (Q1,Q4 ON), (b) Modo III (Q1,Q6 ON).

Para variar la velocidad, estas señales deben ser moduladas por ancho de pulso a una frecuencia mucho más alta que la frecuencia de operación nominal del motor. [2]

La frecuencia de la señal PWM debería ser al menos 10 veces mayor que la frecuencia máxima del motor [8]. Cuando el ciclo de trabajo de la señal PWM se varía, la tensión RMS suministrada al estator también. La velocidad y el par del motor dependen de la fuerza del campo magnético generado por las bobinas del estator, que dependen de la corriente a través de ellos. En conclusión, el ajuste de la tensión del estator cambiará la velocidad del motor.

2.5.3. Modelo Matemático del motor.

Las siguientes asunciones son hechas para definir la ecuación diferencial de un motor BLDC. [22] [23].

- 1) Ignorar la saturación del núcleo, también como las pérdidas por corrientes de Eddy y pérdidas por histéresis.
- 2) Ignorar la reacción del estator y que la distribución del flujo magnético en el entrehierro tiene forma trapezoidal, con un tope plano de 120° eléctricos.
- 3) Suponer que los conductores están distribuidos de forma continua e igual sobre la superficie del estator.
- 4) Los elementos de potencia (IGBT's) y diodos de paso libre del circuito inversor, tienen características ideales.

El voltaje de fase de cada bobinado incluye: la caída de voltaje en la resistencia de fase y la FEM inducida, esto se puede expresar como:

$$\mathbf{u}_x = \mathbf{R}_x * \mathbf{i}_x + \mathbf{e}_{\psi_x} \quad (2.22)$$

Dónde:

\mathbf{u}_x – Voltaje de fase, donde x puedes ser A, B o C

R_x – Resistencia de fase. Para un bobinado trifásico simétrico se tiene que $R_A = R_B = R_C = R$

i_x – Corriente de fase.

e_{ψ_x} – FEM inducida por fase.

Según [24] se puede expresar los voltajes de fase de un motor BLDC como una matriz.

$$\begin{bmatrix} u_A \\ u_B \\ u_C \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} L - M & 0 & 0 \\ 0 & L - M & 0 \\ 0 & 0 & L - M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix} \quad (2.23)$$

Dónde:

L – Inductancia de fase.

M – Inductancia mutua entre fases

Mientras que los voltajes de línea se pueden obtener de sus respectivos voltajes de fase.

$$\begin{bmatrix} u_{AB} \\ u_{BC} \\ u_{CA} \end{bmatrix} = \begin{bmatrix} R & -R & 0 \\ 0 & R & -R \\ -R & 0 & R \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} L - M & M - L & 0 \\ 0 & L - M & M - L \\ M - L & 0 & L - M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} e_A - e_B \\ e_B - e_C \\ e_C - e_A \end{bmatrix} \quad (2.24)$$

La *potencia electromagnética* es igual a la suma del producto de la corriente y la FEM de las tres fases.

$$P_e = e_A i_A + e_B i_B + e_C i_C \quad (2.25)$$

Ignorando las pérdidas mecánicas, la potencia electromagnética es totalmente transformada en *energía cinética*, como

$$P_e = T_e * \omega \quad (2.26)$$

Dónde:

T_e – Torque electromagnético.

ω – Velocidad angular de rotación.

De las ecuaciones anteriores se puede obtener que:

$$T_e = \frac{e_A i_A + e_B i_B + e_C i_C}{\omega} \quad (2.27)$$

Para construir un modelo matemático completo del sistema electromecánico, la ecuación de movimiento tiene que incluir:

$$T_e - T_L = J \frac{d\omega}{dt} + B_v \omega \quad (2.28)$$

Dónde:

T_L – Torque de la carga.

J – Momento de inercia del rotor

B_v – Coeficiente de fricción viscoso.

Las ecuaciones 2.23, 2.27 y 2.28 constituyen el modelo matemático en ecuaciones diferenciales de un motor BLDC.

CAPÍTULO 3

3. DISEÑO DE CONTROLADORES.

3.1. Introducción.

En este capítulo se describe el diseño del controlador difuso y el controlador clásico PID, para el control de velocidad del motor de imanes permanentes sin escobillas (BLDC).

3.2. Descripción del sistema.

Nuestro sistema consiste en un equipo educativo destinado al estudio de inversores y manejo de motores, denominado DMD2000, el cual va a controlar la velocidad de un motor BLDC por medio de un controlador

difuso y un controlador PID, con el objetivo de comparar cuál de los dos controladores tiene una mejor respuesta ante el control de un motor BLDC que es un sistema no lineal. Equipo educativo DMD2000. Fig. 3.1



Figura 3-1 Partes del equipo educativo DMD2000.

El equipo DMD2000 tiene internamente el procesador digital de señales DSP, TMS320F2812 de Texas Instruments, capaz de ser programado por software para el control de motores BLDC, adicionalmente cuenta con una sección de potencia de características industriales donde se encuentra el rectificador y el inversor con su circuitería de disparo.

El motor que va a ser usado en las pruebas será un Motor DC sin escobillas con rotor de imanes permanentes de 1HP y como carga tendremos el equipo DMD2000B que es un freno de polvo magnético, el cual estará acoplado al eje del motor.

Finalmente como software de programación usaremos VisSim junto con TI Code Composer Studio (CCS).

3.3. Diseño del controlador difuso.

El control de velocidad de un motor BLDC es complicado cuando se usa técnicas de control convencional, debido a que se necesita un modelo matemático complejo. Usando Lógica difusa eliminamos la necesidad de modelos matemáticos y se puede implementar una solución fácil. [29]

Para nuestro diseño usaremos un controlador Fuzzy PI con características de un controlador Proporcional-Integral. La función de transferencia de dicho controlador es:

$$G(s) = K \left(1 + \frac{1}{T * s} \right) \quad (3.1)$$

Donde $G(s)$ es la ganancia; K es el parámetro de control y T es la constante de tiempo. La ley de control está dada por la siguiente ecuación.

$$u_{PI} = K_p * e + K_I * \frac{1}{T} \int_0^t e * dt \quad (3.2)$$

Donde u es la señal de control; e es el error dado por $e =$ (*valor referencia – valor real*)

Derivando (3.2) tenemos:

$$\frac{du}{dt} = K_P \frac{de}{dt} + K_I * e \quad (3.3)$$

Podemos escribir la ecuación (3.3) en un sistema de tiempo discreto, como:

$$u(kT) - u(kT - T) = K_p \{e(kT) - e(kT - T)\} + K_I * e(kT) \quad (3.4)$$

$$\Delta u = K_p * \Delta e + K_I * e \quad (3.5)$$

Dónde: Δu es el cambio en u ; Δe es el cambio en el error, y e es el error en un periodo de muestreo.

Para poder diseñar nuestro controlador difuso, basado en una estructura de un contralor PID, se realizarán las siguientes definiciones.

E será la variable lingüística del Error e ; ΔE será la variable lingüística para el cambio del error Δe ; ΔU será la variable lingüística para la salida de control Δu .

La figura 3.2 muestra un diagrama de bloques con la implementación de un controlador de lógica difusa (FLC) para el control de la velocidad de un motor BLDC.

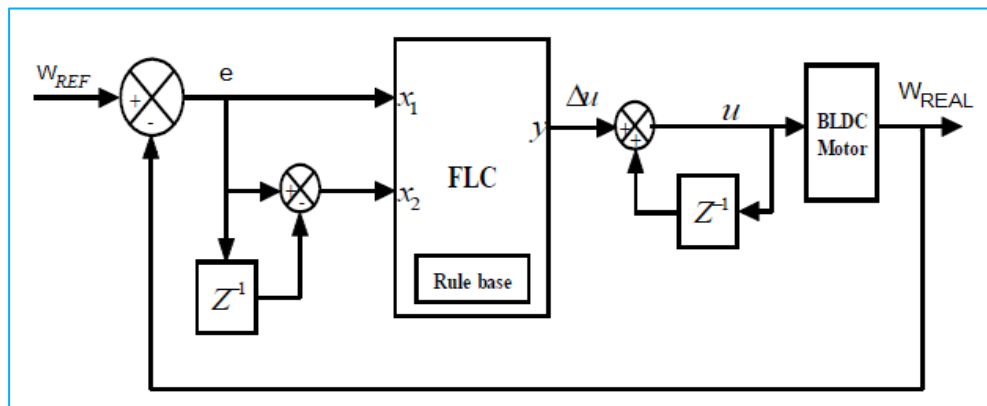


Figura 3-2 Diagrama de bloques de un FLC para el control de un motor BLDC.

Podemos concluir que nuestro sistema tendrá las siguientes interfaces de entrada y salida.

Entrada:

$$X1 = e; \quad e = W_{ref} - W_{real}; \quad X2 = e - e * Z^{-1} \quad (3.6)$$

Salida:

$$y = \Delta u; \quad u = \Delta u + u * Z^{-1} \quad (3.7)$$

3.3.1. Entradas y salidas del controlador.

Las entradas de nuestro controlador difuso serán dos: el *error* de la velocidad de rotación y la *diferencia del error* de rotación.

$$e = \omega_{ref} - \omega_{actual} \quad (3.8)$$

$$\Delta e = error_{previo} - error_{actual} \quad (3.9)$$

Mientras que la salida del controlador difuso será la variación de torque necesaria para mover el motor.

$$\Delta u = \text{variación de torque necesario para mover el motor} \quad (3.10)$$

Realizando nuestro controlador en MATLAB tenemos que:

Escribimos “fuzzy” en la ventana de comandos, se nos abrirá la ventana “FIS Editor” en la cual añadimos una nueva variable de entrada. Figura 3.3.

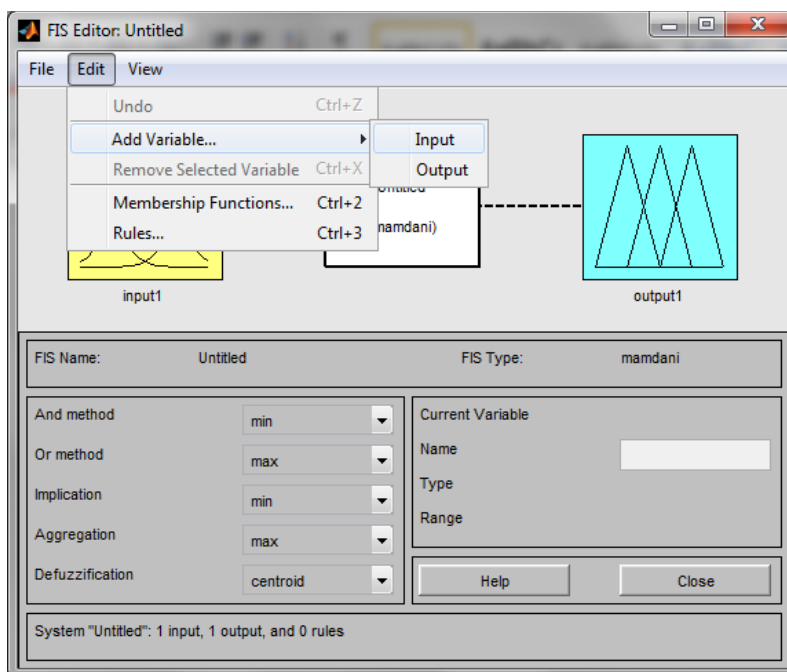


Figura 3-3 Ventana FIS Editor: nueva variable de entrada.

Luego dando clic en cada variable de entrada y de salida, cambiamos el nombre de cada una de ellas Figura 3.4.

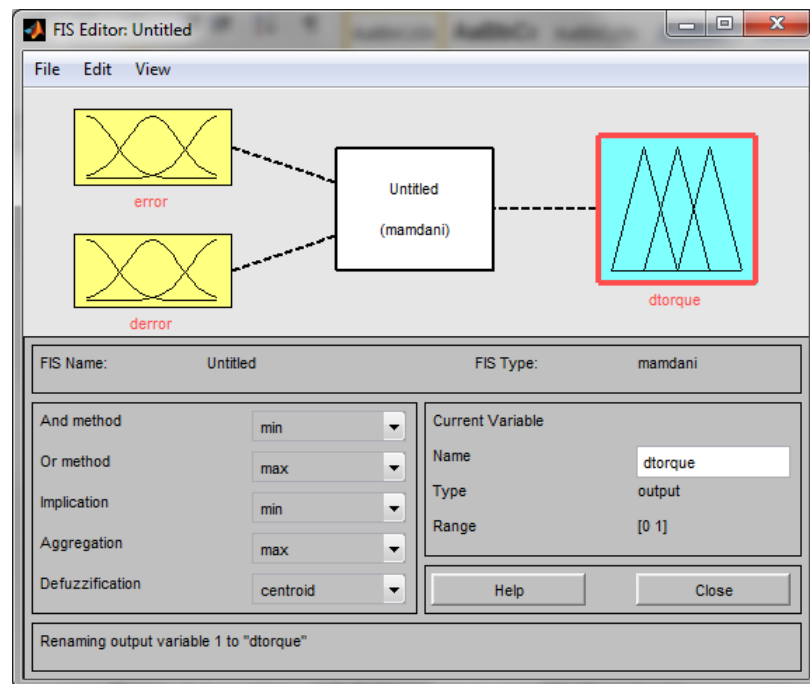


Figura 3-4 Ventana FIS Editor: Asignación de nombres a las entradas.

3.3.2. Definición de conjuntos difusos.

Los conjuntos difusos o funciones de membresía para las entradas y salidas serán de forma triangular y el valor máximo es escalado a 1. Definiremos cinco conjuntos difusos, para cada variable:

1. Grande Negativo [**GN**]
2. Negativo [**N**]
3. Cero [**C**]
4. Positivo [**P**]
5. Grande Positivo [**GP**]

Para añadir funciones de membresía en MATLAB debemos hacer lo siguiente: Damos doble clic sobre una variable de entrada, eliminamos todas las funciones de membresía que están creadas por defecto y luego añadimos las que vamos a usar en este caso son 5 en forma triangular. Figura 3.5.

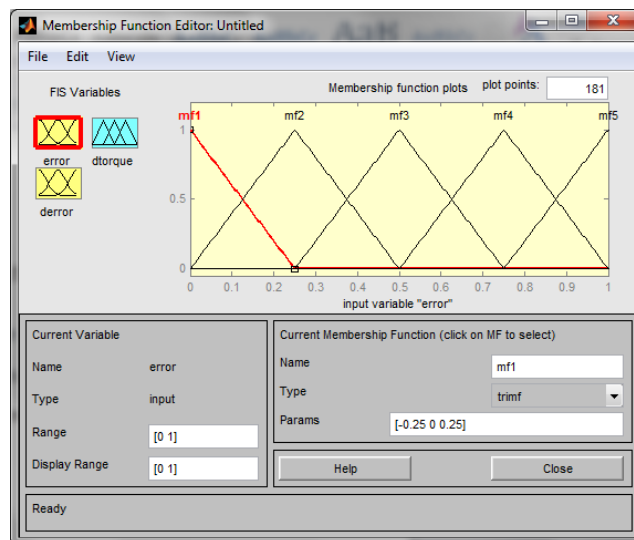


Figura 3-5 Ventana FIS Editor: Funciones de membresía.

Adicionalmente se debe definir el rango de entrada de la variable o el universo de discurso. Se han seleccionado los siguientes rangos:

Entrada “*error*”: **-1000 a +1000**

Entrada “*derror*”: **-100 a +100**

Salida “*dtorque*”: **-15 a +15**

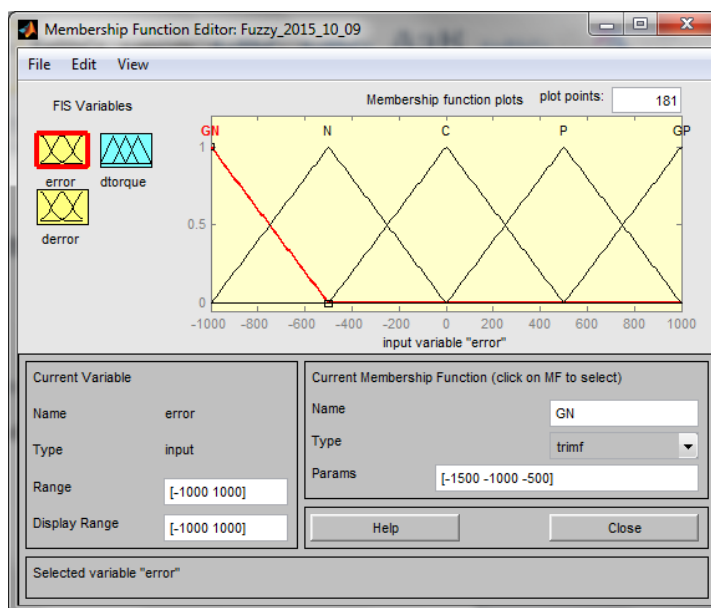


Figura 3-6 Ventana FIS Editor: Rango de la variable de entrada error.

Para la variable de salida se tendrá las siguientes funciones de membresía junto con su rango. Figura 3.7

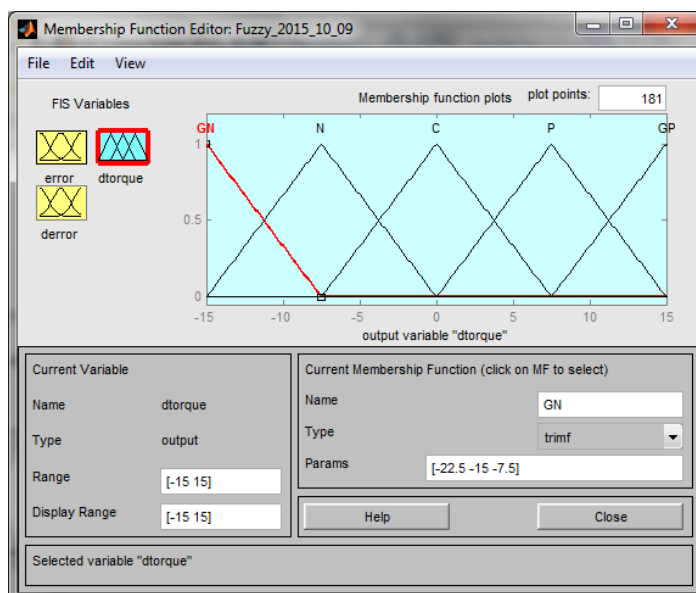


Figura 3-7 Ventana FIS Editor: Rango de la variable de salida dtorque.

3.3.3. Reglas difusas del sistema.

Para el proceso de inferencia debemos definir las reglas que regirán el comportamiento del controlador, estas nacen de la experiencia de la persona que conoce el proceso. Cada variable de entrada (error y derivada del error) puede tomar uno de los cinco valores lingüísticos (GN, N, C, P, GP), por lo tanto 25 (= 5x5) reglas serán formuladas. Las reglas tienen una estructura típica, usando las variables lingüísticas en el antecedente y el consecuente, las mismas que son expresadas en la forma IF-THEN. Las reglas fuzzy tienen la siguiente forma:

$$R^K: IF X_1 \text{ is } A_1^K \text{ AND } X_2 \text{ is } A_2^K, THEN y \text{ is } C^K \quad (3.11)$$

Las reglas para nuestro sistema se muestran en la tabla 3.1

Tabla 3.1 Tabla de reglas para el controlador difuso.

Δe \ e	GN	N	C	P	GP
GN	Grande Negativo	Grande Negativo	Grande Negativo	Negativo	Cero
N	Grande Negativo	Grande Negativo	Negativo	Cero	Positivo
C	Grande Negativo	Negativo	Cero	Positivo	Grande Positivo
P	Negativo	Cero	Positivo	Grande Positivo	Grande Positivo
GP	Cero	Positivo	Grande Positivo	Grande Positivo	Grande Positivo

Podemos leer la primera regla como sigue:

Si el *error* es **GN** (grande negativo) y *derror* es **GN** entonces *dtorque* es **GN**. En la figura 3.8 vemos el editor de reglas del FIS Editor de MATLAB, donde podemos añadir todas las reglas del sistema.

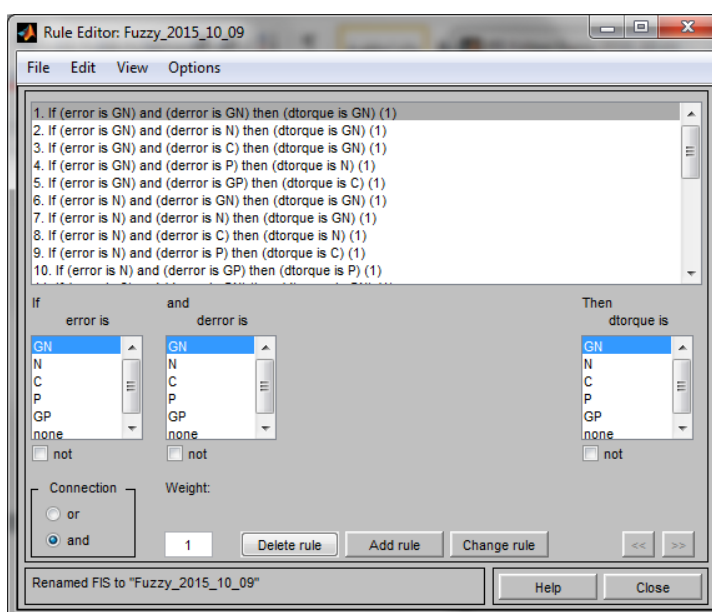


Figura 3-8 Ventana para editar las reglas del controlador difuso.

Adicionalmente se puede obtener la superficie que se crea por las reglas de nuestro controlador a través de la ventana Surface Viewer.

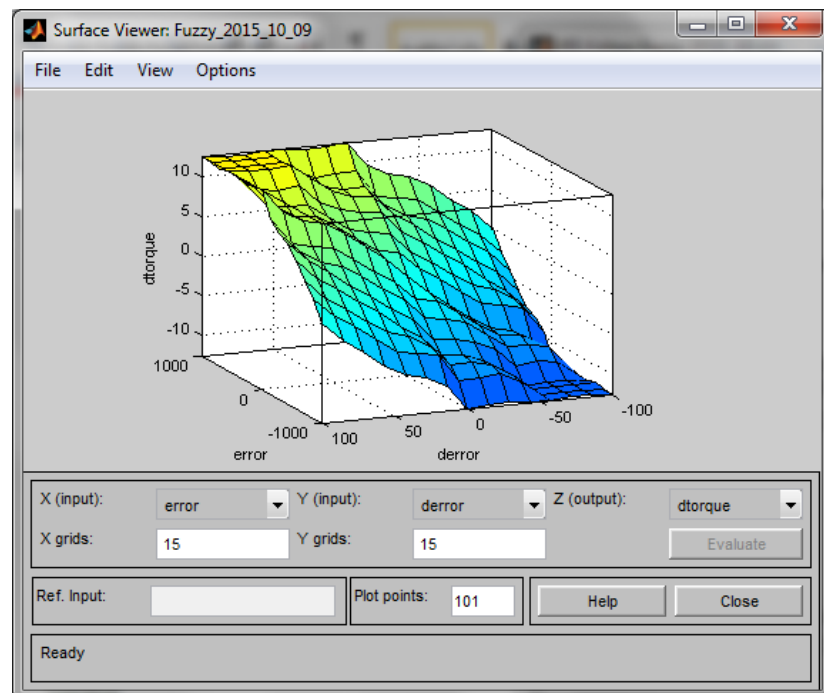


Figura 3-9 Superficie creada por las reglas del controlador.

3.3.4. Defusificación.

El proceso de defusificación, es el que produce un número que refleja de mejor forma el conjunto difuso que resultó del mecanismo de inferencia. [14]

En el desarrollo de este trabajo utilizaremos el Método del centro de gravedad o método del centroide, que es el más usado en aplicaciones de ingeniería, el cual consiste en calcular el centro del área del conjunto difuso resultante.

El asistente para configurar el controlador difuso en MATLAB, nos permite ver como se realiza este proceso de defusificación. Figura 3.10. El resultado del método del centroide nos da un valor puntual que puede ser aplicado a nuestra planta, para la figura mostrada es: 1.76.

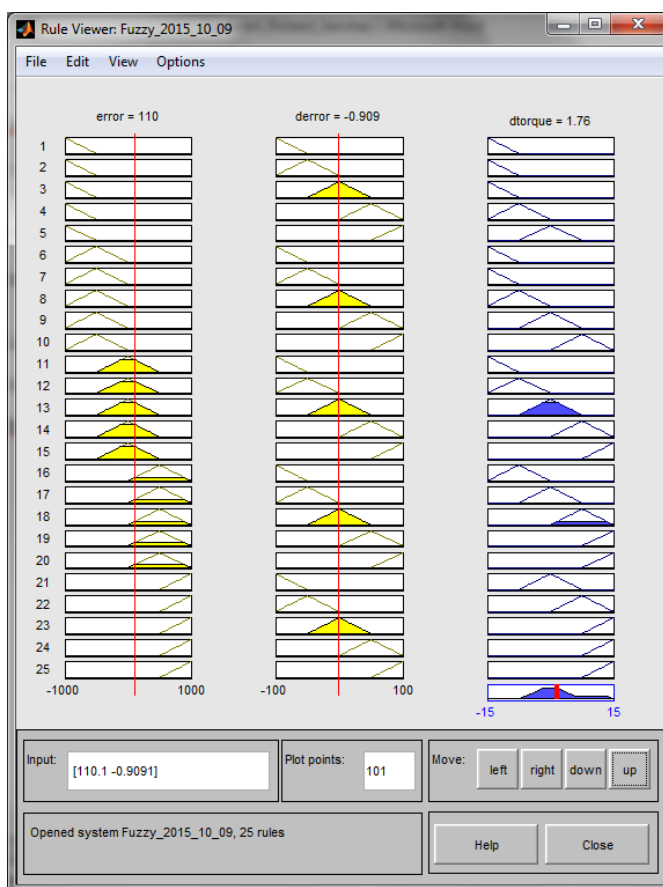


Figura 3-10 Ventana donde se muestra el valor de salida del controlador difuso

3.4. Diseño del controlador PID.

EL sistema usado en este trabajo se muestra en la figura 3.11. Consta de una entrada de voltaje trifásico, el controlador para el *motor BLDC*, un bloque *demux* para seleccionar las variables a ser mostradas, y un *scope* para visualizar las señales más importantes de nuestro sistema.

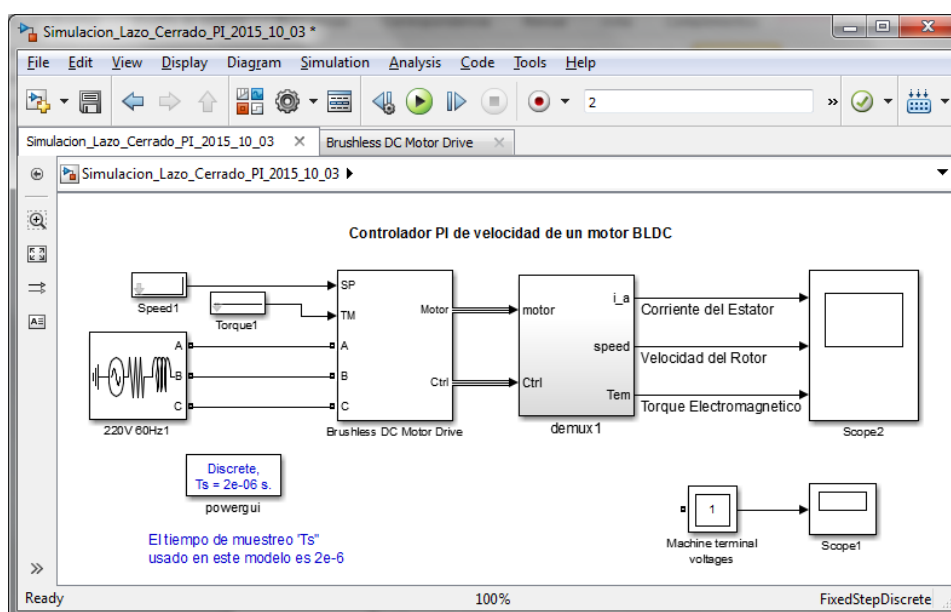


Figura 3-11 Controlador PI de velocidad de un Motor BLDC

Dentro del bloque Brushless DC Motor Drive se encuentra el controlador del Motor DC sin escobillas. El cual consta de los siguientes bloques: Rectificador Trifásico, inversor Trifásico, Motor sincrónico de magneto permanente, un controlador de corriente del tipo ON-OFF y el controlador PI del lazo de velocidad. Figura 3.12.

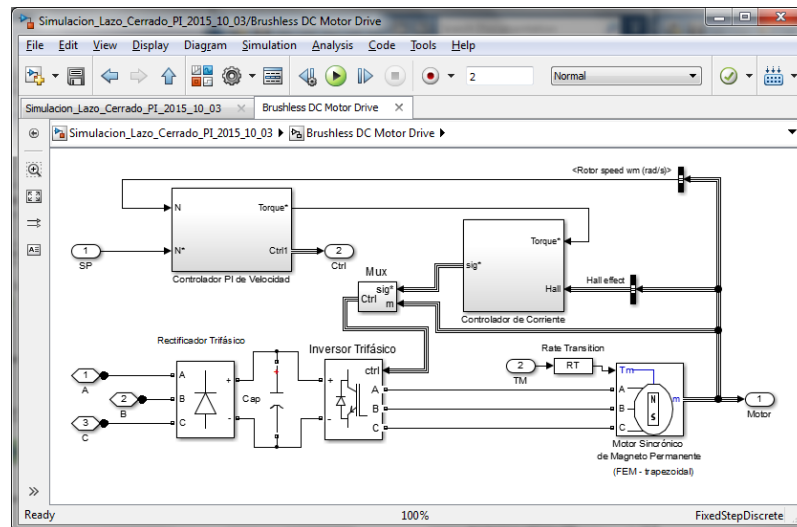


Figura 3-12 Sistema de control de un motor BLDC.

El bloque controlador PI de velocidad se muestra en la figura 3.13, básicamente está constituido por: las ganancias K_p y K_i , varios saturadores que harán que el sistema no se salga de los rangos de trabajo nominales, las señales de referencia de velocidad, la velocidad actual del motor y la salida del controlador que es el torque de referencia al motor.

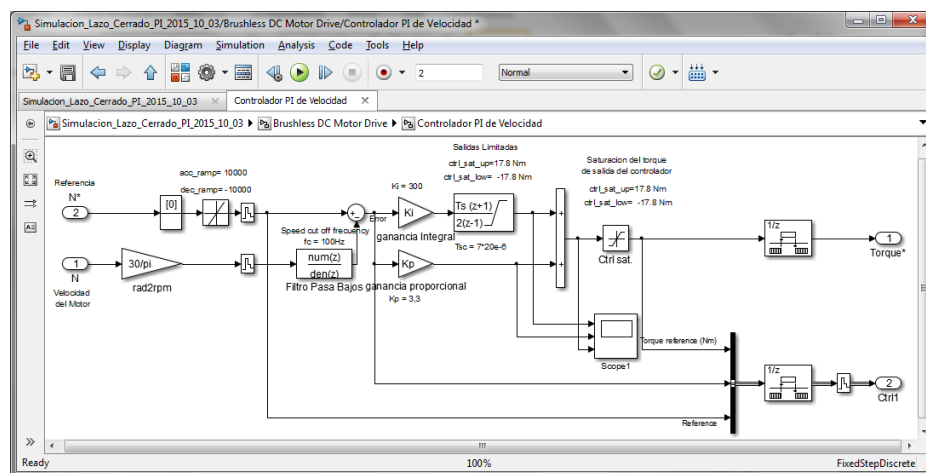


Figura 3-13 Controlador PI de velocidad.

El bloque controlador de corriente es un control ON-OFF que tiene como referencia el Torque impuesto por el controlador PI de velocidad, sabiendo que

$$Te = 2 * Lambda * \#parPolos * i \quad (3.12)$$

Este valor de corriente calculado es multiplicado con la señal que se obtiene del decodificador de los sensores de efecto Hall. Como se muestra en la figura 3.14.

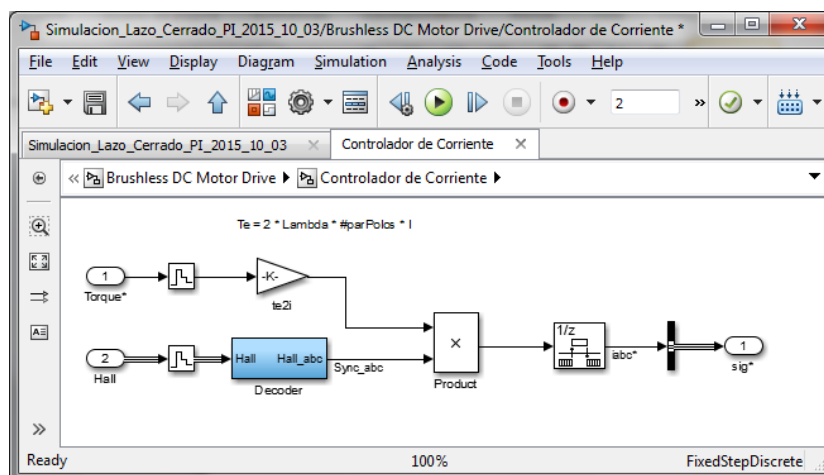


Figura 3-14 Controlador de Corriente.

El decodificador de los sensores de efecto hall se basa en la tabla 3.2 la cual nos sirve para determinar la posición del rotor y la activación de los IGBT's del inversor.

Tabla 3.2 Tabla del decodificador de los sensores de efecto Hall.

Sector	ha	hb	hc	emf_a	emf_b	emf_c	Q1	Q2	Q3	Q4	Q5	Q6
-	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	-1	+1	0	0	0	1	1	0
2	0	1	0	-1	+1	0	0	1	1	0	0	0
6	0	1	1	-1	0	+1	0	1	0	0	1	0
1	1	0	0	+1	0	-1	1	0	0	0	0	1
5	1	0	1	+1	-1	0	1	0	0	1	0	0
3	1	1	0	0	+1	-1	0	0	1	0	0	1
-	1	1	1	0	0	0	0	0	0	0	0	0

En la figura 3.15 se muestran los bloques internos del bloque Decoder.

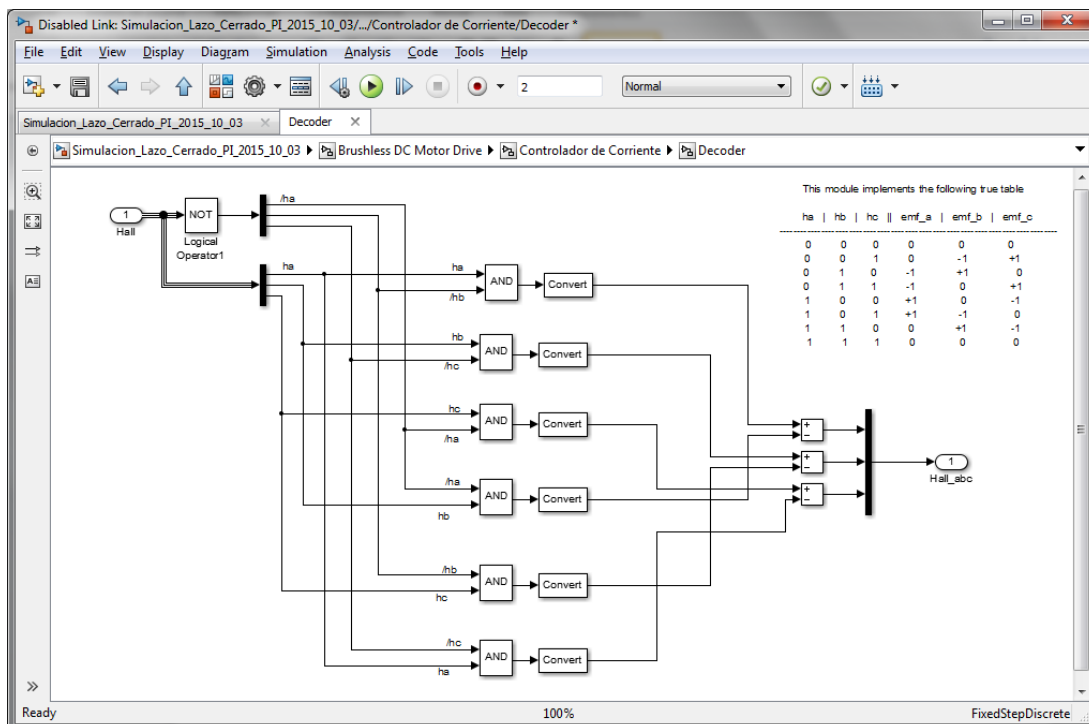


Figura 3-15 Bloque Decoder.

3.4.1. Método de Ziegler y Nichols en lazo abierto.

Este método también es conocido como Método de la respuesta al Paso, está basado en el registro de la respuesta del sistema en lazo abierto, el cual es caracterizado por los parámetros τ , L y K . dando como resultado la obtención de una modelo simplificado de la planta a controlar como un sistema de primer orden más un tiempo muerto: [18].

Podemos calcular los parámetros del sistema de la siguiente forma:

$$K = \frac{u_1 - u_0}{y_1 - y_0}; \quad L = t_1 - t_0; \quad \tau = t_2 - t_1; \quad (3.13)$$

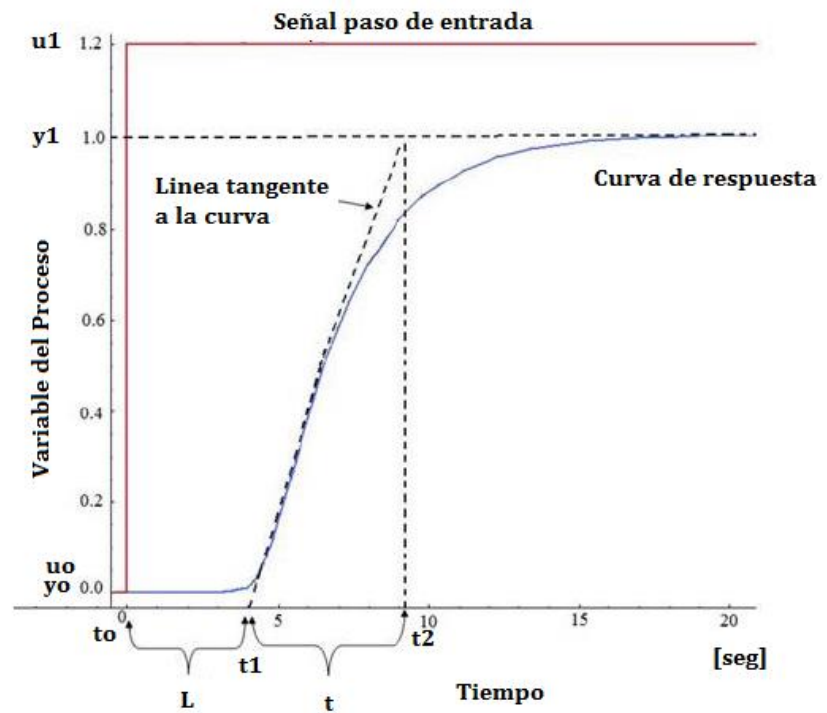


Figura 3-16 Curva de respuesta y obtención de tiempos para Z & N

A partir de estos valores obtendremos la constante K_0 .

$$K_0 = \frac{\tau}{K * L} \quad (3.14)$$

Mediante la constante K_0 se pueden calcular los parámetros del controlador PID por la siguiente tabla:

Tabla 3.3 Parámetros del PID: K_p , T_i y T_d en lazo abierto con método Z-N

Controlador	K_p	T_i	T_d
P	K_0	-	-
PI	$0,9 \cdot K_0$	$3,3 \cdot L$	-
PID	$1,2 \cdot K_0$	$2 \cdot L$	$L/2$

3.4.2. Método de Ziegler y Nichols en lazo cerrado.

En este método usaremos solo la ganancia proporcional K_c del controlador PI, para ello hacemos cero la ganancia integral, y aumentamos K_c hasta que la salida del controlador empiece a oscilar de forma sostenida. En este momento obtenemos la ganancia K_u "Ganancia última" y el periodo de las oscilaciones τ_u denominado "Periodo último", [18] En nuestro caso la ganancia $K_u = 33$ y $\tau_u = 0.005$ como se muestra en la figura 3.17

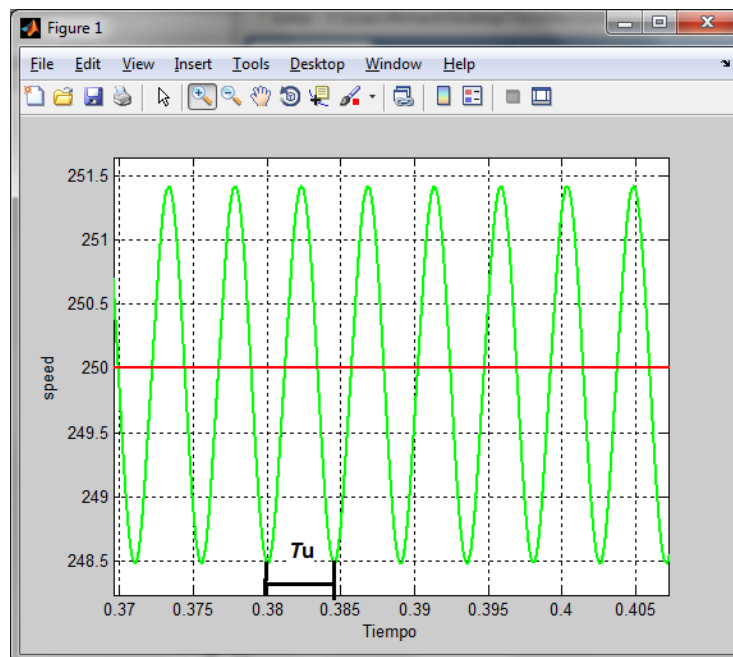


Figura 3-17 Respuesta oscilatoria en lazo cerrado.

Para un controlador PI nos da los siguientes resultados:

Tabla 3.4 Valores del controlador PI por el método de Ziegler-Nichols

Controlador	K_p	T_i	K_i
PI	$K_u/2,2$	$\tau_u/1,2$	K_p/T_i
PI	15	0.0042	3570

Estos valores son un punto de partida para la sintonización de nuestro controlador.

CAPÍTULO 4

4. IMPLEMENTACIÓN DE LA SOLUCIÓN.

4.1. Introducción.

En el capítulo 4, se detallan las características del sistema de desarrollo para el control de motores (DMD2000) mediante procesadores digitales de señal (DSP), usado en el control de velocidad de motores BLDC; así como la implementación del controlador difuso y del controlador PID.

4.2. Diagrama de bloques del sistema de desarrollo experimental DMD2000.

En la figura 4.1, se muestra el diagrama de bloques simplificado de la plataforma experimental para el control de motores. El sistema electrónico está dividido en secciones bien diferenciadas que son [25]:

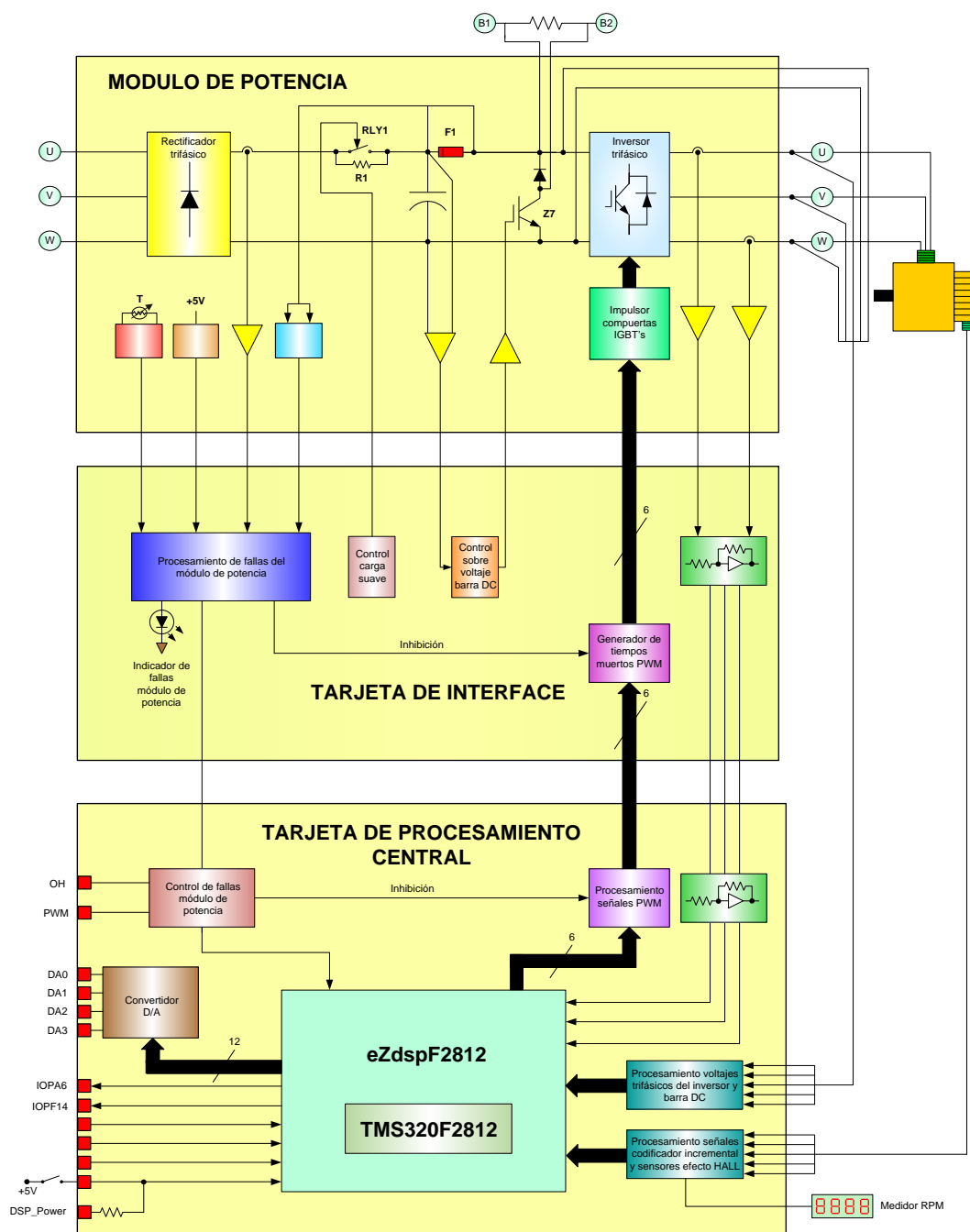


Figura 4-1 Diagrama de bloques de la plataforma experimental DMD2000

Módulo de Potencia

Tarjeta de Interface

Tarjeta de procesamiento central

Tarjeta de desarrollo eZdspF2812

4.2.1. Módulo de Potencia.

Esta sección comprende: el rectificador trifásico no controlado tipo puente, el circuito de carga suave de los condensadores de la barra DC, sistema de control de sobre voltaje de la barra DC con IGBT y resistencia, el inversor trifásico tipo puente de dos niveles. En la tarjeta electrónica del módulo de potencia se estructura la sección de detección de fallas tales como: sobre temperatura del módulo PIM (Power Integrated Module), monitoreo de la fuente de +5 V, sobre corriente de la barra DC y la apertura del fusible F1. Además, se tiene el relé de manejo del circuito de carga suave, el circuito de detección del voltaje de la barra DC, el impulsor para la compuerta del IGBT Z7 que controla el voltaje de la barra DC, el circuito de impulsión para las compuertas de los IGBT's del inversor, los sensores de efecto Hall para la captación de las corrientes de salida en las fases U y W del inversor con los respectivos amplificadores.

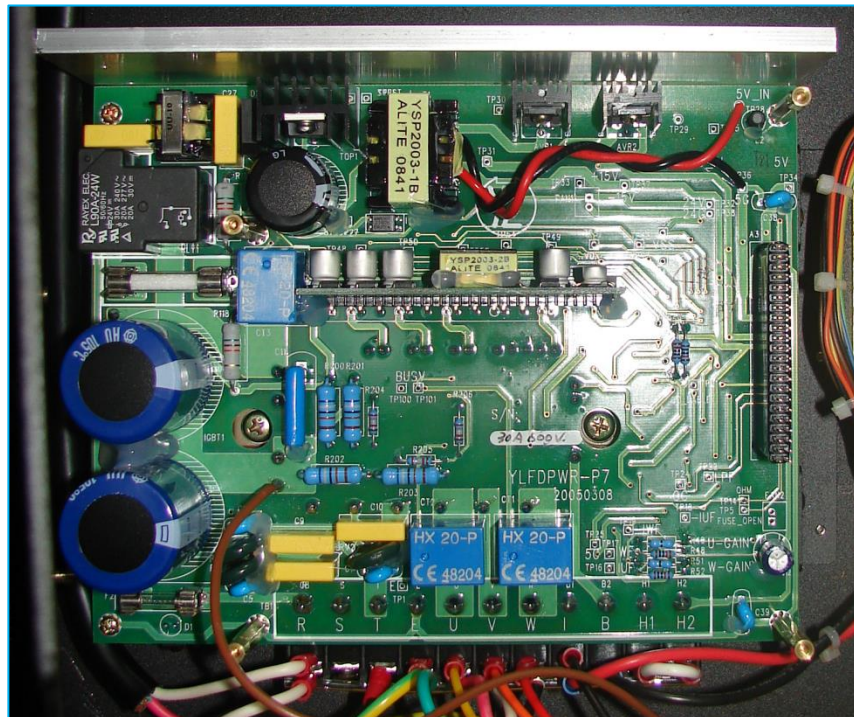


Figura 4-2 Módulo de potencia de la plataforma experimental DMD2000

4.2.2. Tarjetas de procesamiento de señales.

Básicamente las tarjetas de procesamiento de señales son dos: la tarjeta de interface y la tarjeta de procesamiento central.

La tarjeta de Interface, constituida por: el circuito de procesamiento de fallas del módulo de potencia con el LED de visualización respectivo, el circuito de control de carga suave que comanda al relé de carga suave en el módulo de potencia, para que cortocircuite la resistencia de carga suave luego de 1.6

segundos, el control de sobre voltaje de la barra DC que activa el IGBT Z7 cuando el voltaje de la barra supera los 365 Voltios, el generador de tiempos muertos de las señales PWM para los IGBT's del inversor y finalmente el bloque que sintetiza la corriente en la fase V del inversor. Se la muestra en la figura 4.3.

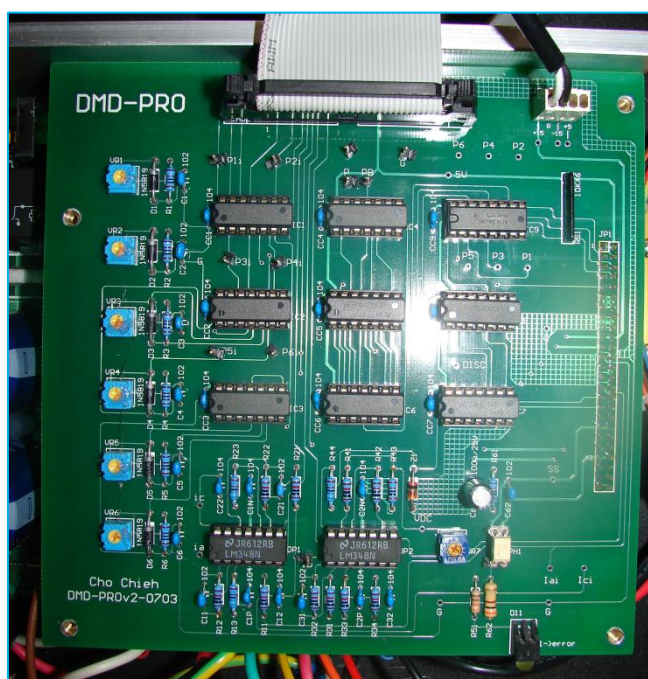


Figura 4-3 Tarjeta de interface.

La tarjeta de procesamiento central consta de: el circuito de control de fallas que tiene la capacidad de visualizarlas mediante el LED OH (Over Heated) e inhibir las señales PWM, el circuito de procesamiento de las señales PWM; el circuito de adición de niveles DC a las señales de corriente del inversor; el circuito de procesamiento de los voltajes del inversor y barra DC; el bloque de

procesamiento de las señales del codificador incremental y sensores de efecto Hall, el mismo que tiene una salida para el medidor de RPM del motor; y, finalmente el bloque de conversión D/A para la visualización de variables del DSP en el panel frontal [25]. Esta tarjeta se visualiza en la figura 4.4.

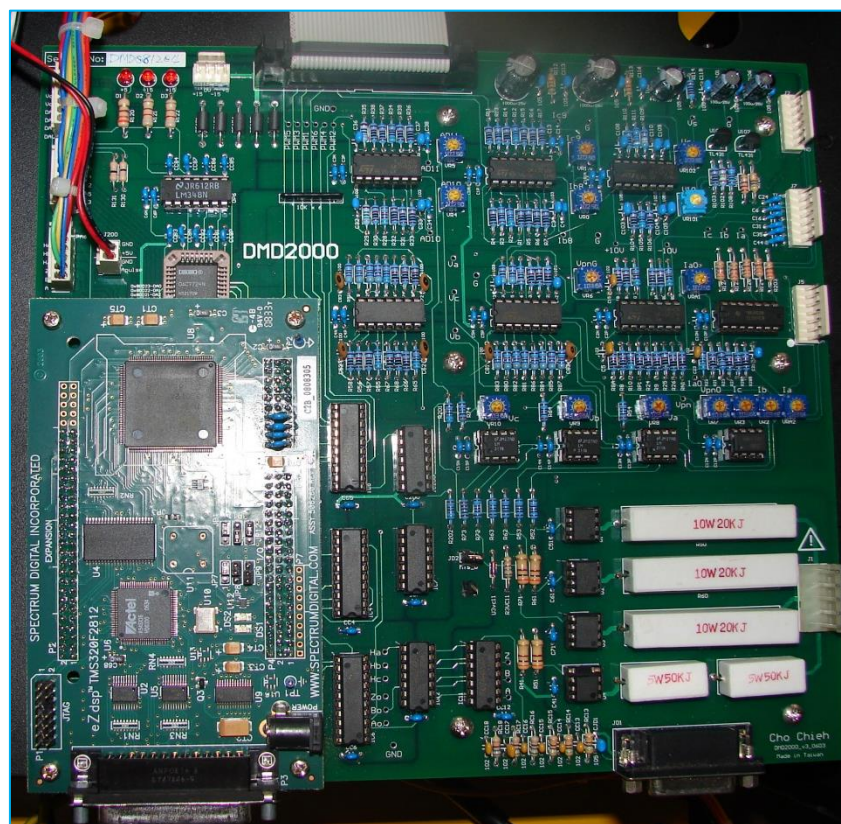


Figura 4-4 Tarjeta de procesamiento central.

4.2.3. Tarjeta de desarrollo de Spectrum Digital (eZdspF2812).

La tarjeta de desarrollo eZdspF2812 diseñada por Spectrum Digital está basada en el procesador digital de señales (DSP)

TMS320F2812 y ubicada sobre la tarjeta de procesamiento central. El computador se conecta a la tarjeta de desarrollo a través del puerto paralelo mediante el emulador de JTAG XDS510PP PLUS. Entre las principales características de esta tarjeta tenemos: Procesador de señales digitales TMS320F2812, velocidad de operación de 150 MIPS, RAM de 18K words, Memoria flash de 128K, Memoria SRAM de 64K, Reloj de 30Mhz, dos conectores de expansión, controlador JTAG, operación con +5 voltios, TI Code Composer Studio driver, conector de emulación de JTAG. [26]

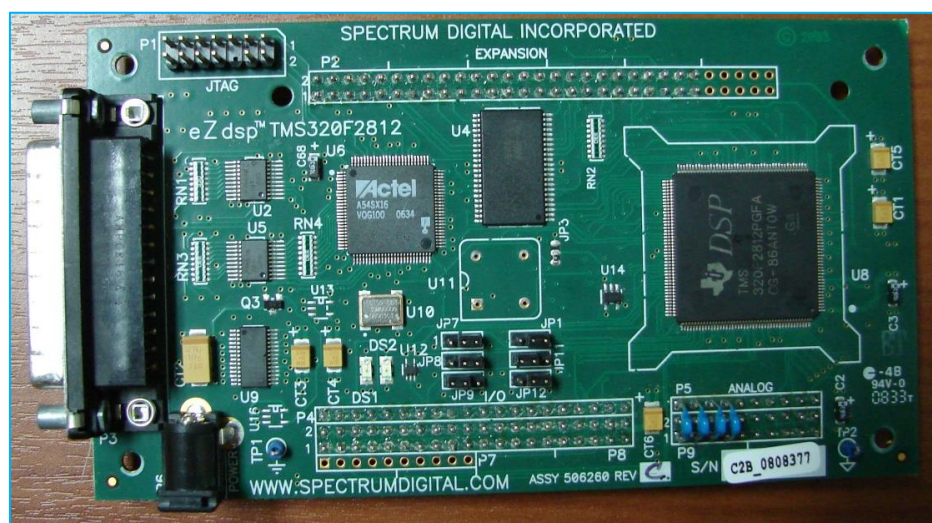


Figura 4-5 Tarjeta de desarrollo eZdspF2812.

4.3. Características del Motor DC sin escobillas a utilizar.

El motor BLDC utilizado en las pruebas experimentales tiene las siguientes características:

Tabla 4.1 Características del Motor BLDC usado en las pruebas

Característica	Valor
Marca	HeadLine
Tipo	130-750MS-ZK-L2
Potencia	750 W - 1 HP
Torque Nominal	28 Kg-cm - 275 N-cm
Torque máximo instantáneo:	84 Kg-cm - 824 N-cm
Velocidad nominal:	3.000 RPM
Corriente nominal:	2,2 A (calculado con voltaje de 220V y $\cos \varphi=0.9$)
Codificador incremental:	1.000 ppr

Algunas fórmulas útiles para el cálculo de variables del motor se describen a continuación.

$$P_m = T * \omega \quad (4.1)$$

Dónde:

P_m – Potencia mecánica del motor. [Wattios]

T – Torque del motor. [N*m]

ω – Velocidad angular de rotación. [rad/seg]

$$P_e = \sqrt{3} * V * I * \cos \varphi \quad (4.2)$$

Dónde:

P_e – Potencia eléctrica del motor. [Vatios]

V – Voltaje del motor. [Voltios]

I – Corriente del motor. [Amperios]

$\cos \varphi$ – Factor de potencia del motor. [<1]

HEADLINE		SERVO MOTOR	
TYPE	130-750MS-ZK-L2		
P R W/HP	750/1	Pole	4
T R Kg-cm	28	MAX	84
N R R.P.M	3000	MAX	4000
SER. NO.	HA0833270006		
HEADLINE ELECTRIC CO;LTD.			

Figura 4-6 Datos de placa del motor BLDC.

Todo el sistema en conjunto se visualiza en la figura 4.7.



Figura 4-7 Sistema de pruebas DMD2000.

4.4. Detalle del software de programación, VisSim.

VisSim, es un entorno visual de desarrollo basado en modelos y simulación dinámica de sistemas complejos. Este combina una interfaz gráfica muy intuitiva con un sistema de cálculo matemático muy poderoso para representar con precisión sistemas lineales y no lineales y simular su comportamiento en tiempo continuo, en tiempo discreto o en ambos.

Además, el entorno de desarrollo de VisSim hace fácil pasar entre las etapas de construcción del modelo, simulación, optimización, validación y generación de código, lo que permite crear, verificar y validar prototipos antes de comprometerse con el diseño.

VisSim es usado en muchos ámbitos: comercial, gubernamental y académico a través del mundo para resolver problemas del mundo real en una variedad de disciplinas de ingeniería como: Comunicaciones, Electrónica, HVAC, dispositivos médicos, Transmisión de energía eléctrica y gas, etc. [27].

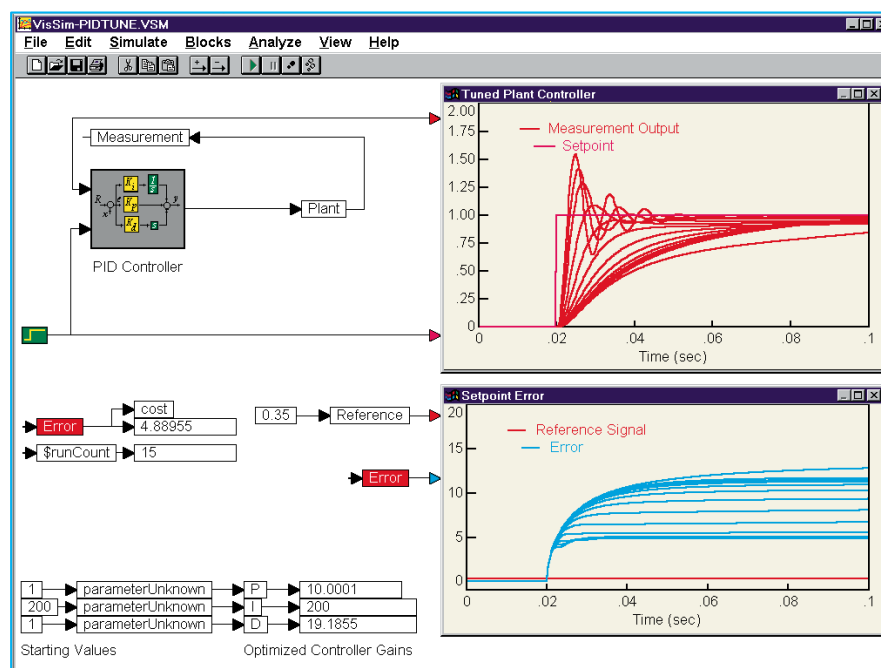


Figura 4-8 Entorno gráfico de VisSim.

VisSim con su programación gráfica (similar a Simulink de MATLAB) tiene librerías especializadas para el diseño de accionamiento de motores eléctricos, este software se puede enlazar con tarjetas de desarrollo de la compañía Spectrum Digital Incorporated, tales como la eZdspF2812 que embebe al procesador TMS320F2812. VisSim interactúa con el software Code Composer Studio (CCS) para la carga y prueba de las aplicaciones en el DSP.

4.5. Diseño de pruebas.

Las pruebas serán realizadas en el equipo didáctico DMD2000 que se encuentra disponible en el laboratorio de Electrónica de Potencia de la FIEC-ESPOL. Este equipo consta de un procesador digital de señales,

que puede ser programado mediante el software VisSim en un computador, junto con todos sus periféricos para poder manejar los disparos de un inversor trifásico, y así controlar la secuencia de disparo para movilizar un motor de magneto permanente BLDC. Como primer paso se desarrollará un sistema de control de velocidad en lazo abierto, es decir, se verificará la secuencia de disparo de los IGBTs, se tomarán las corrientes del motor así como la velocidad en su eje a través del encoder incremental, sin tener un control exacto de la velocidad del motor.

Seguido se desarrollará el controlador de lógica difusa para el control de velocidad de un motor BLDC para esto se usará el software VisSim en la programación y se harán pruebas sucesivas para determinar la respuesta de este controlador ante variaciones en la referencia de velocidad y en la carga acoplada al eje del motor.

Finalmente, se aplicará un controlador PI al sistema, realizando pruebas sucesivas similares a las ejecutadas con el controlador de lógica difusa, esto nos ayudará en las conclusiones de nuestro trabajo.

4.5.1. Control de velocidad en lazo abierto.

Para ilustrar el funcionamiento del equipo DMD2000 y su interacción con el motor BLDC se desarrollará un ejercicio donde se describen ciertos bloques útiles para nuestro estudio [28],

controlando a dicho motor en lazo abierto. El sistema de control se muestra en la figura 4.9. Con este ejercicio podemos modificar la referencia, verificando el cambio en velocidad del motor.

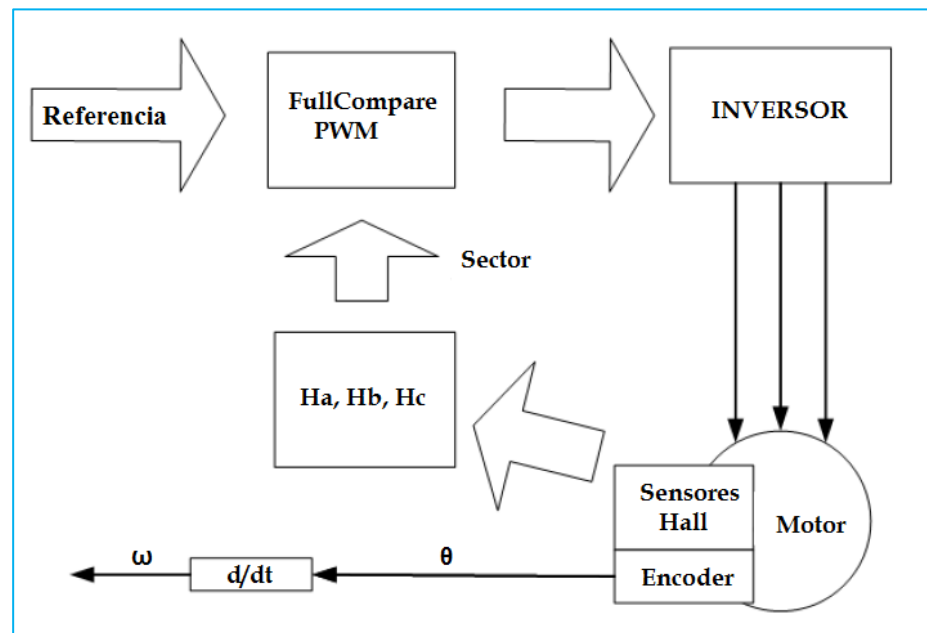


Figura 4-9 Sistema de control de velocidad en lazo abierto de un motor BLDC.

El desarrollo en VisSim del sistema que vamos a usar se encuentra en la figura 4.10

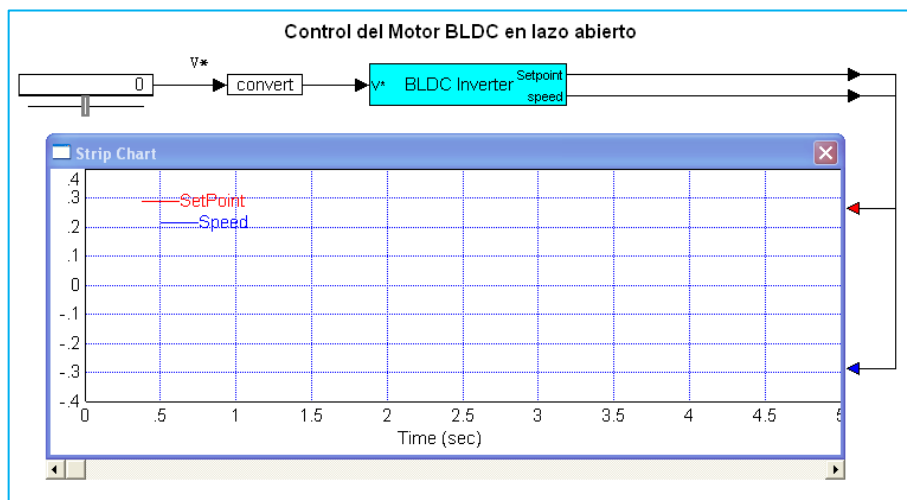


Figura 4-10 Diagrama de bloques “2BLDCOpenU.vsm”

El comando de velocidad está dado por el slider, mientras que el bloque compuesto “BLDC inverter”, que controla el motor en lazo abierto, puede ser separado en tres partes: La primera parte es un bloque DAC, que es usado para obtener cualquier señal analógica interna, en la parte frontal del DMD2000. La segunda parte la forman los bloques “Current Sensor” y “Current Sector” y la última parte está compuesta por los bloques “Position & Speed Sensor” y “BLDC Driver”. Dichos bloques se muestran en la figura 4.11.

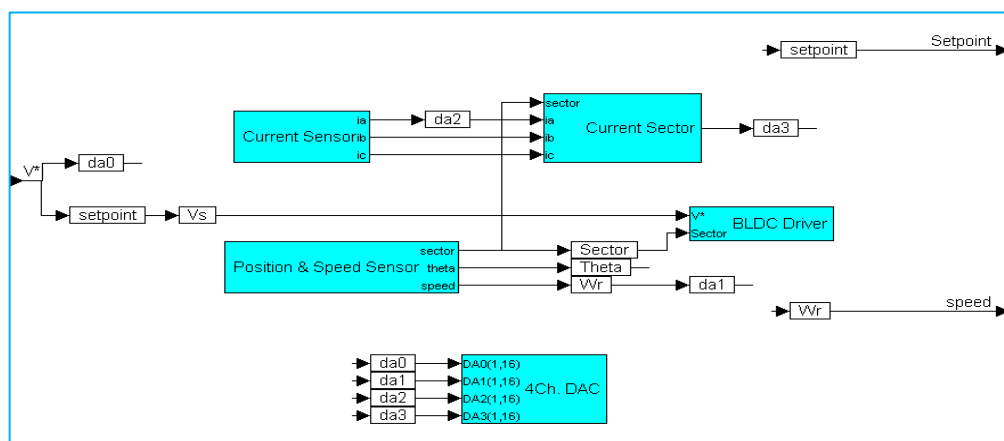


Figura 4-11 Diagramas internos del Bloque compuesto "BLDC Inverter".

En la Figura 4.12 veremos primero el bloque compuesto "Position & Speed Sensor".

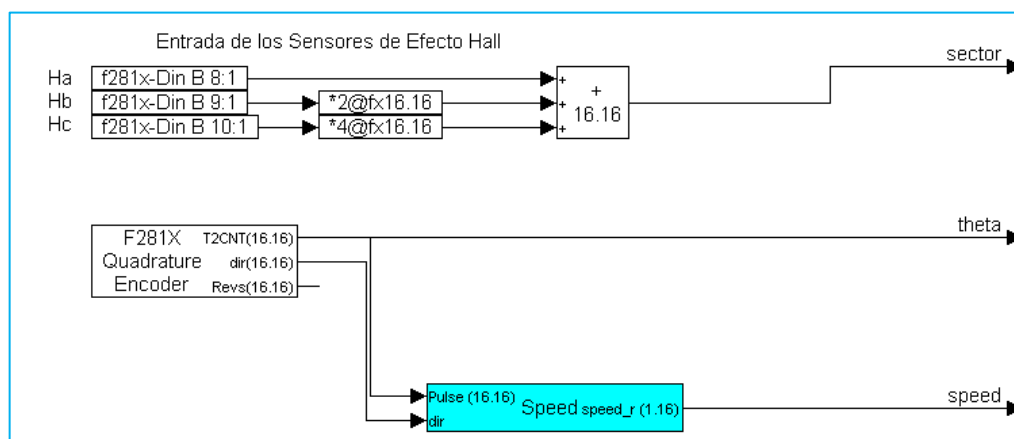


Figura 4-12 Bloque compuesto "Position & Speed Sensor"

Existen tres salidas en este bloque compuesto: $sector$, $theta$ y $speed$. Usaremos las entradas digitales del DSP nombradas como IOPB8, IOPB9 e IOPB10, para leer el estado de los sensores de efecto Hall, dos de estas señales serán multiplicadas por una

ganancia y luego sumadas para obtener la variable *sector* que consta desde el estado 0 hasta el 7.

En este bloque “Position & Speed Sensor” también tenemos el bloque “Quadrature encoder”, que captura las señales del encoder de cuadratura acoplado al motor BLDC, de aquí tomamos dos salidas: la “T2CNT” y “dir”. Donde “T2CNT” es un contador que cuenta cada flanco, ascendente y descendente, de las señales del encoder A y B este conteo es encerrado por la señal Z del encoder, en nuestro caso al ser un encoder de **1000 Pulsos/Rev** al girar **360°** el rotor, este bloque contará **4000** pulsos. Mientras que la salida “dir” determina la dirección de rotación +1: CW, -1: CCW.

En el bloque compuesto “Speed”, figura 4.12, tenemos el cálculo de la velocidad a partir de la posición. Se calculará el **$d\theta/dt$** para hallar la velocidad con los siguientes bloques.

De **Blocks**→**Fixed Point** añadir: el bloque **unitDelay**, el bloque **SampleHold** y el bloque **Sum**. Conectar y crear un bloque compuesto, como se muestra en la figura 4.13.

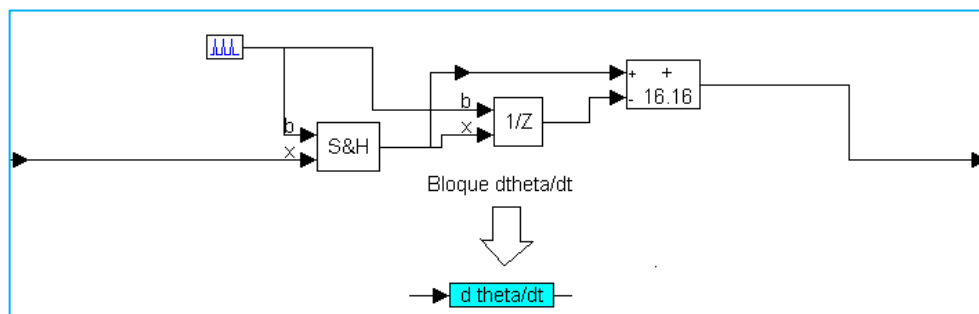


Figura 4-13 Bloque compuesto $d\theta/dt$.

Al bloque $d\theta/dt$ le ingresa una señal diente de sierra que toma valores entre 0 y 4000, por lo tanto la salida del bloque generará una señal constante proporcional a la pendiente de la diente de sierra, que representa la velocidad, sin embargo cada vez que se resetee dicha señal del contador, la derivada será muy grande con un valor negativo, figura 4.14, por lo que añadiremos unos bloques para minimizar el efecto de este cambio abrupto. Esto se muestra en la figura 4.14.

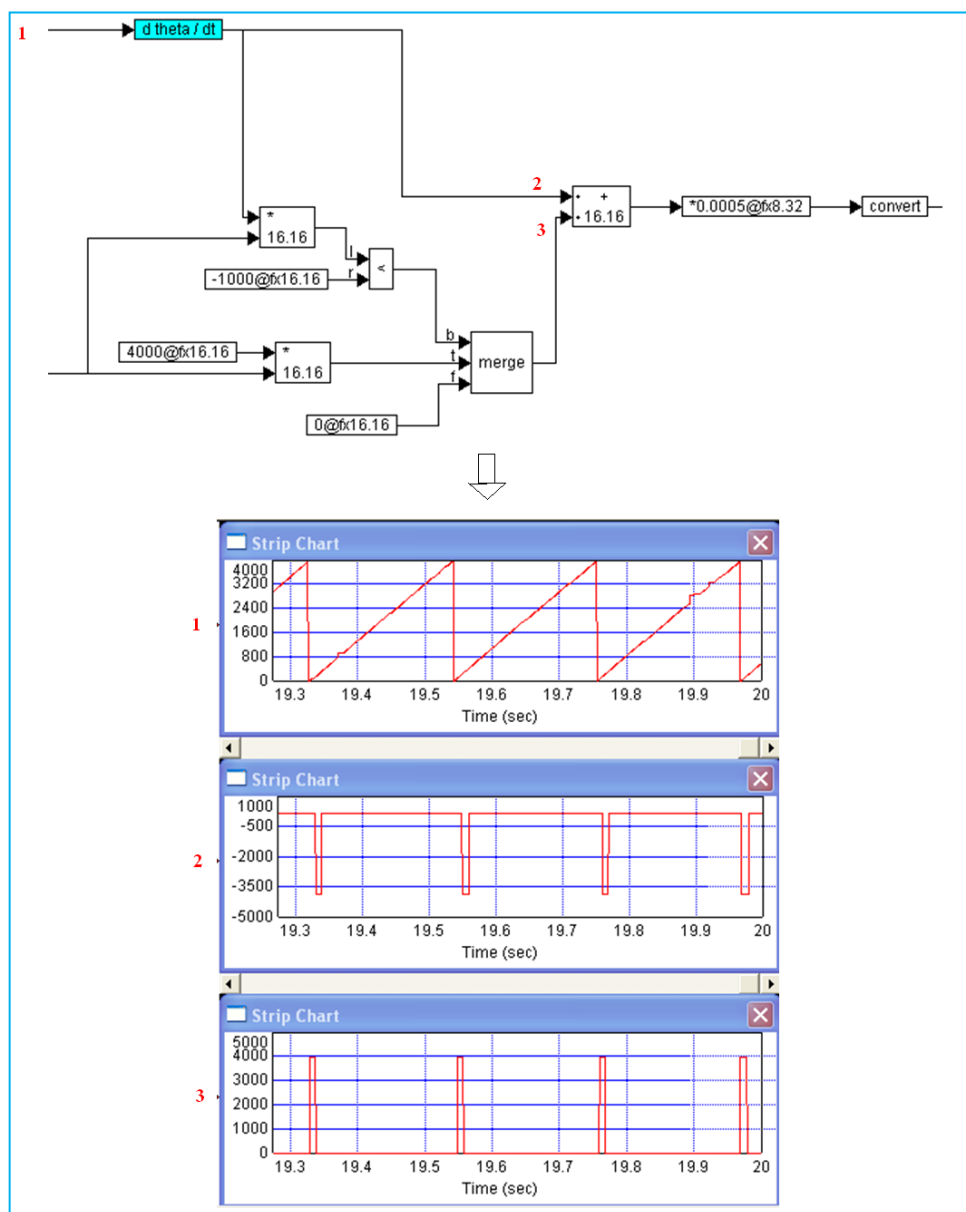


Figura 4-14 Proceso de cálculo de velocidad.

En el bloque compuesto “current sensor” se leen las corrientes I_a , I_b e I_c además se les resta 1.5 ya que estas señales tienen este offset agregado por hardware (el rango de voltaje admisible para una entrada del DSP es 0 a 3.3V [26])

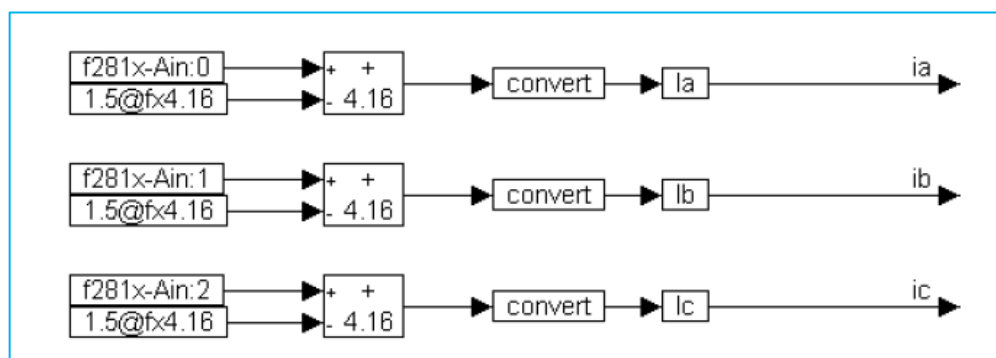


Figura 4-15 Bloque compuesto “Current Sensor”

En el bloque compuesto “Current Sector”, se usa la variable “Sector” que viene del bloque compuesto “Position & Speed Sensor” para seleccionar la corriente correcta.

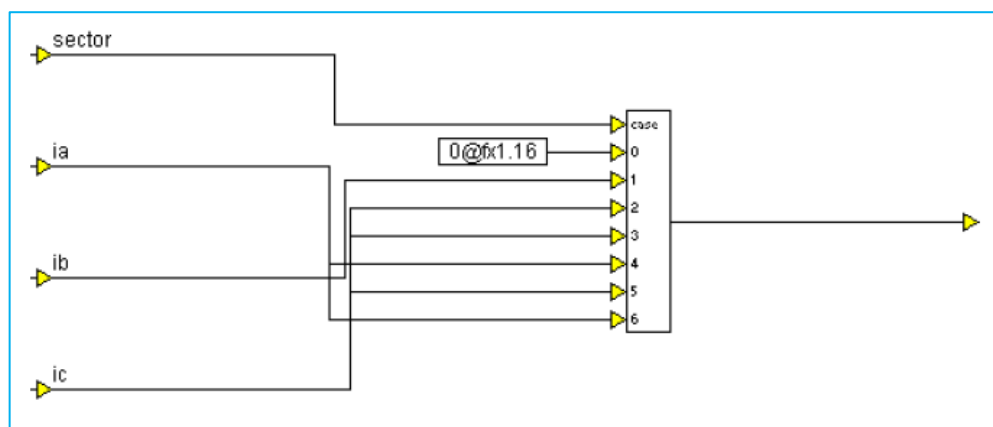


Figura 4-16 Bloque compuesto “Current Sector”

En el bloque compuesto “BLDC Driver” el ciclo de trabajo del PWM es controlado por V^* y el Pin “action” del bloque PWM es controlado por el bloque compuesto “PWM Action Control”.

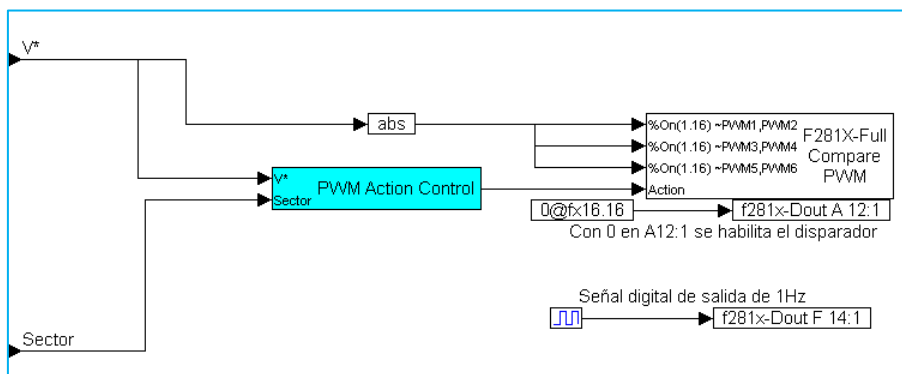


Figura 4-17 Bloque compuesto “BLDC Driver”.

El bloque compuesto “PWM Action Control” se muestra en la figura 4.18, donde la secuencia del sector cambiará en función del signo de V^* : Si V^* es positivo, se usa la secuencia de sector por defecto, pero si V^* es negativo la variable sector se la pasará por un XOR junto con el valor 7, lo cual hará cambiar la secuencia de disparo. La acción del PWM para cada sector se muestra en la tabla 4.2.

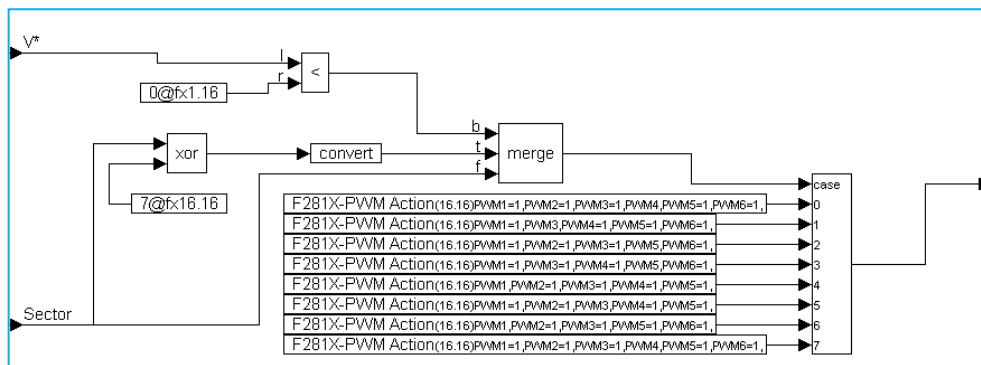


Figura 4-18 Bloque compuesto “PWM Action Control”

Tabla 4.2 Acción del PWM para cada sector

Sector	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6
0	Force High	Force High	Force High	Force High	Force High	Force High
1	Force High	Force Low	Active High	Force High	Force High	Force High
2	Force High	Force High	Force High	Force Low	Active High	Force High
3	Force High	Force Low	Force High	Force High	Active High	Force High
4	Active High	Force High	Force High	Force High	Force High	Force Low
5	Force High	Force High	Active High	Force High	Force High	Force Low
6	Active High	Force High	Force High	Force Low	Force High	Force High
7	Force High	Force High	Force High	Force High	Force High	Force High

En la figura 4.19 vemos como se relacionan los modos descritos en el capítulo 2 con los sectores necesarios para la conmutación de los IGBTs

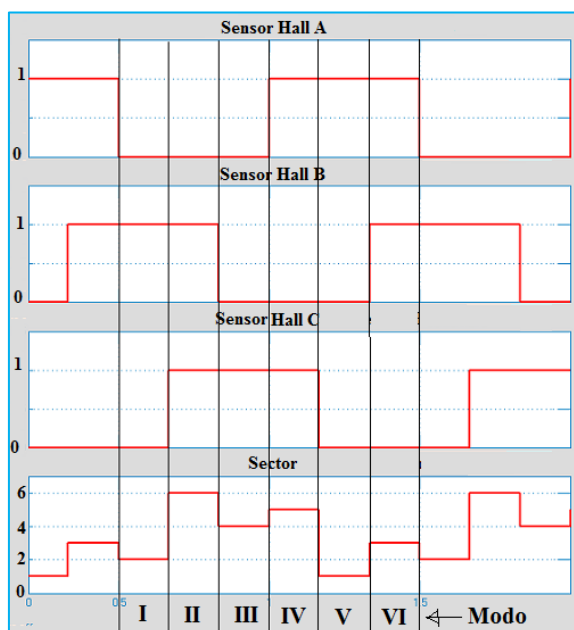


Figura 4-19 Relación entre las variables Sector y Modo.

El procedimiento para compilar y cargar el programa en la DSP se describe con más detalle en [25], entre los pasos a seguir tenemos:

Seleccionar el bloque compuesto “**BLDC Inverter**”, el bloque cambiará de color (rojo) como se muestra en la figura 4.20.

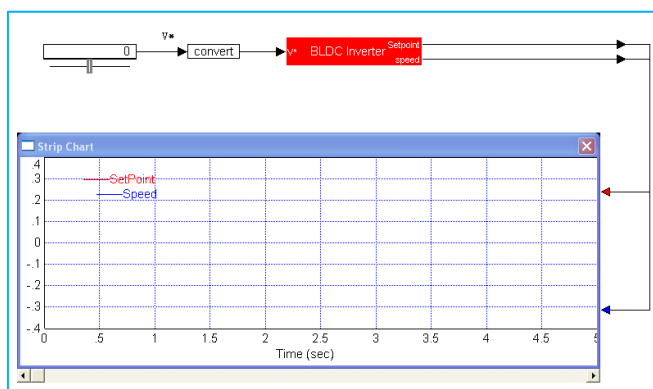


Figura 4-20 Selección del Bloque “BLDC Inverter”.

De la barra de herramientas seleccionar **Tools**→**Code Gen...**, entonces aparecerá el cuadro de diálogo de las propiedades de generación de Código (**Code Generation Properties**). Activar la opción **Include VisSim Communication Interface**, como se muestra en la figura 4.21.

La carpeta en la cual se guardará el archivo **.c** debe ser:
C:\VisSim70\dsp, Target : F281X

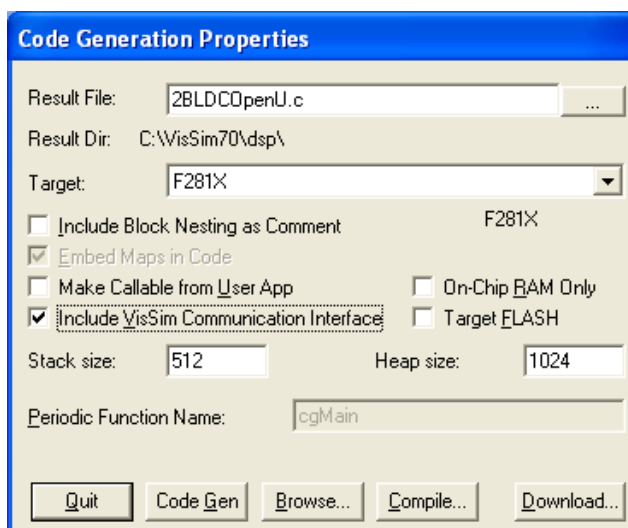
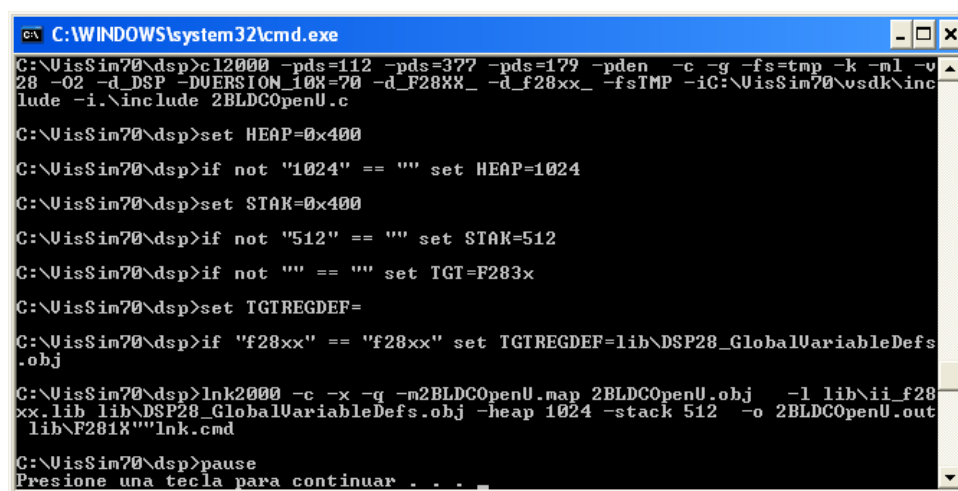


Figura 4-21 Cuadro de dialogo Code Generation Properties.

Dar clic en el botón **Compile** para generar el archivo **.c** y el archivo **.out**. Inmediatamente se despliega la ventana de comandos MS/DOS, como se muestra en la Figura 4.22, donde se realiza la generación de código, compilación y enlace.

Una vez finalizado el proceso y no existen mensajes de error o advertencias (**No Errors, No Warnings**), se debe presionar cualquier tecla para cerrar la ventana de comando y luego dar clic en el botón **Quit**. Si se tiene algún problema en la compilación, probablemente falte instalar **VisSimEcdWeb70.exe**, el cual es descargado de la página WEB de **VisSim**.



```

C:\WINDOWS\system32\cmd.exe
C:\VisSim70\dsp>c 12000 -pds=112 -pds=377 -pds=179 -pden -c -g -fs=tmp -k -ml -u
28 -02 -d_DSP -DUERSION_10X=70 -d_F28XX_ -d_f28xx_ -fsTMP -iC:\VisSim70\vsdk\inc
lude -i.\include 2BLDCOpenU.c
C:\VisSim70\dsp>set HEAP=0x400
C:\VisSim70\dsp>if not "1024" == "" set HEAP=1024
C:\VisSim70\dsp>set STACK=0x400
C:\VisSim70\dsp>if not "512" == "" set STACK=512
C:\VisSim70\dsp>if not "" == "" set TGT=F281X
C:\VisSim70\dsp>set TGTREGDEF=
C:\VisSim70\dsp>if "f28xx" == "f28xx" set TGTREGDEF=lib\DSP28_GlobalVariableDefs
.obj
C:\VisSim70\dsp>lnk2000 -c -x -g -m2BLDCOpenU.map 2BLDCOpenU.obj -l lib\ii_f28
xx.lib lib\DSP28_GlobalVariableDefs.obj -heap 1024 -stack 512 -o 2BLDCOpenU.out
lib\F281X""lnk.cmd
C:\VisSim70\dsp>pause
Presione una tecla para continuar . . .

```

Figura 4-22 Mensajes de compilación y enlace en la ventana de comandos de Windows.

Para poder descargar el código a la DSP y simular seguimos los siguientes pasos.

Guarde el archivo con un nuevo nombre “2BLDCOpenUD”

Seleccionar el bloque compuesto “**BLDC Inverter**” y eliminarlo.

Seleccionar **VisSim/DSP**→**F281X**→**DSPinterface**, pegarlo sobre la hoja de trabajo, dar clic derecho sobre el bloque, se desplegará la ventana **F281X DSP Interface Properties**, (figura 4.23).

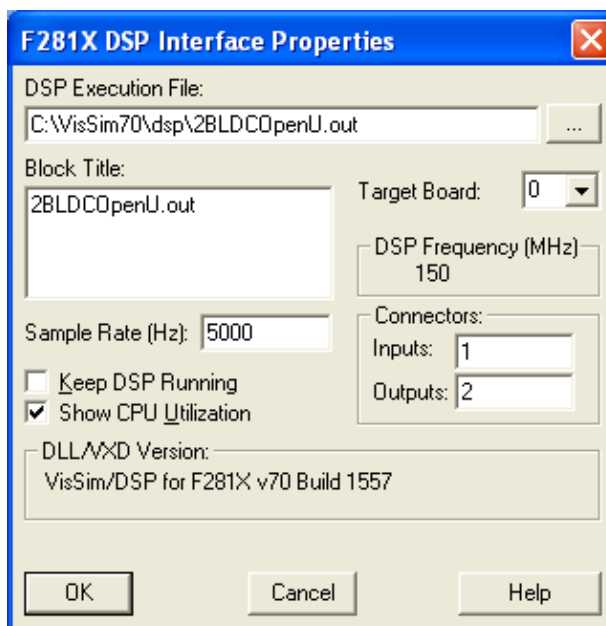


Figura 4-23 Ventana F281X DSP Interface Properties.

Observar que en el campo *DSP Execution File* esté la dirección correcta junto con el nombre del archivo que fue compilado anteriormente. Además, la frecuencia de muestreo del DSP será de 5 KHz, el número de entradas es uno (Referencia) y el de salidas es dos (Referencia y Velocidad). Dar clic en OK y cablear como se muestra en la figura 4.24.

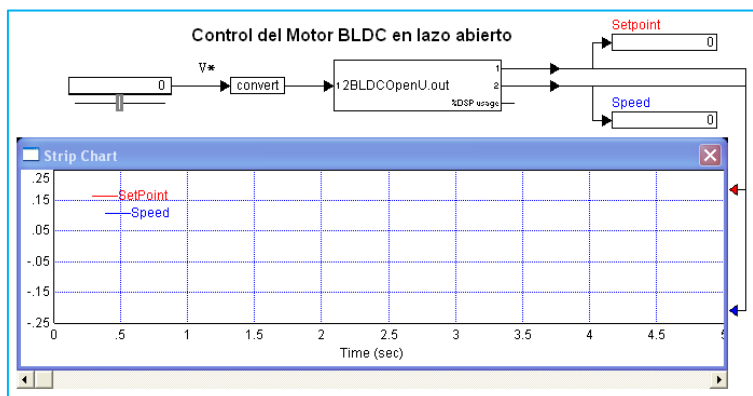


Figura 4-24 Reemplazo del bloque compuesto con el bloque DSPinterface.

Ahora cambiaremos las opciones de simulación, para esto seleccionar **Simulate**→**Simulation Properties**. Se abrirá el cuadro de dialogo de las propiedades de simulación, luego cambie el **Time Step** a 0.001 (1 KHz), **Observar que esta frecuencia es menor que la frecuencia de muestreo del DSP (5KHz)**, dar clic en la opción **Run in Real Time** y luego en Aceptar (figura 4.25).

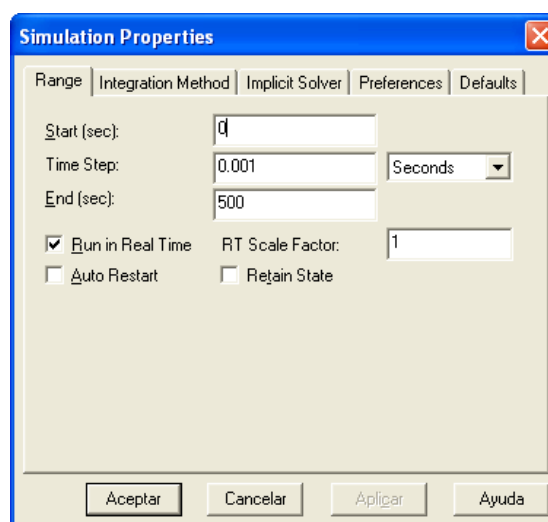



Figura 4-25 Ventana Simulation Properties.

Verifique que el valor del slider de setpoint esté en el valor de cero. Guarde el archivo y presione RUN  para empezar la simulación, en ese momento el bloque **DSPinterface** descargará el archivo compilado **.out** al DSP del equipo DMD2000. Se procederá a modificar el valor del slider poco a poco y se observará como la velocidad del motor va incrementando, figura 4.26.

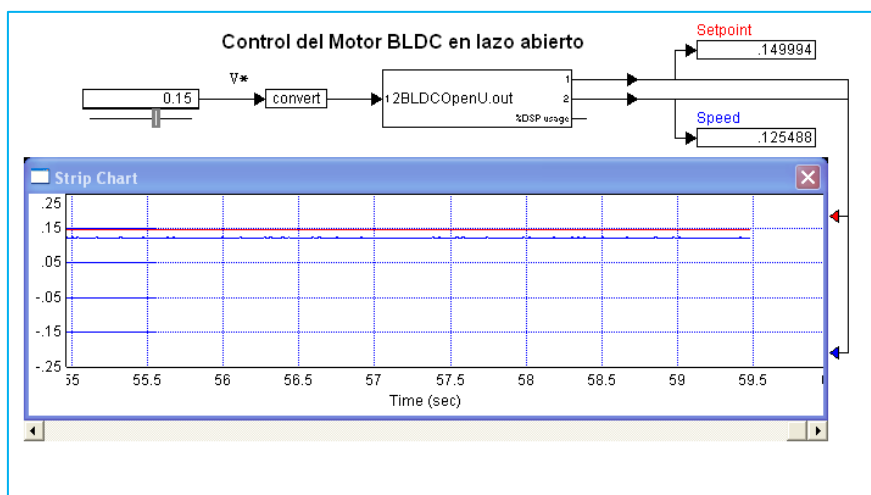


Figura 4-26 Referencia y velocidad actual en lazo abierto.

4.5.2. Control de velocidad en lazo cerrado con lógica difusa.

El control de velocidad de un motor BLDC con un controlador difuso se muestra en la figura 4.27, Esto se desarrollará en el software VisSim en el cual veremos la respuesta del sistema en tiempo real.

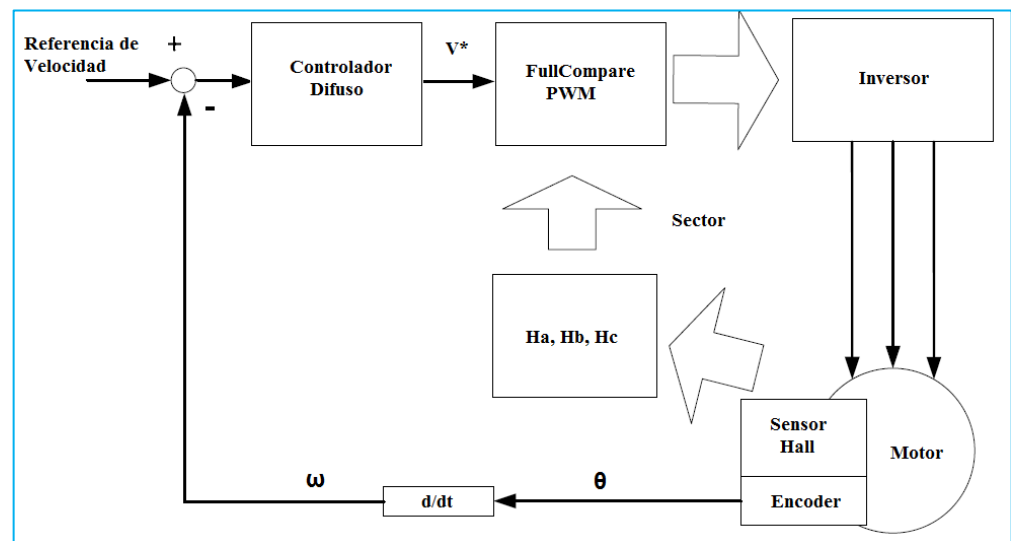


Figura 4-27 Sistema de control difuso de velocidad para un motor BLDC

En nuestro sistema tendremos como entrada la señal de referencia, y como salida el setpoint, la velocidad real y la señal U del controlador difuso. En la figura 4.28 se muestra el esquema principal de nuestro controlador.

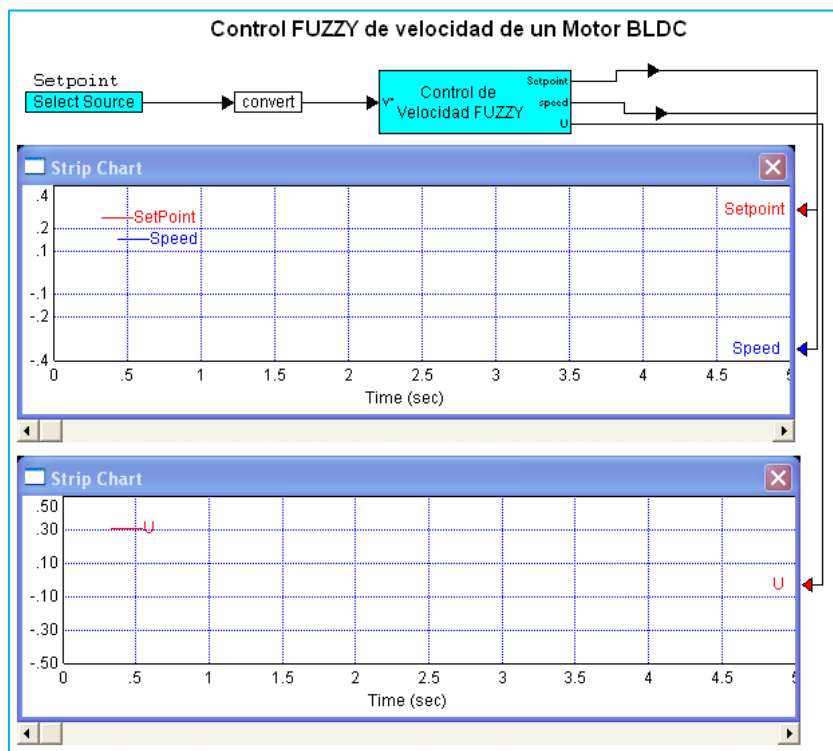


Figura 4-28 Diagrama de bloques del controlador Fuzzy para un motor BLDC.

Los bloques internos del bloque compuesto “Control de velocidad Fuzzy” se muestran en la figura 4.29. Donde los bloques “4Ch.DAC”, “Current Sensor”, “Current Sector”, “Position & Speed Sensor” y “BLDC Driver” son similares a los descritos en el punto 4.5.1 de este capítulo.

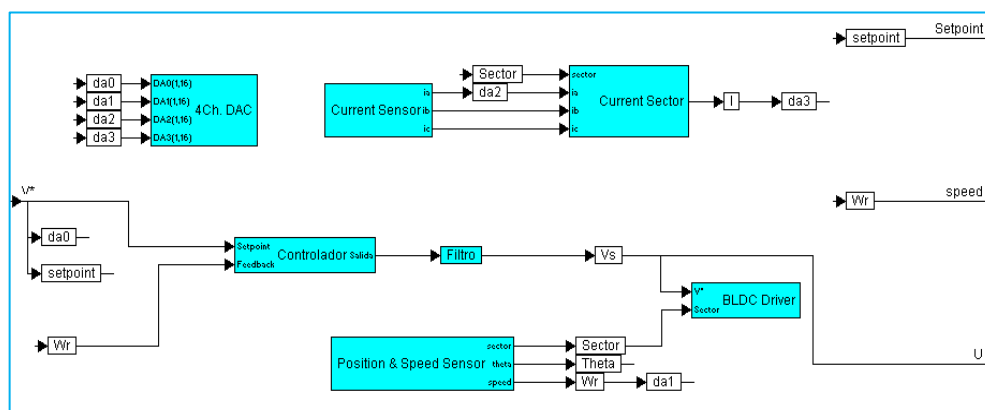


Figura 4-29 Diagrama de bloques del controlador Fuzzy para un motor BLDC.

En el interior del bloque “Controlador” se encuentra el controlador Difuso, figura 4.30. El cual tiene como entrada el error y la derivada del error y como salida la variación del torque, Estas señales se ven modificadas por las ganancias **K_{iF}=0.25**, **K_{pF}=0.002** y **K=0.019** propias de este controlador, las cuales fueron obtenidas mediante prueba y error. La modificación de estos valores cambiará la respuesta del sistema.

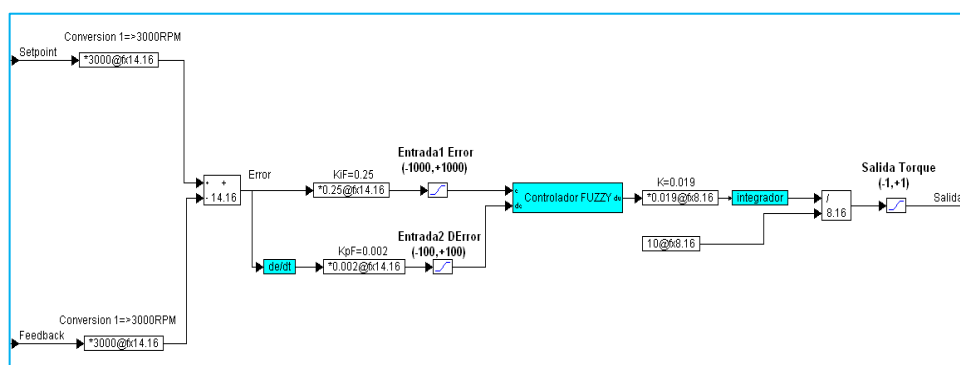


Figura 4-30 Diagrama interno del bloque “Controlador”.

Dentro del bloque “Controlador FUZZY” se encuentra el desarrollo del controlador difuso, Figura 4.31. Las partes que constituyen el controlador difuso son: fuzificación, inferencia y desfuzificación.

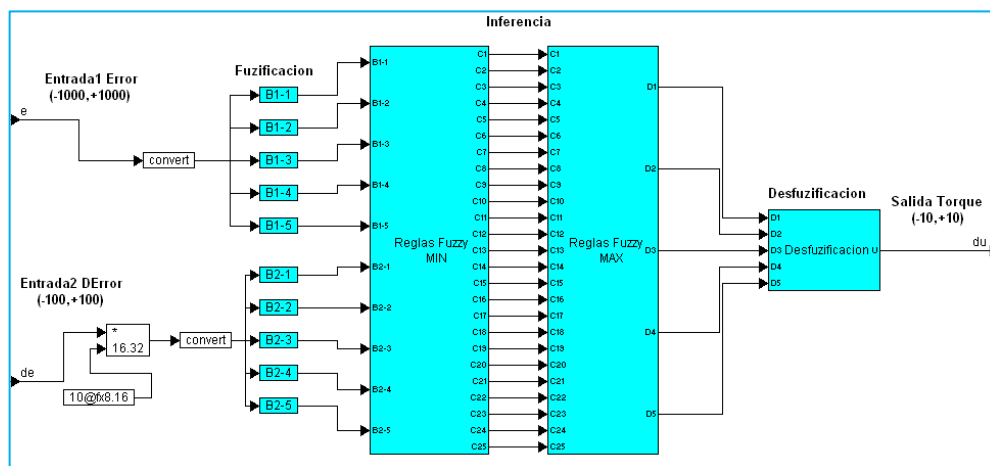


Figura 4-31 Diagrama interno del bloque “Controlador FUZZY”.

La Fuzificación es el proceso en el cual se determina el grado de pertenencia de una entrada respecto a los conjuntos difusos. Para este caso las entradas al controlador son: el ***error = referencia - velocidad_actual*** y la derivada del error, el cual se obtiene sustrayendo el valor del error previo menos el valor actual es decir: ***dError = Error - Error_Previo***. Se han seleccionado cinco conjuntos difusos, *Grande Negativo (GN)*; *Negativo (N)*; *Cero (C)*; *Positivo (P)*; *Grande Positivo (GP)*.

Las funciones de membresía, figura 4.32, son de forma triangular y el máximo valor escalado es 500 en vez de 1 el cual es usado

típicamente en los controladores difusos. Con esto la complejidad del cálculo se reduce, ya que la operación de fuzificación se reduce a sumas y restas.

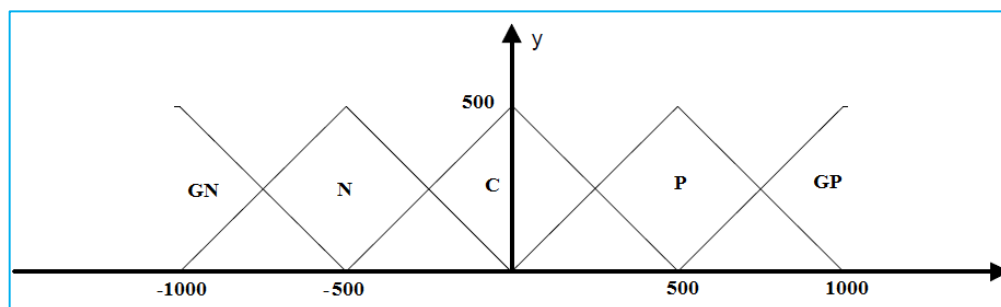


Figura 4-32 Funciones de membresía del controlador difuso.

El resultado de la fuzificación de cada entrada es un vector de cinco elementos ya que existen cinco funciones de membresía, este valor define el grado de membresía del valor de entrada respecto a cada conjunto difuso.

En nuestra implementación la fuzificación se realiza considerando cada una de las funciones de membresía, es así que se obtiene un código para cada función. Por ejemplo, para la función **GN**, figura 4.33, se tendrá el siguiente código:

```
IF(Valor<-500)
    Dato=500-(Valor+1000)
ELSE
    Dato=0
```

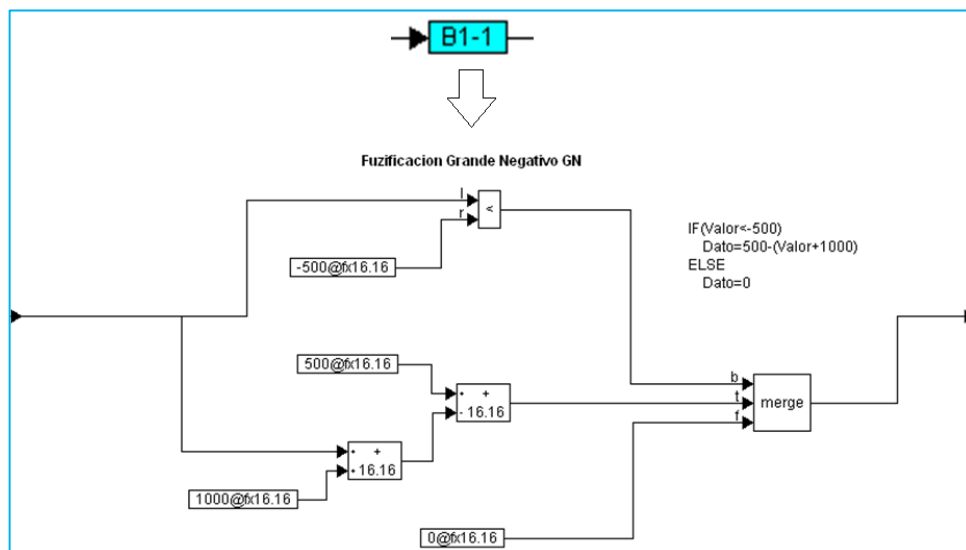


Figura 4-33 Fuzificación de la entrada 1, Función de membresía Grande Negativo (GN).

Para la función **N**, figura 4.34, se tendrá el siguiente código:

IF(Valor<0)

IF(Valor < -500)

Dato=Valor+1000

ELSE

Dato=500-(Valor+500)

ELSE

Dato=0

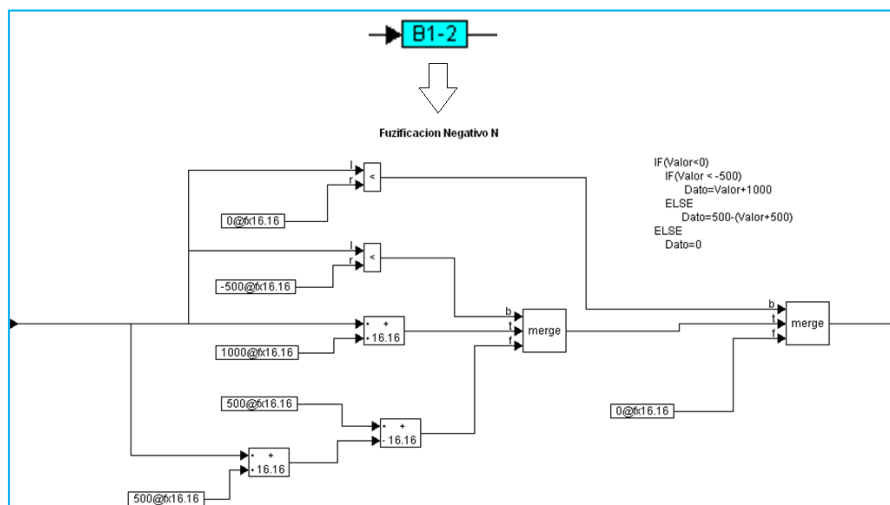


Figura 4-34 Fuzificación de la entrada 1, Función de membresía Negativo (N).

Esto se realiza con cada una de las funciones de Membresía.

La etapa de inferencia se desarrollará con el método de Mandani *min-max* para obtener una salida fuzzy como una consecuencia de la base de reglas. Este método es mostrado en la figura 4.35, de esta forma puede ser implementado fácilmente en el software VisSim. La parte izquierda muestra la implicación de las 25 condiciones de salida debido al antecedente de las reglas difusas aplicando el operador *min*. Cada condición de salida modela la consecuencia de una regla. Entonces para la regla:

$$R^1: \text{IF } x_1 \text{ is } B_1^1 \text{ and } x_2 \text{ is } B_2^1, \text{ THEN } U \text{ is } C^1 \quad (4.3)$$

Será:

$$C^1 = \min[B_1^1, B_2^1] \quad (4.4)$$

El operador **max** es usado para tomar en consideración los efectos de todas las reglas. Las 25 condiciones de salida son agrupadas dentro de cinco valores lingüísticos (D1 a D5) basado en las condiciones de las reglas. Esta operación es mostrada en el lado derecho de la figura 4.35.

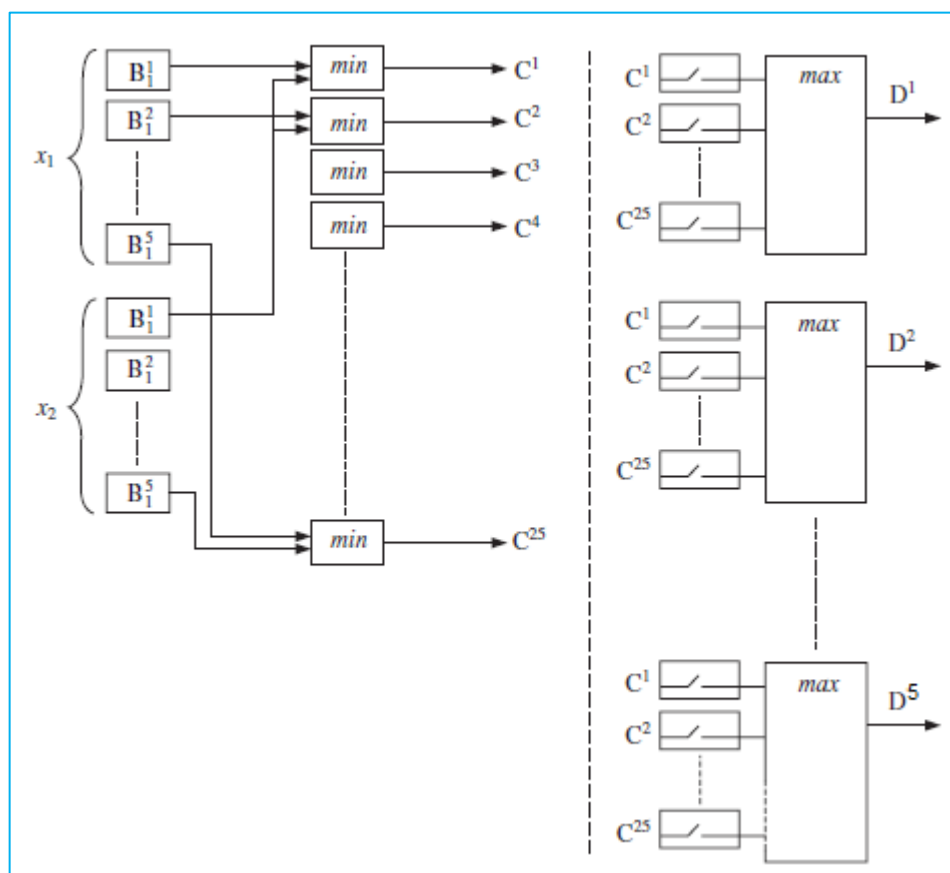


Figura 4-35 Modelo del algoritmo de inferencia.

Las salidas D1 a D5 buscarán el máximo entre las salidas obtenidas de C1 a C25 dependiendo de que regla sea disparada en ese momento, siguiendo la tabla de reglas definida en el diseño. Tabla 4.3.

Tabla 4.3 Tabla de reglas para la implementación del controlador difuso.

$e(X1)$ $\Delta e(X2)$	(B1-1) GN	(B1-2) N	(B1-3) C	(B1-4) P	(B1-5) GP
(B2-1) GN	(C1) Grande Negativo	(C6) Grande Negativo	(C11) Grande Negativo	(C16) Negativo	(C21) Cero
(B2-2) N	(C2) Grande Negativo	(C7) Grande Negativo	(C12) Negativo	(C17) Cero	(C22) Positivo
(B2-3) C	(C3) Grande Negativo	(C8) Negativo	(C13) Cero	(C18) Positivo	(C23) Grande Positivo
(B2-4) P	(C4) Negativo	(C9) Cero	(C14) Positivo	(C19) Grande Positivo	(C24) Grande Positivo
(B2-5) GP	(C5) Cero	(C10) Positivo	(C15) Grande Positivo	(C20) Grande Positivo	(C25) Grande Positivo

Por tanto las salidas D1 a D5 se pueden definir como sigue:

$$D1 = \max [C1, C2, C3, C6, C7, C11].$$

$$D2 = \max [C4, C8, C3, C12, C16].$$

$$D3 = \max [C5, C9, C13, C17, C21].$$

$$D4 = \max [C10, C14, C18, C22].$$

$$D5 = \max [C15, C19, C20, C23, C24, C25].$$

El diagrama interno del bloque Reglas Fuzzy MIN se muestra en la figura 4.36. Están agrupadas en cinco bloques para hacer más fácil el entendimiento de cada sección. Por ejemplo en el bloque MIN 1-5, figura 4.37, está relacionada la señal fuzificada de la entrada 1 (X1) respecto a la primera función de membresía (GN), esto es B1-1, con todas las señales obtenidas de la fuzificación de la entrada 2 (X2) respecto a todas las funciones de membresía. Dando como resultado 25 reglas que nacen de la inferencia del valor mínimo.

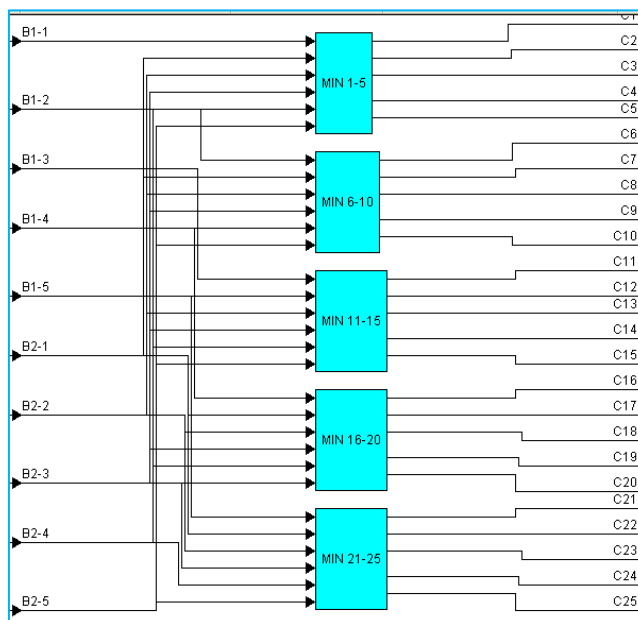


Figura 4-36 Diagrama interno del bloque “Reglas Fuzzy MIN”.

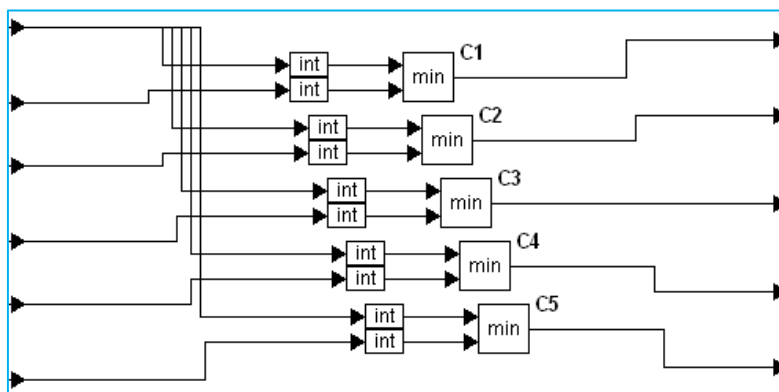


Figura 4-37 Diagrama interno del bloque "MIN 1 - 5".

El diagrama interno del bloque "Reglas Fuzzy MAX" se muestra en la figura 4.38. Que partiendo de la tabla 4.3 se obtiene el conjunto de reglas a las cuales se les buscará el máximo. Por ejemplo:
 $D3 = \text{MAX} [C5, C9, C13, C17, C21]$

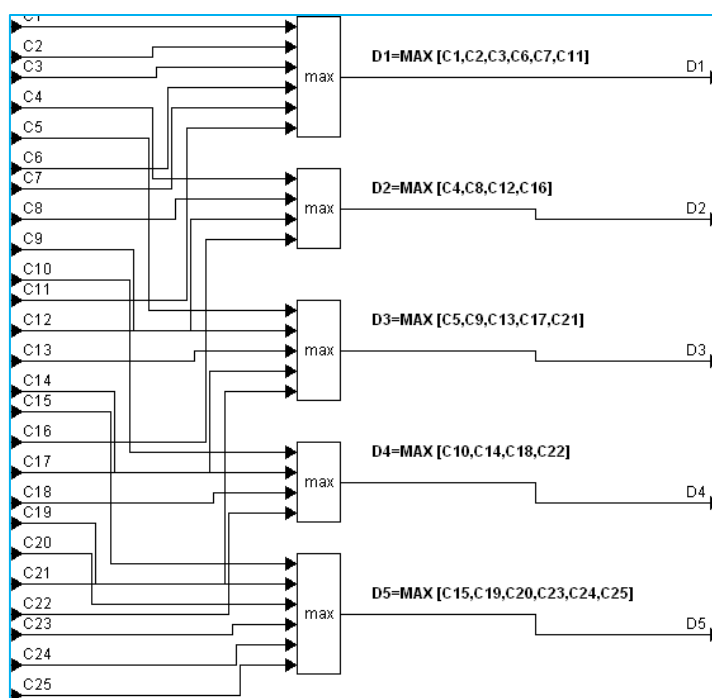


Figura 4-38 Diagrama interno del bloque "Reglas Fuzzy MAX".

Con los valores D1 hasta D5 obtenidos en el paso anterior podemos implementar un método de desfuzificación, en nuestro caso usaremos el método del centroide, este valor determinará el valor del PWM que se ingresará al driver del motor BLDC. La fórmula matemática usada para calcular el centro de gravedad es:

$$Defuz = \frac{\sum_1^5 Y[i]*Coef[i]}{\sum_1^5 Y[i]} \quad (4.5)$$

Donde $Y[i]$ es el i -ésimo miembro del vector de salida del bloque de inferencia, es decir D1 a D5; El vector $Coef[i]$ es el i -ésimo coeficiente que está relacionado con cada función de membresía de la salida. En la figura 4.39 se muestra una representación gráfica de las funciones de membresía que se usan en esta aplicación, donde los coeficientes son $Coef[i] = [-20, -10, 0, 10, 20]$.

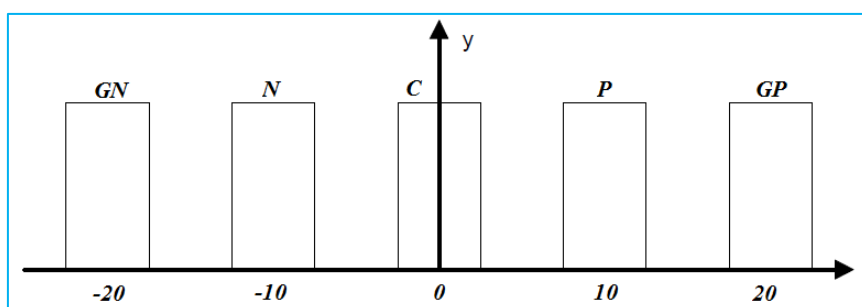


Figura 4-39 Funciones de Membresía de la salida.

El diagrama interno del bloque “Desfuzificación” se muestra en la figura 4.40 donde está implementada la fórmula 4.5.

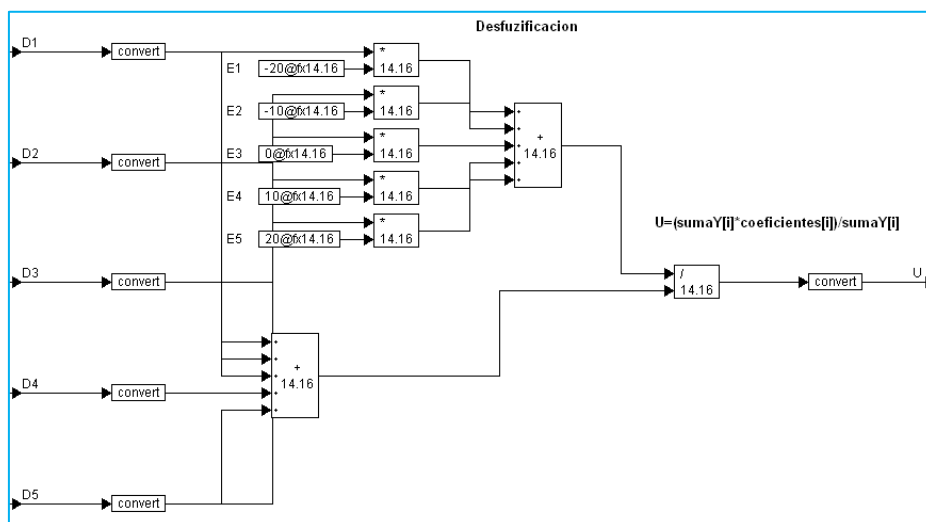


Figura 4-40 Diagrama interno del bloque “Desfuzificación”.

Luego de tener la salida para nuestro sistema necesitamos de un integrador que sume la contribución de la última variación del controlador. Lo cual lo realizamos con los bloques de la figura 4.41

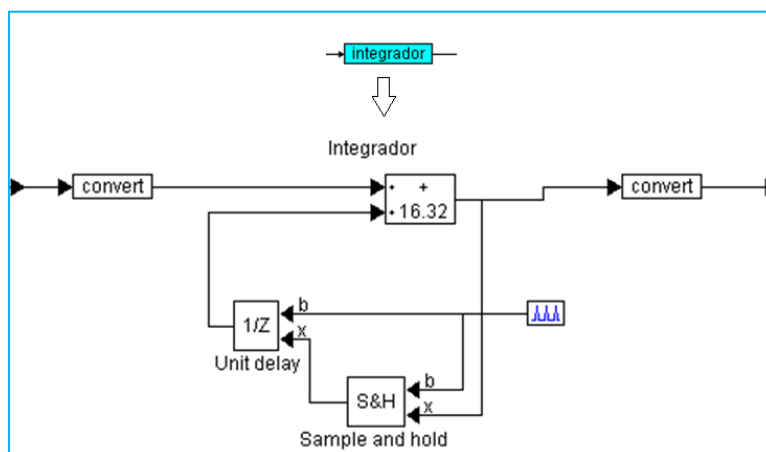


Figura 4-41 Diagrama interno del bloque “Integrador”.

Finalmente para mejorar la respuesta de la salida del controlador colocamos un filtro pasa bajos calculado a unos 100Hz (628rad/seg), este bloque lo podemos obtener de **Blocks**→**Fixed Point**→**TransferFunction (filter)**, pegarlo a la salida del controlador y parametrizarlo como muestra la figura 4.42.

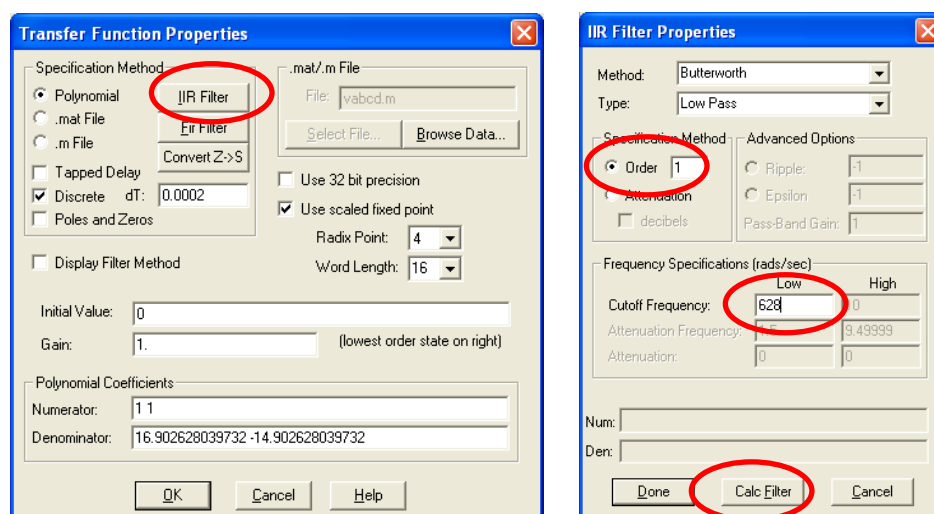
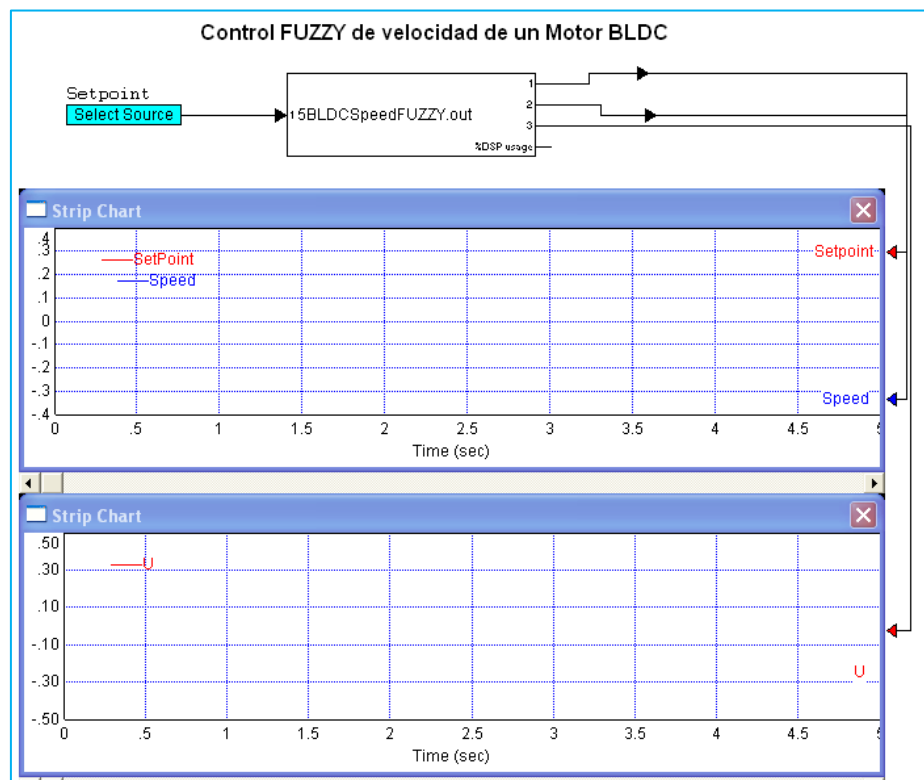


Figura 4-42 Parámetros del bloque “TransferFunction (filter)”.

Guardar el archivo actual con el nombre “5BLDCSpeedFUZZY.vsm”.

Luego damos a *guardar como*, para guardar el mismo archivo con otro nombre “5BLDCSpeedFUZZYD.vsm” donde va a estar el bloque “DSP Interface” para descargar el programa en la DSP, realizar el mismo procedimiento del numeral 4.5.1. El nuevo diagrama de bloques se muestra en la figura 4.43



**Figura 4-43 Diagrama de bloques del archivo
“5BLDCSpeedFUZZYD.vsm”**

Setear el bloque “Select Source” como se muestra en la figura 4.44 para una entrada de velocidad en forma cuadrada con frecuencia 0.3 y amplitud de 0.2. En el bloque compuesto “Select Source” se puede seleccionar una señal Sinusoidal, triangular, cuadrada o una constante, junto con su amplitud y frecuencia. Este Bloque es desarrollado en [25].

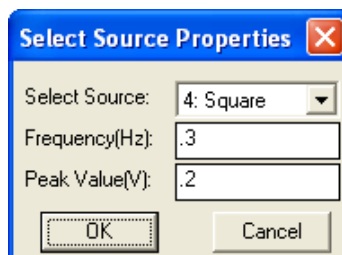


Figura 4-44 Parámetros del bloque Select Source.

Finalmente el comportamiento del sistema es como se muestra en la figura 4.45. En el primer Plot en color Rojo la referencia de velocidad y en color azul la velocidad actual. Mientras que en el segundo Plot en rojo se muestra la señal U del controlador.

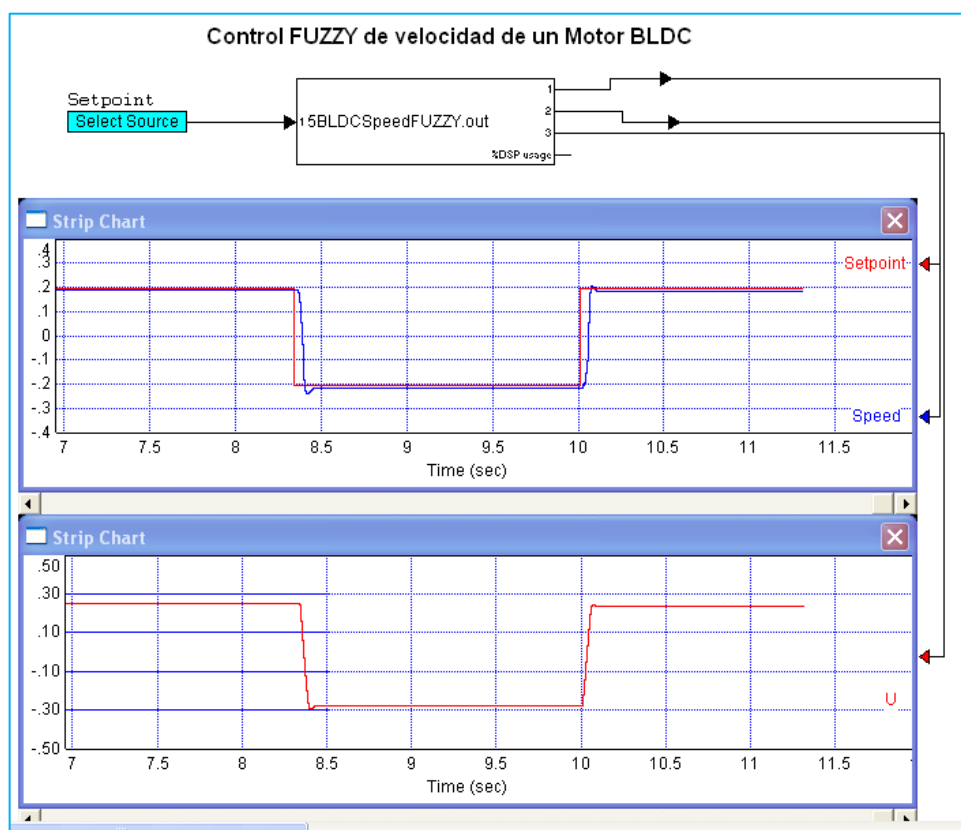


Figura 4-45 Respuesta del Controlador PID ante una entrada de velocidad.

4.5.3. Control de velocidad en lazo cerrado con controlador PID.

Se desarrollará el controlador PID para el control de velocidad de un motor BLDC, mediante el software VisSim, donde el usuario puede modificar en tiempo real la referencia, las constantes del PID y observar la respuesta de velocidad. En la figura 4.46 se muestra un diagrama de bloques del sistema.

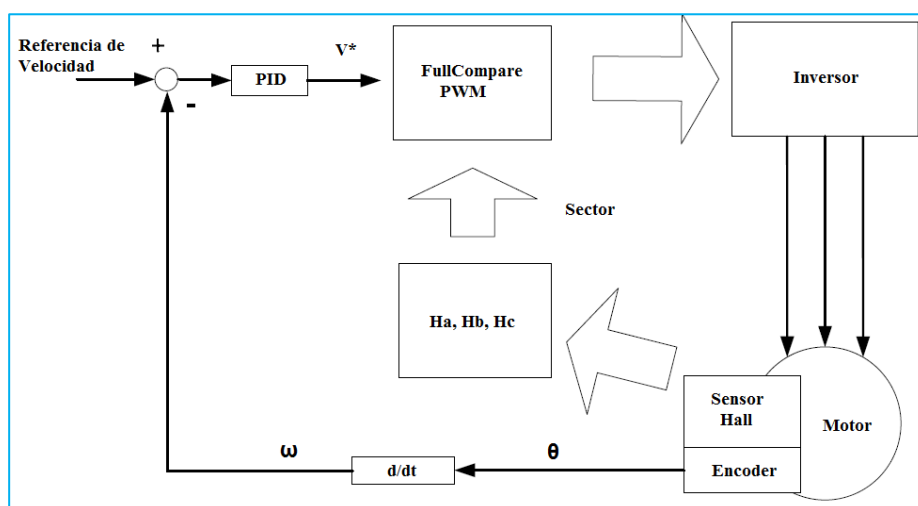


Figura 4-46 Sistema de control PI de velocidad para un motor BLDC.

Los bloques en VisSim que se van a utilizar se encuentran en la figura 4.47.

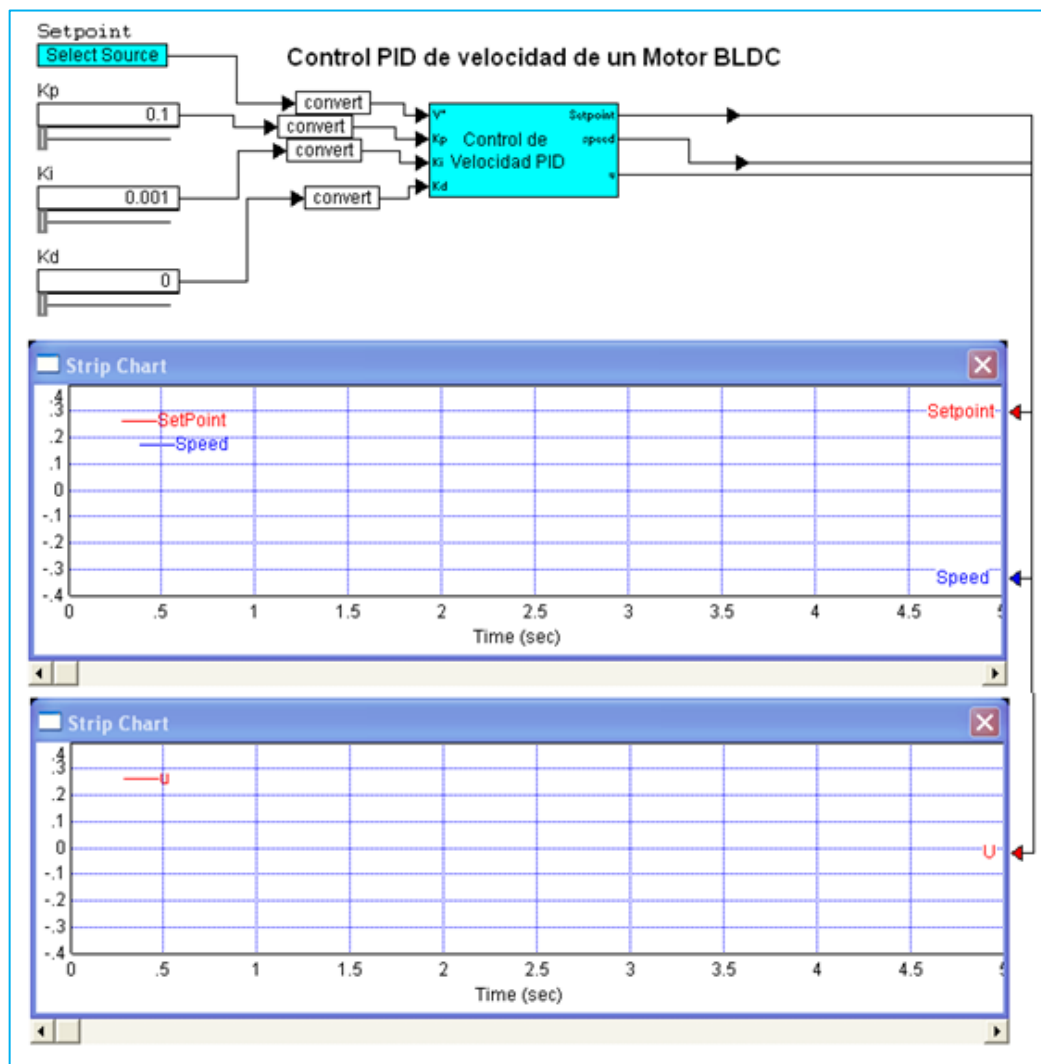


Figura 4-47 Diagrama de bloques del controlador PID para un motor BLDC.

Podemos observar: el bloque “Select Source” que dará el valor de referencia, los tres slider a la izquierda que modifican las ganancias K_p , K_i y K_d del controlador PID, el bloque “Control de velocidad PID” y los 2 scopes “Strip Chart” donde se graficarán las

señales de referencia, la señal de velocidad y la señal u del controlador.

El diagrama interno del bloque “Control de velocidad PID” se muestra en la figura 4.48

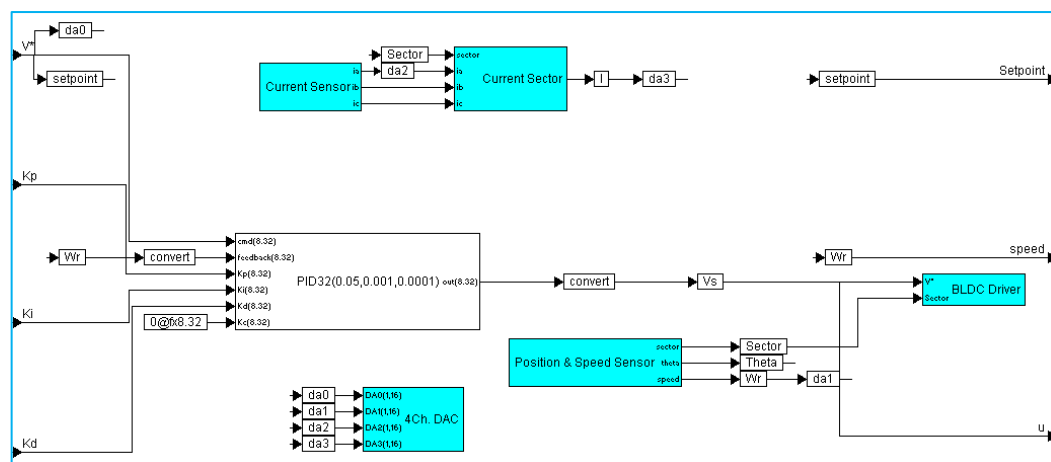


Figura 4-48 Bloque compuesto “Control de velocidad PID”

Donde los bloques “Current Sensor”, “Current Sector”, “Position & Speed Sensor” y “BLDC Driver” son similares a los descritos en el punto 4.5.1 de este capítulo.

Para obtener el bloque PID, vamos a VisSim/DSP-> TI 32-bit Digital Motor Control blocks y seleccionamos PID Regulator, damos clic derecho sobre el bloque y lo completamos como se muestra en la figura 4.49.

Parameter	Value	Range
Kp:	0.05	-128 .. 127.999
Ki:	0.001	-128 .. 127.999
Kd:	0	-128 .. 127.999
Kc:	0	-128 .. 127.999
Upper Limit:	0.4	-128 .. 127.999
Lower Limit:	-0.4	-128 .. 127.999

Use External Gains

OK Cancel Help

Figura 4-49 Parámetros del bloque “PID Regulator”.

Guardar el archivo actual con el nombre “4BLDCSpeedPID.vsm”.

Luego damos a *guardar como*, para guardar el mismo archivo con otro nombre “4BLDCSpeedPIDD.vsm” donde va a estar el bloque “DSP Interface” para descargar el programa en la DSP, realizar el mismo procedimiento del numeral 4.5.1. El nuevo diagrama de bloques se muestra en la figura 4.50, donde se han colocado las ganancias **Kp=0.16**, **Ki=0.036** y **Kd=0**, las que proporcionan una respuesta satisfactoria del sistema.

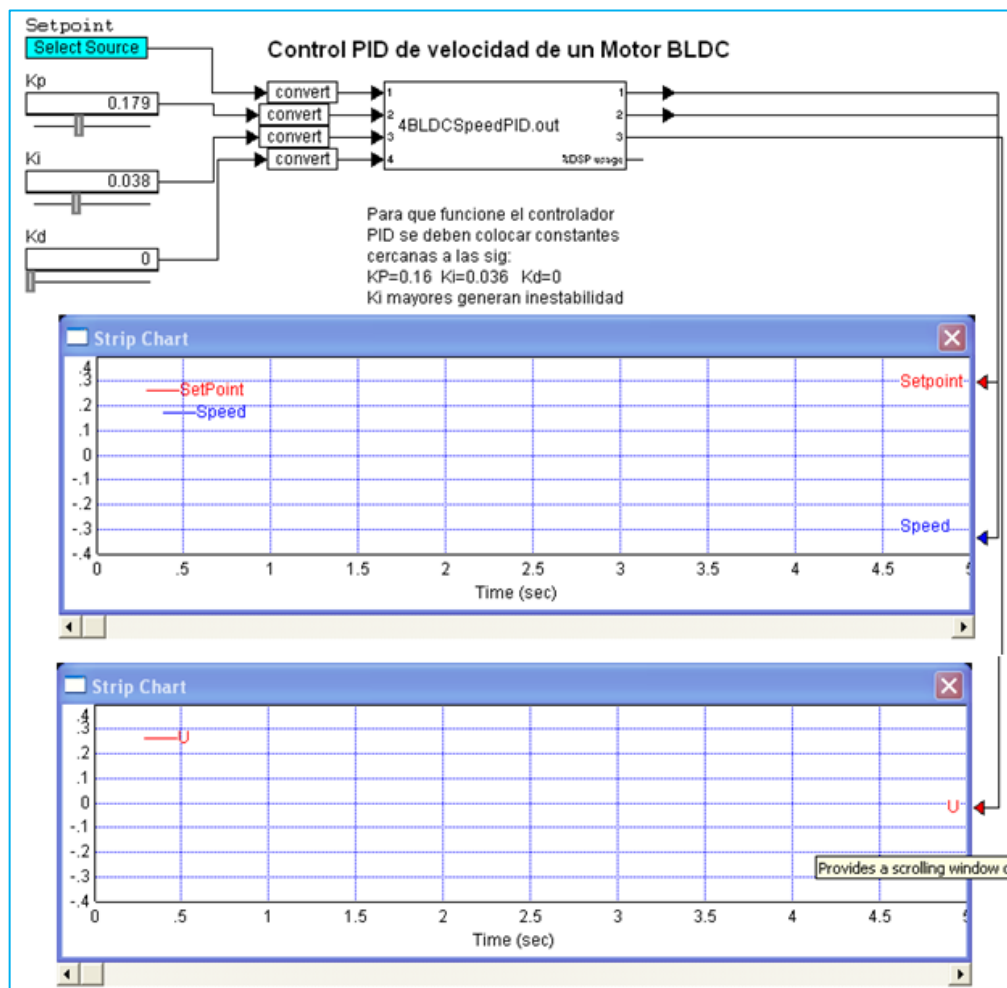


Figura 4-50 Diagrama de bloques del archivo "4BLDCSpeedPID.vsm"

Setear el bloque "Select Source" como se muestra en la figura 4.51 para una entrada de velocidad en forma cuadrada con frecuencia 0.3 y amplitud de 0.2. En el bloque compuesto "Select Source" se puede seleccionar una señal Sinusoidal, triangular, cuadrada o una constante, junto con su amplitud y frecuencia. Este Bloque es desarrollado en [25].

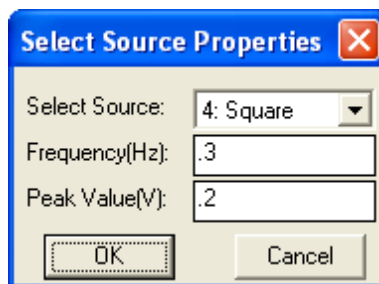


Figura 4-51 Parámetros del bloque Select Source.

Finalmente el comportamiento del sistema es como se muestra en la figura 4.52. En el primer Plot en color Rojo la referencia de velocidad y en color azul la velocidad actual. Mientras que en el segundo Plot en rojo se muestra la señal U del controlador.

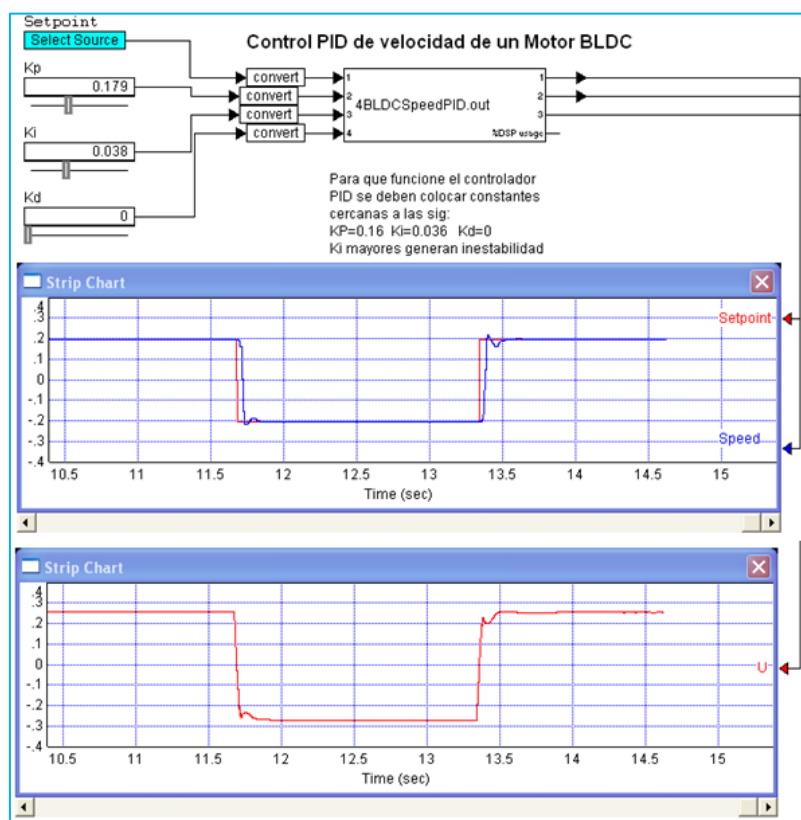


Figura 4-52 Respuesta del Controlador PID ante una entrada de velocidad.

CAPÍTULO 5

5. SIMULACIONES Y PRUEBAS EXPERIMENTALES.

5.1. Introducción.

En el capítulo 5, se presentan los resultados de las simulaciones junto con los resultados experimentales, comparando la respuesta del sistema entre el control PID y el control difuso.

5.2. Resultados de la Simulación.

Para el *controlador PI* se tiene el diagrama de bloques de la figura 5.1.

Ajustando el controlador PI con una ganancia proporcional **$K_p=3.3$** y una ganancia Integral **$K_i=300$** . Para una variación de velocidad de 0 a 250RPM en $t=0$ seg y de 250 a 500RPM en $t=1$ seg, manteniendo el torque de carga constante e igual a 1 N-m. La respuesta del controlador

es la que se muestra en la figura 5.2 donde se observan tres gráficas. Primero la corriente del motor, segundo la respuesta de velocidad del motor y tercero la variación en el torque del motor.

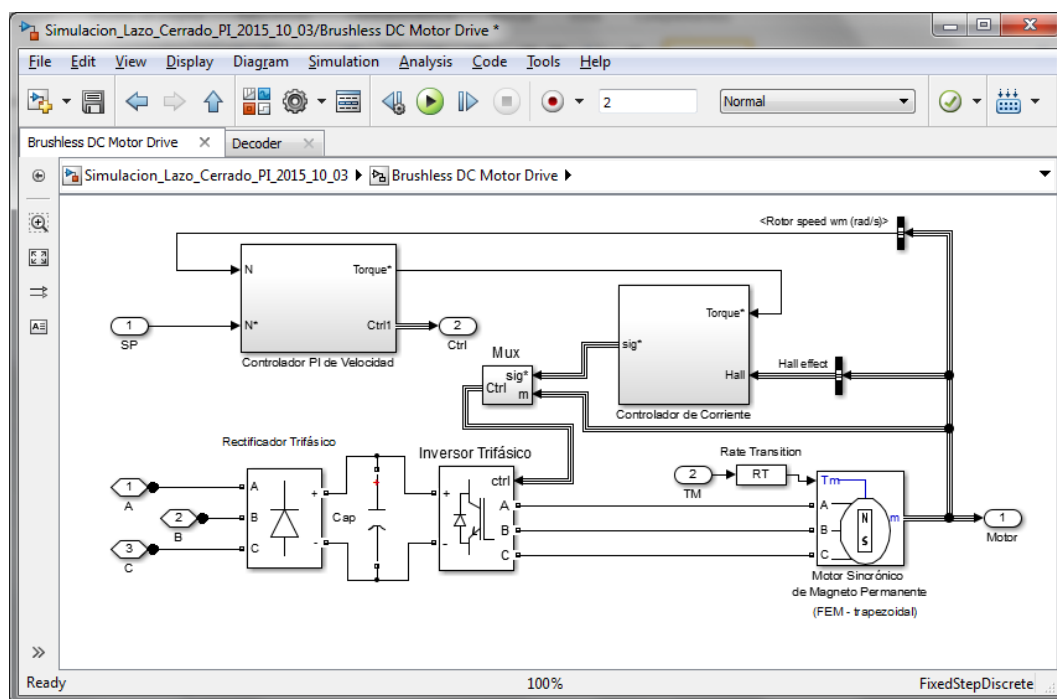


Figura 5-1 Diagrama de bloques del controlador PI en la simulación.

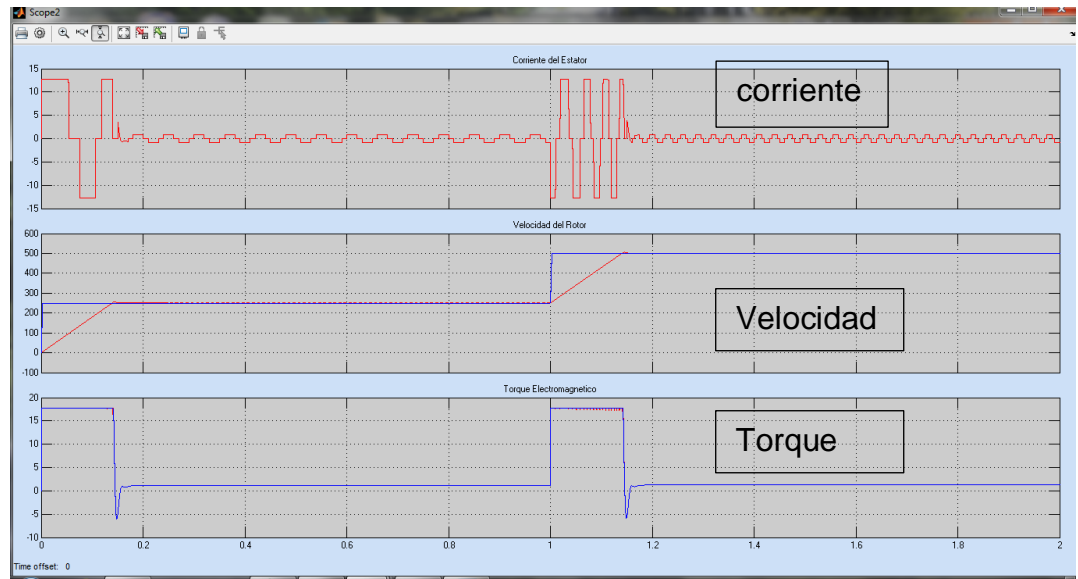


Figura 5-2 Señales de corriente, velocidad y torque del controlador PI, con variación de velocidad y torque de carga constante.

Para las constantes $K_p=3.3$ y $K_i=300$ la respuesta de velocidad es como se muestra en la figura 5.3 donde la variación de **250RPM** a **500RPM** tiene un tiempo de estabilización de $t_s=0.15$ segundos y un sobre-nivel porcentual $M_p=2.4\%$ $((506-250)*100/250)$.

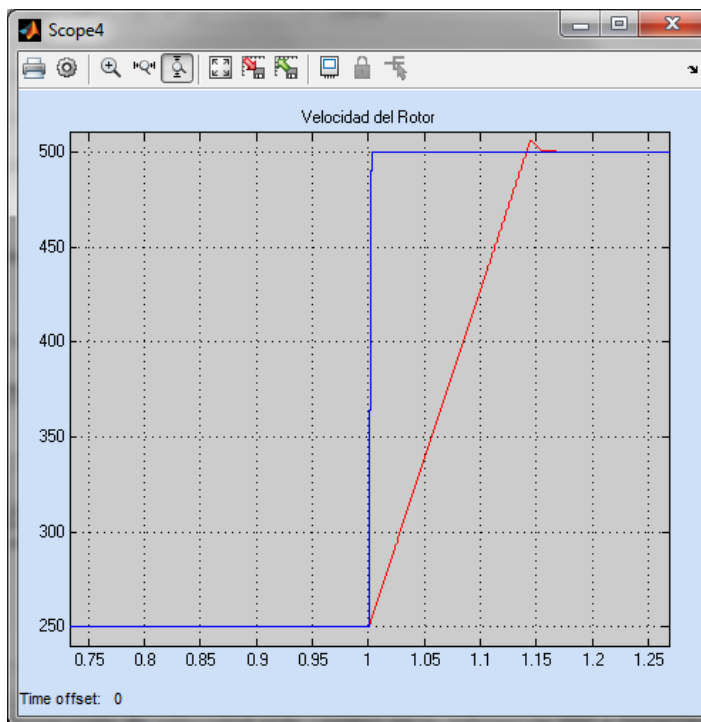


Figura 5-3 Respuesta de velocidad, Controlador PI, ante cambio en la referencia 250 a 500RPM.

Si mantenemos **constante la referencia de velocidad** en 250 RPM y **variamos la carga aplicada al eje del motor** de 1N-m a 2.5N-m (2.75N-m nominal), en la figura 5.4 vemos cómo la velocidad casi no sufre cambio y el controlador PI responde de una manera aceptable ante esta perturbación. Como resultado la corriente pico se eleva de 1 a 2 amperios, las RPM del motor disminuyen menos de 1 RPM pero rápidamente se recupera, y el torque generado por el motor sube de 1.1N-m a 2.7N-m aproximadamente.

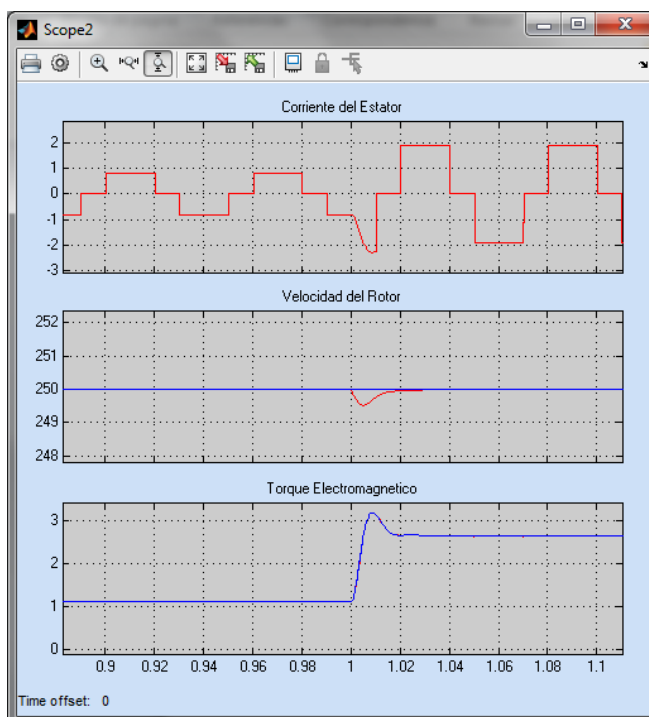


Figura 5-4 Respuesta del sistema, Controlador PI, ante un cambio repentino de torque de carga.

Para el **controlador difuso** se muestra en la figura 5.4 el diagrama de bloques, el cual es similar al del controlador PI, la diferencia está dentro del bloque del controlador lo que se muestra en la figura 5.5, en esta ocasión se necesitan de ganancias en las entradas y en la salida del controlador difuso las que harán que su respuesta mejore ante las variaciones de consigna.

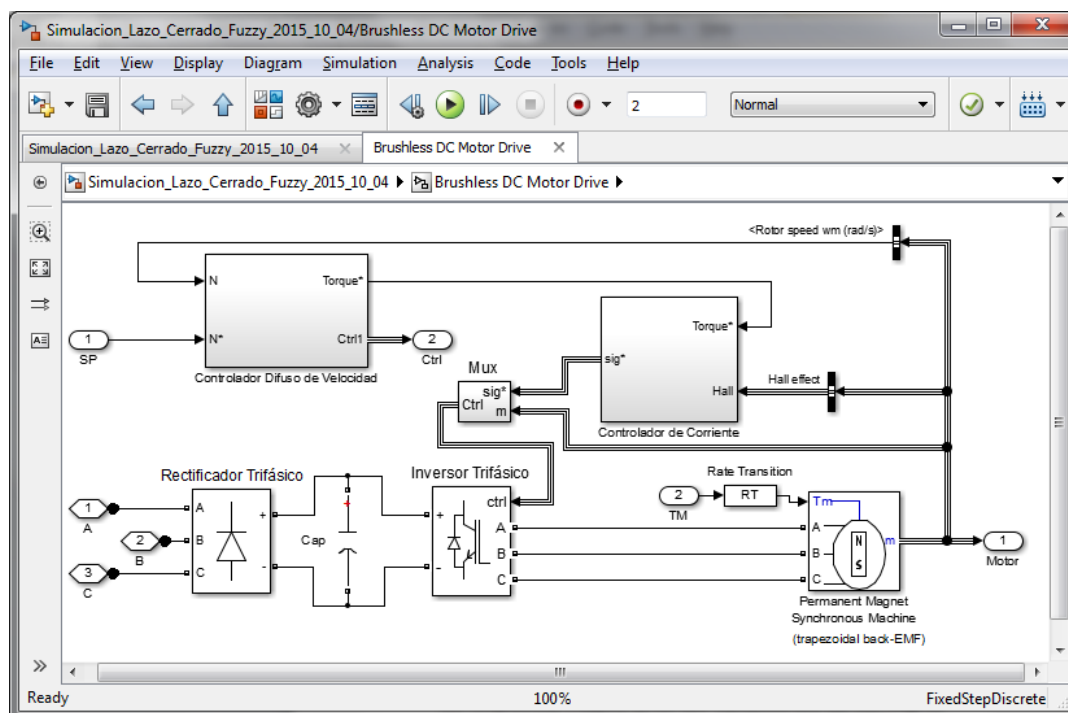


Figura 5-5 Diagrama de bloques del controlador Difuso en la simulación.

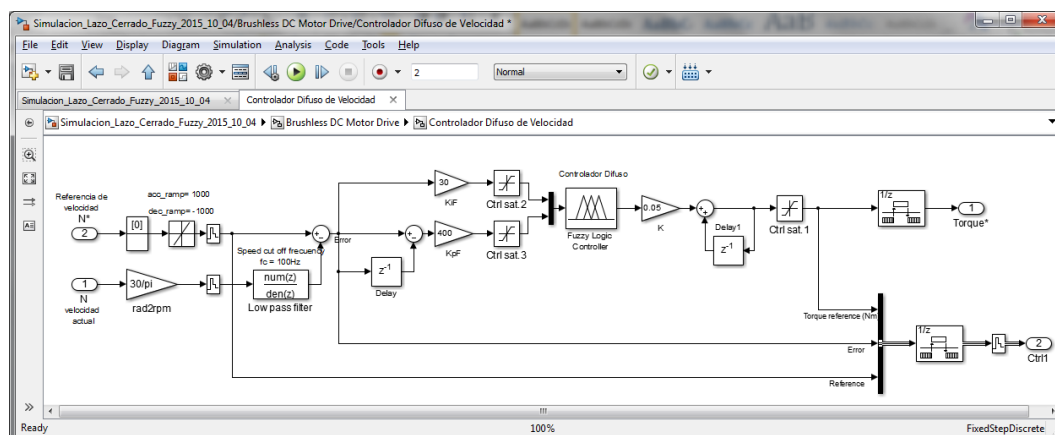


Figura 5-6 Controlador Difuso de velocidad.

Si se ajusta el controlador difuso, figura 5.6, tal que las ganancias sean $K_iF=30$ (Ganancia integral), $K_pF=400$ (Ganancia Proporcional) y $K=0.05$ (Ganancia de Salida), para una variación de velocidad de 0 a

250RPM en $t=0$ segundos y de 250 a 500RPM en $t=1$ segundo, se obtiene la respuesta de la figura 5.7 donde se observa: la corriente del motor, la velocidad del motor y la variación del torque electromagnético.

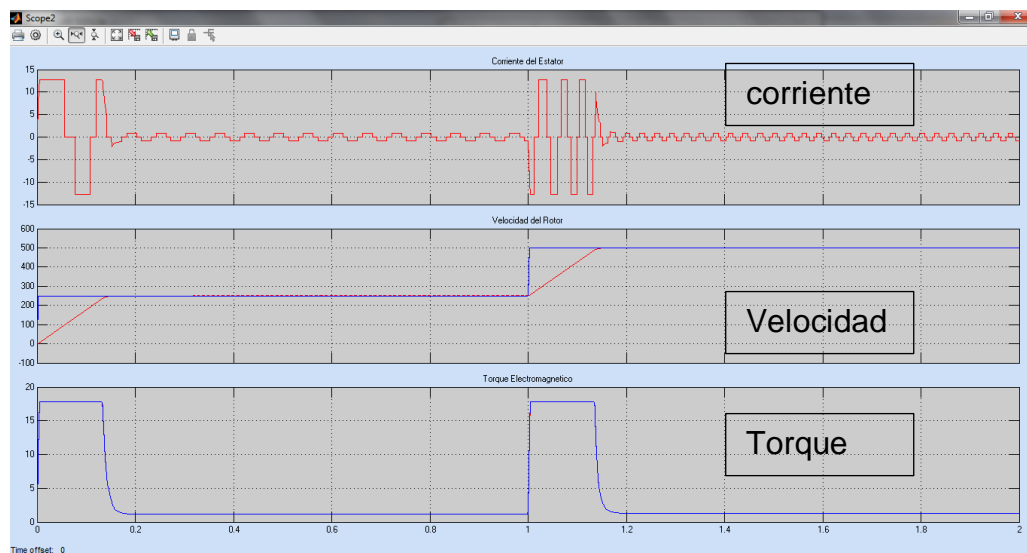


Figura 5-7 Señales de corriente, velocidad y torque del controlador Difuso, con variación de velocidad y torque de carga constante.

Si se toma en consideración el cambio de velocidad de **250 a 500 RPM** en **$t=1$ segundo**, se tiene una respuesta con tiempo de estabilización **$t_s=0.15$ seg** y sin sobre-nivel porcentual ya que la respuesta es críticamente amortiguada. Esto se muestra en la figura 5.8.

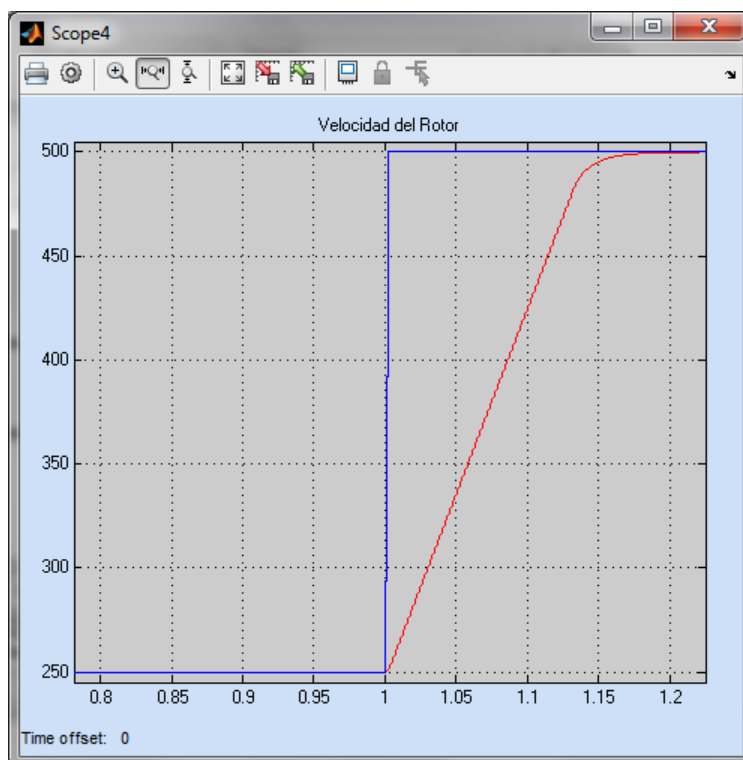


Figura 5-8 Respuesta de velocidad, Controlador Difuso, ante cambio en la referencia de 250 a 500RPM.

En el caso de mantener **constante la referencia de velocidad** en 250 RPM y **variar la carga aplicada al eje del motor** de 1N-m a 2.5N-m (2.75N-m nominal), en la figura 5.9 vemos cómo la velocidad casi no sufre cambio y el controlador Difuso responde de una manera aceptable ante esta perturbación. Como resultado la corriente pico se eleva de 1 a 2 amperios, las RPM del motor disminuyen menos de 1 RPM pero rápidamente se recupera, y el torque generado por el motor sube de 1.1N-m a 2.7N-m aproximadamente. Este comportamiento es similar al presentado por el controlador PI ante variación en el torque de carga.

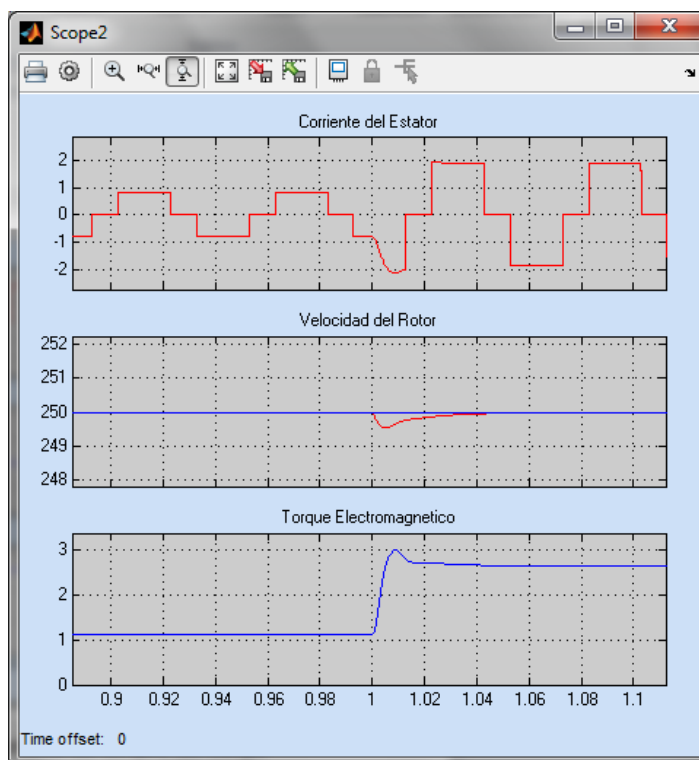


Figura 5-9 Respuesta del sistema, Controlador Difuso, ante un cambio repentino de torque de carga.

5.3. Resultados de la Implementación.

Para el **controlador PI** implementado en VisSim se tiene que ante una variación en la consigna de velocidad de $-600(-0.2)$ a $+600(0.2)$ RPM la respuesta del sistema da un tiempo de estabilización de $T_s = 3.48-3.34 = 0.14$ seg y un sobre-nivel porcentual $M_p = (0.21-(-0.2))*100/0.4 = 2.5\%$ lo que se muestra en la figura 5.10.

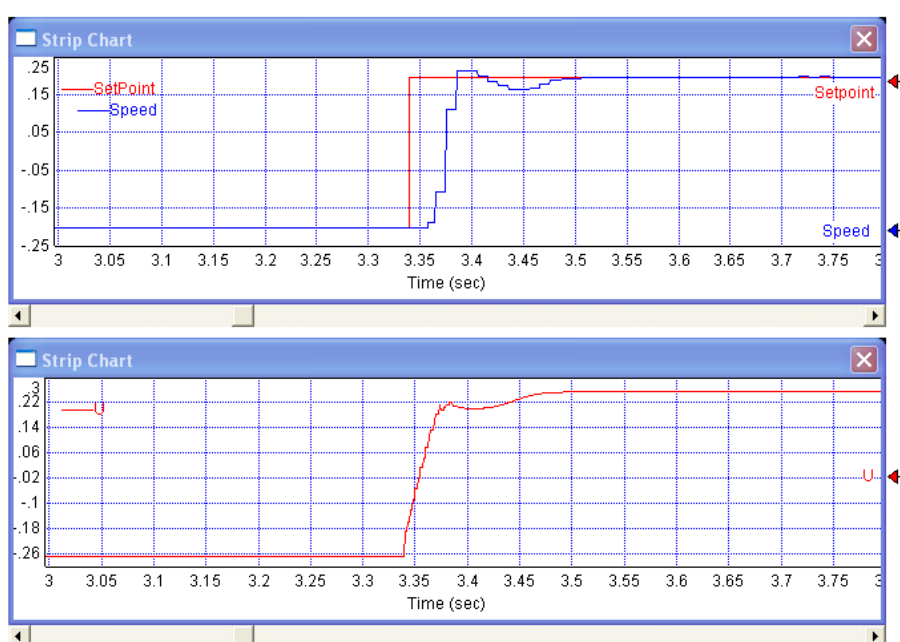


Figura 5-10 Respuesta del sistema, Controlador PI, implementación. Cambio en referencia de velocidad.

Para la implementación del controlador PI las constantes son $K_p=0.16$ y $K_i=0.036$, lo que nos da una respuesta aceptable.

En el evento de mantener la velocidad constante y existiera una perturbación en la carga la respuesta del sistema es como se muestra en la figura 5.11.

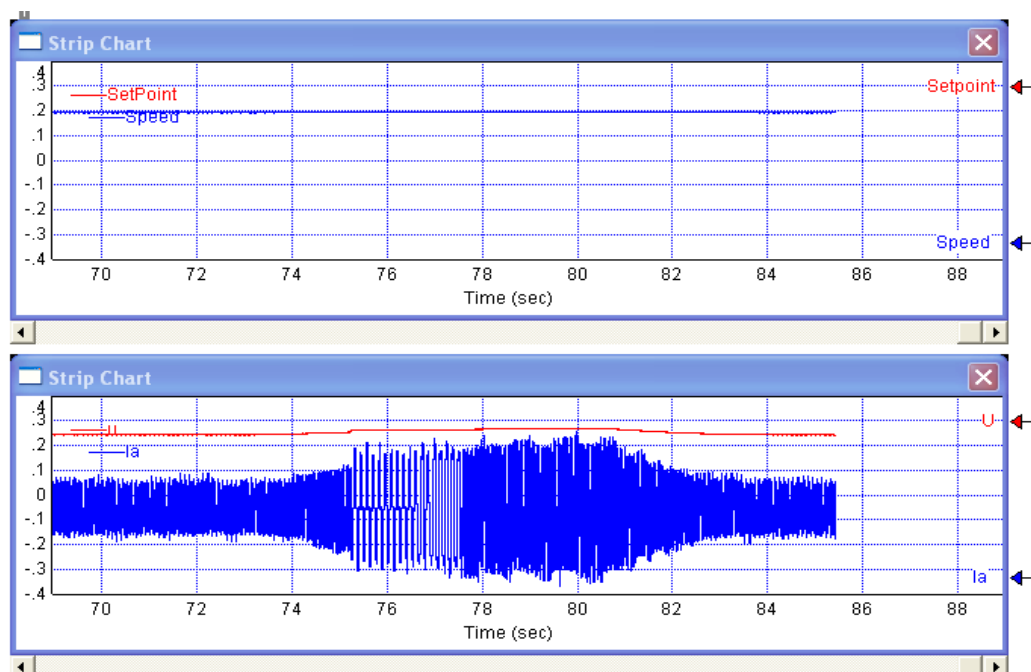


Figura 5-11 Respuesta del sistema, Controlador PI, implementación. Perturbación de la carga.

En las mismas condiciones anteriores la respuesta del controlador Difuso ante variaciones en la consigna de velocidad de $-600(-0.2)$ a $+600(0.2)$ RPM, se muestran en la figura 5.12

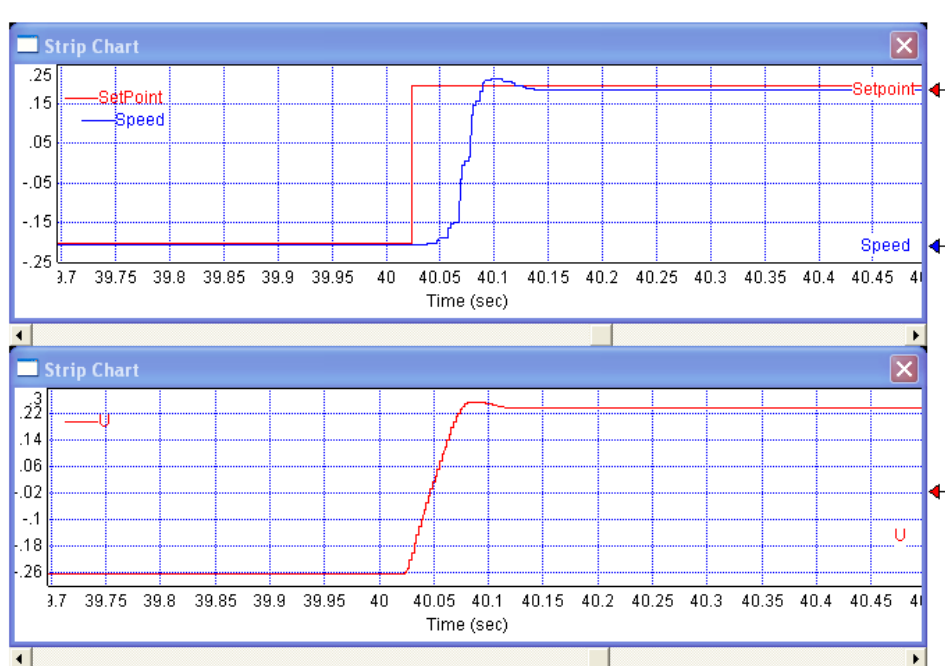


Figura 5-12 Respuesta del sistema, Controlador Difuso, implementación. Cambio en referencia de velocidad.

En este caso las constantes del controlador Difuso fueron: **$K_pF=0.002$** , **$K_iF=0.25$** y **$K=0.019$** lo que nos da una respuesta aceptable.

Simulando una perturbación en la carga aplicada al eje del motor, el controlador difuso responde como se muestra en la figura 5.13

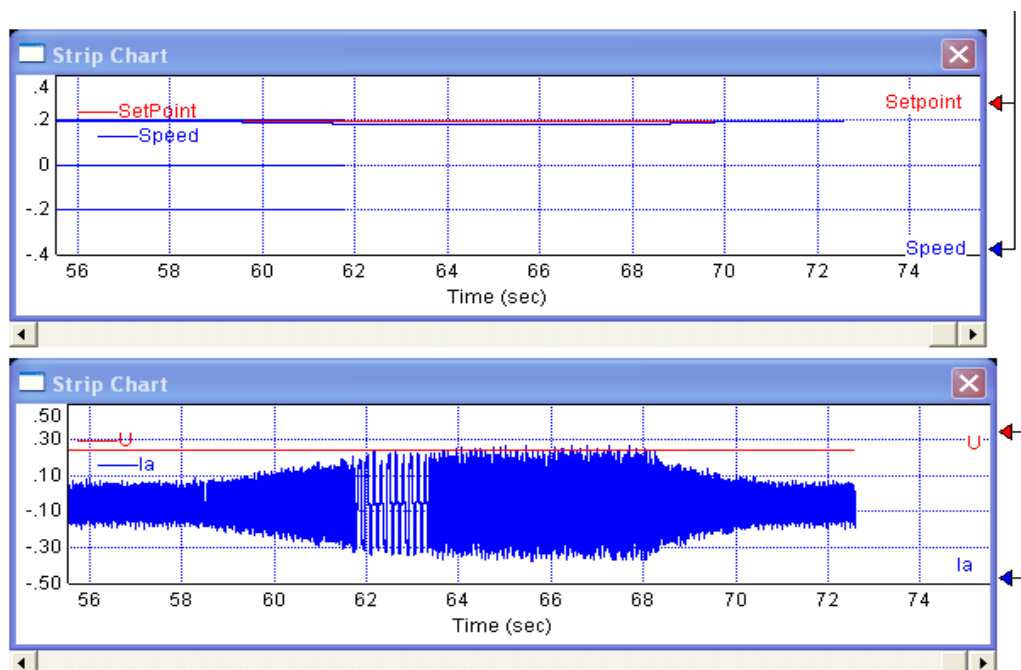


Figura 5-13 Respuesta del sistema, Controlador Difuso, implementación. Perturbación de la carga.

5.4. Análisis comparativo de los controladores.

Por el lado de la simulación se puede decir que tanto el controlador PI como el controlador Difuso tienen un buen control del motor BLDC ante cambios en la referencia de velocidad o perturbaciones en la carga. La ventaja del controlador Difuso ante el controlador PI es que la señal de control U en este caso el torque de referencia, es más suave para el controlador Difuso mientras que en el PI presenta cambios bruscos, lo cual hace que la dinámica de conmutación de los IGBT's de la sección inversora sean fácilmente expuestos a daños. La respuesta de la simulación de ambos controladores se muestra en la figura 5.14.

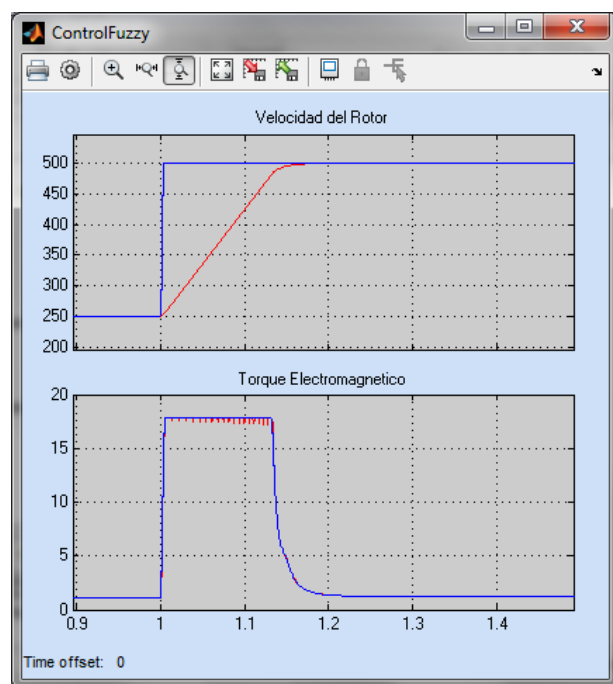
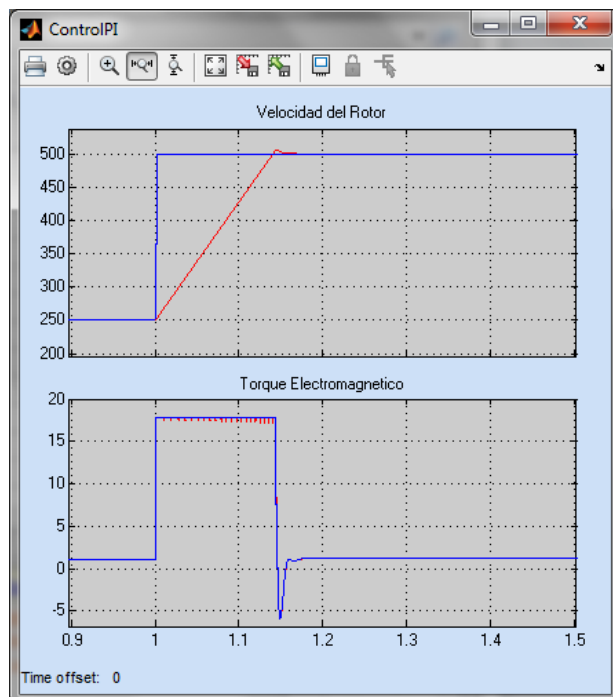
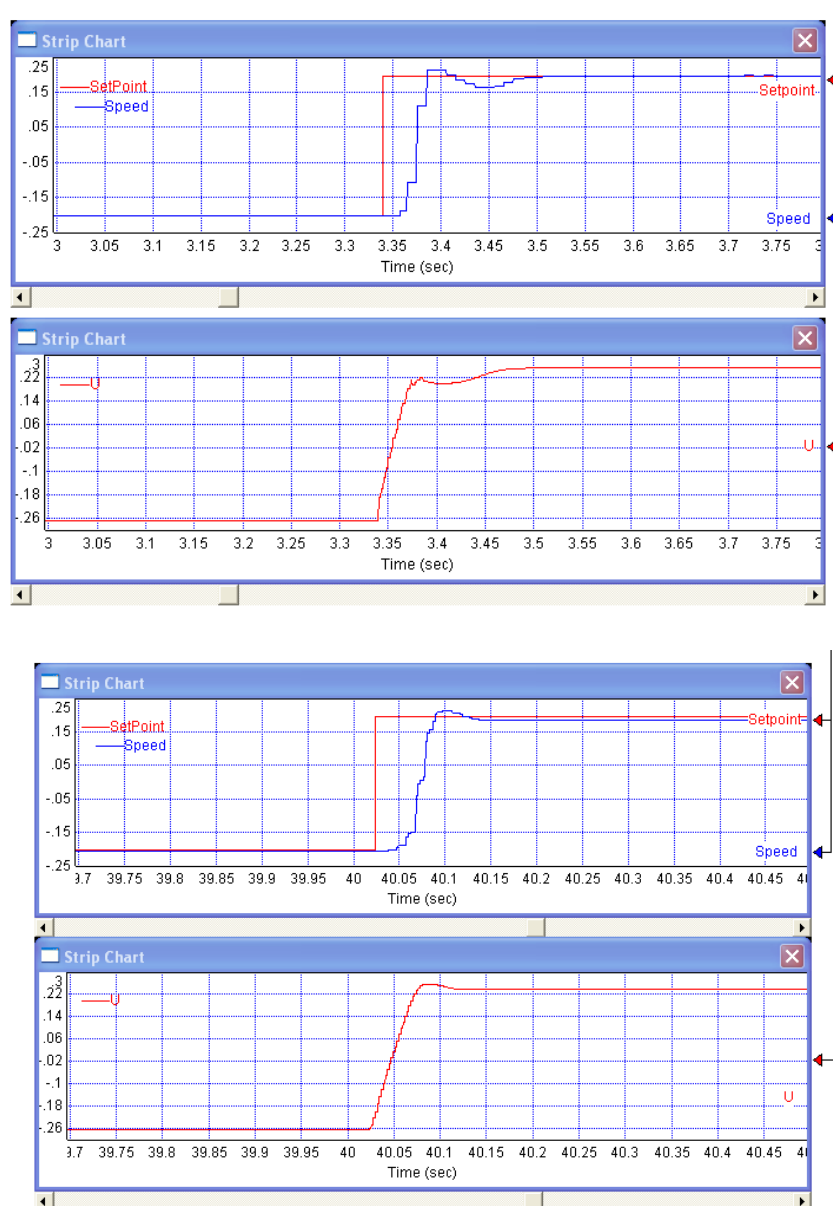


Figura 5-14 Comparación de resultados de la simulación.

Mientras que la respuesta del sistema real se muestra en la figura 5.15,



**Figura 5-15 Comparación de resultados de la implementación;
Arriba: PID; Abajo: Fuzzy**

En la tabla 5.1 se puede observar el resultado de los controladores difuso y PID, tanto en la simulación como en la implementación.

Tabla 5.1 Resultado de los controladores implementados.

Controlador	Simulación		Implementación	
	Fuzzy	PID	Fuzzy	PID
Max Sobre Nivel Porcentual, %	0	2.4	3	2.5
Tiempo de estabilizacion, sec	0.15	0.15	0.11	0.14

CONCLUSIONES

En las pruebas realizadas entre el controlador de lógica difusa y el controlador PID se puede determinar que ambos controladores tienen una respuesta muy similar en cuanto a la variable tiempo de estabilización (T_s), sin embargo el controlador difuso tiene un menor sobre nivel porcentual (%MP) y la salida del controlador (U) tiene una forma más suavizada sin cambios bruscos, lo que ayuda para que no existan variaciones perjudiciales en la señal de control y estas lleguen a dañar la parte electrónica por las variaciones bruscas de esta señal.

Se pudo desarrollar un controlador difuso para el control de velocidad de un motor DC sin escobillas, identificando las entradas del controlador: error y la derivada del error. Así también la salida como una variación de torque para la simulación y como una variación del ciclo de trabajo en la implementación. Sin la necesidad del modelo matemático del sistema.

La lógica difusa convierte los problemas complejos en problemas más simples usando el razonamiento aproximado. El sistema se describe por reglas difusas y el número de funciones de membresía, usando expresiones humanas y las variables lingüísticas. De esta manera se

puede usar el conocimiento del experto eficazmente para describir la conducta del sistema.

Cuando la complejidad del sistema aumenta, es más difícil determinar el conjunto de reglas correctas y funciones de membresía que mejor describan la conducta del sistema. Se necesita emplear una gran cantidad de tiempo para poner a punto las funciones de membresía, ajustar las reglas y las ganancias del controlador para obtener una buena solución. Mientras los sistemas son más complejos, son necesarias más reglas y aumenta la dificultad para relacionarlas entre sí.

El equipo educativo DMD2000 disponible en el laboratorio de potencia usa un procesador digital de señales, TMS320F2812 de Texas Instruments, el cual tiene las características necesarias en cuanto a periféricos de entrada y salida para realizar el control de motores AC, servo motores con FEM sinusoidal y FEM trapezoidal, así como también motores DC con escobillas.

El software usado en la simulación, MATLAB, es una potente herramienta para desarrollar algoritmos de control con controladores PID o mediante Lógica difusa mediante su toolbox FUZZY

BIBLIOGRAFÍA

- [1] M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, "Neural and Fuzzy Logic Control of Drives and Power Systems", Editorial Newnes, England, 2002.
- [2] G.Sakthivel, "Real Time Implementation of DSP based Fuzzy Logic Controller for Speed Control of BLDC Motor", Dept of E&I . Annamalai University, INDIA. Volume 10– No.8, November 2010.
- [3] Gómez juan C, Fuzzy Control, edUTecNe Universidad Tecnológica Nacional, Argentina, 2008.
- [4] Benito Matías, Durán Vicente, Lógica Borrosa, <https://ingtecnologia.files.wordpress.com/2011/06/logica-difusa2pdf.pdf>
[Último acceso: 05 Febrero 2015].
- [5] Bryan Klingenberg, Fuzzy applications, Calvin College Engineering Department, <https://www.calvin.edu/~pribeiro/othrlnks/Fuzzy/apps.htm>,
[Último acceso: 06 Febrero 2015].
- [6] K. Naga Sujatha, K. Vaisakh and Anand. G.. Artificial Intelligence based speed control of brushless DC motor. IEEE 978-1-4244-6551-4/10. 2010

- [7] Cheng-Tsung Lin, Chung-Wen Hung and Chih-Wen Liu.. Fuzzy PI controller for BLDC motors considering Variable Sampling Effect. IEEE Industrial Electronics Society (IECON). Taipei, Taiwan, Nov. 5-8, 2007
- [8] M. V. Ramesh, J. Amarnath, S. Kamakshaiah and G. S. Rao, Speed control of brushless DC motor by using Fuzzy logic PI controller, ARPN journal, http://www.arpnjournals.com/jeas/research_papers/rp_2011/jeas_0911_555.pdf, [Último acceso: 06 Febrero 2015].
- [9] Díaz Duque J, Metodología de la investigación científica, La Habana, 2009.
- [10] M. Santos, Aplicaciones Exitosas de Control Inteligente a Casos Reales, Revista Iberoamericana de Automática e Informática industrial, 2011
- [11] Ph.D. Misa Llorca, Control Inteligente, Maestría en Automatización y Control Industrial, Guayaquil, 2013.
- [12] K. Passino, S. Yurkovich, Fuzzy Control, Addison-Wesley, 1997
- [13] The MathWorks, Inc. MATLAB, Fuzzy Logic Toolbox. User's Guide, Octubre 2014
http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf [Último acceso: 11 Febrero 2015].

- [14] Hung T. Nguyen, Nadipuram R. Prasad, Carol L. Walker, Elbert A. Walker, A first course in Fuzzy and neural control, Chapman & Hall/CRC, 2003.
- [15] Pérez Pueyo, R. Procesado y Optimización de Espectros Raman mediante Técnicas de Lógica Difusa: Aplicación a la identificación de Materiales Pictóricos. Universidad Politécnica de Cataluña, Tesis doctoral, <http://hdl.handle.net/10803/6887> [Último acceso: 11 Febrero 2015].
- [16] Sugeno, M., Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985.
- [17] Åström, K. J. and Hägglund, T. PID controllers: Theory, Design, and Tuning. Research Triangle Park, NC 27709, Segunda Edición, USA. (1995).
- [18] Herrera, Métodos de sintonización de controladores PID, Control de Procesos, Universidad Nacional de Tucumán, Argentina. http://www.herrera.unt.edu.ar/controldeprocesos/Tema_4/Tp4a.pdf [Último acceso: 14 Febrero 2015].
- [19] Jianwen Shao, Direct Back EMF Detection Method for Sensorless Brushless DC (BLDC) Motor Drives, Instituto Politécnico de Virginia, USA, 2003.

- [20] Shane W. Colton, Design and Prototyping Methods for Brushless Motors and Motor Control, Massachusetts Institute of Technology (MIT), 2010
- [21] Padmaraja Yedamale, AN885 - Brushless DC (BLDC) Motor Fundamentals. Microchip Technology Inc. 2003
- [22] Krause, P.C Analisis of Electrical Machinery. Kinsport Press Inc. Kinsport Town, 1986
- [23] Pillay, P., Krishman, R. Modeling, simulation and analysis of permanent magnetic motor drives, Part I: The permanent magnet synchronous motor drive. IEEE transaction on industry Aplication, 1989
- [24] Chang-Liang Xia, Permanent magnet brushless DC motor drives and controls, Tianjin University, P.R China. Wiley, 2012
- [25] Damián Larco Gómez, Diseño y construcción de una plataforma experimental de desarrollo y pruebas de sistemas de control vectorial de motores eléctricos mediante procesamiento digital de señales y programación gráfica asistida por computador, Universidad ESPOL, FIEC Guayaquil-Ecuador. 2013.
- [26] Spectrum Digital, eZdspF2812 Technical Reference, Spectrum Digital Inc, Septiembre 2003.

- [27] Visual Solutions, Inc. VisSim User's Guide Version 8.0, 2010.
- [28] Cho Chieh Tech. Enterprise Ltd. DSP Motor control Developing System DMD2000, 2009.
- [29] Andreas Dannenberg, Fuzzy Logic Motor Control with MSP430x14x, Texas Instrument Application Report SLAA235, Febrero 2005.
- [30] Karr, C.L., Gentry, E.J.: 'Fuzzy Control of pH Using Genetic Algorithm', IEEE Trans. Fuzzy Syst., vol. 1, no. 1, Jan. 1993, pp. 46–53.
- [31] Hmaifar, A., McCormick, E.: 'Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms', IEEE Trans. Fuzzy Syst., vol. 3, no. 2, May 1995, pp. 129–139.

6. ANEXOS

6.1. ANEXO A: Diagrama interno del controlador Difuso.

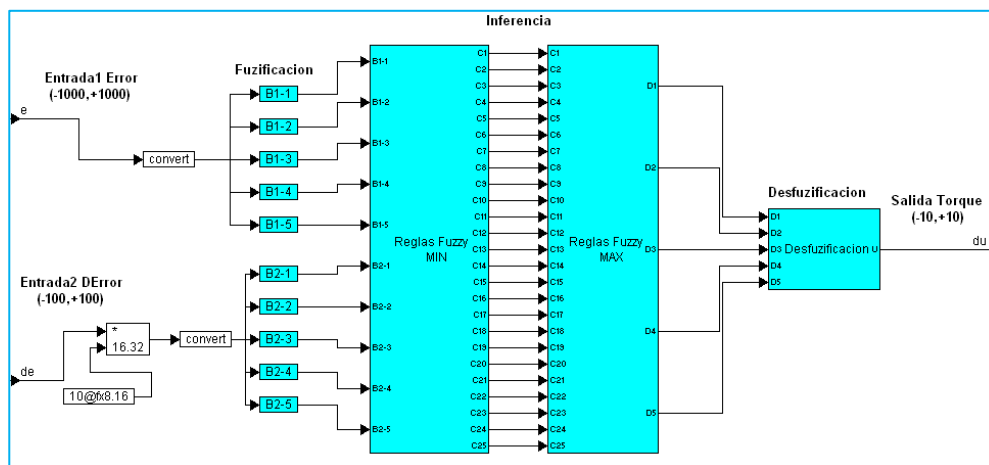


Diagrama general del controlador difuso

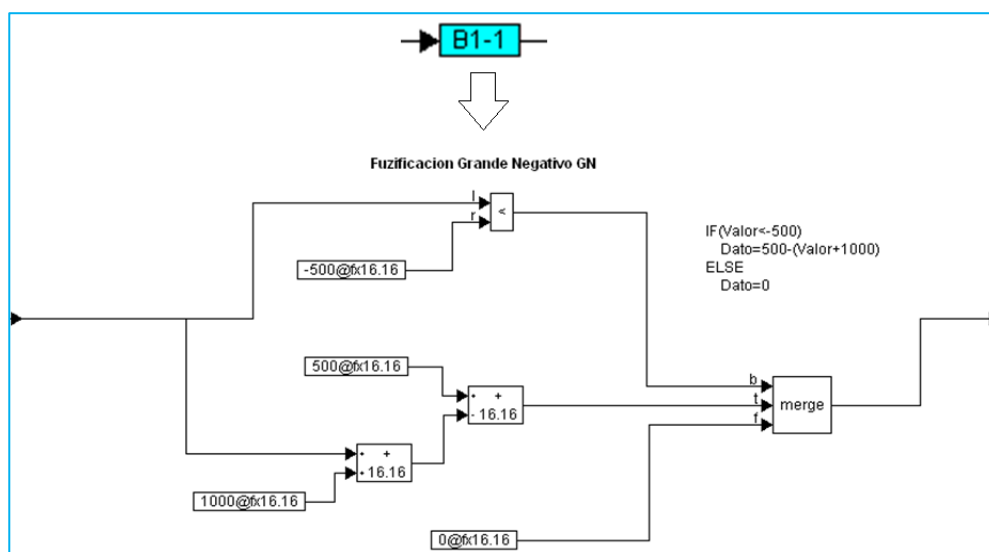


Diagrama Interno del bloque de fusificación B1-1

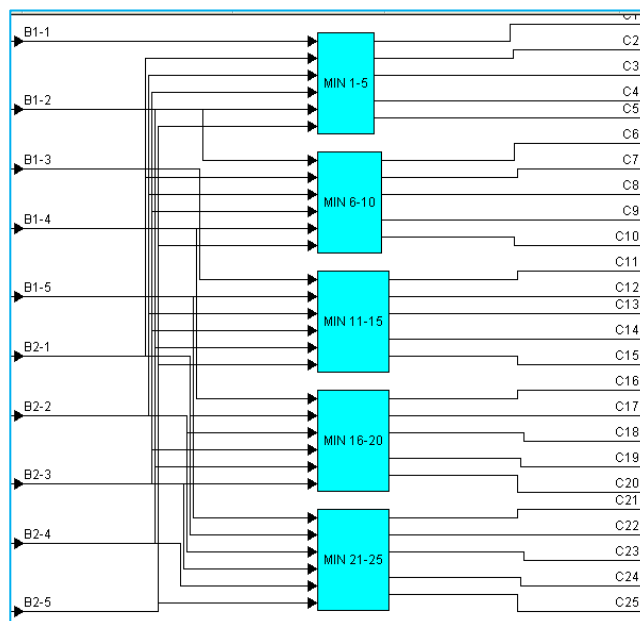


Diagrama interno del bloque Reglas Fuzzy MIN

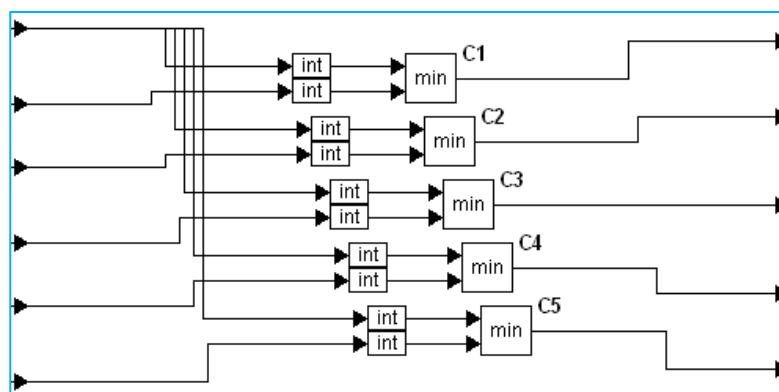


Diagrama interno del bloque MIN 1-5

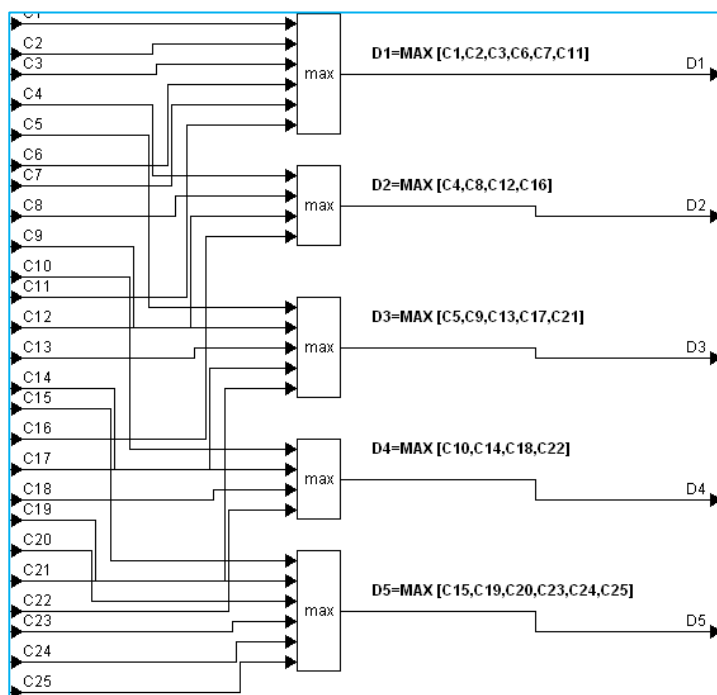


Diagrama interno del bloque Reglas Fuzzy MAX

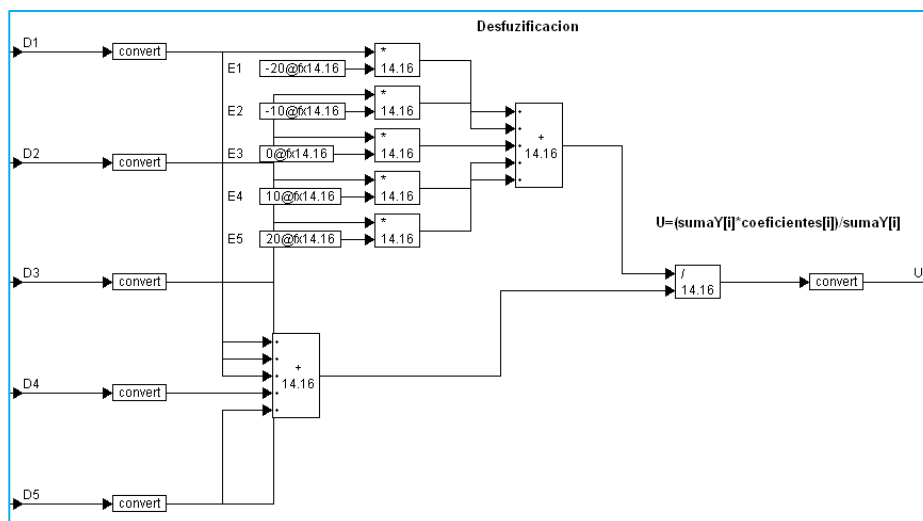


Diagrama interno del bloque de desfuzificación.

6.2. ANEXO B: Freno Magnético DMD-2000B

Magnetic Power Brake DMD-2000B

DMD-2000B is a select option for DMD2000 DSP Motor Control Developing System, it is a load unit for PMSM, DC sever motor and Induction Servo Motor.

Specifications :

1. Control range: 0~50/0 ~ 12 Kg-cm
2. Maximum speed: 1800 rpm
3. Load cell: 10Kg, 0.01 % FS
4. Manual control of the brake
5. Digital display: 0 ~ +/- 50.00 Kg-cm
6. External torque control: 0~10 Vdc
7. Torque output signal: -10~10 Vdc

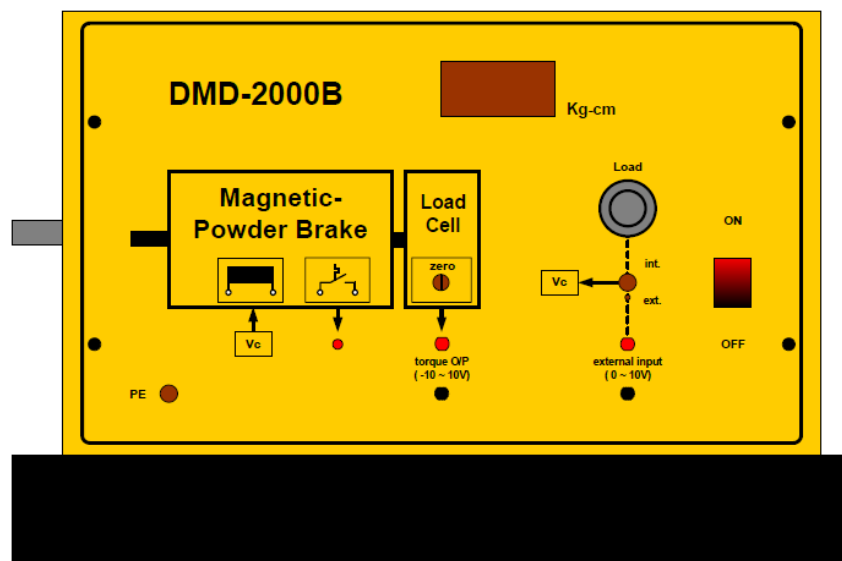


Figure C-1 DMD-2000B front View

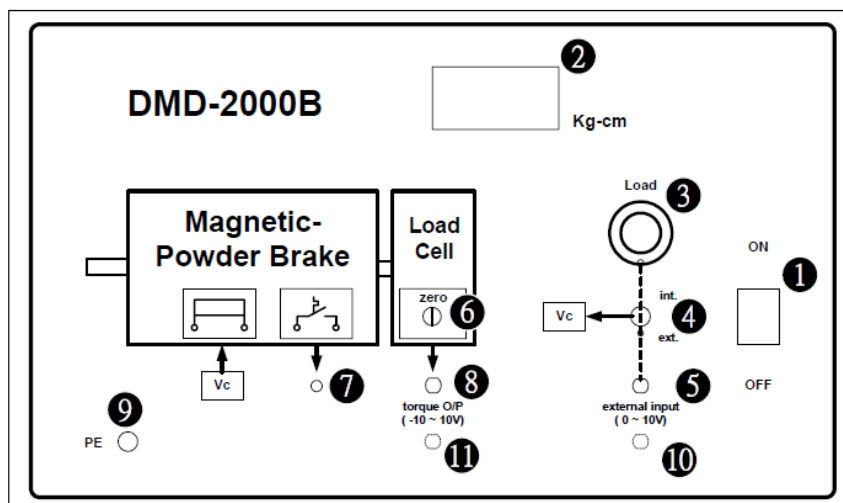


Figure C-2 DMD-2000B front panel and test point

NO	Name	Description
1	Main Power	Control power supply, Light when On.
2	Digital torque display	Display torque by voltage
3	Torque tuning knob	Control torque when using int. torque control
4	Torque control source switch	Select int. or ext. torque control source
5	Ext. torque control input point	Ext. torque source control input point. Input Range: 0~10V
6	Zero regulator	Use when there is bias voltage.
7	Overheat indicator LED	It lights when system overheat, then protection circuit works.
8	Torque output point	Let use can measure torque in voltage by a scope.
9	GND	Connect to Ground
10	GND	Connect to Ground