



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

implementación de técnicas de control avanzado en un sistema embebido para el control de
velocidad de un motor trifásico asíncrono de la universidad de pamplona

Elmer Erney Meneses Jurado

Contacto: ingelmer95@gmail.com

Director:

Msc. Juan Martín Cáceres

trabajo de grado para optar por el título de ingeniero mecatrónico

Universidad de Pamplona

facultad de ingenierías y arquitecturas

programa de Ingeniería Mecatrónica



*Formando líderes para la construcción de un
nuevo país en paz*



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Este trabajo lo dedico a Argenis Jurado Eraso y Alirio Meneses Pantoja por su apoyo incondicional que me han brindado desde siempre, por su valentía y deseos profundos de darme una educación primero desde casa, y acompañarme en mis estudios universitarios.

A Rubí Jurado Eraso, por creer en mí en todo momento y brindarme todo su apoyo cuando más lo necesité.



**Formando líderes para la construcción de un
nuevo país en paz**



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Agradecimientos

Agradezco en primera instancia a la Universidad de Pamplona por permitirme ser parte de su comunidad y estudiar esta carrea, agradezco al profesor Juan Cáceres, por brindarme su asesoría durante el proyecto con la cual logré llevar a cabo todos los objetivos con normalidad, agradezco a mis profesores que, con la mejor voluntad estuvieron dispuestos a compartir y brindar sus conocimientos, que me ayudaron a formarme hasta llegar a este punto, a mis amigos que me brindaron su ayuda en momentos difíciles por los que se pasa en una carrera profesional, a mi familia que siempre han creído en mí y en este proyecto, a todos ellos y ellas con un profundo y sincero sentimiento digo, gracias.



**Formando líderes para la construcción de un
nuevo país en paz**



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Tabla de Contenidos

Tabla de Contenidos	4
Lista de Tablas.....	7
Lista de Figuras.....	8
Abstract.....	11
Introducción.....	12
Planteamiento del problema	14
Justificación.....	15
1 Objetivos.....	16
1.1 Objetivo General.....	16
1.2 Objetivos Específicos.....	16
2 Marco teórico.....	18
2.1 Lazos de Control.....	18
2.1.1 Control lazo abierto	18
2.1.2 Control lazo cerrado.....	19
2.2 Control PID.....	19
2.3 Tarjetas de desarrollo	21
2.3.1 Microcontrolador ARM	21
2.3.2 Tarjeta de desarrollo teensy	23
2.3.3 Raspberry PI 3	24
2.3.4 Dreamer maple	26
2.4 Técnicas de control avanzado.....	29
2.4.1 Control difuso.....	30
2.4.2 Redes neuronales.....	34
2.5 Motores trifásicos asíncronos.....	40
3 Estado del arte	43
4 Capítulo 1 Reconocimiento de la planta	47



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

4.1	Componentes.....	47
4.2	Descripción de componentes.....	48
4.2.1	Motor trifásico asíncrono.....	48
4.2.2	Variador de frecuencia.....	49
4.2.3	Taco – Generador.....	50
4.3	Conexiones y empalmes	51
4.3.1	Conexión del motor trifásico.....	53
4.3.2	Configuración del variador.....	55
5	Capítulo 2 Acondicionamiento de la planta.....	56
5.1	Criterio de selección.....	57
5.2	Circuitos de acondicionamiento	57
5.2.1	Circuito Amplificador	58
5.2.2	Circuito divisor de tensión	60
5.3	Diseñando circuito impreso	63
5.4	Caracterización del Taco – Generador.....	65
6	Capítulo 3 Identificación de la planta	67
6.1	Identificación de la planta.....	67
6.2	Comprobación del modelo encontrado.....	72
7	Capítulo 4 Controladores simulados.....	74
7.1	Control PID discreto	74
7.2	Control Fuzzy.....	86
7.2.1	Creando algoritmo de control.....	91
7.2.2	Simulación control fuzzy.	98
7.3	Control neuronal directo.....	101
7.3.1	Desarrollando algoritmo para control neuronal.....	102
7.3.2	Simulando control neuronal directo	107



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

8	Capítulo 5 Aplicando controladores diseñados a la planta física.....	114
8.1	Control PID discreto	115
8.2	Control fuzzy	117
8.3	Control neuronal directo.....	120
8.4	Comparación de las técnicas de control	124
9	Conclusiones	132
10	Recomendaciones	135
11	Referencias.....	136





IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Lista de Tablas

Tabla 1. Velocidades de sincronismo. (Etolocka, 2014).....	42
Tabla 2. Componentes de la planta a utilizar. (Autor)	48
Tabla 3. Precios de tarjetas de desarrollo.....	57
Tabla 4 Cálculo constantes Ziegler y Nichols.(Villajulca, 2011)	82
Tabla 5 Primera matriz de reglas del controlador fuzzy. (Autor)	91
Tabla 6 Matriz de reglas compuesta controlador fuzzy. (Autor)	91



Lista de Figuras

Figura 1 Lazo abierto de control.(fmeautomatismos, n.d.)	18
Figura 2 Lazo cerrado de control.(fmeautomatismos, n.d.)	19
Figura 3 Arquitectura de un microcontrolador ARM.(Pérez, 2017)	22
Figura 4 Tarjeta de desarrollo teensy 3.6 (Digi-Key, 2017)	24
Figura 5 Tarjeta raspberry con sus componentes.(Ro-botica, 2017).....	25
Figura 6 Vista general de la tarjeta de desarrollo maple. (Df robot, 2017).....	28
Figura 7 Pines y periféricos de la tarjeta maple. (Df robot, 2017)	29
Figura 8 Funciones de pertenencia.(Greg, 2013)	32
Figura 9 Estructura de una red neuronal artificial multicapa (RNA). (Marulanda, 2010)	38
Figura 10 Partes de motor trifásico asíncrono. (FRMza, n.d.).....	41
Figura 11 Esquema control neuronal auto-ajutable. (Noriega et al., 2008).....	47
Figura 12 Motor trifásico asíncrono. (Autor).....	49
Figura 13 Variador de frecuencia académico. (Autor).....	50
Figura 14 Taco- generador. (Autor).....	51
Figura 15 Acoples de motor y taco-generador. (Autor).....	52
Figura 16 Acople diseñado para la planta. (Autor).....	52
Figura 17 Planta con el acople. (Autor)	53
Figura 18 Conexiones del motor trifásico. (Autor).....	54
Figura 19 Conexión del motor al variador. (Autor)	54
Figura 20 Teclado y pantalla del variador. (Autor).....	55
Figura 21 Amplificador no inversor. (Hetpro, 2018).....	59
Figura 22 Circuito divisor de tensión. (Autor).....	61
Figura 23 Circuito general acondicionamiento. (Autor)	64
Figura 24 Circuito acondicionamiento impreso. (Autor).....	64
Figura 25 Diseño caja circuito. (Autor)	65
Figura 26 Datos de Voltaje vs RPM	66
Figura 27 Gráfica de Voltaje Vs RPM. (Autor).....	66
Figura 28 Función de transferencia. (Autor).....	68
Figura 29 Señal de entrada vs salida de la planta. (Autor).....	69
Figura 30 Datos en herramienta identificación (Autor)	70
Figura 31 Función de transferencia en Simulink. (Autor)	72



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Figura 32 Entrada Vs salida del modelo virtual. (Autor).....	73
Figura 33 Señal real Vs simulada. (Autor)	74
Figura 34. Esquema general de simulación. (Autor).....	75
Figura 35 Esquema control PID discreto Simulink. (Autor).....	79
Figura 36 Efecto wind up en señal de control. (Autor).....	80
Figura 37 Sintonización Por Ziegler y Nichols. (Autor).....	82
Figura 38 Esquema Simulink control PID. (Autor)	83
Figura 39 Respuesta control PID discreto 1. (Autor).....	84
Figura 40 Señal de control PID. (Autor).....	85
Figura 41 Respuesta control PID ajustado. (Autor).....	85
Figura 42 Señal de control ajustada. (Autor)	86
Figura 43 Conjuntos difusos de entrada. (Autor).....	88
Figura 44 Conjuntos difusos para delta error. (Autor).....	88
Figura 45 Análisis de error y delta error. (Autor).....	89
Figura 46 Conjuntos difusos de salida. (Autor)	90
Figura 47 Estructura de inferencia Mamdani.(Sun et al., 2018)	92
Figura 48 Funciones para aplicar control fuzzy. (Autor).....	94
Figura 49 Función trapezoidal control fuzzy. (Autor)	96
Figura 50 Función inferencia mamdani. (Autor)	97
Figura 51 Función defuzzificación. (Autor)	97
Figura 52 Esquema Simulink planta control fuzzy. (Autor).....	98
Figura 53 Controlador difuso en Simulink. (Autor).....	98
Figura 54 Señal control fuzzy Vs toolbox. (Autor).....	99
Figura 55 Salida de la planta control difuso. (Autor).....	100
Figura 56 Esquema simulink para control neuronal directo. (Autor)	101
Figura 57 Estructura de la red neuronal para el control. (Autor)	102
Figura 58 Controlador neuronal directo. (Autor).....	107
Figura 59 Código control neuronal directo. (Autor).....	108
Figura 60 Respuesta del controlador neuronal. (Autor).....	109
Figura 61 Señal control neuronal. (Autor).....	110
Figura 62 Esquema control pid, difuso, neuronal. (Autor)	111
Figura 63 Respuesta de control pid, difuso y neuronal. (Autor).....	111
Figura 64 Respuesta ante cambios de setpoint. (Autor).....	112
Figura 65 PWM vs RPM. (Autor).....	113



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Figura 66 Referencia trayectoria compleja. (Autor)	114
Figura 67 Controlador PID planta física. (Autor)	115
Figura 68 Respuesta planta física a controlador PID. (Autor)	116
Figura 69 Respuesta2 control PID. (Autor)	117
Figura 70 Código para control fuzzy. (Autor)	118
Figura 71 Código control fuzzy2. (Autor)	118
Figura 72 Respuesta planta física a control fuzzy. (Autor).....	120
Figura 73 Respuesta2 controlador fuzzy. (Autor).....	120
Figura 74 Código control neuronal. (Autor)	121
Figura 75 Código actualización de pesos red neuronal. (Autor).....	122
Figura 76 Código señal de control neuronal. (Autor)	122
Figura 77 Respuesta control neuronal. (Autor).....	123
Figura 78 Controles a señal paso. (Autor)	124
Figura 79 Controles cambio setpoint. (Autor)	125
Figura 80 Controles cambio setpoint2. (Autor)	126
Figura 81 Controles cambio setpoint3. (Autor)	127
Figura 82 Controles trayectoria compleja1. (Autor).....	128
Figura 83 Setpoint subida señal compleja. (Autor).....	128
Figura 84 Rampa positiva señal compleja. (Autor)	129
Figura 85 Rampa negativa señal compleja. (Autor).....	130
Figura 86 Paso bajada señal compleja. (Autor)	131





IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Abstract

En la actualidad casi todos los procesos de producción incorporan dispositivos mecánicos, eléctricos, electrónicos, robóticos, etc, optimizando el proceso manual que se hacía en tiempos pasados teniendo una sinergia entre estos dispositivos, siendo capaces de realizar las tareas más complejas. Existen distintos métodos de control de estos dispositivos que ayudan al rendimiento de las máquinas encargadas del proceso, por lo general en la industria se usa la técnica de control PID (proporcional, integral, derivativo) para lograr este objetivo, sin embargo, con el avance de la tecnología y la computación, actualmente existen técnicas avanzadas que se podrían aplicar como control difuso, algoritmos genéticos, redes neuronales.

Uno de los inconvenientes al utilizar estas técnicas avanzadas de control es que requieren de software pago instalado en un computador para llevarlas a cabo, esto requiere un estudio de costo-beneficio para evaluar si es rentable implementarlo en la máquina. En el siguiente trabajo se aborda esta problemática, implementando algunas de estas técnicas en un micro controlador el cual es más económico. Para ello se usó como máquina o planta a controlar, un motor trifásico asíncrono de corriente alterna, al cual se le aplicaron distintas técnicas avanzadas para controlar su velocidad de giro, se plantean las técnicas de: control PID, control Difuso y control neuronal directo, se realizó comparaciones y análisis entre ellos para determinar con cual se obtiene los mejores resultados.

Palabras claves: Control PID, control difuso, control neuronal, técnicas de control avanzadas, planta, variador de frecuencia, motor trifásico asíncrono, micro controlador.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Introducción

En la actualidad la mayoría de procesos industriales tienen algún tipo de controlador para funcionar de manera correcta, en muchos casos se utiliza control para optimizar y aumentar la producción puesto que las máquinas con el tiempo sufren desgaste y no tienen el mismo rendimiento inicial, de ahí la importancia de tener un buen sistema de control para optimizar procesos y garantizar la confiabilidad de los mismos, por ejemplo en la industria alimenticia donde las empresas deben garantizar unos estándares de calidad de sus productos (Tecnicontrol) . Pero en sí el uso de un controlador no se limita solamente a la industria, hoy en día es muy común ver electro domésticos que usan algoritmos de control fuzzy, procesos industriales a menor escala, en la academia, en proyectos caseros etc. Esto sin duda ha popularizado el término “control” y cada vez es más frecuente su uso en distintos procesos, de ahí la importancia de conocer el concepto y saber que técnicas son las que se usan y con qué medios se pueden llevar a cabo.

Existen diferentes técnicas de control, la más usada en la industria es el control proporcional, integral, derivativo (PID), aunque con el desarrollo de la computación al día de hoy también es posible implementar otras técnicas como control difuso, control neuronal etc.

En el mercado se encuentran gran variedad de dispositivos que fueron diseñados especialmente para dichas tareas de control, aunque no todas las personas cuentan con los recursos para adquirirlos y no todos se ajustan al proceso en particular que quieren controlar, también se pueden



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

encontrar software que brindan las herramientas necesarias para controlar algún proceso, aunque la mayoría de dicho software son pagos y costosos.

Otro inconveniente que se presenta al hacer uso de un computador para hacer control es la comunicación entre el software y el sistema físico, existen micro controladores que facilitan la comunicación pero en muchos casos nuevamente afecta el factor económico o capacidad del micro para llevar a cabo las tareas de control, añadiendo el problema de tiempos de respuesta ya que se debe tener en cuenta el tiempo que se demora la comunicación entre el computador y el sistema físico, y eso alteraría de alguna forma los parámetros de control que se usen.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Planteamiento del problema

Cuando se va a hacer control de una planta o proceso, se requiere hacer un modelado matemático para determinar su función de transferencia y así analizar a profundidad la planta para determinar la mejor técnica de control. Realizar un modelado matemático por lo general es complicado, puesto que se debe tener en cuenta todos los factores que puedan afectar al proceso como: condiciones ambientales, climáticas, temperatura, entre otros.

Actualmente es común determinar esta función de transferencia por medio de identificación, la cual se hace con datos de entrada y salida del sistema, y por medio del uso de redes neuronales identificar un modelo más exacto de la planta, ya que se puede someter la planta a diferentes factores externos, y estos se verán reflejados en los datos tomados de ella, con el modelo de la planta ya se puede someter a simulaciones y ver su comportamiento frente a distintas técnicas de control.

El control clásico como pid el cual es uno de los más usados en la industria trae consigo algunas limitaciones como: poca efectividad en plantas no lineales, se tiene que sintonizar en un punto fijo de operación. Las máquinas en su funcionamiento real están cambiando sus parámetros periódicamente, por lo tanto, el control pid clásico no podría reaccionar y adaptarse a estos nuevos cambios que tendría la planta. (Villajulca, 2011)

La implementación de técnicas de control avanzadas en una planta de producción o en cualquier máquina, actualmente se ve limitada por el costo del software que brinda librerías y herramientas,



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

esto básicamente representa un obstáculo al momento de presentar una propuesta de implementación de estas técnicas, puesto que por lo general las máquinas ya tienen un control clásico y el balance costo-beneficio no favorecería a la implementación de un control avanzado.

Cabe destacar que los estudios realizados referente a técnicas de control avanzado aplicadas a motores, la mayoría se limitan a simulaciones, pocos han implementado esas técnicas de control a una planta real, esto presenta un inconveniente puesto que en simulaciones no se tendría en cuenta las condiciones físicas del motor, condiciones ambientales, entre otros, que afectarían el rendimiento y los parámetros de cualquier controlador a utilizar.(Pardo, Díaz, & Martinez, 2012)

Justificación

Con base en lo anterior se plantea usar una solución económica para la implementación de estas técnicas, la cual es utilizar un sistema embebido de bajo costo para aplicar estas técnicas de control a un motor trifásico asíncrono.

Para llevar a cabo el objetivo principal de este trabajo, se plantean 3 plataformas diferentes para la programación e implementación de las distintas técnicas de control descritas.

Como primera opción se planea usar un microcontrolador ARM incorporado en una tarjeta teensy 3.6, con una velocidad de 180 MHz, el cual presenta un buen rendimiento y alta velocidad de procesamiento, tiene entradas y salidas analógicas y digitales las cuales se usarán para el control de la planta.(Digi-Key, 2017)



En segunda opción se tiene una Raspberry pi 3, el cual tiene un procesador de 1.4 GHz, más veloz que el microcontrolador ARM y con más accesorios como, conexiones HDMI, Wifi para conexión a internet, puertos USB y una memoria RAM de 1Gb, esta tarjeta cuenta con mejores prestaciones, aunque su precio es más elevado.(Ro-botica, 2017)

Como tercera opción se tiene a la tarjeta de desarrollo Dreamer Maple RET6, la cual consta de entradas tanto análogas como digitales, tiene un procesador de 32 bits a 72Mhz.(Df robot, 2017)

1 Objetivos

1.1 Objetivo General

- ✓ Implementar técnicas de control avanzado en un sistema embebido para el control de velocidad de un motor trifásico asíncrono de la Universidad de Pamplona.

1.2 Objetivos Específicos

- ✓ Seleccionar una plataforma de desarrollo en base al coste y beneficios que preste.
- ✓ Identificar el modelo de la planta (motor- taco generador) empleando la herramienta de identificación de Matlab.
- ✓ Desarrollar los algoritmos que permitan implementar las diferentes técnicas de control en un sistema embebido.
- ✓ Implementar un control PID discreto, un control fuzzy, y un control con redes neuronales artificiales.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

- ✓ Validar el funcionamiento de los algoritmos controlando la planta real usando el material disponible del laboratorio de control de la Universidad de Pamplona.
- ✓ Comparar las diferentes técnicas de control utilizadas.

2 Marco teórico

2.1 Lazos de Control

2.1.1 Control lazo abierto

Un lazo de control es un “arreglo de elementos orientados al mantenimiento de condiciones específicas en un proceso, maquinaria o sistema”. (Núñez, 2009)

Un lazo de control básico consta de ciertos elementos como son: La señal de entrada, o referencia a la cual se espera que la planta llegue, un controlador, la planta o proceso, y por último el actuador como se presenta en Figura 1. (fmeautomatismos, n.d.)

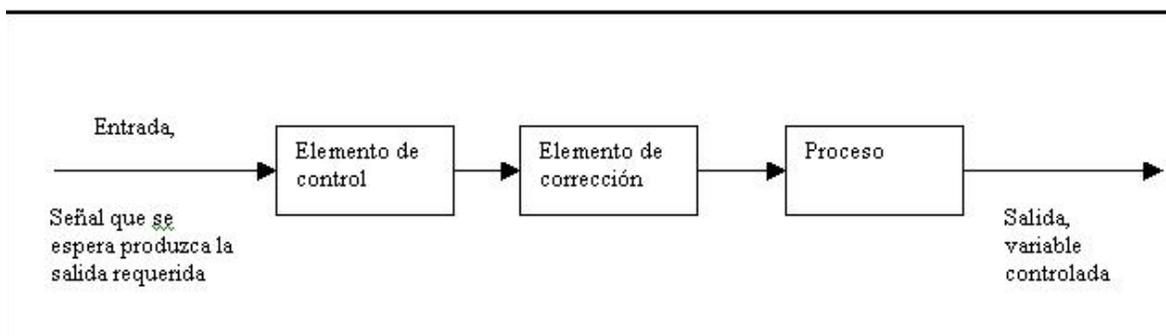


Figura 1 Lazo abierto de control.(fmeautomatismos, n.d.)

Básicamente existen dos tipos de lazos de control, lazo abierto y lazo cerrado, un Control de lazo abierto no tiene realimentación, es decir la salida del sistema no influye en nada en su entrada, un ejemplo de lazo abierto se presenta en Figura 1, en la cual se observa que su salida no interfiere en nada en la entrada del elemento de control. (fmeautomatismos, n.d.)

2.1.2 Control lazo cerrado

Por otro lado, se tiene el lazo de control cerrado, a diferencia del primero en este sistema si se tiene retroalimentación, es decir que la entrada al controlador si depende del valor de salida actual de la planta, el esquema de bloques quedará como se muestra en Figura 2

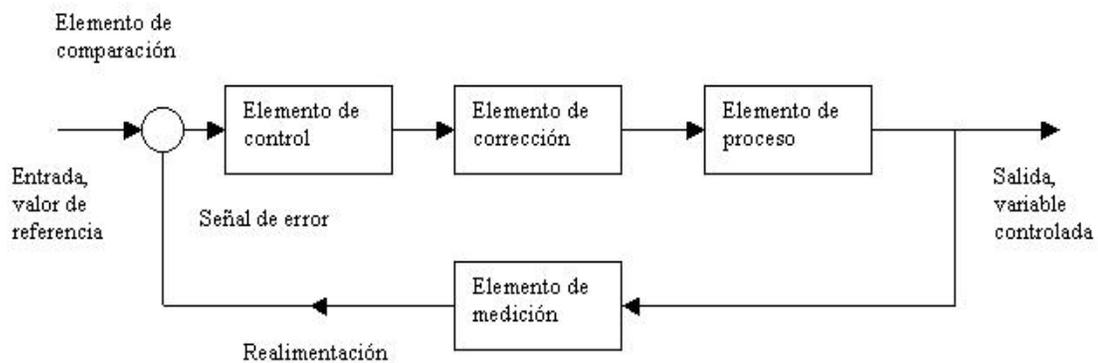


Figura 2 Lazo cerrado de control. (fmeautomatismos, n.d.)

En Figura 2, se observa que al elemento de control está entrando una nueva señal denominada señal de error, esta señal se obtiene al restar el valor de entrada con el valor de salida el cual se obtiene generalmente mediante sensores, el cual en condiciones ideales, si la señal de entrada es idéntica a la de la salida, el error sería cero, esto sería algo muy difícil de conseguir, lo que se busca es que esa señal de error se aproxime a cero. (Rica & S, 2001)

2.2 Control PID

En Figura 2, se observa el bloque de elemento de control el cual regula la señal de entrada a la planta o máquina, en ese bloque se sitúa el controlador, en este caso un PID por sus siglas en ingles

de (proporcional, integral y derivativo). Como su nombre lo da a entender, este controlador incluye tres acciones: La proporcional, integral y derivativa. (Mazzone, 2002)

Hoy en día es uno de los mecanismos de control realimentado más utilizados en la industria, a pesar que su aparición data en los inicios del siglo XX, esto debido al esfuerzo en el desarrollo de nuevos algoritmos de control basadas en estructuras PID. (Rodríguez Rosa, 2012)

A la salida un controlador P , se produce una señal proporcional al error que se esté midiendo, es decir que su salida será la multiplicación de la constante k_p por el error. (Mazzone, 2002)

- Proporcional p: Depende del error actual y modifica la salida en proporción a este.

$$P = K_p * e(t)$$

Ecuación 1. Parámetro proporcional. (Rodríguez Rosa, 2012)

En la acción integral su salida está dada por el error acumulado, por lo tanto, es un control lento, y en cuanto a la acción derivativa tiene un carácter de previsión, lo que quiere decir que hace más rápida la acción de control, la desventaja es que puede amplificar ruidos. (Mazzone, 2002)

- Integral I: Depende de la suma de error acumulado.

$$I = K_i * \int_0^t e(t) dt$$

Ecuación 2. Parámetro integral. (Rodríguez Rosa, 2012)

- Derivativo D: Es la predicción de errores futuros, calcula la variación del error mediante su pendiente en cada instante de tiempo.



$$D = Kd * \frac{d(et)}{dt}$$

Ecuación 3. Parámetro derivativo. (Rodríguez Rosa, 2012)

En la acción proporcional, integral y derivativa (PID) se combinan las ventajas de cada una por lo tanto es el modelo de control más usado. (Mazzone, 2002)

2.3 Tarjetas de desarrollo

“Los microcontroladores constituyen una de las principales áreas de la electrónica aplicada porque facilitan la introducción de los procesadores digitales en numerosos productos industriales” (Mandado Pérez, Menéndez Fuertes, Fernández Ferreira, & López Matos, 2007).

Con el paso de los años y el avance de la tecnología los microcontroladores cada vez son más robustos, muchas veces acompañados de un circuito de acondicionamiento, protección etc que los hacen cada vez más viables para aplicaciones tanto industriales como académicas y de ocio.

2.3.1 Microcontrolador ARM

En la actualidad, casi todo equipo electrónico posee microcontroladores ARM (Advanced RISC Machine), por ejemplo, computadores, celulares, electrodomésticos, equipos industriales de comunicación, medicina, industria automovilística, etc. Casi todas las empresas fabricantes de microcontroladores producen microcontroladores ARM, por ejemplo, Intel, Texas Instruments,

NXP, ST, Atmel, Freescale, Etc. Por estos motivos, es muy importante conocer y programar estos microcontroladores. (Pérez, 2017)

Microcontroladores de 8, 16, 32 bits son iguales en lo que se refiere a su funcionamiento, su principal diferencia está en la cantidad de información que manejan. (Pérez, 2017)

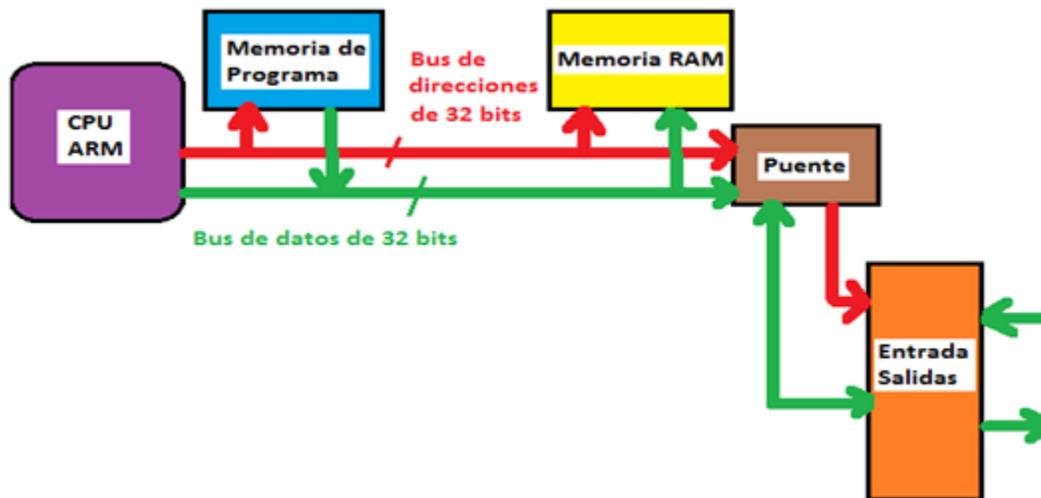


Figura 3 Arquitectura de un microcontrolador ARM.(Pérez, 2017)

En Figura 3 se puede observar la arquitectura común de un microcontrolador ARM. Lo importante aquí es resaltar que las entradas y salidas del sistema pasan primero por un puente (Bridge), esto es porque el microcontrolador procesa instrucciones a una gran velocidad, y los periféricos de entrada y salida necesitan una velocidad más baja para funcionar, así se puede notar que la memoria de programa y la memoria RAM están conectados directamente al procesador y los periféricos a través del puente.(Pérez, 2017)



2.3.2 Tarjeta de desarrollo teensy

La Teensy 3.6 es una placa de desarrollo de protoboard con numerosas funciones, diseñada por PJRC y previamente cargada con un gestor de arranque. Esto permite fácil programación usando conexión USB en placa sin necesidad de un programador externo. Teensy se puede programar en C o mediante el IDE de Arduino instalando el complemento Teensyduino. (Digi-Key, 2017)

El microcontrolador Kinetis K66 en Teensy también tiene la capacidad de emular a cualquier tipo de dispositivo USB, lo que es ideal para USB-MIDI y otros proyectos HID. El microcontrolador de 32 bits y 180 MHz ofrece múltiples canales de acceso directo de memoria, varios ADC de alta resolución y una interfaz de audio digital I2S. Hay cuatro temporizadores de intervalo separado, además de un temporizador de retardo. Todos los pines digitales tienen capacidad de interrupción y operan a niveles de lógica de 3,3 V. (Digi-Key, 2017)

Todas estas características se incluyen en la placa de 62,3 mm x 18,0 mm con todos los puntos de soldadura en una cuadrícula de 0,1" compatible con protoboard. La placa también ofrece un puerto de tarjeta SD micro a bordo. Un ARM® Cortex-MCU mejorado (180 MHz de 72 MHz) y más memoria (a 1 M de 256 K), así como más RAM, EEPROM y pines accesibles constituyen las principales características de esta placa "Teensy" en relación con la anterior Teensy 3.2. Teensy 3.6 es una ampliación de Teensy 3.5 y es una placa completo destacada en la línea Teensy. (Digi-Key, 2017)

En Figura 4 se puede observar la tarjeta de desarrollo teensy en su versión 3.6

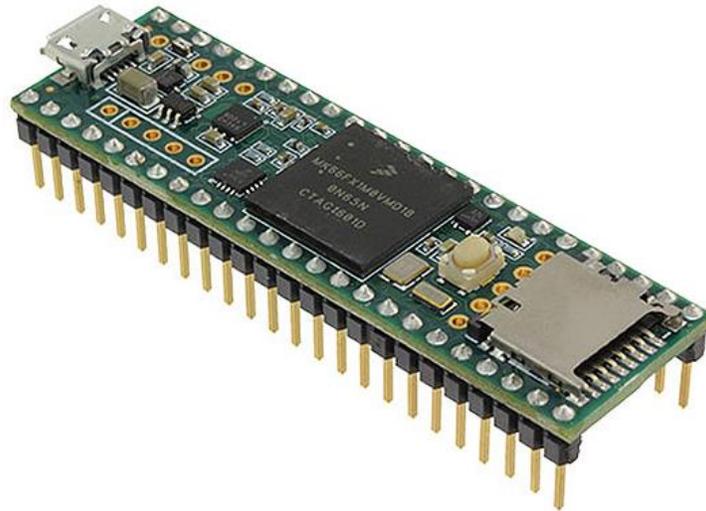


Figura 4 Tarjeta de desarrollo teensy 3.6 (Digi-Key, 2017)

2.3.3 Raspberry PI 3

Raspberry es una placa computadora de bajo coste desarrollada en el Reino Unido por la Fundación Raspberry pi, con el objetivo de estimular la enseñanza de la informática en las escuelas. La placa Raspberry Pi 3 modelo B es la tercera generación de Raspberry Pi y viene con una presentación impecable, dentro de su caja y envuelta en un sobre con el logo de Raspberry Pi.(Ro-botica, 2017)

Muchos sistemas operativos Linux han sido optimizados para la Raspberry Pi y además para darle cualquier uso a esta placa se tiene un sistema operativo muy versátil es el RASPBIAN OS que se puede descargar de la página oficial.

El archivo descargado tiene extensión “.img” y se tiene que guardar en la tarjeta SD de la Raspberry Pi. Para ello hay que utilizar Win32 Disk Imager, que se puede descargar de la página oficial, para finalmente insertar la tarjeta SD en la raspberry y proceder a instalar el sistema operativo y colocar en funcionamiento la tarjeta. (Ro-botica, 2017)

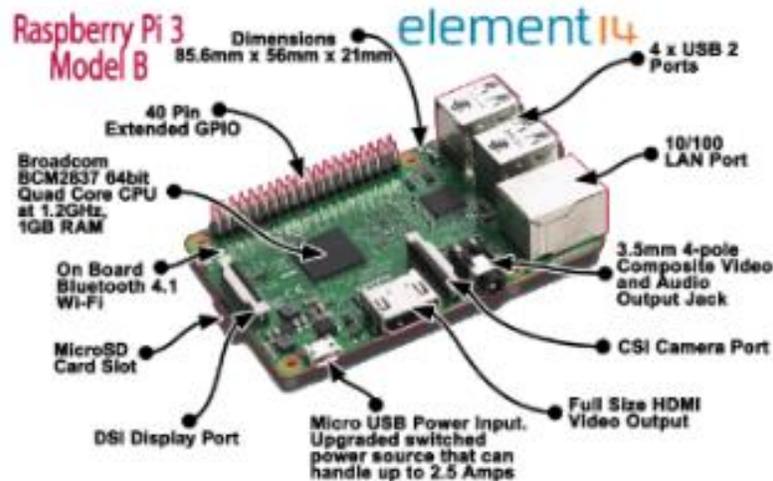


Figura 5 Tarjeta raspberry con sus componentes. (Ro-botica, 2017)

Como se puede observar en Figura 5, esta tarjeta tiene una serie de componentes con los cuales se puede aplicar el proyecto descrito.

Especificaciones técnicas:

- CPU Quad Core 1.2GHz Broadcom BCM2837 de 64 bits
- 1 GB de RAM



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

- BCM43438 wireless LAN y Bluetooth Low Energy (BLE) a bordo
- GPIO extendido de 40 pines
- 4 puertos USB 2
- Salida estéreo de 4 polos y puerto de vídeo compuesto
- HDMI de tamaño completo
- Puerto de cámara CSI para conectar una cámara Raspberry Pi
- Puerto de pantalla DSI para conectar una pantalla táctil Raspberry Pi
- Puerto Micro SD para cargar su sistema operativo y almacenar datos
- Fuente de alimentación Micro USB conmutada mejorada hasta 2.5A (Ro-botica, 2017)

2.3.4 Dreamer maple

El Maple es una placa de microcontrolador basada en el microprocesador STM32F103RET6. El Maple funciona a un máximo de 72 MHz, tiene 43 pines de entrada / salida digital, 15 entradas analógicas, USB nativo de velocidad completa, 3 USART (puertos serie de hardware), soporte SPI / I2C integrado, un conector de alimentación y un botón de reinicio. Maple es programable a través de USB a través del gestor de arranque DFU provisto, no se requiere hardware adicional. Los usuarios también pueden programar a través de la interfaz JTAG externa. Maple ofrece soporte de seguridad para direcciones protegidas de lectura / escritura, así como modos de procesador de usuario y controlador. Tiene un sysTick en tiempo real, útil para un sistema operativo en tiempo



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

real o cualquier aplicación sensible al tiempo. 6 temporizadores de interrupción avanzados que serán de gran ayuda. (Df robot, 2017)

La tarjeta se puede alimentar a través de USB o puede funcionar con una batería lipo recargable. El Maple es compatible con escudos diseñados para Arduino Duemilanove o Diecimila. Maple es una excelente manera de comenzar con un procesador avanzado de 32 bits que, hasta ahora, ha vivido principalmente en el dominio comercial. Se puede optimizar los proyectos con el reloj rápido de Maple, la sofisticada arquitectura de interrupción y una serie de periféricos incorporados. Maple se puede programar utilizando un entorno de programación de estilo Arduino, basado en bocetos, que es de código abierto y se puede descargar en línea.(Df robot, 2017)

Especificaciones:

- STM32F103RET6: un microprocesador ARM Cortex-M3 de 32 bits a 72Mhz
- Voltaje de funcionamiento: 3.3 V
- 64KB RAM y 512KB Flash
- Puerto USB dedicado para programación y comunicaciones.
- 2 temporizadores avanzados, 4 de propósito general y 2 básicos
- 15 entradas analógicas a resolución de 12 bits
- 18 pines PWM a resolución de 16 bits
- 2 periféricos SPI con soporte I2S

- 2 periféricos I2C
- 3 puertos serie USART
- 12 canales de acceso directo a memoria (DMA) (dma.h) con 2 controladores DMA
- Baja potencia y modos de reposo.
- Voltaje de entrada (recomendado): 7 V - 12 V
- Apoyo a la interfaz inalámbrica XBEE (Df robot, 2017)

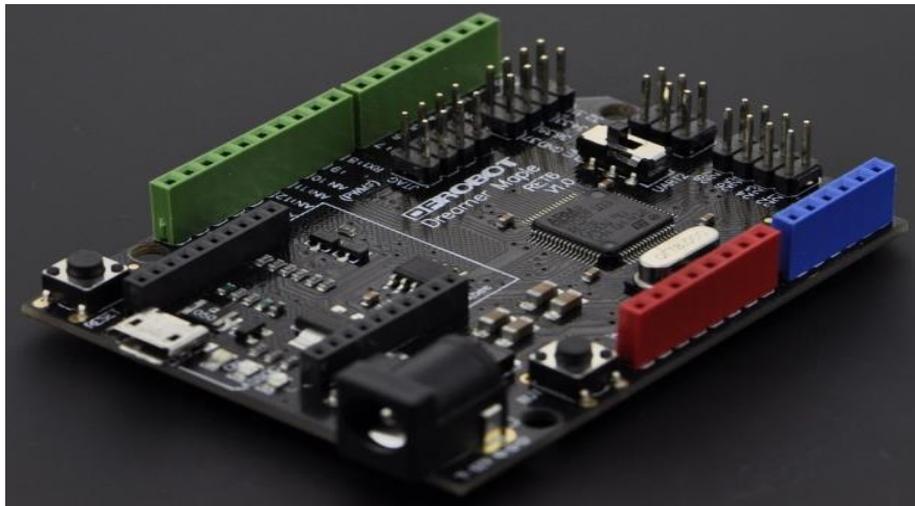


Figura 6 Vista general de la tarjeta de desarrollo maple. (Df robot, 2017)

En Figura 7 se pueden observar todos los pines y periféricos de los que consta esta tarjeta de desarrollo como los pines digitales GPIO, pines PWM (Modulación por ancho de pulso), entradas análogas etc.

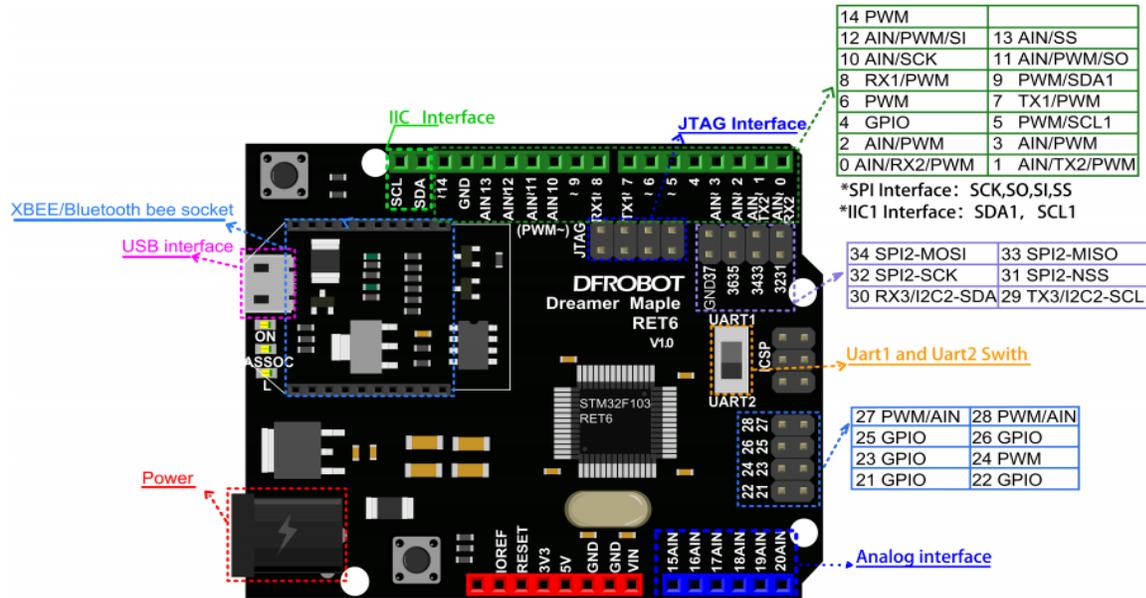


Figura 7 Pines y perifericos de la tarjeta maple. (Df robot, 2017)

2.4 Técnicas de control avanzado

Existen diferentes formas de control automático que responden a este tipo de denominación. En general, todas se basan en un conocimiento más o menos acabado del sistema a controlar, tanto de su fenomenología como de sus condiciones de operación, a través de un modelo matemático que lo describa en forma bastante aproximada, en todo su rango de operación y para cualquier instante de tiempo (situación relevante en procesos por lotes o “batch”). Este modelo se usa tanto como parte integrante del sistema de control como para evaluar el desempeño del sistema a través de técnicas de simulación. (Fernandez, n.d.)

Sin duda, y siguiendo la definición dada de control avanzado, existen diversidad de estrategias de control que responden a esta clasificación, principalmente provenientes del mundo académico-



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

científico. Sin embargo, la dificultad para poder ser operadas por personal con escaso nivel de formación profesional, así como para generalizarlas a sistemas de diferente naturaleza, ha provocado que pocas sean las que han trascendido del ámbito académico al industrial. A continuación, se dará una definición de algunas de estas técnicas que se usarán en el desarrollo de este proyecto. (Fernandez, n.d.)

2.4.1 Control difuso

Este tipo de control se basa en la recopilación de conocimiento de un sistema, a partir de operadores del mismo que pueden considerarse como “expertos” en su área de conocimiento. Este conocimiento se transforma en una serie de reglas del tipo “SI x_1, x_2, \dots, x_n , ENTONCES $y_1, y_2, \dots, y_m, \dots$ ” (“IF ..., THEN ...”), donde “ x_1, x_2, \dots, x_n ” representan los antecedentes o premisas e “ y_1, y_2, \dots, y_m ” las consecuencias o acciones. (Fernandez, n.d.)

Este tipo de control se basa en la aplicación de un álgebra difusa planteada por Lofti Zadeh, al tratar de representar el pensamiento lógico humano. Para utilizar esta lógica, las variables controladas y observadas que provienen de los distintos sensores presentes en el proceso, cuya información será utilizada por el sistema de control, debe “pasarse” al mundo “difuso” a través de un procedimiento conocido como “fuzzificación”, teniendo en cuenta el grado de pertenencia que tiene cada variable a los conjuntos difusos en que se subdivide el rango de aplicación de cada variable, a través de la definición de funciones de pertenencia asignadas a las mismas. Cuando las variables están ya definidas en forma difusa, se contrasta la información del sistema con la base



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

de conocimiento disponible del proceso y, mediante aplicación de álgebra difusa, se determina la acción de control requerida para el sistema, pero con carácter “difuso”. (Fernandez, n.d.)

Para entregar esta información al mundo real, la acción de control debe ser “defusificada” (defuzzification”), y entonces estará disponible para ser aplicada a los actuadores del proceso.

Entre las principales ventajas de esta técnica puede destacarse la simpleza de su implantación, la extensión directa que tiene para aplicarse tanto a sistemas SISO como MIMO y, en este último caso, sin requerir un mismo número de entradas que de salidas. Además, como se basa en una estructura de información similar a la del pensamiento humano, la forma de proceder tiene muy buena aceptación por parte de los operadores del sistema. (Fernandez, n.d.)

Por ejemplo, considere un sistema de frenos antibloqueo, dirigido por un chip de microcontrolador. El microcontrolador tiene que tomar decisiones basadas en la temperatura del freno, la velocidad y otras variables en el sistema.(Greg, 2013)

La variable "temperatura" en este sistema se puede dividir en un rango de "estados", tales como: "frío", “muy frío”, "moderado", "cálido", "caliente", "muy caliente". Definir los límites de estos estados es un poco complicado. Un umbral arbitrario podría configurarse para dividir "caliente" de "cálido", pero esto daría como resultado un cambio discontinuo cuando el valor de entrada pasara por encima de ese umbral. (Greg, 2013)

La forma de evitar esto es hacer que los estados estén "difusos", es decir, permitirles cambiar gradualmente de un estado a otro. Puede definir los estados de temperatura de entrada usando "funciones de membresía o pertenencia" como se observa en Figura 8:

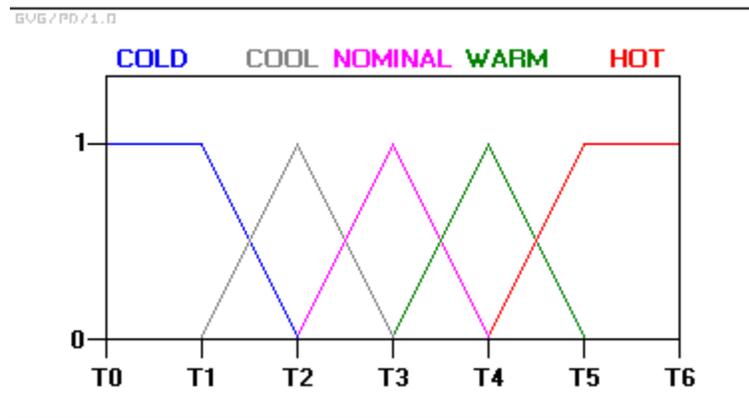


Figura 8 Funciones de pertenencia. (Greg, 2013)

Con este esquema, el estado de la variable de entrada ya no salta bruscamente de un estado al siguiente. En cambio, a medida que cambia la temperatura, pierde valor en una función de membresía y gana valor en la siguiente. En cualquier momento, el "valor de verdad" de la temperatura del freno será casi siempre parte de las funciones de dos miembros: 0.6 nominal y 0.4 caliente, o 0.7 nominal y 0.3 frío, y así sucesivamente. (Greg, 2013)

Las variables de entrada en un sistema de control difuso son mapeadas en general por conjuntos de funciones de pertenencia similares a estas, conocidas como "conjuntos difusos". El proceso de convertir un valor de entrada nítido en un valor difuso se denomina "fuzzificación". (Greg, 2013)



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Dadas las "asignaciones" de variables de entrada en funciones de pertenencia y valores de verdad, el microcontrolador toma decisiones sobre qué acción tomar en función de un conjunto de "reglas", cada una de las formas:

SI la temperatura del freno **ESTÁ caliente** Y la velocidad **NO** es muy **rápida**

ENTONCES la presión de los frenos ha **disminuido ligeramente**

En este ejemplo, las dos variables de entrada son "temperatura de frenado" y "velocidad" que tienen valores definidos como conjuntos difusos. La variable de salida, "presión de frenado", también se define por un conjunto difuso que puede tener valores como "estático", "ligeramente aumentado", "ligeramente disminuido", etc. Recuerde que este es solo un ejemplo, en la realidad la decisión final y el valor de salida se definen en base a un conjunto de reglas. (Greg, 2013)

Ahora este es el sistema de control avanzado que ha tenido mejor aceptación entre sus exponentes, principalmente por la industria japonesa, y se encuentran aplicaciones de estrategias de control basadas en lógica difusa en el ajuste automático del foco en cámaras fotográficas, resolución del sistema de frenos ABS en la industria automotriz, lavadoras automáticas, etc. Inclusive, las principales empresas fabricantes de PLC's incorporan módulos de control de lógica difusa, y las plataformas SCADA o DCS también los están incluyendo como herramientas de control. (Fernandez, n.d.)



2.4.2 Redes neuronales

Las redes neuronales imitan la estructura y la función del cerebro humano mediante el procesamiento de información no lineal y la computación paralela a gran escala con una capacidad de almacenamiento distribuido. Cuando se trata con varios tipos de información, las redes neuronales aprenden por sí mismas y ajustan constantemente sus sistemas para adaptarse a un entorno variable. Por lo tanto, se usan ampliamente en el reconocimiento de patrones, para predecir cambios en las cosas, optimizar decisiones, mejorar el control del proceso y otras tareas. (Ye & Kim, 2018)

El uso del método de red neuronal tiene varias ventajas, por ejemplo, cuando se aplica al pronóstico del consumo de electricidad. Primero, puede imitar al cerebro humano para obtener información para el procesamiento inteligente. En segundo lugar, puede adaptarse a muchas reglas irregulares e inexactas. Las redes neuronales también tienen la capacidad de memorizar información, así como el autoaprendizaje, el razonamiento del conocimiento y la optimización de la información. La auto adaptabilidad es la característica más importante de las redes neuronales en comparación con otros algoritmos. Por lo tanto, las redes neuronales artificiales pueden tener un papel importante en la predicción del consumo de electricidad en los edificios. Si los datos de consumo de electricidad se pueden recopilar de un edificio completo, se puede utilizar una red neuronal para calcular el consumo de electricidad en el edificio durante la próxima temporada o el próximo mes. (Ye & Kim, 2018)



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Se han propuesto varios tipos de redes neuronales. En la actualidad, los cuatro tipos principales de modelos básicos y sus modelos mejorados son la red neuronal Hopfield, la red neuronal BP, la red neuronal RBF y los mapas Kohonen. Las redes neuronales de BP se han aplicado principalmente en áreas como reconocimiento de patrones, procesamiento de información, procesamiento de imágenes, detección de fallas, control inteligente y gestión empresarial. Las mejoras en las redes neuronales de BP han considerado principalmente la estructura de la red y el algoritmo de aprendizaje. Los algoritmos propuestos con estructuras de red mejoradas incluyen el método de impulso adicional, tasa de aprendizaje gradualmente ley pequeña y algoritmo BP elástico, y tasa de aprendizaje adaptativo. Los algoritmos propuestos para mejorar el algoritmo de aprendizaje incluyen el método de gradiente conjugado, el método Cuasi-Newton y el método de optimización LM. (Ye & Kim, 2018)

El algoritmo LM es una red neuronal utilizada en un algoritmo de entrenamiento no lineal, donde combina el método de descenso de gradiente y el método Quasi-Newton para garantizar la velocidad de convergencia localmente rápida y mantener un mejor rendimiento general. La idea básica de este algoritmo es que cada iteración no es más larga de lo necesario a lo largo de un solo gradiente en la dirección negativa y permite buscar el error en la dirección de descenso. Además, los pesos de la red pueden optimizarse mediante el ajuste adaptativo entre el método de descenso de gradiente más pronunciado y el método de Gauss-Newton, que permite que la red converja de



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

manera efectiva y esto mejora en gran medida la velocidad de convergencia y la generalización de la red. (Ye & Kim, 2018)

Las redes neuronales artificiales pertenecen a un grupo de técnicas de procesamiento de información que se pueden utilizar para encontrar conocimiento, patrones o modelos a partir de una gran cantidad de datos. Además, se pueden lograr ventajas intrigantes combinando redes neuronales artificiales con otros modelos computacionales (FDM, FEM, FVM), que pueden proporcionar los datos para entrenar la red neuronal artificial para crear el modelo. Los algoritmos genéticos se pueden usar de manera eficiente para encontrar una solución adecuada a un problema complejo de optimización. Los algoritmos genéticos son una herramienta eficaz que está disponible para explorar espacios de búsqueda grandes y complejos basados en la teoría de Darwin de supervivencia del más apto. En los últimos años, se ha presentado mucho trabajo combinando algoritmos genéticos y redes neuronales artificiales en el contexto del desarrollo de métodos híbridos. Los principales enfoques que se han utilizado en ese sentido son los siguientes: usar algoritmos genéticos para mejorar el aprendizaje de redes neuronales artificiales, y usar algoritmos genéticos para realizar una búsqueda global y optimizar las entradas de la red neuronal modelo versus los resultados de salida creados por una red. (Aleksendrić, Carlone, Aleksendrić, & Carlone, 2015)

Con las Redes Neuronales Artificiales (RNA's) se busca la solución de problemas complejos, no como una secuencia de pasos, sino como la evolución de sistemas de computación inspirados en



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

el cerebro humano, y dotados por tanto de cierta "inteligencia", los cuales no son sino la combinación de elementos simples de proceso (neuronas) interconectados, que operando de forma paralela, consiguen resolver problemas relacionados con el reconocimiento de formas o patrones, predicción, codificación, control y optimización, entre otras aplicaciones (Marulanda, 2010), en este trabajo se enfoca más al uso de redes neuronales para el control de procesos.

Una red neuronal artificial es capaz de aprender por si misma para controlar sistemas dinámicos no lineales (es decir que sus parámetros cambian conforme pasa el tiempo), esto es quizá una de las razones por las que más se utiliza este tipo de control. Para el control de un sistema dinámico se requiere un emulador el cual es una red neuronal de múltiples capas, que aprende a identificar las características dinámicas del sistema. Un controlador, otra red neuronal multicapa, que aprende a controlar al emulador. El controlador autocapacitado se usa para controlar el sistema dinámico real. El proceso de aprendizaje continúa a medida que el emulador y el controlador mejoran y rastrean el proceso físico el cual es el cálculo constante de las características dinámicas del sistema o los parámetros de la planta real. (Nguyen & Widrow, 1990)

Las redes neuronales artificiales quedan integradas dentro de las técnicas conexionistas, lo cual significa que el funcionamiento de cada uno de sus componentes elementales es idéntico, y la complejidad funcional se obtiene mediante una gran interconectividad entre estos elementos. Al igual que su homóloga biológica, una neurona artificial recibe información, bien sea de sensores o de otras neuronas; posteriormente realiza unas operaciones simples, y finalmente transmite el resultado a otras neuronas vecinas. La ventaja de las redes neuronales artificiales frente a sistemas matemáticos o expertos es que la función gana complejidad cuanto mayor es el número y las combinaciones de estas. Al igual que las neuronas biológicas, la muerte o deterioro de una neurona afecta cuantitativamente, pero no cualitativamente. Esto les confiere características que las hacen muy adecuadas para la realización de tareas tales como identificación, reconocimiento de patrones y sobre todo control. (Marulanda, 2010)

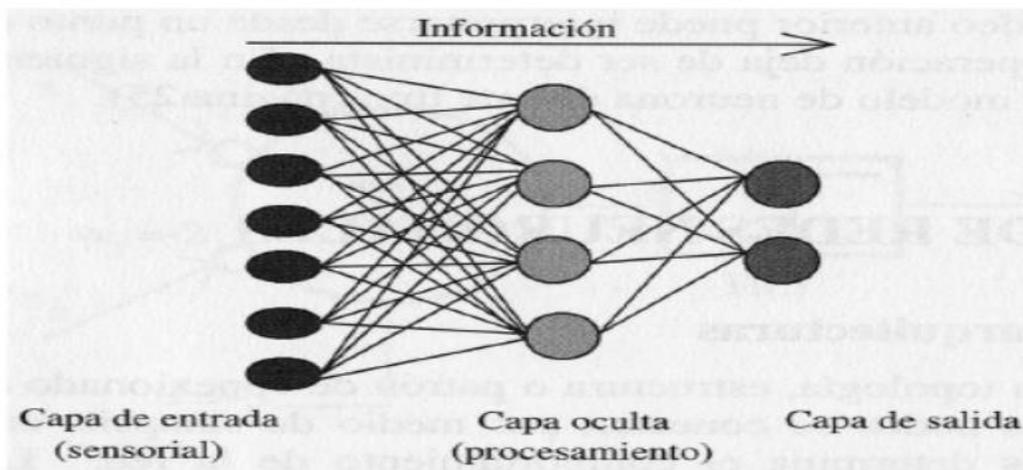


Figura 9 Estructura de una red neuronal artificial multicapa (RNA). (Marulanda, 2010)



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

En Figura 9 se puede observar la estructura básica de una red neuronal artificial. se tiene 3 capas, en la capa de entrada por lo general se tienen sensores, e incluso salidas de las mismas neuronas o las salidas de la red pueden servir como entradas de las mismas, también retardos de estas mismas, es decir salidas anteriores. La capa oculta que pueden ser más de una capa, y la capa de salida de la red, la cual es la salida general de la red neuronal.

Las redes neuronales han sido muy utilizadas últimamente en el control de sistemas con dinámica difícil o sistemas que cambian de parámetros constantemente, como sistemas con altas no linealidades, es por ello que existen dos clases de redes neuronales que han recibido importancia en el área de inteligencia artificial en los últimos años, por una parte, se tiene las redes neuronales multicapa y las redes recurrentes. Las redes multicapa han demostrado ser extremadamente exitosas en problemas de reconocimiento de patrones, mientras que las redes recurrentes se han utilizado en memorias asociativas, así como para la solución de problemas de optimización. Desde el punto de vista teórico de los sistemas, las redes multicapa representan mapas no lineales estáticos, mientras que las redes recurrentes están representadas por sistemas de retroalimentación dinámica no lineal. A pesar de las aparentes diferencias entre las dos clases de redes, existen razones convincentes para verlas de manera unificada. De hecho, los elementos dinámicos y la retroalimentación se utilizarán cada vez más en el futuro, lo que dará como resultado sistemas complejos que contengan ambos tipos de redes. Esto, a su vez, requerirá un tratamiento unificado de tales redes. (Narendra & Parthasarathy, 1990)



2.5 Motores trifásicos asíncronos

Los motores asíncronos trifásicos, también llamados de inducción son los más baratos de construir, los de menor mantenimiento y los más usados en la industria. Existen dos tipos, según que el rotor sea bobinado (motor de anillos rozantes) o en cortocircuito (motor de jaula de ardilla). Estos últimos son los más comunes en la industria, sobre todo porque requiere poco mantenimiento al carecer de anillos rozantes o escobillas que estén sujetas a permanente desgaste. (Etolocka, 2014)

Como todo motor, consta básicamente de dos partes, una fija denominada estator, que es la encargada de generar el campo magnético giratorio a través de una serie de bobinados situados a 120° uno del otro, y otra que gira en su interior llamada rotor. (Etolocka, 2014)

En el estator se tienen 3 devanados distribuidos en ranuras a 120° separados entre sí, en el rotor se dijo que hay dos tipos, en el rotor bobinado los devanados son similares a los del estator con el que está asociado, los devanados del rotor están conectados a anillos rozantes montados sobre el mismo eje, por eso también se llaman rotor bobinado o de anillos rozantes. Para el rotor jaula de ardilla los conductores del rotor están igualmente distribuidos por la periferia del rotor, los extremos de estos conductores están cortocircuitados por tanto no hay posibilidad de conexión del devanado del rotor con el exterior. (FRMza, n.d.)



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

La velocidad del campo magnético giratorio del estator (llamada velocidad síncrona o velocidad de sincronismo) depende de la cantidad de polos del motor y de la frecuencia de la tensión de alimentación. Esto puede expresarse matemáticamente como se observa en Ecuación 4

$$N_s = \frac{60 * f}{p}$$

Ecuación 4. Velocidad de sincronismo de un motor trifásico

Donde NS es la velocidad de sincronismo, f la frecuencia de la red y P el número de polos de la máquina. (Etolocka, 2014)

Esto significa que el motor no puede girar a una velocidad cualquiera, sino que existen unos valores predeterminados por la frecuencia de la tensión de alimentación y las características constructivas.

La Tabla 1 muestra algunos valores para las cantidades de polos más usuales de motores que se encuentran en el mercado.

N° de polos	Rotación síncrona por minuto	
	60 Hertz	50 Hertz
2	3600	3000
4	1800	1500
6	1200	1000
8	900	750
10	720	600

Tabla 1. Velocidades de sincronismo. (Etolocka, 2014)



En el motor asíncrono, el rotor siempre gira más despacio que el campo magnético del estator, esta velocidad se llama asíncrona y la diferencia entre ambas se denomina deslizamiento (S). Generalmente los valores del deslizamiento están entre el 2% y el 4%. (Etolocka, 2014)

3 Estado del arte

En un estudio publicado llamado “Modeling of Photovoltaic Array and Maximum Power Point Tracker Using ANN”, los desarrolladores implementan una red neuronal para encontrar el modelo matemático de un panel solar usando datos reales del panel como datos de entrada para la predicción, luego usan datos climáticos históricos para entrenar la red y la salida que obtienen con respecto a datos reales del panel tienen menos del 10% de error según sus conclusiones. (Kamath, Aithal, Kumar, & Danak, 2012)

Por otra parte, se usa un controlador de redes neuronales para determinar la mejor precisión y aprovechamiento de la energía de un panel solar, creando una red neuronal dinámica, es decir con capacidad de cambiar sus neuronas o conexiones dependiendo de lo que se requiera. (Fernández, 2015)

También se ha usado un motor de corriente trifásica para hacer el análisis de su comportamiento frente a distintos tipos de controladores en simulación usando el software simulink, desde un control PI (proporcional integral), control PID (proporcional, integral y derivativo), al cual se le aplicaron velocidades de referencia y posteriormente una perturbación, siendo estas técnicas de control



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

clásicas poco eficaces para responder ante perturbaciones según sus conclusiones. (Pardo et al., 2012) Luego mediante el uso de redes neuronales, clonaron el control PID clásico, una vez implementado este modelo neuronal la planta presentó un mejor rendimiento y una rápida reacción ante las perturbaciones siendo capaz de llegar a su valor de referencia, este modelo PID clonado con redes neuronales es el modelo que los desarrolladores recomiendan. (Pardo et al., 2012)

Por otro lado, se han implementado redes neuronales para determinar la velocidad de giro de un motor de inducción, con el fin de eliminar sensores de velocidad, en su reemplazo se usa sensores de corriente tanto del estator, como del rotor, usando los valores de corrientes y entrenando redes neuronales se hace una estimación de la velocidad del motor. (Brandstetter & Kuchar, 2017) Para ello se usa el software Simulink y primero se implementa una red neuronal tipo LMP o de perceptrón multicapa, pero no se tuvo la mejor estimación final de velocidad pues presenta fluctuaciones importantes a su salida, posteriormente se usa una RBF la cual es una red neuronal de base radial que tiene la particularidad que solo tiene 3 capas, la de entrada, una capa oculta y una de salida, los desarrolladores obtuvieron mejores resultados con la implementación de este tipo de red, aunque a un mayor costo computacional como especifican sus conclusiones.(Brandstetter & Kuchar, 2017)

Para el control de torque se plantea un control de par directo, generalmente en este control se usa un PI (proporcional e integral) para el control de par y velocidad, pero el problema es que el control PI debe estar bien definido, y muchas veces es complicado y no se llega a las constantes ideales



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

para el controlador. Los desarrolladores sugieren un cambio del controlador clásico por un ANFIS (Neuro difuso) y redes neuronales, las cuales trabajando en conjunto determinan los tiempos de muestreo para los cuales se aplican el estado de conmutación, importante en este tipo de control, obteniendo mejores resultados en tiempos transitorios como en estables. (H., S.F., & B., 2018)

En un estudio publicado en IEE se aborda la problemática que tienen un control convencional ante algunos factores como: sistemas con zonas muertas y saturaciones en los mecanismos del actuador, incertidumbre del modelo o parámetros y ruidos en el proceso, también se habla sobre el control coordinado de múltiples robots. Por ejemplo, en un sistema de múltiples robots, cada robot es un dispositivo independiente, cuando estos robots sostienen un objeto sólido a pesar que son independientes, una pequeña falla en la coordinación de ellos puede dañar el objeto, incluso hasta los mismos robots. Es por esto que los autores proponen usar redes neuronales como control adaptativo para solucionar todos los inconvenientes mencionados. Por lo general una red neuronal se entrena con los valores de salida de ella misma, pero para este caso eso no sería viable puesto que no se conoce los valores deseados de la red neuronal, es decir no se sabe cuál es la señal de control que necesita la planta a controlar, es por ello que utilizan los errores de salida de la planta para entrenar la red neuronal. El único conocimiento a priori que se tiene de la planta es la dirección de su respuesta de salida y con esto se puede encontrar un modelo neuronal para controlar dicho proceso. En el estudio se hace entrenamiento de la red neuronal, aunque se puede hacer que la red



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

se esté adaptando en todo momento y omitir la etapa de entrenamiento previo. (Xianzhong Cui & Shin, 1993)

En el instituto tecnológico de Querétaro en México, se llevó a cabo un estudio con RNA (redes neuronales artificiales) en el que abordan la problemática de las redes neuronales en el campo de control, y es que se necesitan tiempos de ejecución cortos y evitar los procesos de entrenamiento exhaustivo de las redes que resulta un impedimento para usar esta técnica en el campo industrial. Los autores proponen un esquema de red neuronal en el que se omite la parte de entrenamiento previo y se sustituye por una retro propagación del error continua, es decir que la red se estará entrenando continuamente, de este modo se aseguraría de que la red funcione como control adaptativo, pues estaría calculando en cada iteración una señal de control óptima para el error en cada instante de tiempo. Otra particularidad a destacar en este trabajo es que la red no se entrena con el error calculado a la salida de la red neuronal, sino que se toma la salida de la planta y el setpoint para calcular el error como se observa en

Figura 11 en donde "ye" que es el error se calcula con la resta entre "yt" (entrada) y la salida de la planta "y", de esta forma dentro de la red neuronal estaría quedando inmersa la dinámica de la planta a pesar de no conocerla, la red calcularía los valores de control necesarios para que esa diferencia entre el setpoint y la salida de la planta converja a cero. Como conclusión, se pueden solucionar las limitaciones de control convencional como: la aplicación de sistemas MISO o MIMO múltiples entradas múltiples salidas, a sistemas no lineales con zonas muertas o

saturaciones y otras más, aplicando redes neuronales que funcionan como un control adaptativo y no es necesario conocer la dinámica de la planta para poder controlarla. (Noriega, Barrera, & Ordez, 2008)

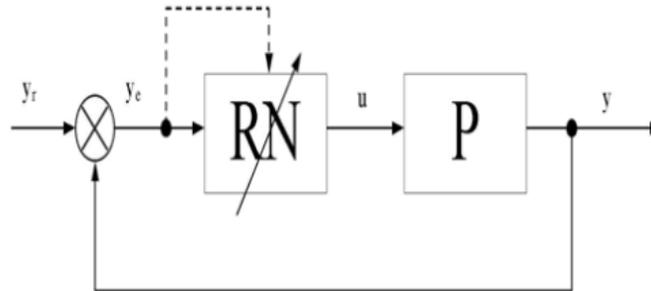


Figura 11 Esquema control neuronal auto-ajustable. (Noriega et al., 2008)

Al final los autores como conclusión recalcan la eficiencia e importancia de implementar esta opción en sistemas con altas no linealidades, con cambios constantes en sus parámetros y dinámica difícil, lo cual es frecuente encontrarse con este tipo de sistemas. (Noriega et al., 2008)

4 Capítulo 1 Reconocimiento de la planta.

4.1 Componentes

El presente trabajo se desarrolló dentro de los laboratorios de la Universidad de Pamplona, por lo tanto, se dispuso de los componentes necesarios para llevarlo a cabo, la lista de los elementos que componen el sistema o planta se los puede apreciar en Tabla 2.

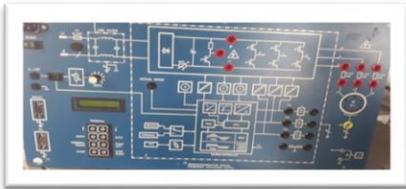
Nombre	Descripción	Imagen
Motor trifásico	Motor de corriente alterna funcionando a 400V en configuración delta.	
Variador de frecuencia	Variador con un rango de frecuencia desde 0.5Hz hasta 120Hz, tiene entrada de voltaje directo como referencia.	
Taco- Generador	Motor de corriente directa adecuado como taco – generador, es decir que genera un voltaje de salida dependiendo de la velocidad a la que esté girando.	

Tabla 2. Componentes de la planta a utilizar. (Autor)

4.2 Descripción de componentes

4.2.1 Motor trifásico asíncrono.

Como planta a controlar se escogió un motor trifásico asíncrono que funciona a 400 voltios en configuración delta y a 600 voltios en configuración estrella. Para este trabajo se escogió configuración delta puesto que tiene un tablero con las borneras para elegir la configuración como se observa en Figura 12. También tiene una velocidad nominal de 1440 revoluciones por minuto, con una potencia de 0.37Kw.



Figura 12 Motor trifásico asíncrono. (Autor)

4.2.2 Variador de frecuencia

El variador de frecuencia, permite controlar la velocidad de un motor trifásico de corriente alterna mediante el cambio de frecuencia en su salida, de esta forma variará la velocidad de sincronismo expresada en Ecuación 4 la cual hace girar a distintas velocidades el campo magnético giratorio, que a su vez también altera la velocidad con la que gira el rotor, que en este caso será la variable de salida a controlar.

El variador usado para este proyecto es de tipo académico el cual se conecta a 220 voltios alternos, cuenta con un interruptor de encendido y apagado. Se puede variar la frecuencia por dos formas, ya sea por medio de un potenciómetro que viene incorporado, o por una señal de referencia en voltaje, que va desde los 0 voltios hasta los 10 voltios.

Cuenta con una pantalla LCD para configurar sus parámetros principales como son: Frecuencia nominal, frecuencia mínima y frecuencia máxima, activar o desactivar el potenciómetro, activar o desactivar sus salidas ttl etc. También tiene los pines de conexión de las tres fases que van conectadas al motor, y unos pines de protección contra sobrecalentamiento del motor.

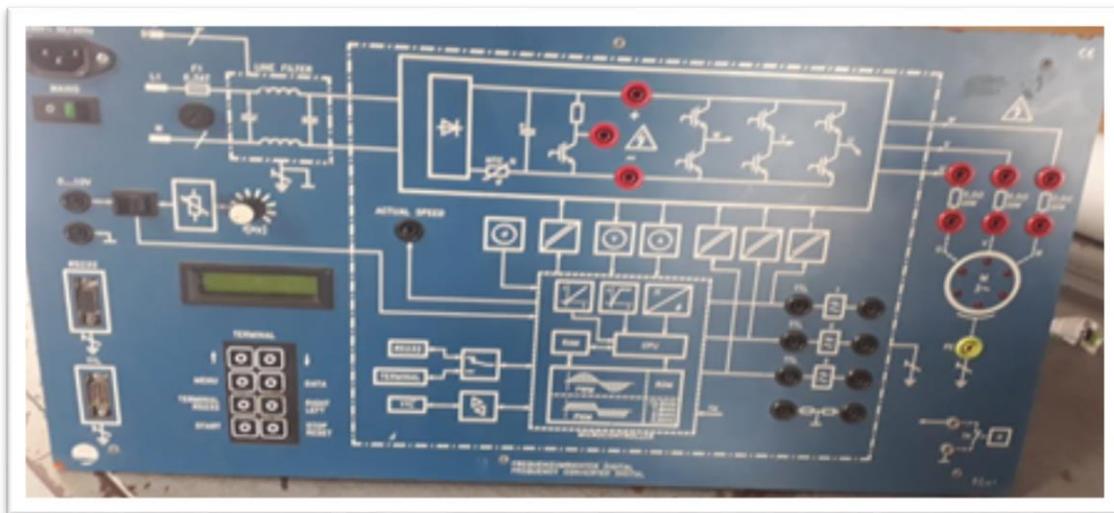


Figura 13 Variador de frecuencia académico. (Autor)

4.2.3 Taco – Generador

El taco – generador es un motor de corriente directa el cual esta adecuado de tal forma que al girar generará un voltaje a su salida, este voltaje puede ser utilizado para hacer una medición de las revoluciones por minuto a la que su rotor está girando, posee dos escalas, una de 0 a 1500 rpm con 0 y 10 voltios de salida, y otra escala de 0 a 3000 rpm con 0 y 10 voltios a su salida, el uso de alguna de las dos difiere en la cantidad de resolución que se quiere en el voltaje de salida. Para el presente proyecto se utilizó la escala de 0 a 3000 rpm puesto que es la que está en correcto

funcionamiento actualmente. Cabe mencionar que también se lo puede conectar como motor de corriente continua, ya que trae esa opción en su tablero de control como se observa en Figura 14.



Figura 14 Taco- generador. (Autor)

En Figura 14 se puede apreciar los pines E1, E2, A1, A2 que se podrían utilizar para conectar el taco como motor. En la parte izquierda con color rojo están marcados los pines de salida para medir las revoluciones por minuto (rpm) a la cual está girando, y por último y en color azul esta la conexión común a tierra.

4.3 Conexiones y empalmes

El motor y el taco – generador van acoplados, cada uno tiene en su eje un acople como se muestra en Figura 15, para unirlos se requiere diseñar una pieza que permita el acople entre ellos. Para

diseñar la pieza se utilizó el software de diseño SolidWorks, la pieza final se puede ver en Figura 16, la cual posteriormente se imprimió en 3D en material flexible.



Figura 15 Acoples de motor y taco-generator. (Autor)

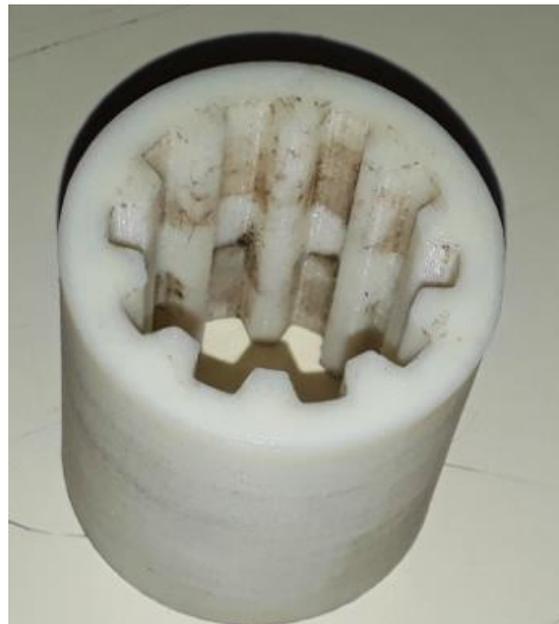


Figura 16 Acople diseñado para la planta. (Autor)

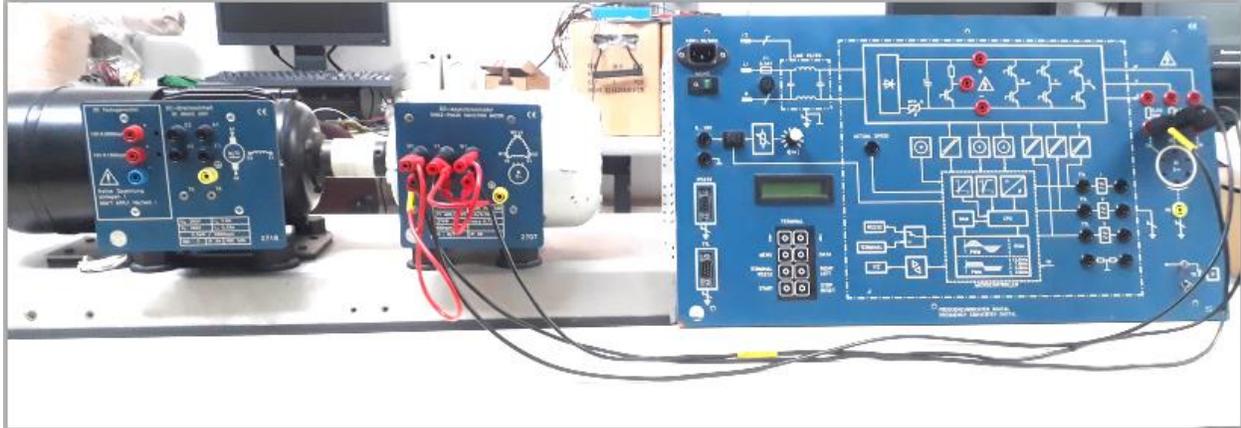


Figura 17 Planta con el acople. (Autor)

Con la impresión de la pieza en 3D, se procede a acoplar el motor y el taco- generador, como se muestra en Figura 17, quedando la planta a controlar completa, junto al variador de frecuencia.

4.3.1 Conexión del motor trifásico

El motor como se dijo anteriormente se lo conectó en configuración delta, en su tablero de control se encuentra el esquema de las tres bobinas y sus respectivas etiquetas como se muestra en Figura

18



Figura 18 Conexiones del motor trifásico. (Autor)

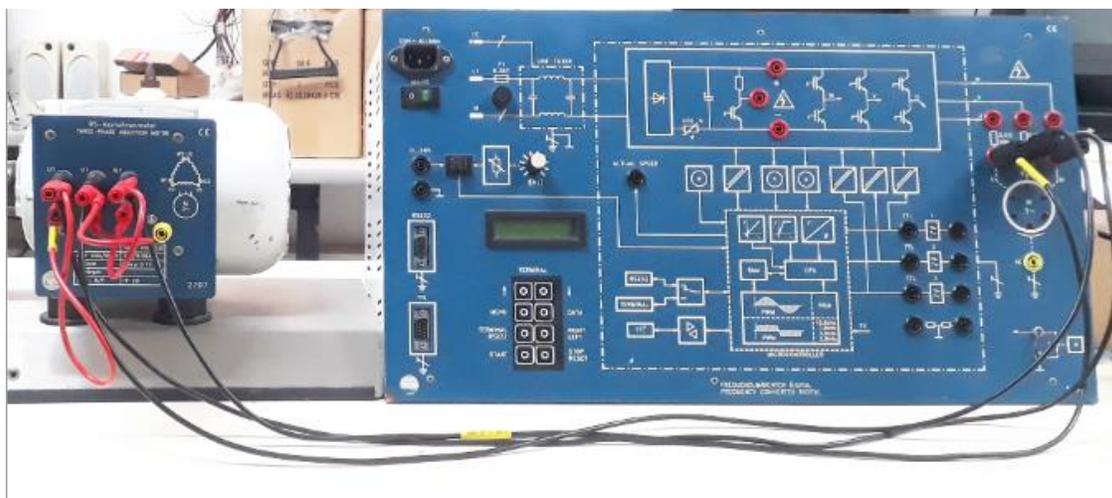


Figura 19 Conexión del motor al variador. (Autor)

Para finalizar su conexión se ubica los pines en el variador, como se muestra en Figura 19.

4.3.2 Configuración del variador

El variador que se utilizó tiene un teclado y una pantalla LCD para configurar, como se muestra en Figura 20.



Figura 20 Teclado y pantalla del variador. (Autor)

A continuación, se mencionan las configuraciones más importantes para usar el variador.

Definir la frecuencia mínima con la que el motor va a arrancar, en este caso la frecuencia mínima que acepta el variador es de 0.5Hz, así mismo se define la frecuencia máxima que en este caso es de 93.7 Hz (aunque la frecuencia máxima que soporta el variador es de 120Hz).



se eligió el valor de 93.7 Hz debido a que la salida del taco-generador está descalibrada, esto genera un voltaje mayor a 10V si se superan 2800 RPM, por lo tanto, se eligió ese valor de frecuencia que trabaja por debajo de 2800 RPM y de esta manera no genere conflictos con la toma de datos.

Se debe activar el potenciómetro, ya sea para trabajar con él o para usar la entrada de 0 a 10V, con esto el variador está listo para empezar a funcionar presionando la tecla start ver Figura 20.

5 Capítulo 2 Acondicionamiento de la planta.

Al inicio de este documento se plantearon tres tarjetas de desarrollo para llevar a cabo el presente proyecto, se hizo cotizaciones de cada una para tener un referente económico al momento de escoger, los precios de cada una se pueden ver en Tabla 3, se consultó en distintas tiendas electrónicas y se sacó un promedio de precios.

Nombre	Descripción	Precio (Pesos colombianos)
Raspberry pi3 modelo b+	Procesador Broadcom BCM2837B0, Cortex-A53 SoC de 64 bits a 1,4 GHz, con encabezado GPIO de 40 pines digitales.	197000 Vistrónica
Dreamer Maple	Procesador 32 bits a 72Mhz, voltaje de operación 3.3 V, 15 entradas análogas a 12 bits de resolución, 18 pines pwm a 16 bits.	140000 didácticas electrónica
Teensy 3.6	Procesador de 180Mhz, 22 salidas PWM, 11 entradas de detección táctil, 25 entradas analógicas, dos salidas analógicas (DAC) de 12 bits	144000 Electronilab

	de resolución, voltaje de operación 3.3 V, reloj en tiempo real.	
--	--	--

Tabla 3. Precios de tarjetas de desarrollo.

En Tabla 3 se pueden ver los precios de cada una de las tarjetas de desarrollo.

5.1 Criterio de selección

Para la selección de la placa de desarrollo se tiene en cuenta aspectos como, velocidad de procesamiento, el costo económico, la existencia de entradas y salidas tanto digitales como análogas.

Una vez vista la lista de precios y descripción en Tabla 3, se escoge la opción de adquirir la tarjeta de desarrollo teensy 3.6, puesto que la tarjeta raspberry no tiene entradas análogas, y para el desarrollo de este proyecto se requiere de entradas análogas con buena resolución, de igual forma salidas análogas de buena resolución. Por otra parte, la tarjeta dreamer maple tiene un precio similar a la teensy 3.6, sin embargo, su velocidad de procesamiento es mucho menor que la teensy, por lo cual la opción más favorable es la tarjeta teensy 3.6, ya que cuenta con dos salidas análogas DAC de 12 bits y un procesador de 180Mhz.

5.2 Circuitos de acondicionamiento

En términos generales se podría utilizar el variador sin problema, pues usando el potenciómetro incorporado se puede variar la velocidad del motor. Para el desarrollo de este proyecto se necesitaría un dispositivo capaz de enviar una señal de voltaje de 0 a 10 voltios, y que sea capaz de leer un voltaje de 0 a 10V.



Anteriormente se escogió como plataforma de desarrollo la tarjeta teensy 3.6, de esta forma toda la planta se debe llevar a los voltajes de salida y entrada de la tarjeta escogida. Como se puede ver en Tabla 3, la tarjeta cuenta con un voltaje de operación de 3.3 voltios, es decir que sus salidas y entradas analógicas aceptan valores de voltaje máximo de 3.3 voltios.

Es por esta razón que se necesita diseñar circuitos de acondicionamiento de la señal para poder trabajar con el variador, poder leer el voltaje del taco-generator y de esta forma controlar la planta completa.

5.2.1 Circuito Amplificador

Para controlar el voltaje que se le va a aplicar al variador de frecuencia se propone un circuito amplificador de voltaje, de tal forma que el voltaje de salida de la tarjeta de 3.3 voltios, se lo pueda amplificar hasta 10 voltios, el cual es el máximo valor de voltaje que acepta el variador.

Se hizo uso de un amplificador operacional configurado como no inversor, como se puede ver en Figura 21, para llevar el voltaje de 3.3 a 10V.

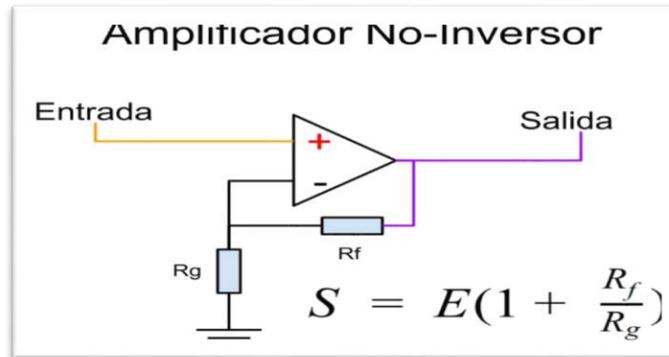


Figura 21 Amplificador no inversor. (Hetpro, 2018)

La ganancia del amplificador en esta configuración se puede ver en Ecuación 5 , en la entrada se conectó la señal de salida analógica de la tarjeta teensy, y la salida del amplificador se conecta al variador de frecuencia.

$$\text{Ganancia} = 1 + \frac{R_f}{R_g}$$

Ecuación 5. Ganancia de un amplificador no inversor. ((Hetpro, 2018))

Siendo Rf y Rg resistencias, de este modo solo se deben escoger las resistencias adecuadas para obtener la ganancia necesaria.

$$V_{out} = V_{in} * \left(1 + \frac{R_f}{R_g}\right)$$

Ecuación 6. Voltaje de salida amplificador no inversor.(Hetpro, 2018)

Despejando R_g de Ecuación 6 se obtiene:

$$R_g = \frac{R_f}{\frac{V_{out}}{V_{in}} - 1}$$

Ecuación 7. Valor de R_g . (Autor)

Para calcular el valor de ganancia deseada se usó Ecuación 7, donde V_{out} es voltaje de salida, V_{in} es el voltaje de entrada, estos dos valores ya se conocen, pues voltaje de salida (en términos máximos) es 10 voltios y voltaje de entrada (voltaje de la tarjeta teensy) es 3.3 voltios, y se asigna un valor a R_f de $4.14K\Omega$ (este valor es arbitrario puede ser cualquiera), y se calcula el valor de R_g :

$$R_g = \frac{4.14 * 10^3 \Omega}{\frac{10V}{3.3V} - 1} = 2.04K\Omega$$

De esta forma se obtiene el valor de la resistencia R_g para que el circuito eleve el voltaje de 3.3 voltios a 10 voltios.

5.2.2 Circuito divisor de tensión

Como se puede observar en Tabla 3, el máximo voltaje en la entrada análoga de la tarjeta es de 3.3 V, y el taco-generador entrega un voltaje máximo de 10V, por esta razón se debe diseñar un circuito divisor de tensión de tal forma que si el taco-generador saca 10V, el divisor de tensión los convierta a 3.3V, para poder conectar de esta forma a la entrada de la tarjeta y así evitar daños.

Para diseñar el circuito se utilizó el software ISIS de proteus, quedando el circuito como se observa en Figura 22.

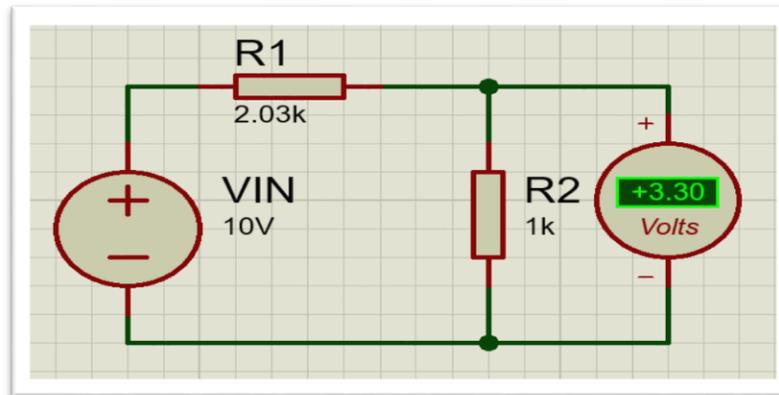


Figura 22 Circuito divisor de tensión. (Autor)

Colocando la entrada analógica de la tarjeta en R2, que es donde se dividiría el voltaje proveniente del taco-generador. Ahora para hacer el cálculo de las resistencias se utiliza la ley de Kirchoff.

Aplicando leyes de Kirchoff al circuito se tiene:

$$-V_{in} + i * R1 + i * R2 = 0$$

Ecuación 8. Ley general de Kirchoff. (Autor)

Donde R1 y R2 son resistencias, V_{in} es el voltaje proveniente del taco – generador, y por último i que es la corriente que circula por el circuito.

Despejando corriente i de Ecuación 8 se obtiene:

$$i = \frac{V_{in}}{R1 + R2}$$



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Ecuación 9. Corriente en el circuito divisor. (Autor)

Ahora lo que interesa es encontrar el voltaje en R2 que es donde se conectará la entrada analógica, por tanto y por ley de ohm se sabe que:

$$VR2 = i * R2$$

Ecuación 10. Voltaje en resistencia R2. (Autor)

Si tomo Ecuación 9 y reemplazo en Ecuación 10 se obtiene lo siguiente:

$$VR2 = \frac{Vin * R2}{R1 + R2}$$

Ecuación 11. Expresión voltaje en R2. (Autor)

Despejando R1 de Ecuación 11 se obtiene:

$$R1 = \frac{Vin * R2}{VR2} - R2$$

Ecuación 12. Expresión para el valor de R1. (Autor)

De Ecuación 12 se sabe el valor de VR2 que sería 3.3V, Vin que sería 10V, y se asigna un valor de 1KΩ a R2 (este valor es arbitrario, puede ser cualquiera), de este modo el valor final de R1 sería:

$$R1 = \frac{10V * 1 * 10^3}{3.3V} - 1 * 10^3 = 2.030 K\Omega$$

Con esos valores de resistencias se obtendría el valor deseado de 3.3 voltios a su salida, como se observa en Figura 22.

5.3 Diseñando circuito impreso

Una vez hechos los cálculos de los componentes necesarios para realizar tanto el circuito elevador de tensión como el divisor de tensión, ahora se procede a escoger los componentes. Para el amplificador operacional se adquirió el encapsulado LM324N, el cual contiene internamente 4 amplificadores operacionales como lo muestra su fabricante (Instruments, 2018), para las resistencias se escogieron trimmers de $5K\Omega$, los cuales son resistencias variables, esto con el fin de darle futuros usos distintos tanto al amplificador como al divisor.

Para realizar el esquema general del circuito completo de acondicionamiento (tanto como amplificador como divisor) se utilizó el software Eagle de Autodesk, el circuito se lo puede ver en Figura 23, de igual forma se puede ver completo en anexo 1.

Con el esquemático conectado correctamente se empezó a diseñar el circuito impreso, como se ve en Figura 24, el cual finalmente se imprimió y se pasó a una placa de cobre para soldar los componentes.

Para una mayor comodidad y evitar errores humanos a la hora de conectar, se diseñó una caja con los pines necesarios para conectar tanto el variador de frecuencia como el taco-generator, para el diseño de la caja se utilizó el software de SolidWorks, el diseño final se puede ver en Figura 25, de esta forma se completa la adecuación de la planta, se tiene entonces el motor, el taco-generator, el variador de frecuencia y la caja con el circuito de acondicionamiento, lo siguiente será empezar a tomar datos de entrada y salida.

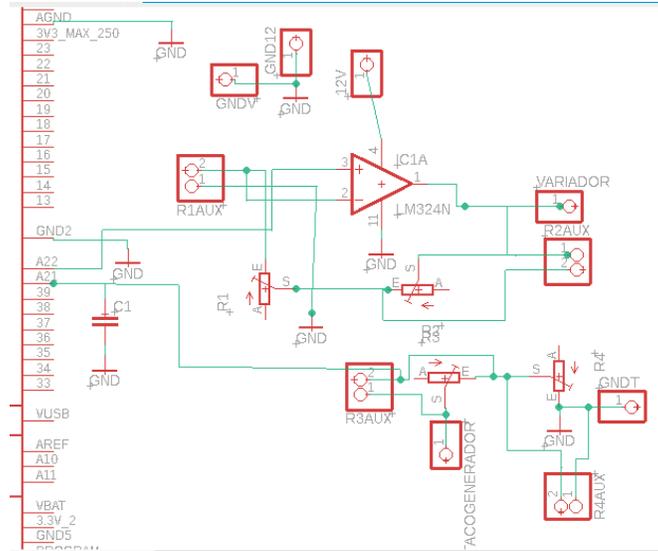


Figura 23 Circuito general acondicionamiento. (Autor)

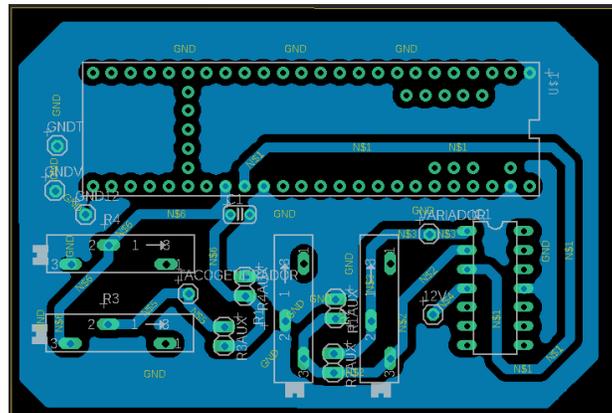


Figura 24 Circuito acondicionamiento impreso. (Autor)



Figura 25 Diseño caja circuito. (Autor)

5.4 Caracterización del Taco – Generador

Es necesario encontrar una expresión que relacione el voltaje de salida del taco-generador con las revoluciones por minuto del motor. Para ello se tomó datos de lectura del taco vs RPM medidas con un tacómetro digital.

En Figura 26 se observan algunos datos tomados para la caracterización del taco-generador que se usó como sensor de RPM, el resto de datos se pueden ver en anexo2.

Posterior a esto se graficaron en el software de Excel, sacando una ecuación que relacione las magnitudes de voltaje y revoluciones por minuto como se observa en Figura 27

Voltaje Taco generador (V)	RPM	PWM
0,035	46,2	40
0,055	70	80
0,076	100	150
0,145	182,3	250
0,195	241,2	350
0,25	320	450
0,3	379	510
0,325	408	600
0,417	510	800
0,575	715	1000
0,689	856	1200
0,801	996	1400
0,869	1079	1500
0,915	1135	1600
1,05	1303	1800
1,092	1355	1900

Figura 26 Datos de Voltaje vs RPM

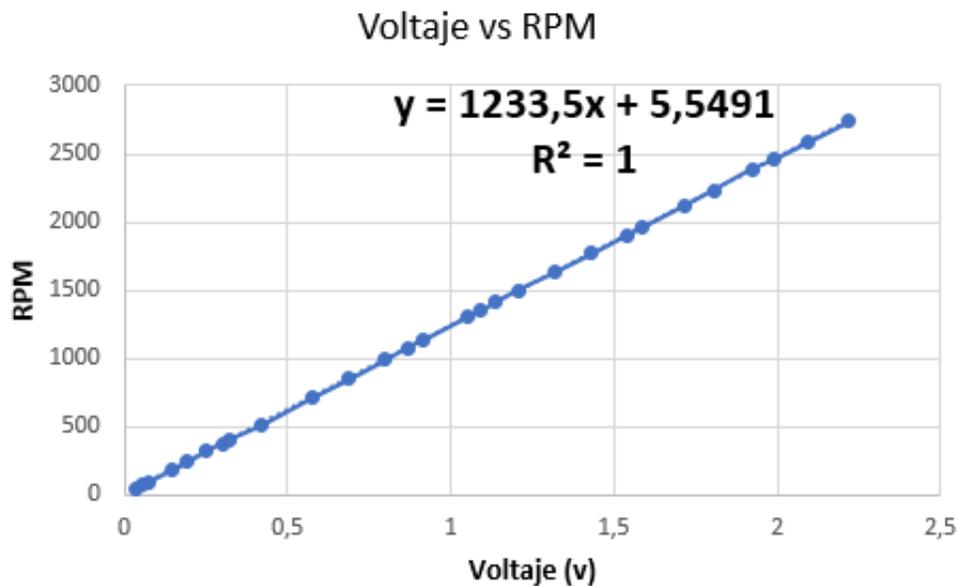


Figura 27 Gráfica de Voltaje Vs RPM. (Autor)



En Figura 27 se puede ver que el comportamiento del sensor taco-generador es de tipo lineal, por lo tanto, la ecuación que representa dicho comportamiento es la ecuación de una recta con pendiente de 1233.5 y un desfase con el eje y de 5.5491.

6 Capítulo 3 Identificación de la planta

6.1 Identificación de la planta

Para empezar a trabajar en las diferentes técnicas de control, primero se debe tener un modelo matemático de la planta, al cual se le puedan hacer simulaciones para tener una mejor idea de lo que sería el comportamiento real de la planta.

Generalmente lo que se hace es encontrar una función de transferencia que relacione la entrada de la planta y la respuesta a la salida como se observa en

Figura 28 donde la función de transferencia se describe como $G(s)$, porque ahí va una expresión matemática en forma de fracción descrita en Ecuación 13.

$$G(s) = \frac{\text{Salida}}{\text{entrada}}$$

Ecuación 13 Expresión de una función de transferencia. (Autor)

Para encontrar una expresión matemática que describa el comportamiento de la planta se suelen usar leyes físicas, ecuaciones y demás, pero este proceso se vuelve tedioso y complicado cuando se tiene una planta compleja. Otro aspecto a tener en cuenta si se modela con ecuaciones es que se estaría trabajando en condiciones ideales, es decir que no se tienen en cuenta factores externos que afectarían el funcionamiento de la planta como desgaste, rozamiento, humedad de las piezas etc. Es por esto que en este trabajo se decidió usar otro método para encontrar la función de transferencia.

Otro método para encontrar el modelo matemático es por medio de toma de datos, se estimula la planta con una señal de entrada y se observa la salida de esta, tomando estos datos y almacenándolos para un posterior tratamiento se puede encontrar una función de transferencia. Existen distintos programas que ofrecen herramientas de identificación a partir de datos, para este trabajo se eligió la herramienta identificación de Matlab.



Figura 28 Función de transferencia. (Autor)

Para la señal de entrada a la planta se propuso la que se observa en Figura 29 la cual está conformada de señales paso, rampa y constantes, esto con el fin de obtener una variedad de datos que permita una buena identificación. Se puede apreciar que el valor máximo de voltaje en la entrada es de

3.3V, ya que es el voltaje al cual trabaja la tarjeta, pero con el circuito de acondicionamiento visto en el capítulo anterior se estaría enviando una señal máxima de 10V, identificando la planta con estos datos quedaría inmerso dentro de la planta el circuito de acondicionamiento, tanto a la entrada como a la salida.

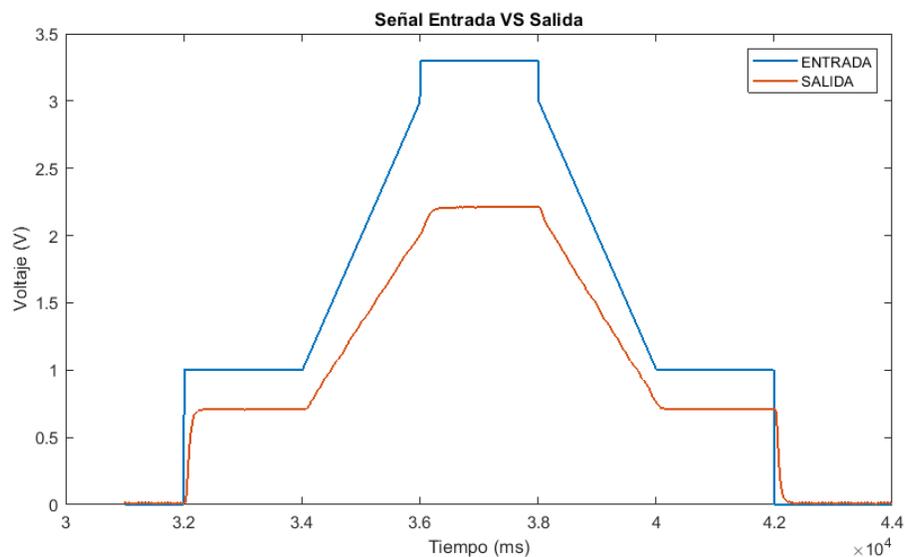


Figura 29 Señal de entrada vs salida de la planta. (Autor)

En Figura 29, se puede observar tanto la señal de entrada (línea azul), como la respuesta que se obtuvo por parte de la planta (línea roja), se aprecia que el voltaje máximo de la señal de salida es 2.2V por lo que se espera en el proceso de identificación una ganancia menor a 1.

Una vez se obtuvieron los datos se ingresaron a la herramienta de identificación de Matlab como se aprecia en Figura 30

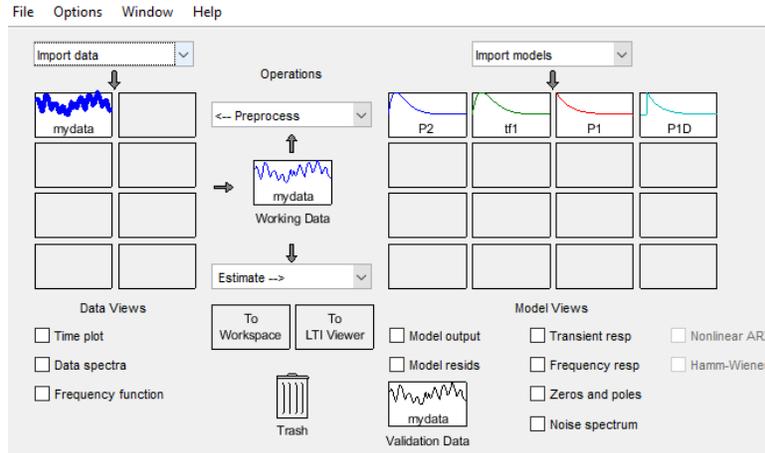


Figura 30 Datos en herramienta identificación (Autor)

Dentro de los parámetros de la herramienta de identificación se escogió una función de transferencia con polo doble puesto que el motor evoluciona rápido en el tiempo y una función de transferencia de un solo polo es decir de primer orden no tenía una buena aproximación.

Luego de distintas estimaciones se tenían dos modelos que presentaban una buena aproximación, el primero era un modelo de segundo orden sin atraso, y el segundo un modelo de primer orden con atraso, se escogió el modelo de segundo orden sin atraso, puesto que el otro presentaba un problema al momento de cambios bruscos en la entrada y esto podría alterar las simulaciones.

$$G(s) = \frac{0.6861}{0.001822 * S^2 + 0.08536 * S + 1}$$

Ecuación 14. Función de transferencia de la planta (Autor)

En Ecuación 14 se observa la función de transferencia encontrada mediante identificación de la planta motor-taco generador, es una función de transferencia de segundo orden que está en el



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

dominio de S , es decir en el dominio de frecuencia continua, se encontró el modelo en S , porque directamente identificando en Z presentaba un error menor al 70% con funciones de transferencia que no seguían idealmente a los datos del modelo real, así que se encontró el modelo en S y se optó pasarlo a Z mediante aproximación bilineal.

Se podrían hacer simulaciones directamente con el modelo encontrado, pero lo ideal es llevar esa función de transferencia al dominio discreto, conocido como el dominio de Z , ya que para el control de la planta se usará una tarjeta con un microcontrolador el cual es un sistema discreto, es decir que trabaja por intervalos de tiempo. Para hacer este proceso se utilizó la transformación bilineal que se ve en Ecuación 15, siendo T el tiempo de muestreo.

$$S = \frac{2}{T} * \frac{(Z - 1)}{(Z + 1)}$$

Ecuación 15. Transformación de S . (Ogata, 1996)

Reemplazando Ecuación 15 en Ecuación 14 con un tiempo de muestreo de 10 milisegundos (que fue el tiempo de muestreo con el que se tomaron los datos), se obtiene lo siguiente:

$$G(Z) = \frac{0.6861}{0.001822 * \left(\frac{2*(Z-1)}{T*(Z+1)}\right)^2 + 0.08536 * \frac{2*(Z-1)}{T*(Z+1)} + 1}$$

Ecuación 16. Reemplazo función transferencia en Z . (Autor)

Tomando la Ecuación 16 y simplificándola se obtiene la función de transferencia con la que se pueden hacer simulaciones.

$$G(Z) = \frac{6.861e - 7 * Z^3 + 2.0583e - 6 * Z^2 + 2.0853e - 6 * Z + 6.861e - 7}{9.0952e - 5 * Z^3 - 5.2808e - 5 * Z^2 - 8.6952e - 5 * Z + 5.6808e - 5}$$

Ecuación 17. Función de transferencia de la planta en Z. (Autor)

6.2 Comprobación del modelo encontrado

Para verificar que se haya hecho una buena identificación de la planta, se hizo una simulación de Ecuación 17 usando el software de Simulink de Mathworks, en él se implementó la función de transferencia encontrada usando como señal de entrada la misma que se utilizó para identificar el sistema vista en Figura 29. En Figura 31 se observa el esquema en Simulink para comprobar la función de transferencia obtenida.

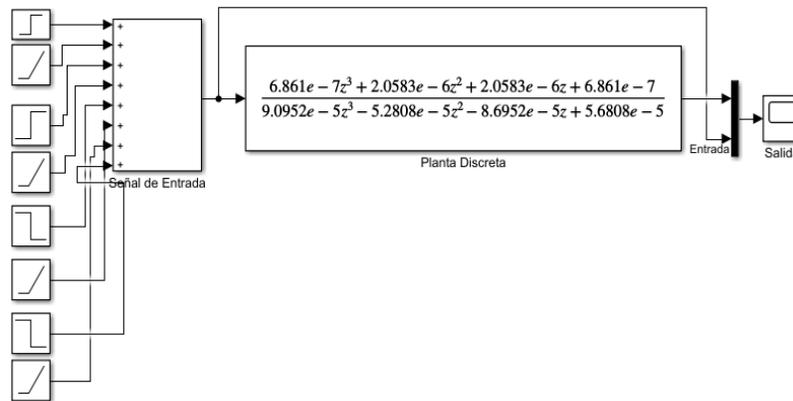


Figura 31 Función de transferencia en Simulink. (Autor)

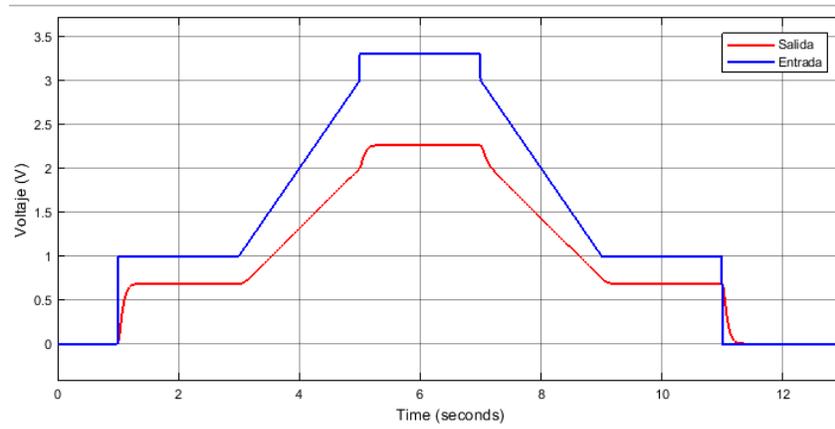


Figura 32 Entrada Vs salida del modelo virtual. (Autor)

En Figura 32 se puede observar las señales tanto de entrada como de salida del modelo virtual encontrado mediante identificación, a primera vista son muy similares con los datos reales, pero no se podría inferir mucho si no se hace una comparativa en una misma gráfica de los datos reales y los datos obtenidos mediante simulación. Es por ello que se comparó en un mismo gráfico los datos reales y los datos simulados, exportando como un vector los datos obtenidos en simulink, para graficar las 4 señales al tiempo.

En Figura 33 se puede apreciar que no existe mucha diferencia entre los datos de salida reales (línea color rojo) y lo datos de salida generados por simulación (línea color morado), esto teniendo en cuenta que se le aplicó la misma señal de entrada con la que se estimuló la planta real. Ahora si se puede afirmar que se obtuvo un modelo aceptable para hacer pruebas de las distintas técnicas de control a nivel de simulación para finalmente aplicarlas a la planta real.

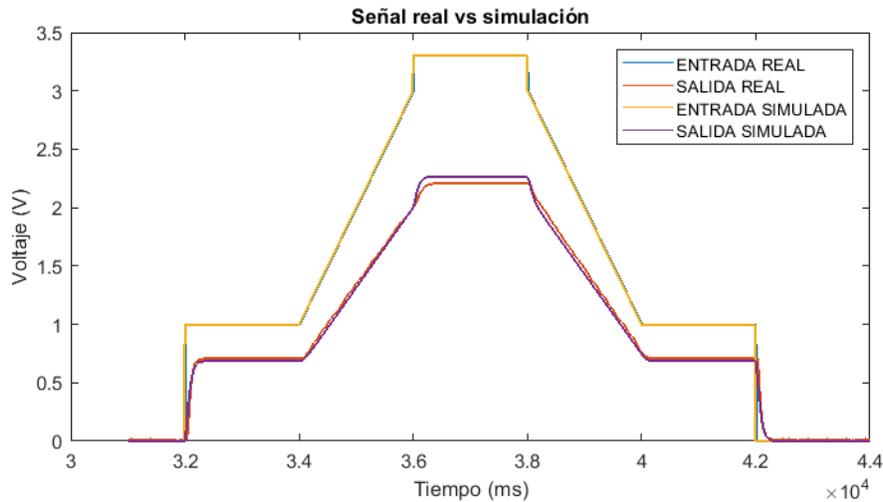


Figura 33 Señal real Vs simulada. (Autor)

7 Capítulo 4 Controladores simulados

En este capítulo se plantearon y aplicaron a nivel de simulación las distintas técnicas de control para el modelo de la planta encontrado en el capítulo 3, para ello se comenzó con el control PID discreto, luego control fuzzy y por último control neuronal directo.

7.1 Control PID discreto

Antes de aplicar cualquier control a la planta, se plantea el sistema de bloques general que se van a usar y con los cuales se hará la simulación completa, para ello se sugiere el esquema visto en Figura 34.

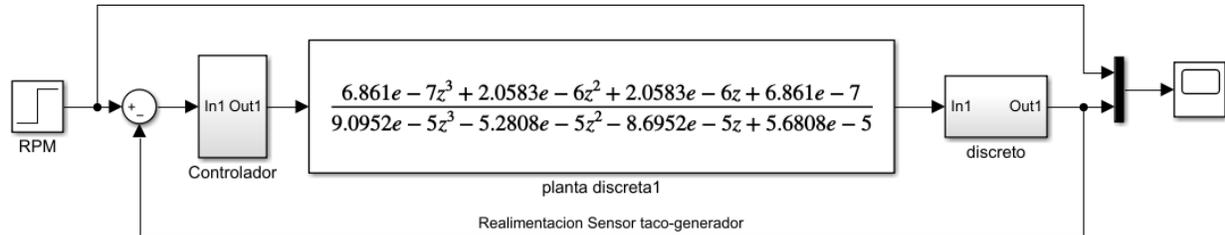


Figura 34. Esquema general de simulación. (Autor)

En Figura 34 se observa el diagrama de bloques usado para hacer simulación, donde el bloque de RPM es la entrada o el valor de velocidad deseado, luego viene el controlador ya sea tipo PID discreto, difuso o neuronal, el cual es el encargado de enviarle la señal a la planta. La salida de la planta está en voltaje, pero la señal de referencia está en revoluciones por minuto, entonces estas dos señales no se pueden comparar directamente pues se estaría cometiendo un error en cuanto a unidades, por lo que se tiene que convertir la salida de voltaje de la planta a RPM para así hacer la comparación con la señal de entrada y que ese sea el error que ingresa al controlador. La conversión de voltaje a RPM se hace mediante el bloque a la salida de la función de transferencia dentro del cual está aplicada la fórmula vista en Ecuación 18 y en Figura 27.

$$RPM = (1233.5 * voltaje) + 5.5491$$

Ecuación 18 Caracterización del sensor taco-generador. (Autor)

Para aplicar un control PID se parte de la ecuación básica que representa la señal de control como se muestra en Ecuación 19 donde $U(t)$ es la señal de control, K_p la constante proporcional, T_i el tiempo diferencial y T_d el tiempo derivativo.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

$$U(t) = Kp * e(t) + \frac{Kp}{Ti} \int e(t)dt + Kp * Td * \frac{de(t)}{dt}$$

Ecuación 19. Expresión general de control PID. (Rodríguez Rosa, 2012)

En algunos textos colocan K_i como Kp/T_i y K_d como $T_d * Kp$ pero es una forma de simplificar la expresión, se puede trabajar con cualquiera de las dos formas. En Ecuación 19 existe un problema y es que la expresión está dada en el dominio de tiempo continuo, y el micro controlador que se va a usar trabaja en tiempo discreto, es decir existe un tiempo de muestreo o un retardo entre cada señal aplicada al actuador. Para solucionar este problema se sugiere pasar toda la expresión al dominio de la frecuencia usando la transformada de Laplace, posteriormente pasar al dominio de Z usando transformada Z , y finalmente pasar la expresión a tiempo discreto.

Para pasar la expresión de la Ecuación 19 al dominio de la frecuencia S , se hará uso de las siguientes igualdades:

$$\mathcal{L} \left[\int f(t)dt \right] = \frac{F(s)}{S}$$

Ecuación 20 Transformada de Laplace de una integral. (Ogata, 1995)

$$\mathcal{L} \left[\frac{df(t)}{dt} \right] = S * F(s)$$

Ecuación 21 Transformada de Laplace de una derivada. (Ogata, 1995)

Haciendo uso de Ecuación 20 y Ecuación 21 para pasar del dominio de tiempo continuo t , al dominio de frecuencia S , se tiene la siguiente expresión:

$$U(s) = Kp * e(s) + \frac{Kp}{Ti} * \frac{e(s)}{s} + Kp * Td * s * e(s)$$

Ecuación 22 Expresión de un controlador PID en S. (Autor)

Teniendo la señal de control en el dominio de frecuencia S , ahora se usa Ecuación 15 para hacer una transformación bilineal al dominio discreto de Z obteniendo lo siguiente.

$$U(z) = Kp * E(z) + \frac{Kp}{Ti} * \frac{2}{T} * \frac{(z-1)}{(z+1)} + Kp * Td * \frac{2}{T} * \frac{(z-1)}{(z+1)} * E(z)$$

Ecuación 23 Transformación bilineal a Z de la señal control. (Autor)

En Ecuación 23 se observa una expresión para la señal de control en el dominio discreto de Z , teniendo a T como tiempo de muestreo del controlador.

Operando y simplificando la expresión dada en Ecuación 23 se obtiene lo siguiente:

$$\frac{U(z)}{E(z)} = \frac{\left[-Kp + \frac{Kp*T}{2*Ti} + \frac{Kp*Td*2}{T}\right] + \left[\frac{Kp*T}{Ti} - \frac{4*Kp*td}{T}\right] * Z + \left[Kp + \frac{Kp*T}{2*Ti} + \frac{Kp*Td*2}{T}\right] * Z^2}{Z^2 + 1}$$

Ecuación 24 Función transferencia controlador en Z. (Autor)

La Ecuación 24 es una función de transferencia del controlador discreto en el dominio de Z , puesto que es una expresión que relaciona la salida (señal de control $U(z)$), con la entrada (error $E(z)$), se observa que las expresiones que están dentro de los corchetes son constantes, por lo cual el

resultado es otra constante, entonces se podría simplificar de la siguiente manera reemplazando lo que está dentro de los corchetes por A0, A1, A2:

$$A0 = \left[-Kp + \frac{Kp * T}{2 * Ti} + \frac{Kp * Td * 2}{T} \right]$$

Ecuación 25 Expresión para A0. (Autor)

$$A1 = \left[\frac{Kp * T}{Ti} - \frac{4 * Kp * td}{T} \right]$$

Ecuación 26 Expresión para A1 (Autor)

$$A2 = \left[Kp + \frac{Kp * T}{2 * Ti} + \frac{Kp * Td * 2}{T} \right]$$

Ecuación 27 Expresión para A3 (Autor)

Reemplazando las ecuaciones Ecuación 25, Ecuación 26, Ecuación 27, en Ecuación 24 se obtiene:

$$\frac{U(z)}{E(z)} = \frac{A0 + A1 * Z + A2 * Z^2}{Z^2 - 1} * \frac{Z^{-2}}{Z^{-2}} = \frac{A2 + A1 * Z^{-1} + A0 * Z^{-2}}{1 - Z^{-2}}$$

Ecuación 28 Función transferencia simplificada en Z. (Autor)

Pero para este trabajo no se necesita la función de transferencia del controlador, si no encontrar una expresión que defina la señal de entrada a la planta, o sea la salida del controlador, por lo tanto, se despeja U(z) de Ecuación 28 obteniendo lo siguiente:

$$U(z) = U(Z^{-2}) + A2 * E(Z) + A1 * E(Z^{-1}) + A0 * E(Z^{-2})$$

Ecuación 29 Señal de control en dominio Z. (Autor)

Los Z-2 que se aprecian en Ecuación 29 se conocen como retardos, entonces en este caso se tiene retardos de la señal de error (E), y la señal de control (U), así que el paso final es pasar esta expresión a tiempo discreto teniendo en cuenta lo anterior:

$$U(k) = U(k - 2) + A2 * E(k) + A1 * E(k - 1) + A0 * E(k - 2)$$

Ecuación 30 Señal de control en tiempo discreto. (Autor)

Ya obtenida una expresión para calcular la señal de control en tiempo discreto, se hace el esquema en Simulink que representaría la ecuación encontrada en Ecuación 30

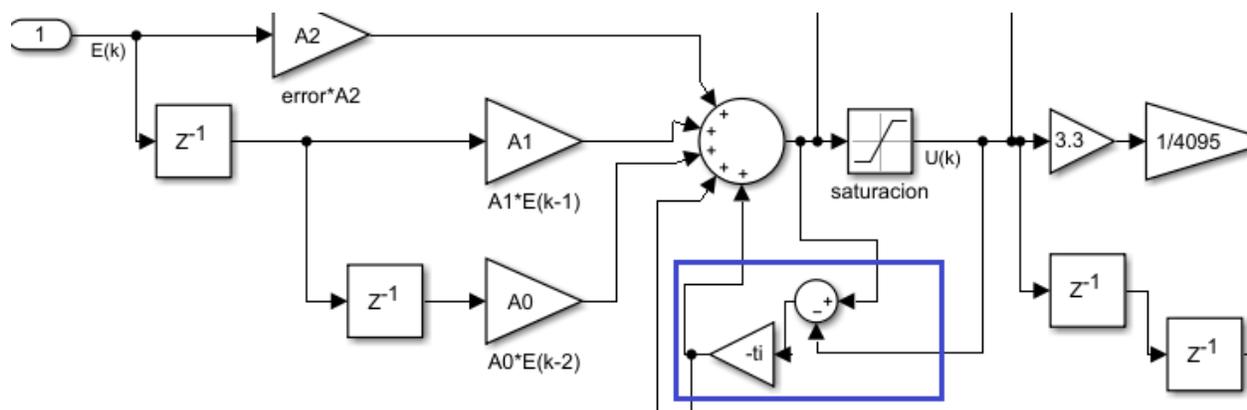


Figura 35 Esquema control PID discreto Simulink. (Autor)

En el esquema presentado en Figura 35 se hace uso de un bloque de saturación, esto con el fin de que la máxima señal de salida sea de 3.3V, puesto que ese es el máximo valor que acepta la tarjeta utilizada en este trabajo. Los bloques que se encuentran dentro del cuadro marcado de color azul es un esquema anti windup. La señal de control a la salida se puede ver afectada por el parámetro integral de la ecuación, esto provoca un desbordamiento fuera de los rangos aceptados por la

planta, inicialmente se puede corregir este problema con el ajuste de saturación, pero internamente la señal irá aumentando o disminuyendo provocando un tiempo muerto, o tiempo en el cual no se estaría haciendo control a la planta.

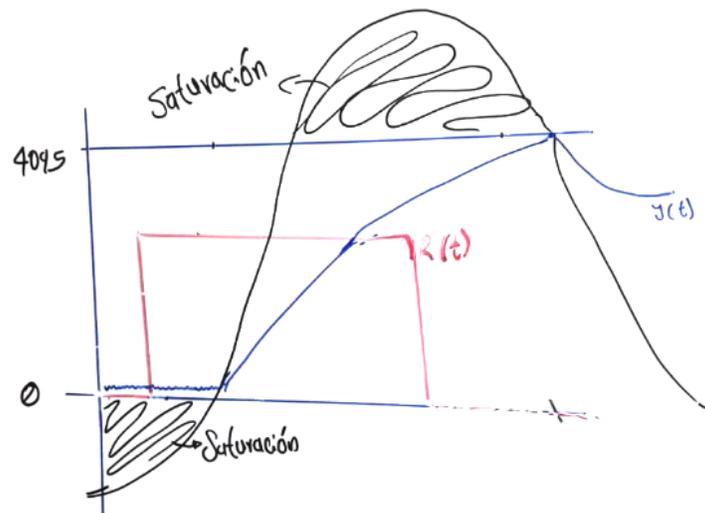


Figura 36 Efecto wind up en señal de control. (Autor)

La planta a controlar en este trabajo acepta valores de 0 a 4095, en Figura 36 se observa que señales por debajo de ese rango están saturadas, y la salida $y(t)$ presenta un retardo ante una entrada $r(t)$ presentada por color rojo, y vuelve a reaccionar cuando la señal de control (presentada con color negro) vuelve a caer en la zona activa de la planta.

Para contrarrestar el efecto wind-up de la señal de control, se resta el valor antes de pasar por saturación con el valor saturado, si la señal de control está en la zona activa esta resta dará cero por lo que el efecto anti wind-up quedaría anulado, pero si no es así ese valor se multiplica por el



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

término t_i , como se muestra en Figura 35, y este término se suma a la ecuación de señal de control vista en Ecuación 30.

Ahora lo único que falta para implementar este controlador es encontrar los valores de A_0 , A_1 , A_2 , y para ello se necesitan encontrar las constantes K_p , T_i , T_d del controlador y ajustarlas para obtener una respuesta deseada.

Para encontrar las constantes K_p , T_i , T_d se usó el método de sintonización de Ziegler y Nichols (Villajulca, 2011) con la señal de un paso de 40% del valor máximo de entrada, el método dice que se debe trazar una recta tangente al punto de inflexión a la curva de salida de la señal. Hay que aclarar que se usó el método para tener una aproximación de las constantes, pues ya se tiene el controlador y planta implementados en Simulink, con los cuales se pueden ajustar los valores encontrados en este método para una mejor respuesta.

En Figura 37 se puede ver la señal paso de entrada equivalente a 1V, y la señal roja es la respuesta de la planta, este método se hizo de forma manual pues no es crítico el proceso de encontrar las constantes exactas ya que esa no es la intención principal, si no tener una aproximación.

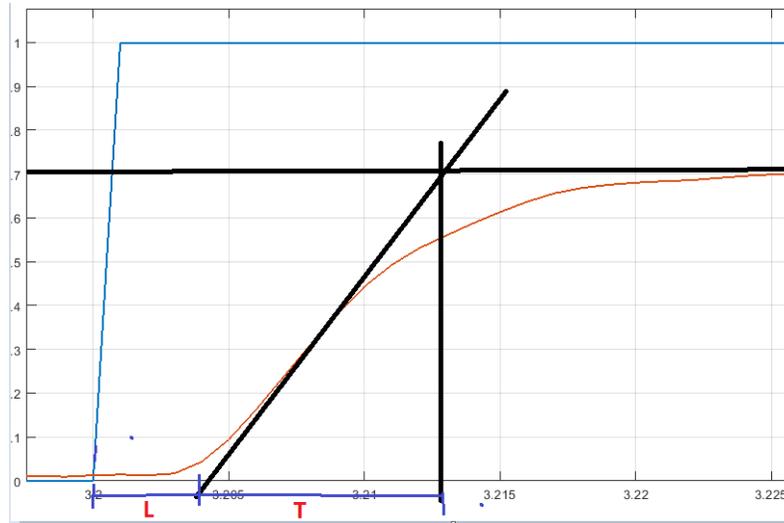


Figura 37 Sintonización Por Ziegler y Nichols. (Autor)

Controlador	Kp	Ti	Td
PID	$1.2 * \frac{T}{Ks * L}$	2*L	0.5*L

Tabla 4 Cálculo constantes Ziegler y Nichols. (Villajulca, 2011)

Para calcular las constantes se sacaron los valores de L y T de la gráfica vista en

Figura 37, los cuales son 0.004 y 0.015 respectivamente, el valor de Ks es la ganancia de la función de transferencia, ese valor se lo puede encontrar en Ecuación 14 y es de 0.6861.

Aplicando las fórmulas que aparecen en Tabla 4 se obtienen las constantes $K_p = 6.558$, $T_i = 0.008$ y $T_d = 0.002$. Estos valores encontrados se los reemplaza junto con un tiempo de muestreo de 0.004 segundos en las ecuaciones Ecuación 25, Ecuación 26, Ecuación 27, para encontrar los valores de A_0 , A_1 , A_2 y hacer la primera simulación de un control PID discreto.

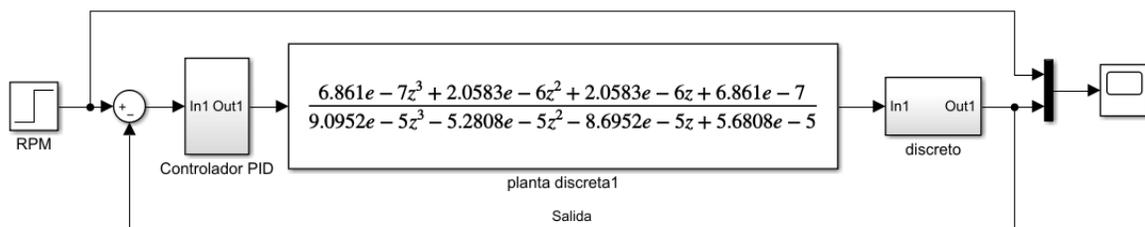


Figura 38 Esquema Simulink control PID. (Autor)

Se implementó el esquema en Simulink que aparece en Figura 38 para hacer la primera simulación de la planta con un controlador PID discreto, dentro del bloque controlador PID está inmerso el esquema que se ve en Figura 35, se lo hizo así por motivos de orden y estética, esta temática se aplicará a las demás técnicas de control.

En Figura 39 se observa la respuesta del controlador PID discreto ante una señal de paso con un set point de 1000 RPM, alrededor del punto de operación donde se lo sintonizó, se puede apreciar una respuesta inmediata de la planta y se estabiliza en un intervalo de tiempo corto, idealmente esta respuesta estaría bien, pues la señal de salida (Línea roja) sigue a la perfección la señal de referencia (Línea azul) que es lo que se busca en un control.

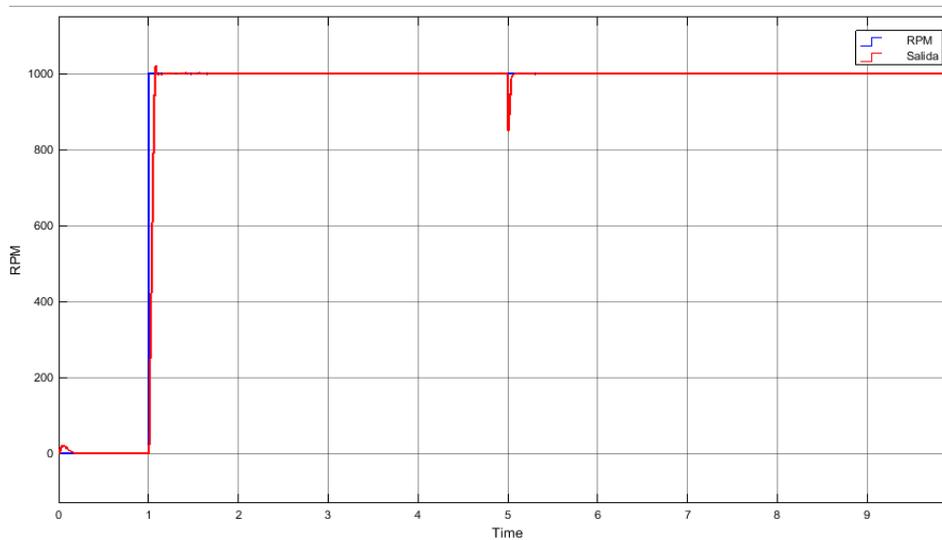


Figura 39 Respuesta control PID discreto 1. (Autor)

En esta primera simulación se le aplicó una perturbación al sistema a los 5 segundos, el controlador es capaz de recuperar el set point. La planta real no trabaja en condiciones ideales por lo tanto es importante analizar la señal de control que se le está aplicando.

En *Figura 40* se observa la señal de control, existe un problema, pues está muy saturada, esto puede traer inconvenientes al momento de aplicar esta señal a la planta física, pues la planta no podría reaccionar a esta señal, además se comprobó experimentalmente que el motor no reacciona tan rápido a un set point de 1000 RPM.

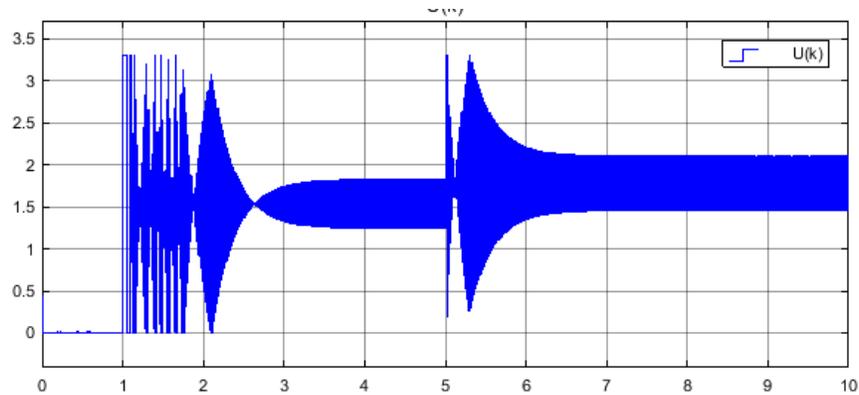


Figura 40 Señal de control PID. (Autor)

Lo que se hizo es ajustar manualmente las constantes K_p , T_d , T_i hasta obtener una señal de salida y una señal de control aceptables, se aplicó la técnica de empezar variando K_p , luego T_i , y por último T_d hasta obtener una respuesta ideal.

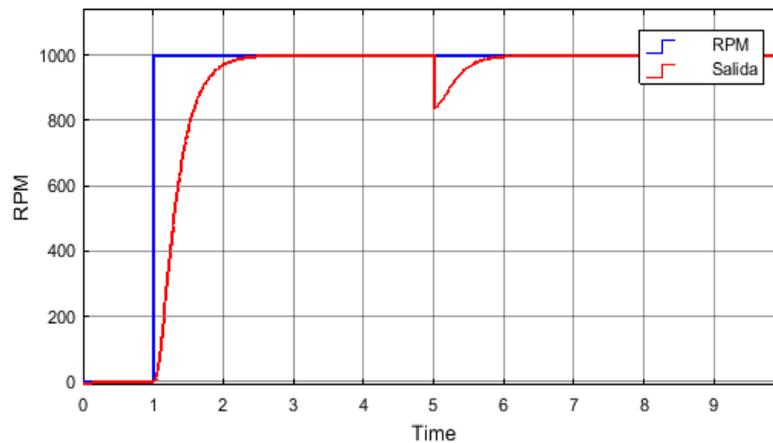


Figura 41 Respuesta control PID ajustado. (Autor)

En Figura 41 se ve la respuesta del sistema con las variables ajustadas, se observa una respuesta sin oscilaciones y un tiempo de establecimiento de 1.3 segundos, es un tiempo aceptable, y en Figura

42 se mira la señal de control, ya no está saturada y los cambios que presenta si los puede asimilar la planta real. Las constantes finales encontradas fueron: $K_p = 0.0015$, $T_i = 0.15$ y $T_d = 0.5$.

Se puede inferir que la señal de control presentada en Figura 42 es la ideal puesto que a los 5 segundos se aplica una perturbación que disminuye la velocidad de la planta, por lo que se espera que la señal de control aumente para compensar la perturbación, tal y como se muestra en Figura 42.

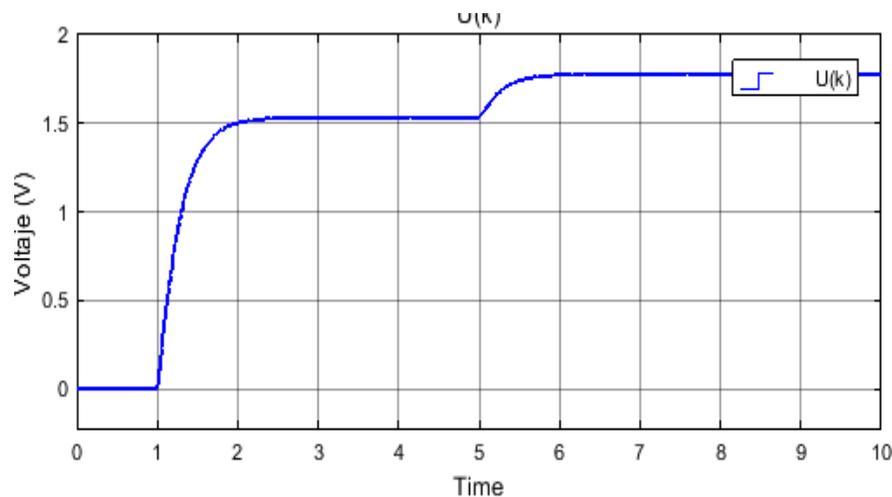


Figura 42 Señal de control ajustada. (Autor)

7.2 Control Fuzzy

La segunda técnica de control que se aplicó a la planta, es control por medio de lógica difusa control fuzzy. Hay distintos tipos de control fuzzy, para este trabajo se utilizó la inferencia de Mamdani, el cual consiste en definir una serie de conjuntos en las señales de entrada como de



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

salida denominados funciones de membresía, y unas reglas que describen el funcionamiento de la planta.

Para empezar, se definió el número de entradas y salidas, para este caso en concreto se tomó el error y un Δ error como entradas, para la salida se tomó un porcentaje de incremento o decremento de la variable de control.

Para la entrada de error se definieron 7 conjuntos llamados: ENG (error negativo grande), ENM (error negativo medio), ENP (error negativo pequeño), EC (error cero), EPP (error positivo pequeño), EPM (error positivo medio), EPG (error positivo grande), como se ve en *Figura 43*.

El universo de discurso de error para la variable de entrada error es de -2800 hasta 2800, pues son las RPM máximas a las que llega el motor.

Inicialmente el controlador tenía un buen comportamiento solo con una entrada, pero presentaba un error en estado estacionario por lo que se decidió agregarle una entrada más, y esta es la ventaja de trabajar con un control fuzzy, que fácilmente se puede controlar un sistema MISO (múltiples entradas, una salida).

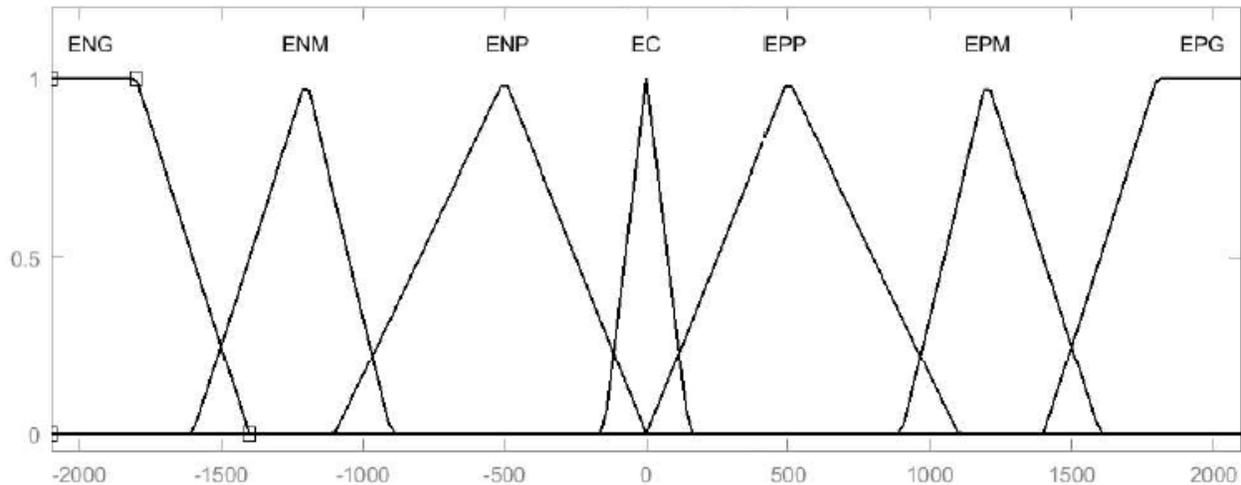


Figura 43 Conjuntos difusos de entrada. (Autor)

La segunda entrada al controlador difuso es un Δ error definido como la diferencia entre el error actual menos el anterior.

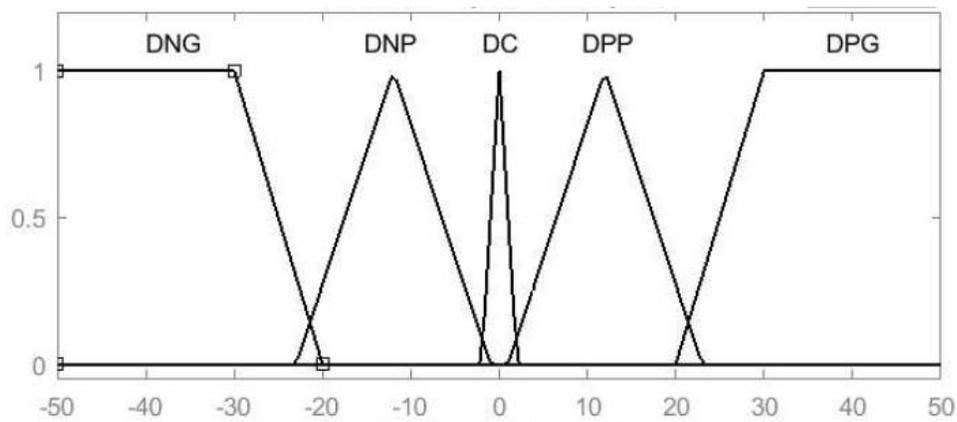


Figura 44 Conjuntos difusos para delta error. (Autor)

El universo de discurso para la segunda variable de entrada (delta error) es de -50 hasta 50 como se observa en *Figura 44*, este valor se lo escogió luego de hacer varias simulaciones y observando la respuesta de la planta, de igual forma la mayor parte del tiempo los valores de delta de error

oscilan entre esos rangos. Los conjuntos de la variable delta error son: DNG(derivada negativa grande), DNP(derivada negativa pequeña), DC(derivada cero), DPP(derivada positiva pequeña), DPG(derivada positiva grande). Cuando el valor de Δ error sea negativo significa que la velocidad está aumentando, y cuando es positivo significa que la velocidad está disminuyendo, esto es importante a tener en cuenta al momento de plantear las reglas.

El error es positivo cuando la variable de salida está por debajo del setpoint, y es negativo cuando la variable de salida está por encima del setpoint, esto se lo puede apreciar mejor en *Figura 45*

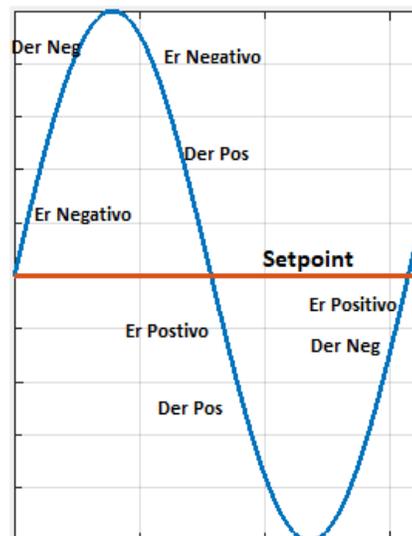


Figura 45 Análisis de error y delta error. (Autor)

Para controlar al motor se definió una variable de salida como porcentaje de aumento o decremento de la señal de control, en términos generales si la planta necesita más velocidad la salida aumentará

el voltaje, si necesita disminuir la velocidad la salida restará voltaje, y la salida será cero cuando el motor llegue a la velocidad deseada.

Se decidió trabajar la salida en bits, puesto que la tarjeta del micro controlador maneja un conversor DAC de 12 bits, por tanto, la salida en bits máxima será de 4095, siendo esto equivalente a 3.3 voltios.

El universo de discurso de los conjuntos de salida se definió desde -120 hasta 120, siendo este el aumento o decremento de la señal de control que se le aplicará a la planta.

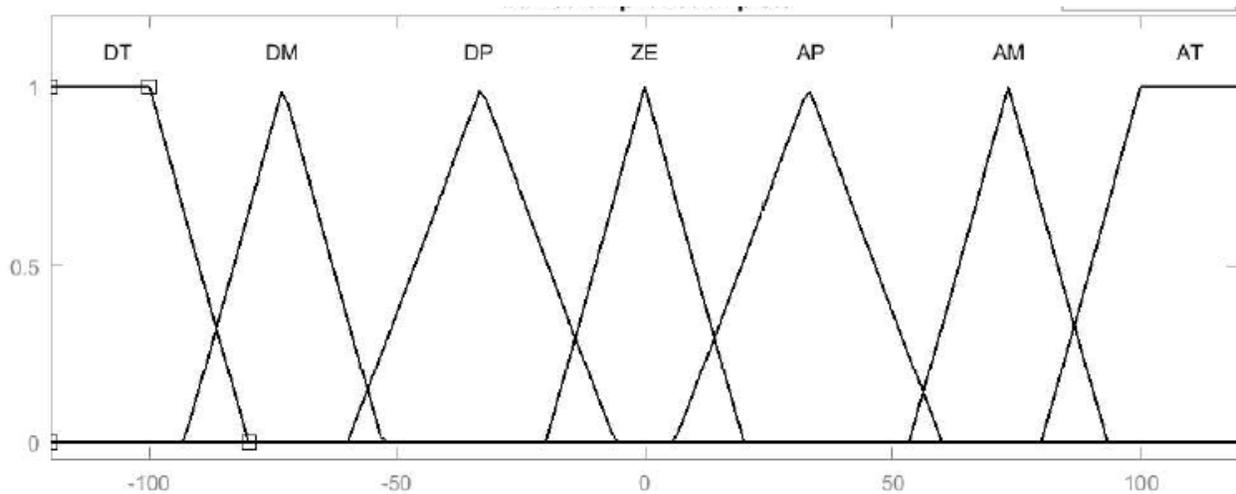


Figura 46 Conjuntos difusos de salida. (Autor)

En Figura 46 se observan los conjuntos definidos para la salida del controlador llamados: DT(decremento total), DM(decremento medio), DP(decremento pequeño), ZE(salida cero), AP(aumento pequeño), AM(aumento medio), AT(aumento total).

Ya definidos los conjuntos o funciones de membresía de entrada como de salida, se procede a crear las matrices de reglas, primero la matriz que relaciona el error con la salida, y luego otra matriz compuesta que relaciona el Δ error y el error con la salida, en Tabla 5 se observa la primera matriz.

Error	Salida
ENG	DT
ENM	DM
ENP	DP
EC	ZE
EPP	AP
EPM	AM
EPG	AT

Tabla 5 Primera matriz de reglas del controlador fuzzy. (Autor)

Ahora se crea la segunda matriz que relaciona el valor de Δ error y error con la salida como se ve en Tabla 6

Δ error	Error	Salida
DNG		
DNP	EC	DP
DC	ENP	DP
DPP	EC	AP
DPG		

Tabla 6 Matriz de reglas compuesta controlador fuzzy. (Autor)

La matriz de reglas que se aprecia en Tabla 6 es la que se usa para corregir el error en estado estacionario de la planta.

7.2.1 Creando algoritmo de control

Como se dijo anteriormente para aplicar el control difuso se usó el método de inferencia de mamdani, el cual es un proceso de mapeo no lineal basado en la lógica difusa, que fue sugerido

por Zadeh (1965). Este método podría modelar los sistemas intrincados en condiciones vagas y / o imprecisas, involucra tres componentes: fuzzificación, reglas de inferencia y defuzzificación como se ve en Figura 47, en donde las palabras subrayadas de color amarillo son las entradas y la palabra subrayada de color verde es la salida del controlador. En detalle, fuzzificación convierte los valores nítidos (entradas y salidas) en variables lingüísticas difusas utilizando las funciones de pertenencia o membresía. Las reglas de inferencia son el núcleo del sistema de inferencia que conecta las variables de entrada y salida mediante un conjunto de reglas IF-THEN (que contienen una parte 'if' y una parte 'then' que también se denominan antecedente y consecuente, respectivamente). (Sun, Li, Gao, & Xia, 2018)

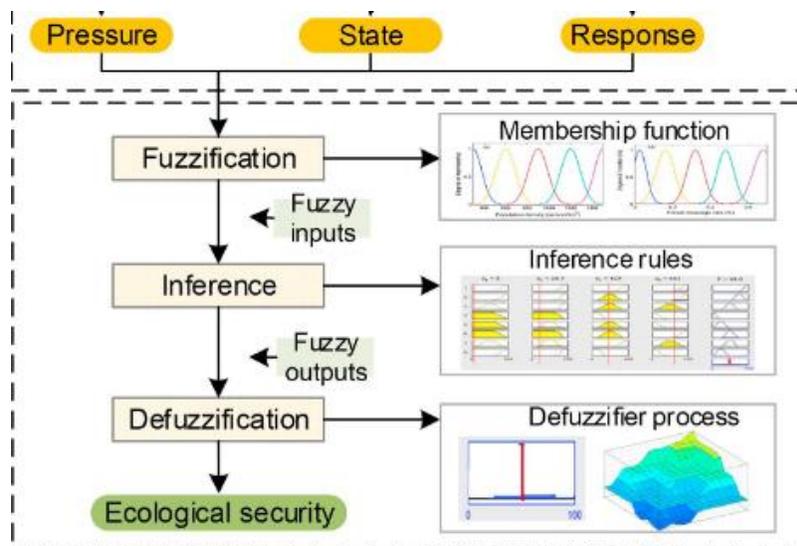


Figura 47 Estructura de inferencia Mamdani.(Sun et al., 2018)

Las reglas se obtienen principalmente de los conocimientos de los expertos. Finalmente, se emplea un método de defuzzificación para convertir la salida difusa en un valor nítido.(Sun et al., 2018)



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

El proceso de fuzzificación consiste en tomar cada una de las entradas y evaluar ese valor en cada una de las funciones de pertenencia asociadas a esa entrada, a ese valor resultante se lo conoce como grado de pertenencia del valor de entrada en cada uno de los conjuntos difusos. Por ejemplo, se tiene la siguiente regla:

“Si error es ENG entonces DT”, se está diciendo que si el valor de error pertenece a error negativo grande (ENG) entonces el conjunto de salida será decremente o disminuya total (DT), el proceso de fuzzificación consiste en tomar el valor de error y evaluarlo en el conjunto ENG, de este modo por cada regla se tendrá un valor asociado a un grado de pertenencia.

El siguiente paso es evaluar reglas de inferencia, esto se hace tomando cada uno de esos grados de pertenencia ya obtenidos anteriormente en el proceso de fuzzificación, y cortar todos los conjuntos difusos de la variable de salida a la altura del grado de pertenencia de la entrada, una vez se tengan los conjuntos difusos de la salida cortados, se los une para formar un único conjunto de salida.

El último paso es defuzzificar el conjunto obtenido, hay distintos métodos de defuzzificación, comúnmente se usa el método de centroide el cual se calcula mediante Ecuación 31, donde S es el conjunto de salida obtenido y U son los valores del universo de discurso de la salida.

$$resultado = \frac{\sum S(i) * U(i)}{\sum S(i)}$$

Ecuación 31 Cálculo del centroide. (Autor)

Para el desarrollo del código que permita implementar control difuso en un micro controlador se usaron funciones de pertenencia tipo trapezoidales y triangulares, aunque el código es abierto para futuras modificaciones donde se puedan agregarle funciones de tipo gaussianas. Para optimizar memoria en el micro controlador, el algoritmo que se desarrolló en este trabajo cuenta con un solo vector de salida, el cual contiene los datos del conjunto difuso resultante que inicialmente se llena de valores cero, y la dinámica del código es ir pasando ese mismo conjunto por todas las reglas definidas, para que estas vayan modificando sus valores y al final obtener un conjunto de salida listo para pasarlo por el proceso de defuzzificación. Por último, se aplicó la fórmula de la Ecuación 31 para obtener el valor final de control.

```
1 #include <Arduino.h>
2 #include "fuzzy.h"
3 void fuzzy::inicio(float B[],int tama){
4 }
5 void fuzzy::borrar(float B[],int tama){
6 }
7
8 void fuzzy::calc_size(float vec[], float paso){
9 }
10
11 void fuzzy::regla_simple(float in[],float Uin[],float err,float V[]
12 }
13
14 void fuzzy::regla_compuesta2(float in1[],float in2[],float U1[],flo
15 res = min(r1,r2);
16 inferencia_mamdani2(B,res,V,UV,tam);
17 }
18
19 void fuzzy::regla_compuesta3(float in1[],float in2[],float in3[],fl
20 res = min(r1,r2);
21 res = min(r3,res);
22 inferencia_mamdani2(B,res,V,UV,tam);
23 }
24
25 float fuzzy::absoluto(float dat){ // Función utilizada para calcular
26 }
27
28 float fuzzy::triangular(float x[],float params[],float date){
29 }
30
31 float fuzzy::trapezoidal(float x[],float params[],float date){
32 }
33
34 void fuzzy::inferencia_mamdani2(float B[],float c,float stru[],float
35 }
36
37 float fuzzy::defusi(float conju[],int tipo,float U_VOL[],int tama){
38 }
```

Figura 48 Funciones para aplicar control fuzzy. (Autor)



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Todas las funciones necesarias que se desarrollaron para aplicar un controlador difuso en el micro controlador se observan en Figura 48 el código completo se encuentra en anexo3. Para la construcción de las funciones de membresía se partió de la forma que tiene función trapezoidal, que básicamente consta de dos rectas una con pendiente positiva, otra con pendiente negativa y una zona constante. Por tanto, se divide la función en 3 partes, las dos pendientes se rigen por la ecuación de recta que se observa en Ecuación 32.

$$y - y1 = m(x - x1)$$

Ecuación 32 Ecuación de recta. (Autor)

$$m = \frac{y2 - y1}{x2 - x1}$$

Ecuación 33 Ecuación de pendiente (Autor)

Teniendo en cuenta lo anterior se calcula la posición del valor de error, y dependiendo de su posición se ubica en línea con pendiente positiva, línea con pendiente negativa o en la zona constante, y dependiendo de la zona en la que esté se calcula el valor de pertenencia, teniendo en cuenta que en conjuntos difusos el mayor valor de pertenencia será uno. Como se dijo anteriormente para poder evaluar la señal de error se dividió a la función trapezoidal en 3 partes, primera pendiente, zona constante y segunda pendiente como se observa en Figura 49. La función triangular se la desarrolló con la misma lógica, solo que en la función triangular se la divide en dos secciones, las dos rectas una con pendiente positiva, y otra con pendiente negativa.

```
float fuzzy::trapezoidal(float x[],float params[],float date){
    float y;
    if(date < params[0]){ // Se rellena con ceros a la izquierda
        y = 0;
    }
    if(date > params[2]){ // Se rellena con ceros a la derecha
        y = 0;
    }
    if (params[0]<=date && date<= params[1]){ // Primera Pendiente
        float p = 1/(params[1]-params[0]);
        y = (p*(date-params[0])+1);
    }
    if(params[1]<date && date<= params[2]){ // Región Constante
        y = 1;
    }
    if (params[2]<=date && date<= params[3]){ // Segunda Pendiente
        float p = -1/(params[3]-params[2]);
        y = (p*(date-params[2])+1);
    }
    if(params[0] <x[0] || params[2] >x[1]){
        y = 0;
    }
    if((date == params[2]&& date ==params[3])|| (date == params[0]&&
        y = 1;
    }
    return y;
}
```

Figura 49 Función trapezoidal control fuzzy. (Autor)

Con estas funciones creadas ya se puede obtener el grado de pertenencia en los conjuntos difusos que es el primer paso, ahora se debe pasar ese grado de pertenencia por las reglas y cortar los conjuntos difusos de salida a la altura del valor de pertenencia, esto se hace con la función de inferencia de mamdani. La función inferencia de mamdani se encarga de tomar los valores de pertenencia, cortar los conjuntos difusos a la salida y unirlos al mismo tiempo, para así crear un único vector con los valores del conjunto de salida como se observa en *Figura 50*. Finalmente se aplicó la formula vista en Ecuación 31 para calcular la señal de salida del controlador, como se observa en *Figura 51*.

```
float v[tama]; // Tamano del vector del universo de discurso de v
for(int i=0; i <tama;i++){ //Aqui genero vector v
    v[i] = j;
    j = j+paso;
}
if(tipo ==1){
    for(int i =0;i<tama;i++){
        if(v[i]<stru[0] || v[i]==stru[0]){
            float y=0;
            if(y>B[i]){
                B[i]=y;
            }
        }
        if(stru[0]<= v[i] && v[i]<=stru[1]){ //Primera Pendiente
            float p= 1/(stru[1]-stru[0]);
            float y = (p*(v[i]-stru[1]))+1;
            if(y>=c){
                y=c;
            }
            if(y>B[i]){
                B[i]=y;
            }
        }
        if(stru[1]<=v[i] && v[i]<=stru[2]){ //zona constante
            float y=c;
            if(y>B[i]){
                B[i]=y;
            }
        }
    }
}
```

Figura 50 Función inferencia mamdani. (Autor)

```
float fuzzy::defusi(float conju[],int tipo,float U_VOL[
    float j=U_VOL[0],paso;
    paso = (absoluto(U_VOL[0])+absoluto(U_VOL[1]))/tama;
    float x[tama];
    //float j=-12;
    float y0,area=0,res1=0;
    for(int i=0; i <tama;i++){
        x[i] = j;
        j = j+paso;
        area= area+conju[i];
        res1= res1+(conju[i]*x[i]);
        conju[i]=0;
    }
    if(area ==0){
        y0 = 0;
    }
    else{
        y0 = res1/area;
    }
    return y0;
}//////////
```

Figura 51 Función defuzzificación. (Autor)

7.2.2 Simulación control fuzzy.

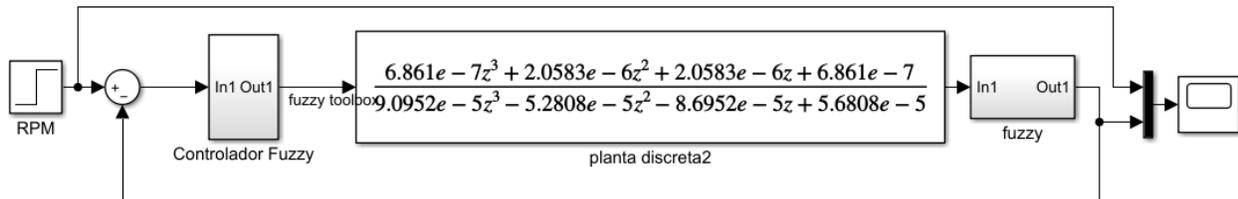


Figura 52 Esquema Simulink planta control fuzzy. (Autor)

En Figura 52 se tiene el esquema usado en simulink con el bloque de controlador fuzzy, dentro de este bloque se encuentra el controlador difuso. Simulink trae un bloque propio para simular un controlador difuso, pero en este trabajo no se lo puede usar directamente porque el objetivo final es aplicar dicho control en la tarjeta teensy y controlar el motor real, así que es importante hacer simulaciones que vayan acorde a lo que se tiene en la planta física, y dicho bloque no se lo puede llevar al microcontrolador escogido, por lo tanto, es necesario diseñar un algoritmo que permita aplicar el controlador a la planta. Para diseñar el algoritmo se utilizaron solo funciones básicas que fácilmente se las puede programar en el micro controlador, de igual forma se diseñó una librería que facilita la implementación de este controlador en la tarjeta teensy.

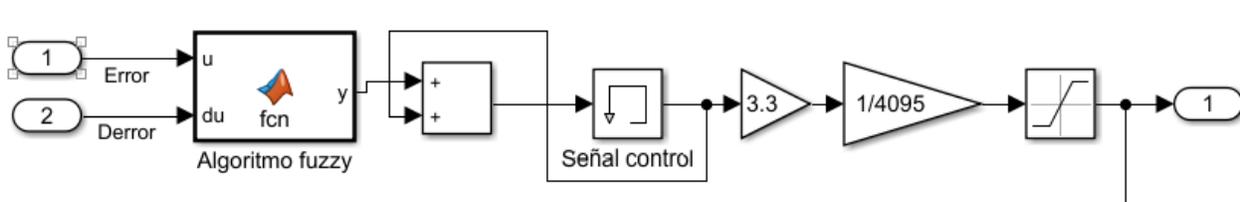


Figura 53 Controlador difuso en Simulink. (Autor)

Para poder usar el controlador difuso en Simulink se empleó el bloque que permite ejecutar funciones escritas en lenguaje de Matlab como se ve en Figura 53 el bloque llamado Algoritmo fuzzy, dentro se encuentran las funciones que permiten aplicar el controlador diseñado anteriormente.

Ahora se requiere comprobar que las funciones creadas para aplicar el control hayan quedado bien, para ello se usó la toolbox de Matlab y se implementó en ella el controlador difuso con los mismos conjuntos de entrada, las mismas reglas para comprobar que el código diseñado esté funcionando correctamente.

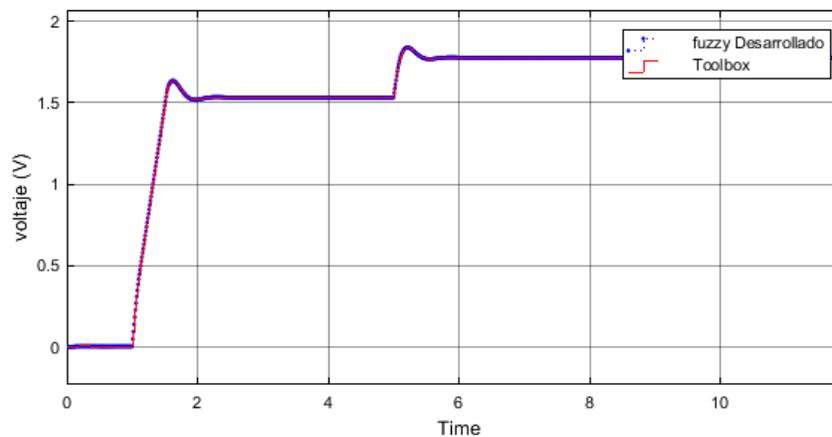


Figura 54 Señal control fuzzy Vs toolbox. (Autor)

Se aplicó tanto el controlador desarrollado por medio de funciones a base de código y el controlador desarrollado en la toolbox a la misma planta, teniendo como resultado el que se observa en Figura 54, las señales de control son idénticas, la señal de control desarrollada mediante

funciones (línea azul) esta sobrepuesta a la señal generada por la toolbox propia de Matlab, esto quiere decir que el algoritmo que se desarrolló está funcionando correctamente.

Una vez comprobado que el código de control difuso está funcionando bien, se procede a hacer la simulación para ver el comportamiento de la planta frente a dicho controlador.

La salida de la planta con el controlador difuso se puede apreciar en *Figura 55*, tiene un tiempo de establecimiento de 1.1 segundos, es un tiempo de establecimiento aceptable para un motor con un set point de 1000 RPM, se puede apreciar un pequeño sobre pico, pero rápidamente el controlador corrige la señal para disminuir el error.

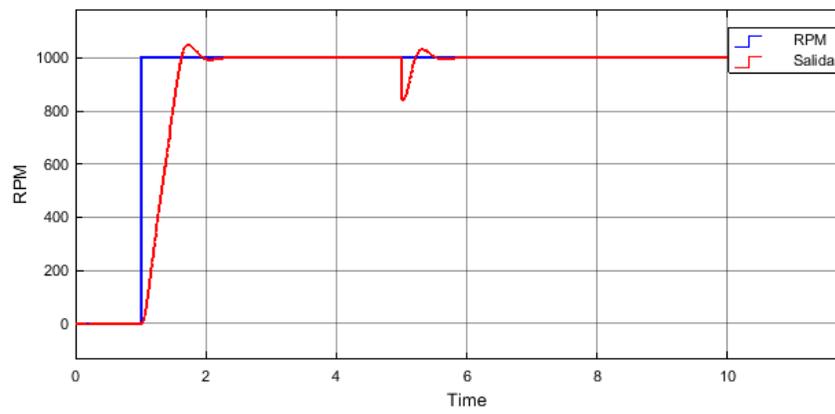


Figura 55 Salida de la planta control difuso. (Autor)

También se aplicó una perturbación a los 5 segundos que provocaba que la planta perdiera velocidad, el controlador reacciona rápido ante perturbaciones por lo que rápidamente corrigió la salida. En la anterior técnica de control (PID) se analizó la señal de control, para asegurarse que no haya ningún inconveniente al momento de aplicarle esa señal a la planta real.

La gráfica de señal de control difuso se puede ver en *Figura 54* se ve que no está saturada, su comportamiento es acorde a las necesidades, pues ante la perturbación, aumentó la señal de voltaje aplicado a la planta para corregir la disminución de velocidad como se esperaba.

7.3 Control neuronal directo

La última técnica de control que se aplicó a la planta, es el control por medio de redes neuronales artificiales (RNA) aplicando el algoritmo de aprendizaje de back propagation.

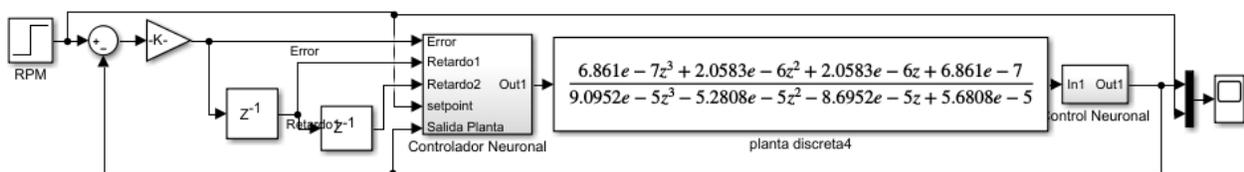


Figura 56 Esquema simulink para control neuronal directo. (Autor)

Para llevar a cabo la simulación de un control neuronal se realizó el esquema que se observa en *Figura 56* en el software de simulink. Para entrenar la red se usaron los valores de error, y dos retardos del error, por lo tanto, la RNA tiene 3 variables de entrada como se observa en *Figura 56*, también se envían al algoritmo los valores de setpoint y salida de la planta con el fin de calcular el error de entrenamiento.

Por lo general las redes neuronales tienen un entrenamiento previo para el ajuste de pesos, en este caso se propone una estructura de entrenamiento continuo, es decir la red neuronal estará ajustando sus pesos periódicamente con el fin de converger el error a cero. Por otra parte, casi siempre se usa la salida de la red neuronal para estimar el error, en este caso, para estimar el error se tomó la salida

de la planta, es decir el valor de rpm a la cual está girando el motor, se compara con el valor de setpoint para estimar el error y con este valor hacer el entrenamiento de la red neuronal.

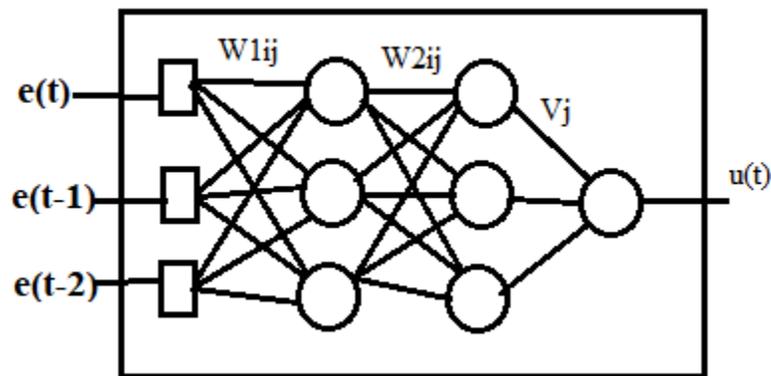


Figura 57 Estructura de la red neuronal para el control. (Autor)

7.3.1 Desarrollando algoritmo para control neuronal

Para la estructura de la red neuronal se propuso la que se observa en Figura 57, teniendo 3 neuronas en la capa de entrada, tres neuronas en la capa oculta y una neurona en la capa de salida la cual será la señal de control, en las entradas se puede ver el error, y dos retardos de error como se ve en Ecuación 34.

$$x(t) = [ey(t), ey(t - 1), ey(t - 2) \dots]$$

Ecuación 34 Expresión para el vector de entradas. (Noriega et al., 2008)

En Figura 57 se observan los coeficientes de pesos w_1, w_2 y v , de las conexiones de entrada a la capa de entrada, oculta y a la capa de salida respectivamente (Noriega et al., 2008), estas variables w y v tienen los pesos asociados a cada una de las neuronas correspondientes. El algoritmo que se

usó se llama back propagation, y se trata de minimizar el error de control que se puede ver en Ecuación 35, propagando las variables desde la capa de entrada hasta la de salida, devolviéndose recalculando pesos y volviendo a propagar, esto se hace de forma cíclica hasta que el error tienda a cero o a un valor aproximado a cero.

$$ey(t) = \text{setpoint} - \text{salida planta}$$

Ecuación 35 Valor de error en la red neuronal. (Autor)

La salida de las neuronas de la capa oculta depende del tipo de función de activación que se escoja, para este caso se escogió la función de activación logsig para las capas de entrada y oculta, de este modo la salida de las neuronas de la capa de entrada está dada por:(Noriega et al., 2008)

$$g(j) = \frac{1}{1 + \exp(-S(j))}, j = 1,2,3,4 \dots$$

Ecuación 36 Salida de neuronas de la capa entrada.(Noriega et al., 2008)

donde:

$$S(j) = \sum_{i=1}^3 w_{1ij} * x_i$$

La salida de las neuronas de la capa oculta está dada por:

$$h(j) = \frac{1}{1 + \exp(-S(j))}, j = 1,2,3,4 \dots$$

Ecuación 37 Salida de las neuronas capa oculta. (Noriega et al., 2008)



donde:

$$S(j) = \sum_{i=1}^3 w_{2ij} * g_i$$

La señal de control viene dada entonces por la neurona de salida y la función de activación asociada a ella, para este trabajo se utilizó la función de activación purelin, la cual es la función identidad, es decir que la salida viene dada por:

$$u(t) = \sum_{j=1}^3 v_j * h_j$$

Ecuación 38 Salida de la red neuronal. (Noriega et al., 2008)

Se define como criterio a minimizar el error que viene dado por:

$$E(t) = \frac{1}{2} * \sum_{k=1}^t e y^2(k)$$

El proceso de minimización consiste en moverse en la dirección del negativo del gradiente de la función E(t) con respecto a los coeficientes de pesos v_j , w_{1ji} y w_{2ji} (Noriega et al., 2008).

La ecuación general para la actualización de pesos es:

$$W(n + 1) = w_n - \mu * \nabla w E(t)$$

Ecuación 39 Expresión general de actualización de pesos. (Milone & Rufiner, 2011)

En donde w_n es el peso actual, μ es el factor de aprendizaje, ∇ es el vector gradiente que se define como:

$$\nabla w_{ji}(n) = -\mu * \frac{dE(n)}{dw_{ji}(n)}$$

Ecuación 40 Vector gradiente. (Milone & Rufiner, 2011)

Luego se tiene que:

$$\frac{dE(n)}{dw_{ji}(n)} = \frac{dE(n)}{dey(n)} * \frac{dey(n)}{dyj(n)} * \frac{dyj(n)}{dvj(n)} * \frac{dvj(n)}{dw_{ji}(n)}$$

Ecuación 41 (Milone & Rufiner, 2011)

Pero se sabe que:

$$\frac{dvj(n)}{dw_{ji}(n)} = y_i(n)$$

Ecuación 42 (Milone & Rufiner, 2011)

De esta forma la Ecuación 41 quedaría reducida a:

$$\frac{dE(n)}{dw_{ji}(n)} = \frac{dE(n)}{dey(n)} * \frac{dey(n)}{dyj(n)} * \frac{dyj(n)}{dvj(n)} * y_i(n)$$

Ecuación 43 Vector gradiente reducido. (Milone & Rufiner, 2011)

Ahora se define el gradiente de error local instantáneo como:



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

$$\delta_j = \frac{dE(n)}{dy_j(n)} \frac{dy_j(n)}{dv_j(n)}$$

Ecuación 44 Gradiente instantáneo. (Milone & Rufiner, 2011)

Por tanto:

$$\nabla w_{ji}(n) = \mu * \delta_j(n) * y_i(n)$$

Ecuación 45 Aplicación general del gradiente. (Milone & Rufiner, 2011)

El gradiente de la capa de salida o sea la capa 3, en este caso quedaría:

$$\nabla w_{3ji}(n) = \mu * e_y(n) * dfact * y_{2i}(n)$$

Ecuación 46 Gradiente capa salida. (Milone & Rufiner, 2011)

En donde se utiliza el número 3 para decir que es de la capa 3 y el número 2 para decir que es de la capa 2, también el término *dfact* significa que es la derivada de la función de activación escogida, para este trabajo se escogió la función de activación *purelin*, por lo tanto, su derivada es 1. De este modo la actualización de pesos para la capa de salida teniendo en cuenta Ecuación 39 sería:

$$w(n + 1) = w_n - \mu * 1 * y_{2i}(n) * e_y(n)$$

Ecuación 47 Actualización de pesos capa salida. (Milone & Rufiner, 2011)

Para las capas ocultas cambia la derivada de función de activación que se escogió *logsig*, la derivada de la función *logsig* es:

$$derlogsig = y * (1 - y)$$

Ecuación 48 Derivada función logsig. (Noriega et al., 2008)

Teniendo en cuenta Ecuación 48 se tiene que la ecuación de actualización de pesos para las capas ocultas es:

$$w(n + 1) = wn - \mu * y2i(n) * (1 - y2i(n)) * y1i(n) * E(n)$$

Ecuación 49 Actualización de pesos capas ocultas. (Milone & Rufiner, 2011)

7.3.2 Simulando control neuronal directo

Una vez encontradas las ecuaciones de actualización de pesos para todas las capas, se hizo el código para aplicar el algoritmo de aprendizaje en el micro controlador, primero desarrollando el mismo código para correr en el software de Matlab y de esta forma poder hacer simulaciones para tener un referente al momento de implementarlo en la planta física.

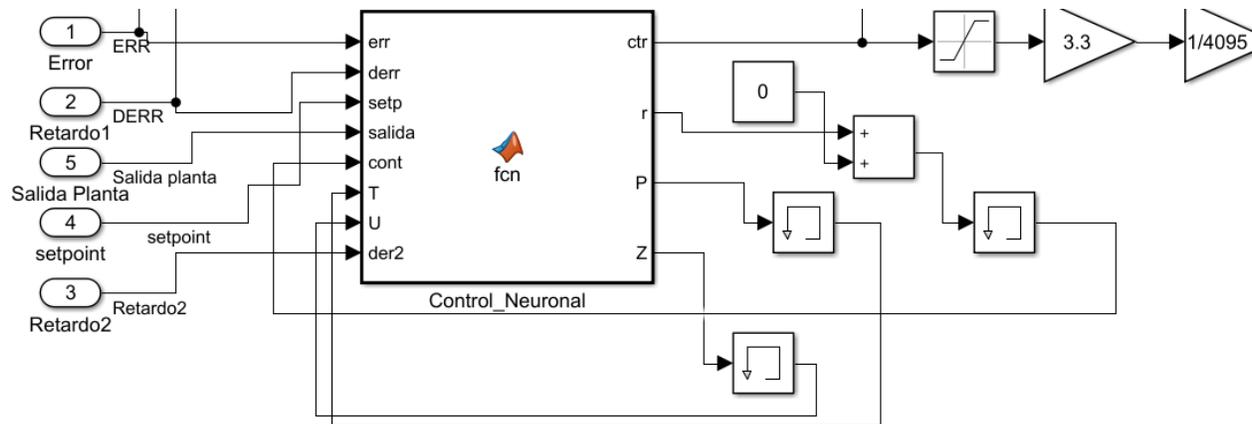


Figura 58 Controlador neuronal directo. (Autor)

El bloque que se observa en Figura 58 está inmerso en el esquema general que se montó en simulink y que se lo puede apreciar en Figura 56.

Nuevamente se hace uso del bloque function de Matlab como se observa en Figura 58, dentro del cual se encuentra el código con el algoritmo de aprendizaje de la red neuronal propuesta. En la red neuronal artificial también se presenta el fenómeno de wind-up, es decir la red puede ajustar pesos para generar una salida por fuera del rango de operación de la planta, dentro del código se tuvo en cuenta este problema, y se lo diseñó de tal forma que si la red detecta que se ha alcanzado los límites para la señal de control, deje de aprender y de esta forma evitar generar una señal de control que la planta no pueda asimilar. En Figura 58 se ven algunos bloques extra, esto con el fin de adecuar y guardar las matrices de pesos y otras variables importantes en la ejecución del algoritmo.

```
1 function [ctr,r,P,Z] = control_neuronal(err,derr,setp,salida,cont,T,U,der2)
2 r = 0;
3 P = zeros(3,3,4);
4 Z = zeros(3,3,4);
5 D=setp;
6 ysal = salida;
7 X=[err derr,der2];
8 a=size(X);
9 b=size(D);
10 N=[a(2) 3 1];
11 L=length(N);
12 W = zeros(3,3,4);
13 M = zeros(3,3,4);
14 r = 0;
15 for capa=2:L
16 for nodo=1:(N(capa))
17 for i=1:(N(capa-1)+1)
18 W(capa,nodo,i)=0.0;
19 end
20 end
21 end
```

Figura 59 Código control neuronal directo. (Autor)

Parte del código que se encuentra dentro del bloque Control_Neuronal que se aprecia en Figura 58, se lo puede ver en Figura 59, empezando por definir las matrices para guardar los pesos o variables auxiliares, seguido del vector X de entradas, en el cual se dijo que estarían el error y dos retardos del error nombrados como derr y derr2, el código completo se encuentra en anexo4.

Se le aplicó la misma señal que a los dos controladores anteriores (PID y fuzzy), una señal paso con una amplitud de 1000 rpm como setpoint como se observa en Figura 60.

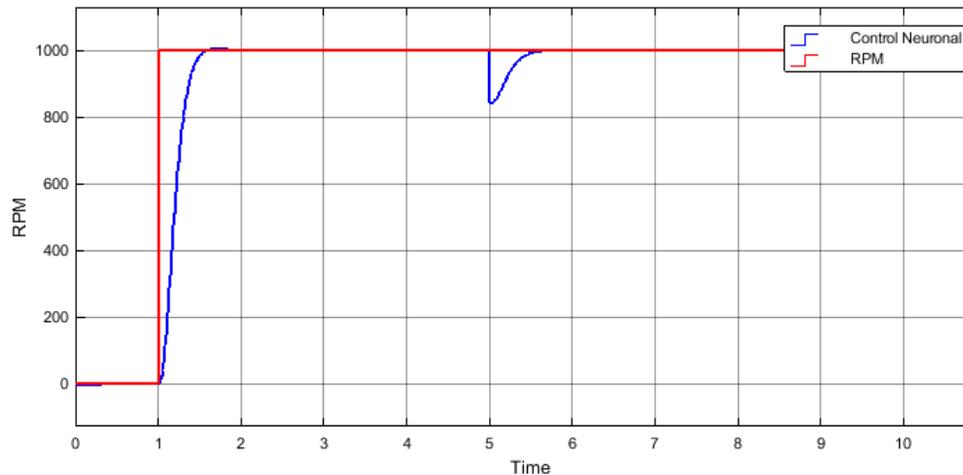


Figura 60 Respuesta del controlador neuronal. (Autor)

En Figura 60 se observa la respuesta de la planta ante un setpoint de 1000 rpm con controlador neuronal directo, se ve claramente que mejora el tiempo de respuesta, alrededor de los 500 milisegundos, no presenta sobre oscilación y reacciona rápido ante perturbaciones, la cual se puede ver en el segundo 5.

Como anteriormente se había hecho es necesario hacer un análisis de la señal de control para asegurarse de que no esté saturada. La señal de control que se observa en Figura 61 es ideal, pues no está saturada, lo que quiere decir que la planta la puede seguir sin problema, y reacciona bien ante perturbaciones, puesto que en el segundo 5 aumentó la señal de control o sea el voltaje para mantener el mismo valor de rpm ante la disminución de velocidad.

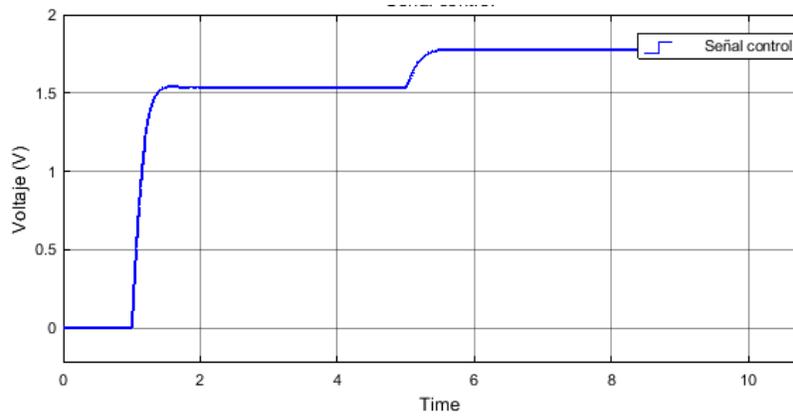


Figura 61 Señal control neuronal. (Autor)

Ahora se colocaron tres funciones de transferencia en paralelo, cada una con uno de los controladores diseñados, y se miró la respuesta en un mismo gráfico, para ello se montó en simulink el esquema que se observa en Figura 62, en este esquema se implementaron las distintas técnicas de control propuestas en este trabajo. El valor de setpoint es igual para todos.

En Figura 63 se observa la respuesta de los tres controladores con la señal azul como referencia, se ve claramente que la señal de control neuronal es la más rápida (color morado), le sigue la señal del control difuso, y por último la señal de control pid.

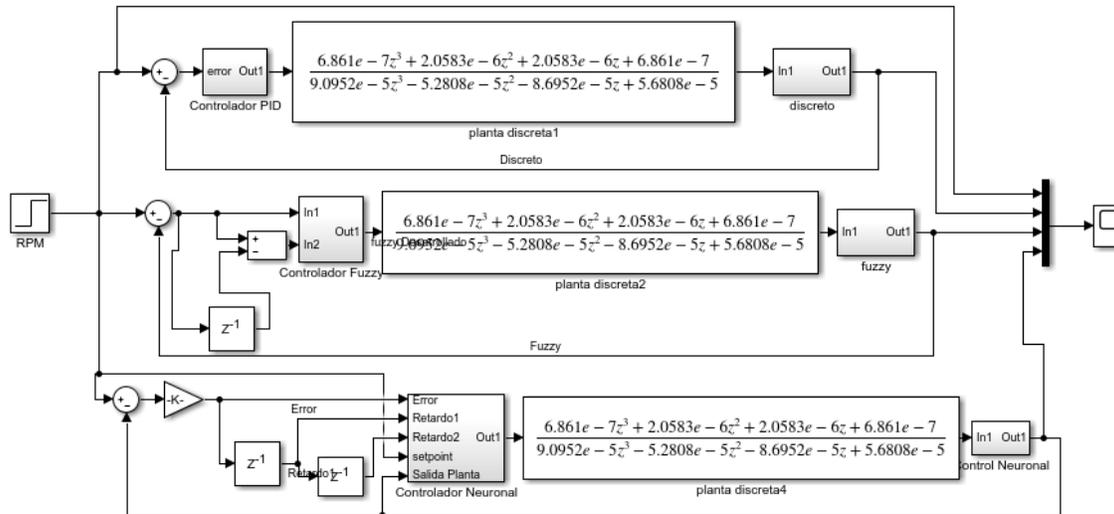


Figura 62 Esquema control pid, difuso, neuronal. (Autor)

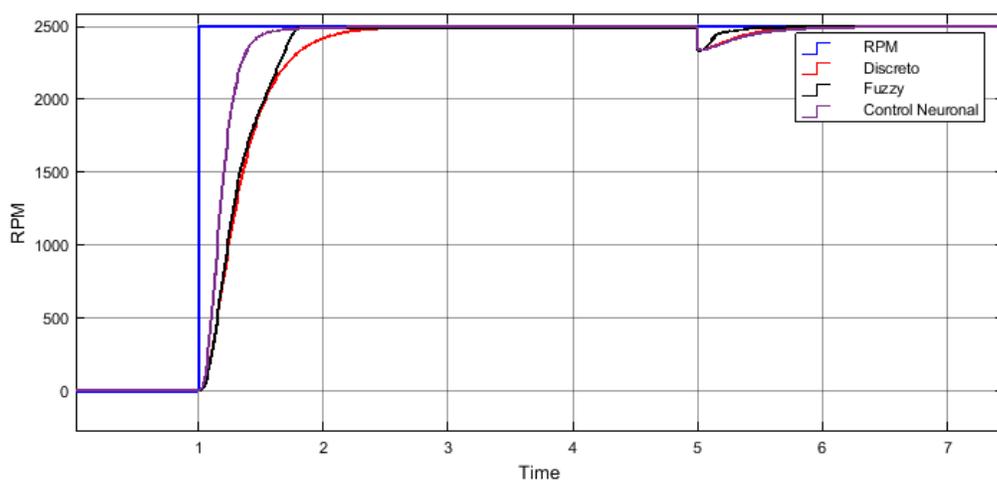


Figura 63 Respuesta de control pid, difuso y neuronal. (Autor)

Los tres controladores reaccionan bien ante una perturbación, pues todos son capaces de recuperar el valor deseado de setpoint.

Ahora se procede a analizar el comportamiento de los 3 controladores ante cambios de setpoint, pues algunos pueden tener buen funcionamiento en un valor de referencia dado, pero para otros valores de referencia puede que no funcionen bien.

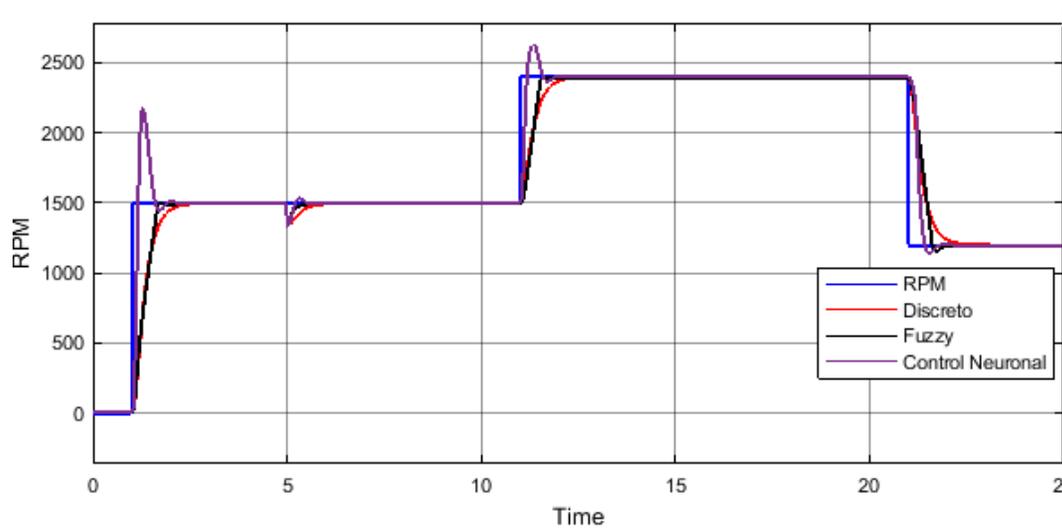


Figura 64 Respuesta ante cambios de setpoint. (Autor)

En Figura 64 se observa que la señal de respuesta generada por el controlador neuronal presenta sobre picos con valores de setpoint bajos, esto se debe principalmente al ajuste de factor de aprendizaje, que en la red neuronal es un valor crítico, aunque rápidamente corrige la señal de control para ajustarse al valor deseado, la señal que mejor se comporta es la del controlador difuso, pues presenta rápidos tiempo de respuesta sin sobre picos considerables.

Al inicio de este trabajo se desconocía el comportamiento en conjunto de la planta, se sabe que el motor trifásico por separado es un sistema altamente no lineal, pero el conjunto motor, taco – generador y variador si se comporta como un sistema líneal, y esto se puede comprobar en la

caracterización del taco – generador, pues la gráfica de señal pwm (modulación por ancho de pulso) vs rpm del motor es completamente lineal y se podría determinar por medio de la ecuación de una recta. En Figura 65 se puede apreciar mejor el comportamiento lineal de la planta.

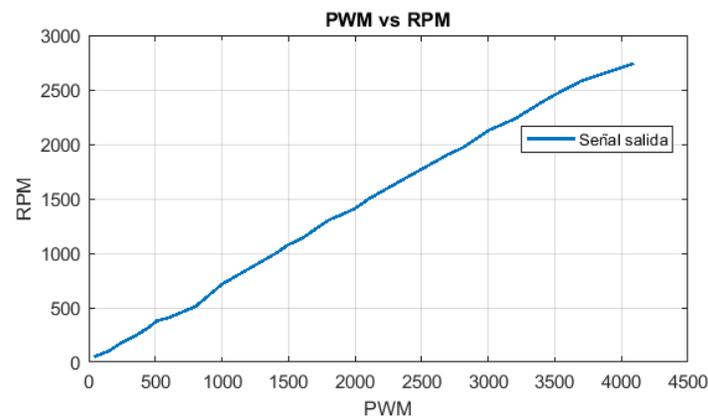


Figura 65 PWM vs RPM. (Autor)

Por lo general el control tipo PID funciona bien en distintos puntos de operación en sistemas lineales, pero en sistemas con alta no linealidad el algoritmo de control PID no funciona correctamente, esto explica porque en la respuesta de Figura 64 ante cambios de setpoint el controlador tiene un buen desempeño, a pesar de estar trabajando en distintos puntos de operación.

Finalmente, a nivel de simulación, se creó una trayectoria con distintos perfiles de seguimiento de setpoint o puntos de referencia, para ello se implementó la señal de entrada usada para identificar la planta visto en Figura 29, se simularon las 3 técnicas de control al tiempo y la respuesta se puede observar en Figura 66.

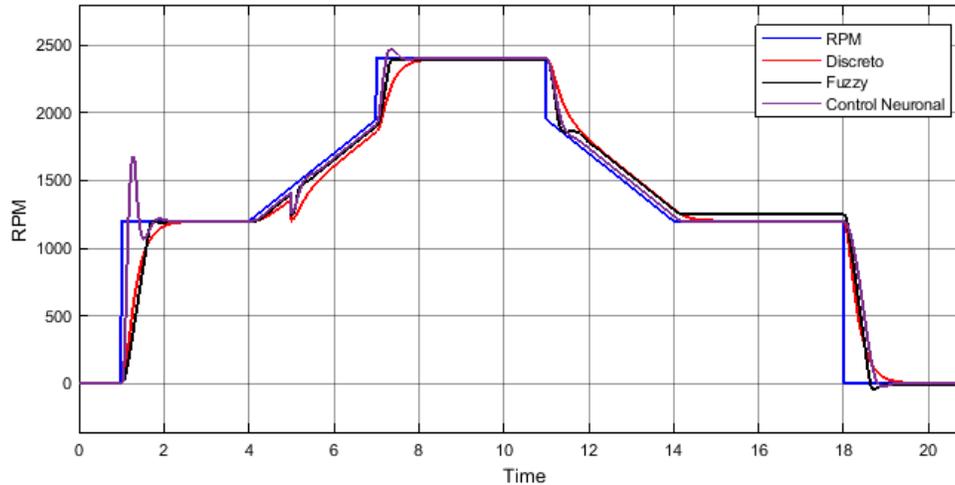


Figura 66 Referencia a perfil de seguimiento. (Autor)

En Figura 66 se muestra la señal de referencia (color azul), se ve que el controlador neuronal presenta un sobre impulso en el primer paso, pero es el que reacciona con mayor rapidez ante los cambios de set point, a los 5 segundos se aplicó una perturbación y todos fueron capaces de corregir la señal, nuevamente se observa que el controlador pid funciona bien en todos los puntos de operación, a pesar de tener un tiempo de establecimiento considerablemente alto en comparación con los demás, en promedio 1 segundo.

8 Capítulo 5 Aplicando controladores diseñados a la planta física.

Se aplicaron las distintas técnicas de control a la planta física, usando la tarjeta teensy 3.6 y el circuito de acondicionamiento visto en Figura 25, en dicho circuito se tienen las entradas para recibir la señal del taco – generador, y las salidas para aplicar el voltaje necesario al variador de frecuencia, de este modo se crea el control en lazo cerrado visto en Figura 34.

Para tener una mejor comprensión del comportamiento de la planta, se estimó la resolución que este ofrece, se hizo experimentalmente puesto que no se tenía un manual o alguna hoja de características, luego de distintas pruebas se estimó que el paso mínimo de frecuencia que presenta el variador es de 1Hz, con este paso mínimo se logra una variación de velocidad de 30 rpm, es decir el paso mínimo en velocidad es de 30 rpm.

8.1 Control PID discreto

Para aplicar el controlador PID discreto, se llevaron a código, las ecuaciones de control como Ecuación 30 encontradas anteriormente, como se observa en Figura 67.

```
1 int control = 0,rpm=0,setpoint=0,u=0,u_2=0,u_1=0,u_sat =0,cont =0,cc
2 IntervalTimer myTimer;
3 float entrada=0,error=0,error_1,error_2,sat=0;
4 float T = 4,Kc = 0.0015,Ti = 0.7,Td = 0.5,a0=0,a1=0,a2=0;//Constante
5 //Crear Variables para el filtro de Kalman////////////////////////////////////
6 float varVolt = 1.12184278324081E-03; // variance determined using
7 float varProcess = 1e-4;
8 float Pc = 0.0;
9 float G = 0.0;
10 float P = 1.0;
11 float Xp = 0.0;
12 float Zp = 0.0;
13 float Xe = 0.0;
14 ////////////////////////////////////////
15 void controlador() {
16     cont = cont + 1;
17     cont2 = cont2+4;
18     u = (a2*error) + (a1*error_1) + (a0*error_2)+ u_2 + (sat*Ti*(-1));
19     u_sat = u; //Lo hago para guardar el valor sin saturar para el anti
20     if(u>=4095){
21         u_sat=4095; //saturó la salida por encima de 4095
22     }
```

Figura 67 Controlador PID planta física. (Autor)

Parte del código cargado al micro controlador se observa en Figura 67, en donde se empieza por definir las variables necesarias para la ejecución del algoritmo de control, también se usó una interrupción por timer para tener exactitud con el tiempo de muestreo definido como 4

milisegundos, también se usó un filtro Kalman para la señal de entrada, puesto que la planta física presentó un ruido considerable, el código completo se puede ver en anexo5.

Para este controlador fue necesario modificar ligeramente las constantes encontradas en simulación, puesto que la resolución del variador de frecuencia es baja y este es un parámetro que no se puede tener en cuenta en simulación. Por cada 1Hz de frecuencia en el variador se tiene una variación promedio de 30 RPM, por tanto, las constantes finales fueron: $K_p = 0.0015$, $T_i = 0.7$ y $T_d = 0.5$, como se puede apreciar la constante que se modificó fue T_i que anteriormente en simulación se había estimado en 0.15.

Los datos tanto de tiempo, setpoint y valor de salida de la planta se enviaron por comunicación serial y se los recibió mediante el software PIC C compiler, para posteriormente almacenarlos, graficarlos en el software de Matlab, y hacer un análisis de su respuesta.

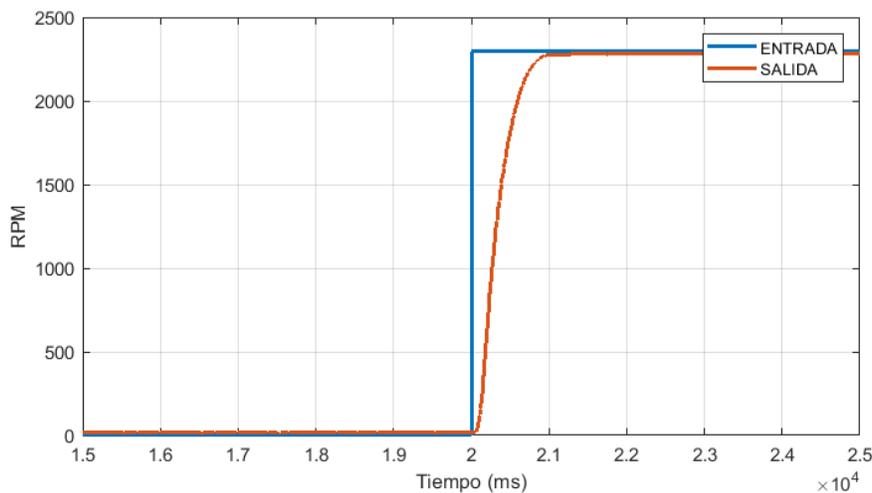


Figura 68 Respuesta planta física a controlador PID. (Autor)

Se definió como set point el valor de 2300 RPM, el cual en Figura 68 se puede apreciar que alcanzó ese valor de referencia en un tiempo aceptable y sin oscilaciones, el tiempo de asentamiento es de alrededor de 1.20 segundos, es de esperarse pues el motor tiene una dinámica que evoluciona rápido en el tiempo.

Haciendo un zoom a la gráfica se puede ver un pequeño error en estado estable como se muestra en Figura 69, el error es de 10 RPM y esto se debe principalmente a la baja resolución que presenta el variador de frecuencia.

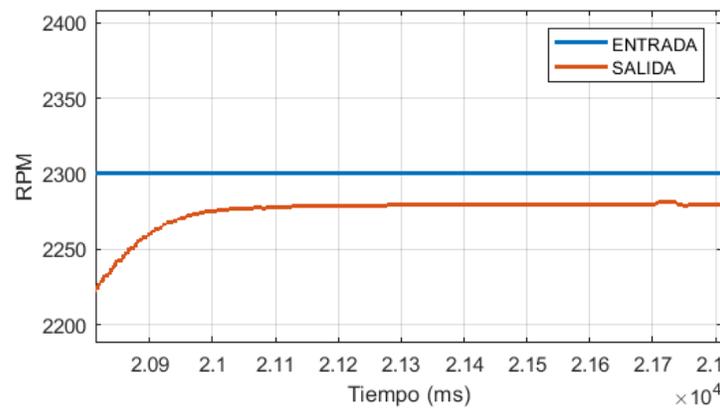


Figura 69 Respuesta2 control PID. (Autor)

8.2 Control fuzzy

Para aplicar la técnica de control fuzzy, como se dijo anteriormente se desarrolló una librería para una mejor organización del código, parte de dicho código se puede apreciar en Figura 70.

```
1 #include <fuzzy.h>
2 fuzzy controlfuzzy;
3 IntervalTimer myTimer;
4 float UE[] = {-2800,2800};
5 float UDER[] = {-50,50};
6 float USAL[] = {-120,120};
7 float ENG[] = {-2800,-2800,-2400,-1860,'R'};
8 float ENM[] = {-2133,-1600,-1200,'T'};
9 float ENP[] = {-1467,-666,-66,'T'};
10 float EC[] = {-133,0,133,'T'};
11 float EPP[] = {6,666,1467,'T'};
12 float EPM[] = {1200,1600,2133,'T'};
13 float EPG[] = {1860,2400,2800,2800,'R'};
14 /////////////// Se agrega los conjuntos difusos de la segunda en
15 float DNG[]={-50,-50,-30,-20,'R'};
16 float DNP[]={-21,-11,-7,'T'};
17 float DC[]={-8,0,8,'T'};
18 float DPP[]={7,11,21,'T'};
19 float DPG[]={20,30,50,50,'R'};
20 ///////////////Definidos conjuntos segunda entrada Delt
```

Figura 70 Código para control fuzzy. (Autor)

```
86 float B[tam];
87 controlfuzzy.inicio(B,tam);
88 entrada = analogRead(A21);
89 //kalman process
90 Pc = P + varProcess;
91 G = Pc/(Pc + varVolt); // kalman gain
92 P = (1-G)*Pc;
93 Xp = Xe;
94 Zp = Xp;
95 Xe = G*(entrada-Zp)+Xp; // the kalman estimate of the sensor vo
96 entrada = float (Xe*3.30)/4095.00;
97 rpm = ((1233.5*entrada)+5.5491);
98 error=setpoint-rpm;
99 der = error-error_1;
100 controlfuzzy.regla_simple(ENG,UE,error,DT,USAL,B,tam);
101 controlfuzzy.regla_simple(ENM,UE,error,DM,USAL,B,tam);
102 controlfuzzy.regla_simple(ENP,UE,error,DP,USAL,B,tam);
103 controlfuzzy.regla_simple(EC,UE,error,ZE,USAL,B,tam);
104 controlfuzzy.regla_simple(EPP,UE,error,AP,USAL,B,tam);
105 controlfuzzy.regla_simple(EPM,UE,error,AM,USAL,B,tam);
106 controlfuzzy.regla_simple(EPG,UE,error,AT,USAL,B,tam);
107 controlfuzzy.regla_compuesta2(EC,DNP,UE,UDER,error,der,DP,USAL,B);
108 controlfuzzy.regla_compuesta2(EC,DPP,UE,UDER,error,der,AP,USAL,B);
109 controlfuzzy.regla_compuesta2(ENP,DC,UE,UDER,error,der,DP,USAL,B);
110 float res = controlfuzzy.defusi(B,1,USAL,tam);
111 control = control+res;
```

Figura 71 Código control fuzzy2. (Autor)

Inicialmente se agrega la librería para poder usar sus funciones, también se definen los conjuntos difusos que ya se los había diseñado en simulación como se observa Figura 70.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

En Figura 71 se establece el vector para guardar el conjunto difuso de salida, luego se inicializa los conjuntos difusos, luego se pasa la señal de entrada por el filtro de Kalman, se calcula el valor de error, se agregan las reglas difusas y finalmente se calcula la señal de control. El código completo se encuentra en anexo 6.

Se propuso el mismo valor de referencia que el del controlador pid, con un setpoint de 2300 RPM y nuevamente enviando los datos por comunicación serial para graficarlos en Matlab, los datos se enviaron usando interrupciones para asegurarse que estén sincronizados con los enviados en el anterior controlador y poder hacer una comparación justa. Se obtuvo la respuesta que se observa en Figura 72, con un tiempo de estabilización de 1.50 segundos, y con un pequeño sobre pico que se aprecia mejor en Figura 73, el sobre pico también se había visto en simulación. La toma de datos completa dura 25 segundos, pero en el segundo 20 es que se aplica el setpoint, esto se hace con los demás controladores con el fin de tener las mismas condiciones para todos.

En Figura 73 se puede ver que la señal de salida presenta un error en estado estable de 10 RPM, se está presentando un comportamiento similar al anterior controlador, y es que con la resolución del variador se presentan dificultades y limitaciones al momento de aplicar los controladores, sin embargo, la salida converge al valor deseado.

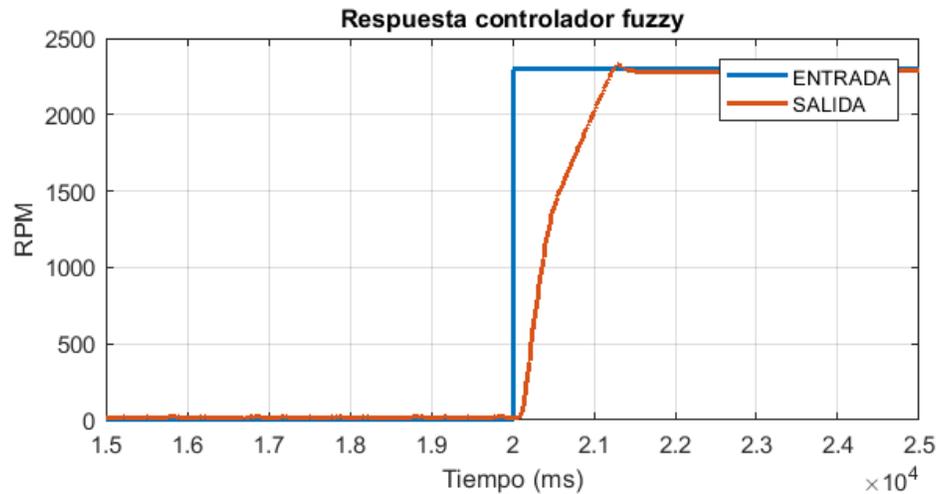


Figura 72 Respuesta planta física a control fuzzy. (Autor)

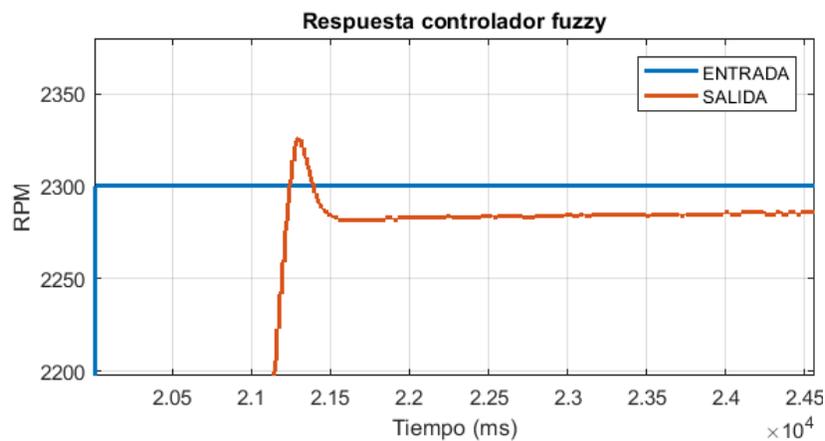


Figura 73 Respuesta2 controlador fuzzy. (Autor)

8.3 Control neuronal directo

Las ecuaciones necesarias para aplicar un algoritmo de control neuronal ya se abordaron anteriormente, teniendo como resultado las ecuaciones principales *Ecuación 47* y *Ecuación 49*, las cuales permiten actualizar pesos para que el error converja a cero, es decir que la red aprenda.

```
//Crear Variables para el filtro de Kalman////////////////////////////////////
const int t1 = 20000, t2 = 30000, t3 =33000 , t4 = 40000, t5 = 4
float varVolt = 1.12184278324081E-03; // variance determined us
float varProcess = 1e-4;
float Pc = 0.0;
float Q = 0.0;
float P = 1.0;
float Xp = 0.0;
float Zp = 0.0;
float Xe = 0.0;
////////////////////////////////////
IntervalTimer myTimer;
long int c=0;
int rev=0,rev2=0;
float n=0.1,rpm=0,D=0;
int a1=1; //numero de patrones
int a2=3; //numero de entradas
int b1=1; //numero de salidas
int b2=1; //numero de patrones
//int capa,nodo,i;
float err=0,win=0;
float X[5][6]={{0.0, 0.0, 0.0, 0.0},{0.0, 0.2, 0.1, 0.7},{0.0, 0
float er[5][3]={{0.0, 0.0, 0.0},{0.0, 1.0, 1.0},{0.0, 1.0, 1.0},
float error=1.0,error_1=0.0,error_2=0.0;
int N[]={0, a2, 3, b1};
int L=3; //NUMERO DE CAPAS
float W[6][6][6];
float M[6][6][6];
float G[6][6][6];
```

Figura 74 Código control neuronal. (Autor)

Para aplicar esta técnica en el micro controlador, se parte definiendo las variables y matrices importantes como se observa en *Figura 74*, también se configura un timer con el fin de enviar datos de forma sincronizada y hacer un análisis en conjunto con los demás controladores.

En términos generales el algoritmo consiste en multiplicar cada peso por el valor de entrada, que en este caso se escogió el error y dos retardos de él, luego se calcula el error de la neurona de la capa de salida, se devuelve calculando el error de las demás capas hasta llegar a las entradas, es en ese paso que se aplica la actualización de pesos, y de eso depende que la red aprenda de manera efectiva, en *Figura 75* se puede ver la sección de código que actualiza los pesos, se están aplicando las fórmulas *Ecuación 47* y *Ecuación 49* para finalmente en *Figura 76* calcular la señal de control

definida como la variable “control”. En *Figura 75* aparece un término denominado “n”, este es el factor de aprendizaje de la red, es un parámetro crítico y hay distintas técnicas para hacerlo adaptativo, para este trabajo se tomó un valor que solo depende del error.

```
189 for (int capa=2; capa<=L; capa++){
190 for (int nodo=1; nodo<=(N[capa]); nodo++){
191 for (int i=1; i<=(N[capa-1]+1); i++){
192 if((abs(error)*2800)<20){
193     W[capa][nodo][i]= W[capa][nodo][i];
194     n=0;
195 }
196 else{
197 if(ind==0){
198     ind = 10;
199     W[capa][nodo][i]= W[capa][nodo][i]+n*G[capa][nodo][i];
200     M[capa][nodo][i]=n*G[capa][nodo][i];
201 }
202 else{
203     W[capa][nodo][i]= W[capa][nodo][i]+n*G[capa][nodo][i]+(0.05*M[capa][nodo][i]-n*G[capa][nodo][i]);
204     M[capa][nodo][i]=n*G[capa][nodo][i];
205 }
206 }
207 }
208 }
209 }
```

Figura 75 Código actualización de pesos red neuronal. (Autor)

```
111 for (int i=1; i<=N[1]; i++){
112 y[1][i] = X[k][i];
113 }
114 y[1][N[1]+1]=1.00;
115 s=0.00;
116
117 for (int capa=2; capa<=L; capa++)
118 {
119 for (int nodo=1 ;nodo<=(N[capa]); nodo++)
120 {
121 for (int i=1; i<=(N[capa-1]+1); i++)
122 {
123 s=s+W[capa][nodo][i]*y[capa-1][i];
124 }
125 if(capa==L){
126 y[capa][nodo]=s;
127 control = s;
128 if(control<0){
129     control=0;
130 }
131 if(control>4095){
132     control = 4095;
133 }
134 analogWrite(A22,control);
```

Figura 76 Código señal de control neuronal. (Autor)

Una vez finalizado el código se establece un set point de 2300, como se había establecido para los anteriores controladores, los valores de salida de la planta se enviaron por comunicación serial.

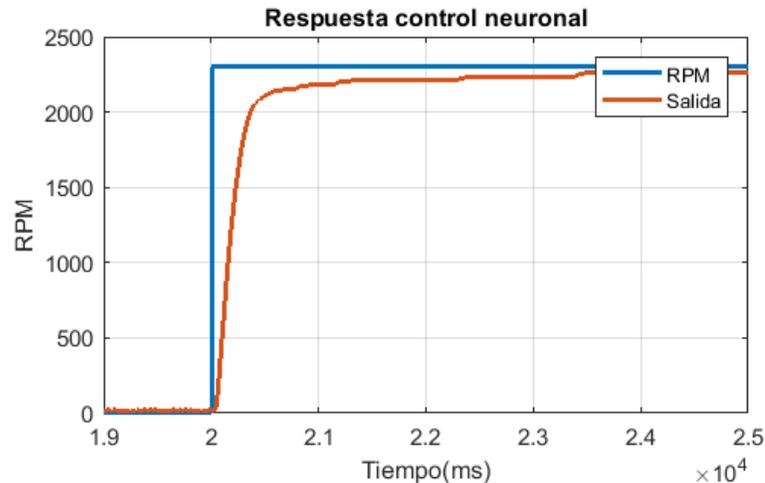


Figura 77 Respuesta control neuronal. (Autor)

En Figura 77, se observa la respuesta de la planta ante el controlador neuronal, se establece en un tiempo de 1.8 segundos, no presenta sobre impulso, la respuesta es rápida, pero en controladores anteriores se tuvo una respuesta más rápida, esto se debe al cálculo del factor de aprendizaje de la red neuronal “n”, con una mejor estrategia para su cálculo se obtendría respuestas más rápidas. Algo importante a tener en cuenta es que la red neuronal sigue aprendiendo, es decir trata de converger el error a cero, y cada vez se acerca más al valor deseado, aunque la resolución del variador no se lo permita, es por esto que se dice que un controlador neuronal es adaptativo, a pesar de no conocer el modelo matemático de la planta, pues en el algoritmo de aprendizaje nunca

se especifica los parámetros de la planta aun así es capaz de calcular el valor que necesita el actuador para llevar a la planta al valor deseado.

8.4 Comparación de las técnicas de control

Hasta ahora se han analizado las respuestas de los controladores por separado, es importante ver en un solo gráfico los 3 controladores para una misma señal de entrada.

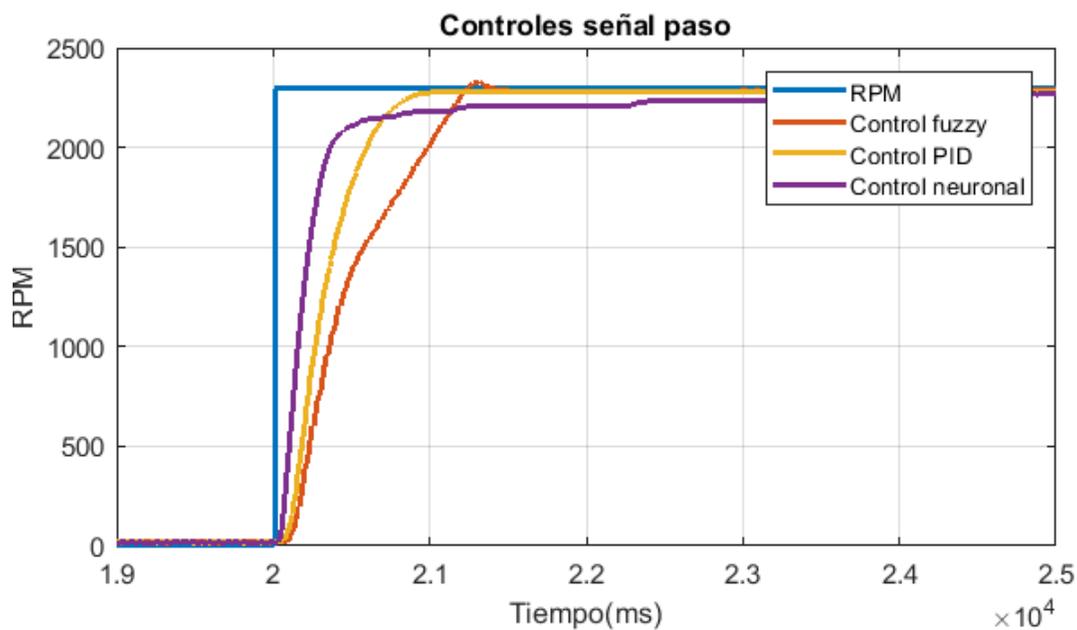


Figura 78 Controles a señal paso. (Autor)

En *Figura 78* se observan los 3 controladores identificados por colores aplicados a la planta física, en esta gráfica si se puede ver que el control neuronal (línea color morado), es el que reacciona más rápido, a pesar que se demora más para llegar al estado estable, debido principalmente al factor de aprendizaje mencionado anteriormente, el controlador fuzzy presenta un sobre impulso

pequeño y reacciona más lento que los demás, aunque hay que tener en cuenta que los tiempos de diferencia son de milisegundos.

Siguiendo lo que se había hecho en simulación, se llevaron los controladores ante un cambio de setpoint, para ello se usaron los mismos valores de referencia en simulación.

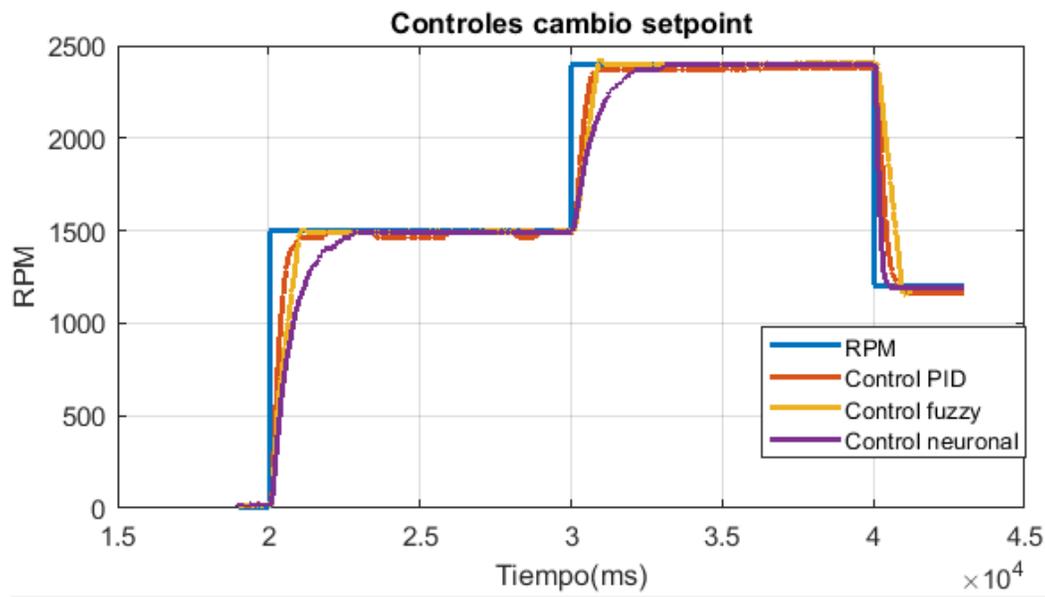


Figura 79 Controles cambio setpoint. (Autor)

Para la prueba de cambio de setpoint, se usó primero el valor de 1500, luego se pasó a uno de 2400 y finalmente el valor de 1200 como se ve en *Figura 79* línea de color azul. A pesar que es un sistema lineal como se dijo anteriormente, el controlador PID tiene problemas en algunos puntos de operación, por ejemplo, el de 1500 presenta una oscilación de 30rpm en estado estable, esto se debe en parte a la resolución del variador pues se sabe que tiene un paso mínimo de 30 rpm y algunos setpoint no son posibles con esa resolución. El controlador fuzzy presenta un mejor

comportamiento ante cambios de setpoint pues se ajusta mejor en estado estable y con un tiempo de estabilización aceptable de 1.2 segundos.

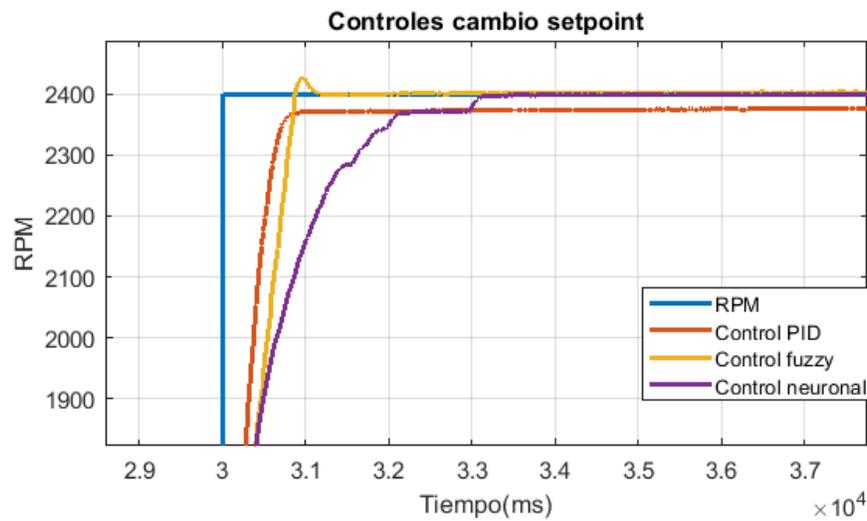


Figura 80 Controles cambio setpoint2. (Autor)

Lo dicho anteriormente se puede apreciar y analizar mejor si se hace un zoom a la gráfica en el setpoint de 2400 como se ve en Figura 80, en donde se ve claramente que la salida de PID (línea roja) tiene un error mayor a los otros controladores, la red neuronal se demora más, pero termina ajustando sus pesos para llegar al valor de referencia de línea azul, lo que no pasa con el control PID. Ahora el control neuronal presenta mejor comportamiento en bajada como se observa en Figura 81, donde se cambia de 2400rpm a 1200 rpm en este caso, el control neuronal es el primero en llevar la señal al setpoint, también se observa que el control PID es el segundo en llevar la señal al setpoint pero con un error en estado estable mayor que el de control fuzzy.

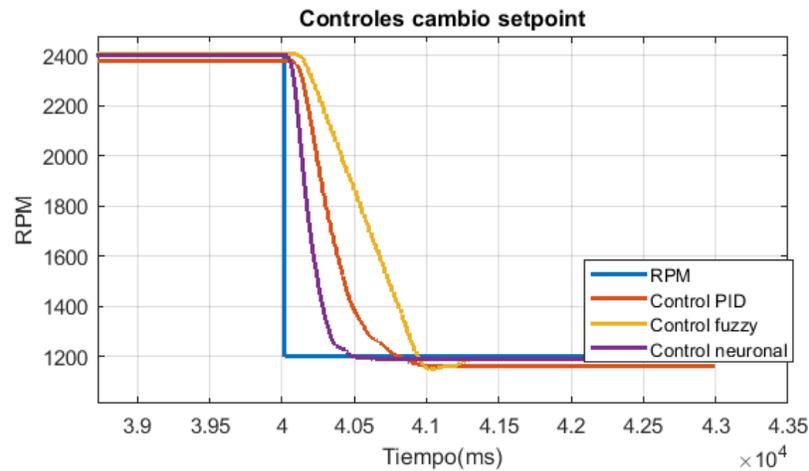


Figura 81 Controles cambio setpoint3. (Autor)

Para finalizar se sometieron todos los controladores a una señal de referencia con perfil de seguimiento más compleja, que consta de dos rampas y una parte constante como se muestra en Figura 82, en esta gráfica se puede ver que todos los controladores son capaces de seguir el perfil de la señal de referencia marcada con una línea azul, esto quiere decir que el modelo virtual de la planta encontrado si corresponde con la planta física, pues en las simulaciones los controladores también eran capaces de seguir la señal de referencia.

En Figura 82 no se puede analizar a detalle los controladores, para ello se dividió la gráfica en 4 secciones, paso de subida, las dos rampas y paso de bajada.

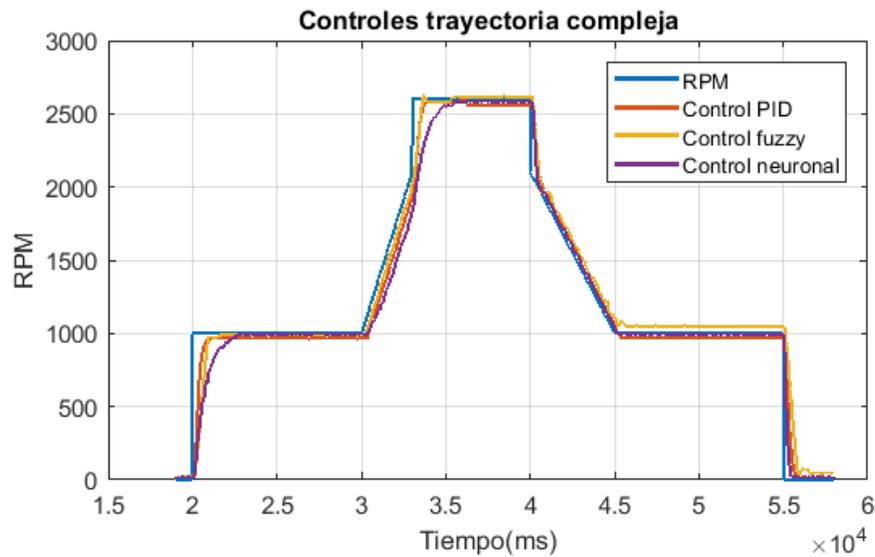


Figura 82 Controles trayectoria compleja1. (Autor)

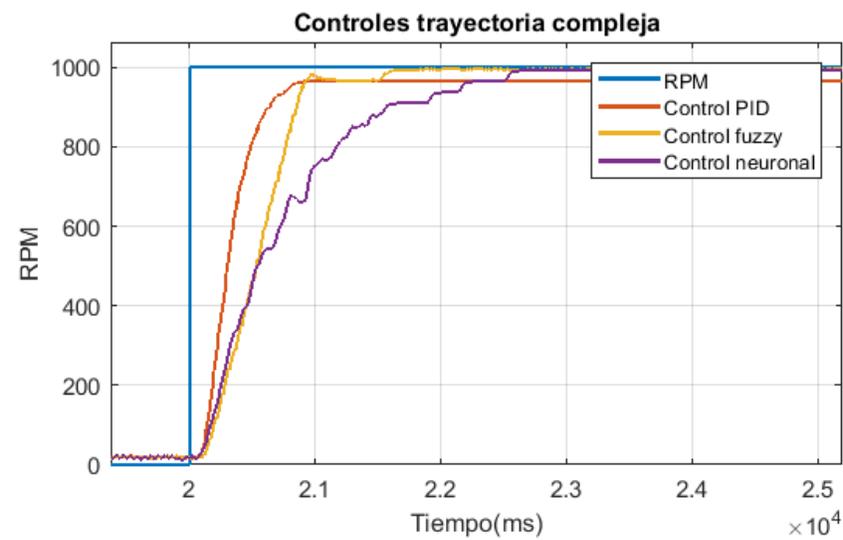


Figura 83 Setpoint subida señal compleja. (Autor)

En la sección de paso de subida visto en Figura 83, se confirma lo que se había deducido del anterior análisis del controlador neuronal, se demora más que los demás en estabilizarse, con la ventaja que en estado estable siempre busca corregir el error asociado, a diferencia del PID el cual si presenta un error más grande en estado estable. El controlador difuso es el que presenta mejor comportamiento, pues tiene un tiempo menor de estabilización que el controlador neuronal, y también corrige el error en estado estacionario. Anteriormente se había mencionado que se le agregó una entrada más al controlador difuso, y fue precisamente para corregir el error en estado estable.

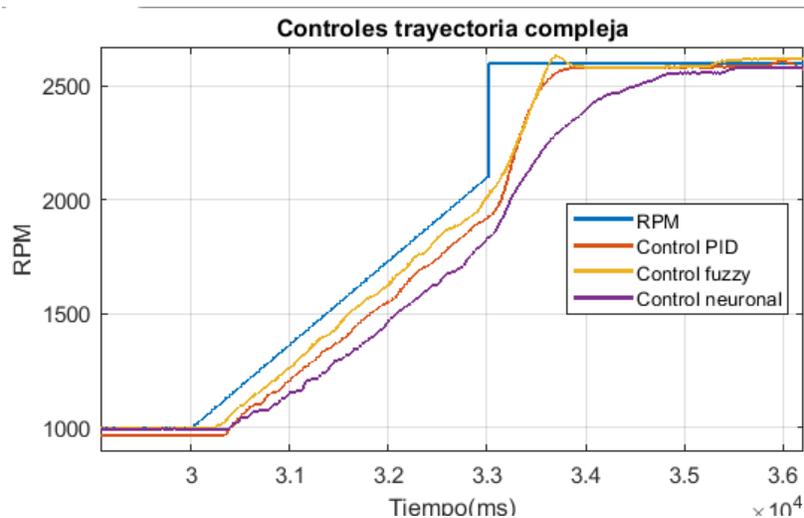


Figura 84 Rampa positiva señal compleja. (Autor)

En la rampa positiva el controlador difuso (línea amarilla), es el que reacciona más rápido en comparación con los demás controladores como se puede ver en Figura 84, nuevamente se tiene al

controlador neuronal reaccionando más lento, pero siempre entrenando la red y modificando la señal de control para tratar de seguir la señal de referencia.

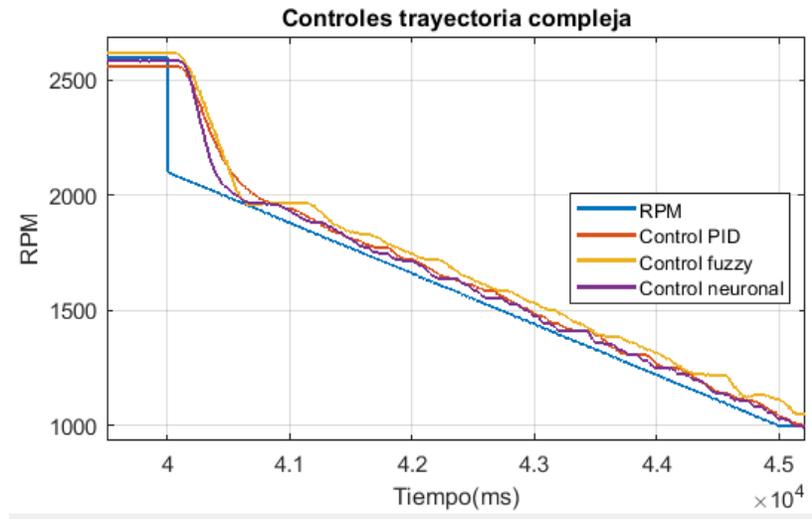


Figura 85 Rampa negativa señal compleja. (Autor)

En Figura 85 se observa la respuesta de los controladores ante la señal de entrada de rampa con pendiente negativa, nuevamente se observa un mejor comportamiento del controlador neuronal en bajada, pues es el primero que reacciona y el que más se acerca a la señal de referencia.

Por último, se analiza la parte de paso de bajada, se espera que la señal de control neuronal se comporte mejor que las demás, con un tiempo de establecimiento corto y un error en estado estable pequeño.

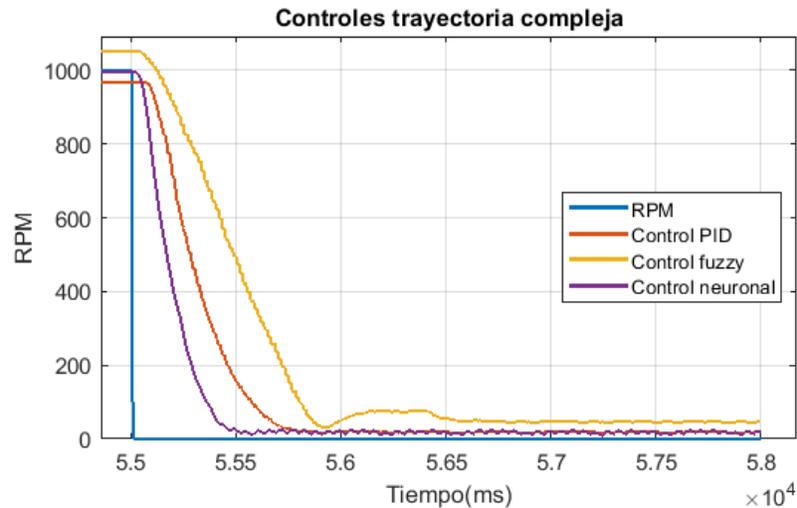


Figura 86 Paso bajada señal compleja. (Autor)

En Figura 86 se observa la respuesta ante el paso de bajada de la señal compleja empleada en Figura 82, se ve que efectivamente como se esperaba la señal de control neuronal desciende más rápido que los demás, aun así, con la linealidad del sistema las 3 técnicas de control ofrecen un seguimiento óptimo para la planta en su totalidad, variador, taco-generator y circuito de acondicionamiento. Las técnicas de control implementadas en este trabajo se pueden fácilmente replicar en cualquier microcontrolador y para cualquier planta, hoy en día se ha popularizado la plataforma de Arduino y sería una buena herramienta para aplicar los algoritmos de control desarrollados en este trabajo, variando algunos parámetros para adecuar al sistema en sí que se vaya a controlar.



9 Conclusiones

Para hacer una correcta identificación es necesario estimular la planta con una señal de entrada que la someta a diferentes puntos de operación, por lo general se suele usar sólo una señal paso en los procesos de identificación, esto es un problema si por ejemplo dicha planta funcionase bien en ese punto, pero tenga alguna no linealidad como zona muerta o saturación en otro punto, entonces con solo la señal paso se estaría asumiendo que la planta es lineal en todos los puntos de operación cuando en realidad no lo es, si se va a usar el modelo identificado para hacer simulaciones, estas no tendrían en cuenta la no linealidad de la planta y solo serviría para hacer simulaciones en ese punto en específico.

Un buen control e identificación, depende en gran medida de la calidad de los instrumentos que se empleen en la medición de las variables que se desean controlar, esto involucra resolución del instrumento, exactitud y precisión del mismo, también factores como el tiempo de muestreo. El dispositivo de medición taco-generator, presentó un ruido considerable de promedio 15rpm, se acondicionó la señal con filtro a nivel software (Kalman) y hardware. Es importante mitigar el ruido de la señal teniendo en cuenta que los algoritmos de identificación que permiten encontrar un modelo virtual trabajan con datos de la planta, y si estos presentan mucha varianza se dificulta la identificación y también el modelo encontrado presenta un mayor porcentaje de error.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

El modelo virtual encontrado presentó un comportamiento muy similar a la planta física, ya que no fueron necesarios muchos cambios a los controladores diseñados en simulación para que funcionen bien en la planta real.

El variador de frecuencia que se usó no contó con buena resolución de salida, esto afectó al momento de aplicar los controladores al motor, aun así, se tuvieron buenos resultados y se comprobó que los algoritmos de control desarrollados funcionan.

La planta en su totalidad presentó un comportamiento lineal, a pesar que por separado algunos de sus componentes sean no lineales, por ejemplo, el motor trifásico. Por ser una planta lineal el controlador PID presentó un buen comportamiento ante cambios de setpoint, aunque por la baja resolución del variador presentaba un error en estado estable.

Se comprobó que el algoritmo de control difuso desarrollado funciona y que es una buena estrategia de control para implementar en otros micro controladores como Arduino, al igual se comprobó la funcionalidad del algoritmo de control neuronal y que puede fácilmente modificarse para usarlo en otros procesos de control con micro controladores.

El parámetro clave para aplicar correctamente un control con red neuronal es el factor de aprendizaje de la misma, buscar una estrategia para calcularlo garantizará un rendimiento adecuado de la planta, sobre todo en plantas de difícil dinámica o no lineales.



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

Con una correcta identificación fue posible encontrar los mejores parámetros para cada técnica de control en simulación, y así garantizar que la planta física presentará un comportamiento similar al del modelo virtual. En control PID solo fue necesario un cambio en una de sus constantes calculadas en simulación para que funcione bien en la planta física, en el control difuso no fue necesaria ninguna modificación de lo que se había hecho en simulación al momento de llevarla a la planta física y, por último, en el control neuronal se modificó el factor de aprendizaje de la red, pues es un parámetro crítico al momento de implementar la red neuronal como controlador.

Se logró entender y plasmar los modelos matemáticos de cada técnica de control, desarrollarlos en código de programación y llevarlos a un sistema embebido para aplicarlos en una planta física y validar su funcionamiento.



10 Recomendaciones

Cuando se vaya a hacer control a una planta, primero se debe medir y definir el rango en el que opera, es importante para saber que señal de control es la adecuada, también medir el rango de operación de los sensores para evitar tener problemas con las tarjetas de adquisición de la señal.

Para usar la librería de control fuzzy desarrollada, tener en cuenta el parámetro de “paso”, el cual es la resolución por decirlo así, del conjunto difuso de salida, es recomendable escoger un valor promedio de resolución de 100 datos para el vector de salida, puesto que se comprobó que escoger más datos es redundante.

Modificar y ajustar la librería, agregarle mas funciones de membresía, para trabajar con más variedad de funciones y explorar métodos de optimización del código.

Si se va a usar un micro controlador para hacer control PID, que por lo general es el más usado, es más óptimo usar el algoritmo de control discreto, pues el micro controlador es un dispositivo digital.

Cuando se vaya a sintonizar un controlador tener en cuenta la resolución que tiene el actuador, en este caso el variador de frecuencia, es una limitante contar con resoluciones bajas ya que el actuador no es capaz de reaccionar a señales de control con cambios pequeños.

11 Referencias

- Aleksendrić, D., Carlone, P., Aleksendrić, D., & Carlone, P. (2015). Soft computing techniques. *Soft Computing in the Design and Manufacturing of Composite Materials*, 39–60. <https://doi.org/10.1533/9781782421801.39>
- Brandstetter, P., & Kuchar, M. (2017). Sensorless control of variable speed induction motor drive using RBF neural network. *Journal of Applied Logic*, 24, 97–108. <https://doi.org/10.1016/J.JAL.2016.11.017>
- Df robot. (2017). Dreamer_Maple_V1.0_SKU_DFR0283-DFRobot. Retrieved June 23, 2019, from https://wiki.dfrobot.com/Dreamer_Maple_V1.0_SKU_DFR0283
- Digi-Key. (2017). Placas de desarrollo Teensy 3.6 - Sparkfun | DigiKey. Retrieved June 23, 2019, from <https://www.digikey.com/es/product-highlight/s/sparkfun/teensy-3-6-development-boards>
- Etolocka. (2014). Funcionamiento del motor asíncrono trifásico. Retrieved June 24, 2019, from <http://www.profetolocka.com.ar/2014/10/23/motores-de-induccion-trifasicos-funcionamiento/>
- Fernandez, M. (n.d.). *Control avanzado*.
- Fernández, S. (2015). Diseño e implementación de una red neuronal para el seguimiento del punto máximo de poder de un panel solar. H. Ravishankar Kamath, R.S Althal, P.K Singh Ashls Kumar, Sinha y Atlt R Danak, “Modeling of Photovoltaic Array and Maximun Power Point Tracker Using ANN,” de Website: Http://Journal.Esrgroups.Org/Jes/Papers/4_3_4.Pdf. Retrieved from <https://repository.unimilitar.edu.co/handle/10654/6762>
- fmeautomatismos. (n.d.). Tipos de control (lazo abierto y cerrado). Retrieved June 23, 2019, from <https://sites.google.com/site/03fmeautomatismos/home/tipos-control>
- FRMza, U. (n.d.). *Máquinas e instalaciones eléctricas*.
- Greg, G. (2013). An Introduction To Fuzzy Control Systems. Retrieved June 24, 2019, from <http://www.faqs.org/docs/fuzzy/>
- H., S., S.F., K., & B., S. (2018). Improvements in direct torque control of induction motor for wide range of speed operation using fuzzy logic. *Journal of Electrical Systems and Information Technology*, 5(3), 813–828. <https://doi.org/10.1016/J.JESIT.2016.12.015>



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

- Hetpro. (2018). Amplificador no-Inversor. Retrieved June 25, 2019, from <https://hetpro-store.com/TUTORIALES/amplificador-no-inversor/>
- Instruments, T. (2018). *LM124, LM224, LM324, LM2902 Quadruple Operational Amplifiers datasheet (Rev. W)*. (September). Retrieved from <http://www.ti.com/lit/ds/symlink/lm324.pdf>
- Kamath, H. R., Aithal, R. S., Kumar, P. K. S. A., & Danak, A. R. (2012). *Modeling of Photovoltaic Array and Maximum Power Point Tracker Using ANN*.
- Mandado Pérez, E., Menéndez Fuertes, L., Fernández Ferreira, L., & López Matos, E. (2007). *Microcontroladores PIC. Sistema integrado para el autoaprendizaje*.
- Marulanda, L. (2010). *CONTROL POR MEDIO DE REDES NEURONALES*. Retrieved from https://www.academia.edu/10356050/CONTROL_POR_MEDIO_DE_REDES_NEURONALES_SISTEMAS_AVANZADOS_DE_CONTROL_PRESENTADO_A
- Mazzone, V. (2002). *Automatización y control industrial*. Retrieved from http://www.rocatek.com/forum_automatizacion_industrial.php
- Milone, D., & Rufiner, L. (2011). *Perceptrón multicapa* (p. 83). p. 83.
- Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1), 4–27. <https://doi.org/10.1109/72.80202>
- Nguyen, D. H., & Widrow, B. (1990). Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, 10(3), 18–23. <https://doi.org/10.1109/37.55119>
- Noriega, A., Barrera, A., & Ordez, A. (2008). Diseño de un controlador neuronal y su implantación en un microcontrolador. *XIII Congreso Latinoamericano de Control Automático / VI Congreso Venezolano de Automatización y Control*, (1).
- Núñez, L. (2009). Lazos de control. *Estrategias de Control*. Retrieved from [http://materias.fi.uba.ar/6509/Estrategias de control 2009.pdf](http://materias.fi.uba.ar/6509/Estrategias%20de%20control%202009.pdf)
- Ogata, K. (1995). *Ingeniería de Control Moderna* (5ta Edición).
- Ogata, K. (1996). *Sistemas de control en tiempo discreto* (2 Edición).
- Pardo, A., Díaz, J., & Martínez, C. (2012). Aplicación de redes neuronales al control de velocidad en motores de corriente alterna. *Revista Colombiana de Tecnologías de*



IMPLEMENTACIÓN DE TÉCNICAS DE CONTROL AVANZADAS EN UN SISTEMA EMBEBIDO

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750 - www.unipamplona.edu.co

- Avanzada*. Retrieved from http://revistas.unipamplona.edu.co/ojs_viceinves/index.php/RCTA/article/view/197
- Pérez, A. (2017). Empezando a programar los microcontroladores ARM (32 bits) (MIC011s). Retrieved June 24, 2019, from <http://www.incb.com.mx/index.php/articulos/78-microcontroladores-y-dsps/1716-empezando-a-programar-los-microcontroladores-arm-32-bits-mic011s>
- Rica, U. D. C., & S, A. A. (2001). Controladores PID y PWM , aplicación al control de velocidad de un motor de CC. *Control*.
- Ro-botica. (2017). RASPBERRY PI 3 MODELO B. Retrieved June 23, 2019, from <https://www.ro-botica.com/Producto/RASPBERRY-PI-3-MODELO-B/>
- Rodríguez Rosa, D. (2012). *Implementación de bloques de Simulink para control adaptativo de un motor de corriente continua*. Retrieved from <https://e-archivo.uc3m.es/handle/10016/16273?show=full#preview>
- Sun, J., Li, Y. P., Gao, P. P., & Xia, B. C. (2018). A Mamdani fuzzy inference approach for assessing ecological security in the Pearl River Delta urban agglomeration, China. *Ecological Indicators*, 94, 386–396. <https://doi.org/10.1016/J.ECOLIND.2018.07.011>
- Tecnicontrol. (n.d.). Instrumentación y Control de procesos industriales. Retrieved from <http://portal.tc.com.co/tecnicontrol/confiabilidad-operacional/instrumentacion-y-control>
- Villajulca, J. (2011). Empezando: Sintonización PID Cuantitativa (Ziegler y Nichols) – Instrumentacion, Control y Automatizacion Industrial. Retrieved July 22, 2019, from <https://instrumentacionycontrol.net/empezando-sintonizacion-pid-cuantitativa-ziegler-y-nichols/>
- Xianzhong Cui, & Shin, K. G. (1993). Direct control and coordination using neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 686–697. <https://doi.org/10.1109/21.256542>
- Ye, Z., & Kim, M. K. (2018). Predicting electricity consumption in a building using an optimized back-propagation and Levenberg–Marquardt back-propagation neural network: Case study of a shopping mall in China. *Sustainable Cities and Society*, 42, 176–183. <https://doi.org/10.1016/J.SCS.2018.05.050>