



Universidad Popular Autónoma del
Estado de Puebla

Centro Interdisciplinario de Posgrados Investigación y
Consultoría

Departamento de Ingeniería y Tecnologías de Información

Doctorado en Ingeniería de Software

**Algoritmos inteligentes para el
control de un convertidor CD-CD
bidireccional de medio puente:
aplicación de lógica difusa y redes
neuronales.**

Tesis que para obtener el grado de
Doctor en Ingeniería de Software

Presenta

Luz María Moreno Aguilar

Director

Dr. Edgar Peralta Sánchez



UPAEP – Secretaría General

Dirección General de Apoyos Académicos

Dirección del Centro de Recursos para el Aprendizaje y la Investigación.

Biblioteca Central - **Karol Wojtyla**

Tesis Digitales Restricciones de uso:

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de textos, imágenes, gráficas, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente de donde la obtuvo mencionando el autor o autores involucrados en el documento.

Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimiento

Al fondo mixto CONACYT-Gobierno del D.F. por el apoyo para la realización de este trabajo a través del proyecto “Sistemas de ahorro de energía para los trenes del Metro de la Ciudad de México” con clave 152954.

Índice

1. Introducción.....	1
1.1. Contexto.....	1
1.2. Antecedentes	1
1.2.1. Proyecto general	1
1.2.2. Convertidor CD-CD	3
1.3. Problemática	3
1.4. Propuesta de solución.....	4
1.5. Objetivo.....	5
1.6. Objetivos específicos.....	5
1.7. Metodología	5
1.8. Organización del documento	6
2. Convertidor CD-CD.....	7
2.1 Convertidores CD-CD	7
2.1.1. Conceptos básicos.....	7
2.1.2. Tipos de configuraciones.....	8
2.1.3. Funcionamiento	9
2.2 Caso particular de Convertidor CD-CD.....	10
2.2.1. Convertidor buck.....	11
2.2.2. Convertidor boost	14
2.3 Sistemas de control para un convertidor CD-CD	16
2.4. Lógica difusa.....	18
2.5. Redes neuronales.....	19
3. Software de control por lógica difusa	21
3.1. Análisis	21
3.1.1. Especificaciones	21

3.1.2. Especificaciones técnicas del convertidor	22
3.2. Diseño	23
3.2.1. Diseño general del módulo de control difuso	23
3.2.1.1. Metodología para el diseño	23
3.2.1.2. Entradas y salidas.....	24
3.2.1.3. Definición de variables lingüísticas	25
3.2.1.3. Definición de funciones de membresía	25
3.2.1.4. Definición de reglas de inferencia	26
3.2.2. Diseño del módulo difuso para el convertidor buck.....	27
3.2.3. Diseño del módulo difuso para el convertidor boost	28
3.3. Simulación	28
3.3.1. Simulación del circuito de control para el convertidor buck.....	29
3.3.1.1. Resultados de la simulación.....	30
3.3.2. Simulación del circuito de control para el convertidor boost	30
3.3.2.1. Resultados de la simulación.....	31
3.4. Implementación	32
3.4.1. Implementación del software de control para el convertidor buck.....	32
3.4.2. Implementación del software de control para el convertidor boost	34
3.4.3. Resultados	35
3.4.3.1. Resultados de la ejecución del control para el convertidor buck.....	35
3.4.3.2. Resultados de la ejecución del control para el convertidor boost	37
4. Software de control por redes neuronales	39
4.1. Análisis	39
4.1.1. Especificaciones	39
4.2. Diseño	40
4.2.1. Diseño general de la red neuronal.....	40
4.1.1.1. Selección de la topología	41
4.1.1.2. Diseño de la red	41
4.1.1.3. Selección del método de entrenamiento	43
4.1.2. Diseño del control buck	44
4.1.3. Diseño del control boost	44
4.2. Simulación	45

4.2.1. Simulación del circuito de control para el convertidor buck.....	45
4.2.1.1. Resultados de la simulación.....	46
4.2.2. Simulación del circuito de control para el convertidor boost	47
4.3.2. Resultados de la simulación.....	47
5. Comparación de algoritmos	49
5.1 Pruebas de ejecución	49
5.1.1. Pruebas de ejecución para el convertidor buck.....	49
5.1.2. Pruebas de ejecución para el convertidor boost	51
5.2. Pruebas de tiempo de ejecución	52
5.2.1. Pruebas de tiempo de ejecución para el convertidor buck	52
5.2.1. Pruebas de tiempo de ejecución para el convertidor boostt	52
5.2. Discusión	54
6. Conclusiones	56
6.1. Trabajo a futuro	58
Apéndice A. Simulación del módulo de lógica difusa.....	59
Apéndice B. Implementación del módulo de lógica difusa	64
Apéndice C. Simulación de una red neuronal	69
Referencias	77

Índice de figuras

Figura 1. Diagrama de bloques propuesto para los sistemas de ahorro de energía	2
Figura 2. Configuraciones básicas de convertidores CD-CD.	9
Figura 3. Modulador de anchura de impulsos.....	10
Figura 4. Ciclo de trabajo representado en periodo de tiempo.....	11
Figura 5. Circuito de control de lazo abierto.	16
Figura 6. Circuito de control de lazo cerrado.	17
Figura 7. Conexiones entre neuronas.....	19
Figura 8. Arquitectura del circuito de control de lazo cerrado para el convertidor cd-cd...	22
Figura 9. Circuito de control basado en lógica difusa	24
Figura 10. Modelo de simulación del circuito de control difuso para el convertidor buck .	29
Figura 11. Resultados de la simulación del control para el convertidor buck	30
Figura 12. Modelo de simulación del circuito de control para el convertidor boost.....	31
Figura 13. Resultados de la simulación del control para el convertidor en modo boost	31
Figura 14. Diagrama físico para la implementación	32
Figura 15. Implementación del software de control para el convertidor buck	33
Figura 16. Implementación del software de control para el convertidor boost.....	35
Figura 17. Convertidor CD-CD buck/boost de prueba: aspecto físico del convertidor, esquemático del convertidor, parámetros y configuración de los puertos de conexión.....	36
Figura 18. Resultados en osciloscopio de la ejecución del software de control para el convertidor buck.....	37
Figura 19. Resultados en osciloscopio de la ejecución del software de control para el convertidor boost	38
Figura 20. Arquitectura general del sistema de control.....	40
Figura 21. Red perceptrón multicapa.....	41
Figura 22. Topología de la red para el software de control para el convertidor	42
Figura 23. Red neuronal diseñada	42

Figura 24. Modelo de simulación del circuito de control basado en redes neuronales para el convertidor buck.....	46
Figura 25. Resultado de ejecución del control en modo buck operado por la red neuronal	46
Figura 26. Modelo de simulación del circuito de control basado en redes neuronales para el convertidor boost	47
Figura 27. Resultado de ejecución del control en modo boost operado por la red neuronal	48
Figura 28. Resultados de la simulación del control para el convertidor buck	50
Figura 29. Resultados de la simulación del control para el convertidor boost.....	51
Figura 30. Resultados de la simulación del control para el convertidor buck después de 7 segundos del reloj interno.....	53
Figura A.1. Configuración de variable de entrada error.....	59
Figura A.2. Declaración de las funciones miembro para la variable de entrada error	60
Figura A.3. Configuración de variable de entrada cambioError.....	61
Figura A.4. Declaración de las funciones miembro para la variable de entrada cambioError	61
Figura A.5. Configuración de variable de salida DeltaD	62
Figura A.6. Declaración de las funciones miembro para la variable de salida DeltaD	62
Figura A.7. Declaración de las reglas de inferencia.....	63
Figura A.8. Visor de reglas	63
Figura B.1. Apertura del editor de lógica difusa en LabVIEW	64
Figura B.2. Alta de la variable Error.....	65
Figura B.3. Declaración de funciones de membresía para la variable de entrada Error	65
Figura B.4. Alta de la variable CambioError	66
Figura B.5. Declaración de funciones de membresía para la variable de entrada CambioError.....	66
Figura B.6. Alta de la variable DeltaD	67
Figura B.7. Declaración de funciones de membresía para la variable de entrada DeltaD... ..	67
Figura B.8. Declaración de reglas de inferencia	68
Figura B.9. Pruebas al módulo difuso	68
Figura C.1. Detalle de los conjuntos de valores para asignar a las neuronas de entrada....	69
Figura C.2. Detalle del conjunto de valores para asignar a la neurona de salida	70

Figura C.3. Diagrama de la red Feed-Forward.....	72
Figura C.5. Gráfica de rendimiento	73
Figura C.6. Gráfica de estado de entrenamiento	74
Figura C.7. Histograma de error	74
Figura C.8. Gráfica de regresión	75
Figura C.9. Simulación de la red de tipo Feed-Forward	75

Índice de tablas

Tabla 1. Características del convertidor en modo buck	22
Tabla 2. Características del convertidor en modo boost	23
Tabla 3. Variables lingüísticas	25
Tabla 4. Tabla de reglas con la estructura Toeplitz	26
Tabla 5. Intervalos de valores para las variables de entrada y salida del controlador difuso para el funcionamiento del convertidor en modo buck.....	27
Tabla 6. Valores definidos para las funciones de membresía de las variables de entrada y la de salida del control difuso para el convertidor en modo buck.....	27
Tabla 7. Intervalos de valores para las variables de entrada y salida del controlador difuso para el funcionamiento del convertidor en modo boost	28
Tabla 8. Valores definidos para las funciones de membresía de las variables de entrada y la de salida del control difuso para el convertidor en modo boost	28
Tabla 9. Características del convertidor buck de prueba.	32
Tabla 10. Intervalos de valores para las variables de entrada y salida del módulo difuso para el convertidor buck de prueba.	33
Tabla 11. Características del convertidor boost de prueba.	34
Tabla 12. Intervalos de valores para las variables de entrada y salida del módulo difuso para el convertidor boost de prueba.....	34
Tabla 13. Intervalos de valores para las variables de entrada y salida de la red para el funcionamiento del convertidor en modo buck.....	44
Tabla 14. Intervalos de valores para las variables de entrada y salida de la red para el funcionamiento del convertidor en modo boost	45

1. Introducción

En este capítulo se presenta el panorama general del proyecto de tesis denominado “Algoritmos inteligentes para el control de un convertidor CD-CD bidireccional de medio puente: aplicación de lógica difusa y redes neuronales”, que propone el desarrollo de un software de control para un convertidor CD-CD, basado en algoritmos inteligentes y bajo una metodología de ingeniería de software. Por ello se presenta el contexto, antecedentes, problemática, propuesta de solución, objetivos y metodología que motivan este desarrollo.

1.1. Contexto

El trabajo que ampara el presente documento es parte del proyecto denominado “Sistemas de Ahorro de Energía para los trenes del metro de la Ciudad de México”, el cual es apoyado por el fondo mixto del Consejo Nacional de Ciencia y Tecnología (CONACYT) y Gobierno del Distrito Federal, con clave 152954, el cual es desarrollado en su totalidad en la Universidad Popular Autónoma del Estado de Puebla.

1.2. Antecedentes

Los antecedentes del proyecto de tesis que se presentan son el panorama del proyecto general al cual pertenece, y una descripción sobre un convertidor CD-CD que será sujeto a ser controlado.

1.2.1. Proyecto general

El proyecto denominado “Sistema de Ahorro de Energía para los trenes de la Ciudad de México” es uno de los proyectos solicitados por el Sistema de Transporte Colectivo

Metro al Consejo Nacional de Ciencia y Tecnología como parte de la Innovación Tecnológica en el Sistema de Transporte Colectivo (Toro, 2011). El proyecto fue concursado y obtenido a través de Fondos Mixtos en la Convocatoria 2010-01.

Este proyecto adquiere especial relevancia en el momento en que una de las megatendencias tecnológicas en el mundo es el uso de nuevas tecnologías energéticas (Grupo Coordinador de Desarrollo Regional del Tecnológico de Monterrey, 2009).

La idea principal de los sistemas de ahorro de energía propuestos en el proyecto es la de almacenar temporalmente la energía regenerada por los trenes del Metro de la Ciudad de México durante el frenado y reutilizarla después, durante el arranque de los mismos. Esto debido a que, actualmente, durante la etapa de frenado del tren, el motor regenera energía eléctrica, la cual es enviada de regreso a la barra guía y, pueden ocurrir dos situaciones: a) si existen trenes cercanos traccionando, esa energía es reutilizada por los mismos, b) si no existen trenes cercanos se usa un frenado de tipo reostático; ésto significa que cuando el tren es frenado la energía regenerada es disipada a través de resistores en forma de calor, por lo que es desperdiciada (Propuesta de realización del proyecto, 2011).

La propuesta del proyecto plantea la construcción de un sistema que permita el almacenamiento de la energía que se produce en el momento de frenado de un tren, en un banco de supercapacitores debido a que los niveles de potencia requeridos para estas operaciones son altos, y su recuperación en el momento de la aceleración.

El objetivo general del proyecto es “Reducir el consumo de energía en los trenes del Sistema de Transporte Colectivo METRO de la Ciudad de México, mediante el desarrollo de un prototipo de ahorro de energía basado en supercapacitores (metrorecuperación)” (Propuesta de realización del proyecto, 2011).

El proyecto contempla la construcción de un sistema que se instalará en tierra, es decir, en una estación del Metro, y estará conectado a la barra guía a través de un convertidor de corriente directa a corriente directa (CD-CD) que trabajará en modo bidireccional, permitiendo que los supercapacitores se carguen cuando exista energía regenerada (frenado) y que se descarguen durante la demanda de energía de los trenes (tracción). La Figura 1 muestra el diagrama de bloques del sistema propuesto.

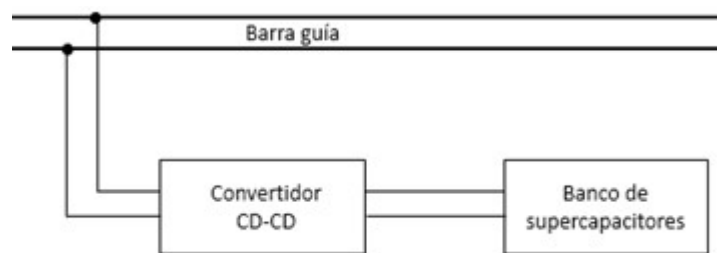


Figura 1. Diagrama de bloques propuesto para los sistemas de ahorro de energía
Fuente: Elaboración propia

De esta manera, el sistema propuesto en el proyecto plantea la construcción de dos dispositivos electrónicos, un convertidor CD-CD y un banco de supercapacitores, los cuales, dados los niveles de corriente y voltaje que procesarán y almacenarán, respectivamente, no pueden ser adquiridos y deben ser contruidos completamente en laboratorio.

1.2.2. Convertidor CD-CD

En este proyecto de tesis, se hace énfasis en el primer elemento, el convertidor CD-CD y en un sistema de software que controle su funcionamiento.

Los convertidores CD-CD son dispositivos electrónicos que se usan para convertir la entrada de corriente directa (CD) no regulada, en una salida de corriente directa controlada en el nivel de voltaje deseado (Mohan, Tore, Robbins, 2011).

Para su funcionamiento, el convertidor se coloca como parte de un circuito de control. Este control puede realizarse por hardware o por software. El objetivo del circuito de control es regular la tensión de salida o la corriente de salida del convertidor.

Generalmente, la estrategia de control consiste en comparar la tensión de referencia con la medición de ésta. El error es procesado por un controlador. La salida del controlador es una señal de referencia que permita hacer los ajustes necesarios para corregir la tensión de salida del convertidor.

En el presente proyecto de tesis, se plantea el desarrollo de un software de control para el convertidor CD-CD que se construirá como parte del proyecto “Sistema de Ahorro de Energía para los trenes de la Ciudad de México”.

1.3. Problemática

La principal dificultad de este proyecto se centra en el desarrollo del software de control para el convertidor CD-CD, debido a que se requiere realizar mediciones del voltaje de salida del convertidor con una frecuencia muy alta y, con la misma frecuencia, realizar el procesamiento específico para obtener el dato necesario para realizar la modificación del ciclo de trabajo del convertidor.

La problemática identificada en el análisis inicial es la siguiente:

1. Los dispositivos electrónicos de los Sistemas de Ahorro de Energía del Metro de la Ciudad de México, en especial el convertidor CD-CD, van a ser contruidos en su totalidad en el laboratorio de pruebas de la UPAEP, con características propias y, por lo tanto, requieren de un sistema de control, también con características propias.
2. Aunque existen propuestas elaboradas y contruidas de sistemas de control para un convertidor, la mayoría de las propuestas revisadas, (Elmas, Deperlioglu, Sayan,

2009), (Govindaraj, Rasila, 2010), (Khaburi, Rostami, 2011), (Muruganandam, Thangaraju, Madheswaran, 2014), (Salam, Taeed, Ayob, 2011), (Sayed, 2013), (Utomo, Haron, Bakar, Ahmad, 2011), han sido elaboradas para convertidores de menor capacidad de procesamiento de voltaje, que van desde los 5 hasta los 24 V, con excepción de (Gouse Basha, Kishore, 2011) que va de los 20 a los 40V. En este proyecto se requiere un sistema adecuado a las características de nivel de voltaje, que se manejan en el Metro, que son de entre 600 y 1000 V. Por ello, es necesario, que antes de la construcción del software de control se realice la simulación del mismo para determinar posibles fallas de funcionamiento y corregirlas en tiempo de diseño.

3. Las propuestas revisadas incluyen estrategias de control que van desde las convencionales, como el uso de los controladores PID (controlador proporcional, controlador integral y controlador derivativo), hasta el uso de algoritmos inteligentes tales como lógica difusa, algoritmos genéticos y redes neuronales. Un análisis mayor de cada una de estas estrategias permitió seleccionar dos de ellas para realizar el diseño y simulación del sistema de control con la finalidad de hacer un comparativo de funcionamiento: la lógica difusa y las redes neuronales. De esta manera, el sistema de control debe estar basado en ellas.
4. Debido a que la cantidad de voltaje disponible en la barra guía es variable, el convertidor CD-CD debe regular este voltaje para entregarlo a los supercapacitores en forma constante, ya que, de no hacerlo así, éstos se dañarían y/o su vida útil se acortaría. También hay que considerar que el tiempo que transcurre entre el arribo, frenado, espera y aceleración, entre un tren y otro del metro, es muy corto. Por estas razones se requiere que el sistema de control sea automático, procese una gran cantidad de datos en un periodo de tiempo corto y logre una estabilidad del voltaje enviado a los supercapacitores, todo ello con el mínimo de error.

La pregunta de investigación que se plantea es: ¿es posible construir un sistema de control por software para el convertidor CD-CD, adecuado a las características específicas del mismo, usando una estrategia de control basada en algoritmos inteligentes?

1.4. Propuesta de solución

Para dar solución a la problemática planteada, se propone el desarrollo de un sistema de control, basado en algoritmos inteligentes, que regule el ciclo de control del convertidor CD-CD con la finalidad de obtener un voltaje de salida constante. Con base en ello, se plantean las siguientes hipótesis:

1. Debido a la automatización que implica, el control por software es más eficiente que un control manual o por hardware.
 2. Se requiere tener una estrategia de control o algoritmo adecuado que ejecute la estabilización del voltaje con rapidez y eficiencia. El uso de los llamados algoritmos de control inteligentes permitirá lograr esto.
-

-
3. El contar con la simulación de dos tipos de algoritmos, permitirá realizar pruebas de eficiencia en la ejecución a fin de decidir cuál de ellos debe ser implementado.
 4. El desarrollo del sistema de control debe seguir una metodología que aplique la ingeniería de software.

1.5. Objetivo

Desarrollar un control automático para un convertidor CD-CD que permita lograr que el voltaje de salida se estabilice en un valor de referencia, en un periodo de tiempo corto, a pesar de las variaciones del voltaje de entrada, utilizando para ello una metodología ingenieril de desarrollo de software.

1.6. Objetivos específicos

- Identificar el estado del arte de los sistemas de control para convertidores.
- Identificar el estado del arte de diferentes metodologías empleadas en la construcción de algoritmos de control inteligente.
- Desarrollar la simulación de dos prototipos del sistema de control, que sean implementadas usando dos metodologías diferentes y comparar los resultados de estas simulaciones, con el fin de determinar la más adecuada para su construcción

1.7. Metodología

La metodología que se seguirá para la realización de este proyecto contempla las siguientes actividades a desarrollar:

1. Planeación
 - 1.1. Elaboración del plan de configuración de los artefactos del proyecto
 - 1.2. Desarrollo del plan de proyecto del sistema de control
 2. Análisis de referentes teóricos
 - 2.1. Estudio y caracterización de algoritmos de control para convertidores.
 - 2.2. Obtención del estado del arte de sistemas de control para convertidores.
 - 2.3. Caracterización de diferentes metodologías para la implementación del algoritmo de control
 - 2.4. Obtención de los requerimientos necesarios para el desarrollo del algoritmo
 3. Definición de la estrategia de desarrollo del algoritmo
 - 3.1. Planteamiento de una propuesta general
 - 3.2. Diseño
 - 3.3. Especificación
 - 3.4. Verificación y validación
-

-
4. Simulación basada en lógica difusa
 - 4.1. Elaboración del diseño de la simulación
 - 4.2. Implementación de la simulación
 - 4.3. Validación de la simulación con respecto a los requerimientos
 5. Implementación del sistema de control basado en lógica difusa
 - 5.1. Análisis
 - 5.2. Diseño
 - 5.3. Elaboración del código del sistema
 - 5.4. Pruebas unitarias
 - 5.5. Validación de la aplicación con respecto a los requerimientos
 - 5.6. Pruebas de integración con dispositivos externo (convertidor CD-CD)
 - 5.7. Pruebas de integración en hardware
 - 5.8. Pruebas de ejecución
 6. Simulación basada en redes neuronales
 - 6.1. Elaboración del diseño de la simulación
 - 6.2. Implementación de la simulación
 - 6.3. Validación de la simulación con respecto a los requerimientos
 7. Selección de la mejor propuesta de solución
 - 7.1. Comparación de resultados de pruebas de ejecución
 - 7.2. Determinación de la mejor propuesta de solución, a partir de las implementaciones desarrolladas
 - 7.3. Elaboración de un artículo sobre los resultados obtenidos
 8. Entrega

1.8. Organización del documento

El contenido del documento se organiza de la siguiente manera: : en el capítulo 2 se presenta el estado del arte de los convertidores CD-CD, el caso particular del convertidor a controlar y una revisión general de las metodologías de control por medio de algoritmos inteligentes a ser usadas en el desarrollo del sistema de control; en el capítulo 3 se presenta el reporte del análisis, diseño, simulación e implementación del control para el convertidor CD-CD usando un algoritmo basado en lógica difusa; en el capítulo 4 se presenta el reporte del análisis, diseño y simulación del convertidor CD-CD usando un algoritmo basado en redes neuronales, en el capítulo 5 se presenta el comparativo de los dos algoritmos; finalmente se presentan las conclusiones y recomendaciones para trabajo futuro.

2. Convertidor CD-CD

En este capítulo se presenta un acercamiento a los convertidores CD-CD, además el reporte del análisis inicial para realizar el diseño de un caso particular de convertidor y, finalmente, se explican los sistemas de control para un convertidor CD-CD y muestra una visión general de dos tipos de metodologías de control, lógica difusa y redes neuronales. Todo ello como marco de referencia para comprender el dispositivo electrónico al que está encaminado el desarrollo del software de control que es la finalidad de este proyecto de tesis.

2.1 Convertidores CD-CD

En esta sección se presentan los conceptos básicos relacionados con los convertidores CD-CD, sus tipos de configuraciones y su funcionamiento.

2.1.1. Conceptos básicos

Los convertidores CD-CD son circuitos electrónicos de potencia que convierten una tensión continua en otro nivel de tensión continua y, normalmente, proporcionan una salida regulada (Hart, 2001).

Son dispositivos electrónicos ampliamente utilizados en sistemas de suministro de energía y tienen múltiples aplicaciones: fuentes de poder en computadoras, sistemas distribuidos de potencia, sistemas de potencia en vehículos eléctricos, etc. (Campos, 2003).

Las funciones de los convertidores CD-CD son (Rashid, 2010):

- Convertir un voltaje de entrada de CD en una tensión de salida de CD.
- Regular la tensión de salida de CD en función de las variaciones de carga y línea.
- Reducir la ondulación de la tensión de CA en la tensión de salida de CD por debajo del nivel requerido.

-
- Proporcionar aislamiento entre la fuente de entrada y la carga (aunque no siempre se requiere aislamiento).
 - Proteger el sistema suministrado y la fuente de entrada contra interferencias electromagnéticas.
 - Satisfacer las diversas normas internacionales y nacionales de seguridad.

Los convertidores CD-CD se pueden dividir en dos tipos principales: convertidores de conmutación dura a través de modulación de ancho de pulso (pulse-width modulation, PWM), y convertidores resonantes de conmutación suave. De éstos los del primer tipo han sido muy populares en las últimas décadas. Las ventajas de los convertidores PWM incluyen: alta eficiencia, operación de frecuencia constante, control simple y disponibilidad comercial de circuitos integrados controladores (Rashid, 2010).

2.1.2. Tipos de configuraciones

Existen diferentes tipos de configuraciones de los convertidores CD-CD (Mohan, Undelan, Robbins, 2009):

- Convertidor reductor (buck).
- Convertidor elevador (boost).
- Convertidor reductor/elevador (buck-boost).
- Convertidor Cúk.
- Convertidor de puente completo.

De estos cinco convertidores, sólo el reductor y el elevador son las topologías básicas del convertidor. Tanto los convertidores buck-boost, también llamado bidireccional, así como los de Cúk son combinaciones de las dos topologías básicas.

Los convertidores contienen cuatro elementos básicos: bobina, capacitor, diodo y un interruptor controlado.

Cada configuración a su vez contiene cuatro elementos básicos: bobina, capacitor, diodo y un interruptor controlado; así las propiedades de cada topología dependen de la ubicación de estos cuatro elementos (Figuras 2a, 2b y 2c) (Campos, 2003).

Donde:

V_s Voltaje de entrada

V_o Voltaje de salida

L Bobina

C Capacitor

Q interruptor controlado

R Carga resistiva

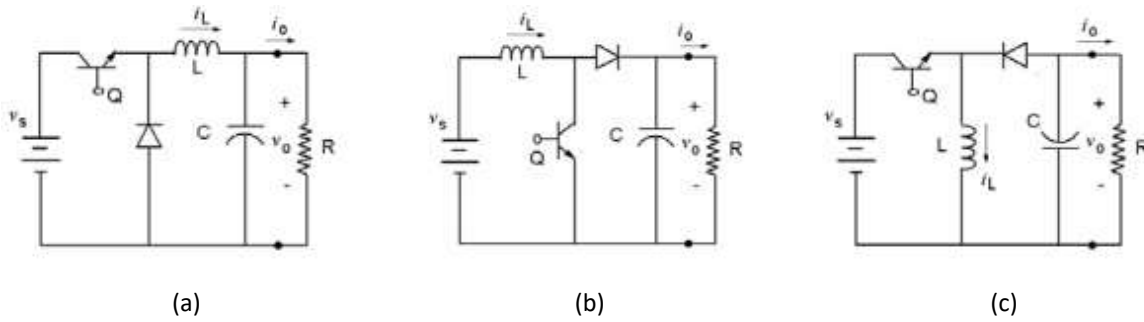


Figura 2. Configuraciones básicas de convertidores CD-CD. (a) Convertidor buck (reductor). (b) Convertidor boost (elevador). (c) Convertidor buck/boost (reductor/elevador).
Fuente: (Campos, 2003).

2.1.3. Funcionamiento

En los convertidores de CD-CD, el voltaje medio de salida de CD debe controlarse para que iguale un nivel deseado, aunque quizá fluctúen el voltaje de entrada y la carga de salida. En un convertidor CD-CD con un voltaje de entrada dado, el voltaje medio de salida se controla mediante el control de los tiempos de encendido y apagado (t_{enc} y t_{apag}). (Mohan, Undelan, Robbins, 2009).

Un método para controlar el voltaje de salida emplea la conmutación con una frecuencia constante (por ende, un periodo de conmutación constante $T_s = t_{enc} + t_{apag}$) y el ajuste de la duración de encendido del interruptor para controlar el voltaje medio de salida. En este método, llamado conmutación por modulación de ancho de pulsos (pulse-width modulation, PWM), el ciclo de trabajo D del interruptor controlado, que se define como la proporción de la duración de encendido con el periodo de conmutación, es variado.

En la conmutación de PWM con una frecuencia de conmutación constante, la señal de control del interruptor que controla el estado (encendido o apagado) del interruptor se genera por medio de la comparación de un voltaje de control en el nivel de señales $v_{control}$ con una forma de onda repetitiva, como se muestra en las figuras 3a y 3b. La señal del voltaje de control se obtiene por lo general mediante la amplificación del error, o la diferencia entre el voltaje de salida real y su valor deseado.

La frecuencia de la forma de onda repetitiva con un pico constante, el cual se muestra como diente de sierra, establece la frecuencia de conmutación. Esta frecuencia se mantiene constante en un control PWM. Cuando la señal de error amplificada, que varía muy lentamente con el tiempo en relación con la frecuencia de conmutación, es más grande que la forma de onda de diente de sierra, la señal de control del interruptor se vuelve alta, lo que causa que el interruptor se encienda. De lo contrario, el interruptor está apagado.

En términos de $v_{control}$ y el pico de la forma de onda de diente de sierra $\hat{V}_{st}$ en la figura 3, la relación de trabajo del interruptor se expresa como (1)

$$D = \frac{t_{enc}}{T_s} = \frac{v_{control}}{\hat{V}_{st}} \quad (1)$$

2.2 Caso particular de Convertidor CD-CD

En esta sección se muestra el análisis matemático realizado como punto de partida para el diseño y la construcción del convertidor CD-CD que es uno de los dispositivos electrónicos principales de los Sistemas de Ahorro de Energía para el metro de la Ciudad de México.

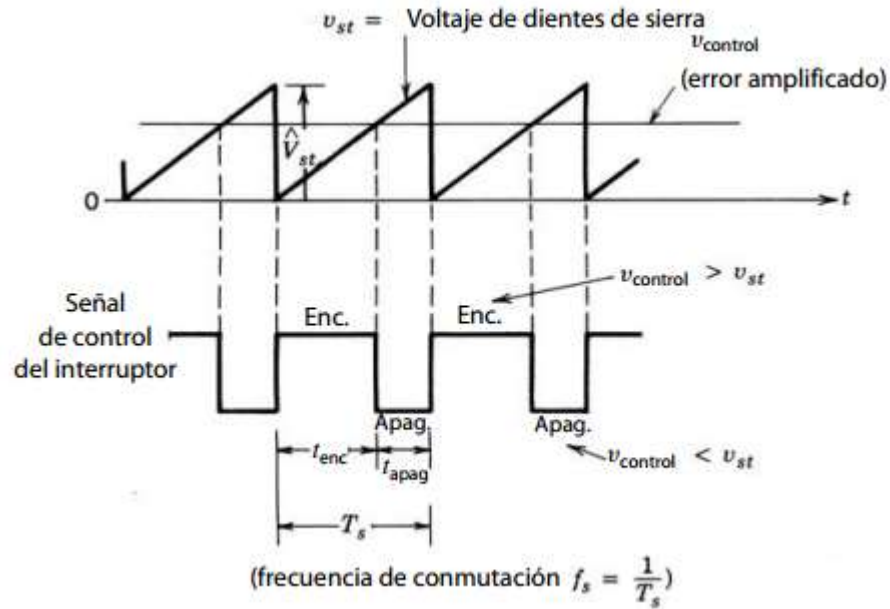
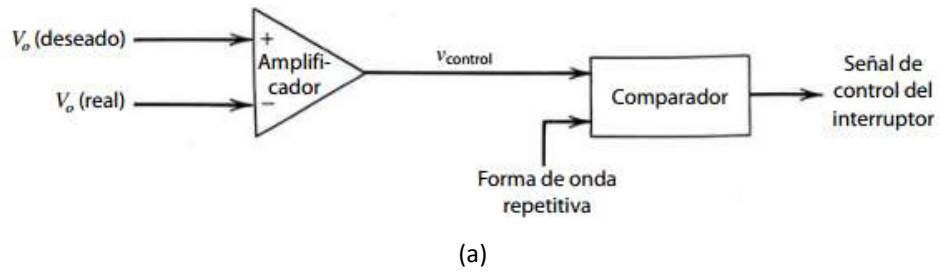


Figura 3. Modulador de anchura de impulsos. (a) diagrama de bloques.(b) señales del comparador
Fuente: (Mohan, Undelan, Robbins, 2009).

Se trata de un convertidor bidireccional, es decir, reductor/elevador o buck/boost, por lo que cual el análisis fue realizado para cada una de las partes componentes. Los resultados fueron reportados en (Cardeña-Moreno, 2012) y son reproducidos en las siguientes dos secciones, con la finalidad de comprender mejor el funcionamiento matemático del convertidor CD-CD.

2.2.1. Convertidor buck

En un convertidor CD-CD reductor también llamado regulador buck el voltaje de promedio de salida es menor al voltaje de entrada. El funcionamiento se divide en dos modos de acuerdo al estado del switch, cerrado o abierto.

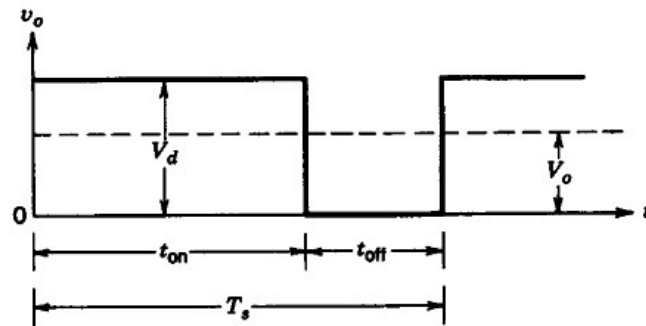


Figura 4. Ciclo de trabajo representado en periodo de tiempo.
Fuente: (Cardeña-Moreno, 2012)

La conmutación entre ambos modos ocurre en un lapso de tiempo en donde el modo 1 (t_{ON}) sucede en $t=0$ y el modo 2 (t_{OFF}) en $t=t_1$ representado en la figura 4. El tiempo total de duración entre los dos modos es el periodo (T) y se entiende que t_1 puede ocurrir en cualquier rango de T ($0 \leq t_1 \leq T$).

$$T = t_{ON} + t_{OFF} \quad (2)$$

Con lo anterior se puede definir el ciclo de trabajo (D) como la relación entre el tiempo de duración del modo 1 o el tiempo de encendido y el periodo total:

$$D = \frac{t_{ON}}{T} \quad (3)$$

Los valores de D varían entre 0 y 1 ($0 \leq D \leq 1$). Luego:

$$t_{ON} = DT \quad (4)$$

Tomando las ecuaciones (2) y (4):

$$t_{OFF} = T(1 - D) \quad (5)$$

Se sabe que el voltaje a través del inductor es:

$$V_L = L \frac{di}{dt} \quad (6)$$

Se supone que la corriente en el inductor tiene un comportamiento lineal, por lo que ésta aumenta en el modo 1 y se expresa a partir de (6):

$$V_i - V_o = L \frac{\Delta I}{t_{on}} \quad (7)$$

En el modo 2, la corriente baja en el inductor,

$$-V_o = -L \frac{\Delta I}{t_{OFF}} \quad (8)$$

La corriente de rizo pico a pico en el inductor se expresa con ΔI , luego se igualan las ecuaciones (7) y (8):

$$\Delta I = \frac{(V_i - V_o)t_{ON}}{L} = \frac{V_o t_{OFF}}{L} \quad (9)$$

Para obtener el voltaje promedio de salida se sustituyen las ecuaciones (4) y (5) en (9):

$$V_o = DV_i \quad (10)$$

Se supone que no hay pérdidas de corriente en el circuito:

$$V_i I_i = V_o I_o \quad (11)$$

Sustituyendo la ecuación (10) en (11):

$$V_i I_i = DV_i I_o \quad (12)$$

Se obtiene la corriente promedio de entrada:

$$I_i = DI_o \quad (13)$$

La corriente de rizo pico a pico se puede calcular a partir de la ecuación (2) sustituyendo las expresiones de t_{on} y t_{OFF} de las ecuaciones (7) y (8):

$$T = \frac{L \Delta I}{V_i - V_o} + \frac{L \Delta I}{V_o} \quad (14)$$

Siendo que $T = 1/f$ y despejando ΔI :

$$\Delta I = \frac{V_o(V_i - V_o)}{V_i f L} \quad (15)$$

Utilizando la ecuación (10) se puede presentar en términos de ciclo de trabajo:

$$\Delta I = \frac{V_i D (1 - D)}{L f} \quad (16)$$

A partir de la configuración del convertidor reductor (Figura 2a) y de la ley de corriente de Kirchhoff, la corriente en el inductor se expresa como $i_L = i_c + i_o$, donde i_c e i_o son las corrientes del capacitor y de la carga R respectivamente. Si se considera que el rizo de corriente de la carga es despreciable entonces $2I_L = \Delta I_L = \Delta I_C = 2I_C$.

Con lo anterior la corriente promedio en el capacitor (en $T/2$) es $I_c = \frac{\Delta I}{4}$. A partir de la ecuación de corriente en el capacitor $i_c = C \frac{dV_c}{dt}$, el rizo de voltaje se define por:

$$\Delta V_c = \frac{1}{C} \int_0^{T/2} \frac{\Delta I}{4} dt = \frac{\Delta I T}{8C} = \frac{\Delta I}{8fC} \quad (17)$$

Y sustituyendo las ecuaciones (15) y (16) en (17) se obtienen:

$$\Delta V_c = \frac{V_o(V_i - V_o)}{8LCf^2V_i} \quad (18)$$

$$\Delta V_c = \frac{DV_i(1 - D)}{8LCf^2} \quad (19)$$

Se requiere que el convertidor funcione en modo continuo; es decir, que la corriente en el inductor sea en todo momento mayor a cero. Para el diseño es necesario obtener valores críticos de inductancia L_c y capacitancia C_c para cumplir con la condición.

Siendo I_L la corriente promedio que pasa por el inductor, la corriente de rizo pico a pico se define como $\Delta I = 2I_L$. Utilizando las ecuaciones (10) y (16):

$$\Delta I = 2I_L = \frac{V_i D (1 - D)}{L f} = \frac{2V_o}{R} = 2I_o = \frac{2DV_i}{R} \quad (20)$$

Luego,

$$\frac{V_i D (1 - D)}{L f} = \frac{2DV_i}{R} \quad (21)$$

Despejando L se obtiene el valor crítico para la inductancia:

$$L_c = \frac{R(1 - D)}{2f} \quad (22)$$

Similarmente, para el valor crítico del capacitor se sabe que V_c es su voltaje promedio; entonces $\Delta V_c = 2V_c = 2V_o$.

$$\Delta V_c = 2V_c = \frac{DV_i(1 - D)}{8LCf^2} = 2V_o = 2DV_i \quad (23)$$

Luego,

$$\frac{DV_i(1 - D)}{8LCf^2} = 2DV_i \quad (24)$$

Despejando C se obtiene el valor crítico para el capacitor:

$$C_c = \frac{1 - D}{16f^2L} \quad (25)$$

2.2.2. Convertidor boost

El convertidor CD-CD elevador o regulador boost tiene una salida de voltaje mayor al voltaje de entrada. El funcionamiento de este convertidor se divide en dos modos dependiendo del estado del switch, cerrado o abierto.

El voltaje a través del inductor es:

$$V_L = L \frac{di}{dt} \quad (26)$$

Considerando que la corriente a través del inductor tiene un comportamiento lineal, aumentando durante el modo 1, el voltaje en el inductor se expresa:

$$V_i = L \frac{\Delta I}{t_{ON}} \quad (27)$$

Durante el modo 2, la corriente baja en el inductor y se expresa:

$$V_i - V_o = -L \frac{\Delta I}{t_{OFF}} \quad (28)$$

Donde ΔI es la corriente rizo pico a pico en el inductor. Igualando las ecuaciones (27) y (28) se obtiene:

$$\Delta I = \frac{V_i t_{ON}}{L} = \frac{(V_o - V_i) t_{OFF}}{L} \quad (29)$$

Recordando las ecuaciones (4) y (5) y sustituyéndolas en la ecuación (29):

$$V_o = \frac{V_i}{(1 - D)} \quad (30)$$

Se asume que no hay pérdidas en el circuito:

$$V_i I_i = V_o I_o \quad (31)$$

Sustituyendo la ecuación (30) en (31):

$$V_i I_i = I_o \frac{V_i}{(1 - D)} \quad (32)$$

Se obtiene la corriente promedio de entrada

$$I_i = \frac{I_o}{(1 - D)} \quad (33)$$

Si $T = t_{ON} + t_{OFF}$ y sustituyendo con las ecuaciones (27) y (28) se obtiene:

$$T = \frac{\Delta I L}{V_i} + \frac{\Delta I L}{(V_o - V_i)} \quad (34)$$

Si $T = 1/f$ y despejando ΔI de la (34):

$$\Delta I = \frac{V_i(V_o - V_i)}{V_o f L} \quad (35)$$

Utilizando la ecuación (30) se puede presentar en términos de ciclo de trabajo:

$$\Delta I = \frac{V_i D}{L f} \quad (36)$$

El voltaje rizo del capacitor se expresa de la siguiente manera:

$$\Delta V_c = \frac{I_o D}{f C} \quad (37)$$

Al igual que en el convertidor buck, se requiere que el convertidor boost opere de manera continua, por lo que es necesario obtener ecuaciones de valores críticos para capacitancia e inductancia.

Si I_L es la corriente promedio que pasa por el inductor, la corriente de rizo pico a pico es $\Delta I = 2I_L$. Utilizando las ecuaciones (36) y (30):

$$\Delta I = 2I_L = \frac{V_i D}{L f} = \frac{2V_o}{R} = 2I_o = \frac{2V_i}{(1-D)R} \quad (38)$$

Luego,

$$\frac{V_i D}{L f} = \frac{2V_i}{(1-D)R} \quad (39)$$

Se despeja L para obtener el valor crítico para la inductancia:

$$L_c = \frac{R(1-D)D}{2f} \quad (40)$$

El voltaje medio del capacitor es:

$$\Delta V_c = 2V_c = \frac{I_o D}{f C} = 2V_o = 2I_o R \quad (41)$$

Despejando C se obtiene el valor crítico para el capacitor:

$$C_c = \frac{D}{2fR} \quad (42)$$

2.3 Sistemas de control para un convertidor CD-CD

Un sistema de control es un sistema que dirige o regula el comportamiento de otro sistema o proceso, formado por un conjunto de componentes capaz de gobernar una salida a partir de unos elementos de entrada (Juárez-Herrero, Cañadas-Martínez, Marín-Morales, R. 2008).

En cualquier tipo de sistema de control se pueden encontrar los siguientes elementos fundamentales:

- La planta. Es el proceso o sistema sobre el que se pretende actuar para establecer el control.
- El controlador. Gestiona el valor de la salida de control a partir de la entrada de referencia.
- La entrada de referencia (o entrada de mando). Es la señal externa aplicada a un controlador como un parámetro de ajuste.
- La señal de control. Es la variable, manipulada por el sistema de control, aplicada a la planta.
- La salida controlada. Es la variable de salida de la planta que está siendo controlada.

Los sistemas de control se clasifican habitualmente según su estructura. Así, se establecen dos grandes categorías: sistemas de control de lazo abierto y sistemas de control de lazo cerrado.

Los sistemas de control más sencillos que existen son los denominados controladores de lazo abierto. En ellos, la acción del controlador es independiente de la salida controlada de la planta. Así, los sistemas de control de lazo abierto están compuestos únicamente por el controlador, la planta, la entrada de referencia al controlador, la señal de control de la planta y la salida controlada como se muestra en la Figura 5.



Figura 5. Circuito de control de lazo abierto.

Fuente: (Juárez-Herrero, Cañadas-Martínez, Marín-Morales, R. 2008).

Los sistemas de control de lazo abierto se comportan de forma adecuada en entornos relativamente estables. Sin embargo, cuando existen perturbaciones inesperadas, un sistema de control de lazo abierto es, a menudo, bastante ineficaz. Por lo tanto, se aplica la idea de la retroalimentación para ofrecer mayor capacidad de actuación a los controladores.

Los controladores que utilizan las propiedades de la retroalimentación de sistemas se denominan controladores de lazo cerrado. La retroalimentación en estos sistemas es la característica que permite que la salida controlada se compare con la entrada de

referencia, de forma que el controlador actúe de forma apropiada considerando la diferencia producida entre ambos (error). Es decir, el control en los sistemas de lazo cerrado depende tanto de la entrada como del valor de salida, siendo su objetivo la minimización del error.

Los elementos fundamentales de un controlador de lazo cerrado son la entrada de referencia, la señal de error, el comparador, el controlador, la señal de control, la planta, la salida controlada, el retroalimentador y la señal de retroalimentación (Figura 6).

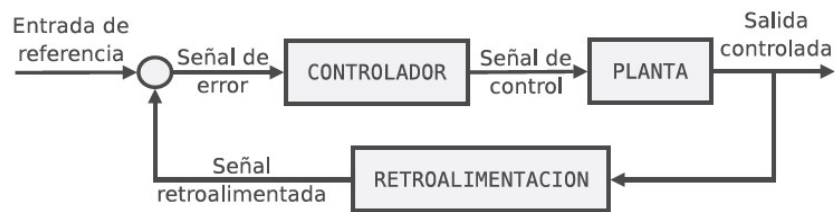


Figura 6. Circuito de control de lazo cerrado.

Fuente: (Juárez-Herrero, Cañadas-Martínez, Marín-Morales, R. 2008).

Los sistemas de lazo cerrado, en comparación con los de lazo abierto son, a menudo, más exactos y menos sensibles a las perturbaciones del exterior. Sin embargo, hay que tener en cuenta que estos sistemas pueden producir cierta inestabilidad en la planta debido a su continuo reajuste.

Los métodos o estrategias de control para convertidores CD-CD más utilizados son los controladores PID. La característica principal de un controlador PID es la forma en la que se calcula la señal de control, estableciendo tres medidas para procesar la señal de salida controlada: anular el error del instante actual, medir la cantidad de tiempo que la señal se ha mantenido con valores incorrectos e intentar anticipar errores futuros midiendo la tasa de variación de la señal en cada instante. A estos controladores se les denomina PID debido al tipo de cálculo que se realiza para la corrección de errores: Producto, Integración y Derivación.

En la actualidad, la mayoría de controladores simples PID que se comercializan, se desarrollan sobre microprocesadores (controladores digitales). Un controlador digital funciona de forma discreta, muestreando la señal analógica de entrada mediante dispositivos sensores. En dichos controladores, los algoritmos de control se implementan directamente en el lenguaje del microprocesador, distribuidos en bloques de cálculo y almacenados en la memoria del dispositivo controlador.

A pesar del gran éxito de los controladores PID, también tienen sus límites. En primer lugar, están diseñados para resolver problemas de control lineal; además, existen situaciones, como la denominada saturación de acción integral, en la que la combinación de la acción integral del controlador junto con la aportación proporcional podría llevar a una situación fuera de control que debe ser subsanada mediante intervención manual.

La estructura de un sistema de control convencional es insuficiente para el control de procesos complejos, debido a los factores tales como: sistemas mal definidos, la disponibilidad de la representación analítica, la existencia de incertidumbre e imprecisión, la necesidad de gestionar conceptos no numéricos, y la necesidad de aumentar la seguridad de funcionamiento, por exigencia de mayor confiabilidad, o por exigencia de funcionamiento continuo.

El control inteligente surge como soporte para la sistematización de las tareas complejas con el fin de suplir sus posibles carencias producidas, entre otros motivos, por la sobrecarga de información. Se trata de automatizar, en la medida en que sea posible, tareas como el análisis de los datos, la detección de fallos, el diagnóstico de los mismos y la toma de decisiones o propuesta de acciones concretas.

Tradicionalmente, los sistemas de control inteligente combinan la teoría clásica de control con técnicas de inteligencia artificial para obtener un sistema basado en un modelo físico, o matemático, dirigido por cierto conocimiento adicional. El estudio de estos sistemas se ha apoyado en campos tales como las redes neuronales, los algoritmos genéticos, los sistemas multiagente o la lógica difusa.

En las secciones siguiente se presenta un panorama general de las dos metodologías seleccionadas para el desarrollo del software de control para un convertidor CD-CD: la lógica difusa y las redes neuronales.

2.4. Lógica difusa

Desde su aparición en la década de los 60's hasta nuestros días, las aplicaciones de la lógica difusa se han ido consolidando, paulatinamente al comienzo, y con un desbordado crecimiento en los últimos años. Se encuentran en soluciones a problemas de control industrial, en predicción de series de tiempo, como metodologías de archivo y búsqueda de bases de datos, en investigación operacional, en estrategias de mantenimiento predictivo y en otros campos más (Duarte-Velazco, 1999).

Las principales razones para tal proliferación de aplicaciones quizás sean la sencillez conceptual de los Sistemas basados en Lógica Difusa, su facilidad para adaptarse a casos particulares con pocas variaciones de parámetros, su habilidad para combinar en forma unificada expresiones lingüísticas con datos numéricos, y el no requerir de algoritmos muy sofisticados para su implementación.

La lógica difusa es sinónimo de la teoría de conjuntos difusos, una teoría que relaciona clases de objetos con fronteras poco claras en las que miembros son sujetos de ser asignados a un cierto grado. Puede ser vista como una metodología para computación con palabras más que con números, por ello es considerada como la codificación del sentido común (Guide, 2012).

Algunas de las razones para usar lógica difusa son las siguientes:

-
- Es conceptualmente fácil de entender.
 - Es flexible.
 - Es tolerante con los datos imprecisos.
 - Puede modelar funciones no lineales de complejidad arbitraria.
 - Puede construirse sobre la experiencia de los expertos.
 - Puede mezclarse con técnicas de control convencionales.
 - Se basa en el lenguaje natural.

2.5. Redes neuronales

Una neurona artificial es una pieza de software capaz de simular las cuatro tareas principales que una neurona biológica ejecuta: i) Dendritas: Aceptar entradas, ii) Soma: Procesar las entradas, iii) Axón: Convertir las entradas procesadas en salidas, iv) Sinápsis: El contacto electromecánico entre neuronas.

Las redes neuronales artificiales pueden ser definidas como modelos computacionales cuya operación está basada en procesamiento paralelo (Kessal, Rahmani, Mostefai, Gaubert, 2012).

En una red neuronal, múltiples neuronas están interconectadas para formar una red que facilite el cómputo distribuido. A la configuración de las interconexiones se le denomina topología y se denota por un grafo. Las conexiones se realizan por medio de un peso de ajuste o peso sináptico. En la Figura 7, la notación w_{12} indica el peso sináptico entre las neuronas 1 y 2. Los subíndices se refieren a los números de las neuronas conectadas.

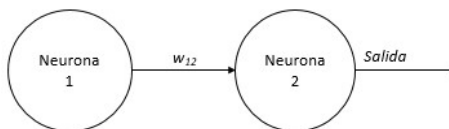


Figura 7. Conexiones entre neuronas
Fuente: Elaboración propia

Algunas características de las redes neuronales son (Matich, 2001):

- Consisten de unidades de procesamiento que intercambian datos o información.
- Se utilizan para reconocer patrones, incluyendo imágenes, manuscritos y secuencias de tiempo (por ejemplo: tendencias financieras).
- Tienen capacidad de aprender y mejorar su funcionamiento.

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos

anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Entre las ventajas se incluyen:

- Aprendizaje Adaptativo. Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- Auto-organización. Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- Tolerancia a fallos. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- Operación en tiempo real. Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- Fácil inserción dentro de la tecnología existente. Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

Una red neuronal es una forma de permitir a un sistema aprender con base en una experiencia previa, por lo que es necesario que, una vez definida la topología de la red, se seleccione un algoritmo de entrenamiento que permita a la red “aprender”.

El proceso de aprendizaje también conocido como entrenamiento de la red puede ser supervisado o no supervisado (Salas, 2004).

El aprendizaje supervisado consiste en entrenar la red a partir de un conjunto de datos o patrones de entrenamiento compuesto por patrones de entrada y salida. El objetivo del algoritmo de aprendizaje es ajustar los pesos de la red w_{ij} de manera tal que la salida generada por la red neuronal sea lo más cercanamente posible a la verdadera salida dada una cierta entrada. Es decir, la red neuronal trata de encontrar un modelo al proceso desconocido que generó la salida. Este aprendizaje se llama supervisado pues se conoce el patrón de salida el cual hace el papel de supervisor de la red.

En el aprendizaje no supervisado se presenta sólo un conjunto de patrones a la red neuronal, y el objetivo del algoritmo de aprendizaje es ajustar los pesos de la red de manera tal que la red encuentre alguna estructura o configuración presente en los datos.

3. Software de control por lógica difusa

Para realizar el control del convertidor CD-CD se requieren métodos de control inteligente que aseguren una salida óptima a través del manejo y transformación de las entradas. Para este trabajo, se seleccionó en primer lugar el uso de lógica difusa debido a que proporciona la facilidad de resolver situaciones complejas y dinámicas, que son más fácilmente expresadas con palabras que con modelos matemáticos.

El control se realizó siguiendo una metodología de desarrollo de software que implica las etapas de análisis, diseño, simulación, e implementación. Las siguientes secciones muestran el reporte de cada uno de esas etapas.

3.1. Análisis

Como parte del análisis, se incluyen las especificaciones del software de control, así como las especificaciones técnicas del convertidor.

3.1.1. Especificaciones

Se tiene un convertidor CD-CD el cual soporta dos configuraciones, buck y boost. El convertidor se coloca como parte de un circuito de control de lazo cerrado (Figura 8). La estrategia de control consiste en comparar un voltaje de referencia V_{ref} con la medición del voltaje de salida del convertidor V_o , a la diferencia resultante se le conoce como el error E y es procesado por un software de control. La salida del software de control incide en el ciclo de trabajo D del interruptor controlado del convertidor para modificar el voltaje de salida de éste.

El objetivo del software de control es lograr que la salida del convertidor se estabilice en el valor de referencia en un periodo de tiempo corto, haciendo que el error se convierta en 0 y sin importar las variaciones del voltaje de entrada V_i .

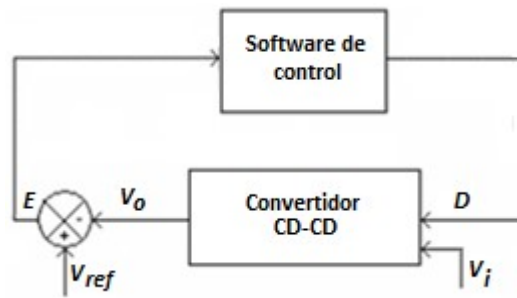


Figura 8. Arquitectura del circuito de control de lazo cerrado para el convertidor cd-cd
Fuente: Elaboración propia

Para el convertidor funcionando en modo buck, el voltaje de salida se obtiene a través de la ecuación (43), mientras que para el convertidor funcionando en modo boost, el voltaje de salida se obtiene a través de la ecuación (44).

$$V_o = V_i \cdot D \quad (43)$$

$$V_o = V_i / (1-D) \quad (44)$$

Donde:

V_i Es el valor de alimentación del convertidor

D Es el valor del ciclo de trabajo del interruptor controlado

V_o Es el valor de salida del convertidor

Para el software de control, la entrada será el error y la salida el valor trabajo D ajustado a las condiciones actuales, el cual puede tomar valores en un intervalo $[0,1]$.

Se requiere desarrollar un software de control automático para cada una de las configuraciones y deberán implementarse usando lógica difusa.

3.1.2. Especificaciones técnicas del convertidor

Adicionalmente, se especifican las características del convertidor para cada una de las configuraciones. La Tabla 1 muestra las características técnicas del convertidor funcionando en modo buck y la Tabla 2 las características técnicas del convertidor funcionando en modo boost.

Tabla 1. Características del convertidor en modo buck
Fuente: Elaboración propia

Característica	Valores
Intervalo de valores de voltaje de alimentación V_i	[600 - 1000 V]
Voltaje de referencia V_{ref}	375 V
Intervalo de valores del ciclo de trabajo D	[0 - 1]

Tabla 2. Características del convertidor en modo boost
Fuente: Elaboración propia

Característica	Valores
Intervalo de valores de voltaje de alimentación V_i	[187.5 - 375 V]
Voltaje de referencia V_{ref}	750 V
Intervalo de valores del ciclo de trabajo D	[0 - 1]

3.2. Diseño

Un software de control difuso contiene un algoritmo que usa una estrategia de control lingüística en lugar de una de control automático.

El software de control por lógica difusa, a diferencia de los controles convencionales no requiere un modelo matemático del sistema que debe ser controlado (Kessal, Rahmani, Mostefai, Gaubert, 2012), sin embargo, es necesaria una comprensión adecuada del sistema y de los requisitos de control. En el diseño del controlador difuso se debe clarificar cómo es procesada la información (estrategia de control y decisión), y cómo fluye a través del sistema (solución / variable de salida) (Huh, S. H., & Park, G. T. 1999).

3.2.1. Diseño general del módulo de control difuso

La Figura 9 es una modificación de la arquitectura del circuito de control mostrada en la figura 8, en el cual se resalta el diseño típico del módulo de control basado en lógica difusa (delimitado por la línea punteada), que está formado por cuatro elementos (So, Tse y Lee, 1996):

- el fuzzificador, que convierte los datos de entrada en valores lingüísticos,
- una base de conocimiento, que consiste de una base de datos con sus correspondientes definiciones lingüísticas y un conjunto de reglas de control,
- un mecanismo de inferencia, que simula un proceso de decisión humano, a través de inferir acciones de control por medio de las reglas de la base de conocimiento, y
- un defuzzificador, que convierte los datos inferidos en datos reales

3.2.1.1. Metodología para el diseño

Los pasos a seguir en el diseño del módulo de control por medio de lógica difusa son (Rameshkumar, Arumugam, 2009):

1. Seleccionar parámetros y elementos de control como factores de escala para las señales de entrada y salida.
2. Particionar el universo de discurso del intervalo abarcado por cada variable en un número de subconjuntos difusos, asignando una etiqueta lingüística cada una.
3. Asignar una función de membresía para cada subconjunto difuso.

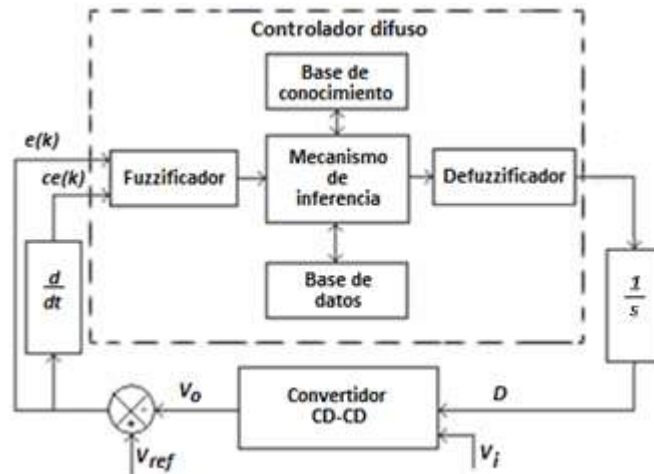


Figura 9. Circuito de control basado en lógica difusa
Fuente: Elaboración propia

4. Asignar relacionales difusas entre los subconjuntos de entrada y salida, formando la base de conocimiento.
 5. Elegir factores de escala apropiados para las variables de entrada y de salida con el fin de normalizar los valores en intervalos de $[-1, 1]$ o $[0, 1]$.
 6. Fuzzificar las entradas, es decir, clasificar los datos de entrada en valores o conjuntos lingüísticos adecuados para el controlador.
 7. Usar un razonamiento difuso apropiado para inferir la salida contribuida por cada regla (relación).
 8. Agregar las salidas difusas recomendadas por cada regla.
 9. Aplicar la técnica de defuzzificación para formar una salida nítida.
 10. Enviar la acción de cambio control hacia la planta, en este caso, el convertidor CD-CD.
- Las siguientes secciones reportan la manera en que se aplicó esta metodología para el diseño del módulo de control difuso.

3.2.1.2. Entradas y salidas

El módulo de control difuso tiene dos entradas controladas:

- El error $e(k)$. Se refiere a la diferencia existente entre la salida del convertidor y el valor de referencia.
- El cambio en el error $ce(k)$. Se refiere a la variación existente entre el error actual $e(k)$ y el error del ciclo anterior $e(k-1)$ y se considera la derivada del error.

La obtención de los valores de entrada se realiza por medio de (45) y (46):

$$e(k) = V_{ref} - V_o \quad (45)$$

$$ce(k) = e(k) - e(k-1) \quad (46)$$

Donde:

V_{ref} Es el valor de referencia

V_0 Es el valor de salida del convertidor

La salida única del módulo de control difuso es un factor de ajuste ΔD que se aplica al ciclo de trabajo D del generador de PWM (interruptor controlado del convertidor). El factor de ajuste puede ser positivo o negativo y es integrado al valor actual de D .

3.2.1.3. Definición de variables lingüísticas

Para el módulo de control difuso se definieron nueve variables lingüísticas, cuyo nombre y significado se muestran en la tabla 3, las cuales representan conjuntos difusos para los valores del error y cambio de error.

Tabla 3. Variables lingüísticas

Fuente: Elaboración propia

Variable lingüística	Significado
NE	Negativo extra grande
NG	Negativo grande
NM	Negativo mediano
NP	Negativo pequeño
CE	Cero
PP	Positivo pequeño
PM	Positivo mediano
PG	Positivo grande
PE	Positivo extra grande

3.2.1.3. Definición de funciones de membresía

Las variables mostradas en la tabla 3 deben ser expresadas en un formato difuso o fuzzificadas, ésto se logra definiendo la membresía como una función, es decir, definir funciones de membresía para cada uno de los conjuntos difusos que van a estar representados por las variables lingüísticas ya definidas.

Las funciones de membresía, están basadas la distribución normal dada por error en el voltaje de salida y el cambio en el error. En este caso se utilizó la función de membresía triangular, definida por medio de (47). Donde a es el valor central de la función de membresía (es decir, que a toma el valor máximo), $a-b$ es el valor inicial y $a+c$ es el valor final de la función. El rango de la función va de $a-b$ a $a+c$.

$$\mu(z) = \Lambda(z) = \begin{cases} 1 - (a - z)/b & a - b \leq z < a \\ 1 - (z - a)/c & a \leq z < a + c, \\ 0 & \text{otro caso} \end{cases} \quad (47)$$

Los intervalos de cada función de membresía están dados por la variable lingüística representativa (de las definidas en la tabla 3), los valores de las variables van

incrementando, es decir, la primera función de membresía inicia con la primera variable lingüística (NG) el centro a es el valor $a - c$ de la siguiente función y así sucesivamente.

3.2.1.4. Definición de reglas de inferencia

Para el diseño de las reglas de inferencia del módulo de control difuso se utilizó el método de observación de la distancia con signo (Salam, Taeed, Ayob, 2011). Este método utiliza una tabla de reglas para definir el comportamiento del módulo de control para obtener una salida adecuada. El método se puede resumir de la siguiente manera:

1. Cuando la salida del convertidor está lejos del punto de referencia, el cambio de ciclo de trabajo debe ser amplio con el fin de llevar la salida al punto de ajuste rápidamente.
2. Cuando la salida del convertidor se aproxima al punto de referencia, se considera un pequeño cambio de ciclo de trabajo.
3. Cuando la salida del convertidor se encuentra cerca del punto de referencia y se acerca rápidamente, el ciclo de trabajo debe mantenerse constante a fin de evitar rebasarlo.
4. Cuando se alcanza el punto de referencia y la salida todavía está cambiando, el ciclo de trabajo debe ser cambiado un poco para prevenir un alejamiento.
5. Cuando se alcanza el punto de referencia y la salida es constante, el ciclo de trabajo se mantiene sin cambios.
6. Cuando la salida está por encima del punto de referencia, el signo del ajuste de ciclo de trabajo debe ser negativo, y viceversa.

A partir de este método, se puede construir una tabla de reglas de dos dimensiones (Tabla 4). Los encabezados de las filas y las columnas de la tabla son las variables lingüísticas ya definidas, que muestran las combinaciones posibles de mediciones para el error y el cambio el error. Tal tabla de reglas se conoce como con la estructura de Toeplitz y define el comportamiento esperado para el módulo de control difuso, ya que los cruces representan las decisiones de salida, de acuerdo a las reglas ya mencionadas.

Tabla 4. Tabla de reglas con la estructura Toeplitz
Fuente: Elaboración propia

		Cambio en el error								
		NE	NG	NM	NP	CE	PP	PM	PG	PE
Error	NE	CE	PP	PM	PG	PE	PE	PE	PE	PE
	PG	NP	CE	PP	PM	PG	PE	PE	PE	PE
	PM	NM	NP	CE	PP	PM	PG	PE	PE	PE
	PP	NG	NM	CE	PP	PM	PG	PE	PE	PE
	CE	NE	NG	NM	NP	CE	PP	PM	PG	PE
	NP	NE	NE	NG	NM	NP	CE	PP	PM	PG
	NM	NE	NE	NE	NG	NM	NP	CE	PP	PM
	NG	NE	NE	NE	NE	NG	NM	NP	CE	PP
	NE	NE	NE	NE	NE	NE	NG	NM	NP	CE

Las decisiones de diseño tanto de variables lingüísticas, funciones de membresía y reglas de inferencia mencionadas son válidas para los módulos de control de cada configuración del convertidor, es decir, tanto para el convertidor buck como para el convertidor boost.

En las siguientes secciones se muestran las características de diseño particulares de cada uno de ellos.

3.2.2. Diseño del módulo difuso para el convertidor buck

A partir de los valores especificados en la Tabla 1, se realizaron las operaciones matemáticas para definir los intervalos de posibles valores que pudieran tomar las variables de entrada, error y cambio en el error, y la variable de salida, cambio en el ciclo de trabajo, del módulo de control difuso para el convertidor buck, Estos intervalos se muestran en la tabla 5.

Tabla 5. Intervalos de valores para las variables de entrada y salida del controlador difuso para el funcionamiento del convertidor en modo buck

Fuente: Elaboración propia

Tipo	Característica	Valores
Entrada	Intervalo de valores del error	[-2250, 563]
Entrada	Intervalo de valores del cambio en el error	[-2815, 2815]
Salida	Intervalo de valores del cambio del ciclo de trabajo D	[-0.625, 0.625]

Con estos datos se definieron los valores de los centros a para las funciones de membresía de las variables tanto de entrada como de salida, los cuales se muestran en la tabla 6.

Tabla 6. Valores definidos para las funciones de membresía de las variables de entrada y la de salida del control difuso para el convertidor en modo buck

Fuente: Elaboración propia

Entradas		Salida	
Error e	Cambio en el error ce	Cambio en el ciclo de trabajo ΔD	
PE 563	PE 2815	PE	0.625
PG 450	PG 1600	PG	0.3
PM 250	PM 800	PM	0.075
PP 100	PP 200	PP	0.0025
CE 0	CE 0	CD	0.0
NP -100	NP -200	NP	-0.0025
NM -500	NM -800	NM	-0.075
NG -1800	NG -1600	NG	-0.3
NE -2250	NE -2815	NE	-0.625

3.2.3. Diseño del módulo difuso para el convertidor boost

Aplicando el mismo método que para el módulo de control difuso del convertidor buck, se definieron los intervalos de posibles valores de las variables de entrada, error y cambio en el error, y la variable de salida, cambio en el ciclo de trabajo, del módulo de control difuso para el convertidor boost. En la tabla 7 se muestran estos intervalos.

Tabla 7. Intervalos de valores para las variables de entrada y salida del controlador difuso para el funcionamiento del convertidor en modo boost

Fuente: Elaboración propia

Tipo	Característica	Valores
Entrada	Intervalo de valores del error	$[-625, 375]$
Entrada	Intervalo de valores del cambio en el error	$[-1000, 1000]$
Salida	Intervalo de valores del cambio del ciclo de trabajo D	$[-0.625, 0.625]$.

A partir de estos datos, se definieron los centros a para las funciones de membresía de las variables tanto de entrada como de salida, así como sus valores, los cuales se muestran en la Tabla 8.

Tabla 8. Valores definidos para las funciones de membresía de las variables de entrada y la de salida del control difuso para el convertidor en modo boost

Fuente: Elaboración propia

Entradas		Salida	
Error e	Cambio en el error ce	Cambio en el ciclo de trabajo ΔD	
PE 375	PE 1000	PE	0.625
PG 250	PG 500	PG	0.3
PM 125	PM 125	PM	0.075
PP 25	PP 50	PP	0.0025
CE 0	CE 0	CD	0.0
NP -25	NP -50	NP	-0.0025
NM -218	NM -125	NM	-0.075
NG -417	NG -500	NG	-0.3
NE -625	NE -1000	NE	-0.625

3.3. Simulación

Debido a las características técnicas del convertidor y con la finalidad de evaluar que los módulos de control diseñados funcionen de manera adecuada, se realizó la simulación de los circuitos de control para cada una de las configuraciones del convertidor.

Las simulaciones se realizaron en Matlab®, usando los módulos Simulink y Fuzzy Logic Toolbox™.

Las siguientes secciones describen cómo se realizó esta simulación y los resultados de su ejecución.

3.3.1. Simulación del circuito de control para el convertidor buck

Como punto inicial fue necesario simular el módulo de control difuso, para ello se usó el Fuzzy Logic Toolbox™ de Matlab®. Los pasos detallados para construir este módulo se describen en el Apéndice A.

La Figura 10 muestra el modelo de la simulación del circuito de control. La simulación se realizó en Simulink® y funciona de la siguiente manera:

1. El emulador de señal permite generar una señal que puede ser configurada para realizar variaciones en un rango de entre 600V y 1000V.
2. Esta señal sirve como entrada a para la fuente que la transforma en voltaje de alimentación para el convertidor CD-CD.
3. El convertidor CD-CD es un subsistema, también construido por simulación. Durante el primer ciclo, el valor del ciclo de trabajo D es 1, a fin de obtener el mismo voltaje en la salida.
4. La señal de salida del convertidor pasa por una resistencia para tener una medición confiable.
5. La medición es el dato que se compara con un valor de referencia de 375 V, que también se puede considerar la salida deseada. La diferencia entre estos dos valores es el error.
6. Se utiliza un retardador para guardar el valor del error en el ciclo anterior. La diferencia entre éste y el error actual es el cambio en el error.
7. El módulo de control difuso recibe como entradas el error y el cambio en el error, procesa estos valores con base en las funciones de membresía definidas en la Tabla 6 y a las reglas de inferencia establecidas en forma interna, y regresa una variación en el ciclo de trabajo ΔD que puede ser negativa o positiva.

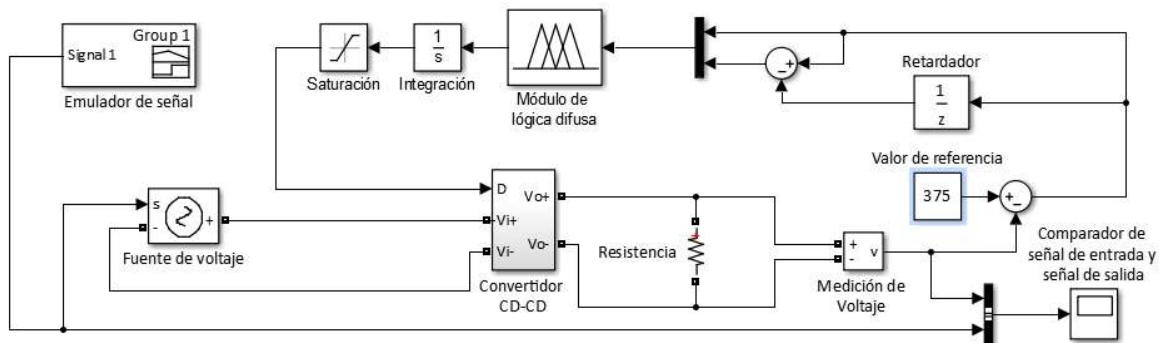


Figura 10. Modelo de simulación del circuito de control difuso para el convertidor buck
Fuente: Elaboración propia

8. El integrador obtiene el ciclo de trabajo a través de la sumatoria de todas las variaciones de éste.
9. Finalmente, se colocó un saturador, que impide que el ciclo de trabajo rebase los límites establecidos.
10. El comparador de señal de entrada y salida permite ver los resultados de la ejecución.

3.3.1.1. Resultados de la simulación

El controlador funciona para mantener el voltaje de salida a un valor constante a pesar de las variaciones del voltaje de alimentación. La Figura 11 muestra los resultados de la simulación del circuito en un transcurso de tiempo de 20 segundos.

En la gráfica, la señal superior (magenta) muestra la variación del voltaje de alimentación al convertidor, en un rango que varía entre los 600V y los 1000V y la señal inferior (azul), indica cómo la salida del convertidor se va ajustando y tiende a la estabilización en 375V (valor de referencia), para lograr un voltaje constante en un periodo de tiempo corto.

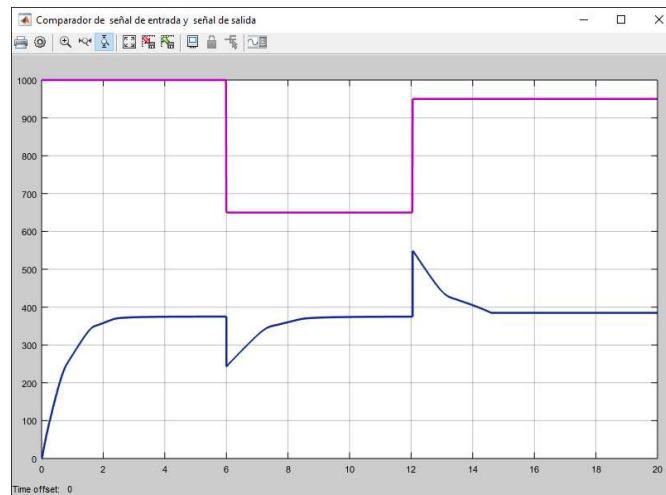


Figura 11. Resultados de la simulación del control para el convertidor buck
Fuente: Elaboración propia

3.3.2. Simulación del circuito de control para el convertidor boost

El proceso para elaborar la simulación del circuito de control para el convertidor boost, el cual se muestra en la Figura 12, fue similar al seguido para elaborar la simulación del circuito de control para el convertidor buck, primero se creó el módulo de control difuso y, posteriormente, el circuito de control completo.

El funcionamiento de ambos circuitos de control es, por lo tanto, similar. Las características diferenciadoras son:

1. El emulador de señal permite generar una señal discreta que puede ser configurada para realizar variaciones en un rango de entre 187.5 V y 375 V.
2. El valor de referencia o salida deseada es de 750 V.
3. El módulo de lógica difusa procesa el error y el cambio en el error con base en las funciones de membresía y sus valores, definidos en la Tabla 8.

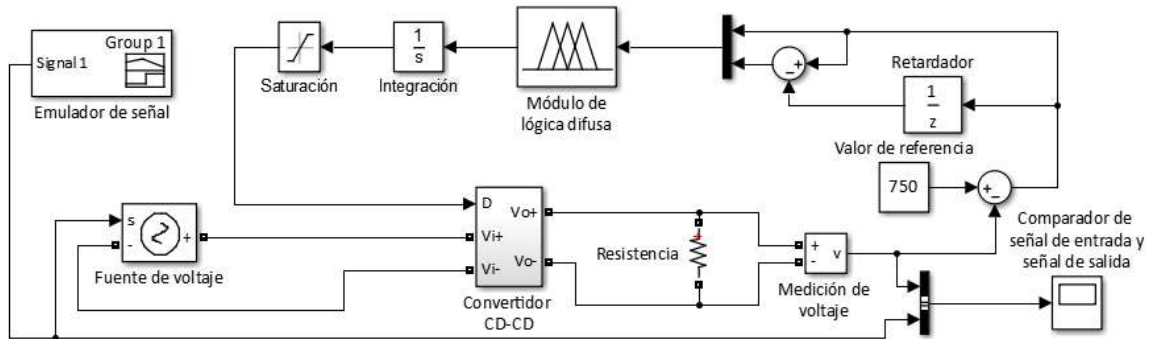


Figura 12. Modelo de simulación del circuito de control para el convertidor boost
Fuente: Elaboración propia

3.3.2.1. Resultados de la simulación

En la figura 13 se muestran los resultados de la simulación del circuito de control difuso para el convertidor boost en un intervalo de 20 segundos. Se puede apreciar la variación en el voltaje de entrada (señal inferior, color magenta), en un rango que va de 187.5V a 375V, y cómo el voltaje de salida (señal superior, color azul) tiende a la estabilización para lograr una salida constante de 750V que es el valor de referencia.

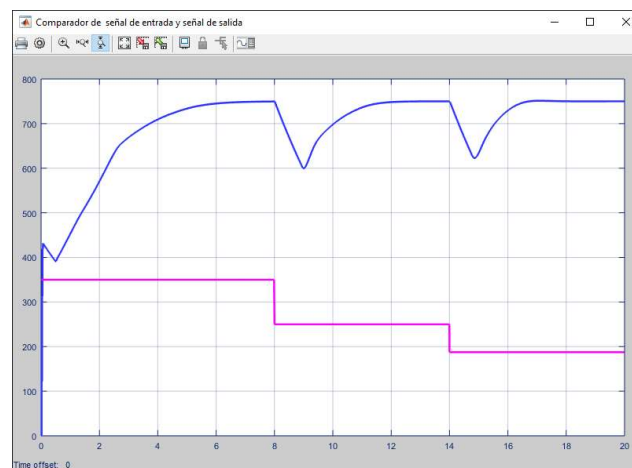


Figura 13. Resultados de la simulación del control para el convertidor en modo boost
Fuente: Elaboración propia

3.4. Implementación

El diagrama físico, para la implementación, se muestra en la Figura 14. El convertidor CD-CD es un componente electrónico y la aplicación es un componente de software. Para la comunicación entre ambos es necesario contar con un dispositivo de adquisición de señales. Debido a que la cantidad de mediciones por segundo que se requiere de la señal de salida convertidor debe ser alta, a fin de obtener una mejor calidad en los datos, se optó por usar el dispositivo de adquisición NI CompactRIO, ya que permite la configuración del periodo de tiempo en el que se harán las mediciones (National Instruments, 2014).

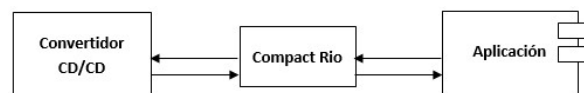


Figura 14. Diagrama físico para la implementación
Fuente: Elaboración propia

La implementación del controlador tanto para el modo buck como en modo boost se realizó en LabVIEW™ debido a las ventajas que presenta en cuanto a que: es un lenguaje visual, permite desarrollar aplicaciones rápidamente, permite una comunicación transparente con el CompactRIO y cuenta con una interfaz para crear el módulo de lógica difusa.

La implementación se realizó para ser probada en un circuito físico completo usando un convertidor CD-CD de prueba cuyo sensor de voltaje tiene un rango de medición de 0 a 55 V entregando una señal de medición acondicionada de 0 V a 3 V (Castellanos, 2013), por lo cual fue necesario realizar un escalamiento de los valores usados en la simulación.

3.4.1. Implementación del software de control para el convertidor buck

La Tabla 9 muestra las características de funcionamiento del convertidor de prueba en modo buck y la Tabla 10 muestra los intervalos de valores de las variables de entrada y salida para el controlador difuso, usados para la implementación.

Tabla 9. Características del convertidor buck de prueba.
Fuente: Elaboración propia

Característica	Valores
Intervalo de valores de voltaje de alimentación V_i	[21 - 35 V]
Voltaje de referencia V_{ref}	13.1 V
Intervalo de valores del ciclo de trabajo D	[0.38 - 0.62]

De manera análoga a la simulación, para la implementación del software de control primero fue necesario crear el módulo de control difuso. Esto se hizo utilizando el Fuzzy

System Designer integrado LabVIEW. Los pasos para implementar el módulo se describen en el Apéndice B.

Tabla 10. Intervalos de valores para las variables de entrada y salida del módulo difuso para el convertidor buck de prueba.

Fuente: Elaboración propia

Tipo	Característica	Valores
Entrada	Intervalo de valores del error	$[-21.5, 131 \text{ V}]$
Entrada	Intervalo de valores del cambio en el error	$[-34.5, 34.5 \text{ V}]$
Salida	Intervalo de valores del cambio del ciclo de trabajo D	$[-0.625, 0.625]$.

La aplicación terminada del software de control se muestra en la Figura 15.

La aplicación se desarrolló en dos partes:

- La primera parte obtiene la señal de salida del convertidor (Figura 15a) a través del puerto 0 del módulo de adquisición del CompactRIO y obtiene el error y el cambio en el error de la señal.

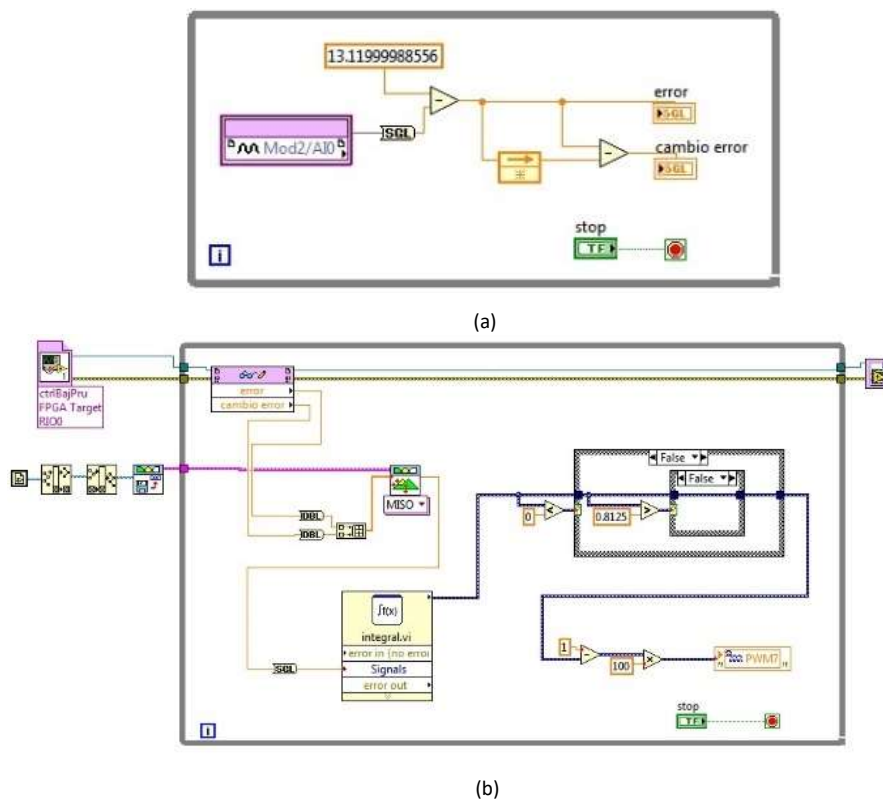


Figura 15. Implementación del software de control para el convertidor buck. (a) Adquisición de señal y obtención del error y cambio en el error, (b) Obtención del ciclo de trabajo

Fuente: Elaboración propia

- La segunda parte (Figura 15b), obtiene el error y el cambio en el error y los envía como entradas al módulo de lógica difusa. La salida del módulo es integrada para obtener el ciclo de trabajo y saturada para evitar el desbordamiento. Posteriormente, es enviada al puerto 7 del módulo de salida del CompactRIO el cual la convierte en una señal modulada de ancho de pulso.

3.4.2. Implementación del software de control para el convertidor boost

Los valores definidos en tiempo de diseño para el software de control para el convertidor boost también fueron escalados para poder ejecutarlo sobre el convertidor boost de prueba. La Tabla 11 muestra las características de funcionamiento del convertidor de prueba en modo buck y la Tabla 12 muestra los intervalos de valores de las variables de entrada y salida para el controlador difuso, usados para la implementación.

Tabla 11. Características del convertidor boost de prueba.

Fuente: Elaboración propia

Característica	Valores
Intervalo de valores de voltaje de alimentación V_i	[21 - 35 V]
Voltaje de referencia V_{ref}	13.1 V
Intervalo de valores del ciclo de trabajo D	[0.38 – 0.62]

Tabla 12. Intervalos de valores para las variables de entrada y salida del módulo difuso para el convertidor boost de prueba.

Fuente: Elaboración propia

Tipo	Característica	Valores
Entrada	Intervalo de valores del error	[-21.5, 131 V]
Entrada	Intervalo de valores del cambio en el error	[-34.5, 34.5 V]
Salida	Intervalo de valores del cambio del ciclo de trabajo D	[-0.625, 0.625].

La implementación del software de control para el convertidor boost se muestra en la figura 16.

Al igual que el software de control para el convertidor buck, esta aplicación requirió la implementación previa del módulo de lógica difusa y se realizó también en dos partes:

- La primera parte está compuesta por la adquisición de datos, a través del puerto 1 del módulo de adquisición del CompactRIO, y obtención de error y cambio en el error, (Figura 16a)
- La segunda parte se compone de la obtención del ciclo de trabajo, a través de la ejecución del módulo de lógica difusa, y su posterior envío al puerto 6 del módulo de salida del CompactRIO para convertirla en una señal modulada de ancho de pulso (Figura 16b).

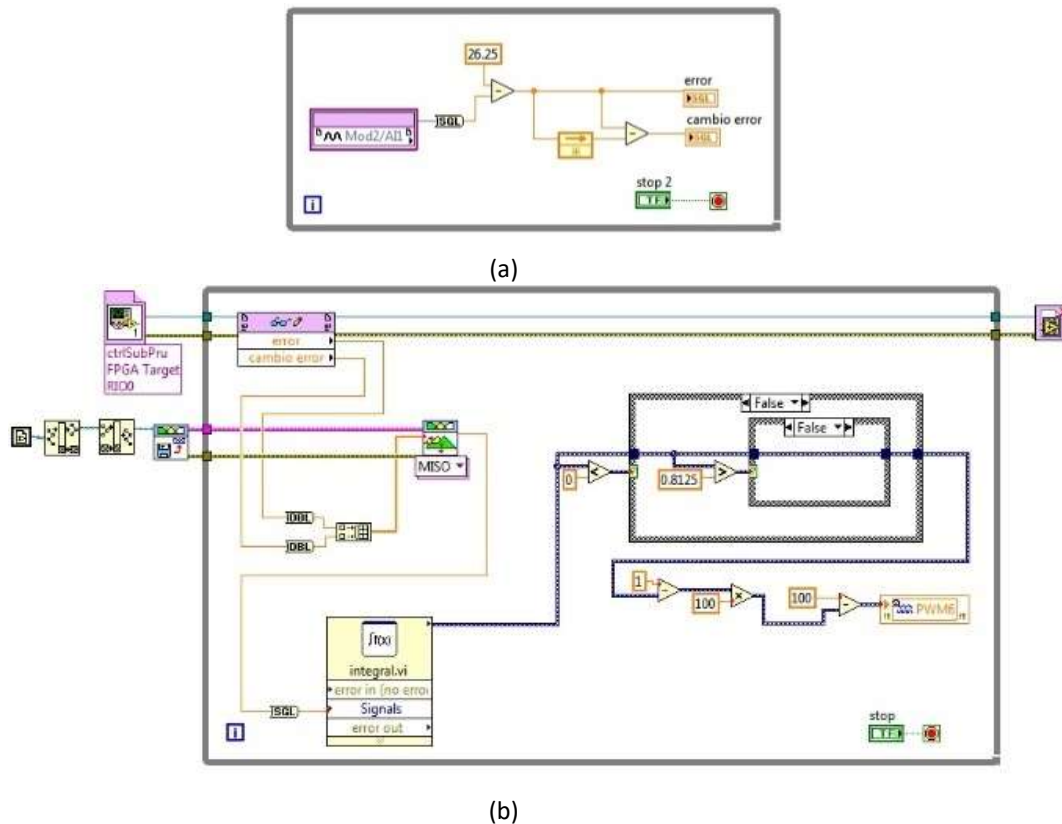


Figura 16. Implementación del software de control para el convertidor boost. (a) Adquisición de señal y obtención del error y cambio en el error, (b) Obtención del ciclo de trabajo
Fuente: Elaboración propia

3.4.3. Resultados

El controlador fue ejecutado en conjunto con el circuito físico completo para probar su funcionamiento. El circuito incluye un convertidor CD-CD de prueba, de 2kW, el cual se muestra en la Figura 17.

Para la visualización de los resultados se utilizó un osciloscopio para mostrar, al mismo tiempo, la señal alimentadora del convertidor y la señal de salida, de manera similar a lo que se realizó en la simulación. Esto se hizo para ambas configuraciones del convertidor.

3.4.3.1. Resultados de la ejecución del control para el convertidor buck

La figura 18 muestra las capturas de pantalla del osciloscopio en diferentes momentos de la ejecución. La señal inferior (azul) corresponde al voltaje de alimentación del convertidor y la señal superior (amarilla) a la señal de salida del mismo.

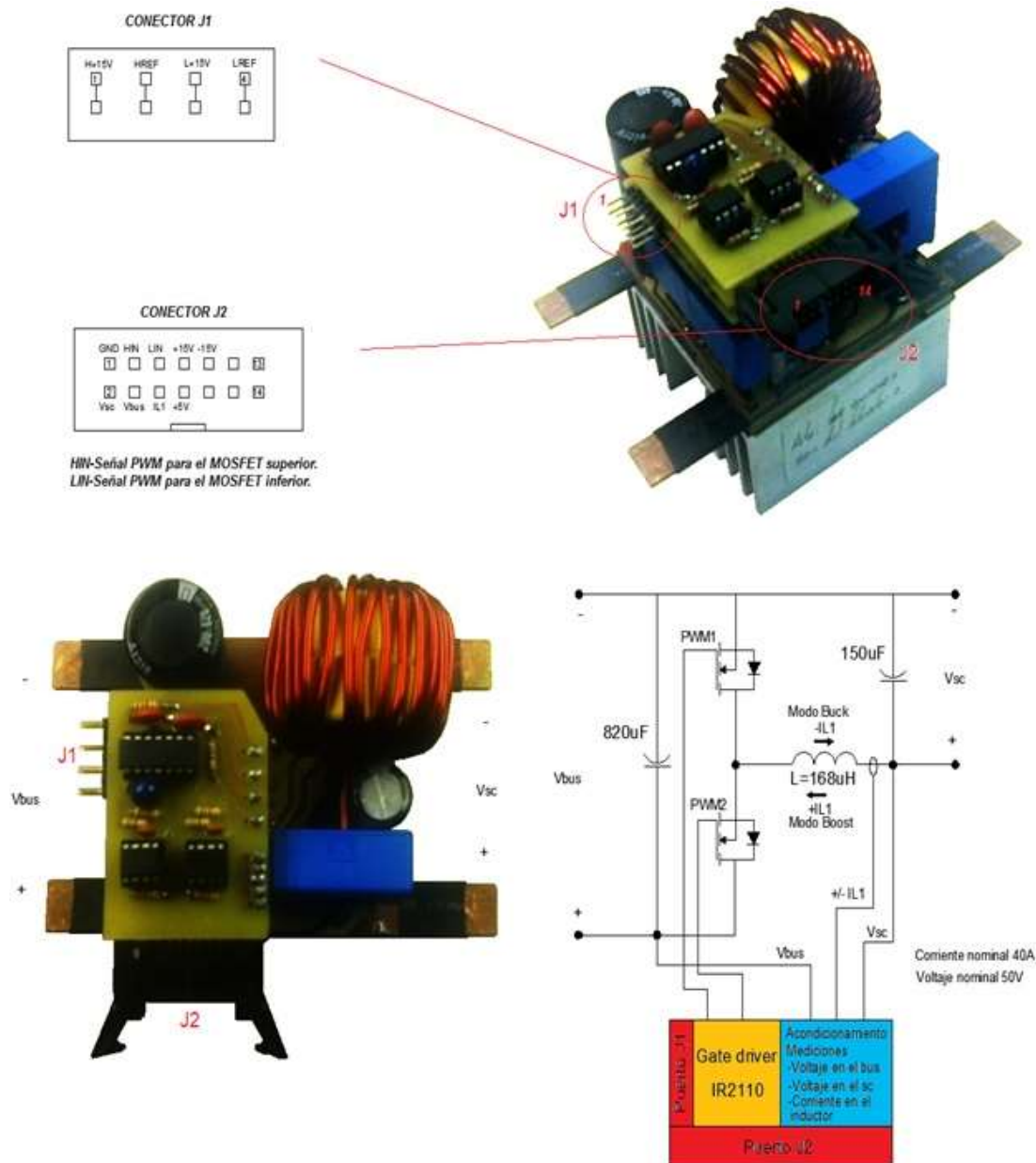
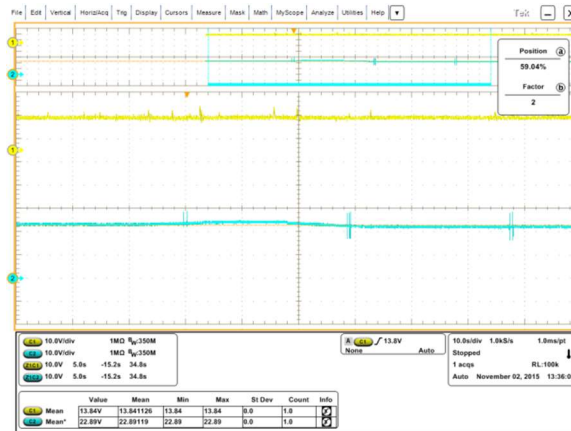
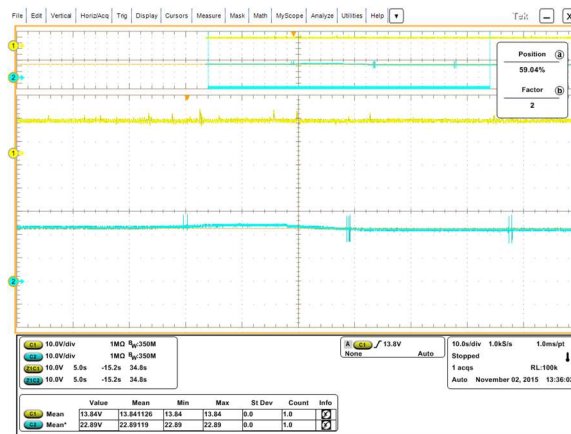


Figura 17. Convertidor CD-CD buck/boost de prueba: aspecto físico del convertidor, esquemático del convertidor, parámetros y configuración de los puertos de conexión.
Fuente: (Castellanos, 2013)

En la parte inferior de cada gráfica se muestra el valor de cada señal en el momento de la captura. La señal alimentadora varía de 22.89V (Figura 18a) a 21.13V (Figura 18b), mientras que la señal de salida varía de 13.84V a 13.59V, respectivamente, teniendo un promedio de 13.7, lo cual indica que se aproxima a la referencia de 13.1V marcada. La diferencia de entre 0.6V en ejecución se considera razonable.



(a)



(b)

Figura 18. Resultados en osciloscopio de la ejecución del software de control para el convertidor buck
Fuente: Elaboración propia

3.4.3.2. Resultados de la ejecución del control para el convertidor boost

De la misma manera, se probó la ejecución del controlador en modo boost usando el circuito físico completo.

La figura 19 muestra los resultados de la ejecución en dos momentos distintos. La señal superior (amarilla) corresponde al voltaje de alimentación del convertidor y la señal inferior (azul) a la señal de salida del mismo.

En la parte inferior de cada gráfica se muestra el valor de cada señal en el momento de la captura. La señal alimentadora se encuentra en el rango de los 8V (Figura 19a) y (Figura 19b) mientras que la señal de salida se encuentra en el rango de los 27V, conteniendo un promedio de 27.01V, lo cual indica que se aproxima a la referencia de 26.25V marcada. La diferencia de 0.76V en ejecución se considera razonable.

4. Software de control por redes neuronales

La teoría y el diseño de redes neuronales artificiales han avanzado significativamente durante las últimas décadas. Gran parte de ese progreso tiene una relación directa con el procesamiento de la señal. En particular, la naturaleza no lineal de las redes neuronales, su capacidad para aprender del entorno de una manera tanto supervisada como no supervisada, así como la propiedad de aproximación universal de las neuronas, hacen a las redes neuronales muy adecuadas para resolver problemas difíciles de procesamiento de señales (Hu, Hwang, 2001).

Lo anterior explica por qué se seleccionó el uso de redes neuronales para la construcción del software de control para un convertidor CD-CD bidireccional, junto con la lógica difusa, lo cual es el objeto de estudio que ocupa el proyecto de tesis que ampara este documento.

El desarrollo del software de control basado en redes neuronales, que es el tema de este capítulo, incluyó las etapas de análisis, diseño, simulación y obtención de resultados. Las siguientes secciones describen cada una de esas etapas.

4.1. Análisis

Para la etapa de análisis se consideran las especificaciones del módulo de control, así como las especificaciones técnicas de convertidor.

4.1.1. Especificaciones

Las especificaciones generales descritas en la sección 3.1.1. son válidas para este software de control.

La diferencia única es que el software de control automático para cada una de las configuraciones del convertidor deberá implementarse usando redes neuronales.

Las especificaciones técnicas del convertidor, descritas en la sección 3.1.2. son válidas en su totalidad para este control.

4.2. Diseño

Para la etapa de diseño se considera el diseño general de la red neuronal, el diseño del módulo de control para el convertidor buck y el diseño del módulo de control para el convertidor boost.

4.2.1. Diseño general de la red neuronal

La figura 20 presenta una instancia de la arquitectura del circuito de control mostrada en la figura 8, donde se resalta el lugar que ocupa la red neuronal como parte del software de control del convertidor (línea punteada).

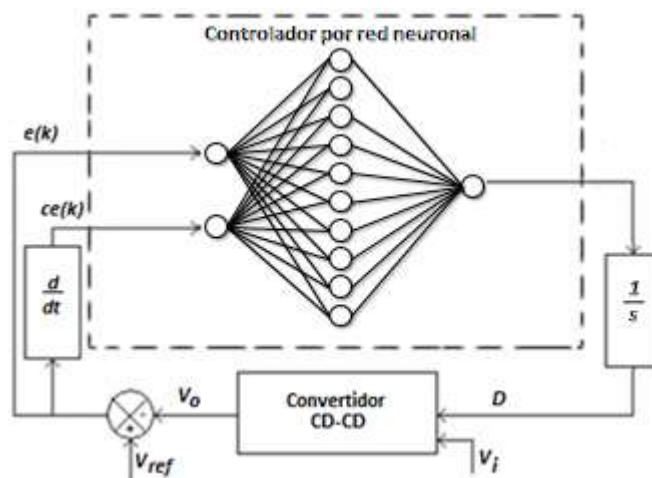


Figura 20. Arquitectura general del sistema de control
Fuente: Elaboración propia

Las entradas y salidas son las mismas que para el software de control por lógica difusa que fueron definidas en la sección 3.2.1.2.

El diseño involucra las siguientes actividades:

1. Selección de la topología de la red.
2. Diseño de la red.
3. Selección del método de entrenamiento.

4.1.1.1. Selección de la topología

La topología o arquitectura de una red consiste en la organización y disposición de las neuronas en la red formando capas de neuronas. En este sentido, los parámetros fundamentales para esta distinción son el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas (Aguilar, 1998). Esta es la razón de que existan muchas topologías para las redes neuronales.

Después de realizar un estudio de diferentes tipos de topologías, su uso y su funcionamiento, la topología seleccionada para el desarrollo del software de control fue la del perceptrón multicapa.

El perceptrón multicapa es una variante del modelo de perceptrón original propuesto por Rosenblatt (Rosenblatt, 1958). Una red neuronal del modelo perceptrón multicapa consiste de una red de neuronas organizadas en capas, las cuales son “alimentadas hacia atrás” (Kessal, Rahmani, Mostefai, Gaubert, 2012). Está compuesta por dos o más capas de neuronas. A la primera capa se le conoce como capa de entrada, las capas intermedias, que pueden ser una o más se conocen como capas ocultas y la última capa se conoce como capa de salida (Figura 21).

Cada una de las capas tiene asociada una función de activación no lineal. Algunas de las funciones de activación más frecuentemente usadas son la sigmoideal logarítmica, sigmoideal tangente hiperbólica o lineal.

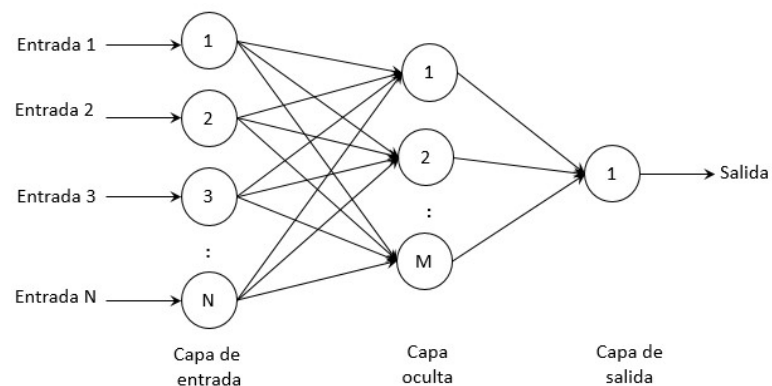


Figura 21. Red perceptrón multicapa
Fuente: Elaboración propia

4.1.1.2. Diseño de la red

Tomando en cuenta las características mencionadas en la sección anterior, se realizó el diseño de la red neuronal. La figura 22 muestra la topología de la red neuronal diseñada para el software de control para el convertidor.

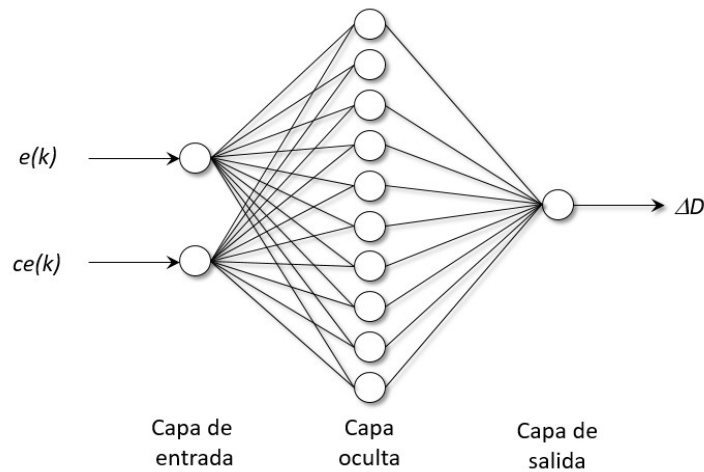


Figura 22. Topología de la red para el software de control para el convertidor
Fuente: Elaboración propia

Las decisiones adicionales resultantes del diseño son las siguientes:

1. La capa de entrada de las redes contiene dos neuronas que representan el error $e(k)$ y el cambio en el error $ce(k)$.
2. El número de neuronas de la capa oculta será 10 ya que es el número que crea por defecto el sistema que será usado para la simulación.
3. La función de transferencia para la capa oculta será lineal
4. La capa de salida contiene una sola neurona que representa el cambio en el ciclo de trabajo D .
5. Se utilizará la función de transferencia sigmoideal para la capa de salida.

Una gráfica de la red neuronal a mayor detalle, considerando las decisiones de diseño anteriores, se muestra en la Figura 23.

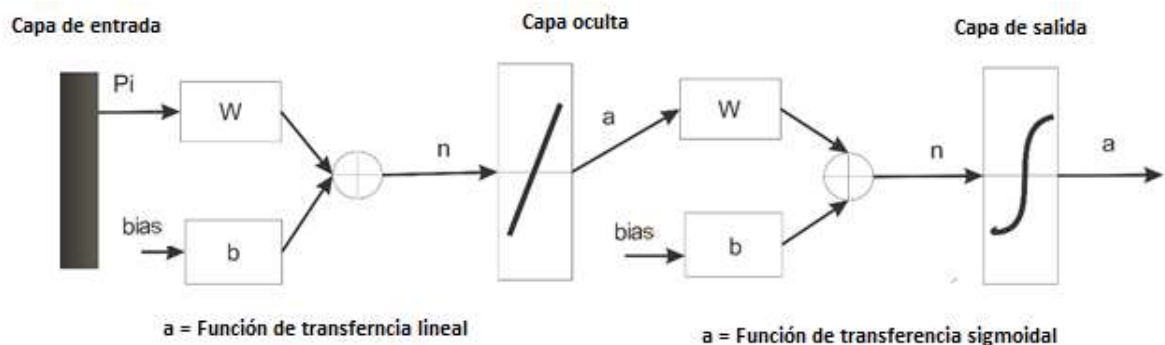


Figura 23. Red neuronal diseñada
Fuente: Elaboración propia

4.1.1.3. Selección del método de entrenamiento

La red perceptrón multicapa es entrenada mediante el algoritmo de retropropagación, ésta es la razón por la que también se conozca como red de retropropagación o red de propagación hacia atrás (Hu, Hwang, 2001).

El aprendizaje de una red neuronal consiste en hallar el conjunto adecuado de pesos sinápticos, para que la red pueda resolver correctamente un problema.

El perceptrón multicapa es una red de aprendizaje supervisado; por tanto, para su entrenamiento necesita un conjunto de pares entrada-salida como ejemplos. Los pesos sinápticos son ajustados de forma que al final del entrenamiento se obtengan las salidas esperadas para cada una de estas entradas de ejemplo.

Durante el proceso de entrenamiento, el perceptrón multicapa tiende a desarrollar relaciones internas entre neuronas.

La retropropagación consiste en propagar el error hacia atrás durante el entrenamiento, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas. El error es la diferencia entre la salida obtenida al ejecutar la red neuronal y la salida deseada.

El número de neuronas de las capas ocultas debe ser lo suficientemente grande como para que se forme una región compleja que pueda resolver el problema, sin embargo, no debe ser muy grande, pues las regiones de decisión se limitarían solo a los ejemplos de entrenamiento.

La ecuación matemática para el algoritmo está dada por (48)

$$n \sum_{i=1}^{i=q} (w_i * p_i) + b_i \quad (48)$$

Donde

p_i Son las entradas a la neurona

w_i Es la matriz de pesos asignados a las entradas de la red neuronal

b_i Es la ganancia de la red

Para calcular el valor de la neurona la salida de la se modifica la red de neuronas de entrada con una función de transferencia, que se muestra a continuación

$$a = f(n) = f \left(n \sum_{i=1}^{i=q} (w_i * p_i) + b_i \right) \quad (49)$$

Los pasos del algoritmo de retropropagación son los siguientes (Hu, Hwang, 2001):

1. Se inicializan los valores de los pesos sinápticos de toda la red; por lo general se asignan valores aleatorios.
2. Se presenta el primer ejemplo de entrada a la red, junto con su salida esperada.
3. Se calcula la entrada total de cada neurona de la capa oculta.
4. Se calculan las salidas de la red
5. Se calculan los términos de error para las neuronas de la capa de salida.
6. Se calculan los términos de error para las neuronas de la capa oculta.
7. Se actualizan los pesos sinápticos de las neuronas de la capa de salida.
8. Se actualizan los pesos sinápticos de la capa oculta
9. Se presenta el siguiente ejemplo de entrada a la red, junto con su salida esperada y se regresa al paso 3. Si no existen más ejemplos se calcula el error cuadrático medio de todos los ejemplos.

El entrenamiento finaliza cuando el error cuadrático medio de todos los ejemplos, resulta aceptable para el funcionamiento de la red.

4.1.2. Diseño del control buck

Considerando el análisis de los datos para las variables de entrada y salida del módulo de control descrito en la sección 3.2.2., se realizaron los ajustes necesarios para adecuar a las características de la red. En la Tabla 13 se muestran los intervalos de valores tanto para las variables de entrada como para la variable de salida de la red neuronal para el control para el convertidor buck.

Tabla 13. Intervalos de valores para las variables de entrada y salida de la red para el funcionamiento del convertidor en modo buck

Fuente: Elaboración propia

Tipo	Característica	Valores
Entrada	Intervalo de valores del error	[-700 a 400]
Entrada	Intervalo de valores del cambio en el error	[-1000 a 1000]
Salida	Intervalo de valores del cambio del ciclo de trabajo D	[-0.5 a 0.5]

Se resolvió generar un arreglo de 21 valores, usando una distribución normal con el valor 0 como centro del arreglo (elemento 11), para cada una de las variables, tanto de entrada como de salida, para la alimentación de la red en el proceso de entrenamiento.

4.1.3. Diseño del control boost

De la misma manera, se ajustaron los intervalos de valores para las variables del módulo control para el convertidor boost. En la tabla 14 se muestran estos intervalos de valores.

Tabla 14. Intervalos de valores para las variables de entrada y salida de la red para el funcionamiento del convertidor en modo boost
Fuente: Elaboración propia

Tipo	Característica	Valores
Entrada	Intervalo de valores del error	[-2250 a 500]
Entrada	Intervalo de valores del cambio en el error	[-2800 a 2800]
Salida	Intervalo de valores del cambio del ciclo de trabajo D	[-0.8 a 0.8]

Para definir los valores de las variables de este módulo de control, que serán usados para alimentar la red en el proceso de aprendizaje, se resolvió también usar arreglos de 21 elementos para cada una de ellas.

4.2. Simulación

La siguiente etapa del desarrollo del software de control basado en redes neuronales consiste en realizar la simulación de los circuitos de control para cada una de las configuraciones del convertidor.

Las simulaciones se realizaron en Matlab®, usando el módulo Simulink.

Las secciones siguientes describen cómo se realizó esta simulación y los resultados de su ejecución.

4.2.1. Simulación del circuito de control para el convertidor buck

Antes de crear el modelo del circuito fue necesario crear la red neuronal, configurarla, entrenarla y presentarla como un bloque de simulación. Los pasos detallados para construir este módulo se describen en el Apéndice C.

A partir del modelo de simulación creado previamente para el convertidor buck y mostrado en la Figura 10, se creó el modelo de simulación del circuito de control para el convertidor buck basado en redes neuronales, sustituyendo el módulo de lógica difusa por el módulo de la red neuronal.

El modelo del circuito de control se muestra en la Figura 24.

El funcionamiento del modelo es similar al descrito en la sección 3.3.1. Las diferencias son:

1. El módulo basado en redes neuronales procesa los valores del error y el cambio de error, que recibe como entradas, con base en los valores internos de los pesos sinápticos de las neuronas, para regresar una variación en el ciclo de trabajo ΔD que puede ser negativa o positiva.
2. Los valores internos de los pesos sinápticos de las neuronas de la red, fueron previamente calculados durante el proceso de entrenamiento, basados en los

arreglos creados a partir en los intervalos de valores de las variables de entrada y salida definidos en la tabla 13.

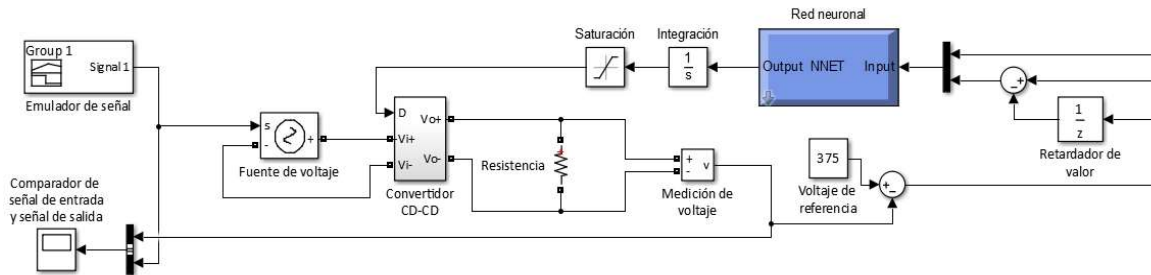


Figura 24. Modelo de simulación del circuito de control basado en redes neuronales para el convertidor buck
Fuente: Elaboración propia

4.2.1.1. Resultados de la simulación

La Figura 25 muestra los resultados de la simulación del circuito en un transcurso de tiempo de 20 segundos.

En la gráfica, la señal superior (color azul) muestra la variación del voltaje de alimentación al convertidor, en un rango que va de los 1000V a los 800V y luego a los 900V. La señal inferior, indica cómo la salida del convertidor se va ajustando y tiende a la estabilización para lograr un voltaje constante de 375V en un periodo de tiempo corto, lo cual se logra por la ejecución de la red neuronal que calcula el cambio que debe aplicarse al ciclo de trabajo cuando existen variaciones en el voltaje de alimentación.

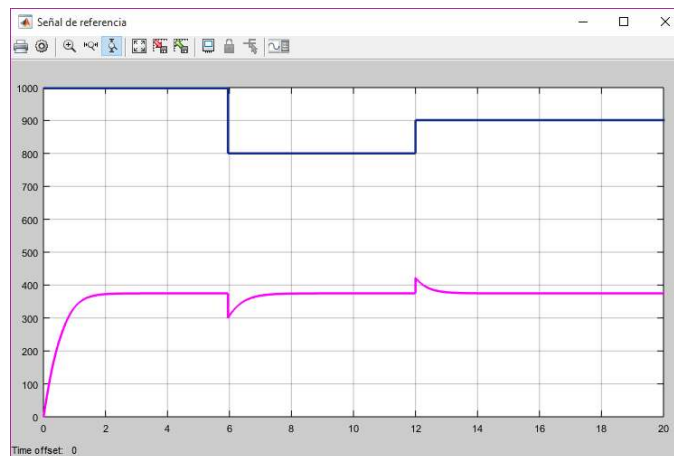


Figura 25. Resultado de ejecución del control en modo buck operado por la red neuronal
Fuente: Elaboración propia

4.2.2. Simulación del circuito de control para el convertidor boost

El desarrollo de la simulación del circuito de control para el convertidor boost, fue similar al seguido para elaborar la simulación del circuito de control para el convertidor buck, es decir, primero se creó, configuró, entrenó y generó la simulación de la red neuronal y, posteriormente, el circuito de control completo.

La Figura 26 muestra el modelo de simulación del circuito completo para el convertidor boost. El funcionamiento del modelo es similar al descrito en la sección 3.3.1. Las diferencias son:

1. El emulador de señal permite generar una señal discreta que puede ser configurada para realizar variaciones en un rango de entre 187.5 V y 375 V.
2. El valor de referencia o salida deseada es de 750 V.
3. El módulo basado en redes neuronales procesa los valores del error y el cambio de error, que recibe como entradas, con base en los valores internos de los pesos sinápticos de las neuronas, para regresar una variación en el ciclo de trabajo ΔD que puede ser negativa o positiva.
4. Los valores internos de los pesos sinápticos de las neuronas de la red, fueron previamente calculados durante el proceso de entrenamiento, basados en los arreglos creados a partir en los intervalos de valores de las variables de entrada y salida definidos en la tabla 13.

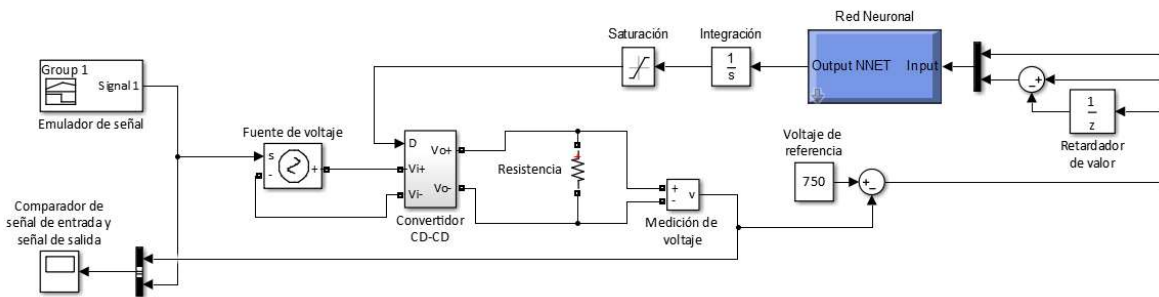


Figura 26. Modelo de simulación del circuito de control basado en redes neuronales para el convertidor boost

Fuente: Elaboración propia

4.3.2. Resultados de la simulación

En la figura 27 se muestran los resultados de la simulación del circuito para el convertidor buck. Se puede apreciar la variación en el voltaje de entrada (señal inferior, color magenta), en un rango que va de 200V a 375V y luego a 250V. Se muestra también cómo el voltaje de salida (señal superior, color azul) tiende a la estabilización para lograr una salida constante de 750V que es el valor de referencia.

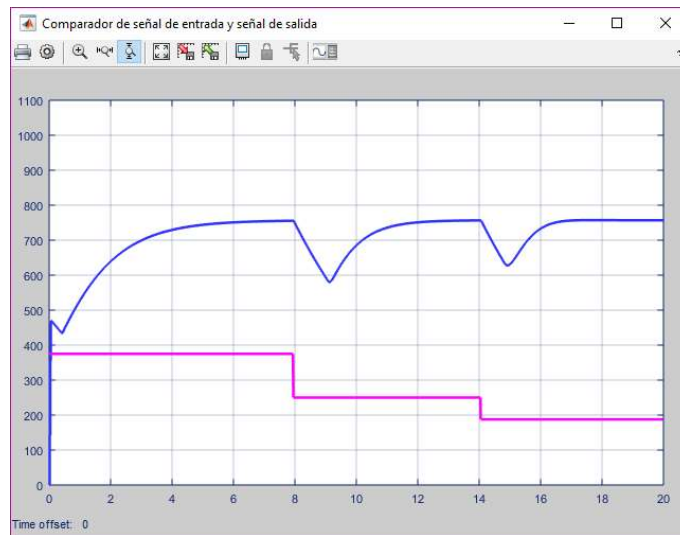


Figura 27. Resultado de ejecución del control en modo boost operado por la red neuronal
Fuente: Elaboración propia

5. Comparación de algoritmos

En los capítulos anteriores se ha presentado el desarrollo de un software de control para un convertidor CD-CD bidireccional, usando dos algoritmos inteligentes diferentes: la lógica difusa y las redes neuronales. Ambos softwares de control fueron probados por separado y se demostró que funcionan de manera correcta, es decir, cumplen la función de estabilizar el convertidor en un el voltaje de salida determinado por voltaje de referencia, para ambas configuraciones del mismo.

Como parte de la evaluación, se resolvió comparar los dos algoritmos para determinar cuál es la mejor opción a ser implementada para el convertidor CD-CD y cuyas características técnicas se definieron en la sección 3.1.2.

Se realizaron pruebas de ejecución y pruebas de tiempo de ejecución.

5.1 Pruebas de ejecución

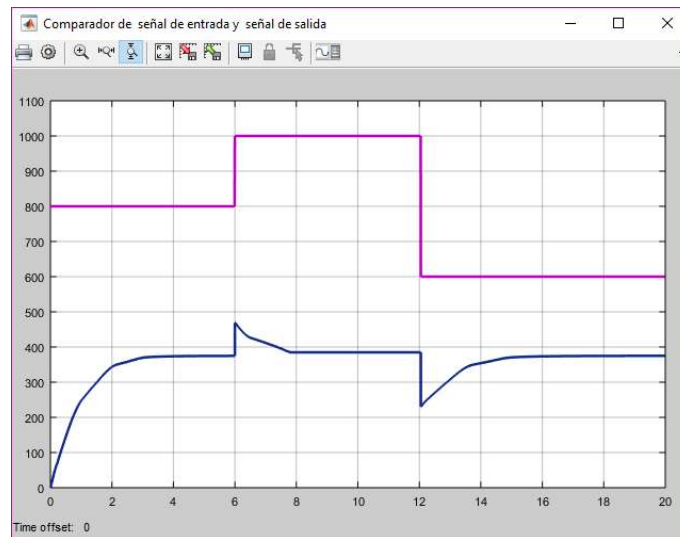
Ambos algoritmos se probaron para mostrar su ejecución. Para realizar esta prueba no fue necesario la sincronización por lo que las ejecuciones se realizaron por separado y se tomaron las capturas de pantalla correspondientes.

Se realizaron pruebas para ambas configuraciones del convertidor.

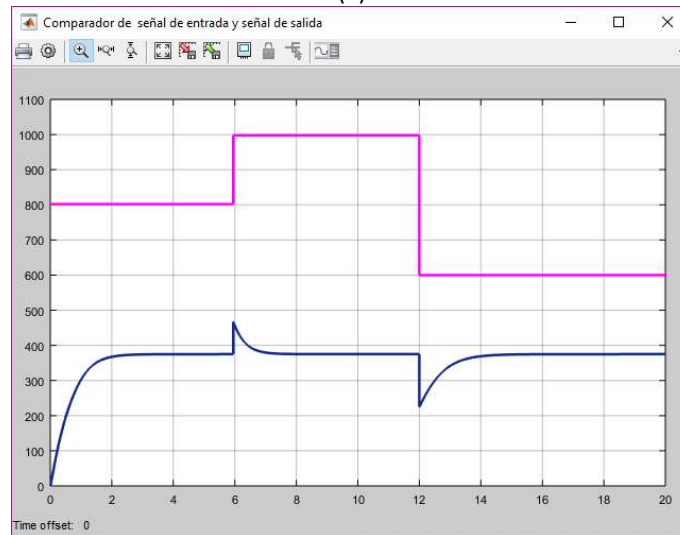
5.1.1. Pruebas de ejecución para el convertidor buck

La Figura 28 muestra la ejecución de los circuitos de control para el convertidor buck. La imagen de la Figura 28a muestra la ejecución del algoritmo difuso mientras que la Figura 28b muestra la ejecución del algoritmo basado en redes neuronales. Las conclusiones del análisis de ambas señales es el siguiente:

- La ejecución de ambos algoritmos es correcta.
- La señal del algoritmo basado en redes neuronales presenta una mejor continuidad que la del algoritmo difuso.



(a)



(b)

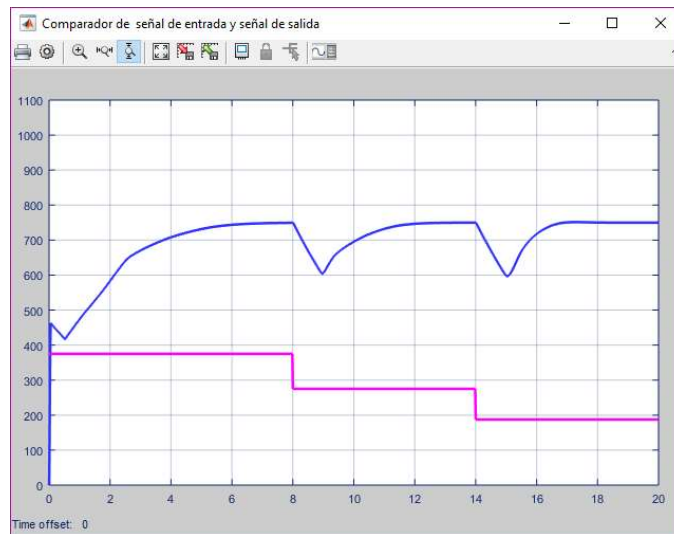
Figura 28. Resultados de la simulación del control para el convertidor buck (a) Módulo de control difuso (b) Módulo de control por redes neuronales
Fuente: Elaboración propia

- Los picos que se generan en las señales de ambos algoritmos cuando existe un cambio en el voltaje son muy similares.
- El algoritmo basado en redes neuronales permite que la señal se estabilice más rápido en el punto de referencia que el algoritmo difuso. Para este caso, la señal neuronal se estabiliza pasados 2 segundos de iniciar la ejecución, mientras la señal neuronal se estabiliza después de 3 segundos. Después del cambio en el voltaje de entrada, que ocurre en el segundo 6, la señal neuronal se estabiliza cerca del segundo 7, mientras la señal neuronal se estabiliza cerca del segundo 8. Finalmente, después del segundo cambio en el voltaje de entrada, que ocurre en

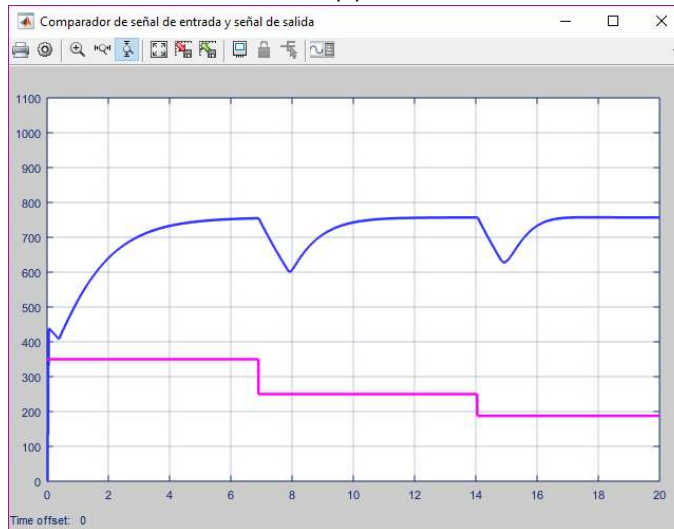
el segundo 12, la señal neuronal se estabiliza alrededor del segundo 14, mientras la señal neuronal se estabiliza alrededor segundo 15.

5.1.2. Pruebas de ejecución para el convertidor boost

La Figura 29 muestra la ejecución de los circuitos de control para el convertidor boost.



(a)



(b)

Figura 29. Resultados de la simulación del control para el convertidor boost. (a) Módulo de control difuso.

(b) Módulo de control por redes neuronales.

Fuente: Elaboración propia

La imagen de la Figura 29a muestra la ejecución del algoritmo difuso mientras que la Figura 29b muestra la ejecución del algoritmo basado en redes neuronales. El análisis de ambas ejecuciones permite obtener conclusiones similares para el convertidor boost que para el convertidor buck:

- La ejecución de ambos algoritmos es correcta.
- La señal del algoritmo basado en redes neuronales presenta una mejor continuidad que la del algoritmo difuso.
- Los picos que se generan en las señales de ambos algoritmos cuando existe un cambio en el voltaje son muy similares.
- El algoritmo basado en redes neuronales permite que la señal se estabilice más rápido en el punto de referencia que el algoritmo difuso.

5.2. Pruebas de tiempo de ejecución

Ambos algoritmos se probaron para comparar su tiempo de ejecución. Para realizar esta prueba fue necesaria la sincronización, es decir, ambos algoritmos se ejecutaron al mismo tiempo y se tomaron las capturas de pantalla correspondientes.

Se realizaron pruebas para ambas configuraciones del convertidor.

5.2.1. Pruebas de tiempo de ejecución para el convertidor buck

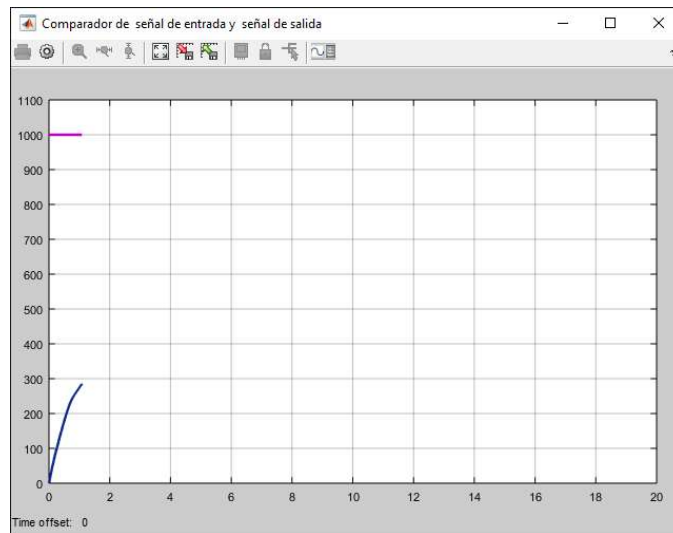
Para el modo buck se resolvió ejecutar ambos algoritmos por un intervalo de tiempo de 7 segundos medidos con el reloj interno de la computadora. Después de transcurrido este tiempo, se detuvo la ejecución y se observaron las gráficas resultantes.

La Figura 30 muestra la ejecución de los circuitos de control para el convertidor buck. La imagen de la Figura 30a muestra la ejecución del algoritmo difuso mientras que la Figura 30b muestra la ejecución del algoritmo basado en redes neuronales.

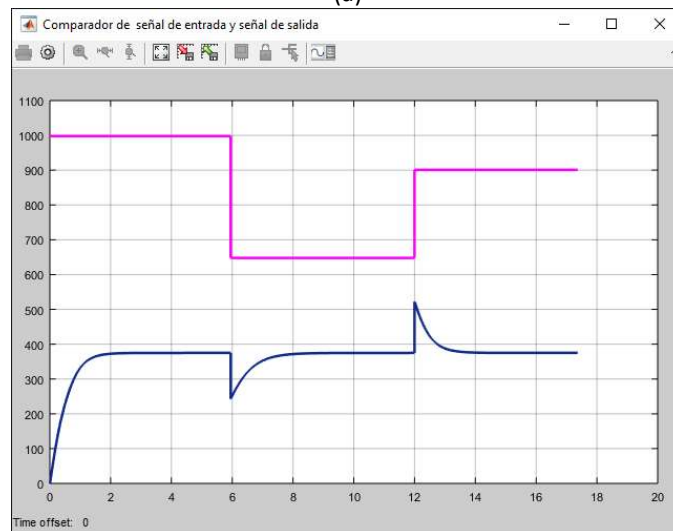
Se puede observar claramente que la ejecución del algoritmo basado en redes neuronales es más rápida que la del algoritmo difuso.

5.2.1. Pruebas de tiempo de ejecución para el convertidor boost

La prueba se realizó también para el convertidor boost. Debido a que la ejecución es más lenta, se resolvió que el tiempo de la muestra sería de 10 segundos del reloj interno de la computadora. Después de transcurrido este tiempo, se detuvo la ejecución y se observaron las gráficas resultantes



(a)



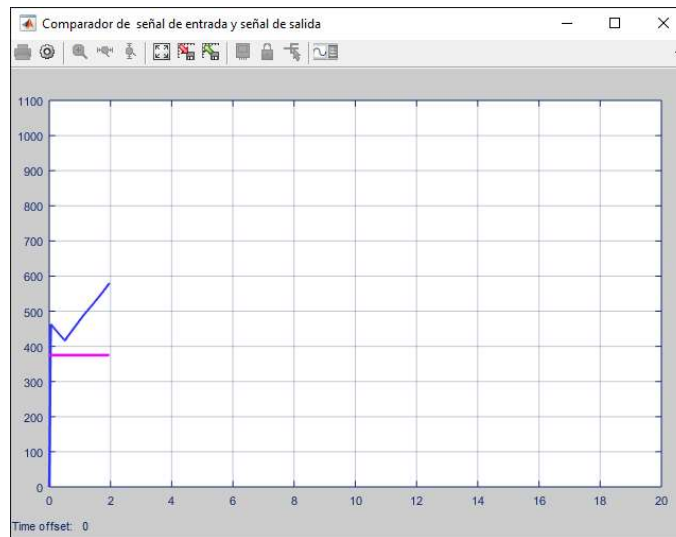
(b)

Figura 30. Resultados de la simulación del control para el convertidor buck después de 7 segundos del reloj interno. (a) Controlador difuso (b) Controlador neuronal

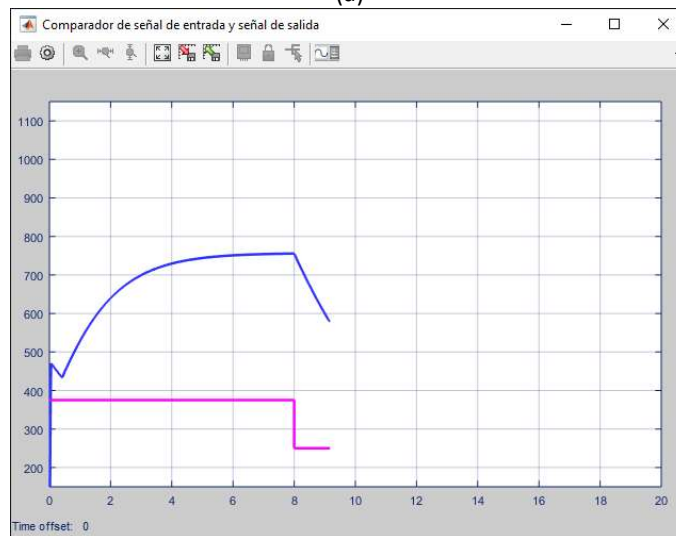
Fuente: Elaboración propia

La Figura 31 muestra la ejecución de ambos algoritmos. La Figura 31a muestra el funcionamiento del algoritmo difuso y la Figura 31b el funcionamiento del algoritmo basado en redes neuronales.

Los resultados obtenidos para el convertidor boost son muy similares a los obtenidos para el convertidor buck, ya que se puede apreciar claramente que la ejecución del algoritmo basado en redes neuronales es más rápido que el algoritmo difuso.



(a)



(b)

Figura 31. Resultados de la simulación del control para el convertidor buck después de 10 segundos del reloj interno. (a) Controlador difuso (b) Controlador neuronal

Fuente: Elaboración propia

5.2. Discusión

Con base en los resultados anteriores se considera lo siguiente:

1. Como se mencionó, el objetivo del controlador es lograr la estabilización de la señal de alimentación en un valor de referencia indicado, en este sentido, como se puede apreciar en las Figuras 27 y 28, los algoritmos de ambos controladores funcionan de manera similar, tanto para el modo buck como modo boost.

-
2. La estabilidad y continuidad de las señales es un factor importante puesto que protege la vida útil de los dispositivos. Aunque este factor no es determinante, el algoritmo basado en redes neuronales posee esta característica mejor que el algoritmo difuso.
 3. El tiempo sí es un factor determinante en esta comparación. El algoritmo basado en redes supera al algoritmo difuso tanto en tiempo de ejecución como en tiempo de estabilización.

Por las razones anteriores se determina el algoritmo basado en redes neuronales es el candidato ideal a ser seleccionado para la implementación del software de control para el convertidor CD-CD que es pieza fundamental en el proyecto denominado “Sistemas de Ahorro de Energía para los trenes del metro de la Ciudad de México”.

6. Conclusiones

A lo largo de este documento, se ha presentado el desarrollo de un software de control para un convertidor de CD-CD bidireccional, es decir, que funciona en dos sentidos, el convertidor buck y el convertido boost.

El desarrollo del software de control se realizó usando dos variantes de algoritmo de control, uno basado en lógica difusa y otro basado en redes neuronales, y bajo una metodología de ingeniería de software.

Las conclusiones obtenidas de este desarrollo se agrupan en dos categorías: conclusiones sobre el proceso de desarrollo y conclusiones sobre el proyecto.

De esta manera, se obtuvieron las siguientes conclusiones sobre el proceso:

1. El realizar un análisis previo es fundamental para la realización de cualquier proyecto de software. En este proyecto de tesis el análisis previo permitió:
 - Entender el contexto en el que se encuentra inmerso el proyecto, es decir, el papel que juega dentro del proyecto general y la importancia que tiene para el logro de su objetivo.
 - Tener un panorama general del proyecto al entender la relación que tiene con el convertidor, el cual es un dispositivo electrónico, y delimitarlo.
 - Obtener especificaciones claras del convertidor, en sus dos configuraciones, así como del control de software a construir para él.
 - Comprender los algoritmos de control inteligente, las diferencias entre ellos y poder seleccionar dos de ellos completamente distintos en concepción: la lógica difusa, basada en la teoría de conjuntos difusos, es decir, sin una delimitación clara, y las redes neuronales, que tratan de emular el proceso de razonamiento de las neuronas humanas.
 - Tener clara la estrategia de control para el convertidor, la cual es fundamental para todas las demás etapas del desarrollo.
2. El diseño descendente, que va de lo general a lo particular, permitió tener un panorama claro de lo que se tenía que construir en las simulaciones:

-
- Primero, del software de control general, que tiene como entrada la diferencia entre el voltaje de salida del convertidor y un voltaje de referencia, es decir, el error; y como salida el valor del ciclo de trabajo a aplicar al convertidor para obtener el voltaje de salida deseado.
 - Después, del módulo de control, que tiene como entradas el error, pero también la diferencia del valor actual de éste con el valor obtenido en el ciclo anterior, es decir, un cambio en el error; y como salida un factor de ajuste para el ciclo de trabajo, que se podría considerar como la diferencia entre el valor del ciclo de trabajo anterior y el ciclo de trabajo actual que se espera obtener.
 - El hacer un diseño por niveles permite realizar, posteriormente, pruebas por niveles.
3. La realización de la simulación del software de control tiene ventajas para el uso de los recursos disponibles del proyecto, entre ellas:
 - Ahorro de tiempo al contar con un prototipo del software, funcionando de manera correcta, sin necesidad de construir el software final.
 - Hacer de este prototipo de software el sujeto de pruebas para determinar la efectividad del diseño elaborado, su funcionalidad y su eficiencia.
 4. Las pruebas permitieron:
 - Probar la funcionalidad de cada uno de módulos de control por separado.
 - Probar elementos no funcionales tales como la exactitud, que se refiere al grado en el que el producto cumple con su especificación, y la eficiencia, se refiere al uso de los recursos tales como tiempo de ejecución y memoria de ejecución, cuando se ejecutaron al mismo tiempo.

Adicionalmente, se obtuvieron las siguientes conclusiones sobre el proyecto:

1. La estrategia de control determinada durante la etapa de análisis demostró ser la correcta para resolver el tipo de problemática que se planteó para el proyecto en el capítulo de introducción.
 2. El seleccionar el uso de dos tipos de algoritmo diferentes tanto en su concepción como en su implementación, pero adecuado para el mismo proyecto, permitió hacer una comparación pertinente de ello y, posteriormente, hacer una selección informada sobre cuál de ellos es el mejor candidato a utilizar para realizar la implementación del software de control final para el proyecto general.
 3. La implementación, en forma de simulación, de los dos tipos algoritmos de control y su evaluación posterior, hicieron posible construir un sistema de control por software para el convertidor CD-CD, adecuado a las características específicas del mismo, lo cual contesta de manera afirmativa a la pregunta de investigación que se planteó en el capítulo de introducción.
 4. El realizar el desarrollo del software de control bajo una metodología de ingeniería de software permitió cumplir, en cierta medida, las dos metas de esta disciplina que son el contar con un producto de calidad, puesto que las pruebas
-

realizadas fueron exitosas, y mejorar la eficiencia del desarrollador, puesto que las conclusiones obtenidas sobre el proceso lo demuestran.

5. La realización de este proyecto permitió la presentación de un artículo de divulgación en un congreso internacional (Moreno-Aguilar, L. M., Peralta-Sánchez, E., 2016), y la publicación de un artículo de divulgación en una revista arbitrada (Moreno-Aguilar, L. M., Peralta-Sánchez, E., Morales-Salgado, M. R. 2016).

Finalmente, se puede concluir que el objetivo planteado para el proyecto y definido en el capítulo de introducción se cumplió debido a que, con los resultados obtenidos en la ejecución del controlador sobre el convertidor, se comprobó la efectividad del algoritmo de control para lograr que la señal de salida del convertidor se mantenga a un voltaje determinado por una referencia, a pesar de las variaciones de voltaje de la señal de alimentación del mismo.

6.1. Trabajo a futuro

Un proyecto de software nunca puede ser considerado como completamente concluido, puesto que siempre habrá alguna adición que hacer, alguna parte del mismo que actualizar o alguna opción que mejorar. Las direcciones en las que se puede enfocar el desarrollo futuro de este proyecto son:

- Probar los módulos de control en un convertidor simulado por hardware y no por software como en este proyecto. El simulador por hardware es un dispositivo electrónico cuyo funcionamiento, a través de las configuraciones correctas, se apega en forma más fiel al funcionamiento de un convertidor real y se encuentra disponible en el laboratorio de pruebas.
- Se podría extender a pruebas realizadas a los elementos no funcionales del software. Dado que para este proyecto las pruebas se enfocaron a la exactitud y la eficiencia, sería conveniente realizar pruebas al software de control sobre conceptos tales como la confiabilidad, que se refiere a la habilidad del producto de responder ante situaciones no esperadas, la integridad que se refiere a la protección del programa y sus datos de accesos no autorizados o, en este caso a interferencias, o la usabilidad, que se refiere a la facilidad de operación del producto por parte de los usuario.
- Realizar la implementación del software de control basado en redes neuronales y probarlo con el convertidor de prueba. Este sería el trabajos a futuro más importante de realizar debido a que el control basado lógica difusa ya fue implementado en este proyecto y a que éste es el algoritmo que demostró mejores cualidades de ejecución tanto funcionales como no funcionales y se seleccionó como mejor candidato para la implementación.

Apéndice A. Simulación del módulo de lógica difusa

El módulo de control basado en lógica difusa se creó usando el Fuzzy Logic Toolbox™ de Matlab®. Las características seleccionadas para su funcionamiento son las siguientes:

- Fuzzificación: singleton.
- Norma S: máximo.
- Norma T: producto.
- Implicación: mamdani.
- Tipo de funciones miembro: triangular.
- Defuzzificación: promedio centro ponderado.

A continuación, se describe paso a paso el proceso elaborar la simulación.

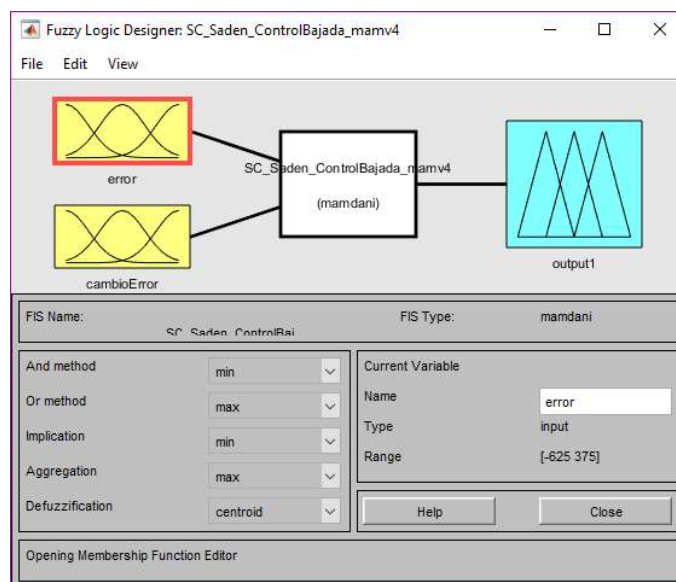


Figura A.1. Configuración de variable de entrada *error*

Paso 1. Se selecciona la variable de entrada por defecto y se configura para convertirla en la variable *error* definida en el diseño. (Figura A.1)

Paso 2. Se edita la variable entrada *error* (Figura A.2) y declararan las funciones d membresía, para ello:

- Se seleccionan el tipo *trimf* que identifica a la función triangular
- Se toman como parámetros los valores de la variable lingüística anterior, la de la actual y la de la siguiente, definidos en la Tabla 6 y de acuerdo con la definición de la sección 3.2.1.3.

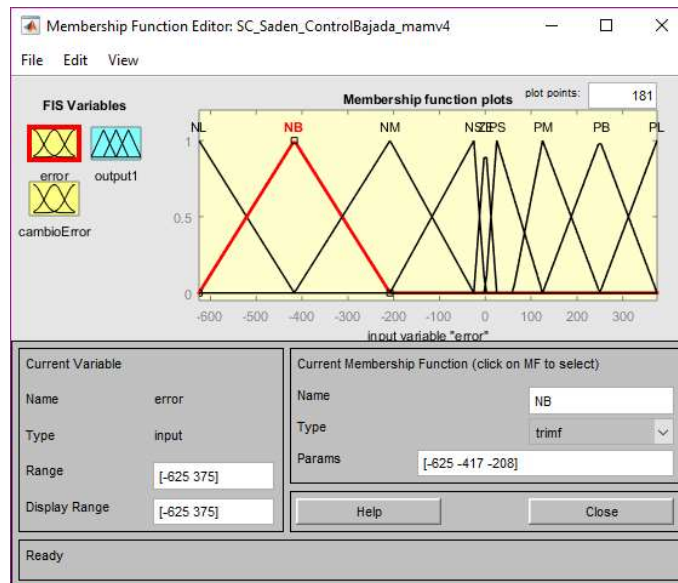


Figura A.2. Declaración de las funciones miembro para la variable de entrada *error*

Paso 3. Se adiciona una nueva variable de entrada y se configura para ser la variable *cambioError*, definida durante el diseño (Figura A.3).

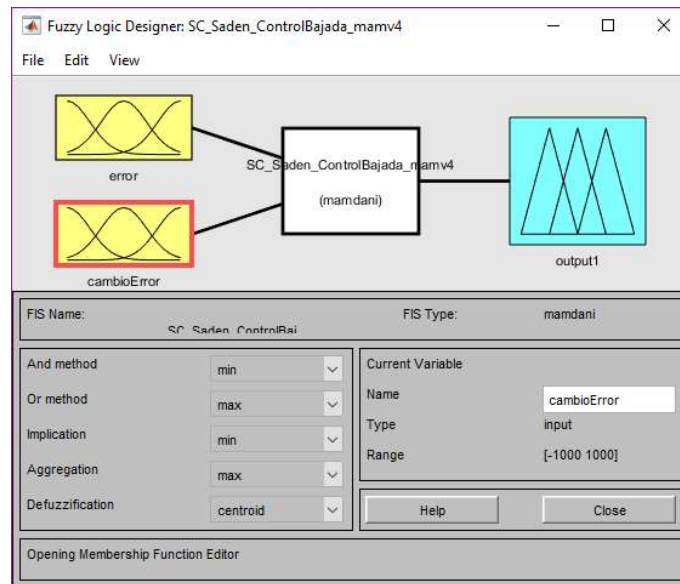


Figura A.3. Configuración de variable de entrada *cambioError*

Paso 4. Se edita la variable entrada *cambioError* y declararan las funciones miembro (Figura A.4)

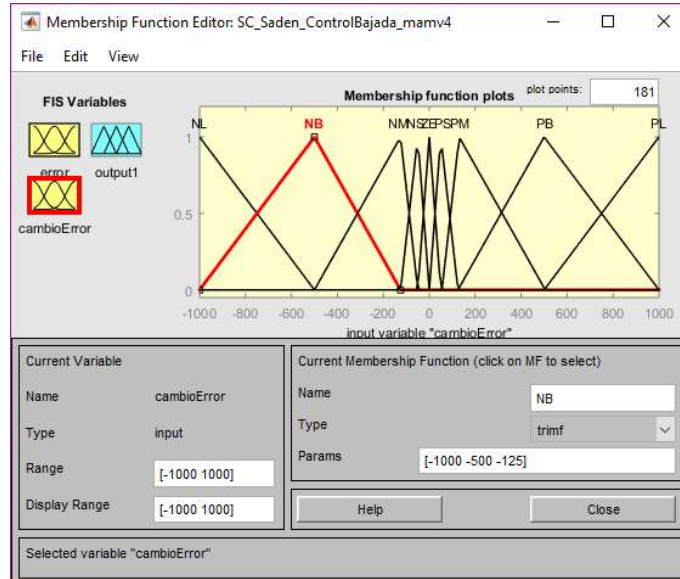


Figura A.4. Declaración de las funciones miembro para la variable de entrada *cambioError*

Paso 5. Se selecciona la variable de salida por defecto y se configura para convertirla en la variable *deltaD* definida en el diseño (Figura A.5).

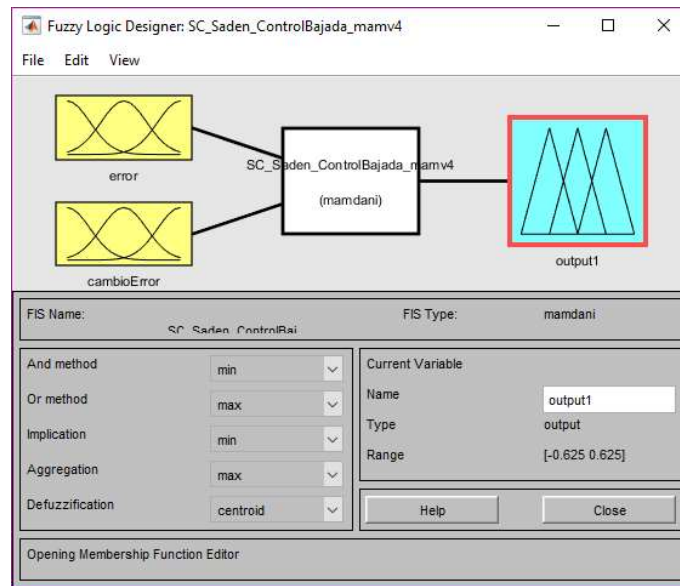


Figura A.5. Configuración de variable de salida *DeltaD*

Paso 6. Se edita la variable salida *DeltaD* y se declararan las funciones miembro (Figura A.6)

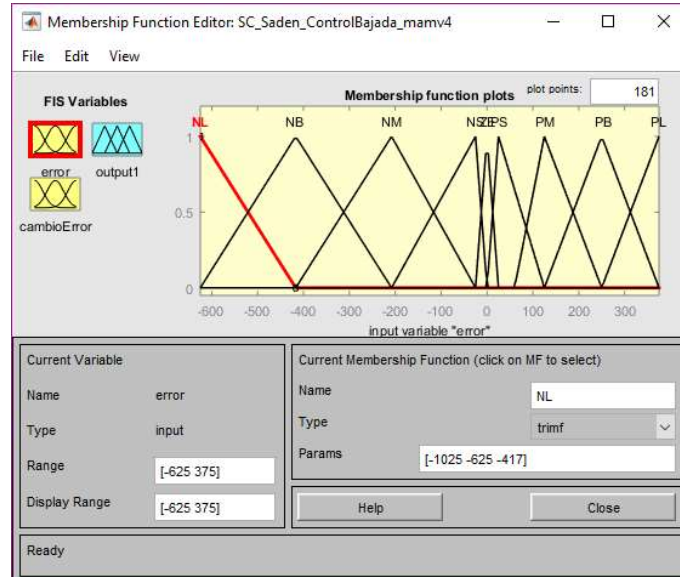


Figura A.6. Declaración de las funciones miembro para la variable de salida *DeltaD*

Paso 7. Se abre el editor de reglas y se definen cada una de las reglas de inferencia como fueron diseñadas. Al combinar 9 funciones miembro por cada una de las variables de entradas, obtenemos en total son 81 reglas, de acuerdo con la Tabla 4. (Figura A.7).

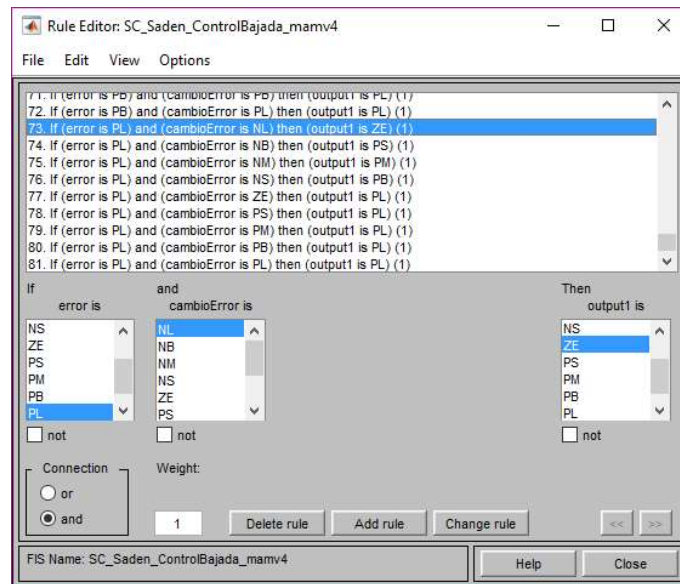


Figura A.7. Declaración de las reglas de inferencia

Paso 8. Es posible probar el funcionamiento del módulo a través del visor de reglas (Figura A.8). Se asignan valores de prueba a las variables de entrada y se visualiza el valor de la variable de salida, de acuerdo con las reglas de inferencia establecidas.

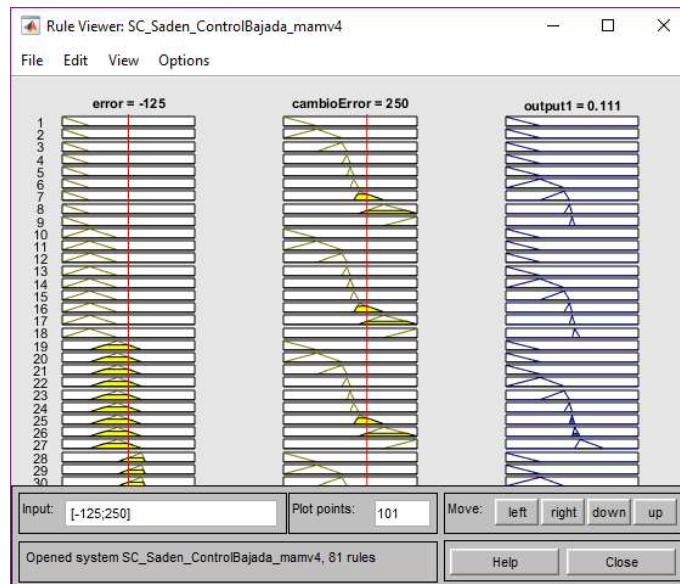


Figura A.8. Visor de reglas

Apéndice B. Implementación del módulo de lógica difusa

La implementación del módulo de control difuso se realizó usando el Fuzzy System Designer de LabVIEW™

A continuación, se describe paso a paso la implementación.

Paso 1. Abir el editor de lógica difusa seleccionando la ruta de menús Tools – Control Design and Simularion – Fuzzy System Designer (Figura B.1).

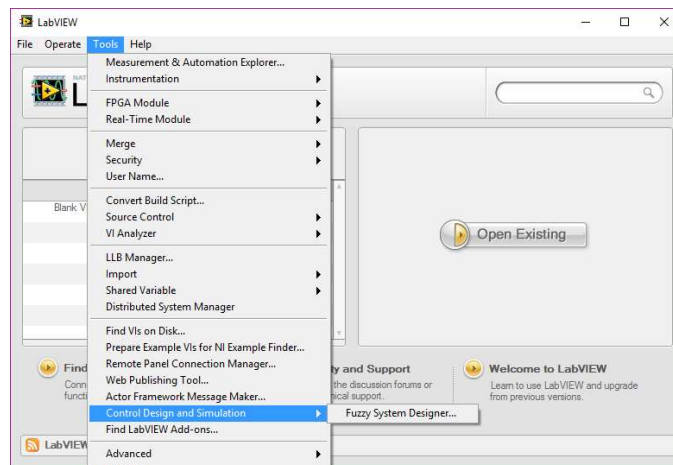


Figura B.1. Apertura del editor de lógica difusa en LabVIEW

Paso 2. Agregar la variable de entrada *Error* utilizando el botón  (Figura B.2).

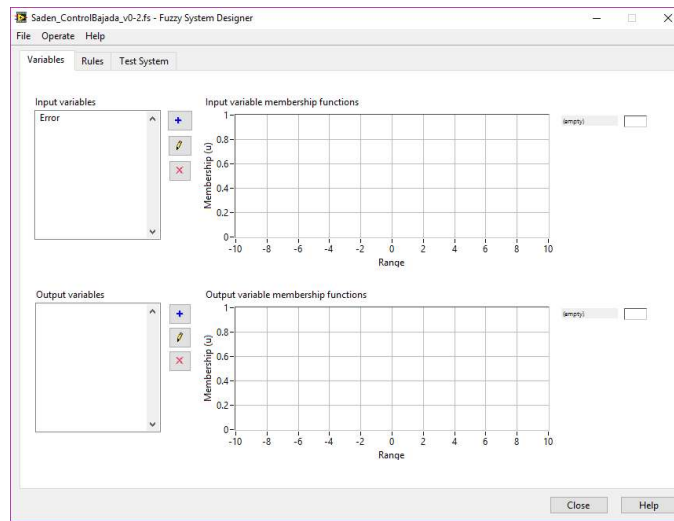


Figura B.2. Alta de la variable *Error*

Paso 3. Editar la variable *Error* (con el botón ) y declarar las nueve funciones de mebresía definidas en el diseño y con los mismos parámetros usados en la simulación (Figura B.3). Al terminar, seleccionar el botón  para terminar la edición y regresar a la ventana anterior.

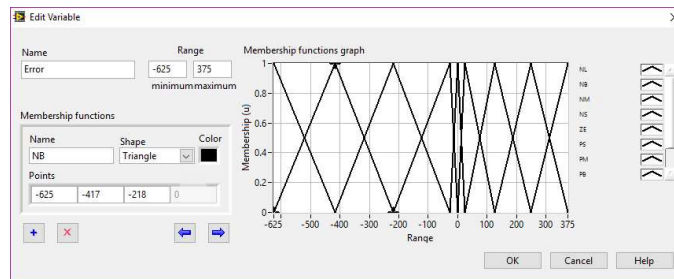


Figura B.3. Declaración de funciones de membresía para la variable de entrada *Error*

Paso 4. Agregar la variable de entrada *CambioError* utilizando el botón  (Figura B.4).

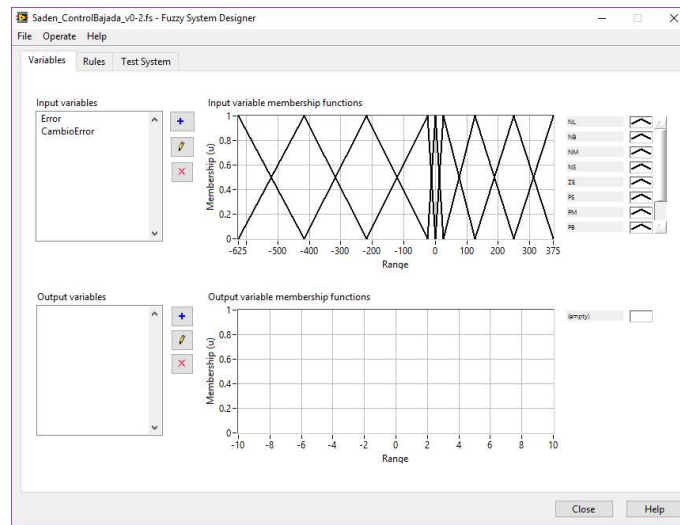


Figura B.4. Alta de la variable *CambioError*

Paso 5. Editar la variable *CambioError* y declarar las nueve funciones de mebresía definidas en el diseño y con los mismos parámetros usados en la simulación (Figura B.5).

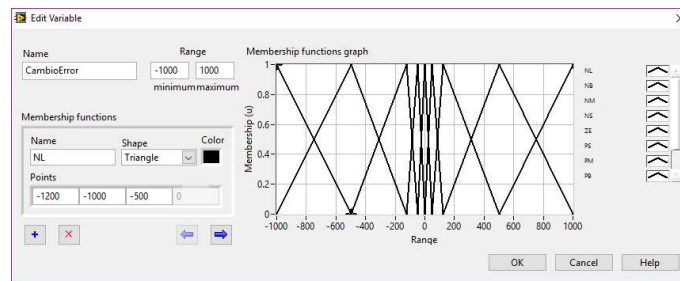


Figura B.5. Declaración de funciones de membresía para la variable de entrada *CambioError*

Paso 6. Agregar la variable de salida *DeltaD* utilizando el botón  (Figura B.6).

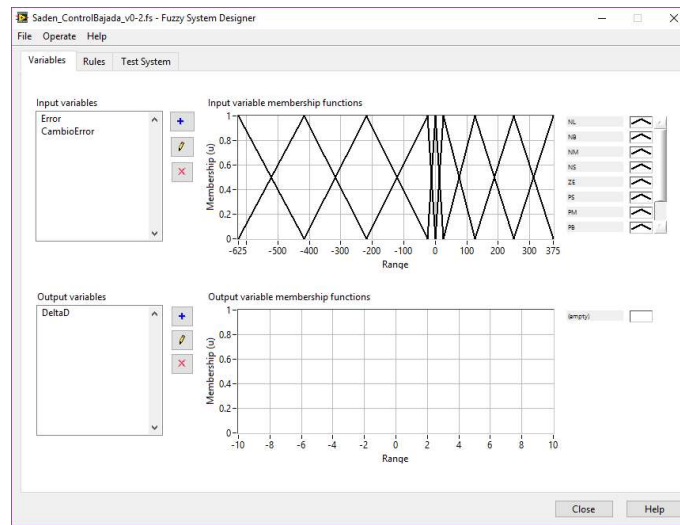


Figura B.6. Alta de la variable *DeltaD*

Paso 7. Editar la variable *DeltaD* y declarar las nueve funciones de mebresía definidas en el diseño y con los mismos parámetros usados en la simulación (Figura B.7).

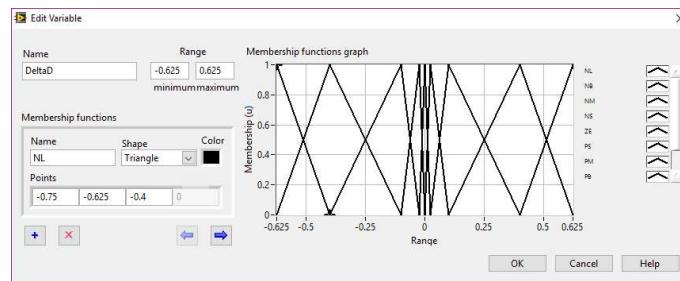


Figura B.7. Declaración de funciones de mebresía para la variable de entrada *DeltaD*.

Paso 8. Seleccionar el folder *Rules* y declarar las 81 reglas de inferencia tal como se diseñaron (Figura B.8).

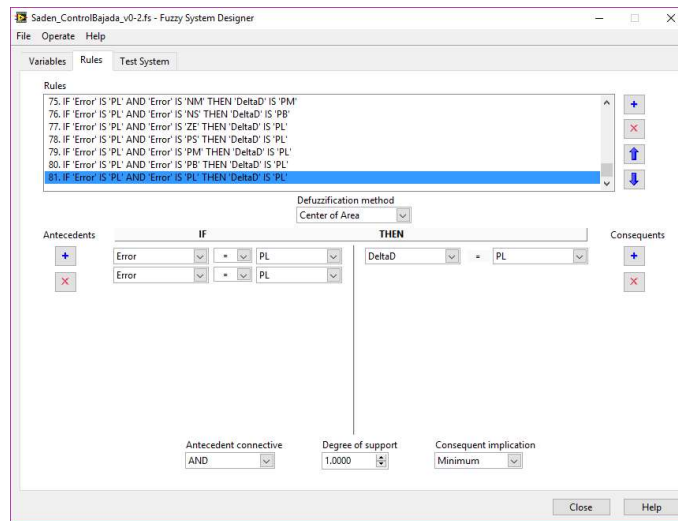


Figura B.8. Declaración de reglas de inferencia

Paso 9. Seleccionar el folder *Test System* para hacer pruebas de funcionamiento del módulo (Figura B.9).

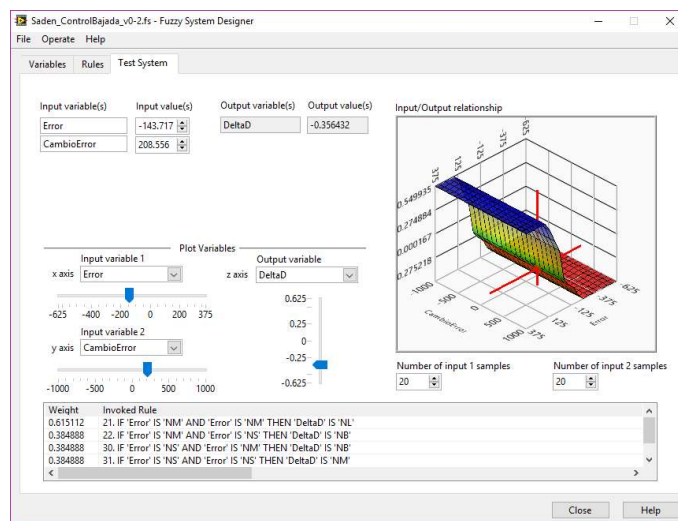


Figura B.9. Pruebas al módulo difuso

Dar valores de prueba a las variables de entrada Error y Cambio de error y el sistema mostrará las reglas invocadas, el valor de la variable de salida DeltaD, calculado con base en estas reglas, y una gráfica de la relación entre entradas y salidas.

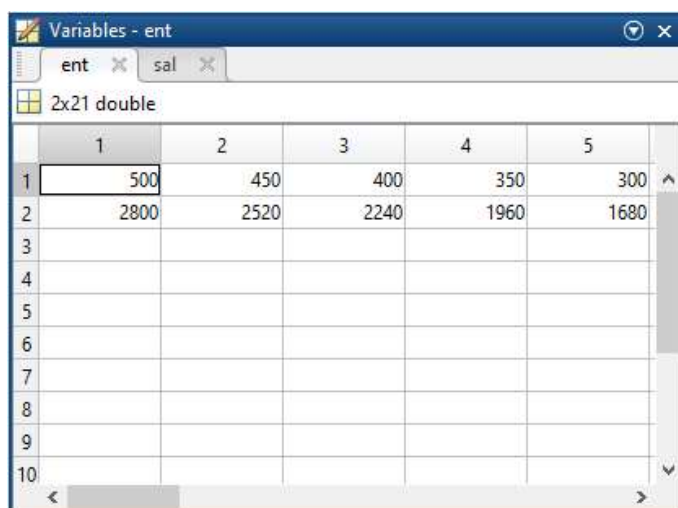
Apéndice C. Simulación de una red neuronal

El módulo de control basado en redes neuronales fue desarrollado usando Matlab®.

A continuación, se describe paso a paso el proceso elaborar la simulación.

Paso 1. Generar los arreglos de valores para las neuronas de entrada y salida (Figuras C.1 y C.2).

En este caso, la variable de entrada *ent* contiene dos conjuntos de valores, uno para cada una de las neuronas de entrada. La variable *sal* solo tiene un conjunto de valores.



	1	2	3	4	5
1	500	450	400	350	300
2	2800	2520	2240	1960	1680
3					
4					
5					
6					
7					
8					
9					
10					

Figura C.1. Detalle de los conjuntos de valores para asignar a las neuronas de entrada

	1	2	3	4	5
1	0.8000	0.7500	0.7000	0.6000	0.4000
2					
3					
4					
5					
6					
7					
8					
9					
10					

Figura C.2. Detalle del conjunto de valores para asignar a la neurona de salida

Paso 2. Crear la red de tipo Feed-Fordward. Para ello, en la ventana de comando escribir

```
red = feedforwardnet
```

Este comando genera una red con características por defecto, que pueden ser visualizadas inmediatamente en la pantalla. Las características son las siguientes:

```
red =
```

```
Neural Network
```

```
name: 'Feed-Forward Neural Network'
userdata: (your custom info)
```

```
dimensions:
```

```
numInputs: 1
numLayers: 2
numOutputs: 1
numInputDelays: 0
numLayerDelays: 0
numFeedbackDelays: 0
numWeightElements: 10
sampleTime: 1
```

```
connections:
```

```
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
```

```
subobjects:
```

```

        input: Equivalent to inputs{1}
        output: Equivalent to outputs{2}

        inputs: {1x1 cell array of 1 input}
        layers: {2x1 cell array of 2 layers}
        outputs: {1x2 cell array of 1 output}
        biases: {2x1 cell array of 2 biases}
        inputWeights: {2x1 cell array of 1 weight}
        layerWeights: {2x2 cell array of 1 weight}

functions:

    adaptFcn: 'adaptwb'
    adaptParam: (none)
    derivFcn: 'defaultderiv'
    divideFcn: 'dividerand'
    divideParam: .trainRatio, .valRatio, .testRatio
    divideMode: 'sample'
    initFcn: 'initlay'
    performFcn: 'mse'
    performParam: .regularization, .normalization
    plotFcns: {'plotperform', plottrainstate, ploterrhist,
               plotregression}
    plotParams: {1x4 cell array of 4 params}
    trainFcn: 'trainlm'
    trainParam: .showWindow, .showCommandLine, .show, .epochs,
               .time, .goal, .min_grad, .max_fail, .mu, .mu_dec,
               .mu_inc, .mu_max

weight and bias values:

    IW: {2x1 cell} containing 1 input weight matrix
    LW: {2x2 cell} containing 1 layer weight matrix
    b: {2x1 cell} containing 2 bias vectors

methods:

    adapt: Learn while in continuous use
    configure: Configure inputs & outputs
    gensim: Generate Simulink model
    init: Initialize weights & biases
    perform: Calculate performance
    sim: Evaluate network outputs given inputs
    train: Train network with examples
    view: View diagram
    unconfigure: Unconfigure inputs & outputs

evaluate:      outputs = red(inputs)

```

Paso 3. Cambiar las opciones por defecto. Para ello escribir el comando

```
red= configure(red,ent,sal);
```

Este comando cambiará el número de entradas a dos, inicializará los pesos y bias con los valores incluidos en las variables *ent* y *sal* y pondrá la red a punto para poder entrenarla.

Utilizando el comando

```
view(net)
```

Es posible visualizar la red en forma de diagrama (Figura C.3), que resalta 2 variables de entrada, 10 neuronas en la capa oculta, 1 neurona en la capa de salida y la variable de salida.

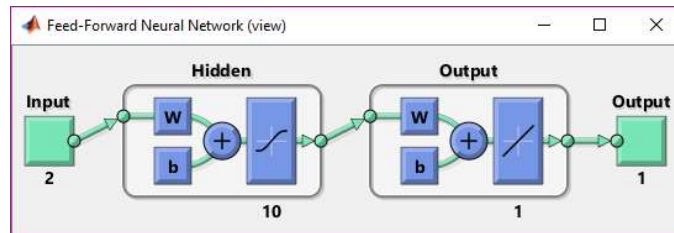


Figura C.3. Diagrama de la red Feed-Forward

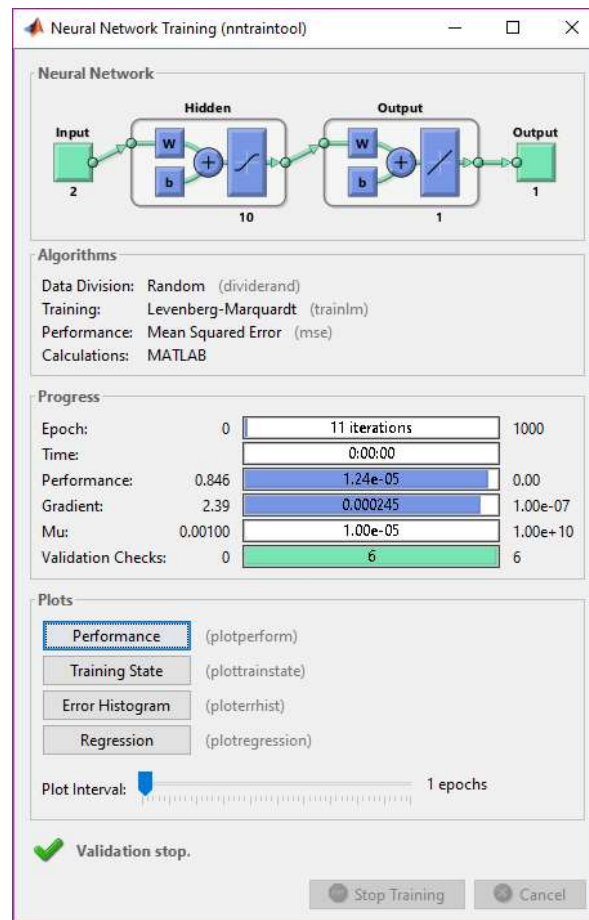


Figura C.4. Interfaz de la herramienta ntraintool

Paso 3. Entrenar la red. Para ello, escribir el comando

```
[red,tr] = train(red,ent,sal);
```

que lanzará el proceso de entrenamiento usando la herramienta nntaintool. La Figura C.4 muestra la interfaz de esta herramienta.

La ventana muestra el progreso del entrenamiento.

Desde esta ventana es posible obtener más información del proceso del entrenamiento a través cuatro gráficas: rendimiento, estado de entrenamiento, histograma de error y regresión.

La gráfica de rendimiento (Figura C.5) muestra el valor de la función de rendimiento frente al número de iteración, además hace una traza el entrenamiento, la validación, y los funcionamientos de la prueba. Se accede a ella a través del botón Performance.

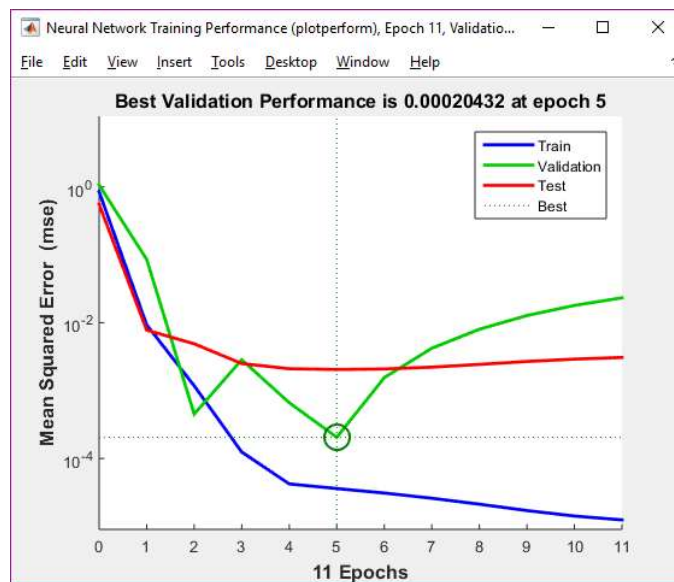


Figura C.5. Gráfica de rendimiento

La gráfica de estado de entrenamiento (Figura C.6) muestra el progreso de otras variables de entrenamiento, como la magnitud del gradiente, el número de verificaciones de validación, etc. Se accede a ella a través del botón Training State.

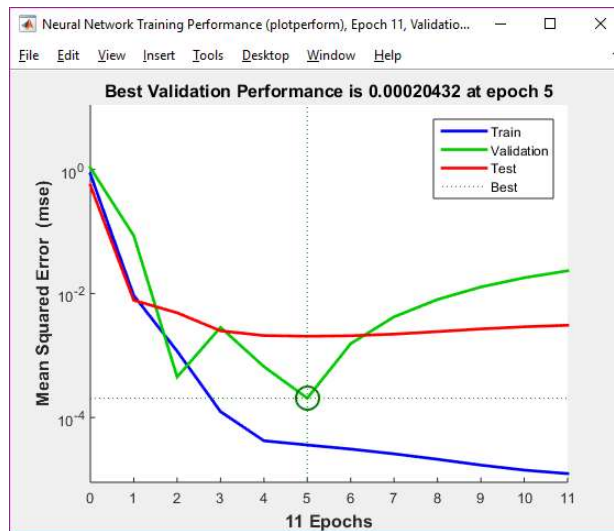


Figura C.6. Gráfica de estado de entrenamiento

La gráfica de histogramas de error (Figura C.7) muestra la distribución de los errores de red. Se accede a ella a través del botón [Error Histogram](#).

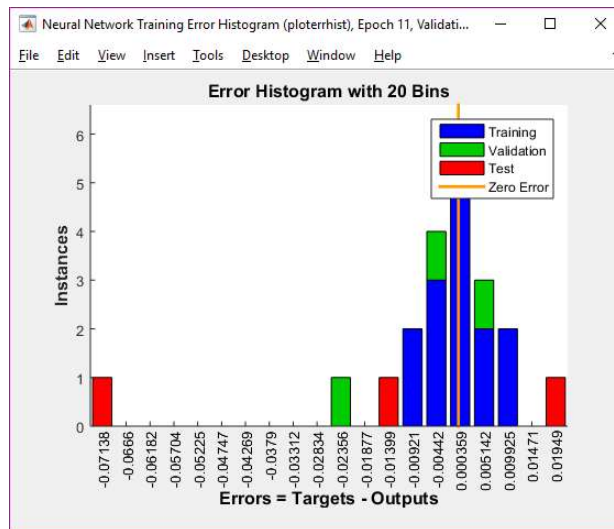


Figura C.7. Histograma de error

La gráfica de regresión (Figura C.8) muestra una regresión entre las salidas de la red y los objetivos de la red. Se accede a ella a través del botón [Regression](#).

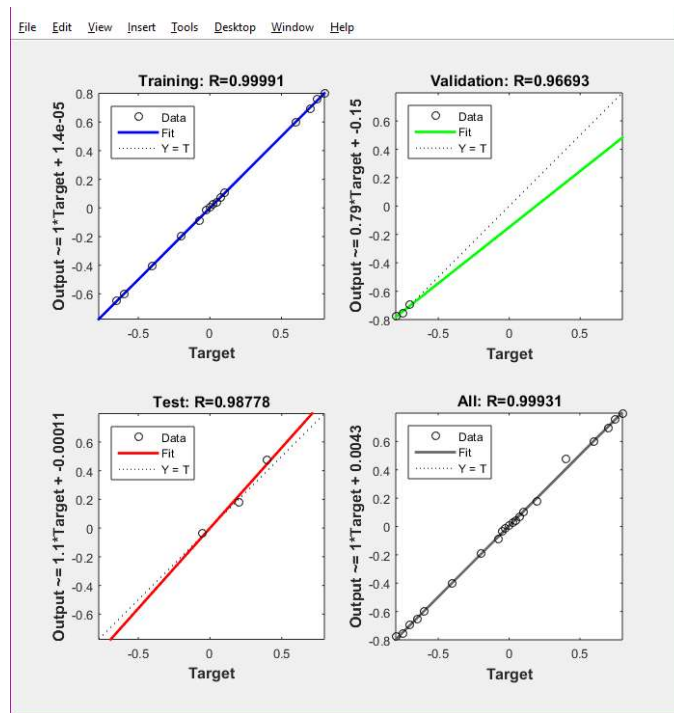


Figura C.8. Gráfica de regresión

Paso 4. Generar la simulación.

Una vez que la red ha sido creada, configurada y entrenada, es necesario convertirla en un bloque para ser incluida como parte de un modelo de simulación. Para ello, en la ventana de comando, escribir

```
gensim(red)
```

Lo cual generara un modelo en Simulink de la red (Figura C.9.).

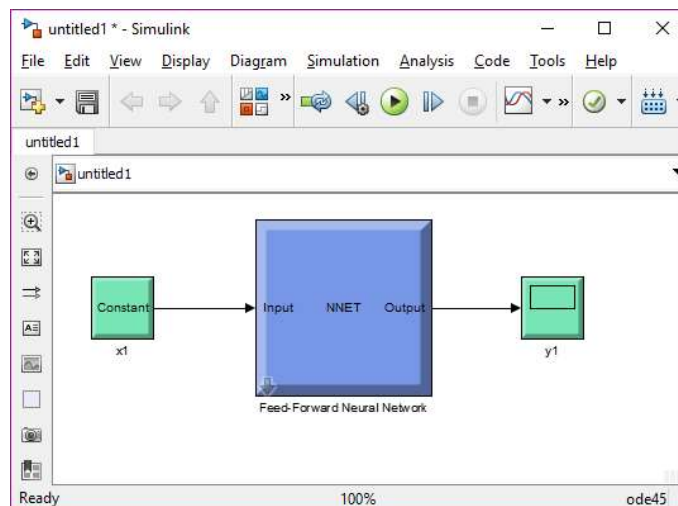


Figura C.9. Simulación de la red de tipo Feed-Forward

El bloque Feed-Forward Neural Network puede ser tomado para conectarlo a un modelo de simulación más amplio.

Referencias

Aguilar, J. (1998). Introducción a las Redes Neuronales. Pontificia Universidad Católica del Ecuador, Facultad de Ingeniería.

Campos, D. (2003). Experimentos en Teoría de Control: Convertidores CD-CD. *Facultad de Ciencias (UASLP)*.

Cardeña-Moreno, S. A. (2016). Diseño de un Convertidor CD/CD Bidireccional de Medio Puente. *Tesis*. Universidad Popular Autónoma del Estado de Puebla.3

Castellanos M., Pedro, Programación del control para un convertidor CD/CD en un DSP, *Reporte interno*

Duarte-Velasco, O. G. (1999). Sistemas de lógica difusa: fundamentos. *Ingeniería e Investigación*, (42), 22-30.

Elmas, C., Deperlioglu, O., & Sayan, H. H. (2009). Adaptive fuzzy logic controller for DC–DC converters. *Expert Systems with Applications*, 36(2), 1540-1548.

Gouse Basha, S., & Kishore, P. D. V. S. K. Implementation and Performance of Fuzzy Logic Controller on DC-DC Boost Converter

Govindaraj, T., & Rasila, R. (2010). Development of fuzzy logic controller for dc–dc buck converters. *International Journal of Engineering Techsci*, 2(2), 192-198.

Grupo Coordinador de Desarrollo Regional del Tecnológico de Monterrey. (2009). Las megatendencias tecnológicas actuales y su impacto en la identificación de oportunidades

estratégicas de negocios. Recuperado de: [http://gestionestrategica.wikispaces.com/file/view/MegatendenciasTecnologicas_y_su_ImpactoOportunidadesEstrategicas\(1\)+copia.pdf](http://gestionestrategica.wikispaces.com/file/view/MegatendenciasTecnologicas_y_su_ImpactoOportunidadesEstrategicas(1)+copia.pdf)

Guide, M. U. S. (2012). Version a, Fuzzy Logic Toolbox. *Math Works*, USA, 235.

Hart, D. W. (2001). Electrónica de potencia. *Pearson Educación, SA*.

Hu, Y. H., & Hwang, J. N. (Eds.). (2001). Handbook of neural network signal processing. CRC press

Huh, S. H., & Park, G. T. (1999). An adaptive fuzzy controller for power converters. *In Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International* (Vol. 1, pp. 434-439). IEEE.

Juárez-Herrero, J. M., Cañadas-Martínez, J. J., Marín-Morales, R. (2008) Control. En: Inteligencia Artificial: Técnicas, Métodos y Aplicaciones. *McGraw Hill*. Pág. 589-646. ISBN 978-84-481-5618-3

Kessal, A., Rahmani, L., Mostefai, M., Gaubert, J., (2012), Power Factor Correction based on Fuzzy Logic Controller with Fixed Switching Frequency, *Electronics & Electrical Engineering*, Issue 118, p 67.

Khaburi, D. A., & Rostami, H. (2011). Controlling the both dc boost and ac output voltages of a z-source inverter using neural network controller with minimization of voltage stress across devices. *Iranian Journal of Electrical & Electronic Engineering*, 7(1).

Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. *Cátedra de Informática Aplicada a la Ingeniería de Procesos–Orientación I*.

Mohan, N., Undelan, T. U., Robbins W. P. (2009). Electrónica de potencia: convertidores, aplicaciones y diseño (no. Tk7881. 15. M64 2009.).

Moreno-Aguilar, L. M., Peralta-Sánchez, E., (2016). Control para un convertidor CD-CD basado en redes neuronales. En: *Avances en las Tecnologías de la Información*. Ed. Alfa-Omega, pp. 429-438, ISBN: 978-607-622-834-0.

Moreno-Aguilar, L. M., Peralta-Sánchez, E., Morales-Salgado, M. R. (2016). Control difuso para un convertidor CD-CD bidireccional de medio puente. *Ingenio Magno*, 7(1), pp.116-132.

Muruganandam, M., Thangaraju, I., & Madheswaran, M. (2014). Simulation and Implementation of an Embedded Hybrid Fuzzy Trained Artificial Neural Network Controller for Different DC Motor. *Muthayammal Engineering College, Rasipuram, Tamilnadu, India*.

National Instruments. (2014). NI LabVIEW for CompactRIO: Developer's Guide.

Propuesta de realización del proyecto "Sistema de Ahorro de Energía para los trenes de la Ciudad de México" (2011). Documento interno.

Rameshkumar, A., & Arumugam, S. (2009). Design and simulation of fuzzy controlled quasi resonant buck converter. *Journal of Engineering & Applied Sciences*, 4(5).

Rashid, M. H. (2010). Power electronics handbook: devices, circuits and applications. *Academic press*.

Rosenblatt, F., (1958) The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, vol. 65, pp. 386–408.

Salam, Z., Taeed, F., & Ayob, S. M. (2011). Design and implementation of a single input fuzzy logic controller for boost converters. *Journal of Power Electronics*, 11(4), 542-550.

Salas, R. (2004). Redes neuronales artificiales. *Universidad de Valparaíso. Departamento de Computación*, 1-15.

Sayed, H. K. (2013). Modeling and Control of a Buck DC-DC Converter Based on Artificial Neural Network. *International Journal of Scientific & Engineering Research*, 4(11), 1224-1233.

So, W. C., Tse, C. K. y Lee, Y. S. (1996). Development of a fuzzy logic controller for DC/DC converters: design, computer simulation, and experimental evaluation. *IEEE Transactions on Power Electronics*, 11(1), 24-32.

Toro, J. (2011). La Innovación Tecnológica en el Sistema de Transporte Colectivo. 3er. *Congreso anual ECORFAN*. Recuperado de: <http://www.ecorfan.org/congresos/3er anual/La%20innovacion%20tecnologica%20en%20el%20sistema%20de%20transporte%20colectivo.pdf>

Utomo, W. M., Haron, Z. A., Bakar, A. A., & Ahmad, M. Z. (2011). Taufik, Voltage tracking of a DC–DC buck-boost converter using neural network control. *Int. J. Comput. Technol. Electron. Eng*, 1(3), 108-113
