

UNIVERSIDAD SANTO TOMÁS

---

SINTONIZACIÓN DE UN CONTROLADOR PID  
MEDIANTE ALGORITMOS BIO-INSPIRADOS PARA LA  
CORRECCIÓN DEL FACTOR DE POTENCIA DE UN  
CONVERTIDOR ELEVADOR

---

Realizado por

David Páez Ramírez  
Juan Pablo Romero Camacho

Un proyecto enviado en cumplimiento del requisito parcial  
para optar por el grado de Ingeniero Electrónico



Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)  
Facultad de Ingeniería Electrónica  
División de Ingenierías

Noviembre de 2021

---

SINTONIZACIÓN DE UN CONTROLADOR PID  
MEDIANTE ALGORITMOS BIO-INSPIRADOS PARA LA  
CORRECCIÓN DEL FACTOR DE POTENCIA DE UN  
CONVERTIDOR ELEVADOR

---

Realizado por

David Páez Ramírez

Juan Pablo Romero Camacho

Un proyecto enviado en cumplimiento del requisito parcial  
para optar por el grado de Ingeniero Electrónico

Dirigido por

José Guillermo Guarnizo Marín

Co-Dirigido por

Jhon Fredy Bayona Navarro

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)  
Facultad de Ingeniería Electrónica  
División de Ingenierías

Noviembre de 2021

---

## DECLARACION DE AUTORÍA

Yo, **DAVID PÁEZ RAMÍREZ** con C.C. 1.072.671.429 y **JUAN PABLO ROMERO CAMACHO** con C.C. 1.019.126.721; declaramos que el proyecto de grado que tiene como nombre **“SINTONIZACIÓN DE UN CONTROLADOR PID MEDIANTE ALGORITMOS BIO-INSPIRADOS PARA LA CORRECCIÓN DEL FACTOR DE POTENCIA DE UN CONVERTIDOR ELEVADOR”**, se ha desarrollado de manera íntegra, respetando derechos intelectuales de las personas que han desarrollado conceptos mediante las citas en las cuales indican la autoría, y cuyos datos se detallan de manera más completa en la bibliografía. En virtud de esta declaración, nos responsabilizamos del contenido, autenticidad y alcance del presente proyecto.

Bogotá, Noviembre de 2021

---

**DAVID PAEZ RAMIREZ**  
C.C. 1.072.671.429

---

**JUAN PABLO ROMERO CAMACHO**  
C.C. 1.019.126.721

---

## RESUMEN

El presente trabajo de grado se basa en el estudio de algoritmos bio-inspirados para la sintonización de controladores aplicados a la eficiencia de convertidores conmutados, para ello se realiza un trabajo de investigación con respecto a diferentes algoritmos aplicados a lo largo del tiempo para finalmente escoger entre ellos el candidato con mejor historial de publicaciones que aplicaciones. de los algoritmos seleccionados si opta por el PSO (Particle Swarm Optimization), los parámetros del controlador son inicializados aleatoriamente en un sector reducido del espacio  $[Kp \ Ki \ Kd \ N]$ , debido a su procesamiento y paradero y su velocidad al buscar una solución en el algoritmo PS o es evolucionado a un multi PC o que permite un sistema de búsqueda múltiple con una mayor tasa de éxito como la simulación del convertidor y el controlador son implementados en MATLAB SIMULINK.

## ABSTRACT

The present degree work is based on the study of bio-inspired algorithms for the tuning of controllers applied to the efficiency of switched converters, for this purpose a research work is carried out with respect to different algorithms applied over time to finally choose among them the candidate with the best publication records which applications. of the selected algorithms if it opts for the PSO (particle swarm optimization), the controller parameters are randomly initialized in a reduced sector of the space  $[Kp \ Ki \ Kd \ N]$ , due to its processing and whereabouts and its speed when searching for a solution in the PS algorithm or is evolved to a multi PC or that allows a multiple search system with a higher success rate as the simulation of the converter and the controller are implemented in MATLAB SIMULINK.

---

## DEDICATORIAS

*A mi madre Luz, mi hermana Elizabeth y mi abuela Cecilia, quienes con su amor y sabiduría me dieron la fuerza para completar las cosas en los momentos difíciles. A mi novia Juliana por su entusiasmo, amor y fortaleza para recordarme la persona que soy y lo que debía hacer. A mi leal y apreciado amigo Alexander por su apoyo y motivación a lo largo de tanto tiempo. A Juan Pablo y Brian David por su dedicación y apoyo en cada trabajo de estos últimos años. A nuestros tutores José Guarnizo y Jhon Bayona, y en especial al profesor Juan Manuel Calderón, pues sin su ayuda y conocimiento no habría sido posible realizar este proyecto. Y a todos mis compañeros y amigos a lo largo del tiempo que me brindaron una mano sin esperar nada a cambio.*

*David Paez Ramirez*

*Agradezco a Dios, mis padres María Nelsy Camacho y Jorge Arturo Romero (Q.E.P.D), Mis Abuelos María Julia Reyes de Romero, Juan B. Romero y Felisa Leyton de Camacho (Q.E.P.D) por apoyarme en todo momento y confiar en mí. A mi querida novia María Alejandra Barrios por ser parte de mi vida y para el desarrollo exitoso de este proyecto. También a mis mejores amigos por siempre formar un gran equipo y estar allí cuando más nos necesitamos. A los directores José Guarnizo y Jhon Bayona, y en especial al profesor Juan Manuel Calderón, por guiarnos y su disposición en todo momento. Finalmente, a mi compañero Brian Noriega y sobre todo a mi gran amigo David Páez por su trabajo para lograr culminar este proyecto.*

*Juan Pablo Romero Camacho*

---

# Contenido

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Lista de Figuras .....                                                     | 7  |
| Introducción.....                                                          | 9  |
| Planteamiento del Problema .....                                           | 10 |
| Justificación .....                                                        | 11 |
| Impacto Social .....                                                       | 12 |
| Objetivos .....                                                            | 13 |
| 5.1. Objetivo General .....                                                | 13 |
| 5.2. Objetivos Específicos.....                                            | 13 |
| Fundamentos Teóricos.....                                                  | 14 |
| 6.1. Generalidades de Circuitos Eléctricos .....                           | 14 |
| 6.1.1. Valor Eficaz o RMS.....                                             | 14 |
| 6.1.2. Potencia.....                                                       | 14 |
| 6.1.3. Rectificador Monofásico de Puente Completo .....                    | 16 |
| 6.1.4. Convertidor de Voltaje DC .....                                     | 18 |
| 6.1.4.1. Modo de Conducción Continua .....                                 | 19 |
| 6.1.4.2. Modo de Conducción Discontinua .....                              | 22 |
| 6.2 Consideraciones del PSO.....                                           | 23 |
| 6.3 Controlador PID .....                                                  | 24 |
| 6.4. Algoritmos Bio-inspirados.....                                        | 27 |
| 6.4.1. Basados en Biología .....                                           | 27 |
| 6.4.1.1. Basados en Evolución.....                                         | 27 |
| 6.4.1.2. Basados en Comportamiento de Enjambre .....                       | 27 |
| 6.5.2.1. Basados en Leyes Físicas .....                                    | 28 |
| Estado del Arte .....                                                      | 29 |
| 7.1. Corrección del Factor de Potencia .....                               | 29 |
| 7.2. Sintonización de Controladores.....                                   | 29 |
| 7.2.1. Sintonización de acuerdo con criterios empíricos o analíticos ..... | 29 |
| 7.2.2. Sintonización por métodos de inteligencia artificial .....          | 30 |
| Metodología .....                                                          | 32 |
| 8.1 Diagrama Metodológico.....                                             | 32 |

---

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 8.1.1. Diagrama de Flujo .....                                     | 32 |
| 8.1.2. Análisis e Investigación.....                               | 33 |
| 8.1.3. Diseño del Convertidor .....                                | 33 |
| 8.1.4. Desarrollo del algoritmo.....                               | 33 |
| 8.1.5. Simulación del modelo obtenido .....                        | 33 |
| 8.1.6. Ajustes Pertinentes.....                                    | 34 |
| 8.1.7. Implementación: Convertidor + Controlador .....             | 34 |
| 8.1.8. Comparación de Resultados.....                              | 34 |
| Diseño .....                                                       | 35 |
| 9.1. Estrategia de Control Inteligente.....                        | 35 |
| 9.1.1. Análisis y Determinación de Objetivo .....                  | 35 |
| 9.1.2. Selección Estrategia .....                                  | 35 |
| 9.1.3. Algoritmo Implementado.....                                 | 36 |
| 9.1.3.1. Descripción general del algoritmo del PSO .....           | 36 |
| 9.1.3.2. Variación de las constantes de aprendizaje .....          | 37 |
| 9.2. Convertidor AC-DC con corrección del factor de potencia.....  | 38 |
| 9.3. Implementación en Simulación SIMULINK MATLAB.....             | 39 |
| 9.4. Diseño e implementación física del circuito.....              | 41 |
| 9.4. 1. Diseño de la carga.....                                    | 41 |
| 9.4.2. Diseño del circuito esquemático. ....                       | 41 |
| 9.4.3. Consideraciones de Diseño del circuito impreso (PCB) .....  | 42 |
| 9.4.3.2. Listado de materiales .....                               | 43 |
| Resultados .....                                                   | 45 |
| 10.1. Simulaciones .....                                           | 45 |
| 10.1.1. Controlador Sintonizado por Métodos Comunes.....           | 45 |
| 10.1.1. Controlador Sintonizado por Algoritmo Bio- inspirado ..... | 49 |
| 10.2. Resultados del Montaje Real.....                             | 56 |
| 10.2.1. Controlador Sintonizado por métodos comunes .....          | 56 |
| 10.2.2. Controlador Sintonizado por Algoritmo Bio-inspirado .....  | 57 |
| Conclusiones.....                                                  | 60 |
| Trabajos Futuros.....                                              | 61 |
| Anexos.....                                                        | 62 |
| Bibliografía.....                                                  | 80 |

---

# Lista de Tablas

|                                                                  |    |
|------------------------------------------------------------------|----|
| TABLA 1: PARÁMETROS DEL MODELO. FUENTE: ELABORACIÓN PROPIA.....  | 38 |
| TABLA 2: LISTADO DE MATERIALES. FUENTE: ELABORACIÓN PROPIA. .... | 43 |
| TABLA 3: CONFIGURACIÓN ENTRENAMIENTO .....                       | 49 |
| TABLA 4, DATOS OBTENIDOS DEL PF CON ANALIZADOR DE REDES .....    | 58 |



---

# Lista de Figuras

|                                                                                                                                                                                                                    |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| FIGURA 1 : VALOR RMS, PICO Y PICO DE UNA SEÑAL DE VOLTAJE (2021). LEARNING ABOUT ELECTRONIC.....                                                                                                                   | 14 |
| FIGURA 2: RECTIFICADOR MONOFÁSICO DE PUENTE COMPLETO. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                              | 17 |
| FIGURA 3: CONVERTIDOR DE VOLTAJE EN SU TOPOLOGÍA ELEVADOR. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                         | 19 |
| FIGURA 4: REPRESENTACIÓN ESQUEMÁTICA DE TOFF EN EL CONVERTIDOR. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                    | 19 |
| FIGURA 5: REPRESENTACIÓN ESQUEMÁTICA DE TON EN EL CONVERTIDOR. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                     | 20 |
| FIGURA 6: FORMA DE ONDA DEL CONVERTIDOR. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                                           | 21 |
| FIGURA 7: FORMAS DE ONDA DEL CONVERTIDOR EN MODO DISCONTINUO. FUENTE: [12]. .                                                                                                                                      | 23 |
| FIGURA 8: DIAGRAMA DE BLOQUES DEL CONTROLADOR. FUENTE: ELABORACIÓN PROPIA.                                                                                                                                         | 24 |
| FIGURA 9: DIAGRAMA DE FLUJO DE ACTIVIDADES REALIZADAS A LO LARGO DEL PROYECTO - ACTIVIDADES DESCRITAS DE FORMA CRONOLÓGICA DE ACUERDO CON LOS OBJETIVOS ESPECÍFICOS DEL PROYECTO. FUENTE: ELABORACIÓN PROPIA. .... | 32 |
| FIGURA 10: MOVIMIENTO DE PARTÍCULAS PSO EN UN ESPACIO DE BÚSQUEDA MULTIDIMENSIONAL.....                                                                                                                            | 37 |
| FIGURA 11: CONVERTIDOR ELEVADOR (TOPOLOGÍA ELEVADOR). FUENTE: ELABORACIÓN PROPIA.....                                                                                                                              | 39 |
| FIGURA 12: CONVERTIDOR ELEVADOR CON CARGAS AL 100 %/80 %/60 %/50 % Y 20 %. FUENTE: ELABORACIÓN PROPIA. ....                                                                                                        | 40 |
| FIGURA 13: CARGA DE 500 - 2500 A 100W. FUENTE: ELABORACIÓN PROPIA. ....                                                                                                                                            | 41 |
| FIGURA 14: DIAGRAMA ESQUEMATIC DEL CIRCUITO IMPRESO (PARA MÁS DETALLE VER EN LOS ANEXOS). FUENTE: ELABORACIÓN PROPIA.....                                                                                          | 42 |
| FIGURA 15: CIRCUITO IMPRESO, CAPA SUPERIOR, CARA INFERIOR. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                         | 44 |
| FIGURA 16: IMPLEMENTACIÓN REAL DE LA PLANTA Y SISTEMA DE CONTROL. FUENTE: ELABORACIÓN PROPIA.....                                                                                                                  | 44 |
| FIGURA : VOLTAJE DE ENTRADA Y CORRIENTE DE ENTRADA .....                                                                                                                                                           | 46 |
| FIGURA 18: VOLTAJE RECTIFICADO Y CORRIENTE RECTIFICADA O CORRIENTE EN EL INDUCTOR.....                                                                                                                             | 47 |
| FIGURA : VOLTAJE DE SALIDA Y CORRIENTE DE SALIDA .....                                                                                                                                                             | 48 |
| FIGURA 20: POTENCIA DE SALIDA.....                                                                                                                                                                                 | 48 |
| FIGURA 21: FACTOR DE POTENCIA .....                                                                                                                                                                                | 49 |
| FIGURA 22: FACTOR DE POTENCIA .....                                                                                                                                                                                | 50 |
| FIGURA 23: CONTROLADOR DE REFERENCIA Y CONTROLADOR MPSO .....                                                                                                                                                      | 50 |
| FIGURA 24: VOLTAJE RECTIFICADO Y CORRIENTE RECTIFICADA O CORRIENTE EN EL INDUCTOR.....                                                                                                                             | 51 |
| FIGURA 25: ENTRENAMIENTOS EN VARIACIONES DE CARGA.....                                                                                                                                                             | 51 |
| FIGURA 26: VOLTAJE DE ENTRADA Y CORRIENTE DE ENTRADA .....                                                                                                                                                         | 52 |
| FIGURA 27: VOLTAJE DE ENTRADA Y CORRIENTE DE ENTRADA .....                                                                                                                                                         | 53 |
| FIGURA 28: VOLTAJE DE SALIDA Y CORRIENTE DE SALIDA .....                                                                                                                                                           | 54 |
| FIGURA 29: POTENCIA DE SALIDA.....                                                                                                                                                                                 | 54 |
| FIGURA 30: FACTOR DE POTENCIA .....                                                                                                                                                                                | 55 |
| FIGURA 31: COMPARACIÓN FACTOR DE POTENCIA .....                                                                                                                                                                    | 55 |

---

|                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------|----|
| FIGURA 32: FORMAS DE ONDA DE VOLTAJE Y CORRIENTE DE ENTRADA CONTROLADOR<br>CLÁSICO. FUENTE: ELABORACIÓN PROPIA. .... | 56 |
| FIGURA 33: DESFACE DE CORRIENTE Y VOLTAJE 3° .....                                                                   | 56 |
| FIGURA 34: FORMAS DE ONDA DE VOLTAJE Y CORRIENTE DE ENTRADA CONTROLADOR<br>CLÁSICO. FUENTE: ELABORACIÓN PROPIA. .... | 57 |
| FIGURA 35: DESFACE DE CORRIENTE Y VOLTAJE 1° .....                                                                   | 57 |
| FIGURA 36: COMPARACIÓN FACTOR DE POTENCIA, IMPLEMENTACIÓN REAL. FUENTE:<br>ELABORACIÓN PROPIA. ....                  | 58 |

---

# Capítulo 1

## Introducción

Al hablar de controladores **PID** se puede decir que gracias a sus diversas ventajas son muy usados en la industria, de hecho, existe una rama de la ingeniería la cual se dedica al estudio de controladores y sus diferentes métodos de sintonización. Todo esto con la intención de poder aplicar toda esta teoría en el desarrollo de procesos industriales ya sea para automatizar procesos complejos o controlar variables de interés. La gran popularidad de los controladores **PID** se da por su excelente rendimiento con algunas condiciones de funcionamiento y la simplicidad a la hora de su implementación, es por esto que a medida que se ha estudiado la teoría de control **PID** se han propuesto diferentes métodos que permiten su sintonización. Entre algunos métodos convencionales conocidos se puede mencionar desde el método de Ziegler-Nichols [1], hasta algunos más elaborados como los métodos de auto-tune que traen algunos softwares de desarrollo [2], [3].

Los métodos convencionales poseen gran variedad de estudios los cuales permiten ser una excelente opción de sintonización a la hora de ser implementados y representado por modelos de sistemas dinámicos, sin embargo, existe la posibilidad de enfrentarse con inconvenientes relacionados con su sintonización. Algunos de los problemas más comunes pueden ser un tiempo de establecimiento largo o un gran overshoot, además, también existe la posibilidad de enfrentarse a sistemas demasiado complejos en donde un método convencional no pueda ser usado con certeza y es ahí en donde se ve la necesidad de explorar otros métodos de sintonización para el controlador **PID**. En los últimos años se han hecho estudios en donde se exploran alternativas de sintonización haciendo uso de la inteligencia artificial, de hecho, algunos algoritmos muestran resultados prometedores, argumentando así, que es posible realizar una sintonización de un controlador **PID** haciendo uso de ellos.

La inteligencia artificial es una disciplina cuyas aplicaciones más comunes son el tratamiento de datos e identificación de sistemas, los cuales son implementados en algoritmos [4]. Algunos de estos algoritmos están basados en el comportamiento de manadas de animales los cuales trabajan en conjunto, estos son conocidos como algoritmos bio-inspirados. Existen gran variedad de algoritmos, pero en este trabajo se explora el algoritmo de Particle Swarm Optimization (**PSO**), el cual es uno de los algoritmos de procesamiento paralelo más popular, desarrollado por James Kennedy y Russell Eberhard en 1995. Este algoritmo de **PSO** se ha usado para resolver problemas prácticos de optimización, y también problemas de la ciencia y la ingeniería.

---

## Capítulo 2

# Planteamiento del Problema

La corrección del factor de potencia es un elemento el cual es necesario, no solo en todas aquellas industrias (Textil, Minera, Agrícola, Alimenticia, entre otras...) las cuales dependan de un consumo energético para su funcionamiento, si, no también se debe tener presente en los hogares sobre todo aquellos los cuales tengan conexión a una red eléctrica, un bajo factor de potencia, presenta una gran cantidad de problemas, tanto para el usuario como para la empresa prestadora del servicio eléctrico, entre los diferentes problemas que pueden aparecer, se encuentran; pérdidas por disipación en los elementos, lo cual aumenta su temperatura y con ello puede reducir su vida útil, elevadas caídas de tensión y baja regulación de voltaje, lo cual puede afectar la estabilidad de la red eléctrica e incremento de la facturación eléctrica (resolución CREG 015 de 2018) [5], [6].

Los convertidores conmutados generalmente poseen sistemas de control para la corrección del factor de potencia, esto es debido a que son ampliamente utilizados en los sistemas eléctricos para regulación y distribución de energía, el tipo de controlador más comúnmente utilizado es el controlador PID, sintonizado por métodos empíricos o analíticos como lo es el método de Ziegler Nichols [5], el uso de métodos empíricos para la determinación de los parámetros de sintonización de controladores, generalmente se obtienen mediante experimentación de prueba y error, fórmulas que se han mejorado y han dado resultado a métodos basados en este tipo de análisis. Sin embargo, existen problemáticas en las que el control convencional no puede llegar a una solución aceptable [7].

El uso de los controladores sintonizados de formas convencionales (métodos empíricos o analíticos) en sistemas acoplados, no lineales o inciertos, poseen cuatro deficiencias notables, las cuales son; error de cálculo, degradación del ruido en el control derivativo, sobre simplificación, pérdida de rendimiento en la ley de control en forma de suma ponderada lineal y algunas dificultades también provenientes del control integral [8].

Basado en la investigación y en las diferentes problemáticas existentes en la sintonización de controladores PID por criterios convencionales, se formula la siguiente pregunta; ¿Qué ventajas son notables en la sintonización de controladores PID mediante criterios de inteligencia artificial para la corrección del factor de potencia, con respecto a los controladores sintonizados por criterios empíricos?

---

## Capítulo 3

# Justificación

El factor de potencia es determinante en la eficiencia de los elementos, los sistemas de control sintonizados de formas empíricas pueden sufrir perturbaciones externas como desgastes, que harán que la corrección del factor de potencia sea menos efectiva, sin embargo, con controladores sintonizados mediante técnicas de inteligencia artificial, el controlador se adapta a las condiciones en las que se encuentre el sistema, además de poseer la capacidad de encontrar con mayor facilidad los parámetros de sintonización para respuestas más rápidas manteniendo la estabilidad.

El control de procesos en la industria es un elemento que se va enriqueciendo cada vez más, la cantidad de elementos que puede llegar a poseer un sistema para dar solución a una problemática, hacen que la sintonización de su controlador pueda llegar a ser extensa, por lo que los sistemas de control inteligente son una opción muy llamativa la cual puede ser viable de implementar. Los métodos de sintonización de controladores PID empíricos son implementados como una solución práctica, pero no pueden dar una solución efectiva a sistemas inciertos o de caja negra, acoplados y no lineales [8], de esta forma se puede ver que, en la industria, los algoritmos computacionales para el procesamiento de grandes cantidades de información son una opción viable, debido precisamente a la versatilidad de estos para manejar la información de forma práctica.

El algoritmo bio-inspirado a desarrollar en el trabajo, presenta la posibilidad de adaptarse a diferentes tipos de sistemas dinámicos, pues la forma de funcionamiento de este busca la disminución del error, de esta forma con los parámetros correctos, el controlador puede ser aplicado a diferentes sistemas de control.

---

## Capítulo 4

# Impacto Social

La eficiencia en sistemas de potencia es fundamental. El uso correcto de la energía beneficia a la humanidad desde diferentes puntos de vista ya sea económico, tecnológico, ambiental o social. La corrección del factor de potencia trae consigo numerosas ventajas a la hora de desarrollar algún tipo de actividad industrial o doméstica que requiera el uso, y la transformación de energía, por ejemplo, la mejora en el nivel de tensión de salida, reducción de las pérdidas en los elementos, aumento de esperanza de vida de los elementos, reducción de costos de energía, entre otras.

Para el empresario minorista o pequeño, familiarizado con la industria de energía, la aplicación de sistemas de control inteligente para la corrección del factor de potencia es una excelente optativa para un alto rendimiento de los componentes, algo que no solo se verá reflejado en la calidad del trabajo, también a nivel económico en ahorros de pérdidas por disipación, también en forma ambiental, pues una alta eficiencia además de reducir las pérdidas como ya se dijo, reduce el consumo, ya que para realizar el mismo trabajo necesita menor energía.

Las comunidades que se encuentran de alguna forma vulnerables, debido a su lejanía con los principales sistemas de distribución eléctrica, tienen una prioridad en cuanto al manejo de la energía alternativa que utilizan en elementos eléctricos del hogar, al no tener un sistema fijo de distribución eléctrica, el rendimiento de la energía es un elemento prioritario, mejorar el sistema de adquisición y distribución mediante sistemas de control inteligente es una opción practica a la vez que efectiva.

---

## Capítulo 5

# Objetivos

### 5.1. Objetivo General

Implementar un algoritmo basado en sistemas bio-inspirados con el fin de sintonizar de forma automática el controlador **PID** de un convertidor **AC/DC** con corrección de factor de potencia (**FP**), en topología elevador para obtener un **FP** superior a 0.93.

### 5.2. Objetivos Específicos

- Diseñar un convertidor **AC/DC** con corrección del factor de potencia en topología elevador a partir de su estado estacionario. Los parámetros de este convertidor son: voltaje de salida directo 220 V, voltaje de entrada alterno 120 V y potencia 100 W.
- Adaptar un algoritmo basado en sistemas Bio-Inspirados para la sintonización de un controlador **PID** a partir del error cuadrático medio, aplicado al convertidor **AC/DC** con corrección del factor de potencia, con el propósito de determinar los parámetros del controlador.
- Implementar en simulación el convertidor **AC/DC** con corrección del factor de potencia en topología elevador mediante el software de **MATLAB & SIMULINK** con fines de simular el modelo de sintonización obtenido para realizar ajustes pertinentes.
- Implementar el controlador **PID** sintonizado por algoritmos Bio - inspirados en el convertidor **AC/DC** con corrección del factor de potencia en topología elevador.
- Comparar los resultados obtenidos por el controlador sintonizado con una técnica de control clásica **PID** para el mismo convertidor reportado en la literatura.

---

## Capítulo 6

# Fundamentos Teóricos

### 6.1. Generalidades de Circuitos Eléctricos

#### 6.1.1. Valor Eficaz o RMS

Es un método para denotar una forma de onda senoidal de voltaje (forma de onda de *AC*) como un voltaje equivalente que representa el valor de voltaje *DC* que producirá el mismo efecto de disipación de potencia en el circuito, como esta tensión de *AC*. En otras palabras, la forma de onda es una forma de onda *AC*, pero el valor RMS permite que esta forma de onda se especifique como *DC*, porque es la tensión *DC* equivalente que entrega la misma cantidad de energía a una carga en un circuito como la señal *AC* hace sobre su ciclo [9].

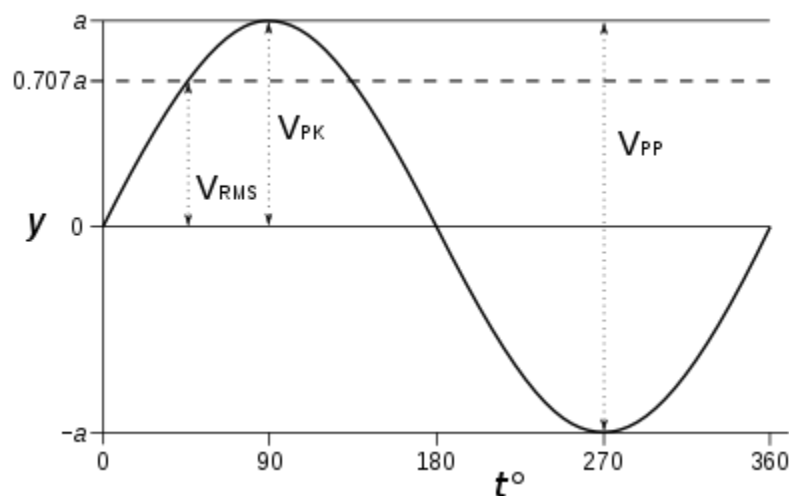


Figura 1 : Valor RMS, pico y pico de una señal de voltaje (2021). Learning About electronic.

En la Figura 1, se observa una señal de voltaje sinusoidal de un ciclo, en donde está la representación gráfica de los conceptos mencionados anteriormente.

#### 6.1.2. Potencia



---

Es la capacidad de producir o demandar energía de una máquina eléctrica, equipo o instalación por unidad de tiempo. En todo circuito eléctrico, para el funcionamiento de los diferentes equipos y máquinas se encuentran presentes las siguientes potencias [10]:

- Potencia Aparente
- Potencia Activa
- Potencia Reactiva

#### 6.1.2.1. Potencia Aparente

La potencia total o aparente es el producto de la corriente y el voltaje. Se la representa con la letra **S** y su unidad de medida se expresa en voltamperios (VA) [10].

Se calcula como:

$$S = \sqrt{P^2 + Q^2} \quad (1)$$

#### 6.1.2.2. Potencia Activa

Es la potencia que en el proceso de transformación de la energía eléctrica se aprovecha como trabajo, es decir, en energía utilizable. Se la representa con la letra **P** y es expresada en vatios (W) [10].

Se calcula como:

$$P = V \times I \times \cos \varphi \quad (2)$$

Siendo  $\varphi$ :

$$\varphi = \arctg \frac{X}{R} \quad (3)$$

Donde **X** es la reactancia y **R** es la resistencia de la carga conectada siendo la impedancia:

$$\vec{Z} = R + jX \quad (4)$$

Esta potencia también puede calcularse como:

$$P = R \times I^2 = \frac{V^2}{R} \quad (5)$$

---

#### 6.1.2.3. Potencia Reactiva

Se simboliza con la letra **Q** expresada en voltamperios reactivos (**VAR**), resulta necesaria para el funcionamiento de ciertas máquinas y dispositivos eléctricos (motores, transformadores, bobinas, relés, etc.) pero no puede transformarse en potencia mecánica o calorífica útil, y causa pérdidas adicionales en los equipos que transportan la energía [10].

Se calcula como:

$$Q = V \times I \times \sin \varphi \quad (6)$$

También puede obtenerse como:

$$Q = X \times I^2 = \frac{V^2}{X} \quad (7)$$

#### 6.1.2.4. FACTOR DE POTENCIA

Es la relación entre la potencia activa y la potencia aparente, y describe como la relación entre la potencia de trabajo o real y la potencia total consumida [7]. Este está definido por la siguiente ecuación:

$$\cos \varphi = \frac{P}{S} \quad (8)$$

Este expresa en términos generales, el desfase de la corriente con relación al voltaje y es utilizado como indicador del correcto aprovechamiento de la energía eléctrica, el cual puede tomar valores entre 0 y 1.0, siendo este último el valor máximo de Factor de Potencia y por tanto, indica el mejor aprovechamiento de energía.

#### 6.1.3. Rectificador Monofásico de Puente Completo

Es el circuito observado en la Figura 2 que contiene un conjunto de cuatro diodos (**D1**, **D2**, **D3**, **D4**) conectados de tal forma que, al ser alimentados con una fuente alterna (**Ve:Vs**), dos de los diodos conducen en un semi-ciclo positivo mientras que el otro par conduce en el semi-ciclo negativo, produciendo así dos señales desfasadas, obteniendo así una onda completa rectificada en la carga (**R**)[11].

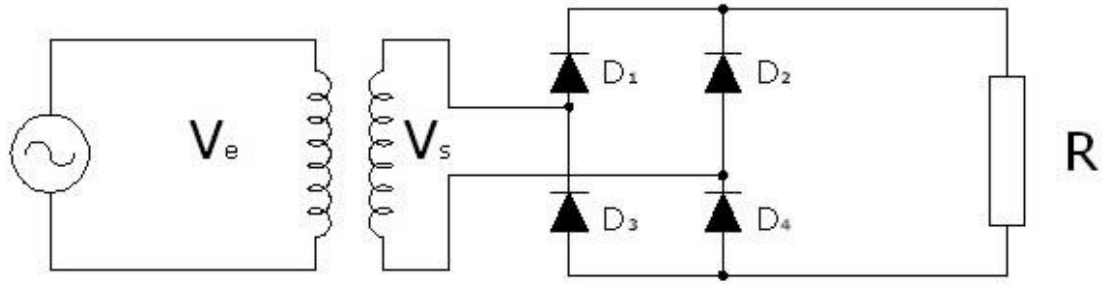


FIGURA 2: Rectificador Monofásico de Puente Completo. Fuente: Elaboración propia.

Para este rectificador se tienen las siguientes fórmulas para calcular los valores de sus parámetros de rendimiento:

- Voltaje DC:

$$V_{dc} = \frac{2V_p}{\pi} \quad (9)$$

Donde  $V_p$  es el voltaje pico en la carga

- Voltaje RMS:

$$V_{rms} = \frac{V_p}{\sqrt{2}} \quad (10)$$

Donde  $V_p$  es el voltaje pico en la carga

- Corriente DC:

$$I_{dc} = \frac{V_{dc}}{R} \quad (11)$$

Donde  $V_{dc}$  es el voltaje medido en corriente directa sobre la carga y  $R$  corresponde al valor de la carga que tiene el circuito.

- Corriente RMS:

$$I_{rms} = \frac{V_{rms}}{R} \quad (12)$$

---

Donde  $V_{rms}$  es el voltaje RMS sobre la carga y  $R$  corresponde al valor de la carga que tiene el circuito.

- Potencia DC:

$$P_{dc} = V_{dc} * I_{dc} \quad (13)$$

Donde  $V_{dc}$  es el voltaje en corriente directa sobre la carga e  $I_{dc}$  es la corriente directa sobre la carga.

- Potencia AC:

$$P_{ac} = V_{rms} * I_{rms} \quad (14)$$

Donde  $V_{rms}$  es el voltaje RMS sobre la carga e  $I_{rms}$  es la corriente RMS sobre la carga.

- Eficiencia del Circuito:

$$V_{ac} = \sqrt{V_{rms}^2 - V_{dc}^2} \quad (15)$$

Donde  $V_{rms}$  es el voltaje RMS sobre la carga y  $V_{dc}$  es el voltaje medido en corriente directa sobre la carga.

#### 6.1.4. Convertidor de Voltaje DC

Existen diferentes aplicaciones para los convertidores DC/DC, usualmente estos son usados en aplicaciones relacionadas a las fuentes electrónicas de baterías, control de motores DC, y en sistemas fotovoltaicos. El convertidor de tensión (Boost), es un circuito elevador de tensión el cual aprovecha la naturaleza de sus componentes para amplificar la señal de voltaje de salida con respecto a su señal de entrada. Este circuito se compone de: Transistor (MOSFET), diodo (D), Inductancia (L), capacitancia (C).

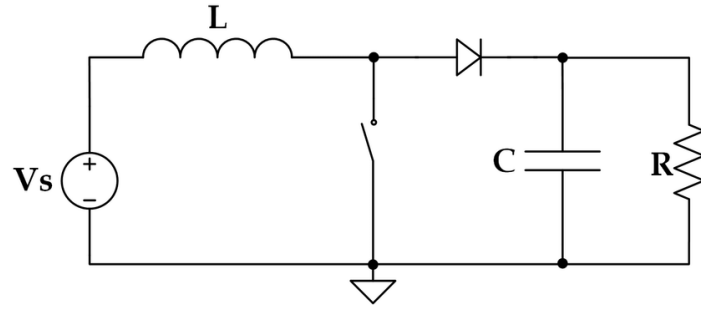


FIGURA 3: Convertidor de voltaje en su topología elevador. Fuente: Elaboración propia.

En la Figura 3, se observa el convertidor de voltaje el cual posee dos modos de conducción, los cuales se van alternando conforme se va activando o desactivando el transistor. Hay que tener presente que mediante la conmutación del MOSFET se genera un efecto de carga y descarga en la inductancia, pasando así por el diodo una corriente hacia el capacitor.

#### 6.1.4.1. Modo de Conducción Continua

Este modelo se produce cuando la corriente que circula por la Inductancia nunca llega a un valor nulo. Al cerrar el interruptor el circuito queda de la siguiente manera.

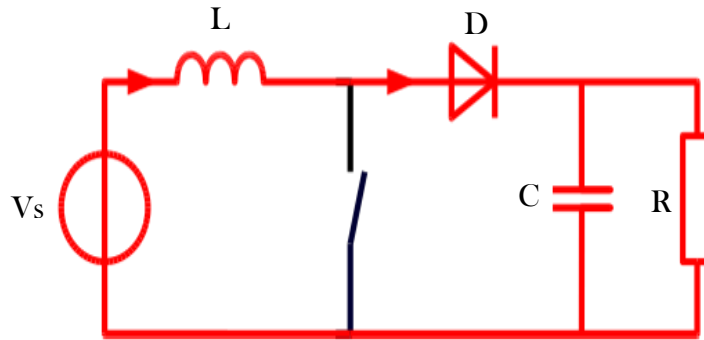


FIGURA 4: Representación esquemática de  $T_{OFF}$  en el convertidor. Fuente: Elaboración propia.

En la figura 4, se observa que el circuito solamente tiene en operación la mitad del convertidor elevador, esto se debe a que el diodo se encuentra polarizado en inversa, y es por ello por lo que el voltaje de la inductancia se encuentra definido por:

$$V_s = L \frac{dI_L}{dt} \quad (16)$$

Sabiendo que el periodo que varía la corriente es constante, la variación de la corriente en la bobina se encuentra definida por:

$$(\Delta i_L)_{abierto} = \frac{V_s D T}{L} \quad (17)$$

Al modificar el estado del interruptor se crea un circuito abierto, es entonces cuando la corriente sigue el lazo y entra en contacto con los demás componentes del convertidor.

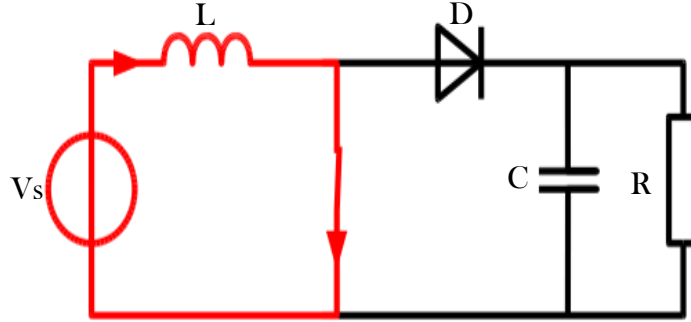


FIGURA 5: Representación esquemática de  $T_{ON}$  en el convertidor. Fuente: Elaboración propia.

En la figura 5, se muestra como la corriente en el inductor ( $L$ ) no varía instantáneamente, esto genera que el diodo se polarice directamente. Al polarizarse el diodo ( $D$ ), la energía almacenada en la inductancia pasará por la capacitancia, teniendo en cuenta lo anterior la tensión que cae en la inductancia es:

$$\frac{dI_L}{dt} = \frac{V_s - V_o}{L} \quad (18)$$

Sabiendo que el interruptor esta activado, la corriente debe variar linealmente, esta variación es constante y se puede demostrar con la siguiente expresión:

$$(\Delta i_L)_{cerrado} = \frac{(V_s - V_o)(1 - D)T}{L} \quad (19)$$

Dadas las expresiones 17 y 19, y sabiendo que la variación neta de corriente es de 0, se obtiene:

$$V_s(D + 1 - D) - V_o(1 - D) = 0 \quad (20)$$

Donde se despeja la tensión de salida  $V_o$ , obteniendo así la siguiente expresión:

$$V_o = \frac{V_s}{1 - D} \quad (21)$$

De la expresión 21, se observa que si el valor del ciclo de trabajo ( $D$ ) es cero, el voltaje de salida  $V_o$ , será igual al voltaje de entrada  $V_s$ . Ahora bien, si el ciclo de trabajo aumenta, en dicha relación se observa cómo va a tender a tener un valor alto el voltaje de salida  $V_o$ , sin duda en la implementación existirá un límite de voltaje, pero en condiciones ideales un ciclo de trabajo alto dará como resultado un voltaje de salida  $V_o$ , bastante alto.

en la figura 6, se muestran las correspondientes formas de onda de la tensión y corriente en el convertidor. Para el análisis general del convertidor elevador también es importante tener en cuenta los valores de corriente que circulan por la bobina. Valores tales como corriente mínimas y máximas con las que opera el convertidor.

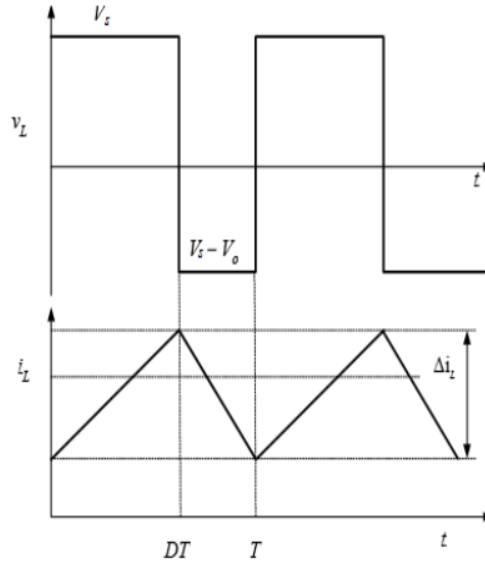


FIGURA 6: forma de onda del convertidor. Fuente: Elaboración propia.

Para conocer la corriente que circula en la bobina se debe tener presente la potencia suministrada por la fuente, la cual debe ser igual a la absorbida por la carga, por lo que igualando dichas potencias y usando la ecuación de tensión de salida se tiene la siguiente expresión:

$$i_L = \frac{V_s}{(1 - D)^2 R} \quad (22)$$

De lo anterior se deduce:

$$i_{Max} = i_L + \frac{\Delta i_L}{2} \quad (23)$$

---


$$i_{Min} = i_L - \frac{\Delta i_L}{2} \quad (24)$$

Donde dicha variación de corriente  $\frac{\Delta i_L}{2}$ , se calcula:

$$\frac{\Delta i_L}{2} = \frac{V_s D T}{2L} \quad (25)$$

Otro valor importante es el de la inductancia mínima, este valor indica el valor mínimo requerido de la inductancia ( $L$ ) con el que el convertidor elevador permanecerá en modo continuo con corriente permanente (CCM).

$$L_{Min} = \frac{D(1-D)^2 R}{2f} \quad (26)$$

Donde  $f$  es la frecuencia de conmutación del convertidor,  $D$  es el ciclo de trabajo y  $R$  es la carga. Por otro lado, es importante mencionar algunas ventajas que se pueda presentar en el Continuous Conection Mode (CCM).

- La ganancia de voltaje no depende de la carga
- La corriente de entrada es continua y no pulsante
- Se pueden lograr mayores eficiencias en comparación con el Discontinuous Conection Mode (DCM).

#### 6.1.4.2. Modo de Conducción Discontinua

Este modo de operación se relaciona al instante en donde la corriente que pasa por la inductancia ( $L$ ) es 0 o tiene un valor nulo. En esta oportunidad la ecuación de voltaje de la inductancia ( $L$ ) es igual a 0 suponiendo así que el voltaje del diodo  $V_d$  y el ciclo de trabajo  $D$  son valores constantes.

$$\frac{V_o}{V_d} = \frac{\Delta L + D}{\Delta L} \quad (27)$$

$$\frac{I_o}{I_d} = \frac{\Delta L}{\Delta L + D} \quad (28)$$

En la figura 7, se observa como la corriente media de entrada tiende a ser igual a la corriente de la bobina, obteniendo así:



$$I_d = \frac{V_d}{2L} DT_s(D + \Delta L) \quad (29)$$

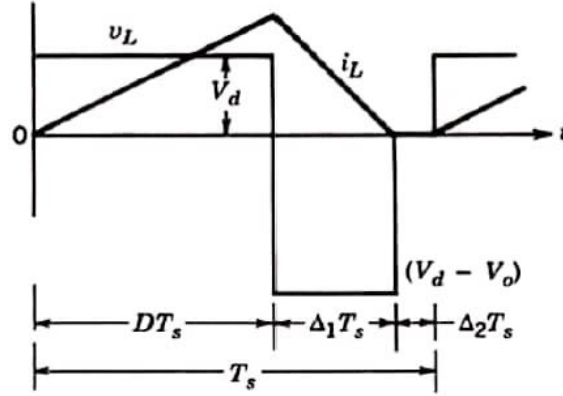


Figura 7: Formas de Onda del Convertidor en Modo Discontinuo. Fuente: [12].

Conociendo el valor anterior, es posible hallar el valor de la corriente de salida, para ello se usa la siguiente expresión:

$$i_o = \left( \frac{V_d T_s}{2L} \right) D \Delta_L \quad (30)$$

En esta oportunidad también es importante mencionar algunas ventajas que se pueda presentar en el Discontinue Connection Mode (DCM).

- La ganancia de la carga depende de los parámetros de diseño ( $L$  y  $F_s$ ).
- El concepto de ondulación de la corriente del inductor es mayor y su valor RMS es mayor.
- La conmutación de encendido del interruptor controlado se realiza con corriente cero, reduciendo pérdidas de conmutación.

## 6.2 Consideraciones del PSO

Es importante considerar algunas de las desventajas presentes en el algoritmo del PSO. Uno de los problemas más recurrentes está relacionado a los mínimos locales y a la precisión [13], esto sin duda dificulta su implementación, sin embargo, existen algunas maneras de implementar un sistema de búsqueda alternativo [14], donde básicamente el PSO tiene un grupo del enjambre en la mejor posición encontrada, mientras otro grupo busca en el espacio los mejores valores. En este método se aplica una disminución de la constante inercial, así, al principio hay un amplio campo de búsqueda, en donde se les da mayor peso a las otras constantes.

## 6.3 Controlador PID

Los controladores PID constituyen aproximadamente la mitad de los controladores implementados en procesos industriales [15]. Su uso elevado está arraigado a su aplicabilidad a la mayoría de los sistemas de control, simplicidad al no necesitar de un modelo matemático de la planta y resultados sobresalientes.

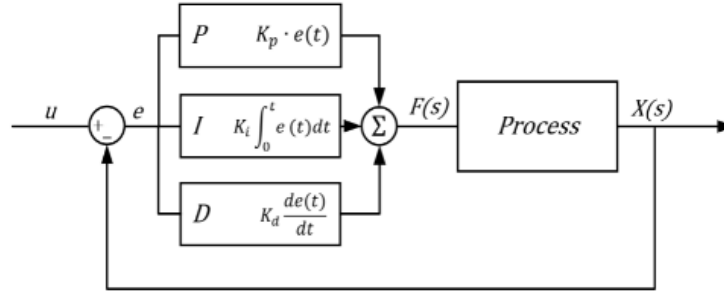


FIGURA 8: Diagrama de Bloques del controlador. Fuente: Elaboración propia.

$$u(t) = K_p e(t) + k_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (31)$$

De la ecuación 31 se puede observar las características generales del controlador PID, en donde  $y$  es la variable medida del proceso,  $r$  la variable de referencia,  $u$  es la señal de control y  $e$  es el error de control [16].

$$e = y_{sp} - y \quad (32)$$

Cada componente del controlador tiene una acción directa sobre el sistema de control. El control proporcional afecta la ganancia del sistema, sin embargo, siempre existe error de estado estacionario y aunque este disminuya de acuerdo con el aumentar su ganancia, también provoca el aumento de oscilaciones en la señal de control. La acción derivativa puede ser interpretada como una extrapolación lineal del tiempo  $T_d$  por lo que al tratarse de una predicción ser demasiado corta puede traer oscilaciones y demasiado extensa no ayuda al sistema. Finalmente, la acción integral elimina el error de estado estacionario, aunque un tiempo elevado  $T_i$  hará que el sistema de control esté más cerca de la acción proporcional que la integral.

### 6.3.0.1. Filtro Derivativo

Normalmente los controladores PID no funcionan de forma óptima debido a que ignoran el ruido y efectos no lineales presentes en el sistema como el wind up el cual se discutirá

---

más adelante [17]. Sin embargo, existen formas de atenuar estos problemas, en este caso el filtro al componente derivativo se aplica como medida a la existencia de ruido en todo sistema de censado, generalmente de alta frecuencia, la derivada de este censado producirá un error en la lectura que aumentará con el tiempo, desplazando el valor real y llevando el sistema a entradas con valores incorrectos.

Por esta razón se aplica un filtro de primer orden en la acción derivativa con el propósito de anular la componente de alta frecuencia en el censado, ya que simplemente no incluir la acción derivativa (PD), puede tener desventajas al no tener la capacidad predicativa de dicha componente.

#### 6.3.0.2. Wind Up

Wind up es conocido como un fenómeno no lineal presente en los controladores, este aparece cuando la acción integral y la saturación se encuentran en el sistema de control, haciendo que el error aumenta desmedidamente debido a la saturación en la lectura de un valor, dando resultado a grandes y lentas oscilaciones en el control [17]. Existen diferentes formas de dar solución a dicho problema, la limitación de la acción integral o “clamping” suele funcionar sin problemas, este método desactiva dicha acción dejándola en 0 cuando se alcanzan los límites del controlador.

#### 6.3.0.3 Discretización

El controlador PID es visto como la sumatoria de las diferentes acciones vistas (Proporcional, Integral y Derivativa), cada una de las cuales cumple con un propósito específico, en el dominio del tiempo el controlador es dado por 7.1, al momento de utilizar un dispositivo electrónico como controlador, se debe tener en cuenta que el sistema no puede leer la información de forma continua, por lo que el controlador es discretizado [18].

Dependiendo del periodo de muestreo  $T_s$  es discretizado el controlador, es necesario tener en cuenta las cualidades del dispositivo que efectuará el control digital, puesto que dependiendo de su velocidad y resolución se deberá elegir un periodo de muestreo y una escala conforme al valor de la lectura.

Los componentes Integral y derivativo del controlador pueden ser expresados en tiempo discreto de 3 formas distintas de acuerdo con el método de transformada z seleccionado (Euler directo, Euler inverso y trapezoidal).

Integrador Discretizado Método Euler Directo:

$$\frac{K_i T_s}{z - 1} \quad (333)$$

---

En la ecuación 33 del método de discretización Euler Directo, se observa  $K_i$  la cual es la constante proporcional,  $T_s$  corresponde al periodo de muestreo y la constante  $z$  de la transformada.

Integrador Discretizado Método Euler Inverso:

$$\frac{K_i T_s z}{z - 1} \quad (344)$$

La ecuación 34 de método de discretización Euler inverso, se observa  $K_i$  la cual es la constante integral,  $T_s$  corresponde al periodo de muestreo y la constante  $z$  de la transformada con un valor adicional en el numerador.

Integrador Discretizado Método Trapezoidal:

$$\frac{K_i T_s}{2} + \frac{z + 1}{z - 1} \quad (355)$$

El método trapezoidal de la ecuación 35 igualmente posee la constante integral  $K_i$  y el periodo de muestreo  $T_s$ , agregando la constante de la transformada en el numerador y denominador.

Derivador Discretizado Método Euler Directo:

$$\frac{K_d N}{1 + \frac{NT_s}{z - 1}} = \frac{K_d N(z - 1)}{z - 1 + NT_s} \quad (366)$$

De la ecuación 36 se observan las constantes de  $K_d$  la cual es la constante derivativa del controlador,  $N$  como el filtro derivativo,  $T_s$  el periodo de muestreo, y  $z$  de la transformada.

Derivador Discretizado Método Euler Inverso:

$$\frac{K_d N}{1 + \frac{NT_s z}{z - 1}} = \frac{K_d N(z - 1)}{(1 + NT_s)z - 1} \quad (377)$$

El método de Euler inverso de la ecuación 37 difiere con el método Directo en el denominador, agregando un  $z$  multiplicando al periodo de muestreo  $T_s$  y la constante  $N$ .

Dependiendo del método del controlador se establece una función de transferencia en tiempo discreto del controlador, la cual es la base para programarlo de forma digital.

$$C(z) = K_p + \frac{K_i T_s z}{z - 1} + \frac{K_d N(z - 1)}{(1 + NT_s)z - 1} \quad (388)$$

Con la función de transferencia de la ecuación 38 se establece la relación de la salida con respecto a la entrada, despejando la solución de la salida actual con respecto a entradas y salidas en momentos anteriores.

---

## 6.4. Algoritmos Bio-inspirados

El análisis del entorno ha sido un elemento importante en el desarrollo de tecnología durante mucho tiempo, los comportamientos de la biología permiten modelar soluciones a problemas complejos de la actualidad, gracias a esto se han desarrollado diferentes metodologías metaheurísticas, transfiriendo el conocimiento de diferentes formas de vida a la tecnología humana [19].

### 6.4.1. Basados en Biología

#### 6.4.1.1. Basados en Evolución

Este tipo de algoritmos tienen un comportamiento a partir de una población, donde hay crecimiento, selección, y mutación generalmente, de acuerdo con estas normas básicas, se estudia el comportamiento del grupo para que mute generación tras generación hasta que los individuos cumplan con las condiciones de la función objetivo [20]. En esta clasificación se encuentran los siguientes algoritmos.

- Algoritmos Genéticos
- Evolución Diferencial
- Evolución Cultural
- Estrategias Evolutivas
- Programación Genética

#### 6.4.1.2. Basados en Comportamiento de Enjambre

En este caso el comportamiento de la población es de acuerdo con el movimiento de los miembros en el espacio, lo que quiere decir que no existe un factor de mutación que cambie a la población por una nueva generación, en vez de eso, la población se ubica por sus vecinos y su evaluación en el entorno cercano. De acuerdo con el tipo de comunicación que exista entre los miembros de la población (enjambre), estos se desplazarán en cierto patrón. Algunos de los algoritmos que tienen este tipo de comportamiento son los siguientes.

- Optimización por enjambre de partículas
- Optimización por colonia de hormigas
- Abejas artificiales
- Termitas
- Murciélagos
- Pájaros
- Gatos
- Búsqueda de Bacterias

- 
- Búsqueda cuckoo

## 6.5.2. Basados en fenómenos naturales

### 6.5.2.1. Basados en Leyes Físicas

Existe un grupo que, de acuerdo con el comportamiento del ambiente, puede usarse para encontrar solución a diferentes problemáticas, en este caso nuevamente una población se comporta de acuerdo con un fenómeno físico con restricciones específicas. Algunos ejemplos de este tipo de algoritmos son.

- Algoritmo de Búsqueda Gravitacional
- Algoritmo de Big Bang Big Crush
- Algoritmo Gotas de Agua

---

## Capítulo 7

# Estado del Arte

### 7.1. Corrección del Factor de Potencia

Las técnicas de controladores actuales para los convertidores con corrección del factor de potencia, especialmente en el modo de conducción continuo, han ganado importancia en los convertidores AC/DC [21], [22], [23] que se utilizan en aplicaciones de alto rendimiento, donde la respuesta rápida y alta precisión son importantes. Existen varios métodos de control de corriente, incluido el control de histéresis, control lineal, control de modo de corriente promedio y control predictivo. En cualquier método de control de corriente, la señal de corriente del inductor se convierte en su señal analógica de voltaje y se realimenta en el sistema para ser comparada con el control de voltaje, de esta forma se varía el ciclo útil del convertidor [24].

Mediante un control basado en un doble lazo, uno para la corriente y otro el voltaje, se diseñan los controladores de acuerdo con el análisis frecuencial de la ecuación del convertidor. Los resultados de acuerdo con el documento muestran una reducción efectiva de la distorsión armónica total en el convertidor (THD) la cual sin controlador tenía un valor del 78 %, con controlador su nuevo valor encontrado fue del 11 %. Se encontró un aumento del factor de potencia, el cual originalmente se encontraba en 0.82 sin controlador y con controlador su valor final fue de 0.995. El porcentaje de regulación del voltaje tuvo un aumento hasta el 19.9 % con respecto al 3.6 % inicial [2].

La corrección del factor de potencia en convertidores mediante la sintonización de controladores es un elemento de estudio no solamente con sintonización empírica, también métodos alternativos, como el uso de lógica difusa [12]. En el documento se explica las diferentes funciones de membresía que se utilizaron en el error, variación del error y el ciclo útil, para obtener un factor de potencia cercano a la unidad y una disminución de la distorsión armónica total, pasando del 128.26 % al 0.42 %.

### 7.2. Sintonización de Controladores

#### 7.2.1. Sintonización de acuerdo con criterios empíricos o analíticos

El desarrollo de métodos de sintonización de controladores, ha sido un tema extenso a lo largo del tiempo, desde la propuesta de control Ziegler Nichols en 1942. Normalmente los métodos de sintonización se dividen en métodos de lazo abierto y de lazo cerrado. Teniendo en el controlador de lazo cerrado, métodos comúnmente utilizados como; Ziegler y Nichols, Cohen y Coon, López et al, kaya y Sheib o Sung et al [25].

Mediante un diseño de controlador basado en la respuesta en frecuencia de la función de transferencia de un convertidor elevador, se construye con un controlador con el fin de corregir el factor de potencia, los resultados del controlador son efectivos en la corrección

---

y la reducción de la distorsión armónica total hasta 6.13% [26]. El controlador fue diseñado por métodos analíticos y su sintonización depende de parámetros que podrían verse afectados por el tiempo o desgaste de la planta.

El control robusto es utilizado ampliamente por tener ventajas en la determinación efectiva de los parámetros del controlador, para sistemas que presentan comportamientos no lineales, como es el caso del convertidor elevador, estrategias de sintonización convencionales no poseen este tipo de ventajas, ya que al tener una parte o un modelo aproximado de funcionamiento sin tener en cuenta las incertidumbres del convertidor, el modelo no puede adaptarse a cambios bruscos en la carga o en el ciclo útil [27]. El controlador robusto es un planteamiento a partir de un modelo analítico, que tiene en cuenta diferentes incertidumbres que aproximan el sistema a un comportamiento más real, esta estrategia es más extensa y sin duda, posee mejores resultados que un controlador convencional, pero por otro lado también está demostrado que la sintonización por métodos de aprendizaje de máquina pueden llegar a encontrar los parámetros del controlador sin necesidad de tener que reconstruir un modelo del sistema, ni tener que incluir incertidumbres para aproximarlos a un modelo real.

### 7.2.2. Sintonización por métodos de inteligencia artificial

Los métodos de sintonización inteligente dan nuevas formas de emplear la sintonización de controladores, el objetivo de los métodos de inteligencia artificial es la adaptación del controlador al problema mediante procesamiento, encontrar el grupo de parámetros que cumpla con los criterios de diseño, estos elementos pueden ser utilizados en conjunto y aplicados efectivamente a convertidores [28].

Los sistemas inmunes artificiales consisten en la proliferación de una población que dé solución a un tipo de problemática, la población se adapta con cada generación al problema hasta darle solución efectivamente [29], [30].

Haciendo uso de sistemas inmunes artificiales se realiza el control de temperatura de una planta con resultados efectivos y una comparación con controladores sintonizados de forma convencional. El controlador sintonizado por métodos empíricos poseía, un sobre pico máximo cercano al 20 % y un tiempo de estabilización de 800 segundos, por su parte el controlador sintonizado por sistemas inmunes artificiales poseía un sobre pico máximo nulo además de reducir el tiempo de estabilización del controlador a la mitad [17].

El documento presenta un algoritmo de control PID, este algoritmo basado en un sistema bio-inspirado inmune artificial que puede adaptarse a las ruedas del motor de un vehículo eléctrico [4]. El algoritmo resuelve la limitación del PID clásico cuando se usa en el controlador de las ruedas del motor del vehículo eléctrico. Los resultados del documento fueron un exitoso control del voltaje de PWM con respecto al control de velocidad del motor, lo cual genera una relación estable entre la aceleración y la velocidad del motor.

[31] Trabajo de la universidad Santo Tomas en el que, de acuerdo con la evolución de un algoritmo bio inspirado, con tan solo 15 generaciones ya se puede tener una población efectiva, la cual es seleccionada para el controlador de la planta, tras 22 generaciones el algoritmo de población muestra excelentes resultados. Nuevamente se realiza una comparación con métodos de sintonización de controladores PID empíricos, el trabajo



---

selecciona el método Ziegler Nichols, los resultados fueron un sobre pico máximo del 66 %, error de estado estacionario de 0.018 % y un tiempo de establecimiento de 0.68 segundos, el controlador Bio-inspirado además de presentar una adaptación casi total al problema, después de 22 de generaciones, obtuvo los siguientes resultados; sobre pico máximo de 5.2587 %, tiempo de establecimiento de 0.1862 segundos y error de estado estacionario de 0.0083 %.

Diferentes algoritmos bio-inspirados son utilizados para resolver problemáticas variadas, en la sintonización de controladores, se recolectaron diferentes técnicas para este trabajo; los sistemas inmunes artificiales [5], [4], [18], [32], los algoritmos genéticos [33] y de enjambre [34], [35], [36], [37], todas estas técnicas son aplicaciones directas de los algoritmos bio-inspirados.

---

## Capítulo 8

# Metodología

### 8.1 Diagrama Metodológico

En esta sección se realiza una representación de las actividades del proyecto, mediante un diagrama de flujo y una descripción detallada de cada una de las actividades desarrolladas a lo largo del tiempo, teniendo en cuenta los objetivos específicos como puntos de referencia en los avances de forma cronológica y detallando puntos clave del desarrollo de cada una las fases.

#### 8.1.1. Diagrama de Flujo

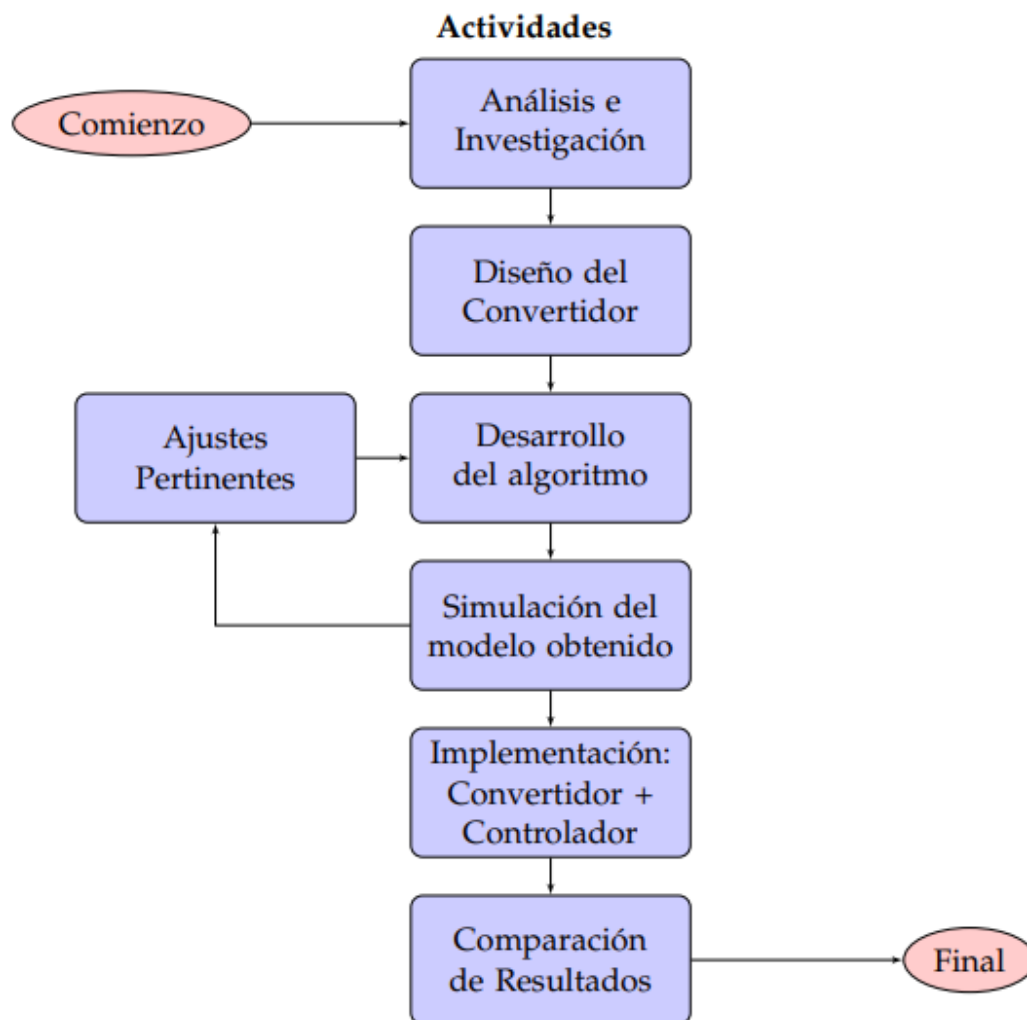


Figura 9: Diagrama de Flujo de actividades realizadas a lo largo del proyecto - Actividades descritas de forma cronológica de acuerdo con los objetivos específicos del proyecto. Fuente: Elaboración propia.

---

El diagrama de flujo de la Figura 9 muestra las 7 actividades principales del proyecto, la rama de realimentación y de ajustes pertinentes está estructurada como una actividad de a la par del desarrollo del algoritmo y de la simulación del modelo obtenido, en esta rama es en donde se lleva a cabo el objetivo general del proyecto.

### 8.1.2. Análisis e Investigación.

Se realizó la recolección de diferente literatura, referente al desarrollo de controladores convencionales y controladores adaptativos inteligentes, posteriormente se seleccionó el criterio de sintonización más acorde tanto al propósito del convertidor (regular el voltaje de salida) como del proyecto (Corrección del factor de potencia).

### 8.1.3. Diseño del Convertidor

Con la investigación y de acuerdo con el trabajo del codirector del proyecto Jhon Bayona, en el desarrollo de convertidores conmutados [33]; se desarrolló e implementó un convertidor AC/DC con corrección del factor de potencia en topología elevador, con la finalidad, de implementar un controlador PID sintonizado mediante algoritmos bio-inspirados, que realice la corrección del factor de potencia de dicho convertidor.

Las especificaciones de fabricación del convertidor AC/DC son: voltaje de salida directo 220 V, voltaje de entrada alterno 120 V y potencia 100 W.

### 8.1.4. Desarrollo del algoritmo

Teniendo en cuenta los antecedentes del documento y el trabajo de investigación, se seleccionó un tipo de algoritmo a trabajar (Sistemas inmunes artificiales, Colonia, Algoritmos Genéticos, redes neuronales, etc.), dadas unas variables de análisis del convertidor (Voltajes y Corrientes) con las que el sistema mediante cada iteración aprende a minimizar el error.

### 8.1.5. Simulación del modelo obtenido

Mediante el software de MATLAB & SIMULINK, se implementó el modelo desarrollado del convertidor AC/DC con corrección del factor de potencia en topología elevador y el modelo del controlador PID sintonizado por algoritmos bio-inspirados. La simulación busca la verificación de resultados de acuerdo con la disminución del error medio cuadrático en el algoritmo, mediante cada iteración, haciendo uso de esta herramienta el

---

sistema encontró una la solución del problema teniendo un valor objetivo (factor de potencia).

#### 8.1.6. Ajustes Pertinentes

De acuerdo con la eficiencia en la capacidad del algoritmo para obtener la solución deseada, haciendo uso de métricas tales como: disminución del error medio cuadrático y el número de iteraciones necesarias para la obtención de la solución, se ajustó diferentes parámetros del procesamiento, para adaptar mejor el sistema al problema.

#### 8.1.7. Implementación: Convertidor + Controlador

La implementación física se realizó de acuerdo con el trabajo y supervisión del codirector del proyecto Ing. Jhon Bayona, en cuanto a los elementos necesarios para la construcción del convertidor AC/DC con corrección del factor de potencia en topología elevador. Por su parte y siguiendo los criterios seleccionados para el desarrollo del algoritmo, el director del proyecto Ing. José Guarnizo, dio las pautas principales para la lectura y procesamiento correcto de las variables a leer en el controlador.

#### 8.1.8. Comparación de Resultados

Con el uso de métricas de desempeño en el procesamiento interno del controlador (Numero de Iteraciones, Voltajes, Corrientes, Error(es), Tiempos de estabilización, etc.), se realizó un listado de acuerdo con los parámetros más importantes y también se realizó una comparación con un controlador PID convencional, utilizado para el mismo convertidor, en un trabajo publicado y reportado en la literatura [38], de autoría del codirector del proyecto Ing. Jhon Fredy Bayona Navarro.

---

## Capítulo 9

# Diseño

### 9.1. Estrategia de Control Inteligente

#### 9.1.1. Análisis y Determinación de Objetivo

El convertidor elevador posee diferentes elementos y comportamientos que por sus características hacen que exista un desfase en las señales de corriente y voltaje, esto genera un problema en el factor de potencia de dicho convertidor. Ya que se desea corregir el factor de potencia, es necesario corregir este desfase que existe entre ambas señales, para ello se debe tener en cuenta el voltaje de entrada del convertidor  $V_i$ , este pasa por un puente rectificador y se obtiene una señal senoide con solamente el semiciclo positivo por lo que la corriente debería tener la misma forma de onda.

#### 9.1.2. Selección Estrategia

El ciclo útil del convertidor es la variable de entrada al sistema, este controla el comportamiento del convertidor, haciéndolo pasar entre el estado de encendido y apagado, de esta forma la corriente en el circuito pasa a tener el valor de la fuente de entrada a dividirse entre el condensador y la carga, de esta forma si se cambia el ciclo útil, también se afecta la corriente en el convertidor.

Como se menciona es necesario controlar la corriente en el convertidor, la forma seleccionada de hacerlo es con un controlador PID, como se vio en el capítulo anterior, lo mejor es que el controlador tenga la cuarta variable  $N$  del filtro derivativo para evitar ruido en la señal de lectura, por lo que realmente se están buscando las cuatro constantes con sus respectivos valores PIDN, estos valores conforman un espacio de 4 dimensiones, los cuales de acuerdo a una función objetivo deben ser explorados en busca de un controlador que cumpla con las exigencias del sistema.

La función objetivo debe tener en cuenta que es necesario que voltaje y corriente se encuentren en fase, por lo que el error que existe en el desfase de ambas señales debería ser la función objetivo, sin embargo verificar solamente la fase de las señales puede ser algo complicado, por lo que comúnmente en correctores del factor de potencia se implementa un control de corriente promedio, en donde la referencia del controlador de corriente será el voltaje de entrada rectificado y escalado al mismo valor de la corriente, de forma que el error que existe entre las señales, al estar en la misma escala, será solamente el desfase que existe entre ambas, haciendo así que el controlador tenga como objetivo hacer que las dos señales estén en fase, corrigiendo así el factor de potencia.

---

Para que el control sea inteligente, debería poder probar diferentes controladores, a medida que el algoritmo va avanzando debería, mejorar la respuesta de los controladores y haciendo que el sistema mejore después de entrenar muy poco tiempo, es por ello que se realiza un análisis exhaustivo de los algoritmos bio-inspirados presentes hasta el momento.

### 9.1.3. Algoritmo Implementado

De acuerdo con diferentes documentos comparativos y recopilatorios, el candidato seleccionado para sintonizar las constantes de un controlador, siendo rápido, simple, eficiente, y escalable es el algoritmo PSO (particle Swarm Optimization), esto debido a su robustez, años de publicaciones, donde se analizan sus características y demás elementos a mejorar, la posibilidad de modificaciones al algoritmo original a nivel muy alto sin afectar su eje central, entre otros. Por estas razones el algoritmo PSO es un candidato más que sobresaliente para este trabajo.

#### 9.1.3.1. Descripción general del algoritmo del PSO

El algoritmo de Particle Swarm Optimización (PSO), es uno de los algoritmos metaheurísticos inspirados en la naturaleza más populares hasta la fecha. El PSO fue desarrollado por James Kennedy y Russell Eberhart en el año de 1995 [39], y desde entonces se ha podido observar como a lo largo del tiempo el PSO se puede implementar para abordar problemas relacionados con la optimización, así mismo esto permitió relacionar el algoritmo en campos de la ciencia y la ingeniería [40], [41].

La descripción general del algoritmo del PSO para cada partícula en cada paso de tiempo debe atravesar el espacio, donde encontrará un conjunto de posibles soluciones, la intención es buscar la mejor solución en el espacio de búsqueda.

$$\vec{\bar{x}}_i(t) \in \mathcal{X} \tag{399}$$

Ahora bien, para un grupo de varias partículas (enjambre), cada partícula del enjambre debe que tener en cuenta los siguientes parámetros de optimización.

- $\vec{\bar{x}}_i(t) \in \mathcal{X}$  Representando su posición.
- $\vec{\bar{v}}_i(t)$  Representando su velocidad.
- $\vec{\bar{p}}_i(t)$  Representando su mejor posición (experiencia).
- $\vec{\bar{p}}_i(t)$  Representando la mejor experiencia entre todas las partículas (enjambre).

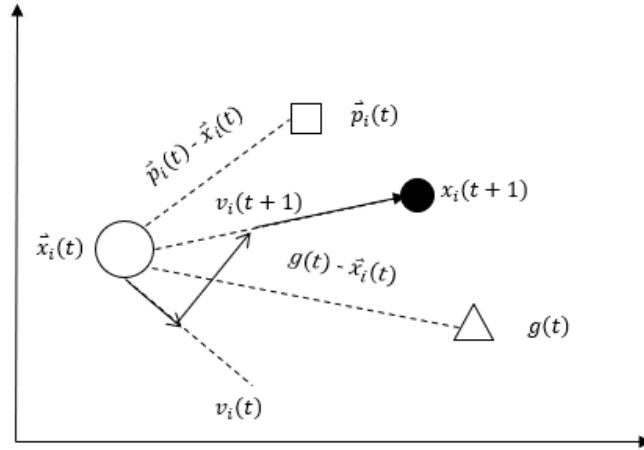


Figura 10: Movimiento de partículas PSO en un espacio de búsqueda multidimensional.

En la figura 10, se observa el modelo metafísico del PSO, allí por cada movimiento de una partícula de un enjambre se tienen en cuenta tres vectores que establecen cómo obtendrá la mejor ubicación dentro del espacio de búsqueda.

Sin embargo, la siguiente ecuación establece la actualización de la posición de las partículas pertenecientes al enjambre.

$$x_i(t) = x_i(t) + v_i(t + 1) \quad (40)$$

Donde:

$$v_i(t + 1) = wv_i(t) + c_1(p_i(t) - x_i(t)) + c_2(g(t) - x_i(t)) \quad (401)$$

Donde, el termino inercial está representado por esta expresión  $wv_i(t)$  y  $w$  Es una constante que representa el valor inercial. Esta expresión  $c_1(p_i(t) - x_i(t))$  Representa el componente cognitivo, y esta última expresión  $c_2(g(t) - x_i(t))$  representa el componente social. Cabe destacar que  $c_1$  es la constante que premia la experiencia personal de cada partícula y  $C_2$  es la constante que premia la experiencia colectiva del enjambre, Ambas representan las constantes de aceleración que varían su valor entre 0 y 1.

### 9.1.3.2. Variación de las constantes de aprendizaje

- $C_1 > 0$  and  $C_2 = 0$ , Representa partículas independientes, sin líder
- $C_1 = 0$  and  $C_2 > 0$ , Partícula que premia la experiencia colectiva, es decir, es atraída por un único líder.
- $C_1 > 0$  and  $C_2 > 0$ , Partículas atraídas por una posición media (independiente)

---

**Algorithm 1** Multi - PSO

---

```
1: Initialize Parameters: c, dt, epochs, s, n, Lim
2: for  $swarm = 1, 2, \dots, s$  do
3:   for  $particle = 1, 2, \dots, n$  do
4:     Initialize:  $x, v$ 
5:   end for
6: end for
7:
8: for  $epochs = 1, 2, \dots, epochs$  do
9:   for  $swarm = 1, 2, \dots, s$  do
10:    for  $particle = 1, 2, \dots, n$  do
11:      Evaluate particle position with (Eq.: 3.24)
12:    end for
13:  end for
14:
15:  for  $swarm = 1, 2, \dots, s$  do
16:    Find Best Value
17:    Find Best Position
18:  end for
19:
20:  for  $swarm = 1, 2, \dots, s$  do
21:    for  $particle = 1, 2, \dots, n$  do
22:      Update Velocity with (Eq.: 3.25)
23:      Update Position with (Eq.: 3.26)
24:    end for
25:  end for
26: end for
27: Select best Swarm behavior:  $x_n$ 
```

---

Pseudocódigo 1, Descripción de alto nivel del algoritmo MPSO.

En el Pseudocódigo 1, se encuentra expresado de manera general los pasos que realiza el algoritmo del PSO junto a la modificación que se realizó obteniendo así la descripción general del MPSO.

## 9.2. Convertidor AC-DC con corrección del factor de potencia

El sistema de control de retroalimentación cuyo propósito es corregir el factor de potencia se aplica al convertidor elevador, el cual posee los siguientes parámetros de diseño que reposan en la Tabla 1:

Tabla 1: Parámetros del modelo. Fuente: Elaboración propia.

| Variable | Valor        |
|----------|--------------|
| Vout DC  | 220 V        |
| Vin AC   | 120 V        |
| Potencia | 100 W        |
| R        | 500 $\Omega$ |
| L        | 500 mH       |
| C        | 256 $\mu F$  |



Es importante mencionar que los valores RLC, se calcularon mediante el análisis característico del convertidor en esta topología. Por otro lado, tal cómo se especificó en la metodología, se parte como un trabajo complementario a la investigación realizada por el codirector del proyecto Ing. Jhon Fredy Bayona Navarro, y es por esta razón que el procedimiento no se incluye en este reporte. La investigación a la cual se hace referencia es la sintonización del controlador PID en un circuito PFC [33].

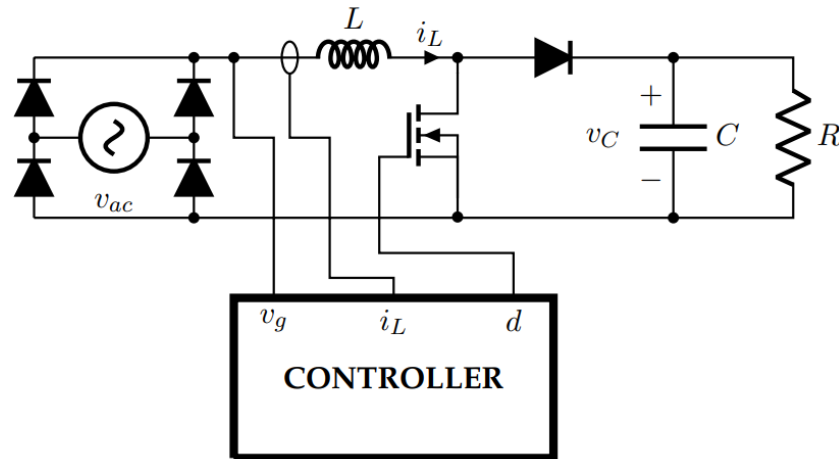


FIGURA 11: Convertidor elevador (Topología Elevador). Fuente: Elaboración propia.

En la Figura 11, Se encuentra la representación gráfica del sistema junto con la representación de la caja blanca del controlador de corriente así mismo se puede visualizar las variables de interés para el sistema de controlado. Donde:

- $v_{ac}$  Es el voltaje de entrada AC.
- $L$  Es el inductor del convertidor.
- $i_l$  Es la corriente del inductor.
- $v_c$  Es el voltaje del condensador.
- $C$  Es el condensador del convertidor.
- $R$  Es la carga del convertidor.
- $v_g$  Es la señal de voltaje de referencia.
- $i_l$  Es la señal de corriente de referencia.
- $d$  Es el ciclo útil.

### 9.3. Implementación en Simulación SIMULINK MATLAB

Mediante la herramienta de SIMULINK MATLAB, se construye un modelo del sistema completo teniendo en cuenta la lectura de diferentes variables con el fin de monitorizar el comportamiento y emplear los controladores de voltaje y de corriente.

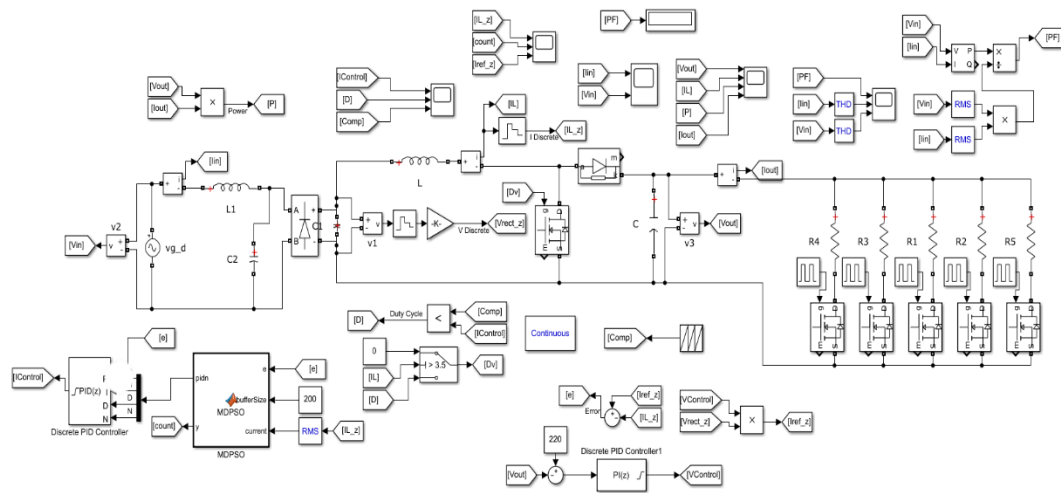


Figura 12: Convertidor Elevador con Cargas al 100 %/80 %/60 %/50 % y 20 %. Fuente: Elaboración propia.

Con el fin de comprobar la robustez de los controladores que se obtienen durante los entrenamientos, estos son sometidos a variaciones de carga como se puede ver en la Figura 12, la cual tiene 5 valores de carga diferentes, cambiando cada vez que un entrenamiento termine, esto debido a que los entrenamientos son realizados de acuerdo con la variación de la corriente RMS del convertidor, cuando esta cambia, activa un conteo para que el sistema se estabilice y comience un nuevo entrenamiento con el nuevo valor de carga.

El controlador  $PI$  tiene el propósito de mantener el voltaje de salida  $V_{out}$  en  $220V$ , independientemente de la carga que se ponga, así que cuando el convertidor necesita mantener el voltaje de salida, pero existe una variación en la carga, la corriente de referencia  $I_{ref}$  es reducida y con ella el ciclo útil  $D$  por lo que el voltaje se mantiene, pero a una potencia menor.

Normalmente al trabajar con el error medio cuadrático, basta con tomar el error que existe entre la corriente de referencia  $I_{ref}$  y la corriente en la bobina  $I_L$ , donde la corriente de referencia va a ser el valor del voltaje de entrada  $V_{in}$  escalado  $1.45A$  a  $170V$  multiplicado por la salida del controlador de voltaje  $PI$ .

el controlador de corriente tiene como propósito el seguimiento del voltaje de entrada  $V_{in}$  escalado, para que solamente exista error en el desfase de las señales, para esto un algoritmo MPSO es entrenado en donde evalúa diferentes controladores cada 500 ciclos de lectura, durante ese tiempo un controlador es colocado al sistema en tiempo real, permitiendo el entrenamiento del sistema de forma completa.

Finalmente, para evitar picos debido a controladores ineficientes, que se implementen durante el entrenamiento del convertidor, se implementa un paso de seguridad donde la lectura al sobrepasar los  $3A$  apaga la conmutación en el MOSFET.

---

## 9.4. Diseño e implementación física del circuito.

### 9.4. 1. Diseño de la carga.

Según los criterios de diseño del convertidor elevador, se debe implementar una carga de  $500\Omega$  la cual soporte una potencia aproximada a los  $100W$ , es por ello por lo que se decide realizar un banco de resistencias usando resistencias de un valor de  $10K\Omega$  a una potencia de  $10W$  cada una, al formar el paralelo entre las 20 resistencias de las mismas especificaciones se logrará obtener una carga total de  $500\Omega$ . Ahora bien, se necesita implementar un sistema que ayude a disipar el calor en el banco de resistencias, para ello se usan láminas de acero cold rolled calibre 16 las cuales están entre las resistencias separadas unas de otras y unidas con ayuda de un alambre dulce.

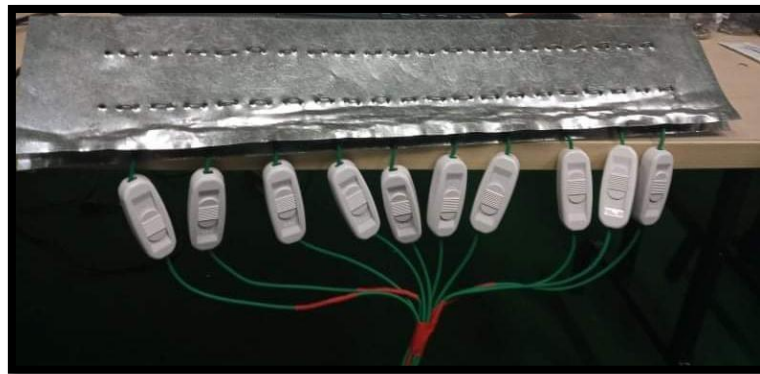


Figura 13: Carga de 500 - 2500 a 100W. Fuente: elaboración Propia.

En la Figura 13, se puede evidenciar el resultado final del banco de resistencias variables, el valor de la resistencia puede ser modificado al accionar cada interruptor de paso.

### 9.4.2. Diseño del circuito esquemático.

Para la implementación física del convertidor fue necesario implementar un circuito impreso (PCB) del convertidor elevador, esto sin antes tener el plano esquemático del convertidor. En el diseño se tomó en cuenta todos los elementos y componentes necesarios para implementar el convertidor junto a el lazo del control PID digital.



$$S_a = \frac{I_{rms}}{J} \quad (412)$$

Donde IRMS, es la corriente (IRMS) máxima que va a pasar por el convertidor. En esta oportunidad  $IRMS = 1A$ . Conociendo el valor de la densidad de corriente  $J$ ,  $S_a = 0,33 \text{ mm}^2$ . Con un espesor de conductor en el impreso  $a = 0,15 \text{ mm}$ . Para calcular la medida del ancho de la pista, se relaciona la siguiente ecuación:

$$b = \frac{0,33 \text{ mm}^2}{0,15 \text{ mm}} = 2,2 \text{ mm} \quad (423)$$

En donde en el numerador se relaciona el área de la sección transversal del material conductor ( $S_a$ ), y en el denominador se relaciona el grosor del material en el impreso (a). Cabe mencionar que también dentro de las consideraciones de diseño se quita la máscara de soldadura en las pistas de potencia. Esto debido a que de ser necesario se podría aumentar en área de la sección transversal ( $S_a$ ) del conductor con estaño.

#### 9.4.3.2. Listado de materiales

Tabla 2: Listado de Materiales. Fuente: Elaboración propia.

| Componente      | Referencia    | Cantidad |
|-----------------|---------------|----------|
| Condensador     | 0,1uF         | 1        |
| Condensador     | 0,1uF         | 2        |
| Condensador     | 10nF          | 1        |
| Condensador     | 10uF          | 2        |
| Condensador     | 220uF         | 1        |
| Condensador     | 4,7uF         | 1        |
| Bobina          | 5.5mH         | 1        |
| Diodo           | MUR 460       | 1        |
| Diodo           | 1N5404        | 1        |
| Rectificador    | 6A 200V       | 1        |
| Regulador       | De 15v a 3.3v | 1        |
| Driver          | MC33152P      | 1        |
| Transistor      | IRF 740       | 1        |
| OPAM            | 23741         | 1        |
| Resistencia     | 0.22Ω         | 1        |
| Resistencia     | 240KΩ         | 2        |
| Resistencia     | 270KΩ         | 2        |
| Resistencia     | 1,2KΩ         | 2        |
| Resistencia     | 10KΩ          | 1        |
| Resistencia     | 15Ω           | 2        |
| Resistencia SMD | 15kΩ          | 2        |
| Resistencia SMD | 4,7KΩ         | 2        |
| Resistencia SMD | 3,3KΩ         | 2        |
| Resistencia SMD | 1KΩ           | 3        |

En la Tabla 2, se observa detenidamente cada componente usado para la implementación del convertidor, cabe resaltar que se tiene información de la referencia y la cantidad de componentes necesarios para el ensamble del circuito PCB.

#### 9.4.3.1. Montaje real del circuito PCB

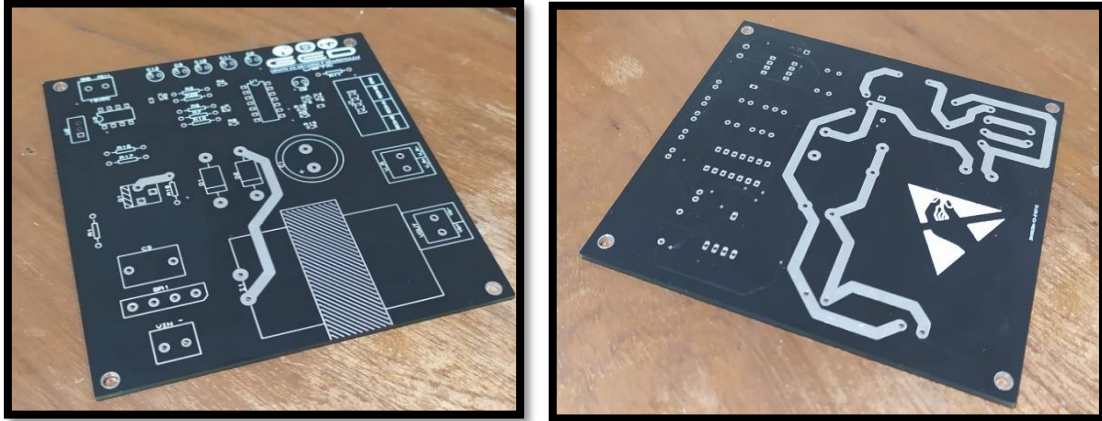


Figura 15: Circuito impreso, capa superior, cara inferior. Fuente: Elaboración propia.

Como se observa en la Figura 15, culminada la etapa de diseño del circuito impreso, se procede a realizar el ensamble de los componentes de la planta, también se procede a hacer uso de la tarjeta DSP C2000 la cual servirá para implementar los sistemas de control, dicho lo anterior en la Figura 16 se observa el circuito ensamblado:

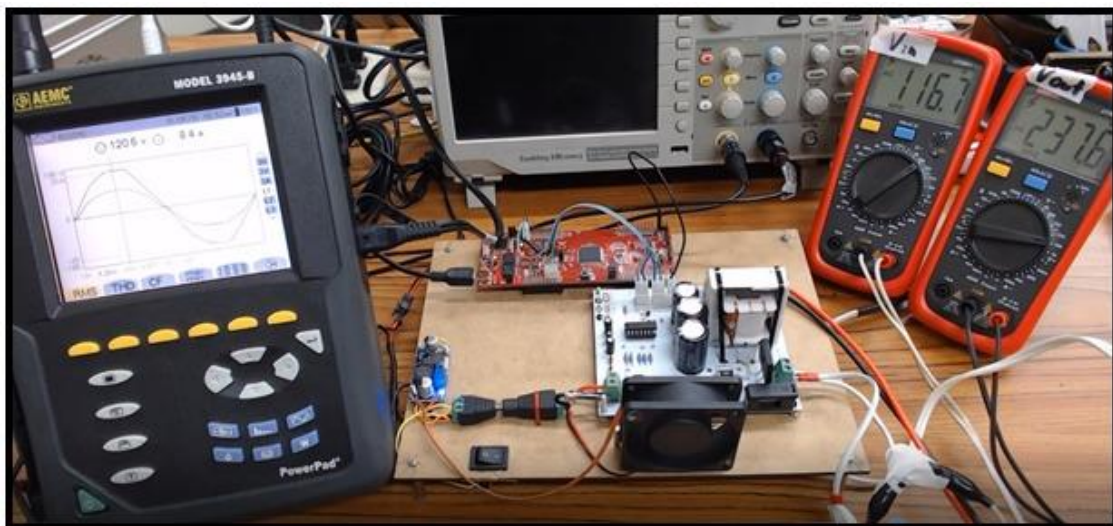


Figura 16: Implementación real de la planta y sistema de control. Fuente: Elaboración propia.

---

## Capítulo 10

# Resultados

Una vez implementado el control digital en la tarjeta C2000 de Texas Instruments se procede a sintonizar el lazo de corriente con los parámetros obtenidos, primeramente, por el método de sintonización auto tuner PID del software de Simulink, y por último con los parámetros obtenidos del algoritmo Multi-PSO. Por cada sintonización del PID se procede a realizar la lectura y visualización de las variables de entrada con la intención de obtener resultados. Para la lectura de datos se hace uso del Analizador de Calidad de Energía Ecléctica Trifásica PowerPaD 3945-B, el cual es el modelo con el que cuenta la Universidad Santo Tomás. En esta oportunidad se realiza la lectura de la red monofásica de entrada del convertidor elevador, es por eso por lo que solamente es necesario hacer uso de una de las pinzas de voltaje y una de las pinzas milimétricas de corriente con las que cuentan el equipo.

### 10.1. Simulaciones

De acuerdo con el modelo propuesto, se implementa una simulación en la cual se pueda ver el funcionamiento del algoritmo MPSO, en la búsqueda de constantes para el controlador PID. En un principio solamente se observará el controlador de referencia ( $PID_{REF}$ ), y su comportamiento con respecto a variación de carga en el sistema, en donde se parte de una carga de  $500\Omega$  y progresivamente aumenta su valor hasta  $2500\Omega$  (500, 625, 833,33, 1000, 2500). de igual forma el controlador encontrado por MPSO ( $PID_{MPSO}$ ) tendrá variaciones de carga para verificar su efectividad en el rechazo a perturbaciones.

#### 10.1.1. Controlador Sintonizado por Métodos Comunes

El controlador sintonizado por métodos comunes hace parte del trabajo sobre el cual se desarrolló este proyecto, los valores de constantes del controlador PID fueron halladas con ayuda del software de MATLAB mediante la herramienta de PID Tuner, este controlador va a ser el controlador de referencia sobre el cual se basa el espacio de búsqueda del  $PID_{MPSO}$ . En el caso de ambos controladores se realizan simulaciones con respecto a voltajes y corrientes, tanto de entrada como de salida, para el caso de este controlador es posible verificar como a medida que el sistema cambia ( $\uparrow R$ ), la forma de onda y por consiguiente el factor de potencia se ve afectado.

A continuación se va a describir la configuración de la simulación; existe un total de 5 cargas distintas a probar en el circuito a lo largo de la simulación ( $500\Omega$ ,  $625\Omega$ ,  $833\Omega$ ,  $1000\Omega$  y  $2500\Omega$ ), estas cargas se activaran en el circuito cada 10 segundos, por lo que el tiempo estimado de simulación es de 50 segundos, en este caso el controlador sintonizado por métodos comunes no puede ser cambiado o mejorado a lo largo de toda la simulación, por



lo que no existe ningún tipo de entrenamiento para este. Las constantes del controlador PID que se encuentra en el circuito son:  $K_p = 0.504$ ,  $K_i = 3000$ ,  $K_d = 4.1e - 6$ ,  $N = 4.8e6$ .

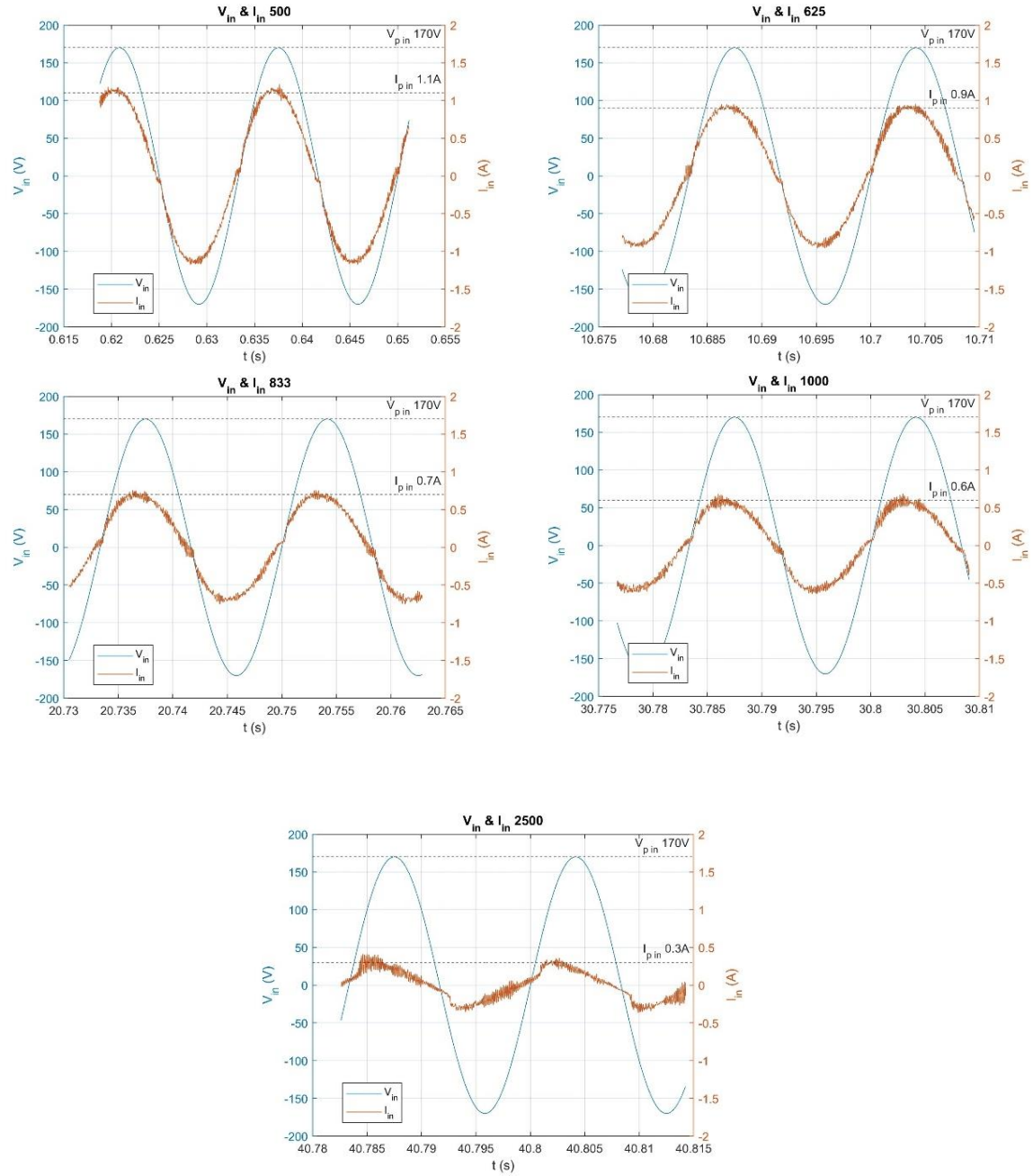


Figura 17: Voltaje de Entrada y Corriente de Entrada

El voltaje y la corriente de entrada se visualizan en la línea de alimentación del convertidor, como se puede observar en la Figura 17, a medida que la carga cambia, la corriente en el convertidor baja, hasta que el sistema cambia lo suficiente para que la forma de onda pierda su factor de potencia, sin embargo, este cambio es de acuerdo a las constantes del controlador, por lo que cuando la carga cambia, el sistema debería readaptarse, en este caso, el sistema no puede hacer eso, ya que necesitaría de un procedimiento matemático



para mejorar a medida que avanza el tiempo, y como no fue implementado de esta forma, este controlador no puede cambiarse, sin embargo esta problemática es abordada y resuelta en el controlador bio-inspirado.

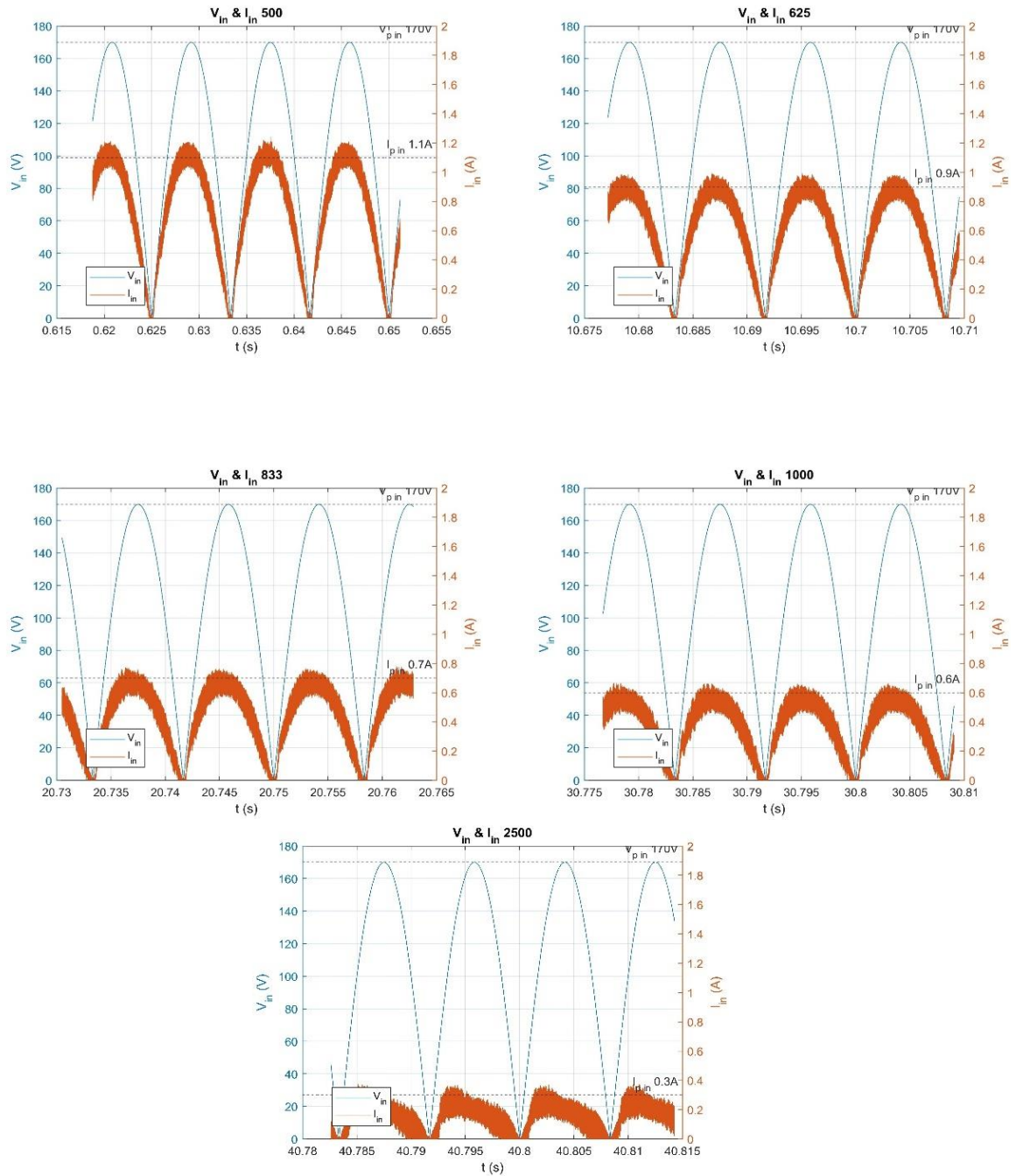


Figura 18: Voltaje Rectificado y Corriente Rectificada o Corriente en el Inductor

De igual forma que en la gráfica anterior, en la Figura 18, se puede visualizar el cambio en la corriente con respecto al cambio de la carga a lo largo del tiempo, sin embargo, la medición de la corriente en la bobina deja ver una cantidad de ruido significativa, este de acuerdo con el efecto del MOSFET, pasando de conexión a desconexión continuamente. Aunque esta distorsión de la señal puede ser corregida de acuerdo con los parámetros del controlador, no desaparece del todo.

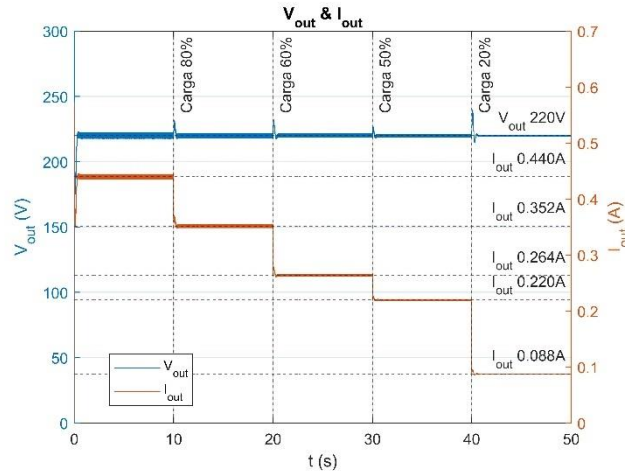


Figura 19: Voltaje de Salida y Corriente de Salida

El voltaje de salida se ve afectado cada vez que existe una perturbación, sin embargo, es trabajo del controlador de voltaje hacer que el sistema se adapte para mantener la referencia a  $220V$ , esto no ocurre con la corriente, pues al cambiar la carga obligatoriamente esta va a descender, por lo mismo, aunque el voltaje se mantenga la potencia decaerá igualmente.

En la Figura 19 se observa el cambio de corriente y el rechazo a perturbaciones del controlador de voltaje. Los porcentajes de carga de la gráfica hacen referencia al número de interruptores que son accionados para cambiar la carga, como se explica en la sección del diseño de carga, pues en total son 10 interruptores cada uno de  $5K\Omega$  al activar 2 la carga pasa de  $500\Omega$  a  $625\Omega$ , lo cual equivale al 80% de los interruptores manteniendo su posición.

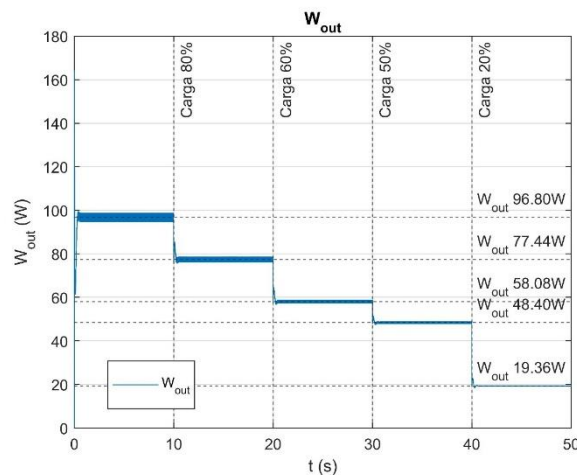


Figura 20: Potencia de Salida.

El convertidor fue diseñado con una potencia de  $100W$  en este caso en la Figura 20 solamente se evidencian  $96.8W$  a plena carga debido a que la simulación está hecha con las cargas del sistema real, que van desde los  $500\Omega$  hasta  $5000\Omega$ , es por esta razón que la potencia está por debajo, ya que si se realiza con la carga para la que fue diseñado el sistema ( $487\Omega$ ), entonces se podrían evidenciar plenamente los  $100W$ .

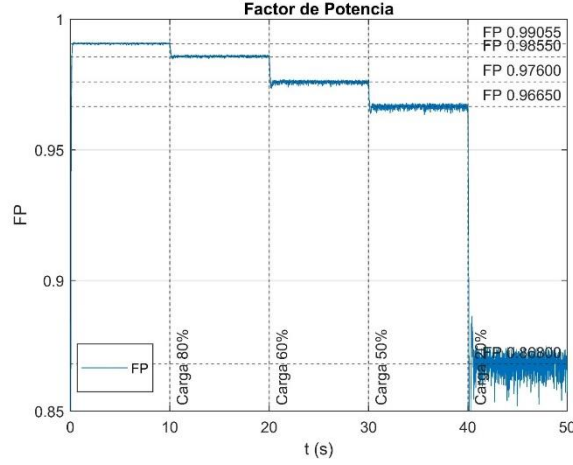


Figura 21: Factor de potencia

El sistema cambia de con cada nueva carga que se coloca en el sistema, de esta forma, el controlador ya no está sintonizado para ese sistema, por lo que el factor de potencia se verá afectado como se puede ver en la Figura 21, el sistema tiene un rendimiento por encima del 90% hasta el punto en donde la carga ha cambiado hasta el doble de su valor, sin embargo, cuando el sistema llega a **2500Ω**, este decae abruptamente hasta un poco mas del 85% de eficiencia.

#### 10.1.1. Controlador Sintonizado por Algoritmo Bio- inspirado

Este controlador a diferencia de la versión clásica posee un algoritmo de sintonización basada en el comportamiento descrito del MPSO, donde diferentes enjambres buscaran en un espacio reducido los valores de constantes que hagan que el sistema mejore su comportamiento con respecto al seguimiento de la referencia.

La configuración de esta simulación es la siguiente: 10 segundos para cada entrenamiento, debido a esto, las perturbaciones, deben realizarse cada 12 segundos, donde los primeros 0.5 segundos de la perturbación es para que el sistema se estabilice con el nuevo valor de carga, es decir, la etapa transitoria de la perturbación, esto debido a que no es recomendable entrenar al sistema durante este periodo, puesto que los resultados podrían verse afectados.

Tabla 3: Configuración Entrenamiento

| Parámetros           | Valor                                                            |
|----------------------|------------------------------------------------------------------|
| <i>PIDN</i>          | $K_p = 0.504, K_i = 3000, K_d = 4.1e - 6, N = 4.8e6.$            |
| <i>Lim</i>           | $K_p = 0 - 10, K_i = 0 - 10e3, K_d = 0 - 1e - 6, N = 0 - 100e8.$ |
| <i>C</i>             | $C_1 = 0.05 C_2 = 0.35 C_3 = 0.75$                               |
| <i>d<sub>t</sub></i> | 0.5                                                              |
| <i>Particulas</i>    | 10                                                               |
| <i>Enjambres</i>     | 5                                                                |
| <i>Epocas</i>        | 50                                                               |

De acuerdo con los parámetros de entrenamiento establecidos en la tabla 3, se realiza un entrenamiento para el sistema, en este caso el convertidor inicia con un controlador base, el cual es el controlador de referencia, pasados 0.5 segundos de la variación de corriente en el circuito, el sistema comenzara a realizar un entrenamiento, con este el sistema evaluara los controladores del MPSO con el objetivo de encontrar alguno que mejore el sistema, dependiendo del comportamiento del enjambre este comenzara a acercarse a la respuesta.

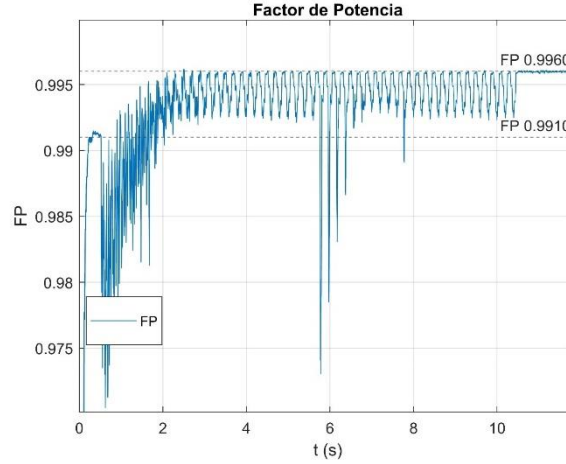


Figura 22: Factor de Potencia

Durante el entrenamiento de 10 segundos, el sistema tiene diferentes controladores los cuales mejoran a lo largo del entrenamiento, como se puede ver en la Figura 22, los primeros controladores por lo general tienen un comportamiento por debajo de la inicial, es decir el controlador de referencia, pero a medida que el entrenamiento avanza, el sistema mejora hasta el punto de mejorar el controlador inicial.

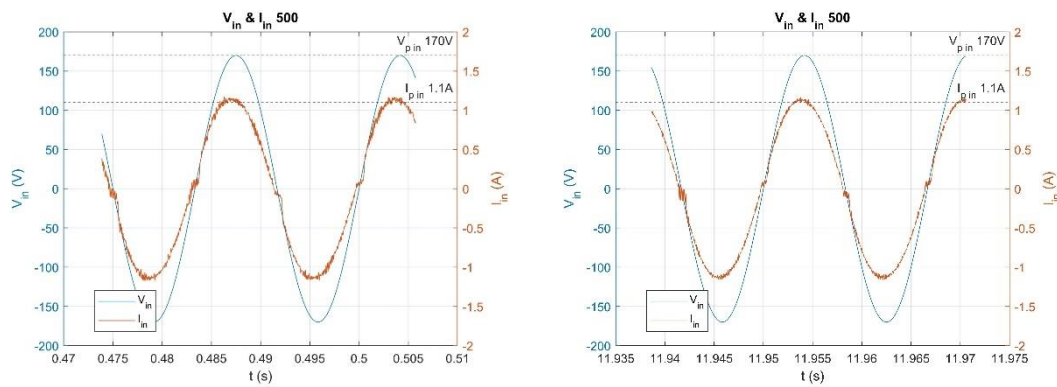


Figura 23: Controlador de referencia y controlador MPSO

En la Figura 23 se observa la diferencia de los controladores al seguir la referencia, en donde antes del entrenamiento, en la primera imagen, el sistema tiene una buena forma de onda, pero tiene un cierto nivel de desfase y distorsión en los puntos cercanos a 0, esto

disminuye en buena manera para la segunda imagen, en donde el entrenamiento fue completado y el controlador que tuvo el mejor comportamiento fue seleccionado.

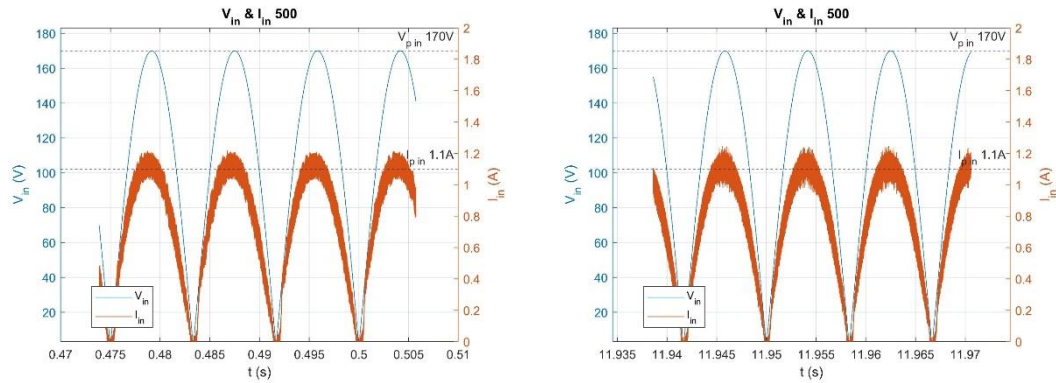


Figura 24: Voltaje Rectificado y Corriente Rectificada o Corriente en el Inductor

La corriente en la bobina parece no sufrir un gran cambio, sin embargo, en el desfase se nota más el cambio de la señal, también como se evidencia en la Figura 24, el valor de la corriente supera el amperio, pues debido a pérdidas de potencia a lo largo del circuito, el sistema por lo general necesita un poco más del valor de la referencia para mantener su voltaje.

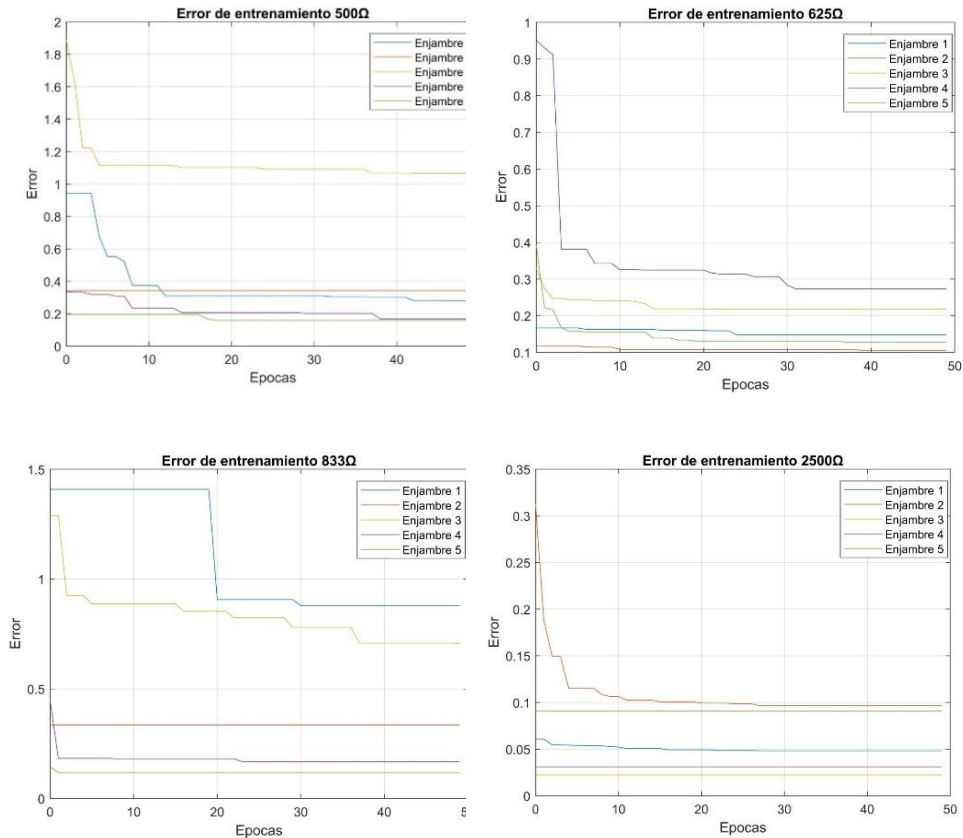


Figura 25: Entrenamientos en variaciones de carga

Como se mencionó al comienzo de la sección, un controlador sin un algoritmo de sintonización, no puede variar a lo largo del tiempo, por lo que está sujeto a tener un mal rendimiento frente a una planta que presenta variaciones, como en el caso de las perturbaciones realizadas al convertidor, es por esta razón que se destina un elemento de detección en los cambios de corriente del sistema, ya que como se puede ver en la figura 17, los diferentes valores de corriente pueden ser interpretados como cambios en la carga del convertidor, de esta forma se tiene un delta de corriente, el cual cuando se supera, inicia el entrenamiento, se debe tener cuidado pues si el delta es demasiado pequeño, los entrenamientos prácticamente no se detendrán, uno detrás de otro, por lo que una variación de  $0.2A$  se considera idónea. En el caso de la Figura 25 no existe un entrenamiento para  $1000\Omega$  de carga, puesto que la variación de corriente con respecto a  $833\Omega$  no es lo suficientemente grande para ser considerado un cambio de sistema, es por esta razón que solamente se pueden observar los entrenamientos para 4 variaciones.

Cuando el sistema entrena a plena carga, es decir, para el modelo el cual fue preparado, se puede dejar el primer controlador encontrado, es decir, no realizar más entrenamientos, esto se realiza con el fin de comprobar el rechazo a perturbaciones de un solo controlador con respecto al controlador que se tenía inicialmente sintonizado por métodos clásicos matemáticos.

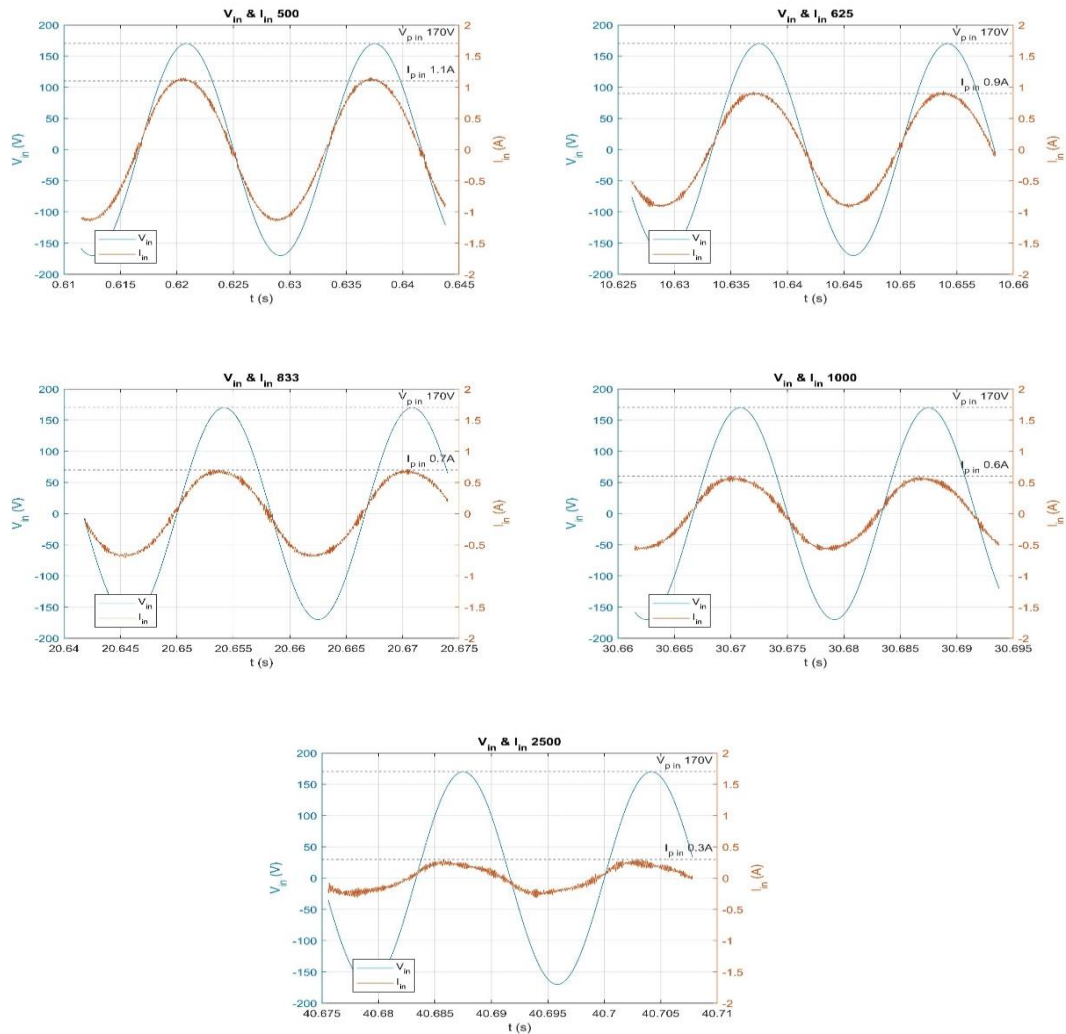


Figura 26: Voltaje de Entrada y Corriente de Entrada



En este caso en la Figura 26 muestra las perturbaciones sobre el convertidor, en este caso se observa que solamente el controlador entrenado está actuando sobre el circuito, con el fin de validar el rechazo a perturbaciones del controlador. Se puede observar que la referencia de la corriente en la entrada mantiene la forma de onda incluso al 20% de la carga total por lo que el controlador tiene un factor de potencia superior.

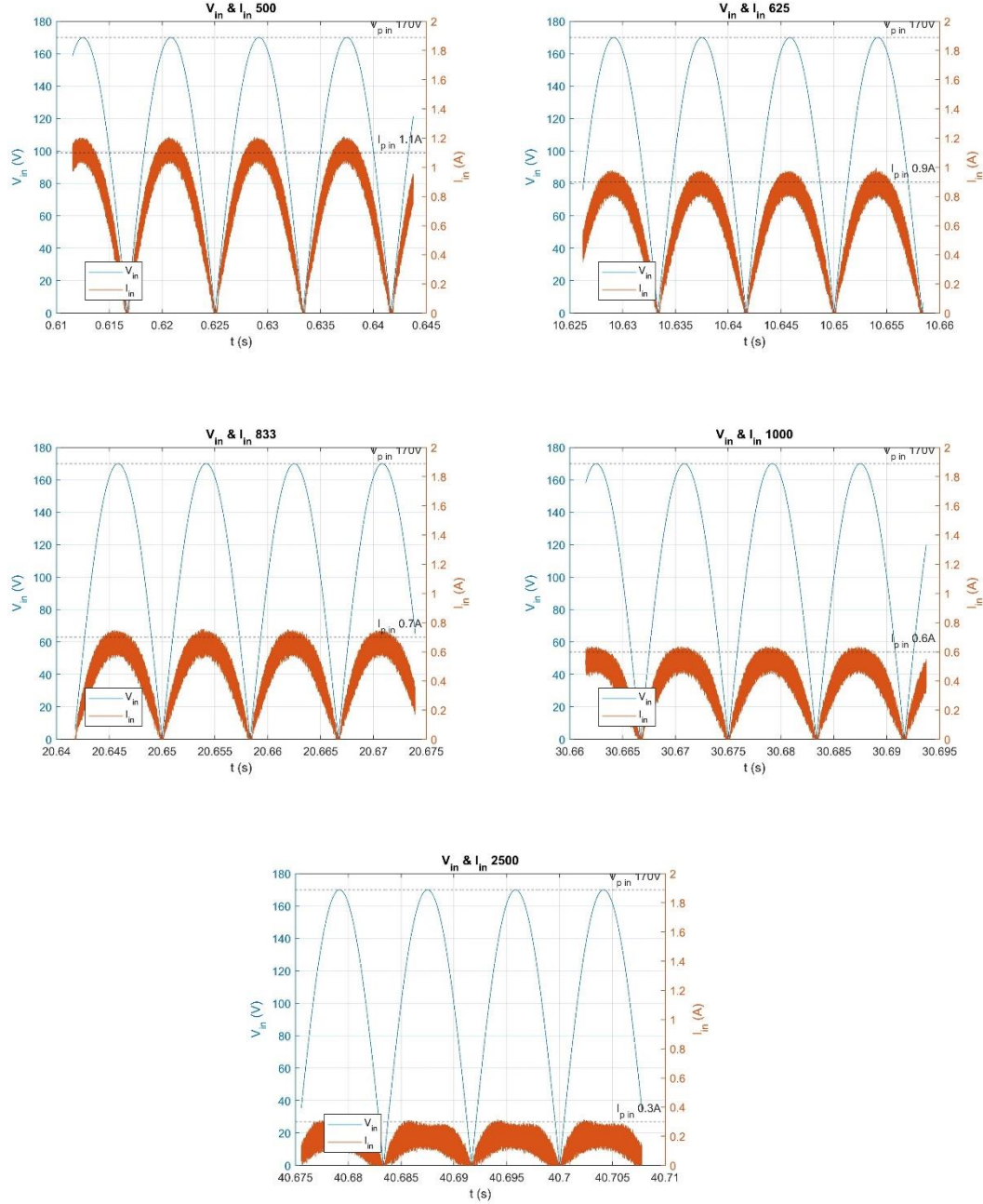


Figura 27: Voltaje de Entrada y Corriente de Entrada

El voltaje de entrada de la corriente de entrada rectificadas que se observan en la Figura 27, se tiene una forma de onda similar en comparación al controlador sintonizado por métodos convencionales.

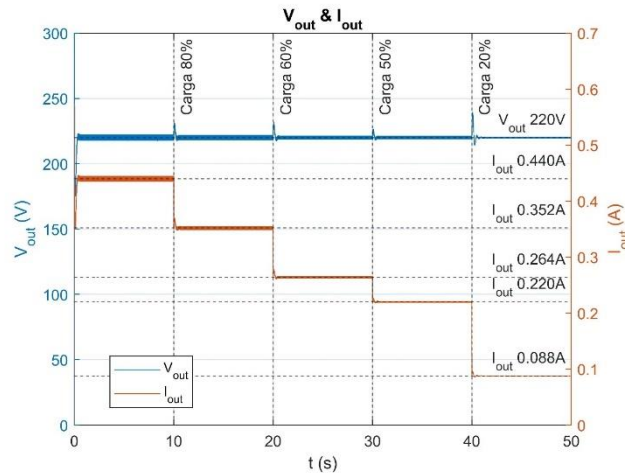


Figura 28: Voltaje de Salida y Corriente de Salida

Al igual que con el controlador sintonizado por métodos convencionales, se realiza un análisis de la corriente en función de la variación de las cargas, cómo se puede evidenciar el cambio en la corriente entre el 50 y el 60% de la carga es menor al Delta de corriente, por lo que durante la carga del 50% el sistema no realizará un entrenamiento.

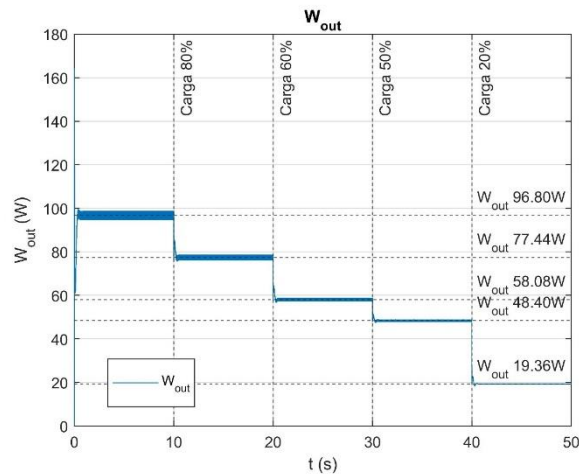


Figura 29: Potencia de Salida

La Figura 29 muestra la potencia de salida el convertidor, al igual que con el controlador de referencia el sistema pierde potencia a medida que varía la carga del convertidor, en este caso no existe una diferencia notable.



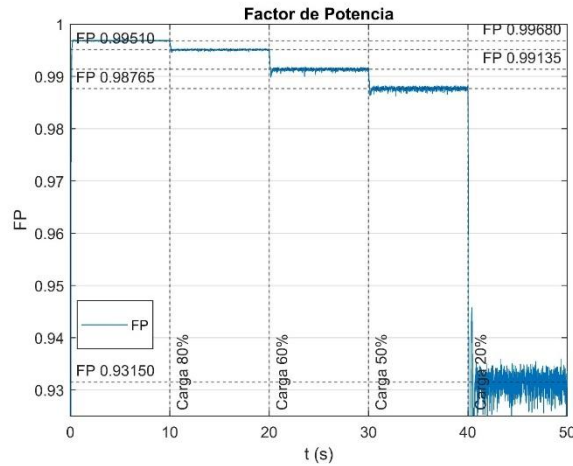


Figura 30: Factor de Potencia

En la Figura 30, se observa cómo el controlador sintonizado por MPSO tiene un comportamiento superior con respecto al controlador de referencia o el controlador sintonizado por métodos convencionales, en este caso se observa que incluso con una carga al 50% del valor original el factor de potencia se mantiene por encima del 99%, cuando existe una caída al 20% del valor de la carga el factor de potencia sigue manteniendo un valor por encima del 90% a diferencia del controlador de referencia.

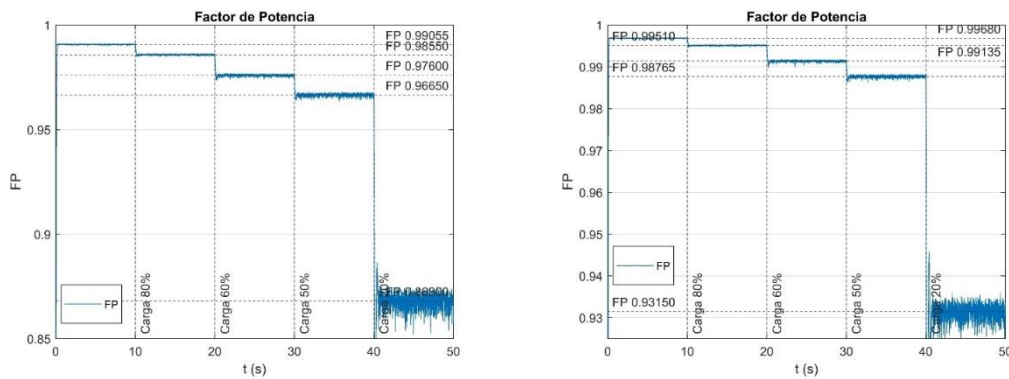


Figura 31: Comparación Factor de Potencia

En la primera imagen de la figura 31 se evidencia el factor de potencia que corresponde al controlador sintonizado por métodos clásicos, este tiene un decaimiento en su eficiencia considerable con cada nueva carga, el sistema, aunque se mantiene por encima del 95% de eficiencia hasta el 50% de la carga inicial, decae hasta cerca del 85% de eficiencia en el sistema. Por otra parte, el controlador que fue encontrado por el MPSO tiene una eficiencia superior al 90% en todas las variaciones y superior al 99% en todas las variaciones del 50% de la carga hasta plena carga. Por lo que el algoritmo cumple efectivamente con mejorar el sistema.

## 10.2. Resultados de la Implementación Real.

### 10.2.1. Controlador Sintonizado por métodos comunes

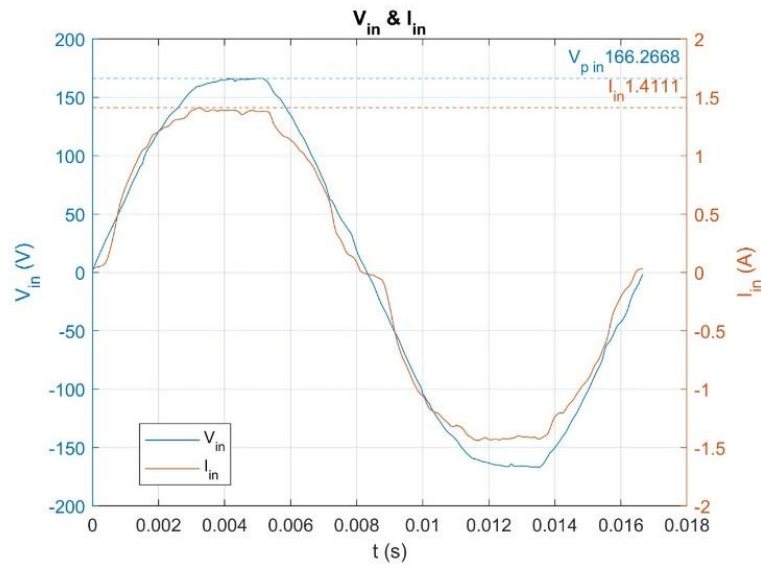


Figura 32: Formas de onda de voltaje y corriente de entrada Controlador Clásico. Fuente: elaboración propia.

La Figura 32, representa gráficamente las formas de onda del voltaje y la corriente de entrada, nótese que en la gráfica aparece la escala y los valores pico de cada señal. La figura 32 fue realizada con el software de Matlab, para su visualización se usaron datos reales generados por el analizador de redes eléctricas.  $V_{RMS} = 120,8 V$ ,  $I_{RMS} = 1,069 A$ ,  $V_p = 166,3 V$ ,  $I_p = 1.41 A$ .

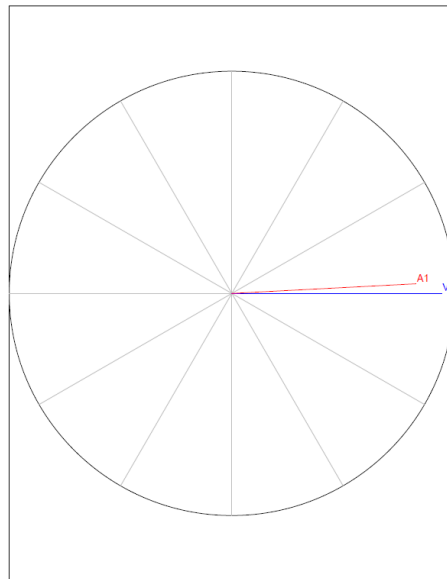


Figura 3333: Desfase de Corriente y Voltaje  $3^\circ$

En la Figura 33, se observa la representación gráfica del diagrama fasorial, esta representación fue generada y exportada por el analizador de redes eléctricas. La figura 33, permite visualizar el relacionamiento entre las variables eléctricas de voltaje de entrada y de corriente de entrada ( $V_{in}$ ,  $I_{in}$ ) también permite visualizar el desfase que tiene la corriente respecto al voltaje, en esta oportunidad evaluando el resultado se logra determinar que gracias al sistema de control PID para el lazo de corriente, sintonizado con los parámetros de auto tune se obtiene un desfase de  $3^\circ$ .

### 10.2.2. Controlador Sintonizado por Algoritmo Bio-inspirado

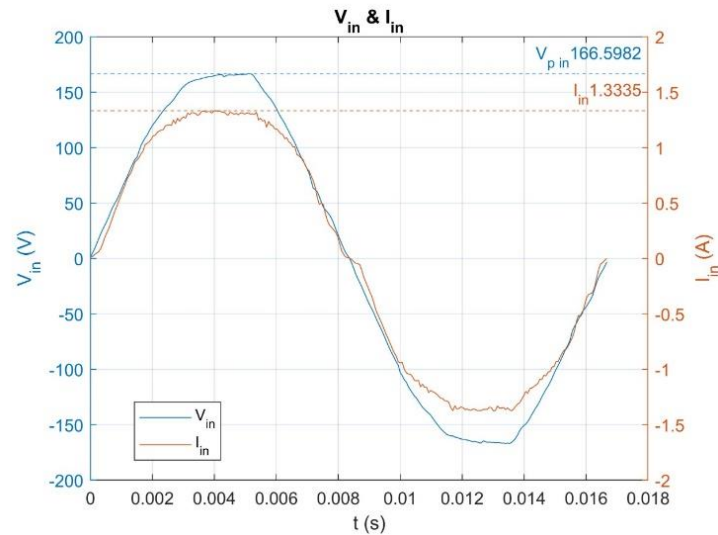


Figura 34: Formas de onda de voltaje y corriente de entrada Controlador Clásico. Fuente: elaboración propia.

La Figura 34, representa gráficamente las formas de onda del voltaje y la corriente de entrada, nótese que en la gráfica aparece la escala y los valores pico de cada señal. La Figura 34 fue realizada con el software de Matlab, para su visualización se usaron datos reales generados por el analizador de redes eléctricas.  $V_{RMS} = 120,7 V$ ,  $I_{RMS} = 1,072 A$ ,  $V_p = 166,6 V$ ,  $I_p = 1.33 A$ .

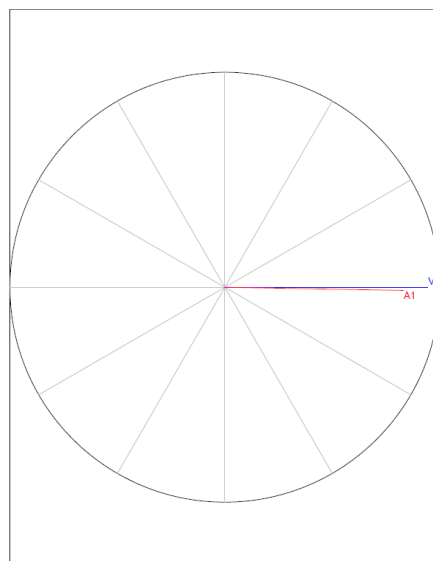


Figura 3535: Desfase de Corriente y Voltaje  $1^\circ$

En la Figura 35, se observa la representación gráfica del diagrama fasorial, esta representación fue generada y exportada por el analizador de redes eléctricas. La Figura 35, permite visualizar el relacionamiento entre las variables eléctricas de voltaje de entrada y de corriente de entrada ( $V_{in}$ ,  $I_{in}$ ), también permite visualizar el desfase que tiene la corriente respecto al voltaje, en esta oportunidad evaluando el resultado se logra determinar que gracias al sistema de control PID para el lazo de corriente con parámetros de sintonización hallados gracias al Multi-PSO se obtiene un desfase de  $1^\circ$ .

### 10.2.3. Descripción de resultados

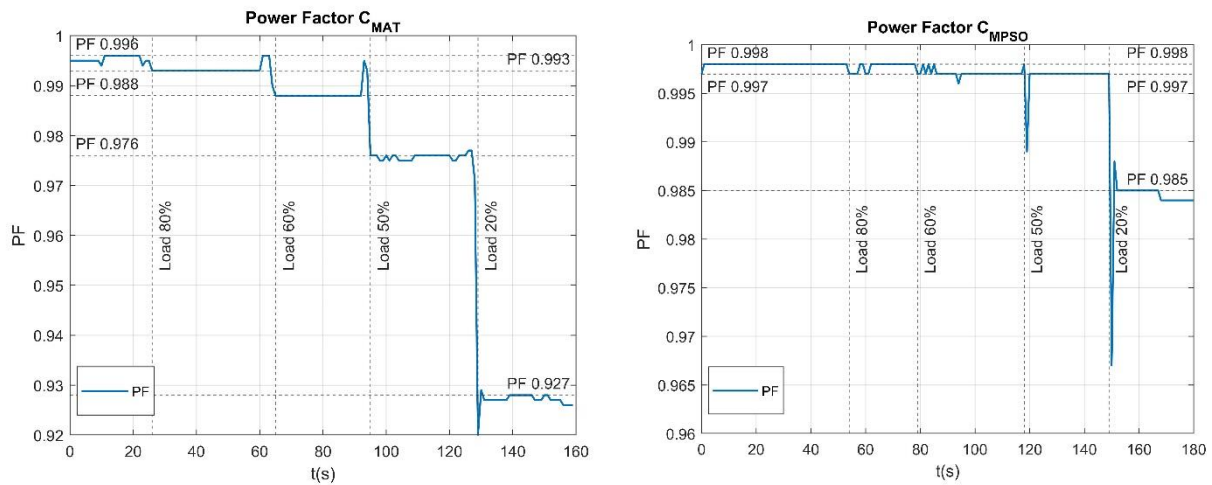


Figura 36: Comparación Factor de Potencia, Implementación Real. Fuente: elaboración propia.

En la primera imagen de la Figura 36 se evidencia el factor de potencia que corresponde al controlador sintonizado por métodos clásicos, este tiene un decaimiento en su eficiencia considerable con cada nueva carga, el sistema, aunque se mantiene por encima del 95% de eficiencia hasta el 50% de la carga inicial, decae hasta cerca del 85% de eficiencia en el sistema. Por otra parte, el controlador que fue encontrado por el MPSO tiene una eficiencia superior al 90% en todas las variaciones y superior al 98% en todas las variaciones del 50% de la carga hasta plena carga. Por lo que el algoritmo cumple efectivamente con mejorar el sistema.

Tabla 4, Datos obtenidos del PF con analizador de redes

| PF      |       |
|---------|-------|
| Clasico | 0,996 |
| MPSO    | 0,999 |

En la Tabla 4, se relacionan los valores de factor de potencia (PF) obtenidos por el Analizador de Calidad de Energía Ecléctica Trifásica PowerPaD 3945-B. La intención de usar este equipo se debe a que es posible visualizar el valor del factor de potencia hasta con tres cifras significativas, sin duda esto nos permite observar el desempeño entre ambos métodos de sintonización. Se observa cómo el PF usando los valores obtenidos por el

---

MPSO obtiene una mejora en la tercera cifra, lo que da a entender que el algoritmo logró hallar unos valores capaces de sintonizar el lazo de corriente del controlador **PID** del convertidor elevador.

---

## Capítulo 11

# Conclusiones

- Mediante el algoritmo bio-inspirado PSO y la versión utilizada en este trabajo, MPSO, es posible mejorar el factor de potencia del convertidor AC/DC en topología elevador con un valor inicial de 0.996 a 0.999.
- Los límites en las constantes de aprendizaje del algoritmo de búsqueda y en las constantes del controlador, son decisivas al momento de realizar la búsqueda en el espacio de posibles candidatos para la sintonización del controlador.
- El algoritmo en un espacio de búsqueda limitado, trata de obtener una mejor solución con respecto al controlador inicial, por lo que la exploración no es prioridad, ya que se está dando un espacio limitado desde el principio, de esta forma el algoritmo se enfoca en su comunicación.
- En el caso de los métodos de sintonización basados en inteligencia artificial se tiene la ventaja del aprendizaje sin necesidad de un modelo matemático, pero también la problemática de desconocer el comportamiento de la planta.

---

## Capítulo 12

# Trabajos Futuros

Basado en el trabajo presentado en este documento, es importante no ignorar algunos puntos que permitan desarrollar nuevos proyectos para las personas que opten por complementar el proyecto de este documento.

- Como se evidencia en la tabla 1, existen gran variedad de algoritmos bio-inspirados, es así como se plantea la posibilidad de adaptar algunos de ellos para realizar la sintonización del controlador **PID**, para así comparar los resultados con los valores obtenidos en este proyecto.
- De igual manera será interesante ver cómo se realiza la propuesta para usar otro método de aprendizaje de máquina para aplicar un modelo el cual sea capaz de sustituir en controlador **PID**, quizás haciendo uso de redes neuronales se pueda cumplir con ese objetivo.
- Sera interesante evaluar el desempeño de convertidor elevador, es por ello por lo que se pretende usar como fuente de alimentación para un convertidor forward.

---

## Capítulo 13

# Anexos

### 13.1. Código de MATLAB SIMULINK

---

```
function [pidn, y]= MDPSO(e,bufferSize,current)
    persistent buffer pidn_value
    persistent counter_buffer counter_swarms
    persistent counter_particles counter counter_freeze
    persistent epoch current_rms
    persistent swarms f fBest gBest fRecord trigger

    num_particles = 10;
    fs = 50000;
    num_swarms = 5;
    num_dimensions = 4;
    %num_dimensions = 3;
    epochs = 50;

    if (isempty(buffer))
        buffer = zeros(1,bufferSize);
    end
    if (isempty(epoch))
        epoch = 1;
        disp([' trainnig time ' sprintf('%0.5g',...
            num_particles*num_swarms*epochs*bufferSize/fs) ' ']);
    end
    if (isempty(current_rms))
        current_rms = current;
        trigger = 0;
    end

    if (isempty(counter))
        counter = 0;
    end

    if (isempty(counter_buffer))
        counter_buffer = 0;
    end
    if (isempty(counter_swarms))
        counter_swarms = 1;
    end
    if (isempty(counter_particles))
        counter_particles = 1;
    end
    if (isempty(pidn_value))
        pidn_value = [0.504,3000,4.1e-6,4.8e6];%controlador Ing.Jhon Bayona
    end
    if (isempty(swarms))
        particles = cell(num_swarms,num_particles);
        f = zeros(1:num_swarms);
        fBest = zeros(1,num_swarms);
        gBest = zeros(num_swarms,num_dimensions);
        fRecord = zeros(num_swarms,epochs);
        counter_freeze = zeros(1,num_swarms);
```



```

Lim = [0 10;
       0 10e3;
       0 1e-6;
       0 100e8];

xLim = zeros(num_dimensions,2);

for swarm = 1:num_swarms
    for particle = 1:num_particles
        for dimension = 1:num_dimensions
            xLim(dimension,:) = randsample(linspace...
            (Lim(dimension,1), Lim(dimension,2),1000),2);
        end
        particles{swarm,particle} = particula(...
            xLim(:,1)',xLim(:,2)');
    end
end
particles{1,1} = particula(pidn_value, xLim(:,2)');
swarms = particles;
end

if current > (current_rms + 0.2) || current < (current_rms - 0.2)
    current_rms = current;
    if counter ~= 25000
        disp([' current ' sprintf('%0.5g', current) '']);
        epoch = 1;
        trigger = 1;
    end
end

if (counter < 25000 && trigger == 1)
    counter = counter + 1;
end

if (epoch <= epochs && counter == 25000)
    if (e ~= 0)
        counter_buffer = counter_buffer + 1;
        pidn_value = swarms{counter_swarms,counter_particles}.x;
    end

    for i=1:bufferSize-1
        buffer(i) = buffer(i+1);
    end

    if (counter_buffer == bufferSize)
        counter_buffer = 0;
        if (counter_swarms <= num_swarms)
            if (counter_particles <= num_particles)
                %disp(' ');
                %disp(['P : ' sprintf('%0.5g', pidn_value(1))...
                    '% I : ' sprintf('%0.5g', pidn_value(2)) ' D : '...
                    %sprintf('%0.5g', pidn_value(3)) ' N : '...
                    %sprintf('%0.5g', pidn_value(4))]);
                e = error_data('e^2', buffer);
                %disp(['error : ' sprintf('%0.5g', e) ' swarm : '...
                    %sprintf('%0.5g', counter_swarms) ' particle : '...
                    %sprintf('%0.5g', counter_particles) ' epoch : '...
                    %sprintf('%0.5g', epoch) ' of '...

```

```

        %sprintf('%0.5g', epochs) ' ');
    swarms{counter_swarms, counter_particles}.evaluate(...
        error_data('e^2', buffer));
    if (e <= 0)
        if (e < min(fBest))
            counter_particles = num_particles;
            counter_swarms = num_swarms;
            epoch = epochs;
        end
    end
    counter_particles = counter_particles + 1;
    if (counter_particles > num_particles)
        counter_particles = 1;
        counter_swarms = counter_swarms + 1;
        if (counter_swarms > num_swarms)
            % update_particles
            fBest_ = fBest;
            [swarms, f, fBest, gBest, fRecord] = ...
                update_particles(num_particles,...
                    num_swarms,...
                    swarms,...
                    epochs,...
                    epoch,...
                    f,...
                    fBest,...
                    gBest,...
                    fRecord);

            disp(' Best Values : ');
            disp(fBest);
            for swarm = 1:num_swarms
                if (fBest(swarm) == fBest_(swarm))
                    counter_freeze(swarm) = ...
                        counter_freeze(swarm) + 1;
                    if (counter_freeze(swarm) == epochs/2)
                        disp(['Swarm Freeze :'...
                            sprintf('%0.5g', swarm)]);
                        if (min(fBest) ~= fBest(swarm))
                            disp(' Setting New Positions');
                            Lim = [0 10;
                                0 10e3;
                                0 1e-6;
                                0 100e8];
                            for particle = ...
                                1:num_particles
                                    for dimension = ...
                                        1:num_dimensions
                                            xLim(dimension,:) = ...
                                                randsample(...
                                                    linspace(Lim...
                                                        (dimension,1),...
                                                        Lim(dimension,2)...
                                                            ,1000),2);
                                            end
                                            swarms{swarm,particle}.x...
                                                = xLim(:,1)';
                                            swarms{swarm,particle}.v...
                                                = xLim(:,2)';
                                        end
                                    end
                                end
                                counter_freeze(swarm) = 0;
                            end
                        else
                            counter_freeze(swarm) = 0;
                        end
                    end
                disp(' Freeze Counter ');
            end
        end
    end
end

```

```

disp(counter_freeze);
counter_swarms = 1;
epoch = epoch + 1;
if (epoch > epochs)
    disp('Trainning Complete');

    counter = 0;
    trigger = 0;

    f = zeros(1:num_swarms);
    fBest = zeros(1,num_swarms);
    gBest = zeros(num_swarms,num_dimensions);
    fRecord = zeros(num_swarms,epochs);
    pidn_value = swarms{1,1}.x;

    Lim = [0 10;
           0 10e3;
           0 1e-6;
           0 100e8];

    xLim = zeros(num_dimensions,2);
    for swarm = 1:num_swarms
        for particle = 1:num_particles
            for dimension = 1:num_dimensions
                xLim(dimension,:)...
                    = randsample(linspace...
                                (Lim(dimension,1),...
                                 Lim(dimension,2),100),2);
            end
            swarms{swarm,particle}.x...
                = xLim(:,1)';
            swarms{swarm,particle}.v...
                = xLim(:,2)';
            swarms{swarm,particle}.pBest...
                = xLim(:,1)';
            swarms{swarm,particle}.fBest...
                = Inf;
            swarms{swarm,particle}.f...
                = Inf;

            end
        end
        swarms{1,1}.x = pidn_value;
        disp(['P :'. ...
              sprintf('%0.5g', pidn_value(1))...
              ' I :'. ...
              sprintf('%0.5g', pidn_value(2))...
              ' D :'. ...
              sprintf('%0.5g', pidn_value(3))...
              ' N :'. ...
              sprintf('%0.5g', pidn_value(4))]);

        %
        %
        %
        %
        %
        disp(['P :'. ...
              sprintf('%0.5g', pidn_value(1))...
              ' I :'. ...
              sprintf('%0.5g', pidn_value(2))...
              ' D :'. ...
              sprintf('%0.5g', pidn_value(3))]);
        %disp([' current '. ...
              %sprintf('%0.5g', current) ']);

    end
else
    end
else
    end
else
    end
else
    end

```

```

        end
    else
        end
    else
        end

    pidn = pidn_value;
    buffer(bufferSize) = e;
    y = counter;
else
    pidn_value = swarms{1,1}.x;
    pidn = pidn_value;
    y = counter;
end
end
end

function [swarms, f, fBest, gBest, fRecord]...
    = update_particles(num_particles,...
        num_swarms,...
        swarms,...
        epochs,...
        epoch,...
        f,...
        fBest,...
        gBest,...
        fRecord)
%inertia_decrease_rate = (0.65*epoch)/epochs;
learning_constants = [0.05 , 0.35 , 0.75, 0];
dt = [0.5 0.5 0.5 0.5];

for swarm = 1:num_swarms
    f(swarm) = Inf;
end
for swarm = 1:num_swarms
    for particle = 1:num_particles
        if swarms{swarm,particle}.fBest < f(swarm)
            f(swarm) = swarms{swarm,particle}.fBest;
            fBest(swarm) = f(swarm);
            gBest(swarm,:) = swarms{swarm,particle}.pBest;
        end
    end
end

Best_index = fBest == min(fBest);
pBestSwarm = 0;
for index = 1:num_swarms
    if Best_index(index) == 1
        pBestSwarm = gBest(index,:);
    end
end
for swarm = 1:num_swarms
    for particle = 1:num_particles
        swarms{swarm,particle}.Update(gBest(swarm,:),...
            pBestSwarm,learning_constants,dt);
    end
end

for swarm = 1:num_swarms
    fRecord(swarm,epoch) = fBest(swarm);
end

```

---

```

        if(epoch == epochs)

            Best_index = fBest == min(fBest);
            disp(Best_index);
            for index = 1:num_swarms
                if Best_index(index) == 1
                    swarms{1,1}.x = gBest(index,:);
                    % disp("PIDN Best Value")
                    % disp(gBest(index,:))
                end
            end

        end

    end

end

function y = error_data(type,...
                        error)
switch type
    case 'e^2'
        y = sum(power(error,2));
    case 'te^2'
    case 'eu^2'
    case 'ets'
end
end

```

---

## 13.2. Código CCS

---

```
#include "PeripheralHeaderIncludes.h"
#include "F2806x_EPwm_defines.h" // useful defines for
initialization
#include "F2806x_Cla.h" //Control Law Acelerator para operaciones con
punto flotante
#include "math.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

#define num_swarms 5
#define num_particles 10
#define epochs 50

void DeviceInit(void);
void InitCpuTimers();
void InitFlash();
void MemCopy(Uint16 *SourceAddr, Uint16* SourceEndAddr, Uint16*
DestAddr);

interrupt void adc_isr(void);
extern Uint16 RamfuncsLoadStart, RamfuncsLoadEnd, RamfuncsRunStart;

float32 vc=0;
float32 i=0;
float32 ir=0;
float32 vg=0;
float32 el=0;
float32 el_1=0;
float32 w1=0;
float32 x1=0;
float32 x1_1=0;
float32 y1=0;
float32 y1_1=0;
float32 u1=0;
float32 u11=0;
float32 eaw1_1=0;
float32 eaw1_2=0;
float32 e2=0;
float32 e2_1=0;
float32 u21=0;
float32 u2=0;
float32 w2=0;
float32 x2=0;
float32 x2_1=0;
float32 eaw2_1=0;
```

---

```

float32 eaw2_2=0;
float32 m=0;
float32 m_1=0;
float32 M=0;
float32 b=0;
float32 b_1=0;
float32 B=0;
float32 s=0;

Uint16 count=0;

Uint16 current_rms_counter = 0;
float32 current_buffer[416] = {}; // 416 es el numero de
datos que hay en un semiciclo de una funcion seno, necesarios para
hacer la integral de valor rms
float32 current_rms[2] = {0.0,0.0}; // el valor 0 es aquel
que se esta leyendo continuamente, el valor 1 es para guardar el
registro del cambio superior a 0.2 A
Uint16 counter_trans = 0; // Contador de
transitorio 0.5s, luego de esto el sistema comienza a entrenar.
char train_trigger = 0;
char epoch = 0;
float32 current_error[200] = {};
Uint16 counter_error = 0;
Uint16 counter_swarms = 0;
Uint16 counter_particles = 0;

float32 fBest[num_swarms] = {}; // mejores valores de
cada enjambre
float32 pBest[num_swarms][4] = {}; // mejores posiciones
de cada enjambre
float32 freeze_counter[num_swarms] = {}; // contador de
enjambre en optimo local

float32 kp = 0.504,
ki = 3000,
kd = 0.000001,
N = 4800000;

struct Particle
{
    /* Properties */
    float x_0, x_1, x_2, x_3;
    float v_0, v_1, v_2, v_3;
    float pBest_0, pBest_1, pBest_2, pBest_3;
    float f;
    float fBest;
};

```

---

```

extern const struct Particleclass
{
    struct Particle (*new)(
        float x_0, float x_1, float x_2, float x_3,
        float v_0, float v_1, float v_2, float v_3,
        float pBest_0, float pBest_1, float pBest_2, float pBest_3,
        float f,
        float fBest);

} Particle;

static struct Particle new(

    float x_0, float x_1, float x_2, float x_3,
    float v_0, float v_1, float v_2, float v_3,
    float fBest,
    float pBest_0, float pBest_1, float pBest_2, float pBest_3,
    float f)
{
    return (struct Particle){
        .x_0 = x_0, .x_1 = x_1, .x_2 = x_2, .x_3 = x_3,
        .v_0 = v_0, .v_1 = v_1, .v_2 = v_2, .v_3 = v_3,
        .pBest_0 = pBest_0, .pBest_1 = pBest_1, .pBest_2 = pBest_2,
        .pBest_3 = pBest_3,
        .f = f,
        .fBest = fBest};
}

const struct Particleclass Particle = {.new = &new};

void evaluate(struct Particle *particle, float e)
{
    if (e < particle->fBest)
    {
        particle->fBest = e;
        particle->pBest_0 = particle->x_0;
        particle->pBest_1 = particle->x_1;
        particle->pBest_2 = particle->x_2;
        particle->pBest_3 = particle->x_3;
    }
    particle->f = e;
}

void update(struct Particle *particle, float gBest[4], float c[3],
float dt)
{
    float pParticle[4] = {0,0,0,0};
    float pSwarm[4] = {0,0,0,0};

    pParticle[0] = particle->pBest_0 - particle->x_0;
    pParticle[1] = particle->pBest_1 - particle->x_1;
    pParticle[2] = particle->pBest_2 - particle->x_2;
    pParticle[3] = particle->pBest_3 - particle->x_3;

    pSwarm[0] = gBest[0] - particle->x_0;
    pSwarm[1] = gBest[1] - particle->x_1;

```



---

```

pSwarm[2] = gBest[2] - particle->x_2;
pSwarm[3] = gBest[3] - particle->x_3;

particle->v_0 = c[0]*particle->v_0;
particle->v_0 = particle->v_0 + c[1]*rand()*pParticle[0];
particle->v_0 = particle->v_0 + c[2]*rand()*pSwarm[0];

particle->v_1 = c[0]*particle->v_1;
particle->v_1 = particle->v_1 + c[1]*rand()*pParticle[1];
particle->v_1 = particle->v_1 + c[2]*rand()*pSwarm[1];

particle->v_2 = c[0]*particle->v_2;
particle->v_2 = particle->v_2 + c[1]*rand()*pParticle[2];
particle->v_2 = particle->v_2 + c[2]*rand()*pSwarm[2];

particle->v_3 = c[0]*particle->v_3;
particle->v_3 = particle->v_3 + c[1]*rand()*pParticle[3];
particle->v_3 = particle->v_3 + c[2]*rand()*pSwarm[3];

particle->x_0 = particle->x_0 + particle->v_0 * dt;
particle->x_1 = particle->x_1 + particle->v_1 * dt;
particle->x_2 = particle->x_2 + particle->v_2 * dt;
particle->x_3 = particle->x_3 + particle->v_3 * dt;

if(particle->x_0 <= 0)
{
    particle->x_0 = 0.000001;
}
if(particle->x_1 <= 0)
{
    particle->x_1 = 0.000001;
}
if(particle->x_2 <= 0)
{
    particle->x_2 = 0.000001;
}
if(particle->x_3 <= 0)
{
    particle->x_3 = 0.000001;
}
}

float float_rand( float min, float max )
{
    float scale = rand() / (float) RAND_MAX; /* [0, 1.0] */
    return min + scale * ( max - min );      /* [min, max] */
}

struct Particle p[num_swarms][num_particles];

```

---

```

void main(void)
{
    Uint16 n = 0;
    DeviceInit();    // Device Life support & GPIO mux settings
    #ifdef FLASH
    MemCopy(&RamfuncsLoadStart, &RamfuncsLoadEnd, &RamfuncsRunStart);
    InitFlash();    // Call the flash wrapper init function
    #endif // (FLASH)
    EALLOW;
    AdcRegs.ADCCTL2.bit.ADCNONOVERLAP=1;           // Enable
non-overlap mode
    AdcRegs.ADCCTL1.bit.INTPULSEPOS=1;             // ADCINT1 trips
after AdcResults latch
    AdcRegs.INTSEL1N2.bit.INT1E=1;                 //activa la
interrupcion del ADCINT1
    AdcRegs.INTSEL1N2.bit.INT1CONT=0;              //Disable
ADCINT1 Continuous mode
    AdcRegs.INTSEL1N2.bit.INT1SEL=3;               // dispere la
interrupcion ADCINT1 cuando termie la conversion del SOC7 setup EOC7
to trigger ADCINT1 to fire
    AdcRegs.ADCCTL1.bit.ADCBGPWD=1;               // Power up
band gap
    AdcRegs.ADCCTL1.bit.ADCREFSEL=0;               // Use
internal bandgap
    AdcRegs.ADCCTL1.bit.ADCREFPWD=1;               // Power up
reference
    AdcRegs.ADCCTL1.bit.ADCPWDN=1;                 // Power up
rest of ADC
    AdcRegs.ADCCTL1.bit.ADCENABLE=1;               // set timing
to 13cycles as required by ADC module
    AdcRegs.ADCCTL1.bit.VREFLOCONV=0;
    for(n=0; n<4445; n++)
    {
        // wait 80000 cycles = 1ms (each iteration is 12 cycles)
    }
    AdcRegs.ADCCTL1.bit.INTPULSEPOS=1;             // create int
pulses 1 cycle prior to output latch
    AdcRegs.ADCSAMPLEMODE.bit.SIMULEN0=1;          // PARA
CONVERSION SIMULTANEA SOC0 Y SOC1
    AdcRegs.ADCSOC0CTL.bit.CHSEL=0;                //set SOC0
channel select to ADCINA2 si fuera 0 seria el ADCIN0 pin 2 tabla 8-18
    AdcRegs.ADCSOC0CTL.bit.TRIGSEL=5;              //set SOC0
start trigger on EPWM1A, due to round-robin SOC0 converts first then
SOC1
    AdcRegs.ADCSOC0CTL.bit.ACQPS=6;                //set SOC0 S/H
Window to 26 ADC Clock Cycles, (9 ACQPS plus 1)
    AdcRegs.ADCSAMPLEMODE.bit.SIMULEN2=1;          //pARA
CONVERSION SIMULTANEA SOC0 Y SOC1
    AdcRegs.ADCSOC2CTL.bit.CHSEL=1;                //set SOC0
channel select to ADCINA2 si fuera 0 seria el ADCIN0 pin 2 tabla 8-18
    AdcRegs.ADCSOC2CTL.bit.TRIGSEL=5;              //set SOC0
start trigger on EPWM1A, due to round-robin SOC0 converts first then
SOC1
    AdcRegs.ADCSOC2CTL.bit.ACQPS=6;                //set SOC0 S/H
Window to 26 ADC Clock Cycles, (9 ACQPS plus 1)
    EDIS;

```

---

```

    EPwm1Regs.ETSEL.bit.SOCAEN=1; // Activa
inicio de conversion ADC canal A Enable SOC on A group
    EPwm1Regs.ETSEL.bit.SOCASEL=1; // Select SOC
from activa evento cuando es igual al periodo
    EPwm1Regs.ETPS.bit.SOCAPRD=1; // Generate
pulse on 1st event
    EPwm1Regs.TBPRD=1599; // Set period,
50kHz when PLL is set to 0x10 (80MHz) Tpwm=2xTBPRDxTtblck donde
Ttblck=1/80MHz
    EPwm1Regs.TBCTL.bit.CTRMODE=TB_COUNT_UP; // count up
and start
    EPwm1Regs.TBPHS.half.TBPHS=0; //TBPHS.all
// Time-Base Phase Register
    EPwm1Regs.TBCTL.bit.PRDL=TB_SHADOW;
//TB_IMMEDIATE; // Set Immediate load
    EPwm1Regs.TBCTL.bit.PHSEN=TB_DISABLE; // Disable
phase loading
    EPwm1Regs.TBCTL.bit.SYNCSEL=TB_SYNC_DISABLE;
//TB_SYNC_DISABLE;
    EPwm1Regs.TBCTL.bit.HSPCLKDIV=TB_DIV1; //CON 6 DE
65Hz
    EPwm1Regs.TBCTL.bit.CLKDIV=TB_DIV1; //CON 6
    EPwm1Regs.DBCTL.bit.OUT_MODE=DB_FULL_ENABLE; // Dead-band
is fully enabled for both rising-edge delay on output EPWMxA and
falling-edge
    EPwm1Regs.DBCTL.bit.POLSEL=DB_ACTV_HIC; //invierte
epwmB
    EPwm1Regs.DBCTL.bit.IN_MODE=DBA_ALL;
    EPwm1Regs.DBFED=24; //tiempo
muerto bajada
    EPwm1Regs.DBRED=24; //tiempo
muerto subida
    EPwm1Regs.CMPCTL.bit.SHDWAMODE=CC_SHADOW;
    EPwm1Regs.CMPCTL.bit.LOADAMODE=CC_CTR_ZERO; // load on
CTR=Zero
    EPwm1Regs.AQCTLA.bit.CAU=AQ_CLEAR; // Accion
cuando el contador es igual a CMPA y el contador esta decrementando.
Clear: force EPWMxA output low.
    EPwm1Regs.AQCTLA.bit.ZRO=AQ_SET; // Accion
cuando el contador es igual a CMPA y el contador es incrementando.
Set: force EPWMxA output high.
    CpuTimer0Regs.PRD.all = 25000000;
    IER=0x0000;
    IFR=0x0000;
    EALLOW; // This is
needed to write to EALLOW protected registers
    PieVectTable.ADCINT1=&adc_isr; //Para que
reconozca a que parte del codigo debe saltar
    EDIS; // This is
needed to disable write to EALLOW protected registers
    PieCtrlRegs.PIEIER1.bit.INTx1 = 1; // Enable INT
1.1 in the PIE
    IER |= M_INT1; // Enable CPU
Interrupt 1
    EINT; // Enable
Global interrupt INTM
    ERTM;

```

---

```

EPwm1Regs.CMPA.half.CMPA=0;

//MPSO Initialization

int swarm;
int particle;

srand( (unsigned)time( NULL ) );

for (swarm = 0; swarm < num_swarms; swarm++)
{
    for (particle = 0; particle < num_particles; particle++)
    {
        float x[4] =
{float_rand(0,10),float_rand(0,10000),float_rand(0,0.000001),float_rand(0,10000000000)};
        float v[4] =
{float_rand(0,10),float_rand(0,10000),float_rand(0,0.000001),float_rand(0,10000000000)};
        float pBest[4] = {0,0,0,0};
        memcpy(pBest,x,sizeof(pBest));

        p[swarm][particle] = Particle.new(
            x[0], x[1], x[2], x[3],
            v[0], v[1], v[2], v[3],
            pBest[0], pBest[1], pBest[2], pBest[3],
            9999.0,
            9999.0);
    }
}

p[0][0].x_0 = kp;
p[0][0].x_1 = ki;
p[0][0].x_2 = kd;
p[0][0].x_3 = N;

for(;;) //infinite loop
{
    if (EPwm1Regs.ETFLG.bit.SOCA==1)
    {
        GpioDataRegs.GPBTOGGLE.bit.GPIO33 = 1;
        EPwm1Regs.ETCLR.bit.SOCA=1;
        if (count<416)
        {
            m=m_1+0.00002*vg;
            m_1=m;
            b=b_1+0.00002*vc;
            b_1=b;
            count++;
        }
        else
        {

```

---

```

        M=120*m;
        m_1=0;
        m=0;
        B=120*b;
        b_1=0;
        b=0;
        count=0;
    }
    if (GpioDataRegs.GPBDAT.bit.GPIO32==1)
    {
        if (M<70)
        {
            GpioDataRegs.GPBCLEAR.bit.GPIO32 = 1;
        }
    }
    else
    {
        if (M>90)
        {
            GpioDataRegs.GPBSET.bit.GPIO32 = 1;
        }
    }
    if (GpioDataRegs.GPADAT.bit.GPIO22==1)
    {
        if (B<160)
        {
            GpioDataRegs.GPACLEAR.bit.GPIO22 = 1;
        }
    }
    else
    {
        if (B>200)
        {
            GpioDataRegs.GPASET.bit.GPIO22 = 1;
        }
    }
}
if(CpuTimer0Regs.TCR.bit.TIF == 1)
{
    CpuTimer0Regs.TCR.bit.TIF = 1;    // clear flag
    //-----
    GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1;    //Toggle GPIO34
    (LD2)
    //-----
}
}
}
//END MAIN CODE

interrupt void adc_isr(void)
{
    i = (0.000269*(float32)AdcResult.ADCRESULT0);
    // Lectura de Corriente
    ir = (0.000269*(float32)AdcResult.ADCRESULT1);
    // Corriente de referencia - Voltaje de entrada / Corriente Deseada

```

---

```

    vc = (0.059*(float32)AdcResult.ADCRESULT2);
// Lectura de Voltaje de Salida
    vg = (0.0456*(float32)AdcResult.ADCRESULT1);
// Lectura de Voltaje de Entrada

    current_buffer[current_rms_counter] = ir;
    if(current_rms_counter == 415)
    {
        current_rms_counter = 0;
    }
    else current_rms_counter ++;

    int counter;
    for(counter = 0; counter == 415; counter++)
    {
        current_rms[0] = current_rms[0] +
pow(current_buffer[counter],2);
        current_rms[0] = sqrt(current_rms[0]*0.00240963855);

        if(current_rms[0] + 0.2 >= current_rms[1] || current_rms[0] - 0.2
<= current_rms[1])
        {
            current_rms[1] = current_rms[0];
            if(counter_trans != 25000)
            {
                epoch = 0;
                train_trigger = 1;
            }
        }
        if(counter_trans < 25000 && train_trigger == 1)
        {
            counter_trans = counter_trans + 1;
        }

        e2=220-vc;
        w2=0.0048*e2;
        x2=x2_1+0.0000012*(e2+e2_1)-eaw2_1-eaw2_2;
        u2=w2+x2;
        if (u2<-0.1)
        {
            u2l=-0.1;
        }
        else
        {
            if (u2>1)
            {
                u2l=1;
            }
            else
            {
                u2l=u2;
            }
        }

        e1=u2l*ir-i;

        current_error[counter_error] = e1;
        if(counter_error == 199)
        {
            counter_error = 0;

```

---

```

    if(counter_swarms < num_swarms)
    {
        if(counter_particles < num_particles)
        {
            float32 e;
            for(counter = 0; counter == 199; counter++)
            {
                e = e + pow(current_error[counter],2);
            }
            //e = sqrt(e*0.005);

            evaluate(&p[counter_swarms][counter_particles],e);
            counter_particles++;
            if(counter_particles == num_particles)
            {
                counter_particles = 0;
                counter_swarms++;
                if(counter_swarms == num_swarms)
                {
                    counter_swarms = 0;
                    epoch++;

                    if(epoch == epochs)
                    {
                        // Training Complete
                        counter_trans = 0;
                        train_trigger = 0;

                    }
                }
            }
        }
    }
}
else counter_error++;

// PID Controller

w1= 0.504*e1;
x1=x1_1+0.03*(e1+e1_1)-eaw1_1-eaw1_2;
y1= 0.01029*y1_1+ 0.2028*(e1-e1_1);
u1=w1+x1+y1;

if (u1<0)
{
    u1l=0;
}
else
{
    if (u1>1)
    {
        u1l=1;
    }
}

```

---

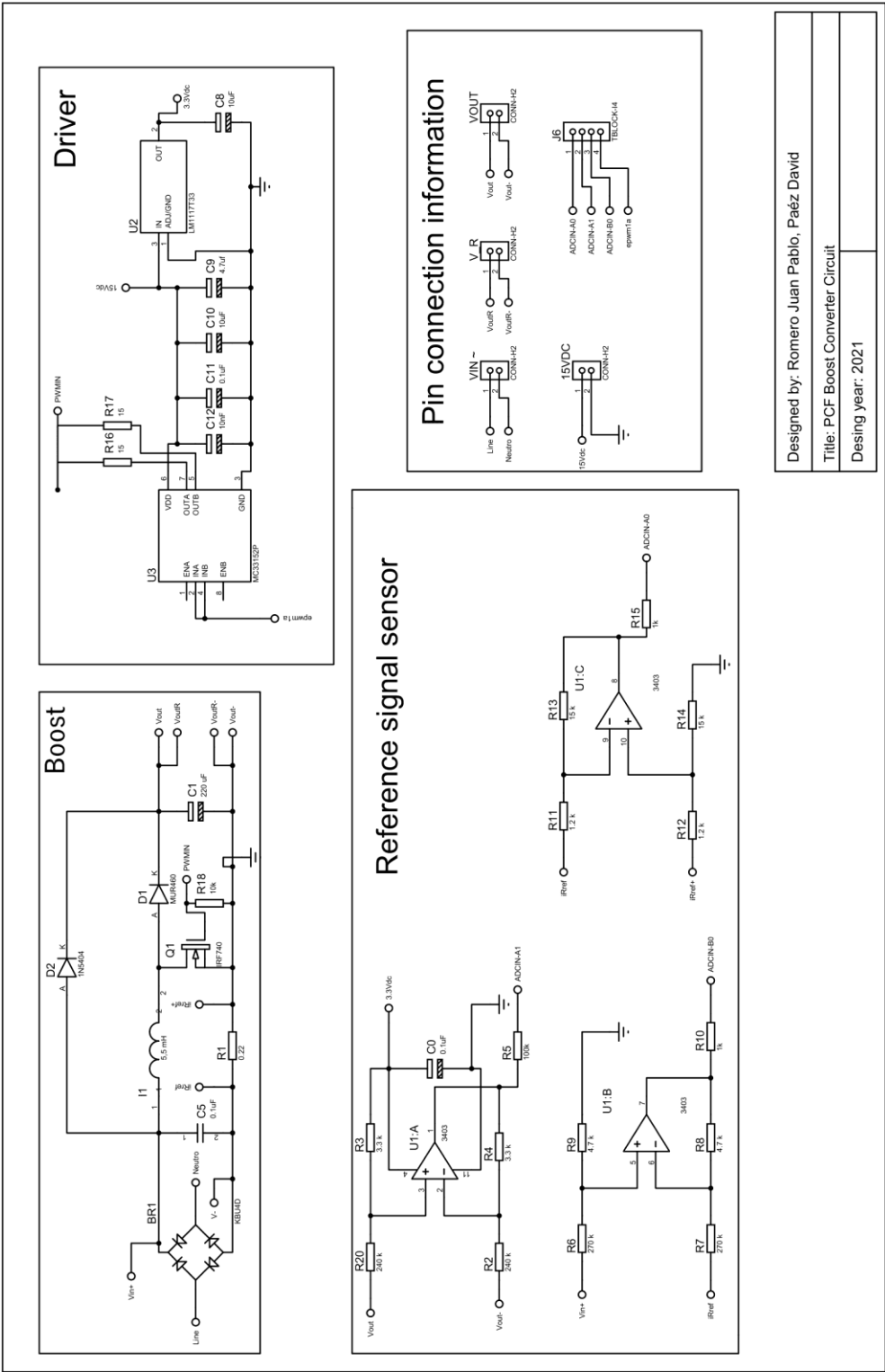
```

        }
    else
    {
        ull=u1;
    }
}
eaw2_2=eaw2_1;
eaw2_1=u2-u2l;
e2_1=e2;
x2_1=x2;
eaw1_2=eaw1_1;
eaw1_1=u1-ull;
e1_1=e1;
x1_1=x1;
y1_1=y1;
if (GpioDataRegs.GPBDAT.bit.GPIO32==1)
{
    if (s<1)
    {
        s=s+0.000001;
    }
}
else
{
    s=0;
}
EPwm1Regs.CMPA.half.CMPA=1599*ull*s;
AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //Clear ADCINT1 flag
reinitialize for next SOC
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;

```



# 13.2. Plano Esquemático del Convertidor



---

## Bibliografía

- [1] C. C. Hang, K. J. Astrom, and W. K. Ho, “Refinements of the Ziegler-Nichols tuning formula,” *IEE Proc. D Control Theory Appl.*, vol. 138, no. 2, pp. 111–118, 1991, doi: 10.1049/ip-d.1991.0015.
- [2] J. Melchi, M. Pagnucco, and E. Lo, “AUTOTUNE IMPLEMENTATION IN MATLAB.”
- [3] W. : Www, S. S. Gade, and S. B. Shendage, “International Journal of Emerging Technology and Advanced Engineering MATLAB Based Response of Systems Using Auto Tune PID Controller,” *Certif. J.*, vol. 9001, no. 12, 2008.
- [4] R. Ben\`vitez, G. Escudero, S. Kanaan, and D. M. Rodó, *Inteligencia artificial avanzada*. Editorial UOC, 2014.
- [5] L. A. H. Fonseca, D. R. G. Leon, and O. M. H. Gómez, “Single-phase rectifier with Power factor correction using a Boost converter,” *Tecnura*, vol. 16, pp. 23–34, 2012.
- [6] G. C. J. M. Serrano, “Corrector de factor de potencia reductor-elevador monofásico.” Cholula, Puebla, México, 2014.
- [7] Q. Meng and T. Liu, “Study on Immune PID Control Method of an In-wheel Motor,” in *Proceedings of the 36th Chinese Control Conference*, 2017, pp. 9554–9559.
- [8] W. A. Shutnan and T. Y. Abdalla, “Artificial Immune system based Optimal Fractional Order PID Control Scheme for Path Tracking of Robot Manipulator ,” in *International Conference on Advances in Sustainable Engineering and Applications*, 2018, pp. 19–24.
- [9] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, Second Edition. Boulder, Colorado, 2004.
- [10] F. S. Llumiquinga Loya, “Diseño de un banco de condensadores para la corrección del factor de potencia de la empresa Banchisfoof SA,” 2012.
- [11] M. H. Rashid, *Electrónica de potencia: circuitos, dispositivos y aplicaciones*. Pearson Educación, 2004.
- [12] I. A. ROSALES SAN MARTÍN and C. A. VICTORIANO GUTIÉRREZ, “ESTUDIO DE CONVERTIDORES DC-DC E IMPLEMENTACIÓN DE UN CONVERTDOR BOOST (ELEVADOR),” 2019.

- 
- [13] J. Xu and F. Deng, "A camera self-calibration method based on IOS-PSO," in *2015 Chinese Automation Congress (CAC)*, 2015, pp. 489–494, doi: 10.1109/CAC.2015.7382550.
- [14] L. Poempool, B. Kruatrachue, and K. Siriboon, "Combine Multi Particle Swarm in Supporting Trapping in Local Optima," in *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, 2018, pp. 1–4, doi: 10.1109/ICEAST.2018.8434398.
- [15] O. Katsuhiko, *Modern control engineering*. 2010.
- [16] K. J. Åström, "Control system design lecture notes for me 155a," *Dep. Mech. Environ. Eng. Univ. Calif. St. Barbar.*, vol. 333, 2002.
- [17] K. M. Passino and N. Quijano, "Proportional-Integral-Derivative Control with Derivative Filtering and Integral Anti-Windup for a DC Servo," *Dept. Electr. Eng. Ohio State Univ.*, 2002.
- [18] M. S. Fadali and A. Visioli, *Digital control engineering: analysis and design*. Academic Press, 2013.
- [19] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari, "Review and classification of bio-inspired algorithms and their applications," *J. Bionic Eng.*, vol. 17, pp. 611–631, 2020.
- [20] F. Valdez, O. Castillo, and P. Melin, "Bio-Inspired Algorithms and Its Applications for Optimization in Fuzzy Clustering," *Algorithms*, vol. 14, no. 4, p. 122, 2021.
- [21] F. L. Luo, "Single-Stage Power Factor Correction AC/DC Converter," *IEEE*, 2005.
- [22] F. H. Martínez, "AC / DC Conversion Techniques in Single-Phase Systems with Unit Power Factor," *Tecnura*, vol. 12, pp. 31–41, 2003.
- [23] J. Muñoz, G. Osorio, and F. Angulo, "Boost converter control with ZAD for power factor correction based on FPGA," *IEEE - Hermes*, 2013.
- [24] A. Krishna and L. Abraham, "Boost Converter based Power Factor Correction for Single Phase Rectifier using Fuzzy Logic Control," in *First International Conference on Computational Systems and Communications (ICCSC)*, 2014, pp. 122–126.
- [25] V. M. A. Ruíz, "PID Controller tuning methods that operate as regulators," *Ingeniería*, vol. 12, pp. 21–36, 2002.
- [26] A. Ullah, S. Khan, and S. Zhaoyun, "Power Factor Control Using Boost Type Converter," in *Advanced Information Technology, Electronic and Automation Control Conference*, 2018, pp. 54–60.

- 
- [27] K. H. Y. Ohta, T. Kajikawa, T. Matsushima, and M. Suzuki, "Robust Digital Control for PFC Boost Converter," 2011.
- [28] J. Yuan *et al.*, "An Immune-Algorithm-Based Dead-Time Elimination PWM Control Strategy in a Single-Phase Inverter," *Trans. power Electron.*, vol. 30, pp. 3964–3975, 2015.
- [29] D. Pen, H. Zhang, L. Yang, and L. Xu, "Study of Immune PID Adaptive Controller and Its Applications in Thermal Control System," in *International Conference on Computational Intelligence and Security*, 2007, pp. 470–474.
- [30] G. M. J. Guillermo, G. M. J. Antonio, and T. R. C. Leonardo, "General inverse neural current control for buck converter," in *Novel Algorithms and Techniques in Telecommunications, Automation and Industrial Electronics*, Springer, 2008, pp. 117–122.
- [31] G. C. C. Duarte, "Desing of an automatic parameter tuning algorith of a PID Controller using bioinspired machine learning techniques." Bogotá, Colombia, 2019.
- [32] H. M. S.F. Toha A.F. Abdul Rahim and R. Akmeliawati, "PID tuned Artificial Immune System Hover Control," *IEEE*, vol. 20, pp. 1235–1240, 2015.
- [33] F. H. M. Sarmiento, E. J. Gomez, and M. C. Ortiz, "Power factor corrector with PID loop fit by genetic algorithm," *Tecnura*, vol. 17, pp. 10–17, 2013.
- [34] R. J. Rajesh and D. A. C. Ananda, "PSO Tuned PID Controller for Controlling Camera Position in UAV Using 2-Axis Gimbal," in *International Conference on Power and Advanced Control Engineering*, 2015, pp. 128–133.
- [35] A. K. Mishra, V. K. Tiwari, R. Kumar, and T. Verma, "Speed Control of DC motor using artificial bee colony optimization technique," 2013.
- [36] A. K. R. Singh and R. Sharma, "Fractional Order PID Control using Ant Colony Optimization," 2016.
- [37] L. C. Cagnina, "Mono and Multiobjective Optimization through a Collective Intelligence Heuristic," Universidad Nacional de San Luis, 2010.
- [38] J. F. Bayona, H. R. Chamorro, A. C. Sanchez, J. Aguilon-Garcia, and D. A. Rubio, "Linear control of a power factor correction rectifier in half-bridge configuration," *IEEE Xplore*, vol. 20, pp. 1–6, 2016.
- [39] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4, pp. 1942–1948.
- [40] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering

- 
- voltage security assessment,” *IEEE Trans. power Syst.*, vol. 15, no. 4, pp. 1232–1239, 2000.
- [41] Z.-L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” *IEEE Trans. power Syst.*, vol. 18, no. 3, pp. 1187–1195, 2003.