

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero Electrónico.



---

**Dr. HERNAN CAMILO ROCHA**  
Decano de la Facultad de Ingeniería



---

**Dr. LUIS EDUARDO TOBON**  
Director Carrera Ingeniería Electronica.



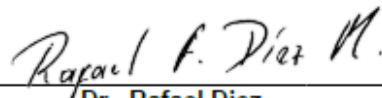
---

**Dr. CARLOS ANDRES GIRALDO CASTAÑEDA**  
Director(a) Trabajo



---

**Dr. Manuel Vicente Valencia**  
Jurado 1



---

**Dr. Rafael Diez**  
Jurado 2



Pontificia Universidad  
**JAVERIANA**  
Cali

**Facultad de Ingeniería**  
**Secretaría de la Facultad**

**Acta de Correcciones al Proyecto de Grado**  
**Ingeniería Electrónica**

**Fecha:** 27 de julio de 2021

**Autores:** Camilo Alexander Diaz Soledad

**Nombre del Proyecto de Grado:** Conversor DC-DC controlado por MPPT de Salto Selectivo para sistema fotovoltaico

**Director:** Carlos Andrés Giraldo Castañeda

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de director(a) del Proyecto de Grado

*Facultad de Ingeniería*

*Calle 18 118-250 Av. Cañasgordas PBX 321 8200 • FAX 555 28 23 • [www.javerianacali.edu.co](http://www.javerianacali.edu.co)*

---

Santiago de Cali, 27 de junio de 2021

Señores  
**Pontificia Universidad Javeriana Cali.**  
Dr. Luis Eduardo Tobón Llano.  
Director Carrera de Ingeniería Electrónica.  
Cali.

Cordial Saludo,

Por medio de la presente me permito informarle que he revisado el proyecto de grado del estudiante de Ingeniería Electrónica Camilo Alexander Díaz Soledad con código 5590137, denominado “Convertor DC-DC controlado por MPPT de Salto Selectivo para sistema fotovoltaico” del cual soy director y considero apto para presentarse al jurado.

Atentamente,



Carlos Andrés Giraldo Castañeda

---

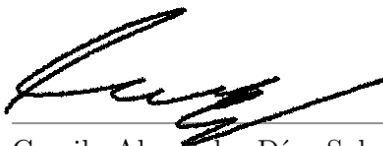
Santiago de Cali, 27 de junio de 2021

Señores  
**Pontificia Universidad Javeriana Cali.**  
Dr. Luis Eduardo Tobón.  
Director Carrera de Ingeniería Electrónica.  
Cali.

Cordial Saludo,

Me permito presentar a su consideración el trabajo de grado denominado “Convertor DC-DC controlado por MPPT de Salto Selectivo para sistema fotovoltaico” con el fin de cumplir con los requisitos exigidos por la Universidad para optar por el título de Ingeniero Electrónico.

Atentamente,



---

Camilo Alexander Díaz Soledad  
Código: 5590137.



Pontificia Universidad  
**JAVERIANA**  
Cali

Pontificia Universidad Javeriana Cali  
Facultad de Ingeniería  
Ingeniería Electrónica

Trabajo de grado:  
Convertor DC-DC controlado por  
MPPT de Salto Selectivo para sistema  
fotovoltaico

Presentado por: Camilo Alexander Díaz Soledad  
Director: Carlos Andrés Giraldo Castañeda

Julio 2021  
Cali

# Agradecimientos

Quiero agradecer primero a Dios, quien me ha dado muchas oportunidades para crecer tanto personal, como profesionalmente y me guía para ser cada día mejor. También le agradezco a mi familia, por acompañarme en cada proceso de mi formación y enseñarme los valores y principios necesarios para ser una persona íntegra. A mi novia y amigos, por ser un gran apoyo emocional durante mi carrera. Finalmente, un especial agradecimiento a mi director, el profesor Carlos Giraldo, por dedicar su tiempo y esforzarse por brindar una guía muy valiosa en el desarrollo de este trabajo.

# Índice

Índice	2
Índice de figuras	4
Índice de cuadros	6
Índice de códigos	7
<b>1. Introducción</b>	<b>9</b>
1.1. Definición del problema . . . . .	9
1.2. Formulación . . . . .	10
1.3. Justificación . . . . .	10
1.4. Objetivos . . . . .	10
1.4.1. Objetivo general . . . . .	10
1.4.2. Objetivos Específicos . . . . .	11
<b>2. Marco teórico</b>	<b>12</b>
2.1. Antecedentes de la investigación . . . . .	12
2.2. Sistemas fotovoltaicos . . . . .	12
2.3. Tipos de Módulos fotovoltaicos . . . . .	13
2.4. Eficiencia de un panel fotovoltaico . . . . .	14
2.4.1. Factor de forma . . . . .	14
2.4.2. Eficiencia en la conversión de energía . . . . .	14
2.5. Perdidas energéticas . . . . .	14
2.5.1. Perdidas por potencia nominal . . . . .	15
2.5.2. Perdidas por suciedad en los paneles . . . . .	15
2.5.3. Pérdidas angulares y espectrales . . . . .	15
2.5.4. Perdidas por temperatura y variaciones climáticas . . . . .	15
2.5.5. Pérdidas por <i>mismatch</i> . . . . .	15
2.6. Convertidores DC-DC . . . . .	16
2.6.1. Parámetros de rendimiento . . . . .	16
2.6.2. Tipos de convertidores DC-DC . . . . .	16
2.7. Algoritmos MPPT . . . . .	24
2.7.1. Algoritmo P&O/Hill Climbing . . . . .	27
2.7.2. Algoritmo de salto selectivo (Selective Hopping o SH) . . . . .	28
2.8. Uso de convertidores DC-DC con algoritmos MPPT . . . . .	30
<b>3. Diseño e implementación</b>	<b>32</b>
3.1. Planteamiento del sistema . . . . .	32
3.2. Características del panel solar . . . . .	32
3.3. Características de las cargas . . . . .	34
3.4. Diseño e implementación del convertidor . . . . .	35
3.4.1. Diseño del convertidor . . . . .	35
3.4.2. Implementación del convertidor . . . . .	39

3.4.3.	Pruebas del convertor . . . . .	41
3.4.4.	Máximos absolutos de operación del convertor . . . . .	47
3.5.	Diseño e implementación del control PWM . . . . .	47
3.5.1.	Diseño del control PWM . . . . .	47
3.5.2.	Implementación del control PWM . . . . .	50
3.5.3.	Pruebas del control PWM . . . . .	51
3.6.	Microcontrolador . . . . .	53
3.7.	Sensores . . . . .	54
3.7.1.	Sensor de voltaje . . . . .	54
3.7.2.	Sensor de corriente . . . . .	57
3.8.	Implementación de algoritmos MPPT . . . . .	59
3.9.	Fuente de alimentación del sistema . . . . .	61
3.10.	Sistema implementado . . . . .	62
<b>4.</b>	<b>Pruebas y análisis de resultados</b>	<b>64</b>
4.1.	Resultados de simulación . . . . .	64
4.2.	Pruebas de algoritmo P&O . . . . .	67
4.3.	Pruebas de algoritmo SH . . . . .	70
4.4.	Comparación de desempeño de los algoritmos . . . . .	73
4.5.	Ciclos de trabajo habituales para ubicar el punto de máxima potencia . . . . .	76
<b>5.</b>	<b>Conclusiones y trabajos futuros</b>	<b>78</b>
	<b>Referencias</b>	<b>80</b>
<b>6.</b>	<b>Anexos</b>	<b>82</b>
6.1.	Diseño de PCB de convertor DC-DC . . . . .	82
6.2.	Pruebas de convertor con carga de 140W . . . . .	83
6.3.	Pruebas de convertor con carga de 250W . . . . .	84
6.4.	Diseño de PCB para control PWM . . . . .	85
6.5.	Código de implementación de algoritmo SH . . . . .	85
6.6.	Código de implementación de algoritmo P&O . . . . .	95
6.7.	Diseño de PCB de la fuente de alimentación . . . . .	103
6.8.	Código de simulación de algoritmo SH en Matlab Simulink . . . . .	103
6.9.	Código de simulación de algoritmo P&O en Matlab Simulink . . . . .	105

# Índice de figuras

1.	Esquema típico de un sistema fotovoltaico. [6]	12
2.	Esquema del convertor Buck. [9]	17
3.	Esquema del convertor Boost. [10]	18
4.	Esquema del convertor Buck-Boost. [11]	20
5.	Esquema del convertor Cúk. [12]	22
6.	Curvas de potencia vs. voltaje y corriente vs. voltaje. [13].	25
7.	Modelo circuital de un panel solar.	25
8.	Cambio de curvas de voltaje vs. corriente debido a Irradiancia y Temperatura. [15]	26
9.	Diagrama de flujo del algoritmo P&O. [16]	27
10.	Recorrido de curva de potencia con P&O. [17]	28
11.	Diagrama de flujo del método de salto selectivo (SH).	29
12.	Búsqueda de punto de máxima potencia por el método de salto selectivo (SH). [2].	30
13.	Esquema de uso de convertidores DC-DC para MPPT [18].	31
14.	Esquema base del proyecto.	32
15.	Curvas del panel a 25°C y diferentes valores de irradiancia. Fuente: Matlab Simulink.	33
16.	Curvas del panel con irradiancia de $1kW/m^2$ y diferentes valores de temperatura. Fuente: Matlab Simulink.	33
17.	Esquema de conexión de las cargas DC.	34
18.	Topología de un convertor DC/DC tipo Buck.	35
19.	Topología de convertor Buck con mosfet de lado inferior.	35
20.	Convertor Buck con filtros LC.	37
21.	Esquema de conexión del driver MCP14E7 en el convertor.	38
22.	Esquemático final de diseño del convertor.	39
23.	Ancho de pista de PCB mínimo para corriente de 10.42A y temperatura de 30°C.	40
24.	Convertor implementado.	41
25.	Prueba de eficiencia de conversión con carga de 140W.	41
26.	Prueba de eficiencia de conversión con carga de 250W.	42
27.	$V_{out}$ con carga de 140W y ciclo de trabajo de 15 %	43
28.	$V_{out}$ con carga de 140W y ciclo de trabajo de 47,3 %	43
29.	$V_{out}$ con carga de 140W y ciclo de trabajo de 96 %	44
30.	$V_{out}$ con carga de 250W y ciclo de trabajo de 15 %	44
31.	$V_{out}$ con carga de 250W y ciclo de trabajo de 47,3 %	45
32.	$V_{out}$ con carga de 250W y ciclo de trabajo de 96 %	45
33.	Visualización de la frecuencia de conmutación y la frecuencia del ruido.	46
34.	Respuesta de circuito tanque con oscilaciones amortiguadas [23].	46
35.	Configuración de 555 para control PWM.	47
36.	Corrientes de carga y descarga del capacitor C2 en control PWM.	48
37.	Esquemático completo de control PWM	49
38.	Control PWM implementado.	50
39.	Terminales del potenciómetro digital.	51

40.	Comparación del ciclo de trabajo real y el del código. . . . .	51
41.	Ciclo de trabajo máximo (96 %), medio (50 %) y mínimo (15 %) respectivamente	52
42.	Curva típica de corriente forward vs. voltaje forward del diodo 1N4148. . . . .	52
43.	Diagrama de pines del Arduino Nano. [26] . . . . .	53
44.	Montaje de Arduino Nano en Shield I/O. . . . .	54
45.	Sensor de voltaje modificado. . . . .	55
46.	Sensor de corriente ACS712-20A. [27] . . . . .	57
47.	Transición de 0V a 24V en la carga. . . . .	59
48.	Diagrama de flujo del algoritmo SH implementado para aplicación en conver- sor DC/DC. . . . .	60
49.	Diagrama de flujo del algoritmo P&O implementado para aplicación en con- versor DC/DC. . . . .	61
50.	Fuente de alimentación del sistema. . . . .	62
51.	Fuente de alimentación implementada en PCB. . . . .	62
52.	Montaje del sistema . . . . .	62
53.	Panel solar utilizado. . . . .	63
54.	Esquema de simulación del sistema en Simulink. . . . .	65
55.	Simulación de rastreo con SH usando condiciones ideales. . . . .	66
56.	Simulación de rastreo con P&O usando condiciones ideales. . . . .	67
57.	Curvas de rastreo usando P&O. . . . .	68
58.	Simulación de rastreo usando P&O con condiciones $Irradiancia = 805W/m^2$ y $Temperatura = 68^{\circ}C$ . . . . .	69
59.	Curvas de rastreos consecutivos usando P&O. . . . .	70
60.	Curvas de rastreo usando SH. . . . .	71
61.	Curvas de rastreo simulando SH con $Irradiancia = 802W/m^2$ y $Temperatura =$ $62^{\circ}C$ . . . . .	72
62.	Curvas de rastreos consecutivos usando SH. . . . .	73
63.	Comparación de curvas de rastreos con SH y P&O. . . . .	74
64.	Comparación de curvas de rastreos con P&O y SH. . . . .	75
65.	Rastreo P&O con ciclo de trabajo inicial de 78%. . . . .	76
66.	Rastreo SH con rango de búsqueda entre el 55 % y el 80%. . . . .	77
67.	Diseño de PCB del convertor DC-DC . . . . .	82
68.	Vista 3D de diseño de PCB del convertor DC-DC . . . . .	82
69.	Tabla de pruebas del convertor con carga de 140W . . . . .	83
70.	Tabla de pruebas del convertor con carga de 250W . . . . .	84
71.	Diseño de PCB de control PWM . . . . .	85
72.	Vista 3D de diseño de PCB de control PWM . . . . .	85
73.	Diseño de PCB de la fuente de alimentación . . . . .	103
74.	Vista 3D del diseño de PCB de la fuente de alimentación . . . . .	103

## Índice de cuadros

1.	Parámetros de diseño del convertor . . . . .	36
2.	Cálculo de valores mínimos de $C$ y $L$ del convertor. . . . .	36
3.	Características principales del MOSFET NCE8580. [21] . . . . .	37
4.	Características principales del diodo F30S60S. [22] . . . . .	38
5.	Máximos absolutos de operación del convertor implementado . . . . .	47
6.	Valores de diseño de control PWM con 555 . . . . .	49
7.	Fragmento de tabla para calibración de sensor por rangos de voltaje. . . . .	57
8.	Resultado de pruebas de P&O y SH . . . . .	75

## Índice de códigos

1.	Código de definición de ciclo de trabajo. . . . .	50
2.	Función básica para medición de voltaje. . . . .	55
3.	Calibración por rangos de voltaje y ciclo de trabajo del conversor. . . . .	55
4.	Función de medición de corriente. . . . .	58
5.	Código de algoritmo SH. . . . .	85
6.	Código de algoritmo P&O. . . . .	95

# Resumen

En este documento se presenta el diseño de un mecanismo de control para un convertidor DC-DC reductor, con el cual se logra una conversión eficiente para un sistema fotovoltaico regulado por un método MPPT. Se trabajó con el algoritmo de salto selectivo (Selective Hopping o SH) [2], el cual cuenta con un tiempo de convergencia mas bajo respecto al algoritmo en el que fue basado (Perturb and Observe o P&O). Con este algoritmo implementado y acoplado al convertidor DC/DC, se alimentaron unas cargas lumínicas de corriente directa que extraen la máxima potencia de un panel solar.

Finalmente, se evaluó el funcionamiento del sistema con el algoritmo MPPT de salto selectivo y se hizo la comparación de su desempeño frente al método MPPT convencional P&O.

# Abstract

This document presents the design of a control mechanism for a step-down DC-DC converter, which achieves an efficient conversion for a photovoltaic system regulated by an MPPT method. The Selective Hopping algorithm (SH) [2] was used, which has a lower convergence time regarding to the algorithm on which it was based (Perturb and Observe or P&O). With this algorithm implemented and coupled to the DC/DC converter, direct current light loads that extract the maximum power from a solar panel are fed.

Finally, the performance of the system was evaluated with the MPPT Selective Hopping algorithm and its performance was compared to the conventional MPPT P&O method.

# Capítulo 1

## 1. Introducción

La energía eléctrica es una fuente esencial para mover el desarrollo de los países, dinamiza efectivamente el comercio y hace posible el funcionamiento de la industria de producción. El sistema eléctrico está constituido por un conjunto de subsistemas que funcionan a distintos niveles de tensión y que tienen como finalidad asegurar el suministro de energía eléctrica a todos los usuarios con las menores pérdidas y la mayor calidad posibles. Dentro de estos sistemas se encuentran las fuentes no convencionales de energía, como la solar. Para que estos sistemas sean útiles, es necesario disponer de la capacidad de generación suficiente y entregarla con eficiencia y de manera segura al usuario final, requiriendo así inversiones grandes de capital, estudios y diseños apropiados, los cuales se rigen por las distintas normas nacionales e internacionales. [1]

Por lo anterior, para migrar a las fuentes no convencionales de energía, como la solar, es necesario aumentar su eficiencia de modo que la relación de costo y beneficio de su instalación y uso final sea comparable al de las energías convencionales, logrando aumentar su demanda. Actualmente existen muchos sistemas fotovoltaicos en los cuales cierta parte de la energía se desperdicia o se utiliza de manera inadecuada, resultando así un sistema ineficiente y costoso. Una solución para aumentar la eficiencia de un sistema fotovoltaico está en la implementación de algoritmos de rastreo de punto máximo de potencia (Maximum Power Point Tracking, MPPT), cuya función es encontrar los parámetros con los que el panel solar o el arreglo fotovoltaico brindará la potencia máxima con las condiciones de irradiancia y temperatura a las que se encuentre sometido. En este proyecto se trabajó con el algoritmo MPPT de salto selectivo [2].

La presente investigación busca lograr una conversión eficiente desarrollando un nuevo mecanismo de control para un convertidor DC-DC, para que éste se adecúe al sistema de un panel solar regulado por un MPPT.

Este documento está compuesto por 5 capítulos distribuidos de la siguiente manera: I. Introducción y Objetivos, donde se presenta la introducción y objetivos que se plantearon como logros con el desarrollo de esta investigación. II. Marco teórico, donde se presenta la base teórica y conceptual sobre la cual se sustenta el proyecto. III. Diseño e implementación, donde se describen las etapas de diseño e implementación del sistema, tanto la parte de hardware como la de software, IV. Pruebas y análisis de resultados, donde se da a conocer los resultados obtenidos con lo implementado, así como un análisis y validación de los mismos; finalmente, capítulo V. Conclusiones, donde se presenta lo concluido de esta investigación y además se mencionan los posibles trabajos a futuro y algunas recomendaciones importantes.

### 1.1. Definición del problema

Los sistemas fotovoltaicos han tenido gran auge en las últimas décadas, sin embargo, estos aún presentan problemas. Problemas de eficiencia debido a diversos factores, que pueden ser clasificados como propios o externos de acuerdo con su naturaleza, problemas de operación o problemas de entorno. Las variaciones climáticas, alta temperatura de la célula solar y exposición al medio ambiente, por ejemplo, pueden afectar el rendimiento de los paneles

solares considerablemente [3]. Toda esta serie de problemas provocan que la producción de energía fotovoltaica no se haya masificado, en comparación con las energías convencionales. Uno de estos problemas son las variaciones climáticas en los sistemas fotovoltaicos, específicamente en las condiciones de radiación solar y temperatura ambiente [4]. Cuando un panel solar es sometido a variaciones climáticas drásticas o repentinas, trae como consecuencia una disminución considerable del pico de potencia que se podría suministrar. Las técnicas de seguimiento del punto de máxima potencia (MPPT) son el método más simple para mejorar el rendimiento del módulo fotovoltaico. Sin embargo, muchos se han diseñado teóricamente pero no son realizables en la práctica y otros se estancan al encontrar un máximo local de potencia bajo condiciones especiales.

Motivado por esta problemática, se desarrolló el algoritmo de salto selectivo. El esquema planteado logra encontrar el punto de operación de máxima potencia donde la energía es consumida en las cargas de prueba que utiliza el sistema. Por esta razón, no se está dando un debido aprovechamiento a la energía generada, es decir, se está encontrando el punto de máxima potencia de manera *offline*. El objetivo principal de [2] era caracterizar el algoritmo de Salto Selectivo sin ser integrado como algoritmo de control en un convertor DC-DC. Para que el sistema tenga un uso práctico y sea viable, es necesario implementarlo como un algoritmo MPPT *online*, por lo cual es necesario un convertor de potencia que entregue la energía generada a cargas reales y que a través de la modificación de su ciclo útil de conmutación, se pueda sintonizar el punto de máxima potencia.

## 1.2. Formulación

¿Cómo mejorar el aprovechamiento de la energía recibida de un sistema fotovoltaico que usa un algoritmo MPPT a través de un convertor de potencia?

## 1.3. Justificación

En la actualidad las fuentes alternativas de energía están desempeñando un papel muy importante en los sistemas de generación eléctrica, siendo el sistema fotovoltaico, la fuente de generación alternativa de mayor crecimiento en los últimos años. [5] Y es que, para el año 2017 la capacidad de generación de este tipo de energía sufrió un crecimiento del 50 %, llegando a ser la fuente de energía con el crecimiento más veloz, incluso por delante del carbón. [5]

Para lograr este crecimiento en la capacidad de generación de energía fue necesario crear sistemas que combinen varios mecanismos de optimización, tales como la refrigeración de los módulos fotovoltaicos, el seguimiento mecánico del sol para lograr mejor ángulo de incidencia de la irradiancia solar a medida que transcurre el día, infinidad de algoritmos MPPT, entre muchos otros. La intención al elaborar este proyecto es contribuir al conjunto de métodos ya implementados y aportar al aumento de eficiencia del sistema de energía solar fotovoltaica.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Diseñar un nuevo mecanismo de control para un convertor DC-DC, para que este se adecúe al sistema de un panel solar regulado por un método MPPT y logre una conversión

eficiente.

#### **1.4.2. Objetivos Específicos**

- Acoplar correctamente el conversor DC-DC al algoritmo MPPT Selective Hopping.
- Definir los rangos de operación del conversor.
- Evaluar el desempeño del sistema con el conversor DC-DC usando una técnica de MPPT tradicional frente al sistema con la técnica MPPT de salto selectivo.
- Caracterizar el sistema implementado.

# Capítulo 2

## 2. Marco teórico

### 2.1. Antecedentes de la investigación

Este proyecto está enfocado a trabajar con algoritmos de rastreo de punto de máxima potencia, en particular con el algoritmo de salto selectivo diseñado e implementado como tesis de maestría en ciencias en ingeniería eléctrica [2], logrando muy buenos resultados. A continuación se definirán los conceptos más importantes referentes al tema tratado.

### 2.2. Sistemas fotovoltaicos

Un sistema Fotovoltaico está integrado por varios componentes. Cada uno de ellos cumpliendo una función específica, a fin de que este pueda suplir la demanda de energía eléctrica impuesto por una carga. Siendo la carga eléctrica la que determina el dimensionamiento del sistema, es decir, las características técnicas de los componentes que lo conforman.

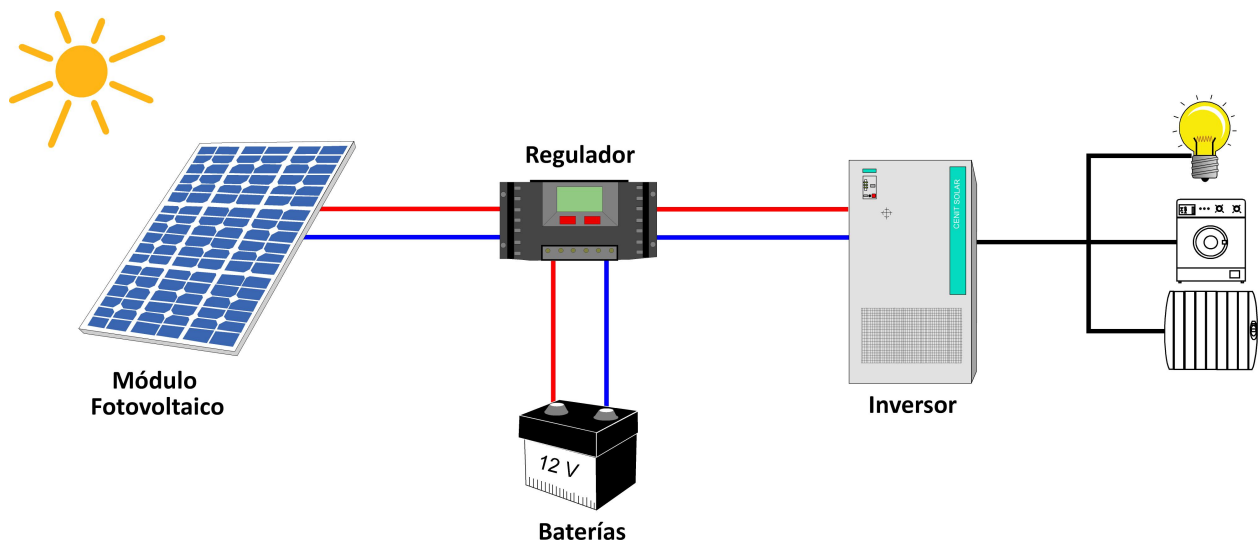


Figura 1: Esquema típico de un sistema fotovoltaico. [6]

En general los bloques que componen a un sistema fotovoltaico son los siguientes:

- **Bloque de generación**

Los paneles fotovoltaicos forman el bloque de generación. El número de ellos depende de varios factores como la radiación solar media del sitio, la potencia de la carga y la potencia que los paneles entregan.

Un panel fotovoltaico está compuesto de un número de células fotovoltaicas interconectadas entre sí y selladas con un recubrimiento de vidrio y un respaldo impermeable. Proporciona en su salida de conexión una tensión continua. Los módulos se construyen con marcos adecuados para su posterior montaje.

- **Bloque de acumulación**

Este bloque contiene tres componentes, el banco de baterías, el control de carga y las protecciones. El control de carga cumple dos funciones: garantizar un régimen de carga adecuado para las baterías, y evita la descarga de las mismas a través de los paneles en las noches. En ocasiones se opta por alimentar cargas DC desde este bloque directamente, por lo cual hace falta que el control de carga trabaje con una técnica MPPT (Maximum Power Point Tracking) y adecúe la potencia para que además de cargar la batería, alimente óptimamente estas cargas DC.

Un controlador de carga básico consiste en un sistema de modulación de ancho de pulso (PWM) con el cual se adecúa el valor medio de voltaje con el que se cargarán las baterías, pero existen controladores de carga más complejos que garantizan mayor eficiencia de carga, los cuales se componen de un convertidor de potencia de entrada en DC y salida en DC, que adecúa el voltaje y la corriente de salida dependiendo del nivel de carga de la batería y de la impedancia equivalente de las cargas conectadas al sistema.

- **Bloque de inversión**

Este bloque está compuesto por el inversor del sistema, encargado de convertir la corriente continua del sistema en corriente alterna, igual a la utilizada en la red eléctrica. Para el caso de Colombia 120 V de valor eficaz y una frecuencia de 60 Hz.

Los inversores para aplicaciones fotovoltaicas incluyen funciones de control para optimizar la potencia de salida (MPPT). La potencia de salida se calcula a partir de la ecuación  $P = V * I$ , es decir la tensión multiplicada por la corriente, y la función MPPT continuamente ajusta la impedancia de la carga para garantizar la potencia óptima.

### 2.3. Tipos de Módulos fotovoltaicos

En el mercado existen diferentes tipos de paneles solares en función del tipo de material empleado para su fabricación. A continuación, se describen alguno de ellos.

- **Silicio Puro monocristalino:** compuesto por una barra de silicio perfectamente cristalizada en una sola pieza. Con un rendimiento teórico alrededor del 24 %, y un rendimiento real entre 17 % y el 20 %. [7]
- **Silicio puro policristalino:** Los materiales son semejantes a los del tipo anterior, aunque en este caso el proceso de cristalización del silicio es diferente. Los paneles policristalinos se basan en secciones de una barra de silicio que se ha estructurado desordenadamente en forma de pequeños cristales. Poseen un rendimiento teórico cercano al 19 %, y un rendimiento real entre 13 % y el 15 %. A pesar de ser menos eficientes, este tipo de paneles son de menor precio en el mercado y tiene un rendimiento mayor en altas temperaturas. [7]
- **Silicio amorfo (TFS):** Este tipo de paneles no sigue aquí estructura cristalina alguna. Las células fotovoltaicas de este tipo son usadas habitualmente para aplicaciones pequeñas como relojes y calculadoras solares. Su rendimiento máximo alcanzado en laboratorio ha sido del 13 % siendo el de los módulos comerciales del 8 %. [7]

## 2.4. Eficiencia de un panel fotovoltaico

El factor de forma y la conversión de energía son dos métodos principales para determinar la eficiencia del panel fotovoltaico. Para ambos métodos se debe conocer el punto de máxima potencia, que corresponde al punto en el cual el voltaje y la corriente alcanzan sus valores máximos de forma simultánea.

Este valor se determina de acuerdo a la siguiente ecuación:

$$P_{max} = V_{pmax} * I_{pmax} \quad (1)$$

Donde:

$P_{max}$ : Punto de máxima potencia

$V_{pmax}$ : Voltaje de máxima potencia

$I_{pmax}$ : Corriente de máxima potencia.

### 2.4.1. Factor de forma

Permite determinar la eficiencia de un panel solar, se calcula mediante el cociente entre la máxima potencia que puede entregar el modulo a la carga y el producto de la tensión de circuito abierto y la corriente de cortocircuito. entre más se acerque a la unidad, mayor potencia proporciona el panel. [29]

$$FF = \frac{P_{max}}{V_{oc} * I_{sc}} \quad (2)$$

Donde:

$P_{max}$ : Punto de máxima potencia.

$V_{oc}$ : Voltaje de circuito abierto

$I_{sc}$ : Corriente de corto circuito

### 2.4.2. Eficiencia en la conversión de energía

Corresponde a la relación entre la cantidad de energía generada y la cantidad de energía luminosa recibida desde el sol, cuando el panel se encuentra conectado a un circuito eléctrico. Se determina con la ecuación. [29]

$$\eta = \frac{P_{max}}{E * AC} \quad (3)$$

Donde:

E: Nivel de radiación sobre la superficie del panel (W/m<sup>2</sup>)

AC: Área de la superficie del panel fotovoltaico.

## 2.5. Perdidas energéticas

Para poder evaluar el rendimiento de los equipos, las pérdidas son un factor determinante a tener en cuenta en todo momento. Diferentes investigaciones se han desarrollado para identificar los factores que afectan el funcionamiento de los paneles fotovoltaicos con el

objetivo de lograr mejoras en la eficiencia para estos sistemas [30]. A continuación se exponen las principales causas de la pérdida energética en los sistemas fotovoltaicos.

### 2.5.1. Pérdidas por potencia nominal

En general los fabricantes de paneles solares garantizan que la potencia de un módulo fotovoltaico de potencia nominal  $P_{nom}$ , está dentro de una banda que oscila entre  $P_{nom} \pm 3\%$ ,  $P_{nom} \pm 5\%$  o  $P_{nom} \pm 10\%$ . Puede suceder que la potencia se sitúa en la banda inferior ofrecida por el fabricante.

### 2.5.2. Pérdidas por suciedad en los paneles

La cantidad de radiación recibida por las células solares en el interior del módulo fotovoltaico es inferior a la que llega a la superficie del mismo, debido a la reflexión y la suciedad de la superficie del módulo. [31] La presencia de una suciedad uniforme da lugar a una disminución de la corriente y tensión entregada por el generador fotovoltaico y por otro lado la presencia de suciedades localizadas da lugar a un aumento de las pérdidas de *mismatch* y a las pérdidas por formación de puntos calientes.

### 2.5.3. Pérdidas angulares y espectrales

El ángulo de inclinación al cual se ubican los paneles, tiene repercusión en la radiación solar que llega a la superficie de las células. Cuando se sitúan perpendicularmente en dirección al sol, reciben la máxima radiación por intervalo de tiempo, considerándose este el ángulo de inclinación óptimo [32]. Este ángulo depende de factores como la latitud, la radiación solar de la zona y el periodo de utilización.

### 2.5.4. Pérdidas por temperatura y variaciones climáticas

Las variaciones climáticas en los sistemas fotovoltaicos tienen un efecto adverso en su rendimiento, las pérdidas de potencia por cambios de temperatura están en el orden de un 4% por cada 10 °C de aumento de su temperatura de operación.

Las técnicas de seguimiento del punto de máxima potencia (MPPT), son unas de las técnicas más utilizadas para reducir las pérdidas de eficiencia debido a las variaciones climáticas, haciendo que el sistema se encuentre siempre en su punto máximo de potencia. En la actualidad existen varios métodos como lo son: Hill Climbing (HC), algoritmo de perturbación y observación (P&O), algoritmo de conductancia incremental (INC), redes neuronales, algoritmos genéticos (GA) y controladores difusos (FC). Sin embargo, los métodos más utilizados son los P&O.

### 2.5.5. Pérdidas por *mismatch*

Los módulos fotovoltaicos a pesar de tener las mismas características técnicas, e incluso ser del mismo fabricante, sus potencias reales, así como su corriente y voltaje pueden variar ligeramente. Al realizar conexiones en serie o paralelo de estos módulos, se generan caídas de tensión y corriente del sistema en conjunto, el módulo de menor corriente limitará la corriente de la serie. De modo semejante ocurre para la tensión de la conexión de módulos en paralelo.

## 2.6. Conversores DC-DC

Los conversores DC-DC se conocen también como pulsadores, reguladores de conmutación o fuentes conmutadas. Permiten que una fuente de voltaje fijo o variable pueda usarse a través de este convertidor como una fuente de voltaje variable y ajustable [8]. El trabajo de los conversores DC-DC es entregar un voltaje mayor y/o menor (dependiendo de la topología) al recibido en su entrada de manera controlada, para esto utiliza dispositivos semiconductores que permiten o interrumpen el flujo de corriente en zonas estratégicas del circuito de manera que se pueda almacenar temporalmente la energía entrante usando capacitores e inductores. Usualmente estos conversores se utilizan a frecuencias de conmutación fijas y lo que varía es el ancho del pulso de conmutación (PWM), de tal manera que el valor de voltaje de salida del convertidor depende directamente del ciclo de trabajo fijado ( $D = \frac{t_{on}}{t_{off}} * 100\%$ ).

### 2.6.1. Parámetros de rendimiento

Ya que se trata de fuentes conmutadas, su rendimiento se mide por las componentes ondulatorias de voltaje y corriente a su salida, entre menor sean las componentes ondulatorias, mas cerca se estará de obtener a la salida unas señales de voltaje y corriente DC ideal y por lo tanto, mejor es el rendimiento.

Para lograr tener componentes ondulatorias pequeñas, es necesario encontrar un balance entre los valores de capacitancia e inductancia de los conversores y la frecuencia de conmutación, mientras mas alta sea la frecuencia, se podrán utilizar valores de capacitancia e inductancia mas pequeños, lo que hace que los componentes sean mas pequeños en tamaño e incluso más económicos.

Esta frecuencia máxima de conmutación está limitada por la física de los dispositivos semiconductores, debido a que estos necesitan cierto tiempo para activarse y desactivarse. Estos tiempos son los tiempos de subida y de bajada y los retardos de subida y de bajada. Por esta misma razón el ciclo de trabajo de conmutación queda limitado entre un máximo y un mínimo. [8]

### 2.6.2. Tipos de conversores DC-DC

Los conversores se clasifican dependiendo de como sea la magnitud y la polaridad del voltaje de salida respecto al del voltaje de entrada.

Indiferentemente del tipo de convertidor, los dispositivos utilizados para la conmutación son BJT, MOSFET o IGBT de potencia y para disminuir el rizado de voltaje y corriente de salida usualmente se utilizan filtros LC.

A continuación se presentan 4 tipos básicos de conversores DC-DC [8]:

#### 1. Conversores reductores (Buck)

Su voltaje de salida es menor que el voltaje de entrada. Requiere de un solo transistor y un diodo de “freewheeling”, aunque en ocasiones se reemplaza dicho diodo por otro transistor y se obtiene un convertidor sincronizado. La corriente de entrada es discontinua y se suele necesitar un filtro suavizante de entrada. Proporciona una polaridad de voltaje de salida igual al voltaje de entrada y una corriente unidireccional de salida.

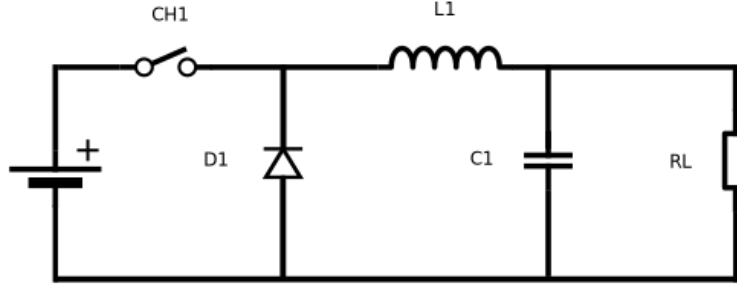


Figura 2: Esquema del convertor Buck. [9]

Este regulador tiene dos fases: la primera es cuando el transistor conduce y fluye corriente a través del inductor, del capacitor y de la carga. La segunda fase es cuando el transistor deja de conducir y el diodo de freewheeling se activa para dejar fluir la corriente almacenada en el inductor, hasta que empiece nuevamente el ciclo.

El voltaje en el inductor es:

$$V_L = L \frac{di}{dt} \quad (4)$$

Suponiendo que la corriente en el inductor crece de manera lineal desde  $I_1$  hasta  $I_2$  en el tiempo  $t_1$  cuando el transistor conduce, se obtiene:

$$V_{in} - V_{out} = L \frac{I_2 - I_1}{t_1} = L \frac{\Delta I}{t_1} \quad (5)$$

Por lo tanto:

$$t_1 = \frac{\Delta I L}{V_{in} - V_{out}} \quad (6)$$

Por otro lado, si la corriente disminuye linealmente de  $I_2$  a  $I_1$  en el tiempo  $t_2$  cuando el transistor deja de conducir, se obtiene:

$$-V_{out} = -L \frac{\Delta I}{t_2} \quad (7)$$

Por lo tanto:

$$t_2 = \frac{\Delta I L}{V_{out}} \quad (8)$$

Donde  $\Delta I = I_2 - I_1$  es el rizado de corriente pico a pico en el inductor. Si se igualan los valores de  $\Delta I$  en las ecuaciones 5 y 7, se obtiene:

$$\Delta I = \frac{(V_{in} - V_{out}) t_1}{L} = \frac{V_{out} t_2}{L} \quad (9)$$

Poniendo  $t_1$  y  $t_2$  en términos del ciclo de trabajo y el periodo, es decir,  $t_1 = D T$  y  $t_2 = (1 - D)T$ , se obtiene el voltaje promedio a la salida del convertor:

$$V_{out} = V_{in} \frac{t_1}{T} = D V_{in} \quad (10)$$

Este tipo de convertor entrega una corriente de salida mayor a la de entrada, puesto que el voltaje de salida es menor al de entrada y debe compensar la potencia de salida,

de manera que la potencia de entrada y de salida del conversor sean prácticamente iguales.

El periodo de conmutación se expresa de la siguiente manera:

$$T = \frac{1}{f} = t_1 + t_2 = \frac{\Delta I L}{V_{in} - V_{out}} + \frac{\Delta I L}{V_{out}} = \frac{\Delta I L V_{in}}{V_{out} (V_{in} - V_{out})} \quad (11)$$

De allí, es puede extraer el rizado de corriente en el inductor como:

$$\Delta I = \frac{V_{in} D (1 - D)}{f L} \quad (12)$$

La corriente del inductor es la suma de las corrientes por el capacitor y por la carga, es decir,  $I_L = I_C + I_{out}$ , por lo que si el rizado por la carga es muy pequeño, se tiene que  $\Delta I_L = \Delta I_C$ . La corriente promedio en el capacitor que fluye para  $T/2$ , es:

$$I_C = \frac{\Delta I}{4} \quad (13)$$

El voltaje en el capacitor es:

$$V_C = \frac{1}{C} \int I_C dt + V_C|_{t=0} \quad (14)$$

y el rizado de voltaje en el capacitor es:

$$\Delta V_C = V_C - V_C|_{t=0} = \frac{1}{C} \int_0^{T/2} \frac{\Delta I}{4} dt = \frac{\Delta I T}{8 C} = \frac{\Delta I}{8 f C} \quad (15)$$

Tomando el valor de  $\Delta I$  en la ecuación 24 y sustituyéndolo en la ecuación 15, se obtiene:

$$\Delta V_C = \frac{V_{in} D (1 - D)}{8 L C f^2} \quad (16)$$

La topología reductora tiene una desventaja respecto al control de la conmutación de los transistores Mosfet, debido a que se requiere de circuitería adicional para obtener un  $V_{GS}$  adecuado para activar el Mosfet, ya que el *source* no está conectado a tierra.

## 2. Conversores elevadores (Boost)

Su voltaje de salida es mayor que el voltaje de entrada. Solo requiere un transistor, por lo cual tiene pocas perdidas por conmutación. El voltaje de salida es muy sensible a los cambios que se produzcan en el ciclo de trabajo y en algunos casos es difícil estabilizar la salida. La corriente promedio de salida es menor que la corriente promedio del inductor y una corriente RMS mas alta fluye a través del capacitor del filtro, por lo que, comparando este regulador con un regulador reductor, se necesitaría un capacitor y un inductor de mayor tamaño.

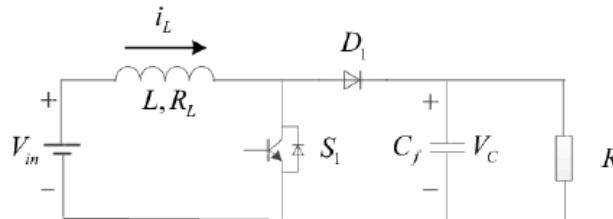


Figura 3: Esquema del conversor Boost. [10]

Las dos fases de este circuito son: primera fase, cuando el transistor conduce, la corriente de entrada aumenta y fluye por el inductor; y segunda fase, cuando el transistor deja de conducir y la corriente fluye por el inductor, el diodo, el capacitor y la carga. Suponiendo que la corriente en el inductor crece de manera lineal desde  $I_1$  hasta  $I_2$  en el tiempo  $t_1$  cuando el transistor conduce, se obtiene:

$$V_{in} = L \frac{I_2 - I_1}{t_1} = L \frac{\Delta I}{t_1} \quad (17)$$

Por lo tanto:

$$t_1 = \frac{\Delta I L}{V_{in}} \quad (18)$$

Por otro lado, si la corriente disminuye linealmente de  $I_2$  a  $I_1$  en el tiempo  $t_2$  cuando el transistor deja de conducir, se obtiene:

$$V_{in} - V_{out} = -L \frac{\Delta I}{t_2} \quad (19)$$

Por lo tanto:

$$t_2 = \frac{\Delta I L}{V_{out} - V_{in}} \quad (20)$$

Donde  $\Delta I = I_2 - I_1$  es el rizado de corriente pico a pico en el inductor. De las ecuaciones 17 y 19, se obtiene:

$$\Delta I = \frac{V_{in} t_1}{L} = \frac{(V_{out} - V_{in}) t_2}{L} \quad (21)$$

Poniendo  $t_1$  y  $t_2$  en términos del ciclo de trabajo y el periodo, como  $t_1 = D T$  y  $t_2 = (1 - D)T$ , se obtiene el voltaje promedio a la salida del conversor:

$$V_{out} = V_{in} \frac{T}{t_2} = \frac{V_{in}}{1 - D} \quad (22)$$

Este tipo de conversor entrega una corriente de salida menor a la de entrada, puesto que el voltaje de salida es mayor al de entrada y debe compensar la potencia de salida, de manera que la potencia de entrada y de salida del conversor sean prácticamente iguales.

El periodo de conmutación se expresa de la siguiente manera:

$$T = \frac{1}{f} = t_1 + t_2 = \frac{\Delta I L}{V_{in}} + \frac{\Delta I L}{V_{out} - V_{in}} = \frac{\Delta I L V_{out}}{V_{in} (V_{out} - V_{in})} \quad (23)$$

De allí, es puede extraer el rizado de corriente en el inductor como:

$$\Delta I = \frac{V_{in} D}{f L} \quad (24)$$

Cuando el transistor conduce, el capacitor entrega la corriente de carga en tiempo  $t_1$ . La corriente media del capacitor durante  $t_1$  es  $I_C = I_{out}$  y el rizado de voltaje es:

$$\Delta V_C = V_C - V_C|_{t=0} = \frac{1}{C} \int_0^{t_1} I_C dt = \frac{1}{C} \int_0^{t_1} I_{out} dt = \frac{I_{out} t_1}{C} \quad (25)$$

De la ecuación 22 se obtiene:

$$t_1 = \frac{V_{out} - V_{in}}{V_{out} f} \quad (26)$$

Reemplazando  $t_1$  en la ecuación 25, se obtiene:

$$\Delta V_C = \frac{I_{out} D}{f C} \quad (27)$$

Esta topología brinda facilidad para controlar la conmutación del Mosfet, ya que el *source* si está conectado a tierra.

### 3. Conversores reductores-elevadores (Buck-Boost)

El voltaje de salida puede ser menor o mayor que el de entrada y sus polaridades son opuestas. También se conoce como regulador inversor. Proporciona una alta eficiencia de conversión. La corriente de entrada es discontinua.

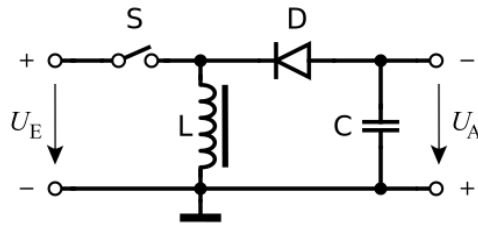


Figura 4: Esquema del convertor Buck-Boost. [11]

Las dos fases de este circuito son: primera fase, cuando el transistor conduce, el diodo tiene polarización inversa y la corriente aumenta y fluye por el inductor; y la segunda fase, cuando el transistor deja de conducir y la corriente fluye a través del inductor, el diodo, el capacitor y la carga.

Suponiendo que la corriente en el inductor crece de manera lineal desde  $I_1$  hasta  $I_2$  en el tiempo  $t_1$  cuando el transistor conduce, se obtiene:

$$V_{in} = L \frac{I_2 - I_1}{t_1} = L \frac{\Delta I}{t_1} \quad (28)$$

Por lo tanto:

$$t_1 = \frac{\Delta I L}{V_{in}} \quad (29)$$

Por otro lado, si la corriente disminuye linealmente de  $I_2$  a  $I_1$  en el tiempo  $t_2$  cuando el transistor deja de conducir, se obtiene:

$$V_{out} = -L \frac{\Delta I}{t_2} \quad (30)$$

Por lo tanto:

$$t_2 = \frac{-\Delta I L}{V_{out}} \quad (31)$$

Donde  $\Delta I = I_2 - I_1$  es el rizado de corriente pico a pico en el inductor. De las ecuaciones 28 y 30, se obtiene:

$$\Delta I = \frac{V_{in} t_1}{L} = \frac{-V_{out} t_2}{L} \quad (32)$$

Poniendo  $t_1$  y  $t_2$  en términos del ciclo de trabajo y el periodo, como  $t_1 = D T$  y  $t_2 = (1 - D)T$ , se obtiene el voltaje promedio a la salida del conversor:

$$V_{out} = -\frac{V_{in} D}{1 - D} \quad (33)$$

Para mantener la relación de potencia de  $P_{out} = P_{in}$ , cuando el conversor está elevando voltaje, también reduce la corriente; y cuando está reduciendo voltaje, también eleva la corriente.

El periodo de conmutación se expresa de la siguiente manera:

$$T = \frac{1}{C} = t_1 + t_2 = \frac{\Delta I L}{V_{in}} - \frac{\Delta I L}{V_{out}} = \frac{\Delta I L (V_{out} - V_{in})}{V_{in} V_{out}} \quad (34)$$

De allí, es puede extraer el rizado de corriente en el inductor como:

$$\Delta I = \frac{V_{in} D}{f L} \quad (35)$$

Cuando el transistor conduce, el capacitor entrega la corriente de carga en tiempo  $t_1$ . La corriente media del capacitor durante  $t_1$  es  $I_C = I_{out}$  y el rizado de voltaje es:

$$\Delta V_C = \frac{1}{C} \int_0^{t_1} I_C dt = \frac{1}{C} \int_0^{t_1} I_{out} dt = \frac{I_{out} t_1}{C} \quad (36)$$

De la ecuación 33 se obtiene:

$$t_1 = \frac{V_{out}}{(V_{out} - V_{in}) f} \quad (37)$$

Reemplazando  $t_1$  en la ecuación 36, se obtiene:

$$\Delta V_C = \frac{I_{out} D}{f C} \quad (38)$$

A pesar de que el mosfet no tiene su *source* conectada a tierra, es sencillo obtener circuitos equivalentes a este en donde el *source* si se conecte a tierra, facilitando el control de la conmutación.

4. **Conversores Cúk:** A su salida, el voltaje puede ser menor o mayor que el voltaje de entrada y su polaridad es opuesta. Su corriente de entrada es continua, tiene bajas pérdidas de conmutación y provee una alta eficiencia de conversión. Este conversor requiere de dos inductores, por lo que para aplicaciones de alta potencia se puede ocupar un espacio significativo de hardware.

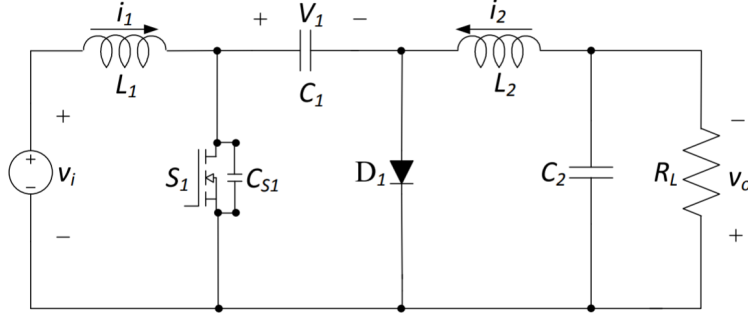


Figura 5: Esquema del convertor Cúk. [12]

Las dos fases de este circuito son: primera fase, en donde el transistor conduce, la corriente aumenta en el inductor  $L_1$ , el voltaje del capacitor  $C_1$  polariza inversamente el diodo y se descarga el capacitor en el circuito constituido por los dos capacitores, el inductor  $L_2$  y la carga; y la segunda fase, donde el transistor deja de conducir, el capacitor  $C_1$  se carga y la energía del inductor  $L_2$  se transfiere a la carga.

Suponiendo que la corriente en el inductor crece de manera lineal desde  $I_{L11}$  hasta  $I_{L12}$  en el tiempo  $t_1$  cuando el transistor conduce, se obtiene:

$$V_{in} = L_1 \frac{I_{L12} - I_{L11}}{t_1} = L_1 \frac{\Delta I_1}{t_1} \quad (39)$$

Por lo tanto:

$$t_1 = \frac{\Delta I_1 L_1}{V_{in}} \quad (40)$$

Por otro lado, si la corriente disminuye linealmente de  $I_{L12}$  a  $I_{L11}$  en el tiempo  $t_2$  cuando el transistor deja de conducir, se obtiene:

$$V_{in} - V_{C1} = -L_1 \frac{\Delta I_1}{t_2} \quad (41)$$

Por lo tanto:

$$t_2 = \frac{-\Delta I_1 L_1}{V_{in} - V_{C1}} \quad (42)$$

Donde  $\Delta I_1 = I_{L12} - I_{L11}$  es el rizado de corriente pico a pico en el inductor  $L_1$  y  $V_{C1}$  es el voltaje medio del capacitor  $C_1$ . De las ecuaciones 39 y 41, se obtiene:

$$\Delta I_1 = \frac{V_{in} t_1}{L_1} = \frac{-(V_{in} - V_{C1}) t_2}{L_1} \quad (43)$$

Poniendo  $t_1$  y  $t_2$  en términos del ciclo de trabajo y el periodo, como  $t_1 = D T$  y  $t_2 = (1 - D)T$ , se obtiene el voltaje promedio del capacitor  $C_1$ :

$$V_{C1} = \frac{V_{in}}{1 - D} \quad (44)$$

Suponiendo que la corriente en  $L_2$  incrementa de manera lineal desde  $I_{L21}$  hasta  $I_{L22}$  en el tiempo  $t_1$ , se tiene que:

$$V_{C1} + V_{out} = L_2 \frac{I_{L22} - I_{L21}}{t_1} = L_2 \frac{\Delta I_2}{t_1} \quad (45)$$

Por lo tanto:

$$t_1 = \frac{\Delta I_2 L_2}{V_{C_1} + V_{out}} \quad (46)$$

Por otro lado, si la corriente disminuye linealmente de  $I_{L_2 2}$  a  $I_{L_2 1}$  en el tiempo  $t_2$ , se obtiene:

$$V_{out} = -L_2 \frac{\Delta I_2}{t_2} \quad (47)$$

Por lo tanto:

$$t_2 = -\frac{\Delta I_2 L_2}{V_{out}} \quad (48)$$

Donde  $\Delta I_2 = I_{L_2 2} - I_{L_2 1}$  es el rizado de corriente pico a pico en el inductor  $L_2$ . De las ecuaciones 45 y 47, se obtiene:

$$\Delta I_2 = \frac{(V_{C_1} + V_{out}) t_1}{L_2} = -\frac{V_{out} t_2}{L_2} \quad (49)$$

Poniendo  $t_1$  y  $t_2$  en términos del ciclo de trabajo y el periodo, como  $t_1 = D T$  y  $t_2 = (1 - D)T$ , se obtiene el voltaje promedio del capacitor  $C_1$ :

$$V_{C_1} = -\frac{V_{out}}{D} \quad (50)$$

De las ecuaciones 44 y 50 se puede hallar el voltaje medio de salida del convertidor:

$$V_{out} = -\frac{D V_{in}}{1 - D} \quad (51)$$

Para mantener la relación de potencia de  $P_{out} = P_{in}$ , cuando el convertidor está elevando voltaje, también reduce la corriente; y cuando está reduciendo voltaje, también eleva la corriente.

El periodo de conmutación se expresa de la siguiente manera:

$$T = \frac{1}{f} = t_1 + t_2 = \frac{\Delta I_1 L_1}{V_{in}} - \frac{\Delta I_1 L_1}{V_{in} - V_{C_1}} = -\frac{\Delta I_1 L_1 V_{C_1}}{V_{in} (V_{in} - V_{C_1})} \quad (52)$$

De allí, se puede extraer el rizado de corriente en el inductor  $L_1$  como:

$$\Delta I_1 = -\frac{V_{in} (V_{in} - V_{C_1})}{f L_1 V_{C_1}} = \frac{V_{in} D}{f L_1} \quad (53)$$

El periodo de conmutación puede hallarse también con las ecuaciones 46 y 48:

$$T = \frac{1}{f} = t_1 + t_2 = \frac{\Delta I_2 L_2}{V_{C_1} + V_{out}} - \frac{\Delta I_2 L_2}{V_{out}} = -\frac{\Delta I_2 L_2 V_{C_1}}{V_{out} (V_{C_1} + V_{out})} \quad (54)$$

De allí, se puede extraer el rizado de corriente en el inductor  $L_2$  como:

$$\Delta I_2 = -\frac{V_{out} (V_{C_1} + V_{out})}{f L_2 V_{C_1}} = \frac{D V_{in}}{f L_2} \quad (55)$$

Cuando el transistor no conduce, el capacitor  $C_1$  se carga por la corriente de entrada durante el tiempo  $t_2$ . La corriente media de carga del capacitor  $C_1$  es  $I_{C_1} = I_{in}$  y el rizado de voltaje es:

$$\Delta V_{C_1} = \frac{1}{C_1} \int_0^{t_2} I_{C_1} dt = \frac{1}{C_1} \int_0^{t_2} I_{in} dt = \frac{I_{in} t_2}{C_1} \quad (56)$$

De la ecuación 51 se obtiene:

$$t_2 = \frac{V_{in}}{(V_{in} - V_{out}) f} \quad (57)$$

Reemplazando  $t_2$  en la ecuación 56, se obtiene:

$$\Delta V_{C_1} = \frac{I_{in} (1 - D)}{f C_1} \quad (58)$$

Si se supone que el rizado de corriente en la carga es despreciable, se tiene que  $\Delta I_{L_2} = \Delta I_{C_2}$ . La corriente media de carga del capacitor  $C_2$ , que fluye en  $T/2$ , es  $I_{C_2} = \Delta I_2/4$ ; y el rizado de voltaje en el  $C_2$  es:

$$\Delta V_{C_2} = \frac{1}{C_2} \int_0^{T/2} I_{C_2} dt = \frac{1}{C_2} \int_0^{T/2} \frac{\Delta I_2}{4} dt = \frac{\Delta I_2}{8 f C_2} = \frac{D V_{in}}{8 C_2 L_2 f^2} \quad (59)$$

De igual manera que en el convertor elevador, el convertor Cúk proporciona un facil control de conmutación del Mosfet, ya que el *source* si está conectado a tierra.

Lo mostrado anteriormente son los esquemas básicos de los convertidores DC-DC, de los cuales se han creado gran cantidad de variaciones, adaptaciones y mejoras para diferentes aplicaciones, pero partiendo de los mismos principios que estos.

## 2.7. Algoritmos MPPT

En sistemas fotovoltaicos, se utilizan los algoritmos MPPT para encontrar el balance de voltaje y corriente con el cual se obtiene la máxima potencia del arreglo fotovoltaico. Cada algoritmo opera bajo principios distintos, tienen distinto tiempo de convergencia y distinta complejidad, por lo que para cada aplicación se debe seleccionar el algoritmo mas adecuado tomando en cuenta estas características.

Estos algoritmos se basan en las curvas de comportamiento de los módulos fotovoltaicos, las cuales se muestran en la **figura 6**.

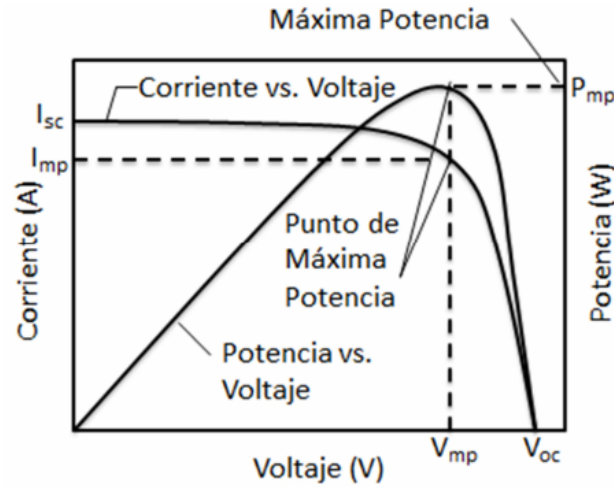


Figura 6: Curvas de potencia vs. voltaje y corriente vs. voltaje. [13].

Dependiendo de las condiciones de irradiancia y temperatura a las que se somete el panel solar o el arreglo fotovoltaico, las magnitudes de las curvas de la **figura 6** cambian, es decir, que el punto de máxima potencia no es fijo, sino que depende de dichas variables y está constantemente en movimiento. Es por esta razón que es necesario el rastreo de punto de máxima potencia.

Para entender el cambio en las magnitudes de las curvas de la **figura 6** es necesario analizar el modelo de circuito equivalente de un panel fotovoltaico. En la **figura 7** se muestra dicho modelo.

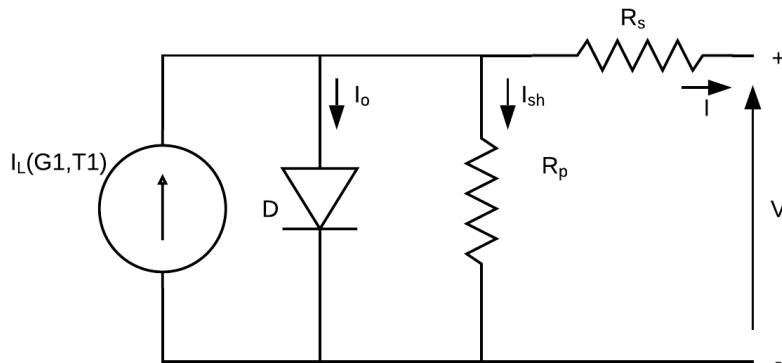


Figura 7: Modelo circuital de un panel solar.

El comportamiento de este modelo está descrito por las ecuaciones 60, 61 y 62. [14]

$$I = I_L(G1, T1) - N_p I_o \left[ e^{\left( \frac{V + IR_s}{V_t} - 1 \right)} \right] - \frac{V + IR_s}{R_p} \quad (60)$$

$$V_t = m N_s k \frac{T_1 + 273}{q_e} \quad (61)$$

$$I_o = \frac{I_{SC} - \frac{V_{OC}}{R_p}}{e^{\frac{V_{OC}}{V_t}} - 1} \quad (62)$$

En donde:

- $N_s$  es el número de celdas en serie.
- $N_p$  es el número de celdas en paralelo.
- $k$  es la constante de Boltzman.
- $q_e$  es la carga del electrón.
- $m$  es el factor de idealidad del diodo, comprendido entre 1 y 2.
- $T_1$  es la temperatura del panel solar medida en  $^{\circ}C$ .
- $R_s$  es la resistencia serie.
- $R_p$  es la resistencia paralela.
- $I_L(G1, T1)$  es la corriente fotogenerada, aproximadamente igual a la corriente de corto circuito  $I_{SC}(G1, T1)$ .
- $I_o$  es la corriente inversa de saturación del diodo.
- $V_{OC}$  es el voltaje de circuito abierto.

Para ejemplificar la manera en que la irradiancia y la temperatura afectan estas curvas, se muestra la **figura 8**.

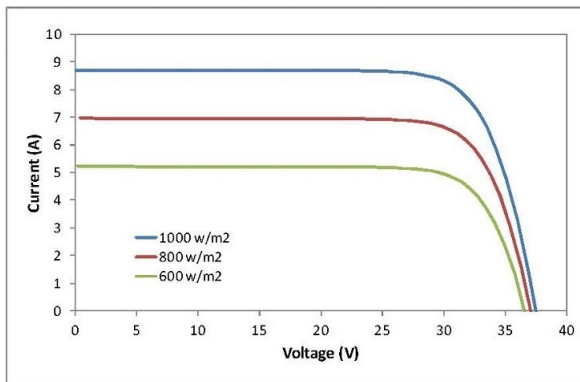


Figura Fa. Curvas I-V para distintos valores de la irradiancia solar  
Temperatura constante de la célula ( $25^{\circ}C$ )

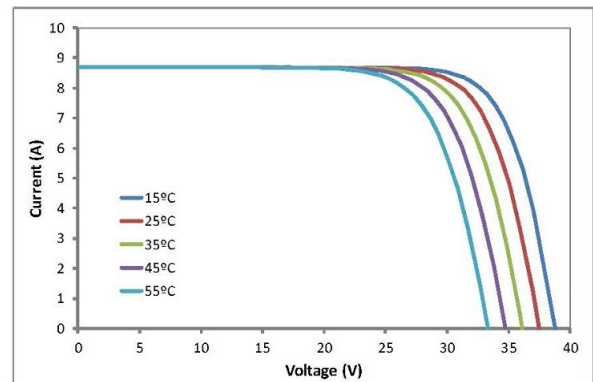


Figura Fb. Curvas I-V para distintas temperatura de la célula  
Valor de irradiancia solar constante ( $1000\text{ W/m}^2$ )

Figura 8: Cambio de curvas de voltaje vs. corriente debido a Irradiancia y Temperatura. [15]

Los algoritmos MPPT tienen diferentes formas de encontrar el punto de máxima potencia, algunos utilizan sensores de irradiancia y temperatura para calcular dicho punto, mientras que otros pueden usar sensores de voltaje y corriente y calculan la potencia de

salida acoplando la impedancia de carga hasta ubicarse en el pico de la curva de potencia; entre otra infinitud de maneras.

A continuación se explican los algoritmos en los cuales se basa este proyecto.

### 2.7.1. Algoritmo P&O/Hill Climbing

Este algoritmo mide voltaje y corriente a la salida del arreglo fotovoltaico, calcula la potencia y toma decisiones basadas en este cálculo. La manera en la que opera es simple:

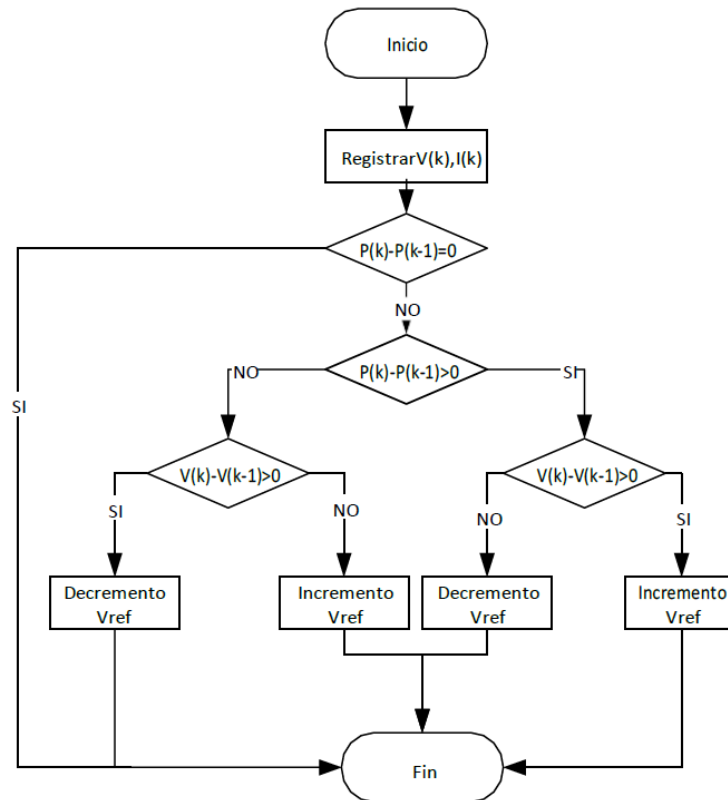


Figura 9: Diagrama de flujo del algoritmo P&O. [16]

Este método trabaja recorriendo la curva de potencia en pequeños pasos. Primero define un punto de prueba (generalmente para este método se toma como punto inicial  $V = 0$ ) y se calcula la potencia. Si la potencia obtenida es mayor a la obtenida en la iteración anterior, verifica si también el voltaje es mayor que en la iteración anterior; de ser así, se mueve hacia la derecha en el eje de voltaje para hacer otra medición, y si no, entonces se mueve a la izquierda en el eje de voltaje. En caso de que la potencia en la iteración anterior sea mayor que en la actual, se mueve hacia la derecha en el eje de voltaje si el voltaje anterior es mayor al actual, de lo contrario se mueve a la izquierda. En la siguiente figura se muestra como se recorre la curva de potencia:

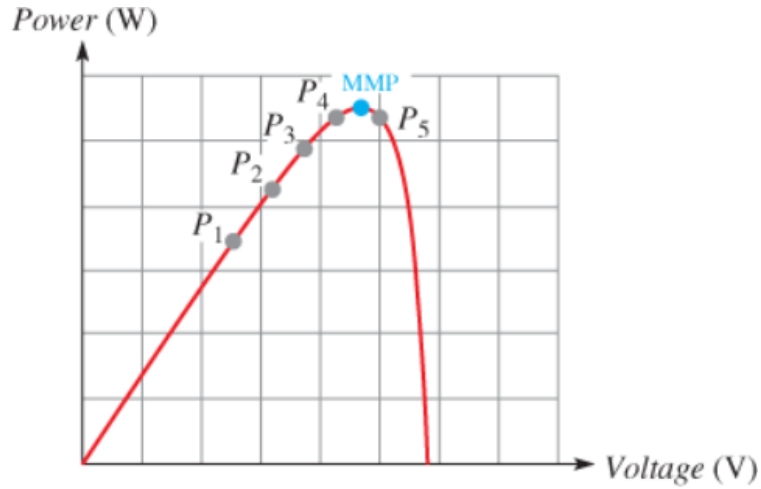


Figura 10: Recorrido de curva de potencia con P&O. [17]

La mayoría de controladores de carga comerciales utilizan este método de rastreo de punto de máxima potencia debido a su simplicidad, ya que comparado a otras técnicas, no es muy exigente respecto al hardware que requiere ni a la complejidad de su algoritmo.

### 2.7.2. Algoritmo de salto selectivo (Selective Hopping o SH)

El algoritmo de salto selectivo está basado en el algoritmo P&O. Consiste en un método iterativo en el que se definen dos puntos de voltaje y se comparan las potencias obtenidas con dichos parámetros. En donde se obtenga la potencia menor, se define el punto de corte de la sección que se eliminará del rango de búsqueda de la siguiente iteración. Además, se definirá otro punto de voltaje en sentido contrario al que se tomó el punto de potencia mas baja respecto al de potencia más alta. El diagrama de flujo que describe este algoritmo se muestra en la **figura 11**.

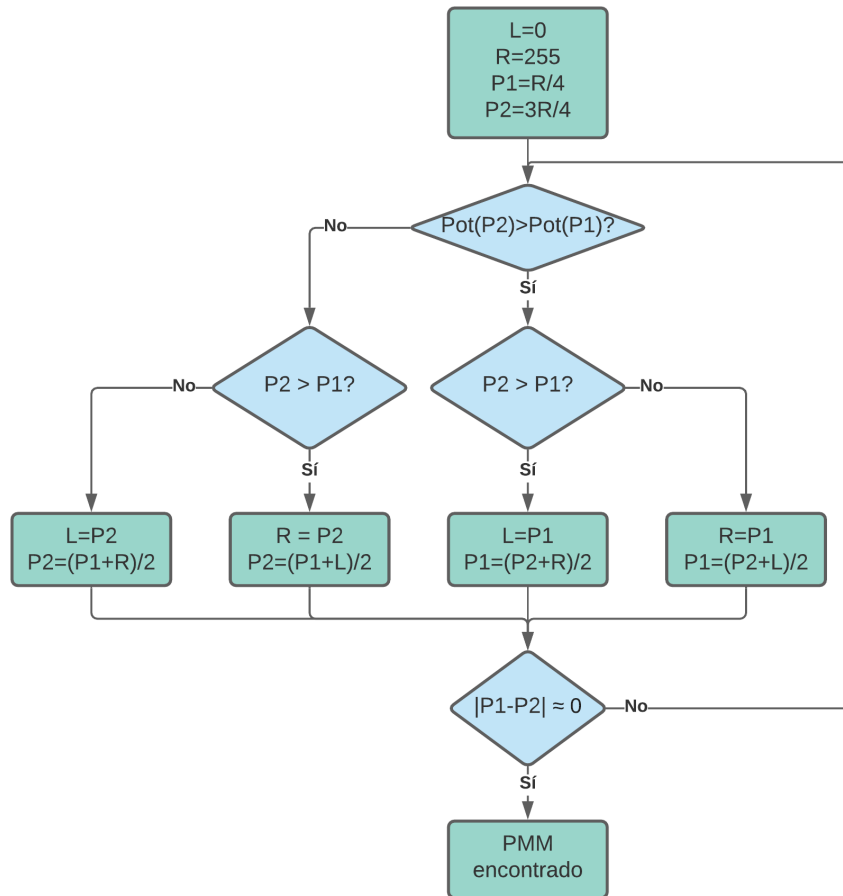


Figura 11: Diagrama de flujo del método de salto selectivo (SH).

Donde:

- $L$ : es el límite mínimo de búsqueda
- $R$ : es el límite máximo de búsqueda
- $P1$  y  $P2$ : son los puntos de voltaje para los cuales se mide la potencia

En las condiciones iniciales los valores de  $L$  y  $R$  se deben al número posible de puntos de prueba que se pueden utilizar. Para el caso particular que se muestra en el diagrama de flujo, serían 256 puntos de prueba posibles, por lo que a cada valor entre 0 y 255 le corresponde un punto del eje de voltaje en la curva de potencia del arreglo fotovoltaico.

La búsqueda del punto de máxima potencia sobre la curva de potencia del panel efectuada por el algoritmo se muestra en la **figura 12**.

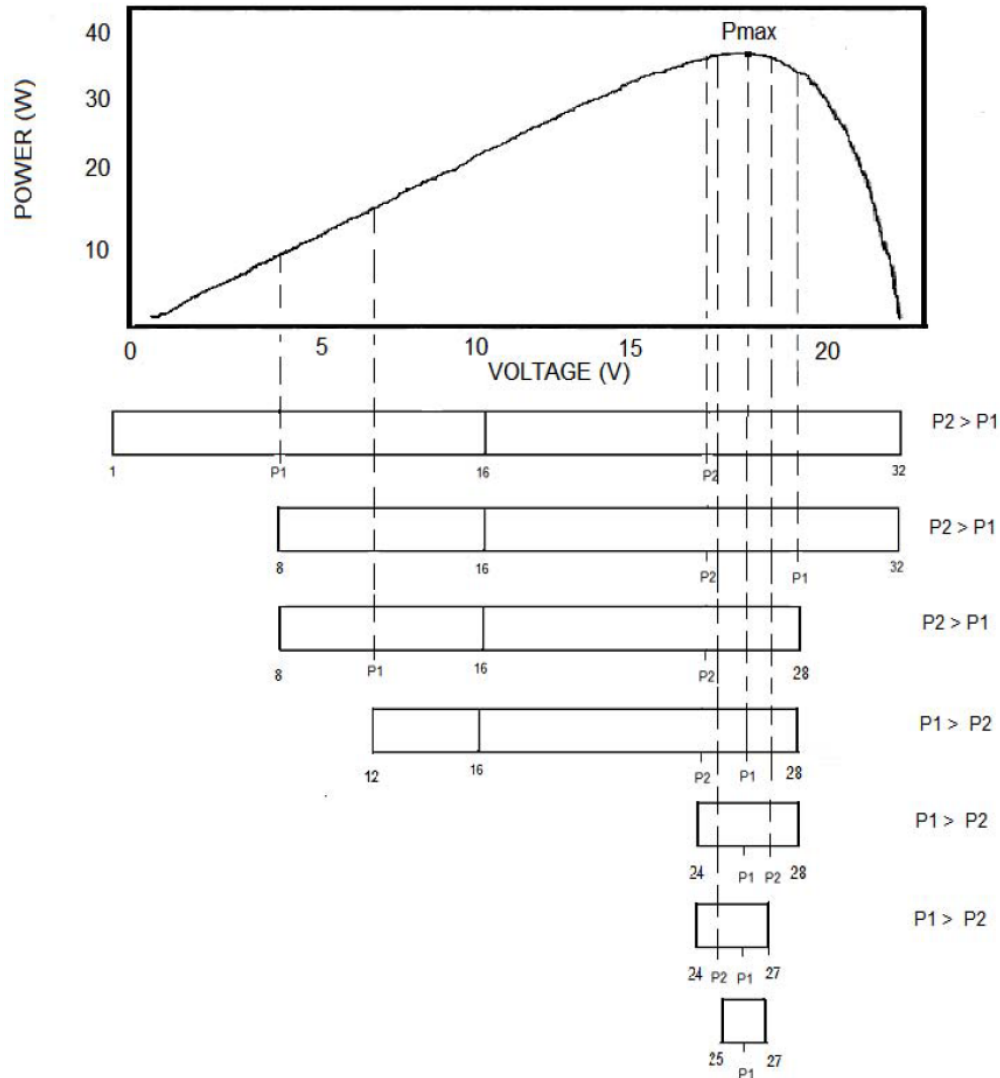


Figura 12: Búsqueda de punto de máxima potencia por el método de salto selectivo (SH). [2].

Con este método se descartan rápidamente partes de la curva que no contienen el punto de máxima potencia, a diferencia del método P&O que en cada iteración puede descartar únicamente su punto anterior de prueba.

Esta técnica no se ha implementado con convertidores de potencia, únicamente manipula la carga para sintonizar el punto de máxima potencia utilizando un arreglo de resistencias en paralelo conmutadas a través de mosfets, es decir, se alimentan cargas de prueba para hallar el punto de máxima potencia pero no se está dando una aplicación real a este método, por lo tanto, es un rastreo de punto de máxima potencia de manera *offline*.

## 2.8. Uso de convertidores DC-DC con algoritmos MPPT

Para rastrear el punto de máxima potencia de un arreglo fotovoltaico de manera *online*, se utilizan convertidores DC-DC, los cuales permiten ubicarse en distintos puntos de la curva de potencia variando su ciclo de trabajo. En la siguiente figura se muestra el esquema de uso de los convertidores para aplicaciones MPPT:

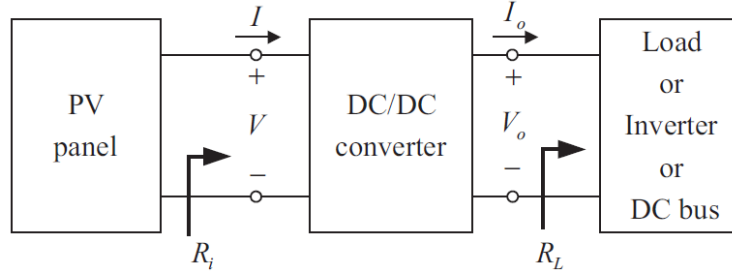


Figura 13: Esquema de uso de convertidores DC-DC para MPPT [18].

Existe una relación entre las resistencias equivalentes mostradas en la figura denotadas como  $R_i$  y  $R_L$ . Esta relación depende de la topología de convertidor utilizada. A continuación se muestran las ecuaciones que relacionan estas resistencias, en topologías buck, boost y buck/boost respectivamente:

$$R_i = \frac{R_L}{D^2} \quad (63)$$

$$R_i = (1 - D)^2 R_L \quad (64)$$

$$R_i = \frac{(1 - D)^2}{D^2} R_L \quad (65)$$

Estar en el punto de máxima potencia implica tener  $V_{mpp}$  e  $I_{mpp}$  a la salida del panel, con esto se puede definir  $R_i = R_{mpp} = \frac{V_{mpp}}{I_{mpp}}$ . De esta manera, al variar el ciclo de trabajo del convertidor se puede obtener el valor de  $R_i$  deseado a partir de la carga  $R_L$ .

En caso de que el  $R_i$  obtenido sea menor a  $R_{mpp}$ , la potencia obtenida será un punto de la parte de pendiente positiva de la curva de potencia contra voltaje. Si el  $R_i$  obtenido es mayor a  $R_{mpp}$ , la potencia obtenida será un punto de la parte de pendiente negativa. Es de esta manera que variando el ciclo de trabajo se pueden escoger puntos de prueba y ejecutar los algoritmos.

# Capítulo 3

## 3. Diseño e implementación

### 3.1. Planteamiento del sistema

El esquema planteado para este proyecto es una simplificación del esquema típico mostrado en la **figura 1**, en donde no se hará uso de baterías ni de un inversor (convertor DC/AC), sino que se alimentarán directamente unas cargas DC con la potencia máxima que el panel pueda entregar.

El esquema se muestra a continuación:

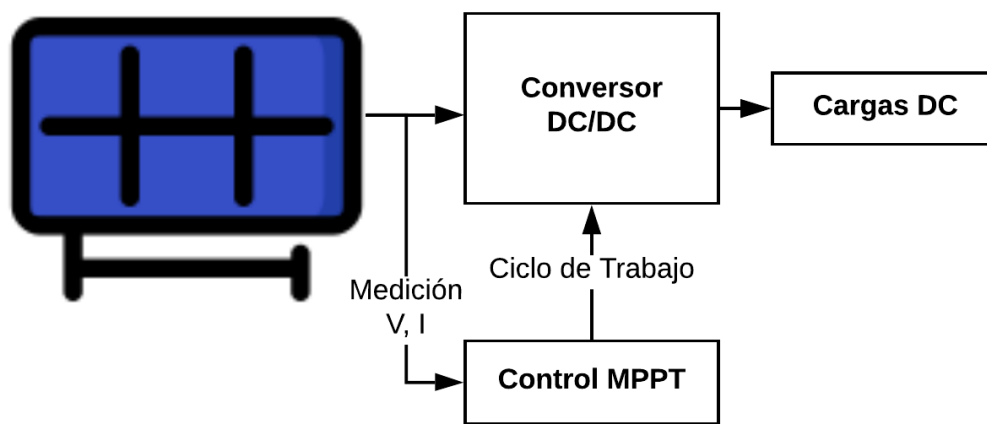


Figura 14: Esquema base del proyecto.

Este esquema es válido para la prueba del MPPT con el convertor DC-DC, ya que con el esquema típico se ejecutan los rastreos teniendo en cuenta la resistencia equivalente a la salida del convertor, mientras que con este esquema simplificado hay una carga directamente conectada a la salida del convertor.

### 3.2. Características del panel solar

El panel solar que se utilizará es el YL255P-29b de Yingli Solar, cuyo comportamiento está descrito por las siguientes curvas:

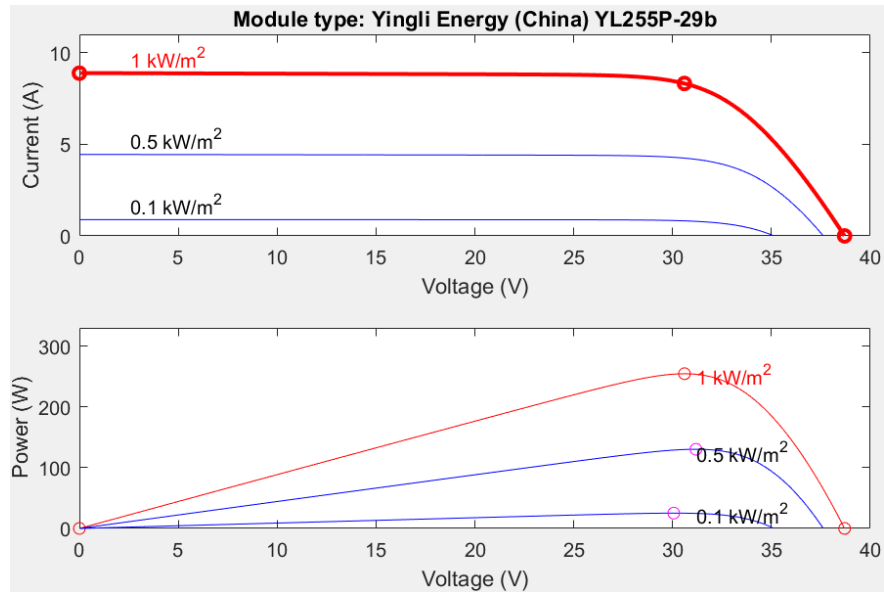


Figura 15: Curvas del panel a 25°C y diferentes valores de irradiancia. Fuente: Matlab Simulink.

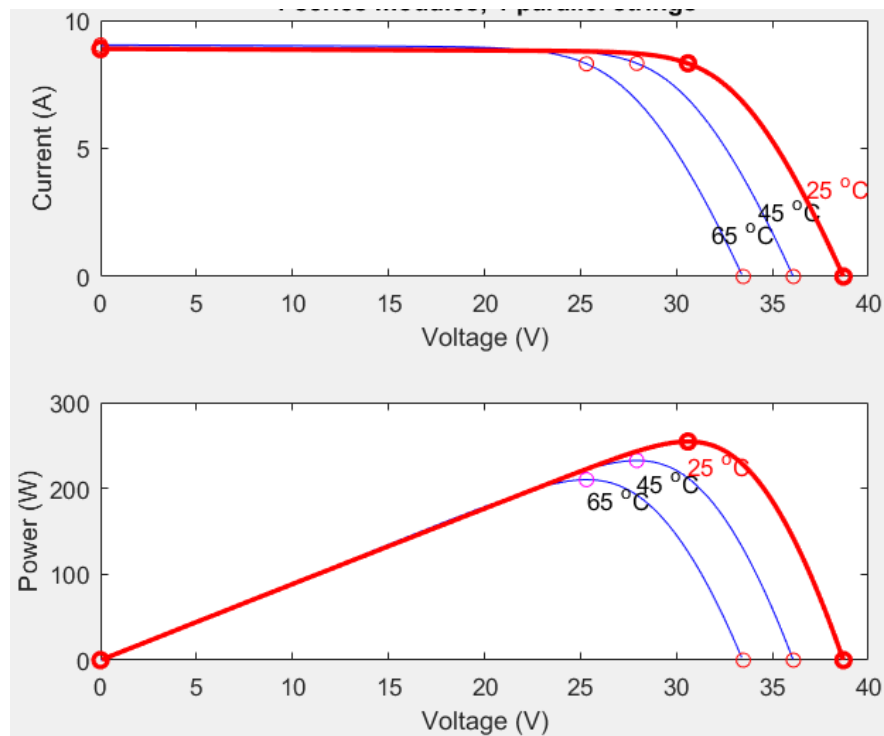


Figura 16: Curvas del panel con irradiancia de  $1\text{ kW/m}^2$  y diferentes valores de temperatura. Fuente: Matlab Simulink.

Como se puede observar en las curvas, el voltaje máximo de circuito abierto del panel es 38.7VDC y su corriente máxima de corto circuito es 8.88A. En su punto de máxima potencia alcanza un voltaje de 30.6VDC y una corriente de 8.32A, para un total de 254.6W

de potencia.

El panel consta de 60 celdas de silicio policristalino de 156mm x 156mm cubiertas por un vidrio templado de texturizado de 3.2mm de espesor. El módulo provee una eficiencia de hasta 15.6%. [19]

### 3.3. Características de las cargas

El objetivo principal es trabajar siempre en el punto de máxima potencia del panel solar, por lo que se requiere una carga que consuma la máxima potencia disponible en todo momento. El panel provee una potencia máxima de 255W, por esta razón, se plantea utilizar seis cargas de 35W y dos de 20W, siendo esto lo equivalente a una carga de 250W. Dichas cargas son bombillos halógenos que trabajan a un voltaje máximo de 12VDC.

Debido a los niveles de voltaje que entrega el panel solar ( $V_{mpp} = 30,6V$ ), se decidió conectar los bombillos de la siguiente manera:

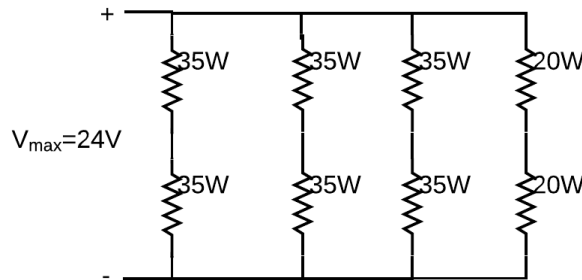


Figura 17: Esquema de conexión de las cargas DC.

Con este esquema de conexión, el voltaje máximo a la salida del conversor deberá ser de 24V, por lo que se facilitará la reducción de voltaje efectuada por este; lo cual beneficia a la eficiencia de conversión.

Observando la **ecuación 63** se puede concluir que es necesario que la resistencia equivalente de la carga sea menor que la resistencia MPP del panel solar. En este caso, las cargas de 35W operan a un máximo de 12V, por lo que se calcula que su resistencia máxima será de:  $R_{35W} = V^2/P = (12V)^2/35W = 4,11\Omega$ . Por otro lado, las cargas de 20W también operan a un máximo de 12V, por lo que se calcula su resistencia máxima como:  $R_{20W} = V^2/P = (12V)^2/20W = 7,2\Omega$ . Con esto se calcula la resistencia equivalente máxima con todas las cargas como:

$$R_{Lmax} = \frac{1}{\frac{1}{4,11+4,11} + \frac{1}{4,11+4,11} + \frac{1}{4,11+4,11} + \frac{1}{7,2+7,2}} = 2,3\Omega \quad (66)$$

La resistencia máxima MPP del panel es  $R_{mpp} = V_{mpp}/I_{mpp} = 30,6V/9,32A = 3,68\Omega$ , por lo que la carga cumple con el requerimiento.

### 3.4. Diseño e implementación del convertor

#### 3.4.1. Diseño del convertor

El convertor que se utilizará será de topología Buck, su esquema se puede observar en la siguiente figura:

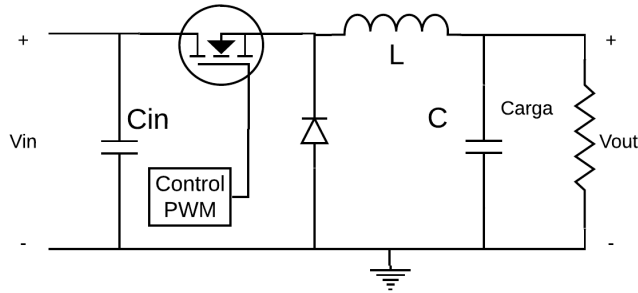


Figura 18: Topología de un convertor DC/DC tipo Buck.

A la salida de este convertor siempre habrá un voltaje inferior al voltaje de entrada, pero para cumplir  $P_{in} \approx P_{out}$ , la corriente de salida de este convertor es mayor a la corriente de entrada.

El ciclo útil de la señal PWM que controla el mosfet, define que voltaje de salida hay respecto al de entrada de acuerdo a la relación  $V_{out} = D * V_{in}$ , donde  $D$  es el ciclo útil y su valor está entre 0 y 1.

En la topología de la **figura 18** se muestra un mosfet de lado superior, el cual en la práctica dificulta el control del  $V_{GS}$  ya que el *source* del mosfet no está conectado a tierra. Por esta razón se decidió optar por modificar la topología y utilizar el mosfet de lado inferior, lo cual evita la necesidad de circuitería adicional para controlar el disparo del mosfet. El circuito resultante se muestra en la siguiente figura:

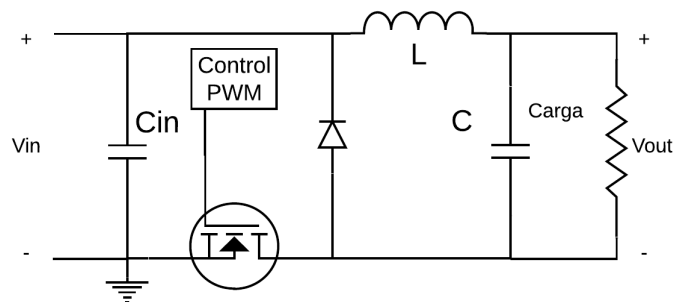


Figura 19: Topología de convertor Buck con mosfet de lado inferior.

#### Componentes pasivos:

Para la elección de los componentes del circuito, se utilizaron las siguientes ecuaciones [8]:

$$C = \frac{\Delta I}{8\Delta V_c F_{sw}} \quad (67)$$

$$L = \frac{V_{out}(V_{in} - V_{out})}{\Delta I F_{sw} V_{in}} \quad (68)$$

En donde  $\Delta I$  es el rizado de corriente que pasa por el inductor L,  $\Delta V_c$  es el rizado de voltaje en el capacitor C; y  $F_{sw}$  es la frecuencia de conmutación del mosfet. Los parámetros utilizados para obtener el valor mínimo de inductancia y capacitancia se muestran en el siguiente cuadro:

Cuadro 1: Parámetros de diseño del conversor

Parám	Valor	Descripción
$V_{in}$	$V_{inmax} = 30,6V$	Voltaje MPP del panel solar
$V_{out}$	$V_{outmax} = 24V$	Voltaje máximo de salida del conversor
$\Delta V_c$	$0,01V_{outmax} = 0,01 * 24 = 0,24$	Rizo de 1% del voltaje máximo de salida
$\Delta I$	$0,08I_{inmax} = 0,08 * 8,32 = 0,71$	Rizo de 8% de corriente MPP de entrada
$F_{sw}$	$72,6kHz$	Frecuencia fija de conmutación

Con estos parámetros se realizó el cálculo de los valores mínimos de inductancia y capacitancia que cumplen dichas especificaciones. Se obtuvieron los siguientes resultados:

Cuadro 2: Cálculo de valores mínimos de C y L del conversor.

Componente	Valor calculado
C	$4,8\mu F$
L	$110\mu H$

Un criterio muy importante para escoger los capacitores es su **resistencia serie equivalente (ESR)**, puesto que es un parámetro clave para determinar el nivel de ruido de salida que manejará el conversor. Tanto el capacitor de entrada como el de salida generan ruido debido a que su ESR interactúa con la corriente AC que fluye a través de ellos [20], por lo que es imprescindible utilizar capacitores en los cuales el valor ESR sea muy pequeño. Para el caso del capacitor de entrada también es muy importante su **inductancia serie equivalente (ESL)**, ya que si esta es muy grande, se genera un ruido de alta frecuencia muy considerable al crear un circuito oscilatorio (tanque) con las inductancias parásitas de las pistas del circuito impreso y los componentes no ideales de los demás elementos del circuito [20]. Idealmente se debería utilizar un capacitor comercial tipo chip con bajos ESR y ESL, sin embargo, conseguir un capacitor con esas características es complicado, por lo que se decidió utilizar un capacitor con bajo ESR y un filtro LC a la salida del conversor para atenuar en lo posible el ruido debido al ESL del capacitor de entrada. Adicionalmente, se añadió un pequeño inductor en serie con la entrada de voltaje, de modo que junto al

capacitor de entrada se crea un filtro de entrada LC para el convertor y evitar el ruido que puede propagarse hasta la fuente de alimentación.

El circuito resultante se muestra en la siguiente figura:

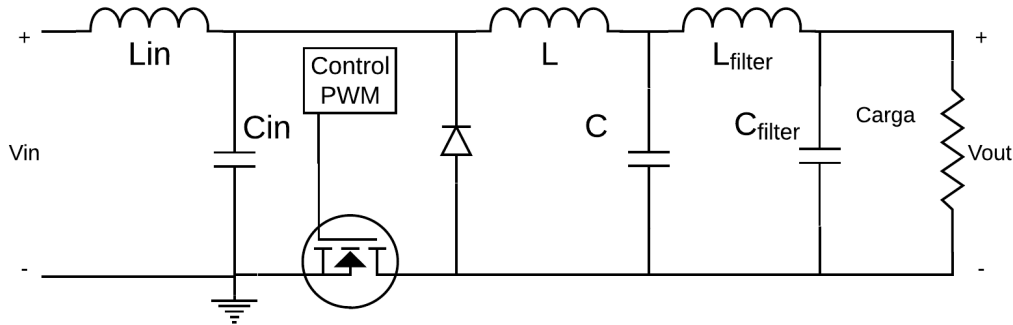


Figura 20: Convertor Buck con filtros LC.

Los componentes mencionados anteriormente se escogieron por ensayo y error entre los componentes rescatados de una fuente de computador desechada, pero respetando los valores mínimos de diseño de capacitancia e inductancia. Se hablará mas a detalle de esto en la sección de implementación del convertor (3.4.2).

#### MOSFET:

Para la conmutación se escogió el mosfet NCE8580, ya que es un MOSFET canal N diseñado para trabajar con control PWM de alta frecuencia, altas corrientes y una resistencia  $R_{DS\_on}$  muy pequeña.

Las características mas importantes de este MOSFET se muestran en la siguiente tabla:

Cuadro 3: Características principales del MOSFET NCE8580. [21]

Parámetro	Valor
Voltaje Drain-Source ( $V_{DS}$ )	85V (máximo)
Corriente continua de Drain ( $I_D$ )	80A (25°C) y 60A (100°C) (máximo)
Pulso de corriente de Drain ( $I_{DM}$ )	320A (máximo)
Disipación de potencia máxima ( $P_D$ )	170W (máximo)
Resistencia Drain-Source en estado ON ( $R_{DS\_on}$ )	8,5mΩ (máximo)
Voltaje forward del diodo ( $V_{SD}$ )	1,2V (máximo)
Tiempo de retardo de encendido ( $t_{d(on)}$ )	18nS (típico)
Tiempo de subida de encendido ( $t_r$ )	12nS (típico)
Tiempo de retardo de apagado ( $t_{d(off)}$ )	56nS (típico)
Tiempo de caída de apagado ( $t_f$ )	15nS (típico)

#### Diodo:

El diodo de *freewheeling* escogido fue el F30S60S, puesto que cumple con las exigencias de frecuencia y corrientes continuas con las que debe trabajar. Además, es comercialmente facil

de adquirir.

Las características mas relevantes de este diodo se listan en el siguiente cuadro:

Cuadro 4: Características principales del diodo F30S60S. [22]

Parámetro	Valor
Pico repetitivo de voltaje inverso ( $V_{RRM}$ )	600V (máximo)
Voltaje inverso pico de trabajo ( $V_{RWM}$ )	600V (máximo)
Voltaje de ruptura ( $V_R$ )	600V (máximo)
Corriente forward promedio ( $I_{F\_AV}$ )	30A (máximo)
Pico de corriente no repetitiva ( $I_{FSM}$ )	300A (máximo)
Voltaje forward ( $V_F$ )	2V ( $I_F = 15A$ )
Tiempo de recuperación inversa ( $t_{rr}$ )	40nS ( $T = 25^\circ$ )(máximo) y 75nS ( $T = 125^\circ$ )(típico)

### Controlador de disparo:

Para activar el disparo del mosfet se utilizó un driver de referencia MCP14E7, el cual recibe la señal de PWM y la transmite al *gate* del mosfet. El esquema de las conexiones del driver se muestran en la siguiente figura:

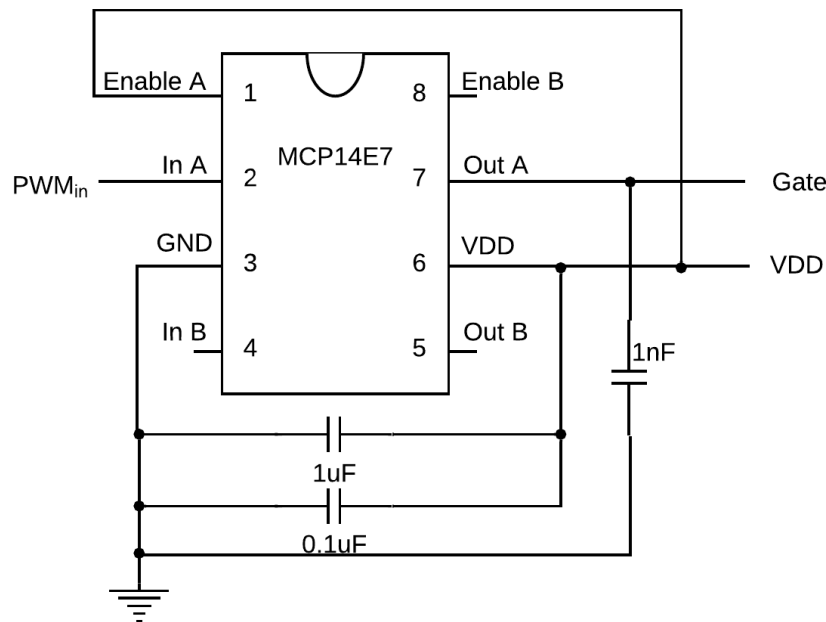


Figura 21: Esquema de conexión del driver MCP14E7 en el convertidor.

Los capacitores de desacoplo de  $1\mu F$  y  $0,1\mu F$ , deben ser de película plástica y cerámico respectivamente, debido a recomendaciones en la hoja de datos del driver. Los pines 4, 5 y 8 al no ser utilizados, se dejan sin conexión.

La alimentación del driver no proviene del panel solar, sino de una fuente DC de la que se

hablará en la **sección 3.9**. Por este motivo el convertor funcionará sin un mínimo de voltaje de entrada.

Esta etapa del circuito se incluyó en la misma PCB que el convertor, mientras que el control PWM se implementó en una PCB por aparte.

**Circuito final:**

El esquemático completo de lo que será la PCB del convertor se muestra a continuación:

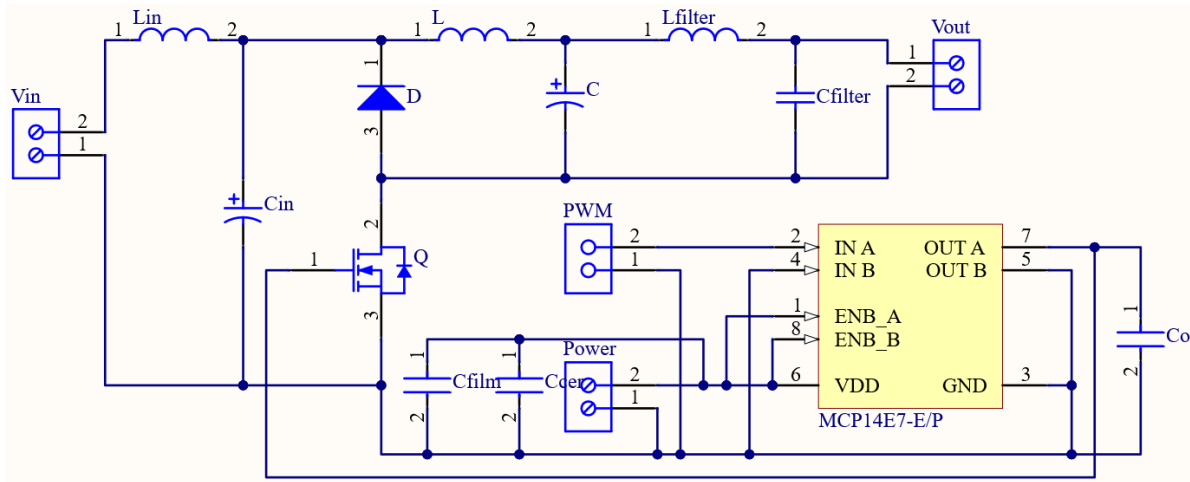


Figura 22: Esquemático final de diseño del convertor.

**3.4.2. Implementación del convertor**

**Componentes pasivos:**

Conociendo los valores mínimos de inductancia y capacitancia, se procedió a escoger y reutilizar los componentes de una fuente de computador desechada.

Para el caso de la **inductancia L**, se tomó un núcleo magnético toroidal de referencia CS270075 y se enrolló un alambre de cobre esmaltado de 1.1mm de diámetro con un total de 36 vueltas, logrando una inductancia de  $127\mu H$ . Se escogió dicho núcleo debido a su alto índice de auto-inducción ( $A_L = 94nH/N^2$ ), lo que implica que se pueden lograr grandes valores de inductancia sin tener que enrollar demasiado el alambre de cobre. La relación entre las variable mencionadas se indica mediante la siguiente expresión:

$$L = A_L * 10^{-9} * N^2 \tag{69}$$

Donde  $L$  es el valor de inductancia,  $A_L$  es el índice de auto-inductancia y  $N$  es el número de vueltas.

Para escoger el **capacitor C**, se tuvo en consideración la necesidad de que tenga baja ESR para disminuir en lo posible el ruido a la salida del convertor, por consiguiente se escogió un capacitor de aluminio Low-ESR de  $470\mu F$  de 50V.

De igual manera, como **capacitor Cin**, se escogió un capacitor de aluminio Low-ESR de  $470\mu F$  de 50V, debido a que fue el capacitor que menor ruido generó.

En cuanto a las **inductancias Lin** y **Lfilter**, se utilizaron inductores de  $1\mu H$ ; y para completar el filtro LC a la salida del convertor, se utilizó como **capacitor Cfilter** un capacitor supresor de ruido de referencia MKP-X2 de  $0,47\mu F$ .

Como ya se había comentado anteriormente, estos componentes (Lin, Cin, C, Lfilter y Cfilter) se escogieron por método de ensayo y error, buscando un equilibrio entre bajo ruido y alta eficiencia de conversión; mientras que para la inductancia L lo que se escogió fue el núcleo toroidal con mayor índice de auto-inductancia.

### Circuito impreso:

Para el diseño de la PCB se tuvieron en cuenta las características de la tarjeta donde se planteó la fabricación y la corriente máxima que fluirá en el circuito para elegir el ancho de las pistas.

- La tarjeta es una lámina FR-4 con espesor de cobre de  $1\text{oz}/\text{ft}^2$ , de referencia KB-6160; que resiste un máximo de  $130^\circ\text{C}$ .
- La corriente máxima del circuito es la corriente de salida máxima, ya que se trata de un convertor Buck. Esta corriente es:  $\frac{P_{outmax}}{V_{outmax}} = \frac{250\text{W}}{24\text{V}} = 10,42\text{A}$

Para calcular el ancho de pista mínimo para que fluya esta corriente sin inconveniente, se utilizó la calculadora de ancho de trazas de PCB de Digikey. Esta es una herramienta online disponible en [www.digikey.com](http://www.digikey.com).

Los resultados fueron los siguientes:

**Corriente**

 A

**Espesor**

 oz/ft<sup>2</sup>

**Aumento de temperatura**

 °C

Ancho mínimo de traza

3.909063011 mm



Capas externas en el aire

Ancho de traza requerido

= 3,909063011 mm

Figura 23: Ancho de pista de PCB mínimo para corriente de 10.42A y temperatura de  $30^\circ\text{C}$ .

El convertor fue implementado con un ancho de pista de  $4\text{mm}$  (ver diseño de PCB en **anexo 6.1**). El resultado se muestra en la siguiente figura:

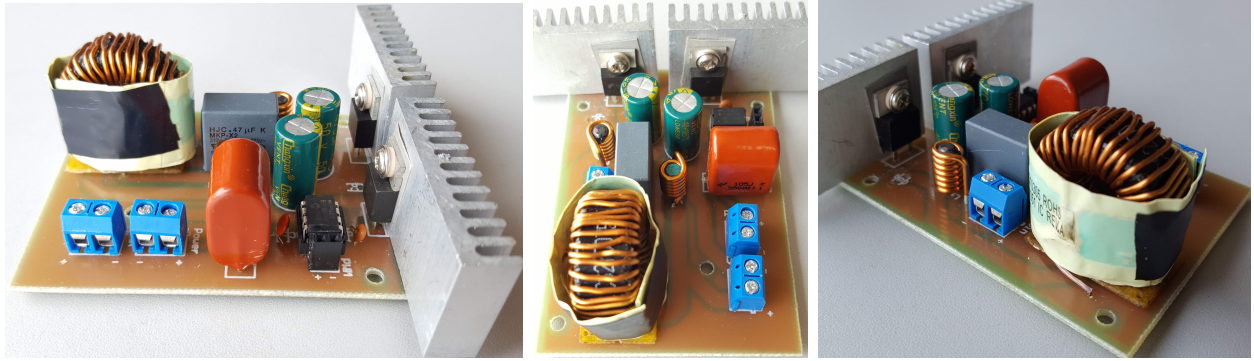


Figura 24: Conversor implementado.

### 3.4.3. Pruebas del conversor

Para probar el funcionamiento del conversor, se utilizaron cargas de distinta potencia para evaluar la eficiencia y el ruido del conversor con cada carga y diferentes ciclos de trabajo. Estas pruebas se realizaron utilizando fuentes de voltaje variable, entregando a la entrada del conversor voltajes cercanos a  $30,6V$ , el cual es el voltaje de máxima potencia del panel solar.

#### Pruebas de potencia:

Con una carga de  $140W$  se obtuvieron los siguientes resultados (tabla de datos en **anexo 6.2**):

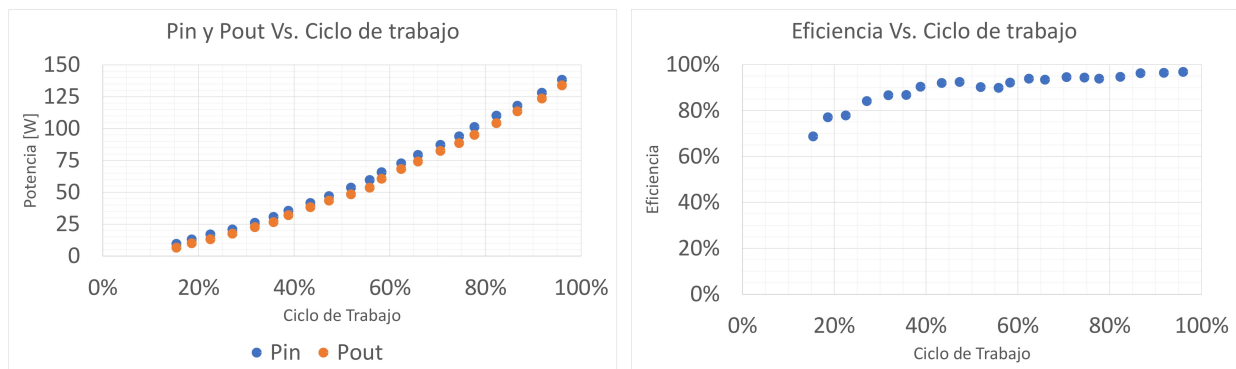


Figura 25: Prueba de eficiencia de conversión con carga de  $140W$ .

Mientras que con la carga con la que se implementará el algoritmo MPPT, que es de  $250W$ , se obtuvieron los siguientes resultados (tabla de datos en **anexo 6.3**):

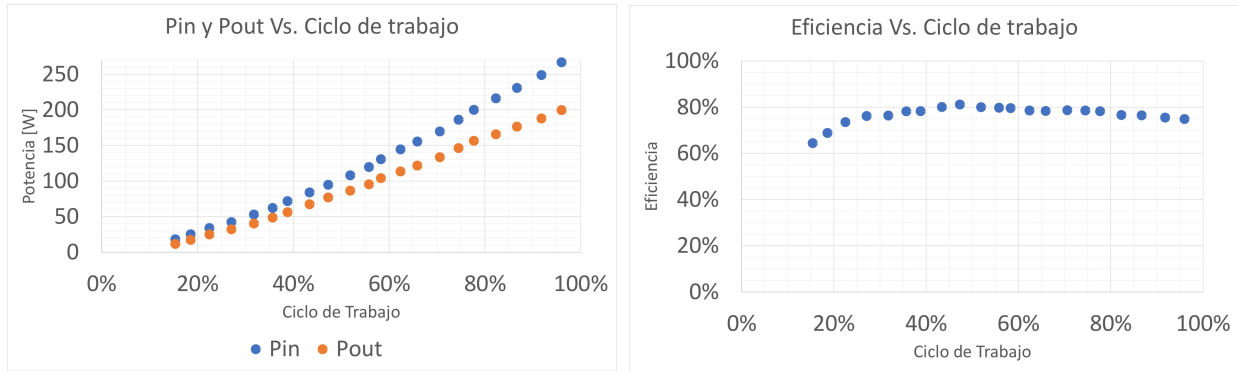


Figura 26: Prueba de eficiencia de conversión con carga de 250W.

Como se puede observar en las figuras 25 y 26, existe una diferencia considerable en la eficiencia de conversión cuando se aumenta la potencia de la carga, con la carga de 140W se alcanzó una eficiencia máxima de 96,8 %, a diferencia de cuando se usa la carga de 250W, que se alcanza una eficiencia máxima de 81,2 %. Esto se debe a que las corrientes que se deben conmutar son mayores, por lo que los tiempos de subida y bajada del mosfet aumentan, produciendo así, disminución en la eficiencia de conversión por pérdidas en forma de calor. Es posible disminuir los tiempos de subida y bajada del mosfet, sin embargo, esto incrementa el ruido a la salida del conversor e incluso inyecta ruido hasta el microcontrolador, interfiriendo en su alimentación, sus comunicaciones y sus conversores analógico-digitales, dejándolo completamente inoperante. Por lo dicho anteriormente, se decidió trabajar con los valores de eficiencia de la **figura 26**.

Las bajas eficiencias obtenidas con ciclos de trabajo pequeños (ciclos de trabajo menores a 40 %), no afectan el desempeño final del sistema, puesto que los ciclos de trabajo en los que se operará se encuentran entre el 50 % y el 100 %.

#### Pruebas de ruido:

Para analizar la magnitud del ruido, se compararán las señales de voltaje obtenidas cuando se utiliza el ciclo de trabajo mínimo, el máximo, y aproximadamente el 50 % del control PWM (del que se habla mas adelante en la **sección 3.5**) con cada carga. En cada prueba se tomó una imagen del voltaje de salida del conversor con un osciloscopio digital; los cursores miden el ancho de la señal para determinar el ruido.

Con la carga de 140W se obtuvieron las siguientes imágenes:

- $DutyCycle = 15\%$ ,  $V_{in} = 32V$ ,  $V_{outmean} = 4V$ ,  $Ruido \approx 2,1V_{pp}$

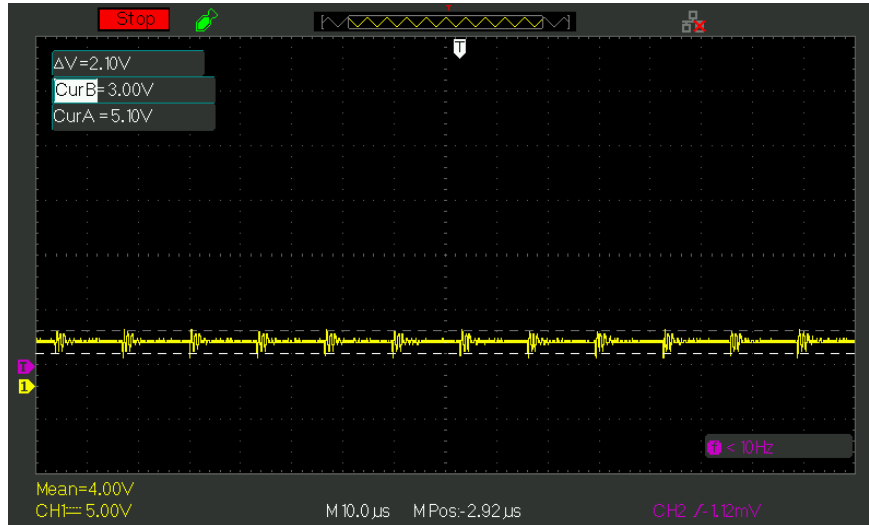


Figura 27:  $V_{out}$  con carga de  $140W$  y ciclo de trabajo de  $15\%$

- $DutyCycle = 47,3\%$ ,  $V_{in} = 32V$ ,  $V_{outmean} = 13,6V$ ,  $Ruido \approx 2V_{pp}$

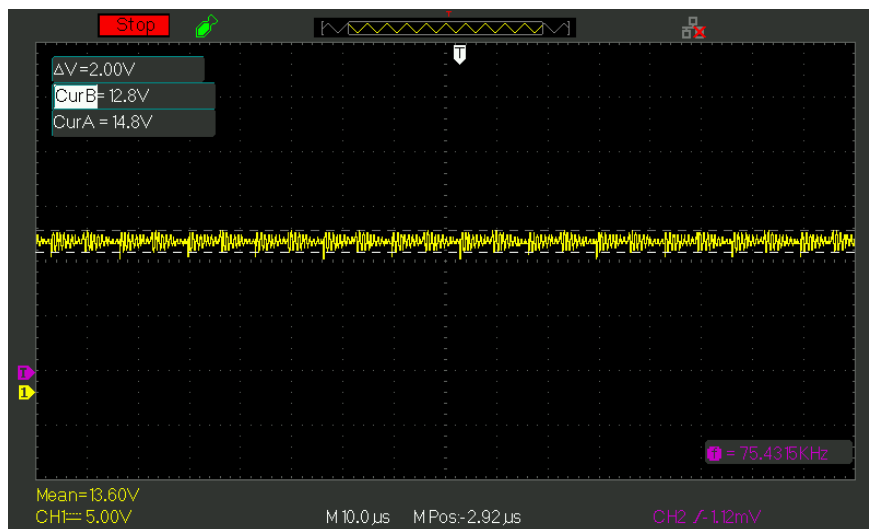


Figura 28:  $V_{out}$  con carga de  $140W$  y ciclo de trabajo de  $47,3\%$

- $DutyCycle = 96\%$ ,  $V_{in} = 25,7V$ ,  $V_{outmean} = 24V$ ,  $Ruido \approx 3,1V_{pp}$

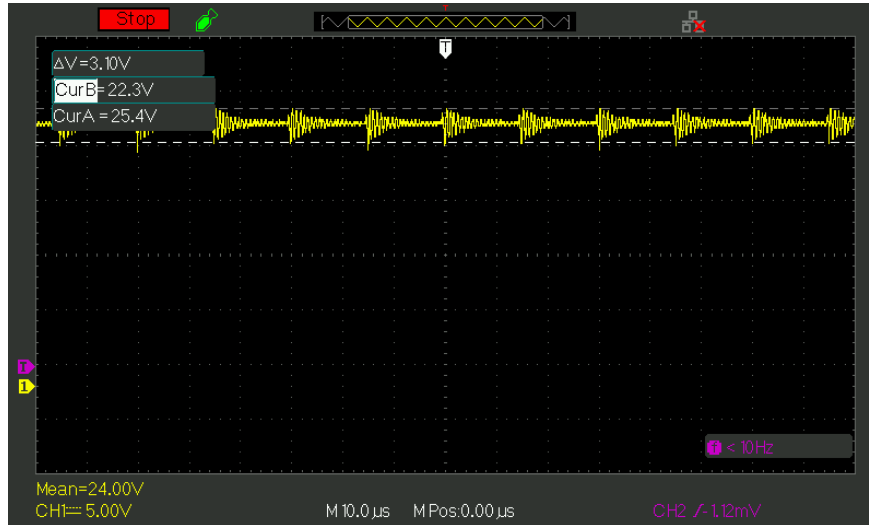


Figura 29:  $V_{out}$  con carga de  $140W$  y ciclo de trabajo de  $96\%$

Con la carga de  $250W$  se obtuvieron las siguientes imágenes:

- $DutyCycle = 15\%$ ,  $V_{in} = 29,96V$ ,  $V_{outmean} = 3,6V$ ,  $Ruido \approx 2,4V_{pp}$

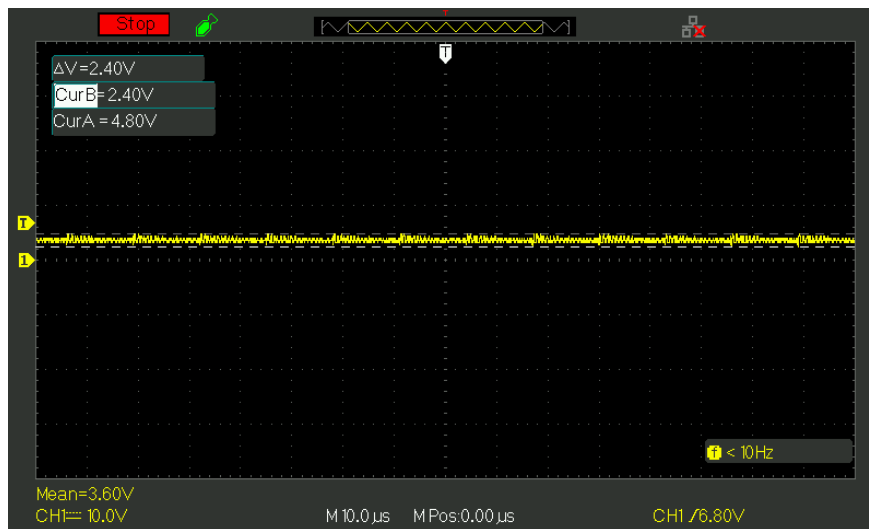


Figura 30:  $V_{out}$  con carga de  $250W$  y ciclo de trabajo de  $15\%$

- $DutyCycle = 47,3\%$ ,  $V_{in} = 29,78V$ ,  $V_{outmean} = 12,4V$ ,  $Ruido \approx 3,0V_{pp}$

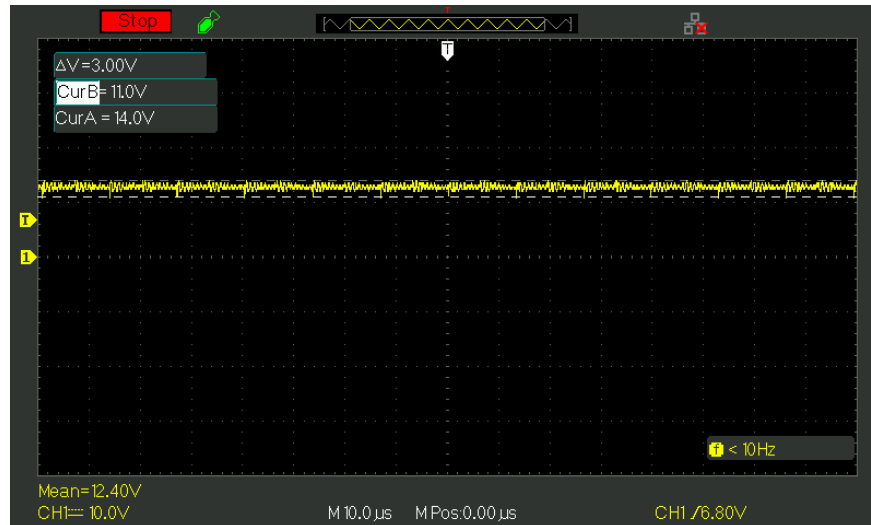


Figura 31:  $V_{out}$  con carga de  $250W$  y ciclo de trabajo de  $47,3\%$

- $DutyCycle = 96\%$ ,  $V_{in} = 25,1V$ ,  $V_{outmean} = 22,4V$ ,  $Ruido \approx 2,8V_{pp}$

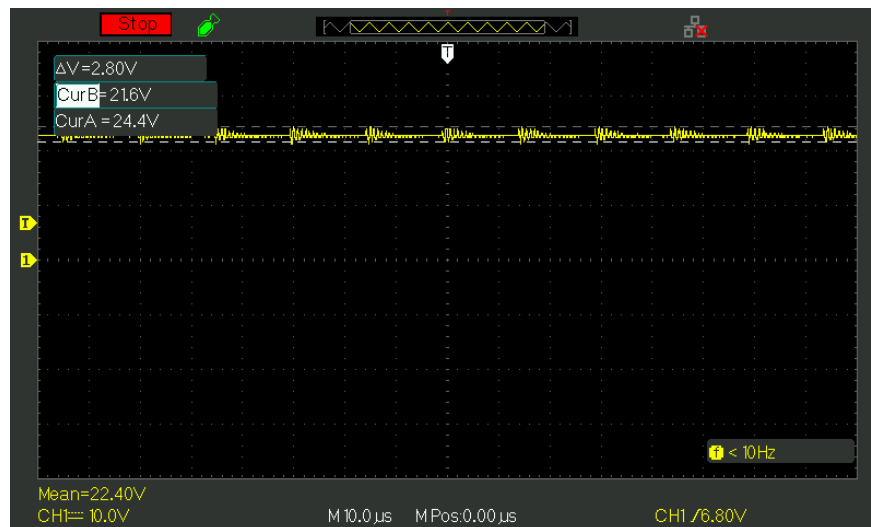


Figura 32:  $V_{out}$  con carga de  $250W$  y ciclo de trabajo de  $96\%$

Como puede observarse en las figuras, la amplitud del ruido se mantiene entre  $2V_{pp}$  y  $3,1V_{pp}$  aproximadamente. Además, analizando las eficiencias en **figuras 25** y **26**; y comparando con los ruidos obtenidos con algunos ciclos de trabajo, se observa la relación que hay entre eficiencia de conversión y ruido de salida, debido a que al tener mayor eficiencia el ruido se incrementa, y viceversa.

Como se mencionó en la **sección 3.4.1**, este ruido de salida se debe a la inductancia serie equivalente del capacitor de entrada del convertor, al crear una respuesta de circuito tanque con los elementos parasitarios del circuito al efectuarse las transiciones de corriente, lo cual explica la forma del ruido y su frecuencia mucho mayor a la frecuencia de conmutación. En la siguiente figura se observan las diferencias entre estas frecuencias y la forma de onda del ruido:

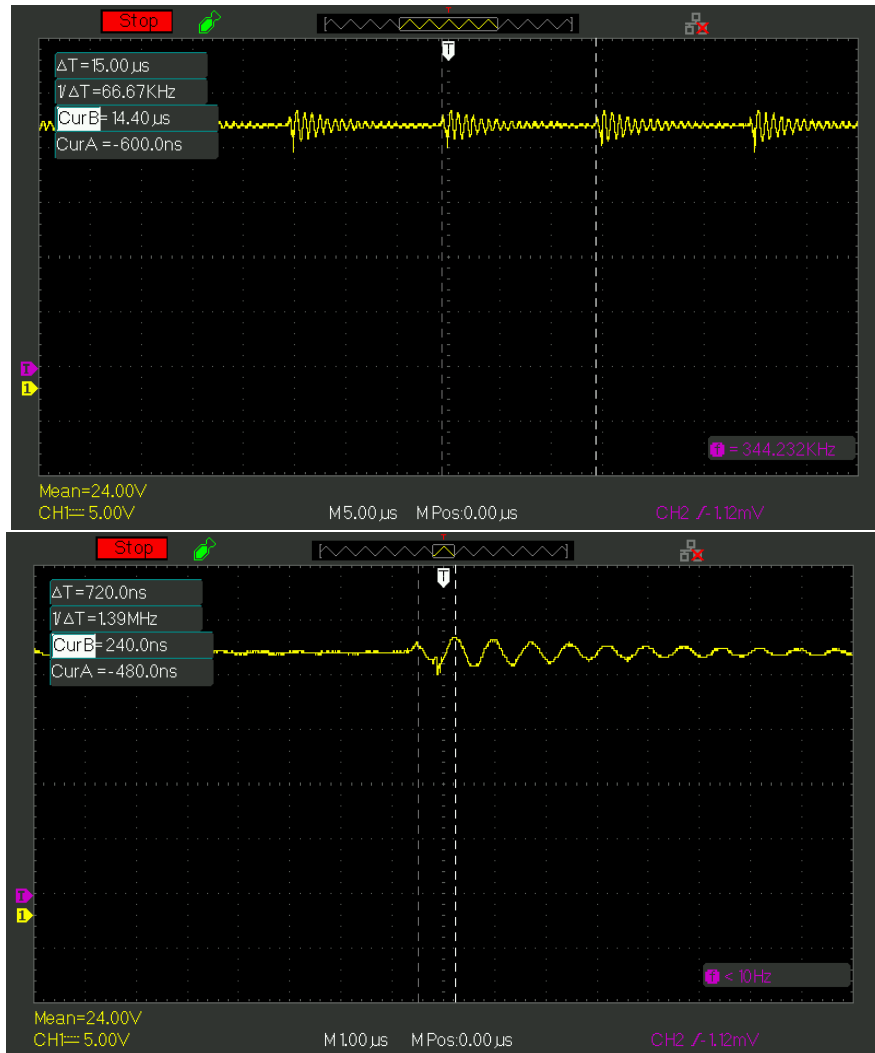


Figura 33: Visualización de la frecuencia de conmutación y la frecuencia del ruido.

La frecuencia de conmutación en este caso es  $66,67kHz$ , mientras que el ruido tiene una frecuencia de aproximadamente  $1,39MHz$ . Además la forma del ruido es igual a la respuesta oscilatoria amortiguada de un circuito tanque:

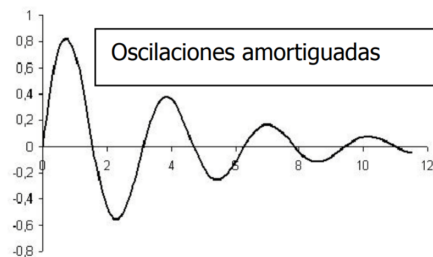


Figura 34: Respuesta de circuito tanque con oscilaciones amortiguadas [23].

### 3.4.4. Máximos absolutos de operación del convertor

El convertor fue diseñado para trabajar con la carga mencionada y con el panel solar de 255W. Sin embargo, el convertor puede trabajar con valores más amplios que los definidos para esta aplicación.

A continuación se muestran los valores de operación máximos del convertor:

Cuadro 5: Máximos absolutos de operación del convertor implementado

Parámetro	Valor	Limitante
$V_{in}$	50V	Voltaje máximo del capacitor de entrada
$V_{out}$	50V	Voltaje máximo del capacitor de salida
$I$ pulsada	300A	Corriente máxima de pulso no repetitivo del diodo
$I$ continua	20A	Corriente con la que la PCB alcanzará 125°C
$T$	125°C	Temperatura máxima del driver de disparo y el material de PCB

## 3.5. Diseño e implementación del control PWM

### 3.5.1. Diseño del control PWM

Para el control PWM se hizo uso del circuito oscilador 555, puesto que este proporciona señales cuadradas de salida de muy alta frecuencia; y con la configuración que se mostrará a continuación, se puede obtener una señal PWM controlada por un potenciómetro. Como se mencionó anteriormente en la **sección 2.6.1**, se requiere una señal PWM de alta frecuencia, debido a que así se disminuye el valor necesario de capacitancia e inductancia para el convertor, facilitando principalmente el proceso de fabricación del inductor L.

El esquema de configuración del circuito 555 para control de PWM se observa en la siguiente figura:

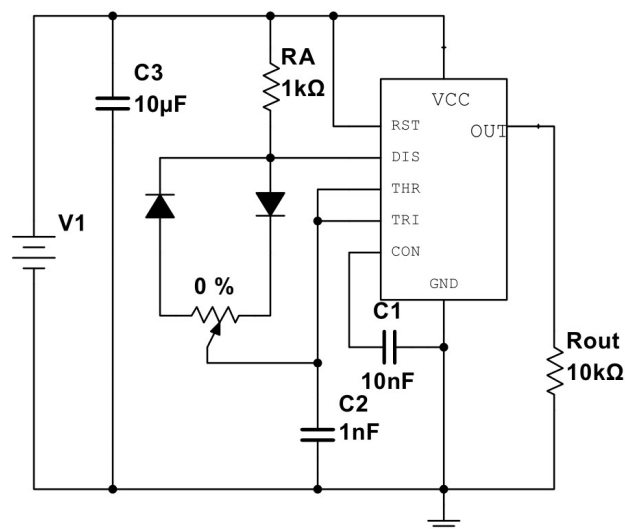


Figura 35: Configuración de 555 para control PWM.

El circuito se utiliza en configuración astable, por lo que su salida será una señal cuadrada cuya frecuencia depende de la suma del tiempo de carga y descarga del capacitor C2 de la siguiente manera: El tiempo de salida en alto es igual al tiempo que el capacitor C2 tarda en cargarse iniciando con un voltaje de  $\frac{1}{3}V_{cc}$  y llegando a  $\frac{2}{3}V_{cc}$ ; el tiempo de salida en bajo es igual al tiempo que este capacitor tarda en descargarse iniciando con un voltaje de  $\frac{2}{3}V_{cc}$  y disminuyendo hasta  $\frac{1}{3}V_{cc}$ .

Como se observa en la **figura 35**, se utilizan dos diodos de recuperación rápida y un potenciómetro para controlar la carga y descarga de dicho capacitor, de esta manera, si se aumenta el tiempo de carga en cierta cantidad al variar el potenciómetro, el tiempo de descarga va a disminuir en esa misma proporción y viceversa, controlando así, el ancho del pulso de la señal cuadrada de salida y manteniendo una frecuencia constante. En la siguiente figura se señala por donde fluyen las corrientes de carga y descarga del capacitor:

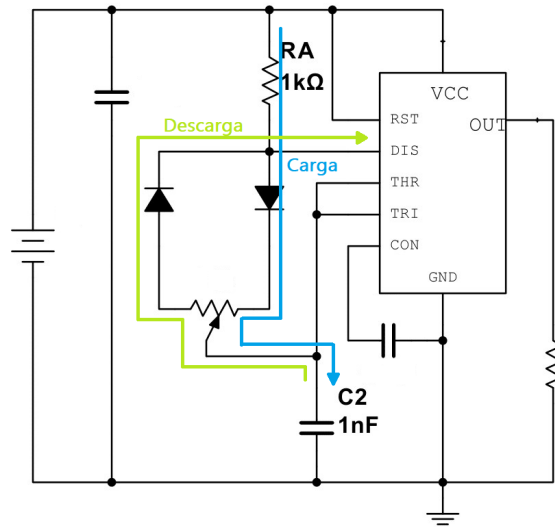


Figura 36: Corrientes de carga y descarga del capacitor C2 en control PWM.

Las ecuaciones de tiempo de encendido, tiempo de apagado y frecuencia de la señal a la salida del circuito 555 para esta configuración, son las siguientes [24]:

$$t_{on} = \left[ \ln \left| 1 - \frac{\frac{1}{3}V_1}{V_1 - V_D} \right| - \ln \left| 1 - \frac{\frac{2}{3}V_1}{V_1 - V_D} \right| \right] [C_2 (R_A + X R_{Pot})] \quad (70)$$

$$t_{off} = \left[ \ln \left| \frac{\frac{2}{3}V_1 - V_D}{V_1 - V_D} \right| - \ln \left| \frac{\frac{1}{3}V_1 - V_D}{V_1 - V_D} \right| \right] [C_2 R_{Pot} (1 - X)] \quad (71)$$

$$T = t_{on} + t_{off} \quad (72)$$

$$f = \frac{1}{T}$$

Donde:

- $V_1$  es el voltaje de alimentación.

- $V_D$  es el voltaje de caída en los diodos, que se asume el mismo para ambos.
- $R_{Pot}$  es la resistencia total del potenciómetro.
- $X$  es el porcentaje de variación del potenciómetro.

Para continuar con el diseño, se realizó el cómputo de estas ecuaciones para encontrar valores comerciales de los componentes del circuito y una frecuencia de conmutación adecuada para el convertor. Los valores calculados se muestran en el siguiente cuadro:

Cuadro 6: Valores de diseño de control PWM con 555

Parám./Comp.	Valor
$R_A$	$1k\Omega$
$R_{Pot}$	$10k\Omega$
$C_2$	$1nF$
$V_D$	$1V$
$V_1$	$5V$
Frecuencia	$72,6kHz$

Para poder seleccionar el ancho del pulso desde el microcontrolador, se utilizó un potenciómetro digital. Se seleccionó el potenciómetro de referencia MCP4141-103, el cual es un potenciómetro de  $10k\Omega$  compuesto de una red de resistencias de 7 bits que se comunica por protocolo serial SPI.

El esquemático completo incluyendo el potenciómetro digital se muestra en la siguiente figura:

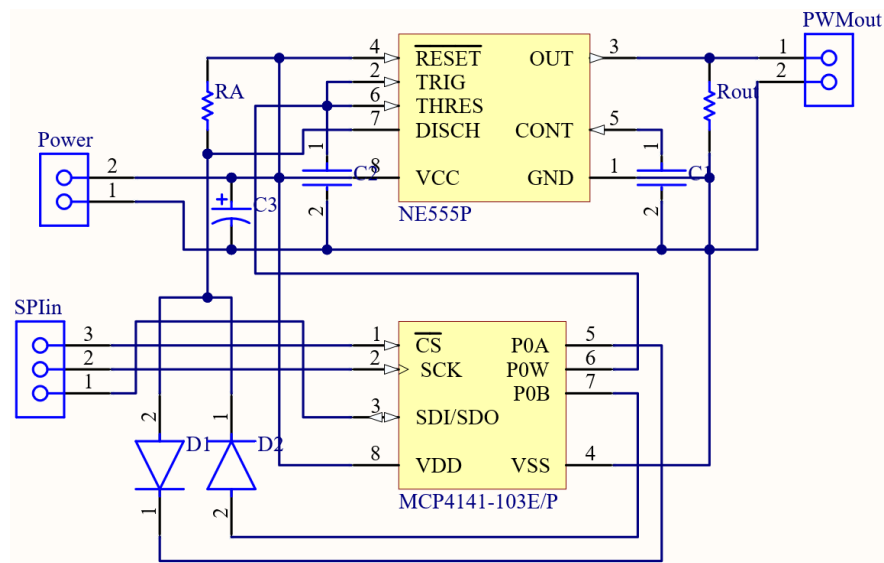


Figura 37: Esquemático completo de control PWM

### 3.5.2. Implementación del control PWM

Luego de tener el diseño completo, se implementó el circuito en PCB (ver diseño de PCB en **anexo 6.4**). El resultado se muestra a continuación:

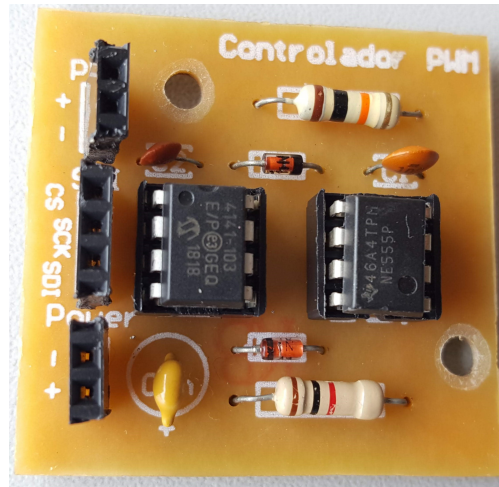


Figura 38: Control PWM implementado.

Con este montaje, el control PWM se comunicará con el microcontrolador y el driver del mosfet del convertor por medio de *Jumpers*.

Cómo se explicará mas adelante, el microcontrolador utilizado en el sistema es Arduino, por lo que el lenguaje de programación usado para definir las interacciones del microcontrolador con el circuito de control PWM es C++. El código utilizado para definir el ciclo de trabajo del control PWM se muestra a continuación:

```
1 #include <SPI.h>
2 byte address = 0x00;
3 int CS = 10;
4 float Duty = 0.5; //Ciclo util , entre 0 y 1.
5 void setup() {
6     pinMode (CS, OUTPUT);
7     SPI.begin();
8 }
9 void loop() {
10    digitalPotWrite(round(128 * (1 - Duty)));
11    delay(2000);
12 }
13 int digitalPotWrite(int value) {
14     digitalWrite(CS, LOW);
15     SPI.transfer(address);
16     SPI.transfer(value);
17     digitalWrite(CS, HIGH);
18 }
```

Código 1: Código de definición de ciclo de trabajo.

Éste código envía al potenciómetro digital un valor entero entre 0 y 128 (por medio de la función `digitalPotWrite(int value)`) para definir el valor de resistencia entre sus terminales.

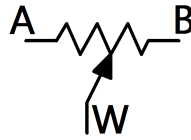


Figura 39: Terminales del potenciómetro digital.

Siendo estos los terminales, el valor entero a la entrada de la función `digitalPotWrite` define la resistencia entre terminales de la siguiente manera (aproximadamente):

$$R_{AW} = 10k\Omega * \left(1 - \frac{int\_value}{128}\right) \quad (73)$$

$$R_{BW} = 10k\Omega - R_{AW} \quad (74)$$

El ciclo de trabajo es un valor entre 0 y 1, por lo que para convertirlo en un valor adecuado para enviar al potenciómetro, se multiplica el valor máximo que éste puede recibir (128) con el valor de ciclo de trabajo (en el código se denota como  $1 - Duty$  en la multiplicación, debido a que por la manera en que se conectaron los terminales del potenciómetro digital, entre más grande sea el valor enviado al potenciómetro, menor es el ciclo de trabajo). El valor finalmente es redondeado, ya que solo se envían al potenciómetro valores enteros.

### 3.5.3. Pruebas del control PWM

Se realizaron pruebas de laboratorio con osciloscopio para capturar las distintas señales PWM para cada valor de *Duty* en el código y encontrar sus diferencias con el ciclo de trabajo real obtenido a la salida del circuito. En la siguiente figura se muestran dichas diferencias:

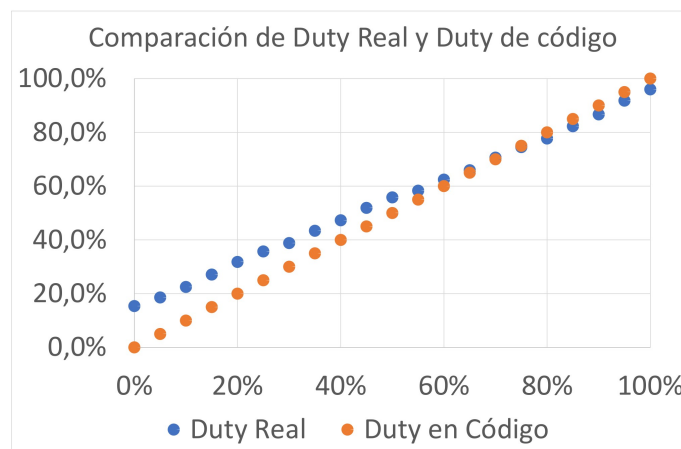


Figura 40: Comparación del ciclo de trabajo real y el del código.

Como se puede observar, la diferencia mas grande se encuentra al definir los ciclos de trabajo mas pequeños. Esto es debido a que la resistencia  $R_A$  actúa como limitador inferior

del ciclo de trabajo, puesto que en el proceso de carga del capacitor, la resistencia  $R_A$  está en serie a la resistencia  $R_{AW}$  del potenciómetro. A partir de 50% aproximadamente, las diferencias son mínimas.

Para la aplicación que se le dará a éste control de PWM, estas diferencias no tienen consecuencias negativas, pues esto no afectará el número de iteraciones que debe hacer el algoritmo para converger al punto de máxima potencia.

Algunas imágenes de la señal PWM capturadas con el osciloscopio se muestran a continuación:

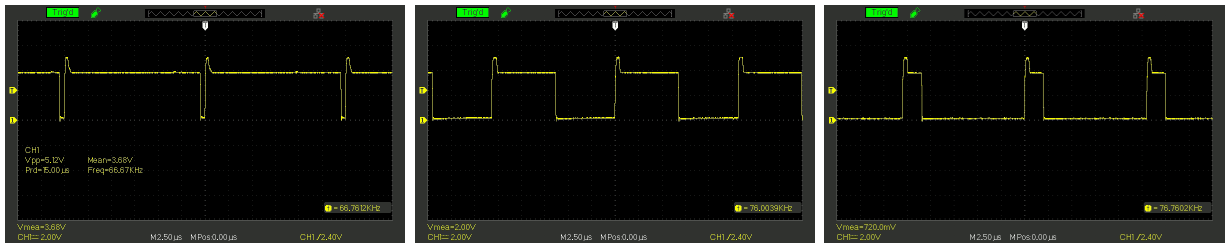


Figura 41: Ciclo de trabajo máximo (96%), medio (50%) y mínimo (15%) respectivamente

Observando estas señales de salida también se nota un cambio de frecuencia de la señal al variar el ancho del pulso. La frecuencia varía desde  $66,8\text{kHz}$  cuando el ciclo de trabajo es el máximo, hasta  $76,8\text{kHz}$  cuando el ciclo de trabajo es el mínimo. Esto se debe a que la frecuencia de conmutación depende del voltaje *forward* de los diodos, como se muestra en las ecuaciones 70, 71 y 72; y al variar  $R_{AW}$  y  $R_{BW}$ , varían las corrientes que fluyen por los diodos, afectando su voltaje *forward*. En la implementación se usaron diodos de referencia 1N4148, la curva que relaciona su corriente con el voltaje *forward* se muestra a continuación:

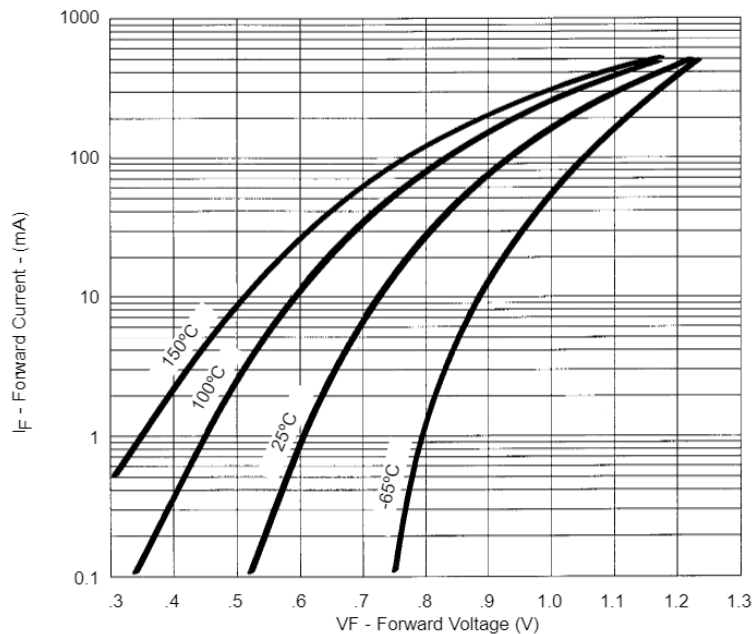


Figura 42: Curva típica de corriente forward vs. voltaje forward del diodo 1N4148.

[25]

Esta curva varía para diodos de distintos fabricantes, por lo que se puede analizar el comportamiento de estas curvas siempre y cuando se tenga en cuenta que los valores en los ejes no describen exactamente los diodos usados en la práctica, puesto que estos son de un fabricante desconocido.

Según las variaciones de frecuencia obtenidas en la salida PWM, se calculó que el voltaje *forward* de los diodos varía entre  $0,9V$  y  $1,1V$  aproximadamente.

Una observación importante es que el conversor se diseñó para una frecuencia de  $72,6kHz$  y la frecuencia mínima del control PWM es aproximadamente  $66,8kHz$ . Sin embargo, usando esta frecuencia en las **ecuaciones 67** y **68** se obtienen valores de inductancia y capacitancia de  $120\mu H$  y  $5,2\mu F$ , mientras que los usados fueron  $127\mu H$  y  $470\mu F$ , así que se siguen cumpliendo los parámetros de diseño del conversor.

### 3.6. Microcontrolador

El sistema requiere un microcontrolador con suficientes entradas analógicas para el manejo de los sensores de voltaje y corriente; y cuenta con comunicación por protocolo SPI para su interacción con el potenciómetro digital usado en el control PWM. Además es necesario que el microcontrolador sea poco voluminoso y consuma poca energía. Es por esto que se seleccionó el Arduino Nano, que es un microcontrolador que cumple con estos requerimientos y brinda muchas facilidades para su programación y conexión con elementos periféricos.

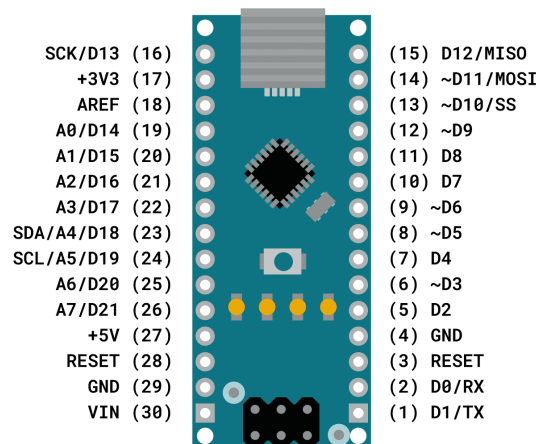


Figura 43: Diagrama de pines del Arduino Nano. [26]

Este microcontrolador tiene un total de 22 pines I/O digitales y 8 pines de entradas analógicas, puede comunicarse por bus  $I^2C$ , bus  $SPI$  o comunicación serial USB, se alimenta con  $5VDC$  y consume  $19mA$  (y se suma un máximo de  $40mA$  por cada I/O usada). Algunos pines digitales tienen función de PWM, pero en este caso no se usarán, ya que sus frecuencias máximas son muy bajas para los requerimientos de este proyecto. [26]

Para facilidad en las conexiones del sistema, se utilizó un Shield de Arduino. Su montaje se muestra a continuación:

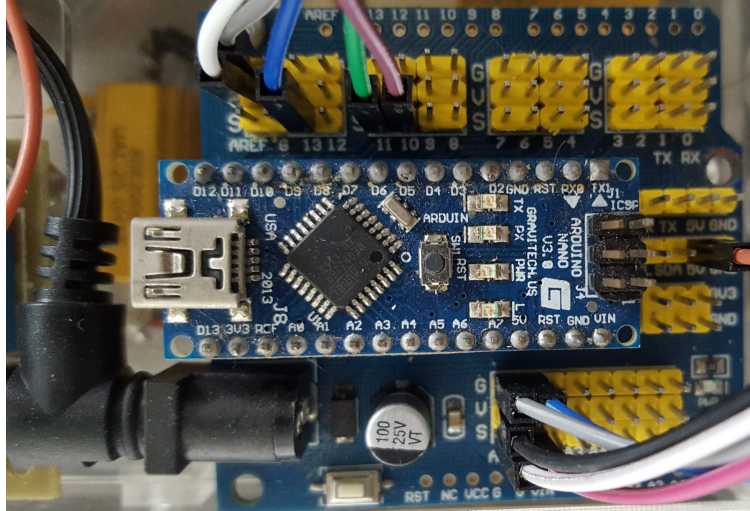


Figura 44: Montaje de Arduino Nano en Shield I/O.

### 3.7. Sensores

Para la implementación del algoritmo MPPT se requiere el monitoreo de las variables de voltaje y corriente suministradas por el panel solar, de manera se puedan tomar decisiones respecto a estas mediciones y moverse por la curva de potencia del panel hasta encontrar el punto de máxima potencia.

#### 3.7.1. Sensor de voltaje

Se trabajó con el sensor FZ0430. Este sencillo sensor solo consta de un divisor de tensión, el cual adecua el voltaje medido al rango aceptable por la entrada analógica del Arduino, cuya entrada máxima es  $5V$ . Comercialmente se consigue dicho sensor como un pequeño circuito con dos resistencias SMD, de  $7,5k\Omega$  y de  $30k\Omega$ , lo que permite medir hasta  $25V$  al Arduino Nano, por lo cual fue necesario modificar este sensor para poder medir hasta  $38,7V$ , que es el voltaje en circuito abierto del panel solar. La modificación necesaria fue reemplazar la resistencia de  $30k\Omega$ ; para esto se escogió una de  $62k\Omega$ , de esta manera es posible medir un voltaje hasta de  $46V$ , lo que es un voltaje mayor al máximo que puede medirse en el panel; sin embargo, proporciona un margen de error para proteger la entrada analógica del Arduino.

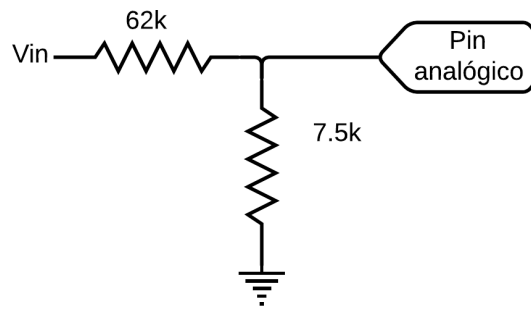


Figura 45: Sensor de voltaje modificado.

A continuación se detalla el software implementado para la utilización de este sensor:

```

1 float VoltageFunction(int n_muestrasV) { //Definicion de funcio n
2   float voltaje = 0;
3   float voltajeEntrada;
4   for (int i = 0; i < n_muestrasV; i++) {
5     voltajeEntrada=analogRead(A1)*(5.0/1023.0); //lectura de voltaje...
        en pin A1
6     voltaje = voltaje + voltajeEntrada * 24.67; //Se multiplica el ...
        voltaje de entrada por el factor de calibracion y se acumula el ...
        valor
7   }
8   voltaje = voltaje / n_muestrasV; //Se calcula el voltaje promedio ...
        de las muestras
9   return (voltaje);
10 }

```

Código 2: Función básica para medición de voltaje.

Con esta función se puede obtener el voltaje promedio del número de muestras especificadas como entrada de la función. Esto se hace tomando el voltaje a la entrada analógica del conversor y multiplicando por un factor que devolverá el voltaje antes del divisor de tensión. No obstante, para obtener medidas de voltaje precisas fue necesario crear varios casos de calibración que dependen del valor de voltaje de entrada y el ciclo de trabajo definido para el conversor, ya que el ruido generado afecta la medición del voltaje.

```

1 float VoltageFunction(int n_muestrasV) { //Definicion de funcion
2   float voltaje = 0;
3   float voltajeIn = 0;
4   float voltajeEntrada;
5   for (int i = 0; i < n_muestrasV; i++) {
6     voltajeEntrada = (analogRead(A1) * (5.0 / 1023.0));
7     if (Duty >= 0.8) { //Calibracion para ciclo de trabajo mayor a ...
        0.8

```

```

8     if (voltajeEntrada < 0.174) { //hasta ...
2V
9         voltaje = voltaje + voltajeEntrada * 24.67;}
10    else if (voltajeEntrada>=0.174&&voltajeEntrada<0.368){//hasta ...
4V
11        voltaje = voltaje + voltajeEntrada * 13.03;}
12    else if (voltajeEntrada>=0.368&&voltajeEntrada<0.564){//hasta ...
6V
13        voltaje = voltaje + voltajeEntrada * 11.82;}
14    else if (voltajeEntrada>=0.564&&voltajeEntrada<0.764){//hasta ...
8V
15        voltaje = voltaje + voltajeEntrada * 11.33;}
16    else if (voltajeEntrada>=0.764&&voltajeEntrada<0.958){//hasta ...
10V
17        voltaje = voltaje + voltajeEntrada * 11.02;}
18    else if (voltajeEntrada>=0.958&&voltajeEntrada<1.255){//hasta ...
13V
19        voltaje = voltaje + voltajeEntrada * 10.88;}
20    else if (voltajeEntrada>=1.255&&voltajeEntrada<1.547){//hasta ...
16V
21        voltaje = voltaje + voltajeEntrada * 10.71;}
22    else if (voltajeEntrada>=1.547&&voltajeEntrada<1.843){//hasta ...
19V
23        voltaje = voltaje + voltajeEntrada * 10.62;}
24    else if (voltajeEntrada>=1.843&&voltajeEntrada<2.140){//hasta ...
22V
25        voltaje = voltaje + voltajeEntrada * 10.56;}
26    else if (voltajeEntrada>=2.140&&voltajeEntrada<2.437){//hasta ...
25V
27        voltaje = voltaje + voltajeEntrada * 10.48;}
28    else if (voltajeEntrada>=2.437&&voltajeEntrada<2.734){//hasta ...
28V
29        voltaje = voltaje + voltajeEntrada * 10.45;}
30    else if (voltajeEntrada>=2.734) { //mas ...
de 29V
31        voltaje = voltaje + voltajeEntrada * 10.40;}}
32    else if (Duty >= 0.6 && Duty <0.8) {

```

Código 3: Calibración por rangos de voltaje y ciclo de trabajo del conversor.

En el anterior fragmento de código se observa el esquema de calibración utilizado. Para cada rango de ciclo de trabajo se usó un conjunto de calibraciones distribuidas por rangos de voltaje obtenido en la entrada analógica del Arduino. Se definieron cinco rangos de ciclo de trabajo en total, para los cuales se tomó el valor medio de cada rango para crear las tablas de calibración:

- $Duty < 0,2$ . Calibración con  $Duty = 0,1$ .
- $0,2 \leq Duty < 0,4$ . Calibración con  $Duty = 0,3$ .
- $0,4 \leq Duty < 0,6$ . Calibración con  $Duty = 0,5$ .
- $0,6 \leq Duty < 0,8$ . Calibración con  $Duty = 0,7$ .

- $Duty \geq 0,8$ . Calibración con  $Duty = 0,9$ .

Posteriormente, para cada rango de ciclo de trabajo se realizó la calibración por rangos de voltaje con ayuda de un multímetro digital. El valor que se ajusta para calibrar la medición, es el factor por el que se multiplica la entrada de voltaje del arduino para obtener el voltaje medido; mientras que los valores que se ajustan para definir los límites de los rangos de voltaje de entrada se obtuvieron como el valor medio de voltaje de entrada al medir ciertos voltajes de valor entero, de la siguiente forma:

Cuadro 7: Fragmento de tabla para calibración de sensor por rangos de voltaje.

V real	V sensor	Factor
6	0,434	13,82
7	0,537	13,04
8	0,640	12,50
9	0,746	12,06

**V real** es el voltaje medido con el multímetro, **V sensor** es el voltaje que entrega el sensor a la entrada analógica del Arduino, **Factor** es el valor por el cual se debe multiplicar el voltaje **V sensor** para obtener **V real**. El primer **límite de rango** se calculó como el valor promedio entre los valores de **V sensor** obtenidos para **V real** igual a 6 y 7, esto es,  $\frac{0,434+0,537}{2} = 0,486$ ; de igual manera, para el segundo **límite de rango** se usaron los valores de **V sensor** obtenidos para **V real** igual a 8 y 9, esto es,  $\frac{0,640+0,746}{2} = 0,693$ . De este modo, se tiene un rango de voltaje entre 0.486 y 0.693, para el cual el factor de calibración será el promedio entre los valores de **Factor** obtenidos para **V real** igual a 7 y 8, es decir,  $\frac{13,04+12,50}{2} = 12,77$ .

En total se usaron 12 rangos de voltaje en cada tabla de calibración.

### 3.7.2. Sensor de corriente

Se utilizó el sensor de efecto Hall ACS712-20A, el cual puede medir hasta 20A y viene adecuado para su facil conexión a microcontrolador, necesitando únicamente conectar a los pines 5V, GND y a un pin de entrada analógica.

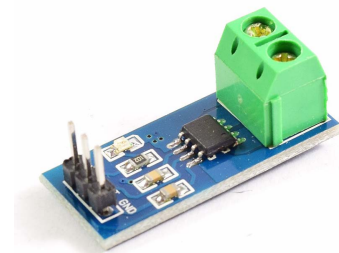


Figura 46: Sensor de corriente ACS712-20A. [27]

Este sensor entrega a su salida un voltaje en función de la corriente medida. Dicha función es la siguiente:

$$V = \text{sensibilidad} * I + \text{constante} \quad (75)$$

Conocer esta función lineal permite calcular el valor de corriente a partir del voltaje obtenido en la entrada analógica del microcontrolador. Además, se aprovecha la linealidad de la ecuación para calibrar la medición, variando los valores de sensibilidad y la constante dados por el fabricante, los cuales son *sensibilidad* = 0,1 y *constante* = 2,5.

En la práctica el valor de voltaje obtenido del sensor no es completamente lineal, por lo que para la calibración fue necesario ajustar el valor de sensibilidad de la **ecuación 75**, por rangos de voltaje, de manera similar a como se hizo con el sensor de voltaje.

Para hallar la constante, se midió una corriente de 0A, por lo que al reemplazar en la ecuación, queda:  $V = \text{sensibilidad} * (0) + \text{constante} = \text{constante}$ . Esta constante es fija para todos los casos de calibración.

Para hallar los valores de sensibilidad, se dividieron los casos por rangos de voltaje de entrada, de manera que el voltaje central de cada rango corresponda a un valor entero de corriente entre 0A y 9A. A cada rango se le asignó la sensibilidad que permita a cada valor central de voltaje obtener el valor entero de corriente ya mencionado. Esta sensibilidad se calculó con la ecuación de la pendiente usando siempre como primer punto el de la corriente de 0A, de la siguiente forma:

$$\text{sensibilidad} = \frac{V_2 - V_1}{I_2 - I_1} = \frac{V - \text{constante}}{I - 0} \quad (76)$$

El código de la función de medición de corriente se muestra a continuación:

```

1 float MedirCorriente(int n_muestras) {
2     float voltaje = 0;
3     float Corriente = 0;
4     for (int i = 0; i < n_muestras; i++) {
5         voltaje = analogRead(A0) * (5.0 / 1023.0);
6         if (voltaje < 2.54) {
7             Corriente = Corriente + (voltaje - 2.488) / 0.0968;} //0A
8         else if (voltaje >= 2.54 && voltaje < 2.64) {
9             Corriente = Corriente + (voltaje - 2.488) / 0.0980;} //1A
10        else if (voltaje >= 2.64 && voltaje < 2.73) {
11            Corriente = Corriente + (voltaje - 2.488) / 0.0975;} //2A
12        else if (voltaje >= 2.73 && voltaje < 2.83) {
13            Corriente = Corriente + (voltaje - 2.488) / 0.0977;} //3A
14        else if (voltaje >= 2.83 && voltaje < 2.93) {
15            Corriente = Corriente + (voltaje - 2.488) / 0.0980;} //4A
16        else if (voltaje >= 2.93 && voltaje < 3.03) {
17            Corriente = Corriente + (voltaje - 2.488) / 0.0982;} //5A
18        else if (voltaje >= 3.03 && voltaje < 3.13) {
19            Corriente = Corriente + (voltaje - 2.488) / 0.0984;} //6A
20        else if (voltaje >= 3.13 && voltaje < 3.23) {
21            Corriente = Corriente + (voltaje - 2.488) / 0.0986;} //7A
22        else if (voltaje >= 3.23 && voltaje < 3.33) {
23            Corriente = Corriente + (voltaje - 2.488) / 0.0988;} //8A
24        else if (voltaje >= 3.33) {

```

```

25     Corriente = Corriente + (voltaje - 2.488) / 0.0990;}}//9A
26 Corriente = Corriente / n_muestras;
27 if (Corriente < 0) {
28     Corriente = 0;}
29 return (Corriente);}

```

Código 4: Función de medición de corriente.

### 3.8. Implementación de algoritmos MPPT

En este proyecto se implementaron dos algoritmos: SH y P&O. Los algoritmos fueron implementados decidiendo el ciclo de trabajo de conmutación (puntos de prueba) a través de la variación del potenciómetro digital. Por cada ajuste de ciclo de trabajo, es decir, por cada iteración; se ejecutaron las funciones de medición de voltaje y corriente con 250 muestras para calcular la potencia. El tiempo de iteración definido tuvo en cuenta el tiempo mínimo necesario para que las variables se estabilicen en el peor caso, es decir, el tiempo que demora la carga en pasar de 0V a 24V. Esta transición se puede observar en la siguiente figura:

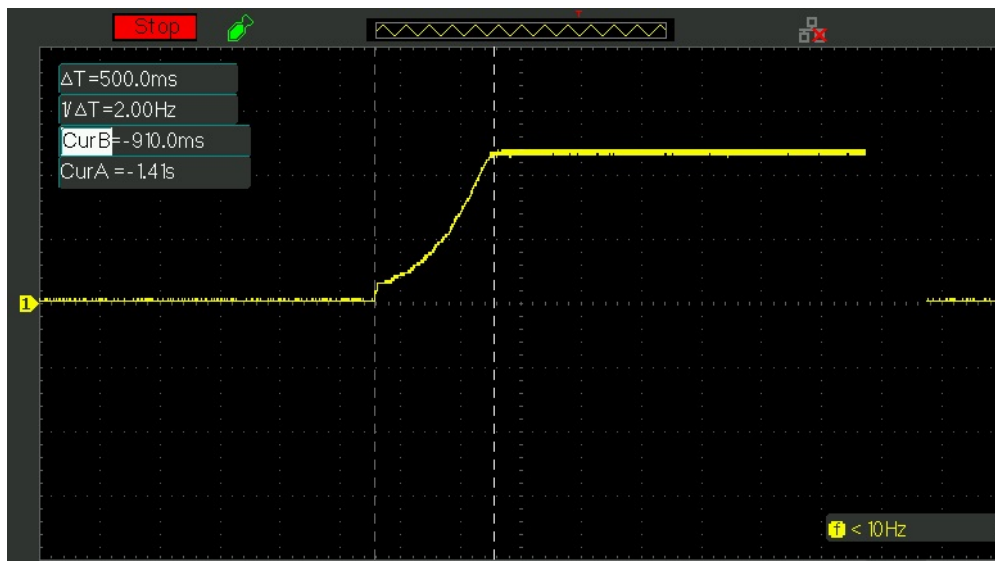


Figura 47: Transición de 0V a 24V en la carga.

La carga tarda  $500\text{ms}$  en estabilizarse en 24V, a este tiempo se sumaron  $300\text{ms}$  como margen de error antes de pasar a la siguiente iteración para poder calcular la potencia alcanzada.

Los diagramas de flujo que describen las operaciones de los algoritmos SH y P&O implementados se muestran a continuación:

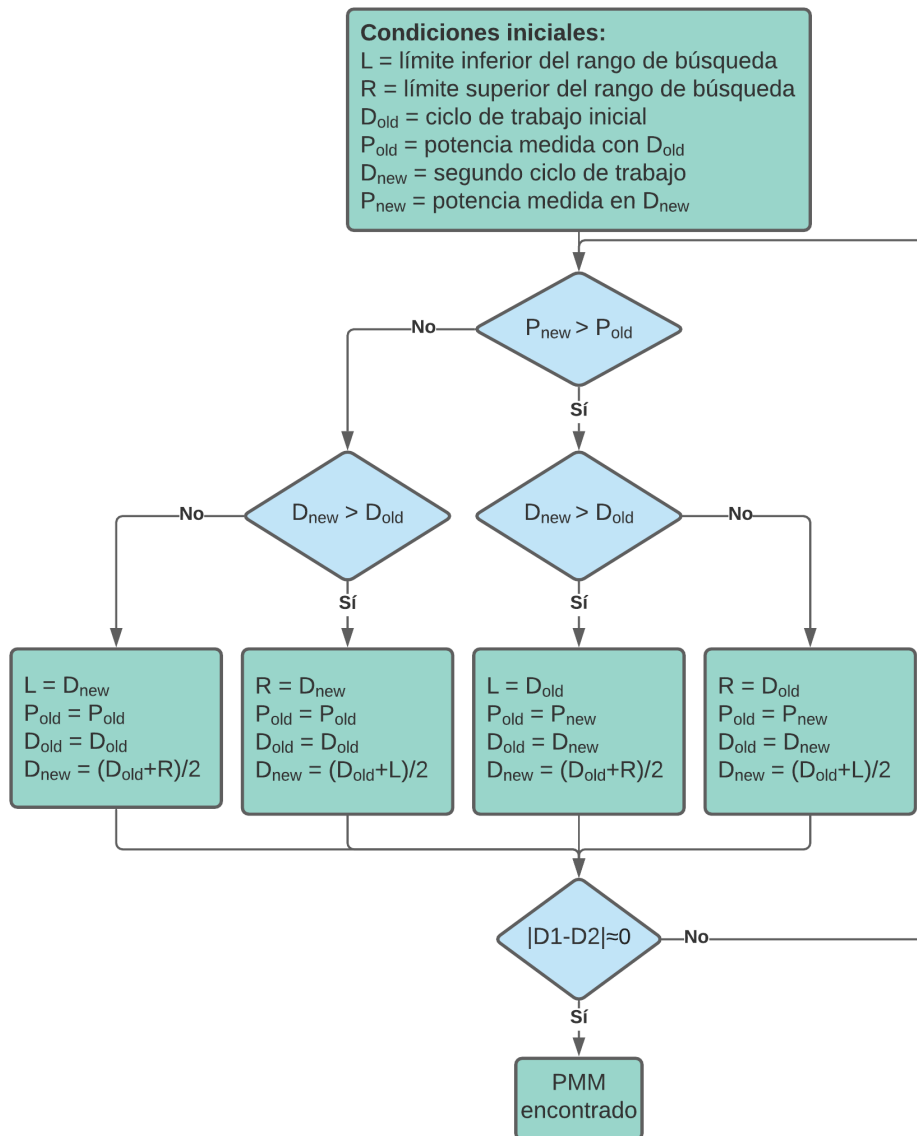


Figura 48: Diagrama de flujo del algoritmo SH implementado para aplicación en convertor DC/DC.

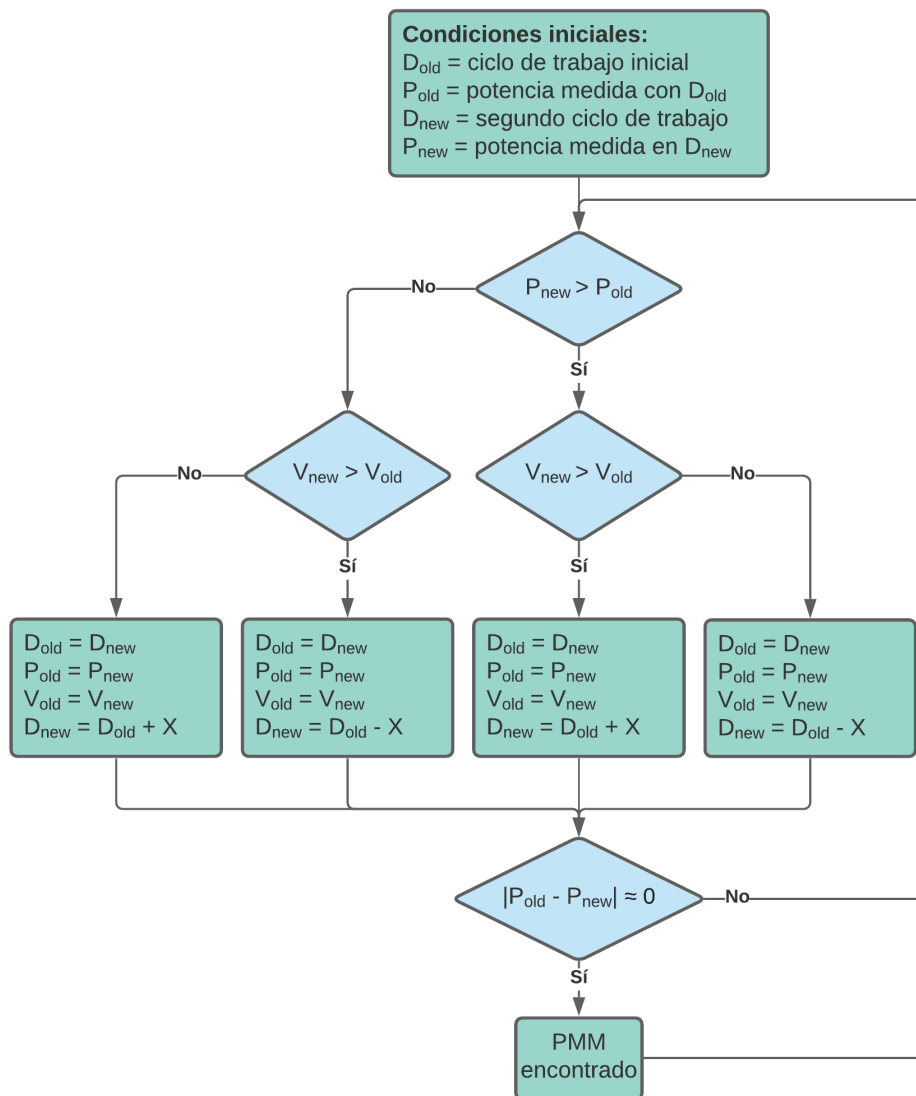


Figura 49: Diagrama de flujo del algoritmo P&O implementado para aplicación en convertor DC/DC.

Los códigos implementados se pueden ver en los **anexos 6.5 y 6.6**.

### 3.9. Fuente de alimentación del sistema

En una aplicación real, lo ideal es que no se necesiten fuentes de alimentación externas y que el panel solar suministre la energía a todos los bloques del sistema. Sin embargo, en este caso se dejará esa adaptación como trabajo futuro.

En el sistema planteado se requiere una fuente de voltaje fija para alimentar el microcontrolador, los sensores de voltaje y corriente, los drivers de los mosfets, el circuito controlador de PWM y el potenciómetro digital, para lo cual se diseñó el siguiente circuito:

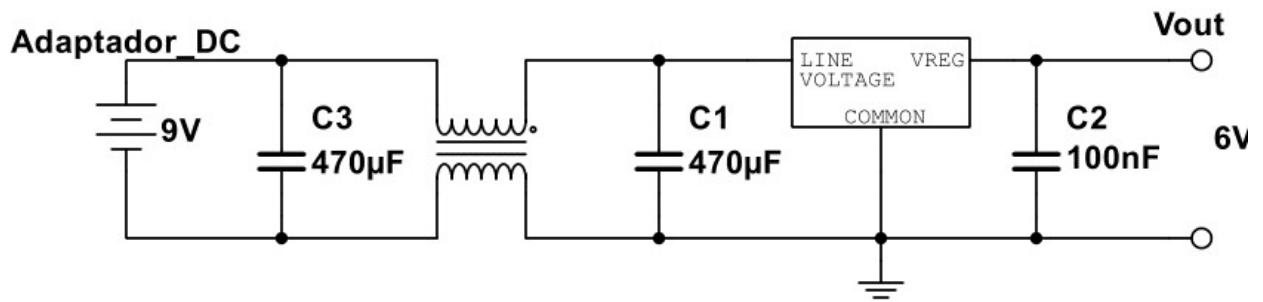


Figura 50: Fuente de alimentación del sistema.

Como se puede observar en la figura, la fuente es conformada por un filtro EMI para evitar picos de corriente y un regulador de voltaje. El diseño de PCB se puede ver en el **anexo 6.7**. El resultado se muestra a continuación:

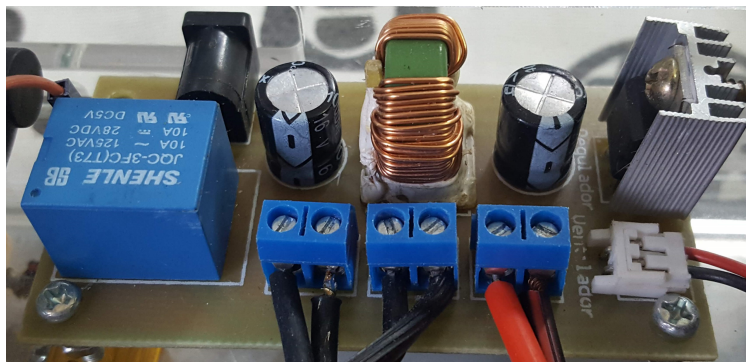


Figura 51: Fuente de alimentación implementada en PCB.

### 3.10. Sistema implementado

En la siguientes figuras se muestra el circuito implementado y el panel solar utilizado:

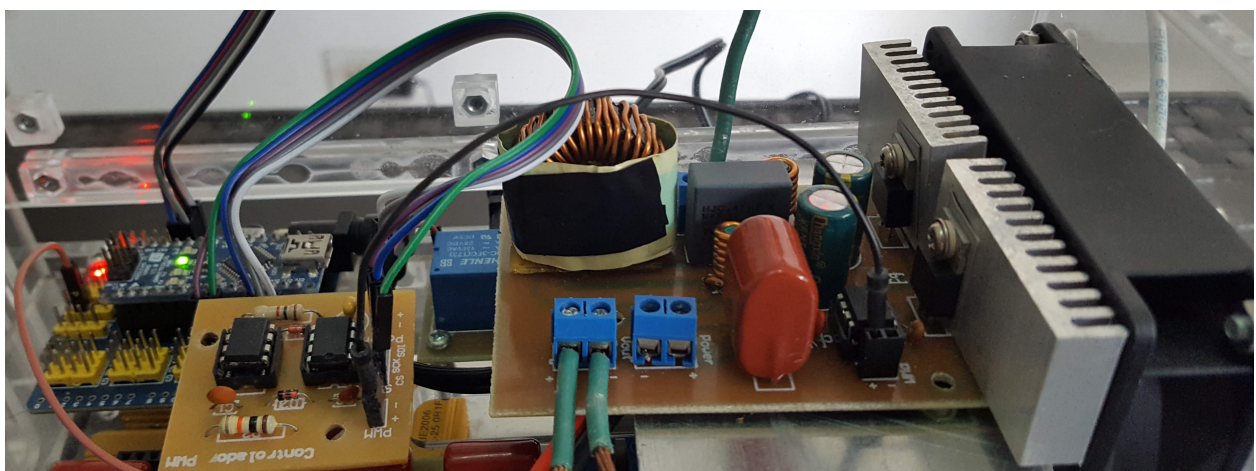


Figura 52: Montaje del sistema



Figura 53: Panel solar utilizado.

# Capítulo 4

## 4. Pruebas y análisis de resultados

El conversor DC-DC implementado se probó con dos algoritmos: el algoritmo P&O, que es el utilizado por la mayoría de reguladores de carga solar MPPT comerciales; y el algoritmo SH. Esto con el fin de encontrar las ventajas y desventajas de la aplicación del algoritmo SH frente a los sistemas tradicionales de rastreo de máxima potencia.

Para entender el comportamiento del voltaje, la corriente y la potencia a la salida del panel solar al ejecutar los rastreos, inicialmente se simuló el sistema implementado en la herramienta Simulink de Matlab.

Posterior a la simulación se realizaron las siguientes pruebas para cada algoritmo con el circuito real:

- **Rastreo simple:** Donde se hará la comparación con lo simulado en Simulink.
- **Rastreos sucesivos:** Donde se verificará que los rastreos siempre convergen al punto de máxima potencia.

Luego se hará una comparación de los resultados de los dos algoritmos respecto a su tiempo de convergencia, número de iteraciones necesarias y la potencia máxima encontrada bajo condiciones climáticas similares.

Finalmente se analizará cuales son los valores de ciclo de trabajo con los que se suele obtener la máxima potencia con el conversor implementado.

Es importante resaltar que para conocer la irradiancia en cada prueba se usó un medidor de potencia solar TM-206 y para encontrar el valor de temperatura se midió el voltaje en circuito abierto entregado por el panel y se ubicó la curva de potencia (de las bases de datos de *Simulink*) acorde a dicho voltaje y a la irradiancia medida.

La conexión entre el panel solar y el conversor se hizo con un cable 14 AWG de 12 metros de largo.

### 4.1. Resultados de simulación

Para la simulación del sistema se utilizó el modelo del panel contenido en las bases de datos de *Simulink*. A este modelo se le definen los parámetros de entrada de irradiancia y temperatura; y a su salida se conectó el modelo del conversor implementado, el cual se hizo utilizando componentes ideales. Además, ya que el simulador no cuenta con un modelo de MOSFET, se reemplazó por un IGBT. Los códigos de Matlab de cada algoritmo se muestran en los **anexos 6.8** y **6.9**.

El esquema de simulación se muestra en la siguiente figura:

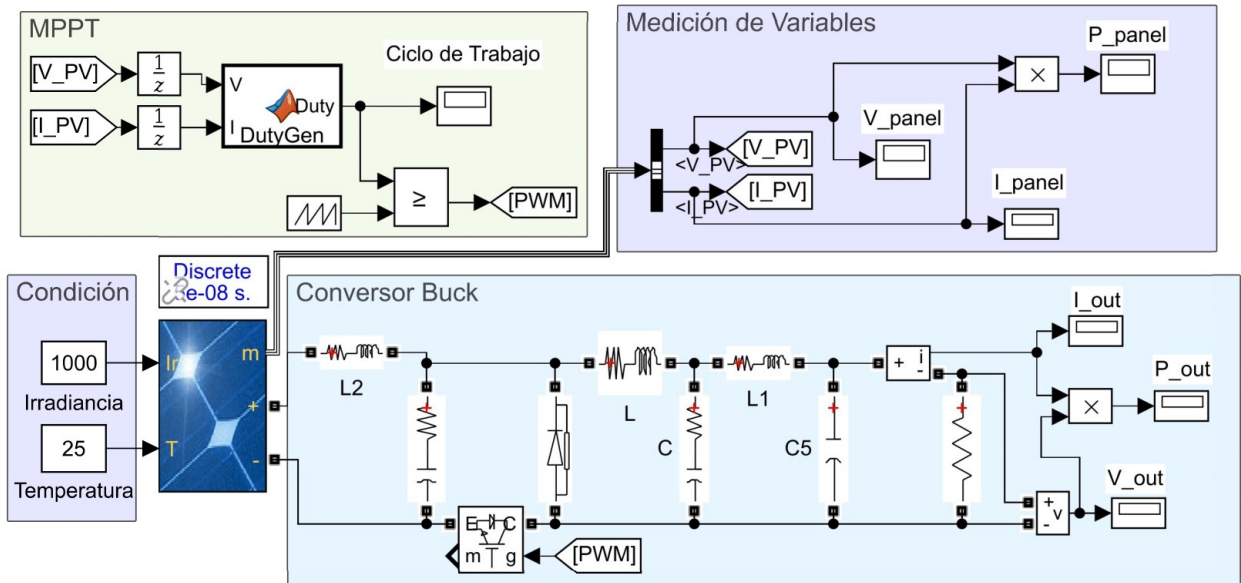


Figura 54: Esquema de simulación del sistema en Simulink.

Con el anterior esquema se realizó la simulación de los algoritmos SH y P&O para condiciones ideales de  $Irradiancia = 1000W/m^2$  y  $Temperatura = 25^{\circ}C$ , los resultados son los siguientes:

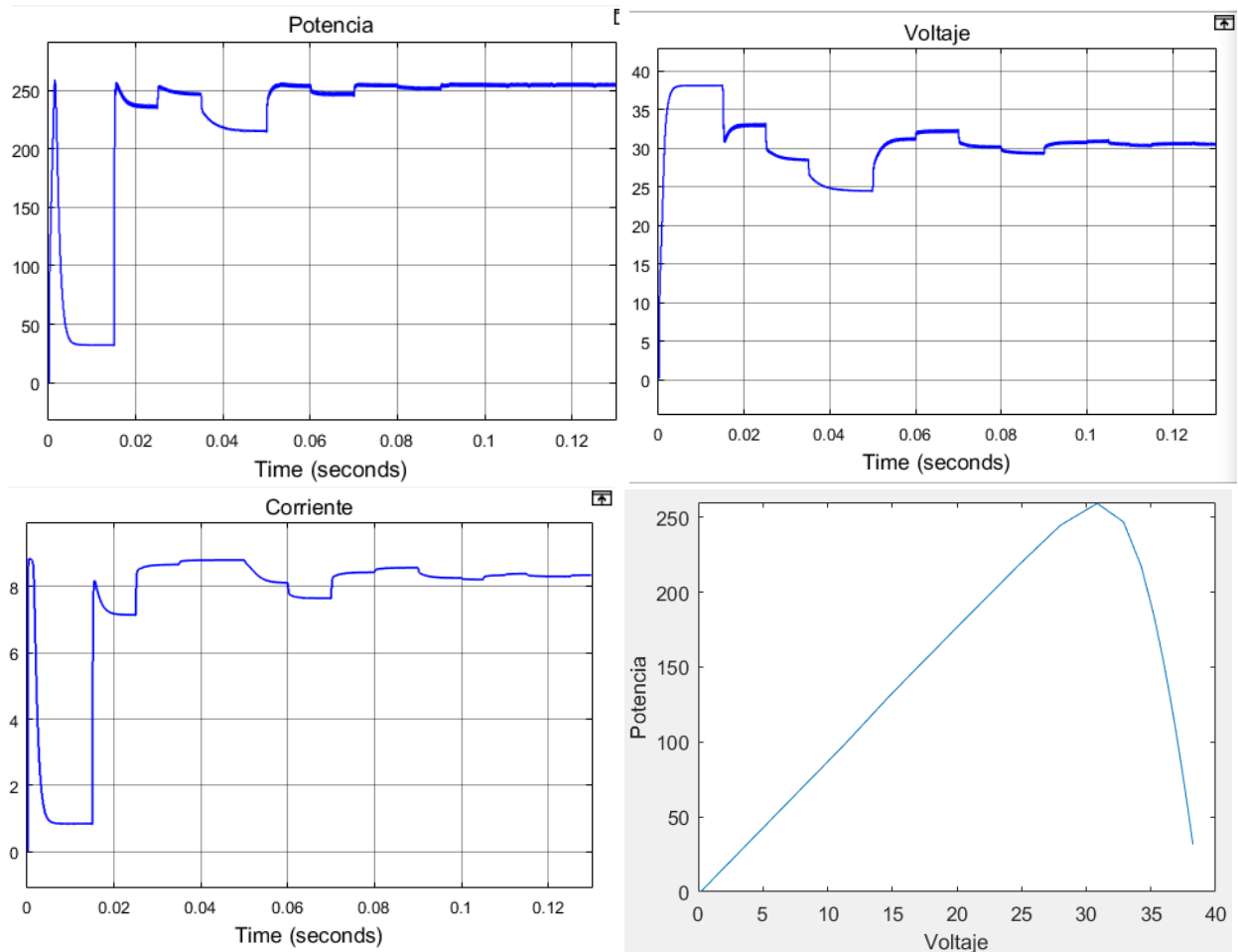


Figura 55: Simulación de rastreo con SH usando condiciones ideales.

Como se mencionó anteriormente, con este algoritmo se puede recorrer la curva de potencia escalando a su punto máximo desde ambos lados, es por esto que al observar los puntos de prueba en el gráfico de potencia contra voltaje, se dibuja gran parte de la curva de potencia. Sin embargo, esta curva se obtuvo tan completa ya que se tomaron en cuenta las transiciones de las curvas al cambiar los ciclos de trabajo de conmutación. En la práctica no se tomaron en cuenta las transiciones, sino los valores ya estabilizados, por lo que las gráficas de potencia contra voltaje tendrán menor resolución que lo obtenido en simulación.

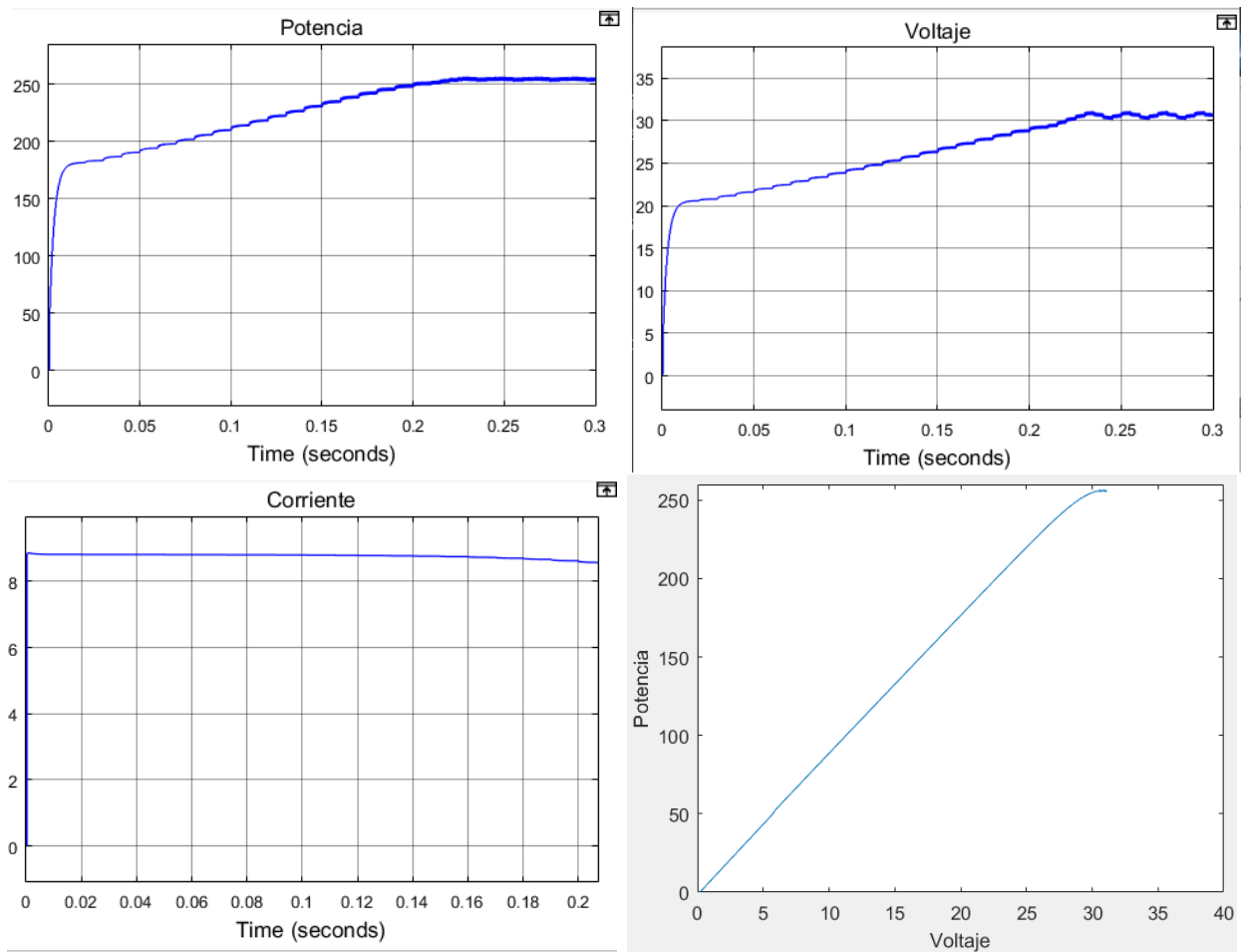


Figura 56: Simulación de rastreo con P&O usando condiciones ideales.

A diferencia del algoritmo SH, con el P&O solo se logra apreciar la sección de pendiente positiva de la curva de potencia contra voltaje. Pero al igual que con la simulación de SH, se obtiene una curva de potencia contra voltaje de mucha resolución al tomar en cuenta las transiciones.

Como se puede observar en las figuras anteriores, bajo las condiciones ideales se consigue la potencia máxima del panel solar. Sin embargo, en la práctica estas condiciones difícilmente se consiguen naturalmente, solo es posible acercarse a estas condiciones con un ambiente controlado, como el propuesto en [28].

## 4.2. Pruebas de algoritmo P&O

### Rastreo simple:

Para observar el comportamiento de las variables al ejecutar el algoritmo P&O y comparar con lo obtenido en simulación, se obtuvieron las gráficas de las variables individualmente en función del tiempo; además, se graficó la curva de potencia contra voltaje para mostrar en donde se ubicaron los puntos de prueba del algoritmo.

Las condiciones bajo las cuales se realizó el rastreo fueron  $Irradiancia = 805W/m^2$  y  $Temperatura = 68^{\circ}C$ . La prueba se realizó el día 23 de junio de 2021 a las 12:28 p.m.

en Cali, Colombia. Las curvas se muestran en la siguiente figura:

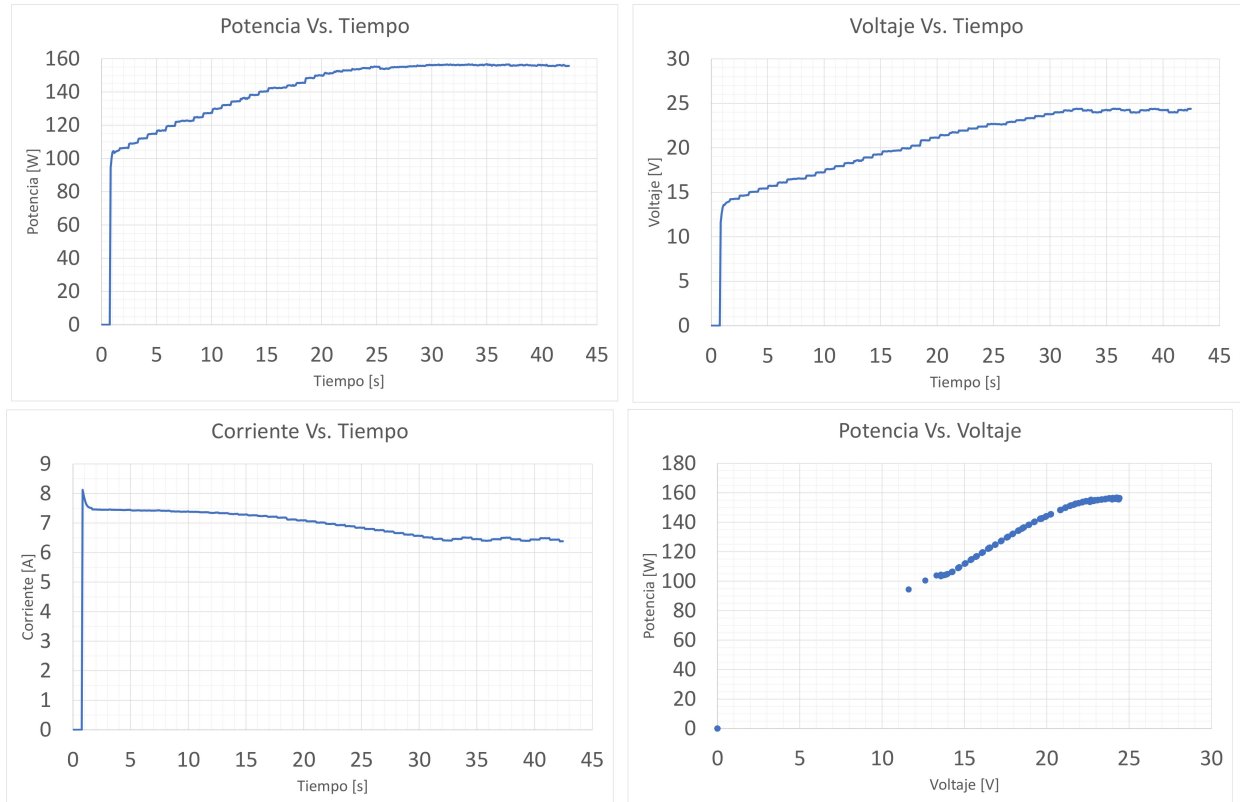


Figura 57: Curvas de rastreo usando P&O.

En las curvas se puede observar desde cual posición se comienza el rastreo con el control PWM en su máximo ciclo de trabajo (96%), que se ubica en un voltaje de 14V (los voltajes por debajo de 14V son los medidos en la transición debida a la repentina alimentación a la carga). Esto indica una limitación en el rango de rastreo, la cual está definida por el valor de la carga, es decir, no es posible rastrear en zonas cercanas a corto circuito, puesto que el máximo ciclo de trabajo (teórico) es 100%, en el cual la resistencia equivalente a la salida del panel sería la misma resistencia de la carga.

Cabe resaltar que con este algoritmo la precisión de los sensores de voltaje y corriente es fundamental, puesto que al aproximarse al punto de máxima potencia, la pendiente de la curva de potencia es muy pequeña, por lo que podría darse el caso de encontrar falsos puntos máximos.

A continuación se muestran los resultados de simulación usando las mismas condiciones climáticas:

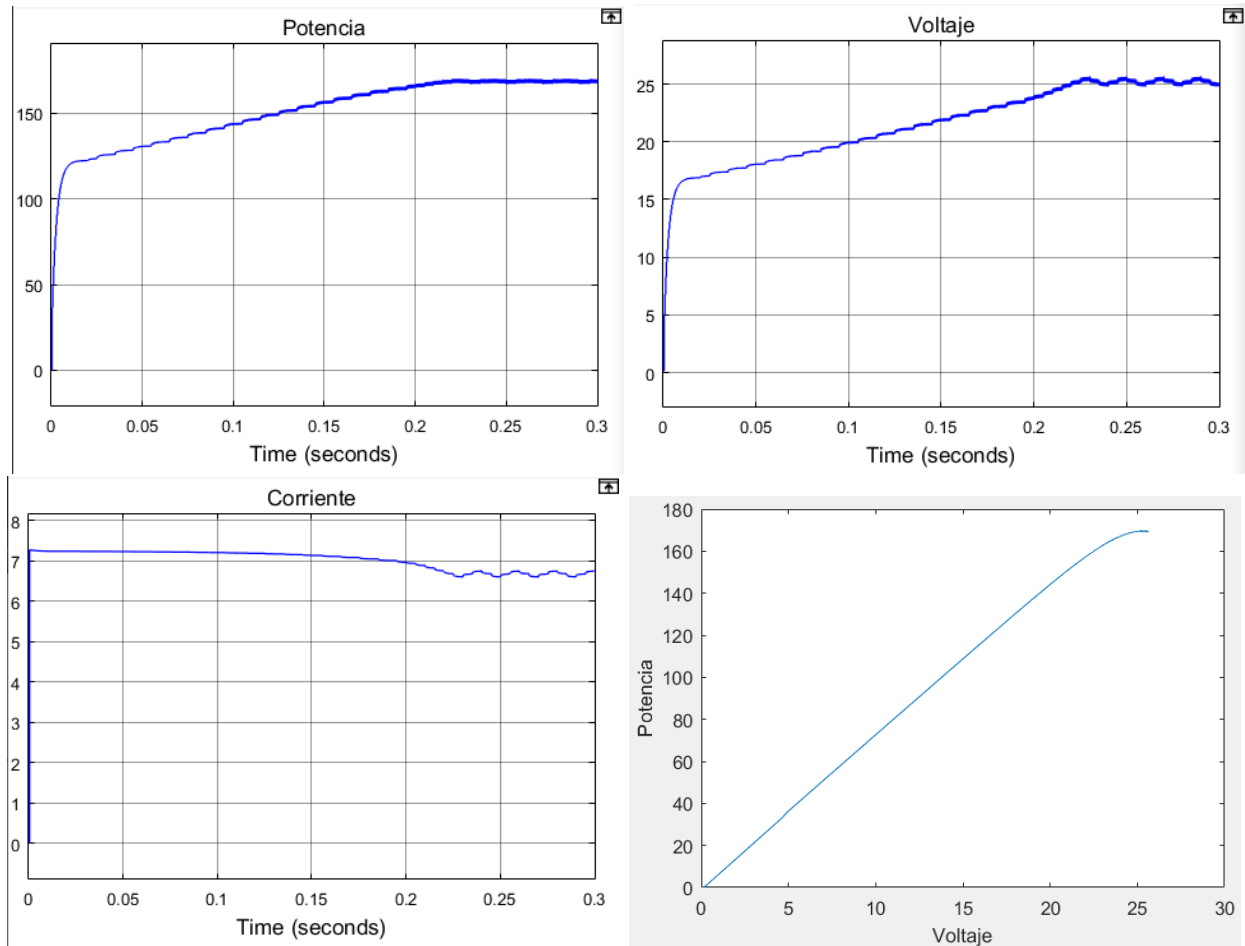


Figura 58: Simulación de rastreo usando P&O con condiciones  $Irradiancia = 805W/m^2$  y  $Temperatura = 68^{\circ}C$ .

En el rastreo simulado se observa que se alcanzó una potencia mayor, con una diferencia de  $11W$ . Esto se debe a que el sistema real presenta caída de tensión en el cable, que en este caso es de  $1,61V$ .

Otra diferencia que se obtiene en las curvas del sistema implementado y el simulado, son las curvas transitorias en la primera iteración, que como ya se mencionó, son debidas a la alimentación repentina a la carga.

#### Rastros consecutivos:

Se realizaron rastreos consecutivos con mismas condiciones climáticas (evitando la sombra de las nubes) para verificar su punto de convergencia. A continuación se muestran las curvas obtenidas con  $Irradiancia = 770W/m^2$  y  $Temperatura = 69^{\circ}C$  el 23 de junio de 2021 a las 11:27 a.m. en Cali, Colombia:

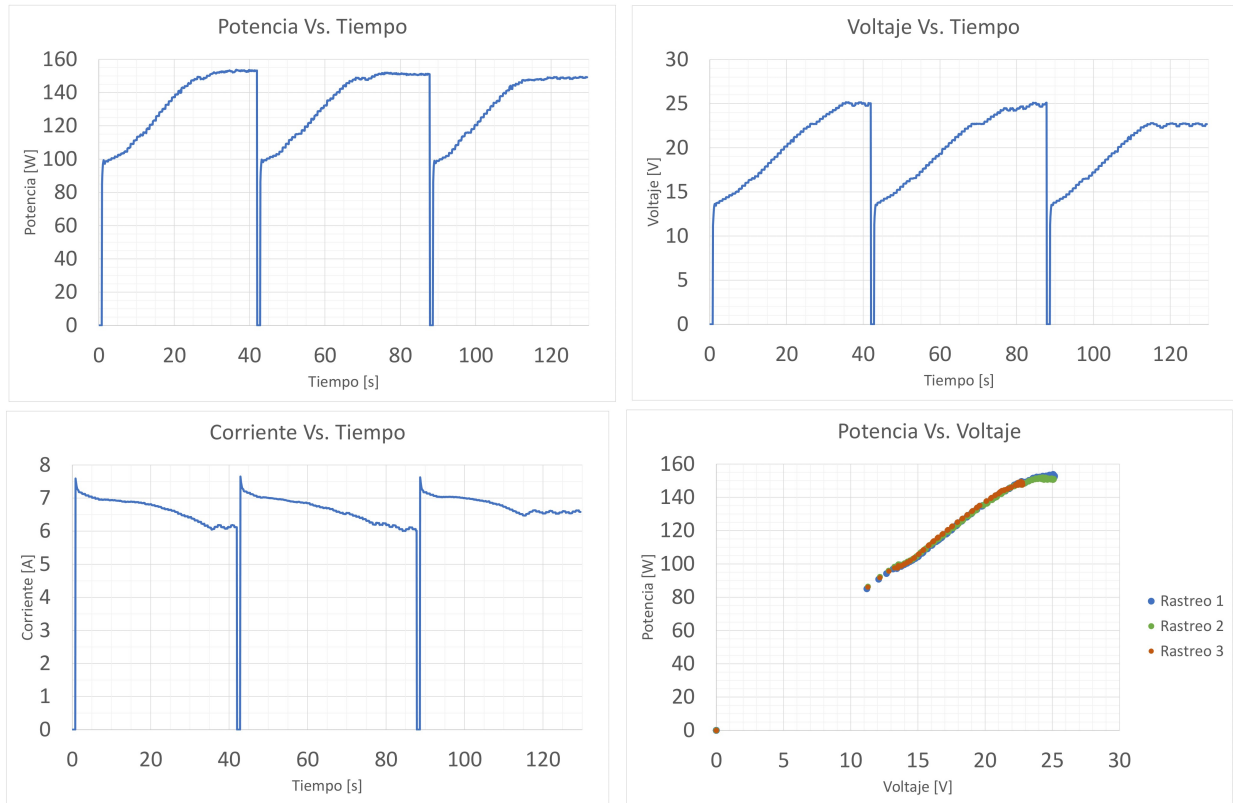


Figura 59: Curvas de rastros consecutivos usando P&O.

En estas curvas se puede apreciar que en los tres rastros se alcanzó el mismo punto de máxima potencia, sin embargo, en el tercer rastreo se obtuvo un menor voltaje y una mayor corriente que en los dos rastros anteriores. Esto se le atribuye a que la curva de potencia en zonas próximas al punto máximo tiene una pendiente cercana a cero, por lo que cualquier variación en la lectura de los sensores puede provocar que se encuentren puntos de máxima potencia similares, pero con voltajes y corrientes distintas. En la implementación realizada, estas variaciones de lectura de las variables se deben a que los sensores son susceptibles al ruido del conversor.

En el primer rastreo se tardó 36 segundos en encontrar el punto de máxima potencia, con un total de 42 iteraciones realizadas. En el segundo rastreo el tiempo fue de 34 segundos, con 40 iteraciones realizadas. En el tercer rastreo se tardó 30 segundos, con 35 iteraciones realizadas.

### 4.3. Pruebas de algoritmo SH

#### Rastreo simple:

De igual manera que con las pruebas de P&O, se obtuvieron las gráficas de las variables individualmente en función del tiempo y la curva de potencia contra voltaje para mostrar los puntos de prueba tomados en la ejecución del algoritmo.

A continuación se muestra un rastreo realizado bajo las condiciones de *Irradiancia* =  $802W/m^2$  y *Temperatura* =  $62^{\circ}C$  el 21 de junio de 2021 a las 12:33 p.m. en Cali, Colombia:

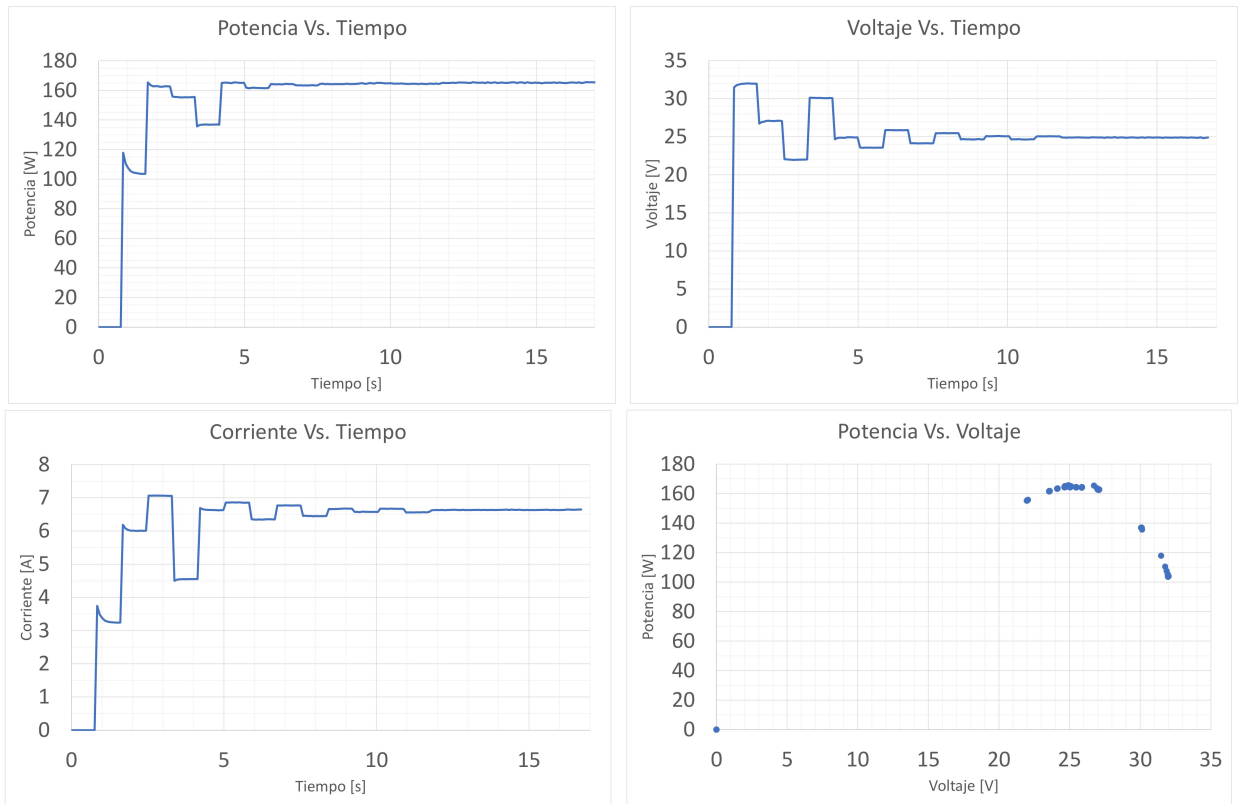


Figura 60: Curvas de rastreo usando SH.

En estas curvas se pueden apreciar más transiciones en las primeras iteraciones, puesto que hay grandes diferencias entre los ciclos de trabajo y la carga se resiste a cambios de voltaje abruptos.

La potencia máxima alcanzada fue de  $165W$  con un voltaje de  $25V$  y una corriente de  $6,6A$ . A continuación se muestran las curvas obtenidas en simulación bajo las mismas condiciones de irradiancia y temperatura:

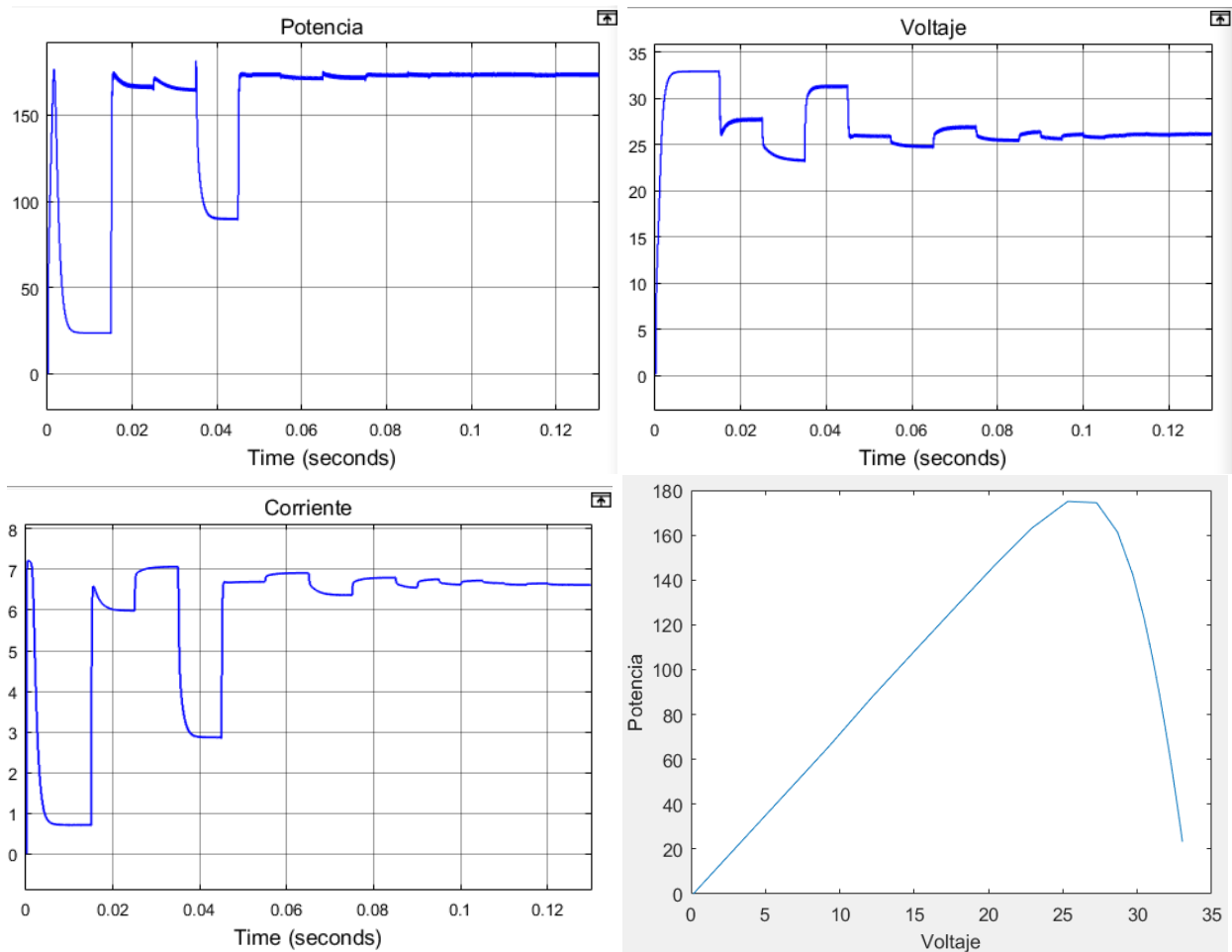


Figura 61: Curvas de rastreo simulando SH con  $Irradiancia = 802W/m^2$  y  $Temperatura = 62^{\circ}C$ .

Se observa que los valores a los que convergió el algoritmo en el sistema real son muy aproximados a lo obtenido en simulación. Existen diferencias en algunas iteraciones, ya que el sistema real utiliza ciclos de trabajo de salida aproximados, por lo que los puntos de prueba varían su posición respecto a los puntos de prueba de simulación; además, en la práctica el valor de impedancia de la carga no es fija, mientras que en simulación se utilizó directamente una resistencia con un valor fijo.

En la simulación se obtuvo una potencia  $10,7W$  por encima de lo obtenido con el sistema real, puesto que la caída de voltaje en el cable que conecta el panel con el conversor es de  $1,6V$  aproximadamente.

Las diferencias en el eje del tiempo se deben a que en el sistema real se necesita tener un tiempo de iteración mucho mayor al de simulación, debido a que la respuesta de la carga no es inmediata, lo que resulta en que al cambiarse de ciclo de trabajo, las variables tardan en estabilizarse. Además, los sensores de voltaje y de corriente tardan un tiempo en calcular los valores promedios, que depende del número de muestras que se especifique en la función de medición.

#### **Rastreos consecutivos:**

Se realizaron rastreos consecutivos en un corto tiempo, de manera que entre los rastreos se

tengan las menores variaciones de irradiancia posibles (evitando la sombra de nubes), para que la potencia máxima encontrada sea la misma en cada uno de ellos. A continuación se muestran los resultados de los rastreos bajo las condiciones de  $Irradiancia = 802W/m^2$  y  $Temperatura = 63^{\circ}C$  el 21 de junio de 2021 a las 12:35 p.m. en Cali, Colombia.

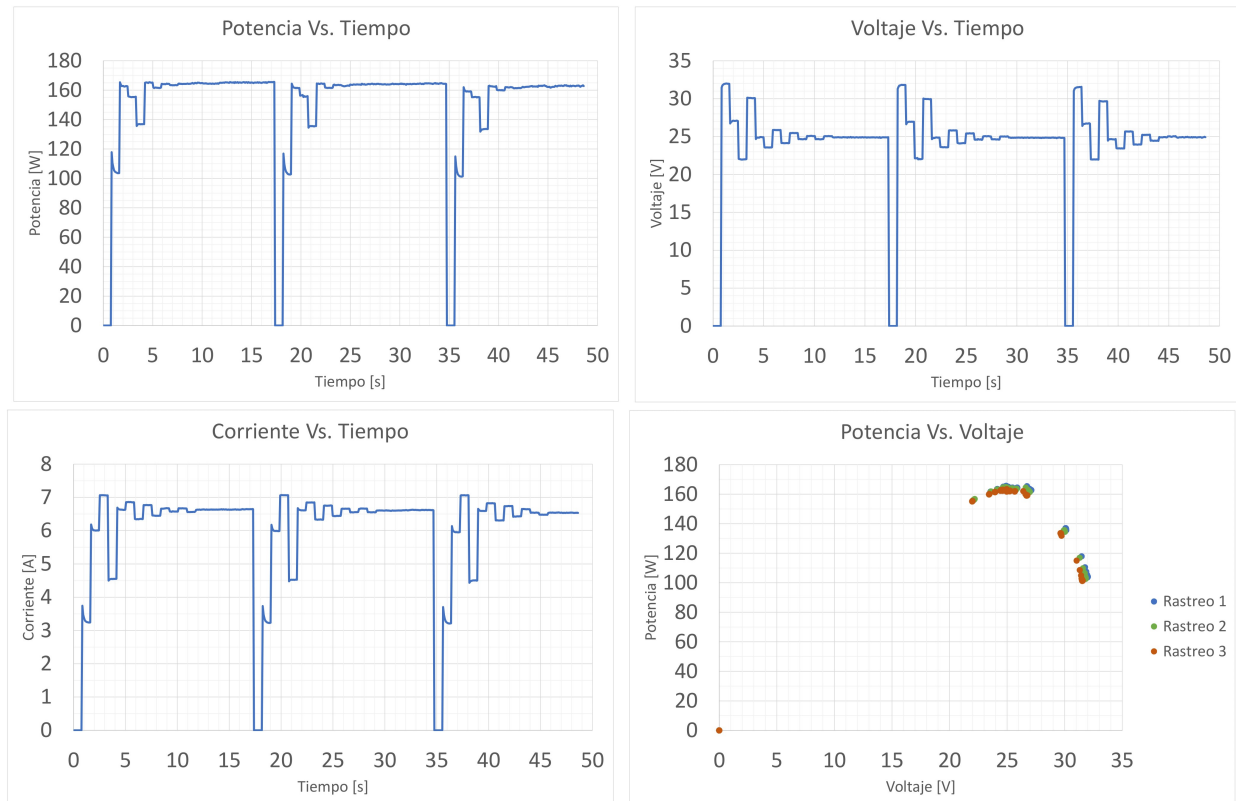


Figura 62: Curvas de rastreos consecutivos usando SH.

En los tres rastreos el algoritmo encontró puntos muy similares de potencia alrededor de  $165W$ , con un voltaje de  $25V$  y una corriente de  $6,6A$ . Se puede apreciar en la curva de potencia contra voltaje que los puntos de prueba fueron casi idénticos. En estas curvas también se puede apreciar que el tiempo de convergencia y el número de iteraciones es el mismo para los tres rastreos. Cada rastreo tardó 11 segundos y realizó 14 iteraciones en total.

#### 4.4. Comparación de desempeño de los algoritmos

Para comparar los desempeños de los algoritmos, se ejecutaron uno tras de otro en el menor tiempo posible, para que las condiciones climáticas afecten mínimamente los resultados. A continuación se mostrarán las curvas de dos de las pruebas, para poder detallar las diferencias comunes de los rastreos con cada algoritmo; y mas adelante se mostrarán en un cuadro los resultados obtenidos en el total de las pruebas realizadas. La siguiente prueba se realizó el 22 de junio de 2021 a las 10:42 a.m. en Cali, Colombia:

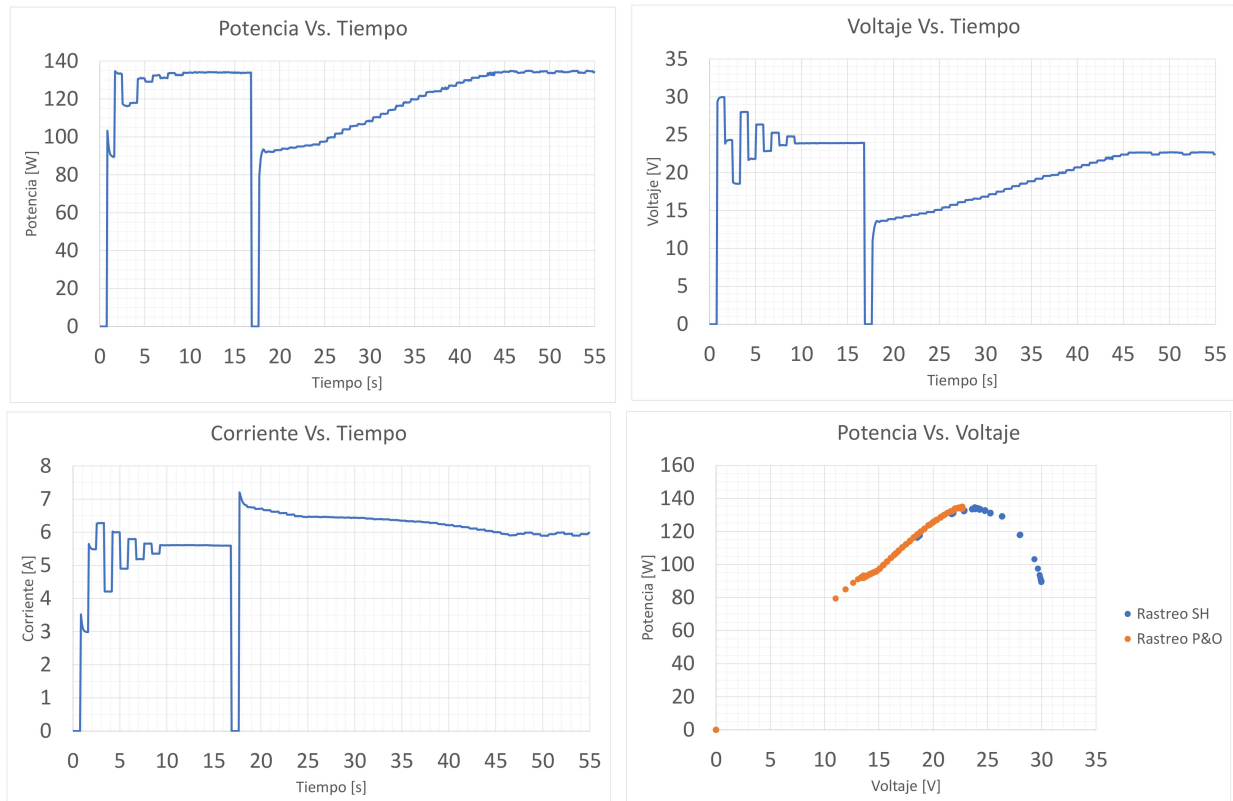


Figura 63: Comparación de curvas de rastreos con SH y P&O.

En esta prueba se observa la significativa diferencia de número de iteraciones y de tiempo de convergencia entre los algoritmos: Para el caso de SH se alcanzó una potencia de  $134W$  en 9 segundos con 11 iteraciones; y en el caso de P&O se alcanzó también una potencia de  $134W$  pero en un tiempo de 29 segundos con 35 iteraciones.

En la curva de potencia contra voltaje se observan los puntos de prueba usados con cada algoritmo: mientras el P&O utiliza todos los puntos posibles de prueba en el lado de pendiente positiva de la curva, el algoritmo SH sitúa puntos de prueba a los dos lados del punto de máxima potencia de manera que escala desde ambos sentidos con saltos mucho mayores. Gracias a esto, es mas probable para el rastreo SH llegar a un punto de máxima potencia con un voltaje mayor, puesto que el P&O suele tomar por encontrado el punto de máximo con un voltaje, que en caso de aumentarse, difícilmente se percibirá un cambio en la potencia al haber una pendiente muy pequeña.

La siguiente prueba muestra otra comparación de rastreos tomada el 23 de junio de 2021 a las 12:00 p.m. en Cali, Colombia. Esta vez los rastreos fueron ejecutados en el orden inverso a como se mostró en la prueba anterior:

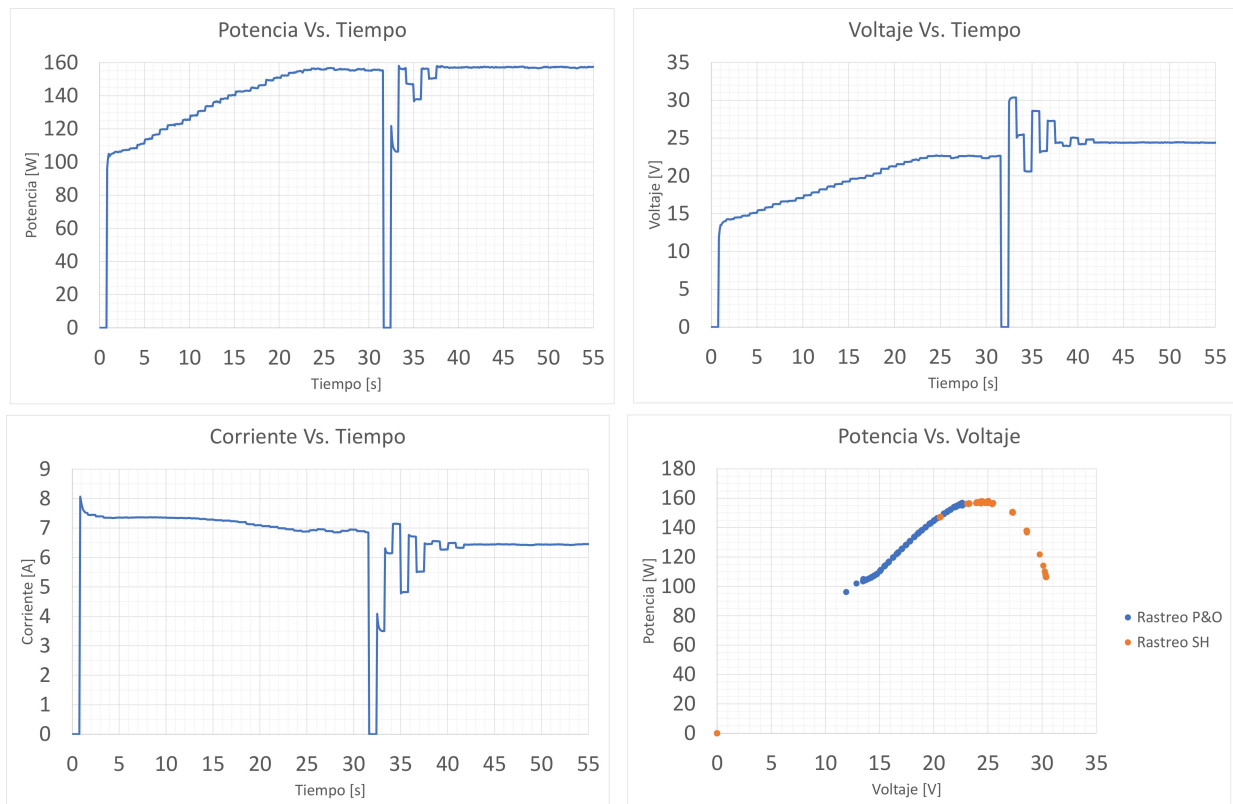


Figura 64: Comparación de curvas de rastreos con P&O y SH.

En esta prueba la potencia encontrada fue de  $156W$  con ambos rastreos. En el caso de P&O se tardó 23 segundos, con un total de 27 iteraciones para encontrar el punto máximo; mientras que con SH se tardó 10 segundos con un total de 12 iteraciones.

De igual manera que con la prueba anterior, el voltaje fijado en el rastreo P&O es menor que el fijado con SH.

A continuación se muestra un cuadro que resume los resultados de varias pruebas con ambos algoritmos:

Cuadro 8: Resultado de pruebas de P&O y SH

Prueba	Tiempo de convergencia		Potencia obtenida	
	Algoritmo P&O	Algoritmo SH	Algoritmo P&O	Algoritmo SH
1	29 seg	9 seg	134W	134W
2	23 seg	10 seg	155W	157W
3	30 seg	12 seg	131W	132W
4	25 seg	10 seg	154W	156W
5	32 seg	12 seg	152W	152W
6	34 seg	12 seg	123W	123W
7	31 seg	9 seg	163W	162W
8	30 seg	12 seg	141W	141W

Con estos resultados se aprecia la considerable mejora en el tiempo de convergencia al usar rastreo SH frente al convencional algoritmo P&O, llegando en algunos casos a ser tres veces mas rápido. Además, con frecuencia se obtiene una mayor potencia con el rastreo SH. Este análisis es válido para los rastreos en donde las condiciones iniciales se definen de manera que se realice un rastreo completo de la curva según la naturaleza de cada método. Sin embargo, conociendo los valores típicos de ciclo de trabajo se pueden lograr variaciones muy significativas en el número de iteraciones, y por ende, en el tiempo de convergencia.

#### 4.5. Ciclos de trabajo habituales para ubicar el punto de máxima potencia

Los ciclos de trabajo con los que se encuentran los puntos de máxima potencia están dentro de un rango independientemente de la potencia obtenida. En las pruebas realizadas, el mínimo ciclo de trabajo con el que se encontró un punto de máxima potencia fue 71 %, mientras que el máximo fue 78 %. Conociendo este rango de ciclos de trabajo es posible definir condiciones iniciales para los métodos de rastreo, de manera que se reduzca el número de iteraciones y el tiempo de convergencia.

Un ejemplo de esto se puede observar en las siguientes curvas:

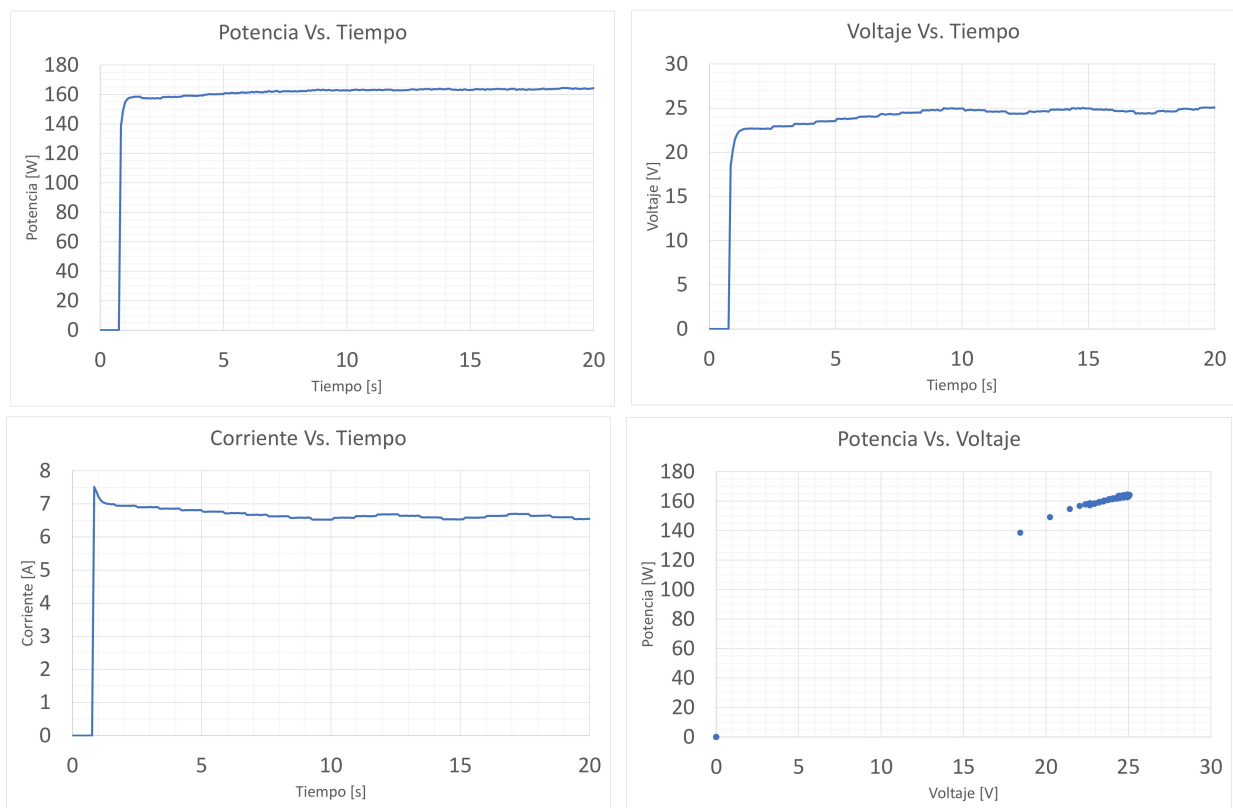


Figura 65: Rastreo P&O con ciclo de trabajo inicial de 78 %.

Al iniciar el rastreo P&O con un ciclo de trabajo del 78 %, el tiempo de convergencia fue de 9 segundos, con un total de 11 iteraciones.

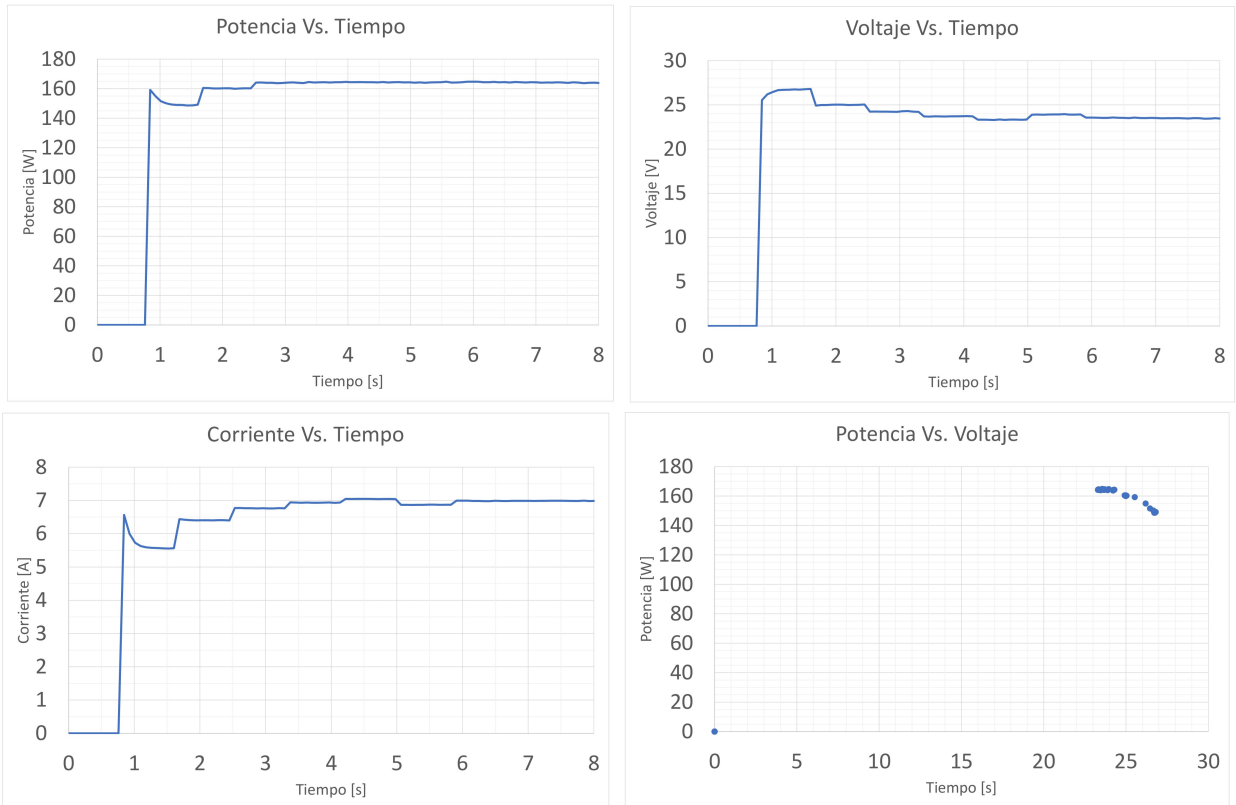


Figura 66: Rastreo SH con rango de búsqueda entre el 55 % y el 80 %.

Al definir la zona de rastreo de SH entre el 50 % y el 80 %, el tiempo de convergencia fue de 6 segundos, con un total de 7 iteraciones.

# Capítulo 5

## 5. Conclusiones y trabajos futuros

- La implementación mostrada en [2] permite un tiempo de convergencia muy bajo para el método SH, llegando a ser incluso de 1 segundo. Sin embargo, con la implementación mostrada en este artículo no se obtienen tiempos tan bajos, debido a que cada iteración debe cumplir con un tiempo mínimo para que las variables se estabilicen, definido por el tiempo de respuesta de la carga y del convertor DC-DC; mientras que en [2] se definió el tiempo de iteración tomando en cuenta únicamente el tiempo de respuesta del panel solar utilizado.
- El método SH muestra ser mucho más rápido que el método P&O cuando la búsqueda del punto de máxima potencia se hace con el rango completo de la curva de potencia, es decir, sin seleccionar condiciones iniciales que optimicen el proceso. Sin embargo, al seleccionar estas condiciones iniciales no se puede predecir cual algoritmo converge más rápido, puesto que en el caso del método P&O el tiempo de convergencia depende de que tan cercano esté el punto inicial de rastreo al punto de máxima potencia, existiendo incluso la posibilidad de que el punto inicial de rastreo coincida con el punto de máxima potencia; mientras que en el caso de SH, el tiempo de convergencia depende de qué tan estrecho sea el rango de búsqueda inicial.
- El sistema implementado tiene la capacidad de acoplar la carga de manera de que se exija la máxima potencia del panel solar. Sin embargo, de la potencia obtenida, entre el 78 % y el 80 % se entrega a la carga, debido a que esta fue la eficiencia de conversión lograda con los componentes seleccionados para el convertor. Esta eficiencia se puede aumentar si se implementa el convertor con componentes pasivos nuevos y de alta calidad que eviten que la conmutación de altas corrientes generen un ruido que afecte el funcionamiento del sistema.
- Para el diseño de convertidores es fundamental conocer los componentes parasitarios de cada elemento del circuito. Sin embargo, los capacitores del convertor implementado son piezas de circuitos desechados. Se desconoce el tiempo de uso de estos componentes y no se cuenta con medidores RLC que permitan la medición del estado actual de los valores de inductancia y resistencia serie equivalente. Es por esto que la selección de capacitores del convertor se hizo por método de ensayo y error, buscando un equilibrio entre alta eficiencia y bajo nivel de ruido.
- El convertor implementado probó una buena calidad de conversión y muy alta eficiencia para cargas de 140W, lo que indica que es ideal para sistemas fotovoltaicos que manejen este nivel de potencia, siempre y cuando la carga utilizada tenga una resistencia equivalente menor a la resistencia  $R_{mpp}$  del panel solar, ya que se trata de un convertor reductor.
- El tiempo que debe tardar cada iteración de los algoritmos está limitado por la naturaleza de la carga y su velocidad de respuesta. Es necesario aumentar el tiempo de iteración si las variables no alcanzan a estabilizarse antes de que inicie la siguiente

iteración para evitar lecturas erróneas, y por ende, puntos de prueba incorrectos. En el caso contrario, de que la carga tenga una velocidad de respuesta rápida, se puede disminuir el tiempo de iteración para lograr tiempos de convergencia más bajos.

- El tiempo de iteración también varía por la cantidad de muestras que se utilicen para obtener mediciones precisas de las variables de voltaje y corriente. Para el sistema implementado se usó el promedio de 250 muestras para obtener valores con poca afectación del ruido del conversor. En caso de implementarse un sistema donde se puedan obtener buenas mediciones sin tomar tantas muestras, se puede reducir el tiempo de iteración.
- El algoritmo SH se ve afectado si durante su ejecución existen cambios en las condiciones climáticas, principalmente en la irradiancia, ya que es el parámetro que varía más rápido. El algoritmo SH estará más completo con un mecanismo que le permita identificar que su ejecución fue afectada por estos cambios o si las condiciones se mantuvieron estables. Esto es posible sin necesidad de sensores externos de irradiancia y temperatura, es suficiente con el análisis de las curvas obtenidas.
- El algoritmo P&O requiere sensores muy precisos de voltaje y corriente, puesto que como se observó en las pruebas, puede no encontrar el verdadero punto de máxima potencia sino un punto cercano.
- Cualquier algoritmo basado en el P&O que mida las mismas variables que este, es implementable en el sistema propuesto en este trabajo, únicamente se necesitará modificación a nivel de software que permita definir el flujo del algoritmo.
- El sistema implementado fue alimentado de manera independiente, esto permite trabajar al conversor sin un voltaje mínimo de entrada. No obstante, el sistema queda limitado a la necesidad de tener acceso a la energía convencional. Para eliminar este obstáculo se requiere una fuente regulada de voltaje que permita alimentar el microcontrolador, el driver del mosfet, los sensores y el potenciómetro digital, a partir de la misma salida del panel solar. Esta fuente deberá tener un robusto blindaje contra el ruido para asegurar el buen funcionamiento del microcontrolador.
- Para añadir baterías al sistema implementado en este proyecto, es necesario adecuar los algoritmos para que trabajen con una correcta lógica de carga de baterías, de manera que se entregue la máxima potencia únicamente cuando las baterías, junto con las cargas conectadas a esta, así lo requieran.
- El sistema MPPT implementado puede trabajar en conjunto con el sistema híbrido planteado en [28] para obtener una mejora muy significativa en la irradiancia y temperatura a la que se somete el panel solar y así alcanzar la máxima potencia absoluta que este puede entregar.

## Referencias

- [1] S. Ramírez Castaño, *Redes de Distribución de Energía*, Tercera ed., Manizales: Centro de Publicaciones Universidad Nacional de Colombia Sede Manizales, 2004
- [2] Giraldo Castañeda, C. (2008). Maximum power point tracking using modified P&O method for the off grid radar. Master. Universidad de Puerto Rico - Mayagüez.
- [3] A. Basnet, “Architectural Integration of Photovoltaic and Solar Thermal Collector Systems into buildings”, Master’s Thesis, no. June, 2012.
- [4] S. Cuestas and L. Lebus, “Diseño de un seguidor de punto máximo de potencia”, pp. 37–42
- [5] Redacción VIVIR, “En 2017 la energía solar creció más rápido que otros combustibles: EL ESPECTADOR”, 8 Enero 2018. [Online]. Available: [www.elespectador.com](http://www.elespectador.com).
- [6] “Esquema de principio - Fotovoltaica aislada - Cenit Solar”, Cenitsolar.com, 2021. [Online]. Available: [www.cenitsolar.com](http://www.cenitsolar.com) [Accessed: 5- Jan- 2021].
- [7] R. Serrano, “Tipos de paneles fotovoltaicos”, TRITEC Intervento, Santiago de Chile, 2016
- [8] Rashid, M. (n.d.). *Electrónica de potencia: Circuitos, dispositivos y aplicaciones*. 2nd ed. pp.[303-342].
- [9] Sarsa, *Circuito Eléctrico de Conversor Buck*. 2008.
- [10] L. Cheng, P. Acuna, R. P. Aguilera, M. Ciobotaru and J. Jiang, “Model predictive control for DC-DC boost converters with constant switching frequency”, 2016 IEEE 2nd Annual Southern Power Electronics Conference (SPEC), Auckland, New Zealand, 2016, pp. 1-6, doi: 10.1109/SPEC.2016.7846189.
- [11] W. Dvorak, *Buckboost converter schematic*. 2010.
- [12] B. Han, M. Kim, S. Lee and J. S. Lee, “Dynamic modeling and integral sliding mode controller design for Cuk converter under load variation”, 2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia), Seoul, Korea (South), 2015, pp. 449-455, doi: 10.1109/ICPE.2015.7167824.
- [13] T. Vargas and A. Abrahamse, “TRAZADOR DE CURVAS I-V DE CÓDIGO ABIERTO PARA MONITOREO DE RENDIMIENTO DE PANELES FOTOVOLTAICOS EN BOLIVIA”, INVESTIGACION & DESARROLLO, vol. 14, no. 1, pp. 100-116, 2014. Available: [www.scielo.org](http://www.scielo.org). [Accessed: 25- Oct- 2019].
- [14] J. J. Rodríguez, “Emulador de un panel solar considerando irradiancia y temperatura”. [Online]. Available: [hdl.handle.net](http://hdl.handle.net).
- [15] “Instalación Solar Fotovoltaica para Vivienda”, Ingemecanica.com, 2021. [Online]. Available: [ingemecanica.com](http://ingemecanica.com). [Accessed: 19- Feb- 2021].

- [16] B. Valenzuela Pazmiño, “ DETERMINACIÓN DEL PUNTO DE MÁXIMA POTENCIA EN SISTEMAS DE BOMBEO SOLAR FOTOVOLTAICO UTILIZANDO EL ALGORITMO PERTURBACIÓN OBSERVACIÓN P&O”. Quito, 2020, p. 14.
- [17] “Maximum Power Point Tracking (MPPT) Charge Controller Working Principle | Electrical Academia”, Electrical Academia. [Online]. Available: [electricalacademia.com](http://electricalacademia.com). [Accessed: 21- Sep- 2019].
- [18] S. Li, H. Liao, H. Yuan, Q. Ai and K. Chen, A MPPT strategy with variable weather parameters through analyzing the effect of the DC/DC converter to the MPP of PV system. ELSEVIER, 2017, p. 2.
- [19] YGE 60 Cell 35mm SERIES. 2014.
- [20] “Input and Output Noise in Buck Converters - Maxim Integrated”, Maximintegrated.com, 2021. [Online]. Available: [www.maximintegrated.com](http://www.maximintegrated.com). [Accessed: 26- Apr- 2021].
- [21] Wuxi NCE Power Semiconductor Co., Ltd, NCE8580 NCE N-Channel Enhancement Mode Power MOSFET. pp. 1-2.
- [22] Thinki Semiconductor Co., Ltd, F30S60S ThinkiSemi 30Amperes,600Volts SwitchMode Ultrafast Recovery Epitaxial Diode. pp. 1 - 3.
- [23] A. Cucueff, Circuito Oscilante Libre. Universidad Nacional del Nordeste, p. 3.
- [24] Generador de señal PWM | Serie del 555 | #5. Mundo Electrónica, 2019.
- [25] Compensated Devices Incorporated, 1N4148. p. 2.
- [26] “Arduino Nano Board Guide (Pinout, Specifications, Comparison)”, Makerguides.com, 2021. [Online]. Available: [www.markerguides.com](http://www.markerguides.com). [Accessed: 23- Jul- 2020].
- [27] “Modulo sensor corriente ACS712-20A”, Lacasadelabanda.com, 2021. [Online]. Available: [www.lacasadelabanda.com](http://www.lacasadelabanda.com). [Accessed: 03- Apr- 2021].
- [28] J. Valencia Espitia and M. Díaz Vásquez, Evaluación de un sistema híbrido para mejorar la eficiencia de conversión de energía en paneles fotovoltaicos. Cali.
- [29] Grupo Simec Chile SRL, “Análisis de un Sistema e Iluminación, utilizando ampollitas de bajo Consumo y alimentado por paneles fotovoltaicos”, vol. 1, p. 138, 2010
- [30] R. Bhol, R. Dash, A. Pradhan, and S. M. Ali, “Environmental effect assessment on performance of solar PV panel”, 2015 Int. Conf. Circuits, Power Comput. Technol. [ICCPCT-2015], pp. 1–5, 2015
- [31] M. Abderrezek and M. Fathi, “Experimental study of the dust effect on photovoltaic panels’ energy yield”, Sol. Energy, vol. 142, pp. 308–320, 2017
- [32] S. Akhlaghi, S. Member, M. Sarailoo, and S. Member, “Study of Sufficient Number of Optimal Tilt Angle Adjustment to Maximize Residential Solar Panels Yield”, 2017.

# Capítulo 6

## 6. Anexos

### 6.1. Diseño de PCB de convertor DC-DC

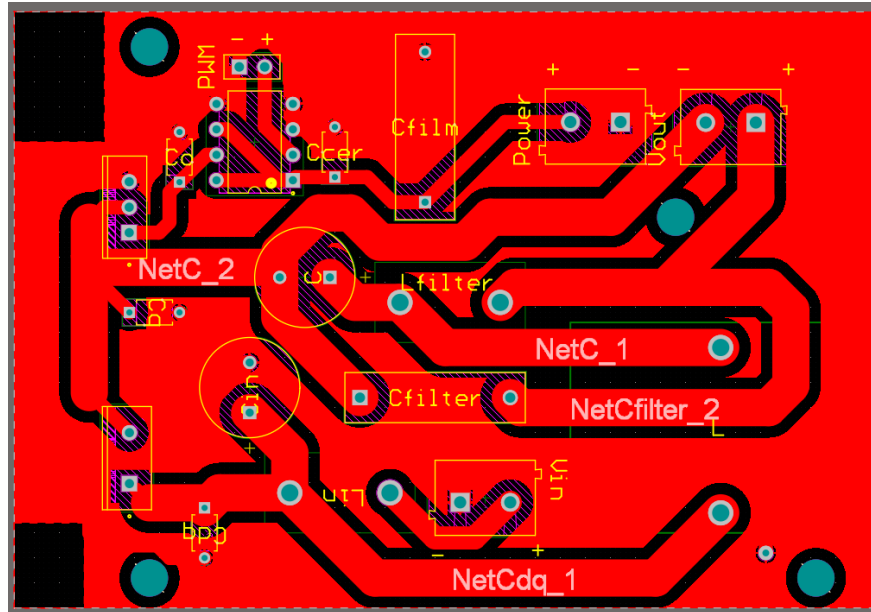


Figura 67: Diseño de PCB del convertor DC-DC

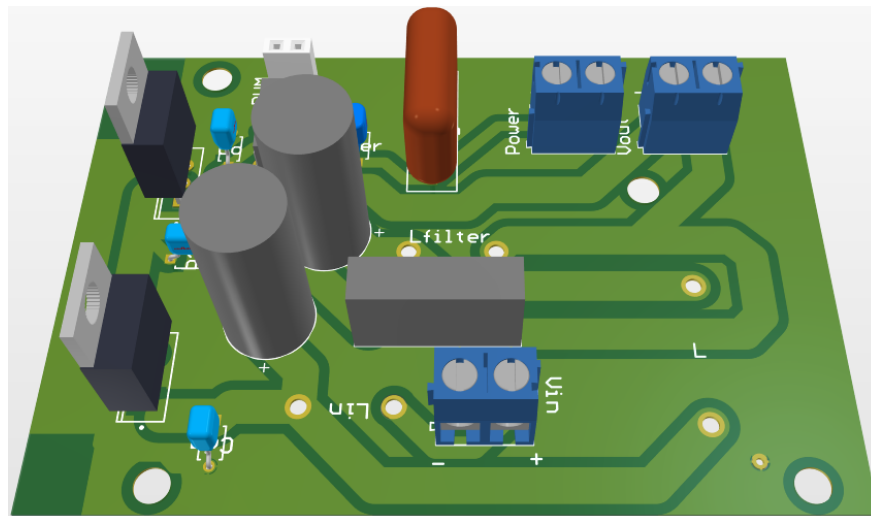


Figura 68: Vista 3D de diseño de PCB del convertor DC-DC

## 6.2. Pruebas de conversor con carga de 140W

Pruebas de potencia y eficiencia del conversor con carga de 140W								
Duty código	Duty Real	Vin	Iin	Vout	Iout	Pin	Pout	Eficiencia
0%	15,4%	27,22	0,352	3,6	1,83	9,58	6,59	68,8%
5%	18,6%	27,19	0,479	4,8	2,091	13,02	10,04	77,1%
10%	22,5%	27,16	0,624	5,6	2,357	16,95	13,20	77,9%
15%	27,1%	27,12	0,768	6,8	2,577	20,83	17,52	84,1%
20%	31,8%	27,06	0,965	8	2,829	26,11	22,63	86,7%
25%	35,7%	27,02	1,135	8,8	3,025	30,67	26,62	86,8%
30%	38,8%	26,98	1,316	10	3,209	35,51	32,09	90,4%
35%	43,4%	26,92	1,547	11,2	3,421	41,65	38,32	92,0%
40%	47,3%	26,87	1,75	12,1	3,592	47,02	43,46	92,4%
45%	51,9%	26,8	2,002	12,8	3,782	53,65	48,41	90,2%
50%	55,8%	26,74	2,23	13,6	3,942	59,63	53,61	89,9%
55%	58,3%	26,68	2,466	14,8	4,098	65,79	60,65	92,2%
60%	62,4%	26,61	2,734	16	4,269	72,75	68,30	93,9%
65%	65,9%	26,54	2,99	16,8	4,414	79,35	74,16	93,4%
70%	70,6%	26,45	3,298	18	4,583	87,23	82,49	94,6%
75%	74,5%	26,38	3,56	18,8	4,714	93,91	88,62	94,4%
80%	77,7%	26,3	3,851	19,6	4,852	101,28	95,10	93,9%
85%	82,3%	26,2	4,204	20,8	5,013	110,14	104,27	94,7%
90%	86,7%	26,11	4,516	22	5,16	117,91	113,52	96,3%
95%	91,8%	25,99	4,931	23,2	5,326	128,16	123,56	96,4%
100%	96,0%	25,87	5,347	24,4	5,49	138,33	133,96	96,8%

Figura 69: Tabla de pruebas del conversor con carga de 140W

### 6.3. Pruebas de conversor con carga de 250W

Pruebas de potencia y eficiencia del conversor con carga de 250W											
Duty código	Duty Real	Vin	Iin	Vout	Iout	Pin	Pout	Eficiencia			
0%	15,4%	30,26	0,601	3,4	3,447	18,19	11,72	64,4%			
5%	18,6%	30,21	0,834	4,4	3,942	25,20	17,34	68,8%			
10%	22,5%	30,15	1,127	5,6	4,462	33,98	24,99	73,5%			
15%	27,1%	30,09	1,404	6,6	4,876	42,25	32,18	76,2%			
20%	31,8%	30,02	1,763	7,6	5,319	52,93	40,42	76,4%			
25%	35,7%	29,95	2,079	8,6	5,663	62,27	48,70	78,2%			
30%	38,8%	29,88	2,403	9,4	5,981	71,80	56,22	78,3%			
35%	43,4%	29,82	2,819	10,6	6,35	84,06	67,31	80,1%			
40%	47,3%	29,78	3,182	11,6	6,63	94,76	76,91	81,2%			
45%	51,9%	29,74	3,633	12,4	6,97	108,05	86,43	80,0%			
50%	55,8%	29,7	4,029	13,2	7,23	119,66	95,44	79,8%			
55%	58,3%	29,66	4,407	14	7,43	130,71	104,02	79,6%			
60%	62,4%	29,61	4,879	14,8	7,67	144,47	113,52	78,6%			
65%	65,9%	29,27	5,311	15,4	7,91	155,45	121,81	78,4%			
70%	70,6%	29,15	5,82	16,4	8,14	169,65	133,50	78,7%			
75%	74,5%	29,47	6,32	17,4	8,41	186,25	146,33	78,6%			
80%	77,7%	29,42	6,8	18,2	8,6	200,06	156,52	78,2%			
85%	82,3%	29,37	7,36	18,8	8,81	216,16	165,63	76,6%			
90%	86,7%	29,32	7,87	19,6	9	230,75	176,40	76,4%			
95%	91,8%	29,25	8,51	20,4	9,21	248,92	187,88	75,5%			
100%	96,0%	29,19	9,14	21,2	9,42	266,80	199,70	74,9%			

Figura 70: Tabla de pruebas del conversor con carga de 250W

#### 6.4. Diseño de PCB para control PWM

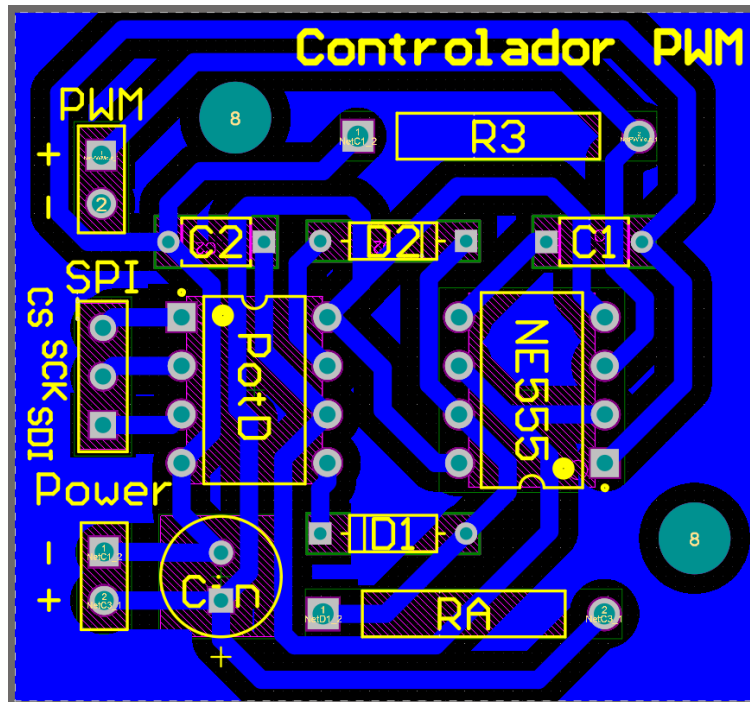


Figura 71: Diseño de PCB de control PWM

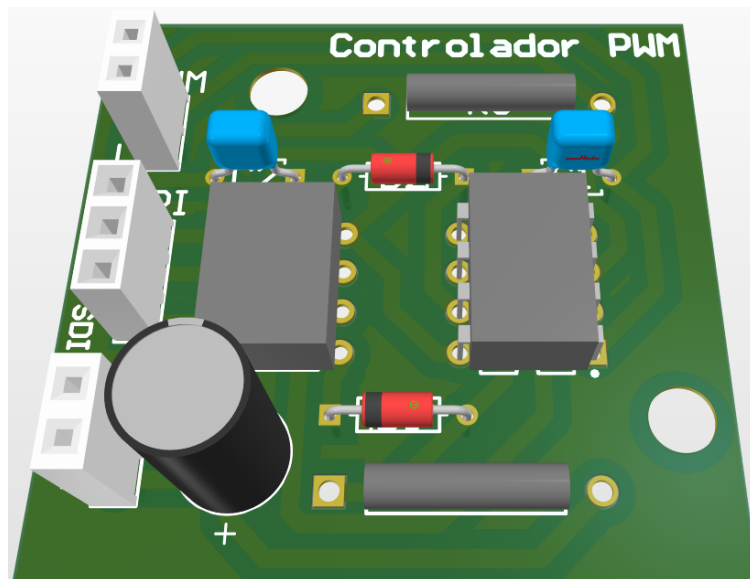


Figura 72: Vista 3D de diseño de PCB de control PWM

#### 6.5. Código de implementación de algoritmo SH

```
1 #include <SPI.h>
```

```

2
3 byte address = 0x00;
4 int CS = 10;
5 int sensorPinV = A1; // Pin para el voltaje
6 int sensorPinC = A0; // Pin para la corriente
7 int puntos_muestra = 10;
8
9 //Condiciones iniciales del algoritmo SH y declaracion de variables
10 float L = 0; //Limite inferior de zona de rastreo
11 float R = 1; //Limite superior de zona de rastreo
12 float Ld;
13 float Rd;
14 float Dmax = 1;
15 float Dmin = 0;
16 bool CorreSH = true;
17 int Contador = 0;
18 int Contador2 = 0;
19 float V;
20 float I;
21 float Pold;
22 float Pold_av;
23 float Pnew;
24 float Pnew_av;
25 float Dold;
26 float Duty = 0.33;
27 float Pmax = 0;
28 float Pumbral_min;
29 float Pumbral_max;
30 float Umbral = 3; //en watts. Umbral de potencia.
31 int TamPotencia = 40;
32 float PotEstab = 2; //variacion de potencia maxima para definir ...
    estabilidad
33 float CompDuty = 10;
34 bool bandera = false;
35
36 void setup() {
37     Serial.begin(9600);
38     pinMode(CS, OUTPUT);
39     SPI.begin();
40     digitalPotWrite(128);
41     delay(5000);
42     for (int h = 0; h < puntos_muestra; h++) {
43         Graficar(0.0, 0.0, 0.0, 1);
44     }
45 }
46 void loop() {
47
48     if (Contador == 0) {
49         digitalPotWrite(round(128 * (1 - Duty)));
50         delay(100);
51         Pold_av = 0;
52         for (int x = 0; x < puntos_muestra; x++) {
53             I = CurrentFunction(250);
54             V = VoltageFunction(250);
55             Pold = V * I;

```

```

56     Pold_av = Pold + Pold_av;
57     Graficar(Pold, V, I, Duty);
58 }
59 Pold_av = Pold_av / puntos_muestra;
60 Dold = Duty;
61 Duty = 0.75;
62 }
63 else {
64     digitalPotWrite(round(128 * (1 - Duty)));
65     delay(100);
66     Pnew_av = 0;
67     for (int x = 0; x < puntos_muestra; x++) {
68         I = CurrentFunction(250);
69         V = VoltageFunction(250);
70         Pnew = V * I;
71         Pnew_av = Pnew + Pnew_av;
72         Graficar(Pnew, V, I, Duty);
73     }
74     Pnew_av = Pnew_av / puntos_muestra;
75     CompDuty = round(128 * (1 - Duty)) - round(128 * (1 - Dold));
76
77     // Criterios de mpp encontrado
78     if (CompDuty == 0) { // mpp encontrado
79         if (bandera == true && (Duty < (Ld+0.005) || Duty > (Rd-0.005))...
80     ) { // Si en la histeresis el mpp queda al borde del rango de ...
81     busqueda
82         L = Duty - 0.1;
83         R = Duty + 0.1;
84         Ld = L;
85         Rd = R;
86         Pold_av = Pnew_av;
87         Dold = Duty;
88         Duty = (Dold + R) / 2;
89         Contador2 = 0;
90     }
91     else {
92         bandera = false;
93     }
94     if (Contador2 == 0) {
95         Pmax = Pnew_av;
96         Pumbral_min = Pmax - (Umbral / 2); // definicion de umbral ...
97     alrededor de
98         Pumbral_max = Pmax + (Umbral / 2); // potencia maxima
99         Contador2 = Contador2 + 1;
100     }
101     Duty = Dold;
102     Dold = Duty;
103     float mayor = 0;
104     float menor = 260;
105     if (Pnew_av > Pumbral_max || Pnew_av < Pumbral_min) { // si la ...
106     potencia sale del umbral de mpp
107         for (int a = 0; a < TamPotencia; a++) {
108             I = CurrentFunction(250);
109             V = VoltageFunction(250);
110             Pnew = V * I;

```

```

107         Pnew_av = Pnew + Pnew_av;
108         Graficar(Pnew, V, I, Duty);
109     }
110     Pnew_av = Pnew_av / puntos_muestra;
111     if (Pnew_av > mayor) {
112         mayor = Pnew_av;
113     }
114     if (Pnew_av < menor) {
115         menor = Pnew_av;
116     }
117
118     if ((mayor - menor < PotEstab)&&(bandera==false)) { //...
potencia estabilizada
119         L = Duty - 0.1;
120         R = Duty + 0.1;
121         Ld = L;
122         Rd = R;
123         Pold_av = Pnew_av;
124         Dold = Duty;
125         Duty = (Dold + R) / 2;
126         Contador2 = 0;
127         bandera = true;
128     }
129 }
130 }
131 else {
132     //Logica de puntos de prueba de potencia
133     if (Pnew_av > Pold_av) {
134         if (Duty > Dold) {
135             L = Dold;
136             Pold_av = Pnew_av;
137             Dold = Duty;
138             Duty = (Dold + R) / 2;
139         }
140         else {
141             R = Dold;
142             Pold_av = Pnew_av;
143             Dold = Duty;
144             Duty = (Dold + L) / 2;
145         }
146     }
147     else {
148         if (Duty > Dold) {
149             R = Duty;
150             Pold_av = Pold_av;
151             Dold = Dold;
152             Duty = (Dold + L) / 2;
153         }
154         else {
155             L = Duty;
156             Pold_av = Pold_av;
157             Dold = Dold;
158             Duty = (Dold + R) / 2;
159         }
160     }
}

```

```

161     }
162 }
163 //asegurar el ciclo util entre 0 y 1
164 if (Duty >= Dmax) {
165     Duty = Dmax;
166 }
167 else if (Duty <= Dmin) {
168     Duty = Dmin;
169 }
170 Contador = Contador + 1;
171 //////////////////////////////////////////////////////////////////// FIN DEL LOOP ...
172 ////////////////////////////////////////////////////////////////////
173 }
174 //DEFINICION DE FUNCIONES...
175 ////////////////////////////////////////////////////////////////////
176 float VoltageFunction(int n_muestrasV) { //Medici n del voltaje
177     float voltaje = 0;
178     float voltajeIn = 0;
179     float voltajeEntrada;
180     for (int i = 0; i < n_muestrasV; i++) {
181         voltajeEntrada = (analogRead(A1) * (5.0 / 1023.0)); ////// ...
182         inicio bloque de calibraci n
183         if (Duty >= 0.8) {
184             if (voltajeEntrada < 0.086) {
185                 voltaje = voltaje + voltajeEntrada * 57.14; //hasta 2V
186             }
187             else if (voltajeEntrada >= 0.086 && voltajeEntrada < 0.285) {...
188                 //hasta 4V
189                 voltaje = voltaje + voltajeEntrada * 19.58;
190             }
191             else if (voltajeEntrada >= 0.285 && voltajeEntrada < 0.486) {...
192                 //hasta 6V
193                 voltaje = voltaje + voltajeEntrada * 14.35;
194             }
195             else if (voltajeEntrada >= 0.486 && voltajeEntrada < 0.693) {...
196                 //hasta 8V
197                 voltaje = voltaje + voltajeEntrada * 12.77;
198             }
199             else if (voltajeEntrada >= 0.693 && voltajeEntrada < 0.898) {...
200                 //hasta 10V
201                 voltaje = voltaje + voltajeEntrada * 11.94;
202             }
203             else if (voltajeEntrada >= 0.898 && voltajeEntrada < 1.206) {...
204                 //hasta 13V
205                 voltaje = voltaje + voltajeEntrada * 11.43;
206             }
207             else if (voltajeEntrada >= 1.206 && voltajeEntrada < 1.514) {...
208                 //hasta 16V
209                 voltaje = voltaje + voltajeEntrada * 11.03;
210             }
211             else if (voltajeEntrada >= 1.514 && voltajeEntrada < 1.821) {...
212                 //hasta 19V
213                 voltaje = voltaje + voltajeEntrada * 10.79;

```

```

206     }
207     else if (voltajeEntrada >= 1.821 && voltajeEntrada < 2.143) {...
//hasta 22V
208         voltaje = voltaje + voltajeEntrada * 10.64;
209     }
210     else if (voltajeEntrada >= 2.143 && voltajeEntrada < 2.452) {...
//hasta 25V
211         voltaje = voltaje + voltajeEntrada * 10.43;
212     }
213     else if (voltajeEntrada >= 2.452 && voltajeEntrada < 2.759) {...
//hasta 28V
214         voltaje = voltaje + voltajeEntrada * 10.37;
215     }
216     else if (voltajeEntrada >= 2.759) { //mas de 29V
217         voltaje = voltaje + voltajeEntrada * 10.26;
218     }
219 }
220 else if (Duty >= 0.6 && Duty < 0.8) {
221     if (voltajeEntrada < 0.101) {
222         voltaje = voltaje + voltajeEntrada * 40.00; //hasta 2V
223     }
224     else if (voltajeEntrada >= 0.101 && voltajeEntrada < 0.307) {...
//hasta 4V
225         voltaje = voltaje + voltajeEntrada * 17.78;
226     }
227     else if (voltajeEntrada >= 0.307 && voltajeEntrada < 0.511) {...
//hasta 6V
228         voltaje = voltaje + voltajeEntrada * 13.49;
229     }
230     else if (voltajeEntrada >= 0.511 && voltajeEntrada < 0.718) {...
//hasta 8V
231         voltaje = voltaje + voltajeEntrada * 12.24;
232     }
233     else if (voltajeEntrada >= 0.718 && voltajeEntrada < 0.923) {...
//hasta 10V
234         voltaje = voltaje + voltajeEntrada * 11.59;
235     }
236     else if (voltajeEntrada >= 0.923 && voltajeEntrada < 1.233) {...
//hasta 13V
237         voltaje = voltaje + voltajeEntrada * 11.16;
238     }
239     else if (voltajeEntrada >= 1.233 && voltajeEntrada < 1.529) {...
//hasta 16V
240         voltaje = voltaje + voltajeEntrada * 10.84;
241     }
242     else if (voltajeEntrada >= 1.529 && voltajeEntrada < 1.823) {...
//hasta 19V
243         voltaje = voltaje + voltajeEntrada * 10.78;
244     }
245     else if (voltajeEntrada >= 1.823 && voltajeEntrada < 2.132) {...
//hasta 22V
246         voltaje = voltaje + voltajeEntrada * 10.66;
247     }
248     else if (voltajeEntrada >= 2.132 && voltajeEntrada < 2.442) {...
//hasta 25V

```

```

249     voltaje = voltaje + voltajeEntrada * 10.47;
250 }
251 else if (voltajeEntrada >= 2.442 && voltajeEntrada < 2.752) {...
//hasta 28V
252     voltaje = voltaje + voltajeEntrada * 10.39;
253 }
254 else if (voltajeEntrada >= 2.752) { //mas de 29V
255     voltaje = voltaje + voltajeEntrada * 10.30;
256 }
257 }
258 else if (Duty >= 0.4 && Duty < 0.6) {
259     if (voltajeEntrada < 0.136) {
260         voltaje = voltaje + voltajeEntrada * 22.73; //hasta 2V
261     }
262     else if (voltajeEntrada >= 0.136 && voltajeEntrada < 0.330) {...
//hasta 4V
263         voltaje = voltaje + voltajeEntrada * 15.27;
264     }
265     else if (voltajeEntrada >= 0.330 && voltajeEntrada < 0.527) {...
//hasta 6V
266         voltaje = voltaje + voltajeEntrada * 12.87;
267     }
268     else if (voltajeEntrada >= 0.527 && voltajeEntrada < 0.725) {...
//hasta 8V
269         voltaje = voltaje + voltajeEntrada * 12.00;
270     }
271     else if (voltajeEntrada >= 0.725 && voltajeEntrada < 0.923) {...
//hasta 10V
272         voltaje = voltaje + voltajeEntrada * 11.53;
273     }
274     else if (voltajeEntrada >= 0.923 && voltajeEntrada < 1.220) {...
//hasta 13V
275         voltaje = voltaje + voltajeEntrada * 11.21;
276     }
277     else if (voltajeEntrada >= 1.220 && voltajeEntrada < 1.519) {...
//hasta 16V
278         voltaje = voltaje + voltajeEntrada * 10.96;
279     }
280     else if (voltajeEntrada >= 1.519 && voltajeEntrada < 1.815) {...
//hasta 19V
281         voltaje = voltaje + voltajeEntrada * 10.79;
282     }
283     else if (voltajeEntrada >= 1.815 && voltajeEntrada < 2.118) {...
//hasta 22V
284         voltaje = voltaje + voltajeEntrada * 10.70;
285     }
286     else if (voltajeEntrada >= 2.118 && voltajeEntrada < 2.421) {...
//hasta 25V
287         voltaje = voltaje + voltajeEntrada * 10.58;
288     }
289     else if (voltajeEntrada >= 2.421 && voltajeEntrada < 2.729) {...
//hasta 28V
290         voltaje = voltaje + voltajeEntrada * 10.48;
291     }
292     else if (voltajeEntrada >= 2.729) { //mas de 29V

```

```

293     voltaje = voltaje + voltajeEntrada * 10.41;
294 }
295 }
296 else if (Duty >= 0.2 && Duty < 0.4) {
297     if (voltajeEntrada < 0.140) {
298         voltaje = voltaje + voltajeEntrada * 22.22; //hasta 2V
299     }
300     else if (voltajeEntrada >= 0.140 && voltajeEntrada < 0.339) {...
//hasta 4V
301         voltaje = voltaje + voltajeEntrada * 14.86;
302     }
303     else if (voltajeEntrada >= 0.339 && voltajeEntrada < 0.540) {...
//hasta 6V
304         voltaje = voltaje + voltajeEntrada * 12.59;
305     }
306     else if (voltajeEntrada >= 0.540 && voltajeEntrada < 0.735) {...
//hasta 8V
307         voltaje = voltaje + voltajeEntrada * 11.76;
308     }
309     else if (voltajeEntrada >= 0.735 && voltajeEntrada < 0.933) {...
//hasta 10V
310         voltaje = voltaje + voltajeEntrada * 11.40;
311     }
312     else if (voltajeEntrada >= 0.933 && voltajeEntrada < 1.233) {...
//hasta 13V
313         voltaje = voltaje + voltajeEntrada * 11.08;
314     }
315     else if (voltajeEntrada >= 1.233 && voltajeEntrada < 1.536) {...
//hasta 16V
316         voltaje = voltaje + voltajeEntrada * 10.85;
317     }
318     else if (voltajeEntrada >= 1.536 && voltajeEntrada < 1.838) {...
//hasta 19V
319         voltaje = voltaje + voltajeEntrada * 10.67;
320     }
321     else if (voltajeEntrada >= 1.838 && voltajeEntrada < 2.141) {...
//hasta 22V
322         voltaje = voltaje + voltajeEntrada * 10.55;
323     }
324     else if (voltajeEntrada >= 2.141 && voltajeEntrada < 2.439) {...
//hasta 25V
325         voltaje = voltaje + voltajeEntrada * 10.48;
326     }
327     else if (voltajeEntrada >= 2.439 && voltajeEntrada < 2.748) {...
//hasta 28V
328         voltaje = voltaje + voltajeEntrada * 10.40;
329     }
330     else if (voltajeEntrada >= 2.748) { //mas de 29V
331         voltaje = voltaje + voltajeEntrada * 10.36;
332     }
333 }
334 else if (Duty < 0.2) {
335     if (voltajeEntrada < 0.143) {
336         voltaje = voltaje + voltajeEntrada * 21.51; //hasta 2V
337     }

```

```

338     else if (voltajeEntrada >= 0.143 && voltajeEntrada < 0.344) {...
//hasta 4V
339         voltaje = voltaje + voltajeEntrada * 14.60;
340     }
341     else if (voltajeEntrada >= 0.344 && voltajeEntrada < 0.543) {...
//hasta 6V
342         voltaje = voltaje + voltajeEntrada * 12.44;
343     }
344     else if (voltajeEntrada >= 0.543 && voltajeEntrada < 0.741) {...
//hasta 8V
345         voltaje = voltaje + voltajeEntrada * 11.68;
346     }
347     else if (voltajeEntrada >= 0.741 && voltajeEntrada < 0.942) {...
//hasta 10V
348         voltaje = voltaje + voltajeEntrada * 11.30;
349     }
350     else if (voltajeEntrada >= 0.942 && voltajeEntrada < 1.246) {...
//hasta 13V
351         voltaje = voltaje + voltajeEntrada * 10.97;
352     }
353     else if (voltajeEntrada >= 1.246 && voltajeEntrada < 1.549) {...
//hasta 16V
354         voltaje = voltaje + voltajeEntrada * 10.74;
355     }
356     else if (voltajeEntrada >= 1.549 && voltajeEntrada < 1.851) {...
//hasta 19V
357         voltaje = voltaje + voltajeEntrada * 10.58;
358     }
359     else if (voltajeEntrada >= 1.851 && voltajeEntrada < 2.155) {...
//hasta 22V
360         voltaje = voltaje + voltajeEntrada * 10.48;
361     }
362     else if (voltajeEntrada >= 2.155 && voltajeEntrada < 2.461) {...
//hasta 25V
363         voltaje = voltaje + voltajeEntrada * 10.41;
364     }
365     else if (voltajeEntrada >= 2.461 && voltajeEntrada < 2.760) {...
//hasta 28V
366         voltaje = voltaje + voltajeEntrada * 10.33;
367     }
368     else if (voltajeEntrada >= 2.760) { //mas de 29V
369         voltaje = voltaje + voltajeEntrada * 10.31;
370     }
371 }
372 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////fin ...
bloque de calibraci n
373     voltajeIn = voltajeIn + voltajeEntrada; //solo es para ...
calibracion
374 }
375     voltaje = voltaje / n_muestrasV;
376     voltajeIn = voltajeIn / n_muestrasV; //solo es para calibracion
377     return (voltaje);
378 }
379
380 float CurrentFunction(int n_muestras) {

```

```

381 float voltaje = 0;
382 float Corriente = 0;
383 float constante = 2.488; //voltaje cuando I=0;
384 for (int i = 0; i < n_muestras; i++) {
385     voltaje = analogRead(A0) * (5.0 / 1023.0); ///////////////inicio ...
    bloque de calibraci n
386     if (voltaje < 2.55) {
387         Corriente = Corriente + (voltaje - constante) / 0.0970; //0A
388     }
389     else if (voltaje >= 2.55 && voltaje < 2.65) {
390         Corriente = Corriente + (voltaje - constante) / 0.0970; //1A
391     }
392     else if (voltaje >= 2.65 && voltaje < 2.74) {
393         Corriente = Corriente + (voltaje - constante) / 0.0970; //2A
394     }
395     else if (voltaje >= 2.74 && voltaje < 2.84) {
396         Corriente = Corriente + (voltaje - constante) / 0.0977; //3A
397     }
398     else if (voltaje >= 2.84 && voltaje < 2.95) {
399         Corriente = Corriente + (voltaje - constante) / 0.0980; //4A
400     }
401     else if (voltaje >= 2.95 && voltaje < 3.04) {
402         Corriente = Corriente + (voltaje - constante) / 0.0980; //5A
403     }
404     else if (voltaje >= 3.04 && voltaje < 3.14) {
405         Corriente = Corriente + (voltaje - constante) / 0.0983; //6A
406     }
407     else if (voltaje >= 3.14 && voltaje < 3.24) {
408         Corriente = Corriente + (voltaje - constante) / 0.0984; //7A
409     }
410     else if (voltaje >= 3.24 && voltaje < 3.34) {
411         Corriente = Corriente + (voltaje - constante) / 0.0985; //8A
412     }
413     else if (voltaje >= 3.34) {
414         Corriente = Corriente + (voltaje - constante) / 0.0985; //9A
415     }
416     ////////////////////////////////////////////////////////////////////fin bloque de calibraci n
417 }
418 Corriente = Corriente / n_muestras;
419 if (Corriente < 0) {
420     Corriente = 0;
421 }
422 return (Corriente);
423 }
424
425 int digitalPotWrite(int value) {
426     digitalWrite(CS, LOW);
427     SPI.transfer(address);
428     SPI.transfer(value);
429     digitalWrite(CS, HIGH);
430 }
431 int Graficar(float Y1, float Y2, float Y3, float Y4) {
432     Serial.print(Y1, 3);
433     Serial.print(", ");
434     Serial.print(Y2, 3);

```

```

435 Serial.print(", ");
436 Serial.print(Y3, 3);
437 Serial.print(", ");
438 Serial.println(Y4, 3);
439 }

```

Código 5: Código de algoritmo SH.

## 6.6. Código de implementación de algoritmo P&O

```

1 #include <SPI.h>
2
3 byte address = 0x00;
4 int CS = 10;
5 int sensorPinV = A1; // Pin para el voltaje
6 int sensorPinC = A0; // Pin para la corriente
7 int puntos_muestra = 10;
8
9 //Condiciones iniciales del algoritmo SH y definicion de variables
10 float L = 0;
11 float R = 1;
12 float Dmax = 1;
13 float Dmin = 0;
14 bool CorreSH = true;
15 int Contador = 0;
16 float V;
17 float I;
18 float Pold = 0;
19 float Vold = 0;
20 float Pnew;
21 float Duty = 1;
22
23 void setup() {
24   Serial.begin(9600);
25   pinMode(CS, OUTPUT);
26   SPI.begin();
27   digitalPotWrite(128);
28   delay(5000);
29   for (int h = 0; h < puntos_muestra; h++) {
30     Graficar(0.0, 0.0, 0.0, 1);
31   }
32 }
33 void loop() {
34   digitalPotWrite(round(128 * (1 - Duty)));
35   delay(100);
36   float Pot_av = 0;
37   float V_av = 0;
38   for (int x = 0; x < puntos_muestra; x++) {
39     I = CurrentFunction(250);
40     V = VoltageFunction(250);
41     Pnew = V * I;
42     Pot_av = Pot_av + Pnew;
43     V_av = V_av + V;

```

```

44     Graficar(Pnew, V, I, Duty);
45 }
46 Pot_av = Pot_av / puntos_muestra;
47 V_av = V_av / puntos_muestra;
48
49 if (Pot_av >= Pold) {
50     if (V_av >= Vold) {
51         Duty = Duty - 0.008;
52     }
53     else {
54         Duty = Duty + 0.008;
55     }
56 }
57 else {
58     if (V_av >= Vold) {
59         Duty = Duty + 0.008;
60     }
61     else {
62         Duty = Duty - 0.008;
63     }
64 }
65
66 //asegurar el ciclo util entre 0 y 1
67 if (Duty >= Dmax) {
68     Duty = Dmax;
69 }
70 else if (Duty <= Dmin) {
71     Duty = Dmin;
72 }
73
74 Contador = Contador + 1;
75 Pold = Pot_av;
76 Vold = V_av;
77
78 ////////////////////////////////////////////////// FIN DEL LOOP ...
79 //////////////////////////////////////////////////
80 }
81 //DEFINICION DE FUNCIONES...
82 //////////////////////////////////////////////////
83 float VoltageFunction(int n_muestrasV) { //Medici n del voltaje
84     float voltaje = 0;
85     float voltajeIn = 0;
86     float voltajeEntrada;
87     for (int i = 0; i < n_muestrasV; i++) {
88         voltajeEntrada = (analogRead(A1) * (5.0 / 1023.0)); //...
89         //////////////////////////////////
90         if (Duty >= 0.8) {
91             if (voltajeEntrada < 0.086) {
92                 voltaje = voltaje + voltajeEntrada * 57.14; //hasta 2V
93             }
94             else if (voltajeEntrada >= 0.086 && voltajeEntrada < 0.285) {...
95                 //hasta 4V
96                 voltaje = voltaje + voltajeEntrada * 19.58;

```

```

95     }
96     else if (voltajeEntrada >= 0.285 && voltajeEntrada < 0.486) {...
//hasta 6V
97         voltaje = voltaje + voltajeEntrada * 14.35;
98     }
99     else if (voltajeEntrada >= 0.486 && voltajeEntrada < 0.693) {...
//hasta 8V
100        voltaje = voltaje + voltajeEntrada * 12.77;
101    }
102    else if (voltajeEntrada >= 0.693 && voltajeEntrada < 0.898) {...
//hasta 10V
103        voltaje = voltaje + voltajeEntrada * 11.94;
104    }
105    else if (voltajeEntrada >= 0.898 && voltajeEntrada < 1.206) {...
//hasta 13V
106        voltaje = voltaje + voltajeEntrada * 11.43;
107    }
108    else if (voltajeEntrada >= 1.206 && voltajeEntrada < 1.514) {...
//hasta 16V
109        voltaje = voltaje + voltajeEntrada * 11.03;
110    }
111    else if (voltajeEntrada >= 1.514 && voltajeEntrada < 1.821) {...
//hasta 19V
112        voltaje = voltaje + voltajeEntrada * 10.79;
113    }
114    else if (voltajeEntrada >= 1.821 && voltajeEntrada < 2.143) {...
//hasta 22V
115        voltaje = voltaje + voltajeEntrada * 10.64;
116    }
117    else if (voltajeEntrada >= 2.143 && voltajeEntrada < 2.452) {...
//hasta 25V
118        voltaje = voltaje + voltajeEntrada * 10.43;
119    }
120    else if (voltajeEntrada >= 2.452 && voltajeEntrada < 2.759) {...
//hasta 28V
121        voltaje = voltaje + voltajeEntrada * 10.37;
122    }
123    else if (voltajeEntrada >= 2.759) { //mas de 29V
124        voltaje = voltaje + voltajeEntrada * 10.26;
125    }
126 }
127 else if (Duty >= 0.6 && Duty < 0.8) {
128     if (voltajeEntrada < 0.101) {
129         voltaje = voltaje + voltajeEntrada * 40.00; //hasta 2V
130     }
131     else if (voltajeEntrada >= 0.101 && voltajeEntrada < 0.307) {...
//hasta 4V
132         voltaje = voltaje + voltajeEntrada * 17.78;
133     }
134     else if (voltajeEntrada >= 0.307 && voltajeEntrada < 0.511) {...
//hasta 6V
135         voltaje = voltaje + voltajeEntrada * 13.49;
136     }
137     else if (voltajeEntrada >= 0.511 && voltajeEntrada < 0.718) {...
//hasta 8V

```

```

138     voltaje = voltaje + voltajeEntrada * 12.24;
139 }
140 else if (voltajeEntrada >= 0.718 && voltajeEntrada < 0.923) {...
//hasta 10V
141     voltaje = voltaje + voltajeEntrada * 11.59;
142 }
143 else if (voltajeEntrada >= 0.923 && voltajeEntrada < 1.233) {...
//hasta 13V
144     voltaje = voltaje + voltajeEntrada * 11.16;
145 }
146 else if (voltajeEntrada >= 1.233 && voltajeEntrada < 1.529) {...
//hasta 16V
147     voltaje = voltaje + voltajeEntrada * 10.84;
148 }
149 else if (voltajeEntrada >= 1.529 && voltajeEntrada < 1.823) {...
//hasta 19V
150     voltaje = voltaje + voltajeEntrada * 10.78;
151 }
152 else if (voltajeEntrada >= 1.823 && voltajeEntrada < 2.132) {...
//hasta 22V
153     voltaje = voltaje + voltajeEntrada * 10.66;
154 }
155 else if (voltajeEntrada >= 2.132 && voltajeEntrada < 2.442) {...
//hasta 25V
156     voltaje = voltaje + voltajeEntrada * 10.47;
157 }
158 else if (voltajeEntrada >= 2.442 && voltajeEntrada < 2.752) {...
//hasta 28V
159     voltaje = voltaje + voltajeEntrada * 10.39;
160 }
161 else if (voltajeEntrada >= 2.752) { //mas de 29V
162     voltaje = voltaje + voltajeEntrada * 10.30;
163 }
164 }
165 else if (Duty >= 0.4 && Duty < 0.6) {
166     if (voltajeEntrada < 0.136) {
167         voltaje = voltaje + voltajeEntrada * 22.73; //hasta 2V
168     }
169     else if (voltajeEntrada >= 0.136 && voltajeEntrada < 0.330) {...
//hasta 4V
170         voltaje = voltaje + voltajeEntrada * 15.27;
171     }
172     else if (voltajeEntrada >= 0.330 && voltajeEntrada < 0.527) {...
//hasta 6V
173         voltaje = voltaje + voltajeEntrada * 12.87;
174     }
175     else if (voltajeEntrada >= 0.527 && voltajeEntrada < 0.725) {...
//hasta 8V
176         voltaje = voltaje + voltajeEntrada * 12.00;
177     }
178     else if (voltajeEntrada >= 0.725 && voltajeEntrada < 0.923) {...
//hasta 10V
179         voltaje = voltaje + voltajeEntrada * 11.53;
180     }

```

```

181     else if (voltajeEntrada >= 0.923 && voltajeEntrada < 1.220) {...
//hasta 13V
182         voltaje = voltaje + voltajeEntrada * 11.21;
183     }
184     else if (voltajeEntrada >= 1.220 && voltajeEntrada < 1.519) {...
//hasta 16V
185         voltaje = voltaje + voltajeEntrada * 10.96;
186     }
187     else if (voltajeEntrada >= 1.519 && voltajeEntrada < 1.815) {...
//hasta 19V
188         voltaje = voltaje + voltajeEntrada * 10.79;
189     }
190     else if (voltajeEntrada >= 1.815 && voltajeEntrada < 2.118) {...
//hasta 22V
191         voltaje = voltaje + voltajeEntrada * 10.70;
192     }
193     else if (voltajeEntrada >= 2.118 && voltajeEntrada < 2.421) {...
//hasta 25V
194         voltaje = voltaje + voltajeEntrada * 10.58;
195     }
196     else if (voltajeEntrada >= 2.421 && voltajeEntrada < 2.729) {...
//hasta 28V
197         voltaje = voltaje + voltajeEntrada * 10.48;
198     }
199     else if (voltajeEntrada >= 2.729) { //mas de 29V
200         voltaje = voltaje + voltajeEntrada * 10.41;
201     }
202 }
203 else if (Duty >= 0.2 && Duty < 0.4) {
204     if (voltajeEntrada < 0.140) {
205         voltaje = voltaje + voltajeEntrada * 22.22; //hasta 2V
206     }
207     else if (voltajeEntrada >= 0.140 && voltajeEntrada < 0.339) {...
//hasta 4V
208         voltaje = voltaje + voltajeEntrada * 14.86;
209     }
210     else if (voltajeEntrada >= 0.339 && voltajeEntrada < 0.540) {...
//hasta 6V
211         voltaje = voltaje + voltajeEntrada * 12.59;
212     }
213     else if (voltajeEntrada >= 0.540 && voltajeEntrada < 0.735) {...
//hasta 8V
214         voltaje = voltaje + voltajeEntrada * 11.76;
215     }
216     else if (voltajeEntrada >= 0.735 && voltajeEntrada < 0.933) {...
//hasta 10V
217         voltaje = voltaje + voltajeEntrada * 11.40;
218     }
219     else if (voltajeEntrada >= 0.933 && voltajeEntrada < 1.233) {...
//hasta 13V
220         voltaje = voltaje + voltajeEntrada * 11.08;
221     }
222     else if (voltajeEntrada >= 1.233 && voltajeEntrada < 1.536) {...
//hasta 16V
223         voltaje = voltaje + voltajeEntrada * 10.85;

```

```

224     }
225     else if (voltajeEntrada >= 1.536 && voltajeEntrada < 1.838) {...
//hasta 19V
226         voltaje = voltaje + voltajeEntrada * 10.67;
227     }
228     else if (voltajeEntrada >= 1.838 && voltajeEntrada < 2.141) {...
//hasta 22V
229         voltaje = voltaje + voltajeEntrada * 10.55;
230     }
231     else if (voltajeEntrada >= 2.141 && voltajeEntrada < 2.439) {...
//hasta 25V
232         voltaje = voltaje + voltajeEntrada * 10.48;
233     }
234     else if (voltajeEntrada >= 2.439 && voltajeEntrada < 2.748) {...
//hasta 28V
235         voltaje = voltaje + voltajeEntrada * 10.40;
236     }
237     else if (voltajeEntrada >= 2.748) { //mas de 29V
238         voltaje = voltaje + voltajeEntrada * 10.36;
239     }
240 }
241 else if (Duty < 0.2) {
242     if (voltajeEntrada < 0.143) {
243         voltaje = voltaje + voltajeEntrada * 21.51; //hasta 2V
244     }
245     else if (voltajeEntrada >= 0.143 && voltajeEntrada < 0.344) {...
//hasta 4V
246         voltaje = voltaje + voltajeEntrada * 14.60;
247     }
248     else if (voltajeEntrada >= 0.344 && voltajeEntrada < 0.543) {...
//hasta 6V
249         voltaje = voltaje + voltajeEntrada * 12.44;
250     }
251     else if (voltajeEntrada >= 0.543 && voltajeEntrada < 0.741) {...
//hasta 8V
252         voltaje = voltaje + voltajeEntrada * 11.68;
253     }
254     else if (voltajeEntrada >= 0.741 && voltajeEntrada < 0.942) {...
//hasta 10V
255         voltaje = voltaje + voltajeEntrada * 11.30;
256     }
257     else if (voltajeEntrada >= 0.942 && voltajeEntrada < 1.246) {...
//hasta 13V
258         voltaje = voltaje + voltajeEntrada * 10.97;
259     }
260     else if (voltajeEntrada >= 1.246 && voltajeEntrada < 1.549) {...
//hasta 16V
261         voltaje = voltaje + voltajeEntrada * 10.74;
262     }
263     else if (voltajeEntrada >= 1.549 && voltajeEntrada < 1.851) {...
//hasta 19V
264         voltaje = voltaje + voltajeEntrada * 10.58;
265     }
266     else if (voltajeEntrada >= 1.851 && voltajeEntrada < 2.155) {...
//hasta 22V

```

```

267     voltaje = voltaje + voltajeEntrada * 10.48;
268 }
269 else if (voltajeEntrada >= 2.155 && voltajeEntrada < 2.461) {...
//hasta 25V
270     voltaje = voltaje + voltajeEntrada * 10.41;
271 }
272 else if (voltajeEntrada >= 2.461 && voltajeEntrada < 2.760) {...
//hasta 28V
273     voltaje = voltaje + voltajeEntrada * 10.33;
274 }
275 else if (voltajeEntrada >= 2.760) { //mas de 29V
276     voltaje = voltaje + voltajeEntrada * 10.31;
277 }
278 }
279 //...
////////////////////////////////////
280 voltajeIn = voltajeIn + voltajeEntrada; //solo es para ...
calibracion
281 }
282 voltaje = voltaje / n_muestrasV;
283 voltajeIn = voltajeIn / n_muestrasV; //solo es para calibracion
284 return (voltaje);
285 }
286
287 float CurrentFunction(int n_muestras) {
288     float voltaje = 0;
289     float Corriente = 0;
290     float constante = 2.488; //voltaje cuando I=0;
291     for (int i = 0; i < n_muestras; i++) {
292         voltaje = analogRead(A0) * (5.0 / 1023.0); ///////////////
293         if (voltaje < 2.55) {
294             Corriente = Corriente + (voltaje - constante) / 0.0970; //0A
295         }
296         else if (voltaje >= 2.55 && voltaje < 2.65) {
297             Corriente = Corriente + (voltaje - constante) / 0.0970; //1A
298         }
299         else if (voltaje >= 2.65 && voltaje < 2.74) {
300             Corriente = Corriente + (voltaje - constante) / 0.0970; //2A
301         }
302         else if (voltaje >= 2.74 && voltaje < 2.84) {
303             Corriente = Corriente + (voltaje - constante) / 0.0970; //3A
304         }
305         else if (voltaje >= 2.84 && voltaje < 2.95) {
306             Corriente = Corriente + (voltaje - constante) / 0.0975; //4A
307         }
308         else if (voltaje >= 2.95 && voltaje < 3.04) {
309             Corriente = Corriente + (voltaje - constante) / 0.0980; //5A
310         }
311         else if (voltaje >= 3.04 && voltaje < 3.14) {
312             Corriente = Corriente + (voltaje - constante) / 0.0983; //6A
313         }
314         else if (voltaje >= 3.14 && voltaje < 3.24) {
315             Corriente = Corriente + (voltaje - constante) / 0.0984; //7A
316         }
317         else if (voltaje >= 3.24 && voltaje < 3.34) {

```

```

318     Corriente = Corriente + (voltaje - constante) / 0.0985; //8A
319 }
320 else if (voltaje >= 3.34) {
321     Corriente = Corriente + (voltaje - constante) / 0.0985; //9A
322 }
323 ///////////////////////////////////////////////////////////////////
324 }
325 Corriente = Corriente / n_muestras;
326 if (Corriente < 0) {
327     Corriente = 0;
328 }
329 return (Corriente);
330 }
331
332 int digitalPotWrite(int value) {
333     digitalWrite(CS, LOW);
334     SPI.transfer(address);
335     SPI.transfer(value);
336     digitalWrite(CS, HIGH);
337 }
338 int Graficar(float Y1, float Y2, float Y3, float Y4) {
339     //Serial.println("Potencia , Voltaje , Corriente");
340     Serial.print(Y1, 3);
341     Serial.print(" , ");
342     Serial.print(Y2, 3);
343     Serial.print(" , ");
344     Serial.print(Y3, 3);
345     Serial.print(" , ");
346     Serial.println(Y4, 3);
347 }

```

Código 6: Código de algoritmo P&O.

## 6.7. Diseño de PCB de la fuente de alimentación

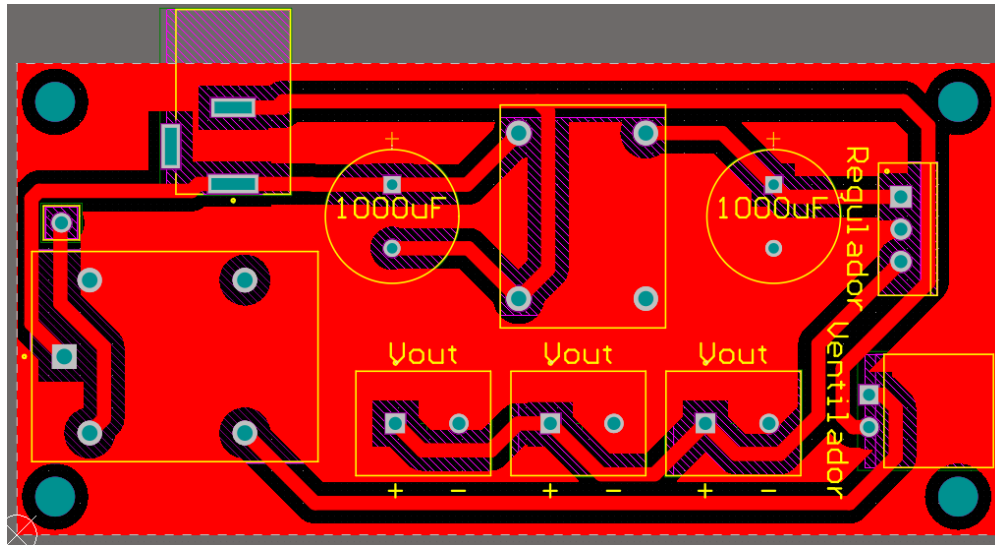


Figura 73: Diseño de PCB de la fuente de alimentación

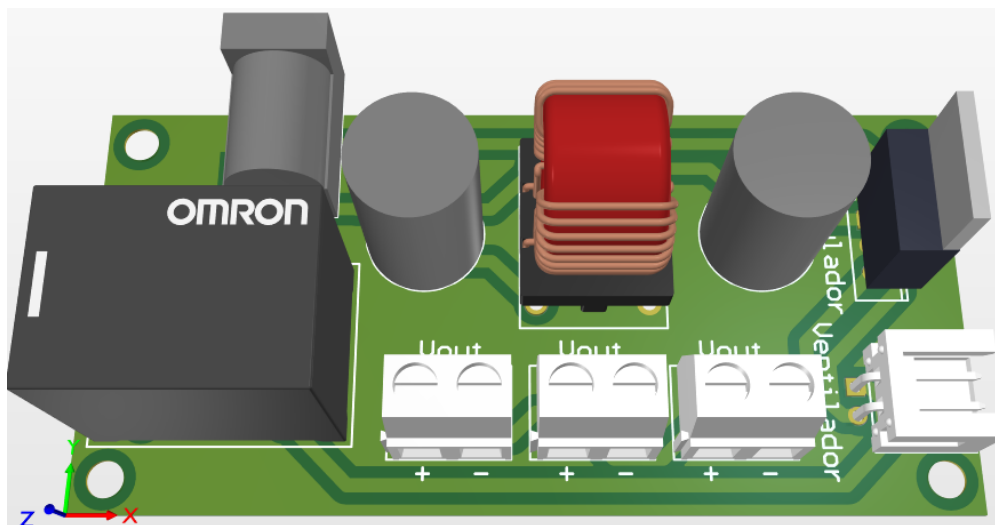


Figura 74: Vista 3D del diseño de PCB de la fuente de alimentación

## 6.8. Código de simulación de algoritmo SH en Matlab Simulink

```
1 function Duty = DutyGen(V,I)
2
3 Dmax = 1; %Limite maximo de D
4 Dmin = 0; %limite minimo de D
5 muestra = 50; %Cantidad de muestras para definir estabilizacion
6 Pot = V*I;
7
```

```

8 persistent L R Dold Dnew Pnew Pold DutyOld Contador MV Bandera;
9
10 if isempty(Bandera)
11     Bandera = 0;
12 end
13
14 if isempty(Dold) || Bandera == 1
15     L = 0.01; %Duty Izquierda
16     R = 0.99; %Duty Derecha
17     Contador = 0;
18     MV = zeros(muestra,1); %Muestra de V
19     Dold = R/4;
20     Dnew = 3*Dold;
21     Duty = Dold;
22     DutyOld = Duty;
23     Pold = 0;
24     Bandera = 0;
25 end
26
27 Contador = Contador + 1;
28 if (abs(Dnew - Dold) < 0.001)
29     Duty = DutyOld;
30     DutyOld = Duty;
31     if (abs(Pold - Pot) > 4)
32         Bandera = 1;
33     end
34 else
35     if (Contador < muestra)
36         MV(Contador) = Pot;
37         Duty = DutyOld;
38     elseif (Contador ≥ muestra)
39         MV(1:muestra-1) = MV(2:muestra);
40         MV(muestra) = Pot;
41         Contador = 0;
42
43         if ((max(MV)-min(MV)) < 1.5) %Delta de potencia para asumir ...
estabilizacion
44             if (DutyOld == 0.99/4)
45                 Pold = V*I;
46                 Dnew = DutyOld*3;
47                 Duty = Dnew;
48                 DutyOld = Duty;
49             else
50                 Pnew = V*I;
51
52                 if (Pnew > Pold)
53                     if (Dnew > Dold)
54                         L = Dold;
55                         Dold = Dnew;
56                         Pold = Pnew;
57                         Dnew = (Dold + R) / 2;
58                         Duty = Dnew;
59                         DutyOld = Duty;
60                     else
61                         R = Dold;

```

```

62         Pold = Pnew;
63         Dold = Dnew;
64         Dnew = (Dold + L) / 2;
65         Duty = Dnew;
66         DutyOld = Duty;
67     end
68 elseif (Pold > Pnew)
69     if (Dnew > Dold)
70         R = Dnew;
71         Pold = Pold;
72         Dold = Dold;
73         Dnew = (Dold + L) / 2;
74         Duty = Dnew;
75         DutyOld = Duty;
76     else
77         L = Dnew;
78         Pold = Pold;
79         Dold = Dold;
80         Dnew = (Dold + R) / 2;
81         Duty = Dnew;
82         DutyOld = Duty;
83     end
84 else
85     Duty = DutyOld;
86     Pold = Pnew;
87     DutyOld = Duty;
88 end
89
90 if (Duty ≥ Dmax)
91     Duty = Dmax;
92     DutyOld = Duty;
93 elseif (Duty ≤ Dmin)
94     Duty = Dmin;
95     DutyOld = Duty;
96 else
97     %No haga nada
98 end
99 end
100 else
101     Duty = DutyOld;
102 end
103 else
104     Duty = DutyOld;
105 end
106 end

```

SH.m

## 6.9. Código de simulación de algoritmo P&O en Matlab Simulink

```

1 function Duty = DutyGen(V, I)
2

```

```

3 Dmax = 1; %Limite maximo de D
4 Dmin = 0; %limite minimo de D
5 muestra = 50; %Cantidad de muestras para definir estabilizacion
6 Pot = V*I;
7 AnchoPaso = 0.0078;
8 DutyInicial = 1;
9
10 persistent Vold Pnew Pold DutyOld Contador MV Bandera;
11
12 if isempty(Bandera)
13     Bandera = 0;
14 end
15
16 if isempty(Vold) || Bandera == 1
17     Contador = 0;
18     MV = zeros(muestra,1); %Muestra de V
19     Vold = 0;
20     Duty = 1;
21     DutyOld = DutyInicial;
22     Pold = 0;
23     Bandera = 0;
24 end
25
26 Contador = Contador + 1;
27
28 if (Contador < muestra)
29     MV(Contador) = Pot;
30     Duty = DutyOld;
31 elseif (Contador ≥ muestra)
32     MV(1:muestra-1) = MV(2:muestra);
33     MV(muestra) = Pot;
34     Contador = 0;
35
36     if ((max(MV)-min(MV)) < 1.5) %Delta de potencia para asumir ...
estabilizacion
37
38         Pnew = V*I;
39
40         if (Pnew > Pold)
41             if (V > Vold)
42                 Duty = DutyOld - AnchoPaso;
43
44             else
45                 Duty = DutyOld + AnchoPaso;
46             end
47         else
48             if (V > Vold)
49                 Duty = DutyOld + AnchoPaso;
50             else
51                 Duty = DutyOld - AnchoPaso;
52             end
53         end
54         Vold = V;
55         DutyOld = Duty;
56         Pold = Pnew;

```

```
57
58     if (Duty ≥ Dmax)
59         Duty = Dmax;
60         DutyOld = Duty;
61     elseif (Duty ≤ Dmin)
62         Duty = Dmin;
63         DutyOld = Duty;
64     else
65         %No haga nada
66     end
67
68     else
69         Duty = DutyOld;
70     end
71 else
72     Duty = DutyOld;
73 end
```

PO.m