

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
DESARROLLO DE UN SIMULADOR DE PANEL FOTOVOLTAICO
MEDIANTE UN CONVERTOR DC-DC**

**AUTORES:
CHRISTIAN ALBERTO JÁCOME TIPÁN
VICTOR GEOVANNY PULLAS RIVADENEIRA**

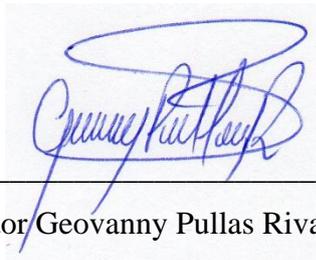
**TUTOR:
JAVIER ARTURO ROJAS URBANO**

Quito, septiembre del 2020

CESIÓN DE DERECHOS DE AUTOR

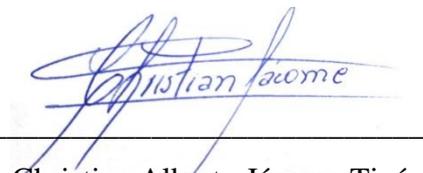
Nosotros, Christian Alberto Jácome Tipán y Victor Geovanny Pullas Rivadeneira con documentos de identificación N°1715273387 y N°1722993282 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “DESARROLLO DE UN SIMULADOR DE PANEL FOTOVOLTAICO MEDIANTE UN CONVERTOR DC-DC”, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Victor Geovanny Pullas Rivadeneira

C.I. 1722993282



Christian Alberto Jácome Tipán

C.I. 1715273387

Fecha: Quito, septiembre 2020.

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, “DESARROLLO DE UN SIMULADOR DE PANEL FOTOVOLTAICO MEDIANTE UN CONVERTOR DC-DC”, realizado por Victor Geovanny Pullas Rivadeneira y Christian Alberto Jácome Tipán, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, septiembre 2020



Javier Arturo Rojas Urbano

C.I. 020189802-0

DEDICATORIA

El trabajo fruto de mi esfuerzo y constancia se lo dedico como mucho amor a mi madre porque ella sembró en mí la semilla del amor, el deseo de triunfar y salir adelante. Te amo y no va haber manera de devolverte tanto que me has ofrecido, es un logro más que llevo a cabo, y sin lugar a dudas ha sido en gran parte gracias ti.

Geovanny Pullas

La culminación de este trabajo representa un sueño alcanzado y para poder lograrlo me ha costado mucho esfuerzo, tenacidad y carácter en los momentos más difíciles para seguir avanzando, esas virtudes y valores se los debo a mi madre a quien se lo dedico con todo mi amor y a mi hermana Lisbeth (†) que fue un ejemplo de perseverancia con una fe inquebrantable siempre imparables en sus metas. Las amo mucho.

Christian Jácome

AGRADECIMIENTO

Quiero expresar las más sinceras palabras de agradecimiento a mi querida madre, por haberme dado un hogar donde crecer, aprender y donde adquirí los valores que definen mi vida. A mi padre gracias por todo el esfuerzo en trabajar para que no me faltara nada y por su amor.

A mis hermanas, aunque la mayoría de las veces parece que estuviéramos en una batalla, hay momentos en los que la guerra cesa y nos unimos para lograr nuestros objetivos, solo un simple GRACIAS...

A mi sobrina que con sus cosas me hace salir de quicio, gracias por ser quién eres, por todo tu amor y entusiasmo para vivir.

A mi novia por su apoyo incondicional, noble. ¡Eres el amor de mi vida!

A mis amigos que en la vida universitaria pasamos tantos buenos momentos, por su apoyo y sentimiento de hermandad.

Geovanny Pullas

Doy gracias a Dios por permitirme vivir este momento tan anhelado y por cuidarme todo el tiempo, por todas las cosas buenas que me han dado felicidad y por todas las cosas malas porque me han hecho más fuerte.

Agradezco el apoyo siempre incondicional de mi madre y mis hermanos que son mi mayor presión y mis sobrinos mi motivación para no desmayar.

A mis tías: María, Inés, Fanny, Irene y a mi tío Alonso por siempre haberme ayudado desde niño y me han apoyado siempre cuando lo necesito

Christian Jácome

ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR.....	i
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	ii
DEDICATORIA	iii
AGRADECIMIENTO.....	iv
ÍNDICE GENERAL.....	v
ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE TABLAS	x
RESUMEN.....	xi
ABSTRACT.....	xii
INTRODUCCIÓN	xiii
CAPÍTULO 1	1
ANTECEDENTES.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2 JUSTIFICACIÓN	1
1.3 OBJETIVOS	2
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos.....	2
CAPÍTULO 2.....	4
MARCO TEÓRICO.....	4
2.1 ENERGÍA SOLAR FOTOVOLTAICA	4
2.1.1 Funcionamiento de una celda fotovoltaica.....	4
2.2 PANEL FOTOVOLTAICO	5
2.2.1 Estructura y componentes	6
2.2.2 Clasificación.....	7
2.2.3 Características eléctricas	7
2.2.4 Aplicaciones.....	8

2.3 MODELO MATEMÁTICO DE UN PANEL FOTOVOLTAICO (FV).....	10
2.4 SIMULADOR DE PANEL FOTOVOLTAICO.....	13
2.4.1 Convertidor AC-DC-DC	15
CAPÍTULO 3	19
DISEÑO DEL SIMULADOR DE PANEL FOTOVOLTAICO	19
3.1 ESQUEMA GENERAL DE LA SOLUCIÓN.....	19
3.2 DIMENSIONAMIENTO DEL SIMULADOR DE PANEL FOTOVOLTAICO	20
3.3 DISEÑO DEL CONVERTOR TRIFÁSICO AC-DC	22
3.4 FILTRADO	23
3.5 REGULADOR DE VOLTAJE	24
3.6 CONTROLADOR PID	26
3.7 IMPLEMENTACIÓN DEL MODELO MATEMÁTICO FV.....	28
3.8 DISEÑO DE LA INTERFAZ GRÁFICA.....	29
3.9 COMUNICACIÓN SERIAL BLUETOOTH	41
3.10 DISEÑO DEL SIMULADOR DE PANEL FOTOVOLTAICO COMPLETO	44
CAPÍTULO 4	46
SIMULACIÓN Y PRUEBAS.....	46
4.1 PRUEBA DEL MODELO MATEMÁTICO FV	46
4.2 PRUEBA DEL CONVERTOR TRIFÁSICO NO CONTROLADO	47
4.3 PRUEBA DEL CONVERTOR REDUCTOR EN LAZO ABIERTO	49
4.3.1 Prueba estática del convertor reductor en lazo abierto	49
4.3.2 Prueba dinámica del convertor reductor en lazo abierto	50
4.4 PRUEBA DEL CONVERTOR REDUCTOR EN LAZO CERRADO	51
4.4.1 Prueba estática del convertor reductor en lazo cerrado	51
4.4.2 Prueba dinámica del convertor reductor en lazo cerrado	52
4.4.3 Prueba del convertor reductor con carga variable	53

4.5 COMPARACIÓN ENTRE LAZO ABIERTO Y CERRADO DE LA PRUEBA ESTÁTICA	54
CONCLUSIONES	56
RECOMENDACIONES	58
REFERENCIAS	59
ANEXOS	

ÍNDICE DE FIGURAS

Figura 2.1. Estructura interna de una celda fotovoltaica.....	5
Figura 2.2. Celdas fotovoltaicas.....	5
Figura 2.3. Estructura de un Panel Fotovoltaico	6
Figura 2.4. Cuadro de clasificación.....	7
Figura 2.5. Sistemas conectados a la red.....	9
Figura 2.6. Componentes de una instalación fotovoltaica aislada	9
Figura 2.7. Circuito equivalente de una celda FV	10
Figura 2.8. Diagrama en bloques FV	14
Figura 2.9. Diagrama en bloques de un convertidor AC/DC/DC	15
Figura 2.10. Convertidores AC/DC/DC.....	16
Figura 2.11. Convertidor reductor.....	16
Figura 2.12. Convertidor elevador	17
Figura 2.13. Topologías de Convertidores DC-DC	18
Figura 3.1. Diagrama en bloques del simulador fotovoltaico	19
Figura 3.2. Conversor AC-DC	22
Figura 3.3. Conversor trifásico y filtrado.....	24
Figura 3.4. Controlador PID.....	26
Figura 3.5. Sintonización con “PID Tunner”	27
Figura 3.6. Algoritmo matemático FV	29
Figura 3.7. Interfaz gráfica del simulador de panel fotovoltaico	31
Figura 3.8. Diagrama de flujo de la interfaz de datos ingresados	32
Figura 3.9. Diagrama de flujo para mostrar de la curva Voltaje vs Irradiancia.....	33
Figura 3.10. Diagrama de flujo para ingresar la irradiancia	34
Figura 3.11. Diagrama de flujo de la interfaz y variables previas a la simulación	35
Figura 3.12. Diagrama de flujo de irradiancia fija	37
Figura 3.13. Diagrama de flujo de la simulación de un panel en irradiancia variable	39
Figura 3.14. Diagrama de flujo de la simulación de un arreglo en irradiancia variable	40
Figura 3.15. APP “Instrument Control”	41
Figura 3.16. Algoritmo de comunicación en la ESP-32.....	42
Figura 3.17. Interrupciones implementadas en la ESP-WROOM-32	43
Figura 3.18. Simulador de panel fotovoltaico diseñado.....	44

Figura 4.1. Simulación conversor trifásico no controlado	47
Figura 4.2. Simulación conversor trifásico no controlado con filtrado.....	48
Figura 4.3. Señales de salida del conversor trifásico no controlado y filtrado	48
Figura 4.4. Diagrama del conversor reductor en lazo abierto	49
Figura 4.5. Respuesta estática del sistema en lazo abierto.....	50
Figura 4.6. Respuesta dinámica del sistema en lazo abierto	50
Figura 4.7. Diagrama del conversor reductor en lazo cerrado	51
Figura 4.8. Respuesta estática del sistema en lazo cerrado.....	52
Figura 4.9. Respuesta dinámica del sistema en lazo cerrado	52
Figura 4.10. Diagrama del conversor reductor en lazo cerrado con carga variable...	53
Figura 4.11. Respuesta del sistema en lazo cerrado frente a cambios de carga.....	54
Figura 4.12. Comparación de respuesta de los sistemas en lazo abierto y cerrado....	55

ÍNDICE DE TABLAS

Tabla 2.1. Cuadro de modelos matemáticos	13
Tabla 3.1. Potencia de consumo de una casa promedio en Quito	21
Tabla 4.1. Parámetros de las curvas de respuesta del sistema lazo abierto y cerrado	54

RESUMEN

El estudio y evaluación de etapas en sistemas fotovoltaicos se dificulta por las condiciones meteorológicas cambiantes, no permitiendo la planificación del trabajo, también se necesita amplios espacios para instalar los paneles y costos elevados de los mismos.

El desarrollo empieza con el estudio del modelo matemático de un panel fotovoltaico que se implementa junto con una interfaz gráfica (HMI) que permite ingresar los valores de los parámetros que intervienen en el modelo matemático, grafica en tiempo real las curvas de Voltaje vs Irradiancia, Corriente vs Voltaje y Potencia vs Voltaje, muestra el punto de trabajo de un convertidor DC-DC reductor.

Se diseñó un convertidor trifásico no controlado de onda completa de corriente alterna (AC) a corriente continua (DC), mediante un puente rectificador trifásico y un filtro capacitivo permitiendo la alimentación al convertidor reductor.

También se diseñó un convertidor reductor que ajusta el voltaje de salida según el ciclo de trabajo que recibe el conmutador desde el controlador (PID) en lazo cerrado y este a su vez toma el dato de voltaje de referencia calculado por el modelo matemático mediante comunicación bluetooth que llega desde el HMI.

El proyecto desarrollado sirve para la planificación de trabajos de investigación de sistemas fotovoltaicos, sin la dependencia de las condiciones meteorológicas, facilitando el estudio dentro de un laboratorio, sin la necesidad de contar con los paneles físicos y áreas amplias que normalmente se usan para su instalación.

ABSTRACT

The study and evaluation of stages in photovoltaic systems is made difficult by the changing meteorological conditions, not allowing the planning of the work, large spaces are also needed to install the panels and high costs thereof.

The development begins with the study of the mathematical model of a photovoltaic panel that is implemented together with a graphical interface (HMI) that allows entering the values of the parameters that intervene in the mathematical model, graphs in real time the curves of Voltage vs Irradiance, Current vs Voltage and Power vs Voltage, shows the duty point of a step-down DC-DC converter.

Not-controlled three-phase full-wave alternating current (AC) to direct current (DC) converter was designed, using a three-phase rectifier bridge and a capacitive filter allowing power to the step-down converter.

A step-down converter was also designed that adjusts the output voltage according to the duty cycle that the switch receives from the controller (PID) in closed loop and this in turn takes the reference voltage data calculated by the mathematical model through Bluetooth communication. coming from the HMI.

The developed project serves for the planning of research work on photovoltaic systems, without dependence on meteorological conditions, facilitating the study within a laboratory, without the need for physical panels and wide areas that are normally used for their installation.

INTRODUCCIÓN

El presente proyecto de titulación consiste en el diseño e implementación de un simulador de paneles fotovoltaicos mediante un convertidor reductor DC-DC, destinado para el estudio de sistemas fotovoltaicos frente a cambios de irradiancia. El simulador es capaz de graficar las curvas de Corriente vs Voltaje, Potencia vs Voltaje, Voltaje vs Irradiancia y el punto de trabajo del convertidor reductor.

La simulación realizada consta de un convertidor de voltaje de corriente alterna trifásica a directa (AC/DC), un conversor reductor por fuente de voltaje (DC/DC) y un controlador en lazo cerrado para el control de voltaje cuyo valor de voltaje de referencia es recibido por medio de bluetooth desde la interfaz gráfica donde se calcula a partir del modelo matemático del panel solar y los parámetros ingresados por el usuario.

En el capítulo 1 se presentan los antecedentes del proyecto, el problema a solucionar y los objetivos a cumplirse en su desarrollo.

El capítulo 2 muestra los aspectos teóricos, conceptuales y técnicos que se investigaron para el desarrollo del proyecto técnico y poder definir la estructura del proyecto, así como su funcionamiento y limitaciones.

El capítulo 3 se enfoca en el diseño y simulación de cada una de las etapas del simulador de panel fotovoltaico dimensionando cada uno de sus componentes, así como los requerimientos para su interconexión.

En el capítulo 4 se presentan las pruebas de simulación realizadas a cada una de las etapas del proyecto, así como a su funcionamiento interconectado, mostrando los valores obtenidos, esquemas de simulación utilizados y análisis de gráficas del simulador fotovoltaico.

El capítulo 5 contiene las conclusiones que se pueden deducir de la investigación realizada, así como del desarrollo del proyecto y algunas recomendaciones que podrían ayudar a obtener mejores resultados en futuras implementaciones del proyecto.

CAPÍTULO 1

ANTECEDENTES

1.1 PLANTEAMIENTO DEL PROBLEMA

Para aprovechar la energía fotovoltaica producida por paneles solares es necesario trabajar en técnicas de transferencia de potencia y algoritmos que permitan obtener su mayor rendimiento posible. La variación de la radiación solar representa un inconveniente para el desarrollo y evaluación de estas técnicas porque dificultan elaborar una planificación de actividades de trabajo o no permiten su cumplimiento puesto que no existen las condiciones de experimentación uniformes o favorables de radiación. (Fărcaș, Ciocan, & Tulbure, 2018; Macaulay, Lin, & Zhou, 2018)

También es necesario invertir una gran cantidad de dinero, ya que dependiendo de su potencia un panel solar puede costar en el Ecuador entre 40,00 USD y 330,00 USD aproximadamente para bajas y altas potencias respectivamente. Así mientras mayor número de paneles solares se requiera en un arreglo fotovoltaico mayor es la cantidad de dinero a invertir. (ProViento S.A., 2019)

El uso de paneles solares requiere de espacio físico para su instalación, si se toma en cuenta que un panel de alta potencia puede medir 1956×992×40 mm para un arreglo fotovoltaico se requiere amplias áreas. Aun cuando se utilicen paneles solares de baja potencia cuyas dimensiones son reducidas (ej. 350x360x17 mm) siempre es necesario para su instalación contar con áreas apropiadas para la recepción de la radiación solar complicando el trabajo en laboratorio. (ProViento S.A., 2019)

1.2 JUSTIFICACIÓN

Disponer de un ambiente de pruebas controlado y compacto para las etapas de potencia y control que intervienen en los sistemas de energización fotovoltaicos facilita su desarrollo, modificación y evaluación ya que se puede trabajar en laboratorio y se elimina la necesidad de disponer de los paneles, así como del espacio físico que estos ocupan, lo cual es considerable cuando se requiere de arreglos conectados en serie y paralelo. Eliminar la dependencia de las condiciones meteorológicas permitirá que se realicen pruebas a una o varias etapas de potencia y control en las mismas condiciones

de experimentación permitiendo su validación y comparación, además se puede simular la variación de la radiación solar de todo un día en un tiempo establecido por el usuario, disminuyendo así los tiempos de prueba.

1.3 OBJETIVOS

1.3.1 Objetivo General

- Desarrollar un convertidor DC-DC para la simulación de paneles fotovoltaicos.

1.3.2 Objetivos Específicos

- Investigar las variables que intervienen en la transformación de energía solar en eléctrica en paneles fotovoltaicos para la obtención de un modelo matemático de su comportamiento.
- Evaluar las diferentes topologías de convertidores DC-DC para la selección del tipo más conveniente en la simulación de paneles fotovoltaicos.
- Diseñar un convertidor DC-DC y su controlador para la simulación del comportamiento de paneles fotovoltaicos ante la variación de la radiación solar.
- Diseñar un controlador para la regulación del voltaje entregado por el convertidor DC-DC ante variaciones de carga.
- Simular el convertidor DC-DC con sus dos lazos de control para la verificación de la generación de potencia variable de acuerdo al modelo del panel solar.
- Implementar una interfaz hombre-máquina (HMI) para el ingreso y visualización de las variables que intervienen en el modelo matemático del panel fotovoltaico.

- Realizar pruebas de funcionamiento mediante simulación para la validación del convertidor y el HMI en base a las curvas de Voltaje vs Radiación solar disponibles en las hojas de datos.

CAPÍTULO 2

MARCO TEÓRICO

Este capítulo contiene los aspectos teóricos y conceptuales utilizados para el desarrollo del proyecto técnico.

2.1 ENERGÍA SOLAR FOTOVOLTAICA

Es la energía eléctrica que se puede obtener a partir del efecto fotoeléctrico y fotovoltaico, que se producen al exponer dos placas, de material semiconductor en contacto, a la luz solar, de manera que los fotones inciden sobre las placas, especialmente sobre el material tipo N, expulsando electrones hacia el material tipo P, generando una corriente eléctrica. (Palaguachi & Guiracocha, 2017)

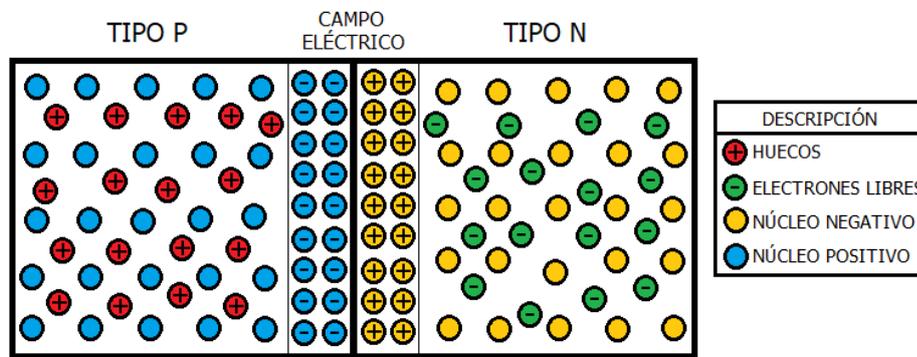
El estudio de esta forma de energía tiene sus inicios con el físico Alexandre Edmond Becquerel, quien al experimentar con baterías expuestas a la luz solar observó un incremento en su energía deduciendo así el efecto fotovoltaico. El primer prototipo para el aprovechamiento de esta energía fue desarrollado por el inventor estadounidense Charles Fritts, uniendo dos placas, una de selenio y una fina capa de oro, que al estar expuestas a la luz solar producían un flujo de electrones entre ellas, dando como resultado una celda que genera energía eléctrica a partir de la energía solar, la celda fotovoltaica. Con el pasar de los años los estudios se concentraron en mejorar el rendimiento de las celdas fotovoltaicas introduciendo el uso de las uniones P-N de materiales semiconductores. (Palaguachi & Guiracocha, 2017)

2.1.1 Funcionamiento de una celda fotovoltaica

Una celda fotovoltaica (FV) está formada por cristales de silicio (Si) y unión de materiales semiconductores tipo P y tipo N, al momento que incide la irradiación solar en la capa tipo N los fotones chocan con los electrones de la capa de valencia del átomo de silicio, si estos poseen la energía necesaria (1.12 [eV]), pueden liberar un electrón de esa capa formando carga positiva y a su vez dejando un hueco, produciendo una reacción en cadena que libera electrones y deja huecos en la unión PN, forzando los huecos hacia el material tipo P y los electrones hacia al tipo N, formando dos capas similares a un diodo con un ánodo (positivo) y cátodo (negativo) respectivamente.

Cuando se conecta una carga en terminales se determina la dirección del flujo de corriente eléctrica generada. En la Figura 2.1 se puede observar la estructura interna de una celda fotovoltaica. (Palaguachi & Guiracocha, 2017)

Figura 2.1. Estructura interna de una celda fotovoltaica

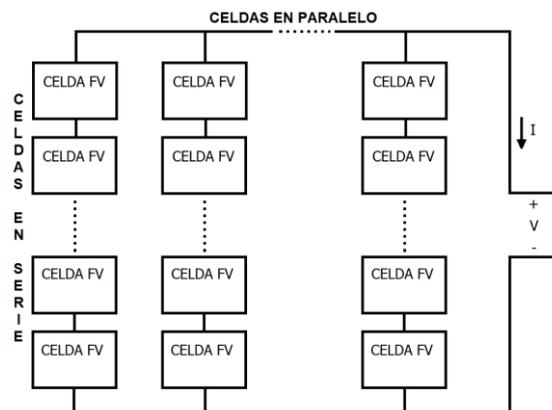


Configuración electrónica de una celda fotovoltaica. Elaborado por: Geovanny Pullas & Christian Jácome

2.2 PANEL FOTOVOLTAICO

Un panel fotovoltaico es un conjunto de celdas FV conectadas en serie y paralelo con el objetivo de obtener en sus terminales distintos niveles de voltaje como se puede observar en la Figura 2.2.

Figura 2.2. Celdas fotovoltaicas



Conexión de celdas dentro de un panel fotovoltaico. Elaborado por: Geovanny Pullas & Christian Jácome

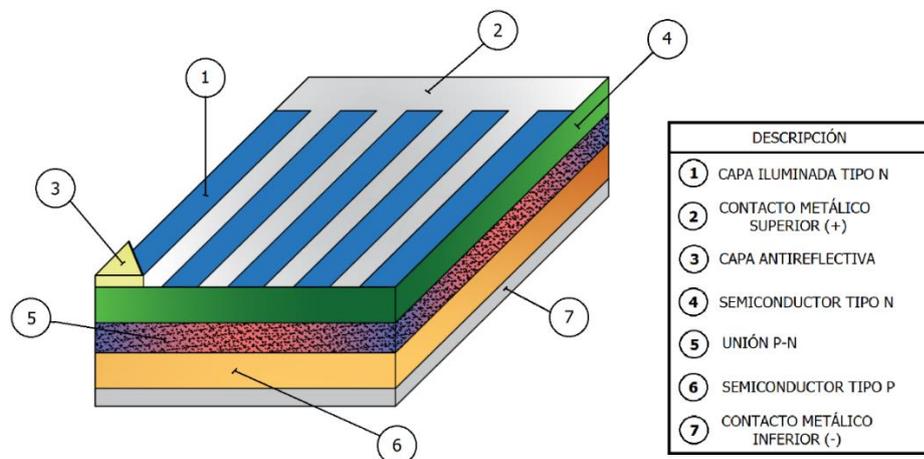
Un panel fotovoltaico tiene una lámina de vidrio en la parte frontal para protección ante condiciones climáticas adversas. Todo el arreglo es colocado en un marco metálico de aluminio para aumentar su robustez y permitir su fijación a un soporte. (Aguirre & Aman, 2017)

Para conseguir distintos niveles de voltaje o mayor capacidad de potencia entregada a la carga se conectan paneles FV en serie y en paralelo, formando lo que se conoce como matriz fotovoltaica. (Tapfumanei, 2016)

2.2.1 Estructura y componentes

La estructura básica de un panel fotovoltaico se muestra en la Figura 2.3, la capa anti reflectiva se construye con materiales que permitan disminuir el porcentaje de fotones que se desaprovechan por efecto de la reflexión, así se consigue que un mayor número de fotones incidan y sean absorbidos por los electrones. La capa iluminada que se encuentra expuesta a la irradiación solar es del tipo N y la del tipo P se encuentra en la zona de la obscuridad. La malla metálica superior crea una resistencia eléctrica por la unión metal-semiconductor y a su vez impide que los fotones lleguen al interior de cada celda por lo que afecta directamente a la eficiencia. (Aguirre & Aman, 2017; Palaguachi & Guiracocha, 2017)

Figura 2.3. Estructura de un Panel Fotovoltaico

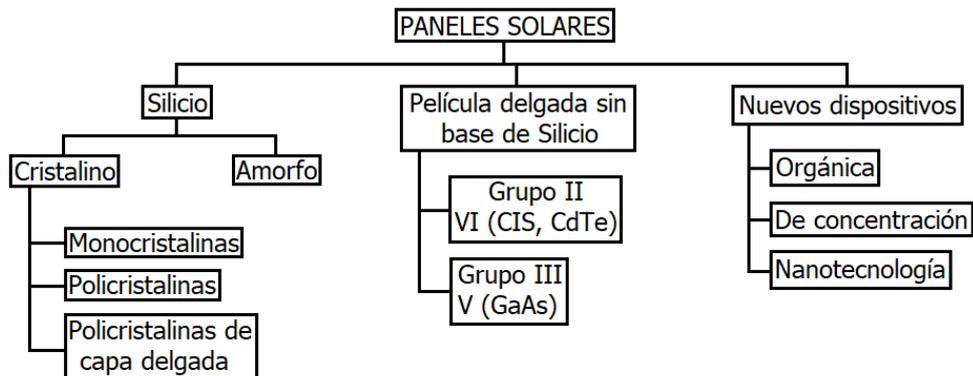


Componentes internos de un panel fotovoltaico. Elaborado por: Geovanny Pullas & Christian Jácome

2.2.2 Clasificación

De acuerdo al material semiconductor del que se encuentran fabricados y a su estructura interna, los paneles FV se pueden clasificar en tres como se pueden apreciar en la Figura 2.4.

Figura 2.4. Cuadro de clasificación



Clasificación de acuerdo al material de fabricación y estructura interna. Elaborado por: Geovanny Pullas & Christian Jácome

Los paneles con cristales de silicio (Si) son los más utilizados por su costo, tienen una eficiencia razonable por celda en monocristalino del 14% al 18% y en policristalino del 15%, los de cristales de silicio amorfo de espesor muy fino usan menor cantidad de material y tienen un rendimiento del 10%. Los de película delgada sin base de silicio son de costo más elevado, rendimiento similar a los de Si, a excepción de los de arseniuro de galio (GaAs). Actualmente se construyen con materia orgánica con un rendimiento bajo, concentración con el 44% y nanotecnología que puede llegar a convertir el 80% de la energía del sol. (Palaguachi & Guiracocha, 2017)

2.2.3 Características eléctricas

Para determinar las características eléctricas, los fabricantes de paneles FV, realizan pruebas de rendimiento bajo condiciones de prueba estándar, STC en el que se especifica el valor de los parámetros como irradiancia de 1000 [W/m²], temperatura ambiente de 25 [°C] y masa de aire de 1.5, los valores obtenidos se indican en la ficha técnica del fabricante y son los siguientes:

Corriente de cortocircuito (I_{sc}): es la mayor corriente que puede obtenerse del panel bajo condiciones STC con voltaje igual a cero y sin carga.

Voltaje de circuito abierto (V_{oc}): es el mayor voltaje que puede obtenerse del panel bajo condiciones STC con corriente igual a cero.

Potencia máxima (P_{max}): es la máxima potencia que puede obtenerse del panel bajo condiciones STC. También corresponde al par máximo de Corriente vs Voltaje.

Corriente de máxima potencia (I_{mp}): es el valor de corriente máxima menor a la corriente de corto circuito (I_{sc}), cuando se encuentre en su máxima potencia en condiciones STC. También corresponde al valor de corriente del par de P_{max} (I_{mp} , V_{mp}).

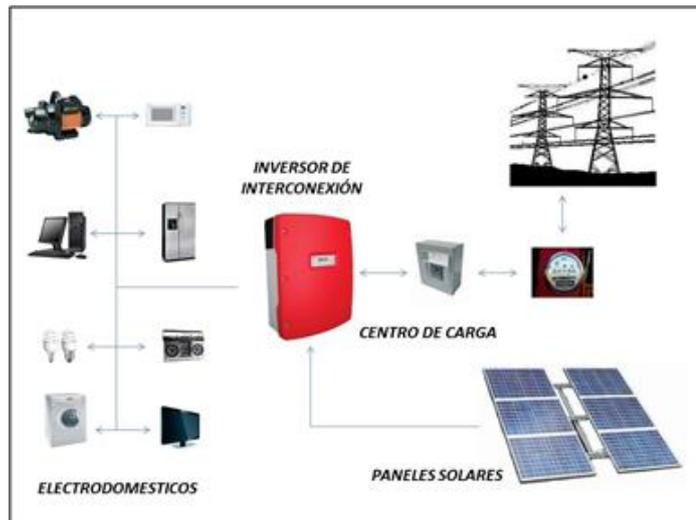
Voltaje de máxima potencia (V_{mp}): es el valor de voltaje máximo menor al voltaje de circuito abierto (V_{oc}), cuando se encuentre en su máxima potencia en condiciones STC. También corresponde al valor de voltaje del par de P_{max} (I_{mp} , V_{mp}). (Aguirre & Aman, 2017)

2.2.4 Aplicaciones

En la actualidad los paneles fotovoltaicos se utilizan en diferentes campos y son diversas sus aplicaciones, las mismas que pueden estar sujetas a sistemas conectados a la red eléctrica y aplicaciones aisladas o de autoconsumo. (Atersa, 2017)

En aplicaciones conectadas a la red eléctrica entregan voltaje de corriente continua a un inversor para entregar corriente alterna, debe disponer de protecciones frente a situaciones como tensiones fuera de rango, corte de la red o desfases en la red. Utilizadas en viviendas o edificios se disminuye el consumo de energía eléctrica desde la red y de existir un excedente de generación se puede alimentar a la red, generando ingresos económicos. (Atersa, 2017)

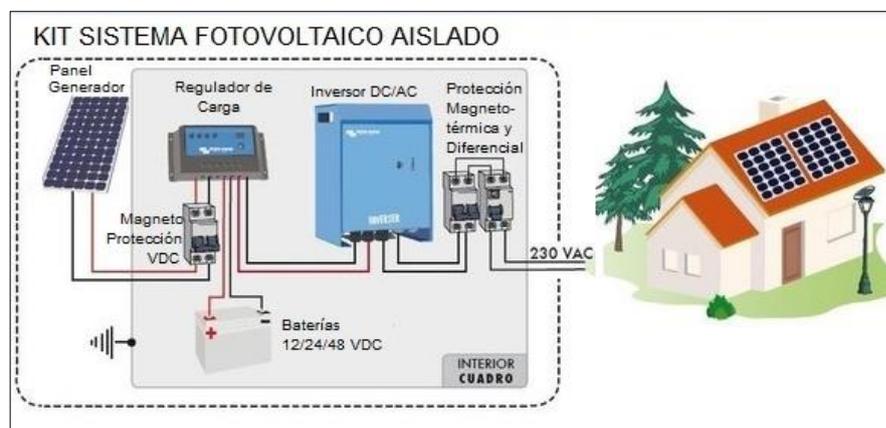
Figura 2.5. Sistemas conectados a la red



Componentes de un sistema conectado a la red. Fuente: (Solar, 2020)

En aplicaciones aisladas se utilizan para disponer de energía eléctrica en lugares de difícil acceso, también se utilizan en sistemas de protección catódica que es un método para proteger estructuras metálicas de la corrosión empleando un pequeño voltaje. Otra aplicación está en sistemas de tratamiento de aguas para la alimentación de una fuente de luz ultravioleta que purifica el agua.

Figura 2.6. Componentes de una instalación fotovoltaica aislada



Arquitectura de un sistema fotovoltaico aislado. Fuente: (Ingemecánica, 2020)

En el campo agrícola se usa energía fotovoltaica para bombas de agua para proveer agua desde pozos o ríos para uso doméstico, alimentar los abrevaderos del ganado, el riego de cultivos y electrificación de cercas. En las telecomunicaciones se energizan equipos de radares, radiotelefonía, repetidoras de radio y televisión, puestos de monitoreo ubicados en zonas remotas y de difícil acceso. (Atersa, 2017)

2.3 MODELO MATEMÁTICO DE UN PANEL FOTOVOLTAICO (FV)

Para un análisis de operación o simulación ante distintas condiciones ambientales o aplicaciones se utiliza un modelo matemático del comportamiento de un panel fotovoltaico, que consiste en una ecuación con una serie de variables y constantes que intervienen en el proceso de obtención de energía eléctrica a partir de energía fotovoltaica, facilitando el tiempo de procesamiento de datos en laboratorio. (De Souza, Behr, Waltrich, Martins, & Coelho, 2019)

El modelo matemático general se obtiene a partir de un circuito equivalente, formado por una fuente de corriente (I_{pv}) que representa la corriente fotogenerada, un diodo (D) que representa la unión de los semiconductores tipo P-N, una resistencia en serie (R_s) que representa la carga propia de los materiales semiconductores P-N y una en paralelo (R_p) que representa la no linealidad de la unión P-N. (De Souza et al., 2019)

Figura 2.7. Circuito equivalente de una celda FV

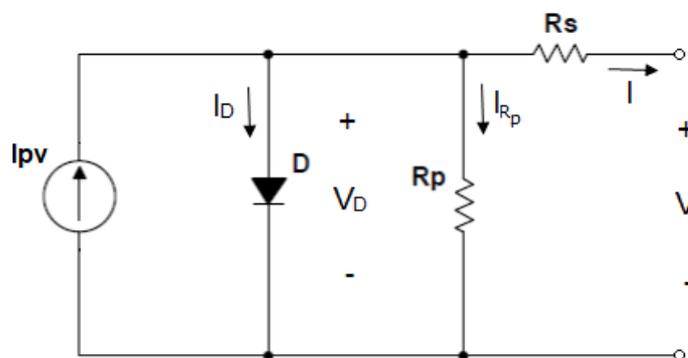


Diagrama equivalente de un Panel Fotovoltaico. Elaborado por: Geovanny Pullas & Christian Jácome

El modelo matemático expresa la corriente de salida en función de una serie de parámetros constantes y variables que se muestran en las siguientes ecuaciones: (De Souza et al., 2019)

$$I = I_{pv} - I_D - I_{Rp} \quad \text{Ec. (2.1)}$$

$$I_{pvn} = \frac{R_s + R_p}{R_p I_{sc}} \quad \text{Ec. (2.2)}$$

$$I_{pv} = \frac{G}{G_n} (I_{pvn} + K_i \Delta_T) \quad \text{Ec. (2.3)}$$

$$V_t = \frac{N_s k T}{q} \quad \text{Ec. (2.4)}$$

$$I_o = \frac{I_{sc}}{e^{\frac{V_{oc}}{a V_{tn}}}} \quad \text{Ec. (2.5)}$$

$$I = I_{ph} - I_o \left(e^{\frac{V+R_s I}{V_t a}} - 1 \right) - \frac{V+R_s I}{R_p} \quad \text{Ec. (2.6)}$$

Donde:

I: Corriente de salida [A]

I_{pvn} : Corriente fotogenerada con valores iniciales de R_s y R_p

I_{pv} : Corriente fotogenerada en [A]

I_D : Corriente del diodo en [A]

I_{Rp} : Corriente de fuga de la unión P-N en [A]

G: Irradiación sobre el panel fotovoltaico en [W/m^2]

G_n : Irradiación nominal a 1000 [W/m^2]

I_{sc} : Corriente de cortocircuito [A]

K_i : Coeficiente de temperatura de corriente en [A/K], incide directamente en la I_{sc} y está dado por el fabricante del panel FV. (Aguirre & Aman, 2017)

Δ_T : Variación de la temperatura actual (T) y la nominal ($T_n = 25 \text{ °C}$) en [K]

V_t : Voltaje térmico [V], es el voltaje en la juntura de la celda a la temperatura que se encuentre el panel.

N_s : Número de celdas [adimensional]
 k : Constante de Boltzmann (1.38×10^{-23} [J/K])
 T : Temperatura del panel en [K]
 q : Carga del electrón (1.6×10^{-19} [C])
 I_o : Corriente inversa de saturación del diodo [A]
 I_{sc} : Corriente de corto circuito [A]
 V_{oc} : Voltaje de circuito abierto en [V]
 a : Factor de idealidad del diodo [adimensional], medida de cuan cerca el diodo real sigue la ecuación del diodo ideal. (Cata & Rodriguez, 2015)
 V_{tn} : Voltaje térmico a temperatura nominal [V]
 V : Voltaje de salida [V]
 R_s : Representa la carga propia de los materiales semiconductores P-N [Ω]
 R_p : Representa la no linealidad de la unión P-N [Ω]

Las Ecuaciones 2.1, 2.2, 2.3, 2.4 y 2.5 complementan a la Ecuación 2.6 que es la principal. El método más conocido de resolución de este tipo de ecuación es el de los cinco parámetros por Newton Raphson (NR), el cual calcula a , I_{pv} , I_o , R_s y R_p . Una vez calculadas estas variables permite obtener los valores de salida de voltaje y corriente teóricos del panel.

Existen diferentes circuitos equivalentes que permiten obtener modelos matemáticos que describen el comportamiento de un panel solar, desde el más simple (modelo ideal), hasta el más complejo como se puede observar en la Tabla 2.1.

El circuito equivalente de diodo ideal contiene una fuente de corriente y un diodo conectados en paralelo lo que lo hace sencillo debido a que no tiene parámetros desconocidos e implica que el método resolutivo de Newton Raphson sea simple con una carga baja de procesamiento y mala precisión, en el de diodo simple se añade una resistencia en serie R_s , que mejora la precisión al considerar las pérdidas por el material semiconductor, apareciendo nuevas variables a considerar, I_{pv} , I_o y R_s , en el método de NR para calcular los nuevos parámetros y realizar la aproximación a la curva con una carga de procesamiento baja, en el modelo de diodo general se añaden dos resistencias R_s y R_p , que le dan al modelo la precisión necesaria para obtener curvas cercanas a la de un panel FV real, pero con una carga de procesamiento media debido a que se

aumentan parámetros a considerar, a , I_{pv} , I_o , R_s y R_p , en este caso al método de NR se lo conoce como el método de los cinco parámetros, en el modelo de dos diodos se añade un diodo que permite simular condiciones de sombra con alta precisión a la aproximación de curvas a costo de un procesamiento elevado y tiempos de espera altos, debido a que se deben calcular siete parámetros, a , a_1 , I_{pv} , I_o , I_{o1} , R_s y R_p , por el método de NR. (Cushicóndor, 2019)

Tabla 2.1. Cuadro de modelos matemáticos

MODELOS CARACTERÍSTICAS	DIODO IDEAL	DIODO SIMPLE	GENERAL	DOS DIODOS
CIRCUITO EQUIVALENTE				
ELEMENTOS	Fuente de corriente, Diodo	Fuente de corriente, Diodo, Resistencia del semiconductor	Fuente de corriente, Diodo, Resistencias del semiconductor y de la unión P-N	Fuente de corriente, Diodos (2), Resistencias del semiconductor y de la unión P-N
PARÁMETROS DESCONOCIDOS	No tiene	a, I_{pv}, I_o, R_s	a, I_{pv}, I_o, R_s, R_p	$a, a_1, I_{pv}, I_o, I_{o1}, R_s, R_p$
MÉTODO DE CÁLCULO	Newton Raphson	Aproximación de curvas junto con Newton Raphson	Método de los 5 parámetros junto con Newton Raphson	Método de los 7 parámetros junto con Newton Raphson
PRECISIÓN	Mala	Buena	Muy buena	Muy buena

Características principales y circuitos equivalentes. Fuente: (Cushicóndor, 2019)

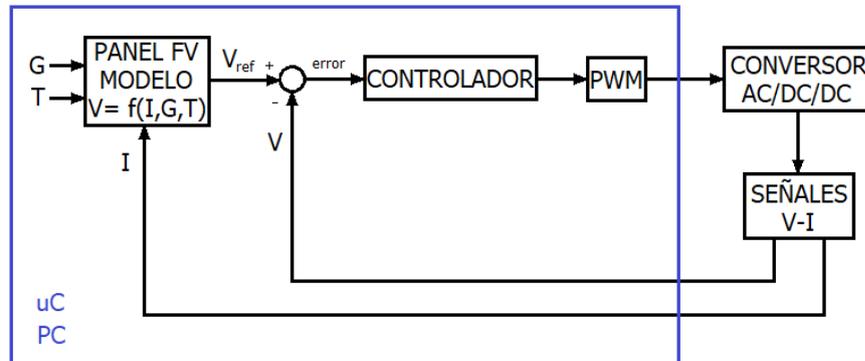
2.4 SIMULADOR DE PANEL FOTOVOLTAICO

Un simulador fotovoltaico es un dispositivo capaz de obtener las curvas características de un panel FV siguiendo un modelo matemático o por medio de tablas de búsqueda de datos. Se basa en un convertidor AC/DC/DC con dos sensores en la salida los cuales permiten medir voltaje y corriente, una de estas señales ingresa directo al modelo matemático dependiendo si el control es por lazo de voltaje o corriente, se calcula en el modelo matemático una señal de referencia (V_{ref} o I_{ref}), para obtener a la salida del controlador un ciclo de trabajo al cual funciona el convertidor, como se puede observar en la Figura 2.8.

Se puede considerar como una fuente de alimentación DC-DC conmutada que es controlada por medio de una computadora o un microcontrolador, y entrega voltajes y corrientes en función del modelo matemático de un panel FV, de tal manera que se

puede reemplazar un panel fotovoltaico real por el simulador para investigación de su operación o del funcionamiento de circuitos que procesan su energía, en la Figura 2.8 se muestra la estructura de un simulador fotovoltaico. (Hasnaoui, Bennani-Ben Abdelghani, & Slama-Belkhodja, 2018)

Figura 2.8. Diagrama en bloques FV



Estructura de un simulador fotovoltaico. Elaborado por: Geovanny Pullas & Christian Jácome

Existen simuladores fotovoltaicos comerciales como el Chroma de la serie 62000H-S, el cual permite almacenar hasta 100 curvas I-V en memoria, con tiempo programado de 1-15000 segundos, incorpora un controlador digital de 16 bits y circuitos de medición con precisión de voltaje del 0.05% y exactitud del 0.1%, para aplicaciones de alta potencia permite que varias de estas fuentes sean conectadas en paralelo hasta conseguir 150 [kW]. (Chroma, 2015)

También existe investigación en este tema como el proyecto (Hasnaoui et al., 2018), en el que se implementa un simulador de generador fotovoltaico basado en un convertidor controlado en puente H de hasta 6 [kW], utiliza una etapa lógica con un algoritmo basado en tablas de búsqueda en lugar de expresiones matemáticas no lineales, junto con un controlador PI por lazo de corriente, que reduce los cambios abruptos de corriente en la salida a causa de la variación de irradiancia en la entrada.

En el proyecto (Erkaya, Moses, Flory, & Marsillac, 2015), se desarrolla un dispositivo capaz de simular módulos fotovoltaicos con un error menor al 10%, donde el usuario selecciona los parámetros de corriente de cortocircuito y voltaje de circuito abierto

asociados a los módulos fotovoltaicos. Utiliza el modelo matemático de diodo ideal y posteriormente el software realiza los cálculos. La etapa de potencia usa un convertidor reductor no síncrono por simplicidad y operación predecible.

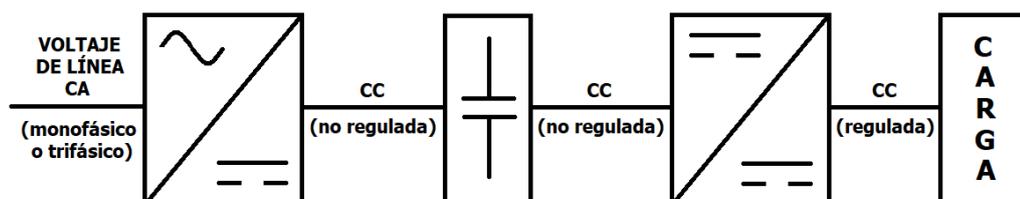
En el proyecto (De Souza et al., 2019), se desarrolla un emulador de matriz fotovoltaica con una potencia de 1kW usando un convertidor reductor que contiene una rama de amortiguación en el filtro de salida para reducir las oscilaciones en condiciones de baja carga. Implementa el modelo general para determinar los parámetros eléctricos de la matriz y resolver completamente la ecuación que describe la curva V-I en un procesador (DSP) junto con un control PI por lazo de corriente.

2.4.1 Convertidor AC-DC-DC

La función principal de un convertidor es suministrar voltaje y corriente continua variable a partir de una fuente AC. Se basa en una operación periódica llamada conmutación, donde se proporciona voltaje de la fuente a la carga durante un tiempo (t_{on}) y posteriormente se aplica un cortocircuito (t_{off}), el resto del período (T). Para la conmutación se necesitan componentes semiconductores totalmente controlados (TBJ, Mosfet, IGBT, GTO, etc.). (Mohan, Ned; Undeland, Tore M; Robbins, 2015)

En la Figura 2.9 se muestra la estructura de un convertidor AC/DC/DC, de tres etapas. El voltaje AC pasa por un convertidor AC/DC, rectificador, convirtiéndolo en voltaje DC, que se filtra con un capacitor para disminuir el ruido y el rizado, este voltaje ingresa al convertidor DC/DC en el cual se establece un nivel DC específico de acuerdo a las necesidades de la aplicación.

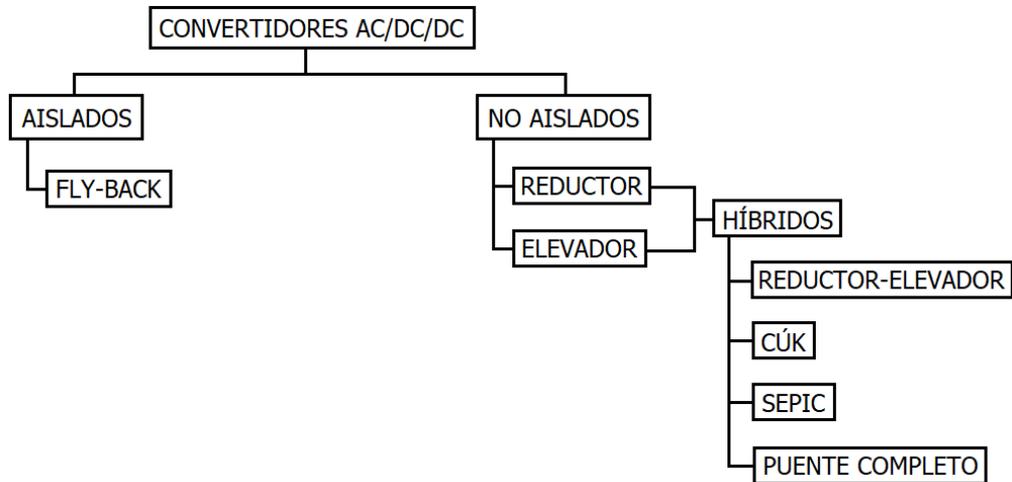
Figura 2.9. Diagrama en bloques de un convertidor AC/DC/DC



Estructura de un convertidor AC/DC/DC. Elaborado por: Geovanny Pullas & Christian Jácome

En la Figura 2.10 se muestra la clasificación de los convertidores. (Mohan, Ned; Undeland, Tore M; Robbins, 2015)

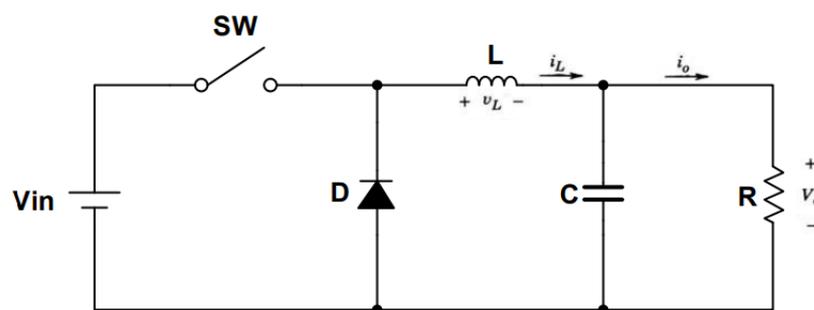
Figura 2.10. Convertidores AC/DC/DC



Clasificación de los convertidores AC/DC/DC. Elaborado por: Geovanny Pullas & Christian Jácome

Convertidor reductor: produce un voltaje de salida menor al presente en su entrada.

Figura 2.11. Convertidor reductor



Topología electrónica Convertidor Buck. Elaborado por: Geovanny Pullas & Christian Jácome

$$V_o = DV_{in}$$

Ec. (2.7)

Donde:

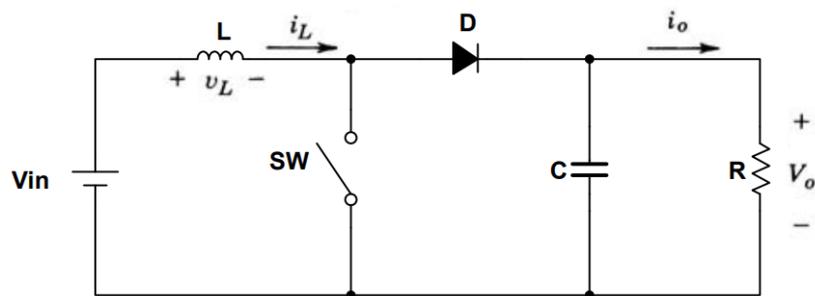
V_o : Voltaje de salida [v]

V_{in} : Voltaje de entrada [v]

D: Ciclo de trabajo

Convertidor elevador: proporciona en la salida siempre un voltaje mayor que su entrada.

Figura 2.12. Convertidor elevador

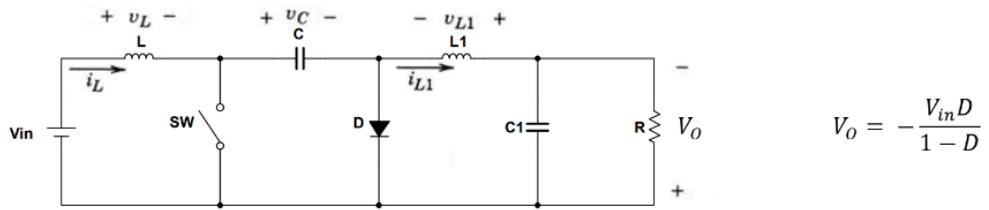


Topología electrónica Convertidor Boost. Elaborado por: Geovanny Pullas & Christian Jácome

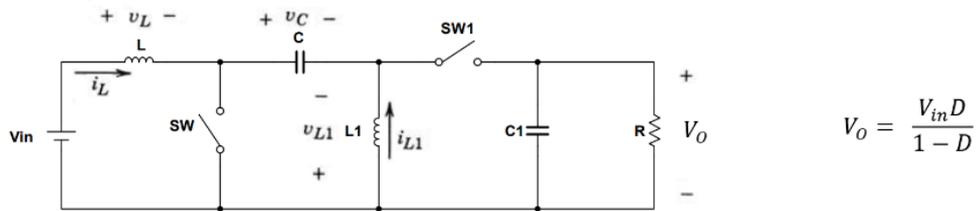
$$V_o = \frac{V_{in}}{1 - D} \quad \text{Ec. (2.8)}$$

A partir de estos convertidores básicos se han desarrollado diversas topologías como los conversores sepic, cúk y zeta, cuya topología y ecuaciones características se muestran en la Figura 2.13. (Mohan, Ned; Undeland, Tore M; Robbins, 2015)

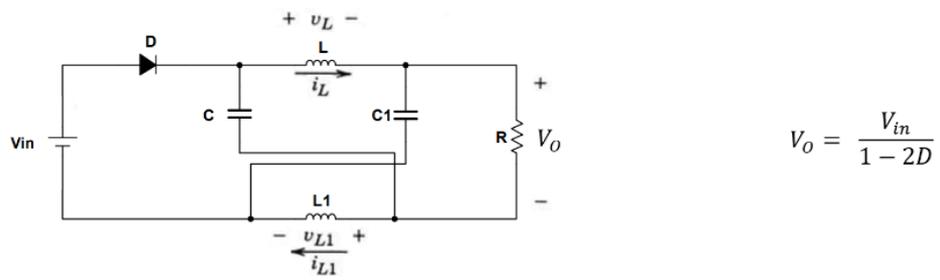
Figura 2.13. Topologías de Convertidores DC-DC



a)



b)



c)

a) Convertidor Cúk. b) Convertidor Sepic c) Convertidor Zeta. Elaborado por: Geovanny Pullas & Christian Jácome

CAPÍTULO 3

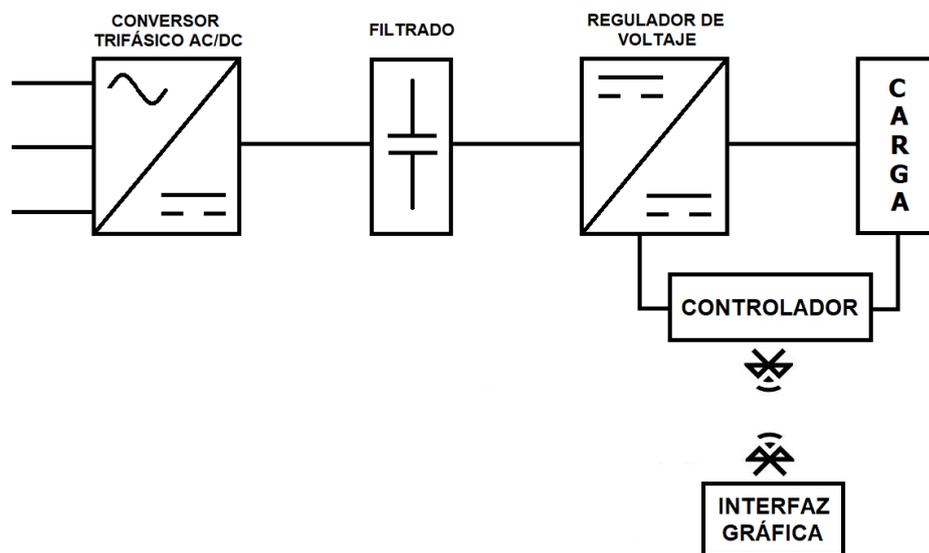
DISEÑO DEL SIMULADOR DE PANEL FOTOVOLTAICO

En el presente capítulo se detalla cada uno de los subsistemas que forman parte del simulador de panel fotovoltaico y su relación de operación dentro del sistema general.

3.1 ESQUEMA GENERAL DE LA SOLUCIÓN

El simulador de panel fotovoltaico se desarrolla de acuerdo con el esquema en bloques que se muestra en la Figura 3.1, en la cual se pueden distinguir las etapas principales para su funcionamiento. El conversor trifásico junto con la etapa de filtrado entrega un voltaje DC de alimentación al regulador, este es realimentado por el controlador, el cual mediante comunicación bluetooth recibe los datos desde la interfaz gráfica.

Figura 3.1. Diagrama en bloques del simulador fotovoltaico



Etapas del diseño del Simulador Fotovoltaico. Elaborado por: Geovanny Pullas & Christian Jácome

- **Conversor trifásico AC-DC**, se encarga de convertir el voltaje alterno que se toma de la red en voltaje DC.
- **Filtrado**, sirve para reducir el rizado de voltaje para obtener una señal con menor distorsión y represente una menor perturbación para la siguiente etapa.

- **Regulador de voltaje**, este bloque entrega un voltaje regulado de acuerdo a un ciclo de trabajo dado.
- **Controlador**, recibe desde la interfaz gráfica el valor del voltaje V_{ref} , calculado con el modelo matemático, lo compara con el voltaje de salida del regulador, y genera la señal de control para el mismo.
- **Interfaz gráfica**, es un sistema HMI que controla los parámetros del panel fotovoltaico a simular y determina la salida que debe tener el regulador de voltaje en función del modelo matemático del panel.
- **Comunicación**, hace posible la transmisión y recepción de datos entre la interfaz gráfica y el controlador.

3.2 DIMENSIONAMIENTO DEL SIMULADOR DE PANEL FOTOVOLTAICO

Con el objetivo de que el simulador permita realizar estudios domiciliarios en el país se ha tomado en cuenta la energía de consumo de una casa promedio en Quito, Ecuador, realizando un estudio práctico de la potencia en paneles solares que se necesita. En la Tabla 3.1 se detallan los diferentes artefactos eléctricos y electrónicos que se han considerado estándar para una casa según la empresa ProViento. Para calcular la energía total (E_{total}) y potencia instalada ($Pot.paneles$) se utilizan las siguientes ecuaciones:

$$E_{total} = \frac{\text{Energía/día}}{\eta} \quad \text{Ec. (3.1)}$$

$$Pot.paneles = \frac{E_{total}}{h/día} \quad \text{Ec. (3.2)}$$

Donde:

Energía/día: Energía total de consumo [Wh/día]

η : rendimiento del arreglo de paneles 90%, en base a la experiencia de implementación de proyectos realizados por parte de la empresa ProViento en el Ecuador.

h/día: número de horas al día donde los niveles de irradiancia son elevados, 3.1 [h/día] (Diaz, 2018)

Tabla 3.1. Potencia de consumo de una casa promedio en Quito

Equipo	Potencia Promedio [W]	Cantidad	Potencia Acumulada	Tiempo horas	Consumo Wh/día
TV	124	1	124	5	620
Licadora	570	1	570	0,08	46
Plancha	1000	1	1000	0,25	250
Mini-Componente	577	1	577	0,50	289
Microondas	1000	1	1000	0,18	180
Lavadora	890	1	890	0,28	249
Ducha	3100	1	3100	0,50	1550
Foco	30	10	300	6	1800
Router	11	1	11	24	264
Teléfono móvil	14	2	28	1	28
Laptop	70	1	70	3	210
Refrigerador	360	1	360	8	2880
		Potencia Total	8030	ENERGÍA TOTAL	8365

Energía/día	8365	Wh/día
Etotal	9295	Wh/día
Pot.paneles	2998	W

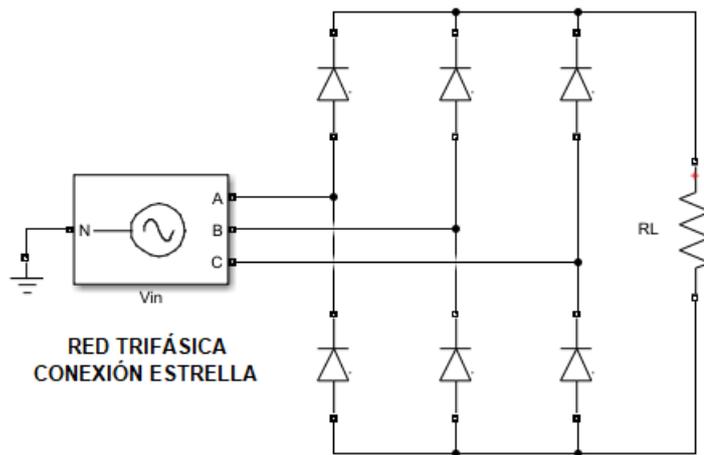
Cálculo de consumo de potencia promedio de electrodomésticos de una casa para reducir el consumo de energía de la red eléctrica. Elaborado por: Geovanny Pullas & Christian Jácome

El dimensionamiento del simulador fotovoltaico se lo realizó con las Ecuaciones (3.1) y (3.2), obteniendo como resultado 3 [kW] de potencia, fijando el valor de voltaje en 250 [V], típico en proyectos fotovoltaicos para una casa promedio, realizados en el Ecuador por parte de la empresa ProViento. Con los valores de potencia y voltaje se obtiene la corriente de 12 [A], parámetros de diseño para el regulador de voltaje (corriente I_o y voltaje V_o de salida).

3.3 DISEÑO DEL CONVERTOR TRIFÁSICO AC-DC

Mediante esta etapa se busca proporcionar un bajo rizado de voltaje al bus DC. Se elije implementar la topología trifásica de puente completo seis pulsos no controlado, considerando su habilidad de manejar mayor potencia frente a la topología de medio puente o cualquier topología monofásica, además se considera un convertor no controlado ya que no se requiere de una salida variable y tiene característica de bajo rizado.

Figura 3.2. Conversor AC-DC



Rectificación trifásica puente completo con valor de carga de 24.75 [Ω], que permite extraer la corriente de 12 [A]. Elaborado por: Geovanny Pullas & Christian Jácome

Previendo el funcionamiento en las instalaciones de laboratorios de la Universidad Politécnica Salesiana este convertor se energiza desde una red eléctrica trifásica en conexión estrella, con un voltaje línea-línea (V_{L-L}) rms de 220 [V_{AC}] que al ser rectificado se puede obtener un voltaje DC (V_{dc}) de acuerdo con la Ecuación (3.3). (Almeida, 2017)

$$V_{dc} = \frac{3\sqrt{2} V_{L-L}}{\pi} \quad \text{Ec. (3.3)}$$

En estas condiciones se puede obtener 297.1 [V], suficiente para satisfacer la entrada del regulador. El diseño del circuito se implementó por medio de Simulink de Matlab, en el cual se midió el rizado de voltaje de 41.82 [V] no siendo óptimo considerando

que el voltaje mínimo es cercano a los 250 [V], y podría reducirse aún más a causa de fluctuaciones en la red trifásica.

3.4 FILTRADO

Esta etapa consiste en la inserción de un capacitor en paralelo con la salida del rectificador, se debe garantizar que la capacidad de descarga sea lo suficientemente grande para sostener el voltaje. Según (Yung, 2016) el límite sugerido está entre el 5% y 10%, si se elige un rizado cercano al 5% el costo del condensador es elevado y un rizado bajo, en cambio al elegir un valor cercano al 10% no se mejora el rizado pero el costo del capacitor es bajo, por lo que se elige un valor intermedio de 7%, logrando una buena relación costo beneficio entre el rizado y precio del capacitor. El capacitor requerido es de 980 [uF], calculado a partir de la Ecuación (3.5). (Almeida, 2017)

$$V_{m_{L-L}} = V_{L-L} \sqrt{2} \quad \text{Ec. (3.4)}$$

$$C_{\min} \geq \frac{I_{dc} \left(\sin^{-1}(1-\gamma) - \frac{\pi}{6} \right)}{V_{m_{L-L}} * \gamma * 2\pi * f} \quad \text{Ec. (3.5)}$$

Donde:

I_{dc} : Corriente DC de salida, 12 [A]

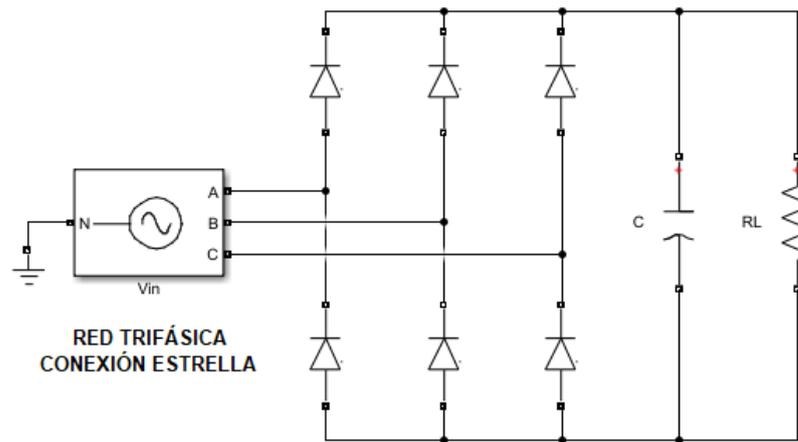
$V_{m_{L-L}}$: Voltaje máximo línea-línea, 311 [V]

γ_v : Porcentaje de rizado de voltaje, 7%

f : Frecuencia de la red eléctrica, 60 [Hz]

Considerando valores estándar se emplea un condensador de 1000 [uF], que por simulación permite obtener un rizado de 13.48 [V] que corresponde al 4.33% del voltaje máximo de línea a línea.

Figura 3.3. Conversor trifásico y filtrado



Rectificación trifásica puente completo con filtrado y carga de $24.75 [\Omega]$ que permite extraer la I_o de $12 [A]$. Elaborado por: Geovanny Pullas & Christian Jácome

3.5 REGULADOR DE VOLTAJE

La salida de voltaje entregada por el rectificador con filtro es DC y en la carga se requiere un voltaje DC regulado por lo que esta etapa está diseñada la topología BUCK, que es un conversor DC-DC reductor sin inversión de polaridad, ya que el voltaje máximo en la salida es menor que el voltaje disponible en la entrada. Para el diseño se consideran las condiciones de operación máxima, que fueron definidas en la sección 3.2, siendo:

- V_{in} : Voltaje de la fuente de $311 [V]$
- V_o : Voltaje de salida de $250 [V]$
- I_o : Corriente de salida de $12 [A]$

Mediante la Ecuación (2.7) así como las ecuaciones (3.6), (3.7), (3.8) y (3.9), que se obtienen a partir del análisis en estado estable del conversor reductor, y con los valores de frecuencia de conmutación (f_s) de $180 [kHz]$, para permitir relaciones de reducción elevadas (Richtek, 2019); rizado de corriente (γ_i) de 5% , que garantiza el modo de conducción continua calculando una inductancia de diseño de $454 [uH]$, aun cuando se trabaje en baja corriente con un ciclo de trabajo mínimo de 0.038 , se obtiene un inductor cuyo valor es mayor al de inductancia mínima.

$$\Delta_i = I_o * \gamma_i \quad \text{Ec. (3.6)}$$

$$I_{o(\min)} = I_o - \frac{\Delta_i}{2} \quad \text{Ec. (3.7)}$$

$$L = \frac{(D * (1 - D) * V_{in})}{f_s * \Delta_i} \quad \text{Ec. (3.8)}$$

$$L_{\min} = \frac{(1 - D) * V_o}{2 * f_s * I_{o(\min)}} \quad \text{Ec. (3.9)}$$

Donde:

Δ_i : Rizado de corriente [A]

L: Inductancia [H]

L_{\min} : Inductancia mínima [H]

$I_{o(\min)}$: Corriente mínima [A]

Para el dimensionamiento del capacitor se consideran las ecuaciones (3.10), (3.11) y (3.12), con los valores de f_s y V_o ya mencionados anteriormente, junto con la consideración de rizado de voltaje del 1% del voltaje de salida del filtrado en base a la medición de un panel real, se obtiene una capacitancia de 170 [nF].

$$\Delta_v = V_o * \gamma_v \quad \text{Ec. (3.10)}$$

$$V_{o(\max)} = V_o + \frac{\Delta_v}{2} \quad \text{Ec. (3.11)}$$

$$C = \frac{V_o * (1 - D)}{8 * L * \Delta_v * (f_s)^2} \quad \text{Ec. (3.12)}$$

Donde:

C: Capacitancia [F]

Δ_v : Rizado de voltaje [V]

El valor de capacitancia teórico de 170 [nF] garantiza obtener un rizado del 5%, sin embargo, se opta por una capacitancia de 470 [nF], debido a que en el mercado ese es el valor más próximo, permitiendo obtener mejores resultados al alimentar una carga variable.

3.6 CONTROLADOR PID

Tomando en cuenta que el rectificador con filtro tiene un rizado reducido, esta variación del voltaje se reflejaría en cambios del nivel de voltaje entregado en la salida del regulador, así mismo al variar los voltajes requeridos en la salida del regulador, el rizado de voltaje se verá afectado, incrementando y disminuyendo según sea el caso, es necesario incluir un lazo de control que mantenga estable el voltaje en la salida del regulador sin importar estas perturbaciones así como para mejorar su respuesta transitoria.

El controlador recibe un setpoint de voltaje (V_{ref}) desde la interfaz gráfica a partir del modelo matemático, también recibe la señal del sensor de voltaje de realimentación colocado en la salida del convertor reductor, para calcular el error que ingresa al compensador y ejecutar la acción de control más conveniente.

Figura 3.4. Controlador PID

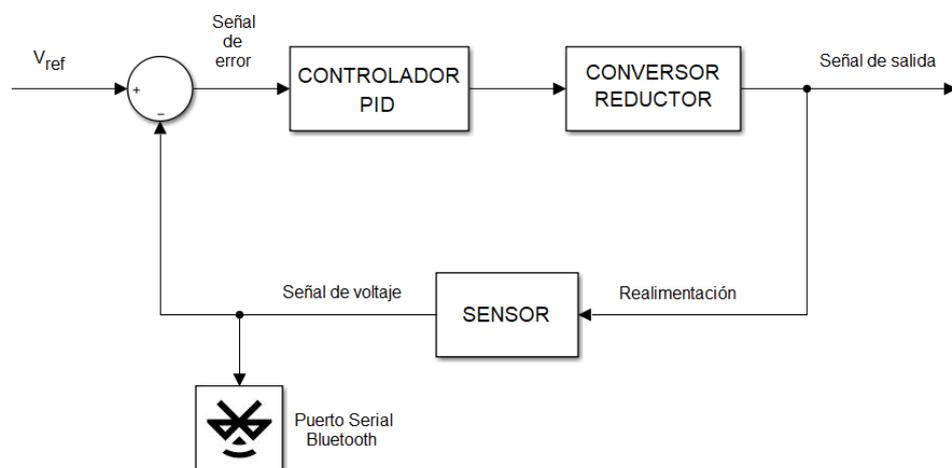


Diagrama en bloques del controlador PID para el convertor reductor. Elaborado por: Geovanny Pullas & Christian Jácome

El controlador utilizado es un PID porque permite alcanzar el estado estacionario en un tiempo de establecimiento mínimo ante variaciones en la carga sintonizándolo de manera adecuada.

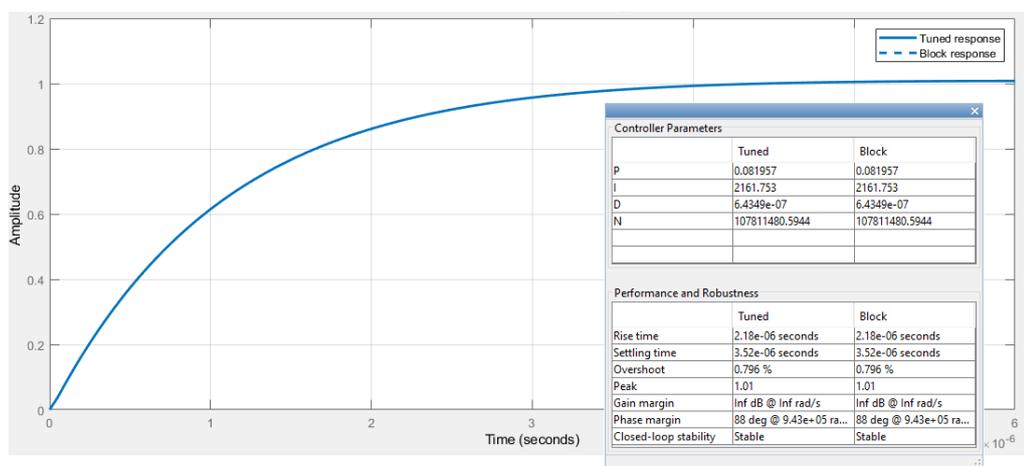
Para el diseño del PID se utiliza la función de transferencia del modelo de pequeña señal del convertidor reductor, que relaciona el voltaje de salida con el ancho de pulso para el interruptor de potencia, como se muestra en la ecuación (3.13). (Vargas, Pérez, Cota, & Castellanos, 2015)

$$\frac{\widehat{V}_o(s)}{\widehat{D}(s)} = \frac{\frac{V_{in}}{LC}}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \quad \text{Ec. (3.13)}$$

$$\frac{\widehat{V}_o(s)}{\widehat{D}(s)} = \frac{1.4575 \times 10^{12}}{s^2 + 1.0214 \times 10^5 s + 4.6865 \times 10^9}$$

La sintonización del controlador se realiza con la ayuda del software Matlab “PID Tuner”.

Figura 3.5. Sintonización con “PID Tuner”



Parámetros de sintonización del controlador PID para el convertidor reductor. Elaborado por: Geovanny Pullas & Christian Jácome

La Figura 3.5 muestra los resultados de los parámetros k_i de 2161.7, k_p de 0.082 y k_d de 6.43×10^{-7} , que producen en la salida del regulador un máximo pico (Mp) de 0.796% y un tiempo de establecimiento (t_s) de 3.52×10^{-6} [s], estos valores dan la mejor respuesta transitoria en la sintonización del PID.

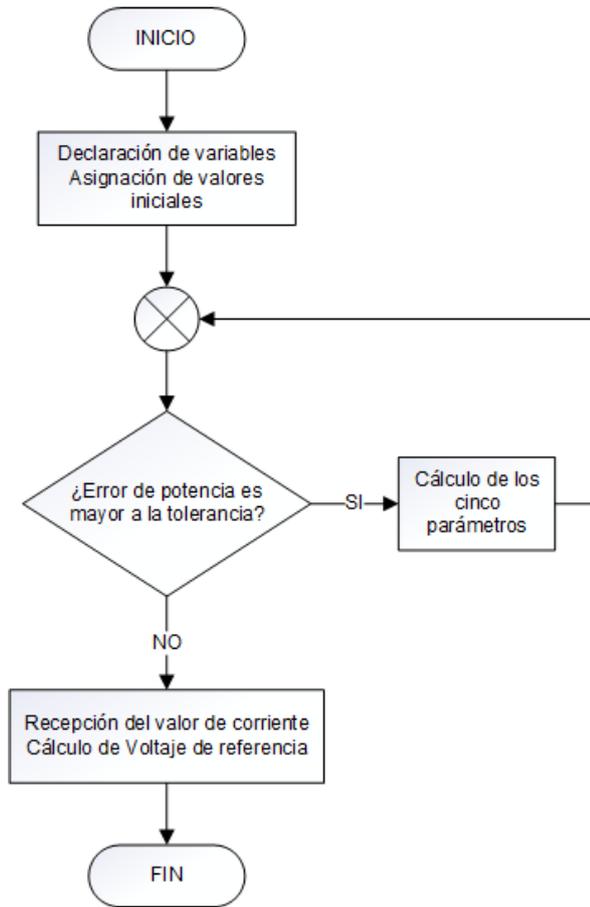
3.7 IMPLEMENTACIÓN DEL MODELO MATEMÁTICO FV

Se desarrolla dentro del algoritmo de la interfaz que se ejecuta en una PC utilizando el software Matlab, con el propósito de aprovechar sus recursos y lograr el procesamiento de datos con mayor rapidez.

El modelo matemático utilizado es el de diodo general, porque se tiene una precisión muy buena en menor tiempo de procesamiento, comparado con los de la Tabla 2.1 de la sección 2.3, el cual requiere que se ingresen los parámetros de la hoja de datos del panel fotovoltaico (V_{oc} , I_{sc} , I_{mp} , V_{mp} , N_s), y además el valor de G. En función de estos valores se calculan los de salida por el método de NR que son a , I_{pv} , I_o , R_s y R_p , detallados en las secciones 2.2 y 2.3. El proceso que sintetiza el algoritmo matemático implementado se muestra en la Figura 3.6.

En el algoritmo matemático FV se inicializan las variables (R_s en 0 y R_p en R_{p_min}) y se fijan las constantes (k , q y K_i) e ingresan las características eléctricas del panel a condiciones STC (N_s , V_{oc} , I_{sc} , V_{mp} , I_{mp}), con estos valores se calcula I_o que ingresa al primer proceso iterativo, con un valor de tolerancia inicializado en 0.001 y error infinito de potencia. Calculando I_{pv} e I_{sc} , el valor de R_p se obtiene de la Ecuación (2.6) igualando I a I_{mp} y V a V_{mp} , ingresa al segundo bucle iterativo y calcula por NR la Ecuación (2.6) de forma vectorial, asignando valores al vector de voltaje se calculan los correspondientes de corriente entre los límites de 0 a V_{oc} con una tolerancia de 0.001, a continuación, con los vectores de voltaje y corriente se consigue el vector de potencia del que se extrae su valor máximo y entre el dato de P_{max_e} se calcula un error, por último se actualiza el valor de R_s en pasos de 0.001. Todo este proceso se repite hasta que el error de potencia sea menor a la tolerancia dada. Cuando el error de potencia es menor a la tolerancia con los valores actualizados de R_s , R_p e I_o se obtienen los valores de “ a e I_{pv} ”. Obtenidos los cinco parámetros se toma el valor de la corriente medida por el sensor, realimentado desde la salida del regulador de voltaje, con estos datos el modelo calcula el voltaje de referencia.

Figura 3.6. Algoritmo matemático FV



Algoritmo matemático del modelo general de panel FV. Elaborado por: Geovanny Pullas & Christian Jácome

Mientras los valores de tolerancia y los pasos de R_s sean menores que los indicados, mayor será el número de iteraciones y demorará más en obtener los parámetros de salida, pero se conseguirán valores más exactos; sin embargo, al reducirlos drásticamente se corre el riesgo de que se quede en el bucle iterando infinitamente debido a que no puede encontrar un error más bajo que la tolerancia. El código implementado se lo realizó en un script de Matlab y se encuentra en el Anexo 2.

3.8 DISEÑO DE LA INTERFAZ GRÁFICA

El HMI ha sido diseñado para permitir el ingreso de parámetros de un panel FV simple o como un arreglo, datos requeridos para ejecutar el modelo matemático, además permite visualizar las curvas teóricas de Voltaje vs Irradiancia, Corriente vs Voltaje,

Potencia vs Voltaje y también los puntos de trabajo a los que se encuentra el regulador de voltaje.

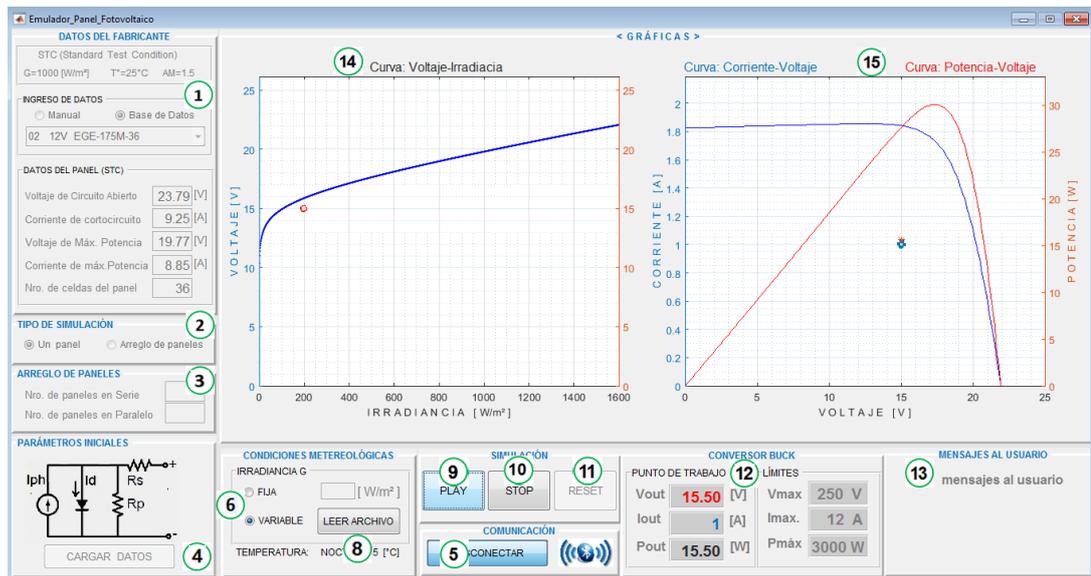
Para la simulación el usuario puede elegir el ingreso de datos del panel FV de forma manual o desde la base de datos, que contiene la información de paneles comerciales comúnmente utilizados en el país; el usuario también puede determinar el tipo de simulación entre un panel simple o un arreglo de conexiones en serie y paralelo; por último, el usuario también puede configurar si el dato de irradiancia se ingresa de forma manual o desde un archivo para dar capacidad de simular el comportamiento ante condiciones reales obtenidas mediante la medición de la irradiancia en un determinado intervalo de tiempo.

La interfaz se encuentra desarrollada con la finalidad de ser amigable con el usuario e intuitiva dado que primero se debe realizar una acción para continuar con la siguiente, los pasos a seguir para el manejo del HMI son los siguientes y se muestra en la Figura 3.7, la interfaz gráfica diseñada:

1. Elegir el ingreso de datos: manual o base de datos.
2. Elegir tipo de simulación: Un panel o Arreglo de paneles.
3. Si es arreglo de paneles definir número de paneles en serie y paralelo.
4. Presionar el botón “CARGAR DATOS”.
5. Presionar el botón “CONECTAR”, establece la comunicación bluetooth.
6. Elegir Irradiancia: fija o variable.
7. Si es irradiancia variable, se debe crear una carpeta en el disco local C llamada “Emulador” y dentro de ella debe existir un archivo Excel “xlsx”, con el nombre “irradiancia” (C:\Emulador\irradiancia). Los datos del archivo irradiancia deben estar en la “Hoja1”, columna “A”, son de tipo entero en el rango de 1-1670 [W/m²] que es la máxima irradiancia en la ciudad de Quito. (Diaz, 2018)
8. Presionar el botón “LEER ARCHIVO”, carga los datos de irradiancia desde un archivo Excel.
9. Presionar el botón “PLAY”, inicia la simulación con los valores configurados.
10. Si se presiona el botón “STOP” cuando se encuentre ejecutando la simulación es interrumpida finalizando el proceso.

11. Para comenzar una nueva simulación pulsar el botón “REINICIAR”.
12. Visualización del punto de trabajo del convertor reductor
13. Mensajes al usuario.
14. Curva Voltaje vs Irradiancia.
15. Curvas de Corriente vs Voltaje y Potencia vs Voltaje.

Figura 3.7. Interfaz gráfica del simulador de panel fotovoltaico



HMI desarrollado por medio del software Matlab. Elaborado por: Geovanny Pullas & Christian Jácome

La interfaz gráfica se diseñó tomando en cuenta los errores que se podrían dar al digitar los datos en el ingreso. Si existe error al cargar los datos, se presentará una ventana de aviso en la cual se informa de qué variable su campo se encuentra vacío, o no se tiene el dato correcto porque no es de tipo numérico (el usuario digita texto) o está fuera de rango. Además de los errores anteriores, la interfaz gráfica realiza una comparación entre voltaje de circuito abierto (V_{oc}) y el voltaje de máxima potencia, si el último es mayor que el primero se despliega un mensaje solicitando que el usuario corrija los valores ingresados, lo mismo ocurre para el caso entre corriente de cortocircuito (I_{sc}) y corriente de máxima potencia (I_{mp}).

Figura 3.8. Diagrama de flujo de la interfaz de datos ingresados

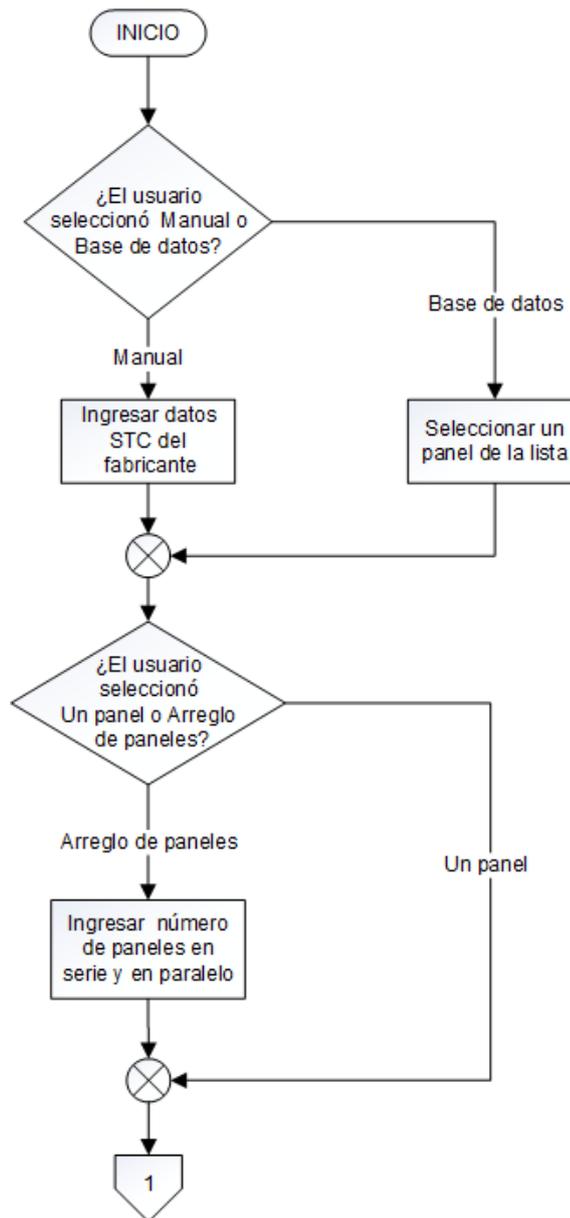


Diagrama de flujo de la interfaz para validar los datos ingresados primera parte. Elaborado por: Geovanny Pullas & Christian Jácome

Mientras el usuario ingrese datos erróneos, la interfaz gráfica seguirá enviando de manera infinita los mensajes detallando el tipo de error y solicitando sea corregido, como se denota en el diagrama de la Figura 3.8, con los valores adecuados los datos se cargan exitosamente, luego las secciones “DATOS DEL FABRICANTE”, “TIPO DE SIMULACIÓN”, “ARREGLO DE PANELES” y el propio botón “CARGAR

DATOS”, (puntos 1, 2, 3, 4 de la Figura 3.7), serán inhabilitados con la finalidad de que no se puedan alterar los datos mientras se realiza la simulación. Luego de bloquear las secciones indicadas se grafica la curva de Voltaje vs Irradiancia en condiciones de circuito abierto.

Figura 3.9. Diagrama de flujo para mostrar de la curva Voltaje vs Irradiancia

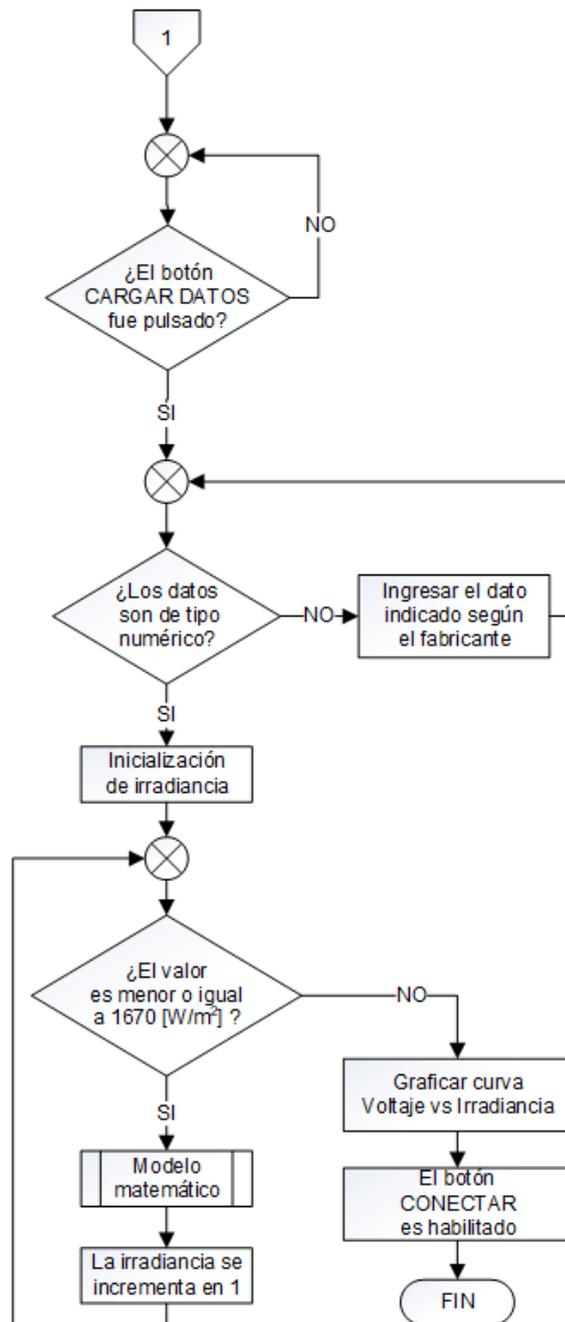


Diagrama de flujo de la interfaz para validar los datos ingresados segunda parte y gráfica de curva de Voltaje vs Irradiancia. Elaborado por: Geovanny Pullas & Christian Jácome

'C:\Emulador\irradiancia', si el archivo no es encontrado, aunque se presione el botón ‘PLAY’ la simulación no correrá hasta que el usuario pegue el archivo en la dirección y con el nombre indicado. En la Figura 3.10, se muestra el diagrama de flujo de los procesos descritos y a continuación se muestra en la Figura 3.11 la inicialización de variables que empiezan al ejecutar el botón ‘PLAY’.

Figura 3.11. Diagrama de flujo de la interfaz y variables previas a la simulación

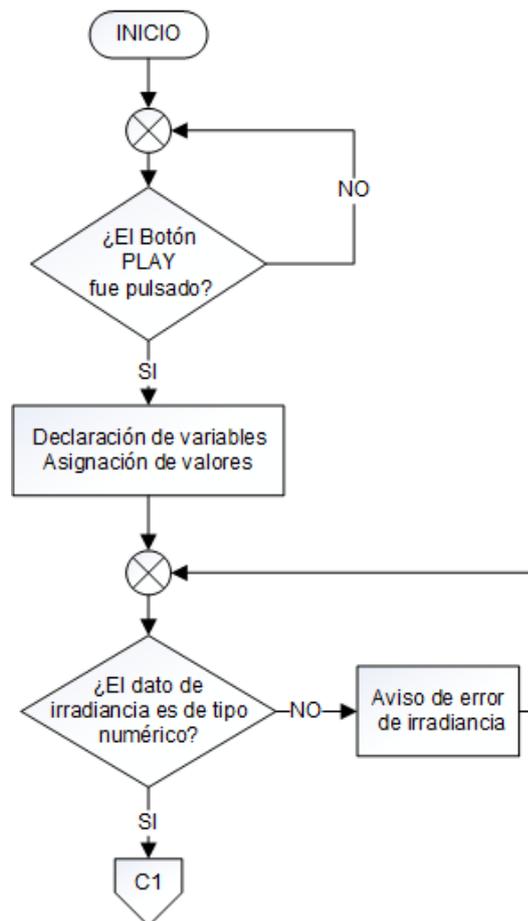


Diagrama de flujo de la interfaz para la creación de variables globales previas al modo de simulación.
Elaborado por: Geovanny Pullas & Christian Jácome

Si el usuario presiona el botón ‘PLAY’ sin importar el tipo de simulación y modo de irradiancia seleccionados se crean variables globales necesarias para el modelo matemático, las detalladas en el Capítulo 2 y además para la funcionalidad de la interfaz, por ejemplo la variable ‘stop’ necesaria para determinar en cualquiera de los casos de simulación el paro de la misma; ‘VectorDatosG’ contiene los datos del

archivo irradiancia; “GQuito” almacena todo el rango valores desde 1 [W/m²] hasta “GmaxQuito” el valor máximo de la irradiancia en la ciudad de Quito; en el vector “Resul_Vref_1” se ingresarán los resultados obtenidos del modelo matemático. A continuación, se evalúa si el dato ingresado manualmente en modo de irradiancia “FIJA” es un valor entero comprendido en los valores de 1-1670 [W/m²], caso contrario muestra el mensaje de aviso respectivo y se consulta la configuración hecha por el usuario mediante las variables “TipoDeSimulación” y “TipoDeIrradiancia”. En la Figura 3.11 se muestra el diagrama de flujo.

Una vez creadas e inicializadas las variables necesarias, el algoritmo busca el tipo de simulación configurado por el usuario siendo posibles las siguientes opciones:

- Opción1, Tipo de Simulación: Un Panel e Irradiancia Fija.
- Opción 2, Tipo de Simulación: Un Panel e Irradiancia Variable.
- Opción 3, Tipo de Simulación: Arreglo e Irradiancia Fija.
- Opción 4, Tipo de Simulación: Arreglo e Irradiancia Variable.

En la Figura 3.12 se muestra el diagrama de flujo para la “Opción 1”, se toma el dato de la irradiancia y se compara con el valor de la variable “Ganterior” inicializada a cero, como el valor es diferente porque la irradiancia empieza en 1 [W/m²], la primera vez ingresa a la condición encargada de la comparación, luego carga el valor de irradiancia actual en la variable “Ganterior”, calcula las variables y ejecuta el algoritmo del modelo matemático. Cuando termina se procede a calcular el voltaje de referencia (V_{ref}) y a graficar las curvas Corriente vs Voltaje, Potencia vs Voltaje, Voltaje vs Irradiancia, así como los respectivos puntos de trabajo en las gráficas, por último, muestra los valores del voltaje, corriente y potencia de salida del convertidor reductor. Si el valor de la irradiancia actual es igual al valor de la variable “Ganterior”, el algoritmo realiza un salto hacia el cálculo de V_{ref} , evitando así que se calcule nuevamente para un valor repetido de irradiancia, disminuyendo el tiempo de procesamiento en la simulación.

Figura 3.12. Diagrama de flujo de irradiancia fija

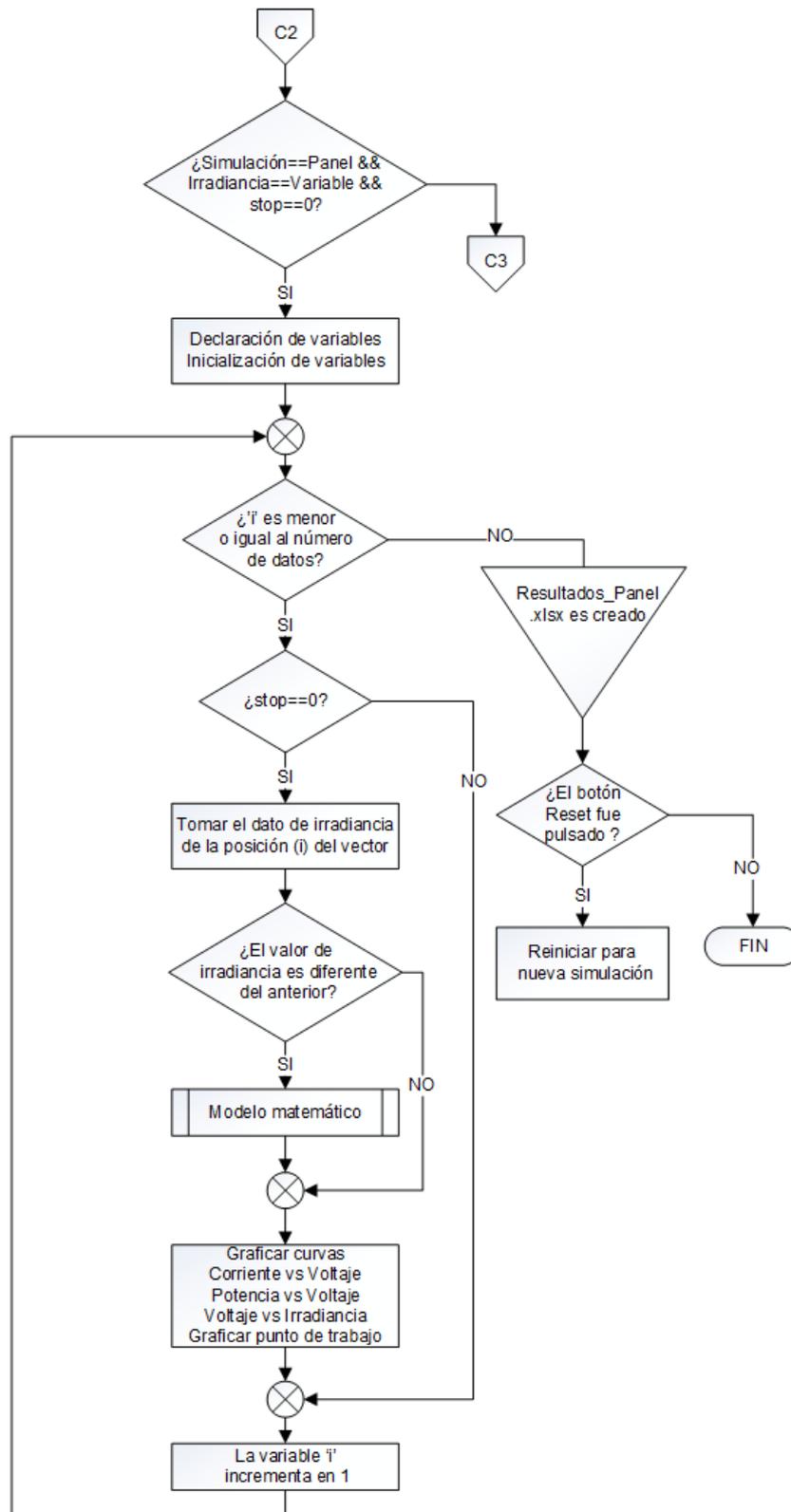


Diagrama de flujo de la interfaz para seleccionar el modo de ingreso de la irradiancia. Elaborado por: Geovanny Pullas & Christian Jácome

Todo el proceso detallado se repite de manera continua e inicia nuevamente cuando se ingresa un nuevo valor y se interrumpe si el usuario pulsa el botón “STOP”. Cuando la simulación se detiene únicamente se puede volver a simular presionando “RESET”.

Para la opción 2 en la que se simula un panel con irradiancia variable se declaran las variables globales “DatosG_Archivo_irradiancia”, que almacena los datos del archivo irradiancia; “GmaxQuito” irradiancia máxima de Quito; y se inicializan “stop”, “G_anterior” y “Resul_Vref_2”.

La simulación procede de manera similar a la simulación de un panel con irradiancia fija, con la diferencia de que la aplicación no ingresa en un lazo infinito simplemente obtiene resultados de acuerdo al número de datos del archivo “irradiancia” y en el cálculo de voltaje de referencia en cada iteración va almacenado su valor en el vector “Resul_Vref_2”.

La aplicación realiza el salto en caso de que el valor actual de irradiancia sea igual al anterior. Al finalizar el cálculo de todos los datos se crea un archivo en formato Excel con el nombre ‘ResultadosVref_PanelSimulado.xlsx’ los resultados del voltaje y corriente de salida del conversor reductor correspondientes a los datos del archivo irradiancia. Finalmente, el usuario puede presionar “RESET” si desea realizar una nueva simulación o cerrar la aplicación. En la Figura 3.13 se muestra el diagrama de flujo.

Cuando se desea simular un arreglo de panel fotovoltaico ya sea con irradiancia “FIJA” o “VARIABLE”, se procede de manera idéntica a la de un panel en cada opción respectivamente, con la diferencia de que en el cálculo del voltaje de referencia se toma en cuenta el número de paneles en serie (N_{sa}) y en el caso de la corriente el número de paneles en paralelo (N_{pa}), además estos valores influyen en otras variables, luego se grafican las curvas. En la Figura 3.14 se muestra el diagrama de flujo.

Figura 3.13. Diagrama de flujo de la simulación de un panel en irradiancia variable

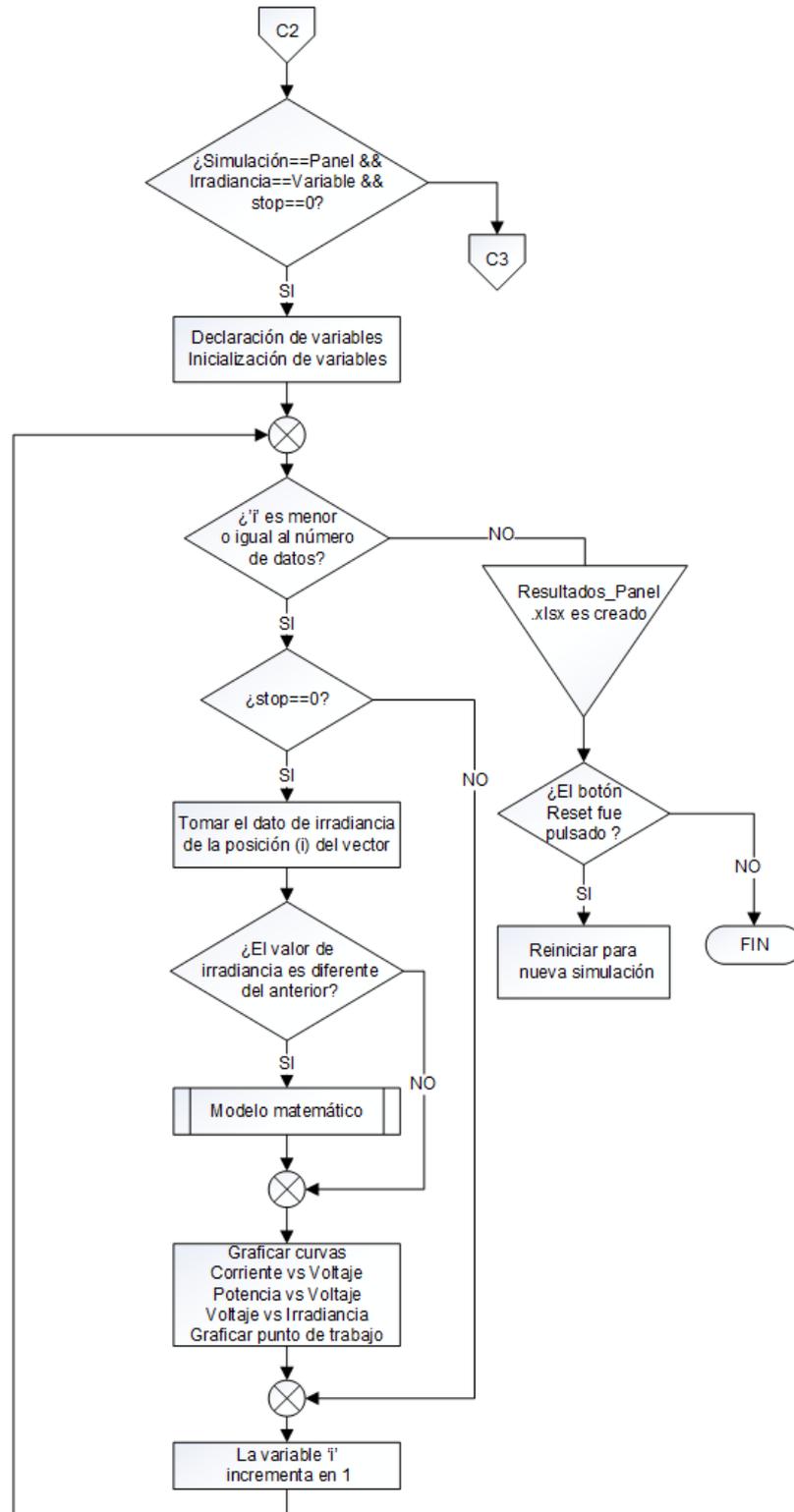


Diagrama de flujo de la interfaz para simular un panel en modo de irradiancia variable. Elaborado por: Geovanny Pullas & Christian Jácome

Figura 3.14. Diagrama de flujo de la simulación de un arreglo en irradiancia variable

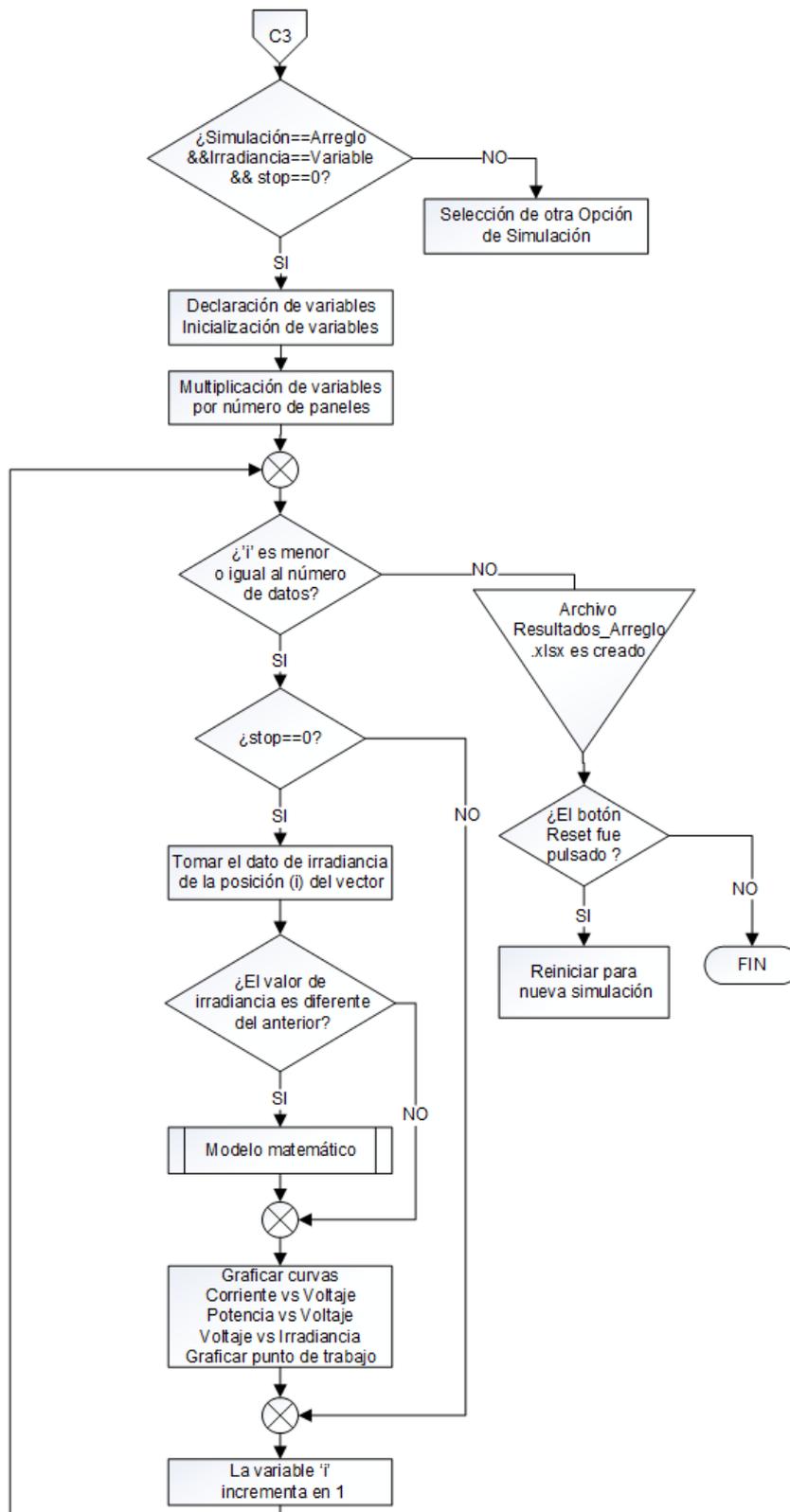


Diagrama de flujo de la interfaz para simular un arreglo de paneles en modo de irradiancia variable.

Elaborado por: Geovanny Pullas & Christian Jácome

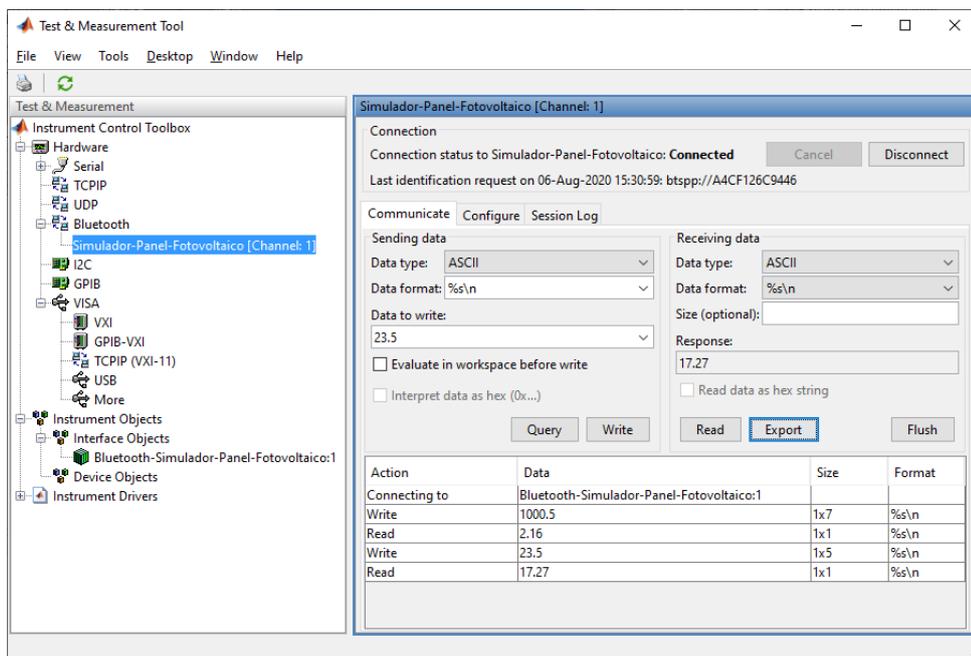
3.9 COMUNICACIÓN SERIAL BLUETOOTH

Para brindar flexibilidad en la operación del simulador, se implementa un medio de comunicación inalámbrico entre la interfaz gráfica y el controlador del regulador de voltaje, donde la interfaz envía el valor de voltaje en función del modelo matemático y recibe la información de los sensores de voltaje y corriente, en el punto de operación del regulador de voltaje.

Se utiliza comunicación serial bluetooth tomando en cuenta su velocidad de transmisión frente a la necesidad de un intercambio rápido de datos e inmunidad al ruido electromagnético, creando una pequeña red inalámbrica que simplemente necesita que se emparejen los dispositivos para comenzar a transmitir la información.

Para la implementación bluetooth el software Matlab contiene la APP “Instrument Control”, que permite buscar el dispositivo y establecer la comunicación, se deben configurar dos parámetros, tipo y formato de datos a enviar. Terminada la configuración en la pestaña “Session Log”, se genera el código de programación que debe contener el algoritmo de la interfaz para emparejar los dispositivos.

Figura 3.15. APP “Instrument Control”

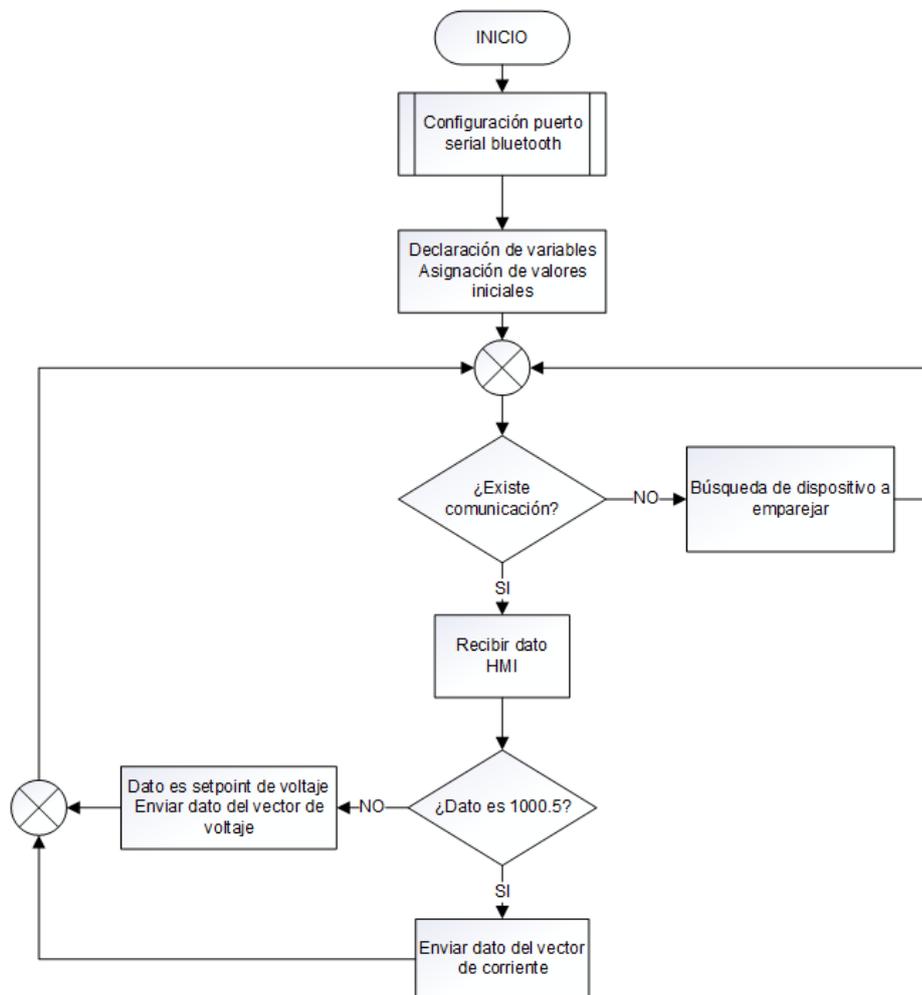


Configuración de la comunicación bluetooth. Elaborado por: Geovanny Pullas & Christian Jácome

El dispositivo usado para validar la comunicación serial bluetooth implementada en Matlab es la tarjeta ESP-WROOM-32. El microcontrolador debe contener la programación bluetooth y estar listo para establecer conexión.

La programación implementada en la ESP-WROOM-32, primero habilita y luego se le asigna el nombre de “Simulador-Panel-Fotovoltaico” dentro de la red inalámbrica, facilitando la búsqueda del dispositivo a ser emparejado. Se conectan dos pulsadores en los pines de entrada de la tarjeta, estos dan la orden para extraer uno a uno los datos de dos vectores de corriente y voltaje.

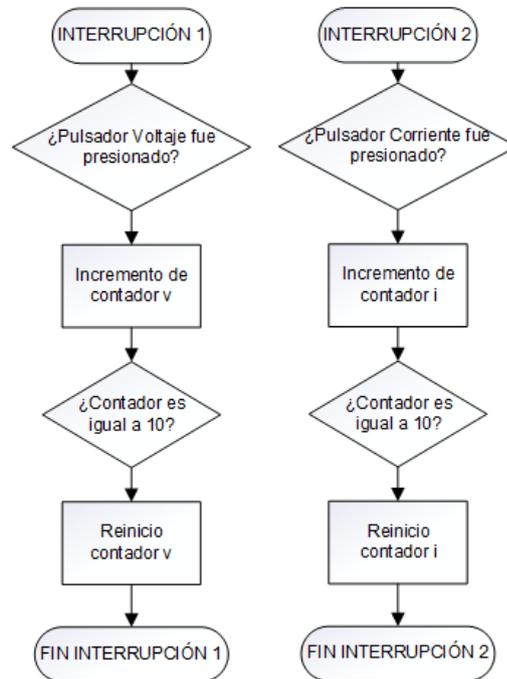
Figura 3.16. Algoritmo de comunicación en la ESP-32



Comunicación bluetooth y transferencia de datos. Elaborado por: Geovanny Pullas & Christian Jácome

El diagrama de flujo de las interrupciones implementadas en el dispositivo receptor bluetooth para poder cambiar los valores de los vectores de voltaje y corriente se muestran a continuación.

Figura 3.17. Interrupciones implementadas en la ESP-WROOM-32



Interrupciones para cambiar los datos de los vectores de corriente y voltaje. Elaborado por: Geovanny Pullas & Christian Jácome

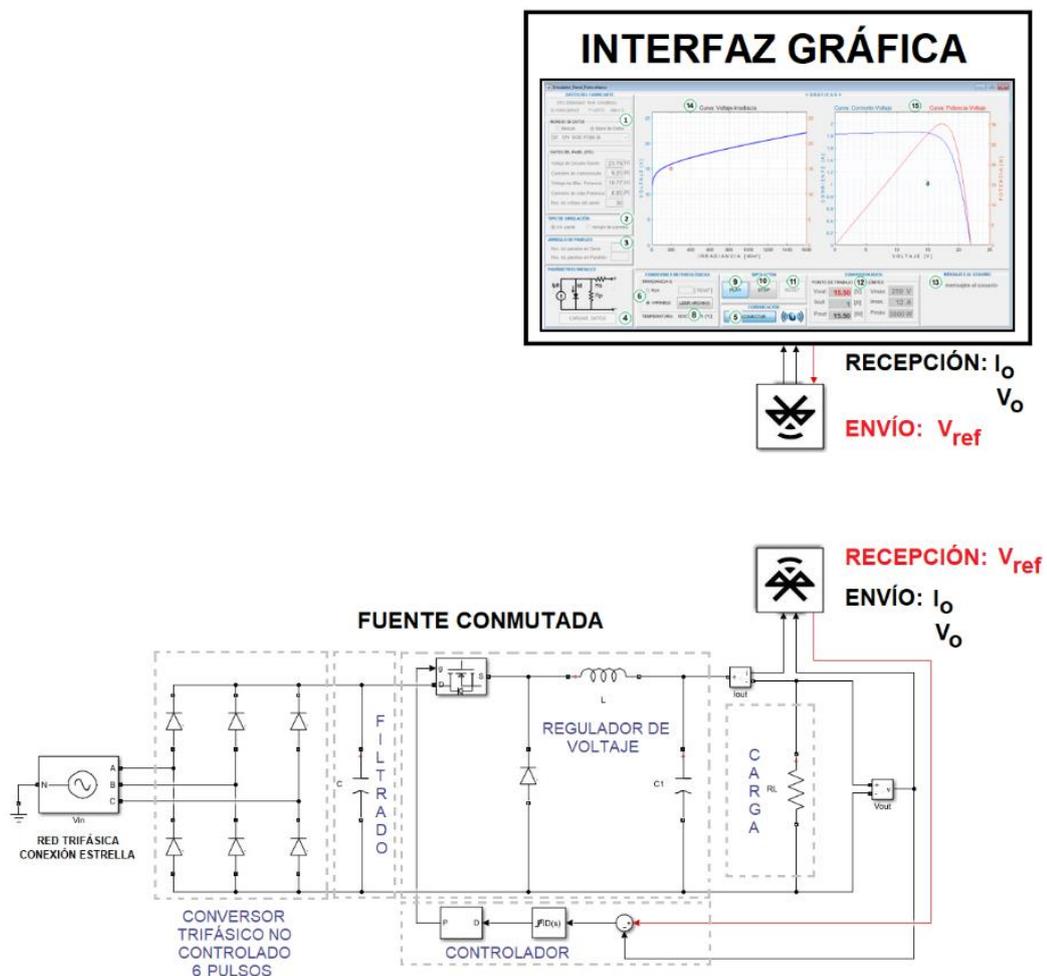
La comunicación es de tipo maestro-esclavo desde la HMI hacia la ESP-WROOM-32, para emparejar los dispositivos se necesita habilitar sus respectivos bluetooth, esto empieza al pulsar el botón “CONECTAR” desde la interfaz. Al correr la simulación desde la HMI se envía al controlador el V_{ref} obtenido con el modelo matemático, este responde con el dato extraído del vector voltaje, que representa el voltaje medido por el sensor a la salida del regulador. El sensor de corriente está representado por un vector, los valores se obtienen mediante la solicitud enviada desde la HMI al receptor ESP-WROOM-32, este valor es necesario en el modelo matemático para calcular un nuevo voltaje de referencia. El siguiente diagrama de flujo complementa lo explicado.

3.10 DISEÑO DEL SIMULADOR DE PANEL FOTOVOLTAICO COMPLETO

El proyecto desarrollado tiene la capacidad de simular un panel o arreglo de paneles fotovoltaicos utilizando en su algoritmo el modelo matemático que determina su comportamiento, para lo cual cuenta con una interfaz gráfica en la que se visualizan las curvas de Corriente vs Voltaje, Potencia vs Voltaje, Voltaje vs Irradiancia y el punto de trabajo.

En la Figura 3.18 se muestra el proyecto del simulador de panel fotovoltaico en el que se detallan todos los bloques mencionados en la sección 3.1 del esquema general de la solución.

Figura 3.18. Simulador de panel fotovoltaico diseñado



Sistema completo del simulador de panel fotovoltaico. Elaborado por: Geovanny Pullas & Christian Jácome

Los datos del panel pueden ser ingresados manualmente o seleccionar un modelo de panel de un listado. Además del tipo de simulación, es posible seleccionar el modo de ingreso de la irradiancia entre fija o variable, en la primera el usuario digita el valor y en el segundo se extrae los valores desde un archivo en formato Excel.

La interfaz gráfica establece comunicación mediante bluetooth con el dispositivo ESP-WROOM-32, solicita el valor de la corriente de salida del regulador para calcular mediante el modelo matemático el voltaje de referencia, recibe el voltaje de salida del regulador para visualizar el punto de trabajo. Ambos datos de corriente y voltaje de salida al ser una simulación se los representa mediante vectores.

El controlador trabaja mediante un lazo de voltaje, siendo realimentado desde la salida del regulador, compara ese valor con el V_{ref} que recibe del HMI y genera la señal de control para el regulador de voltaje, una vez que el mismo llega al valor de setpoint recibido, en ese instante el punto de trabajo es enviado por vía bluetooth hacia la HMI que a su vez es graficado en cada una de las curvas mencionadas.

CAPÍTULO 4

SIMULACIÓN Y PRUEBAS

En el siguiente apartado se muestran mediciones, simulaciones, análisis de gráficas y los resultados que se obtuvieron en cada etapa del simulador de panel fotovoltaico comprobando el funcionamiento de cada subproceso con los resultados teóricos y simulados.

4.1 PRUEBA DEL MODELO MATEMÁTICO FV

Para poder validar el modelo matemático FV se realizaron mediciones de un panel comercial en la empresa ProViento, que proporcionó la instrumentación adecuada para la medición de irradiancia en base a la norma ISO 9060, corriente y voltaje, de un panel FV de la serie EGE-175M-36, desde las 9:30 am a 15:10 pm, se tomaron mediciones en períodos de cinco minutos sin carga, con la finalidad de tener resultados experimentales y comparar con los simulados.

La validación del modelo matemático FV se realizó bajo las siguientes condiciones, primero ingresan en el simulador por medio de un archivo Excel llamado “irradiancia”, los valores de irradiancia experimentales ($G_{\text{experimental}}$) medidos del panel EGM-175-36; luego se hacen las siguientes configuraciones en la interfaz eligiendo el panel “02 12V EGM-175-36” de la base de datos, en tipo de simulación “Un panel” y en irradiancia “VARIABLE”.

Al ingresar los valores de $G_{\text{experimental}}$ al modelo matemático se tiene un voltaje de referencia simulado ($V_{\text{ref(simulado)}}$) y con el experimental ($V_{\text{ref(experimental)}}$) se calcula un error relativo ($\mathcal{E}V_{\text{ref}}$) por medio de la Ecuación (4.1). Para conocer la exactitud del modelo matemático FV se calculó el promedio de errores relativos ($P\mathcal{E}V_{\text{ref}}$) por medio de la Ecuación (4.2) obteniendo como resultado promedio 6.08%. Los valores mencionados se pueden observar en el Anexo 1.

$$\text{error} = \left| \frac{\text{valor experimental} - \text{valor simulado}}{\text{valor experimental}} \right| \times 100 \quad \text{Ec. (4.1)}$$

$$\text{Promedio} = \frac{\sum_{i=0}^n \text{error} (i)}{n}$$

Ec. (4.2)

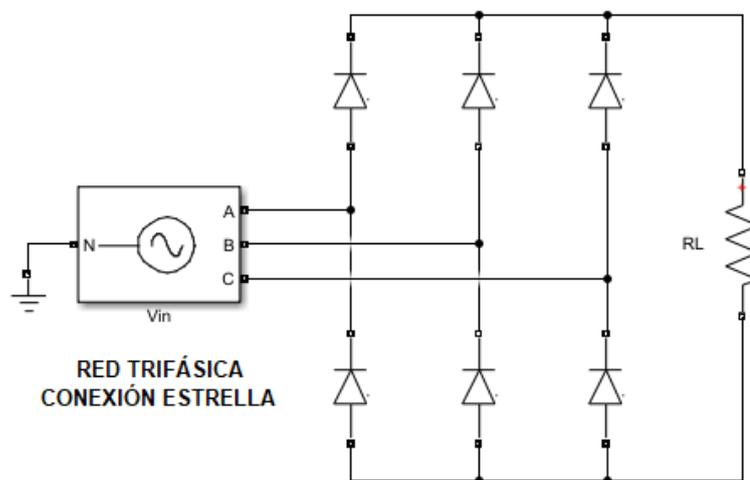
Donde:

n: número de muestras

4.2 PRUEBA DEL CONVERTOR TRIFÁSICO NO CONTROLADO

La prueba de funcionamiento del rectificador trifásico no controlado se realizó en Simulink de Matlab, en el que se colocó una fuente trifásica con V_{L-L} de 220 [V] rms, un puente de diodos trifásico y una resistencia (24.75 [Ω]) con la que se logra extraer la corriente de acuerdo al dimensionamiento de la potencia del simulador FV. Los resultados obtenidos se muestran en la Figura 4.1.

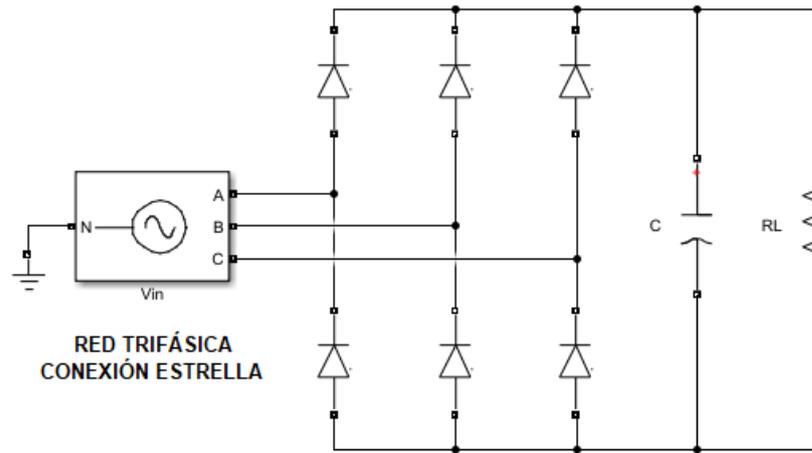
Figura 4.1. Simulación conversor trifásico no controlado



Simulación conversor trifásico no controlado sin filtrado. Elaborado por: Geovanny Pullas & Christian Jácome

En la Figura 4.2 se muestra el conversor trifásico AC-DC con la etapa de filtrado obteniendo las curvas de voltaje DC y rizado de voltaje en la salida.

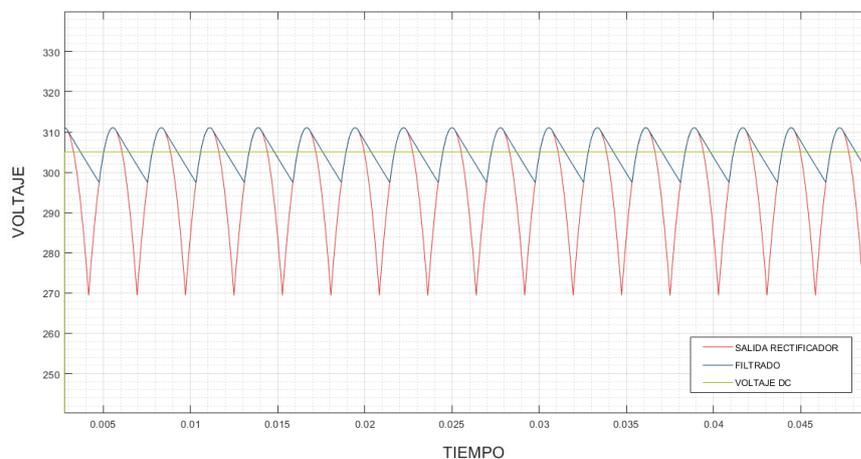
Figura 4.2. Simulación conversor trifásico no controlado con filtrado



Simulación conversor trifásico no controlado con filtrado. Elaborado por: Geovanny Pullas & Christian Jácome

Al realizar la simulación se obtiene un V_{dc} de 297.1 [V] e I_{dc} de 12 [A], sin filtrado, mientras que después del filtrado el V_{dc} de 305.1 [V] e I_{dc} de 12.33 [A], se puede notar que el voltaje tiene un incremento a causa del filtro, obteniendo Δ_v de 13.48 [V], valor medido con osciloscopio de Simulink.

Figura 4.3. Señales de salida del conversor trifásico no controlado y filtrado

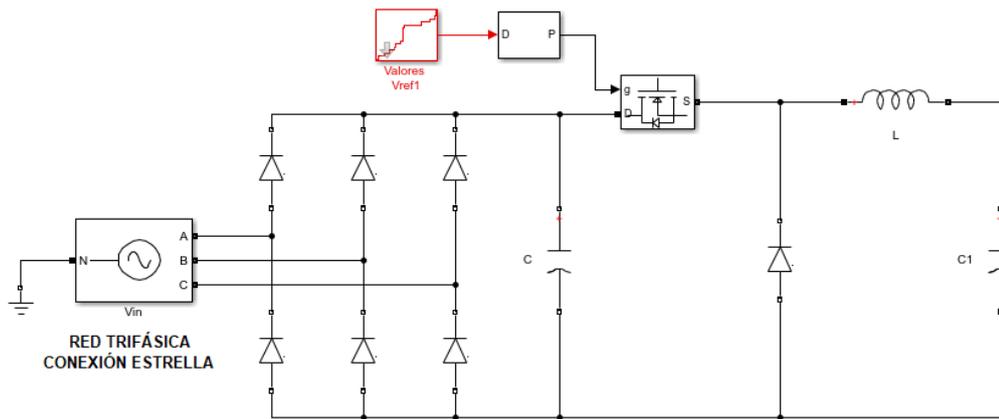


Comparación de curvas entre voltaje rectificado, filtrado y DC. Elaborado por: Geovanny Pullas & Christian Jácome

4.3 PRUEBA DEL CONVERTOR REDUCTOR EN LAZO ABIERTO

Para comprobar el funcionamiento del convertor reductor se realizan dos pruebas, una estática que consiste en ingresar un ciclo de trabajo al conmutador, calculado con la Ecuación (2.7), especificado en la columna “D” en tabla de Anexos 1; y otra dinámica ingresando todos los valores “D”. El diagrama de la Figura 4.4 se empleó para simular las pruebas dinámica y estática.

Figura 4.4. Diagrama del convertor reductor en lazo abierto



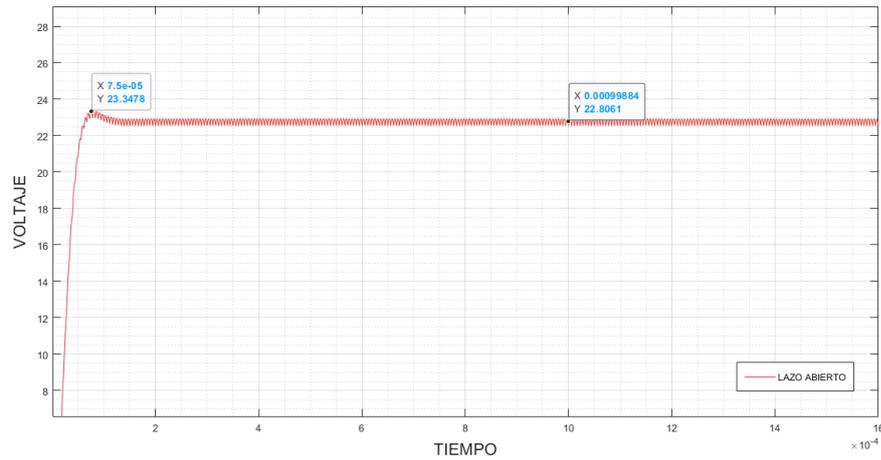
Simulación del convertidor reductor en lazo abierto con ciclo de trabajo D. Elaborado por: Geovanny Pullas & Christian Jácome

4.3.1 Prueba estática del convertor reductor en lazo abierto

La prueba estática del convertor reductor en lazo abierto se realizó ingresando al conmutador un ciclo de trabajo de 0.066 que representa al $V_{ref(simulado)}$ de 20.5 [V] calculado por medio de la Ecuación (2.7).

En la Figura 4.5, el tiempo de establecimiento de la señal es de 1 [ms], mostrando un máximo pico de 1.03%, el rizado de voltaje después del tiempo de establecimiento (t_s) de 350 [mV] y el V_o (lazo abierto) de 22.6 [V] con un error relativo del 11.4% frente al $V_{ref(experimental)}$ de 20.3 [V].

Figura 4.5. Respuesta estática del sistema en lazo abierto

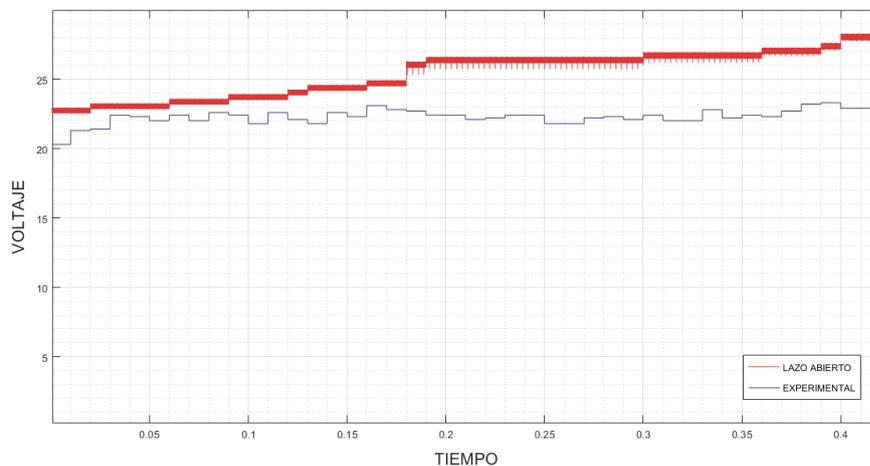


Respuesta del convertidor reductor en lazo abierto frente a un ciclo de trabajo de 0.066 que representa un voltaje de 20.5 [V]. Elaborado por: Geovanny Pullas & Christian Jácome

4.3.2 Prueba dinámica del convertidor reductor en lazo abierto

La prueba dinámica del convertidor reductor en lazo abierto se realizó ingresando al conmutador un vector con datos del ciclo de trabajo (D) que representa a los $V_{ref(simulados)}$. Para obtener los voltajes DC de salida se configura un tiempo de muestreo de 5 [ms], en donde la señal se encuentra estabilizada.

Figura 4.6. Respuesta dinámica del sistema en lazo abierto



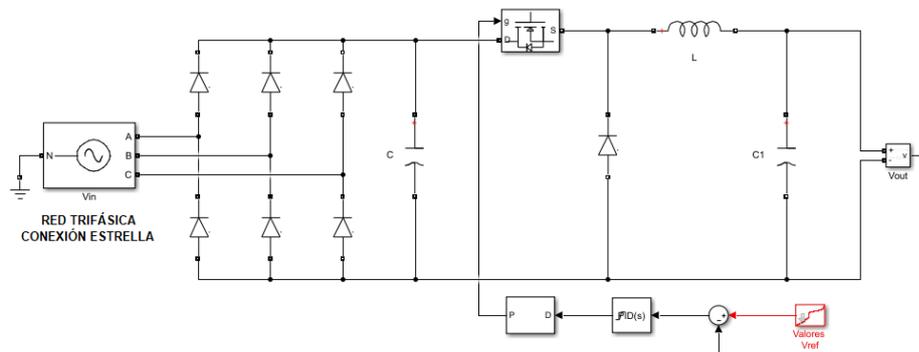
Respuesta dinámica del convertidor reductor en lazo abierto con cambio de ciclo de trabajo comparado con la señal de voltaje experimental. Elaborado por: Geovanny Pullas & Christian Jácome

La gráfica de la Figura 4.6 indica la respuesta del sistema frente a los vectores de tiempo y ciclo de trabajo ingresados al conmutador, los resultados se muestran en Anexos 1. Comparando los voltajes $V_{ref(experimental)}$ con V_o (lazo-abierto), se calcula un error promedio del 12.86% indicando que el sistema en lazo abierto no es el adecuado para esta aplicación.

4.4 PRUEBA DEL CONVERTOR REDUCTOR EN LAZO CERRADO

Para comprobar el funcionamiento del sistema en lazo cerrado se realizan dos pruebas, una estática que consiste en ingresar un setpoint de voltaje de referencia simulado ($V_{ref(simulado)}$), de la tabla de Anexos 1; y otra dinámica ingresando todos los valores $V_{ref(simulado)}$. El diagrama de la Figura 4.7 se empleó para simular las pruebas dinámica y estática.

Figura 4.7. Diagrama del convertidor reductor en lazo cerrado

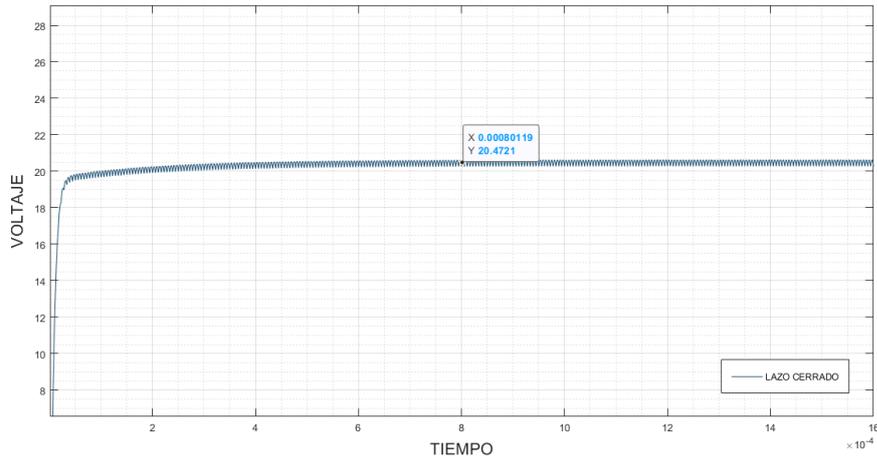


Simulación del convertidor reductor en lazo cerrado con setpoint de voltaje. Elaborado por: Geovanny Pullas & Christian Jácome

4.4.1 Prueba estática del convertidor reductor en lazo cerrado

La prueba estática del convertidor reductor en lazo cerrado se realizó tomando como setpoint para el PID, el valor de $V_{ref(simulado)}$ de 20.5 [V]. Al iniciar la simulación, el compensador reduce el error hasta aproximar el voltaje de salida al valor $V_{ref(simulado)}$. En la Figura 4.7 se muestra que el tiempo de establecimiento de la señal es de 0.8 [ms], no tiene un máximo pico, el rizado de voltaje después del tiempo de establecimiento es de 350 [mV] y el V_o (PID) de 20.28 [V] con un error relativo del 0.2% frente al $V_{ref(experimental)}$ de 20.3 [V].

Figura 4.8. Respuesta estática del sistema en lazo cerrado

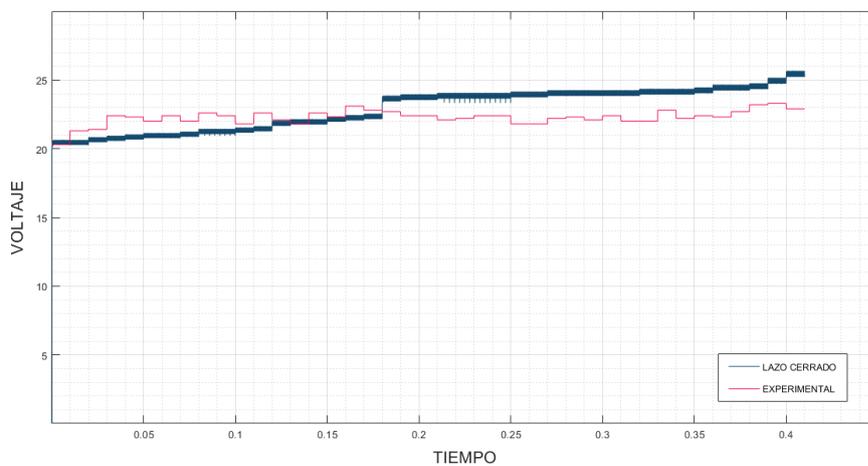


Respuesta del convertor reductor en lazo cerrado con un setpoint de voltaje de 20.5 [V]. Elaborado por: Geovanny Pullas & Christian Jácome

4.4.2 Prueba dinámica del convertor reductor en lazo cerrado

La prueba dinámica del convertor reductor en lazo cerrado se realizó ingresando al setpoint un vector con datos de $V_{ref(simulado)}$. Para obtener los voltajes DC de salida se configura un tiempo de muestreo de 5 [ms], en donde la señal se encuentra estabilizada.

Figura 4.9. Respuesta dinámica del sistema en lazo cerrado



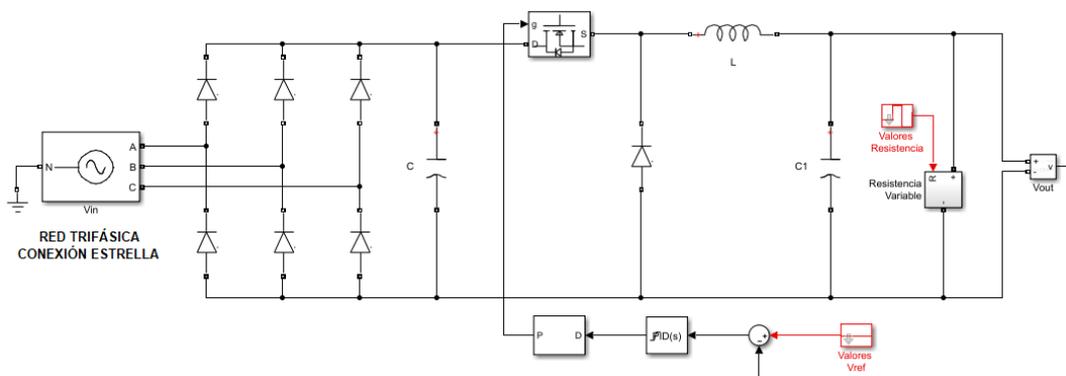
Respuesta dinámica del convertor reductor en lazo cerrado con cambio de setpoint comparado con la señal de voltaje experimental. Elaborado por: Geovanny Pullas & Christian Jácome

La gráfica de la Figura 4.9 indica la respuesta del sistema frente a los vectores de tiempo y setpoint ingresados al PID, los resultados se muestran en Anexos 1. Comparando los voltajes $V_{ref(experimental)}$ con $V_o(PID)$, se calcula un error promedio del 5.83% indicando que el sistema en lazo cerrado es el adecuado para esta aplicación.

4.4.3 Prueba del convertor reductor con carga variable

La prueba del convertor reductor en lazo cerrado con carga variable, se realizó estableciendo el valor de setpoint en 100 [V] el cual se encuentra dentro del rango de voltajes de trabajo del regulador, ingresando un vector con valores de resistencia. El diagrama simulado se muestra en la Figura 4.10.

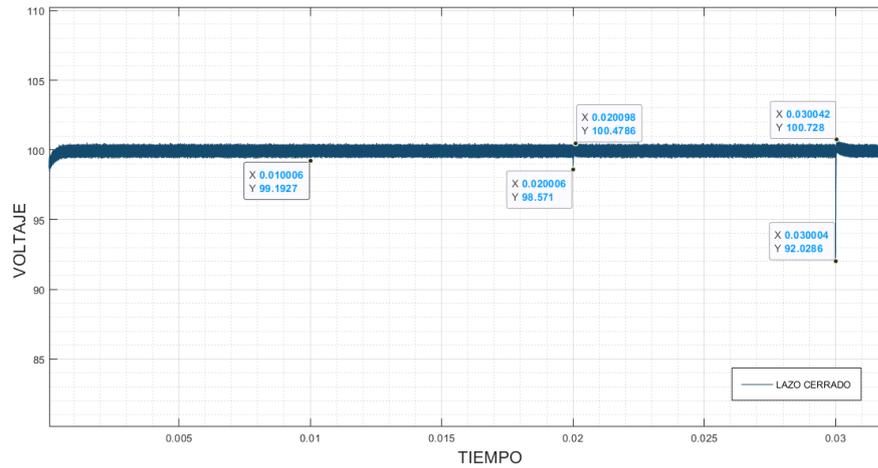
Figura 4.10. Diagrama del convertor reductor en lazo cerrado con carga variable



Respuesta del convertor reductor en lazo cerrado con carga variable. Elaborado por: Geovanny Pullas & Christian Jácome

En la Figura 4.11 se indica por medio de etiquetas la respuesta del controlador frente a cambios de carga en la salida. Al inicio se emplea valores de carga donde se exige al convertor una baja corriente, reduciendo el valor de resistencia hasta conseguir el límite de corriente máxima que se puede obtener del sistema. A medida que el valor de resistencia disminuye, la caída de voltaje es mayor, pero el sistema en lazo cerrado lo compensa inmediatamente y vuelve al valor de setpoint.

Figura 4.11. Respuesta del sistema en lazo cerrado frente a cambios de carga



Respuesta del convertor reductor en lazo cerrado con valor fijo de setpoint de 100 [V] y carga variable. Elaborado por: Geovanny Pullas & Christian Jácome

4.5 COMPARACIÓN ENTRE LAZO ABIERTO Y CERRADO DE LA PRUEBA ESTÁTICA

Recopilando los datos obtenidos en las secciones 4.3.1 y 4.4.1 se ha realizado la Tabla 4.1, la que describe los principales parámetros de las curvas de respuestas del sistema en lazo abierto y cerrado frente a un $V_{ref(simulado)}$ de 20.5 [V].

Tabla 4.1. Parámetros de las curvas de respuesta del sistema lazo abierto y cerrado

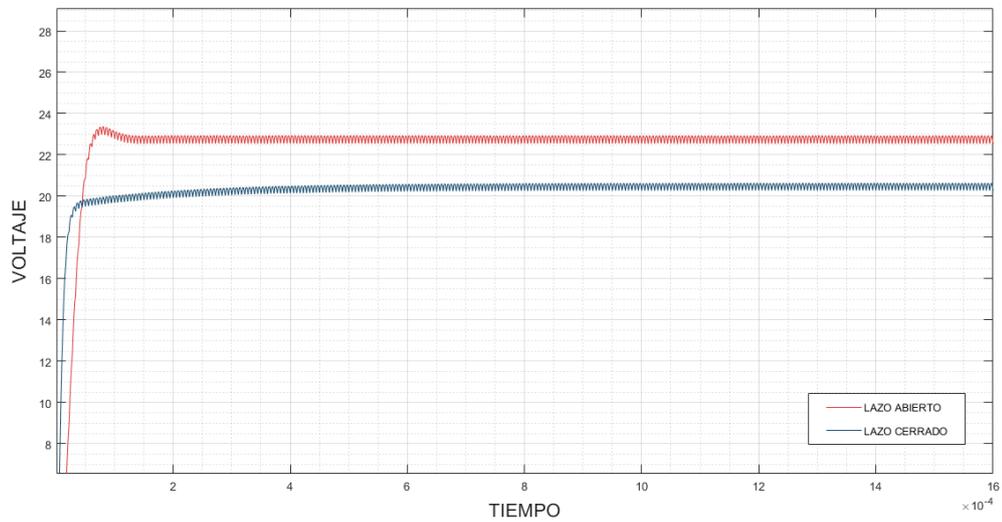
SISTEMA PARÁMETROS	LAZO ABIERTO	LAZO CERRADO
t_s [ms]	1	0.8
M_p [%]	1.03	NO TIENE
Δv [mV]	350	350
V_o [V]	22.6	20.28
$\mathcal{E}V_o$ [%]	11.4	0.2

Valores obtenidos en las simulaciones de lazo abierto y cerrado. Elaborado por: Geovanny Pullas & Christian Jácome

De acuerdo a los datos de la Tabla 4.1 se evidencia que el sistema en lazo cerrado tiene mejor respuesta frente a un valor de referencia dado, deduciendo que este es el más

eficiente para la implementación del simulador, esto se corrobora al visualizar la Figura 4.12 y el valor promedio del error relativo en lazo cerrado, 5.83%, en Anexos 1.

Figura 4.12. Comparación de respuesta de los sistemas en lazo abierto y cerrado



Curvas de los sistemas en lazo abierto y cerrado de Voltaje vs Tiempo. Elaborado por: Geovanny Pullas & Christian Jácome

CONCLUSIONES

La irradiancia solar que recibe un panel fotovoltaico, según su modelo matemático es la de mayor influencia en el valor de corriente de salida del panel, sin embargo el voltaje de salida disminuye abruptamente cuando la corriente en la carga es mayor a la salida del panel FV por lo que el modelo da como resultado un valor numérico complejo para el voltaje de referencia por lo que en la simulación se debe mantener variaciones en la carga que no excedan la corriente de salida del panel.

Por medio del estudio y análisis del modelo matemático FV desarrollado a partir del circuito equivalente de diodo general, en condiciones de prueba estándar, se logra determinar las curvas teóricas características de voltaje vs irradiancia, corriente vs voltaje y potencia vs voltaje, también a cualquier valor de irradiancia comprendido en el rango de 1-1670 [W/m²].

Para seleccionar la topología del convertidor DC-DC a diseñar, se tomó en cuenta el estudio de proyectos realizados con paneles fotovoltaicos, para reducir el consumo de energía en una casa promedio en la ciudad de Quito, 3 [kW], necesitando una alimentación trifásica rectificadora y filtrada, para obtener un voltaje DC de 305 [V], a partir de los cuáles se pueden obtener los 250 [V] como salida máxima del simulador, por esta razón se eligió el convertidor de tipo reductor, ya que cubre todos los valores de voltaje para la simulación de un panel o arreglos fotovoltaicos.

El convertidor reductor es un sistema que no posee error de posición ante variaciones de carga pero esto si afecta a los niveles de rizado de manera proporcional, la variación en el voltaje de alimentación altera el nivel de voltaje entregado, esto hace indispensable la necesidad de utilizar un controlador que haga al sistema inmune a estas perturbaciones, con lo cual se puede reducir el error de 12.86%, a lazo abierto, hasta un error de 5.83%, en lazo cerrado, mejorando la respuesta en estado estable de todo el sistema además de la respuesta dinámica del sistema, haciendo que funcione de manera más adecuada ante variaciones de irradiancia más rápidas con un tiempo de establecimiento de 0.08 [ms], para una corriente de 12 [A].

La comunicación serial bluetooth de datos implementada permitió la visualización del punto de trabajo del regulador de voltaje en tiempo real, eliminando la conexión por cable. El regulador de voltaje es de tipo conmutado y produce interferencia dado que contiene un inductor de alta frecuencia, se puede concluir que este no afecta en nada a la comunicación inalámbrica por lo que es una alternativa acertada.

La HMI está implementada con la capacidad de tomar la información desde un archivo con formato Excel, esto le permite al usuario realizar simulaciones con datos obtenidos desde una estación meteorológica hasta de un día completo, visualizando el punto de trabajo del regulador para los diferentes valores de irradiancia. Terminada o detenida la simulación inmediatamente se crea otro documento Excel, con una matriz que contiene las columnas de irradiancia, voltaje de referencia, voltaje de salida y corriente de salida.

RECOMENDACIONES

La interfaz gráfica fue desarrollada y probada en una PC que contiene disco duro HD y 4GB de memoria RAM. En este equipo se logró una simulación de dinámica aceptable para su validación, sin embargo, se recomienda ejecutar la aplicación del simulador fotovoltaico en una PC de disco sólido SSD con 8GB o 16GB de memoria RAM con la finalidad de disminuir los tiempos de procesamiento para las diferentes funcionalidades del HMI, consiguiendo mayor rapidez en la toma de datos que provienen del convertidor reductor y mayor velocidad de respuesta ante cambios de corriente y voltaje en la carga.

Para la simulación se recomienda utilizar los datos escritos en la etiqueta posterior del panel fotovoltaico que se desea simular, los valores especificados en las hojas de datos pueden variar respecto a los de la etiqueta ya sea porque existen distintas hojas de datos para un mismo modelo de panel, cada una con valores diferentes de los parámetros, o también porque se puede encontrar datos de varios modelos de paneles de diferente potencia en una misma hoja, que contiene una sola gráfica de Corriente vs Voltaje para todos los modelos que detalla.

Para mejorar los resultados del modelo matemático se recomienda incluir la variación de la temperatura de las celdas e ingresar valores de los coeficientes de temperatura tanto para voltaje de circuito abierto como para la corriente de cortocircuito.

REFERENCIAS

- Aguirre, M., & Aman, C. (2017). *Diseño e implementación de un convertidor dc-dc en lazo cerrado que genere curvas I-V/P-V de módulos fotovoltaicos comerciales hasta 100W*. Escuela Politécnica Nacional.
- Almeida, A. (2017). *Diseño e implementación de una fuente de corriente continua de alta potencia*. Escuela Politécnica Nacional.
- Atersa. (2017). Aplicaciones de la energía solar fotovoltaica. Retrieved from <http://www.atersa.es/es/conocenos/aplicaciones-energia-solar-fotovoltaica/>
- Cata, J., & Rodriguez, F. (2015). *Análisis Matemático De Un Panel Solar Fotovoltaico De Silicio*. Retrieved from <http://dspace.ups.edu.ec/handle/123456789/10250>
- Chroma. (2015). *Programmable DC Power Supply Solar Array Simulation, model 62000H-S series*. 8. Retrieved from http://www.chromaate.com/product/62000H_series_Programmable_DC_Power_Supply.htm?fbclid=IwAR1iT6q-6a7ra4E2zMTRVWEoaL6CSC8bjEg1vYkpUw-W-01T_nat4jx8B2s
- Cushicóndor, S. (2019). *Estimación de irradiancia solar basada en modelos matemáticos y medición de variables eléctricas de paneles fotovoltaicos*. Escuela Politécnica Nacional.
- De Souza, V. F., Behr, E. E., Waltrich, G., Martins, D. C., & Coelho, R. F. (2019). Photovoltaic array emulator based on the buck converter. *PEDG 2019 - 2019 IEEE 10th International Symposium on Power Electronics for Distributed Generation Systems*, 843–848. <https://doi.org/10.1109/PEDG.2019.8807464>
- Diaz, S. (2018). *Evaluación del potencial solar para obtención de energía eléctrica en la ciudad de Quito*. Universidad de las Américas.
- Erkaya, Y., Moses, P., Flory, I., & Marsillac, S. (2015). Development of a solar photovoltaic module emulator. *2015 IEEE 42nd Photovoltaic Specialist Conference, PVSC 2015*, 1–3. <https://doi.org/10.1109/PVSC.2015.7356282>
- Fărcaș, C., Ciocan, I., & Tulbure, A. (2018). Solar Emulator for a Photovoltaic Module. *2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging, SIITME 2018 - Proceedings*. <https://doi.org/10.1109/SIITME.2018.8599283>
- Hasnaoui, M., Bennani-Ben Abdelghani, A., & Slama-Belkhodja, I. (2018). Control

- design and experimental set-up for a high-power PV generator emulator. *2018 9th International Renewable Energy Congress, IREC 2018*, (Irec), 1–6. <https://doi.org/10.1109/IREC.2018.8362555>
- Ingemecánica. (2020). Sistema fotovoltaico aislado. Retrieved from <https://ingemecanica.com/tutorialsemanal/tutorialn192.html>
- Macaulay, J., Lin, C. W., & Zhou, Z. (2018). An emulated PV source based on an Indoor Solar Panel with external excitement current and voltage compensation. *SPEEDAM 2018 - Proceedings: International Symposium on Power Electronics, Electrical Drives, Automation and Motion*. <https://doi.org/10.1109/SPEEDAM.2018.8445307>
- Mohan, Ned; Undeland, Tore M; Robbins, W. (2015). *Electrónica de Potencia: Convertidores, aplicaciones y diseño* (Tercera Ed). México, D. F.
- Palaguachi, H. P. M., & Guiracocha, C. V. P. (2017). *Simulación del perfil tensión – corriente para paneles solares mediante convertidor Cc-Cc reductor*. Retrieved from <https://dSPACE.ups.edu.ec/bitstream/123456789/15324/1/UPS-CT007539.pdf>
- ProViento S.A. (2019). Energías Renovables. Retrieved from <https://www.proviento.com.ec/panelesolares.html>
- Richtek. (2019). Criterios de selección del convertidor Buck. Retrieved from [https://www.richtek.com/Design Support/Technical Document/AN033](https://www.richtek.com/Design%20Support/Technical%20Document/AN033)
- Solar, E. (2020). Sistema conectado a la red eléctrica. Retrieved from <http://www.energiasolar.lat/2020/>
- Tapfumanei, T. B. (2016). *Solar Array Emulator*. Stellenbosch.
- Vargas, E., Pérez, J., Cota, J., & Castellanos, M. (2015). Control de un convertidor CD-CD tipo buck por medio del microcontrolador ATMEGA328. *Cultura Científica y Tecnológica*, 0(55).
- Yung, C. (2016, March). ¿En una fuente de poder de C.C. cuánto rizado es demasiado? *Electrical Apparatus Service Association, Inc*. Retrieved from www.easa.com

ANEXOS

Anexo 1: TABLA DE CÁLCULO DE ERRORES EXPERIMENTALES Y SIMULADOS EN LAZO ABIERTO Y CERRADO

$G_{\text{(experimental)}}$ [W/m ²]	EGE-175M-36	MODELO MATEMÁTICO		REGULADOR DE VOLTAJE				
	$V_{\text{ref(experimental)}}$ [V]	$V_{\text{ref(simulado)}}$ [V]	$\epsilon V_{\text{ref(exp-sim)}}$ [%]	D	V_o (lazo-abierto) [V]	ϵV_o (Lazo-abierto) [%]	V_o (PID) [V]	ϵV_o (PID) [%]
91	20,3	20,5	1,00	0,066	22,6	11,4	20,3	0,2
117	21,3	20,5	3,53	0,066	22,6	5,9	20,3	4,7
157	21,4	20,7	3,29	0,067	22,9	6,9	20,5	4,2
196	22,4	20,8	7,10	0,067	22,9	1,9	20,6	8,2
220	22,3	20,9	6,07	0,067	22,9	2,7	20,7	7,2
232	22,0	21,0	4,68	0,067	22,9	3,9	20,8	5,5
243	22,4	21,0	6,12	0,068	23,2	3,6	20,8	7,1
266	22,0	21,1	4,11	0,068	23,2	5,5	20,9	5,1
313	22,6	21,3	5,81	0,068	23,2	2,8	21,1	6,6
329	22,4	21,3	4,79	0,069	23,5	5,0	21,1	5,9
355	21,8	21,4	1,73	0,069	23,5	7,9	21,2	2,8
370	22,6	21,5	4,96	0,069	23,5	4,1	21,3	5,8
481	22,1	21,9	0,97	0,070	23,9	8,0	21,7	1,9
500	21,8	22,0	0,71	0,071	24,2	10,9	21,8	0,1
507	22,6	22,0	2,69	0,071	24,2	7,1	21,8	3,6
567	22,3	22,2	0,62	0,071	24,2	8,3	22,0	1,6
589	23,1	22,3	3,54	0,072	24,5	6,2	22,1	4,4
625	22,8	22,4	1,69	0,072	24,5	7,6	22,2	2,7
975	22,7	23,7	4,41	0,076	25,8	13,8	23,5	3,4
1003	22,4	23,8	6,13	0,077	26,2	16,7	23,6	5,1
1005	22,4	23,8	6,30	0,077	26,2	16,8	23,6	5,2
1017	22,1	23,9	7,80	0,077	26,2	18,3	23,7	6,9
1030	22,2	23,9	7,68	0,077	26,2	17,9	23,7	6,7
1042	22,4	23,9	6,91	0,077	26,2	16,8	23,7	5,7
1042	22,4	23,9	6,91	0,077	26,2	16,9	23,7	5,7
1043	21,8	24,0	9,87	0,077	26,2	20,1	23,8	9,0
1058	21,8	24,0	10,12	0,077	26,2	20,1	23,8	9,0
1070	22,2	24,1	8,34	0,077	26,2	18,0	23,9	7,5
1078	22,3	24,1	7,98	0,077	26,2	17,4	23,9	7,0
1080	22,1	24,1	9,24	0,077	26,2	18,7	23,9	8,2
1085	22,4	24,1	7,62	0,078	26,5	18,3	23,9	6,5
1090	22,0	24,1	9,66	0,078	26,5	20,5	23,9	8,5
1110	22,0	24,2	9,99	0,078	26,5	20,5	24,0	8,9
1115	22,8	24,2	6,45	0,078	26,5	16,5	24,0	5,3
1122	22,2	24,2	9,20	0,078	26,5	19,4	24,0	7,9
1135	22,4	24,3	8,44	0,078	26,5	18,3	24,1	7,4
1190	22,3	24,5	9,83	0,079	26,8	20,4	24,3	8,8
1190	22,7	24,5	7,90	0,079	26,9	18,3	24,3	6,9
1221	23,2	24,6	6,20	0,079	26,8	15,8	24,4	5,1
1337	23,3	25,0	7,53	0,080	27,2	16,7	24,8	6,4
1470	22,9	25,5	11,45	0,082	27,8	21,5	25,3	10,3

Promedio de errores relativos	6,08
--------------------------------------	------

12,86

5,83

Anexo 2: CÓDIGO DE PROGRAMACIÓN DEL MODELO MATEMÁTICO DEL PANEL FOTOVOLTAICO

```

% XXXXXXXXXXXXXXXXXXXX  MODELO MATEMATICO PV  XXXXXXXXXXXXXXXXXXXXXXXXXXXX

% Parámetros

G1=str2double (get(handles.in_G_fija,'string'));% G
ingresada

% obtener Rs y Rp

Ti=Imp1/Iscn1;           % Tasa de cambio de corriente
Tv=Vmp1/Vocn1;          % Tasa de cambio de voltaje
dT=abs (T-Tn);          % Variacion de temperatura
Isc= (Iscn1+Ki*dT)*G1/Gn; % Corriente Isc a irradiacion
G1

Imp= Isc*Ti;             % Actualizacion de Imp
Iscn=Isc;                % Actualizacion de Iscn
Vmp= Vmp1-((Imp1-Imp)/m); % Calculo de Vmp a Imp de
limite

Vocn=Vmp/Tv;            % max de corriente
                        % Actualizacion de Vocn

% Datos obtenidos del fabricante del PV

Np=1;                    % Numero de celdas en paralelo
Pmax_e=Vmp*Imp;          % Potencia maxima nominal

% Corriente nominal de saturacion del diodo

Ion=Iscn/(exp (Vocn/a/Ns/Vtn)-1);
Io=Ion;

% Valores referenciales de Rs y Rp

Rs_max=(Vocn-Vmp)/Imp;
Rp_min= Vmp/(Iscn-Imp)-Rs_max;

% Valores iniciales de Rs y Rp

Rp=Rp_min;
Rs=0;
tol=0.001;                % Error de tolerancia de potencia
P=0;                       % inicialización del vector Potencia
'P'

error=Inf;

%Proceso iterativo para Rs y Rp hasta que Pmax,modelo=Pmax,exp

while (error>tol)
    dT=abs (T-Tn);
    Ipvn= (Rs+Rp)/Rp*Iscn; % Iscn = Isc de datasheet
    Ipv= (Ipvn+Ki*dT)*G/Gn; % Corriente fotogenerada
    Isc= (Iscn+Ki*dT)*G/Gn; % Isc para una determinada G

```

```

        %Resistencia en paralelo
        Rp=Vmp*(Vmp+Imp*Rs)/(Vmp*Ipv-
Vmp*Io*exp((Vmp+Imp*Rs)/Vt/Ns/a)+Vmp*Io-Pmax_e);

        %Resolver la ecuacion I-V para diferentes pares (V,I)
        clear V % Se limpia vector V de
voltajes
        clear I % Se limpia vector I de
corrientes

        V=0:.1:50; % Vector de tensiones
        I=zeros(1,size(V,2)); % Vector de corriente

        for j=1:size(V,2) % Se calcula para todos los
voltajes
            g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);

            while (abs(g(j))>tol)
                g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);
                glin(j)=-Io*Rs/Vt/Ns/a*exp((V(j)+I(j)*Rs)/Vt/Ns/a)-Rs/Rp-1;
                I_(j)=I(j)-g(j)/glin(j); %Newton Raphson
                I(j)=I_(j);
            end
        end
        Pmax_m= max(P);
        error=(Pmax_m-Pmax_e);

        %Incremento de Rs
        Rs=Rs+0.001; %Pasos de aumento de la resistencia
    end

```

Anexo 3: CÓDIGO DE PROGRAMACIÓN DE LA INTERFAZ GRÁFICA DEL PANEL FOTOVOLTAICO

```
function varargout = Emulador_Panel_Fotovoltaico(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Emulador_Panel_Fotovoltaico_OpeningFcn, ...
                  'gui_OutputFcn',  @Emulador_Panel_Fotovoltaico_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Emulador_Panel_Fotovoltaico_OpeningFcn(hObject, eventdata,
handles, varargin)

movegui(hObject, 'center'), %..... centra la
ventana en la pantalla
set(handles.axes1, 'Visible', 'on'); %..... Axes1 activo
set(handles.axes3, 'Visible', 'on'); %..... Axes3 visible
set(handles.radBtn_Aux, 'Value', 1); %..... radBtn_aux'
activo
set(handles.radBtn_Aux, 'Visible', 'off'); %..... radBtn_aux'
oculto
set(handles.BaseDeDatosPaneles, 'Enable', 'off'); %.....
BaseDeDatosPaneles inhabilitado
set(handles.in_PanelesEnSerie, 'Enable', 'off'); %.....
in_PanelesEnSerie Inhabilitado
set(handles.in_PanelesEnParalelo, 'Enable', 'off'); %.....
in_PanelesEnParelelo Inhabilitado
set(handles.radBtn_SimuPanel, 'Value', 1); %..... En Tipo de
simulación por default aparece la opción 'Panel' activo
set(handles.btnGroup_G_fija, 'Enable', 'off'); %..... radButton G
Fija Inhabilitado
set(handles.btnGroup_G_variable, 'Enable', 'off'); %..... radButton G
Variable Inhabilitado

set(handles.btn_Play, 'Enable', 'off'); %..... Button PLAY
Inhabilitado
set(handles.btn_Stop, 'Enable', 'off'); %..... Button STOP
Inhabilitado
set(handles.btn_LeerArchivo, 'Enable', 'off'); %..... Button LEER
ARCHIVO Inhabilitado
set(handles.btn_Reset, 'Enable', 'off'); %..... Button RESET
Inhabilitado
set(handles.tglBtn_Comunicacion, 'Enable', 'off'); %..... Button
COMUNICACION
```

```

axes(handles.axes4)%..... selecciona
axes4
circuito=imread('cell_circuito.jpg');%..... lee la imagen
imshow(circuito);%..... captura la
imagen que va amostrar
set(handles.axes4, 'UserData', circuito);%..... aplica la
imagen al axes

axes(handles.axes5)%..... selecciona
axes4
com=imread('Bluetooth.jpg');%..... lee la imagen
imshow(com);%..... captura la
imagen que va amostrar
set(handles.axes5, 'UserData', com);%..... muestra la
imagen

handles.output = hObject;
guidata(hObject, handles);

function varargout = Emulador_Panel_Fotovoltaico_OutputFcn(hObject,
eventdata, handles)
% Get default command line output from handles structure
varargout{1} = handles.output;

function in_Voc_Callback(hObject, eventdata, handles)
function in_Voc_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function in_Isc_Callback(hObject, eventdata, handles)
function in_Isc_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function in_Vmp_Callback(hObject, eventdata, handles)
function in_Vmp_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function in_Imp_Callback(hObject, eventdata, handles)
function in_Imp_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function in_NumeroCeldasPanel_Callback(hObject, eventdata, handles)
function in_NumeroCeldasPanel_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

end

function in_PanelesEnSerie_Callback(hObject, eventdata, handles)
function in_PanelesEnSerie_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function in_PanelesEnParalelo_Callback(hObject, eventdata, handles)
function in_PanelesEnParalelo_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function in_G_fija_Callback(hObject, eventdata, handles)
function in_G_fija_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function radBtn_Manual_Callback(hObject, eventdata, handles)

    set(handles.in_Voc, 'Enable', 'on', 'String', '');
    set(handles.in_Isc, 'Enable', 'on', 'String', '');
    set(handles.in_Vmp, 'Enable', 'on', 'String', '');
    set(handles.in_Imp, 'Enable', 'on', 'String', '');
    set(handles.in_NumeroCeldasPanel, 'Enable', 'on', 'String', '');
    set(handles.in_PanelesEnSerie, 'Enable', 'off');
    set(handles.in_PanelesEnParalelo, 'Enable', 'off');
    set(handles.BaseDeDatosPaneles, 'Enable', 'off');

function radBtn_BaseDeDatos_Callback(hObject, eventdata, handles)

    set(handles.BaseDeDatosPaneles, 'Enable', 'on');

function BaseDeDatosPaneles_Callback(hObject, eventdata, handles)

    global m
    lista=get(hObject, 'String');%....returns selected item from
BaseDeDatosPaneles
    n=get(hObject, 'Value');% ..... Obtiene el valor del objeto
BaseDePaneles
    ModeloPanel=lista(n);%..... Asigna el valor n de la
lista
    switch cell2mat(ModeloPanel)%....Convierte el dato tipo
celda a matricial
        case'01 12V GP-050P-36' %1 OK
            m=1;% % Modelo de panel
            set(handles.in_Voc, 'String', '22.2');
            set(handles.in_Isc, 'String', '2.90');
            set(handles.in_Vmp, 'String', '18.3');
            set(handles.in_Imp, 'String', '2.73');

set(handles.in_NumeroCeldasPanel, 'String', '36');

```

```

        case'02    12V  EGE-175M-36' % Modelo de panel
            m=2.9;% OK
            set(handles.in_Voc,'String','23.79');
            set(handles.in_Isc,'String','9.25');
            set(handles.in_Vmp,'String','19.77');
            set(handles.in_Imp,'String','8.85');

set(handles.in_NumeroCeldasPanel,'String','36');

        case'03    24V  EGE-255P-60'% Modelo de panel
            m=2.3;%OK
            set(handles.in_Voc,'String','37.82');
            set(handles.in_Isc,'String','8.76');
            set(handles.in_Vmp,'String','30.77');
            set(handles.in_Imp,'String','8.29');

set(handles.in_NumeroCeldasPanel,'String','60');

        case'04    24V  EGE-270P-60'% Modelo de panel
            m=2.3;%OK
            set(handles.in_Voc,'String','38.45');
            set(handles.in_Isc,'String','9.03');
            set(handles.in_Vmp,'String','31.44');
            set(handles.in_Imp,'String','8.59');

set(handles.in_NumeroCeldasPanel,'String','60');

    end

function BaseDeDatosPaneles_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function btnGroup_Irradiancia_SelectionChangedFcn(hObject,
eventdata, handles)
    global Resul_Vref_1 G_Fija_Usuario
    if (hObject == handles.btnGroup_G_fija)

        set(handles.in_G_fija,'Enable','on');
        set(handles.btn_LeerArchivo,'Enable','off');
        set(handles.btn_Play,'Enable','on');
    else % (hObject == btnGroup_G_variable)
        set(handles.in_G_fija,'Enable','off');
        set(handles.btn_LeerArchivo,'Enable','on');
        set(handles.btn_Play,'Enable','off');
        set(handles.in_G_fija,'Enable','off','String','');
    end

function btn_CargarDatos_Callback(hObject, eventdata, handles)

    global GQuito VG_IoutCero m GmaxQuito
    GmaxQuito=1670;% Variable que representa la irradiancia
maxima de Quito
    % Constantes fisicas

```

```

        a=1.3;%..... Factor de idealidad
inicial
        k=1.3806503e-23;%..... Constante de Boltzmann
        q=1.60217646e-19;%..... Carga del electrón
        Kv=-0.080;%..... Coeficiente de
temperatura de voltaje
        Ki= 0.65e-3;%..... Coeficiente de
temperatura de Isc

        %Valores nominales
        Gn=1000;%..... Irradiancia nominal a
25°C en [W/m^2] 'STC'
        Tn=298.15;%..... Temperatura nominal en
[K] Tn=25+273.15 'STC'

        %Algoritmo de ajuste a la condicion nominal
        G=Gn;
        T=Tn;
        Vtn=k*Tn/q;%..... %Voltaje termico
nominal
        Vt=k*T/q;%..... %Voltaje termico a la
temperatura T

        %===== INGRESO DE DATOS POR GUIDE
        %=====
        Vocn1=str2double(get(handles.in_Voc,'string'));
%.....ingreso Voc de datasheet
        Iscn1=str2double(get(handles.in_Isc,'string'));
%.....ingreso Isc de datasheet
        Vmp1=str2double(get(handles.in_Vmp,'string'));
%.....ingreso Vmp de datasheet
        Imp1=str2double(get(handles.in_Imp,'string'));
%.....ingreso Imp de datasheet
        Ns=str2double(get(handles.in_NumeroCeldasPanel,'string'));
%.....ingreso Número de celdas de datasheet
        Nsa=str2double(get(handles.in_PanelesEnSerie,'string'));
%.....ingreso número de paneles en serie (arreglo)

Npa=str2double(get(handles.in_PanelesEnParalelo,'string'));%.....i
ngreso número de paneles en paralelo (arreglo)
        G1=1000;
        %===== FIN Ingreso de datos por guide
        %=====

        %//////// CORRECCIÓN DE ERRORES DE DATOS INGRESADOS
        ////////////////////////////////////////////
        ////////////////////////////////////////////

        if isempty(Vocn1)||isnan(Vocn1)%
|
|           errordlg('El campo < Voltaje de Circuito Abierto > está
vacío o el dato ingresado no es un número','CORREGIR') % |
|           return %
|
|           end %
|

        if isempty(Iscn1)||isnan(Iscn1)

```

```

        errordlg('El campo < Corriente de Corto Circuito> está
vacío o,el dato ingresado no es un valor numérico','CORREGIR')
        return
    end
    if isempty(Vmp1)||isnan(Vmp1)
        errordlg('El campo < Voltaje de Máxima Potencia > está
vacío o el dato ingresado no es un valor numérico','CORREGIR')
        return
    end
    if isempty(Imp1)||isnan(Imp1)
        errordlg('El campo < Corriente de Máxima Potencia > está
vacío o el dato ingresado no es un valor numérico','CORREGIR')
        return
    end

    if(get(handles.btnGroup_G_fija,'Value')==1)
        if isempty(G1)||isnan(G1)
            errordlg('El campo < Irradiancia Fija > está vacío
o el dato ingresado no es un valor numérico entero','CORREGIR')
            return
        end
        if (G1<1||G1>GmaxQuito)
            errordlg('En el campo < Irradiancia Fija > sólo se
puede ingresar un valor numérico entero en el rango de 1 a 1670
W/m2)','CORREGIR')
            return
        end
    end

    if(get(handles.radBtn_SimuArreglo,'Value')==1)
        if isempty(Nsa)||isnan(Nsa)||Nsa<1
            errordlg('ARREGLO DE PANELES: En el campo < Número
de paneles en serie > debe ingresar un número entero mayor o igual a
1','CORREGIR')
            return
        end
        if (isempty(Npa)||isnan(Npa)||Npa<1)
            errordlg('ARREGLO DE PANELES: En el campo < Número
de paneles en paralelo > debe ingresar un número entero mayor o
igual a 1','CORREGIR')
            return
        end
        if(Nsa*Vocn1)>250
            errordlg('El número de paneles en serie excede la
capacidad del convertor DC-DC (250vdc),por favor disminuya el
número de paneles en serie','CORREGIR')
            return
        end
        if(Npa*Iscn1)>12
            errordlg('El número de paneles en paralelo excede
la capacidad del convertor DC-DC (12A),por favor disminuya el número
de paneles en paralelo','CORREGIR')
            return
        end
    end

    if(Vocn1<= Vmp1)

```



```

        %Resistencia en paralelo
        Rp=Vmp* (Vmp+Imp*Rs) / (Vmp*Ipv-
Vmp*Io*exp( (Vmp+Imp*Rs) /Vt/Ns/a)+Vmp*Io-Pmax_e);

        %Resolver la ecuacion I-V para diferentes pares (V,I)
        clear V % Se limpia vector V de
voltajes
        clear I % Se limpia vector I de
corrientes

        V=0:.1:50; % Vector de tensiones
        I=zeros(1,size(V,2)); % Vector de corriente

        for j=1:size(V,2) % Se calcula para todos los
voltajes
            g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);

            while (abs(g(j))>tol)
                g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);
                glin(j)=-Io*Rs/Vt/Ns/a*exp((V(j)+I(j)*Rs)/Vt/Ns/a)-Rs/Rp-1;
                I_(j)=I(j)-g(j)/glin(j); %Newton Raphson
                I(j)=I_(j);
            end
        end
        Pmax_m= max(P);% Se toma el máximo valor del vector
Potencia.(P)
        error=(Pmax_m-Pmax_e);

        %Incremento de Rs
        Rs=Rs+0.001;
    end

    GQuito= 1:GmaxQuito; % Vector irradiancia de Quito
de 1 a 1670
    VG_IoutCero = 1:GmaxQuito; % Vector voltaje vs
Irradiancia con Iout=0[A]

    for i=1:1:GmaxQuito
        % Parámetros cuando la irradiancia es diferente de 1000
        G1=i; % Irradiancia itera de 1 hasta
1300
        Ti=Imp1/Iscn1; % Tasa de cambio de corriente
        Tv=Vmp1/Vocn1; % Tasa de cambio de voltaje
        dT=abs(T-Tn); % Variacion de temperatura
        Isc= (Iscn1+Ki*dT)*G1/Gn; % Corriente Isc a irradiacion
G1
        Imp= Isc*Ti; % Actualizacion de Imp
        Iscn=Isc; % Actualizacion de Iscn
        Ipvn= (Rs+Rp) /Rp*Iscn; % Iscn = Isc de datasheet
        Ipv= (Ipvn+Ki*dT)*G1/Gn; % Corriente fotogenerada
        Vmp= Vmp1-((Imp1-Imp)/m); % Calculo de Vmp a Imp de
limite max de corriente
        Vocn=Vmp/Tv; % Actualizacion de Vocn
        Iout=0; % Iout=0 para cálculo de
voltaje en condiciones de circuito abierto
    end

```

```

        % Vout = Ns*Vt*log((Ipv-Iout)/Imp)+Vmp-Rs*Iout    % formula
general
        % Vout (del modelo matematico es decir el Set Point que va
al PID

        VoutConIoutCero = Ns*Vt*log((Ipv-Iout)/Imp)+Vmp-Rs*Iout; %
        if get(handles.radBtn_SimuPanel,'Value')==1
            VG_IoutCero(i)=VoutConIoutCero;% Se ingresa en el
vector el voltaje calculado
        end
        if get(handles.radBtn_SimuArreglo,'Value')==1
            VG_IoutCero(i)=VoutConIoutCero*Nsa;% Se ingresa en
el vector el voltaje calculado
        end
    end

    % FIN Parámetros cuando la irradiancia es diferente de 1000

    % ////////////////////////////////// GRÁFICA VOLTAJE vs IRRADIANCIA
////////////////////////////////////
        if get(handles.radBtn_SimuArreglo,'Value')==1
            Vocn1=Vocn1*Nsa;% Cuando se simula arreglo Vocn*Nro de
paneles en serie/para la escala de la gráfica
        end

        axes(handles.axes1);%.Se selecciona axes1
        grid on

set(handles.axes1,'XMinorGrid','on','YMinorGrid','on');%...Activa
subcuadrícula al axes1
    hold on
    xlabel('I R R A D I A N C I A [ W/m² ]','FontSize',10)%
    xlim([0 1700])% Límites eje 'Y' de Voltaje (eje 'Y'derecho)
    ylabel('V O L T A J E [ V ]','FontSize',10)%
    ylim([0 1.1* Vocn1])%. Límites eje 'Y' de Voltaje (eje
'Y'derecho)
    plot(GQuitto,VG_IoutCero,'r.','LineWidth',0.1)% Gráfica G-V ,
'r'=rojo ,AnchoDeLinea= 1

    % ////////////////////////////////// FIN GRÁFICA VOLTAJE vs IRRADIANCIA
////////////////////////////////////

    %===== INGRESO DE DATOS POR GUIDE
%=====
        set(handles.in_Voc,'Enable','off'); %..... ingreso Voc
de datasheet
        set(handles.in_Isc,'Enable','off'); %..... ingreso Isc
de datasheet
        set(handles.in_Vmp,'Enable','off'); %..... ingreso Vmp
de datasheet
        set(handles.in_Imp,'Enable','off'); %..... ingreso Imp
de datasheet
        set(handles.in_NumeroCeldasPanel,'Enable','off');% ingreso
Número de celdas de datasheet
        set(handles.in_PanelesEnSerie,'Enable','off'); %... ingreso
número de paneles en serie (arreglo)
        set(handles.in_PanelesEnParalelo,'Enable','off');%. ingreso
número de paneles en paralelo (arreglo)
        set(handles.radBtn_Manual,'Enable','off');%..... se
inhabilita ingreso de datos manual

```

```

        set(handles.radBtn_BaseDeDatos,'Enable','off');%... se
inhabilita ingreso de datos Base de datos
        set(handles.BaseDeDatosPaneles,'Enable','off');%... se
inhabilita BaseDeDatosPaneles
        set(handles.radBtn_SimuPanel,'Enable','off');%..... se
inhabilita Tipo de simulacion panel
        set(handles.radBtn_SimuArreglo,'Enable','off');%... se
inhabilita Tipo de simulacion arreglo
        set(handles.tglBtn_Comunicacion,'Enable','on');%... se habilita
boton CONECTAR

```

```

%===== FIN Ingreso de datos por guide
=====

```

```

%===== TIPO DE SIMULACIÓN: 1PANEL o ARREGLO DE PANELES %
=====

```

```

function btnGrp_TipoSimulacion_SelectionChangedFcn(hObject,
eventdata, handles)
    if (hObject==handles.radBtn_SimuPanel)
        set(handles.in_PanelesEnSerie,'String','','Enable','off');%
Se inhabilita el campo número de paneles en serie

```

```

set(handles.in_PanelesEnParalelo,'String','','Enable','off');% Se
inhabilita el campo número de paneles en paralelo

```

```

    else
        set(handles.in_PanelesEnSerie,'Enable','on');% Se habilita
el campo número de paneles en serie
        set(handles.in_PanelesEnParalelo,'Enable','on');% Se
habilita el campo número de paneles en paralelo
    end

```

```

%===== FIN: TIPO DE SIMULACIÓN: 1PANEL o ARREGLO DE PANELES %
=====

```

```

function btn_LeerArchivo_Callback(hObject, eventdata, handles)
    global DatosG_Archivo_irradiancia
    DatosG_Archivo_irradiancia=
xlsread('C:\Emulador\irradiancia');%lee todos los datos del archivo
irradiancia

```

```

    set(handles.btn_Play,'Enable','on');

```

```

% ..... toggle button

```

```

CONECTAR/DESCONECTAR.....

```

```

function tglBtn_Comunicacion_Callback(hObject, eventdata, handles)
    global obj1

```

```

        set(handles.btn_CargarDatos,'Enable','off');%... se habilita
botón 'CARGAR DATOS'

```

```

        set(handles.btnGroup_G_variable,'Enable','on');% se habilita
botón 'G VARIABLE'

```

```

        set(handles.btnGroup_G_fija,'Enable','on');%.... se habilita
botón 'G FIJA'

```

```

        set(handles.btn_Play,'Enable','on');%..... se habilita
botón 'PLAY'

```

```

        set(handles.btn_Stop,'Enable','on');%..... se habilita
botón 'STOP'

```

```

        a=get(hObject,'Value');% Asigna a la variable 'a'el estado de
tglBtn_Comunicacion OK

```

```

if a==1 % consulta si se ha pulsado el botón 'CONECTAR'
    %.. Conexión de instrumento
    %.. Encuentra un objeto de conexión Bluetooth.
    set(handles.tglBtn_Comunicacion,'String','DESCONECTAR');%

    obj1 = instrfind('Type','bluetooth','Name','Bluetooth-
Simulador-Panel-Fotovoltaico:1','Tag','');
    if isempty(obj1)%isempty=1 Si el objeto de conexión
Bluetooth NO EXISTE
        obj1 = Bluetooth('Simulador-Panel-Fotovoltaico',
1);%Si no existe crea al objeto

        else % caso contrario, USA EL OBJETO QUE ENCONTRÓ.
            fclose(obj1);
            obj1 = obj1(1);
        end
        %.. Connect to instrument object, obj1.
        fopen(obj1);

else
    %.. Disconnect and Clean Up
    %.. Desconectarse del objeto del instrumento, obj1.
    set(handles.tglBtn_Comunicacion,'String','CONECTAR');
    set(handles.btn_Play,'Enable','off');
    set(handles.btn_Stop,'Enable','off');
    fclose(obj1);
end

%ssssssssssssssssssssssssssssssss ...SIMULACION STOP...
ssssssssssssssssssssssssssssssss
function btn_Stop_Callback(hObject, eventdata, handles)
    global stop
    stop=1;
    set(handles.tglBtn_Comunicacion,'Enable','on')
    set(handles.btn_Reset,'Enable','on');%.....
    set(handles.btn_Play,'Enable','off')
    set(handles.btnGroup_G_fija,'Enable','off')
    set(handles.btnGroup_G_variable,'Enable','off')
    set(handles.btn_Play,'Enable','off')

%ssssssssssssssssssssssssssssssss ... FIN SIMULACION STOP...
ssssssssssssssssssssssssssssssss

%XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.... SIMULACION
PLAY.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
function btn_Play_Callback(hObject, eventdata, handles)

    set(handles.tglBtn_Comunicacion,'Enable','off')
    global stop Iout1 obj1 VectorDatosG GQuito VG_IoutCero m
GmaxQuito
    global Resul_Vref_1
    stop=0;
    cont=0;
    G_anterior=0

    a=1.3;                %Factor de idealidad inicial

```

```

k=1.3806503e-23;      %Constante de Boltzmann
q=1.60217646e-19;    %Carga del electrón
Kv=-0.080;           %Coeficiente de temperatura de voltaje
Ki= 0.65e-3;         %Coeficiente de temperatura de Isc
%Valores nominales
Gn=1000;              %Irradiancia nominal a 25°C en [W/m^2]
'STC'
Tn=298.15;           %Temperatura nominal en [K]
Tn=25+273.15 'STC'
%Algoritmo de ajuste a la condicion nominal
G=Gn;
T=Tn      % se debe igualar a Tn
Vtn=k*Tn/q;      %Voltaje termico nominal
Vt=k*T/q;        %Voltaje termico a la temperatura T

%===== INGRESO DE DATOS POR GUIDE =====
Vocn1=str2double(get(handles.in_Voc,'string')); %
Iscn1=str2double(get(handles.in_Isc,'string'));%
Vmp1=str2double(get(handles.in_Vmp,'string'));%
Imp1=str2double(get(handles.in_Imp,'string'));%
Ns=str2double(get(handles.in_NumeroCeldasPanel,'string'));%
Nsa=str2double(get(handles.in_PanelesEnSerie,'string'));%
Npa=str2double(get(handles.in_PanelesEnParalelo,'string'));%
G1=str2double(get(handles.in_G_fija,'string'));% Gfija ingresada
%===== FIN Ingreso de datos por guide =====|

%***** Corrección de errores en ingreso de datos
%*****
% isempty es =1 :Cuando la variable no tiene dato es decir está
vacía
% isnan es =1 :Cuando el dato no es un número;(podría ser que
el usuario ingresó texto.
|

if(get(handles.btnGroup_G_fija,'Value')==1)
    if isempty(G1)||isnan(G1)
        errordlg('El campo < Irradiancia Fija > está vacío o el dato
ingresado no es un valor numérico entero','CORREGIR')
        return
    end
    if(G1<1||G1>GmaxQuito)
        errordlg('En el campo < Irradiancia Fija > Por favor ingrese
un valor dentro del rango 1 a 1670 W/m² . Ingrese un valor dentro
del rango'),'CORREGIR')
        return
    end
end

%***** FIN Corrección de errores en ingreso de datos
%*****

TipoDeSimulacion=get(handles.radBtn_SimuPanel,'Value');
TipoDeIrradiancia=get(handles.btnGroup_G_fija,'Value')

%CASO1: TIPO DE SIMULACIÓN: 1 PANEL - IRRADIANCIA: G_FIJA

```

```

while (TipoDeSimulacion==1) && (TipoDeIrradiancia==1) && (stop==0)

    % XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX MODELO PV
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

    G1=str2double(get(handles.in_G_fija,'string'))% G1=Dato de
    irradiancia

    if(G1~=G_anterior)
        G_anterior=G1;
        Ti=Imp1/Iscn1; % Tasa de cambio de corriente
        Tv=Vmp1/Vocn1; % Tasa de cambio de voltaje
        dT=abs(T-Tn); % Variacion de temperatura
        Isc= (Iscn1+Ki*dT)*G1/Gn; % Corriente Isc a irradiacion G1
        Imp= Isc*Ti; % Actualizacion de Imp
        Iscn=Isc; % Actualizacion de Iscn
        Vmp= Vmp1-((Imp1-Imp)/m); % Calculo de Vmp a Imp de limite
max de corriente
        Vocn=Vmp/Tv; % Actualizacion de Vocn

    % Datos obtenidos del fabricante del PV
        Np=1; %Numero de celdas en paralelo
        Pmax_e=Vmp*Imp; %Potencia maxima nominal

    % Corriente nominal de saturacion del diodo
        Ion=Iscn/(exp(Vocn/a/Ns/Vtn)-1);
        Io=Ion;

    % Valores referenciales de Rs y Rp
        Rs_max=(Vocn-Vmp)/Imp;
        Rp_min= Vmp/(Iscn-Imp)-Rs_max;

    % Valores iniciales de Rs y Rp
        Rp=Rp_min;
        Rs=0;

        tol=0.001; %Error de tolerancia de potencia
        P=0; % inicialización del vector Potencia 'P'
        error=Inf;

    %Proceso iterativo para Rs y Rp hasta que Pmax,modelo=Pmax,exp
        while(error>tol)
            dT=abs(T-Tn);
            Ipvn=(Rs+Rp)/Rp*Iscn; % Iscn = Isc de datasheet
            Ipv=(Ipvn+Ki*dT)*G/Gn; % Corriente fotogenerada
            Isc=(Iscn+Ki*dT)*G/Gn; % Isc para una determinada G

            %Resistencia en paralelo
            Rp=Vmp*(Vmp+Imp*Rs)/(Vmp*Ipv-
Vmp*Io*exp((Vmp+Imp*Rs)/Vt/Ns/a)+Vmp*Io-Pmax_e);

            %Resolver la ecuacion I-V para diferentes pares (V,I)
            clear V
            clear I

            V=0:.1:50; %Vector de tensiones
            I=zeros(1,size(V,2)); %Vector de corriente

```

```

        for j=1:size(V,2)           % Se calcula para todos los
voltagej                             voltajes
            g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);

            while (abs(g(j))>tol)
                g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);
                glin(j)=-Io*Rs/Vt/Ns/a*exp((V(j)+I(j)*Rs)/Vt/Ns/a)-Rs/Rp-1;
                I_(j)=I(j)-g(j)/glin(j);      %Newton Raphson
                I(j)=I_(j);
            end
        end

        %Cálculo de la potencia usando la ecuacion I-V
        P=(Ipv-Io*(exp((V+I.*Rs)/Vt/Ns/a)-1)-(V+I.*Rs)/Rp).*V;

        Pmax_m= max(P); % Asigna a Pmax_m el maximo valor del vector
Potencia (P)
        error=(Pmax_m-Pmax_e);
        %Incremento de Rs
        Rs=Rs+0.001;
    end
    % XXXXXXXXXXXXXXXXXXXXXXXXXXXX      FIN      MODELO PV
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        end% Fin if(G1~=G_anterior)
            Ipvn=(Rs+Rp)/Rp*Iscn           % Iscn = Isc de datasheet
            Ipv=(Ipvn+Ki*dT)*G1/Gn         % Corriente fotogenerada

        % Iout1 del sensor a la salida del Buck para cálculo de Vout=Vref
        fprintf(obj1, 1000.5);           % Clave para enviar dato
de Iout
            Isalida = fscanf(obj1, '%s\n'); % Tomo Isalida puerto
bluetooth
            Iout1= str2double(Isalida);    % Transformacion a
numerico

        %
                " VOLTAJE DE UN SOLO PANEL "

        % Vout = Ns*Vt*log((Ipv-Iout)/Imp)+Vmp-Rs*Iout % formula general
Vout
        fprintf('Vout un solo panel:\n');
        Vout = Ns*Vt*log((Ipv-Iout1)/Imp)+Vmp-Rs*Iout1 % Voltaje de
salida
            if Ipv<Iout1

set(handles.mensajes, 'ForegroundColor', 'white', 'BackgroundColor', 'red', 'String', 'LA CORRIENTE DE EN LA CARGA EXCEDE LA CORRIENTE DEL PANEL FOTOVOLTAICO')
            else

set(handles.mensajes, 'ForegroundColor', 'white', 'BackgroundColor', 'blue', 'String', 'SIMULACION EN CURSO ')
            end
            if(Iout1<0.3)
                Vout=Vocn;
            else
                Vout = Ns*Vt*log((Ipv-Iout1)/Imp)+Vmp-Rs*Iout1 % Voltaje
de salida

```

```

end

Vref= Vout;% .....
fprintf(obj1,Vref);% .....
Vvalida = fscanf(obj1,'%s\n');% .....
Vout1= str2double(Vvalida);% .....

%          " FIN VOLTAJE DE UN SOLO PANEL "

Pmax_PV=Imp*Vmp;%..... Potencia max.
del panel (datasheet)
Pout=Vout1*Iout1 %..... Potencia de
salida real del Buck Converter

set(handles.mostrar_Vout,'String',Vout1);% .... Muestra Vout
del sensor en el panel
set(handles.mostrar_Iout,'String',Iout1);%..... Muestra Iout
del sensor en el panel
set(handles.mostrar_Pout_PV,'String',Pout);%... Muestra Pout en
el punto de trabajo

% ////////////////////////////////// GRÁFICAS I-V & P-V
////////////////////////////////////
axes(handles.axes3);
cla(handles.axes3,'reset');%..... limpia y
restablece propiedades del axes3
set(handles.axes3,'XMinorGrid','on','YMinorGrid','on');% activa
subcuadrícula al axes3
grid on % Activa la
cuadrícula en la gráfica

yyaxis left % Escoge:eje
'Y' izquierdo de la gráfica para(I)
plot(V,I,'b','LineWidth',1) % Gráfica I-V
,'b'=azul ,AnchoDeLinea= 1.5
xlabel('V O L T A J E [ V ]','FontSize',10) % Tamaño de la
etiqueta del eje X: 10
ylabel('C O R R I E N T E [ A ]','FontSize',10) % Tamaño de la
etiqueta del eje Y: 10
ylim([0 1.2*Isc])%..... Límites eje
de Corriente (eje 'Y' izquierdo)
hold on
plot(Vout1,Iout1,'.b','MarkerSize',25);

yyaxis right%..... Escoge:eje
'Y' derecho de la gráfica para(Potencia)
plot(V,P,'r','LineWidth',1)%..... Gráfica P-V
,'r'=rojo ,AnchoDeLinea= 1
ylabel('P O T E N C I A [ W ]','FontSize',10)%.... Tamaño de la
etiqueta del eje Y: 10
ylim([0 1.1*Pmax_e]) %..... Límites eje
'Y' de Potencia (eje 'Y'derecho)
hold on
plot(Vout1,Pout,'.r','MarkerSize',25);

% ////////////////////////////////// FIN GRÁFICAS I-V & P-V
////////////////////////////////////

```

```

% ##### GRÁFICA VOLTAJE vs IRRADIANCIA
#####
axes(handles.axes1);
cla(handles.axes1,'reset'); %
limpia y restablece propiedades del axes3
set(handles.axes1,'XMinorGrid','on','YMinorGrid','on'); %
activa subcuadrícula al axes3
grid on %
Activa la cuadrícula en la gráfica

yyaxis left %
Escoge eje 'Y' izquierdo de la gráfica para (I)
plot(G_Quitto,VG_IoutCero,'.r','LineWidth',0.1)%.....
Gráfica G-V , 'r'=rojo ,AnchoDeLinea= 1
xlabel('I R R A D I A N C I A [ W/m² ]','FontSize',10)%
Tamaño de la etiqueta del eje X: 10
ylabel('V O L T A J E [ V ]','FontSize',10)%.....
xlim([0 1700])%.....
Límites eje 'Y' de Voltaje (eje 'Y'derecho)
ylim([0 1.1* Vocn1])%.....
Límites eje 'Y' de Voltaje (eje 'Y'izquierdo)
hold on
plot(G1,Vout1,'.r','MarkerSize',25);

yyaxis right
ylim([0 1.1* Vocn1])%..... Límites eje
'Y' de Voltaje (eje 'Y'derecho)

% ##### FIN GRÁFICA VOLTAJE vs IRRADIANCIA
#####

cont=cont+1
pause(0.3)
end % fin while CASO1

%CASO2: Tipo de Simulación: PANEL Irradiancia: G_VARIABLE
.....
if((TipoDeSimulacion==1)&&(TipoDeIrradiancia==0)&&stop==0)
fprintf('1 SOLO Panel y G_variable:\n');
global DatosG_Archivo_irradiancia GmaxQuitto
stop=0;
G_anterior=0;
Resul_Vref_2= 1:length(DatosG_Archivo_irradiancia);% almacenar
resultados Vref del modelo de cada irradiancia
Resul_Iout1_2= 1:length(DatosG_Archivo_irradiancia);%
almacenar los resultados Iout que viene del conversor
Resul_Vout1_2= 1:length(DatosG_Archivo_irradiancia);%
almacenar los resultados Vout que viene del conversor

for i=1:length(DatosG_Archivo_irradiancia)%

if stop==0 % Condicion si Botón Stop No ha sido presionado

% XXXXXXXXXXXXXXXXXXXXXXXXXXXX MODELO PV
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

% Parámetros cuando la irradiancia es diferente de 1000

```

```

G1=DatosG_Archivo_irradiancia(i)% G1= Dato de irradiancia
if (G1~=G_anterior)
    G_anterior=G1;
    Ti=Imp1/Iscn1;           % Tasa de cambio de
corriente
    Tv=Vmp1/Vocn1;         % Tasa de cambio de
voltaje
    dT=abs (T-Tn);        % Variacion de
temperatura
    Isc=(Iscn1+Ki*dT)*G1/Gn; % Corriente Isc a
irradiacion G1
    Imp= Isc*Ti;           % Actualizacion de Imp
    Iscn=Isc;             % Actualizacion de Iscn
    Vmp= Vmp1-((Imp1-Imp)/m); % Calculo de Vmp a Imp
de limite max de corriente
    Vocn=Vmp/Tv;         % Actualizacion de Vocn

    % Datos obtenidos del fabricante del PV
    Np=1;                 % Numero de celdas en paralelo
    Pmax_e=Vmp*Imp;      % Potencia maxima nominal

    % Corriente nominal de saturacion del diodo
    Ion=Iscn/(exp(Vocn/a/Ns/Vtn)-1);
    Io=Ion;

    % Valores referenciales de Rs y Rp
    Rs_max=(Vocn-Vmp)/Imp;
    Rp_min= Vmp/(Iscn-Imp)-Rs_max;

    % Valores iniciales de Rs y Rp
    Rp=Rp_min;
    Rs=0;
    tol=0.001;           % Error de tolerancia de potencia
    P=0;                 % inicialización del vector
Potencia 'P'
    error=Inf;

%Proceso iterativo para Rs y Rp hasta que Pmax,modelo=Pmax,exp
while (error>tol)
    dT=abs (T-Tn);
    Ipvn=(Rs+Rp)/Rp*Iscn; % Iscn = Isc de datasheet
    Ipv=(Ipvn+Ki*dT)*G/Gn; % Corriente fotogenerada
    Isc=(Iscn+Ki*dT)*G/Gn; % Isc para una determinada G

    %Resistencia en paralelo
    Rp=Vmp*(Vmp+Imp*Rs)/(Vmp*Ipv-
Vmp*Io*exp((Vmp+Imp*Rs)/Vt/Ns/a)+Vmp*Io-Pmax_e);
    %Resolver la ecuacion I-V para diferentes pares (V,I)
    clear V
    clear I
    V=0:.1:50;           %Vector de tensiones
    I=zeros(1,size(V,2)); %Vector de corriente

    for j=1:size(V,2)    %Se calcula para todos los
voltajes
        g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);

        while (abs(g(j))>tol)

```

```

        g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);
        glin(j)=-Io*Rs/Vt/Ns/a*exp((V(j)+I(j)*Rs)/Vt/Ns/a)-Rs/Rp-1;
        I_(j)=I(j)-g(j)/glin(j);      %Newton Raphson
        I(j)=I_(j);
    end
end

    %Cálculo de la potencia usando la ecuacion I-V
    P=(Ipv-Io*(exp((V+I.*Rs)/Vt/Ns/a)-1)-(V+I.*Rs)/Rp).*V;
    Pmax_m= max(P); % Asigna a Pmax_m el maximo valor del
vector Potencia (P)
    error=(Pmax_m-Pmax_e);
    %Incremento de Rs
    Rs=Rs+0.001;
        end% Fin while(error>tol)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        Ipvn=(Rs+Rp)/Rp*Iscn           % Iscn = Isc de
datasheet
        Ipv=(Ipvn+Ki*dT)*G1/Gn         % Corriente
fotogenerada
        end% ..... FIN Si Ganterior es
diferente de G1 actual
    % Iout1 del sensor a la salida del Buck para cálculo de Vout=Vref
    fprintf(obj1, 1000.5);             % Clave para enviar
dato de Iout
    Isalida = fscanf(obj1, '%s\n');    % Tomo Isalida puerto
bluetooth
    Iout1= str2double(Isalida);        % Transformacion a
numerico

    % Vout = Ns*Vt*log((Ipv-Iout)/Imp)+Vmp-Rs*Iout % formula general
Vout
    fprintf('Vout un solo panel:\n');

    if(Iout1<0.3)
        Vout=Vocn
    else
        Vout = Ns*Vt*log((Ipv-Iout1)/Imp)+Vmp-Rs*Iout1 % Voltaje de
salida
    end

    Vref= Vout;% ..... % Vref= 34.1;
    fprintf(obj1,Vref);% .....
    Vsalida = fscanf(obj1, '%s\n');% .....
    Vout1= str2double(Vsalida);% .....

    if Ipv<Iout1

set(handles.mensajes, 'ForegroundColor', 'white', 'BackgroundColor', 'red', 'String', 'LA CORRIENTE DE CONSUMO EN LA CARGA EXCEDE LA CORRIENTE DEL PANEL FOTOVOLTAICO')
    else

set(handles.mensajes, 'ForegroundColor', 'white', 'BackgroundColor', 'blue', 'String', 'SIMULACION EN CURSO ')

```

```

end

if isreal(Vout)
    Vout=Vout;
else
    Vout=0;
end
Resul_Vref_2(i)=Vout;%
Resul_Iout1_2(i)=Iout1;
Resul_Vout1_2(i)=Vout1;

%          " FIN VOLTAJE DE UN SOLO PANEL "

Pmax_PV=Imp*Vmp; %..... Potencia max.
del panel (datasheet)
Pout=Vout1*Iout1 %...Vout1.....Potencia de
salida real del Buck Converter
set(handles.mostrar_Vout,'String',Vout1);% Muestra Vout
del conversor Buck en el panel
set(handles.mostrar_Iout,'String',Iout1);% Muestra Iout
del conversor Buck en el panel
set(handles.mostrar_Pout_PV,'String',Pout);% Muestra Pout en
el punto de trabajo

% XXXXXXXXXXXXXXXXXXXXXXXXXXXX          FIN   MODELO PV
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

% ////////////////////////////////////  GRÁFICAS I-V  &  P-V
//////////////////////////////////////
axes(handles.axes3);
cla(handles.axes3,'reset'); % limpia y
restablece propiedades del axes3
set(handles.axes3,'XMinorGrid','on','YMinorGrid','on');% activa
subcuadrícula al axes3
grid on % Activa la
cuadrícula en la gráfica

yyaxis left % Escoge:eje
'Y' izquier de la gráfica para(I)
plot(V,I,'b','LineWidth',1) % Gráfica I-V
, 'b'=azul ,AnchoDeLinea= 1.5
xlabel('V O L T A J E [ V ]','FontSize',10) % Tamaño de la
etiqueta del eje X: 10
ylabel('C O R R I E N T E [ A ]','FontSize',10) % Tamaño de la
etiqueta del eje Y: 10
ylim([0 1.2*Isc])%..... Límites eje
de Corriente (eje 'Y' izquierdo)
hold on
plot(Vout1,Iout1,'.b','MarkerSize',25);

yyaxis right%..... Escoge:eje
'Y' derecho de la gráfica para(Potencia)
plot(V,P,'r','LineWidth',1)%..... Gráfica P-V
, 'r'=rojo ,AnchoDeLinea= 1
ylabel('P O T E N C I A [ W ]','FontSize',10)%.... Tamaño de la
etiqueta del eje Y: 10
ylim([0 1.1*Pmax_e]) %..... Límites eje
'Y' de Potencia (eje 'Y'derecho)
hold on
plot(Vout1,Pout,'.r','MarkerSize',25);

```

```

% ////////////////////////////////////// FIN GRÁFICAS I-V & P-V
////////////////////////////////////

% ##### GRÁFICA VOLTAJE vs IRRADIANCIA
#####
axes(handles.axes1);
cla(handles.axes1,'reset'); %
limpia y restablece propiedades del axes3
set(handles.axes1,'XMinorGrid','on','YMinorGrid','on'); %
activa subcuadrícula al axes3
grid on %
Activa la cuadrícula en la gráfica

yyaxis left %
Escoge eje 'Y' izquierdo de la gráfica para (I)
plot(GQuitto,VG_IoutCero,'.r','LineWidth',0.2)%.....
Gráfica G-V , 'r'=rojo ,AnchoDeLinea= 1
xlabel('I R R A D I A N C I A [ W/m² ]','FontSize',10)%
Tamaño de la etiqueta del eje X: 10
ylabel('V O L T A J E [ V ]','FontSize',10)%.....
xlim([0 1700])%..... Límites
eje 'Y' de Voltaje (eje 'Y'derecho)
ylim([0 1.1* Vocn1])%.....
Límites eje 'Y' de Voltaje (eje 'Y'izquierdo)
hold on
plot(G1,Vout1,'.r','MarkerSize',25);

yyaxis right
ylim([0 1.1* Vocn1])%..... Límites eje
'Y' de Voltaje (eje 'Y'derecho)

% ##### FIN GRÁFICA VOLTAJE vs IRRADIANCIA
#####

cont=cont+1;
pause(0.3)

end % FIN Condicion para stop
end% FIN for

cabecera={'Irradiancia G','Vref.del modelo','Iout conversor','Vout
conversor'};

xlswrite('C:\Emulador\Resultados_Panel',[DatosG_Archivo_irradiancia(
:),Resul_Vref_2(:),Resul_Iout1_2(:),Resul_Vout1_2(:)],'Hojal','A1');
%
xlswrite('C:\Emulador\Resultados_Panel',cabecera,'Hojal','A1');%

end %FIN CASO 2
.....

%CASO3: Tipo de Simulación: ARREGLO PV Irradiancia: G_FIJA
.....

while(TipoDeSimulacion==0)&&(TipoDeIrradiancia==1)&&(stop==0)

```

```

    fprintf('Arreglo y G_fija:\n');

    % XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX MODELO PV
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    if cont==0 % Realiza el proceso solo al cambiar irradiancia y
    press PLAY

        % Parámetros cuando la irradiancia es diferente de 1000
        G1=str2double(get(handles.in_G_fija,'string'))% G1= Dato
de irradiancia
        Ti=Imp1/Iscn1; % Tasa de cambio de corriente
        Tv=Vmp1/Vocn1; %% Tasa de cambio de voltaje
        dT=abs(T-Tn); % Variacion de temperatura
        Isc=(Iscn1+Ki*dT)*G1/Gn; % Corriente Isc a irradiacion
G1
        Imp= Isc*Ti; % Actualizacion de Imp
        Iscn=Isc; % Actualizacion de Iscn
        Vmp= Vmp1-((Imp1-Imp)/m); %% Calculo de Vmp a Imp de limite
max de corriente
        Vocn=Vmp/Tv; %% Actualizacion de Vocn

        % Datos obtenidos del fabricante del PV
        Np=1; %Numero de celdas en paralelo
        Pmax_e=Vmp*Imp; %Potencia maxima nominal

        % Corriente nominal de saturacion del diodo
        Ion=Iscn/(exp(Vocn/a/Ns/Vtn)-1);
        Io=Ion;

        % Valores referenciales de Rs y Rp
        Rs_max=(Vocn-Vmp)/Imp;
        Rp_min= Vmp/(Iscn-Imp)-Rs_max;

        % Valores iniciales de Rs y Rp
        Rp=Rp_min;
        Rs=0;

        tol=0.001; %Error de tolerancia de potencia
        P=0; % inicialización del vector Potencia 'P'
        error=Inf;

        %Proceso iterativo para Rs y Rp hasta que Pmax,modelo=Pmax,exp
        while(error>tol)
            dT=abs(T-Tn);
            Ipvn=(Rs+Rp)/Rp*Iscn; % Iscn = Isc de datasheet
            Ipv=(Ipvn+Ki*dT)*G/Gn; % Corriente fotogenerada
            Isc=(Iscn+Ki*dT)*G/Gn; % Isc para una determinada G

            %Resistencia en paralelo
            Rp=Vmp*(Vmp+Imp*Rs)/(Vmp*Ipv-
Vmp*Io*exp((Vmp+Imp*Rs)/Vt/Ns/a)+Vmp*Io-Pmax_e);

            %Resolver la ecuacion I-V para diferentes pares (V,I)
            clear V
            clear I

            V=0:.1:50; %Vector de tensiones

```

```

I=zeros(1,size(V,2)); %Vector de corriente

for j=1:size(V,2) %Se calcula para todos los voltajes
    g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);

    while (abs(g(j))>tol)
        g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);
        glin(j)=-Io*Rs/Vt/Ns/a*exp((V(j)+I(j)*Rs)/Vt/Ns/a)-Rs/Rp-1;
        I_(j)=I(j)-g(j)/glin(j); %Newton Raphson
        I(j)=I_(j);
    end
end

%Cálculo de la potencia usando la ecuacion I-V
P=(Ipv-Io*(exp((V+I.*Rs)/Vt/Ns/a)-1)-(V+I.*Rs)/Rp).*V;

%Voltaje=P/I
Pmax_m= max(P);
error=(Pmax_m-Pmax_e);
%Incremento de Rs
Rs=Rs+0.001;
end

% Cambio variables del modelo para 1 PANEL para graficar como
ARREGLO
Nsa=str2double(get(handles.in_PanelesEnSerie,'string'));%
|
Npa=str2double(get(handles.in_PanelesEnParalelo,'string'));%
V=V*Nsa;% Vector Voltaje
I=I*Npa;
Vocn=Vocn*Nsa;
Isc=Isc*Npa; %Corriente Isc a irradiacion G1
Iscn=Iscn*Npa; %Corriente Iscn fabricante
Imp=Imp*Npa;
Vmp=Vmp*Nsa;
Pmax_e= Imp*Vmp;
P=P*(Nsa*Npa);
Rp=Rp/Npa;
Rs=Rs*Nsa;
cont=cont+1;
end % FIN Realiza el proceso solo al cambiar irradiancia y
press PLAY

% FIN Cambio variables del modelo 1 PANEL para graficar como ARREGLO

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
fprintf('Calculo de voltaje:\n');
Ipvn=(Rs+Rp)/Rp*Iscn % Iscn = Isc de
datasheet
Ipv=(Ipvn+Ki*dT)*G1/Gn % Corriente
fotogenerada

% Iout1 del sensor a la salida del Buck para cálculo de Vout=Vref
fprintf(obj1, 1000.5); % Clave para enviar
dato de Iout
Isalida = fscanf(obj1, '%s\n'); % Tomo Isalida puerto
bluetooth

```

```

        Iout1= str2double(Isalida);           % Transformacion a
numerico

    %
        " VOLTAJE DE UN SOLO PANEL "

    % Vout = Ns*Vt*log((Ipv-Iout)/Imp)+Vmp-Rs*Iout  % formula general
Vout
    fprintf('Vout un solo panel:\n');
    Vout = Ns*Vt*log((Ipv-Iout1)/Imp)+Vmp-Rs*Iout1 % Voltaje de
salida
    Vref= Vout;% ..... % Vref= 34.1;
    fprintf(obj1,Vref);% .....
    Vsalida = fscanf(obj1,'%s\n');% .....
    Vout1= str2double(Vsalida);% .....

    if Ipv<Iout1

set(handles.mensajes,'ForegroundColor','white','BackgroundColor','red',
'String','LA CORRIENTE DE CONSUMO EN LA CARGA EXCEDE LA CORRIENTE
DEL PANEL FOTOVOLTAICO')
    else

set(handles.mensajes,'ForegroundColor','white','BackgroundColor','blue',
'String','SIMULACION EN CURSO ')
    end

    %
        " FIN VOLTAJE DE UN SOLO PANEL "

    Pmax_PV=Imp*Vmp; %..... Potencia max.
del panel (datasheet)
    Pout=Vout1*Iout1 %..... Potencia de
salida

    set(handles.mostrar_Vout,'String',Vout1);% Muestra Vout
del sensor en el panel
    set(handles.mostrar_Iout,'String',Iout1);% Muestra Iout
del sensor en el panel
    set(handles.mostrar_Pout_PV,'String',Pout);% Muestra Pout en
el punto de trabajo

    % XXXXXXXXXXXXXXXXXXXXXXXXXXXX FIN MODELO PV
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

    % ////////////////////////////////// GRÁFICAS I-V & P-V
////////////////////////////////////
    axes(handles.axes3);
    cla(handles.axes3,'reset'); % limpia y
restablece propiedades del axes3
    set(handles.axes3,'XMinorGrid','on','YMinorGrid','on');% activa
subcuadrícula al axes3
    grid on % Activa la
cuadrícula en la gráfica

    yyaxis left % Escoge:eje
'Y' izquier de la gráfica para(I)
    plot(V,I,'b','LineWidth',1) % Gráfica I-V
, 'b'=azul ,AnchoDeLinea= 1.5
    xlabel('V O L T A J E [ V ]','FontSize',10) % Tamaño de la
etiqueta del eje X: 10

```

```

        ylabel('C O R R I E N T E   [ A ]','FontSize',10) % Tamaño de la
etiqueta del eje Y: 10
        ylim([0 1.2*Isc])%..... Límites eje
de Corriente (eje 'Y' izquierdo)
        hold on
        plot(Vout1,Iout1,'.b','MarkerSize',25);

        yyaxis right%..... Escoge:eje
'Y' derecho de la gráfica para(Potencia)
        plot(V,P,'r','LineWidth',1)%..... Gráfica P-V
, 'r'=rojo ,AnchoDeLinea= 1
        ylabel('P O T E N C I A   [ W ]','FontSize',10)%.... Tamaño de la
etiqueta del eje Y: 10
        ylim([0 1.1*Pmax_e]) %..... Límites eje
'Y' de Potencia (eje 'Y'derecho)
        hold on
        plot(Vout1,Pout,'.r','MarkerSize',25);
% ////////////////////////////////// FIN GRÁFICAS I-V   &   P-V
////////////////////////////////////

% ##### GRÁFICA VOLTAJE vs IRRADIANCIA
#####
        axes(handles.axes1);
        cla(handles.axes1,'reset'); %
limpia y restablece propiedades del axes3
        set(handles.axes1,'XMinorGrid','on','YMinorGrid','on'); %
activa subcuadrícula al axes3
        grid on %
Activa la cuadrícula en la gráfica

        yyaxis left %
Escoge:eje 'Y' izquier de la gráfica para(I)
        plot(GQuito,VG_IoutCero,'r.','LineWidth',0.2)%.....
Gráfica G-V , 'r'=rojo ,AnchoDeLinea= 1
        xlabel('I R R A D I A N C I A   [ W/m² ]','FontSize',10)%
Tamaño de la etiqueta del eje X: 10
        ylabel('V O L T A J E   [ V ]','FontSize',10)%.....
        xlim([0 1700])%.....Límites
eje 'Y' de Voltaje (eje 'Y'derecho)
        ylim([0 1.1* (Vocn1*Nsa)])%.....
Límites eje 'Y' de Voltaje (eje 'Y'izquierdo)
        hold on
        plot(G1,Vout1,'.r','MarkerSize',25);%

        yyaxis right
        ylim([0 1.1* (Vocn1*Nsa)])%.....
Límites eje 'Y' de Voltaje (eje 'Y'derecho)

% ##### FIN GRÁFICA VOLTAJE vs IRRADIANCIA
#####

        cont=cont+1
        pause(0.2)

end % FIN CASO 3
.....

```

```

    %CASO4: Tipo de Simulación: ARREGLO PV      Irradiancia:
G_VARIABLE .....
    if (TipoDeSimulacion==0) && (TipoDeIrradiancia==0) && (stop==0)
        fprintf('Arreglo y G_Variable:\n');
        global DatosG_Archivo_irradiancia
        stop=0;
        Resul_Vref_4= 1:length(DatosG_Archivo_irradiancia);%
almacenar resultados Vref del modelo de cada irradiancia
        Resul_Iout1_4= 1:length(DatosG_Archivo_irradiancia);%
almacenar los resultados Iout que vien del conversor
        Resul_Vout1_4= 1:length(DatosG_Archivo_irradiancia);%
almacenar los resultados Iout que vien del conversor

    for i=1:length(DatosG_Archivo_irradiancia)
        if stop==0 % Condición para STOP

            % XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX MODELO PV
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

            % Parámetros cuando la irradiancia es diferente de 1000
            G1=DatosG_Archivo_irradiancia(i)% G1= Dato de irradiancia
            Ti=Imp1/Iscn1; % Tasa de cambio de corriente
            Tv=Vmp1/Vocn1; %% Tasa de cambio de voltaje
            dT=abs(T-Tn); % Variacion de temperatura
            Isc=(Iscn1+Ki*dT)*G1/Gn; % Corriente Isc a irradiacion
G1
            Imp= Isc*Ti; % Actualizacion de Imp
            Iscn=Isc; % Actualizacion de Iscn
            Vmp= Vmp1-((Imp1-Imp)/m); %% Calculo de Vmp a Imp de limite
max de corriente
            Vocn=Vmp/Tv; %% Actualizacion de Vocn

            % Datos obtenidos del fabricante del PV
            Np=1; %Numero de celdas en paralelo
            Pmax_e=Vmp*Imp; %Potencia maxima nominal

            % Corriente nominal de saturacion del diodo
            Ion=Iscn/(exp(Vocn/a/Ns/Vtn)-1);
            Io=Ion;

            % Valores referenciales de Rs y Rp
            Rs_max=(Vocn-Vmp)/Imp;
            Rp_min= Vmp/(Iscn-Imp)-Rs_max;

            % Valores iniciales de Rs y Rp
            Rp=Rp_min;
            Rs=0;

            tol=0.001; %Error de tolerancia de potencia
            P=0; % inicialización del vector Potencia 'P'
            error=Inf;

            %Proceso iterativo para Rs y Rp hasta que Pmax,modelo=Pmax,exp
            while(error>tol)
                dT=abs(T-Tn);
                Ipvn=(Rs+Rp)/Rp*Iscn; % Iscn = Isc de datasheet
                Ipv=(Ipvn+Ki*dT)*G/Gn; % Corriente fotogenerada

```

```

Isc=(Iscn+Ki*dT)*G/Gn; % Isc para una determinada G

%Resistencia en paralelo
Rp=Vmp*(Vmp+Imp*Rs)/(Vmp*Ipv-
Vmp*Io*exp((Vmp+Imp*Rs)/Vt/Ns/a)+Vmp*Io-Pmax_e);

%Resolver la ecuacion I-V para diferentes pares (V,I)
clear V
clear I

V=0:.1:50; %Vector de tensiones
I=zeros(1,size(V,2)); %Vector de corriente

for j=1:size(V,2) %Se calcula para todos los voltajes
g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);

while (abs(g(j))>tol)
g(j)=Ipv-Io*(exp((V(j)+I(j)*Rs)/Vt/Ns/a)-1)-
(V(j)+I(j)*Rs)/Rp-I(j);
glin(j)=-Io*Rs/Vt/Ns/a*exp((V(j)+I(j)*Rs)/Vt/Ns/a)-Rs/Rp-1;
I_(j)=I(j)-g(j)/glin(j); %Newton Raphson
I(j)=I_(j);
end
end

%Cálculo de la potencia usando la ecuacion I-V
P=(Ipv-Io*(exp((V+I.*Rs)/Vt/Ns/a)-1)-(V+I.*Rs)/Rp).*V;

Pmax_m= max(P); % Asigna a Pmax_m el maximo valor del vector
Potencia (P)
error=(Pmax_m-Pmax_e);
%Incremento de Rs
Rs=Rs+0.001;
end
% Cambio variables del modelo para 1 PANEL para graficar como
ARREGLO
Nsa=str2double(get(handles.in_PanelesEnSerie,'string'));%
|
Npa=str2double(get(handles.in_PanelesEnParalelo,'string'));%
V=V*Nsa;% Vector Voltaje
I=I*Npa;
Vocn=Vocn*Nsa;
Isc=Isc*Npa; %Corriente Isc a irradiacion G1
Iscn=Iscn*Npa %Corriente Iscn fabricante
Imp=Imp*Npa
Vmp=Vmp*Nsa
Pmax_e= Imp*Vmp;
P=P*(Nsa*Npa);
Rp=Rp/Npa;
Rs=Rs*Nsa;
cont=cont+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
fprintf('Calculo de voltaje:\n');
Ipv=(Rs+Rp)/Rp*Iscn % Iscn = Isc de
datasheet
Ipv=(Ipv+Ki*dT)*G1/Gn % Corriente
fotogenerada

```

```

    % Iout1 del sensor a la salida del Buck para cálculo de Vout=Vref
    fprintf(obj1, 1000.5); % Clave para enviar
dato de Iout
    Isalida = fscanf(obj1, '%s\n'); % Tomo Isalida puerto
bluetooth
    Iout1= str2double(Isalida); % Transformacion a
numerico

%
% " VOLTAJE DE UN SOLO PANEL "

% Vout = Ns*Vt*log((Ipv-Iout)/Imp)+Vmp-Rs*Iout % formula general
Vout
    fprintf('Vout un solo panel:\n');
    Vout = Ns*Vt*log((Ipv-Iout1)/Imp)+Vmp-Rs*Iout1 % Voltaje de
salida
    Vref= Vout;% ..... % Vref= 34.1;
    fprintf(obj1,Vref);% .....
    Vsalida = fscanf(obj1, '%s\n');% .....
    Vout1= str2double(Vsalida);% .....

    if Ipv<Iout1

set(handles.mensajes, 'ForegroundColor', 'white', 'BackgroundColor', 'red', 'String', 'LA CORRIENTE DE CONSUMO EN LA CARGA EXCEDE LA CORRIENTE DEL PANEL FOTOVOLTAICO')
    else

set(handles.mensajes, 'ForegroundColor', 'white', 'BackgroundColor', 'blue', 'String', 'SIMULACION EN CURSO ')
    end

    if isreal(Vout)
        Vout=Vout;

    else
        Vout=0;

    end

    Resul_Vref_4(i)=Vout;%
    Resul_Iout1_4(i)=Iout1;
    Resul_Vout1_4(i)=Vout1;
%
% " FIN VOLTAJE DE UN SOLO PANEL "

    Pmax_PV=Imp*Vmp; %..... Potencia max.
del panel (datasheet)
    Pout=Vout1*Iout1 %..... Potencia de
salida real del Buck Converter

    set(handles.mostrar_Vout, 'String', Vout1);% .... Muestra Vout
del sensor en el panel
    set(handles.mostrar_Iout, 'String', Iout1);% .... Muestra Iout
del sensor en el panel
    set(handles.mostrar_Pout_PV, 'String', Pout);% .. Muestra Pout en
el punto de trabajo

```

```

% XXXXXXXXXXXXXXXXXXXXXXXXXXXX      FIN      MODELO PV
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

% ////////////////////////////////////  GRÁFICAS I-V  &  P-V
////////////////////////////////////

axes(handles.axes3);
cla(handles.axes3,'reset');           % limpia y
restablece propiedades del axes3
set(handles.axes3,'XMinorGrid','on','YMinorGrid','on');% activa
subcuadrícula al axes3
grid on                               % Activa la
cuadrícula en la gráfica

yyaxis left                           % Escoge:eje
'Y' izquierdo de la gráfica para(I)
plot(V,I,'b','LineWidth',1)          % Gráfica I-V
, 'b'=azul ,AnchoDeLinea= 1.5
xlabel('V O L T A J E   [ V ]','FontSize',10) % Tamaño de la
etiqueta del eje X: 10
ylabel('C O R R I E N T E   [ A ]','FontSize',10) % Tamaño de la
etiqueta del eje Y: 10
ylim([0 1.2*Isc])%..... Límites eje
de Corriente (eje 'Y' izquierdo)
hold on
plot(Vout1,Iout1,'.b','MarkerSize',25);

yyaxis right%..... Escoge:eje
'Y' derecho de la gráfica para(Potencia)
plot(V,P,'r','LineWidth',1)%..... Gráfica P-V
, 'r'=rojo ,AnchoDeLinea= 1
ylabel('P O T E N C I A   [ W ]','FontSize',10)%.... Tamaño de la
etiqueta del eje Y: 10
ylim([0 1.1*Pmax_e]) %..... Límites eje
'Y' de Potencia (eje 'Y'derecho)
hold on
plot(Vout1,Pout,'.r','MarkerSize',25);

% ////////////////////////////////////  FIN GRÁFICAS I-V  &  P-V
////////////////////////////////////
% #####  GRÁFICA VOLTAJE vs IRRADIANCIA
#####
axes(handles.axes1);
cla(handles.axes1,'reset');           %
limpia y restablece propiedades del axes3
set(handles.axes1,'XMinorGrid','on','YMinorGrid','on'); %
activa subcuadrícula al axes3
grid on                               %
Activa la cuadrícula en la gráfica

yyaxis left                           %
Escoge:eje 'Y' izquierdo de la gráfica para(I)
plot(GQuitto,VG_IoutCero,'.r','LineWidth',0.2)%.....
Gráfica G-V , 'r'=rojo ,AnchoDeLinea= 1
xlabel('I R R A D I A N C I A   [ W/m² ]','FontSize',10)%
Tamaño de la etiqueta del eje X: 10
ylabel('V O L T A J E   [ V ]','FontSize',10)%.....
xlim([0 1700])%.....
Límites eje 'Y' de Voltaje (eje 'Y'derecho)
ylim([0 1.1* (Vocn1*Nsa)])%.....
Límites eje 'Y' de Voltaje (eje 'Y'izquierdo)

```

```

hold on
plot(G1,Vout1, '.r', 'MarkerSize',25);

yyaxis right
ylim([0 1.1* (Vocn1*Nsa)])%.....
Límites eje 'Y' de Voltaje (eje 'Y'derecho)

% ##### FIN GRÁFICA VOLTAJE vs IRRADIANCIA
#####

cont=cont+1;
pause(0.3)

end % FIN Condicion para stop
end%
titulos={'Irradiancia G', 'Vref.del modelo', 'Iout conversor', 'Vout
conversor'};

xlswrite('C:\Emulador\Resultados_Arreglo',[DatosG_Archivo_irradianci
a(:),Resul_Vref_4(:),Resul_Iout1_4(:),Resul_Vout1_4(:)], 'Hojal', 'A1'
);%
xlswrite('C:\Emulador\Resultados_Arreglo',titulos, 'Hojal', 'A1');%

end % FIN CASO 4
.....

%XXXXXXXXXXXXXXXXXXXXXXXXXXXX...FIN SIMULACION
PLAY...XXXXXXXXXXXXXXXXXXXXXXXXXXXX

function btn_Reset_Callback(hObject, eventdata, handles)
% global obj1
% fclose(obj1);
close(Emulador_Panel_Fotovoltaico);% cierra la ventana
Base_De_Datos_Paneles
clear all;
Emulador_Panel_Fotovoltaico;%Abre la ventana
Emulador_Panel_Fotovoltaico

```

Anexo 3: CÓDIGO DE PROGRAMACIÓN BLUETOOTH DE LA TARJETA ESP-WROOM-32

```
#include "BluetoothSerial.h"
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

//Declaracion de pines 15 y 2 y estados para los pulsadores
const int botonI = 15;
const int botonV = 2;

volatile int i=0, v=0;    //Contadores para vectores Iout y Vout

String data = "";        //Variable para almacenar datos del puerto serial bluetooth
BluetoothSerial SerialBT;
float Vref, Dato, Iout1, Vout1;
float Iout[10] = {0.00, 2.16, 3.52, 4.91, 5.27, 6.69, 7.38, 8.41, 9.67, 10.53};
float Vout[10] = {17.27, 18.41, 19.66, 17.75, 20.12, 22.57, 18.76, 19.32, 20.91, 21.44};

void setup() {
  Serial.begin(115200);
  SerialBT.begin("Simulador-Panel-Fotovoltaico");    //Nombre Bluetooth
  Serial.println("The device started, now you can pair it with bluetooth!");
  pinMode(botonI, INPUT);    //Declaracion de PIN 15 como entrada
  pinMode(botonV, INPUT);    //Declaracion de PIN 2 como entrada
  attachInterrupt(digitalPinToInterrupt(botonI), Corriente, RISING);
  attachInterrupt(digitalPinToInterrupt(botonV), Voltaje, RISING);
}

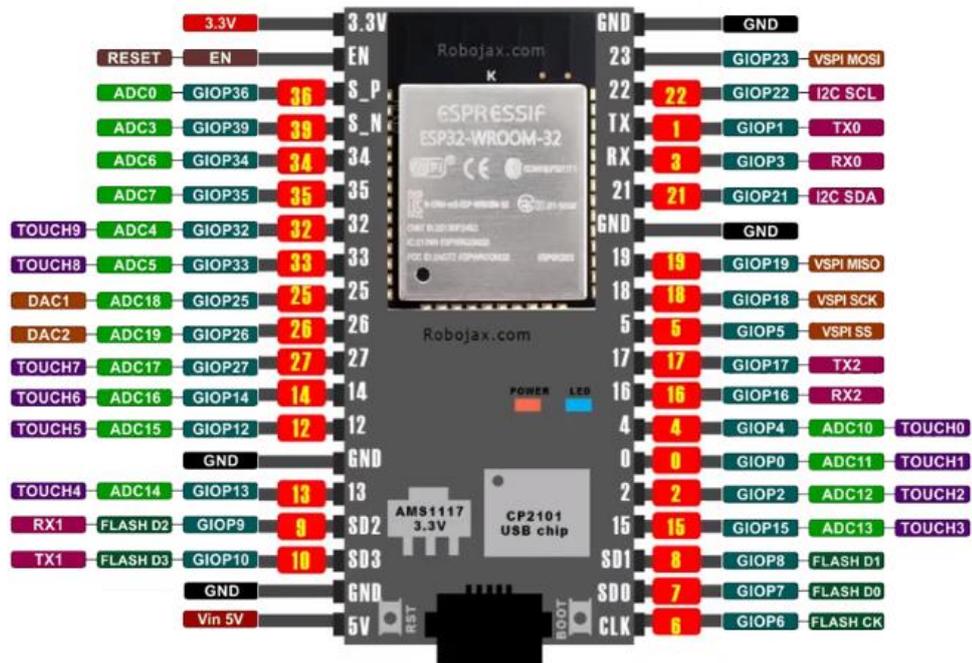
void loop() {
  while (SerialBT.available()) {
    char character = SerialBT.read();    //Lee cada caracter por el puerto serial bluetooth
    if (character != '\n') {    //Condicion de salto de dato
      data.concat(character);    //Concatena los caracteres leidos y almacenados por el puerto serial bluetooth
    }
    else {
      Dato = data.toFloat();    //Convierte el dato al flotante
      Serial.println(Dato);    //Imprime la variable Dato
      data = "";    //Limpia la variable data para lectura de nuevos datos
      delay(500);
      if (Dato==1000.5){    //Condicion para saber que se esta enviando dato de Corriente
        Iout1=Iout[i];    //Se extrae uno a uno los datos del vector de Iout
        SerialBT.println(Iout1);    //Envio por el puerto bluetooth el dato extraido
      }
      else {
        Vref=Dato;    //Si no es dato de Corriente es dato de Setpoint de Voltaje
        Serial.print("Vref= ");    //Imprime dato de setpoint de voltaje
        Serial.println(Vref);
        Vout1=Vout[v];    //Se extrae uno a uno los datos del vector de Vout
        SerialBT.println(Vout1);    //Envio por el puerto bluetooth el dato extraido
      }
    }
  }
}

void Corriente(){
  i++;    //Incremento de contador i
  if (i==10) {    //Condicion si se llega a 10 se reinicie el contador
    i=0;
  }
}

void Voltaje(){
  v++;    //Incremento de contador v
  if (v==10) {    //Condicion si se llega a 10 se reinicie el contador
    v=0;
  }
}
```

Anexo 4: DESCRIPCIÓN DE PINES DE LA TARJETA ESP-WROOM-32

ESP32 38 PINES ESP WROOM 32



Anexo 5: HOJAS DE DATOS DE LOS PANELES EGE-175-36, GP-50P-36, EGE-255P-60, EGE-270-60

SOLAR MODULE EGE-175M-36 MONOCRYSTALLINE MODULE 175 W / 36 Cells / 5 Bus bars



ELECTRICAL DATA AT STC*

Power output (Pmax)	175 W
Power tolerance	0~+5 %
Module efficiency	17.65 %
Maximum power voltage (Vmp)	19.13 V
Maximum power current (Imp)	9.15 A
Open circuit voltage (Voc)	23.49 V
Short circuit current (Isc)	9.45 A

*Standard Test Conditions: Irradiance: 1 000 W / m²
Cell temperature: 25 °C • AM: 1.5

ELECTRICAL DATA AT NOCT*

Power output (Pmax)	130.48 W
Maximum power voltage (Vmp)	17.93 V
Maximum power current (Imp)	7.27 A
Open circuit voltage (Voc)	21.78 V
Short circuit current (Isc)	7.68 A

*Nominal Operating Cell Temperature: Irradiance: 800 W / m²
Ambiant temperature: 20 °C • AM: 1.5 • Wind speed: 1 m/s

TEMPERATURE CHARACTERISTICS

NOCT	45 °C ±2 °C
Temperature coefficient of Pmax	-0.41%/°C
Temperature coefficient of Voc	-0.31%/°C
Temperature coefficient of Isc	+0.06%/°C

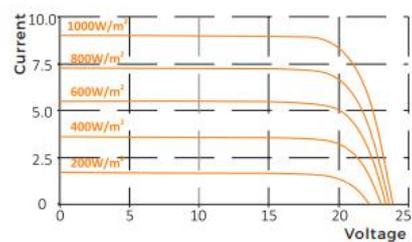
MAXIMUM RATINGS

Operating temperature range	-45 °C ~+85 °C
Maximum system voltage	1000 V
Max series fuse rating	10 A
Max front load (e.g.: snow)	5 400 Pa
Max back load (e.g.: wind)	2 400 Pa

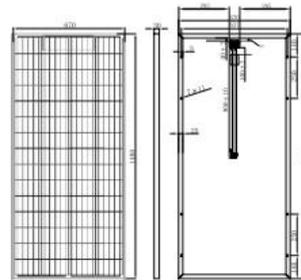
MECHANICAL CHARACTERISTICS

Cell type	Monocrystalline (156x156 mm)
Number of cells	36
Dimensions	1480x670x30 mm
Weight	11.0 kg
Glass	3.2 mm tempered glass
Frame	Anodized aluminium alloy
Junction box	IP65 or IP67
Cable	0.9 m
Connector	MC4 or MC4 compatible

EGE-175M-36 / I-V CURVES



ENGINEERING DRAWINGS (mm)



PACKAGING

Type	Pcs	Weight
Pallet	35 pcs	410 kg
20ft GP Container	720 pcs	8.4 t
40ft GP Container	1400 pcs	16.6 t
40ft HQ Container	1800 pcs	21.2 t



Polycrystalline Module Models

GP-055P-36 GP-050P-36

Electrical Specifications

*STC : Irradiance 1000W/m², AM1.5 spectrum, module temperature 25°C

Model type	GP-055P-36	GP-050P-36
Peak power (Pmax)	55W	50W
Cell Efficiency	19.40%	17.64%
Maximum power voltage (Vmp)	19.4V	18.3V
Maximum power current (Imp)	2.84A	2.73A
Open circuit voltage (Voc)	23.0V	22.2V
Short circuit current (Isc)	2.97A	2.90A
Power Tolerance	±3%	
Maximum system voltage	800V	
Series fuse rating (A)	4	
Number of bypass diode	2	

Temperature Coefficients

Current Temperature Coefficient	$\alpha(Isc)$	0.08%/°C
Voltage Temperature Coefficient	$\beta(Voc)$	-0.32%/°C
Power Temperature Coefficient		-0.38%/°C
Nominal Temperature Of Cells(NOTC)		46±2°C
Application Temperature Range		-40~+85°C

Mechanical Characteristics

Dimensions:	540 x 670 x 25mm (21.26x26.38x0.98 inch)
Weight:	3.8 kg (8.44lbs)
Solar Cells:	36 cells in series(4 x 9)
Construction:	High-transmission low-iron tempered glass, EVA, TPT/TPE
Frame:	Anodized aluminum is mainly for improving

corrosion resistance. Color: silver

Output(Optional)

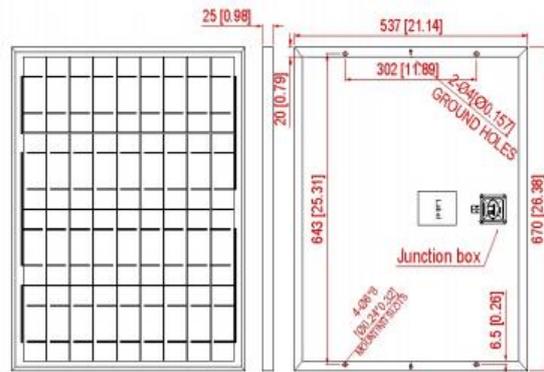
Cable	LAPP(4 mm ²)
Lengths	900mm(-) and 900mm(+)
Connector	MC Plug Type IV

Module Warranty:

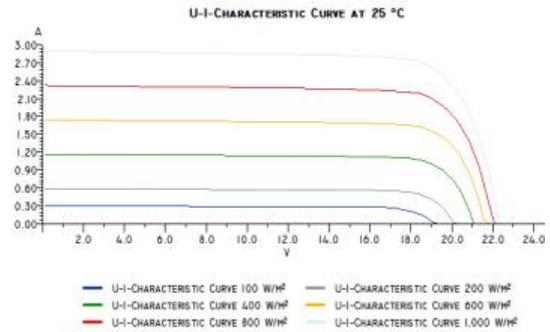
25-year limited warranty of 80% power output;

10-year limited warranty of 90% power output;

5-year limited warranty of materials and workmanship.

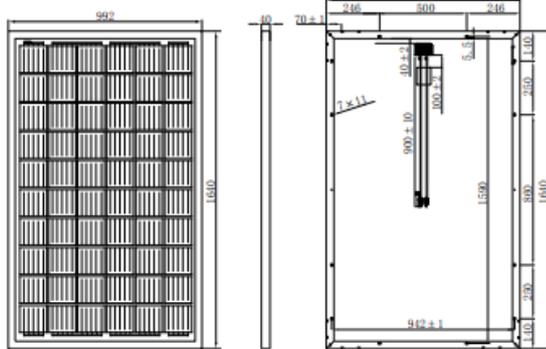


Module Diagram

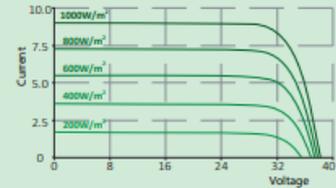


60 CELLS PHOTOVOLTAIC MODULE 255~275 W OUTPUT POWER

Engineering drawings



IV curves



Temperature characteristics

NOCT	45 °C ± 2 °C
Temperature coefficient of Pmax	-0.41 %/°C
Temperature coefficient of Voc	-0.31 %/°C
Temperature coefficient of Isc	+0.06 %/°C

Electrical characteristics at STC*

	255 W	260 W	265 W	270 W	275 W
Power output (Pmax)	255 W	260 W	265 W	270 W	275 W
Panel efficiency	15.67 %	15.99 %	16.29 %	16.60 %	16.90 %
Maximum power voltage (Vmp)	30.77 V	30.90 V	31.16 V	31.44 V	31.70 V
Maximum power current (Imp)	8.29 A	8.41 A	8.50 A	8.59 A	8.68 A
Open circuit voltage (Voc)	37.82 V	38.05 V	38.26 V	38.45 V	38.75 V
Short circuit current (Isc)	8.76 A	8.86 A	8.95 A	9.03 A	9.11 A

*Standard test conditions:
irradiance: 1000 W/m² • cell temperature: 25 °C • AM: 1.5

General characteristics

Operating conditions

Cell type	Polycrystalline (156×156 mm)	Operating temperature range	-45 °C ~ +85 °C
Number of cells	60	Max system voltage	1500 V
Dimensions	1640×992×40 mm	Max series fuse rating	15 A
Weight	18.8 kg	Max front load (ex: snow)	5400 Pa
Glass	3.2 mm tempered glass	Max back load (ex: wind)	2400 Pa
Frame	Anodized aluminium alloy		
Junction box	IP65 or IP67		
Cable	0.9 m		
Connector	MC4 or MC4 compatible		

Packaging

Per pallet	26 pcs
Per 20ft GP container	350 pcs
Per 40ft GP container	780 pcs
Per 40ft HQ container	800 pcs