

DESARROLLO DE UN CONTROL PID, QUE PERMITA MONITOREAR Y MODIFICAR LA TENSIÓN DE SALIDA DE UN VARIAC MONOFÁSICO, UTILIZANDO UN MICROCONTROLADOR ESP32 Y MOTORREDUCTOR ACOPLADO AL VOLANTE.

**Jhon Jairo Suárez Ramírez
Yeici Enrique Herrera Durán**

Universidad Antonio Nariño
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Bucaramanga, Colombia

2020

DESARROLLO DE UN CONTROL PID, QUE PERMITA MONITOREAR Y MODIFICAR LA TENSIÓN DE SALIDA DE UN VARIAC MONOFÁSICO, UTILIZANDO UN MICROCONTROLADOR ESP32 Y MOTORREDUCTOR ACOPLADO AL VOLANTE.

**Jhon Jairo Suárez Ramírez
Yeici Enrique Herrera Durán**

Trabajo de grado presentado para optar por el título de:
Ingeniero Electromecánico.

Director(a)
MsC. Camilo Leonardo Sandoval.

Línea de Investigación:
Automatización Industrial

Universidad Antonio Nariño
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Bucaramanga, Colombia

2020

DEDICATORIA

Los autores de trabajo lo dedican a Dios por darles vida y salud, a sus familiares quienes continuamente los apoyaron en momentos difíciles motivándolos siempre a llegar a la meta trazada.

AGRADECIMIENTOS

Los autores de este trabajo agradecen a Dios y a la Universidad Antonio Nariño por formarnos como ingenieros electromecánicos.

RESUMEN

Título: Desarrollo de un control PID, que permita monitorear y modificar la tensión de salida de un Variac monofásico, utilizando un microcontrolador esp32 y MOTORREDUCTOR acoplado al volante.

Palabras clave: Arduino, Labview, Autotransformador, microcontrolador, PID.

Descripción:

El desarrollo de un PID para controlar la tensión de salida de un Variac es el principal objetivo de este trabajo; esto con el ánimo de resolver la hipótesis que plantea ¿Cuál es la mejor alternativa modernizar el control manual de un Variac y convertirlo en un equipo automatizado? Por esta razón se desarrolla una tarjeta controladora, y caracteriza el sistema para encontrar la ecuación de transferencia y las constantes PID utilizando el IDENT de Matlab, seguido a esto se elaboró elaborar una interfaz HMI en Labview para operar y monitorear el Variac, finalmente se validó la funcionalidad del sistema desarrollado al determinar el error e identificar la línea de tendencia del mismo para reducirlo al máximo.

ABSTRACT.

Title: Development of a PID control, which allows monitoring and modifying the output voltage of a single-phase Variac, using an esp32 microcontroller and MOTOR REDUCER coupled to the steering wheel.

Key words: Arduino, Labview, Autotransformer, microcontroller, PID.

Description:

The development of a PID to control the output voltage of a Variac is the main objective of this work; This with the aim of solving the hypothesis that arises. What is the best alternative to modernize the manual control of a Variac and turn it into automated equipment? For this reason a controller card is developed, and it characterizes the system to find the transfer equation and the PID constants using the Matlab IDENT, followed by the elaboration of an HMI interface in Labview to operate and monitor the Variac, finally it was validated the functionality of the system developed by determining the error and identifying its trend line to reduce it as much as possible.

Contenido

	Pág.
Introducción.....	14
Planteamiento del problema.	15
Objetivos del proyecto.....	17
Objetivo General.	17
1. Marco teórico.	18
1.1. Transformadores.	18
1.1.1. El Autotransformador.....	20
1.2. Sistemas de Control de Posición y velocidad	22
1.2.1. Sistemas de Control en lazo cerrado.....	23
1.3. Creación de instrumentos virtuales en Labview.....	23
1.4. Historia de los microcontroladores.....	24
1.4.1. Que es un microcontrolador.	25
1.5. Sistemas de transmisión de energía mecánica.	25
1.6. Cajas de engranajes.....	26
2. Desarrollo de sistema de control PID.	28
2.1. Driver L298 y realimentación de tensión de salida.	28
2.1.1. Características del Driver L298.....	29
3.1.2 Relación entre posición del volante y tensión de salida.	32
3.1.3. PID utilizando IDENT de Matlab.	34

3.1.4. Tabulación de los resultados.	35
3.1.5. Características adicionales al controlador.	43
3.1.6. Algoritmo programado para la operación del Variac.	45
3.2. Distribución de componentes utilizando el software SolidWords.	49
3.2.1. Esquemas desarrollados en Solidwords.	50
3.2.2. Distribución de los elementos dentro del tablero de control.	52
3.2.3. Implementación final.	52
4. Interfaz Hombre Maquina para operar el Variac.	54
4.1. Panel frontal en Labview.	54
4.1.1. Elementos de interacción con el Variac desde el panel frontal.	56
4.2. Diagrama de bloques.	60
4.2.1. Algoritmo programado en Labview.	60
4.2.2. Programación en lenguaje gráfico.	62
5. Validación del sistema de control implementado en el Variac.	65
5.1. Prueba de control de tensión.	65
6. Resultados y análisis.	68
6.1. Pruebas de comunicación.	68
6.2. Corrección del error de la medida	69
7. Conclusiones.	72
8. Bibliografía.	74

Listado de figuras

	Pág.
Figura 1 Propuesta de automatización.....	15
Figura 1 Devanados de autotransformador.....	16
Figura 2 Transformador de potencia	19
Figura 3 Circuito de un autotransformador.....	21
Figura 4 Autotransformador Anular	22
Figura 5 VISA READ.....	24
Figura 6 VISA WRITE.	24
Figura 7 Diagrama general de un sistema microcontrolador.....	25
Figura 8 Esquema de una transmisión combinada.	26
Figura 9 Interior de una caja reductora	27
Figura 10 Variables a monitorear y controlar por el sistema.	28
Figura 11 Sensor de posición del volante.	29
Figura 12 Lazo de control de tensión de salida del Variac.....	29
Figura 13 Conexiones del L298.	31
Figura 14 Conexiones del ESP32 al L298.....	32
Figura 15 Caracterización de voltaje de salida.....	33
Figura 16 Importar información de PWM y tensión de salida.....	35
Figura 17 Confidence Region un polo 15% PWM.	36
Figura 18 Confidence Region dos polos 15% PWM.	37

Figura 19 Confidence Region tres polos 15% PWM.	37
Figura 20 Confidence Region un polo 37% PWM.	38
Figura 21 Confidence Region dos polos 37% PWM.	39
Figura 22 Confidence Region tres polos 37% PWM.	39
Figura 23 Confidence Region un polo 80% PWM.	40
Figura 24 Confidence Region dos polos 80% PWM.	41
Figura 25 Confidence Region tres polos 80% PWM.	41
Figura 26 Respuesta del control diseñado.	42
Figura 27 Parada de emergencia y salida de tensión.	43
Figura 28 Conexión eléctrica entre parada de emergencia y ESP32.	44
Figura 29 Indicadores piloto.	45
Figura 30 Circuito cableado para alimentar indicadores piloto.	45
Figura 31 Algoritmo 1, operación de entradas/salida digitales.	46
Figura 32 Algoritmo de comunicación.	47
Figura 33 Algoritmo para desactivar la salida del Variac cuando la tensión supera 130V	48
Figura 34 Algoritmo PID para control de posición.	49
Figura 35 Esquema de implementación en SolidWords.	50
Figura 36 Acople de eje del motor a volante del Variac.	51
Figura 37 Implementación final en Solidwords.	51
Figura 38 Distribución de los elementos en la caja de control.	52
Figura 39 Implementación final de Variac controlador por ESP32.	53
Figura 40 Interfaz hombre máquina para operar el Variac.	54

Figura 41 Configuración de puerto serial y alarmas en Labview.....	55
Figura 42 Tendencia de Setpoint y voltaje de salida.....	55
Figura 43 Indicador de Setpoint para control y visualizador de tensión de entrada.	57
Figura 44 Manipulación de MC desde Labview.....	57
Figura 45 Activación de PID desde Labview.	58
Figura 46 Indicación de MC y PID en el tablero de control.	58
Figura 47 Comunicación normal y en falla.	59
Figura 48 Presentación de fallas en panel frontal.	59
Figura 49 Establecer Setpoint desde panel frontal.....	60
Figura 50 Algoritmo de lectura de buffer serial.....	61
Figura 51 Algoritmo para escritura de buffer serial.	62
Figura 52 Programa para escribir en buffer serial.....	63
Figura 53 Programa para leer buffer serial.....	64
Figura 54 Seleccionando puerto de comunicación desde Labview.	65
Figura 55 Resultados de pruebas en ascenso y descenso de voltaje.....	67
Figura 56 Visualización en Labview de pruebas de control de voltaje.	67
Figura 57 Información en IDE de Arduino.	68
Figura 58 Lectura de voltaje después de String Subjet.....	69
Figura 59 Tendencia del error de voltaje.....	70

Listado de tablas.

	Pág.
Tabla 1 Criterios para seleccionar driver de motor.....	30
Tabla 2 Valor de importancia para cada criterio para el driver del motor.	30
Tabla 3 Lógica para operar el L298.	32
Tabla 4 Caracterización voltaje de salida.....	33
Tabla 5 Resultados del modelo de salida con diferentes entradas 15%.	36
Tabla 6 Resultados del modelo de salida con diferentes entradas 37%.	38
Tabla 7 Resultados del modelo de salida con diferentes entradas 80%.	40
Tabla 8 Parámetros del controlador.	42
Tabla 9 String de comunicación entre Esp32 y Labview.	47
Tabla 10 Indicación de control del moto reductor.....	56
Tabla 11 Funciones del algoritmo desarrollado en Labview.....	61
Tabla 12 Bloques utilizados para escritura del buffer serial.	63
Tabla 13 Resultados de pruebas de control de voltaje en ascenso y descenso.	66
Tabla 14 Error de medidas de voltaje.....	70
Tabla 15 Corrección de la tensión de salida.	71

Listado de ecuaciones.

	Pág.
Ecuación 1 Relación de transformación	20
Ecuación 2 Potencia de paso en un autotransformador	21
Ecuación 3 Relación lineal posición y tensión de salida.....	34
Ecuación 4 Relación polinómica grado dos de posición y tensión de salida.....	34
Ecuación 5 Relación polinómica grado tres de posición y tensión de salida.....	34
Ecuación 6 Función de transferencia para Variac.....	42
Ecuación 7 Calculo de error porcentual.	69
Ecuación 8 Tendencia del error de voltaje	70

Introducción

Con la ejecución de trabajo se evidencia que posible automatizar un Variac y dejar las bases técnicas y metodológicas para lograr tal fin. Para ello se hace uso de un microcontrolador ESP32 programado desde el IDE de Arduino a este llegan diferentes señales que permiten relacionar la tensión de salida con la cuales se retroalimenta el lazo de control PID; en la figura 1 se observa una representación de la implementación que sería una posible solución al problema planteado.

Figura 1 Propuesta de automatización



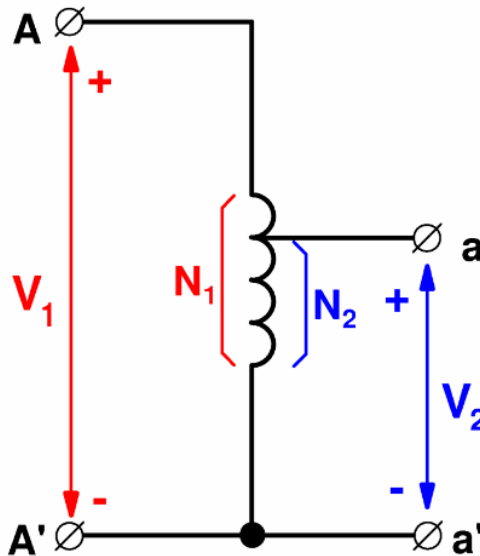
Fuente: Autores.

Planteamiento del problema.

El autotransformador es un tipo de transformador cuya principal diferencia es que este segundo se encuentra aislado eléctricamente entre los devanados del

primario y secundario, dado que el autotransformador solo cuenta con un devanado de donde salen todas las conexiones (ver figura 2).

Figura 2 Devanados de autotransformador.



Fuente: Autotransformador. Recuperado de <http://www.sectorelectricidad.com/11001/el-autotransformador-ventajas-y-desventajas/>

La problemática que se ha identificado es la falta de sistemas de control de posición del volante de un Variac que permita modificar la tensión de salida del mismo, y de esta forma evaluar el comportamiento de los controladores PID para este tipo de sistemas aplicando de esta forma conocimientos adquiridos en asignaturas como sistemas de control, instrumentación virtual, entre otras. Siendo importante señalar que los Variac tienen aplicaciones a nivel industrial para conectar maquinaria fabricada para tensiones nominales diferentes a las de la fuente de alimentación y en la academia los autotransformadores son utilizados por ejemplo en prácticas de arranque de motores de inducción a tensión reducida. Por esta razón estos equipos son usados para diversas pruebas de laboratorio para la enseñanza de carreras como tecnología e

ingeniería, sin embargo se evidencia que ellos deben ser manipulados manualmente mediante un volante que modifica el número de espiras y de esta forma varia la tensión de salida, esto a pesar de la evolución de los sistemas de control, la electrónica de potencia y desarrollo de software, lo cual hace estos equipos se vean obsoletos dentro de un proceso de formación académica, por esta razón se plantea la pregunta de investigación ¿Cuál es la mejor alternativa modernizar el control manual de un Variac y convertirlo en un equipo automatizado? Esta tesis de grado dará solución a este planteamiento ya que identificará las características del sistema para realizar el diseño de un controlador PID utilizando el IDENT de Matlab y un microcontrolador ESP32.

Objetivos del proyecto.

Objetivo General.

Desarrollar un controlador PID, para modificar y monitorear la tensión de salida de un Variac monofásico de 0 – 110V, utilizando un microcontrolador ESP32 y un motorreductor.

Objetivos Específicos.

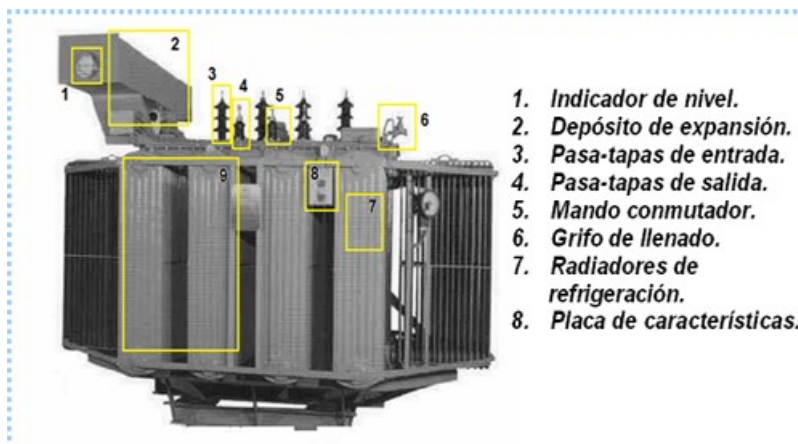
- Desarrollar una tarjeta controladora, para implementar el algoritmo PID, basando su funcionamiento en un microcontrolador ESP32 y caracterizando el sistema utilizando el IDENT de Matlab.
- Desarrollar un algoritmo en Labview, que sirva de interface hombre maquina HMI, utilizando las funciones VISA.
- Validar el funcionamiento del controlador desarrollado, para comprobar que realiza las funciones de modificar y monitorear automáticamente la salida de tensión del autotransformador.

1. Marco teórico.

1.1. Transformadores.

El transformador es una máquina eléctrica estática (ver figura 3), destinada a funcionar con corriente alterna, constituida por dos arrollamientos, primario y secundario, que permiten transformar energía eléctrica con magnitud de V-I determinadas, a otras con valores en general diferentes [1]. La importancia de los transformadores, se debe a que gracias a ellos ha sido posible el enorme desarrollo de la industria eléctrica, al haber logrado el transporte de la energía eléctrica a grandes distancias. Téngase en cuenta que el transporte de energía eléctrica desde centrales de generación, hasta los centros de consumo es más económica cuanto más alto sea el nivel de tensión en la transmisión, puesto que las corrientes disminuyen, reduciendo de esta forma las pérdidas por efecto joule¹.

Figura 3 Transformador de potencia



Fuente: Jesús Fraile Mora, Maquinas eléctricas sexta edición, editorial Mc Graw Hill, p. 181.

Sacrificando rigor, para ganar concreción, y en términos ideales útiles, puede añadirse que la función de esta máquina consiste en transformar la energía (potencia),

¹ Nombre dado en honor al físico inglés **James Prescott Joule**

en el sentido de alterar sus factores según la relación de transformación como lo indica la ecuación 1 [2].

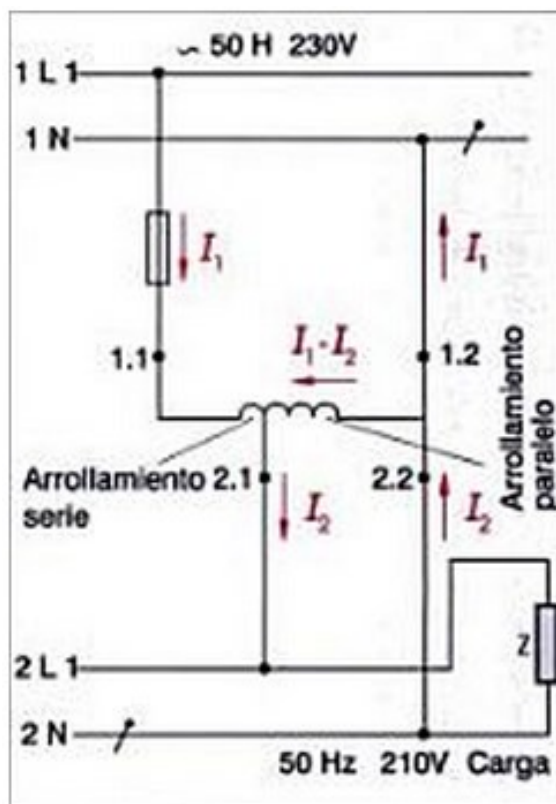
Ecuación 1 Relación de transformación

$$\frac{U_1}{U_2} \approx \frac{I_2}{I_1}$$

Fuente: Transformadores de potencia de medida y protección, pág. 5

1.1.1. El Autotransformador.

En el autotransformador hay dos partes del arrollamiento dispuestas en serie que son el arrollamiento paralelo y el arrollamiento serie (ver figura 4). El arrollamiento paralelo es el arrollamiento de tensión inferior. Al transformar reduciendo queda en paralelo con la carga. El arrollamiento de tensión superior es la conexión en serie del arrollamiento paralelo y del arrollamiento serie.

Figura 4 Circuito de un autotransformador

Fuente: Peter Bastián, electrotecnia, Ediciones AKAL, p. 276.

La potencia total que se puede tomar de un autotransformador se llama potencia de paso. Esta se transmite en parte por el paso de corriente del arrollamiento de entrada, y por otra parte del flujo magnético en el núcleo del hierro. Cuanto más se aproxime la relación de transmisión al valor 1 (ver ecuación 2), tanto menor es la potencia transmitida por inducción [3].

Ecuación 2 Potencia de paso en un autotransformador

$$S_B = \frac{U_2 - U_1}{U_2} \cdot S_D$$

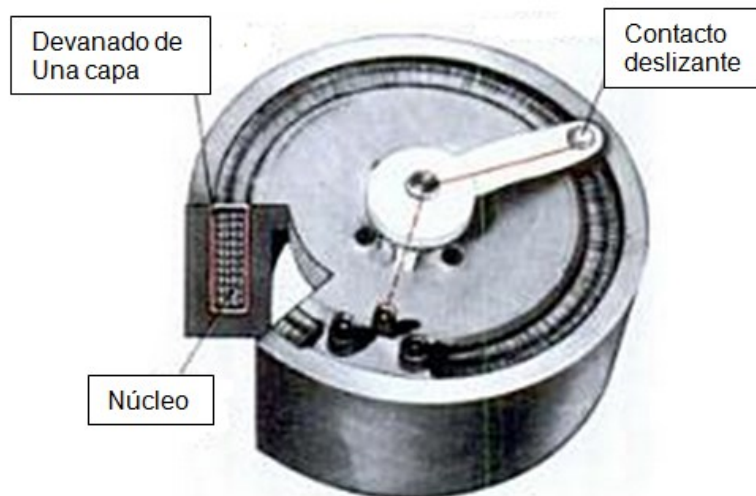
Fuente: Peter Bastian, electrotecnia, Ediciones AKAL, p. 276.

Dónde:

- S_B , Potencia de construcción
- U_1 , Tensión de entrada
- U_2 , Tensión de salida
- S_D , Tensión de paso.

Aun cuando los autotransformadores presentan similitudes con los divisores de tensión, se prefieren los primeros, en la técnica de las corrientes industriales, porque presentan menores pérdidas y porque la tensión de salida es más independiente de la carga que en los divisores de tensión. Los pequeños autotransformadores se construyen en forma de los transformadores variables de núcleo anular (figura 5), de modo similar a los potenciómetros giratorios [4].

Figura 5 Autotransformador Anular



Fuente: Adolf Senner, Principios de electrotecnia, editorial REVERTE, p. 186.

1.2. Sistemas de Control de Posición y velocidad

Este proyecto tiene como fin controlar la tensión de salida de un Variac de 1.2kW, para ello será necesario controlar la posición del volante ubicado en la parte

superior del mismo, es por tal motivo que se hace necesaria la investigación de sistemas de control en lazo cerrado destinados a controlar velocidad y posición.

1.2.1. Sistemas de Control en lazo cerrado

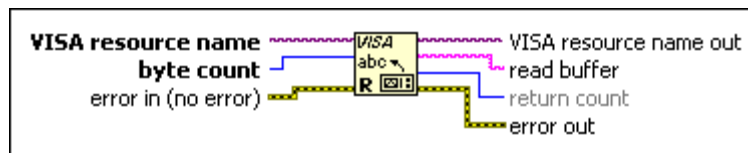
Un sistema de control mantiene una relación determinada entre la salida y la entrada de referencia, comparándolas y usando la diferencia como medio de control, se denomina control realimentado [5]. Un ejemplo sería el control de posición del volante de un autotransformador, midiendo la tensión de salida y comparándola con un valor de referencia, un motor eléctrico se encargaría de mover el volante y variar la tensión de salida hasta lograr que el error entre el valor de referencia y la salida real sea lo más cercano a cero. Los sistemas de control no se limitan a la ingeniería, sino que también se encuentran en diversos campos ajenos a ella. Por ejemplo, el cuerpo humano es un sistema de control realimentado muy avanzado. Tanto la temperatura corporal como la presión sanguínea se conservan constantes mediante una realimentación fisiológica. En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error y llevar la salida del sistema a un valor deseado.

1.3. Creación de instrumentos virtuales en Labview.

En el documento “Entorno gráfico de programación” [6] se indica que la programación se puede realizar mediante el uso de controles que sirven de entrada de datos y

funciones que realizan operaciones con esos datos. Para el caso de este trabajo es necesario tener claridad sobre la comunicación serial cuyas características principales son velocidad, número de bits del dato, paridad, control de flujo. En la figura 6 se presenta el bloque VISA READ, al cual es necesario indicarle el número de bytes que debe leer del buffer, en la figura 7 esta el bloque VISA WRITE, el cual escribe datos en el buffer.

Figura 6 VISA READ.



Fuente: Autores.

Figura 7 VISA WRITE.

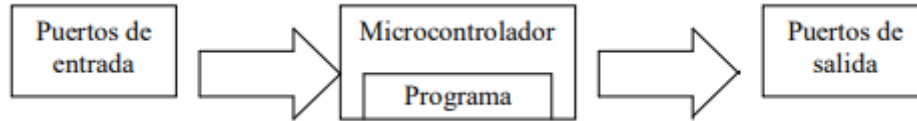


Fuente: Autores.

1.4. Historia de los microcontroladores.

Desde el año de 1970 apareció el microprocesador trayendo consigo la época dorada para el Z-80, 808 entre otros, los cuales eran utilizados como elementos centrales en aparatos de control [10]; en el año de 1980 los fabricantes de estos dispositivos dieron a conocer un nuevo elemento llamado microcontrolador el cual contaba con una CPU, memoria RAM, memoria ROM y circuitos de entrada y salida (ver figura 8), desde ese momento cambio el diseño de los productos.

Figura 8 Diagrama general de un sistema microcontrolador.



Fuente: Pichucho, J. (2007). Modulo para verificar el funcionamiento de los programas grabados en el PIC16F84A.

1.4.1. Que es un microcontrolador.

Es un circuito integrado programable que tiene todos los elementos que componen un computador y es utilizado para controlar el funcionamiento de una tarea determinada; sus pines de entrada y salida pueden ser utilizados para conectar motores, relays etc. [10]. Las partes básicas son:

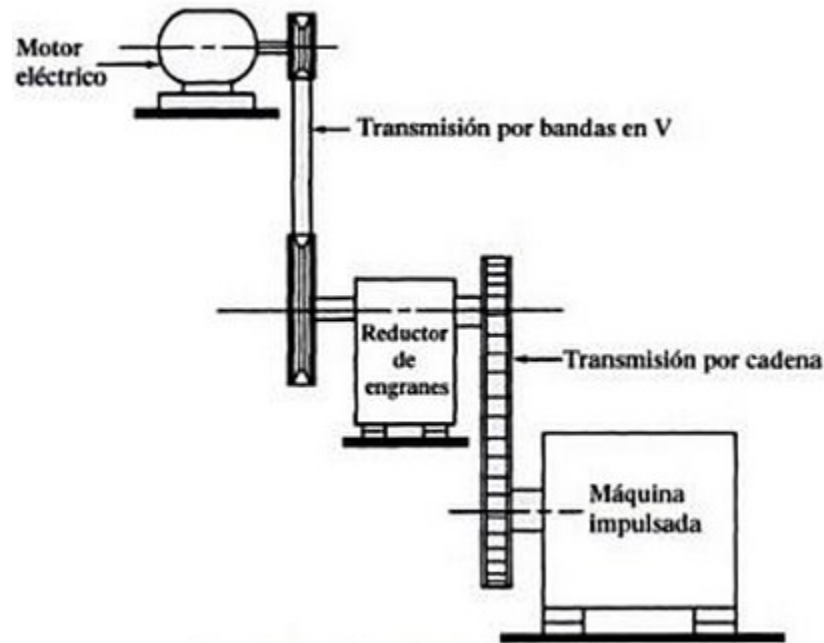
- ROM: memoria de solo lectura.
- RAM: memoria de acceso aleatorio.
- Líneas de I/O: permiten conectar elementos externos.
- Lógica de control: coordina la interacción entre los demás bloques.

1.5. Sistemas de transmisión de energía mecánica.

Las bandas y las cadenas son los tipos principales de elementos flexibles de transmisión de potencia. Las bandas trabajan con poleas, mientras que las cadenas trabajan con ruedas dentadas. La figura 9, muestra una aplicación industrial típica de estos elementos, combinados con un reductor de velocidad con engranajes. Esta

aplicación ilustra donde se usan las bandas, engranes y cadenas con el mayor provecho, es necesario estar documentado sobre estos temas dado que para este trabajo se utilizará un motorreductor [7].

Figura 9 Esquema de una transmisión combinada.



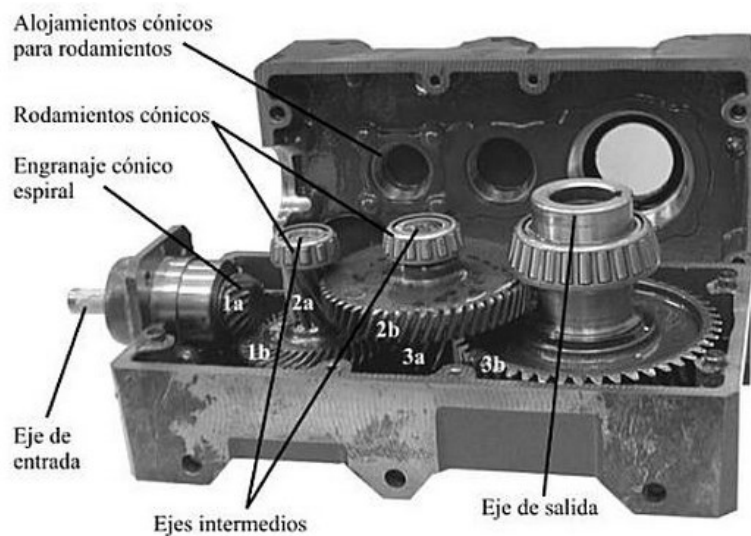
Fuente: Diseño de elementos de máquinas, p. 266.

1.6. Cajas de engranajes

Las cajas de engranajes o cajas reductoras, están compuestas de trenes de engranajes que poseen ejes intermedios con diferentes etapas de reducción. Se utilizan para transmitir potencia cuando se requieren relaciones de transmisión o relaciones de par mayores a las logradas con engranajes simples. También se utilizan cuando se requiere una relación de transmisión variable, un cambio de sentido de giro de los ejes o un ángulo de salida diferente [8]. La mayoría de las cajas reductoras

cuentan con este tipo de trenes. En ellos los ejes mecánicos intermedios no poseen ningún tipo de movimiento de translación, solamente giran alrededor del eje geométrico. En la figura 10 se muestra un ejemplo de una caja reductora, formada por un tren fijo, en la que los ejes de entrada y salida son perpendiculares.

Figura 10 Interior de una caja reductora



Fuente: Mantenimiento mecánico de máquinas, p 55.

2. Desarrollo de sistema de control PID.

Para realizar el desarrollo del sistema de control PID en lazo cerrado se desarrollan los esquemas de conexión de todos los elementos y que se tiene en cuenta para la operación segura del Variac, las variables que se desean monitorear y controlar se presentan en la figura 11.

Figura 11 Variables a monitorear y controlar por el sistema.



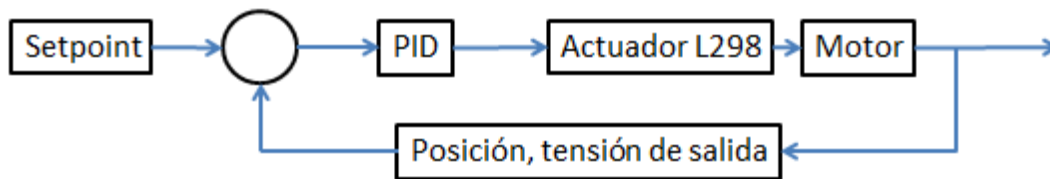
Fuente: Autores.

2.1. Driver L298 y realimentación de tensión de salida.

Como se observa en la figura 12 el sistema es realimentado por la señal de un potenciómetro lineal que se encuentra acoplado al sistema de transmisión del volante del Variac, entonces el lazo de control se presenta en la figura 13 el cual tiene como actuador el Driver L2982 y como señal de entrada la posición del volante.

Figura 12 Sensor de posición del volante.

Fuente: Autores.

Figura 13 Lazo de control de tensión de salida del Variac.

Fuente: Autores.

2.1.1. Características del Driver L298.

Este apartado tiene la intención de presentar las características del Driver que fue seleccionado para controlar el motor que modificara la posición del volante del Variac, y se presenta una comparación con otras posibles opciones mostrando que esta es la mejor alternativa, para ello se utiliza la matriz de Holmes asignando un valor entre 1 y 10 siendo este último tomado como un criterio muy importante y 1 como menos importante, de esto se obtienen los resultados de la tabla 1. Los pasos

realizados a continuación permiten obtener cual es la mejor opción para controlar el motor DC, los criterios a evaluar son:

- **Criterio 1:** Entradas digitales para controlar estados del motor.
- **Criterio 2:** Compatibilidad de niveles lógicos con la salida del microcontrolador ESP32.
- **Criterio 3:** Corriente y voltaje acordes a los consumos del motor.

Tabla 1 Criterios para seleccionar driver de motor.

Criterios para seleccionar driver.						
	1	2	3	Total filas	Valor normalizado	
Criterios	1	5	5	10	25	
	2	10	10	20	50	
	3	5	5	10	25	
	Total				40	100

Fuente: Autores.

Ahora bien, se establecen valores de importancia entre 1 y 10 y multiplicándolo por el valor de prioridad normalizada, al realizar la sumatoria de cada una de las opciones y el mayor será la mejor alternativa (ver tabla 2).

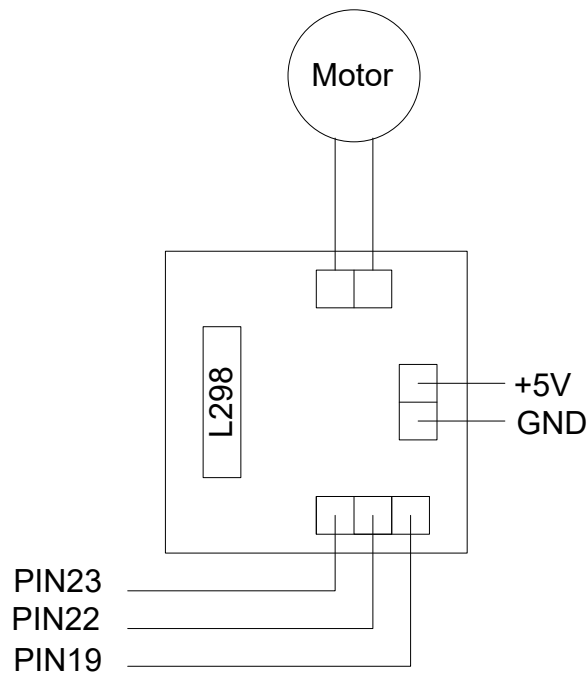
Tabla 2 Valor de importancia para cada criterio para el driver del motor.

	Valor normalizado	Prioridad	Prioridad normalizada	L298		DRV8825		TB6560	
Criterio 1	25	4	0,29	5	1,43	1	0,29	1	0,29
Criterio 2	50	6	0,43	10	4,30	5	2,15	5	2,15
Criterio 3	25	4	0,29	5	1,43	5	2,15	5	2,15
		14	1,00		7,14		4,59		4,59

Fuente: Autores.

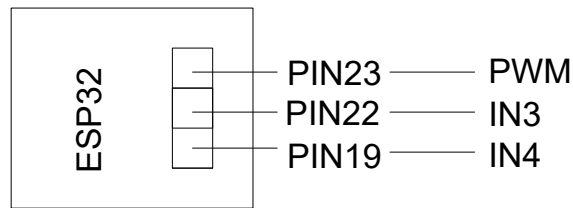
De esta forma es claro como cuales son los elementos que intervienen en el lazo de control de tensión de salida del Variac, por esta razón ahora se presentan las conexiones entre estos elementos. En la figura 14 se encuentran el L298 donde se hace referencia a los pines utilizados y la función que cumplen.

Figura 14 Conexiones del L298.



Fuente: Autores.

Las conexiones del ESP32 que van hacia el Driver L298 son las indicadas en la figura 15, y la lógica para operar este elemento se indican en la tabla 3 haciendo especial énfasis que el PIN EN2 es el que recibe la señal PWM proveniente del control PID, los otros dos pines se encargan de determinar el sentido de giro.

Figura 15 Conexiones del ESP32 al L298.

Fuente: Autores.

Tabla 3 Lógica para operar el L298.

Acción ejecutada por motor.	IN3	IN4
Motor detenido, no gira en ningún sentido.	0	0
Motor gira a la derecha	0	1
Motor gira a la izquierda	1	0
Motor detenido, no gira en ningún sentido.	1	1

Fuente: Autores.

3.1.2 Relación entre posición del volante y tensión de salida.

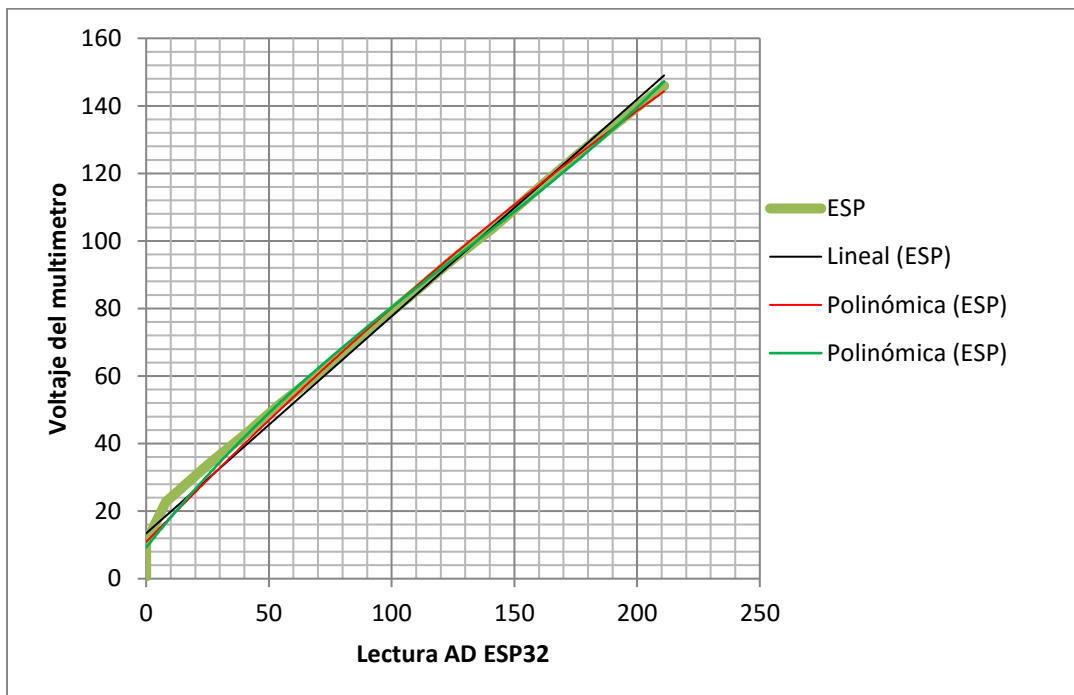
Para establecer la relación que existe entre la tensión de salida respecto a la posición del volante del Variac se realiza el siguiente procedimiento:

1. Se lleva el volante del Variac a la posición inicial.
2. Se modifica la posición del volante y se registra el valor leído por el ESP32 desde el monitor serial de Arduino.
3. Se toman los valores reales con un multímetro a la salida del Variac y se tabulan estos resultados en la tabla 4, con estos valores se logra llegar a lo presentado en la figura 16.

Tabla 4 Caracterización voltaje de salida.

Multímetro	ESP	Lineal	Error Lineal	Polinómica grado 2	Error grado 2	Polinómica grado 3	Error grado 3
22,8	8	18,56	4,24	16,8116	5,9884	16,381	6,419
33,7	25	29,49	4,21	29,2505	4,4495	30,535	3,165
43,7	42	40,41	3,29	41,4004	2,2996	43,375	0,325
51,4	54	48,12	3,28	49,8028	1,5972	51,759	-0,359
62,7	74	60,98	1,72	63,4868	-0,7868	64,701	-2,001
74,3	92	72,54	1,76	75,4604	-1,1604	75,487	-1,187
84,5	109	83,47	1,03	86,4713	-1,9713	85,154	-0,654
94,6	125	93,75	0,85	96,5705	-1,9705	93,978	0,622
103,7	141	104,03	-0,33	106,4137	-2,7137	102,712	0,988
117	162	117,53	-0,53	118,9444	-1,9444	114,329	2,671
128,1	180	129,09	-0,99	129,334	-1,234	124,694	3,406
145,9	211	149,01	-3,11	146,4677	-0,5677	144,171	1,729
		Max	4,235		5,988		6,419

Fuente: Autores.

Figura 16 Caracterización de voltaje de salida.

Fuente: Autores.

Las líneas de tendencia para funciones lineal, polinómica grado 2 y 3 se presentan en las ecuaciones 3, 4 y 5. Según lo indica la tabla 2 los errores encontrados con estas ecuaciones presentan valores de 4.24V (función lineal), 5,98 (Polinómica grado 2), 6,94 (Polinómica grado 3), por esta razón la función lineal será la utilizada para relacionar la tensión de salida con la posición del volante del Variac ya que es la que presenta el menor error.

Ecuación 3 Relación lineal posición y tensión de salida.

$$Tensión = (0,6426 * (Valor AD de ESP32)) + 13,424$$

Ecuación 4 Relación polinómica grado dos de posición y tensión de salida.

$$Tensión = (-0,0005 * (AD ESP32)^2) + (0,7482 * (AD ESP32)) + 10,858$$

Ecuación 5 Relación polinómica grado tres de posición y tensión de salida.

$$Tensión = (-0,0007 * (ESP32)^3) + ((-0,0028 * (ESP32)^2) + (0,9188 * (ESP32)) + 9,2059$$

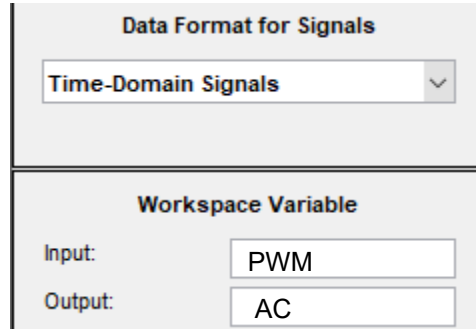
3.1.3. PID utilizando IDENT de Matlab.

Después de tener la ecuación que relaciona la posición con la tensión de salida, es posible iniciar el desarrollo de la sintonización del lazo de control PID mediante el IDENT de Matlab, para ello se realizan los pasos indicados a continuación:

1. Se ubica el volante del Variac en la posición cero o inicial.
2. Se programa la salida PWM que va al Driver L298 y se registra la tensión (ecuación 3) de salida hasta que esta se estabilice.

3. Los archivos son almacenados en formato .txt y se importan a Matlab, tomando como variables de trabajo los valores de PWM y Voltaje de salida en AC (Ver figura 17).

Figura 17 Importar información de PWM y tensión de salida.



Data Format for Signals	
	Time-Domain Signals

Workspace Variable	
Input:	PWM
Output:	AC

Fuente: Autores.

3.1.4. Tabulación de los resultados.

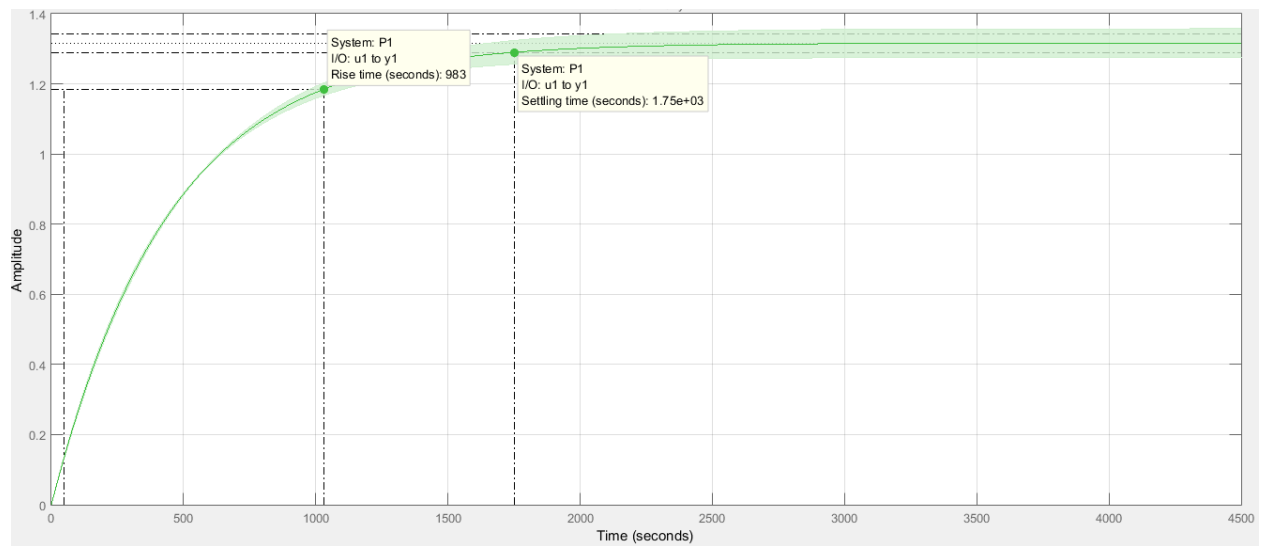
Siendo claro hasta este punto cada uno de los procedimientos realizados con el ánimo de obtener la ecuación de transferencia y las constantes PID del lazo de control, en la tabla 5 se presentan los resultados para valores de PWM de 15%, en las figuras 18, 19 y 20 las diferentes respuestas del sistema, sin embargo al evaluar la región de confianza que es un vector de desviaciones estándar en una distribución multivariada, siendo esto entonces la indicación que tan dispersos estarían los datos respecto a la media.

Tabla 5 Resultados del modelo de salida con diferentes entradas 15%.

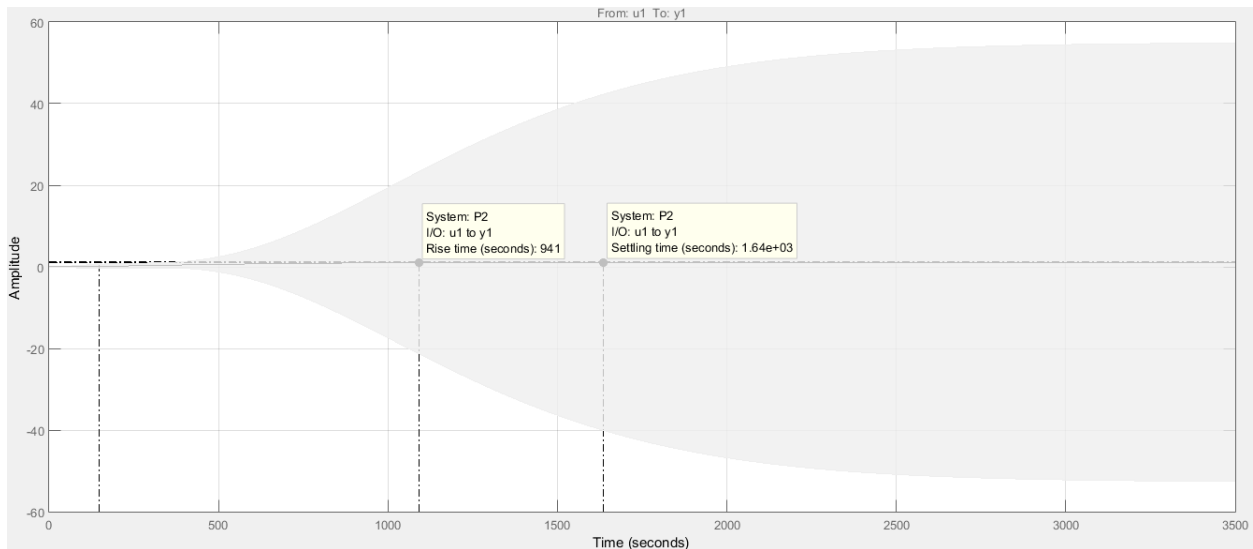
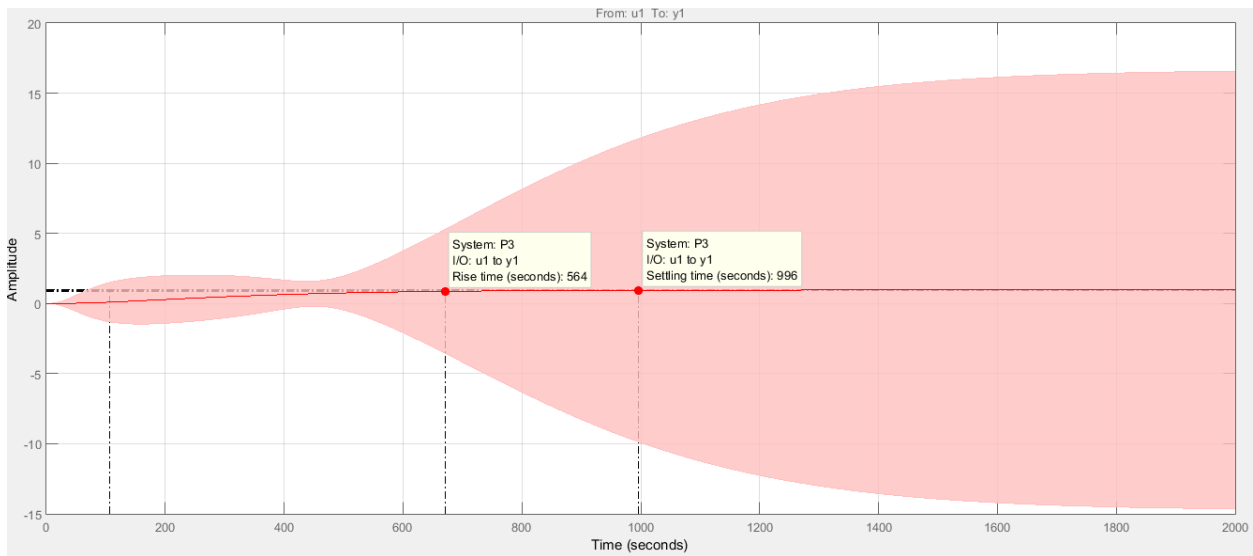
Entrada	Un polo	Dos polos	Tres polos.
15%	Kp= 1.3162 Tp1= 447.2347 Best fits %=88.25	Kp= 1.1311 Tp1= 285.459 Tp2= 275.1524 Best fits %=87.69	Kp= 0.96249 Tp1= 174.4788 Tp2= 18.4096 Tp3= 160.2235 Best fits %=84.85

Fuente: Autores.

Figura 18 Confidence Region un polo 15% PWM.



Fuente: Autores.

Figura 19 Confidence Region dos polos 15% PWM.**Fuente:** Autores.**Figura 20** Confidence Region tres polos 15% PWM.**Fuente:** Autores.

Al realizar las mismas pruebas para una salida del 37% se obtienen los valores indicados en la tabla 6 para uno, dos y tres polos, aquí se observa que el Best Fits

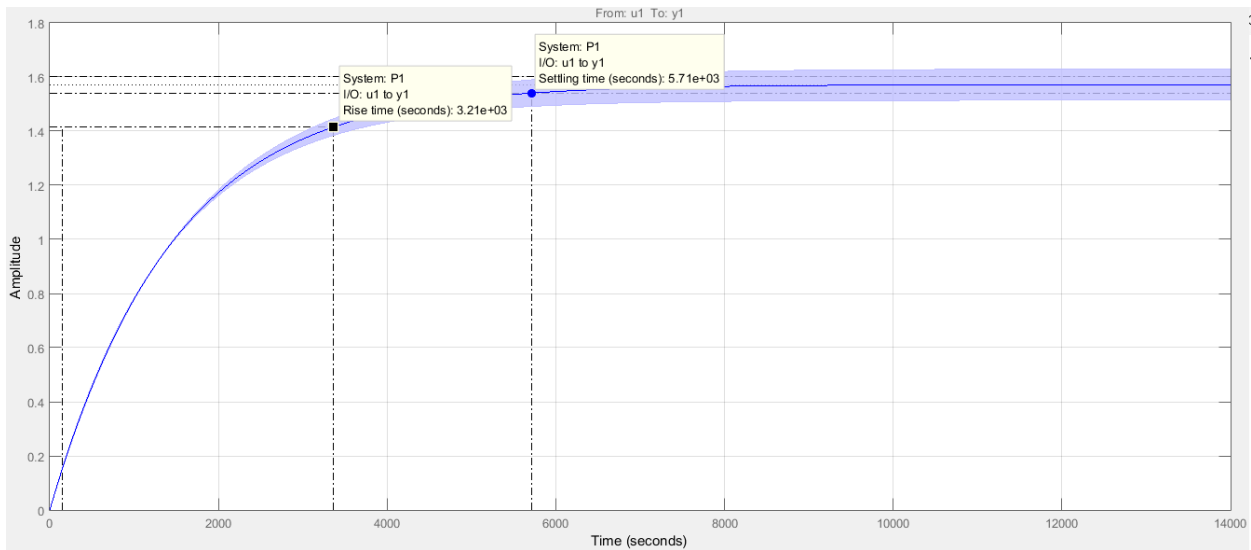
mejora, sin embargo se observa que se aumenta la zona de dispersión de los valores, ver figuras 21, 22 y 23.

Tabla 6 Resultados del modelo de salida con diferentes entradas 37%.

Entrada	Un polo	Dos polos	Tres polos.
37%	Kp= 1.5718 Tp1= 1459.579 Best fits %=92.46	Kp= 0.92902 Tp1= 288.5163 Tp2= 296.5487 Best fits%=92.38	Kp= 1.329 Tp1= 1091.0568 Tp2= 81718.638 Tp3= 0.0037919 Best fits %=92.33

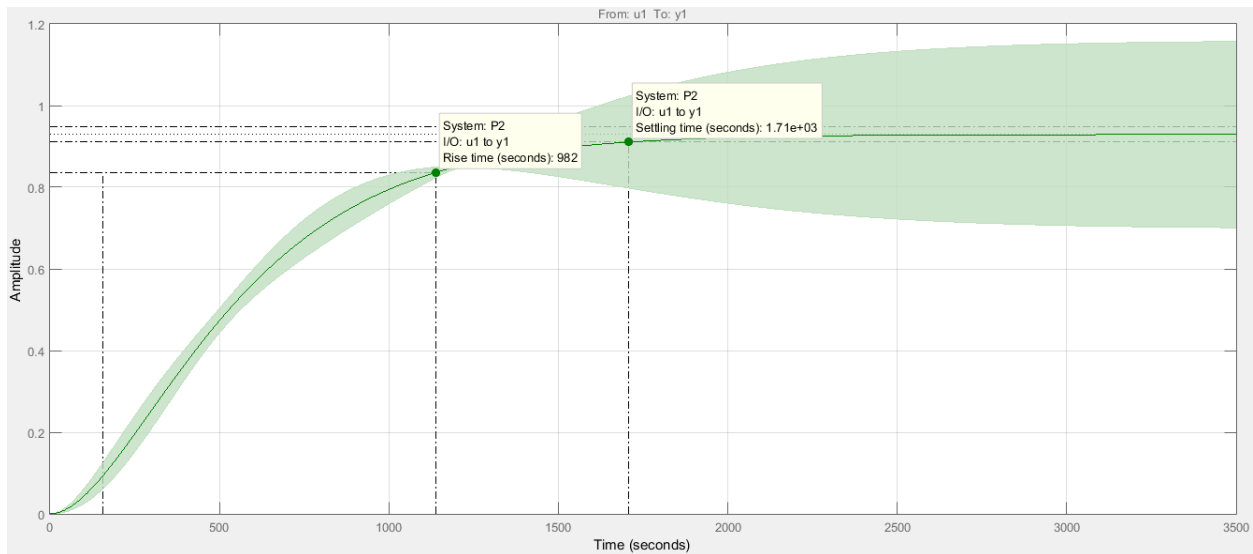
Fuente: Autores.

Figura 21 Confidence Region un polo 37% PWM.



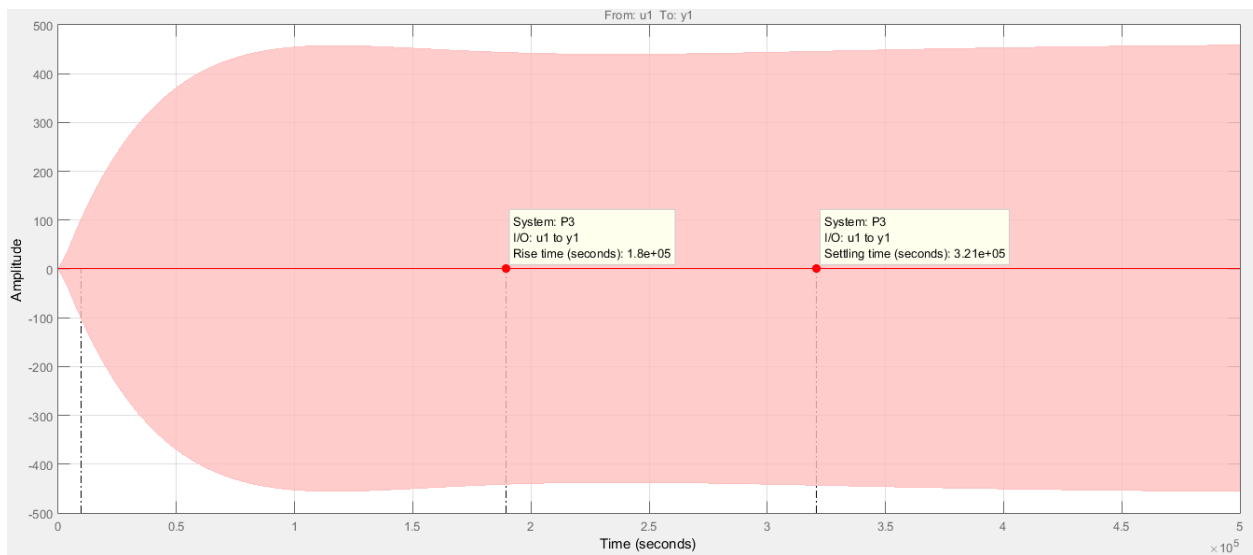
Fuente: Autores.

Figura 22 Confidence Region dos polos 37% PWM.



Fuente: Autores.

Figura 23 Confidence Region tres polos 37% PWM.



Fuente: Autores.

Finalmente se realiza la misma evaluación utilizando ahora una salida del 80% (ver tabla 7), en esta ocasión la zona de dispersión de los valores es aún mayor (ver figuras 24, 25 y 26) por esta razón se decide continuar con el diseño del controlador

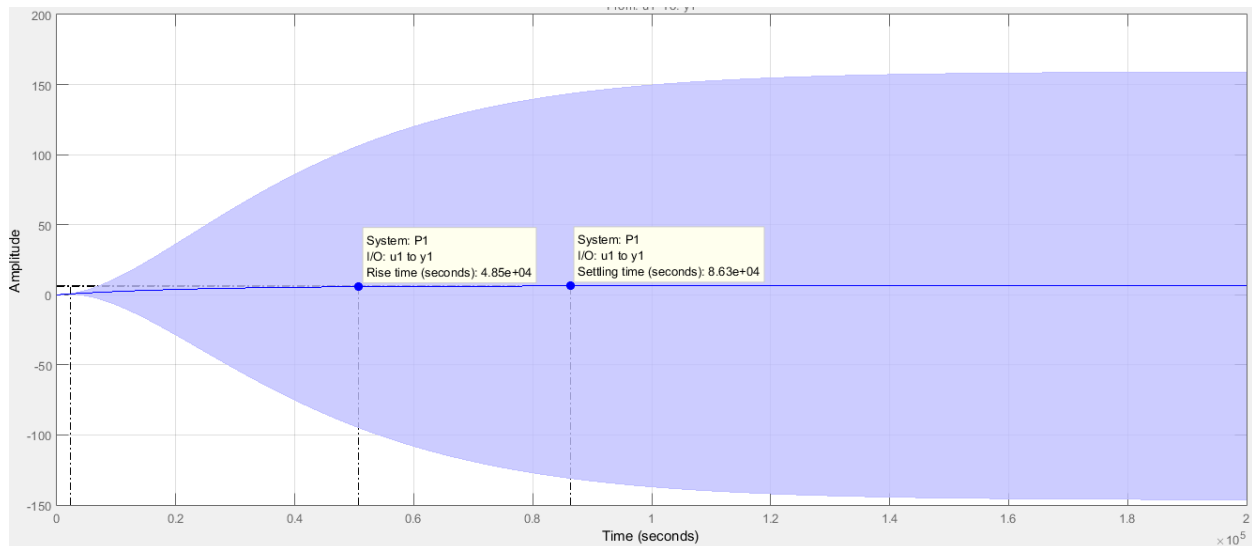
PID utilizando la herramienta “sisotool” de Matlab con los resultados obtenidos a 15% un polo.

Tabla 7 Resultados del modelo de salida con diferentes entradas 80%.

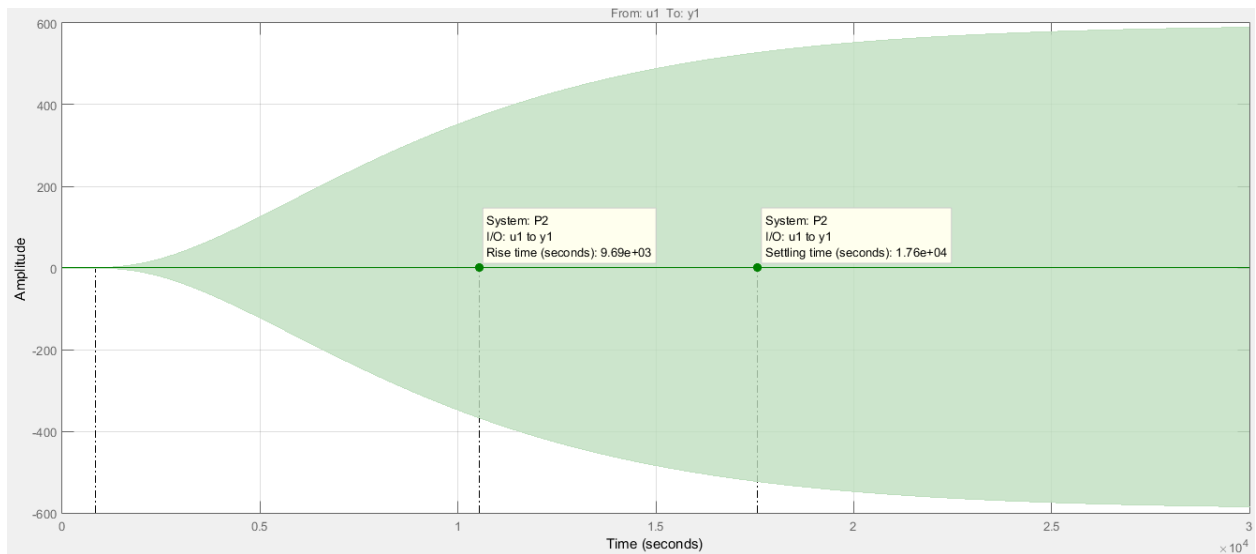
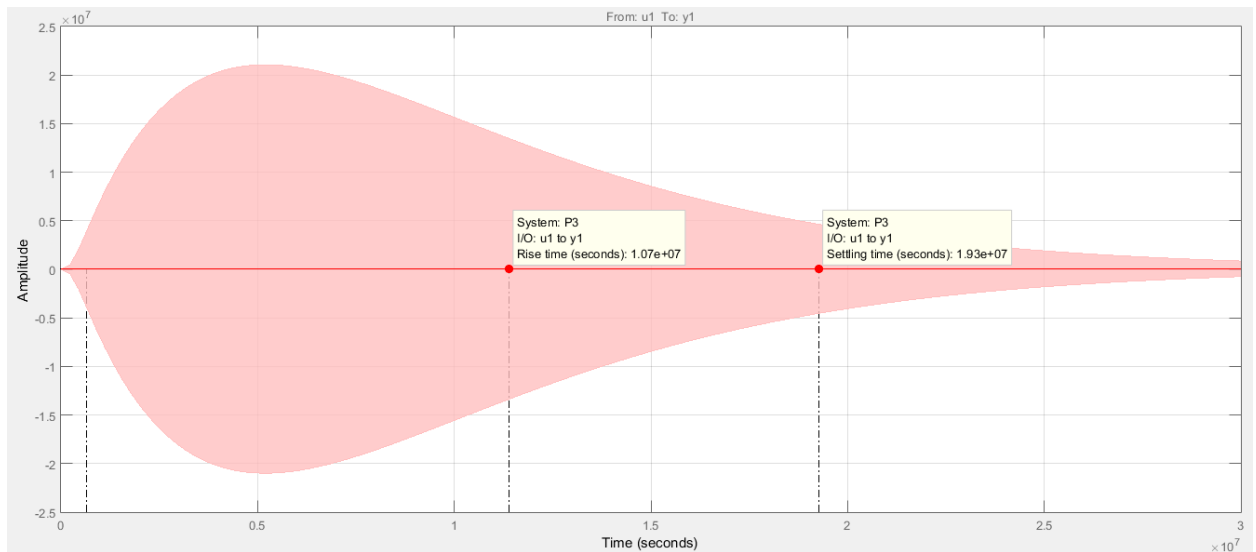
Entrada	Un polo	Dos polos	Tres polos.
80%	Kp= 6.4102 Tp1= 22054.441 Best fits %=65.91	Kp=2.2474 Tp1= 4373.025 Tp2= 441.378 Best fits %=95.34	Kp= 56136.25 Tp1= 4889885.6 Tp2= 139709.0 Tp3= 0.0634 Best fits %=96.69

Fuente: Autores.

Figura 24 Confidence Region un polo 80% PWM.



Fuente: Autores.

Figura 25 Confidence Region dos polos 80% PWM.**Fuente:** Autores.**Figura 26** Confidence Region tres polos 80% PWM.**Fuente:** Autores.

La ecuación 6 es el modelo de la función de transferencia obtenida con Matlab para el sistema implementado, siendo $K_p = 1.3162$ y $T_{p1} = 447.23$, en la tabla 8 todos los valores concernientes al lazo de control PID y en la figura 27 la respuesta del sistema.

Ecuación 6 Función de transferencia para Variac.

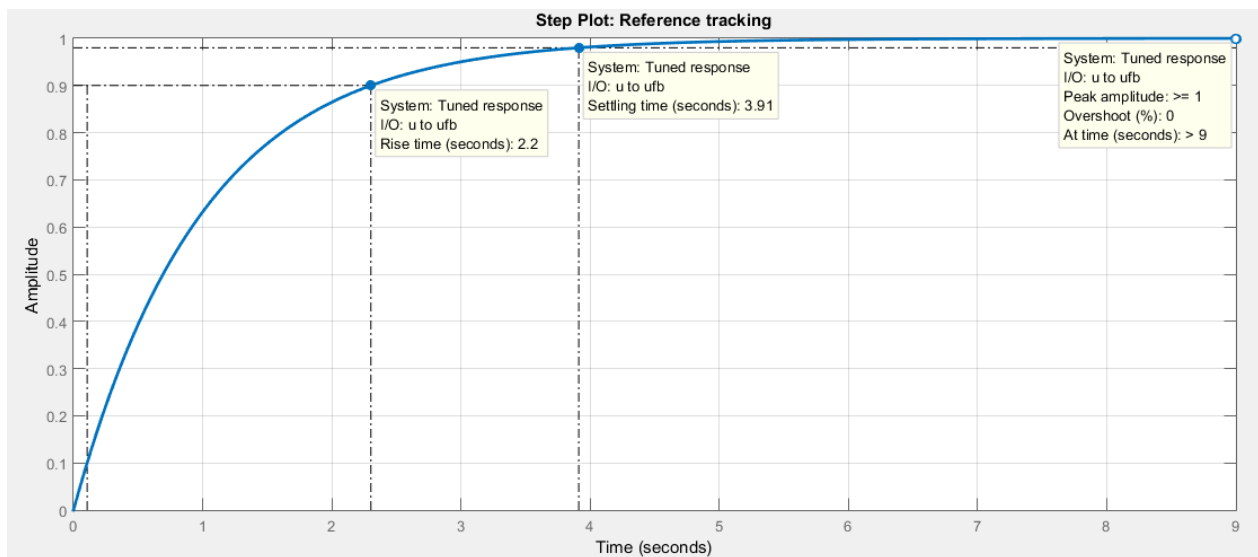
$$G_S = \frac{K_P}{1 + (T_{P1} * s)}$$

Tabla 8 Parámetros del controlador.

Kp	1.3162	Rise time	2.2 seg
Ki	0.0001	Settling time	3.91 seg
Kd	0.0001	Overshoot	0%

Fuente: Autores.

Figura 27 Respuesta del control diseñado.



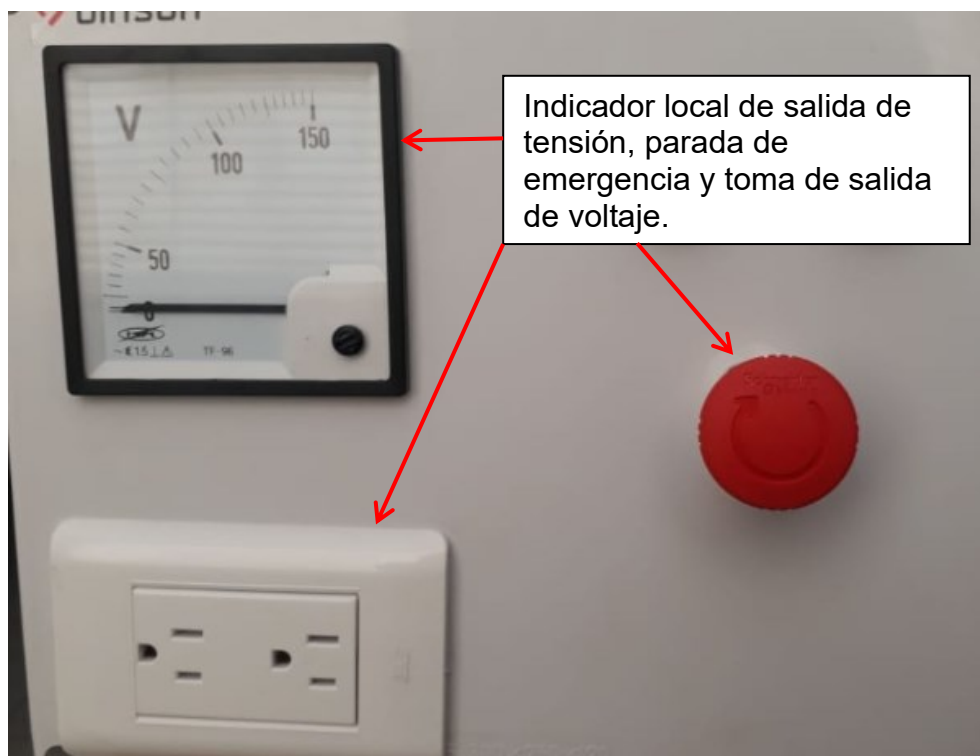
Fuente: Autores.

3.1.5. Características adicionales al controlador.

Con el ánimo de desarrollar un sistema que opere de manera segura y de fácil interpretación por parte del operador del Variac se agregaron una serie funciones al sistema y que a continuación se detallaran.

Parada de emergencia: esta será activada de manera manual y por esta razón se ubica un botón en la caja de control (ver figura 28), esta señal es llevada al microcontrolador ESP32 y mediante el algoritmo programado esto traerá como resultado la desactivación del contactor eliminando de inmediato la salida de tensión en la toma eléctrica.

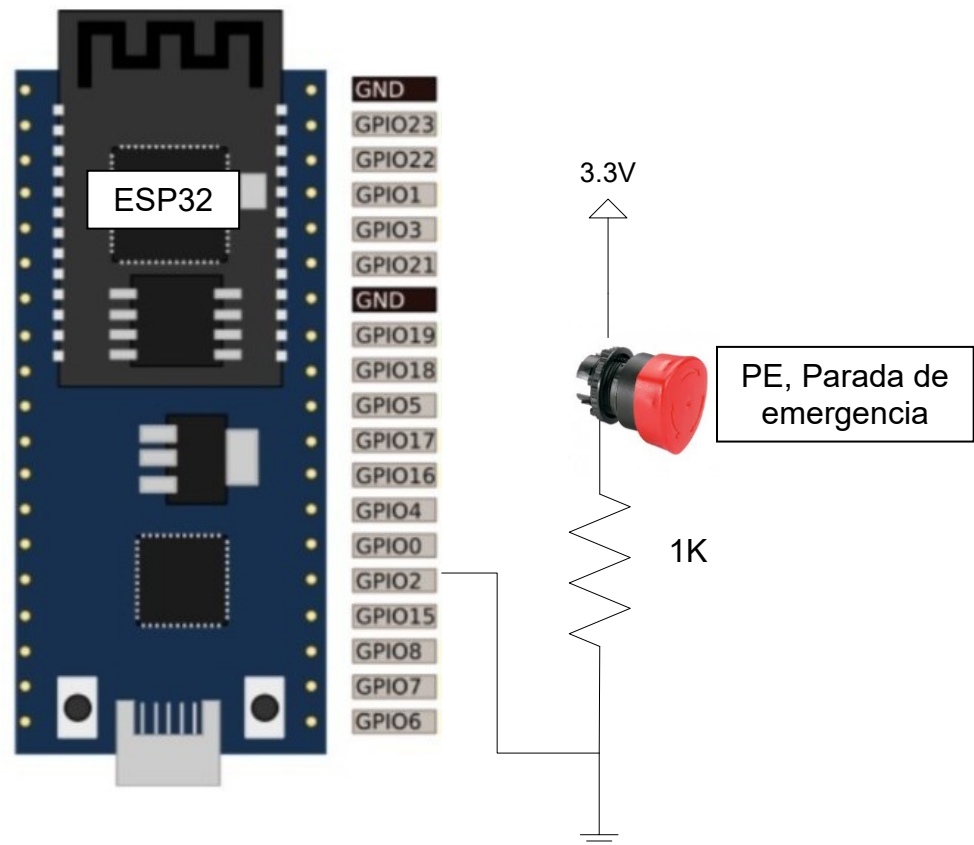
Figura 28 Parada de emergencia y salida de tensión.



Fuente: Autores.

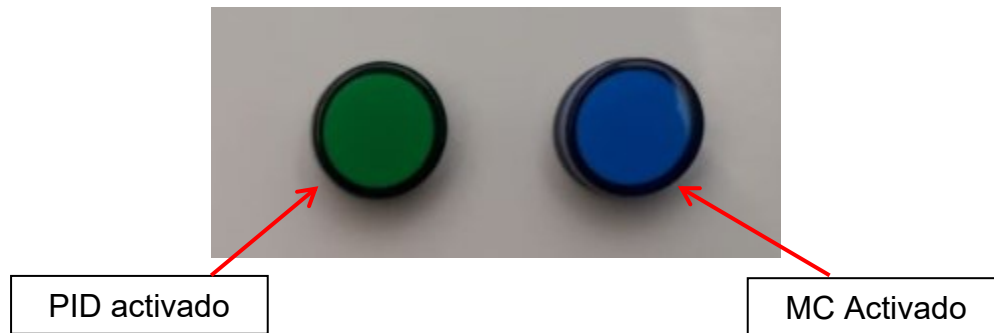
La conexión de la parada de emergencia se realiza como lo indica la figura 29, de forma tal que al activar el pulsador se cierra el contacto y se llevan 3.3V a la entrada digital 2 del ESP32.

Figura 29 Conexión eléctrica entre parada de emergencia y ESP32.



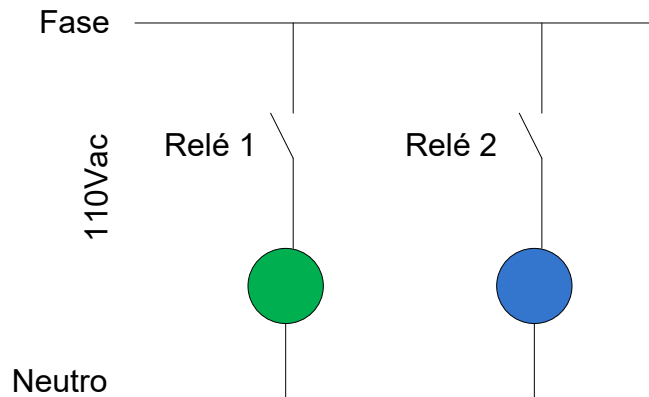
Fuente: Autores.

Señalización: esta pretende presentar en cual estado se encuentra la operación del Variac, los dos indicadores piloto presentados en la figura 30 muestran lo siguiente, el verde señala que el PID se está ejecutando normalmente y el indicador azul que el contactor que permite la salida de tensión está activado.

Figura 30 Indicadores piloto.

Fuente: Autores.

Como la salida de señal digital del ESP 32 es de máximo 3.3 V se decide utilizar relés cuya bobina actúe a este nivel de tensión y cuyo contacto soporte 130vac, el circuito para energizar estos se presenta en la figura 31.

Figura 31 Circuito cableado para alimentar indicadores piloto.

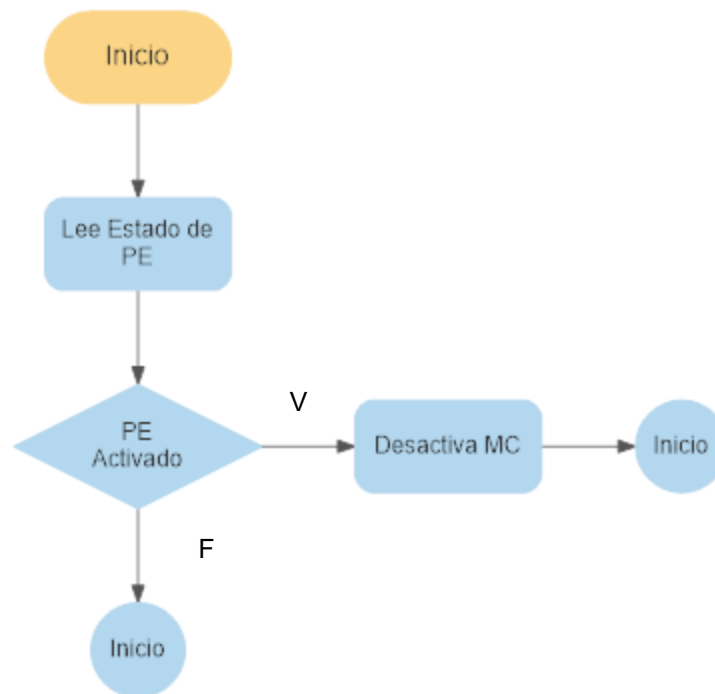
Fuente: Autores.

3.1.6. Algoritmo programado para la operación del Variac.

Todas las acciones que son realizadas por el Variac son tomadas por el algoritmo que se programó en el ESP32, y adicional a ello ejecuta una sub rutina que permite

establecer la comunicación con el software Labview. Para iniciar una parte del código se encarga de realizar la lectura de las entradas digitales para monitorear el estado de la parada de emergencia y actuar sobre el contactor, este diagrama de flujo se presenta en la figura 32.

Figura 32 Algoritmo 1, operación de entradas/salida digitales.



Fuente: Autores.

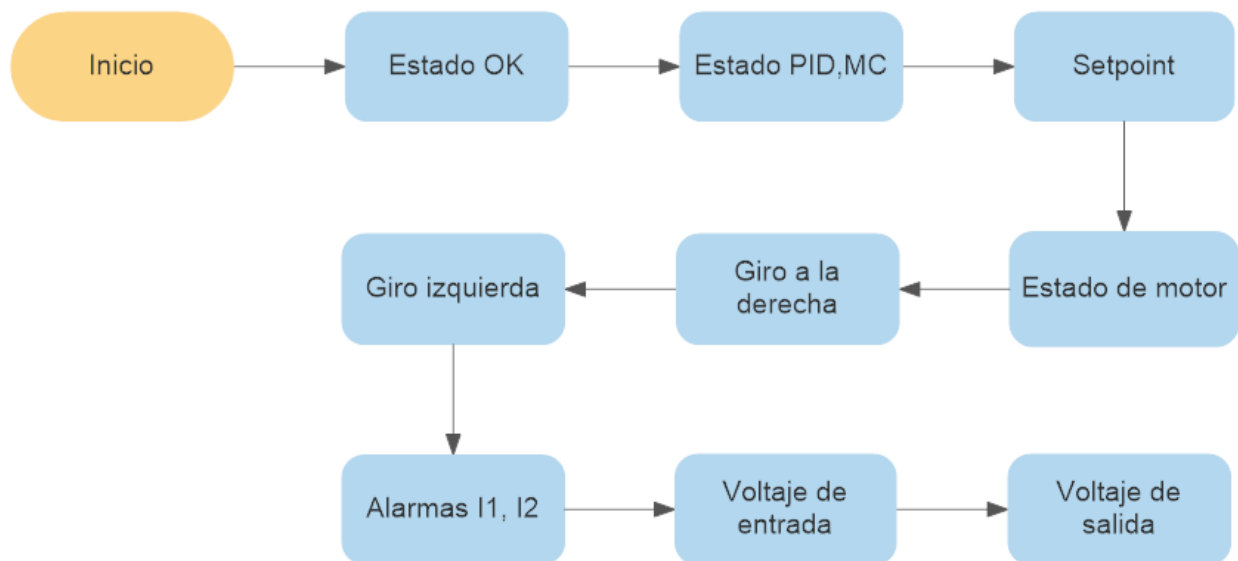
Para programar la rutina que se encarga de realizar la comunicación con Labview se realiza una investigación previa para determinar cuál es el bloque a utilizar que permita interpretar la información enviada por el ESP32, de aquí se decide utilizar el “String Subset” dado que siempre se enviarán la misma cantidad de datos. Para entender mejor lo mencionado anteriormente se escribe la tabla 9.

Tabla 9 String de comunicación entre Esp32 y Labview.

Bits	0-9	10	11	12-14	17	18	19	20	21	22-24	25-27
Des.	Estado de comunicación	Estado de PID	Estado de MC	Setpoint	Motor ON	Giro der.	Giro izq.	Alarma I1	Alarma I2	Voltaje entrada	Voltaje salida.

Fuente: Autores.

Entonces a través del algoritmo de la figura 33 es posible identificar cuáles son las acciones ejecutadas por el ESP32 de tal forma que envíe la información tal como se indica en la tabla 9, de esta forma es posible organizar dichos datos en el panel frontal de Labview.

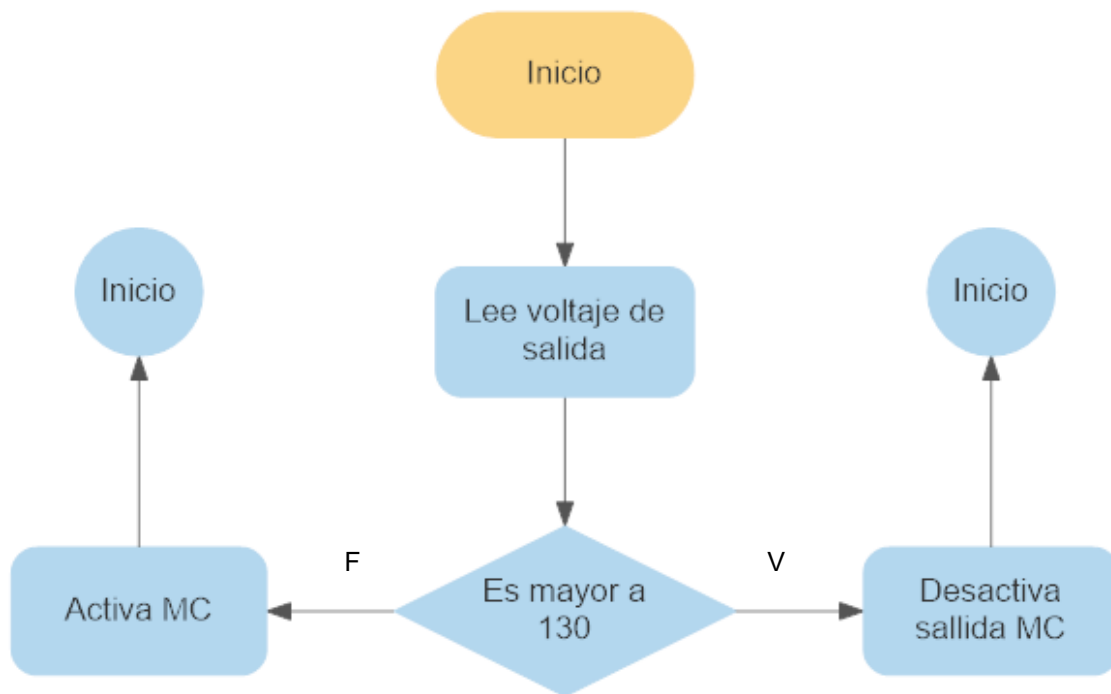
Figura 33 Algoritmo de comunicación.

Fuente: Autores.

Una alarma adicional a la parada de emergencia es la desactivación de la salida del contactor cuando la tensión del Variac sea superior a 130Vac, dado que la mayor parte

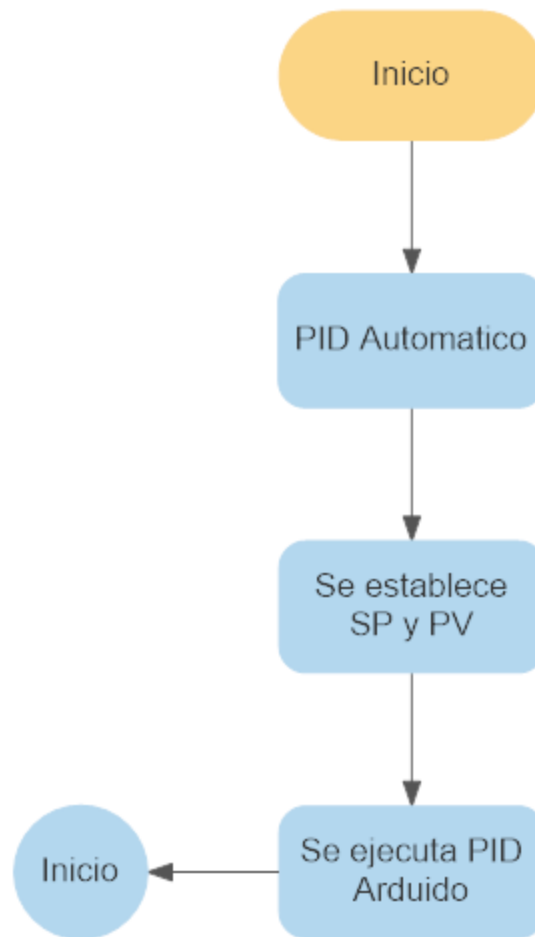
de los equipos electrónicos tiene este valor como umbral máximo, es por ello que se programó el algoritmo de la figura 34.

Figura 34 Algoritmo para desactivar la salida del Variac cuando la tensión supera 130V



Fuente: Autores.

Finalmente, la acción de control es ejecutada utilizando la librería PID de Arduino siendo la salida una señal PWM que modifica la velocidad del motor, adicional a eso se agrega la función que identifica si el motor debe girar a la izquierda o derecha como se puede observar en la figura 35.

Figura 35 Algoritmo PID para control de posición.

Fuente: Autores.

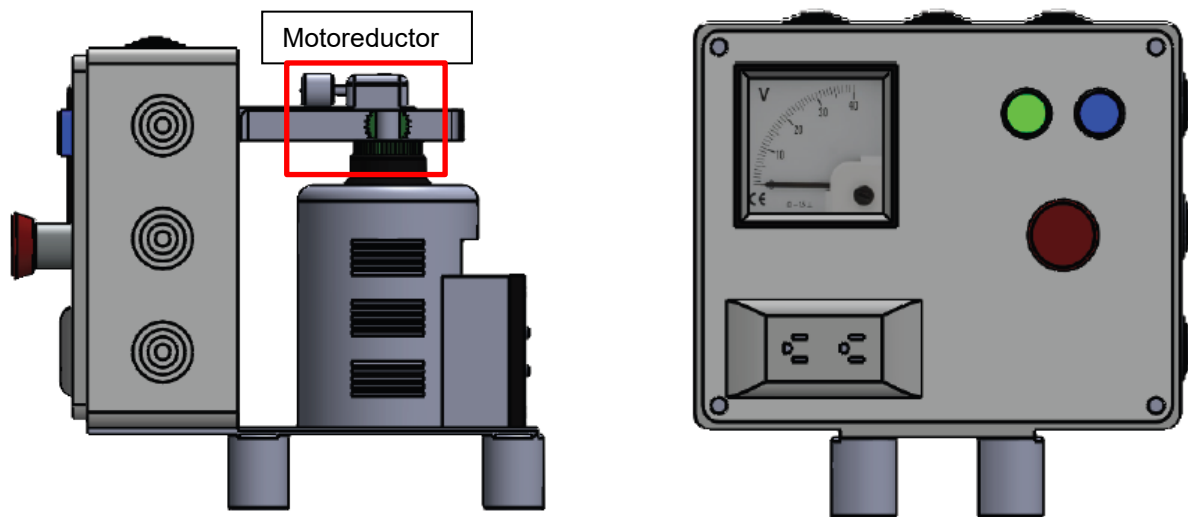
3.2. Distribución de componentes utilizando el software SolidWorks.

En este punto ya es claro cuáles son los elementos necesarios para realizar control sobre la tensión de salida del Variac, se procede a desarrollar cual será la ubicación de cada uno de ellos utilizando el software SolidWorks.

3.2.1. Esquemas desarrollados en SolidWorks.

Antes de iniciar la construcción del sistema de control y visualización de variables, es necesario definir la ubicación de los componentes, para ello se utilizó el software SolidWorks (ver figura 36), aquí de manera general se presenta la como sería la implementación final.

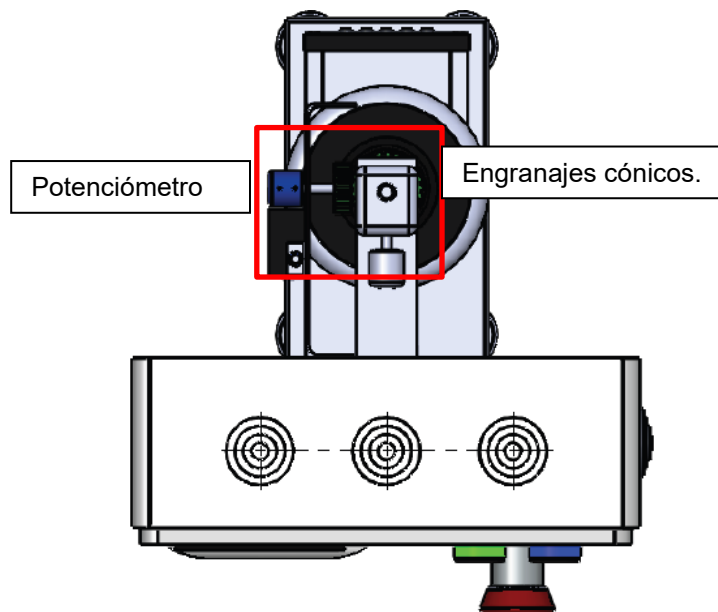
Figura 36 Esquema de implementación en SolidWorks.



Fuente: Autores.

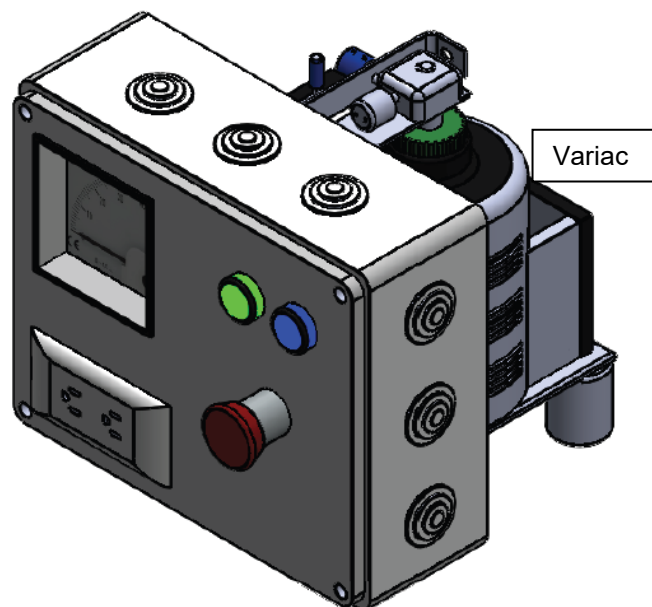
Después de acoplar el motor al Variac se utilizan dos engranajes cónicos para acoplar también el trimer que realimenta la posición del volante ver figura 37 la cual se ajusta mediante dos tornillos prisioneros. En la figura 38 se presenta una figura general de la implementación final del sistema a desarrollar.

Figura 37 Acople de eje del motor a volante del Variac.



Fuente: Autores.

Figura 38 Implementación final en SolidWorks.

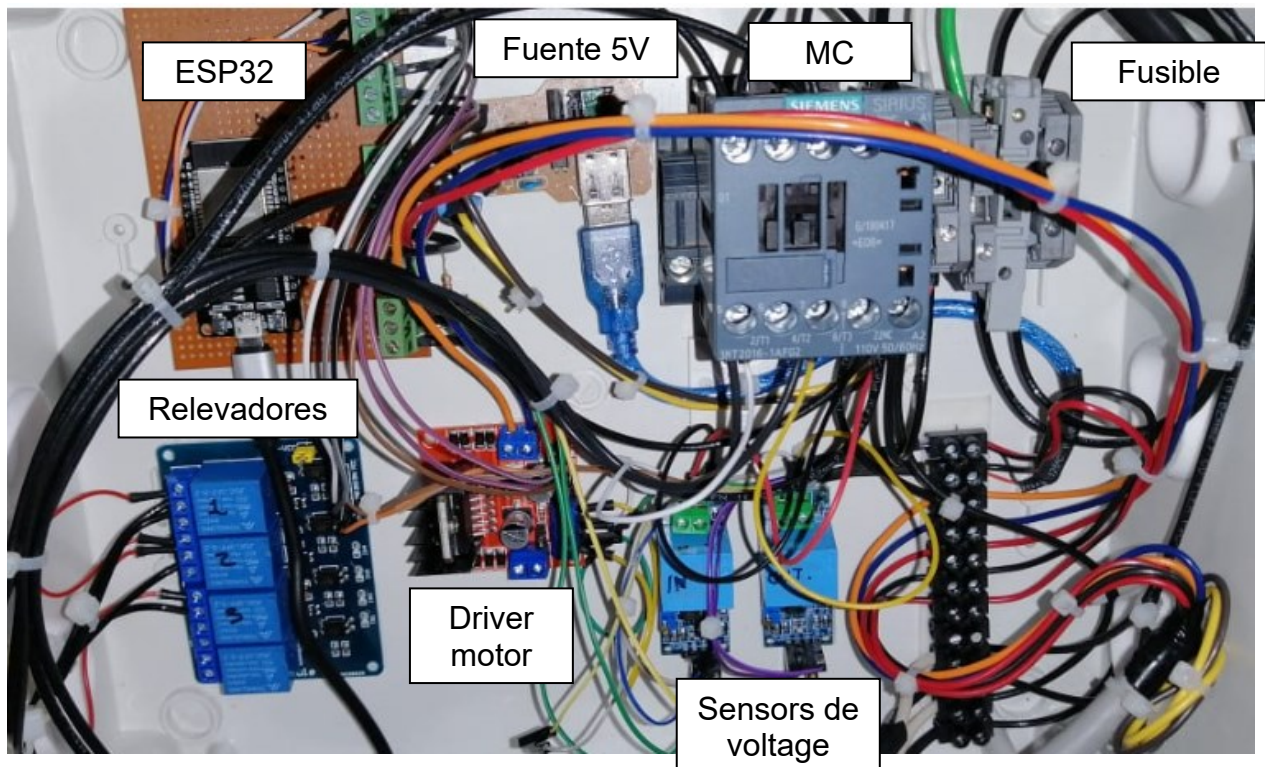


Fuente: Autores.

3.2.2. Distribución de los elementos dentro del tablero de control.

Como el sistema de adquisición de datos y control de voltaje de salida está gobernado por el microcontrolador ESP32 y adicional a ello se cuenta con otros elementos como fuente de DC, sensor de voltaje de entrada, driver L298 y módulo de relés, se utiliza el software SolidWorks para realizar la distribución entre cada uno de los componentes (ver figura 39).

Figura 39 Distribución de los elementos en la caja de control.



Fuente: Autores.

3.2.3. Implementación final.

Después de tener claro cuál será la distribución de los elementos que componen el control del Variac, se realiza la implementación final como lo presenta la figura 40.

Figura 40 Implementación final de Variac controlador por ESP32.

Tablero de control.



Variac acoplado a motorreductor. Con engranajes cónicos.



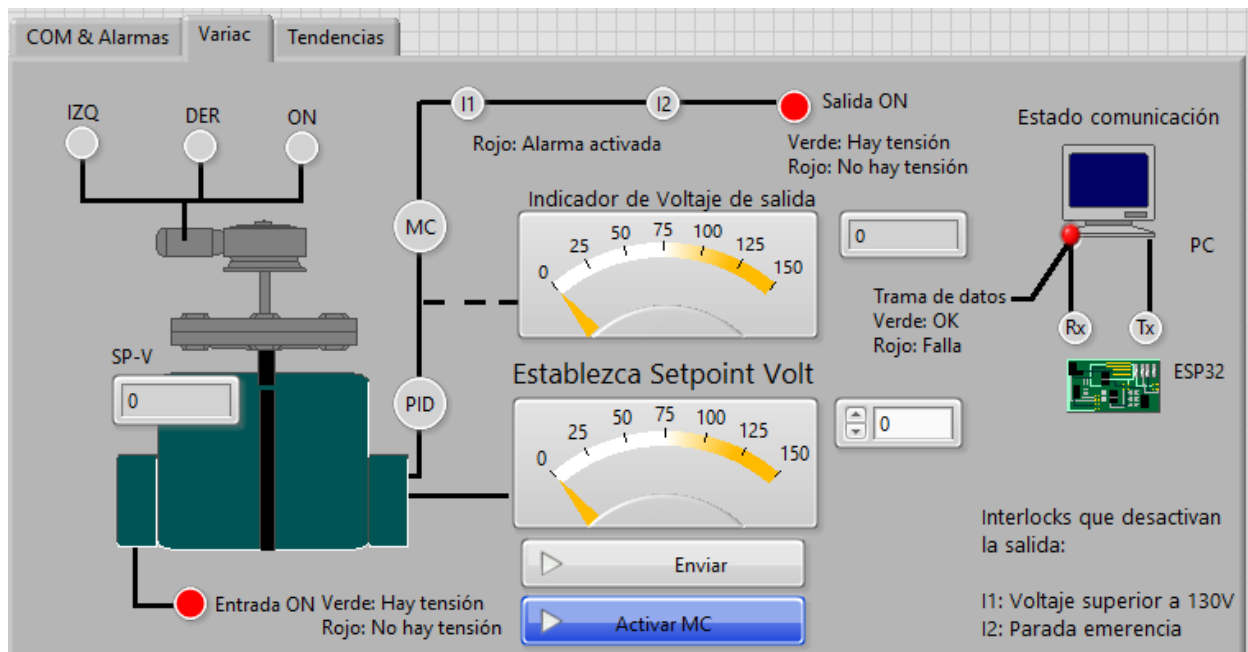
Fuente: Autores.

4. Interfaz Hombre Maquina para operar el Variac.

4.1. Panel frontal en Labview.

A través de la figura 41 se presenta la interfaz HMI que permite interactuar con el proceso y donde se presentan todas las señales que se han mencionado en el capítulo 2, de forma que sea fácil interpretar lo que está sucediendo en el Variac.

Figura 41 Interfaz hombre máquina para operar el Variac.



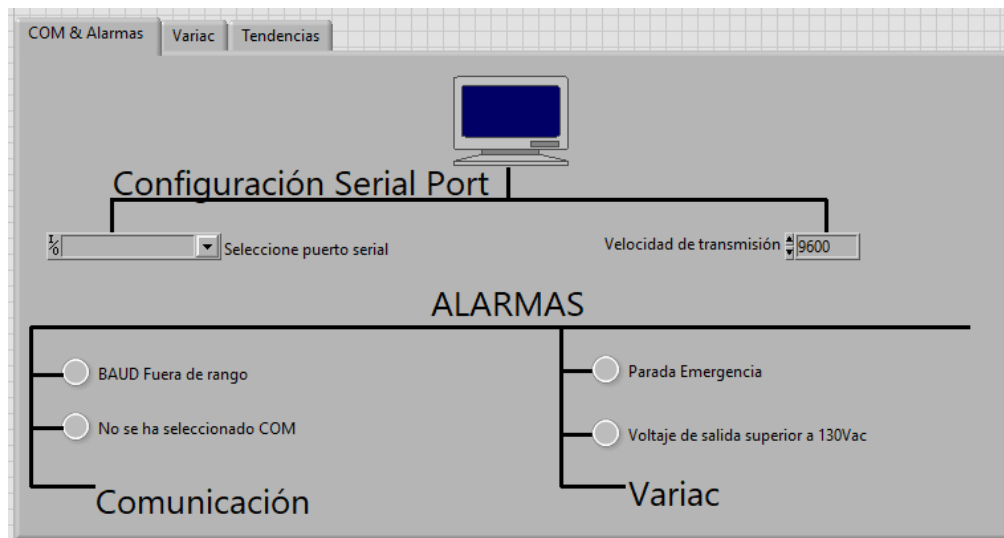
Fuente: Autores.

Adicional a esta ventana también se cuenta con dos más, la primera de ellas (ver figura 42) es utilizada para seleccionar el puerto de comunicación, la velocidad de transmisión que para el caso de este trabajo es de 115200 Baudios y adicional a ello presenta las siguientes alarmas:

- Velocidad de transmisión fuera de rango.

- No se ha seleccionado puerto de comunicación serial.
- Esta activada la parada de emergencia o voltaje de salida superior a 130Vac.

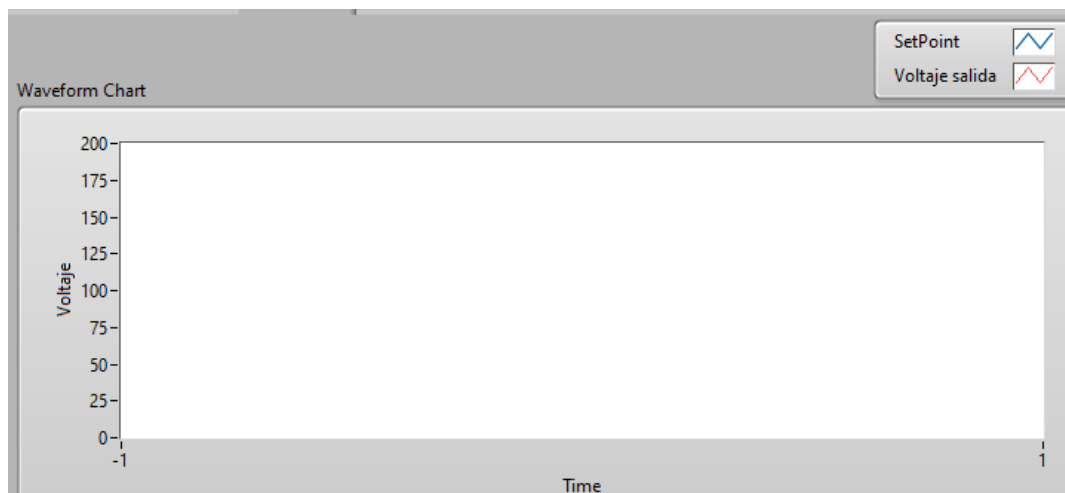
Figura 42 Configuración de puerto serial y alarmas en Labview.



Fuente: Autores.

La última ventana para interactuar desde el software Labview es la indicada en la figura 43, aquí el usuario evidencia las tendencias del valor de set point y el voltaje de salida.

Figura 43 Tendencia de Setpoint y voltaje de salida.

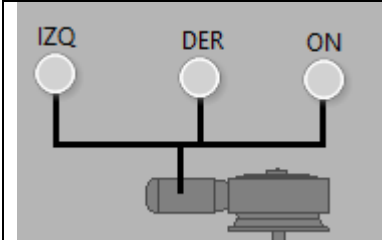


Fuente: Autores.

4.1.1. Elementos de interacción con el Variac desde el panel frontal.

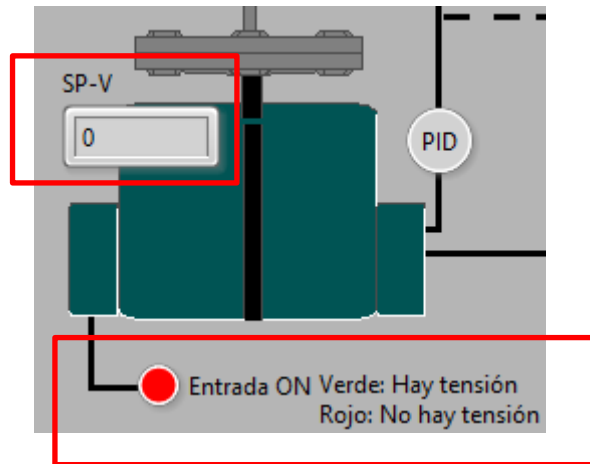
El motor de corriente continua que se encarga de mover el volante del Variac para modificar la tensión de salida y su operación es monitoreada desde el panel frontal de Labview, en la tabla 10 se indican algunos aspectos relevantes de esta señalización.

Tabla 10 Indicación de control del motorreductor.

	ON: Cuando se encuentra en color verde indica que el motor se encuentra activado
	IZQ: Cuando se encuentra en verde indica que el motor gira a la izquierda, disminuye la tensión de salida.
	DER: Cuando se encuentra en verde indica que el motor gira a la derecha, aumenta la tensión de salida.

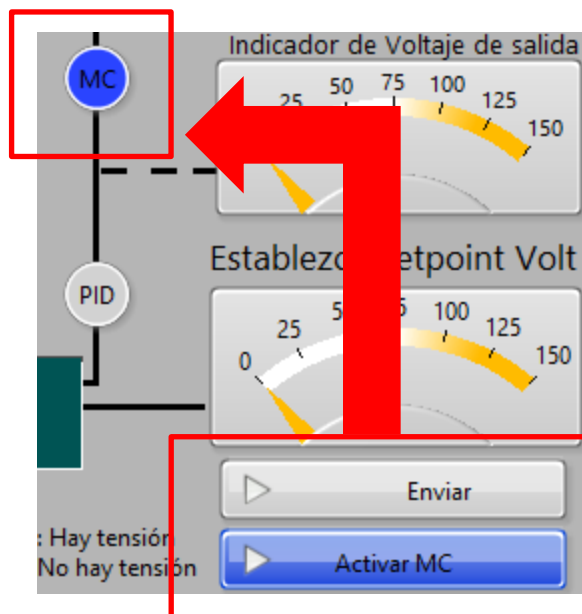
Fuente: Autores.

El indicador llamado SP-V se encarga de presentar cual es el valor de Setpoint sobre el cual la tarjeta de adquisición de datos basada en el ESP32 va a realizar el control, y adiciona a ello el indicador llamado “Entrada ON” que se observa en la figura 44 cuando se encuentra en color rojo indicara que no existe tensión de entrada, al estar en color verde indicara que el Variac se encuentra conectado a la red eléctrica y para ello se hace uso del sensor de voltaje de referencia ZMPT1010B el cual se conecta a una de las entradas análogas de la ESP32.

Figura 44 Indicador de Setpoint para control y visualizador de tensión de entrada.

Fuente: Autores.

Para activar la salida MC que habilita la tensión de salida es necesario inicialmente dar clic sobre el botón llamado “Activar MC” y después clic sobre el botón “Enviar” (ver figura 45) después de esto se activara el indicador MC en color azul.

Figura 45 Manipulación de MC desde Labview.

Fuente: Autores.

Si lo que se desea es habilitar el PID se deberá entonces dar clic sobre el botón “Activar PID” y después de esto clic sobre el botón “Enviar” retornando una señal del microcontrolador que hará que el indicador PID se active en color verde (ver figura 46)

Figura 46 Activación de PID desde Labview.



Fuente: Autores.

Adicional a esto, la misma señalización se indicará desde el tablero de control con la misma secuencia de colores para MC y PID como se puede evidenciar en la figura 47

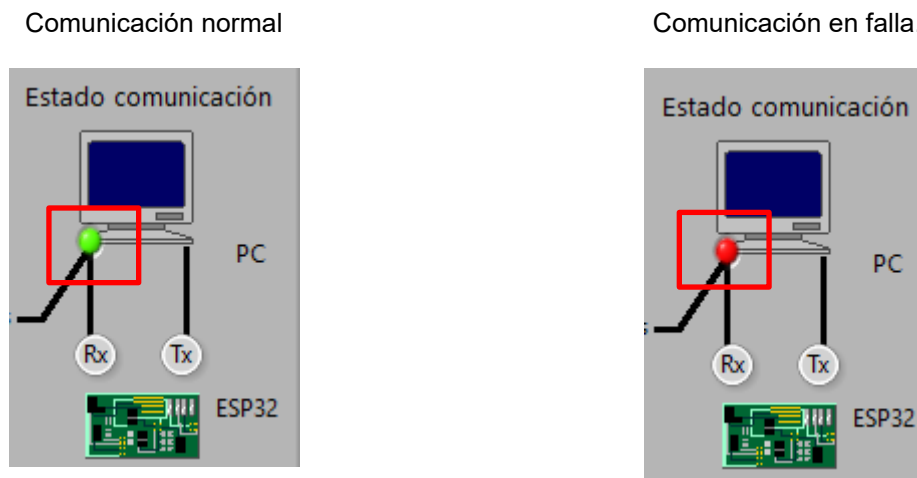
Figura 47 Indicación de MC y PID en el tablero de control.



Fuente: Autores.

Para comprobar que la comunicación entre el microcontrolador ESP32 y Labview se está realizando correctamente se incluye el indicador presentado en la figura 48, aquí cuando el LED se encuentra en color verde toda la información se está transmitiendo normalmente, sin embargo si este se encuentra en color rojo será necesario revisar el cableado y la configuración del ESP32.

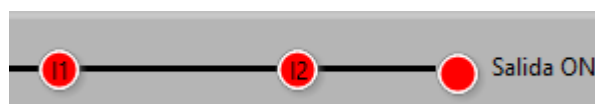
Figura 48 Comunicación normal y en falla.



Fuente: Autores.

La indicación de las dos alarmas que pueden desactivar la salida del Variac se visualizaran como se observa en la figura 49 cuando estén en color rojo indicaran que se encuentra presente la falla o condición.

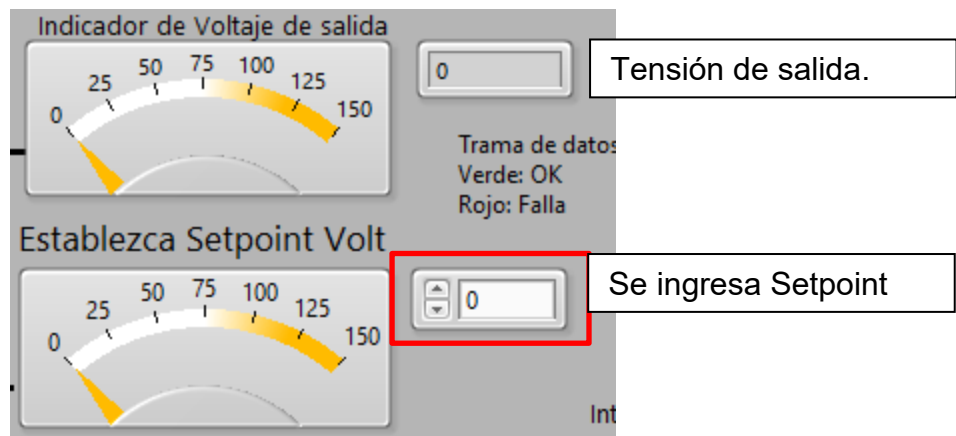
Figura 49 Presentación de fallas en panel frontal.



Fuente: Autores.

Finalmente, para seleccionar cual será el Setpoint que se desea establecer a la salida del Variac es necesario digitar dicho valor en la casilla indicada en la figura 50, y se visualizara a través del Knob meter.

Figura 50 Establecer Setpoint desde panel frontal.



Fuente: Autores.

4.2. Diagrama de bloques.

Antes de mostrar el diagrama de bloques, se es necesario dejar claro cuál es la forma con la cual se van a comunicar Labview con el ESP32, es por esta razón que inicialmente se presentaran los algoritmos y la lógica bajo la cual opera la programación.

4.2.1. Algoritmo programado en Labview.

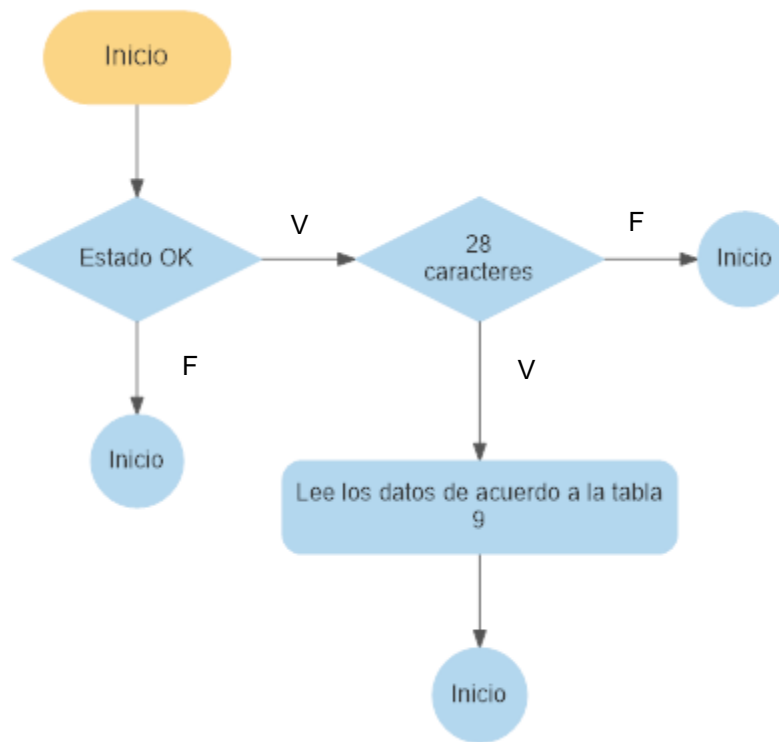
Para realizar la comunicación con el microcontrolador ESP32, se realiza un algoritmo que cumple las funciones indicadas en la tabla 11.

Tabla 11 Funciones del algoritmo desarrollado en Labview.

Función	Descripción
Lectura de buffer serial	Realiza la lectura de una cadena de caracteres en los cuales viene la información relacionada con: Estado de PID y MC, Lectura del Setpoint, voltaje de salida, motor ON, giro izquierda, giro derecha, Alarmas I1 e I2, voltaje de entrada y salida.
Escritura en buffer serial.	Envía los datos necesarios para activar MC, el PID y modificar el Setpoint.

Fuente: Autores.

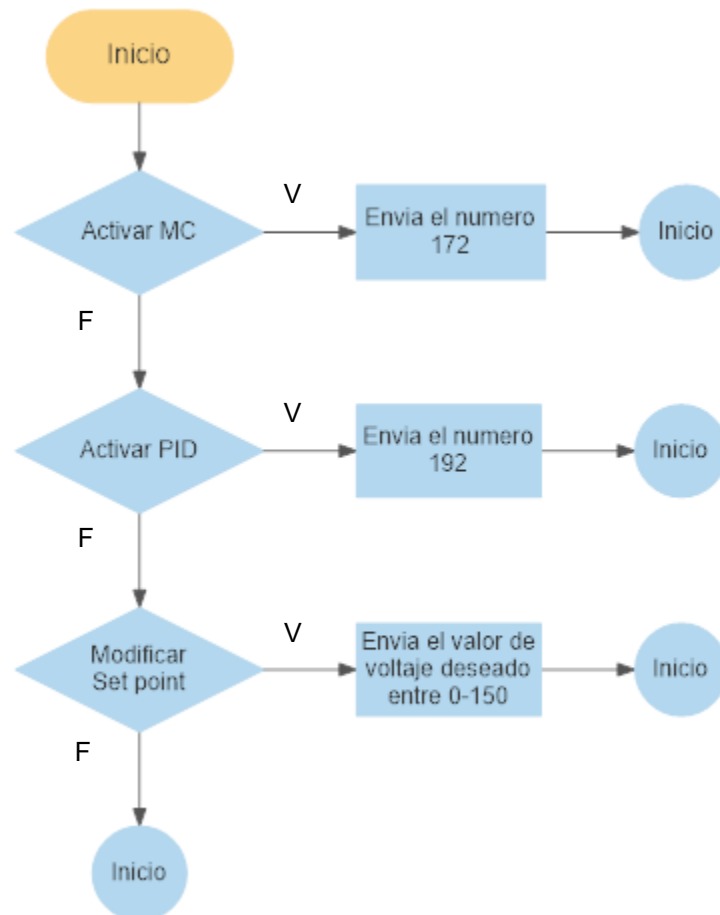
Para realizar la lectura de buffer serial se desarrolla el algoritmo planteado en la figura 51, aquí lo importante es primero identificar que la trama de datos es de 28 caracteres e inicia con la palabra “Estado OK” si están dadas estas dos condiciones el algoritmo pasa a una nueva etapa en la cual se leen los datos de cada variable.

Figura 51 Algoritmo de lectura de buffer serial.

Fuente: Autores.

En el caso de la escritura en el buffer serial, si se desea enviar el Setpoint se enviara un numero entre 0 y 150, para activar el PID se envía el número 172 y para el MC se envía 192, esto se puede observar en el algoritmo de la figura 52.

Figura 52 Algoritmo para escritura de buffer serial.



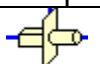



Fuente: Autores.

4.2.2. Programación en lenguaje gráfico.

Labview permite desarrollar un lenguaje de programación gráfico basado en estructuras y bloques. Como se logró evidenciar en el ítem 4.2.1. las funciones desarrolladas son

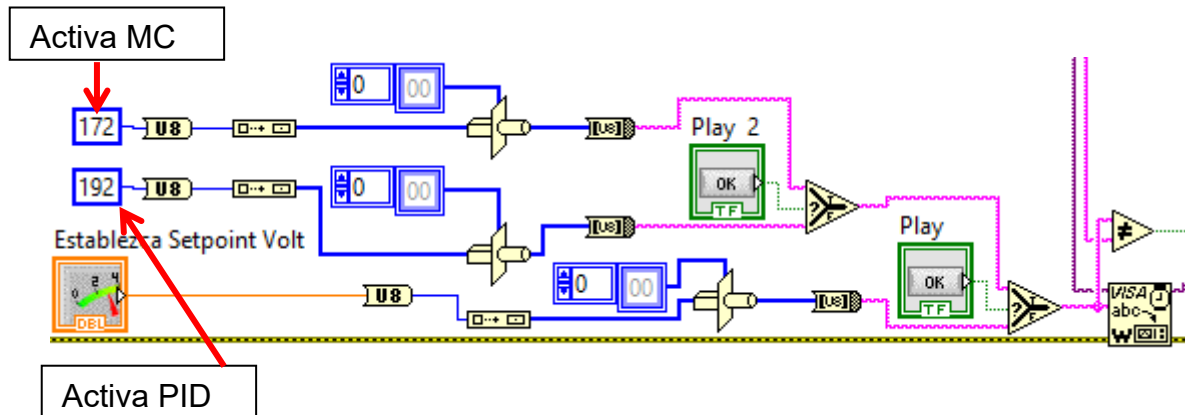
realizar escritura y lectura del buffer serial, por esta razón aquí se mostrará cómo se realizan estas acciones, pero cabe señalar que todo el código se ejecuta dentro de una estructura While-loop. Para realizar la escritura del puerto serial se implementaron los bloques indicados en la figura 53 en ellos se destacan los descritos en la tabla 12.

Tabla 12 Bloques utilizados para escritura del buffer serial.

Bloque	Función.
	Type Cast Function; convierte el numero ingresado en un matriz.
	Byte Array To String Function; Convierte una matriz de bytes sin signo que representan caracteres ASCII en una cadena
	Select; Devuelve el valor conectado en T o F según e estado de “s”
	Visa Write; a raves de este bloque se envía la información por el puerto serial.

Fuente: Autores.

Figura 53 Programa para escribir en buffer serial

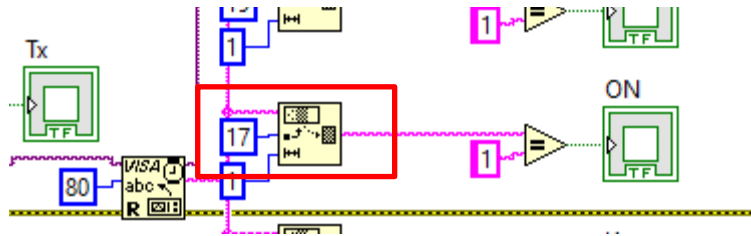


Fuente: Autores.

Para el caso de la lectura del buffer serial se utiliza el bloque llamado “Serial Buffer”, con este le obtiene la información enviada por el microcontrolador ESP32, aquí el bloque de mayor importancia es el indicado en la figura 54, este se encarga de leer

solo “tramos” de la cadena de información enviada utilizando como guía la tabla 9 de este documento.

Figura 54 Programa para leer buffer serial.



Fuente: Autores.

5. Validación del sistema de control implementado en el Variac.

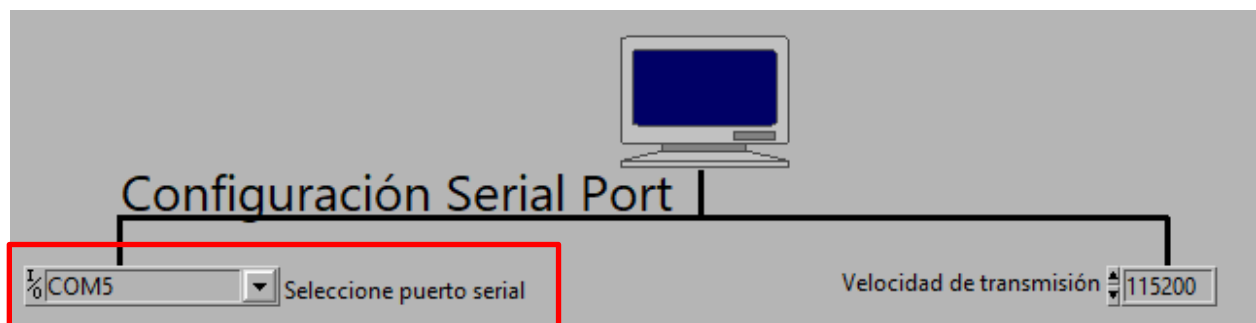
Este capítulo presenta las pruebas funcionales realizadas al controlador desarrollado, para dejar en evidencia que este funciona correctamente.

5.1. Prueba de control de tensión.

Para realizar la operación correcta del software y Hardware implementado se deben realizar los pasos mencionados a continuación:

- Conectar el tablero de control a una toma eléctrica de 110Vac
- Conectar el cable USB a uno de los puertos USB del computador.
- En el administrador de dispositivos verificar cual fue el COM asignado al cable USB.
- Ingresar al aplicativo desarrollado en Labview y asignar el puerto que se encontró en el administrador de dispositivos como lo indica la figura 55, durante la prueba el COM asignado fue el numero 5 pero este puede cambiar.

Figura 55 Seleccionando puerto de comunicación desde Labview.



Fuente: Autores.

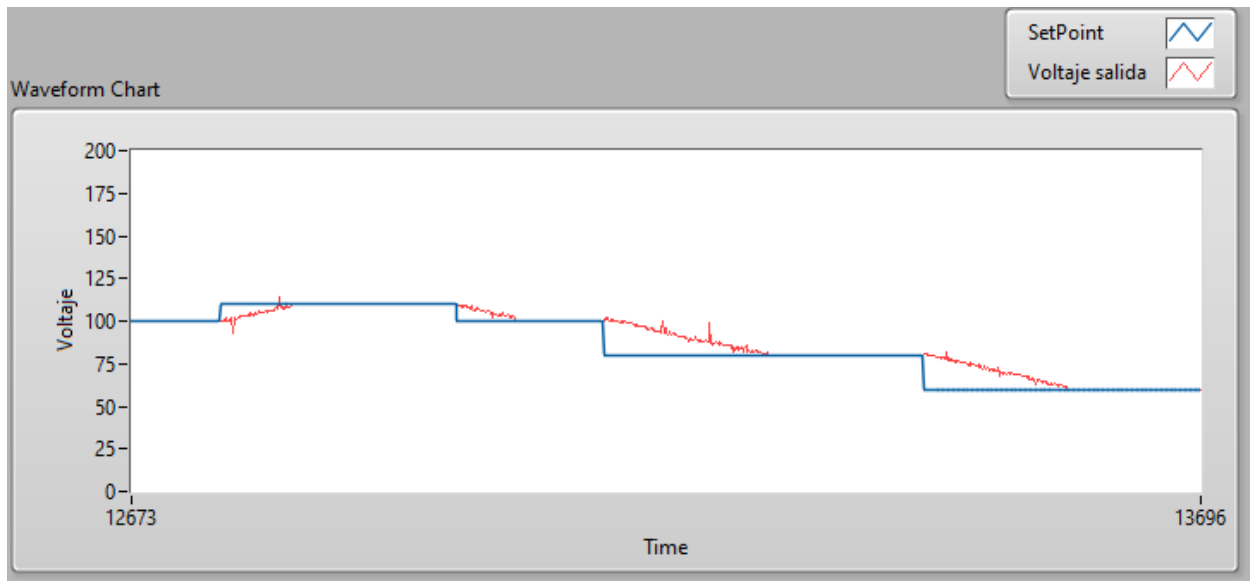
Las pruebas desarrolladas no tienen la intención de determinar el error de las mediciones porque esto se evaluará en el capítulo 6, aquí se presentará que el equipo es capaz de controlar la salida de tensión desde Labview. A continuación, se indica el paso a paso para modificar el voltaje de salida.

- Se activa la salida **MC** desde el panel frontal.
- Se activa la salida **PID** desde el panel frontal.
- Se establece el Setpoint en los puntos indicados en la tabla 13, aquí es posible identificar que tanto en ascenso como en descenso el control es capaz de ubicar el volante del Variac en la posición que corresponde al voltaje deseado, adicional se deja la figura 56 que indica el aumento de la tensión de salida para llegar de 50 a 90Vac.

Tabla 13 Resultados de pruebas de control de voltaje en ascenso y descenso.

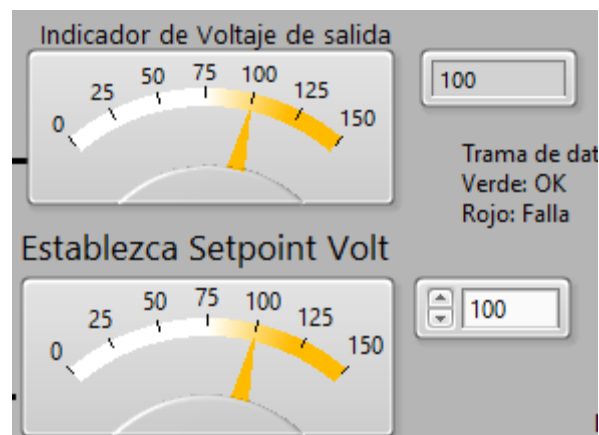
Asenso		Descenso	
Setpoint [V]	Voltaje de salida [V]	Setpoint [V]	Voltaje de salida [V]
40	40	110	110
50	50	100	100
60	60	90	90
90	90	60	60
100	100	50	50
110	110	40	40

Fuente: Autores.

Figura 56 Resultados de pruebas en ascenso y descenso de voltaje.

Fuente: Autores.

Además, en Labview es posible registrar el voltaje de salida del Variac desde el panel frontal de Labview, en la figura 57 se indica que se ha establecido un valor de Setpoint de 100 Vac y el microcontrolador ESP32 está registrando que alcanzo dicho valor.

Figura 57 Visualización en Labview de pruebas de control de voltaje.

Fuente: Autores.

6. Resultados y análisis.

Para realizar el análisis de los resultados se llevarán a cabo una serie de pruebas que permiten determinar que el sistema implementado funciona sin inconvenientes.

6.1. Pruebas de comunicación.

La prueba de comunicación presenta a detalla el tránsito de los datos desde el ESP32 hasta el algoritmo desarrollado en Labview que permite visualizar la información del Variac en el panel frontal. Para empezar en la figura 58 se muestra cómo se presenta como vienen organizados estos datos que corresponde a lo indicado en la tabla 9 de este documento y para ello se utilizó el monitor serial del IDE de Arduino.

Figura 58 Información en IDE de Arduino.

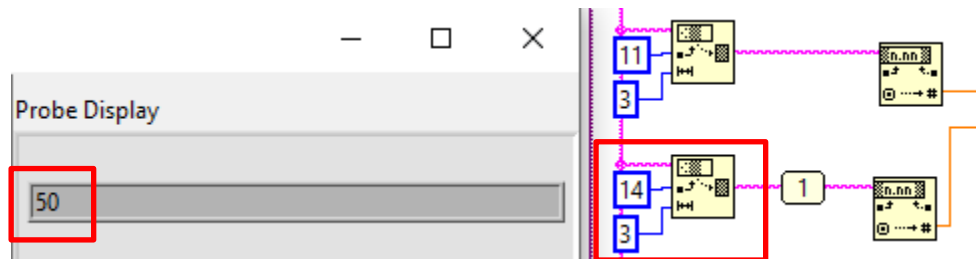
```
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
Estado OK11060060000000721
```

Fuente: Autores.

Después que esta información llega a Labview y es leída mediante el bloque VISA READ, ya solo queda leer los tramos de información que se desean visualizar en el

panel frontal, por eso en la figura 59 se observa que después del bloque “String Subjet” si se desea leer el voltaje de salida se inicia la lectura en el carácter 14 hasta el 16, de esta forma es claro que el cien por ciento de la comunicación entre el hardware y software es funcional.

Figura 59 Lectura de voltaje después de String Subjet.



Fuente: Autores.

6.2. Corrección del error de la medida

Para realizar esta prueba, es necesario saber que el error es calculado utilizando la ecuación 7, tomando como medida de referencia la de un multímetro que se utiliza para medir la salida de tensión del Variac y comparándola con la tensión indicada en el panel frontal de Labview, los resultados se muestran en la tabla 14.

Ecuación 7 Calculo de error porcentual.

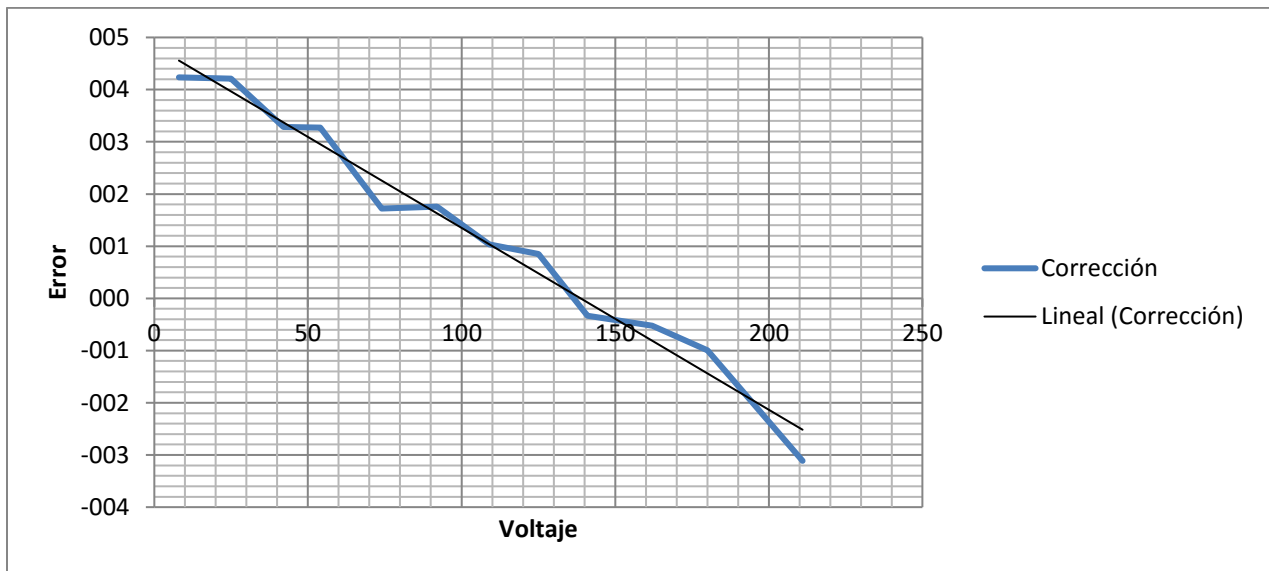
$$Error = Referencia - Labview$$

Para obtener la relación que se produce entre el voltaje de salida con respecto al error de la medición es necesario inicialmente realizar la gráfica de la figura 60, después de esto se obtiene una línea de tendencia del error representada a través de la ecuación 8.

Tabla 14 Error de medidas de voltaje

Multímetro	Labview	Error Lineal
22,8	18,56	4,24
33,7	29,49	4,21
43,7	40,41	3,29
51,4	48,12	3,28
62,7	60,98	1,72
74,3	72,54	1,76
84,5	83,47	1,03
94,6	93,75	0,85
103,7	104,03	-0,33
117	117,53	-0,53
128,1	129,09	-0,99
145,9	149,01	-3,11

Fuente: Autores.

Figura 60 Tendencia del error de voltaje.

Fuente: Autores.

Ecuación 8 Tendencia del error de voltaje

$$Error = -(0.03048 * voltaje Labview) + 4.8357$$

La ecuación obtenida del error ahora es agregada al microcontrolador ESP32 y con ello se logra reducir el error máximo de 4.24 V a -0.32V como se observa en la tabla 15, de esta forma es claro que el sistema de control y visualización desarrollado cumple su función correctamente.

Tabla 15 Corrección de la tensión de salida.

Multímetro	Error inicial.	Tensión corregida	Error final
22,8	4,56	23,12	-0,32
33,7	3,97	33,45	0,25
43,7	3,37	43,79	-0,09
51,4	2,96	51,08	0,32
62,7	2,26	63,24	-0,54
74,3	1,63	74,18	0,12
84,5	1,04	84,51	-0,01
94,6	0,49	94,23	0,37
103,7	-0,07	103,96	-0,26
117	-0,80	116,72	0,28
128,1	-1,43	127,66	0,44
145,9	-2,51	146,51	-0,61

Fuente: Autores.

7. Conclusiones.

El driver L298 fue seleccionado utilizando matrices de selección ya que por sus características como corriente pico de operación 4 A, corriente constante de operación 2 A, corte de operación por sobrecalentamiento, excelente inmunidad al ruido, permite voltaje de alimentación de hasta 46V, además permite realizar el control de la velocidad y sentido de giro mediante señales digitales que se logran conectar directamente de los pines del ESP32.

Para lograr determinar la ecuación de transferencia junto con las constantes PID se utilizó el IDENT de Matlab comparando las salidas al 15%, 37% y 80%, esto dio como resultado constantes $P= 1.32$, $I=0.001$ y $D=0.001$; además de esto el Rise time fue igual a 2.2seg, el Overshoot de 0; y setting time de 3.91seg, garantizando una operación correcta.

A través de Labview es posible crear una interfaz hombre máquina, desde el diagrama de bloques se conectan los indicadores del panel frontal y para realizar la comunicación con el ESP32 los boques Visa permiten establecer la comunicación entre el hardware y software utilizando el puerto serial cuya de transmisión de 115200 Baudios, con la función "String Subject" se logran leer sectores de la trama de comunicación en donde se información como Setpoint, Voltaje de salida, alarmas, operación del motorreductor, sentido de giro.

Al realizar la validación del control desarrollado es posible identificar que tanto el control de posición como la visualización se realizan correctamente, ya que las pruebas iniciales arrojaron un error en la medición de 4.56V, sin embargo, después de encontrar

la línea de tendencia y ecuación que representa el error para agregarlo a la ecuación programada desde el IDE para el ESP32 se logró reducir dicho error a $-0.32V$

8. Bibliografía.

- [1]. Jesús Fraile Mora, Maquinas eléctricas sexta edición, editorial Mc Graw Hill, p. 181.
- [2]. Enrique Ras Oliva, Transformadores de potencia, medida y protección, editorial MACOMBO ediciones técnicas, p. 5, p. 175-200.
- [3]. Peter Bastian, electrotecnia, Ediciones AKAL, p. 276.
- [4]. Adolf Senner, Principios de electrotecnia, editorial REVERTE, p. 186.
- [5]. Katsuhiko Ogata, Ingenieria de control moderna, Prentice Hall, p. 7.
- [6]. Lajara José, LABVIEW entorno grafico de programación, editorial Marcombo ediciones técnicas, p. 50.
- [7]. Robert L. Mott, Diseño de elementos de máquinas, editorial PEARSON, p. 266.
- [8]. Antonio Pérez Gonzales, Mantenimiento mecánico de máquinas, UNIVERSITAT JAUME, p. 55.
- [9]. Pichucho, J. (2007). Modulo para verificar el funcionamiento de los programas grabados en el PIC16F84A.
- [10]. Curso básico de microcontroladores CEKIT, p. 9. 1997.