



Secretaría de Educación Pública

Tecnológico Nacional de México Instituto Tecnológico de Veracruz

Control Inteligente implementado en un sistema embebido, de la temperatura de un horno empleado en la soldadura de componentes de montaje superficial

TESIS

Que para obtener el grado de:

Maestro en Eficiencia Energética y Energías Renovables

Presenta:

José Luis López Riviello

Director:

MC. José Antonio Hernández Reyes

Co-Director:

MC. Carlos Roberto González Escarpeta



"2019, Año del Caudillo del Sur, Emiliano Zapata"

H. Veracruz, Veracruz, **17/diciembre/2019**
Div. Estudios de Posgrado e Investigación
OFICIO: ITV/DEPI/No.605/2019

Asunto: Autorización de Impresión de Tesis

**C. JOSÉ LUIS LÓPEZ RIVIELLO
PRESENTE**

Por este conducto se le comunica que su tesis de Maestría en Eficiencia Energética y Energías Renovables titulada:

"Control inteligente implementado en un sistema embebido, de la temperatura de un horno empleado en la soldadura de componentes de montaje superficial"

Fue aprobada en su totalidad en cuanto a formato y calidad del contenido a satisfacción del H. Jurado del examen, por lo que está Usted autorizado a editar la presentación definitiva del trabajo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
Antorcha y Luz de Fuego Permanente®

EL. H. JURADO


BENIGNO ORTIZ MUÑIZ
JEFE DE LA DIV. EST. POSG. E INVEST

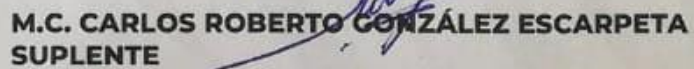

M.C. JOSÉ ANTONIO HERNÁNDEZ REYES
PRESIDENTE




M.C. JOSÉ LUIS FERNANDO PALOMEQUE LOYO
SECRETARIO

INSTITUTO TECNOLÓGICO DE VERACRUZ
DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN


DR. RICARDO FRANCISCO MARTÍNEZ GONZÁLEZ
VOCAL


M.C. CARLOS ROBERTO GONZÁLEZ ESCARPETA
SUPLENTE

c. p. Expediente del Estudiante
Archivo

BOM*





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO


Instituto Tecnológico de Veracruz

"2019, Año del Caudillo del Sur, Emiliano Zapata"

El presente trabajo de tesis titulado **"Control inteligente implementado en un sistema embebido, de la temperatura de un horno empleado en la soldadura de componentes de montaje superficial"** presentado por el (la) C. **José Luis López Riviello** ha sido aprobado por su comité tutorial como requisito parcial para obtener el grado de:

MAESTRO EN EFICIENCIA ENERGÉTICA Y ENERGÍAS RENOVABLES

DIRECTOR


M.C. JOSÉ ANTONIO HERNÁNDEZ REYES

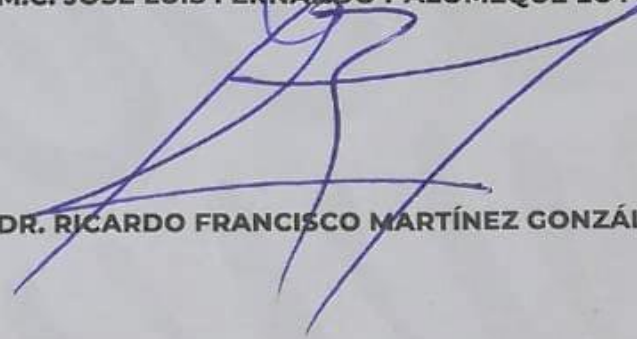
CO-DIRECTOR


M.C. CARLOS ROBERTO GONZÁLEZ ESCARPETA

REVISOR


M.C. JOSÉ LUIS FERNANDO PALOMEQUE LOYO

REVISOR


DR. RICARDO FRANCISCO MARTÍNEZ GONZÁLEZ



Calz. Miguel Ángel de Quevedo 2779, Col. Formando Hogar
C.P. 91897, Veracruz, Ver. Tel. (229) 934 1500
www.veracruz.tecnm.mx



**CONTROL INTELIGENTE IMPLEMENTADO EN UN SISTEMA EMBEBIDO, DE LA
TEMPERATURA DE UN HORNO EN LA SOLDADURA DE COMPONENTES DE
MONTAJE SUPERFICIAL.**

Por:

José Luis López Riviello

Tesis propuesta al

Consejo de Maestría en Eficiencia Energética y Energías Renovables

del

INSTITUTO TECNOLÓGICO DE VERACRUZ

Como requerimiento parcial para obtener el grado de

Maestro en eficiencia energética y energías renovables

Diciembre 2019

AGRADECIMIENTOS

Quiero agradecer a Dios por darme salud, vida y amor.

A mi esposa Jessi por darme el amor y apoyo durante todo este trayecto, inspirarme día a día, comparto contigo este gran logro amor.

A mis padres por que siempre están confiando en mí y están apoyándome durante todo este trayecto, y que me han inculado valores y principios con su ejemplo, por haberme dado todo su amor y confianza sin descanso durante toda mi vida, con ellos comparto la euforia de la conclusión de este magnifico proyecto.

A mi hermano que me apoya a seguir e inspira para poder guiarnos hacia un buen camino, teniendo retos que nos gusten mejorar.

A mis asesores MC. José Antonio y MC. Carlos Roberto por compartir su experiencia, dedicar el tiempo necesario revisado cuidadosamente y poniendo mejoras en mi trabajo.

A mis profesores que me formaron como maestro, con sus grandes enseñanzas y me inculcaron a ser siempre metódico en el mundo de la investigación.

Al Consejo Nacional de Ciencia y Tecnología por el apoyo financiero para realizar mis estudios de maestría.

Al Instituto Tecnológico de Veracruz, por enseñarme los conocimientos los cuales fui adquiriendo en el transcurso de esta meta personal para lograr un desempeño óptimo en la inserción laboral.

Gracias.

RESUMEN

En este documento se presenta el desarrollo e implementación del control de temperatura de un horno para efectuar la soldadura de componentes de montaje superficial. Se evalúa e implementa el programa de control en tres sistemas embebidos seleccionados previamente por sus características de adaptabilidad en el aprendizaje máquina. Por las características individuales de cada uno de los sistemas embebidos propuestos, se estructuraron programas para obtener el máximo desempeño en cada uno de ellos, para elegir el sistema embebido más adecuado y que procese los datos de manera más eficiente.

La implementación inicia con la evaluación de las características y la arquitectura de los sistemas embebidos para la aplicación del control inteligente. La estrategia de control usada es la lógica difusa, desarrollada a través del método de Mamdani y se utiliza la curva de temperatura para tecnología de montaje superficial a seguir, citada en el trabajo de L. Kwang. El uso de esta disciplina antes mencionada se introduce en la aplicación de una inteligencia artificial básica, puesto que, a través de estos procesos, el dispositivo a desarrollar toma las decisiones pertinentes para mantener el control de la temperatura. La inteligencia artificial basada en la lógica difusa acepta múltiples variables, semejando el discernimiento humano, a diferencia de la lógica tradicional o binaria que solo acepta dos valores.

Al alcanzar la semejanza de un pensamiento humano en un dispositivo electrónico, no es necesario un operador experto para llevar a cabo procesos productivos como el que nos ocupa. También fue importante hacer la mejor elección del sistema electrónico embebido con mayor y más eficiente procesamiento de los datos obtenidos en el proceso de soldadura, el cual es nuestro punto de partida.

Después de comparar los resultados obtenidos por cada uno de ellos, se selecciona la Raspberry Pi 3 B para realizar el controlador inteligente.

Para mostrar los resultados físicos, se presentan pruebas de seguimiento de temperatura y de acabado de soldadura.

Los capítulos que componen este documento mostrarán los detalles de la investigación y sus resultados.

El capítulo 1 describe la presentación y descripción del tema partiendo del estado actual, así como los antecedentes, la justificación y los objetivos de este documento.

El capítulo 2 presenta una revisión bibliográfica de los temas de la propuesta, siendo la temperatura, la soldadura, la curva de temperatura en soldadura de montaje superficial, horno de convección, los sistemas embebidos, sensor de temperatura, y los programas necesarios para la aplicación del controlador inteligente.

El capítulo 3 presenta la propuesta metodológica del controlador inteligente, desde la obtención de los datos, la optimización, la selección de los sistemas embebidos para el controlador inteligente, el comportamiento y la evaluación de los sistemas embebidos, la implementación de los programas y librerías del controlador inteligente, la parametrización de las variables del control inteligente y la programación del controlador inteligente.

El capítulo 4 presenta los resultados de la optimización de los datos mediante una red neuronal artificial, los resultados de la aplicación del controlador en el horno eléctrico, los acabados de soldadura y la discusión del controlador inteligente.

El capítulo 5 se presenta la conclusión de este trabajo.

El capítulo 6 presenta la Bibliografía y los anexos de esta tesis.

Palabras clave: *Sistemas Embebidos, Inteligencia Artificial, Lógica Difusa, Red Neuronal Artificial, Control Difuso, Arduino, Tiva C, Raspberry Pi.*

ABSTRACT

This document presents the development and implementation of the temperature control of a furnace for welding surface mount components. The control program is evaluated and implemented in three embedded systems previously selected for their adaptability characteristics in machine learning. Due to the individual characteristics of each of the proposed embedded systems, programs were structured to obtain maximum performance in each of them, to choose the most appropriate embedded system and to process the data more efficient.

The implementation begins with the evaluation of the characteristics and architecture of the embedded systems for the application of intelligent control. The control strategy used is fuzzy logic, developed through the Mamdani method and the temperature curve is used for surface mounting technology to follow, cited in the work of L. Kwang. The use of this discipline mentioned above is introduced in the application of a basic artificial intelligence, since, through these processes, the device to be developed makes the relevant decisions to maintain temperature control. Artificial intelligence based on fuzzy logic accepts multiple variables, resembling human discernment, unlike traditional or binary logic that only accepts two values.

When reaching the similarity of a human thought in an electronic device, an expert operator is not necessary to carry out productive processes like the one in question. It was also important to make the best choice of the embedded electronic system with greater and more efficient processing of the data obtained in the welding process, which is our starting point.

After comparing the results obtained by each of them, the Raspberry Pi 3 B is selected to perform the smart controller.

To show the physical results, temperature monitoring and welding finishing tests are presented.

The chapters that make up this document will show the details of the investigation and its results.

Chapter 1 describes the presentation and description of the topic based on the current status, as well as the background, justification and objectives of this document.

Chapter 2 presents a bibliographic review of the topics of the proposal, being the temperature, the welding, the temperature curve in surface mount welding, convection oven, embedded systems, temperature sensor, and the programs necessary for the application of the smart controller.

Chapter 3 presents the methodological proposal of the intelligent controller, from data collection, optimization, the selection of embedded systems for the intelligent controller, the behavior and evaluation of embedded systems, the implementation of the programs and libraries of the intelligent controller, parameterization of the variables of the intelligent control and programming of the intelligent controller.

Chapter 4 presents the results of data optimization through an artificial neural network, the results of the application of the controller in the electric furnace, welding finishes and the discussion of the intelligent controller.

Chapter 5 presents the conclusion of this work.

Chapter 6 presents the Bibliography and the annexes of this thesis

Keywords: *Temperature, Embedded Systems, Artificial Intelligence, Fuzzy Control, Arduino, Tiva C, Raspberry Pi.*

ÍNDICE

AGRADECIMIENTOS	i
RECONOCIMIENTOS.....	ii
RESUMEN.....	iii
ABSTRACT.....	v
ÍNDICE	vii
LISTA DE TABLAS.....	ix
LISTA DE FIGURAS	ix
ABREVIATURAS	xi
INTRODUCCIÓN	1
I. CAPITULO 1.....	5
1.1. Descripción del problema.....	5
1.2. Preguntas de investigación	5
1.3. Justificación.....	6
1.4. Alcance	8
1.5. Hipótesis	8
1.6. Objetivos.....	9
1.6.1 Objetivo general	9
1.6.2 Objetivos específicos	9
II. CAPITULO 2	10
2.1. Estado del arte	12
2.2. Marco Teórico	12
2.2.1. Temperatura.....	12
2.2.1.1 Medios de transferencia de calor	12
2.2.2. Soldadura	13
2.2.3. Aleaciones de soldadura	13
2.2.4. Pasta de soldar	14
2.2.5. Curva de temperatura en soldadura de montaje superficial	15
2.2.6. Fundente (Flux)	16
2.2.7. Metodología de la soldadura.....	16
2.2.7.1 Efectos en el perfil de recalentado.....	19
2.2.8. Horno de Convección.....	19

2.2.9. Sistemas embebidos.....	20
2.2.9.1. Consideraciones de los sistemas embebidos.....	20
2.2.9.2 Elección de sistemas embebidos	21
2.2.9.2.1.Sistema embebido Arduino	22
2.2.9.2.2 Sistema embebido Tiva C	22
2.2.9.2.3 Sistema embebido Raspberry	23
2.2.9.2.3.1 Raspbian.....	24
2.2.9.2.3.2 Python	25
2.2.9.2.3.3 Librería Scikit-learn	26
2.2.9.2.3.4 Librería Scikit-Fuzzy.....	26
2.2.9.2.3.5 Modulación de ancho de pulso.....	26
2.2.10. Sensor MAX31855.....	27
2.2.11. Inteligencia Artificial	28
2.2.12. Lógica Difusa	28
2.2.12.1. Conjunto difuso.....	30
2.2.12.2. Soporte, Punto cruce, Fuzzy Singleton	30
2.2.12.3. Colocación de operaciones teoreticas.....	31
2.2.12.4. Máquina de inferencia difusa	32
2.2.12.5. Fusificación	32
2.2.12.6. Inferencia difusa.....	33
2.2.12.7. Reglas de conocimiento difuso	33
2.2.12.8. Maquina de inferencia Mamdani.....	33
2.2.12.9. Maquina de inferencia Sugeno	34
2.2.12.10. Defusificación	34
2.2.12.11. Método del Centroides.....	35
2.2.12.12. Método de Centro de Máximos.....	35
2.2.13. Controlador Lógico Difuso	35
2.2.14. Redes Neuronales Artificiales	36
2.2.14.1. Aplicaciones de las Redes Neuronales Artificiales(RNA)	37
2.3.14.2. Entrenamiento de las redes neuronales.....	38
2.3.14.3. Redes de una capa.....	39
2.3.14.4. Perceptrón.....	40
III. CAPITULO 3	41

3. Metodología.....	41
3.1. Obtención de datos	41
3.2. Optimización de datos obtenidos.....	42
3.3. Selección de sistema embebido para el controlador inteligente	45
3.4. Comportamiento y evaluación de los sistemas embebidos.....	45
3.5. Implementación de la lógica difusa al controlador inteligente	47
3.6. Instalación de Raspbian	47
3.6.1. Inicio de Raspberry con Raspbian	49
3.6.2. Elección de lenguaje	50
3.6.3. Instalación de Librerías.....	50
3.6.3.1. Librería Matplotlib.....	50
3.6.3.2. Librería Scikit-Learn	51
3.6.3.3. Librería Python 3 Scipy	52
3.6.3.4. Parametrización de variables difusas en el controlador.....	52
3.7. Programación.....	59
3.7.1. Código en Raspberry Pi 3 B	60
IV.- CAPITULO 4	61
4. Resultados y discusión.....	61
4.1. Resultado de aplicación de Red Neuronal	61
4.2. Resultado de aplicación de Controlador inteligente en horno eléctrico.....	62
4.3. Resultados de acabados de soldadura	64
V. CAPITULO 5	66
5. Bibliografía.....	66
BIBLIOGRAFÍA.....	68
ANEXO A	71
ANEXO B	74
ANEXO C	75
ANEXO D.....	82
ANEXO E	89

LISTA DE TABLAS

<u>1.1. Desempeño de sistemas de control</u>	4
<u>2.1. Cuadro Resumen</u>	11
<u>2.2. Temperaturas de fusión de elementos más comunes en soldadura</u>	13
<u>2.3. Temperatura de fusión de elementos aleados</u>	14
<u>2.4. Beneficios y limitaciones de los métodos</u>	17
<u>2.5. Datos técnicos de los sistemas embebidos</u>	21
<u>3.1. Reglas difusas</u>	56
<u>3.2. Inferencia o memoria difusa</u>	57

LISTA DE FIGURAS

<u>2.1. Diagrama general de proceso</u>	12
<u>2.2. Perfil recomendado de temperatura para el calentado de soldadura de dispositivos SMD</u>	15
<u>2.3. Diagrama de flujo proceso de calentado</u>	18
<u>2.4. Arduino Mega 2560</u>	22
<u>2.5. Tiva C de Texas Instruments TM4C123GXL</u>	23
<u>2.6. Raspberry modelo Pi 3 B</u>	24
<u>2.7. PWM de Raspberry Pi</u>	26
<u>2.8. Sensor de temperatura MAX31855</u>	27
<u>2.9. Respuesta de resistencia RTD vs Temperatura</u>	28
<u>2.10. Inferencia difusa</u>	29
<u>2.11. Red de una capa</u>	39
<u>2.12. Función del nodo</u>	39
<u>2.13. Separación lineal de los conjuntos A y B</u>	40
<u>3.1. Sistema completo aplicado</u>	41
<u>3.2. Curva característica de perfil de temperatura</u>	42
<u>3.3. Diagrama de la red neuronal artificial</u>	43
<u>3.4. Diagrama a bloques de optimización de datos mediante RNA</u>	43
<u>3.5. Error de procesamiento RNA</u>	44

<u>3.6. Datos procesados en RNA</u>	44
<u>3.6. Diagrama de implementación en Raspberry</u>	45
<u>3.7. Diagrama eléctrico de potencia</u>	46
<u>3.8. Comportamiento de los sistemas embebidos</u>	46
<u>3.9. Diagrama de implementación en Raspberry</u>	47
<u>3.10. Sistema operativo Raspbian</u>	48
<u>3.11. Formateo de memoria microSD</u>	48
<u>3.12. Programa Etcher</u>	49
<u>3.13. Inicio de Raspbian en Raspberry Pi 3 B</u>	49
<u>3.14. Actualización de datos en la placa Raspberry</u>	50
<u>3.15. Instalació de librerías matplotlib</u>	51
<u>3.16. Instalación de Scikit-Learn</u>	51
<u>3.17. Instalación de librería Scipy</u>	52
<u>3.18. Universo error de temperatura</u>	53
<u>3.19. Universo derivada de error de temperatura</u>	53
<u>3.20. Universo de salida de ciclo de trabajo PWM</u>	57
<u>3.21. Diagrama de flujo de controlador inteligente</u>	59
<u>4.1. Resultado obtenido de proceso RNA en curva característica</u>	62
<u>4.2. Resultados de caldeo de soldadura</u>	63
<u>4.3. Transistor de mediana potencia</u>	64
<u>4.4. Circuito integrado microcontrolador</u>	64
<u>4.5. Botón metálico de montaje superficial</u>	65

ABREVIATURAS

Símbolo	Descripción
FL	Fuzzy Logic (Lógica Difusa).
FLC	Controlador Lógico Difuso (Fuzzy Logic Controller).
CF	Conjunto Difuso.
RNA	Red Neuronal Artificial.
AG	Algoritmos Genéticos.
FS	Fuzzy Sets (Colocaciones Difusas).
FM	Fuzzy Memory (Memoria Difusa).
Cap	Capacitor (Capacitor).
Res	Resistencia (Resistor).
if	Si (condicional, del inglés “if”).
°K	Grado Kelvin.
°C	Grado Centígrado.
THT	(por sus siglas en inglés Throw Hole Throw.)
SMD	Dispositivos de Montaje de Superficie (por sus siglas en inglés Surface Mountage Devices).
SMT	Tecnología de montaje superficial (por sus siglas en inglés Surface Mount Technology).

INTRODUCCIÓN

En las últimas décadas, la humanidad intenta tomar conciencia del problema que implica el aumento de consumo energético, que generan las crisis y conflictos energéticos, la utilización desmedida de materiales y combustibles fósiles aplicados en procesos de producción, genera una gran cantidad dióxido de carbono, principalmente aquellos productos que utilizan tecnología de vanguardia como son los dispositivos electrónicos, los cuales la venta y el consumo de éstos se ha incrementado considerablemente. Esto motiva el desarrollo e investigación de las energías renovables y el control de procesos energéticos.

Las grandes empresas de equipos electrónicos miniaturizan cada vez más sus dispositivos, mejorando sus procesos de producción y disminuyendo pérdidas, ahorrando espacio en las tarjetas de circuito impreso (Printed Circuit Board por sus siglas en inglés PCB) para el montaje de componentes, aplicando la tecnología de montaje superficial (Surface Mount Technology, SMT) llegando a estándares muy elevados.

Estos equipos utilizados en la producción de productos electrónicos tienen la capacidad de mejorar la respuesta y calidad de la información obtenida en un proceso específico. De tal forma que en la industria y en el sector tecnológico, se encuentran en vías de desarrollo y mejora continua para la aplicación en diversos procesos.[5]

Esto llevó a buscar nuevos métodos de montaje y soldadura cada vez más eficientes de los componentes electrónicos utilizados; que actualmente casi todos los equipos electrónicos utilizan la tecnología de montaje superficial, con esto surge la necesidad del manejo del control de temperatura en la soldadura, que requiere una curva de calentamiento específica para soldar dichos componentes.

A los hornos que usan estos métodos de soldadura se les denomina hornos eléctricos de reflujo. La importancia de estos hornos es fundamental, que de su correcto funcionamiento depende en gran medida la calidad general del transformado de la pasta de soldar y la agilidad en el proceso productivo. Teniendo en cuenta lo señalado, estos hornos deben contar con un sistema de control capaz de satisfacer los requerimientos necesarios. [21]

Si el proceso se encuentra mal calibrado, el caldeo y la soldadura en tarjetas de circuito impreso de montaje superficial, puede dar como resultado anomalías que afecten la calidad de la soldadura y el funcionamiento de dichos componentes, reduciendo el rendimiento del equipo electrónico en general.

Actualmente los hornos eléctricos cuentan con sistemas de control, como el control PID, circuito de control de lazo cerrado, el control difuso y el neuro difuso [28]; por lo cual se pretende implementar un sistema de control inteligente, que tenga la capacidad retentiva de las situaciones que se hallan presentado en los procesos previos de soldadura para ajustar de manera automática, sin necesidad de la intervención de un operador [16].

Estos sistemas de control inteligente traen consigo grandes ventajas con respecto a la toma de decisiones por parte del operador del horno, quien no tiene acceso inmediato a la información de las condiciones en las que se realizará el proceso de soldadura.

Al referirse a un control inteligente, esto lleva a tomar en cuenta la llamada Inteligencia Artificial (I.A.).

Haciendo un breve bosquejo sobre la Inteligencia Artificial encontramos que es la inteligencia exhibida por máquinas. En ciencias de la computación, una máquina inteligente ideal es un agente racional flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea. [24]

Los cuatro pilares básicos en los que se apoya la Inteligencia Artificial son los siguientes [26]:

- Búsqueda del estado requerido en el conjunto de los estados producidos por las acciones posibles.
- Algoritmos genéticos.
- Redes neuronales artificiales.
- Razonamiento mediante una lógica formal.

También existen distintos tipos de percepciones y acciones, que pueden ser obtenidas y producidas, respectivamente, por sensores físicos y sensores mecánicos en máquinas. [24]

La utilización de la Inteligencia Artificial es muy variada, como ejemplos se encuentran en el área de control de sistemas, planificación automática, reconocimiento del habla, reconocimiento de escritura, y reconocimiento de patrones. [26]

En esta tesis se utiliza un sistema control difuso retroalimentado por un sensor para medir la variable física de temperatura, abordar el procesamiento y pruebas de estrés, obteniendo como resultado la temperatura en tiempo real. El sensor se encuentra en un área estratégica para adquirir toda la irradiación en el área interna del horno; el controlador a su vez compensará las variaciones de temperatura de la lectura obtenida.

El sistema de control difuso propuesto tiene una estructura líneal con retropropagación, en donde se emplea un controlador interno y ajusta la posición de la temperatura actual con respecto a la temperatura deseada.

La temperatura es un factor crítico que define la condición del compuesto para la soldadura, por lo cual es muy importante y complejo el control de la temperatura hacia los dispositivos de montaje superficial.

Lograr este sistema de control inteligente implica grandes retos técnicos y de programación, porque se necesita monitorear en tiempo real una cantidad importante de datos de la variable física; esto requiere de la utilización de un microcontrolador para procesar la información obtenida por los sensores que proporcionarán la información necesaria para el control.

La importancia de implementar un sistema de control inteligente para un horno de soldadura de reflujo estriba en la necesidad de lograr una fundición perfecta de la pasta de soldar, los dispositivos y el sustrato sobre los que se montan, ya que requieren características especiales en las mismas [4].

Los sistemas de control descritos en los trabajos de Hwang y Goire, han mostrado gran desempeño en los procesos de soldadura y se aplican en numerosos sistemas de horneado de línea comercial como se muestra en la siguiente tabla 1.1.

Tabla 1.1 Desempeño de sistemas de control.

Horno difuso para el calentado de dispositivos de montaje superficial desarrollado por L. Hwang	Control Neuronal por Modelo de Referencia para un Sistema de Estructura Variable desarrollado por M. Goire.
<ul style="list-style-type: none">• Calidad en la señal procesada.• Procesamiento de la temperatura.• Optimización de producto terminado.	<ul style="list-style-type: none">• Optimización en tiempo real de la señal procesada.• Calidad de la señal procesada.• Mejoramiento en el proceso de datos.

Por lo tanto, en el siguiente capítulo se plantea la problemática, así como los objetivos de esta tesis. Posteriormente se dará a conocer los antecedentes importantes de la utilización de sistemas de control difuso, así como un breve resumen de los trabajos revisados que emplean algún tipo de control inherente a este tema.

I. CAPITULO 1

El presente capítulo muestra la descripción del problema del proyecto, así como las preguntas de investigación. Seguidamente se da la justificación, el alcance y los objetivos de esta tesis.

1.1 Descripción del problema

La gran mayoría de los circuitos electrónicos que se usan actualmente utilizan en casi la totalidad de su diseño, tecnología de montaje en superficie (SMT), dejando una minoría muy marcada únicamente en las etapas de gran potencia, para la utilización de dispositivos de tecnología de agujeros pasantes [34]. Principalmente la tecnología SMT pretende miniaturizar los dispositivos y mejorar la eficiencia de los procesos aplicados en su fabricación.

Con la finalidad de hallar una mayor eficiencia energética en los procesos productivos de los hornos eléctricos de reflujo, se hallan varios modelos de hornos electricos que utilizan sistemas de control proporcional-integral-derivativo (PID), sistemas de control difuso y sistemas de control de red neuronal [1,2,4,13], aplicados en diferentes condiciones para el caldeo de objetos. Por lo que surge la necesidad de diseñar un control de temperatura que cumpla con los requerimientos de eficiencia en el transformado de la soldadura de montaje superficial [4].

Actualmente en México no existe un método de control inteligente aplicado en hornos eléctricos de reflujo, utilizados para la fabricación de tarjetas electrónicas con tecnología SMD, cabe mencionar que existen hornos para tal uso, pero son de tecnología extranjera y los utilizan las empresas maquiladoras en nuestro país. La necesidad de desarrollar un control inteligente para un horno de reflujo, que pueda ser mejorado por medio de la aplicación de lógica difusa y que pueda dotarlo de las características y capacidades de un operador experto.

1.2 Preguntas de investigación

- ¿Qué ventajas tendrá el sistema al aplicar la técnica de control inteligente en el proceso de caldeo de soldadura en dispositivos de montaje superficial?
- ¿Cuáles son las características que debe poseer un controlador inteligente, para mejorar el desempeño en los procesos de caldeo de soldadura en dispositivos de montaje superficial?

1.3 Justificación

El porcentaje utilización de tecnología de montaje superficial depende principalmente de la aplicación a la que será dirigida el diseño del equipo en cuestión. Encontrándose en los sistemas de comunicación y telefonía celular una utilización del 100%, debido a las necesidades de miniaturización y al bajo consumo de potencia, para la operación de estos. Los equipos de cómputo se encuentran en la misma situación, ya que las marcas de vanguardia en este rubro utilizan igualmente la totalidad de sus elementos de montaje de superficie; de la misma manera los equipos de video domésticos y de uso institucional, utilizan el 98% de dispositivos de montaje superficie en su diseño, dejando la etapa de alimentación de potencia con tecnología through hole technology (THT de sus siglas en inglés) [34].

La utilización de estos dispositivos de montaje de superficie ha sido gradual y a la vez muy veloz en cuanto a la transición en el cambio de nuevas tecnologías como Chip sobre placa (COB de sus siglas en inglés) y rejilla de bolas de soldadura (BGA de sus siglas en inglés), causando que los costos de fabricación de los equipos vayan a la baja, a la vez que el peso de estos dispositivos disminuye significativamente en comparación con la tecnología THT. Queda claro que el porcentaje de la utilización de dispositivos de montaje de superficie en combinación con tecnología THT, dependerá del diseño y las necesidades de aplicación de los dispositivos a construir [34].

La evolución del número de dispositivos de montaje de superficie también es regida por la llamada ley de Moore, que vaticina que cada 18 meses se duplica el número de componentes dentro de los equipos electrónicos en un mismo espacio. [34]

En nuestro país las empresas maquiladoras cuentan con equipos para la fabricación de tarjetas electrónicas, pero el desarrollo de estos equipos se originó en el extranjero, por lo tanto, no tenemos acceso a los datos de fabricación ni desarrollo de los productos que se maquilan en dichos lugares, no existe la posibilidad de modificar los diseños ni los métodos de producción ya que todo está regido y controlado por los extranjeros.

Las tendencias tecnológicas apuntan a una gran miniaturización de los productos electrónicos, exigiendo una gran eficiencia energética en los mismos, por lo tanto, el uso de estas tecnologías mejora sustancialmente los tiempos, costos de producción y consumo de energía.

Algunas instituciones educativas, empresas y usuarios particulares del entorno geográfico cercano cuentan con equipos de soldadura de montaje superficial. Algunas empresas pretenden diseñar equipos con tecnología SMT para competir con otros fabricantes de línea comercial. Generalmente estos equipos de soldadura se ocupan para hacer mantenimiento correctivo a equipos de cómputo y telefonía celular, no incluyendo un controlador automático y son operados por un técnico experimentado en soldadura. Por lo que restringe las posibilidades de innovación y desarrollo de nuevos diseños de circuitos electrónicos.

En el entorno geográfico no se dispone de un equipo de soldadura de reflujo controlado automáticamente, por lo que surge la necesidad de un equipo con las características funcionales y de eficiencia energética como las requeridas en un horno de reflujo pero que utilice un sistema de control inteligente desarrollado aquí en México.

Para lograr una autonomía total en el control inteligente se opta por un método de inteligencia artificial basado en lógica difusa. Existen variadas aplicaciones de la lógica difusa que han alcanzado un gran éxito comercial. La experiencia ha mostrado resultados de mayor rendimiento que los obtenidos por un control convencional con algoritmos. El controlador lógico difuso muestra ser útil cuando los procesos son muy complejos para ser analizados por técnicas cuantitativas convencionales o cuando la información disponible es interpretada cualitativamente de manera inexacta o no certera.

Los principales beneficiados con esta investigación será la comunidad tecnológica, los usuarios y las empresas que decidan utilizar estos sistemas, motivando el desarrollo de

nuevos productos electrónicos, a su vez, estos sistemas de control podrán ser utilizados e implementados en equipos de diferente índole para mejorar su desempeño.

1.4 Alcance

El alcance que esta tesis:

- Definir el modelo del control inteligente.
- Simular el sistema de control inteligente.
- Habilitar un horno electrico resistivo para soldadura de montaje superficial.
- Implementar en un sistema embebido el control inteligente utilizando lógica difusa a través de la programación en el lenguaje Python.
- Implementar interfaz de potencia controlada por PWM.
- Ajustar en tiempo real el control difuso en el sistema embebido.

1.5 Hipótesis

Es posible usar Inteligencia Artificial para controlar la temperatura en un horno eléctrico para inducir calor a la soldadura en tarjetas de montaje superficial, asegurando el perfil de reflujo que comprende las tres etapas de calentamiento.

1.6 Objetivos

1.6.1. Objetivo general

Desarrollar un sistema basado en inteligencia artificial, para el control de temperatura del horno eléctrico de reflujo y que asegure una transferencia adecuada de calor.

1.6.2. Objetivos específicos

- Identificar y evaluar en tiempo real las señales de temperatura.
- Seleccionar y programar el sistema embebido.
- Diseñar el sistema de control inteligente.
- Implementar el control inteligente en el sistema embebido.
- Estudiar y acoplar la interfaz de potencia al sistema de control.
- Corroborar el funcionamiento del control de temperatura.

CAPITULO 2

En este capítulo se presenta el estado de arte de acuerdo con a la metodología planteada. Seguidamente se muestra el marco teórico de los conceptos principales de esta investigación.

2.1. Estado del arte

Los comienzos siempre implican un mayor esfuerzo, no importa en qué ámbito de la naturaleza se presente; de tal manera sucede en la generación de calor, en este caso específico, el caldeo para procesar el estado de la pasta de soldar requiere un gran consumo energético, proviniendo éste de una fuente eléctrica. Optimizar el consumo de energía para alcanzar la curva característica de calentamiento necesaria, implica desarrollar un sistema de control que lleve a tal objetivo.

Actualmente existen equipos de línea comercial para procesos de soldadura dispositivos de montaje superficial que utilizan sistemas de control moderno incluidos los sistemas PID, controladores difusos y controladores basados en redes neuronales. [1,2,5,6,7,30]

También existen equipos quizá no tan difundidos, que utilizan control adaptativo, sistema de control dinámico, sistemas de control de potencia lineal, sistemas de control inteligente basados en lógica neuro difusa. [3,10,11, 4]

Para lograr alcanzar un control eficiente es necesario la utilización de sistemas electrónicos ahora llamados sistemas embebidos, los cuales contienen dispositivos capaces de procesar datos a gran velocidad. [31,32,33] Muchos de los equipos de horneado para placas de circuito impreso, utilizan sistemas de control electrónico basados en microprocesadores, por lo tanto, llegan a ser sistemas electrónicos muy complejos y costosos. Un sistema embebido no utiliza un procesador en sí, utiliza un microcontrolador que incluye en su diseño los periféricos electrónicos necesarios para funcionar como una pequeña microcomputadora, esto incluye una memoria aleatoria (SRAM), memoria de solo lectura (EEPROM) en la que se graba el código de programa de control o su firmware, en el cual lleva su código básico.

De la mano con los sistemas embebidos se utiliza el software embebido mismos que crean una extraordinaria sinergia, tales sistemas y software tienen un futuro promisorio por qué tienen un avance vertiginoso [33].

De los anteriores artículos se toman los que de manera explícita definen valores de optimización utilizados en sus trabajos experimentales en la Tabla 2.1. En donde se puede observar la mejora de los controles en diferentes aplicaciones.

Tabla 2.1 Cuadro resumen

Referencia	Método de control	Prueba experimental	Datos Extra
L. Kwang	Control Difuso / Neuro Difuso	Calentado de soldadura en distintos compuestos.	Curva característica del perfil de calentado.
C. Joers	Control Lineal	Calentado en un horno de tubo.	-
B. Costa	Control Difuso	Calentado en un horno solar.	Modelos difusos.
P. López	Control no lineal con retroalimentación	Control de temperatura en reactores.	-
D. Linkens	Control inteligente adaptativo	Control inteligente en dosificación de anestesia.	Modelo de control inteligente.
Y. Boutalis	Control Adaptativo	Control adaptativo y sistemas de identificación.	Modelo de control adaptativo.
R. García	Control de temperatura lineal continuo.	Control de temperatura aplicado en el tostado de café.	Modelo de referencia.

2.2. Marco teórico

A continuación, como parte del documento de tesis se presenta el marco teórico, integrando los temas principales que se requiere conocer.

El proceso por controlar es el siguiente mostrado de forma general en la Figura 2.1, por lo que se necesitan los conocimientos generales de temperatura, soldadura, acabado de soldadura, hornos eléctricos de reflujo, control difuso y redes neuronales artificiales.

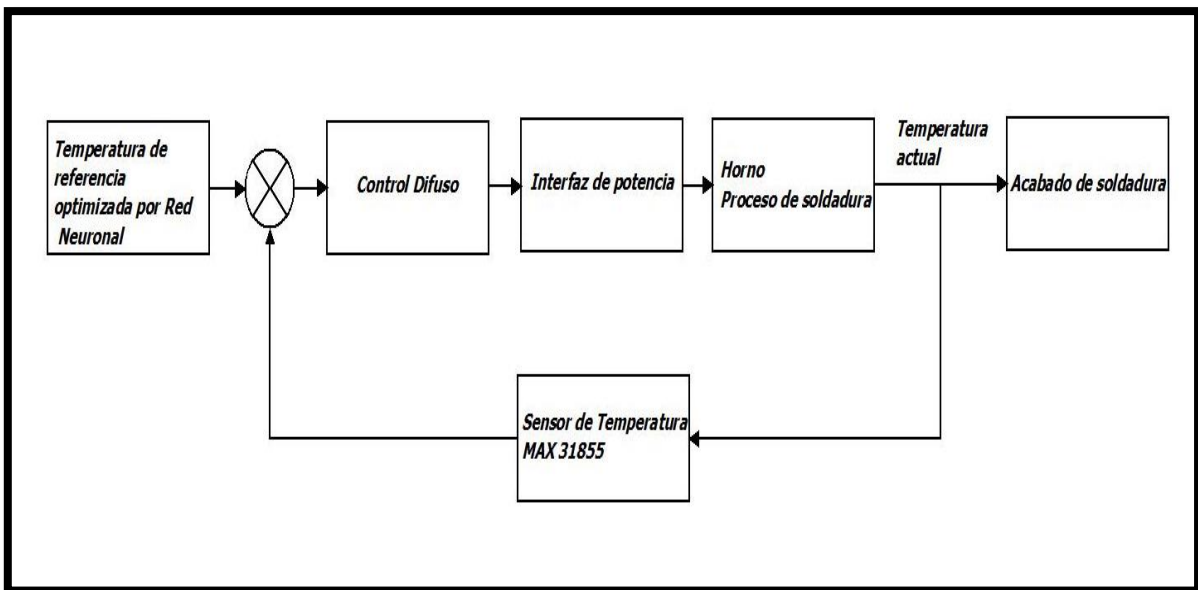


Figura 2.1 Diagrama general de proceso.

En cada uno de los aspectos de esta investigación se presenta la base fundamental para la implementación y óptimo funcionamiento del horno de reflujo.

2.2.1 Temperatura

La temperatura es una magnitud escalar que refleja la cantidad de calor, conocida como energía cinética, que es la energía asociada a los movimientos de las partículas del sistema, siendo en sentido traslacional, rotacional, o en forma de vibraciones y la temperatura se maneja en unidades de grados Kelvin ($^{\circ}\text{K}$) y grados centígrados ($^{\circ}\text{C}$).

2.2.1.1 Medios de transferencia de calor

Los medios de transferencia de calor son los siguientes:

- **Conducción:** La transferencia de calor de un punto a otro de un mismo cuerpo teniendo diferente temperatura.

- **Convección:** La transferencia de calor de un cuerpo a otro que tienen diferente temperatura.
- **Radiación:** La transferencia de calor a través de ondas electromagnéticas o radiantes de un cuerpo a otro con diferente temperatura.

2.2.2 Soldadura

La soldadura es un proceso de fijación en donde se realiza la unión de dos o más piezas de un material separados mediante la transferencia de calor, presión o la combinación de ambos para convertirlos en una nueva pieza.

2.2.3 Aleaciones de soldadura

Las soldaduras son generalmente descritas como aleaciones que licúan a temperaturas aproximadas de 400 ° C. Los elementos comúnmente usados en aleaciones de soldadura son Estaño (Sn), Plomo (Pb), Plata (Ag), Bismuto (Bi), Indio (In), Antimonio (Sb) y Cadmio (Cd). Sus puntos de fusión están listados en la Tabla 2.2. (Hwang, 2004).

Tabla 2.2 Temperaturas de fusión de elementos más comunes en soldadura.

	Sn	Pb	Ag	Bi	In	Sb	Cd
°C	232	328	961	271.5	156.6	630.5	321.2

A la aleación binaria Estaño-plomo que es muy conocida, pueden agregarse otras aleaciones binarias como Estaño-Plata, Estaño-Antimonio, Estaño-Indio, Estaño -Bismuto, Plomo-Indio y Plomo-Bismuto.

Las aleaciones ternarias incluyen Estaño-Plomo-Plata, Estaño -Plomo -Bismuto y Estaño-Plomo-indio indicados en la Tabla 2.3.[4]

Tabla 2.3 Temperaturas de fusión de elementos aleados.

Composición Química	°C	°F
70Sn/30Pb	193	380
63Sn/37Pb	183	361
60Sn/40Pb	190	375
50Sn/50Pb	216	420
40Sn/60Pb	238	460
30Sn/70Pb	255	491
25Sn/75Pb	266	511
10Sn/90Pb	302	575
5Sn/95Pb	312	594
62Sn/36Pb/2Ag	179	355
10Sn/88Pb/2Ag	290	554
5Sn/95Pb	312	594
62.5Sn/36Pb/2.5Ag	179	355
10Sn/88Pb/2Ag	290	554

2.2.4 Pasta de soldar

En definición, es una mezcla química cinéticamente estable de polvo de aleaciones de soldadura, flux y un vehículo, capaces de unirse en una pasta moldeable que logre hacer uniones eléctricas entre componentes y un sustrato.

La pasta de soldar en polvo es el componente mayoritario en la mezcla, el vehículo primario contiene y suministra el polvo de soldar, el flux limpia la aleación en polvo y el sustrato y las terminales de los componentes a soldar para lograr una mejor formación de soldadura y una unión eléctrica eficiente [4].

Las características de la pasta de soldar son descritas en los siguientes parámetros [18]:

- Apariencia física.
- Duración y Estabilidad.
- Dispersabilidad a través de agujas finas.
- Capacidad de impresión en pantalla.
- Capacidad de impresión con estencil.
- Viscosidad.
- Caída fría.
- Tiempo de tachuela.
- Adhesión.
- Tiempo de vida de exposición.
- Calidad y consistencia.

- Compatibilidad con superficies a ser unidas.
- Flujo apropiado.
- Humectabilidad.
- Fenómeno de des humectación.
- Fenómeno de guiño.
- Fenómeno de lixiviación.
- Cantidad y propiedades de residuos.
- Corrosividad de residuos.
- Disponibilidad de limpieza de residuos.
- Apariencia de la unión soldada.
- Vacío de la junta de soldadura.

2.2.5 Curva de temperatura en soldadura de montaje superficial

Un factor muy importante en el calentamiento de los dispositivos de montaje superficial es la relación muestreo-temperatura. La Figura 2.2 muestra el perfil recomendado para el calentado de la soldadura con la mezcla de pasta de soldar en conjunción con el líquido flux para optimizar la curva de reflujo [4].

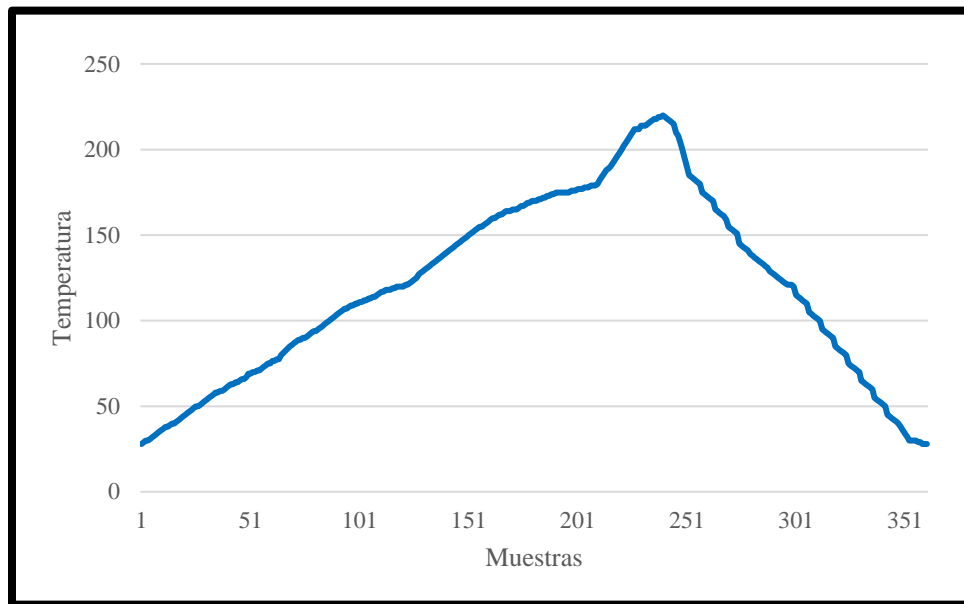


Figura 2.2 Perfil recomendado de temperatura para el calentado de soldadura de dispositivos SMD

El perfil de temperatura de recalentado representa la relación de temperatura y las iteraciones durante el proceso del calentado, dependiendo no solo del parámetro ajustado durante la

capacidad y flexibilidad del equipo. Específicamente, la temperatura instantánea genera condiciones y experiencias determinadas por:

- Ajuste de temperatura en toda la zona de control.
- Temperatura ambiente.
- Masa por placa.
- Masa total en el calentamiento.
- Eficiencia de la fuente generadora y transferencia de calor.

2.2.6 Fundente (FLUX)

La clave principal para lograr una buena soldadura consiste en que las superficies de los materiales a unir estén limpias. La limpieza debe ser mantenida durante el proceso de soldadura para asegurar que las caras metálicas de los componentes y el sustrato logre una buena conductividad eléctrica y mecánica.

Este proceso de limpieza es llamado Fluxing y el material utilizado es el Flux.[4].

2.2.7 Metodologías de soldadura

Tipos: Los métodos de reflujo comercialmente disponibles incluyen conducción, infrarrojos, fase de vapor, gas caliente, convección, inducción, resistencia y laser. Cada uno de estos métodos de reflujo tienen características únicas y méritos en costo, desempeño y conveniencia operacional y se muestra en la siguiente tabla 2.4.

Tabla 2.4 Beneficios y limitaciones de los métodos de reflujo.

METODO DE REFLUJO	BENEFICIOS	LIMITACIONES
Conducción	<ul style="list-style-type: none"> • Equipos de bajo costo. • Cambios rápidos de visibilidad durante el reflujo. 	<ul style="list-style-type: none"> • Superficies planas y de una sola cara, área limitada.
Infrarrojo	<ul style="list-style-type: none"> • Alta producción de calor. • Parámetros de procesamiento y control de temperaturas versátiles. • Perfil reflujo consistente. 	<ul style="list-style-type: none"> • Dependencia geotérmica.
Gas caliente	<ul style="list-style-type: none"> • Bajo costo. • Rango de calentamiento alto. • Calentamiento localizado. 	<ul style="list-style-type: none"> • Control de temperatura limitado.
Convección	<ul style="list-style-type: none"> • Gran desempeño y versatilidad. 	<ul style="list-style-type: none"> • Lento calentamiento, gran demanda de flux.
Inducción	<ul style="list-style-type: none"> • Rango rápido de calentamiento y capacidad. 	<ul style="list-style-type: none"> • Solo aplicable a partes metálicas no magnéticas.
Laser	<ul style="list-style-type: none"> • Calentamiento localizado con gran intensidad, tiempo, corto soldadura superior, previene ruptura de empaques 	<ul style="list-style-type: none"> • Equipo de alto costo, requiere pasta especial limitado para soldar en masa.

Los parámetros que afectan tanto a la producción como a la integridad de las uniones de soldadura se incluyen en el diagrama de flujo se presenta el proceso de calentado de los diferentes tipos aleaciones a seleccionar figura 2.3.[4]

- **Temperatura de Pre calentamiento.**

Es la temperatura medida en el encapsulado, es un límite dado por fabricante oscilando la temperatura de 90°C a 140°C.

- **Tiempo de Pre calentamiento.**

Es el tiempo durante el cual se supera la temperatura de fusión.

- **Pico de Temperatura.**

Es la temperatura medida arriba del encapsulado, no debe superarse la temperatura para evitar daños en el chip. La temperatura oscila entre 245°C y 260°C.

- **Tiempo de Permanencia del Pico de Temperatura.**

Es la temperatura deseada en la junta de soldadura para lograr una correcta unión de soldadura. Importa a temperatura mínima necesaria para que se solde correctamente el encapsulado y la placa.

- **Enfriamiento.**

Es la zona del reflow donde los materiales y componentes vuelven a la temperatura ambiente. Se debe tener cuidado en esta etapa y tener de forma gradual para evitar shock térmico en los componentes y la placa.

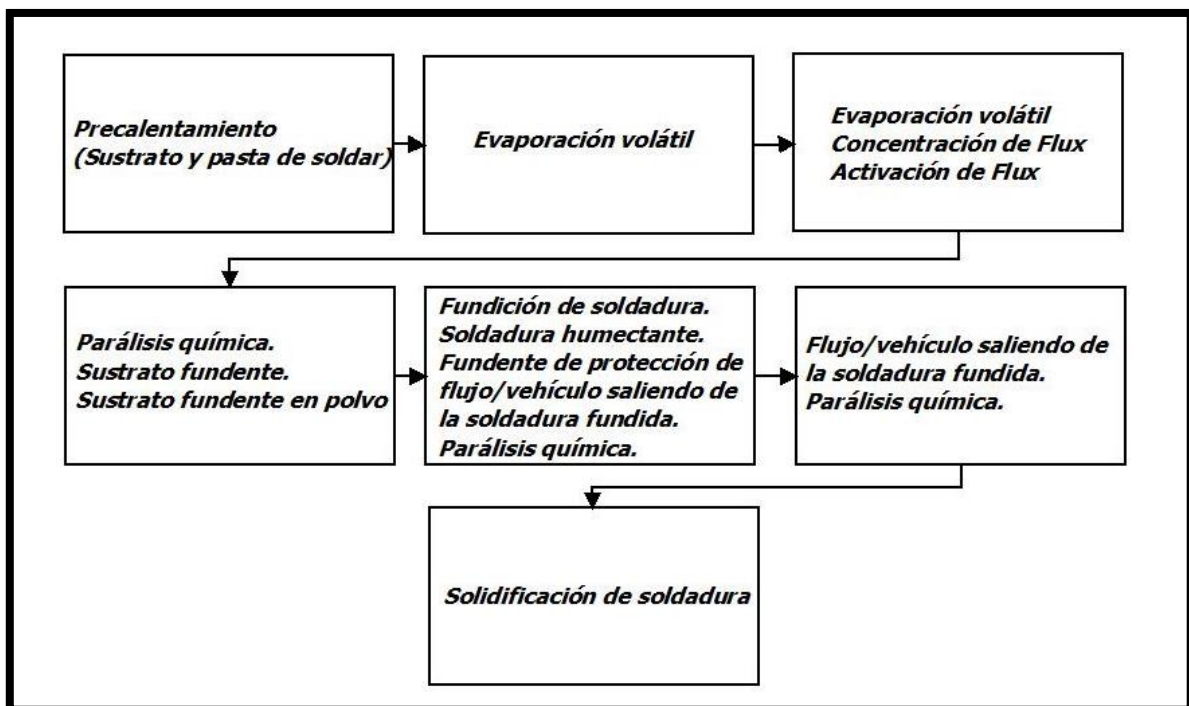


Figura 2.3 Diagrama de flujo proceso de calentado.

2.2.7.1 Efectos en el perfil de recalentado

El perfil de reflujo utilizado para la fabricación de tarjetas de montaje superficial tiene una relación directa con el rendimiento de la fabricación de los componentes, en conjunto con la integridad de la soldadura y su fiabilidad.

Las áreas específicas que tienen gran impacto de afectación son por la siguiente enumeración de daños que ocurren dentro del proceso, cada área puede verse afectada, en escalas de daño, dentro de las tres etapas principales de calentamiento [4]:

- Distribución de temperatura en todo el conjunto.
- Agrietamiento del encapsulado.
- Bolas en la soldadura.
- Reborde de soldadura.
- Capacidad de humectación.
- Capacidad de limpieza de residuos.
- Aspecto y características de los residuos.
- Áreas de vacíos en las juntas de la soldadura.
- Reacción metalúrgica entre la soldadura y la superficie del sustrato.
- Microestructuras entre las juntas de la soldadura.
- Nivel residual de estrés del ensamble de la placa.

2.2.8 Horno de Convección

El horno de convección es un equipo que utiliza energía eléctrica, y que es de un uso muy extendido, por su costo accesible, fácil manejo y gran desempeño. Con los sistemas de programación que se incorporan, son más útiles y confiables [15].

Los hornos de convección tienen grandes ventajas para la fusión de metales, siendo destacadas las siguientes [4]:

- Pueden obtenerse temperaturas muy elevadas hasta de 3500° C en algunos tipos de hornos eléctricos.

- Puede controlarse la velocidad de elevación de temperatura, y mantener entre límites muy precisos, con regulaciones completamente automáticas.
- La carga queda por completo libre de contaminación de gas combustible.
- Puede controlarse perfectamente la atmósfera en contacto con la masa fundida, haciéndola oxidante o reductora a voluntad, e incluso en algún tipo de horno puede operarse en vacío.
- Tienen mayor duración de revestimientos que en los demás tipos de hornos.
- Se instalan en espacios reducidos.

La operación dentro de este tipo de equipos se realiza con mayor higiene que en los hornos de calentado por vía gas y combustibles fósiles.

2.2.9 Sistemas Embebidos

Los sistemas embebidos son productos utilizados por la Inteligencia Artificial que se han convertido en algo común en casi todos los mercados de consumo industrial [30]. El papel fundamental del sistema embebido no es la transformación de datos, si no la interacción del mundo con el ser humano, ofreciendo tres atributos especiales para el desarrollo de software embebido, que son [31]:

- Limitaciones de recursos en el software embebido.
- Confiabilidad en el procesamiento de señales.
- La respuesta en el tiempo real de aplicaciones.

2.2.9.1 Consideraciones de los sistemas embebidos.

El desarrollo del programa genera altos retos ya que en cada sistema embebido tiene su singularidad en el lenguaje de programación dedicado, aunque los dispositivos actuales centran la razón de ser en la comunicación con la inteface humano máquina (HMI) [31], con esto restringe ciertas operaciones físicas, para los sistemas de control.

Existen 4 grandes retos en su desarrollo:

- Planeación del proceso de desarrollo de software embebido.
- Establecimiento de una metodología.
- Aseguramiento de la calidad del software embebido.
- Herramientas de diseño.

2.2.9.2 Elección de sistemas embebidos.

Los sistemas embebidos elegidos y sus características en el presente trabajo son los siguientes:

- Arduino® Mega 2560
Modelo: Atmega 2560.
Lenguaje de programación: C++.
Programa: Arduino IDE.
- Tiva C ®
Modelo: EK-TM4C123GXL.
Lenguaje de programación: C++.
Programa: Energía IDE.
- Raspberry®
Modelo: 3 B 2016.
Lenguaje de programación: Python 3.5.
Programa: Thonny IDE.

A continuación, en la siguiente Tabla 2.5 se designan las características específicas de los sistemas embebidos [31]

Tabla 2.5 Datos técnicos de los sistemas embebidos elegidos.

	Mega (2560)	Tiva c launch pad	Cortex-A53 (ARMv8)
Procesador	ATmega2560	Cortex-M4F	Cortex-A53 (ARMv8)
Anchura	8 bits	32 bits	64 bits
Velocidad	16 MHz	80 MHz	1.4 GHz
Pines digitales	54	40	40
Pines PWM	16	16	16
Convertidores ADC	16	3	16
Convertidores DAC	0	1	0
Interrupciones	24	24	24
Memoria FLASH	256Kb	256Kb	-
SRAM	8kb	32kb	1Gb
EEPROM	4kb	2kb	-

2.2.9.2.1 Sistema embebido Arduino.

Arduino mega 2560 tiene como microcontrolador Atmel 2560. Atmel es una compañía de semiconductores fundada en 1984, comercializa productos como dispositivos de radiofrecuencia, microcontroladores, memorias flash y memorias EEPROM, como se muestra en la siguiente Figura 2.4.

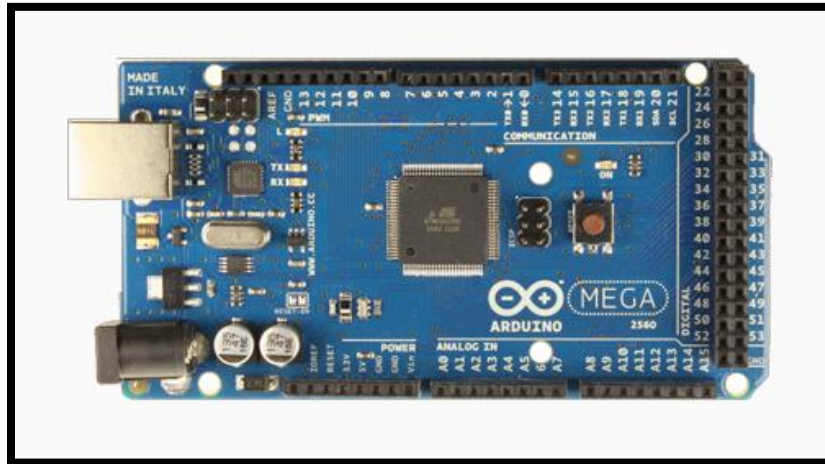


Figura 2.4 Arduino Mega 2560.

Las características del Arduino Mega 2560 son las siguientes:

- 256 KB de memoria flash.
- CPU AVR de 8 Bits.
- 8 KB de memoria SRAM.
- Reloj de cristal 16 MHz.
- 54 puertos de entrada/salida.
- 16 puertos analógicos y 15 puertos digitales PWM.
- Se caracteriza por 135 instrucciones.

2.2.9.2.2 Sistema embebido Tiva C.

Texas Instruments desarrolló microcontroladores con arquitectura ARM, conocidos los dispositivos como ARM Tiva C. El microcontrolador es denominado como encapsulado de bajo consumo de energía, como se muestra en la Figura 2.5.



Figura 2.5 Tiva C de Texas Instruments TM4C123GXL.

Las características del Launch pad Tiva C son las siguientes:

- 256 KB de memoria flash.
- CPU ARM Cortex -M4
- 32 KB de memoria SRAM.
- Reloj de cristal de precisión.
- 40 puertos de entrada/salida.
- 16 puertos analógicos y 16 puertos digitales PWM.

2.2.9.2.3 Sistema embebido Raspberry.

Raspberry Pi es una placa de computadora que se puede considerar como un dispositivo SMD (Dispositivo de montaje superficial por sus siglas en inglés “Surface Mounted Device”) de bajo costo desarrollado en el Reino Unido por la Fundación Raspberry Pi (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, sin embargo, su comercialización dio lugar hasta el año 2012.

El dispositivo con el cuál se lleva a cabo la investigación del controlador inteligente es con el modelo Raspberry Pi 3B que cuenta con las siguientes características de Hardware y se muestra en la siguiente Figura 2.6:

- CPU: Quad-Core Cortex A7 a 900 MHz.
- GPU: VideoCore IV de doble núcleo.
- RAM: 1 GB DDR2
- Puertos:

- 4 x USB 2.0.
- 1 x 40 GPIO pin.
- 1 x HDMI ver. 1.4.
- 1 x Ethernet.
- 1 x Combo audio/mic.
- 1 x Interfaz de cámara (CSI).
- 1 x interfaz de pantalla (DSI).
- 1 x Micro SD.
- 1 x Núcleo Gráfico 3D.
- Módulo Bluetooth.
- Módulo de Wi-Fi b/g/n en banda de 2.4 GHz.

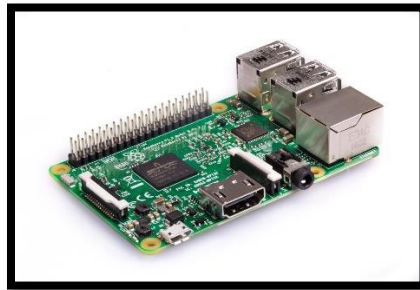


Figura 2.6 Raspberry modelo Pi 3B.

2.2.9.2.3.1 Raspbian.

El software que utiliza el sistema Raspberry Pi está basado en una versión adaptada de Debian SO (Debian es un sistema operativo basado en Linux y es una distribución de software libre) llamado Raspbian. Raspberry Pi también permite el uso de otros sistemas operativos, incluido una versión liviana de Windows 10. Pero, debido a las limitaciones que el sistema operativo de Windows presenta en cuanto al acceso a la GPIO de la Raspberry se opta por instalar Raspbian en nuestro sistema embebido.

Raspbian es una distribución del sistema operativo libre GNU/Linux, este cuenta con optimizaciones para el CPU del Raspberry Pi, permitiendo realizar cálculos con coma flotante por hardware. Raspbian es el sistema operativo oficial de la fundación. Este viene con software preinstalado para educación, programación y usos en general. Contiene Python, Scratch, Sonic Pi, Java y demás programas. La última revisión estable del software tiene fecha al 13 de noviembre del 2018.

2.2.9.2.3.2 Python.

Python es un lenguaje de programación que data del año 1980 y el creador de éste interprete fue Guido Van Rossum. La filosofía que utiliza Python para su sintaxis permite que el código en su ambiente sea más sencillo en comparación a otras plataformas de programación.

Se dice que Python es un lenguaje multiparadigma puesto que permite desenvolverse en múltiples series de instrucciones como lo es la programación orientada a objetos, programación funcional a partir de list, programación imperativa, scripts, aplicaciones web, aplicaciones de escritorio, entre otros.

En la actualidad existen varias librerías para Python que permiten desarrollar proyectos de manera más eficiente, librerías como lo son “Matplotlib” que son series de instrucciones que nos permiten elaborar gráficas 2D con variedad de propiedades, diseños y formatos, así como ambientes interactivos a través de plataformas, el objetivo de Matplotlib es el facilitar la representación gráfica con la menor cantidad de líneas de código.

Otra librería bastante útil es “NumPy” que es una paquetería que contiene herramientas para la computación científica en Python. Esta añade soporte para grandes arreglos y matrices multi-dimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar esos arreglos.

SciPy es una librería basada en el ecosistema Python de código abierto para matemáticas, ciencia e ingeniería. Esta contiene modulos para optimización, algebra linear, integración, interpolación, funciones especiales, imagen y procesado de señales entre otras funciones comunes en la ciencia e ingeniería.

Por último, pero no menos importante está la librería “scikit-learn” la cual está diseñada para el aprendizaje maquina (Machine Learning) y el análisis de información. Estas tres librerías en conjunto son poderosas herramientas que permiten el desarrollo de aplicaciones en Python con un alto grado de exactitud y precisión.

2.2.9.2.3.3 Librería Scikit-learn

Scikit-Learn es la librería más útil para el Machine Learning dentro de Python, su característica principal es que se acomoda en los ambientes de código libre, y se vuelve reutilizable en varios contextos de aplicación, dentro de las áreas académicas e industriales.

Esta librería está constituida por SciPy(Science Python) e incluye los siguientes paquetes:

- Numpy: librería de matriz n-dimensional base.
- Pandas: estructura de datos y análisis.
- SciPy: librería fundamental para la informática científica.
- Matplotlib: trazado completo en 2D.
- Ipython: consola intectiva mejorada.
- SymPy: matemática simbólica.
- Scikit-Fuzzy:

2.2.9.2.3.4 Librería Scikit-Fuzzy.

Scikit es una colección de algoritmos de lógica difusa destinada al uso en SciPy Stack, dirigido al lenguaje de Python. Incluyendo los paquetes de la librería de lógica difusa.

2.2.9.2.3.5 Modulación de ancho de pulso.

La modulación por ancho de pulsos o PWM (por sus siglas en ingles Pulse-Width Modulation) de una señal digital, que modifica el ciclo de trabajo de una señal periódica para controlar la cantidad de energía que se envía, dicho de otro modo, modifica el tiempo de la señal para simular una señal analógica como se muestra en la Figura 2.7.

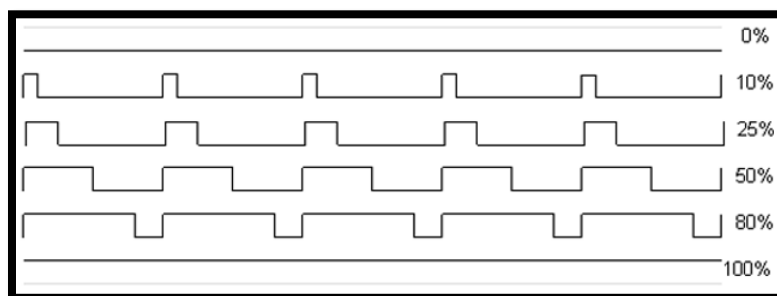


Figura 2.7 PWM de Raspberry Pi.

2.2.10 Sensor MAX31855

Los sensores son dispositivos que detectan magnitudes físicas como presión, temperatura, humedad, velocidad. Al detectar la magnitud física, el sensor transduce la señal en elementos binarios, generalmente en energía eléctrica y convertida en trenes de pulsos.

El sensor de temperatura transforma la energía calórica en energía eléctrica, dando una señal de corriente directa.

Para este trabajo se elige el sensor de temperatura MAX31855, mostrado en la Figura 2.8. que contiene la librería para convertir la variable física en cualquier denominación de medición de temperatura ya sea en grados Celsius °C ó en grados Fahrenheit °F. La característica del sensor de temperatura es la siguiente:

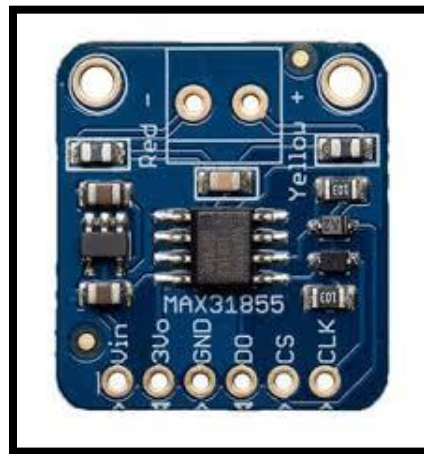


Figura 2.8 Sensor de temperatura MAX31855

- Conversión simple de la resistencia RTD de platino a valor digital como se muestra en la Figura 2.9.
- Manejar RTD de platino de 100 Ohms.
- Compatible con conexiones de sensor de 2, 3, 4 hilos.
- Interfaz compatible con SPI.
- Resolución ADC de 15 bits.
- Precisión total en captura de la señal.
- Tiempo de conversión de 21 ms.
- Protección en entradas por sobreprotección.

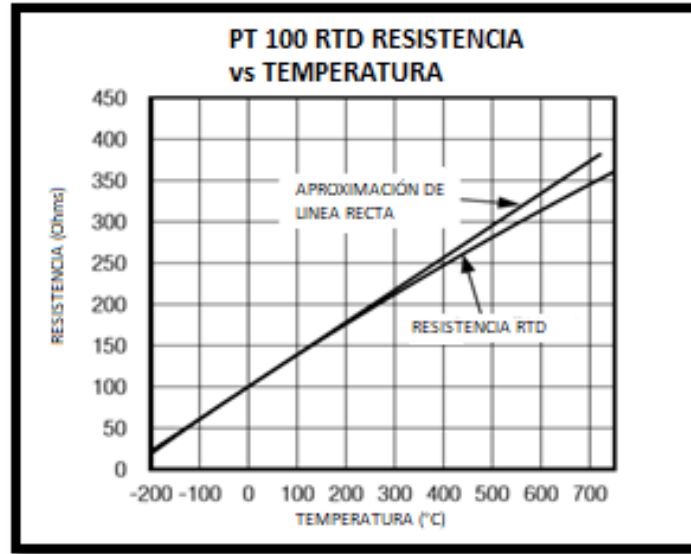


Figura 2.9 Respuesta de resistencia RTD vs Temperatura.

2.2.11 Inteligencia Artificial

La Inteligencia Artificial es una combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano.

Las ramas que componen a la Inteligencia Artificial son principalmente tres disciplinas [30]:

- Lógica Difusa.
- Redes Neuronales Artificiales.
- Algoritmos genéticos.

La elección principal para este trabajo es la Lógica Difusa ya que ofrece la adaptabilidad y la bivalencia para los sistemas embebidos, optimizando los recursos informáticos de los mismos [2,4].

2.2.12 Lógica Difusa

La Lógica Difusa es una rama de la Inteligencia Artificial que permite computar y analizar las cadenas de caracteres [6] comúnmente como binarias, que se puede utilizar para reducir el conjunto de datos [20], convirtiendo la posición de entre informaciones falsas o verdaderas (ceros y unos), agregando valoraciones intermedias para aumentar una infinidad de valores preestablecidos por nuestra lingüística [12].

Al referirse en Lógica Difusa a un conjunto con la Inteligencia Artificial actualmente se afirma que la estabilidad de cualquier sistema experto puede ser 50% cierta o 50% falsa, ya que los dispositivos integrados en la Inteligencia Artificial son bivalentes [26].

La efectividad de los sistemas lógicos difusos motiva los siguientes objetivos:

- Mejorar la robustez con respecto a los métodos clásicos.
- Simplificar el diseño de los modelos e implementar de forma concreta los sistemas.
- Autonomía total y adaptabilidad.

En la Lógica Difusa se cuenta con 2 tipos de métodos aplicados comúnmente en la industria, el método Mamdani y el método Tagashi-Sugeno, ya que estos métodos ofrecen una gran adaptabilidad dentro de los sistemas embebidos [6].

Su principal ventaja de la Lógica Difusa es la eliminación parcial de un modelo matemático a integrar dentro del sistema inteligente, teniendo la fusificación, base del conocimiento y la defusificación, a continuación, se muestra un modelo una máquina de inferencia difusa en Figura 2.10, de esta manera se sustituye la creación de un modelo matemático [6].

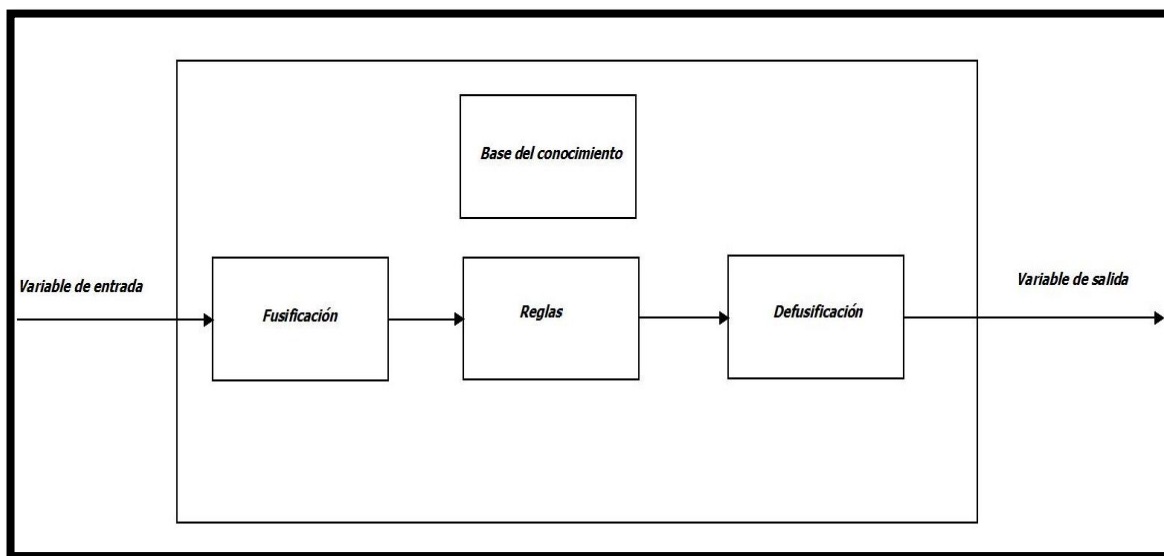


Figura 2.10 Inferencia difusa.

Para tener el proceso del sistema difuso se tiene que indicar, los conjuntos en cada universo, siendo de entrada o de salida, para definir las funciones de pertenencia en cada uno de los valores lingüísticos predefinidos por el operador o por la información adquirida en eventos anteriores, distribuyendo en el universo dedicado a cualquier variable física.

2.2.12.1 Conjunto Difuso

Como la lógica difusa es la definición de grados de pertenencia, emplea valores continuos entre 0 y 1. El concepto se asocia a los valores de pertenencia que genera la inferencia humana, teniendo los valores dentro de la fuzzificación y la defuzzificación dando valores discretos dentro de la lógica. De este modo, se diferencia de la lógica clásica a la idea bica del conjunto difuso que es el elemento de conjunto determinado por el grado de pertenencia.

Un conjunto difuso (CF) en un universo de discurso U, es caracterizado por una función de afiliación u_f el cual toma valores en el intervalo [0,1] llamado $u_f:U[0,1]$.

Un conjunto difuso puede ser visto como una generalización del concepto de un conjunto ordinario cuya función de membresía solo toma 2 valores (0,1). Así que un conjunto difuso F en U puede ser representado como un conjunto de pares ordenados o un elemento genérico u y este grado de pertenencia de función de membresía:

$$F = \{(u, u_f(u)) : u \in U\} \quad \text{Ec. 1}$$

Donde:

- F : Conjunto universo
- U : Universo de discurso
- u : Grado de pertenencia

Cuando U es continuo, un conjunto difuso F puede ser escrito de manera concisa como

$$F = \sum_{n=1}^n \frac{u_f(u_i)}{u_i} \quad \text{Ec.2}$$

Donde:

- u: Universo continuo
- f: Conjunto difuso universo

2.2.12.2 Soporte, Punto cruce, Fuzzy Singleton

El soporte o universo de un conjunto difuso F es el conjunto nítido de todos los puntos u en U tal que $u_f(u) > 0$. En particular el elemento u en U al cual $u_f = 0.5$, es llamado el punto de cruce y un conjunto difuso cuyo apoyo es un simple punto de U con $u_f = 1.0$ es requerido como un singleton difuso.

2.2.12.3 Colocación de operaciones teoreticas.

(Operaciones con conjuntos teoréticos)

Para A y B son 2 conjuntos difusos en U con funciones de membresía u_a y u_b , respectivamente.

Las funciones teoréticas de unión, intesección y complemento de los conjuntos está definida a travez de sus funciones de membresia. Más específicamente, vea lo siguiente:

Unión

La función de membresía $u_{a \cup b}$ de la unión AUB es un punto definido para todos los valores de u en U dado por

$$u_{(A \cup B)(u)} = \max \{u_a(u), u_b(u)\} \quad \text{Ec. 3}$$

Donde:

$u_{(A \cup B)(u)}$: Grado de pertenencia de salida

u_a : Grado de pertenencia 1

u_b : Grado de pertenencia 2

Intersección

La función de membresía $u_{A \cap B}$ de la intersección $A \cap B$ es un punto definido para todos los valores de u en U dado por

$$u_{A \cap B}(u) = \min \{u_A(u), u_B(u)\} \quad \text{Ec.4}$$

Donde:

$u_{(A \cap B)(u)}$: Grado de pertenencia de salida

u_a : Grado de pertenencia 1

u_b : Grado de pertenencia 2

Complemento

La función de membresía u_a del complemento de un conjunto difuso A está definida para todos $u \in U$ dada por

$$u_A(u) = 1 - u_A(u) \quad \text{Ec.5}$$

Donde:

u_A : Grado de pertenencia

Producto cartesiano

Si A_1, \dots, A_n son conjuntos difusos u_1, \dots, u_n , respectivamente, el producto cartesiano de A_1, \dots, A_n es un conjunto difuso en el producto espacial $u_1 * \dots * u_n$ con una función de membresía.

$$\begin{aligned} u_{A_1} \times \dots \times u_{A_n}(u_1, u_2, \dots, u_n) &= \min \{u_{A_1}, \dots, u_{A_n}\} \text{ ó} \\ &= u_{A_1}(u_1) * u_{A_2}(u_2) * \dots * u_{A_n}(u_n) \end{aligned}$$

2.2.12.4 Máquina de inferencia difusa

La máquina de inferencia difusa es una plataforma computacional basada en los conceptos de la teoría de los conjuntos difusos, esta plataforma es aplicable en diferentes áreas de la ingeniería, economía, administración entre otros. La máquina de inferencia difusa se puede aplicar en clasificación de datos, sistemas expertos, robots o reconocimiento de patrones. La máquina de inferencia difusa está formada por tres conceptos esenciales de la lógica difusa. Son la Fusificación, las Reglas de la base del conocimiento y la Defusificación. En la Figura 2.10 se mostró la máquina de inferencia difusa aplicable en cualquier controlador.

2.2.12.5 Fusificación

La fusificación es un proceso mediante el cual se obtienen las variables de entrada y comienza el proceso de la máquina de inferencia difusa, realiza el proceso de la variable que es evaluada y asigna un grado de pertenencia dentro de los conjuntos difusos, ya que los conjuntos difusos son preestablecidos por el operador. Para continuar con el proceso se usan las funciones de pertenencia de los conjuntos difusos.

2.2.12.6 Inferencia difusa

Para poder establecer la o las salidas que nuestra lógica va a producir cuando se ejecute el proceso del controlador difuso se requieren de reglas difusas: Se llaman reglas difusas al conjunto de proposiciones IF-THEN que modelan el problema que se quiere resolver. Una regla difusa simple tiene la forma:

“Si u es A entonces v es B ”

Donde A y B son conjuntos difusos definidos en los rangos de “ u ” y “ v ” respectivamente. Una regla expresa un tipo de relación entre los conjuntos A y B cuya función característica sería $\neq_{AB}(x, y)$ y representa lo que conocemos como implicación lógica. La elección apropiada de esta función característica está sujeta a las reglas de la lógica proporcional.

2.2.12.7 Reglas de conocimiento difuso

Dentro de la máquina de inferencia difusa se encuentran preestablecidas las reglas del conocimiento difuso, en el cual se encarga de evaluar las reglas propuestas por el operador. Las reglas del conocimiento difuso conjuntan las entradas y posteriormente generan la salida difusa. A este proceso se le conoce como “Reglas Difusas”, en el se obtiene la salida difusa para poder ser procesada por la defusificación y obtener la salida deseada del sistema.

Las reglas del conocimiento difuso utilizan la operación de intersección entre los grados de pertenencia de cada entrada difusa (A y B), para ser asignada posteriormente en un grado de pertenencia de salida (C), quedando la representación de la regla del conocimiento difuso como:

si A u B entonces C

2.2.12.8 Máquina de inferencia Mamdani

La máquina de inferencia de tipo Mamdani tiene como formato la siguiente regla, si u_1 es A_1 y u_2 es B_1 entonces es C , en donde los valores de entrada u_1 y u_2 provienen de los valores lingüísticos y C representa los términos lingüísticos asociados a los conjuntos difusos.

$$f = (u_1 \text{ y } u_2) \quad \text{Ec. 6}$$

Donde:

f : Grado de pertenencia total

u_1 : Grado de pertenencia 1

u_2 : Grado de pertenencia 2

Cada regla tiene su formato preestablecido por el lado izquierdo que contiene los antecedentes de uno o varios factores y del lado derecho contiene los valores consecuentes.

La ventaja de este método proyecta un ambiente intuitivo ampliamente aceptado y se adapta mejor al lenguaje humano.

2.2.12.9 Máquina de inferencia Sugeno

La máquina de inferencia de tipo Sugeno tiene como formato la siguiente regla, si u_1 es A_1 y u_2 es B_1 entonces es $f(u_1 \circ u_2)$, donde las variables lingüísticas de entrada se asumen que las variables provienen de valores lingüísticos y la salida representa la función lineal de las entradas.

$$f = (u_1 * u_2) \quad \text{Ec. 7}$$

Donde:

f : Grado de pertenencia total

u_1 : Grado de pertenencia 1

u_2 : Grado de pertenencia 2

Para los sistemas que usan este tipo de máquina se les infiere como sistemas TSK en honor a su desarrollador.

La ventaja de este tipo de método genera en términos de computación eficiencia de corrimiento en el programa, optimiza las técnicas adaptativas, garantiza la continuidad en la superficie de salida y se adapta mejor al análisis matemático.

2.2.12.10 Defusificación

El proceso de la defusificación es convertir los valores difusos en variables de salida que resultan del proceso de inferencia, preprocesados por la máquina de inferencia difusa, en donde esta etapa es necesaria para el sistema de Mamdani, ya que el conjunto difuso de salida no es directamente utilizable para dar una información precisa al operador o enviar una

acción de la inferencia difusa al mundo real. Para realizar la defusificación existen varios métodos, para el cálculo de la misma es importante elegir el más adecuado.

2.2.12.11 Método del Centroide.

Conocido como método de centro de gravedad, es el método más usado para la defusificación, su expresión es la siguiente:

$$Z = \frac{\int \mu(u) * u \, dc}{\int \mu(u) \, dc} \quad \text{Ec.8}$$

Donde:

μ : Grado de pertenencia de conjunto difuso

u_i : Universo de entrada 1

u_{i2} : Universo de entrada 2

Z : Valor de centroide

2.2.12.12 Método de Centro de Máximos

Este método se considera el conjunto difuso de la variable de salida y sus valores de pertenencia, de tal modo que la suma de los valores sea igual a uno. Donde la salida real se obtiene con la siguiente expresión:

$$Z = \frac{\sum_{i=1}^n \mu(u) * u_i}{\sum_{i=1}^n \mu(u_i)} \quad \text{Ec.9}$$

Donde:

μ : Grado de pertenencia de conjunto difuso

u_i : Universo de entrada 1

u_{i2} : Universo de entrada 2

Z: Valor de centroide

2.2.13 Controlador Lógico Difuso

El controlador lógico difuso visto desde la perspectiva de la naturaleza inexacta del mundo real es un conjunto de reglas de control lingüístico relacionadas por los conceptos duales de las implicaciones difusas y la regla de inferencia compositiva.

El control lógico difuso (FLC) provee un algoritmo que puede convertir el control lingüístico basado en el conocimiento de un experto, en una estrategia de control automático.

La principal ventaja de aplicación de un control difuso a comparación de los controladores convencionales, son los siguientes:

- El controlador difuso no requiere identificar el sistema, para esto no se necesita información precisa, por lo que se propone emplear la teoría de conjuntos difusos.
- No es necesario estandarizar o alinear el modelo de la aplicación en el sistema embebido.
- Facilita el diseño del controlador.
- No se requiere algún modelo matemático específico.

2.2.14 Redes Neuronales Artificiales

Las redes neuronales artificiales se basan en generalizar información extraída de datos experimentales, tablas bibliográficas o bases de datos, los cuales se determinan por expertos humanos. Dichas redes neuronales toman en cuenta las entradas y como salidas las señales del sistema. Las redes neuronales artificiales son aproximaciones no lineales a la forma en que funciona el cerebro; por lo tanto, no deben compararse directamente con el cerebro ni confundir los principios que fundamentan el funcionamiento de las redes neuronales artificiales y del cerebro, ni pensar que las redes neuronales se basan únicamente en las redes biológicas ya que sólo emulan en una parte muy simple el funcionamiento del cerebro humano. Además, se debe considerar que las redes biológicas son generadoras de procesos neurobiológicos en que se establecen relaciones de complejidad muy alta, las cuales no se pueden lograr con redes monocapas ni con redes multicapas.

Las neuronas artificiales simples fueron introducidas por McCulloch y Pitts en 1943. Una red neuronal se caracteriza por los siguientes elementos:

1. Un conjunto de unidades de procesamiento o neuronas.
2. Un estado de activación para cada unidad, equivalente a la salida de la unidad.
3. Conexiones entre las unidades, generalmente definidas por un peso que determina el efecto de una señal de entrada en la unidad.
4. Una regla de propagación, que determina la entrada efectiva de una unidad a partir de las entradas externas.
5. Una función de activación que actualiza el nuevo nivel de activación basándose en la entrada efectiva y la activación anterior.
6. Una entrada externa que corresponde a un término como bias para cada unidad.

7. Un método para reunir la información, correspondiente a la regla del aprendizaje.
8. Un ambiente en el que el sistema va a operar, con señales de entrada e incluso señales de error.

En muchas redes las unidades de proceso tienen respuesta de la forma:

$$y = f(\sum_k \omega_k x_k) \quad \text{Ec.10}$$

Donde:

- x_k : señales de salida de otros nodos o entradas externas.
- ω_k : pesos de las ligas de conexión.
- $f(\cdot)$: función no lineal simple

La función puede ser sigmoideal, tangente hiperbólica, escalón, entre otras. En MATLAB® se tiene diferentes funciones de activación como tansig, hardlim y purelin, entre otras, lo cual facilita las aproximaciones que se requieran hacer, empleando RNA.

Cada unidad de proceso tiene una tarea simple: recibe la entrada de otras unidades o de fuentes externas y procesa la información para obtener una salida que se propaga a otras unidades. Una red puede tener una estructura arbitraria, pero las capas que contienen estas estructuras están definidas de acuerdo con su ubicación en la topología de la red neuronal. Las entradas externas son aplicadas en la primera capa, y las salidas se consideran la última capa. Las capas internas que no se observan como entradas o salidas se denominan capas ocultas. Por convención, las entradas no se consideran como capa porque no realizan procesamiento.

2.2.14.1 Aplicaciones de las Redes Neuronales Artificiales (RNA)

Algunas áreas donde se aplican las RNA son:

- Automóviles: Sistemas de piloto automático. Detección de fallas por reconocimiento externo de vibraciones.
- Bancos: Lectura de cheques y otros documentos. Evaluación de aplicaciones de créditos.
- Electrónica: Predicción de secuencia de códigos. Distribución de elementos en CI. Control de procesos. Análisis de fallas. Visión artificial. Reconocimiento de voz.

- Finanzas: Tasación real de los bienes. Asesoría de préstamos. Previsión en la evolución de precios. Seguimiento de hipotecas. Análisis de uso de línea de crédito. Evaluación del riesgo en créditos. Identificación de falsificaciones. Interpretación y reconocimiento de firmas.
- Manufactura: Control de la producción y del proceso. Análisis y diseño de productos. Diagnóstico de fallas en el proceso y maquinarias. Identificación de partículas en tiempo real. Inspección de calidad mediante sistemas visuales. Análisis de mantenimiento de máquinas.
- Medicina: Análisis de células portadoras de cáncer mamario. Análisis de electroencefalograma y de electrocardiograma. Reconocimiento de infartos mediante ECG. Diseño de prótesis. Optimización en tiempos de trasplante. Reducción de gastos hospitalarios.
- Robótica: Control dinámico de trayectoria. Robots elevadores. Controladores. Sistemas ópticos.
- Seguridad: Códigos de seguridad adaptativos. Criptografía. Reconocimiento de huellas digitales.
- Telecomunicaciones: Compresión de datos e imágenes. Automatización de servicios de información. Traducción en tiempo real de lenguaje hablado.
- Transporte: Diagnóstico de frenos en camiones. Sistemas de ruteo y seguimiento de flotas.
- Voz: Reconocimiento de voz. Compresión de voz. Clasificación de vocales. Transformación de texto escrito a voz.

2.2.14.2 Entrenamiento de las redes neuronales.

Se denomina entrenamiento al proceso de configuración de una red neuronal para que las entradas produzcan las salidas deseadas a través del fortalecimiento de las conexiones. Una forma de llevar esto a cabo es a partir del establecimiento de pesos conocidos con anterioridad, y otro método implica el uso de técnicas de retroalimentación y patrones de aprendizaje que cambian los pesos hasta encontrar los adecuados.

Además del aprendizaje puede dividirse en supervisado o asociado y no supervisado o autorganizado.

2.2.14.3 Redes de una capa.

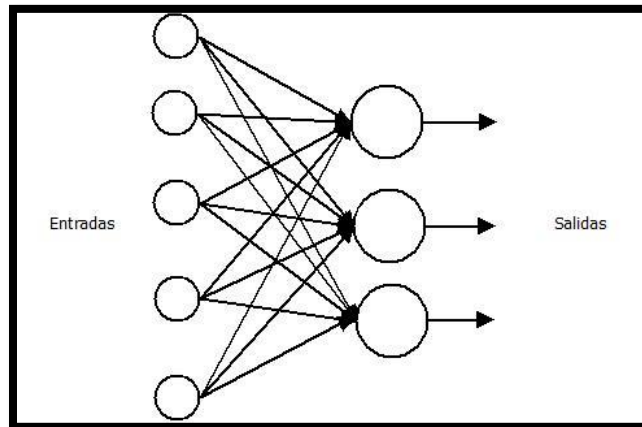


Figura 2.11 Red de una capa.

Las redes de una capa, cómo se muestra en la Figura 2.11 presentan las entradas directamente conectados a las salidas mediante pesos. Las salidas no interactúan, por lo que una red con n salidas puede ser analizada por n redes separadas. Cada unidad cómo se muestra en la Figura 2.12 dos produce salidas de forma:

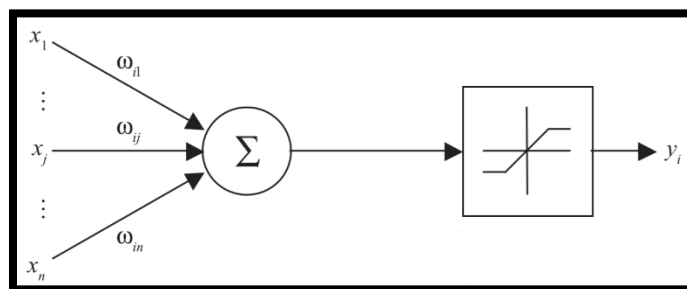


Figura 2.12 Función del nodo.

En notación vectorial se expresa cómo:

$$y(x) = f(\omega^T x)$$

Los puntos de x con una suma constante $\sum \omega_j x$ Define un hiperplano perpendicular al vector ω . la orientación del hiperplano se determina por la dirección de ω , dependiendo de los pesos, pero no de la magnitud de ω .

La suma $\sum \omega_j x = 0$ define un hiperplano que cruza el origen. La introducción de un término teta denominado bias en la suma del hiperplano tiene mayor movilidad y permiten la separación de algunos conjuntos de datos, con lo que se obtienen mejores clasificaciones.

2.2.14.4 Perceptrón.

Los perceptrones de una capa oculta pueden clasificarse correctamente en los conjuntos de datos que son linealmente separables, esto es, que pueden ser separados por el hiperplano en dos dimensiones y puede verse como una línea recta que separa dos clases; como se muestra en la Figura 2.13.

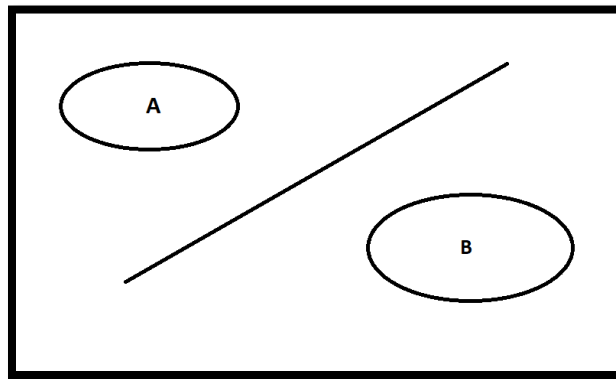


Figura 2.13 Separación lineal de los conjuntos A y B.

El algoritmo de aprendizaje del perceptrón funciona para aprender funciones binarias linealmente separables ya que de otra manera el algoritmo no convergería ni produciría la mejor solución, debido a que las salidas son binarias se emplean unidades lineales de umbral. Cada unidad calcula la suma con pesos de las entradas “ x ”, generando una salida binaria “ y ”.

CAPITULO 3

3. Metodología

A continuación, se muestra la secuencia de cada uno de los procesos necesarios para la aplicación de la inteligencia artificial dentro del controlador, como se muestra en la Figura 3.1.

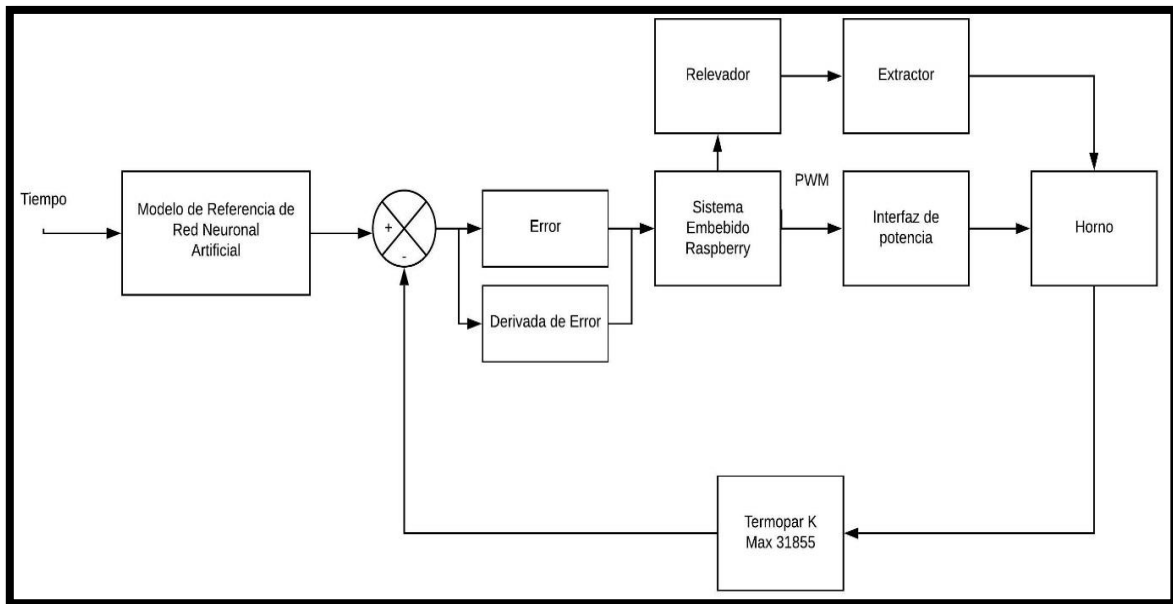


Figura 3.1 Sistema completo aplicado

Con el diagrama mostrado se observa cada uno de los aspectos desarrollados en esta investigación, se presenta el proceso para la implementación y óptimo funcionamiento del horno de reflujo.

3.1. Obtención de datos

Principalmente se recabó la información de los datos de la investigación de L. Kwang [4], como se muestra en el anexo A. Al procesar estos datos se obtuvo la gráfica de la curva característica del perfil de temperatura para el soldado de dispositivos de montaje superficial como se muestra en la Figura 3.2.

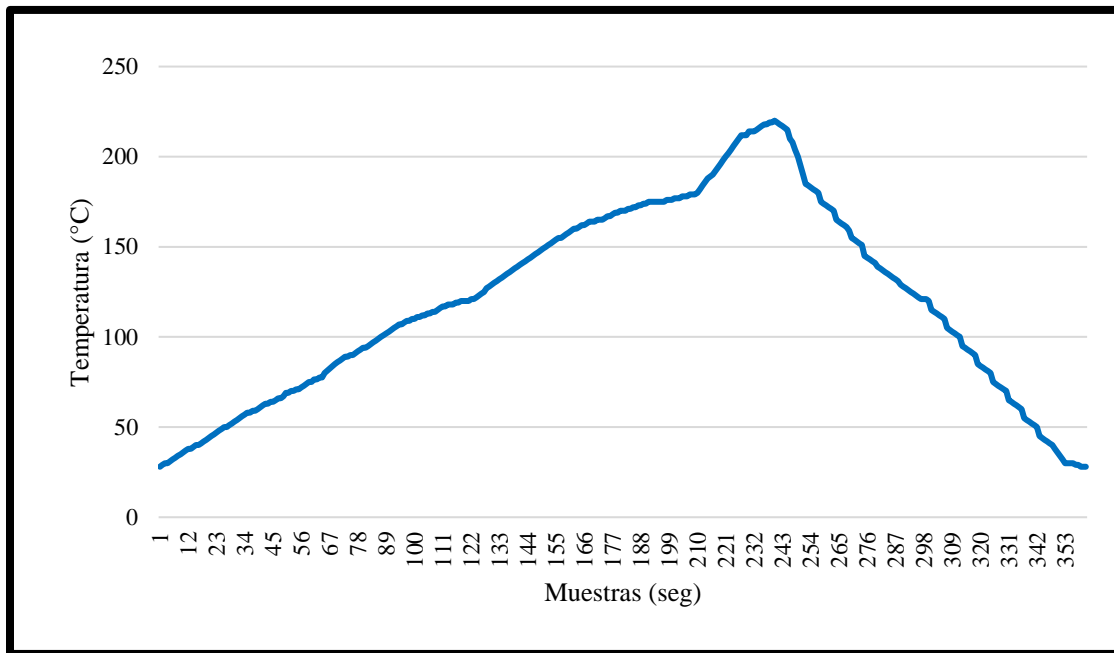


Figura 3.2 Curva característica de perfil de temperatura.

3.2 Optimización de datos obtenidos

Con el fin de sustituir la tabla de datos, se copia la información recabada en la red neuronal artificial con retropropagación. Principalmente se configura la red neuronal para tener un modelo de datos que sirven de referencia, para el seguimiento de temperatura en el horno.

La arquitectura de la RNA utilizada consta de:

- Número de neuronas en la capa oculta: 100.
- Número de neuronas en la capa de salida: 1.
- Algoritmo de entrenamiento: Levenberg- Maquardt.
- Función de activación de la capa oculta : Sigmoidal.
- Función de activación de la capa de salida: Elliot Sigma.
- Parámetros de muestra entrenamiento: 50.
- Número de iteraciones de entrenamiento: 2000.
- División de datos: Aleatorio.
- Criterio de optimización: Error medio cuadrático.

Al introducir toda esta información dentro de la red neuronal, se obtiene en el software Matlab el diagrama de la RNA, como se muestra en la figura 3.3.

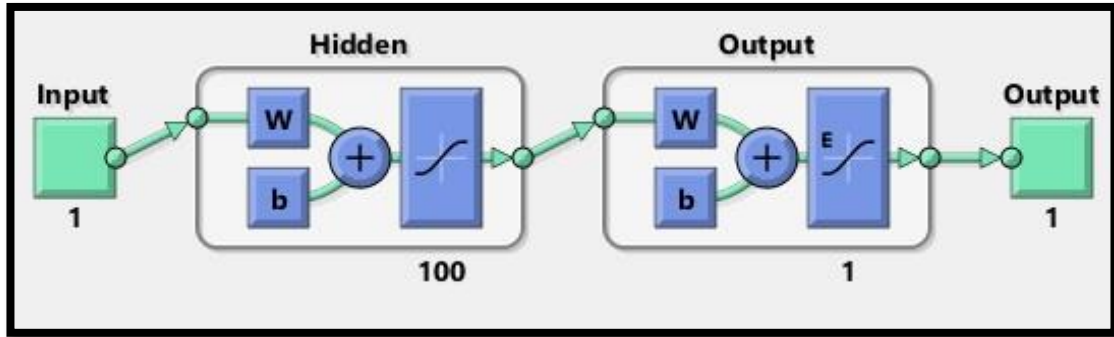


Figura 3.3 Diagrama de la red neuronal artificial.

Los datos de referencia de la temperatura se obtuvieron de la investigación de L.Kwang [4], al utilizar la tabla del anexo A, en el entrenamiento de la red neuronal, como se muestra en la Figura 3.4.

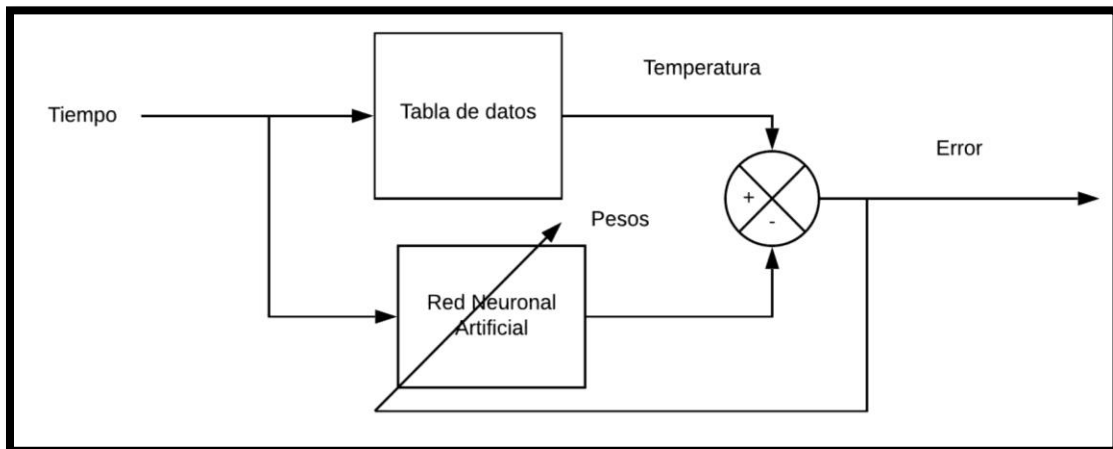


Figura 3.4 Diagrama a bloques de optimización de datos mediante RNA.

Una vez entrenada la red neuronal artificial, se copian los datos de la tabla del Anexo A, y al comparar los valores proporcionados los de la tabla de salida, con esto se obtuvo el error mostrado en la Figura 3.5.

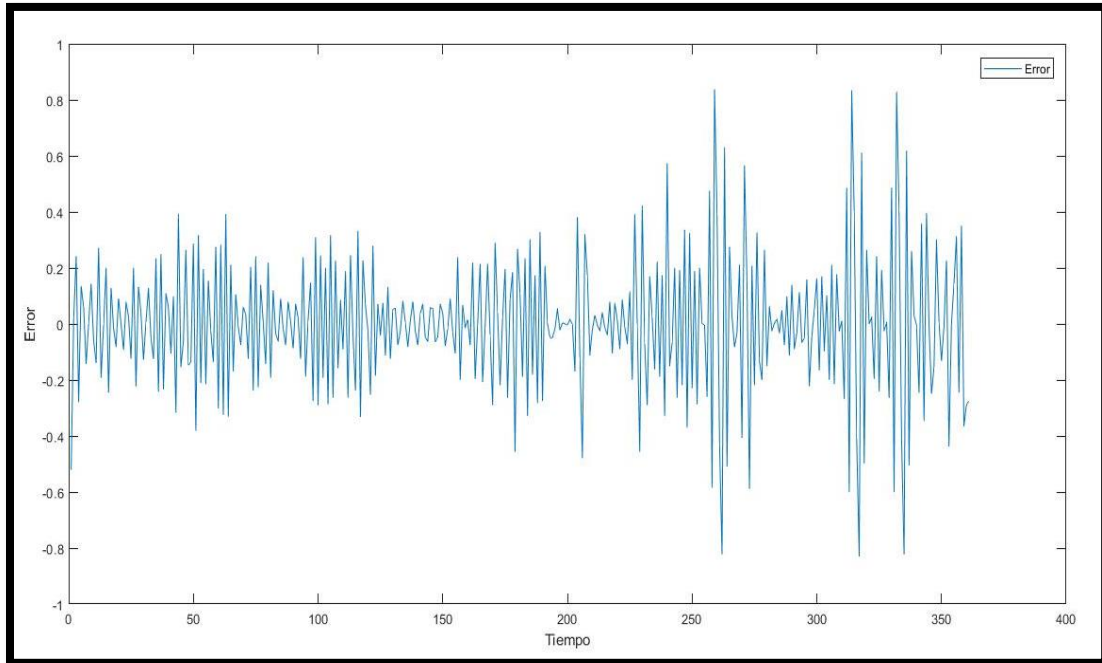


Figura 3.5 Error de procesamiento RNA

Al entrenar la red neuronal, mediante 2000 muestras se obtienen los resultados mostrados en la Figura 3.6, para 360 datos de entrada.

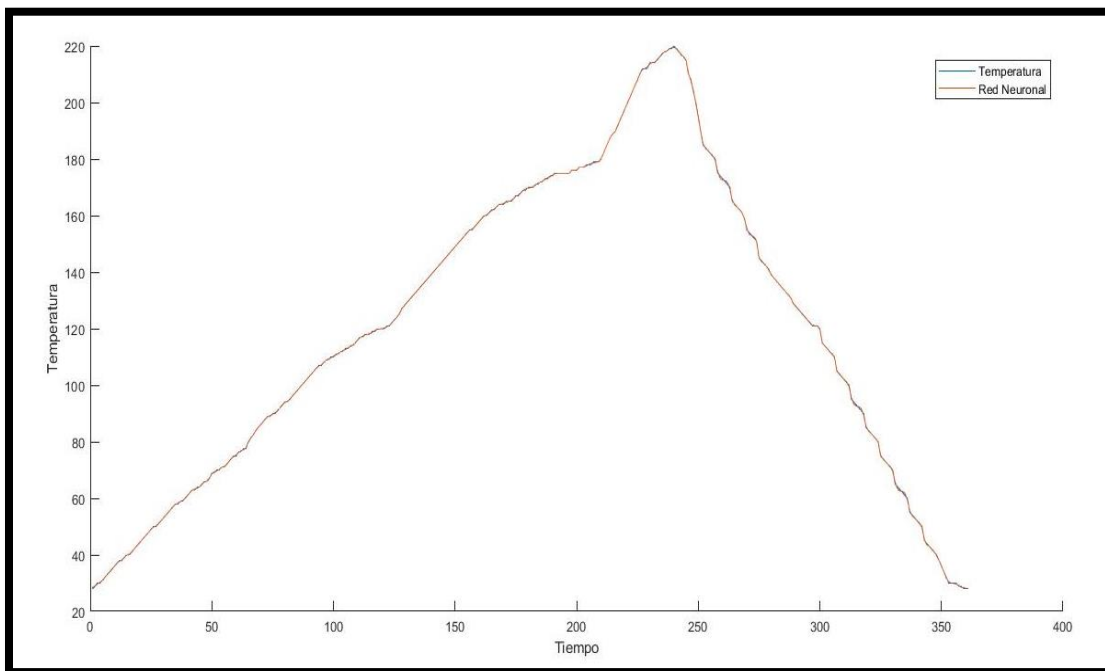


Figura 3.6 Datos procesados en RNA.

Al tener todos estos datos procesados se ingresarán posteriormente dentro del programa para el controlador inteligente.

3.3 Selección de sistema embebido para el controlador inteligente

Con el fin de contar con un sistema embebido apropiado para implementar el controlador difuso, se toman los siguientes tres dispositivos embebidos, en los cuales serán ingresados los datos para evaluar la eficiencia de procesamiento en la lógica seleccionada para el controlador inteligente.

1. Arduino Mega 2560.
2. Tiva C TM4C123GXL.
3. Raspberry Pi 3 B.

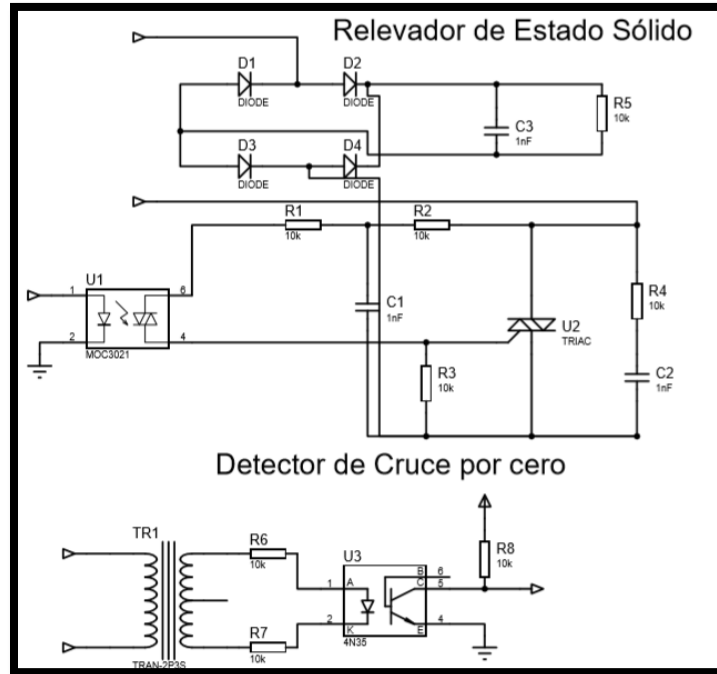
3.4 Comportamiento y evaluación de los sistemas embebidos.

A los sistemas embebidos se les aplica el código de programa correspondiente mostrado en el anexo C para las placas de Arduino y Tiva C, en tanto el anexo D es para la placa Raspberry.

Los sistemas se ajustan respectivamente a un nivel fijo de temperatura de 100°C en la etapa de precalentamiento para evaluar el comportamiento de cada uno de los sistemas del control propuestos.

En cada sistema se genera una salida de control por PWM que activará la etapa de potencia para la resistencia de caldeo, a través de un Triac, el cual suministrará la corriente eléctrica alterna (AC) de 120 V, necesaria para realizar el proceso de caldeo.

En la Figura 3.7 se muestra el diagrama eléctrico para el suministro de potencia en el control de las resistencias internas del horno, así como el detector de cruce por cero para adquirir la información de la corriente eléctrica alterna y traducir la información en los sistemas embebidos.



3.7 Diagrama eléctrico de potencia.

En la Figura 3.8 se muestra el control de temperatura obtenido de los sistemas embebidos.

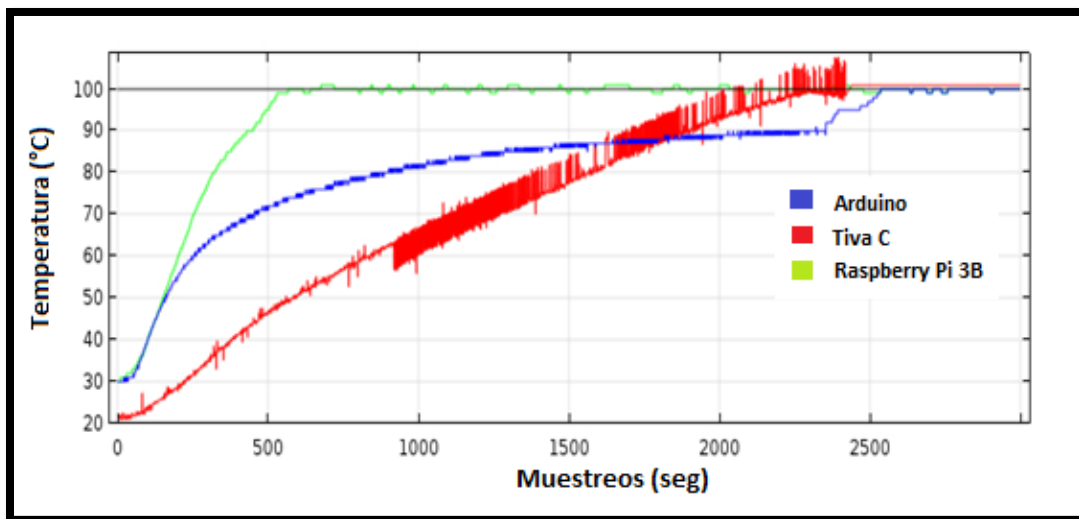


Figura 3.8 Comportamiento de los sistemas embebidos.

- El sistema Arduino se muestra en color azul.
- El sistema Tiva C se muestra en color rojo.
- El sistema Raspberry Pi 3 B se muestra en color verde.

El muestreo de la gráfica se realizó en segundos, con la respuesta obtenida de los tres sistemas ajustados para la optimización de la curva de precalentamiento a 100°C, se denota lo siguiente, el sistema embebido Arduino logra alcanzar la temperatura de 100°C apartir de las 2600 segundos, en tanto el sistema Tiva C logra llegar en un lapso de 2000 segundos, causando una señal de ruido indeseable en la lectura del sensor, en tanto que el sistema Raspberry logra llegar en un lapso de 500 segundos.

Debido a la eficiencia y precisión de control mostrado de los diferentes sistemas embebidos, la elección obvia es el sistema embebido Raspberry Pi 3 B.

3.5 Implementación de la lógica difusa al controlador inteligente

Como se menciona la elección del sistema embebido Raspberry, se plantea la implementación del control inteligente, que se muestra en la Figura 3.9.

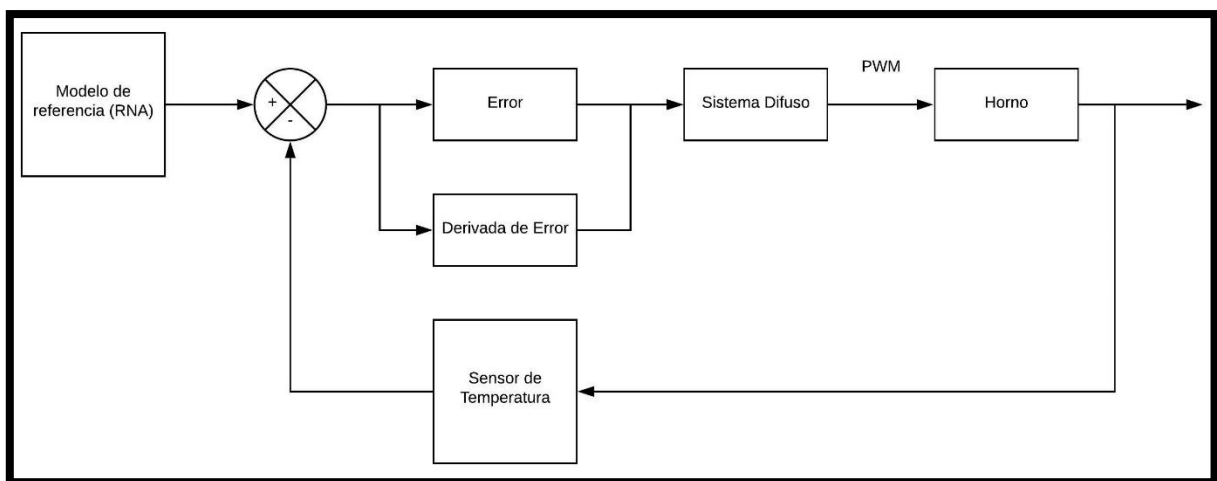


Figura 3.9 Diagrama de implementación en Raspberry.

3.6 Instalación de Raspbian

Se comienza con la instalación del sistema operativo Raspbian, con la imagen obtenida de la página oficial <https://www.raspberrypi.org/downloads/raspbian/> como se muestra en la Figura 3.10.

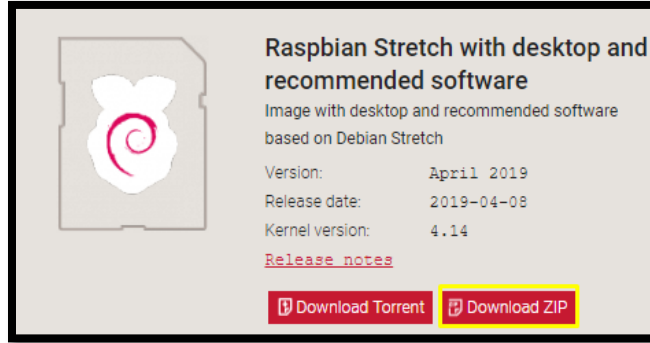


Figura 3.10 Sistema operativo Raspbian

Al obtener el archivo que contiene el sistema operativo Raspbian, se procede a formatear la SD, por lo que se recomienda formatear en un tamaño de almacenamiento de 16 Gb como mínimo, como se muestra en la Figura 3.11.

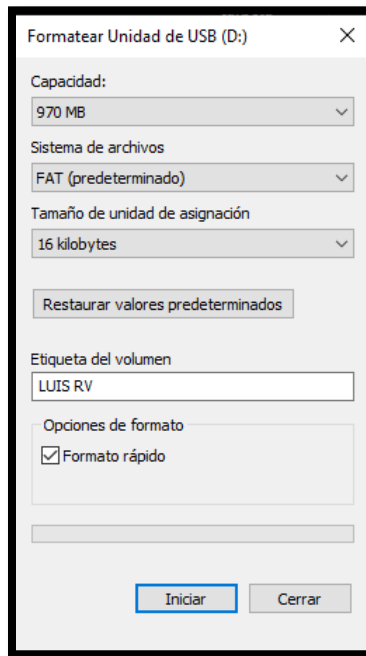


Figura 3.11 Formateo de memoria microSD

Al tener el formateo de la memoria y la imagen del sistema operativo Raspbian, se graba el archivo con terminación .IMG del sistema operativo en la SD, se necesita el programa “Etcher”, para acceder y convertir el sistema operativo en ejecutable para la placa de Raspberry Pi 3 B. Una vez contenido el programa se selecciona la imagen del sistema operativo y se graba el programa dentro de la SD, como se muestra en la Figura 3.12.



Figura 3.12 Programa Etcher

Al tener finalizada la conversión, se retira la memoria SD de la computadora y se procede a insertarla en el equipo Raspberry Pi 3B para el primer arranque. Para el arranque del sistema operativo Raspbian se procede a completar la guía de inicio rápido dentro de la Raspberry Pi 3 B.

3.6.1 Inicio de Raspberry con Raspbian

Una vez realizado el primer arranque, se dirige el inicio del sistema operativo Raspbian, como se muestra en la Figura 3.13.



Figura 3.13 Inicio de Raspbian en Raspberry Pi 3 B

Con el inicio de Raspbian, Raspberry solicita al usuario completar las configuraciones generales de selección de país, elección de zona horaria, contraseña principal del dispositivo, conexión de red inalámbrica y la principal actualización del software a la fecha de la instalación, como se muestra en la Figura 3.14.

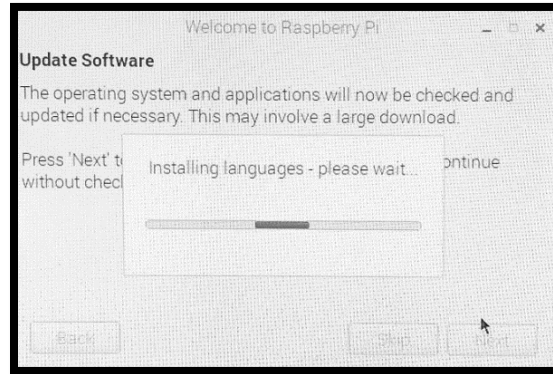


Figura 3.14 Actualización de datos en la placa Raspberry

3.6.2 Elección de lenguaje

Se elije el lenguaje de programación Python 3.5 para el sistema embebido Raspberry Pi 3 B, que es un sistema de desarrollo de software libre para la aplicación del controlador inteligente difuso en el control de temperatura del horno de reflujo. Actualmente existen varias librerías adaptables en Python que se caracterizan por la mejora del Aprendizaje Máquina (Machine Learning).

3.6.3 Instalación de Librerías

Principalmente se seleccionan las librerías elementales para el funcionamiento de la placa Raspberry Pi 3 B con lógica difusa para el controlador inteligente, siendo las siguientes:

- Matplotlib.
- Scikit-Learn.
- Python 3 Scipy.

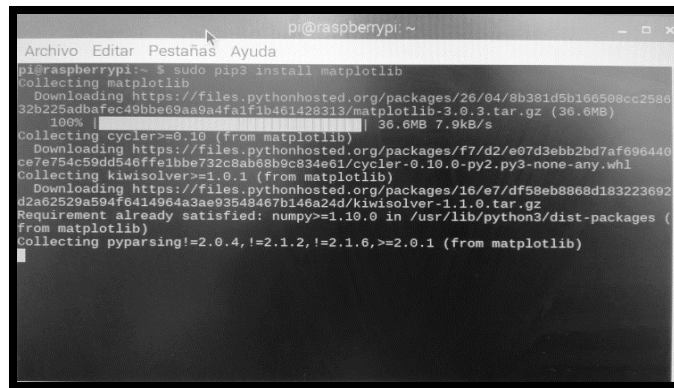
3.6.3.1 Librería Matplotlib

Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en arrays o listas de datos. La librería Matplotlib se ejecuta con la siguiente línea:

```
sudo pip3 install matplotlib
```

En esta línea permitirá a Python, ejecutar el programa y actualizar el entorno para poder manejar el paquete con el archivo numpy, que viene integrado dentro de esta librería, como

se muestra en la Figura 3.15. ya que numpy es una extensión de Python y que agrega soporte para vectores y matrices.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo pip3 install matplotlib  
Collecting matplotlib  
  Downloading https://files.pythonhosted.org/packages/26/04/8b381d5b166508cc258632b225adbafec49bbe69aa9a4fa1f1b461428313/matplotlib-3.0.3.tar.gz (36.6MB)  
    100% |#####| 36.6MB 7.9kB/s  
Collecting cycler>=0.10 (from matplotlib)  
  Downloading https://files.pythonhosted.org/packages/f7/d2/e07d3ebb2bd7af696440ce7e754c59dd546ffe1bbe732c8ab68b9c834e61/cyclor-0.10.0-py2.py3-none-any.whl  
Collecting kiwisolver>=1.0.1 (from matplotlib)  
  Downloading https://files.pythonhosted.org/packages/16/e7/df58eb8868d183223692d2ae2b29a594f6414964a3ae93548467b146a24d/kiwisolver-1.1.0.tar.gz  
Requirement already satisfied: numpy>=1.10.0 in /usr/lib/python3/dist-packages (from matplotlib)  
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 (from matplotlib)
```

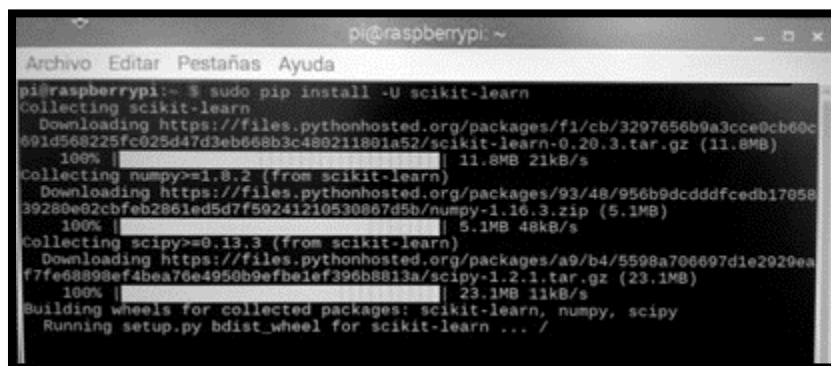
Figura 3.15 Instalación de librería matplotlib.

3.6.3.2 Librería Scikit-Learn

Al tener instalada la librería Matplotlib en la placa de Raspberry Pi 3 B, se procede a instalar la librería Scikit-Learn. Es la librería principal encargada del mejoramiento del procesamiento de Machine Learning, sus funciones permiten implementar la lógica difusa. Para la instalación de esta librería se necesitó la siguiente línea de código:

```
sudo pip install scikit-learn
```

Con esta línea de código se añadirá a Python la librería para la utilización del aprendizaje máquina, como se muestra en la Figura 3.16.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo pip install -U scikit-learn  
Collecting scikit-learn  
  Downloading https://files.pythonhosted.org/packages/f1/cb/3297656b9a3cce0cb60c691d560225fc025d47d3eb668b3c480211801a52/scikit-learn-0.20.3.tar.gz (11.8MB)  
    100% |#####| 11.8MB 21kB/s  
Collecting numpy>=1.8.2 (from scikit-learn)  
  Downloading https://files.pythonhosted.org/packages/93/48/956b9dcdddfcedb1705839280e02cbfeb2861ed5d7f59241210530867d5b/numpy-1.16.3.zip (5.1MB)  
    100% |#####| 5.1MB 48kB/s  
Collecting scipy>=0.13.3 (from scikit-learn)  
  Downloading https://files.pythonhosted.org/packages/a9/b4/5508a706697d1e2920eaf77fe68898ef4bea76e4950b9efbe1ef396b8813a/scipy-1.2.1.tar.gz (23.1MB)  
    100% |#####| 23.1MB 11kB/s  
Building wheels for collected packages: scikit-learn, numpy, scipy  
  Running setup.py bdist_wheel for scikit-learn ... /
```

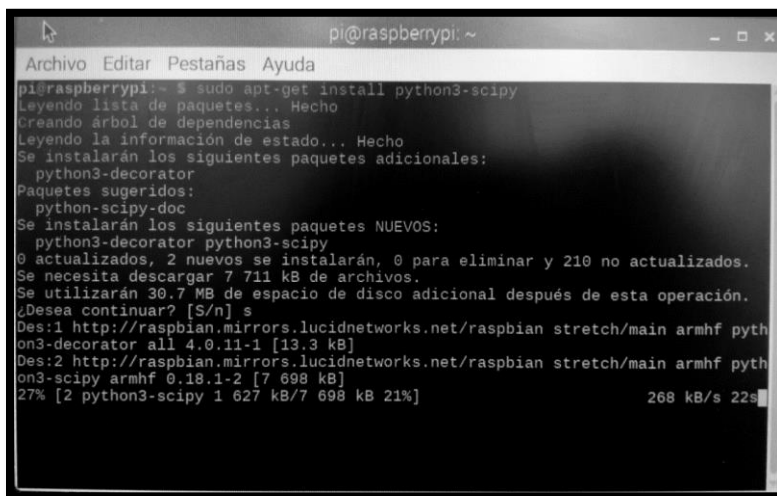
Figura 3.16 Instalación de Scikit-Learn

3.6.3.3 Librería Python 3 Scipy

La librería Scipy se ejecuta con la siguiente línea de código:

```
sudo apt-get install python3-scipy
```

Con la instalación de esta librería Raspberry se encarga de realizar cálculos computacionales complejos tanto lógicos como matemáticos. Scipy se adapta en el entorno de Python, como se muestra en la Figura 3.17.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi1:~$ sudo apt-get install python3-scipy  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
python3-decorator  
Paquetes sugeridos:  
python-scipy-doc  
Se instalarán los siguientes paquetes NUEVOS:  
python3-decorator python3-scipy  
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 210 no actualizados.  
Se necesita descargar 7 711 kB de archivos.  
Se utilizarán 30.7 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] s  
Des:1 http://raspbian.mirrors.lucidnetworks.net/raspbian stretch/main armhf pyth  
on3-decorator all 4.0.11-1 [13.3 kB]  
Des:2 http://raspbian.mirrors.lucidnetworks.net/raspbian stretch/main armhf pyth  
on3-scipy armhf 0.18.1-2 [7 698 kB]  
27% [2 python3-scipy 1 627 kB/7 698 kB 21%] 268 kB/s 22s
```

Figura 3.17 Instalación de librería Scipy

Al tener instalado el módulo del programa de Scipy en el sistema embebido Raspberry se puede iniciar la programación del controlador inteligente.

3.6.4 Parámetros de variables difusas en el controlador

Se designa las variables de entrada como Error de Temperatura y derivada del error. En la entrada Error se establecen los conjuntos difusos Mínimo, Muy bajo, Bajo, Medio, Alto, Muy alto y Máximo. Para la derivada de error se establecen los conjuntos difusos como Negativo, Cero y Positivo quedando así definidos los universos como se muestran en las siguientes Figuras 3.18 y 3.19.

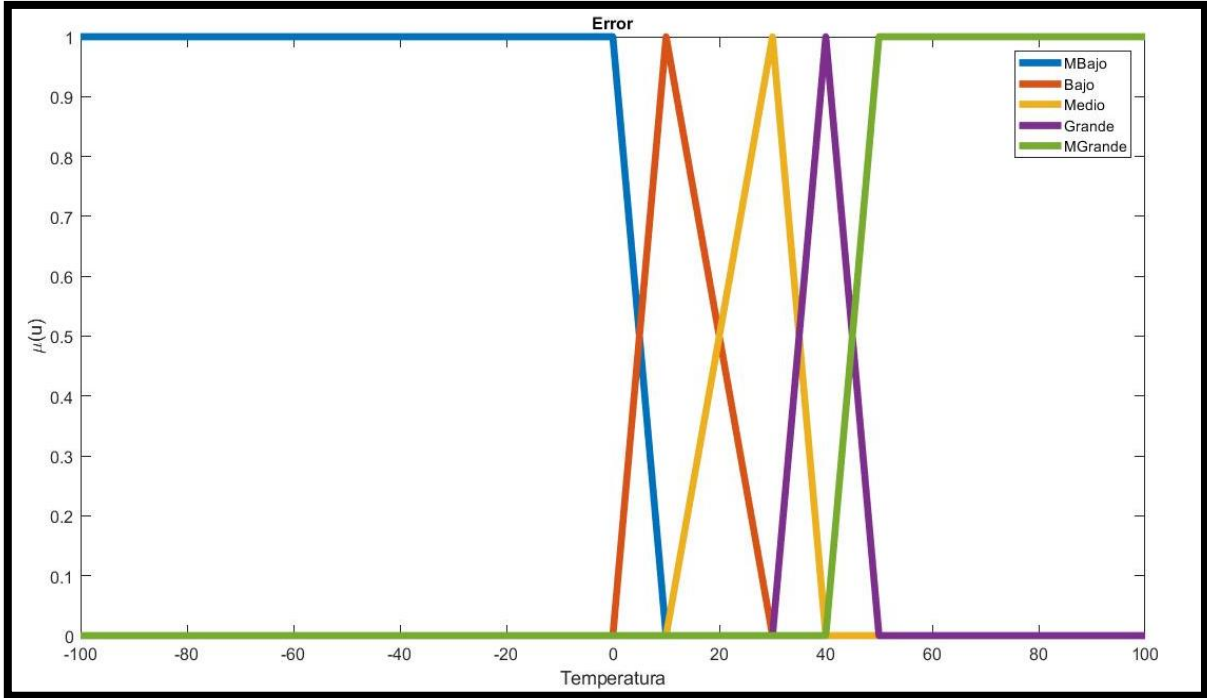


Figura 3.18 Universo error de temperatura.

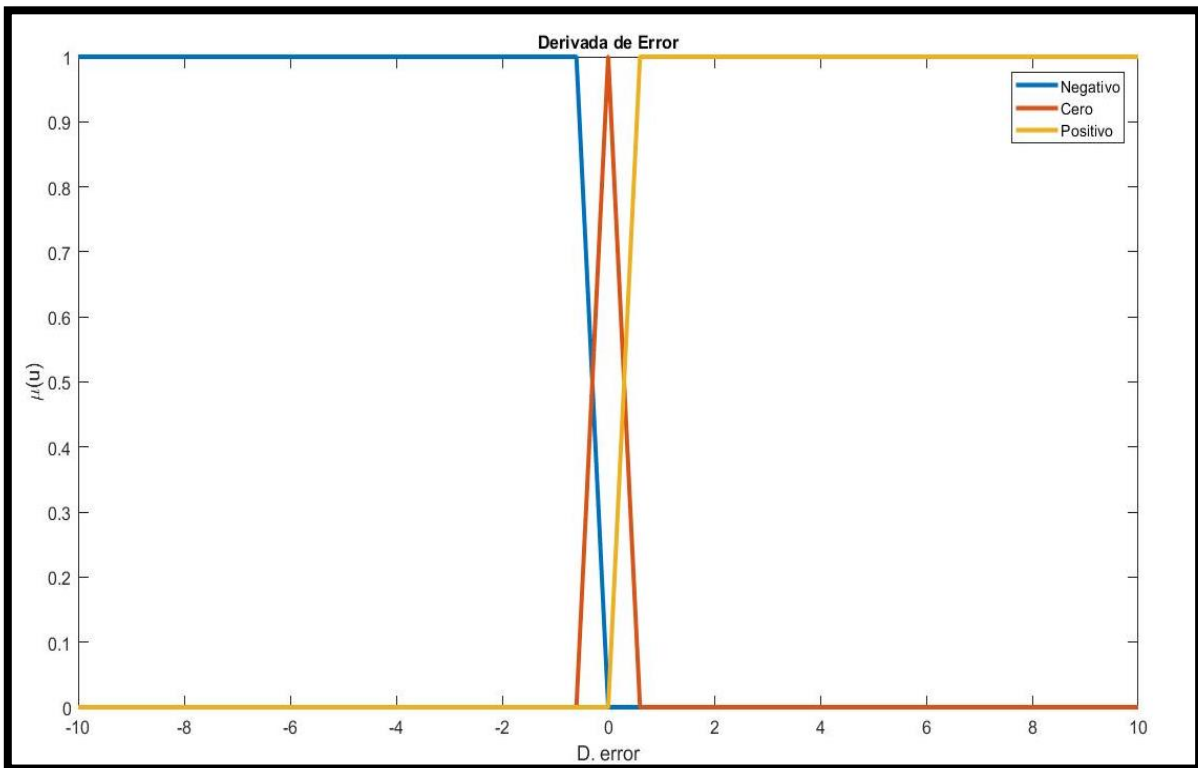


Figura 3.19 Universo derivada error de temperatura

El universo error se describe matemáticamente mediante la ecuación 11:

$$\mu(\text{Error m. bajo})(x) = \begin{cases} y = 1 & \text{si } x \leq 0 \\ y = \frac{B-x}{B-m} \\ y = 0 & \text{si } x \geq 15 \end{cases}$$

$$\mu(\text{Error bajo})(x) = \begin{cases} y = 0 & \text{si } x \leq 0 \\ y = \frac{x-a}{m-a} \\ y = \frac{B-x}{B-m} \\ y = 0 & \text{si } x \geq 30 \end{cases}$$

$$\mu(\text{Error medio})(x) = \begin{cases} y = 0 & \text{si } x \leq 15 \\ y = \frac{x-a}{m-a} \\ y = \frac{B-x}{B-m} \\ y = 0 & \text{si } x \geq 40 \end{cases}$$

$$\mu(\text{Error grande})(x) = \begin{cases} y = 0 & \text{si } x \leq 30 \\ y = \frac{x-a}{m-a} \\ y = \frac{B-x}{B-m} \\ y = 0 & \text{si } x \geq 50 \end{cases}$$

$$\mu(\text{Error muy grande})(x) = \begin{cases} y = 0 & \text{si } x \leq 30 \\ y = \frac{x-a}{m-a} \\ y = 1 & \text{si } x \geq 5 \end{cases}$$

Ec. 11

Con el universo Derivada del error se obtiene la ecuación 12:

$$\mu(\text{Negativo})(x) = \begin{cases} y = 1 \text{ si } x \leq -1 \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 0 \end{cases}$$

$$\mu(\text{Zero})(x) = \begin{cases} y = 0 \text{ si } x \leq -1 \\ y = \frac{x - a}{m - a} \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 1 \end{cases}$$

$$\mu(\text{Positivo})(x) = \begin{cases} y = 0 \text{ si } x \leq 0 \\ y = \frac{x - a}{m - a} \\ y = 1 \text{ si } x \geq 1 \end{cases}$$

Ec.12

Con base a los conjuntos establecidos, se generaron las siguientes reglas mostradas en la Tabla 3.1.

Tabla 3.1 Reglas difusas

	Temperatura (Error)		Temperatura (dError)		PWM
IF	MBajo	AND	Neg	THEN	MBajo
IF	MBajo	AND	Cero	THEN	Mínimo
IF	MBajo	AND	Positivo	THEN	MBajo
IF	Bajo	AND	Neg	THEN	Medio
IF	Bajo	AND	Cero	THEN	Bajo
IF	Bajo	AND	Positivo	THEN	MBajo
IF	Medio	AND	Neg	THEN	Alto
IF	Medio	AND	Cero	THEN	Medio
IF	Medio	AND	Positivo	THEN	Medio
IF	Grande	AND	Neg	THEN	Máximo
IF	Grande	AND	Cero	THEN	Alto
IF	Grande	AND	Positivo	THEN	Alto
IF	MGrande	AND	Neg	THEN	Máximo
IF	MGrande	AND	Cero	THEN	Máximo
IF	MGrande	AND	Positivo	THEN	MAlto

Con estas reglas difusas se obtienen las reglas de inferencia o memoria difusa, destinado para la aplicación en la programación Tabla 3.2.

Tabla 3.2 Inferencia o memoria difusa

Temperatura (Error)	Temperatura (dError)			
	Potencia	Neg	Cero	Positivo
MBajo	MBajo	Mínimo	Apagado	
Bajo	Medio	Bajo	MBajo	
Medio	Alto	Medio	Medio	
Grande	Máximo	MAlto	Alto	
MGrande	Máximo	Máximo	MAlto	

Una vez que se fusifican los datos de entrada y se aplican las reglas correspondientes se requiere calcular la salida del sistema difuso, para lo cual se utiliza el universo de salida del ciclo de trabajo PWM, que se muestra en la Figura 3.20.

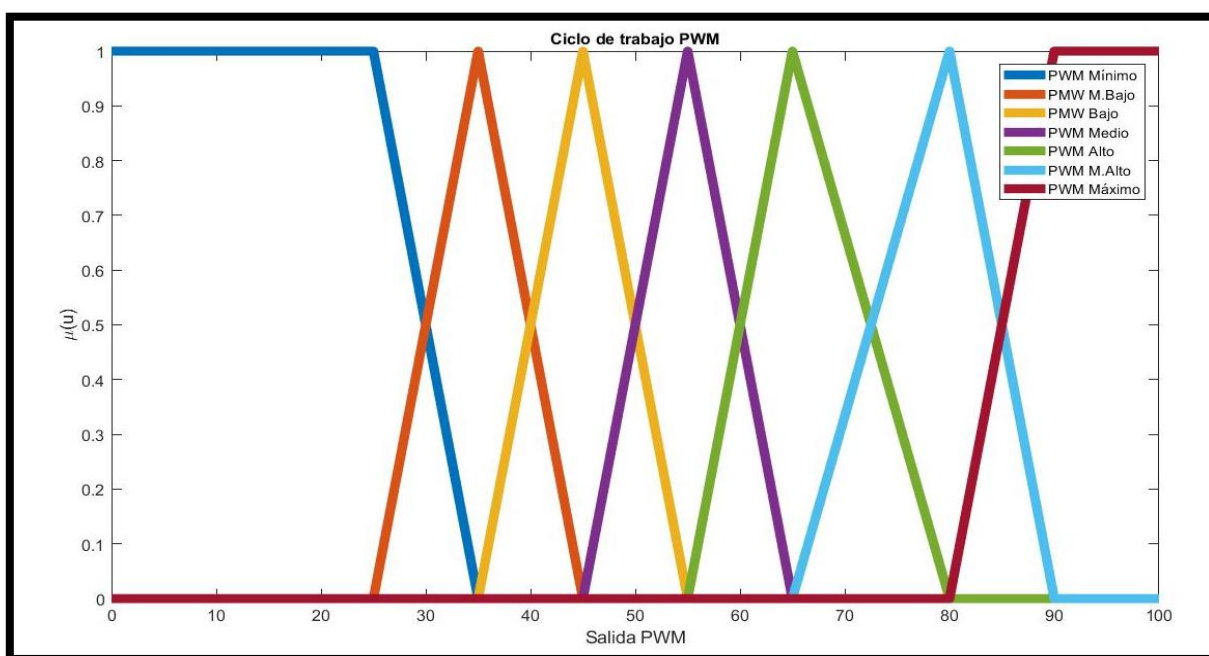


Figura 3.20 Universo de salida de ciclo de trabajo PWM.

Los conjuntos de salida se describen mediante la ecuación 13.

$$\mu(PWM \text{ mínimo})(x) = \begin{cases} y = 1 & \text{si } x \leq 25 \\ y = \frac{B - x}{B - m} & \\ y = 0 & \text{si } x \geq 35 \end{cases}$$

$$\mu(PWM \text{ m. bajo})(x) = \begin{cases} y = 0 \text{ si } x \leq 25 \\ y = \frac{x - a}{m - a} \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 45 \end{cases}$$

$$\mu(PWM \text{ bajo})(x) = \begin{cases} y = 0 \text{ si } x \leq 35 \\ y = \frac{x - a}{m - a} \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 55 \end{cases}$$

$$\mu(PWM \text{ medio})(x) = \begin{cases} y = 0 \text{ si } x \leq 45 \\ y = \frac{x - a}{m - a} \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 65 \end{cases}$$

$$\mu(PWM \text{ alto})(x) = \begin{cases} y = 0 \text{ si } x \leq 55 \\ y = \frac{x - a}{m - a} \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 80 \end{cases}$$

$$\mu(PWM \text{ m. alto})(x) = \begin{cases} y = 0 \text{ si } x \leq 65 \\ y = \frac{x - a}{m - a} \\ y = \frac{B - x}{B - m} \\ y = 0 \text{ si } x \geq 90 \end{cases}$$

$$\mu(PWM \text{ máximo})(x) = \begin{cases} y = 0 \text{ si } x \leq 85 \\ y = \frac{x - a}{m - a} \\ y = 1 \text{ si } x \geq 90 \end{cases} \quad \text{Ec. 13}$$

En la defusificación, se utiliza el método de Mamdani para el proceso de datos y el método del centroide para el cálculo de la salida del control inteligente.

3.7 Programación

El programa de control de temperatura implementado se realizó siguiendo el diagrama de flujo mostrado en la Figura 3.21.

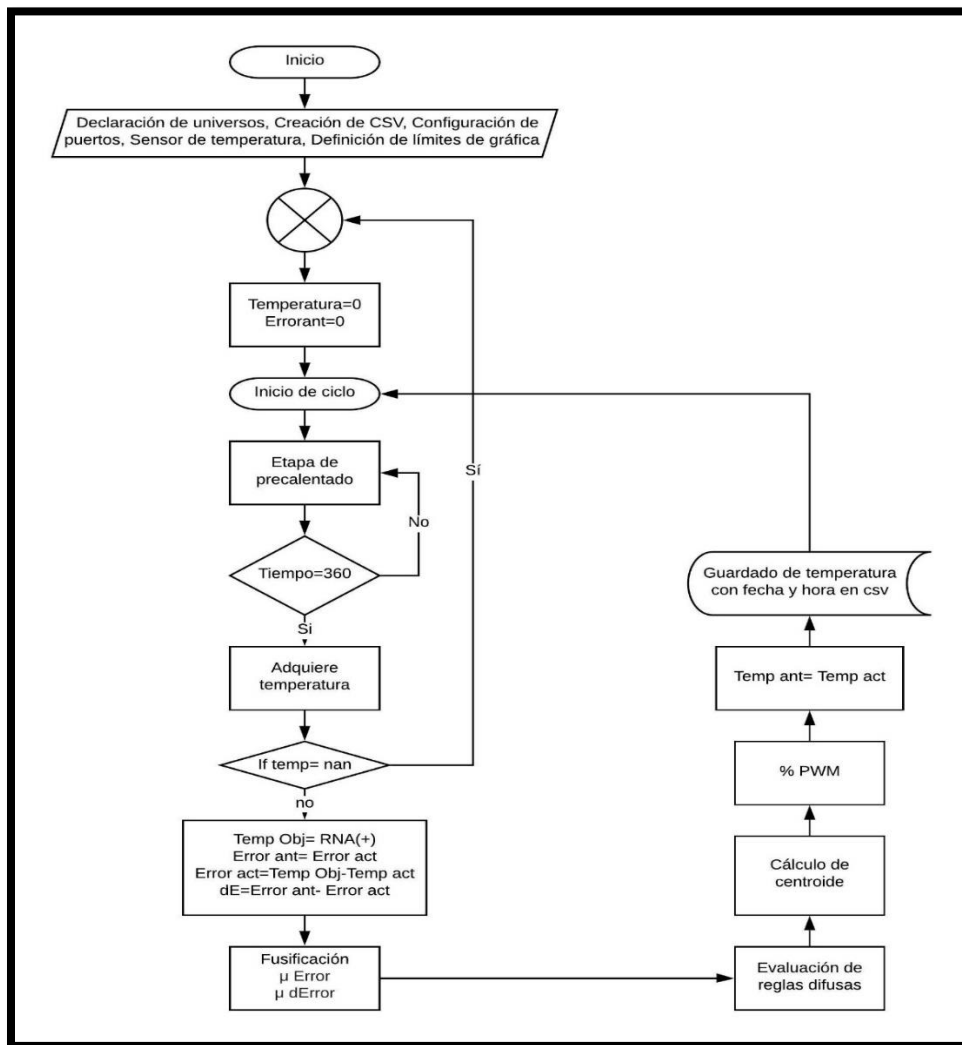


Figura 3.21 Diagrama de flujo de controlador inteligente.

3.7.1 Código en Raspberry Pi 3 B

En el anexo D se muestra detallado el código del programa que se implementa en el controlador inteligente basado Raspberry Pi 3 B. Este código se utilizó en varias pruebas dinámicas para el control de temperatura del horno eléctrico de reflujo.

CAPITULO 4

4. Resultados y discusión

Para verificar el desempeño del control de temperatura del horno se realizaron varias pruebas dinámicas con diferente cantidad y tipos de dispositivos de montaje superficial.

Las pruebas se realizaron con la finalidad de observar como varía la temperatura interior del horno debido a la masa de los dispositivos a soldar.

También se realizaron pruebas con diferentes sensores de temperatura para hallar el que proporciona datos más confiables en tiempo real.

El objetivo de las pruebas de soldadura de componentes SMT fue para observar la calidad del terminado de la soldadura (brillantez y fijación mecánica).

Con la finalidad de optimizar el gasto energético en el proceso de soldadura, se opta por procesar los datos obtenidos por L. Kwang [4] en una red neuronal artificial. Iniciando las pruebas con 100 neuronas en la capa oculta y 1000 iteraciones, con los resultados obtenidos se observó que mejora su aprendizaje aumentando a 2000 iteraciones, teniendo, así como resultado la curva de temperatura utilizada.

Posteriormente se aplica esta información al controlador de temperatura para eficientizar el proceso de caldeo de soldadura.

4.1 Resultado de aplicación en Red Neuronal

Al procesar todos los datos del trabajo de L. Kwang (Anexo A) en el modelo de la red neuronal, se obtiene la gráfica de los datos generados por la propia red, siendo la Figura 4.1.

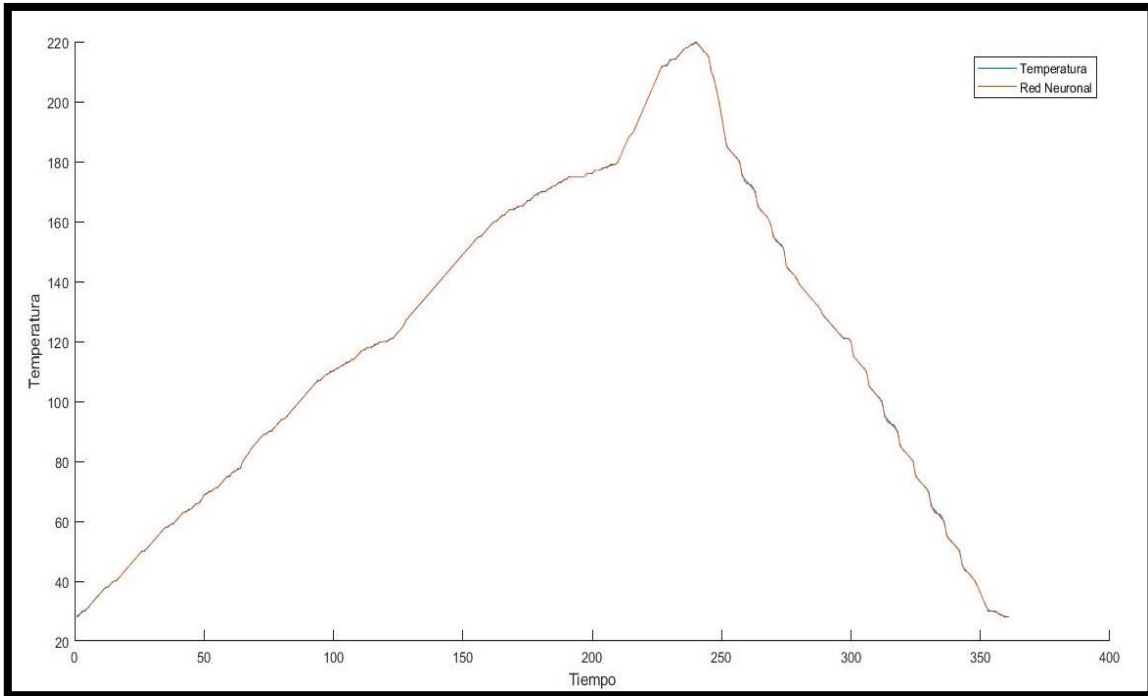


Figura 4.1 Resultado obtenido de proceso RNA en curva característica.

Al tener los datos procesados, se obtiene la matriz de datos para ingresar al controlador difuso que por su gran dimensión sólo será citado. Aunque si fuera posible mostrarla se notaría la precisión de que es capaz este controlador.

4.2 Resultado de aplicación de Controlador inteligente en Horno eléctrico

Después de realizar las pruebas dinámicas en el horno eléctrico de reflujo se obtuvieron los siguientes resultados, logrando alcanzar el perfil deseado de temperatura en la mayoría de las pruebas, mostradas en la Figura 4.2. Cabe mencionar que algunas variaciones de la temperatura también son alteradas por las condiciones del medio ambiente, debido a que este no cuenta con aislamiento térmico.

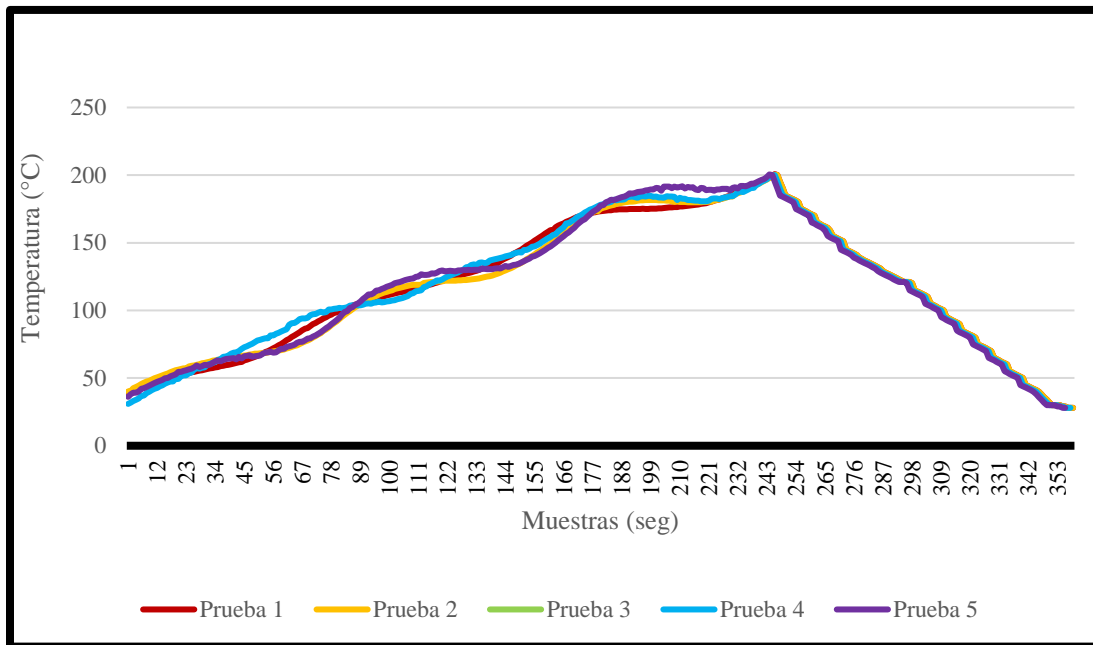


Figura 4.2 Resultados de caldeo de soldadura.

En la prueba 1 se energizó el horno en vacío (sin componentes para soldar).

En la prueba 2 se operó el horno con una placa de 2 x 1 cm, con resistencia SMT.

En la prueba 3 se operó el horno con una placa 2 x 2 cm, con un arreglo de transistor SMT.

En la prueba 4 se operó el horno con una placa 2 x 3 cm, con dispositivos discretos.

En la prueba 5 se operó el hrono con una placa 4 x 5 cm, con un arreglo de microprocesador SMT.

Al observar los resultados en las curvas, se llega a la conclusión que las variaciones de la señal adquirida en cada una de ellas son debido a las variaciones de la temperatura ambiente, porque es notorio que alcanzan la temperatura de pico superior casi en el mismo lapso.

Las pruebas dinámicas en el horno eléctrico de reflujo con el controlador inteligente integrado permitieron obtener muestras de dispositivos soldados con gran calidad en su terminado y sin daño a los dispositivos mismos. En la Figura 4.3 se muestra la diferencia entre la temperatura deseada y la temperatura objetivo.

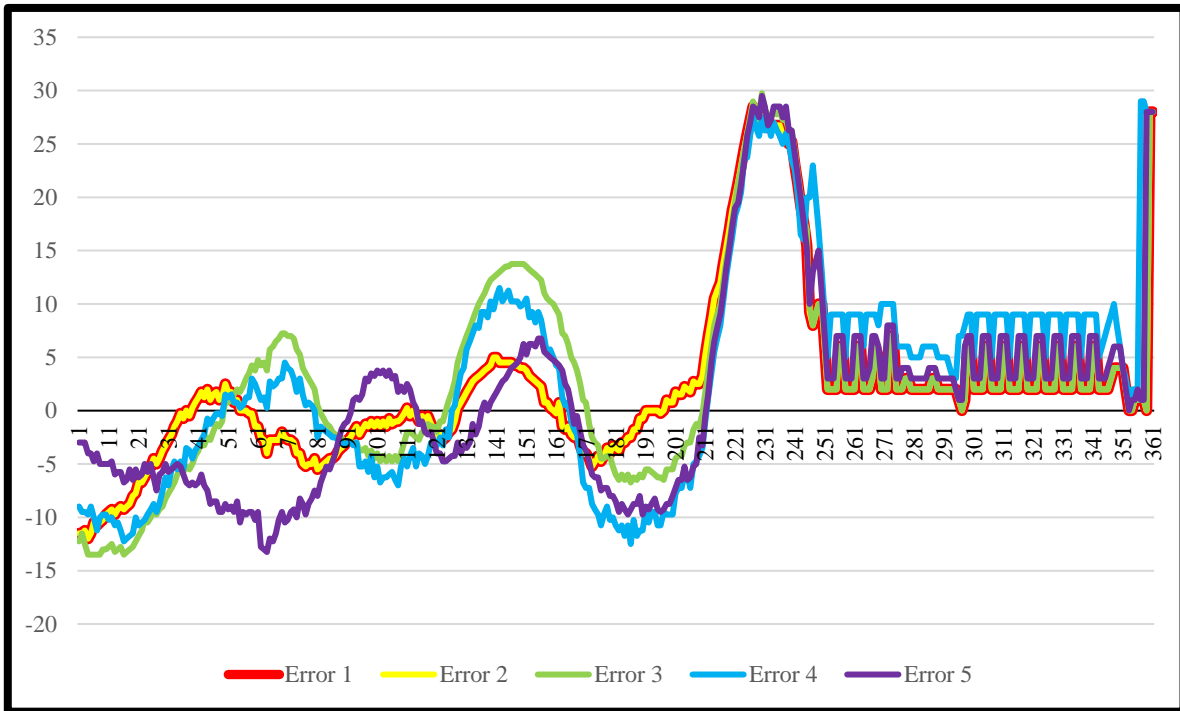


Figura 4.3 Diferencia entre temperatura deseada y temperatura objetivo

4.3 Resultados de acabados de soldadura

A continuación, se presenta el resultado de los acabados de la soldadura de los componentes en placa impresa de fibra de vidrio. En la Figura 4.4 se muestra un dispositivo de mediana potencia, que por lo general disipan la temperatura más rápidamente porque contienen mayor cantidad de metales en su construcción, sin embargo, se solda de manera perfecta.

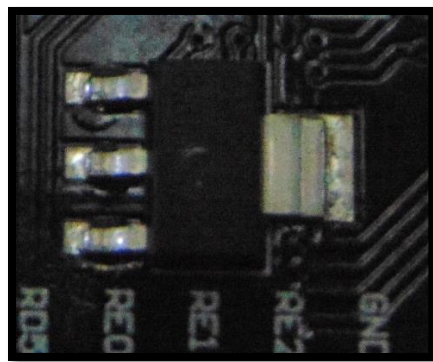


Figura 4.4 Transistor de mediana potencia.

En la Figura 4.5 se presenta la soldadura en componentes SMT como un circuito integrado microcontrolador que requiere una temperatura precisa para evitar daños en el mismo durante el proceso de soldadura.

Se destaca la brillantez que se observa en la soldadura de sus terminales, siendo esto muestra de la buena calidad en el proceso de soldadura.

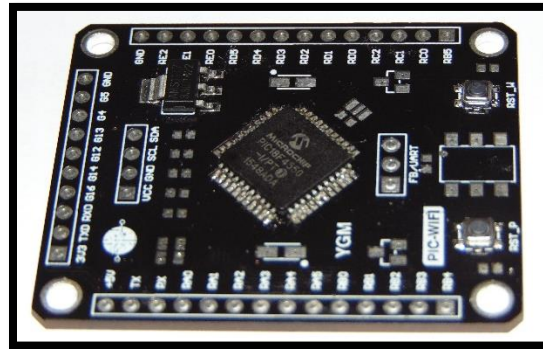


Figura 4.5 Circuito integrado microcontrolador.

En la Figura 4.6 se presenta un botón de montaje superficial miniatura componente SMT de dimensiones de 5mm por 5mm, que requiere una perfecta unión mecánica al sustrato debido a las funciones que realiza, de igual manera se observa la calidad y fuerza del anclaje que forma con la soldadura.

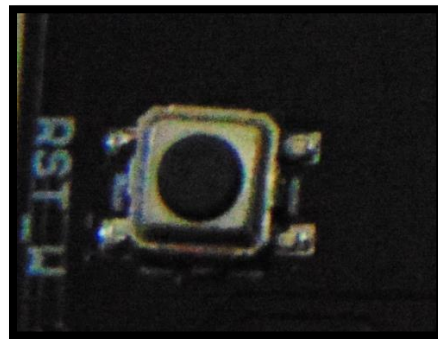


Figura 4.6 Botón metálico de montaje superficial.

Es importante mencionar que todas las imágenes anteriores pertenecen a una misma placa de circuito de montaje superficial y que fueron soldados en un mismo proceso, obteniendo cada uno de ellos la temperatura exacta en las uniones con el sustrato.

Estos resultados nos permiten aseverar que el control inteligente de temperatura del horno de reflujo que sometimos a prueba cumple con los objetivos planteados.

CAPITULO 5

5. Conclusiones

Después de observar los resultados en el proceso de horneado de pequeñas placas de circuitos electrónicos básicos, con dispositivos de montaje superficial en un sustrato de base fenólica y posteriormente en placas de base de fibra de vidrio, y aplicando un control inteligente de temperatura basado en un sistema embebido como el utilizado en este trabajo, es reconfortante ver que se obtienen resultados satisfactorios en cuanto a la calidad del terminado del producto y a la eficiencia energética de un equipo de características básicas, pero con las mejoras que implica la utilización de un sistema inteligente de control difuso. Con la aplicación de Lógica Difusa en el diseño de este controlador, se evita la necesidad de un operador experto en soldadura para realizar las labores de horneado en las placas de montaje superficial, siendo esta acción realizada por el controlador quien sigue una curva de temperatura de manera automática de acuerdo con las necesidades y que es recomendada para dispositivos SMT, debido al volumen y área de las superficies a ser soldadas.

Es importante mencionar que además de un sistema embebido eficiente, para el procesamiento de datos, es indispensable utilizar el mejor software embebido disponible con el que se pueda contar, ya que es notoria la sinergia que ambos pueden lograr.

Con lo anterior expuesto, se llega a la conclusión de que si se cumplieron con los objetivos específicos propuestos en este trabajo. Porque fue posible detectar en tiempo real las señales de temperatura, teniendo un 98.2% de eficiencia con respecto a la referencia de temperatura de la investigación de L. Kwang [4], se diseñó un sistema de control inteligente capaz de procesar los datos obtenidos por un sensor de temperatura, en combinación con el mejor software embebido disponible y que fueron implementados en el controlador inteligente, que a su vez se acopló a una interfaz de potencia, manteniendo el uso de la onda senoidal en un 99.4% con un resultado óptimo en el acabado del caldeo de soldadura, además de alcanzar una eficiencia energética aceptable, debido a que en las pruebas realizadas se observó que la salida de PWM disminuye en un 32%, teniendo apagada esta señal; optimizando el consumo energético por medio del controlador, que verifica la temperatura durante todo el proceso.

A futuro, se podría lograr una mayor optimización energética y de control de temperatura, integrando o fusionando dentro del control un sistema neurodifuso, que podría arrojar mejores resultados en este tipo de procesos.

CAPITULO 6

6. Bibliografía

- [1]Joers C., Gonzalez A., Medellin R. y Salas R. (2007). *System identification and temperatura control of a tuve furnace. TecnoINTELECTO*, pp.103-108.
- [2]Costa B., Lemos J. y Guillot E. (2018). *Solar furnace temperatura control with active cooling. Elsevier*, pp.66-77.
- [3]López P. y Aguilar R. (2009). *Dynamic Nonlinear Feedback for Temperature Control of Continuous Stirred Reactor with Complex Behavior. Journal of Applied Reseach and Tecnology*, pp.202-217.
- [4]Kwang Y. Lee. (2005). *Tutorial on Intelligent Optimization and Control for Power Systems: An introduction. Fellow IEEE*, pp.2-5.
- [5]Kumpati S. Narendra. (2002). *Intelligent Control using Neural Networks and Multiple Models. Proceedings of the 41st IEEE*. pp.1357-1362.
- [6]Chuen C. Lee. (1990). *Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part1. IEEE Transactions on systems*, pp.404-418.
- [7]Joseph Bih. (2006). *Paradigm shift- an introduction to fuzzy logic. IEEE Potentials*, pp.6-10.
- [8]Linkens D.A. (1992). *Adaptive and Intelligent Control in Anesthesia. IEEE Control Systems*, pp.6-11.
- [9]Ravidran B., Kachroo P. y Hegazy T. (2001). *Intelligent Feedback Control-Based Adaptive Resource Management for Asynchronous, Decentralized Real-Time Systems. IEEE Transactions on Systems*,pp.261-265.
- [10]Narendra K. y Balakrishnan J. (1995) *Intelligent Control using Fixed and Adaptive Models. Proceedings of the American ControlConference*, pp.597-601.
- [11]Narendra K. y Balakrishnan J. (1997) *Adaptive Control Using Multiple Models. Fellow IEEE*, pp.171-187.
- [12]Balkeshwar S. y Kumar M.A. (2015) *Fuzzy Logic Control System and its applications. International Research Journal of Engineering and Technology (IRJET)*, pp.742-746.

- [13]Saavedra, A.J. y Fernández, L. (2008) . *Controlling a planar temperature grid. Ingeniería y Desarrollo*, pp.133-148.
- [14]Boutalis,Y. (Ed). (2014). *System Identificatioon and Adaptive Control*. London,United Kindom: Springer.
- [15]García, R. y Servín, R. (2013). *Control y Monitoreo de temperatura en un horno electrico continuo para el tostado de café.(tesis depregrado)* Instituto Politécnico Nacional,Ciudad de México.
- [16]Goire,M.M.,Flores,J.M., Bonilla, M. y Baruch I.S.(2013). *Control Neuronal por Modelo de Referencia para un Sistema de Estructura Variable.Computación y Sistemas*,6(4),pp.284-292.
- [17]Hwang, J. (Ed). (2004). *Digital Engineering Library*. Cleveland, Ohio: McGraw-Hill.
- [18]Indium,C. (2013). *NC-SMQ 92J Solder Paste*. Singapore, Cheoungju, Asia. Recuperado de <http://documents.indium.com/qdynamo/download.php?docid=96>.
- [19]Poma, J., Raffo,E. y Berrospi,E. (2012). *Diseño, construcción e implementación de un sistema automatizado integrador para los módulos de caudar, presion y temperatura del centro de manufactura avanzada. Industrial data.*, pp.90+.
- [20]Mahmoud,M.(2018). *Fuzzy Control, Estimation and diagnosis*. Cham,Switzerland: Springer.
- [21]Quiroga, V. (2012). *Circuito de control para hornos de secadode transformadores. Energética Vol.2, 5*,pp1-3
- [22]Salazar,R.D., Pupo, A.C. y Jiménez, H.J.(2014). *Sistema energéticamente eficiente y de bajo costo para controlar la temperatura y aumentar el oxígeno en estanques de cultivo de alevines de tilapia roja. Revista Facultad de Ingeniería*,23(36),pp.9-23.
- [23]Suárez,L.D (2016). *Diseño de un sistema de control electrónico de voltaje para un auto transformador tipo escobilla móvil para alimentacion de potencia a hornos electricos de curvado de vidrio*. Instituto Politécnico Nacional, Ciudad de México.
- [24]Takeyas,B. (2009). *Introducción a la inteligencia artificial. Instituto tecnológico de Nuevo Laredo*, pp.1-3. Recuperado de <http://www.itnuevolaredo.edu.mx/takeyas/Articulos/Inteligencia%20Artificial/ARTICULO%20Introduccion%20a%20la%20Inteligencia%20Artificial.pdf>.
- [25]Tenenbaum,J(2017,23 de Agosto).*We study the computacional basis of human learning and inference*. Recuparado de <https://www.csail.mit.edu/research/computational-cognitive-science-group>
- [26]Tirso,A. (2002). *Homo Cybersapiens. La inteligencia artificial y la humana*. Madrid,España : EUNSA.

- [27]Trinks,W. (Ed).(2006). **Industrial Furnances**. *Industrial Furnances* (págs. 50- 56). United States of America: Wiley.
- [28]Tsai,T., Yang, T. y Hou, P. (2005). **Neurofuzzy modeling of the reflow thermal profile for sufarcemount assembly**. *International Journal of Systems Science*,6(2),pp.89-101.
- [29] Cortés, T. A. (2016). **Los sistemas embebidos en la industria 4.0. Sistemas Embebidos: Estado actual con visión al futuro**. ISBN 978-607-9394-08-0, 2-6.
- [30]Ponce, P. (2010). **Inteligencia Artificial con aplicaciones a la ingeniería (2ª ed.)**. Ciudad de México, México: Alfaomega.
- [31]Hernández, J. (2010). **El Software Embebido y los Retos que implica su desarrollo**. *Conciencia Tecnológica*, 40, 42–45.
- [32]Triviño, M. (2010). **Servidor Web Embebido en una FPGA con Codiseño como Metodología de Diseño**. *Innovation and Development for the Americas*, 1(4), 1–8.
- [33]Ebert, Christof y Jones Capers (2009), “**Embedded software: facts , figures and future**”. **Innovative Technology for Computer Professionals Computer**. abril. Volumen 42. Núm 4. Washington D.C. USA, pp 42-52.
- [34]Placeres, J.M. (2002) **Desarrollo de prototipos para tecnología de montaje superficial (SMT)**. *División de Ingeniería y arquitectura Tec Monterrey*, 1(1), 1-84.

ANEXO A

Tablas de temperaturas de referencia de curva característica (L. Kwang)

1	28				
2	29				
3	30				
4	30				
5	31				
6	32				
7	33				
8	34				
9	35				
10	36				
11	37				
12	38				
13	38				
14	39				
15	40				
16	40				
17	41				
18	42				
19	43				
20	44				
21	45				
22	46				
23	47				
24	48				
25	49				
26	50				
27	50				
28	51				
29	52				
30	53				
31	54				
32	55				
33	56				
34	57				
35	58				
36	58				
37	59				
38	59				
		39	60	77	91
		40	61	78	92
		41	62	79	93
		42	63	80	94
		43	63	81	94
		44	64	82	95
		45	64	83	96
		46	65	84	97
		47	66	85	98
		48	66	86	99
		49	67	87	100
		50	69	88	101
		51	69	89	102
		52	70	90	103
		53	70	91	104
		54	71	92	105
		55	71	93	106
		56	72	94	107
		57	73	95	107
		58	74	96	108
		59	75	97	109
		60	75	98	109
		61	76.5	99	110
		62	76.5	100	110
		63	77.5	101	111
		64	77.5	102	111
		65	80	103	112
		66	81.25	104	112
		67	82.5	105	113
		68	83.75	106	113
		69	85	107	114
		70	86	108	114
		71	87	109	115
		72	88	110	116
		73	89	111	117
		74	89	112	117
		75	90	113	118
		76	90	114	118

115	118
116	119
117	119
118	120
119	120
120	120
121	120
122	121
123	121
124	122
125	123
126	124
127	125
128	127
129	128
130	129
131	130
132	131
133	132
134	133
135	134
136	135
137	136
138	137
139	138
140	139
141	140
142	141
143	142
144	143
145	144
146	145
147	146
148	147
149	148
150	149
151	150
152	151
153	152
154	153
155	154

156	155
157	155
158	156
159	157
160	158
161	159
162	160
163	160
164	161
165	162
166	162
167	163
168	164
169	164
170	164
171	165
172	165
173	165
174	166
175	167
176	167
177	168
178	169
179	169
180	170
181	170
182	170
183	171
184	171
185	172
186	172
187	173
188	173
189	174
190	174
191	175
192	175
193	175
194	175
195	175
196	175

197	175
198	176
199	176
200	176
201	177
202	177
203	177
204	178
205	178
206	178
207	179
208	179
209	179
210	180
211	182
212	184
213	186
214	188
215	189
216	190
217	192
218	194
219	196
220	198
221	200
222	202
223	204
224	206
225	208
226	210
227	212
228	212
229	212
230	214
231	214
232	214
233	215
234	216
235	217
236	218
237	218

238	219
239	219
240	220
241	219
242	218
243	217
244	216
245	215
246	210
247	208
248	204
249	200
250	195
251	190
252	185
253	184
254	183
255	182
256	181
257	180
258	175
259	174
260	173
261	172
262	171
263	170
264	165
265	164
266	163
267	162
268	161
269	159
270	155
271	154
272	153
273	152
274	151
275	145
276	144
277	143
278	142

279	141
280	139
281	138
282	137
283	136
284	135
285	134
286	133
287	132
288	131
289	129
290	128
291	127
292	126
293	125
294	124
295	123
296	122
297	121
298	121
299	121
300	120
301	115
302	114
303	113
304	112
305	111
306	110
307	105
308	104
309	103
310	102
311	101
312	100
313	95
314	94
315	93
316	92
317	91
318	90
319	85

320	84
321	83
322	82
323	81
324	80
325	75
326	74
327	73
328	72
329	71
330	70
331	65
332	64
333	63
334	62
335	61
336	60
337	55
338	54
339	53
340	52
341	51
342	50
343	45
344	44
345	43
346	42
347	41
348	40
349	38
350	36
351	34
352	32
353	30
354	30
355	30
356	30
357	29
358	29
359	28
360	28

ANEXO B

Código de programa de aplicación en Matlab para optimización de datos mediante red neuronal.

```
clear all; close all; clc; format short;
```

```
Array=csvread('curva.csv');  
tiempo = Array(:, 1);  
temperatura = Array(:, 2);
```

```
input = tiempo;
```

```
B = temperatura;  
target = B - min(B(:));  
target = target ./ max(target(:));
```

```
net = feedforwardnet(100);  
net.trainFcn = 'trainlm';  
net.layers{1}.transferFcn='elliotsig';  
net.divideParam.trainRatio = 1;  
net.divideParam.valRatio = 0;  
net.divideParam.testRatio = 0;  
[net,tr,Y,E] = train(net,input',target');  
view(net)  
MARDUINO= B-net(input');
```

```
hold on  
plot(target)  
plot(net(input'))  
hold on  
ylabel('Temperatura')  
xlabel('Tiempo')  
xlswrite('Valores_Par_Net_Curv_SoldaduraN100.xlsx',MARDUINO, 'Hoja1','A1')  
plot(MARDUINO)
```

ANEXO C

Programa aplicado en Arduino IDE y Energía IDE

```
#include <LiquidCrystal_I2C.h>
unsigned long tiempo = 0;
//Variables difusas
float ErTem[] = {0, 1, 2, 3, 4}, dErTem[] = { -0.1, 0,0.1};
float PotPWM[] = {0.25, 85, 40, 55, 65, 80, 95};
float DifusEr[] = {0, 0, 0, 0, 0}, DifusdER[] = {0, 0, 0};
float Potencia[] = {0, 0, 0, 0, 0, 0, 0};
float ReglaDif[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
float TaT, error, errorAnt, de, real, real2, X, Y, Z;
float Ta1,Ta2,Ta21,Ta22,Ta23,Ta24,Ta25,Ta26,Ta27,Ta28,Ta29,Ta210;
//valor de salida para PWM
int PW, fin;
//valores de contadores
int i = 0, j = 0, n = 0;
// valor de ajuste de setpoint
float setpoint = 0;
//valor de tuneo de setpoint
int setpointmas = 7;
int setpointmenos = 8;
int valor1, valor2;
float temperatura,y,yant=0,xant=0;
//declaracion de salida a PWM
int mosfet = 6;
// Envío de datos vía excel
char Excel1[10], Excel2[10], Excel3[10], Excel4[10], Excel5[10];

LiquidCrystal_I2C lcd(0x3f, 16, 4);

void setup() {
  Serial.begin(9600); // Iniciación de programa vía serial
  pinMode(6, OUTPUT); // Declaración de salida PIN 6
  pinMode(setpointmas, INPUT); //Declaración de valor de suma para tuneo de setpoint
  pinMode(setpointmenos, INPUT); //Declaración de valor de resta para tuneo de setpoint
  pinMode(A0, INPUT); // Declaracion de valor de entrada analógica A0
  pinMode(A1, INPUT);
  lcd.init(); //Inicialización de display
  lcd.backlight();
  lcd.print("Iniciando"); //Impresión en display
  lcd.setCursor(0, 1);
  lcd.print("Control Difuso"); //Impresión en display
  delay(2000);
  lcd.setCursor(0, 0);
```

```

lcd.print("Obteniendo Inf."); //Impresión en display
lcd.setCursor(0, 1);
lcd.print("          "); //Impresión en display
delay(5000);
lcd.clear();
Ta1 = analogRead(A0); // Lectura del Pin A0
delay(2);
Ta1 = Ta1 / 1023 * 5 * 100; // Conversión de la señal
delay(2);
Ta21 = analogRead(A1); // Lectura del Pin A0
delay(5);
//temperatura=Ta21*500/1024;
//y=0.6859*yant+0.3141*xant;
//yant=Ta2;
//xant=temperatura;
//delay(50);
Ta22 = analogRead(A1);
delay(5);
Ta23 = analogRead(A1);
delay(5);
Ta24= analogRead(A1);
delay(5);
Ta25 = analogRead(A1);
delay(5);
Ta26= analogRead(A1);
delay(5);
Ta27 = analogRead(A1);
delay(5);
Ta28 = analogRead(A1);
delay(5);
Ta29 = analogRead(A1);
delay(5);
Ta210 = analogRead(A1);
delay(5);
Ta2=(Ta21+Ta22+Ta23+Ta24+Ta25+Ta26+Ta27+Ta28+Ta29+Ta210)/10;
delay(10);
Ta2 = Ta2 / 1023 * 5 * 100; // Conversión de la señal
delay(100);
real = Ta1;
real2 = Ta2;      // Guardar la señal en dato real
//valor1 = digitalRead(setpointmas);
//valor2 = digitalRead(setpointmenos);
//Condicionales para aumentar o disminuir el setpoint

TaT = real2;
fin = setpoint; // Guardado de la señal de setpoint en variable fin
valor1= int(fin);

```

```

delay(2);
error = valor1 - TaT; // Variable de error de temperatura
delay(2);
}

void loop() {
  tiempo = millis();
  //Condicionales para aumentar o disminuir el setpoint
  if (((digitalRead(setpointmas) == HIGH) && (digitalRead(setpointmenos) == HIGH))
  {
    setpoint = setpoint;
    delay(50);
  }
  else if (((digitalRead(setpointmas) == HIGH) && (digitalRead(setpointmenos) == LOW))
  {
    setpoint = setpoint + 1;
    delay(50);
  }
  else if (((digitalRead(setpointmas) == LOW) && (digitalRead(setpointmenos) == HIGH))
  { setpoint = setpoint - 1;
    delay(50);
  }
  else if (((digitalRead(setpointmas) == LOW) && (digitalRead(setpointmenos) == LOW))
  { setpoint = setpoint;
    delay(50);
  }
  //Vectores difusos
  for (i = 0; i < 5; i++) { //Iguala a 0 cada variable de los vectores.
    DifusEr[i] = 0;
  }
  for (i = 0; i < 3; i++) {
    DifusdER[i] = 0;
  }

  fin = setpoint; // Guardado del valor de setpoint en la variable fin
  valor1=int(fin);
  delay(2);
  Ta1 = analogRead(A0); //Lectura del A0 y guardado en la variable Ta1
  delay(10);
  Ta1 = Ta1 / 1023 * 5 * 100; //Conversión de la variable obtenida
  delay(10);
  Ta21 = analogRead(A1); // Lectura del Pin A0
  delay(5);
  Ta22 = analogRead(A1);
  delay(5);
  Ta23 = analogRead(A1);
  delay(5);
}

```

```

Ta24= analogRead(A1);
delay(5);
Ta25 = analogRead(A1);
delay(5);
Ta26= analogRead(A1);
delay(5);
Ta27 = analogRead(A1);
delay(5);
Ta28 = analogRead(A1);
delay(5);
Ta29 = analogRead(A1);
delay(5);
Ta210 = analogRead(A1);
delay(5);
Ta2=(Ta21+Ta22+Ta23+Ta24+Ta25+Ta26+Ta27+Ta28+Ta29+Ta210)/10;
//Ta2 = analogRead(A1); //Lectura del A0 y guardado en la variable Ta1
//Ta2=(Ta2+Ta2+Ta2+Ta2+Ta2+Ta2+Ta2+Ta2+Ta2+Ta2)/10;
//yant=Ta2;
//xant=temperatura;
//delay(50);
Ta2 = Ta2 / 1023 * 5 * 100; //Conversión de la variable obtenida
delay(100);
real = int(Ta1);      //Envío de la variable Ta1 a la variable real
real2 = Ta2;
TaT = real2;
errorAnt = error; //Traspaso de la variable error en la variable errorAnt
error = valor1 - TaT; //Operación del error
de = errorAnt - error; //Operación Derivada del error
//Impresión en el LCD
lcd.setCursor(0, 0);
lcd.print("TD: "); // Temperatura deseada
lcd.setCursor(3, 0);
lcd.print(fin); //Escribe el valor de la variable
delay(10);
lcd.setCursor(0, 1);
lcd.print("TA: "); // Temperatura actual
lcd.setCursor(3, 1);
lcd.print(real);
delay(10);
lcd.setCursor(9, 1);
lcd.print("TA: "); // Temperatura actual
lcd.setCursor(12, 1);
lcd.print(real2);
delay(10);
//Difusion
//Derivada de Error
if (de < -0.01) { //Función LAMDA NEG

```



```

    DifusdER[0] = 1;
}
else if ((de >= -0.01) && (de <= 0)) { //Función TRIANGULAR CERO
    DifusdER[0] = (de - dErTem[1]) / (dErTem[0] - dErTem[1]);
    DifusdER[1] = 1 - DifusdER[0]; //Complemento conjunto difuso
}
else if ((de > 0) && (de < 0.01)) { //Función TRIANGULAR CERO
    DifusdER[1] = (de - dErTem[2]) / (dErTem[1] - dErTem[2]);
    DifusdER[2] = 1 - DifusdER[1];
}
}
else if (de >= 0.01) {
    DifusdER[2] = 1;
}
}
//Error de Temperatura
if (error < 0) { //Función LAMDA MBajo
    DifusEr[0] = 0;
}
else if ((error >= 0) && (error < 1)) { //Funciones LAMDA MBajo Y TRIANGULAR
Bajo
    DifusEr[0] = (ErTem[1] - error) / (ErTem[1] - ErTem[0]); //LAMDA MBajo
    DifusEr[1] = 1 - DifusEr[0]; //Complemento TRIANGULAR Bajo
}
else if ((error >= 1) && (error < 2)) { //Funciones TRIANGULAR Bajo y
TRIANGULAR Medio
    DifusEr[1] = (ErTem[2] - error) / (ErTem[2] - ErTem[1]); //TRIANGULAR Bajo
    DifusEr[2] = 1 - DifusEr[1]; //Complemento TRIANGULAR Medio
}
else if ((error >= 2) && (error < 4)) {
    DifusEr[2] = (ErTem[3] - error) / (ErTem[3] - ErTem[2]); //TRIANGULAR Medio
    DifusEr[3] = 1 - DifusEr[2]; //Complemento TRIANGULAR Grande
}
}
else if ((error >= 4) && (error < 5)) { //Funciones TRIANGULAR Grande y
TRIANGULAR MGrande
    DifusEr[3] = (ErTem[4] - error) / (ErTem[4] - ErTem[3]); //TRIANGULAR Grande
    DifusEr[4] = 1 - DifusEr[3]; //Complemento GAMMA MGrande
}
}
else if (error >= 5) { //Función Gamma MGRANDE
    DifusEr[4] = 1;
}
}
// Defuzzificación
n = 0; //Numero de regla difusa
for (i = 0; i < 3; i++) { //Contador dError
    for (j = 0; j < 5; j++) { //Contador ErroDifuso
        ReglaDif[n] = min( DifusEr[j], DifusdER[i]);
        n = n + 1;
    }
}
}

```

```

//Operación de máximos
Potencia[0] = ReglaDif[5]; //Potencia[0]=>Minima_PWM
Potencia[1] = max(ReglaDif[11], ReglaDif[0]); //Potencia[1]=>Bajo1_PWM
Potencia[2] = ReglaDif[6]; //Potencia[2]=>Bajo2_PWM
Potencia[3] = ReglaDif[1]; //Potencia[3]=>Media_PWM
Potencia[4] = max(ReglaDif[7], ReglaDif[12]); //Potencia[4]=>Alto1_PWM
Potencia[5] = max(max (max(ReglaDif[2], ReglaDif[8]), ReglaDif[13]), ReglaDif[14]);
//Potencia[5]=>Alto2_PWM
Potencia[6] = max(max(ReglaDif[3], ReglaDif[4]), ReglaDif[9]);

//Cálculo de Centroide
X = 0, Y = 0, Z = 0; //Iguala a 0 las variables X, Y y Z.
for (i = 0; i < 7; i++) {
  X = Potencia[i] * PotPWM[i] + X;
  Y = Potencia[i] + Y;
}
Z = X / Y; //Salida de Z es en porcentaje
// Salida de control del mosfet

float PW = (Z / 100) * 255; //Porcentaje de PWM. 100% PWM=255
PW=PW;
//Impresión de la señal en LCD
lcd.setCursor(6, 0);
lcd.print("PWM:   ");
lcd.setCursor(10, 0);
if (isnan(PW))
{ PW = 0;
}
lcd.print(PW);
//Salida de la señal a Mosfet/Relevador estado sólido
analogWrite(mosfet, PW);
//Serial.print(" ");
//delay(1);
//Serial.print("");
//dtostrf(Ta1,5,3,Excel1);
//Serial.print(Excel1 );
//Serial.println(Ta1 );
//Serial.print(" ");
delay(30);
//dtostrf(Ta2, 5, 3, Excel2);
//Serial.print(Excel2 );
//Serial.println(Ta2 );
//Serial.print(" ");
//delay(30);
//dtostrf(TaT, 5, 3, Excel3);
//Serial.print(Excel3 );
Serial.println(Ta2);

```

```
//Serial.print(" ");  
//delay(30);  
//dtostrf(fin, 5, 3, Excel4);  
//Serial.print(Excel4 );  
//Serial.println(fin );  
//Serial.print(" ");  
//delay(30);  
//dtostrf(PW, 5, 3, Excel5);  
//Serial.println(Excel5 );  
//Serial.println(PW );  
//delay(30);  
}
```

ANEXO D

Programa aplicado en Raspberry Pi 3 B

```
#!/usr/bin/env python
import os
import glob
import time
import subprocess
import time
import os
import RPi.GPIO as GPIO
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt
import sys
import pigpio
import csv
import pandas as pd
import time
import signal
import Adafruit_GPIO.SPI as SPI
import Adafruit_MAX31855.MAX31855 as MAX31855
import numpy

from time import sleep, strftime, time
import matplotlib.pyplot as plt

''' Contactor y Ventilador. '''
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
Contactor = 18 # GPIO = 18, PIN 12
Ventilador = 5 # GPIO = 5, PIN 29
GPIO.setup(Contactor, GPIO.OUT)
GPIO.setup(Ventilador, GPIO.OUT)
''' Raspberry Pi software SPI configuration. '''
CLK = 23
CS = 24
DO = 21
sensor = MAX31855.MAX31855(CLK, CS, DO)

''' Raspberry Pi hardware SPI configuration. '''
SPI_PORT = 0
SPI_DEVICE = 0
sensor = MAX31855.MAX31855(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))

resistor = 2 #Valor GPIO = 2, Pin = 3
t = 0 # Tiempo

GPIO.output(18, GPIO.HIGH)
```

```

GPIO.setup(2, GPIO.OUT)
p = GPIO.PWM(2, 500)
plt.ion()
x = []
y = []

def destroy():
    GPIO.cleanup()

def enfriar():
    while True:
        GPIO.output(5, GPIO.HIGH)
        tempcool = float(sensor.readTempC())
        p.stop()
        sleep(1)
        #tempcool = float(input("Cuáles la temperatura? : "))
        print ("Temperatura",tempcool)

        print("\n")
        if tempcool <= 32:
            print("¡Proceso completado!")
            GPIO.cleanup()
            #break
            sys.exit()

def keyboardInterruptHandler(signal,frame):
    print("KeyboardInterrupt (ID: {}) has been caught. Cleaning up... ".format(signal))
    destroy()
    print("Turning off PWM Signal")
    exit()

#def pwm_stop(temp):
#    if sensor.readTempC() > objt_temp:
#        p.stop()
#        print('PWM Value: ',control_pwm,'%')

def graph(temp):
    y.append(temp)
    x.append(time())
    plt.clf()
    plt.scatter(x,y)
    plt.plot(x,y)
    plt.draw()

def escrituradata():
    tempsave = sensor.readTempC()
    with open("PruebasOctubre.csv", "a") as log:
        log.write("{0},{1},{2},{3}\n".format(strftime("%Y-%m-%d
%H:%M:%S"),str(tempsave),str(value),str(control_value)))
        graph(tempsave)

```

```

signal.signal(signal.SIGINT, keyboardInterruptHandler)
GPIO.output(5, GPIO.LOW)
GPIO.output(18, GPIO.LOW)
error_ant = 0
while True:
    print('Etapa de precalentado')
    error_ant = 0
    for i in range(1,120):  "Repeticiones de precalentado"
        temp0 = float(sensor.readTempC())
        print("Tiempo: ',i, 's')
        print("Temperatura: ',temp0,'°C')
        print('\n')
        p.start(100)
        sleep(1)
    p.stop()
    print ('Etapa concluida')
    print('Iniciando control difuso')
    print('\n')

    temp = float(sensor.readTempC())
    #tempnan = [~numpy.isnan(temp)]
    #if tempnan == [False]:
    # print('Error al obtener la temperatura')
    # print('\n')
    if temp >= 150:
        enfriar()

    else:

        df = pd.read_csv('curvas.csv')
        for index, row in df.iterrows():

            temp = sensor.readTempC()
            #nueva_temp = float(input("Temperatura actual: "))  #Valor variable para pruebas
            if temp >= 200:
                enfriar()
            else:
                print('Ciclo #',index)
                #horno_temp = float(sensor.readTempC())

            #Generar variables universales
            #temperatura objetivo y horno en un rango de 0 a 250
            #el mismo para la variable de salida (PWM)

            x_error = np.arange(-50,50,1)  #Error
            x_dError = np.arange(-250,250,1) #dError
            x_pwm = np.arange(0,100,1)  #PWM

```

```

#Generar funciones de membresia""
error_very_low = fuzz.trapmf(x_error, [-50,-50,-3,1])
error_low = fuzz.trimf(x_error, [0,2,4])
error_med = fuzz.trimf(x_error, [3,6,9])
error_med_high = fuzz.trimf(x_error, [7,10,13])
error_high = fuzz.trapmf(x_error, [11,14,50,50])

"" Derivada del Error ""
dError_minus = fuzz.trapmf(x_dError, [-250,-250,-0.5,0])
dError_zero = fuzz.trimf(x_dError, [-0.5,0,0.5])
dError_plus = fuzz.trapmf(x_dError, [0,0.5,250,250])

"" Control de salidad PWM ""
control_temp_pwm_off = fuzz.trapmf(x_pwm, [0,0,10,20])
control_temp_pwm_very_low = fuzz.trimf(x_pwm, [10,35,40])
control_temp_pwm_low = fuzz.trimf(x_pwm, [35,45,55])
control_temp_pwm_med = fuzz.trimf(x_pwm, [45,55,65])
control_temp_pwm_med_high = fuzz.trimf(x_pwm, [55,65,75])
control_temp_pwm_high = fuzz.trimf(x_pwm, [70,80,95])
control_temp_pwm_on = fuzz.trapmf(x_pwm, [80,95,100,100])

#error_ant = input("Error anterior: ")
#error_ant = float(error_ant)

value = int((row[1]))          #Valor del setpoint obtenido del TXT
#value = float(input('Setpoint: '))

print('Seguimiento: ',value, '°C')

#print('Error anterior: ',error_ant)

objt_temp = value             # Valor del setpoint variable

#nueva_temp = float(input("Temperatura actual: "))    #Valor variable para
pruebas
nueva_temp = float(sensor.readTempC())
error = objt_temp - nueva_temp
#print('Error: ',error)
#print ('El error anterior es ', error_ant)    #Para fines de depuración del programa
dError = error - error_ant
#sleep(2)

#nueva_temp = float(sensor.readTempC())
#nueva_temp = 110

#escriuradata()

print ('Temperatura: ',nueva_temp,'°C')

error_ant = error

```

```

#print('dError',dError)
sleep(7) ""Tiempo de ESPERA""
#print('La derivada del error es ', objt_temp1)

""Calcular grados de membresia""

error_very_low = fuzz.interp_membership(x_error,error_very_low, float(error))
error_low = fuzz.interp_membership(x_error,error_low, float(error))
error_med = fuzz.interp_membership(x_error,error_med, float(error))
error_med_high = fuzz.interp_membership(x_error,error_med_high,
float(error))

error_high = fuzz.interp_membership(x_error,error_high, float(error))
"" Control de Objetivo (Derivada del error) ""

dError_temp_minus = fuzz.interp_membership(x_dError,dError_minus,
float(dError))

dError_temp_zero = fuzz.interp_membership(x_dError,dError_zero,
float(dError))

dError_temp_plus= fuzz.interp_membership(x_dError,dError_plus,
float(dError))

#Aplicar las reglas

#####
active_rule1 = np.fmin(error_very_low,dError_temp_minus)
#R1 = np.fmin(active_rule1,control_temp_pwm_muy_bajo)
R1 = np.fmin(active_rule1,control_temp_pwm_off)
#regla 2
active_rule2 = np.fmin(error_very_low,dError_temp_zero)
#R2 = np.fmin(active_rule2,control_temp_pwm_bajo)
R2 = np.fmin(active_rule2,control_temp_pwm_off)
#regla 3
active_rule3 = np.fmin(error_very_low,dError_temp_plus)
R3 = np.fmin(active_rule3,control_temp_pwm_off)
#R3 = np.fmin(active_rule3,control_temp_pwm_apagado)
#####
#regla 4
active_rule4 = np.fmin(error_low,dError_temp_minus)
#R4 = np.fmin(active_rule4,control_temp_pwm_medio)
R4 = np.fmin(active_rule4,control_temp_pwm_off)
#regla 5
active_rule5 = np.fmin(error_low,dError_temp_zero)
#R5 = np.fmin(active_rule5,control_temp_pwm_bajo)
R5 = np.fmin(active_rule5,control_temp_pwm_off)
#regla 6
active_rule6 = np.fmin(error_low,dError_temp_plus)
R6 = np.fmin(active_rule6,control_temp_pwm_med)
#R6 = np.fmin(active_rule6,control_temp_pwm_muy_bajo)

```



```

#####
#regla 7
active_rule7 = np.fmin(error_med,dError_temp_minus)
#R7 = np.fmin(active_rule7,control_temp_pwm_med_alto)
R7 = np.fmin(active_rule7,control_temp_pwm_very_low)
#regla 8
active_rule8 = np.fmin(error_med,dError_temp_zero)
#R8 = np.fmin(active_rule8,control_temp_pwm_medio)
R8 = np.fmin(active_rule8,control_temp_pwm_low)
#regla 9
active_rule9 = np.fmin(error_med,dError_temp_plus)
#R9 = np.fmin(active_rule9,control_temp_pwm_medio)
R9 = np.fmin(active_rule9,control_temp_pwm_med_high)

#####
#regla 10
active_rule10 = np.fmin(error_med_high,dError_temp_minus)
#R10 = np.fmin(active_rule10,control_temp_pwm_alto)
R10 = np.fmin(active_rule10,control_temp_pwm_med)
#regla 11
active_rule11 = np.fmin(error_med_high,dError_temp_zero)
#R11 = np.fmin(active_rule11,control_temp_pwm_med_alto)
R11 = np.fmin(active_rule11,control_temp_pwm_high)
#regla 12
active_rule12 = np.fmin(error_med_high,dError_temp_plus)
#R12 = np.fmin(active_rule12,control_temp_pwm_med_alto)
R12 = np.fmin(active_rule12,control_temp_pwm_high)

#####
#regla 13
active_rule13 = np.fmin(error_high,dError_temp_minus)
#R13 = np.fmin(active_rule13,control_temp_pwm_encendido)
R13 = np.fmin(active_rule13,control_temp_pwm_med_high)
#regla 14
active_rule14 = np.fmin(error_high,dError_temp_zero)
#R14 = np.fmin(active_rule14,control_temp_pwm_encendido)
R14 = np.fmin(active_rule14,control_temp_pwm_on)
#regla 15
active_rule15 = np.fmin(error_high,dError_temp_plus)
#R15 = np.fmin(active_rule15,control_temp_pwm_alto)
R15 = np.fmin(active_rule15,control_temp_pwm_on)

#####

ctr0 = np.zeros_like(x_pwm)

Conjunto_Array = np.array ([R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11,
R12, R13, R14, R15])

Cantidad_conjunto = np.size (Conjunto_Array)

```

```

CF1 = np.array(Conjunto_Array[0])

for i in range (0,14):
    CF2 = np.array(Conjunto_Array[i+1])
    CF1 = np.fmax(CF1,CF2)

aggregated = CF1
#print (CF1)

control_value = fuzz.defuzz(x_pwm,aggregated, 'centroid')
escribiradata()
control_active = fuzz.interp_membership(x_pwm,aggregated,control_value)
#print ('Centroid: ',control_value)

''' fig, ax4 = plt.subplots(figsize=(8,3))
ax4.plot(x_pwm ,control_temp_pwm_off,'b',linewidth=.5, linestyle='--')
ax4.plot(x_pwm ,control_temp_pwm_very_low,'b',linewidth=.5, linestyle='--')
ax4.plot(x_pwm ,control_temp_pwm_low,'r',linewidth=.5, linestyle='--')
ax4.plot(x_pwm ,control_temp_pwm_med,'b',linewidth=.5, linestyle='--')
ax4.plot(x_pwm ,control_temp_pwm_med_high,'b',linewidth=.5, linestyle='--')
ax4.plot(x_pwm ,control_temp_pwm_high,'g',linewidth=.5, linestyle='--')
ax4.plot(x_pwm ,control_temp_pwm_on,'g',linewidth=.5, linestyle='--')
ax4.fill_between(x_pwm,ctr0,aggregated,facecolor='orange',alpha=0.7)
ax4.plot([control_value,
control_value],[0,control_active],'k',linewidth=1.5,alpha=0.9)
ax4.set_title('Salida PWM')
fig.tight_layout()
plt.ion()'''
'''PWM control conditions'''
if control_value > 0.99 and control_value <= 8.8096:
    p.stop()
    print('PWM Off')

if control_value == 9.367201426024955:
    p.stop()
    print('PWM Off')

elif control_value >= 8.80961:
    p.start(control_value)
    print('PWM Value: ',control_value,'%')

'''Temperature File Writing'''
#temp.to_csv('Seguimientocurva.csv')
#with open("horno_seguimiento200.csv", "a") as log:
#    log.write("{0},{1}\n".format(strftime("%Y-%m-%d
%H:%M:%S"),str(temp)))
#graph(temp)
print("\n')

```

ANEXO E

Trabajo enviado y aceptado en el IEEE Internacional Conference on Engineering Veracruz 2019.

Implementation of fuzzy controller in different embedded systems

López Riviello J.L., López Riviello E., Hernández Reyes J. A., González Escarpeta C. R.

Maestría en Eficiencia Energética y Energías Renovables.

Departamento de Ingeniería Eléctrica-Electrónica.

Tecnológico Nacional de México Campus Veracruz

Veracruz, México.

ing-riviello@ieee.org, ing_emmanuelrivi@ieee.org, jantoniohr@gmail.com, carlosge@hotmail.com

Abstract— This article presents the implementation of different embedded systems in the development of fuzzy logic for its application in the control of a furnace with measures to 100 °C for the optimization of preheating in surface mounted devices. Fuzzy logic accepts multiple variables that the device must process efficiently, so it is very important to make the appropriate choice of the embedded system to achieve adaptability to any problem presented. In addition, the observations of the development of fuzzy logic within the program are discussed.

Keywords— *Fuzzy Logic, Fuzzy control, Embedded Systems, Arduino, Tiva C, Raspberry Pi.*

I. INTRODUCTION

Currently in the 4.0 industries and research centers in Mexico are in development for the application of emerging technologies, although the process is gradual, is applied by 65% in the industry. This applied technology is known as Embedded Systems, since they offer a greater adaptability with the existing technology, generating a support for the efficiency in certain processes. These platforms are encapsulated in large-scale integration devices, predominantly surface-mounted devices, with which the physical size of such systems can be reduced [1-4].

Embedded systems are mainly platforms designed to improve the response capacity and quality of the information obtained in a specific process. In this way in the industry and in the technological sector, they are in the process of being developed for the application of improvements in various processes.

In the dynamic control systems called by artificial intelligence such as fuzzy logic, neural networks, genetic algorithms, among other control systems; There is a perception of adaptability that is currently necessary, in which embedded systems offer the optimization and reduction of computer resources, generating stability in the control systems.

The control systems applied in various devices with fuzzy logic, perform nonlinear work operations in any production or control process.

Fuzzy logic is a control system that aims to mimic the linguistics of the human being, based on prior knowledge and inference, that by obtaining these two steps the what and the why of any process to be performed in any task will be

determined, with which, any process that does not necessarily use a mathematical model is detailed, so it becomes adaptable in any field of application. Having to the diffuse logic at present in a position of preponderance in diverse applications.

The main objective of this work is to offer a wide panorama of the advantages and disadvantages of the operation of the embedded systems giving a balance of the results obtained in each device that are currently developed.

II. EMBEDDED SYSTEMS

Embedded systems are products used by Artificial Intelligence and have become common in almost all industrial consumer markets [2]. The fundamental role of the embedded system is not the transformation of data, but the interaction of the world with the human being. The fields of application are varied, and by joining together multiple synergies are achieved and an almost perfect execution is obtained. The application of the embedded systems in technological equipment is: industrial machines, security systems, robotic equipment, means of transport, among other equipment. So, it generates a symbiotic relationship with different disciplines of engineering.

A. Characteristics

The considerations chosen for this work are mainly three special attributes for the development of embedded software, which are: limitations of resources in the processing and the real-time response in the application of fuzzy logic.

B. Development challenges

The development of the program generates high challenges since in each embedded system it has its singularity in the dedicated programming language, although the current devices focus the rationale in the communication with the human machine interface (HMI), with this it restricts certain physical operations, for control systems. There are four major challenges in its developments: in the planning of the embedded software, development process, in the establishment of a methodology, in the assurance of embedded software quality, in the design tools [6].

C. Choice of embedded systems

The embedded systems chosen in this publication are the following: