



UNIVERSIDAD TÉCNICA DE COTOPAXI

DIRECCIÓN DE POSGRADO

MAESTRÍA EN ELECTROMECAÁNICA

MODALIDAD: PROYECTO DE INVESTIGACIÓN

Título:

Análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura.

Trabajo de titulación previo a la obtención del título de Magíster en Electromecánica

Autor:

Ing. Juan Carlos Remache Coyago

Tutor:

MSc. Byron Paúl Corrales Bastidas

LATACUNGA – ECUADOR

2023

AVAL DEL TUTOR

“En mi calidad de Tutor del Trabajo de Titulación “Análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura” presentado por Remache Coyago Juan Carlos para optar por el título magíster en Electromecánica.

CERTIFICO

Que dicho trabajo de investigación ha sido revisado en todas sus partes y se considera que reúne los requisitos y méritos suficientes para ser sometido a la presentación para la valoración por parte del Tribunal de Lectores que se designe y su exposición y defensa pública.

Latacunga, febrero 28, 2023



M.Sc. Byron Paúl Corrales Bastidas

CC: 0502347768

AVAL DEL TRIBUNAL

El trabajo de Titulación “Análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura”, ha sido revisado, aprobado y autorizado su impresión y empastado, previo a la obtención del título de Magíster en Electromecánica, el presente trabajo reúne los requisitos de fondo y forma para que el estudiante pueda presentarse a la exposición y defensa.

Latacunga, febrero 28, 2023



.....
PhD. Enrique Torres Tamayo

CC: 1757121940

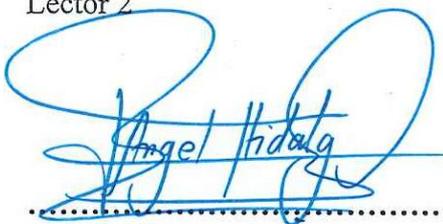
Presidente del tribunal



.....
Ing. MSc. Cristian Fabian Gallardo Molina

CC: 0502847692

Lector 2



.....
Ing. MSc. Ángel Guillermo Hidalgo Oñate

CC: 0503257404

Lector 3

DEDICATORIA

El presente trabajo va dedicado a Dios, fuente de sabiduría.

A mi papá Lorenzo en el cielo, a mi madre Gloria, a quienes de su esfuerzo se refleja lo que soy.

A mi esposa Johanna, mis hijas Arianna y Liz, mis motores e inspiración de vida.

A mi familia en general, por apoyarme sin condición.

A mis amigos y compañeros de trabajo, por estar pendientes y traer el combo estudiantil para que no decaiga este educando.

Va dedicado también a quien lo lea y espero aportar en su crecimiento.

Juan Carlos.

AGRADECIMIENTO

Siempre con gratitud a mi empresa Repsol, luego Petrolia por apoyar a sus trabajadores en el crecimiento continuo.

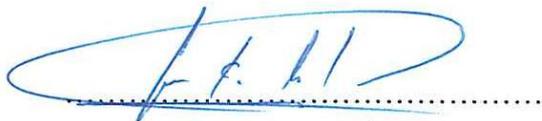
Mi agradecimiento a la Universidad Técnica de Cotopaxi, a la Dirección de Posgrados, y sus destacados profesores.

Juan Carlos Remache Coyago.

RESPONSABILIDAD DE AUTORÍA

Quien suscribe, declara que asume la autoría de los contenidos y los resultados obtenidos en el presente trabajo de titulación.

Latacunga, febrero 28, 2023



Ing. Juan Carlos Remache Coyago

CC: 1712482163

RENUNCIA DE DERECHOS

Quien suscribe, cede los derechos de autoría intelectual total y/o parcial del presente trabajo de titulación a la Universidad Técnica de Cotopaxi.

Latacunga, febrero 28, 2023



Ing. Juan Carlos Remache Coyago

CC: 1712482163

AVAL DEL PRESIDENTE DEL TRIBUNAL

Quien suscribe, declara que el presente Trabajo de Titulación “Análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura”, contiene las correcciones a las observaciones realizadas por los lectores en sesión científica del tribunal.

Latacunga, febrero 28, 2023



PhD. Enrique Torres Tamayo

CC: 1757121940

UNIVERSIDAD TÉCNICA DE COTOPAXI
DIRECCIÓN DE POSGRADO

MAESTRÍA EN ELECTROMECAÁNICA

Título: Análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura.

Autor: Ing. Remache Coyago Juan Carlos

Tutor: MSc. Corrales Bastidas Byron Paúl

RESUMEN

La temperatura es una de las variables siempre presentes en la industria y cuando se pretende llevar a cabo una práctica de control, se halla con dos dificultades, encontrar un modelo matemático de la planta que presente cierta exactitud y probar él o los controladores diseñados con la planta físicamente disponible a fin de estabilizar la variable de salida. Actualmente también el hardware de los PLC industriales permite software con instrucciones para implementar un tipo de control avanzado, sin embargo, el costo de ellos es elevado. Las nuevas tecnologías han permitido disponer en el mercado hardware libre de bajo costo, pero la implementación del algoritmo para control digital en ellos aún está en desarrollo y se tienen algunos fallos en el despliegue de resultados en varios casos. El presente proyecto de titulación implementa un proceso embebido de temperatura basado en las características del transistor TIP31C y el sensor LM35, el controlador empleado es en base a la tarjeta Arduino en donde se desarrollan los algoritmos para el control digital PID y Fuzzy. Se emplea la herramienta Simulink como software avanzado para la identificación del modelo matemático de la planta, la recolección y almacenamiento de datos, y mediante las librerías se elabora una pantalla con bloques de programación para el despliegue gráfico de variables del proceso y el cálculo de índices del desempeño basados en el error. Al finalizar la investigación se dispone de un proceso embebido de temperatura de bajo costo cuyo controlador tiene una programación base que permite experimentar con distintos algoritmos y sus parámetros de diseño para el control digital. El enlace con Simulink permite emplear herramientas actualizadas que fácilmente ayudan a hallar el modelo matemático, desplegar las tendencias de las variables del proceso y tener un acertado criterio de desempeño sobre un determinado control digital en aplicación.

PALABRAS CLAVE: Sistema embebido, controlador electrónico, respuesta transitoria, estado estacionario, índices de desempeño.

UNIVERSIDAD TÉCNICA DE COTOPAXI
POSTGRADUATE DEPARTMENT
MASTER'S DEGREE IN ELECTROMECHANICS

Title: Controllers application comparative analysis in an embedded temperature process.

Author: Eng. Juan Carlos Remache Coyago

Tutor: MSc. Corrales Bastidas Byron Paúl

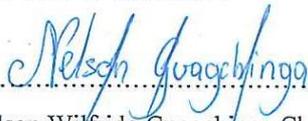
ABSTRACT

Temperature is one of the variables always present in the industry, and when it comes to carry out a control practice, two difficulties are detected. Finding a mathematical model of the plant that has some accuracy and test the controller or controllers designed with the plant physically available in order to stabilize the output variable. Nowadays the industrial PLC hardware also allows a software with instructions to implement a type of advanced control, however, its cost is expensive. New technologies have permitted to provide a low-cost opensource hardware on the market, but the implementation of the algorithm for its digital control is still under development and there are some failures in the deployment of results in several cases. The current degree project implements an embedded temperature process based on the characteristics of the TIP31C transistor and the LM35 sensor, the controller used is based on the Arduino card where the algorithms for PID and Fuzzy digital control are developed. The Simulink tool is used as an advanced software for the mathematical model identification of the plant, data collection and storage; and through the libraries, a screen with programming blocks is developed for the graphic display of process variables and calculation of performance indicators based on the error. At the end of the research, a low-cost embedded temperature process is provided whose controller has a programming base that allows experimenting with different algorithms and their design parameters for digital control. The connection with Simulink allows to use updated tools that easily help to find the mathematical model, displays the trends of the process variables and have an accurate performance criteria on a given digital control application.

KEY WORDS: Embedded system, electronic controller, transient response, steady state, performance indicator.

Nelson Wilfrido Guagchinga Chicaiza con cédula de identidad número: 0503246415 magister en la Enseñanza del Idioma Inglés como Lengua Extranjera con número de registro de la SENESCYT: 1010-2019-2041252; **CERTIFICO** haber revisado y aprobado la traducción al idioma inglés del resumen del trabajo de investigación con el título: "Análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura" de Juan Carlos Remache Coyago, aspirante a magister en Electromecánica.

Latacunga, febrero 28, 2023



MSc. Nelson Wilfrido Guagchinga Chicaiza
CC: 050324641-5

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I	8
FUNDAMENTACIÓN TEÓRICA	8
1.1 Introducción y objetivo del capítulo	8
1.2 Antecedentes de la investigación o fundamentación del estado del arte	8
1.3 Fundamentación teórica	9
1.3.1 Proceso Industrial.....	9
1.3.2 Controladores de proceso	10
1.3.3 Teorías de control.....	11
1.3.4 Tipos de controladores	16
1.3.5 Criterios de desempeño	25
1.3.6 Análisis de respuesta transitoria y error en estado estacionario.....	27
1.3.7 Sistemas embebidos	28
1.3.8 Hardware de código abierto	30
1.3.9 Herramientas de análisis computacional.....	31
1.3.10 Transferencia de calor	32
1.3.11 Modelo eléctrico análogo al proceso de disipación térmica	34
1.4 Descripción del objeto de estudio	35
1.5 Conclusiones	35
CAPÍTULO II	37
MATERIALES Y MÉTODOS	37
2.1 Introducción y objetivo del capítulo	37
2.2 Metodología de Investigación	37
2.3 Diseño del proceso embebido	38
2.3.1 Metodología de diseño	38

2.3.2	Requerimientos y propuesta funcional del sistema embebido	39
2.3.3	Selección de componentes	40
2.3.4	Diseño, modelación y simulación	43
2.3.5	Construcción física.....	57
2.4	Conclusiones	58
CAPÍTULO III	59
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....		59
3.1	Introducción y objetivo del capítulo	59
3.2	Obtención de la función de transferencia.....	59
3.3	Implementación de controladores PID.....	59
3.3.1	Métodos de sintonización.....	61
3.3.2	Resultados Obtenidos.....	62
3.3.3	Análisis usando criterios para estado transitorio y estacionario	64
3.3.4	Análisis usando índices de desempeño	65
3.4	Implementación de controlador con Lógica Difusa.....	66
3.4.1	Resultados Obtenidos.....	71
CAPÍTULO IV	76
CONCLUSIONES Y RECOMENDACIONES.....		76
4.1	Conclusiones	76
4.2	Recomendaciones.....	78
BIBLIOGRAFÍA	79
ANEXOS	83
ANEXO 1. VALIDACIÓN DE DATOS	84
A1.1	Registro de verificación de calibración del sensor LM35.....	84
A1.2	Registro de verificación de calibración del software de análisis:	85
A1.3	Certificado de calibración del instrumento patrón de medida.....	86

A1.3 Registro fotográfico.....	87
ANEXO 2. FUNCIÓN DE TRANSFERENCIA USANDO MATLAB	88
A2.1 Obtención de datos de la planta en lazo abierto	88
A2.2 Separación de datos de una tabla almacenada en Matlab.....	89
A2.3 Identificación de la función de transferencia en Matlab	90
ANEXO 3. IMPLEMENTACIÓN DE LA PANTALLA DE CRISTAL LIQUIDO LCD CON I2C	99
A3.1 Implementación física	99
A3.1 Implementación lógica	101
ANEXO 4. CONSTRUCCIÓN FÍSICA	102
A4.1 Listado de material	102
A4.2 Diagramas de cajas y tarjetas	103
ANEXO 5. DETERMINACIÓN DEL PUNTO DE TRABAJO DEL TIP31C .	106
A5.1 Cálculo de la recta de carga y el punto Q del TIP31C	106

ÍNDICE DE TABLAS

Tabla I. Sistema de tareas con relación a los objetivos específicos.....	3
Tabla II. Variable independiente: Controlador	5
Tabla III. Variable independiente: Sistema embebido.....	6
Tabla IV. Variable dependiente: Respuesta de temperatura	7
Tabla V. Fórmulas de sintonización de Ziegler Nichols.....	21
Tabla VI. Métodos de sintonización de un controlador PID	21
Tabla VII. Selección componentes para el sistema embebido.....	41
Tabla VIII. Señales por acoplar en sistema.....	43
Tabla IX. Características TIP31C	45
Tabla X. Valores de diseño del sistema embebido.	47
Tabla XI. Asignación de canales físicos en tarjeta Arduino.	53
Tabla XII. Valores calculados de sintonía para PID – análogos.....	62
Tabla XIII. Valores corregidos de sintonía para PID – discretos.	62
Tabla XIV. Resultados de aplicación de métodos de sintonía para control PID. .	64
Tabla XV. Resultados índices de desempeño para PID.....	65
Tabla XVI. Definición de variables lingüísticas.....	68
Tabla XVII. Reglas de inferencia.....	70
Tabla XVIII. Resultados estado estable / transitorio para control Fuzzy	72
Tabla XIX. Resultados de los índices de desempeño PID vs. Fuzzy.....	73
Tabla XX. Resultados del uso de memoria para los controladores PID y Fuzzy .	74
Tabla XXI. Características de distintos modelos de procesadores	74
Tabla XXII. Conexión LCD a I2C en Arduino UNO.	100
Tabla XXIII. Listado de material y precio del sistema embebido.	102
Tabla XXIV. Valores calculados de I_c vs. VCE del transistor TIP31C.	106

ÍNDICE DE FIGURAS

Fig. 1. Lazo abierto de control	10
Fig. 2. Lazo cerrado de control	10
Fig. 3. Esquema básico para la representación de la función de red.....	12
Fig. 4. Diagrama de bloques del proceso en el dominio del tiempo	13
Fig. 5. Representación del proceso en espacio de estados en tiempo discreto.....	14
Fig. 6. Transformada Z, representación gráfica	15
Fig. 7. Control todo-nada	17
Fig. 8. Control PID	18
Fig. 9. Respuesta del sistema en lazo abierto ante entrada escalón unitario.....	19
Fig. 10. Respuesta del sistema en lazo abierto ante entrada escalón unitario y medición de los parámetros experimentales.. ..	20
Fig. 11. Ejemplo de establecimiento de reglas en control difuso.....	23
Fig. 12. Ejemplo de un controlador de lógica difusa estático	24
Fig. 13. Ejemplo de función de pertenencia triangular	25
Fig. 14. Criterios de desempeño	27
Fig. 15. Parámetros de respuesta transitoria	28
Fig. 16. Sistemas embebidos	29
Fig. 17. Microcontrolador Arduino UNO	30
Fig. 18. Ambiente de trabajo de Matlab + Simulink.....	31
Fig. 19. Mecanismos de transferencia de calor	32
Fig. 20. Modelo eléctrico análogo al proceso de disipación térmica	34
Fig. 21. Proceso de la Investigación Cuantitativa	38
Fig. 22. Método de diseño para sistemas embebidos	39

Fig. 23. Partes constitutivas del sistema embebido.....	40
Fig. 24. Diseño del sistema embebido	42
Fig. 25. Gráfica de potencia de disipación del TIP31C	44
Fig. 26. Circuito base del sistema embebido	46
Fig. 27. Circuito del TIP31C del sistema embebido	47
Fig. 28. Acondicionamiento señal del LM35 a salida estándar 1 a 5VDC	48
Fig. 29. Amplificador no inversor.....	48
Fig. 30. Circuito sumador.....	49
Fig. 31. Modelamiento de la Planta y simulación en Proteus	50
Fig. 32. Tareas base del controlador Arduino UNO	51
Fig. 34. Modelamiento de la tarjeta Arduino en Proteus	54
Fig. 35. Esquema de despliegue de datos y análisis.....	55
Fig. 36. Librería usada en Simulink para interactuar con Arduino	56
Fig. 37. Ambiente de análisis de datos en Simulink	56
Fig. 38. Sintonización de PID mediante varios métodos.	63
Fig. 39. Diagrama de flujo del controlador PI-Fuzzy.	67
Fig. 40. Función de membresía del error	69
Fig. 41. Función de membresía de la derivada del error.....	69
Fig. 42. Función de membresía de la salida	70
Fig. 43. Respuesta del sistema embebido de temperatura con control difuso.....	72
Fig. 44. Validación de medición y calibración de temperatura.....	87
Fig. 45. Instrumento patrón: Cámara termográfica Fluke Ti32	87
Fig. 46. Imágenes captadas por la cámara termográfica.	87
Fig. 47. Configuración para almacenamiento de datos en Scope.	88
Fig. 48. Almacenamiento de datos en Workspace	89

Fig. 49. Carga del registro de datos.....	89
Fig. 50. Extracción de valores del archivo Datalogging	90
Fig. 51. Comprobación de datos almacenados en tabla de Matlab.	90
Fig. 52. Utilización de comando “systemIdentification”.....	91
Fig. 53. Menú de la herramienta “systemIdentification”.....	91
Fig. 54. Selección de la importación de datos.....	91
Fig. 55. Selección datos de setpoint y de proceso.....	92
Fig. 56. Ambiente de selección de importación de datos.....	93
Fig. 57. Selección de rangos en ventana Operations.....	93
Fig. 58. Selección del área de datos.....	94
Fig. 59. Selección del área de datos (continuación 1).....	94
Fig. 60. Selección del área de datos (continuación 2).....	95
Fig. 61. Selección del área de datos (continuación 3).....	96
Fig. 62. Selección del área de datos (continuación 4).....	96
Fig. 63. Selección del área de datos (continuación 5).....	97
Fig. 64. Obtención del modelo matemático.....	97
Fig. 65. Validación del modelo.....	98
Fig. 66. Estructura del adaptador LCD a I2C, PCF8574T	99
Fig. 67. Comando para incluir librerías en Arduino IDE.....	100
Fig. 68. Cableado display LCD con Arduino usando protocolo I2C	100
Fig. 69. Placa base del circuito de la planta	103
Fig. 70. Diseño de las carátulas de las cajas de conexiones.....	103
Fig. 74. Recta de carga y zona de operación del TIP31C	107

INTRODUCCIÓN

Antecedentes

De acuerdo con lo establecido por el departamento de investigación de la Universidad Técnica de Cotopaxi, la línea de investigación del presente proyecto es Procesos industriales; la sub línea de investigación es Automatización, control y protecciones de sistemas electromecánicos.

El análisis comparativo de la aplicación de controladores en un proceso embebido de temperatura se enmarca en el desarrollo de la Electrónica, automatización y control, permitiendo optimización en procesos industriales.

Planteamiento del problema

En la actualidad, gracias al desarrollo de herramientas informáticas y de modelamiento electrónico, se han desarrollado nuevos sistemas dinámicos. El control de sus variables ha requerido mayor precisión y por lo tanto nuevas técnicas de control han aparecido, sin embargo, éstas se las ha desarrollado por separado, de acuerdo con la necesidad del sistema en estudio. Existe limitada información referente al análisis comparativo actualizado para las nuevas técnicas y filosofías de control, pues la mayoría de bibliografía se ha centrado en mostrar resultados para controladores clásicos.

Los técnicos y profesionales que trabajan en la industria no disponen de información actualizada de primera mano que les permita entender resultados de forma clara, comparar comportamientos en un mismo sistema u homogenizar resultados con una misma variable.

La temperatura es una de las variables siempre presentes en la industria y cuando se pretende llevar a cabo una práctica de control, se hallan dos dificultades, encontrar un modelo matemático de la planta que presente cierta exactitud y probar él o los controladores diseñados con la planta físicamente disponible [1].

Es necesario disponer de un proceso de temperatura embebido en el cual se pueda analizar el correcto desarrollo de controladores a fin de estabilizar la variable de salida, así la respuesta de temperatura podría ser evaluada en su tiempo de estabilización, sus valores máximos y mínimos alcanzados en estado dinámico, y su comportamiento en estado estacionario.

Objeto y campo de acción

Como objeto de la investigación se ha tomado la aplicación de distintos controladores industriales en procesos de temperatura.

El campo de investigación específico es para un proceso embebido de temperatura, diseñado de manera didáctica y enlazado a herramientas de análisis mediante software.

Objetivo General

Realizar un análisis comparativo mediante la aplicación de distintos tipos de controladores en un proceso embebido de temperatura a fin de determinar sus parámetros más relevantes.

Objetivos específicos

- Investigar los tipos de controladores aplicados a procesos de temperatura.
- Diseñar un módulo embebido de temperatura.
- Implementar diferentes tipos de controladores aplicados a un proceso de temperatura.
- Validar los resultados y mediante la comparación entre las respuestas obtenidas de los controladores.

Sistemas de tareas con relación a los objetivos específicos

Se elabora la tabla de tareas para poder llevar a cabo los objetivos, se describen los resultados esperados y las actividades:

Tabla I. Sistema de tareas con relación a los objetivos específicos.

Objetivos específicos	Actividad (tareas)	Resultado de la actividad	Descripción de la actividad
1	Recopilación de información	Selección de controladores a implementar.	Investigación de distintos tipos de controladores
2	Diseño módulo embebido	Selección de un prototipo embebido para realizar la experimentación	Investigación y diseño del sistema controlado de temperatura: selección del actuador, elemento sensor, hardware a usar, adecuada visualización de resultados.
3	Construcción del módulo, configuración de parámetros.	Disponer físicamente del sistema de temperatura	Elaboración de placa electrónica. Ensamble de elementos. Parametrización del sistema.
4	Obtención, validación y presentación de datos	Presentar de manera objetiva los resultados obtenidos.	Pruebas operativas Adquisición de datos Elaboración de informe

Justificación y/o importancia de la investigación

La presente investigación es requisito previo a la obtención del título de Magister en Electromecánica y plantea una respuesta al problema de la falta de un análisis en distintos tipos de controladores aplicados a un proceso de temperatura. Así pues, presenta un módulo embebido que sirve de base para implementar algunos tipos de controladores y utiliza herramientas de análisis para presentar su desempeño.

El estudio además está pensado como un módulo didáctico de entrenamiento, por lo cual sus partes constitutivas permiten flexibilidad en el momento de ensamblar y analizar resultados. El proceso embebido propuesto es robusto para manipulación de estudiantes, el controlador a emplearse estará basado en hardware libre de código abierto, permitiendo adicionar una variedad de algoritmos. También dispondrá la opción de enlazar a distintos softwares de ingeniería que permitan profundizar el estudio, lo que se puede también aplicar técnicas de modelamiento y simulación, además de usar sus poderosas bondades de análisis.

El módulo implementado beneficiará a los alumnos de últimos semestres de carreras de Ingeniería y profesionales interesados en comprender y obtener óptimo control en procesos de temperatura.

En la industria la aplicación es importante pues existe una diversidad de procesos industriales con control de temperatura: controles de calefacción, planchas industriales, sistemas de enfriamiento en motores y generadores, control de energía en turbinas, intercambiadores de calor, entre otras.

La manipulación de la variable temperatura para mantener estables dichos procesos puede llegar a ser muy complejo, la selección del controlador idóneo y eficiente depende del criterio de quien lo manipule o diseñe. Los resultados de un análisis comparativo de controladores permiten un entendimiento claro y se constituyen en una herramienta clave para mejorar el rendimiento del proceso. Las empresas pueden perder competitividad al no trabajar dentro de parámetros óptimos.

Hipótesis

El desarrollo de un análisis en diferentes controladores electrónicos digitales aplicados a un proceso embebido de temperatura, brindará información relevante de las características de desempeño de los controladores.

Variable independiente

Aplicación de controladores. Sistema embebido

Variable dependiente

Respuesta de temperatura.

Operacionalización de variables

Se procede a generar una matriz en las que se define los conceptos, categorías y unidades de medida que servirán para tomar datos y elaboración de cálculos.

Tabla II. Variable independiente: Controlador

Concepto	Categoría	Indicadores	Unidad	Técnicas	Instrumentos
Controlador es un circuito eléctrico o componente electrónico, destinado a manipular una variable eléctrica.	Tipos de Control: PID Difuso	Complejidad de programación	Tamaño del algoritmo (kbytes)	Medición	Software de programación
		Criterios de desempeño	Cálculo del error	Cálculo	Fórmulas de evaluación del error
		Complejidad matemática del principio de operación	Uso de herramientas matemáticas, Ecuaciones lineales, no lineales	Observación	Ecuaciones

Tabla III. Variable independiente: Sistema embebido

Concepto	Categoría	Indicadores	Unidad	Técnicas	Instrumentos
Sistema embebido es un sistema diseñado para un propósito específico	Selección de proceso de temperatura	Dispositivos electrónicos asociados a generar temperatura	Rango de Temperatura (°C)	Selección	Catálogos
			Dispositivos adicionales requeridos	Selección	Catálogos
		Costo	Valoración (USD \$)	Valoración	Catálogos
	Implementación	Propiedades características del sistema	Temperatura (°C)	Medición	Cámara termográfica
			Potencia (W)	Cálculo	Ecuaciones

Tabla IV. Variable dependiente: Respuesta de temperatura

Concepto	Categoría	Indicadores	Unidad	Técnicas	Instrumentos
Respuesta de temperatura es la manera en la cual se comporta la salida del sistema en el tiempo.	Respuesta en estado transitorio	Tiempo de retardo	Tiempo(ms)	Medición	Herramienta de análisis matemático
		Tiempo de crecimiento	Tiempo(ms)	Medición	Herramienta de análisis matemático
		Tiempo de pico	Tiempo(ms)	Medición	Herramienta de análisis matemático
		Sobre impulso máximo	Porcentaje <40%	Medición	Herramienta de análisis matemático
	Respuesta en estado estacionario	Tiempo de estabilización	Tiempo (ms) $3 < t_s < 8\tau$ cte del sistema	Medición	Herramienta de análisis matemático
		Error en estado estacionario	Porcentaje < 2%	Medición	Herramienta de análisis matemático

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción y objetivo del capítulo

En el presente capítulo se revisan los conceptos teóricos necesarios a utilizarse en el desarrollo del proyecto. En primer lugar, se explican los tipos de controladores y los métodos generales para su correcta sintonización, luego se presentan los criterios para evaluar su desempeño. Se explican los procesos de temperatura aplicados en la industria y finalmente se analizan las distintas herramientas para su implementación y análisis.

Todo ello servirá de base para realizar el análisis comparativo de controladores en un sistema embebido de temperatura.

1.2 Antecedentes de la investigación o fundamentación del estado del arte

En la actualidad, nuevas técnicas de control han aparecido gracias al desarrollo de herramientas informáticas y de modelamiento electrónico. Recientes estudios en sus conclusiones quedan en la intriga de experimentar frente a nuevas filosofías de control, por ejemplo, C. Muñoz al término de su análisis comparativo PI versus Fuzzy dice “Se debería comparar las pruebas del controlador Fuzzy con otro controlador moderno. Se sugiere usar un modelo de control predictivo” [2].

E. Basantes en su fundamentación teórica de tesis cita “La falta de conocimientos en herramientas digitales limita el monitoreo y control de procesos industriales” [3]. Estudios comparativos entre controladores se vienen realizando desde hace algunos años y en ellos se recalca que el éxito de la implementación de un control de proceso

depende también del buen entendimiento y experiencia de quien los manipula o diseña [4], A. Ferreyra en 1999, en su comparación de PID versus Fuzzy indica que “para un control PID se puede decir que presenta una buena precisión cuando es bien ajustado” [5]. Para el 2018 C. Saeteros concluye que la tendencia se mantiene al decir “La lógica Fuzzy es programada en base a la experiencia y lógica del programador. El controlador PID funciona bien con niveles cercanos al nivel para el cual se obtuvo los parámetros, perdiendo efectividad con valores de consigna altos, en los que se tiene que ajustar los parámetros del controlador, generando inconvenientes al diseñador” [6].

Es muy interesante el estudio de J. G. Álvarez en Colombia cuya conclusiones reflejan el estado en industrias actuales: “A partir de visitas realizadas a empresas en la región del Valle del Cauca se pudo evidenciar que aún se presenta un gran desconocimiento por los técnicos de las empresas en metodologías de sintonía de controladores PID, detectándose aún problemas comunes en los controladores PID tales como: alta variabilidad, bajo rendimiento, poca eficiencia en los lazos de control, lo cual muestra que abordar temas de la ingeniería en capacitación y transferencia son requeridos buscando mejorar la competitividad y productividad de las empresas. De igual forma se pudo determinar que técnicas avanzadas como controladores inteligentes y robustos no son aún usados en la industria, principalmente debido al desconocimiento en la sintonía, implementación y mantenimiento de estos controles por parte de ingeniería y operación. Implementar, validar y documentar los métodos no usados en la industria en plantas reales es otro reto para la ingeniería y la academia” [7].

1.3 Fundamentación teórica

1.3.1 Proceso Industrial

Es un conjunto de operaciones específicamente diseñadas para convertir una materia prima en un producto final, por lo cual se tienen que aplicar distintas tareas que logran dicha transformación. Ogata define un proceso como a cualquier operación que deba controlarse [8], es decir tomar acciones dirigidas sistemáticamente para obtener una respuesta deseada.

1.3.2 Controladores de proceso

Son aquellos en que su salida mantiene una relación prefijada con alguna entrada de referencia, constantemente se está comparando y su diferencia sirve como medida de control. Para que todo esto sea posible es necesario concebir un sistema de control que incluye elementos definidos como son: el elemento de medida, elemento de transmisión, el registrador y un actuador o elemento final de control. Todo el conjunto se denomina lazo de control y puede ser abierto o cerrado [9].

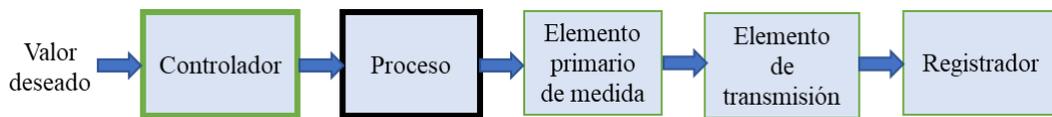


Fig. 1. Lazo abierto de control. [9]

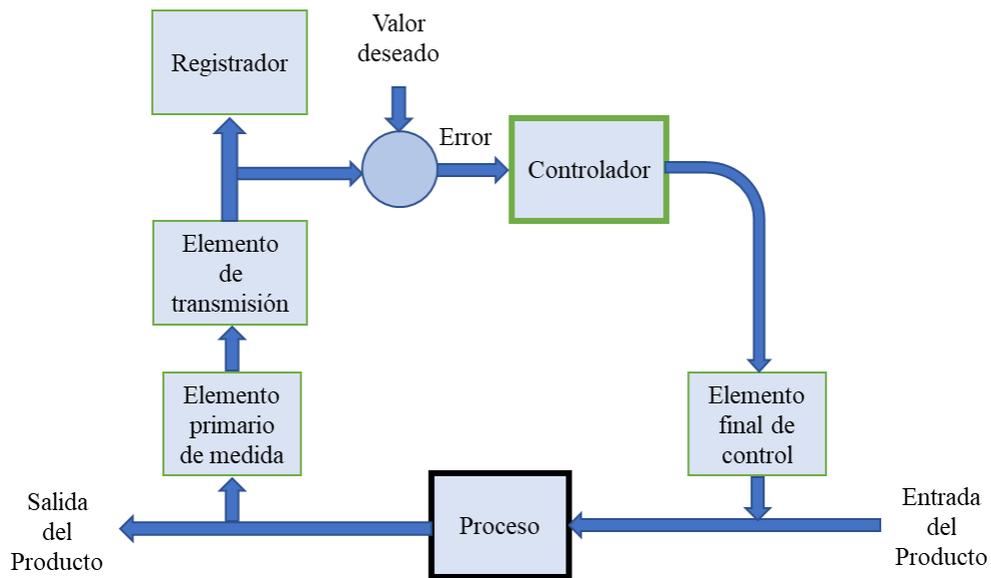


Fig. 2. Lazo cerrado de control. [9]

Como ejemplo lazo abierto es el control de temperatura de un clásico horno de cocina, en el cual se coloca el valor deseado y un determinado tiempo para la cocción, no se evalúa la señal de salida que es el alimento horneado, sino hasta que termina el ciclo. Un ejemplo de lazo cerrado es un control de temperatura del aire acondicionado, donde un sensor realimenta información para mantener el valor deseado. En Fig. 1 y Fig. 2 se muestran la estructura de los dos tipos de lazo.

1.3.3 Teorías de control

Los procesos industriales vienen evolucionando a gran escala y sus sistemas pueden llegar a realizar tareas complejas, teniendo múltiples variables de entrada y salida. Gracias al avance tecnológico nuevos procedimientos y técnicas para análisis y diseño han aparecido. Al momento se tienen dos teorías de control [10], claramente diferenciadas:

Teoría de control clásica:

- Concepto de la función de transferencia.
- Su análisis es en dominio de la frecuencia.
- Uso en sistemas continuos, lineales, invariantes en el tiempo.
- Uso en sistemas de una entrada, una salida.

Teoría de control moderna:

- Concepto de espacios de estado.
- Análisis en dominio del tiempo.
- Su análisis es vectorial matricial.
- Uso en sistemas digitales, lineales o no lineales, variables o invariantes en el tiempo.
- Uso en sistemas multivariable.

Dentro de la comparativa de controladores es necesario diferenciar el tipo de teoría de control que usa, pues puede generar confusión respecto a la terminología presentada. En el caso de procesos de temperatura la mayoría de las respuestas de salida son continuas, lentas en el tiempo y su diseño es meramente clásico, sin embargo, en controles avanzados, como en control de entrega de energía en turbinas, la temperatura no sólo es una parte de las múltiples señales de control, sino que es una variable cuya manipulación inmediata evita daños estructurales del equipo, en estos casos la teoría de control aplicada es moderna.

1.3.3.1 Modelo por función de transferencia

Al requerir modelar algunos procesos industriales, se encontró que estos daban respuestas dinámicas, y se podían representar mediante ecuaciones diferenciales que representan la evolución con el tiempo. Aplicando concepto de Taylor para linealizar ecuaciones diferenciales que no fueran lineales y la herramienta matemáticas conocida como Transformada de Laplace, se puede representar el sistema en tiempo en un nuevo sistema con una nueva variable compleja llamada “s” y cuya representación es más simple.

A continuación, un esquema básico muestra que la señal de salida es igual a la señal de entrada afectada por la red lineal.

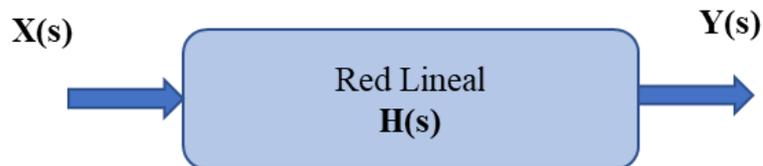


Fig. 3. Esquema básico para la representación de la función de red. Fuente propia.

La señal de salida es $Y(s) = X(s) H(s)$, de donde

$$H(s) = \frac{Y(s)}{X(s)} \quad (1)$$

$H(s)$ es conocida como función de transferencia.

Evaluando $Y(s)=0$ encontramos los llamados “ceros”, con $X(s)=0$ se hallan los “polos”. Al graficar en el plano complejo “s” se puede determinar si una función de transferencia es estable o inestable.

1.3.3.2 Modelo por variables de estado

El control clásico tradicional considera al sistema dinámico lineal bajo representación de funciones de entrada-salida. En la teoría moderna se definen tres tipos de variables: de entrada, salida y variables de estado. El espacio de estado es una forma de representar un sistema dinámico en función de n ecuaciones de

diferencia de primer orden. Las variables de estado son el conjunto más pequeño de variables que determinan el comportamiento dinámico de un sistema [11].

Para un sistema lineal e invariante en el tiempo las ecuaciones desarrolladas ya simplificadas son:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2)$$

En donde A es la matriz de estado del sistema, B la matriz de entrada, C la matriz de salida y D la matriz de transmisión directa.

El diagrama de bloques de un sistema representado en espacios de estado es el siguiente:

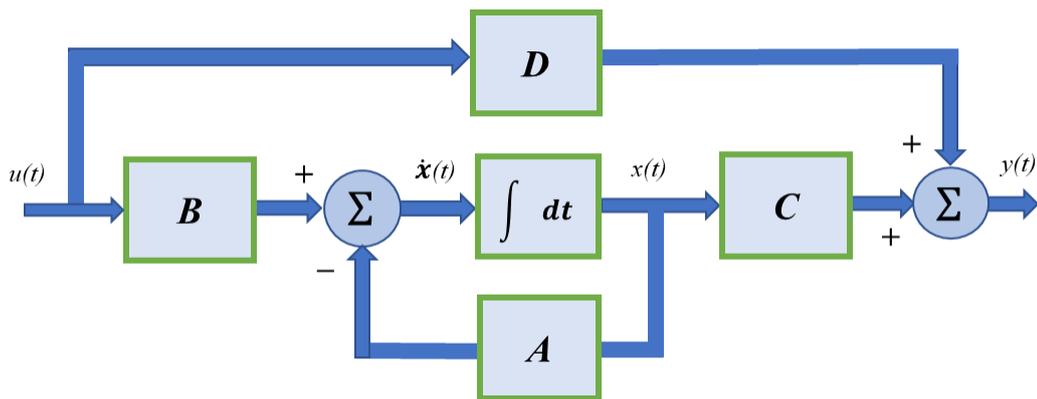


Fig. 4. Diagrama de bloques del proceso en el dominio del tiempo. [11]

Para controladores digitales que trabajan en tiempo discreto, la representación es similar como se muestra en la Fig. 5, la integración se representa por z^{-1} , y la derivada de la variable en adelante de una muestra $x(k + 1)$:

$$\begin{aligned} x(k + 1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (3)$$

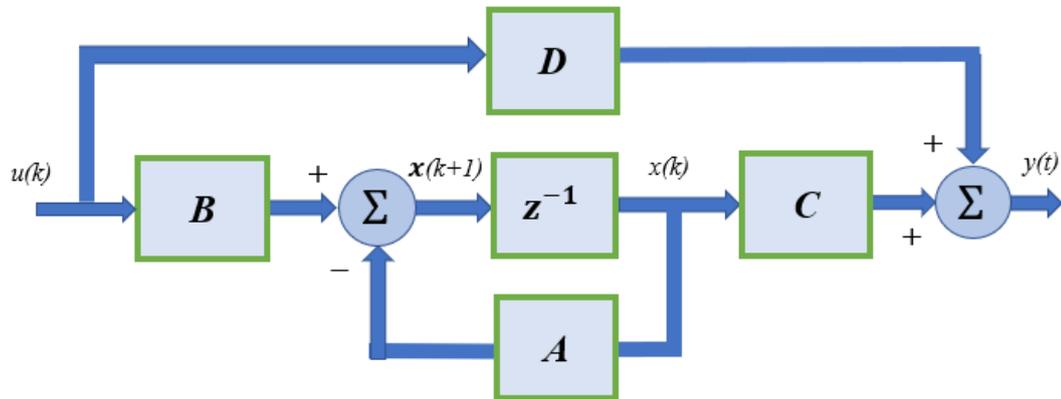


Fig. 5. Representación del proceso en espacio de estados en tiempo discreto. [11]

Ley de Control para Realimentación de estados

La representación matemática de un sistema de control por realimentación de estados viene dada por la fórmula:

$$u(k) = -Kx(k) + r(k) = -[K_1 \quad K_2 \quad \dots \quad K_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + r(k) \quad (4)$$

1.3.3.3 La transformada Z

Mientras que la transformada de Laplace sirve para sistemas continuos, la transformada Z es una estrategia matemática para operar sistemas discretos, cuya definición se muestra:

$$X(z) = Z[x(k)] = \sum_{k=0}^{\infty} x(k) z^{-k} \quad (5)$$

En donde la transformada Z es la sumatoria desde una muestra en la posición cero hasta el infinito, tomadas cada una en un periodo de tiempo T.

$$X(z) = x(0) + x(T)z^{-1} + x(2T)z^{-2} \dots + x(kT)z^{-k} + \dots \quad (6)$$

La representación gráfica del tiempo discreto es la siguiente:

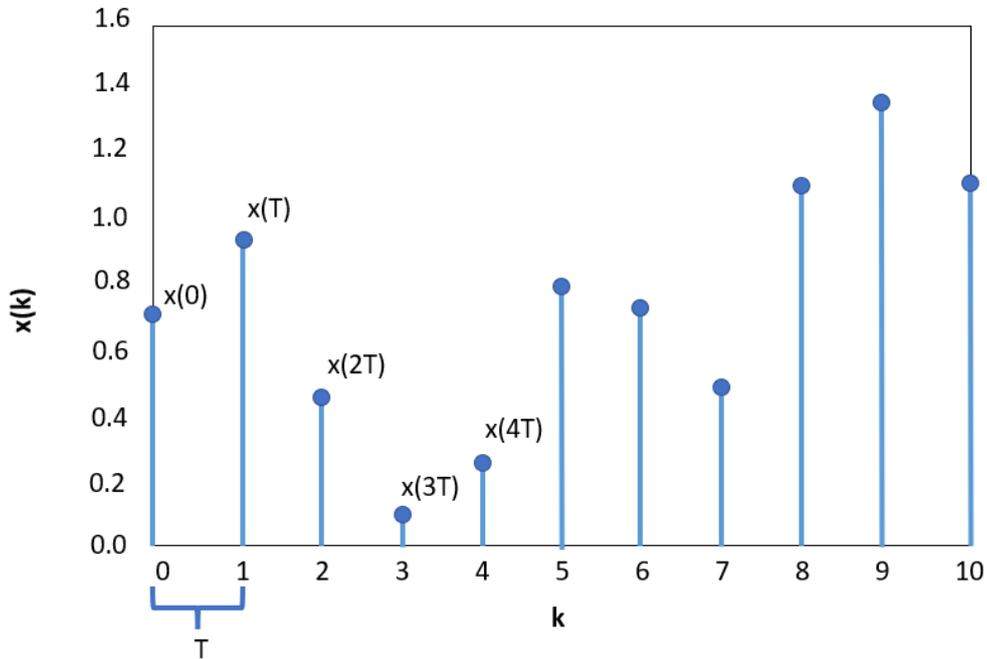


Fig. 6. Transformada Z, representación gráfica. [12]

1.3.3.4 Selección del tiempo de muestreo

Cuando se trabaja dentro de un sistema discreto se debe seleccionar el tiempo de muestreo, de manera que, una vez adquiridos los datos, se pueda recuperar la señal original.

El teorema de Shannon indica que, si la frecuencia de muestreo es suficientemente mayor en comparación con la componente de más alta frecuencia que se incluye en la señal de tiempo continuo, las características de amplitud de la señal de tiempo continuo se pueden preservar en la envolvente de la señal muestreada.

La frecuencia de muestreo se define mediante la expresión:

$$\omega_s = \frac{2\pi}{T_s} \quad (7)$$

Donde,

ω_s es la frecuencia de muestreo, T_s es el periodo de muestreo, y el término ω_c es la componente de más alta frecuencia.

El criterio en general aplicado a sistemas de control [12] se aplica a:

Sistemas en lazo abierto:

$$\omega_s \geq 2\omega_c \quad (8)$$

Sistemas de lazo cerrado oscilatorias:

$$8\omega_c \leq 2\omega_s \leq 12\omega_s \quad (9)$$

Sistemas de lazo cerrado poco oscilatorias:

$$\frac{t_s}{20} \leq T_s \leq \frac{t_s}{10} \quad (10)$$

Donde t_s es el tiempo de establecimiento del sistema en lazo cerrado. Cuando se muestrea una señal sin tener en cuenta la condición del Teorema de Shannon[12] se produce una confusión de frecuencia conocida como Aliasing.

1.3.3.5 Velocidad de la CPU del procesador

El CPU procesa muchas instrucciones, entre ellas cálculos de bajo nivel como los aritméticos, de distintos programas cada segundo. La velocidad de reloj mide el número de ciclos que la CPU ejecuta cada segundo, medidos en Megahercios. Un ciclo es un pulso sincronizado por un oscilador interno. A veces varias instrucciones se completan en un solo ciclo de reloj, pero otras toman múltiples ciclos.

1.3.4 Tipos de controladores

Cuando se diseñan procesos se tienen dos características fundamentales que se toman en cuenta en el comportamiento de la variable controlada y son:

- Cómo cambia de valor de la variable controlada al someterse a una perturbación que afecte al proceso.
- En qué tiempo se requiere que la variable alcance el valor requerido, partiendo de una condición específica.

De acuerdo con estas consideraciones se han clasificado algunos tipos de controladores que pueden encajar para un mismo proceso, pero estaría sobre dimensionado su uso, lo cual incide en el tema económico al momento de implementarse, además de la complejidad al momento de manipular.

1.3.4.1 Control todo-nada

Conocido como “on-off”, la acción de control sobre el elemento final de control toma únicamente dos posiciones, 0% y 100% de acción. La salida de producto varía en una banda diferencial o histéresis. En Fig. 7 se observa la forma de respuesta obtenida, normalmente se tiene respuesta oscilante ya que la acción de control se mantiene cíclicamente con pulsos en alto “on” y en bajo “off”.

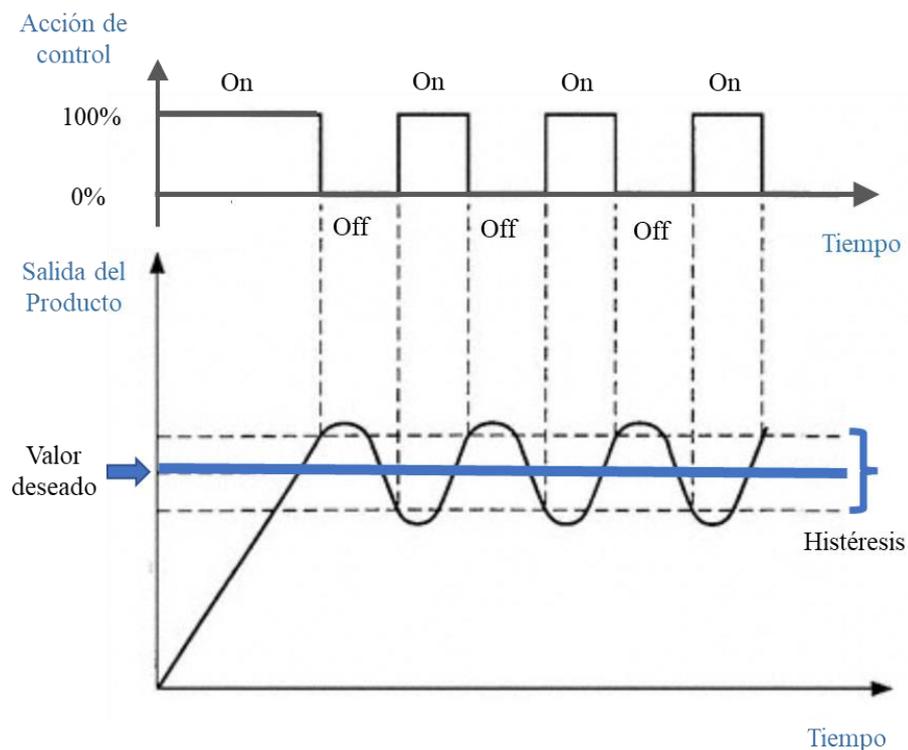


Fig. 7. Control todo-nada. [13]

Este tipo de control opera correctamente en procesos de temperatura de respuesta lenta encontrados por ejemplo en calentadores de aceite, intercambiadores de calor, aplicaciones de aire acondicionado. Su ventaja es que tiene un costo bajo.

1.3.4.2 Control PID

Mediante este tipo de control se pueden realizar tres tipos de acciones: Proporcional, Integral y Derivativa. Analizar en base al error y presentar el gráfico en base al error. La ecuación de los controladores PID está dada por [14]:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (11)$$

La representación gráfica del control se tiene a continuación:

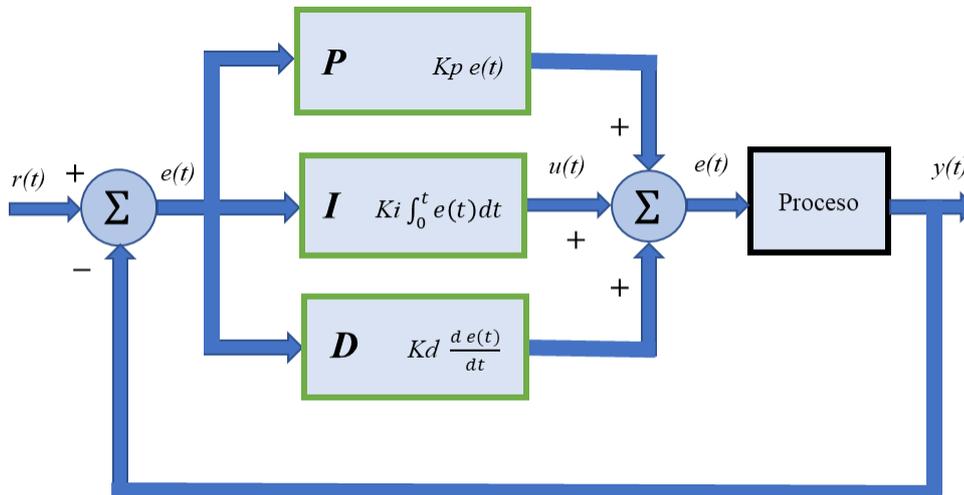


Fig. 8. Control PID.

La acción proporcional varía directamente con la desviación del error respecto a la variable de salida. El efecto es de un amplificador de ganancia constante.

La acción integral es proporcional a la integral del error, es decir suma los errores pasados, permitiendo eliminar el error en régimen permanente.

La acción derivativa es proporcional a la derivada del error, si el error es constante lo discrimina y da como resultado una respuesta de salida rápida, el control actúa cuando hay cambio de valor absoluto en el error.

Métodos de sintonización de un PID

La sintonización de un controlador consiste en encontrar los parámetros óptimos de funcionamiento para obtener una respuesta estable en lazo cerrado.

El diseño de un controlador PID cuando se conoce su modelo matemático es sencillo y hay muchos caminos para conseguirlo.

Dado que un control PID tiene tres parámetros: ganancia K_p , tiempo integral T_i y tiempo derivativo T_d , el ajuste tiene su complejidad debido a que se debe caracterizar a las especificaciones de diseño. Varios procedimientos se han planteado y todos pretenden minimizar la función de error.

Los procedimientos más conocidos son los siguientes:

- Método de oscilaciones sostenidas
- Método de las oscilaciones amortiguadas
- Método de la curva de reacción
- Ziegler-Nichols
- Cohen Coon
- Criterios de desempeño

Como ejemplo Ziegler Nichols es un método en el cual de forma experimental se determinan los parámetros de un control PID, si bien hay varias técnicas, un primer método se obtiene experimentalmente de la respuesta de la planta en lazo abierto ante la entrada de un escalón unitario como se indica en la Fig. 9.

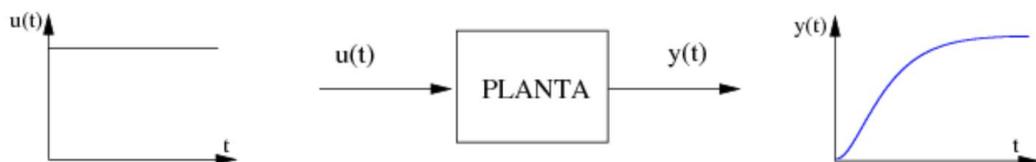


Fig. 9. Respuesta del sistema en lazo abierto ante entrada escalón unitario. [15]

Para aplicar el criterio se varía la respuesta $y(t)$, aumentando progresivamente la ganancia proporcional hasta lograr que oscile el sistema ante un cambio de señal tipo escalón en el valor de consigna. Una respuesta típica con parámetros experimentales se muestra en la Fig. 10, de la misma se tienen valores que únicos que caracterizan a cada sistema, por ejemplo, al inicio se obtiene **el tiempo de retardo L** que nos muestra cuanto demora en reaccionar el sistema, un **periodo T** que evalúa el crecimiento de señal hasta llegar al valor de consigna y el **valor K** que representa la magnitud de salida en la cual se mantiene la respuesta estable ante el escalón aplicado.

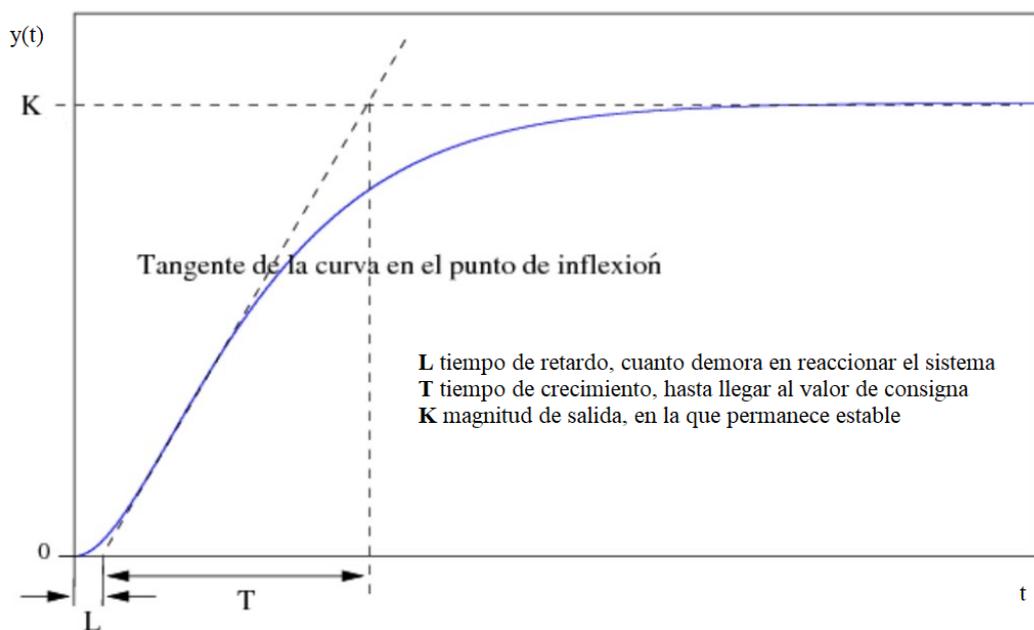


Fig. 10. Respuesta del sistema en lazo abierto ante entrada escalón unitario y medición de los parámetros experimentales. [16]

La forma de respuesta obtenida obedece a tres constantes K , L y T .

La función de transferencia del sistema se caracteriza con dichos parámetros:

$$G(s) = \frac{K}{1 + T * s} * e^{-L*s} \quad (12)$$

Luego para la aplicación del lazo cerrado se usan dichos valores; como ejemplo los valores sugeridos por Ziegler-Nichols para los parámetros de este PID son:

Tabla V. Fórmulas de sintonización de Ziegler Nichols.

Controlador	Parámetro	Fórmula
PID	Constante proporcional, Kp	$1.2 * T/(K*L)$
	Tiempo constante de integración, Ti	$2 * L$
	Tiempo constante de derivación, Td	$0.5 * L$

Fuente: [10]

Los parámetros de lazo cerrado de un PID obedecen a la fórmula:

$$\frac{U(s)}{E(s)} = K_p \left(1 + T_d * s + \frac{1}{T_i * s} \right) \quad (13)$$

Se debe notar que el método Ziegler Nichols fue concebido en su momento para controles analógicos y no para digitales, pudiéndose generar inestabilidad, especialmente porque su filosofía apunta a tener una oscilación inicial del 25%.

La mayoría de los métodos de sintonización reciben el nombre de quien los desarrolló. A continuación, en la tabla VI, se presenta un resumen de sus parámetros:

Tabla VI. Métodos de sintonización de un controlador PID aplicado a un proceso autorregulado.

Regla	Kc	Ti	Td	Comentario
Ziegler_Nichols	$X1 * T / (K * L)$	$2 * L$	$0.5 * L$	$1.2 \leq X1 \leq 2$
Lambda	$(0.5 * L + T) / \{K * [0.5 * L + Tc1]\}$	$T + (0.5 * L)$	$(T * L) / (L + 2 * T)$	Tc1: T a 3T
Borresen_Grindal	$T / (K * L)$	$3 * L$	$0.5 * L$	
Callender	$1.066 / (K * L)$	$1.418 * L$	$0.353 * L$	$(L/T) = 0.3$
Chien	$0.95 * T / (K * L)$	$2.38 * L$	$0.42 * L$	$0.1 < (L/T) < 1$
Moros	$1.2 * T / (K * L)$	$2 * L$	$0.42 * L$	

Fuente: [17]

De acuerdo a la Fig. 10 los parámetros son: **L** es el tiempo de retardo, de cuanto demora en reaccionar el sistema al inicio. **T** es el tiempo de crecimiento, hasta llegar al valor de consiga y **K** el valor de salida que permanece estable.

Sintonización de un controlador PID para un sistema digital

La sintonización de controladores digitales difiere de los análogos, y existen criterios propios a aplicarse. Las consideraciones son:

- El sistema digital deberá estar en una región de control, para lo cual el factor de incontrolabilidad debe estar:

$$0.1 \leq \left(\frac{L}{T}\right) \leq 0.3 \quad (14)$$

- Aumentar el retardo L a un valor igual a la mitad del período de muestreo:

$$L' = L + \frac{T_s}{2} \quad (15)$$

- Disminuir K_p a la mitad o menos, dependiendo del sistema:

$$K' = \frac{K_p}{2} \quad (16)$$

- Ziegler y Nichols también establecieron una relación entre T_i y T_s :

$$T_i = 4 * T_s \quad (17)$$

1.3.4.4 Control Difuso

Es usada para control de procesos no lineales y con un comportamiento variable en el tiempo. La lógica difusa utiliza operadores para describir un sistema mediante reglas y se puede asignar valores denominados grados de pertenencia. Un bosquejo inicial para el diseño de un tipo de control difuso se muestra a continuación en Fig. 11:

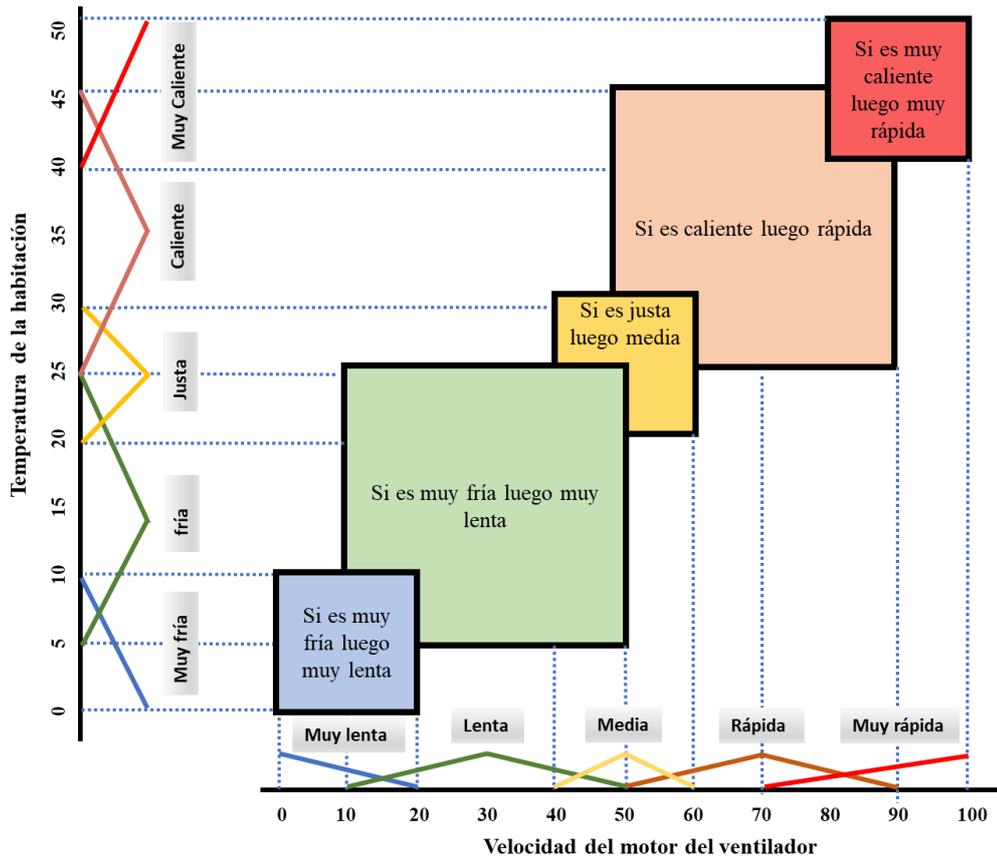


Fig. 11. Ejemplo de establecimiento de reglas en control difuso. [9]

Para el ejemplo de Fig. 11 se requiere elaborar un control de temperatura en una habitación para lo cual se va a controlar un ventilador. Se establecen las reglas para operación tanto de la variable de proceso, en este caso temperatura, como la variable que será controlada, en este caso la velocidad del motor del ventilador. Luego se intercambian reglas para recibir una determinada respuesta a cierta condición que el usuario desea y es el denominado punto de consigna o setpoint. Cada requerimiento tiene ya una respuesta definida de actuación. En el control Fuzzy se determinan de manera similar las respuestas del sistema.

Un esquema simple de controlador difuso es un sistema basado en reglas estáticas, definidas normalmente por la experiencia del diseñador; sus elementos representados formalmente en un diagrama de control se muestran en la Fig. 12.

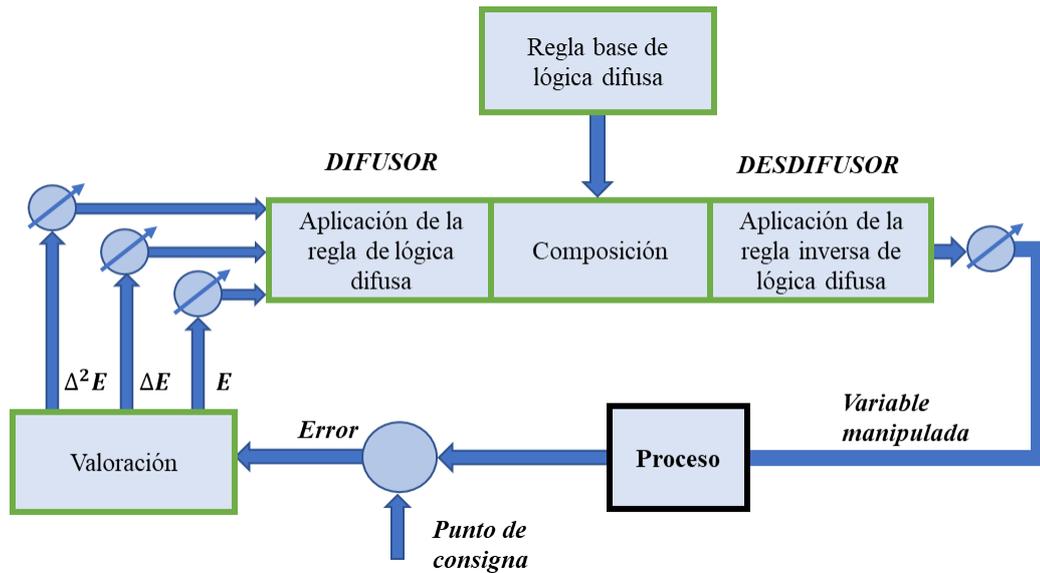


Fig. 12. Ejemplo de un controlador de lógica difusa estático. [9]

Para entender el control difuso es importante conocer algunas definiciones:

Etapa de fusificación, dónde se da el proceso de conversión de los valores de entrada o salida a términos verbales o lingüísticos llamados conjuntos difusos.

Universo del discurso, donde se plasma el conocimiento experto del sistema, aquí se coloca el rango completo donde se ubican tanto del error y su derivada.

Función de pertenencia, se nota por μ o MF y es la curva que define el grado, entre uno y cero, en que las variables pertenecen a los conjuntos difusos. Las formas de la curva más conocidos son: triangulares, trapezoidales, tipo campana de Gauss entre otras, cada una tiene su correspondiente representación matemática y gráfica, por ejemplo para una función de pertenencia tipo triangular, la ecuación y su representación gráfica son:

$$f(x, a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \quad (18)$$

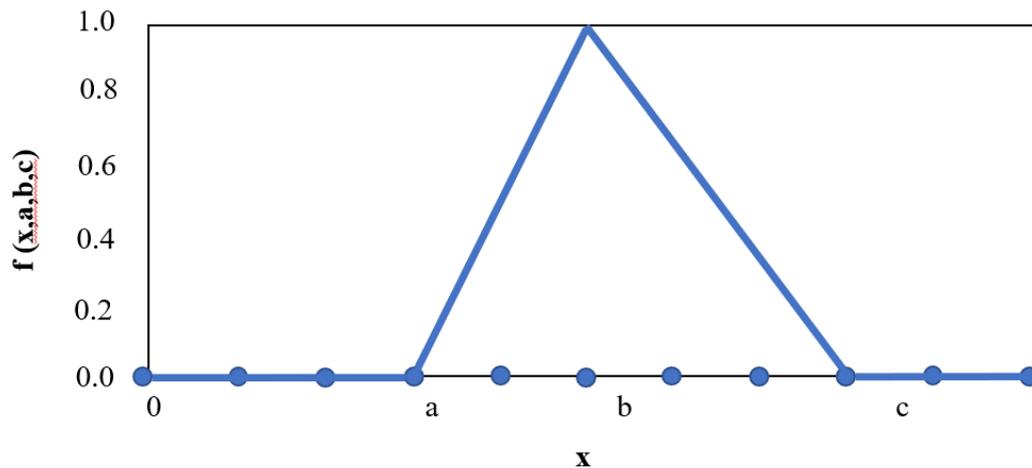


Fig. 13. Ejemplo de función de pertenencia triangular. [6]

El grado de pertenencia del valor de entrada, se expresa finalmente como:

$$f(x, a, b, c) = \max \left(\min \left(\frac{x - a}{b - a}, \frac{c - x}{c - b} \right), 0 \right) \quad (19)$$

Reglas de inferencia, son de tipo causa efecto y toman los niveles de pertenencia para junto a las reglas para generar una salida en el sistema difuso.

Defusificación, es una transformación invertida que mapea la salida del dominio difuso en una acción de control no difusa, es decir lo convierte a un rango de valor de la variable de salida. De los métodos más utilizados se tiene al llamado “Centro de gravedad”, cuya técnica retorna al centro del área donde se generó la etapa de inferencia.

1.3.5 Criterios de desempeño

Luego de una perturbación en el sistema se puede evaluar la característica de respuesta hasta retornar al punto de ajuste. Los criterios para dicho análisis son:

Razón de amortiguamiento.

La relación entre crestas de los dos primeros ciclos es 1 /4. Si el valor es mayor

hay mayor estabilidad del sistema, pero el tiempo de normalización es mayor.

Criterio de área mínima

Se evalúa la integral en función del error absoluto IAE (Integral Absolute Error), mientras su valor sea menor, el tiempo de recuperación será menor. La fórmula es:

$$IAE = \int_0^t |e(t)| dt \quad (20)$$

Criterio de mínima integral del cuadrado del error

Se evalúa la integral en función del cuadrado error ISE (Integral Square Error), y penaliza grandes errores. A menor valor, el tiempo de recuperación será menor.

$$ISE = \int_0^t [e(t)]^2 dt \quad (21)$$

Criterio de mínima integral en el tiempo del valor absoluto del error

Evalúa la integral la integral de la función absoluta del error multiplicada por el tiempo ITAE (Integral Time Absolute error), en un valor $0 \leq t \leq 1$. Por lo que fue diseñada para penalizar pequeños errores.

$$ITAE = \int_0^t |e(t)| * t dt \quad (22)$$

Criterio de mínima perturbación

Requiere una curva de recuperación no repetitiva y se usa para correcciones rápidas de perturbaciones cíclicas.

Criterio de la amplitud mínima

Se evalúa que la amplitud de la cresta de desviación tienda a ser mínima. El criterio da importancia a corregir la desviación antes que el tiempo que toma en recuperarse.

La Fig.14 ilustra claramente sobre los criterios de desempeño explicados:

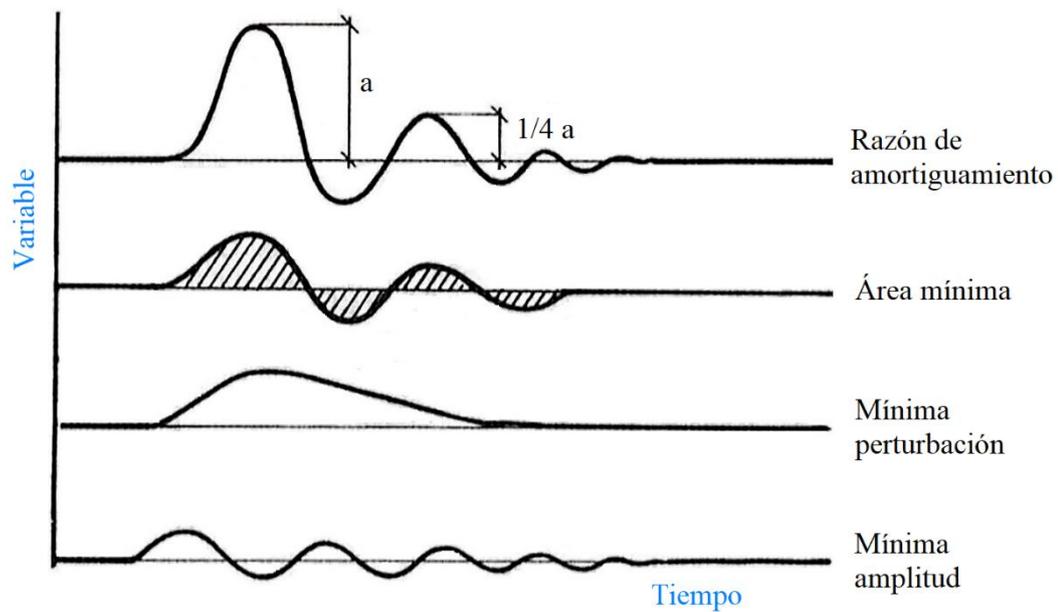


Fig. 14. Criterios de desempeño. [9]

1.3.6 Análisis de respuesta transitoria y error en estado estacionario

Al analizar sistemas de control, es necesario disponer de una base de comparación para evaluar el comportamiento de un determinado proceso. Para efectos de diseño se emplean señales especiales de entrada sean tipo escalón, rampa, impulso, senoidal y se evalúan las respuestas en el tiempo a su salida.

La respuesta en el tiempo de un sistema de control se ha dividido en dos partes: la transitoria y en el estado estacionario. Por respuesta en estado estacionario se entiende al valor que toma la salida cuando el tiempo tiende al infinito [8].

Los parámetros de respuesta transitoria son:

- **Tiempo de retardo t_d**

Es el tiempo requerido para alcanzar el valor final por primera vez

- **Tiempo de crecimiento t_r**

Se ha establecido que sea el tiempo requerido para que la respuesta pase del 10% al 90% del valor final.

- **Tiempo de pico t_p**

Tiempo requerido para que la respuesta alcance el primer pico de sobre impulso

- **Sobre impulso máximo M_p**

Es el valor pico máximo de la curva de respuesta. Normalmente viene expresado porcentualmente y se determina de la siguiente manera:

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} * 100 \% \quad (23)$$

- **Tiempo de establecimiento t_s**

Es el tiempo requerido para que las oscilaciones amortiguadas transitorias alcancen y permanezcan dentro del $\pm 2\%$ o del $\pm 5\%$ del valor final.

Se debe notar que los parámetros son para análisis en dominio del tiempo. Se presenta en Fig. 15 la curva de respuesta típica de una señal $c(t)$ obtenida a la salida de un sistema.

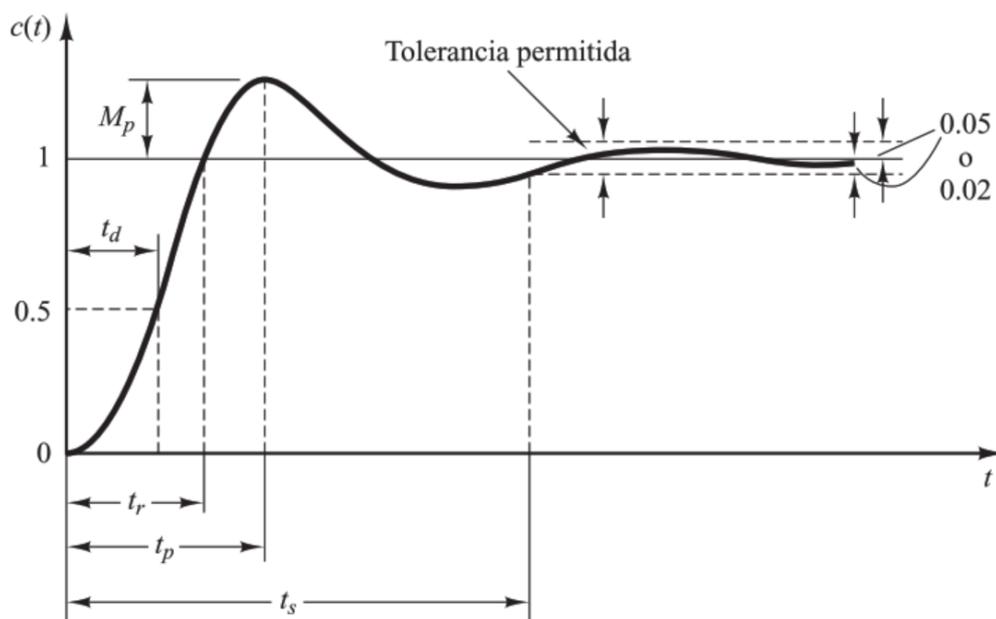


Fig. 15. Parámetros de respuesta transitoria. [8]

1.3.7 Sistemas embebidos

Sistema embebido se denomina a los equipos electrónicos diseñados para un propósito específico, por ejemplo, un sistema de ignición de un automotor, control de temperatura de cilindros de máquinas de combustión interna, un teléfono celular, ruteadores, módems, etc. En Fig. 16 se representa la variedad de aplicaciones.

Los componentes básicos son:

- Arquitectura de hardware
- Elementos de entrada /salida: sensores y actuadores
- Software de control

Las prestaciones:

- Óptimos tiempos de respuesta.
- Bajo consumo de energía.
- Alta fiabilidad.
- Bajo costo de implementación.



Fig. 16. Sistemas embebidos. [18]

Para procesos de temperatura se tiene una variedad de aplicaciones, normalmente vienen en módulos compactos, con software incorporado en microchips.

La presente investigación presenta un sistema embebido, mismo que se ha caracterizado para medición de temperatura, su hardware y software están dedicados a la adquisición de datos y el tratamiento para poder comunicarse de manera óptima con un computador y poder desplegar las curvas características al usar determinadas filosofías de control.

1.3.8 Hardware de código abierto

Opensource.com hace una definición precisa para el hardware de código abierto: “Se refiere a las especificaciones de diseño de un objeto físico que se licencian de tal manera que dicho objeto puede ser estudiado, modificado, creado y distribuido por cualquiera” [19]. Es decir, el creador de un sistema entrega deliberadamente sus especificaciones de diseño y el código de sus algoritmos, permitiendo al usuario final la manipulación absoluta.

Los beneficios del hardware de código abierto son:

- Permite la mejora continua de un diseño, pues al compartir sus diagramas se pueden notar sus debilidades, el usuario los puede corregir, de hecho, hay fórums y páginas especializadas que vienen permitiendo el desarrollo continuo.
- Hardware de menor costo, debido a que hay un mayor número de usuarios que lo entienden y comparten sus experiencias; el resultado, la venta de estos dispositivos aumenta, permitiendo un descenso en el costo.

Ejemplos:

- Arduino, es una plataforma desarrollada para creación de aplicaciones electrónicas. Permite la manipulación libre tanto de hardware como software. Nació en 2013 como parte de un proyecto para obtener placas de bajo costo para estudiantes de electrónica [20]. Consta de una placa que permite conectar periféricos de entrada / salida hacia un microcontrolador ATmega328. Funciona en sistemas operativos Windows, macOS y GNU/Linux.



Fig. 17. Microcontrolador Arduino UNO. [21]

- Raspberry Pi, es un ordenador de bajo costo y ha permitido desarrollar pequeños prototipos en electrónica e informática como servidores web, sistemas de domótica, servidores de correo electrónico, etc.

1.3.9 Herramientas de análisis computacional

Matlab / Simulink es una herramienta para realizar cálculos matemáticos, usa un lenguaje de programación de alto desempeño conocido como lenguaje M.

Características:

- Cálculo numérico preciso.
- Capacidad de manejo simbólico.
- Facilidades de diseño con la interfaz gráfica.
- Extensas bibliotecas de funciones.
- Despliegue gráfico de tendencias de variables.

A través de su herramienta Simulink puede realizar simulación de sistemas dinámicos. Puede trabajar con sistemas lineales, no lineales, continuos y discretos. El ambiente de trabajo en Simulink es más versátil que el ambiente propio de Matlab pues se muestra el proceso de diseño de manera gráfica como se ve en Fig. 18, así mismo se pueden desplegar valores de cálculos o adquiridos mediante tarjetas digitales fácilmente adaptables a los puertos de una computadora personal.

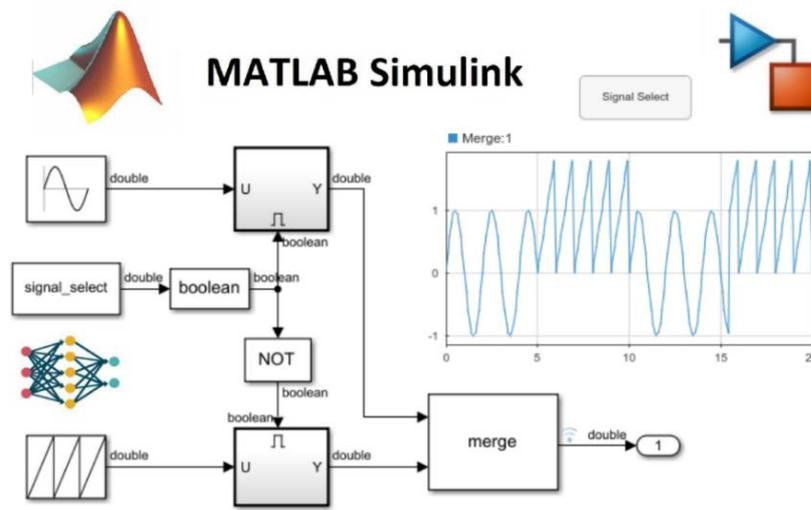


Fig. 18. Ambiente de trabajo de Matlab + Simulink. [22]

1.3.10 Transferencia de calor

El calor se transfiere por tres mecanismos distintos: Conducción, convección y radiación, como se muestra en Fig. 19. En general siempre estarán presentes de forma simultánea en mayor o menor medida [23]. La energía térmica total P_H es la suma de los tres mecanismos y viene expresada en watios [W].

$$P_H = P_{H_{Conducción}} + P_{H_{Convección}} + P_{H_{Radiación}} \quad [W] \quad (24)$$



Fig. 19. Mecanismos de transferencia de calor. [24]

Conducción

Es un proceso de transferencia de energía que tiene lugar mediante un soporte físico, de manera que la energía interna de las zonas de mayor temperatura se desplaza hacia las de menor con el fin de anular el gradiente térmico. La razón de flujo de calor para conducción es proporcional para una gradiente unidimensional $d\vartheta/dx$, donde λ en la dimensión [W/mK] es la conductividad térmica específica y A es el área de cruce seccional para el flujo de calor.

$$P_{H_{Conducción}} = \frac{dQ}{dt} = -\lambda A \frac{d\vartheta}{dx} \quad (25)$$

Para un simple cuerpo cúbico con longitud L y dos interfaces paralelas A en diferentes temperaturas ϑ_1 y ϑ_2 , la ecuación de la transferencia de calor es:

$$P_{H_{Conducción}} = \frac{dQ}{dt} = \lambda A \frac{\vartheta_1 - \vartheta_2}{L} \quad (26)$$

Convección

Es un mecanismo de transmisión de calor basado en el desplazamiento de masa en el interior de un fluido, lo que origina una mezcla de unas zonas con otras y el consiguiente intercambio de calor entre ellas. El sentido de la conducción también es de las temperaturas decrecientes. Aunque como en el caso de la conducción requiere de un soporte físico, este no es físico, sino en movimiento. Este mecanismo es de especial importancia en el intercambio de calor entre una superficie y el fluido que la rodea. Según el motor que dota de movimiento al fluido se puede diferenciar entre convección libre y forzada. La razón de flujo de calor para convección puede ser descrita por la ecuación:

$$P_{H_{Convección}} = \alpha A (\vartheta_1 - \vartheta_2) \quad (27)$$

Donde α es el coeficiente de convección, A es el área de superficie a la temperatura ϑ_1 del objeto, ϑ_2 es la temperatura del fluido circundante.

Radiación

La radiación electromagnética es aquella que todo cuerpo emite si se halla por encima del cero absoluto, es emitida en todas las direcciones del espacio y puede ser interceptada por otros cuerpos dando lugar a un intercambio radiante entre superficies. La radiación emitida en un rango de longitudes de onda adecuado es susceptible de afectar al contenido energético de un cuerpo. A diferencia de los mecanismos anteriores no precisa de un soporte físico, ni de diferencia de temperatura entre cuerpos. La emisión de una superficie es mayor cuanto más elevada sea su temperatura, independiente de su gradiente térmico. El análisis de la radiación se realiza con la ley de Stefan Boltzmann [25].

$$P_{H_{Radiación}} = \varepsilon \sigma A \vartheta^4 \quad (28)$$

Donde ε es la emisividad, $\sigma = 5.67 \times 10^{-8} [Wm^{-2}K^{-4}]$ es la constante de Stefan Boltzmann y ϑ es la temperatura de una superficie A .

1.3.11 Modelo eléctrico análogo al proceso de disipación térmica

El planteo del modelo matemático exacto para la transferencia de calor conduce a ecuaciones lineales complejas, sin embargo, para fines prácticos, se puede aproximar a un modelo simplificado donde prevalece la conducción [26]. Dados dos puntos a temperaturas T_1 y T_2 , la potencia que fluye de T_1 a T_2 se calcula mediante:

$$P = \frac{T_1 - T_2}{R_{th}} \quad (29)$$

Donde R_{th} es la llamada resistencia térmica. La ecuación (12) es válida solo para pequeños saltos térmicos o bien vale en forma de desigualdad, adoptando el máximo valor de R_{th} .

La ecuación (12) es formalmente análoga a la ley de Ohm, como se muestra:

$$I = \frac{V_1 - V_2}{R} \quad (30)$$

Equiparándose de la siguiente manera:

$$V \rightarrow T \text{ [}^\circ\text{C]}, \quad I \rightarrow P \text{ [W]}, \quad R \rightarrow R_{th} \text{ [}^\circ\text{C/W]}$$

Esto permite modelizar situaciones típicas de propagación de calor y se puede representar de manera similar a un esquema eléctrico como se muestra a continuación:

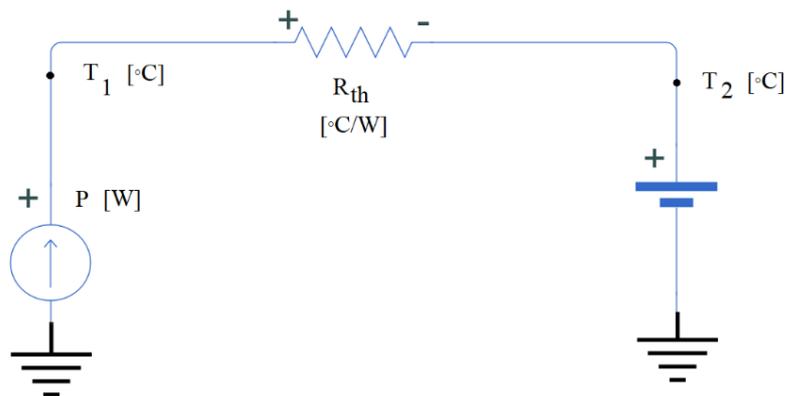


Fig. 20. Modelo eléctrico análogo al proceso de disipación térmica.

1.4 Descripción del objeto de estudio

El objeto de estudio es la aplicación de controladores industriales en procesos de temperatura y el campo de estudio es un proceso embebido de temperatura.

Para el respectivo análisis comparativo se han escogido 2 tipos de controladores:

- El PID, cuya sintonización se hará por el método de Ziegler Nichols
- Control difuso

El PID se ha desarrollado de acuerdo a las teorías de control clásico mientras que el control difuso corresponde a la teoría moderna de control, se aplicarán criterios de desempeño [9] y los resultados se centrarán en el análisis comparativo de la respuestas del sistema tanto en respuesta transitoria y en estado estacionario [8], para una clara presentación de resultados se usa también la herramienta computacional Matlab/ Simulink [22].

La principal característica del estudio es la aplicación en un proceso embebido de temperatura [18], mismo que ha sido diseñado para constituirse en un módulo didáctico, por lo cual la construcción se basa en la utilización de un hardware de código abierto como lo es el Arduino [21]. La metodología escogida para desarrollar el prototipo embebido será la sugerida por MathWorks [27], las interfaces de comunicación se desarrollan usando las bondades de la plataforma de desarrollo Arduino IDE.

1.5 Conclusiones

- Se establece el marco teórico referencial para el análisis comparativo de tipos de controladores aplicados a un sistema embebido de temperatura, de la investigación realizada se observa que a pesar de el transcurso de los años la literatura menciona el constante desarrollo de nuevas tecnologías y la aplicación a nivel industrial demanda una capacitación continua del personal. Probablemente en pocos años se requiera realizar un nuevo análisis comparativo con la nueva tecnología y métodos que vendrán.

- Hoy en día se cuenta con abundante información sobre el tema que se desee tratar sin embargo se debe notar que no todos cuentan con un aval científico reconocido, por lo cual es necesario verificar el origen de la información. Así mismo hay mucha literatura que plantea interesantes hipótesis que nos invitan a seguir investigando, el conocimiento hoy es globalizado y hay mucho interés por compartirlo. Al investigar sobre el hardware de código abierto se evidencia una sed inmensa de desarrolladores que aportan conocimiento día a día.

CAPÍTULO II

MATERIALES Y MÉTODOS

2.1 Introducción y objetivo del capítulo

En el presente capítulo se describe la metodología para la implementación del sistema embebido de temperatura (hardware), el software de control, y las interfaces usadas para la interconexión con el software de análisis de datos.

2.2 Metodología de Investigación

El enfoque de la investigación es cuantitativo pues se basa en la determinación de resultados medibles y comparables para llegar a formar un criterio de selección de un controlador adecuado para un proceso de temperatura. La definición de los alcances son el resultado de una minuciosa revisión bibliográfica, los objetivos planteados y la hipótesis que se mencionan en capítulos anteriores. Los alcances van en el siguiente orden: descriptivos, correlacionales y explicativos.

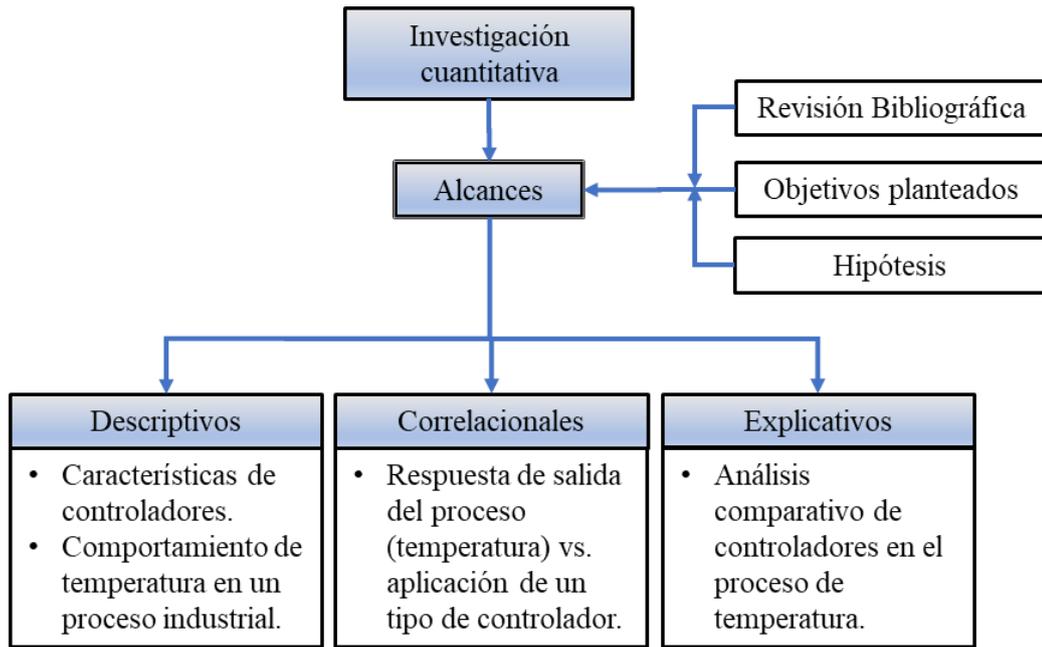


Fig. 21. Proceso de la Investigación Cuantitativa.

El estudio descriptivo busca especificar propiedades y características importantes de los controladores y el proceso de temperatura. El estudio correlacional permitirá asociar las variables mediante los parámetros descritos en la hipótesis. Finalmente, el estudio explicativo permitirá realizar el análisis comparativo propuesto en este proyecto de investigación [28]. Un resumen del proceso de la Investigación Cuantitativa se indica en Fig. 21.

2.3 Diseño del proceso embebido

2.3.1 Metodología de diseño

Dentro del diseño de procesos embebidos MathWorks propietario de la plataforma Matlab sugiere un método simplificado para su desarrollo mostrado en Fig. 22, es un método resultado de la experiencia. Se parte de la investigación y los requerimientos para plasmarlos en un diseño sujeto a modelación y simulación. Si se tienen resultados satisfactorios se procede a la implementación. Finalmente, el proceso embebido se integra al sistema requerido. En todas las etapas se realizan

pruebas y verificación de funcionamiento lo que asegura obtener como resultado un proceso plenamente funcional.

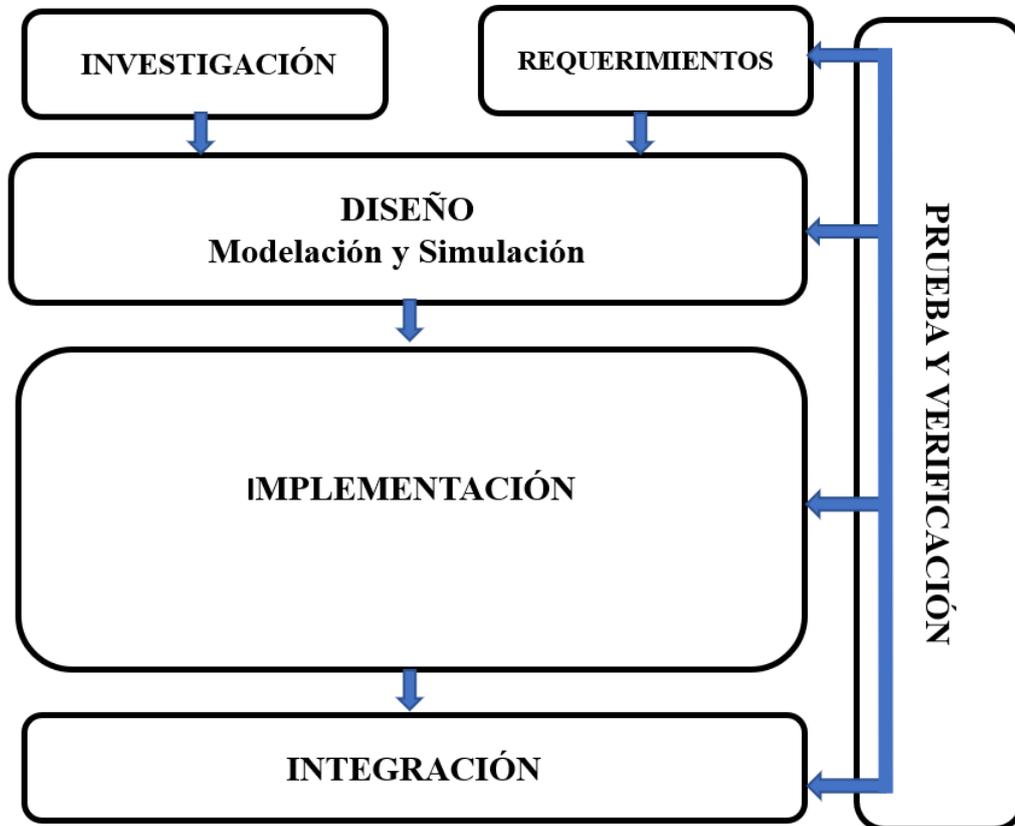


Fig. 22. Método de diseño para sistemas embebidos. [27]

2.3.2 Requerimientos y propuesta funcional del sistema embebido

Como requerimientos del sistema se enuncian:

- Bajo costo: Sensor de temperatura, calentador, controlador, software.
- Rango de operación de 0-100 °C
- Controlador de código abierto que permita implementación de nuevos algoritmos.
- Sistema de análisis con software avanzado.

El sistema embebido consta de las siguientes partes constitutivas, Fig. 23:

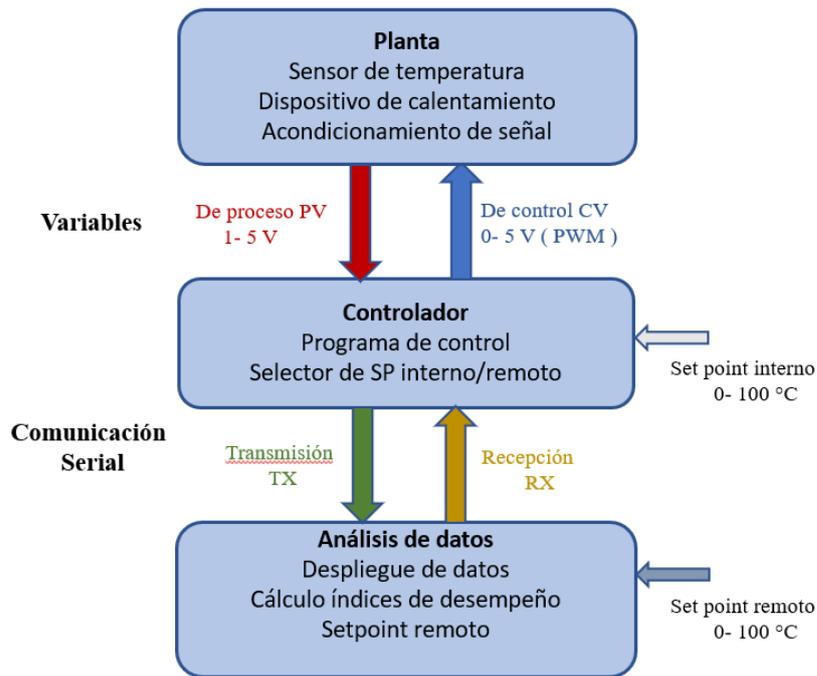


Fig. 23. Partes constitutivas del sistema embebido.

La planta, constituida básicamente de un sensor de temperatura y del dispositivo de calentamiento, la salida normalizada de 1 a 5 V, así mismo la señal de control que ingresará al canal analógico de Arduino es de 0 a 5 V(PWM).

El controlador, donde se incluyen las relaciones matemáticas para un sistema de control de lazo cerrado. Se tiene un setpoint local en un rango de 0 a 100 °C y con un switch selector para escoger setpoint externo (remoto).

El análisis de datos, con comunicación serial y capacidad de visualización del proceso y cálculo de parámetros de desempeño. También se puede manipular el setpoint si en el controlador se ha escogido el modo remoto.

2.3.3 Selección de componentes

Luego de la investigación de elementos idóneos y revisión de disponibilidad de diversos dispositivos electrónicos en el mercado se elabora un cuadro comparativo para la selección de los componentes y se muestra en la tabla VII:

Tabla VII. Selección componentes para el sistema embebido

Sensor de Temperatura	Tipo	CI LM35	RTD PT100	Termistor PTC
	Confiabilidad	Alta	Alta	Media
	Precio USD. \$	2	7.4	6.45
	Extra requerido: Acond. de señal	No	Si	Si
	Precio USD. \$	2	12	2
	Selección	●		
Elemento final de control	Tipo	Transistor TIP31C 40W	Niquelina 40W	
	Confiabilidad	Media	Alta	
	Precio USD. \$	1	4	
	Extra requerido: Relé de aislamiento	No	Si	
	Precio USD. \$	0	17	
	Selección	●		
Controlador	Tipo	Arduino UNO	PIC 16F883	LOGO 6ED1052
	Confiabilidad	Media	Media	Alta
	Precio USD. \$	15	10	170
	Código abierto	Si	Si	No
	Extra requerido: Software progr.	No	21.75	10
	Selección	●		
Software análisis	Tipo	Arduino Plotter	Matlab	Octave
	Versatilidad	Baja	Alta	Media
	Precio USD. \$	0	2000	0
	Código abierto	No	No	Si
	Extra requerido: Programación	Compleja	Baja	Compleja
	Selección		●	

De la tabla VII, se selecciona:

Sensor de temperatura, el circuito integrado LM35 por su confiabilidad que es alta pues, ha sido probado en varios proyectos, la respuesta de temperatura es lineal por lo que no necesita hardware de acondicionamiento de señal extra.

Calentador, se encuentra que se pueden aprovechar las características de disipación de calor de un transistor de juntura bipolar, específicamente el TIP31C cuyo costo es relativamente bajo y se puede conseguir en las electrónicas locales.

Controlador, de las diversas opciones se escoge al Arduino UNO, especialmente porque presenta hardware de código abierto que permite implementar fácilmente nuevos algoritmos y funciones. El costo comercial es bajo, además que su software de programación no tiene valor adicional.

Software de análisis, se halló una buena diferencia de valor entre software pagado y libre, es el caso de Matlab versus Octave; Arduino también ofrece su propio graficador, pero tiene limitadas facilidades para realizar un análisis. Tanto Arduino como Octave serían la elección idónea, sin embargo, son software en desarrollo, que en poco tiempo dispondrán de las librerías adecuadas para un análisis avanzado, al momento no lo disponen. Se escoge Matlab, porque el sistema embebido va dirigido a estudiantes cuyo acceso es mediante licencia estudiantil, por tanto, su uso es viable.

A continuación, en la Fig. 24 se indican los elementos seleccionados.

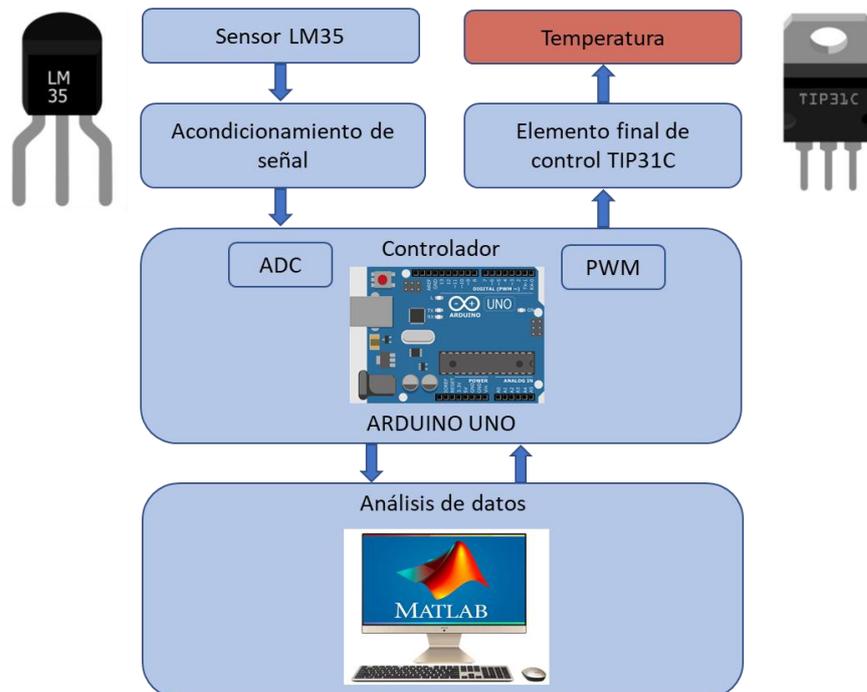


Fig. 24. Diseño del sistema embebido.

2.3.4 Diseño, modelación y simulación

Una vez seleccionados los componentes, y con las características de cada uno se procede a definir los parámetros de operación definitivos y a continuación a realizar el acondicionamiento de señal entre dispositivos. La tabla VIII muestra los valores de salida disponibles en cada parte del sistema.

Tabla VIII. Señales por acoplar en sistema.

Sensor	Salida Planta	Controlador				Proceso
LM35	Acondicionamiento	Entrada analógica	Convertidor A/D	Setpoint D/A	Salida PWM	Temperatura proceso
[mV]	[V]	[V]	[bit]	[bit]	[V]	[°C]
0-1500	1 – 5	0 - 5	0 - 1023	0 - 255	0-5	0-100

En los siguientes subcapítulos se modela el sistema propuesto; así como se acondiciona las señales propuestas.

2.3.4.1 Modelamiento para la transferencia de calor: Transistor TIP31C

Para el diseño de la operación del transistor de juntura bipolar se revisa la curva de la potencia de disipación térmica en su zona operación segura, pues la operación está condicionada por la temperatura en la juntura. La gráfica de la potencia de disipación mostrada en Fig. 25 establece un valor en el área de operación segura.

En este caso se desea una temperatura de operación de 100 ° C, con lo cual el diseño en todo momento debe ser monitoreado para que no sobrepase una disipación de potencia de 15 W.

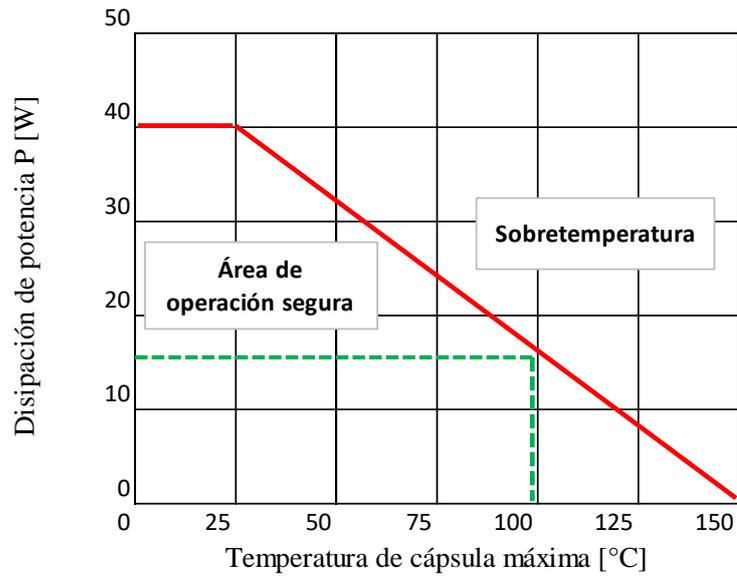


Fig. 25. Gráfica de potencia de disipación del TIP31C. [29]

A continuación, se muestra el modelamiento para la transferencia de calor:

El calor es disipado en un transistor de juntura bipolar BJT, mediante pérdidas eléctricas debidas al efecto Joule resultando un gradiente de temperatura. Cuando se llega a la condición de estado estable la razón de flujo de calor constante P_H corresponde a la potencia eléctrica disipada P_{el}

$$P_H = \frac{dQ}{dt} = -kA \frac{d\theta}{dx} \quad (31)$$

Para fines prácticos y con un flujo de calor constante fluyendo de T_1 a T_2 , como se indicó en (12), para un BJT corresponde:

$$P_H = \frac{T_1 - T_2}{R_{th}} \quad (32)$$

La resistencia térmica total R_{th} en un transistor de juntura bipolar es la suma de las resistencias térmicas de las capas de material que la componen, así:

$$R_{th} = R_{th_{jc}} + R_{th_{ca}} \quad (33)$$

$R_{th_{jc}}$, resistencia térmica de la juntura-cápsula

$R_{th_{ca}}$, resistencia térmica de la cápsula al aire (medio ambiente)

Si se coloca un disipador se suma la resistencia térmica de la pasta conductora $R_{th_{cd}}$ y del disipador $R_{th_{da}}$.

$$R_{th} = R_{th_{jc}} + R_{th_{ca}} + R_{th_{cd}} + R_{th_{da}} \quad (34)$$

La expresión para la potencia total disipada por el transistor de juntura bipolar es:

$$P_H = \frac{T_j - T_a}{R_{th}} \quad (35)$$

T_j , es la temperatura en la juntura y T_a la temperatura del aire (medio ambiente).

Para realizar los cálculos correspondientes se tienen en la tabla IX los datos especificados en la hoja técnica del TIP31C [29].

Tabla IX. Características TIP31C.

Descripción	TIP31C	Transistor de potencia, Tipo NPN	
Disipación de potencia hasta $T_c = 25\text{ °C}$	P_d	40	W
Resistencia térmica Juntura al ambiente (aire libre)	$R_{th_{ja}}$	62.5	°C/W
Temperatura de juntura máxima	T_j	150	°C
Ganancia de corriente DC	h_{fe}	171	
Voltaje colector emisor	V_{ceo}	100	V
Voltaje Colector base	V_{cbo}	100	V
Voltaje Emisor base	V_{ebo}	5	V
Voltaje umbral V_{be}	V_{be}	0.7	V

Fuente: [29]

El circuito base a modelar, se muestra en la Fig. 26 :

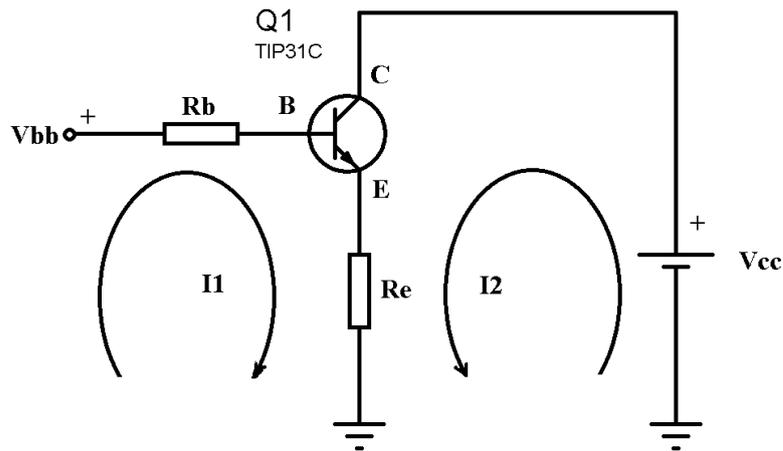


Fig. 26. Circuito base del sistema embebido.

Aplicando el análisis por mallas (Kirchhoff) se tiene:

$$V_{cc} - V_{ce} - V_{Re} = 0 \quad (36)$$

$$V_{bb} - V_{Rb} - V_{be} - V_{Re} = 0 \quad (37)$$

$$I_c = h_{fe} * I_b \quad (38)$$

Donde V_{cc} es el voltaje de polarización, V_{ce} representa la caída de voltaje entre colector y emisor, V_{Re} el voltaje en la resistencia Re , V_{bb} es el voltaje aplicado a la entrada del circuito antes de la resistencia Rb , V_{Rb} la caída de tensión en la resistencia Rb , I_c es la corriente presente en el colector, I_b la corriente en la base y h_{fe} la ganancia del transistor. Se deben asumir algunos valores de operación tales como V_{cc} y V_{bb} ; otros vienen de las hojas técnicas así V_{be} y la ganancia DC h_{fe} . También se debe asumir I_c y V_{Re} .

Las restantes ecuaciones para el cálculo de la potencia térmica disipada se muestran en ec. (39) y la potencia eléctrica generada por el transistor en ec. (40) :

$$P_d = (T_j - T_a) / (R_{Tja} + R_{Tcd} + R_{Tda}) \quad (39)$$

$$P_d = (V_{be} * I_e) + (V_{ce} * I_c) \quad (40)$$

Como se indicó para un valor de temperatura de juntura T_j de 100°C, se debe monitorear que la potencia del TBJ no sobrepase los 15W. Si bien el valor de temperatura de la juntura del transistor T_j es de 100 ° C, se prevé un 10% de factor

de seguridad, para poder medir sobre picos de señal y evitar saturación en los distintos canales de medición, esto es un T_j de 90°C .

A continuación, en la tabla X se presenta el resumen de los valores de diseño, de la hoja técnica y los valores asumidos:

Tabla X. Valores de diseño del sistema embebido.

	Parámetro	Valor	Unidad	Descripción
Datos de diseño	T_j	90	$[\text{°C}]$	Temperatura de operación de la juntura
	T_a	18	$[\text{°C}]$	Temperatura ambiente (aire libre)
	V_{cc}	12	$[\text{V}]$	Voltaje alimentación proceso (TIP31C)
	V_{bb}	5	$[\text{V}]$	Voltaje alimentación señal de control
Datos fabricante / hoja técnica	RT_{ja}	62.5	$[\text{°C/W}]$	Resistencia térmica, juntura a ambiente
	RT_{cd}	0.45	$[\text{°C/W}]$	Resistencia térmica disipador silicona
	R_{tda}	0.825	$[\text{°C/W}]$	Resistencia térmica de un disipador
	h_{fe}	171		Ganancia DC
	V_{be}	0.7	$[\text{V}]$	Voltaje de operación umbral
Valores asumidos	I_c	0.1	$[\text{A}]$	Corriente en colector
	P_d^*	15	$[\text{W}]$	Potencia de disipación máxima
				* Monitoreada en cada cálculo

Una vez obtenidos los resultados del diseño, se busca valores comerciales para las resistencias R_b y R_e ; se recalculan los valores finales. Para todos los casos P_d da un valor menor a 1.5 W.

El circuito diseñado final para el TIP31C se grafica a continuación:

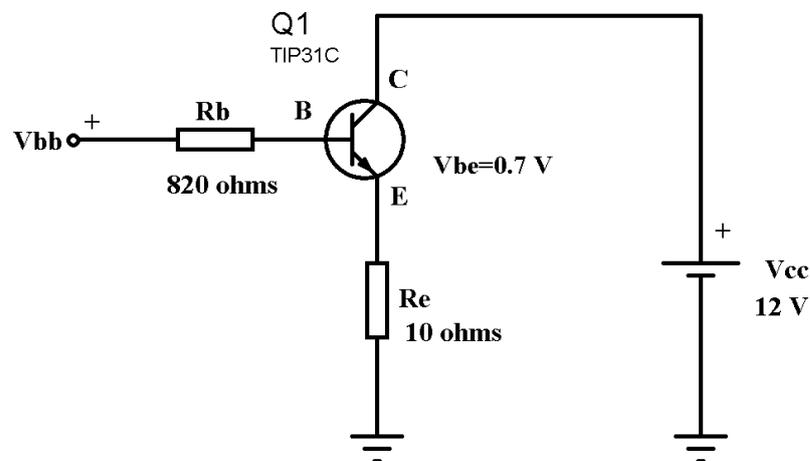


Fig. 27. Circuito del TIP31C del sistema embebido.

2.3.4.2 Acondicionamiento de señal de la variable de proceso: Sensor LM35

De acuerdo con las características de la hoja de datos del LM35 [30], se tiene:

Rango utilizado: 0 - 100 °C
Salida mV: 0 - 1000 mV
Respuesta: Lineal + 10 mV / °C

Para obtener la salida de 1 a 5 V desde la planta del sistema es necesario realizar acondicionamiento de señal desde del LM35 hasta la entrega de señal al controlador como se indica en la Fig. 28. Se usará amplificadores operacionales.

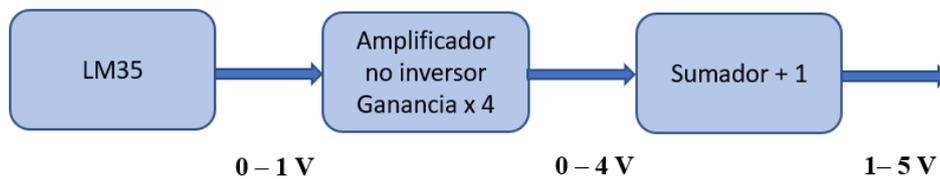


Fig. 28. Acondicionamiento señal del LM35 a salida estándar 1 a 5VDC.

Desde 0-1 V pasamos a 0-4 V usando un amplificador no inversor, mediante la configuración mostrada en la Fig. 29, a continuación, se muestra el cálculo:

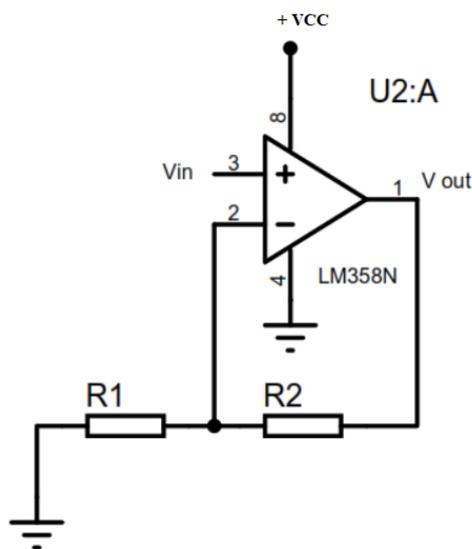


Fig. 29. Amplificador no inversor.

$$V_{in} = \beta * V_{out} \quad (41)$$

$$V_{out} = V_{in} \left(1 + \frac{R_2}{R_1}\right) \quad (42)$$

Aplicando valores se tiene:

$$4 = 1 \left(1 + \frac{R_2}{R_1}\right) \rightarrow 3 = \frac{R_2}{R_1}$$

Asumiendo $R_2 = 30000 \Omega$, con lo cual $R_1 = 10000 \Omega$.

Para pasar de 0-4 V a 1-5 V usamos nuevamente un amplificador operacional en configuración sumador como se muestra en la Fig. 30:

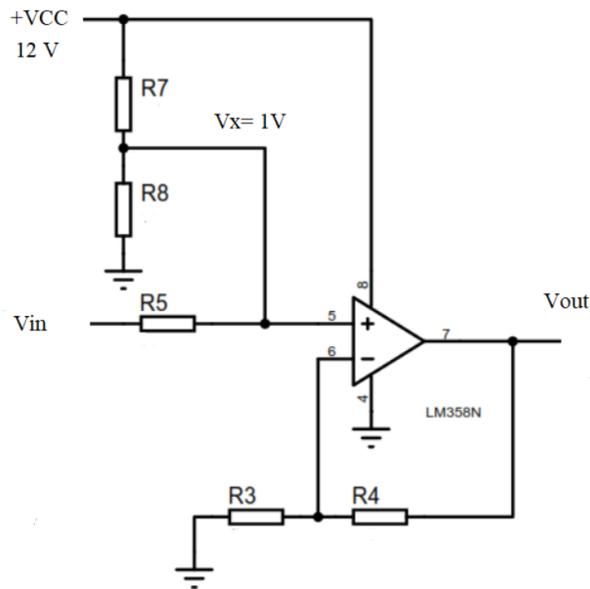


Fig. 30. Circuito sumador.

Las relaciones matemáticas son:

$$V_{out} = V_{in} + V_x$$

$$V_x = V_{out} * \left(\frac{R_3}{R_3+R_4}\right), V_x = V_{cc} * \left(\frac{R_8}{R_8+R_7}\right), V_x = V_{in} * \left(\frac{R_8}{R_8+R_5}\right)$$

Asumiendo $R_3=R_4=470 \Omega$ y considerando los valores máximos, $V_{in}=4 \text{ V}$, $V_x=1$, con $V_{cc}=12\text{V}$, se tiene $R_7= 11 \text{ k} \Omega$, $R_8= 1 \text{ k} \Omega$.

Una vez obtenidos los cálculos de cada parte constitutiva de la planta, se procede a realizar el modelamiento y simulación de funcionamiento mediante el programa Proteus, como se muestra en la Fig. 31.

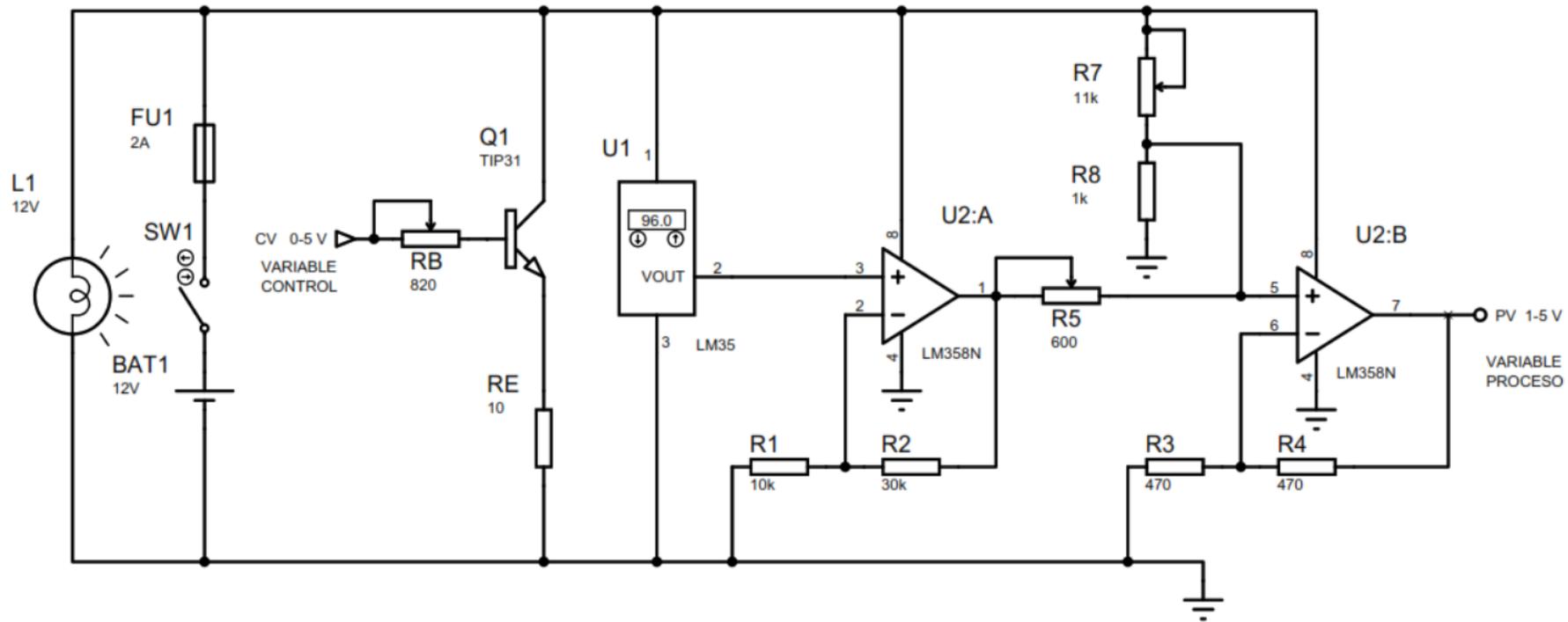


Fig. 31. Modelamiento de la Planta y simulación en Proteus.

2.3.4.3 Programación base: Tarjeta Arduino UNO

Las tareas que debe realizar el controlador se resumen en el siguiente diagrama:

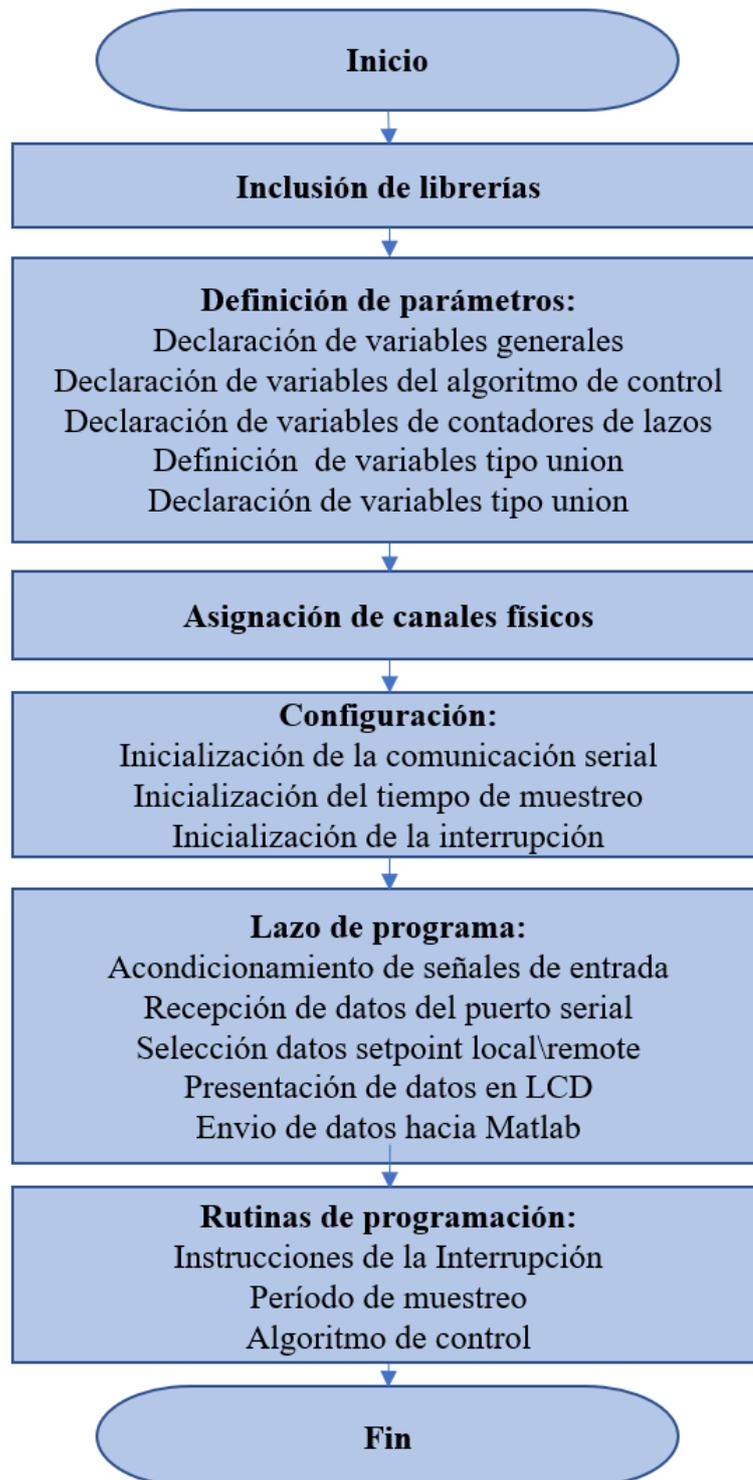


Fig. 32. Tareas base del controlador Arduino UNO.

Inclusión de librerías, se usan dos librerías, “TimerOne.h” [31] que abre un reloj que lleva sincronización del PWM así como usa una interrupción asociada para llevar el tiempo de muestreo T_s y la librería “LiquidCrystal_I2C.h” [32] que permite adaptar el LCD a protocolo I2C, la descripción detallada de esta última librería se puede revisar en el anexo 3.

Definición de parámetros, que permite asignar el tipo de dato a cada variable usada en la programación. En este bloque se resalta el uso de las variables tipo “unión” que permiten interactuar Arduino con Matlab vía comunicación serial, toda vez que Arduino envía en secuencia datos por byte. Las variables para número real en Arduino se denominan tipo float con un tamaño de 32 bits, en Matlab este mismo dato se denominan tipo single. En Arduino para tomar datos desde los canales análogos así como el manejo de escalamientos y operaciones matemáticas se hace con float, ahora bien para la transmisión de datos por comunicación serial, se envían datos de byte en byte (8 bits), es decir un dato float lo debo descomponer en 4 bytes para luego enviarlo, así mismo para recibir debo rearmar 4 bytes para obtener el dato requerido; en Matlab en cambio se configura el envío/recepción de datos como tipo single y los trata de una vez los 32 bits. La variable tipo unión se caracteriza por almacenar en una misma dirección sea un dato tipo float o un arreglo uint8_t de 4 bytes.

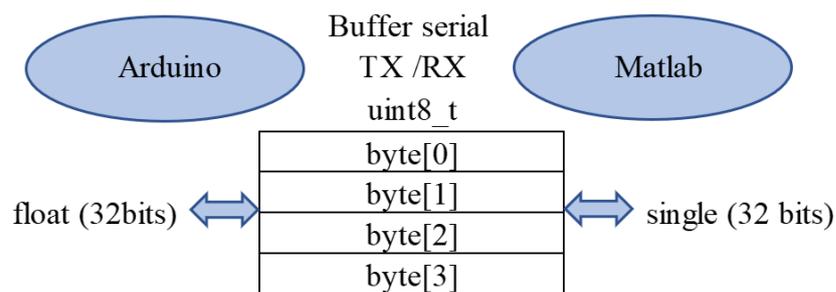


Fig. 33. Manejo de datos tipo real en Arduino – Matlab

Asignación de canales físicos, se refiere a configurar las entradas y salidas de los puertos analógicos y digitales de Arduino, así como la transmisión recepción de datos vía I2C. En la tabla XI se muestra la configuración.

Tabla XI. Asignación de canales físicos en tarjeta Arduino.

Variable	TAG	Canal
Variable del proceso PV	Tproc_CAN	Análogo 0
Setpoint interno	Tsetpm_CAN	Análogo 2
Selector Setpoint externo/interno	SW_Tsetp_CAN	Digital 2
Variable del control CV	Tsetp_CAN	Digital 3
Adaptador LCD a I2C	SDA	Análogo 4
Adaptador LCD a I2C	SCL	Análogo 5

Configuración, en esta sección se realizan tres tareas de inicialización: la comunicación serial, el contador del tiempo de muestreo y la interrupción para realizar las instrucciones de muestreo.

Lazo de programa “void loop()”, se acondiciona las señales recibidas y para atenuar picos y valles de señal se toma un promedio de las lecturas de cada canal; se programa la recepción y salida de datos por comunicación serial hacia Matlab; se recibe dato del selector de setpoint interno / externo en la entrada digital de Arduino; aquí también constan las instrucciones para despliegue de valores en el display LCD.

Rutinas de programación, se colocan al final las instrucciones de la interrupción para realizar el muestreo, se implementa el algoritmo de control y es aquí donde cambiarán las instrucciones dependiendo del tipo de control que se esté aplicando.

2.3.4.4 Hardware: Tarjeta Arduino UNO

Desde la planta se disponen tanto la variable de proceso PV como la variable de control CV, y en base a la programación propuesta se implementa el hardware necesario para la tarjeta Arduino UNO. Cabe mencionar que se coloca el selector de setpoint, el display LCD y el potenciómetro para variar el setpoint local o interno. Al igual que la planta se dibuja el esquema correspondiente en Proteus (Release 8.9 SP2), en la Fig. 34 se muestra la implementación final.

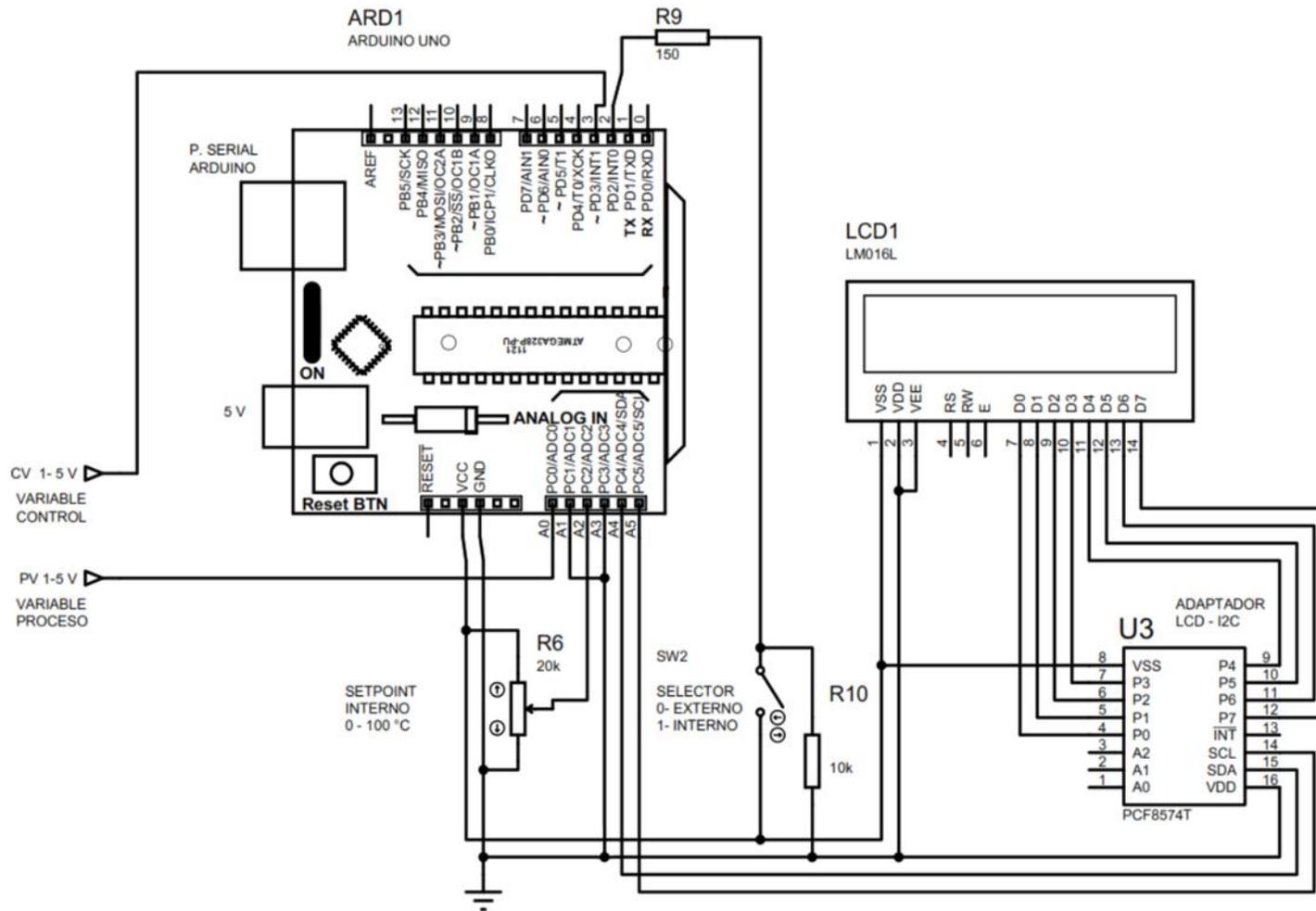


Fig. 34. Modelamiento de la tarjeta Arduino en Proteus.

2.3.4.5 Creación del entorno para análisis de datos: Matlab-Simulink

El esquema para el análisis de datos se resume en el siguiente gráfico:

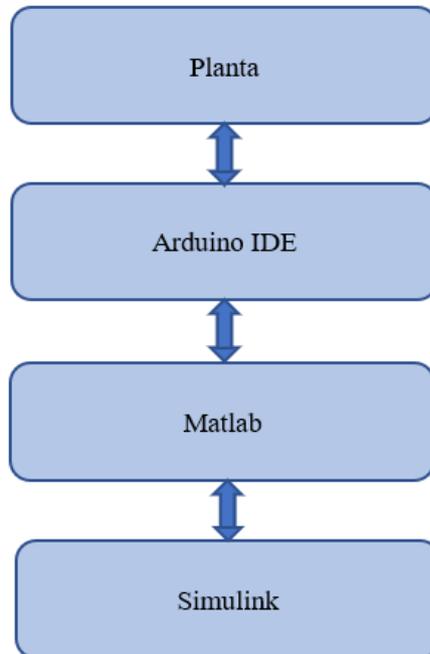


Fig. 35. Esquema de despliegue de datos y análisis.

Para realizar el análisis de datos la planta se comunica con una computadora mediante el software “Arduino IDE”, el protocolo es serial, el flujo de datos es bidireccional. El paquete computacional Matlab - Simulink permite adquirir datos desde la memoria del controlador Arduino. En la Fig. 36 se observa el logo de la librería “Simulink Support Package for Arduino Hardware” que debe ser instalada en Matlab en el menú “Add ons”.

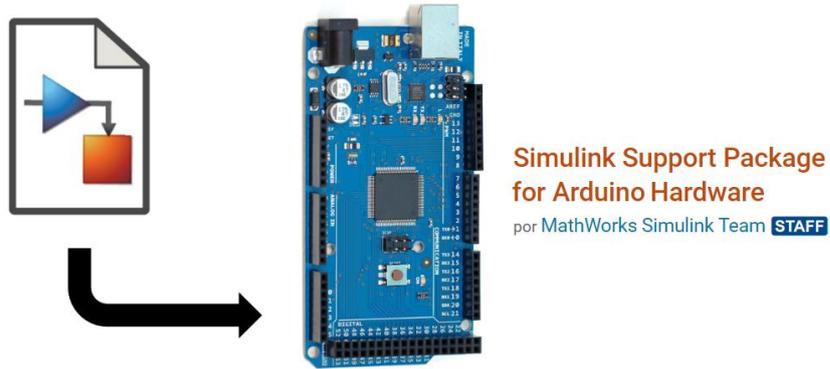


Fig. 36. Librería usada en Simulink para interactuar con Arduino. [22]

El ambiente gráfico desarrollado en Simulink para el análisis es el siguiente:

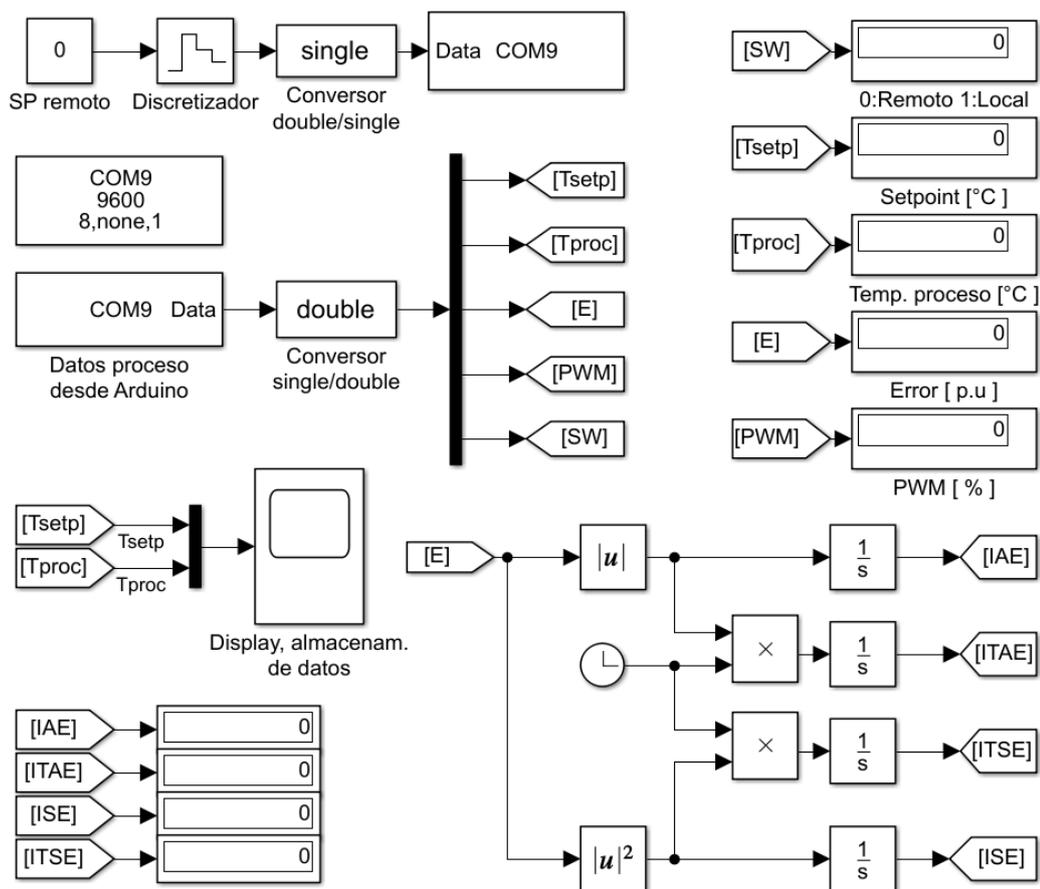


Fig. 37. Ambiente de análisis de datos en Simulink.

A continuación, se describen las partes implementadas en la Fig. 37:

Selección y envío del Setpoint, inicia con la selección en la placa de Arduino mediante el switch SW, cuyo estado selecciona:

0: Setpoint remoto (desde el Simulink)

1: Setpoint interno o local (desde potenciómetro ubicado en placa)

A continuación, se envía el dato hacia el Arduino mediante el puerto serial, para ello es necesario que el valor del setpoint vaya de manera discretizada, además que el tipo de dato que se maneja en protocolo serial es mediante secuencia de bytes, por tanto, el tipo de datos “double” de Simulink se debe transformar en “single” y a continuación ingresar al bloque para la comunicación serial. Se recalca el uso del conversor single - double, caso contrario se tendrá error de comunicación.

Mediante la herramienta de osciloscopio se grafican los datos desde la planta, también se aprovecha de la herramienta Logging del mismo, para almacenar datos cada que finalice un muestreo de datos.

El error obtenido en la planta se lo trabaja mediante bloques gráficos de programación para obtener los índices de desempeño basados en el error. Finalmente se muestran los valores acumulados del error en displays.

2.3.5 Construcción física

Se adjunta en el anexo 4 los detalles para la implementación del proceso embebido de temperatura, esto es el listado de material, el diseño de las cajas, la construcción real y las dimensiones tanto para la planta y el control, finalmente se incluye la placa PCB para el armado en baquelita de la parte electrónica.

2.4 Conclusiones

- El diseño del sistema embebido de temperatura permitirá realizar varias aplicaciones de control únicamente cambiando la programación en el controlador. No es necesario la modificación en hardware, salvo actualización en firmware, para poder obtener distintas pruebas sobre la variable de temperatura.
- Se diseñó el sistema embebido de temperatura, mismo que por ser dedicado, resulta con un conveniente costo, que bien puede ser empleado para producción en gran escala. Otra característica es que emplea es la aplicación de fundamentos avanzados, pero de igual manera ocupa para sus cálculos solo la matemática estrictamente necesaria.
- En la actualidad el uso de herramientas de diseño y cálculo facilitan la realización de operaciones avanzadas y obtención de resultados en menor tiempo. El uso de algoritmos y facilidad de programación permiten realizar varios ensayos hasta llegar a un resultado satisfactorio. El conocimiento de distintas herramientas se convierte en una herramienta poderosa y da gran ventaja al diseñador que las domina.

CAPÍTULO III

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

3.1 Introducción y objetivo del capítulo

En el presente capítulo se revisa el desarrollo para la obtención de los algoritmos de programación de los controladores, así como los criterios específicos que se deben tener en cuenta al aplicar a un sistema embebido. Se despliega los resultados obtenidos y se realiza la comparación y análisis respectivo.

3.2 Obtención de la función de transferencia

Para poder representar mediante una ecuación matemática el proceso de calentamiento del sistema embebido se encuentra la función de transferencia en lazo abierto y con un setpoint específico de 40°C, para ello se utiliza la herramienta “systemIdentification” de Matlab, el procedimiento en detalle se incluye en el Anexo 2. La función de transferencia obtenida es:

$$G(s) = \frac{1.3335}{1 + 177.13 * s} * e^{-5.377*s} \quad (43)$$

3.3 Implementación de controladores PID

Existe una diferencia marcada entre construir un PID análogo y un PID discreto, la aplicación para un PID discreto es para un sistema embebido pues se va a programar en la memoria del controlador las instrucciones respectivas, en este punto aparece el concepto del tiempo de muestreo, que permite almacenar valores actuales y del pasado. El desarrollo del algoritmo de control es el siguiente:

Partiendo de la ecuación (3), para el control tipo PID:

$$u(t) = Kp \cdot e(t) + Ki \cdot \int_0^t e(t) dt + Kd \cdot \frac{de(t)}{dt}$$

Donde aplicando la transformada de Laplace obtenemos:

$$U(s) = Kp \cdot E(s) + \frac{Ki}{s} E(s) + Kd \cdot s E(s)$$

Y la función de transferencia del controlador queda:

$$\frac{U(s)}{E(s)} = Kp + \frac{Ki}{s} + Kd \cdot s$$

Para realizar la conversión a control digital se usa la transformada de Z, mediante el siguiente arreglo:

$$\frac{U(s)}{E(s)} = \frac{Kp \cdot s + Ki + Kd \cdot s^2}{s}$$

Aplicando la transformada Z [33],

$$s = \frac{1 - z^{-1}}{T_s}$$

dónde T_s es el período de muestreo.

$$\frac{U(z)}{E(z)} = \frac{Kp \cdot \left[\frac{1 - z^{-1}}{T_s} \right] + Ki + Kd \cdot \left[\frac{1 - z^{-1}}{T_s} \right]^2}{\left[\frac{1 - z^{-1}}{T_s} \right]}$$

Resolviendo y ordenado,

$$\frac{U(z)}{E(z)} = \frac{\left[Kp + Ki \cdot T_s + \frac{Kd}{T_s} \right] + z^{-1} \left[-2 \frac{Kd}{T_s} - Kp \right] + z^{-2} \left[\frac{Kd}{T_s} \right]}{(1 - z^{-1})}$$

$$U(z) = E(z) \cdot \frac{\left[Kp + Ki \cdot T_s + \frac{Kd}{T_s} \right] + z^{-1} \left[-2 \frac{Kd}{T_s} - Kp \right] + z^{-2} \left[\frac{Kd}{T_s} \right]}{(1 - z^{-1})}$$

$$U(z) - U(z)z^{-1} = E(z) \left[Kp + KiT_s + \frac{Kd}{T_s} \right] + E(z)z^{-1} \left[-2\frac{Kd}{T_s} - Kp \right] + E(z)z^{-2} \left[\frac{Kd}{T_s} \right]$$

Ahora aplicando la transformada inversa de Z, se retorna al tiempo discreto, es decir a una función que depende de las muestras.

$$u(k) - u(k-1) = e(k) \left[Kp + KiT_s + \frac{Kd}{T_s} \right] + e(k-1) \left[-2\frac{Kd}{T_s} - Kp \right] + e(k-2) \left[\frac{Kd}{T_s} \right]$$

Para la implementación en Arduino, se definen como coeficientes K1, K2 y K3:

$$K1 = \left[Kp + KiT_s + \frac{Kd}{T_s} \right]$$

$$K2 = \left[-2\frac{Kd}{T_s} - Kp \right]$$

$$K3 = \left[\frac{Kd}{T_s} \right]$$

Así mismo, se reconoce en la ecuación a $u[k]$ como la salida discreta correspondiente a una muestra k . Así designando como referencia a una muestra con valor $k=0$, $u[0]$ es la muestra inicial, $u[1]$ la muestra siguiente, tomada con periodo de retraso T_s . De la misma manera para el error, $e[k]$ es el valor de error inicial, $e[k-1]$ corresponde al error en retraso de un periodo y $e[k-2]$ es el error tomado en dos periodos anteriores.

Para el cálculo del PID discreto, con $k=0$, la ecuación para implementar en el controlador queda:

$$u[0] = K1 * e[0] + K2 * e[1] + K3 * e[2] + u[1]$$

Los valores de K1, K2 y K3 dependen del criterio de sintonización que se use.

3.3.1 Métodos de sintonización

Se aplican los distintos métodos de sintonización tradicionales de acuerdo con sus criterios dados en la tabla VI, se obtienen los valores mostrados en la tabla XII .

Tabla XII. Valores calculados de sintonía para PID – análogos.

Método	Kc'	Ti	Td
Ziegler_Nichols	16.980	18.754	4.689
Lambda	1.460	181.819	4.568
Borresen-Grindal	14.150	28.131	4.689
Callender	0.085	13.297	3.310
Chien	13.442	22.317	3.938
Moros	16.980	18.754	3.938

Se debe notar que si se aplican exactamente los valores indicados en la tabla XII se obtienen respuestas alejadas del setpoint. Por esta razón es necesario para sistemas discretos (caracterizados por un tiempo de muestreo T_s) aplicar el criterio de incontabilidad [12], con lo cual se obtiene una tabla corregida con valores que presentan resultados coherentes, mismos que se muestran en la tabla XIII.

Tabla XIII. Valores corregidos de sintonía para PID – discretos.

Método	Kc	Ki	Kd
Ziegler_Nichols	8.490	0.453	39.804
Lambda	0.730	0.004	3.335
Borresen-Grindal	7.075	0.251	33.170
Callender	0.043	0.003	0.141
Chien	6.721	0.301	26.470
Moros	8.490	0.453	33.436
Experimental	7.000	0.100	20.000

3.3.2 Resultados Obtenidos.

Para cada método se guardan los datos usando la herramienta Scope de Simulink. Se grafican las tablas obtenidas a fin de poder comparar las distintas respuestas. El despliegue de las distintas curvas se aprecia en la Fig. 38.

Para verificar la exactitud de las respuestas obtenidas se emplea como patrón de verificación al Fluke Ti32, cuyo certificado de calibración se adjunta en el Anexo A1.3. Los valores verificados son para el LM35 y el software de análisis Simulink, los registros de verificación de adjuntan en el anexo A1.1 y A1.2 respectivamente.

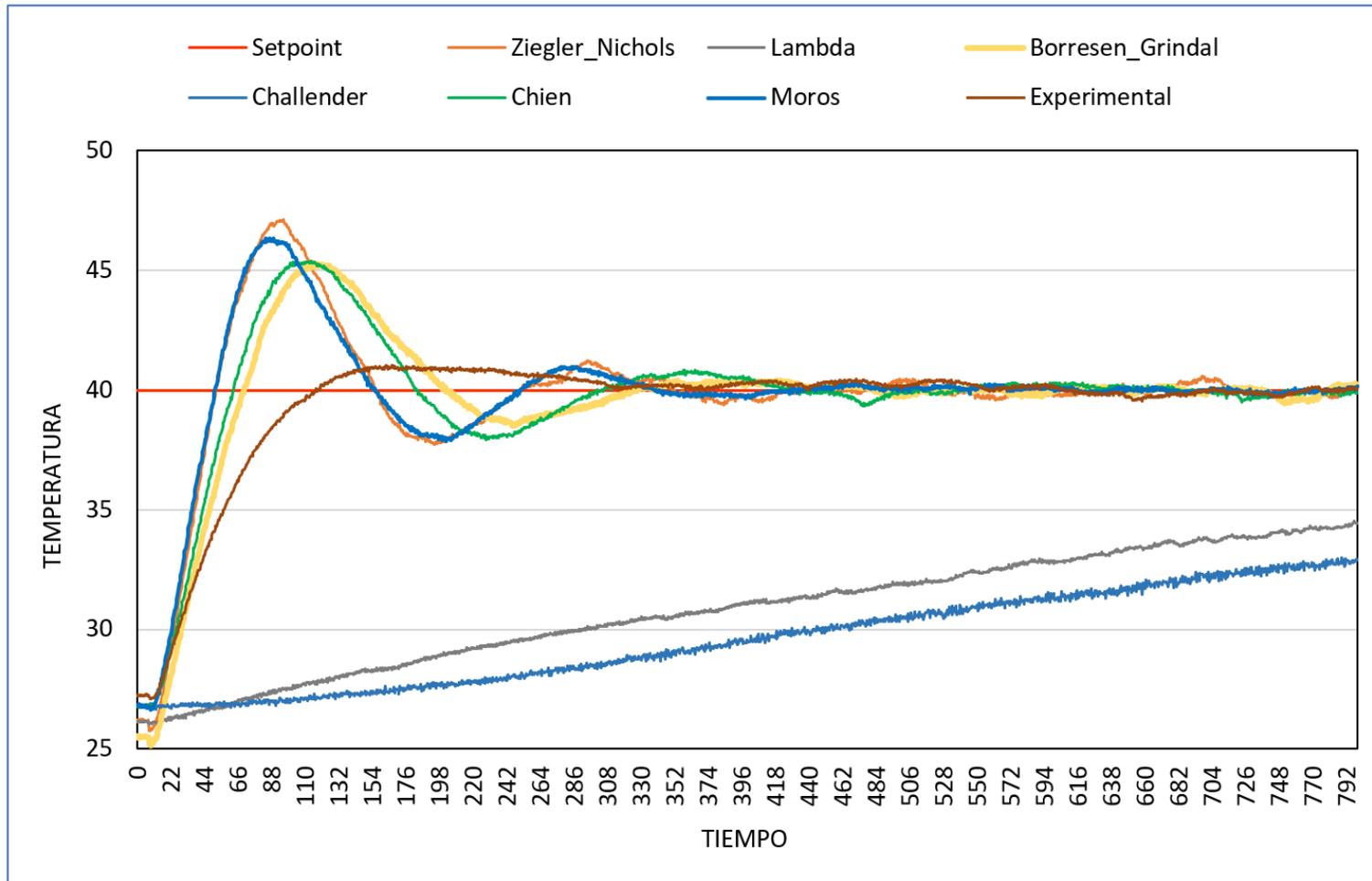


Fig. 38. Sintonización de PID mediante varios métodos.

3.3.3 Análisis usando criterios para estado transitorio y estacionario

De las tablas de datos se extraen los valores para análisis en estado transitorio y estacionario, se plasman los resultados en la tabla XIV.

Tabla XIV. Resultados de la aplicación de métodos de sintonía para control PID.

Método	Respuesta en estado transitorio						Estado estable	
	Set point	Temp. Máxima	Tiempo de retardo	Tiempo de crecimiento	Tiempo de pico	Sobre impulso máximo	Tiempo de estabilización	Error en estado estacionario
	°C	°C	seg.	seg.	seg.	%	seg.	%
Ziegler N.	40	47.12	51.50	40.50	95.50	17.81	309.50	-0.17
Lambda	40	34.53	> 800	>800	>800	-13.67	>800	14.02
Borresen G.	40	45.31	70.50	51.50	116.00	13.27	289.50	-0.79
Callender	40	32.99	>800	>800	>800	-17.54	>800	18.07
Chien	40	45.39	63.00	46.50	114.00	13.47	480.50	0.31
Moros	40	46.35	51.00	38.50	86.50	15.88	295.00	-0.06
Experimental	40	41.03	117.00	64.50	163.00	2.58	239.50	-0.24

Si bien se obtiene una gama de valores, estos nos sirven para tener una idea de los valores dentro del rango de sintonización.

Se puede observar que tenemos tres tipos diferenciados de respuestas,

- Ziegler_Nichols, Borresen_Gridal, Chien y Moros son respuestas sub amortiguadas con picos de temperatura sobre los 45°C, sobre impulso mayor al 10%, un tiempo de estabilización alrededor de 300 segundos y un error estacionario menor al 1%. La diferencia de estos tipos de controladores es que se llega pronto al valor de setpoint pero se tiene una señal oscilante alrededor del mismo, se valida la señal de control una vez que su señal está dentro del rango de error permitido en estado estacionario. Esta característica puede usarse en sistemas que requieren dar permisos a otros sub sistemas, por ejemplo de enfriamiento, mientras se puede esperar hasta que se estabilice completamente el sistema.
- El método experimental, como resultado de las experiencias de los otros métodos permite hallar un punto medio entre controladores rápidos y lentos, denominado críticamente amortiguado, se halla con un mejor tiempo de estabilización de 239 segundos y el sobre impulso apenas de un 2.58%, su

tiempo de retardo es mayor para los de respuesta sub amortiguada en una proporción de casi el doble de valor. La aplicación es muy conveniente para sistema que no soporten sobre impulsos altos, pues se puede llegar a sobrepasar valores de diseño no admitidos para la estructura misma del controlador.

- Los métodos de Lambda y Callender dan una respuesta sobre amortiguada traducida en excesivamente lenta. De la ventana de tiempo de 800 segundos usada para comparar no se llegó a alcanzar el setpoint, aunque la tendencia muestra que, si se puede llegar, la característica es que su parámetro de ganancia integral es muy pequeño, en el orden de milisegundos. Estas propiedades pueden ser aprovechadas en sistemas de calentamiento donde es importante subir la temperatura lentamente para evitar estrés térmico de las partes mecánicas.

3.3.4 Análisis usando índices de desempeño

De los datos desplegados em Simulink se extraen los valores para análisis usando índices de desempeño, se plasman los resultados en la tabla XV.

Tabla XV. Resultados índices de desempeño para PID.

	Ziegler_N.	Lambda	Borresen	Callender	Chien	Moros	Experim.
IAE	22.42	52.01	20.32	176.60	24.35	19.02	24.06
ITAE	2047.00	8152.00	2107.00	6053.00	2578.00	1700.00	2350.00
ISE	4.81	9.61	4.18	42.52	4.91	4.17	4.85
ITSE	132.10	677.80	99.71	1203.00	164.90	81.02	184.30

IAE, este criterio de desempeño es básico y busca el menor valor se sobre amortiguación y una respuesta transitoria satisfactoria, de los valores Moros y Borresen serían los adecuados.

ITAE, Este parámetro es para evaluar pequeños errores, por lo cual es notorio que su valor es alto para método Lambda y Callender cuyas curvas de respuesta no llegaron al setpoint deseado en el tiempo de comparación propuesto. El acumulado ITAE de mayor valor refleja los valores de error significativos en el transcurso del

tiempo. En este caso el de mejor respuesta a este criterio fue Ziegler_Nichols cuyo error en el tiempo de estabilización es menor a los demás.

ISE, penaliza grandes errores, dando valores elevados sobre todo en el sobreimpulso inicial, por esa razón el método de sintonización de Moros alcanza el menor valor ISE acumulado. En la gráfica se observa que se tienen curvas similares entre Ziegler, Borresen, Chien y el Experimental, pero mediante la valorización ISE se puede establecer diferencia en números.

ITSE, permite discriminar permanencia de errores pequeños en el transcurso del tiempo, un buen valor lo alcanza Moros. El método Experimental a pesar de tener una menor sobre amortiguación al inicio, en el transcurso del tiempo no llega a obtener un valor de error menor que Moros, por lo cual ITSE lo penaliza. La comparación de estos dos tipos de controladores es un claro ejemplo de que tener un menor valor de ITSE no necesariamente quiere decir que uno sea mejor que el otro, sino que depende de la aplicación en la que se requiera usar. Es claro que Moros alcanza un menor error en menos tiempo que el Experimental pero su sobreimpulso es mayor. El Experimental se puede usar en procesos que se debe evitar sobre picos de la variable controlada.

El tamaño final del algoritmo de programación fue de **8036 bytes**, es decir un 24% del espacio de almacenamiento del programa cuyo máximo es 32256 bytes.

Las variables globales usan **530 bytes**, es decir un 25% de la memoria dinámica cuyo máximo es 2048 bytes.

La complejidad matemática del PID va desde el uso de ecuaciones no lineales, uso de transformada Z y la aplicación de criterios de sintonía probados y basados en la práctica de varios autores.

3.4 Implementación de controlador con Lógica Difusa

Se implementa un controlador difuso de tipo proporcional integral PI. En este apartado se explica exclusivamente la lógica de control para el bloque de control difuso. Las etapas que se implementan se resumen en la Fig. 39 y a continuación se describe cada una.

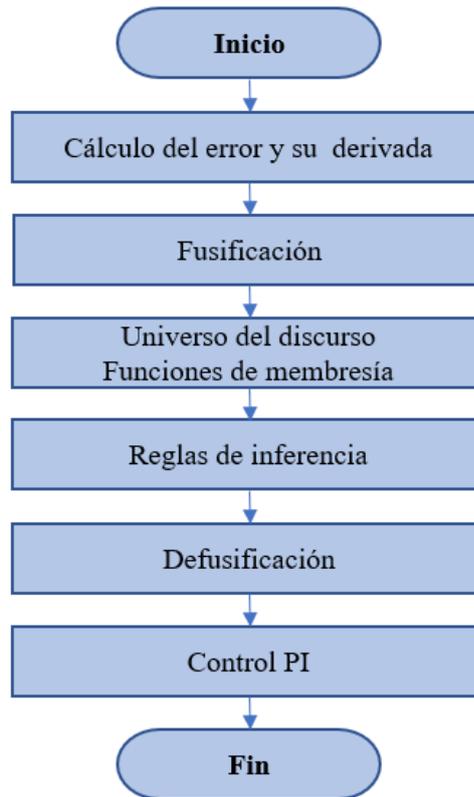


Fig. 39. Diagrama de flujo del controlador PI-Fuzzy.

Cálculo del error y su derivada, el error es la diferencia entre el punto de consigna o setpoint SP y la variable del proceso PV, dentro del presente proyecto se asigna al SP como Tsetp.number y a PV como Tproc.number.

$$e[ke] = Tsetp.number - Tproc.number$$

para la derivada del error en tiempo discreto se tiene:

$$derivada = (e[ke] - e[ke - 1]) / Ts$$

donde $e[ke]$ es error actual, $e[ke-1]$ error anterior, y Ts periodo de muestreo.

Fusificación, en esta etapa se identifican las variables lingüísticas de entrada como son el error y su derivada, se determina también la variable de salida para el control del proceso. A continuación, se definen valores lingüísticos asegurándose de cubrir todo el campo de posibles escenarios de cada variable. En la tabla XVI se presenta la definición propuesta para el control difuso.

Tabla XVI. Definición de variables lingüísticas.

Error	Derivada del error	Salida	Definición
eNMx	deNMx	uNMx	Negativo Máximo
eNMd	deNMd	uNMd	Negativo Medio
eNMn	deNMn	uNMn	Negativo Mínimo
eZ	deZ	uZ	Zero
ePMn	dePMn	uPMn	Positivo Mínimo
ePMd	dePMd	uPMd	Positivo Medio
ePMx	dePMx	uPMx	Positivo Máximo

Universo del discurso, se refiere al rango dentro del cual se definieron las variables del control difuso. Estos valores dependen del diseñador y su experiencia.

Error e , definido como la diferencia entre el punto de referencia y el valor real de la variable del proceso. El valor está entre -100 y 100 considerando el rango de temperatura de 0 a 100 °C y los posibles valores que puede tomar dicho error.

Derivada del error, referida a la razón de cambio del error en el tiempo, esto se ve reflejado en el valor de la pendiente de la curva de respuesta del controlador. Se toma un valor de -1 a 1, sin embargo, se puede evaluar en valores mayores, lo que incide en una respuesta rápida e inestable.

Salida del control u , con un rango de -8 a 8, reflejada en el valor de salida del PWM hacia el elemento de calentamiento de temperatura. La salida toma valores de 0 a 255, que corresponden a la señal entre 0 y 100 % de la salida PWM.

Funciones de membresía, se elaboran las respectivas funciones de membresía, por su menor dificultad de tratamiento matemático se escoge las de tipo triangular tanto para el error Fig. 40, su derivada Fig. 41 y la salida del control Fig. 42. Se colocan valores espaciados para cada definición dada en la Fusificación.

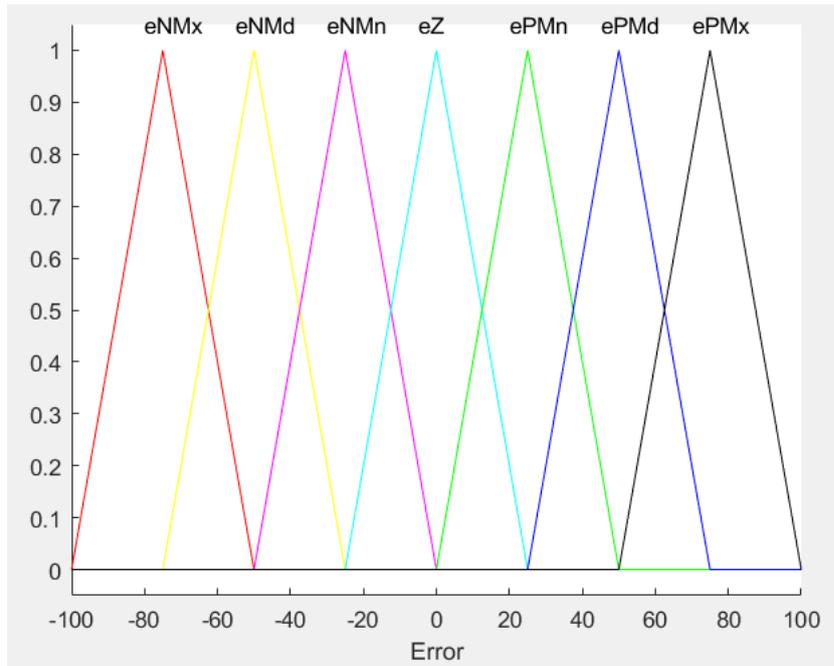


Fig. 40. Función de membresía del error.

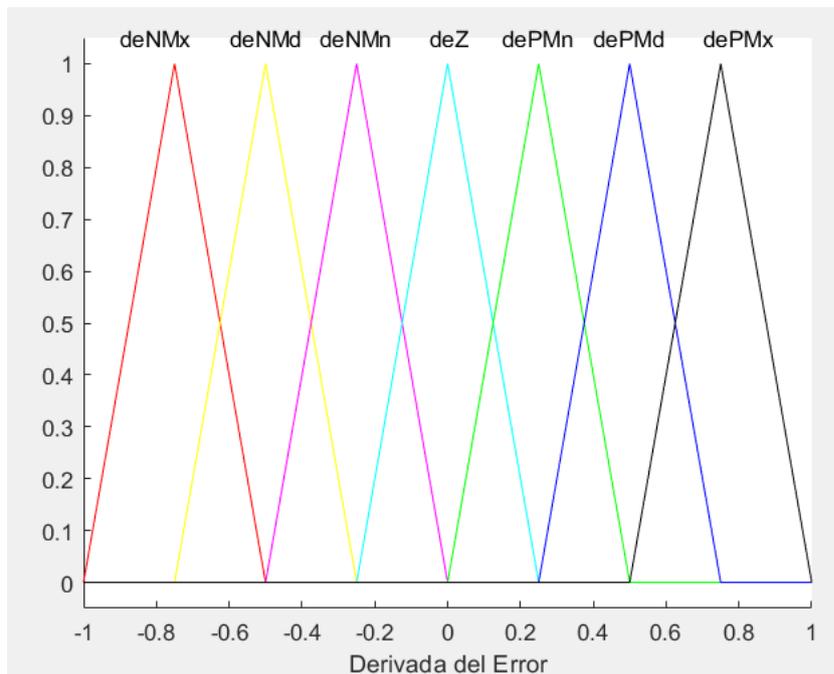


Fig. 41. Función de membresía de la derivada del error.

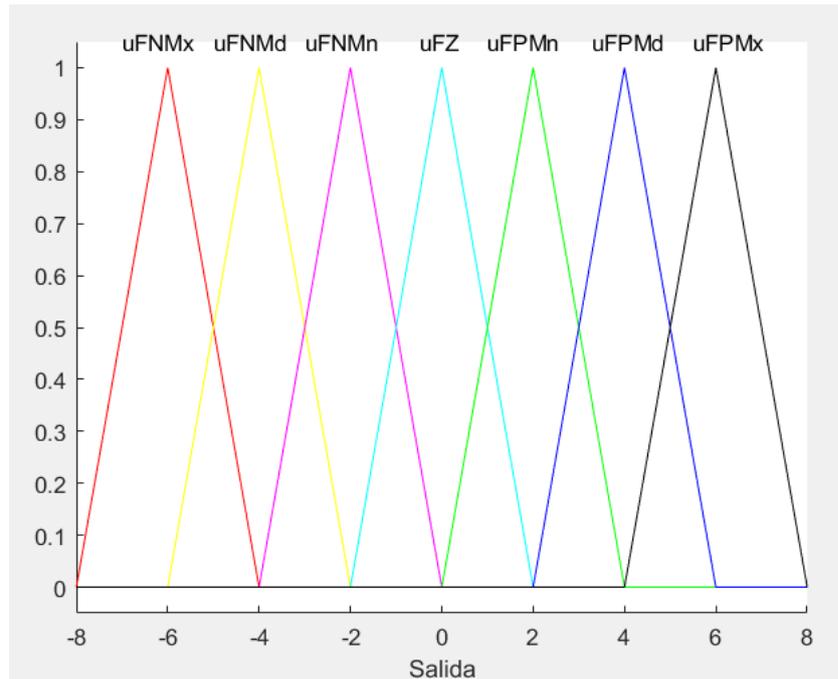


Fig. 42. Función de membresía de la salida.

Reglas de inferencia, se colocan los valores lingüísticos de salida que se desean obtener en base a las combinaciones de las definiciones de las entradas. En la tabla XVII se indica todas las respuestas esperadas a partir del error y su derivada.

Tabla XVII. Reglas de inferencia.

e / de		Derivada del error, de						
		deNMx	deNMd	deNMn	deZ	dePMn	dePMd	dePMx
Error, e	eNMx	uNMx	uNMx	uNMx	uNMx	uNMd	uNMn	uZ
	eNMd	uNMx	uNMx	uNMx	uNMd	uNMn	uZ	uPMn
	eNMn	uNMx	uNMx	uNMd	uNMn	uZ	uPMn	uPMd
	eZ	uNMx	uNMd	uNMn	uZ	uPMn	uPMd	uPMx
	ePMn	uPMd	uNMn	uZ	uPMn	uPMd	uPMx	uPMx
	ePMd	uPMn	uZ	uPMn	uPMd	uPMx	uPMx	uPMx
	ePMx	uZ	uPMn	uPMd	uPMx	uPMx	uPMx	uPMx

Defusificación, en este proceso a la función de salida lingüística se le asignan valores correspondientes a el rango de control. Como se definió en el universo del discurso la salida se le asignó la variable $u[k]$ y toma valores de 0 a 255, que corresponden a la señal entre 0 y 100 % de la salida PWM.

El comando de programación que calcula el valor final es:

$$\text{LeyControl} = \text{ControlFuzzy}(e[ke], \text{derivada}, \text{rango_e}, \text{rango_de}, \text{rango_u})$$

Donde LeyControl es el control defusificado, ControlFuzzy es la función programada para el cálculo difuso, $e[ke]$ es el error actual, derivada corresponde al valor derivativo del error, y a continuación vienen los universos para el error rango_e , para la derivada del error rango_de y la salida de control rango_u .

Control PI, permite manejar los sobre impulsos que puede tener la señal de salida, su línea de control en tiempo discreto es:

$$u[ku] = \text{LeyControl} + u[ku - 1]$$

Donde $u[ku]$ es el valor actual, LeyControl es el control defusificado y $u[ku-1]$ es el valor de la muestra anterior.

Las funciones obtenidas se programan en el microcontrolador Arduino.

3.4.1 Resultados Obtenidos.

De igual manera que para el controlador PID, se registran los datos en Simulink y se los exporta a Excel. Se grafican los resultados y presentan en la Fig. 43. La respuesta obtenida es satisfactoria. La respuesta graficada responde a la programación deseada y las reglas impuestas. El setpoint es alcanzado dentro del tiempo de crecimiento y establecimiento similar a lo obtenido en los controladores PID.

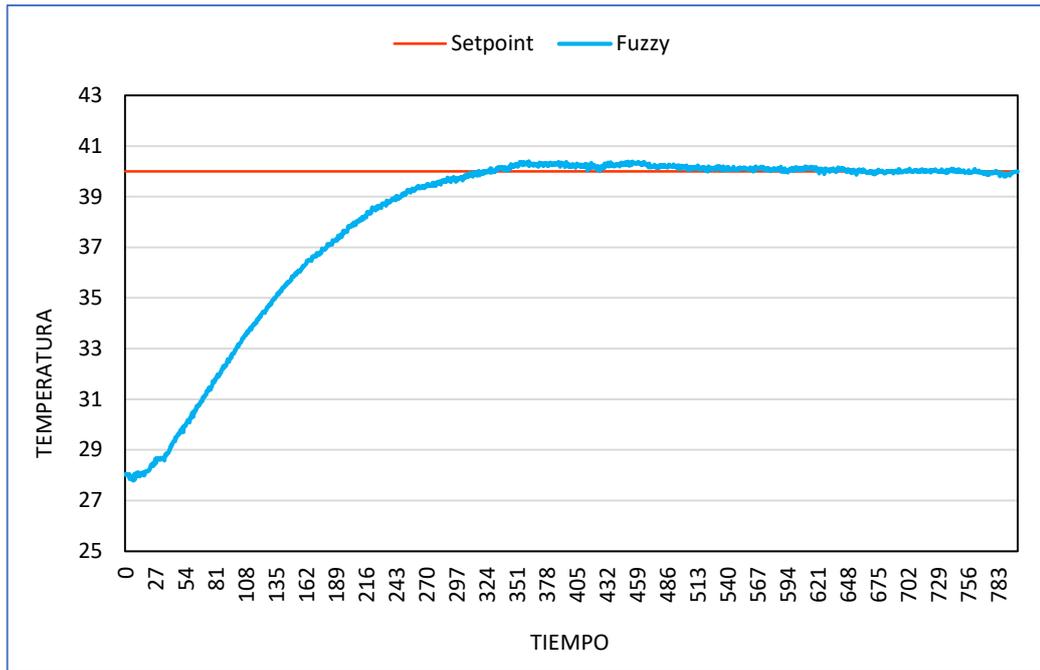


Fig. 43. Respuesta del sistema embebido de temperatura con control difuso.

De los datos obtenidos se extraen los resultados tanto para la respuesta en estado estable como transitorio y se presentan en la tabla XVIII:

Se puede observar que la temperatura máxima alcanzada es menor que los demás métodos de PID, el tiempo de retardo y de crecimiento es mucho mayor sin embargo compensan el obtener un sobre impulso del 1%, que es un resultado excelente.

El error en estado estacionario es de 0.03% que se encuentra muy por debajo de la norma que permite un máximo de 2 a 5%, la estabilización observada del sistema es satisfactoria.

Tabla XVIII. Resultados estado estable / transitorio para control Fuzzy

Método	Respuesta en estado transitorio						Estado estable	
	Set point	Temp. Máxima	Tiempo de retardo	Tiempo de crecimiento	Tiempo de pico	Sobre impulso máximo	Tiempo de estabilización	Error en estado estacionario
	°C	°C	seg.	seg.	seg.	%	seg.	%
Fuzzy	40	40.40	331.00	154.00	362.00	1.01	256.50	0.03

El tiempo de estabilización respecto a un PID es similar, pero se alcanza un error en este estacionario prácticamente nulo.

El análisis de acuerdo con los índices de desempeño indicado en la tabla XIX muestra valores que comparados con los obtenidos con el PID no muestran un valor significativo que permitan una selección determinante, puesto que el tiempo hasta llegar al punto de referencia es lento, y los errores se siguen penalizando, sin importar cuán buena sea la señal de respuesta, en cuanto a sobre impulsos y estabilidad.

Tabla XIX. Resultados de los índices de desempeño PID vs. Fuzzy

	Moros	Fuzzy
IAE	19.02	49.68
ITAE	1700.00	8694.00
ISE	4.17	35.08
ITSE	81.02	14360.00

Dentro de los índices de desempeño se compara el control Fuzzy versus el Control PID, se puede notar que los valores del control difuso son mayores respecto a la sintonía de Moros. El valor ISE que penaliza grandes errores refleja los valores de error al inicio que son significativos en Fuzzy. ITSE en cambio penaliza los errores que mientras más permanezcan en el tiempo los valores se siguen elevando y acumulando. Nuevamente, se nota que esto no quiere decir que el tipo de control sea inadecuado, sino depende de la aplicación que se requiera.

El tamaño final del algoritmo de programación fue de **18912 bytes**, es decir un 58% del espacio de almacenamiento del programa cuyo máximo es 32256 bytes.

Las variables globales usan **659 bytes**, es decir un 32,18% de la memoria dinámica cuyo máximo es 2048 bytes.

La complejidad matemática del Fuzzy va desde el uso de ecuaciones no lineales, uso de transformada Z y la aplicación de comparadores lógicos mediante la técnica IF -THEN.

La tabla comparativa del tamaño de programación tanto para los controladores PID y Fuzzy se indica a continuación:

Tabla XX. Resultados del uso de memoria para los controladores PID y Fuzzy

Memoria	Unidad	PID	Fuzzy	
Flash	Máx	Bytes	32256	32256
	Usado	Bytes	8036	18912
	%		24.91	58.63
RAM	Máx	Bytes	2048	2048
	Usado	Bytes	530	659
	%		25.88	32.18

Frente a la memoria Flash (no volátil) máxima del dispositivo Arduino UNO, la cantidad de memoria usada para PID es un 24.91%, mientras que para Fuzzy es 58.63%, la cantidad de instrucciones implementadas en Fuzzy es mayor. De igual manera el uso de RAM da un 25.88% para PID frente a un 32.18% del Fuzzy.

En el caso de requerir adaptar los recursos desarrollados en este proyecto se debe tener en cuenta algunas consideraciones dependiendo de la plataforma de software y hardware que se quiera utilizar. En la Tabla XXI se muestran las características de distintos modelos de microcontroladores similares a Arduino UNO.

Tabla XXI. Características de distintos modelos de procesadores

Tipo	Procesador	No. E/S	No. ADC	Flash /RAM kB	Frec. Mhz	Volt. VDC	Protoc I2C	No. PWM	Precio US\$.
UNO R3	ATmega328p	14	6-10 bits	32/2	16	5	si	6	18
ESP8266	XtensaLX106	16	1-10 bits	16000/160	80	3.3	si	1	11.8
ESP32	XtensaLX6	36	18-12bits	16000/512	160	3.3	si	18	13
Raspberry PI Model B	ARM 11	8	N/A	SD /256	700	5	si	N/A	59
Raspberry PI Pico	Arm Cortex-M0+	26	3-12bits	2000/264	133	5	si	16	15
P. Embebido de Temp req.		2	2-10 bits	19/0.66	-	5	si	1	-

Las consideraciones son las siguientes:

- Se usa un solo conversor análogo-digital con una resolución de 10 bits (máxima lectura digital 1023), para casos como el ESP32 o el Raspberry PI

Pico disponen un ADC de 12 bits (máxima lectura digital 4095), por lo que se debe reformular el escalamiento de señal.

- EL voltaje de salida en los terminales análogos es de 5V, los modelos ESP8266 y ESP32 manejan 3.3V por lo que se debe tener en cuenta que la planta en su variable de control requiere los 5 V para alcanzar el 100% de control. Modelos antiguos como el Raspberry PI modelo B, inclusive no disponen de salida análoga.
- El tamaño para la memoria Flash es de 19kB, y de la RAM de 0.66 kB por lo que cualquier dispositivo de mayor capacidad lo suplirá sin inconveniente.
- La frecuencia del procesador incide en la cantidad de instrucciones y cálculos aritméticos que procesa, un mejor procesador realizará más operaciones en un periodo determinado. El tiempo de muestreo se calcula a partir de la respuesta en lazo abierto para un escalón unitario, la filosofía del tiempo de muestreo es permitir la recuperación de una señal análoga a partir de datos discretizados. Para el presente proyecto el tiempo de muestreo calculado es de 8 a 16 segundos por lo cual la velocidad del microprocesador no incide.
- Otra característica a observar, es el protocolo de comunicación I2C que se usa para poder desplegar datos hacia el LCD. En caso de no disponer de este protocolo se debe realizar el cableado tradicional usando todos los pines del LCD.
- El presente proyecto usa una salida PWM para la señal de control, en caso de disponer de un microcontrolador que no disponga de esta prestación, se deberá programar, un ejemplo es el modelo antiguo Raspberry Pi modelo B.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se investiga distintos tipos de controladores existentes en el mercado y se seleccionan a los controladores PID y Fuzzy como los más representativos y usados en la industria. El control PID como base de todos los controladores clásicos y el Fuzzy en base a las nuevas tecnologías desarrolladas. Para el sistema embebido de temperatura aplicando ambos tipos de control se determina mayor complejidad de programación para el Fuzzy con 18912 bytes frente a 8036 bytes promedio de un PID. De acuerdo con los criterios de desempeño y comparando PID se obtiene un error absoluto integral de 19.02 frente al control Fuzzy de 49.68, esto indicaría un mejor control mediante el uso de PID, pero en realidad siempre depende de la aplicación en que se requiera usar el controlador. Se determina también que hay complejidad matemática usada para el desarrollo de los algoritmos para el control digital, ambos controladores emplean ecuaciones no lineales, transformadas Z directa e inversa, sin embargo el PID usa métodos de sintonía basados en múltiples pruebas de sus autores, Fuzzy se basa en usar comparadores lógicos y reglas, todas ellas dependientes de la experiencia del programador.
- Se diseña el módulo embebido de temperatura basados en las propiedades para conducción eléctrica y disipación térmica del transistor TIP31C, se

escoge el rango de 0 a 100 °C. Se selecciona al sensor de temperatura LM35 cuyo costo beneficio es inmejorable, el controlador físico usa al Arduino por disponer de hardware y software libre. Para realizar el análisis se emplea a Matlab - Simulink y se aprovecha las herramientas avanzadas incorporadas en sus últimas versiones.

- Se realizan pruebas de respuesta de temperatura del sistema embebido utilizando distintos criterios de sintonización en un control PID. Así mismo se observa la respuesta para el control Fuzzy. Se capturan datos en Matlab y se los tabula para el análisis en estado transitorio y estable. Se presentan en una tabla los parámetros de temperatura máxima, tiempo de retardo, tiempo de crecimiento, tiempo de pico, sobreimpulso máximo, tiempo de estabilización y error en estado estacionario. Se realiza gráfica de cada tipo de control y con lo cual se puede definir tres grupos de respuestas: subamortiguada, críticamente amortiguada y sobreamortiguada , cada uno depende de sus valores de diseño iniciales.
- Se implementa en Simulink una pantalla de análisis para evaluar los criterios de desempeño basados en los cálculos de la integral del error
- Se han analizado distintos criterios de desempeño para los controladores y se concluye que el seleccionar al mejor de ellos no solo dependen de una determinada valoración sino de también de las necesidades del proceso. Se puede requerir alcanzar un mínimo error en el menor tiempo posible o simplemente llegar de manera sostenida al valor de consigna.
- Muchos estudios utilizan computadoras y software especializado para obtener la respuesta de control y el controlador usado se emplea únicamente como interface de datos. El diseño de un proceso embebido de temperatura conlleva en implementar los algoritmos necesarios dentro del controlador lógico, esto en la actualidad ha sido posible gracias al aparecimiento de hardware y software libre como lo es le Arduino, obteniendo una implementación a bajo costo y permitiendo el entendimiento del funcionamiento de la programación, así como la verificación de respuestas a distintas filosofías de control.

4.2 Recomendaciones

- A partir del presente estudio que usa Arduino UNO, se puede investigar aplicaciones con mejores prestaciones para la implementación de controladores digitales para tarjetas embebidas, Arduino UNO tiene una memoria máxima de 32 kbytes, y una velocidad de procesamiento de 16MHz y ha sido diseñado para uso como controlador, no como computador de cálculo avanzado, por lo que resulta interesante la aplicación en sistemas microprocesados como Raspberry o similares.
- Los controladores digitales son desarrollados en base a algoritmos de programación, se puede continuar el estudio con controles avanzados como el Predictivo, Cascada u otros.
- Actualmente ya se disponen de herramientas para análisis con código libre, Matlab por ejemplo puede ya reemplazarse por Octave, quien dispone de un ambiente de programación similar, aunque todavía se deberá esperar un desarrollo similar para Simulink. En todo caso, diariamente se suman nuevos aportes de código libre que los podemos usar y simplificar muchos cálculos y rutinas de programación.

BIBLIOGRAFÍA

- [1] C. Betancor, J. Cerezo, y A. Vega, “Diseño de un sistema de control de temperatura”, *VII Congr. TAAE-2006*, pp. 1–9, 2006, [En línea]. Disponible en: <http://e-spacio.uned.es/fez/eserv/taee:congreso-2006-1116/S3F04.pdf>.
- [2] C. Muñoz, “Aplicación y comparación de sistemas de control PI y Fuzzy en espesador piloto”, Universidad de Concepción, Chile, 2019.
- [3] E. Basantes y P. Calapiña, “Desarrollo de un sistema web server para el monitoreo de una estación de nivel”, Universidad Técnica del Cotopaxi, Ecuador, 2020.
- [4] Y. G. Fuertes, “Diseño e implementación de un módulo educativo para el control de temperatura”, Pontificia Universidad Católica del Perú, Perú, 2017.
- [5] A. Ferreyra y R. Fuentes, “Estudio Comparativo entre Control PID y Difuso”, *XV SOMI Congr. Instrumentación*, vol. CON-4, n° 52, p. 6, 2000.
- [6] C. Saeteros, “Implementación de un control Fuzzy para la regulación de nivel del módulo MPS PA Compact Workstation mediante la tarjeta STM32F4 Discovery”, Universidad Politécnica Salesiana Sede Quito, Ecuador, 2018.
- [7] J. G. Álvarez Díaz, J. M. Armero Viveros, y C. A. Urrutia, “Control de temperatura en intercambiadores de calor tipo coraza-tubo: una revisión realizada a la industria”, *Entre Cienc. e Ing.*, vol. 14, n° 27, pp. 41–49, 2020, doi: 10.31908/19098367.1763.
- [8] K. Ogata, *Ingeniería de Control Moderna*, 5a ed. Madrid, España: Pearson Educación SA, 2013.
- [9] A. Creus, *Instrumentación industrial*, 8a ed. México: Alfaomega, 2010.
- [10] F. Vera, “Metodología para modelar y sintonizar lazos de control del proceso, enfocado a minimizar el consumo energético en el recipiente (U_V5) en la Unidad de tratamiento de gas de Petroecuador Esmeraldas”, Escuela Superior Politécnica de Chimborazo, Ecuador, 2020.

- [11] E. Adam, *Instrumentación y control de procesos: notas de clase*, 3a ed., vol. 148. Colombia: Ediciones Universidad Nacional del Litoral, 2020.
- [12] L. García, *Control Digital, Teoría y práctica.*, 2a ed. Colombia: Politécnico Colombiano JIC, 2009.
- [13] Sapiensman, “Controladores de procesos industriales”, 2019. http://www.sapiensman.com/tecnoficio/electricidad/instrumentacion_industrial4.php (accedido sep. 12, 2021).
- [14] J. Jaramillo y F. Vaca, “Implementación de un sistema de control robusto para seguimiento de trayectoria de tres cuadricópteros en formación”, Escuela Politécnica Nacional, Ecuador, 2018.
- [15] B. P. Corrales, “Desarrollo de un entorno virtual 3D que simule una estación de control de temperatura, mediante la técnica de ‘Hardware in the Loop’, orientado al entrenamiento de Control Automático, en la Universidad Técnica de Cotopaxi”, Universidad de las Fuerzas Armadas ESPE, Ecuador, 2022.
- [16] A. Rosillo Meseguer, “Montaje, programación y puesta en marcha de un robot neumático de estructura paralela.”, Universidad Politécnica de Cartagena, Colombia, 2008.
- [17] O. Aidan, *Handbook of PI and PID controller tuning rules*, 3rd. ed. Londres, Inglaterra: Imperial College Press, 2009.
- [18] Facultad de Tecnología y Ciencias Aplicadas - Universidad Nacional de Catamarca, “Qué es un sistema embebido?”, *Laboratorio de Sistemas Embebidos*, 2016. <https://lasetecno.blogspot.com/p/que-es-un-sistema-embebido.html> (accedido sep. 12, 2021).
- [19] RedHat, “What is open hardware?”, *Open source resources*, 2014. <https://opensource.com/resources/what-open-hardware> (accedido sep. 12, 2021).
- [20] Y. Fernández, “Qué es Arduino, cómo funciona y qué puedes hacer con uno”, *Xataka Basics*, 2020. <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno> (accedido sep. 12, 2021).

- [21] O. Alzate, “Arduino blink”, *Código Electrónica*, 2020. <http://www.codigoelectronica.com/blog/arduino-blink> (accedido sep. 12, 2021).
- [22] MathWorks, “Control merging signals with the push button block”, *Help Center*, 2021. <https://la.mathworks.com/help/simulink/slref/dashboard-merge.html> (accedido sep. 12, 2021).
- [23] Vishay Beyschlag, “Thermal management in surface-mounted resistor applications”, *Resist. Prod. Appl. note No. 28844*, pp. 1–6, 2011, [En línea]. Disponible en: <http://www.vishay.com/docs/28844/tmismra.pdf>.
- [24] E. Coluccio, “15 Ejemplos de conducción, convección y radiación”, *Enciclopedia de ejemplos*, 2022. <https://www.ejemplos.co/10-ejemplos-de-conduccion-conveccion-y-radiacion/> (accedido feb. 10, 2022).
- [25] A. Pinilla, “Simulación termomecánica de un proceso de soldadura”, Universidad Politécnica de Madrid, España, 2017.
- [26] F. Miyara, “Disipación de potencia”, *Electrónica III, 2a ed.*, p. 22, 2006, Accedido: mar. 02, 2022. [En línea]. Disponible en: <http://www.fceia.unr.edu.ar/enica3/disipa.pdf>.
- [27] The MathWorks Inc, “Tools for model-based design”, *Model. Des. Embed. Control Syst.*, pp. 1–14, 2020, [En línea]. Disponible en: <https://la.mathworks.com/campaigns/offers/model-based-design-embedded-control-systems.html>.
- [28] R. Sampieri, C. Fernández, y P. Baptista, *Metodología de la Investigación*, 6a ed. México: McGraw-Hill, 2014.
- [29] Continental Device India Limited, “Transistor TIP31”, *TIP31 Datasheet*, vol. Rev. 32, pp. 1–4, 2002.
- [30] T. Instrument, “LM35 Precision centigrade temperature sensors”, *LM35 Datasheet*, pp. 1–13, 2017, [En línea]. Disponible en: <https://www.ti.com/lit/ds/symlink/lm35.pdf>.
- [31] J. Tane y D. Jérôme, “TimerOne”, *Arduino Library List*, 2015.

- <https://www.arduinolibraries.info/libraries/timer-one> (accedido nov. 23, 2022).
- [32] Electroall, “LiquidCrystal_I2C.zip”, *Electroall / Códigos arduino*, 2018. https://github.com/ELECTROALL/Codigos-arduino/blob/master/LiquidCrystal_I2C.zip (accedido oct. 26, 2022).
- [33] D. Hou, “PID control on PIC16F161X by using a PID peripheral”, *Microchip TB3136*, p. 8, 2015, [En línea]. Disponible en: <https://ww1.microchip.com/downloads/en/Appnotes/90003136A.pdf>.
- [34] G. Montenegro, “Identificación de sistemas con Matlab”, *Universidad Técnica Federico Santamaría, Chile*, 2017. <https://www.youtube.com/watch?v=HvCcFcMEL7E> (accedido jun. 30, 2022).
- [35] M. Naylamp, “Tutorial LCD con I2C, controla un LCD con solo dos pines”, *Tutoriales*, 2021. https://naylampmechatronics.com/blog/35_tutorial-lcd-con-i2c-controla-un-lcd-con-solo-dos-pines.html (accedido oct. 26, 2022).

ANEXOS

ANEXO 1. VALIDACIÓN DE DATOS

A1.1 Registro de verificación de calibración del sensor LM35

Registro de calibración		UTC-001		Método		Comparación directa		
Características metroológicas del instrumento a calibrar								
Tipo	Sensor de temperatura de circuito integrado			Rango	0-150 [°C]			
Marca	Texas Instruments			Resolución	+ / - 0.25 °C			
Modelo	LM35			Apreciación	10mV/ °C			
Serial	N/A			Identificación	U2			
Condiciones Generales								
Hora	21h00			Fecha	9-Sep-22			
Temperatura	Inicial	18 [°C]	Final	18 [°C]				
Humedad	Inicial	81 [%]	Final	81 [%]				
Datos del patrón de calibración								
Patrón utilizado	Cámara Térmica			Unidad medida	°C			
Marca	Fluke			Rango	-20 a 600 °C			
Modelo	Thermal Imager Ti32			Resolución	0.1 °C			
Serial	Ti32-11120187			Certificado	LT-2020-086			
Incertidumbre	k = 2			Código	NPF			
Mediciones								
Ref. de temperatura [°C]	Lecturas					Promedio [°C]	Desviación instrumento [°C]	Error instrumento [%]
	No. Unid.	1	2	3	4			
26.5	Patrón	26.3	26.4	26.4	26.4	26.38	-0.15	0.57
	Instr.	26.5	26.5	26.5	26.6	26.53		
30	Patrón	30.7	30.8	30.8	30.8	30.78	-0.02	0.08
	Instr.	30.8	30.8	30.7	30.9	30.80		
40	Patrón	40.4	40.4	40.4	40.4	40.40	0.52	1.30
	Instr.	39.9	39.8	39.9	39.9	39.88		
50	Patrón	50.6	50.6	50.6	50.6	50.60	0.53	1.04
	Instr.	50.1	50	50.1	50.1	50.08		
60	Patrón	60.2	60.2	60.2	60.2	60.20	0.43	0.71
	Instr.	59.8	59.9	59.7	59.7	59.78		
70	Patrón	70.5	70.5	70.6	70.4	70.50	0.58	0.82
	Instr.	69.9	70.1	69.9	69.8	69.93		
80	Patrón	79.9	79.9	79.9	80.1	79.95	1.10	1.38
	Instr.	78.8	78.8	78.9	78.9	78.85		
90	Patrón	90.6	90.4	90.5	90.5	90.50	1.05	1.16
	Instr.	89.6	89.6	89.5	89.1	89.45		
100	Patrón	101.5	101.5	101	102	101.50	-0.28	0.27
	Instr.	101.4	101.6	102	102.1	101.78		
Observaciones								
Se tuvo una desviación máxima de 1.1 °C, correspondiente al 1.37%, el sensor se puede usar y se puede mejorar compensando en programación del controlador.								
Error instrumento = [(Lectura Patrón -Lectura Instr.)/(Lectura Patrón)] x 100 (%)								
Responsables								
Realizado por				Revisado por				
Nombre	Juan Carlos Remache			Nombre	Paúl Corrales			
Firma				Firma				
Fecha	10-Sep-22			Fecha	10-Sep-22			

A1.2 Registro de verificación de calibración del software de análisis:

Registro de calibración	UTC-002				Método	Comparación directa		
Características metrológicas del instrumento a calibrar								
Tipo	Software de análisis				Rango	0-100 [°C]		
Marca	MathWorks				Resolución			
Modelo	Matlab/ Simulink				Apreciación			
Serial	Licencia 968398				Identificación	R2018b Update 8		
Condiciones Generales								
Hora	21h00				Fecha	9-Sep-22		
Temperatura	Inicial	18	[°C]	Final	18	[°C]		
Humedad	Inicial	81	[%]	Final	81	[%]		
Datos del patrón de calibración								
Patrón utilizado	Cámara Térmica				Unidad medida	°C		
Marca	Fluke				Rango	-20 a 600 °C		
Modelo	Thermal Imager Ti32				Resolución	0.1 °C		
Serial	Ti32-11120187				Certificado	LT-2020-086		
Incertidumbre	k = 2				Código	NPF		
Mediciones								
Ref. de temperatura	Lecturas					Promedio	Desviación instrumento	Error instrumento
	[°C]	No.	1	2	3			
	Unid.	[°C]				[°C]	[°C]	[%]
26.5	Patrón	26.30	26.30	26.30	26.30	26.30	0.27	1.04
	Instr.	26.15	26.08	25.91	25.97	26.03		
30	Patrón	30.80	30.80	30.80	30.80	30.80	0.42	1.36
	Instr.	30.51	30.32	30.44	30.26	30.38		
40	Patrón	40.40	40.40	40.40	40.40	40.40	0.42	1.03
	Instr.	40.04	39.96	40.01	39.92	39.98		
50	Patrón	50.10	50.10	50.10	50.10	50.10	0.01	0.02
	Instr.	49.95	50.05	50.38	49.97	50.09		
60	Patrón	60.20	60.20	60.20	60.20	60.20	0.02	0.04
	Instr.	60.20	60.29	60.21	60.01	60.18		
70	Patrón	69.90	69.90	69.90	69.90	69.90	-0.20	0.29
	Instr.	70.30	70.13	69.92	70.06	70.10		
80	Patrón	79.70	79.70	79.70	79.70	79.70	0.06	0.07
	Instr.	79.81	79.27	79.58	79.91	79.64		
90	Patrón	90.20	90.10	90.00	89.80	90.03	0.01	0.01
	Instr.	90.05	90.07	89.99	89.94	90.01		
100	Patrón	100.10	100.60	100.70	100.80	100.55	0.65	0.65
	Instr.	100.00	99.80	99.90	99.90	99.90		
Observaciones								
La mayor desviación es 0.65 y un error del 0,64%, menor al del sensor LM35.								
Se observa mayor inestabilidad en la medida debida al tipo de control usado en el sistema embebido.								
Error instrumento = [(Lectura Patrón -Lectura Instr.)/(Lectura Patrón)] x 100 (%)								
Responsables								
Realizado por					Revisado por			
Nombre	Juan Carlos Remache				Nombre	Paúl Corrales		
Firma					Firma			
Fecha	10-Sep-22				Fecha	10-Sep-22		

A1.3 Certificado de calibración del instrumento patrón de medida



LABORATORIO DE TEMPERATURA CERTIFICADO DE CALIBRACIÓN

Página 01 de 01

1.- INFORMACION Y DATOS		CERTIFICADO N°: LT-2020-086
Cliente: CONSORCIO PETROLERO BLOQUE 16		Fecha de recepción: 2020-02-27
Solicitante: Ing. Javier Mosquera		Fecha de calibración: 2020-02-27
Dirección: Isabel La Católica N24-430 Y Luis Cordero, Torre Rf.		Fecha de emisión: 2020-03-02
Teléfono: 02 976600		

2.- CONDICIONES AMBIENTALES		Lugar de calibración: Tecniprecisión
Temp. Amb.: (23 ± 5) °C		
Humedad Relativa: (47.5 ± 12.5) % HR		

3.- IDENTIFICACIÓN DEL EQUIPO BAJO CALIBRACIÓN (EBC)		Unidad de medida: °C
Equipo: CÁMARA TÉRMICA		Rango: -20 °C a 600 °C
Marca: FLUKE		Resolución: 0,1°C
Modelo: TI 32		Código: NPF
Serie: T132-11120187		

4.- TRAZABILIDAD

Método Utilizado: Método de calibración bajo criterios de la Norma ASTM E2847, por comparación directa LCT-PTCI-01.
Incertidumbre de Medida: El cálculo de la incertidumbre se estimó multiplicando la incertidumbre típica por el factor de cobertura (k=2), que para una distribución t de Student que corresponde a una probabilidad de cobertura de aproximadamente el 95%. Según documento "Guide to the Expression of Uncertainty in Measurement" de la ISO.

Patrones utilizados:

Patrón Utilizado:	Marca:	Modelo:	Serie:	Fecha de Calibración:
CALIBRAIDOR DE INFRARROJOS	FLUKE	9132	B26926	2019-04-23
Trazabilidad:	Certificado de Calibración:			Próxima Calibración:
C.M.E.E.	TE190196CIR			2021-04

5.- RESULTADOS DE LA CALIBRACIÓN

TEMPERATURA °C				
ITEM #	LECTURA PATRÓN	LECTURA MEDIDA (EBC)	CORRECCIÓN °C	INCERTID. k=2 (±°C)
1	30,49 °C	33,6	-3,1100	1,6
2	150,04 °C	152,1	-2,0400	1,7
3	300,07 °C	299,4	0,6700	2,0
4	400,09 °C	396,2	3,8700	2,1

6.- OBSERVACIONES

- Los resultados obtenidos hacen referencia únicamente al ítem mencionado en el punto 3 del presente documento.
- Este certificado tiene validez únicamente en su forma íntegra y original, no se permite la reproducción parcial o total de este documento, sin la autorización por escrito de Tecniprecisión Cia. Ltda.
- Los resultados del presente certificado deben interpretarse como antes y después de la calibración, ya que no se realizan ajustes o reparaciones durante el proceso.
- La estabilidad y funcionamiento del Equipo, depende de varios factores, los cuales están fuera del control de nuestros laboratorios
- Este certificado solo se aplicará para el ítem identificado, únicamente se podrá reproducir en forma completa y con la aprobación escrita específica de Tecniprecisión.
- La fecha de próxima calibración se incluye únicamente cuando el cliente lo haya solicitado, el Laboratorio no incluye recomendaciones sobre intervalos de próxima calibración. Norma ISO 17025:2017, literal 7.8.4.3.

7.- FIRMAS DE RESPONSABILIDAD

Calibrado por: Tnlgo. Marco Caiza
Técnico de Laboratorio de
Temperatura



Autorizado por: Ing. Fernando Avilés
Director de Laboratorio

Firma:

FIN DEL CERTIFICADO

LCT-PTCI-01-REV-00.2020
 Av. Gelo Plaza Lasso N65-85 y Bellavista. Edificio Morb 3er piso
 Sector Parque de los Recuerdos - Guito - Ecuador
 Tel: 593 02 6 035 811 / 3 464 324 / 6 001 375
 Cel: 0987 836 855 / 0984 950 765
 E-mail: ventas@tecniprecision.com / calidad@tecniprecision.com
 laboratorio@tecniprecision.com / asistente@tecniprecision.com
 facebook / Tecniprecisión Cia. Ltda. - Página Oficial

LABORATORIO DE METROLOGIA ECUATORIANO
 www.tecniprecision.com

A1.3 Registro fotográfico



Fig. 44. Validación de medición y calibración de temperatura.



Fig. 45. Instrumento patrón: Cámara termográfica Fluke Ti32.

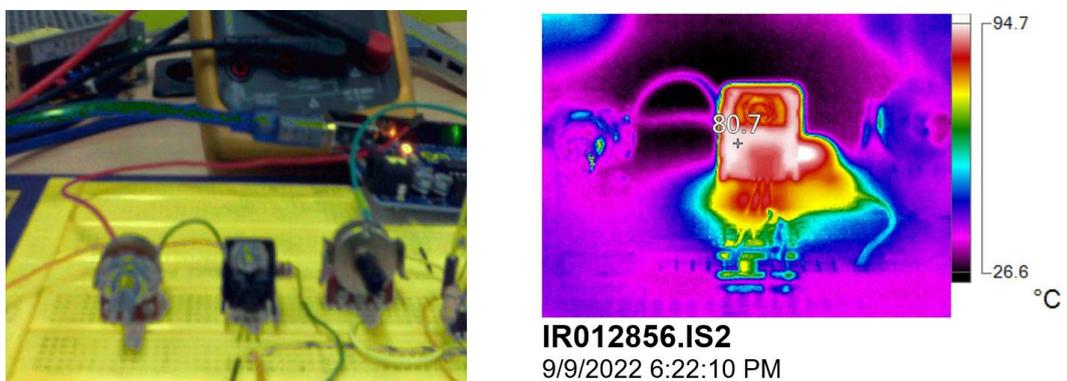


Fig. 46. Imágenes captadas por la cámara termográfica.

ANEXO 2. FUNCIÓN DE TRANSFERENCIA USANDO MATLAB

A2.1 Obtención de datos de la planta en lazo abierto

Utilizando el ambiente de análisis creado en Simulink, se coloca como entrada el valor de la señal tipo escalón es decir el setpoint de temperatura, se escogió 40 °C (Tsetp_int); luego en Arduino se escala el valor del setpoint para la respectiva salida PWM cuyo canal se denominó como Tsetp_CAN. La instrucción para escalamiento en Arduino se llama map y se escribe de la siguiente manera :

```
analogWrite(Tsetp_CAN, map(Tsetp_int,0,100,0,255));
```

A continuación, se configura la herramienta Scope para almacenar datos, para ello se selecciona “Log data to workspace”, se asigna un valor de variable por ejemplo “Data” y se guarda en formato “Structure with Time” como lo indica la Fig. 47.

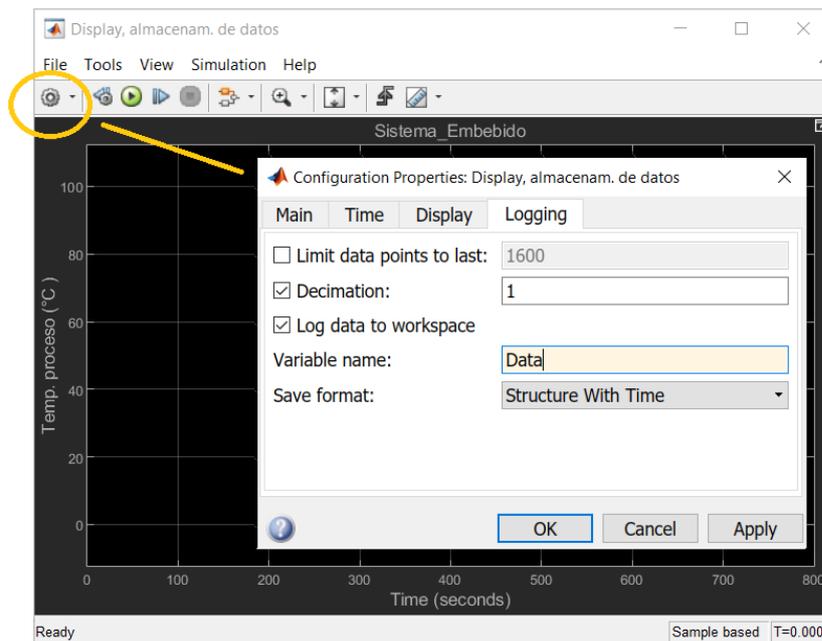


Fig. 47. Configuración para almacenamiento de datos en Scope.

La selección mencionada hará que se almacenen los datos cuando se detenga el modo Run del programa, inmediatamente en Workspace de Matlab aparecerá el archivo de datos con el nombre de la variable “Data” como se muestra en Fig. 48,

se debe guardar dicho dato para lo cual nos solicitará un nombre, para el caso presente se nombra como “Datalogging” y este se guardará con la extensión *.mat

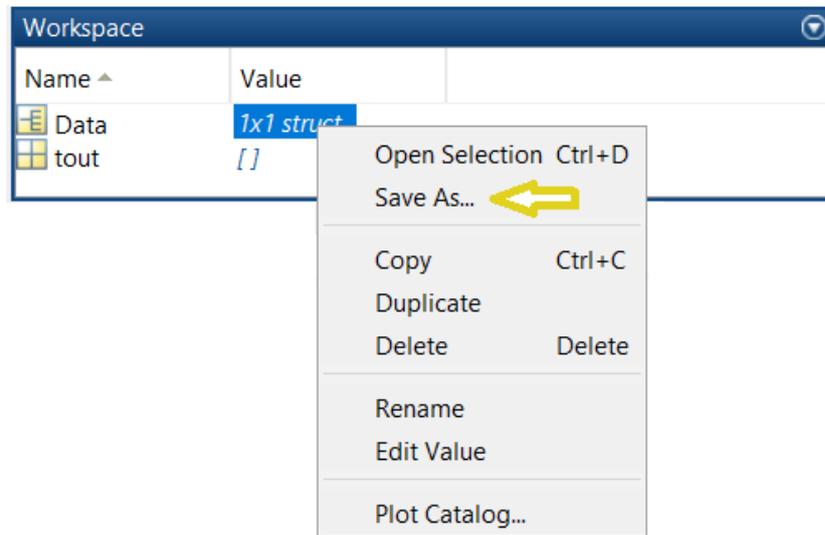


Fig. 48. Almacenamiento de datos en Workspace.

A2.2 Separación de datos de una tabla almacenada en Matlab

Una vez que se han almacenado los datos desde el “Datalogging.mat”, es necesario disponerlos separadamente de la siguiente manera:

- a) En el Command Window de Matlab, ver Fig. 49, se carga el archivo de datos guardado de extensión *.mat, por ejemplo ‘DataLogging.mat’:

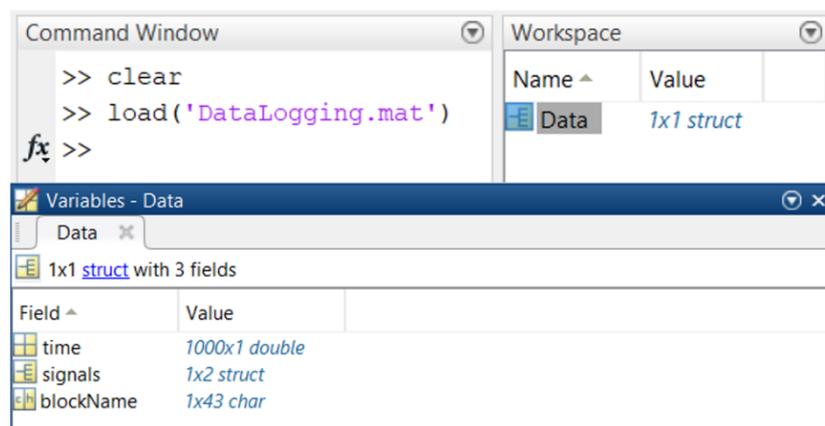
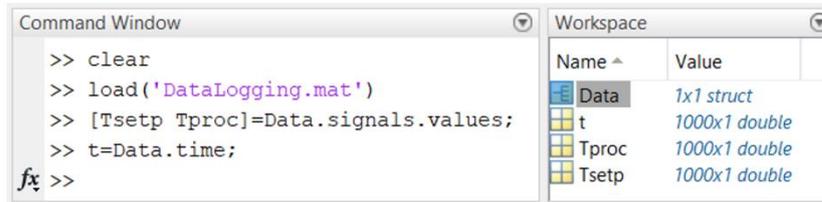


Fig. 49. Carga del registro de datos.

- b) Se almacena en vectores las variables: setpoint “Tsetp”, la respuesta obtenida del proceso “Tproc”, y el tiempo de muestreo “t”; se usan las propiedades del objeto Data, en este caso accediendo a las propiedades de señales y valores [34], como se muestra en Fig. 50:



```
>> clear
>> load('DataLogging.mat')
>> [Tsetp Tproc]=Data.signals.values;
>> t=Data.time;
fx >>
```

Name	Value
Data	1x1 struct
t	1000x1 double
Tproc	1000x1 double
Tsetp	1000x1 double

Fig. 50. Extracción de valores del archivo Datalogging.mat

- c) Como comprobación se puede dibujar las señales, mediante el uso del comando plot, dicha instrucción se muestra en Fig. 51.

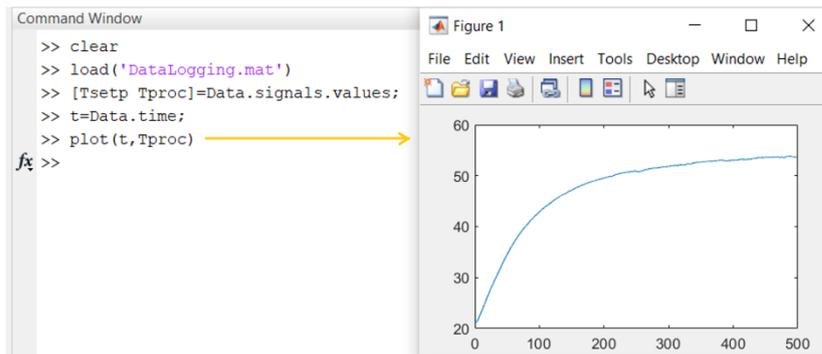


Fig. 51. Comprobación de datos almacenados en tabla de Matlab.

Es necesario recalcar que los datos almacenados en el Datalogging deben ser del tipo double para poder desplegar la respectiva gráfica con el comando plot.

A2.3 Identificación de la función de transferencia en Matlab

- a) Una vez que se tienen los valores por separado del registro de datos Datalogging, se usa la herramienta systemIdentification. La misma se instala usando el Add-Ons y está dentro del paquete “System Identification Toolbox”.

El comando en Matlab 2018b es “systemIdentification”, como se muestra en Fig. 52, seguidamente se abre un cuadro de diálogo como se observa en Fig. 53

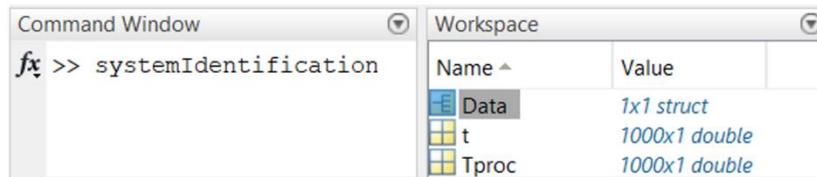


Fig. 52. Utilización de comando “systemIdentification”.

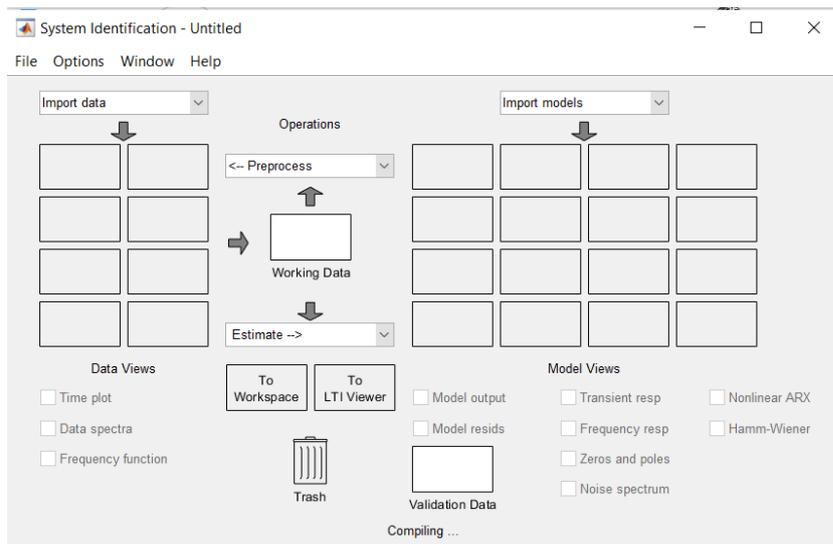


Fig. 53. Menú de la herramienta “systemIdentification”.

b) A continuación se importan los datos, en el dominio del tiempo como se indica:

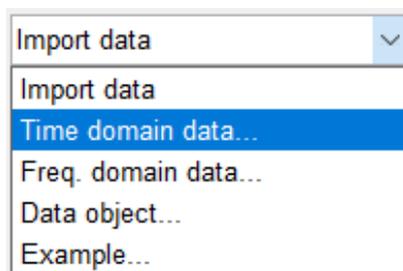


Fig. 54. Selección de la importación de datos.

Y se llenan los campos solicitados en el menú mostrado en Fig. 55.

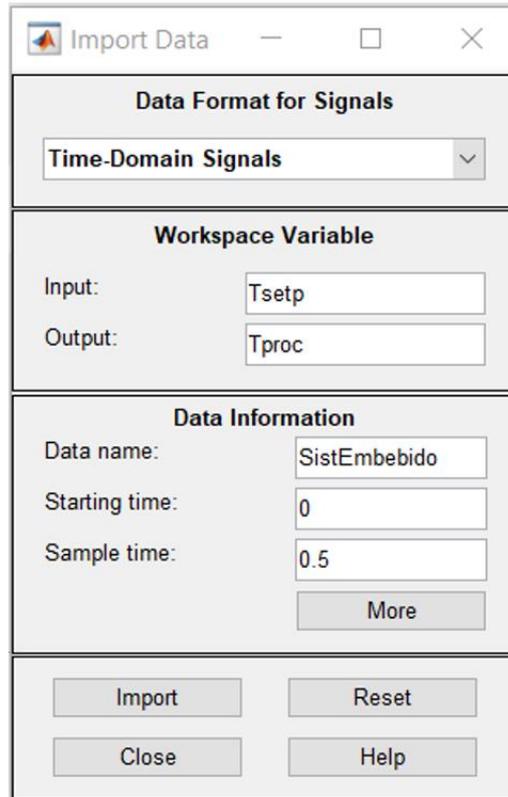


Fig. 55. Selección datos de setpoint y de proceso.

Como variable de entrada se coloca el setpoint deseado T_{setp} , que representa la entrada tipo escalón. La variable de salida del proceso es la respuesta de temperatura T_{proc} . También se colocan un nombre nuestro proceso, el tiempo de inicio y el tiempo de muestreo de los datos, mismo que se puede confirmar abriendo el vector de tiempo t y observando el valor del intervalo entre muestra y muestra.

- c) Se selecciona Import y cierra el cuadro de diálogo Import Data. Se despliega un primer reconocimiento de los datos en los recuadros de Operations, como se muestra en la Fig. 56.

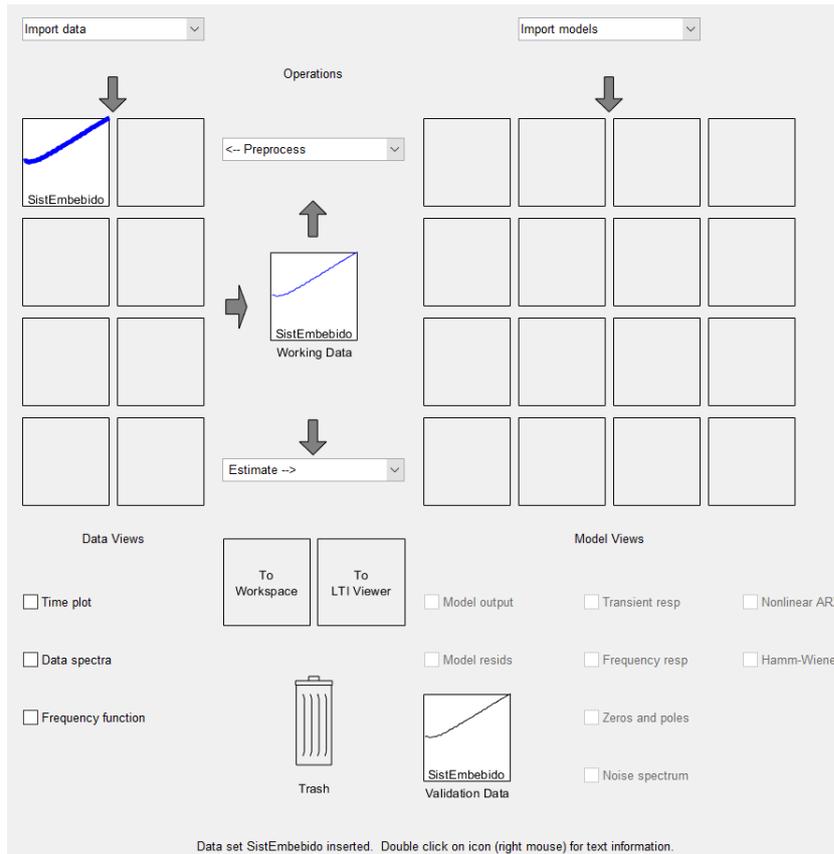


Fig. 56. Ambiente de selección de importación de datos.

- d) A continuación, se valida el modelo de la siguiente manera:
 En Operations, se selecciona el rango de mediciones para la entrada escalón

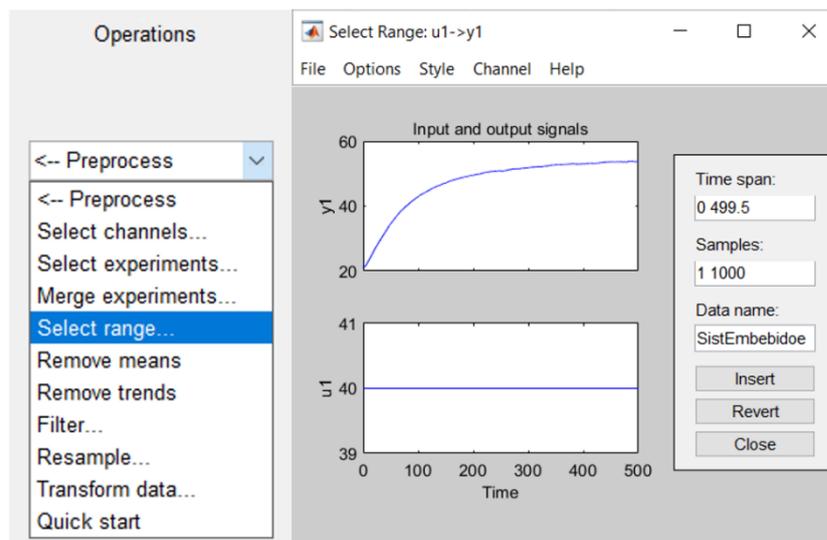


Fig. 57. Selección de rangos en ventana Operations.

Inmediatamente se marca el área de datos que deseamos analizar como se muestra:

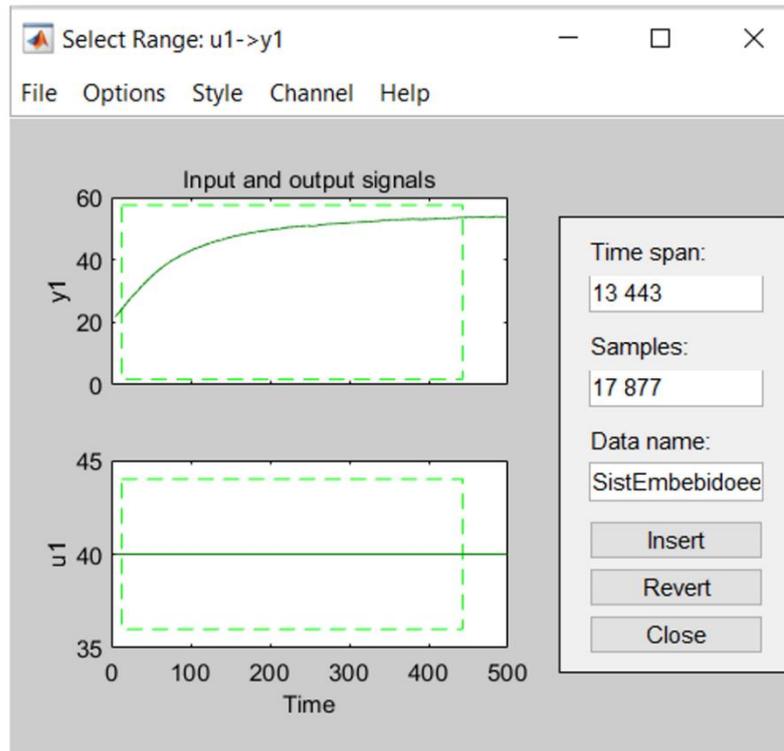


Fig. 58. Selección del área de datos.

Se escoge Insert por primera vez, y se genera el modelo cuyo nombre es el que se asignó a los datos seguido de una e, para el ejemplo SistEmbebidoee, luego se selecciona Insert por segunda vez y se genera un segundo modelo de validación, el nombre es el asignado a los datos seguidos de la letra v, ejemplo "Sist Embebido v".

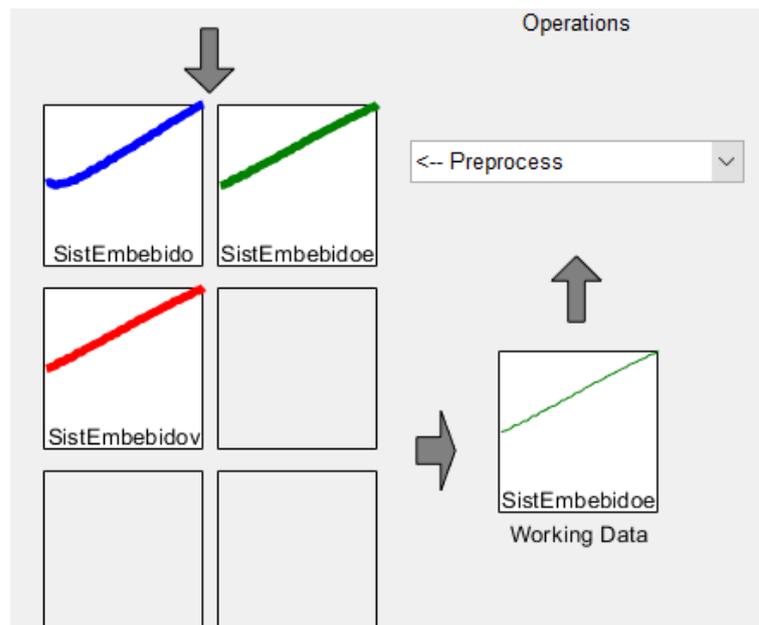


Fig. 59. Selección del área de datos (continuación 1).

Se arrastra el modelo generado al cuadro Working Data y el modelo de validación a cuadro Validation data, como se indica en Fig. 60.

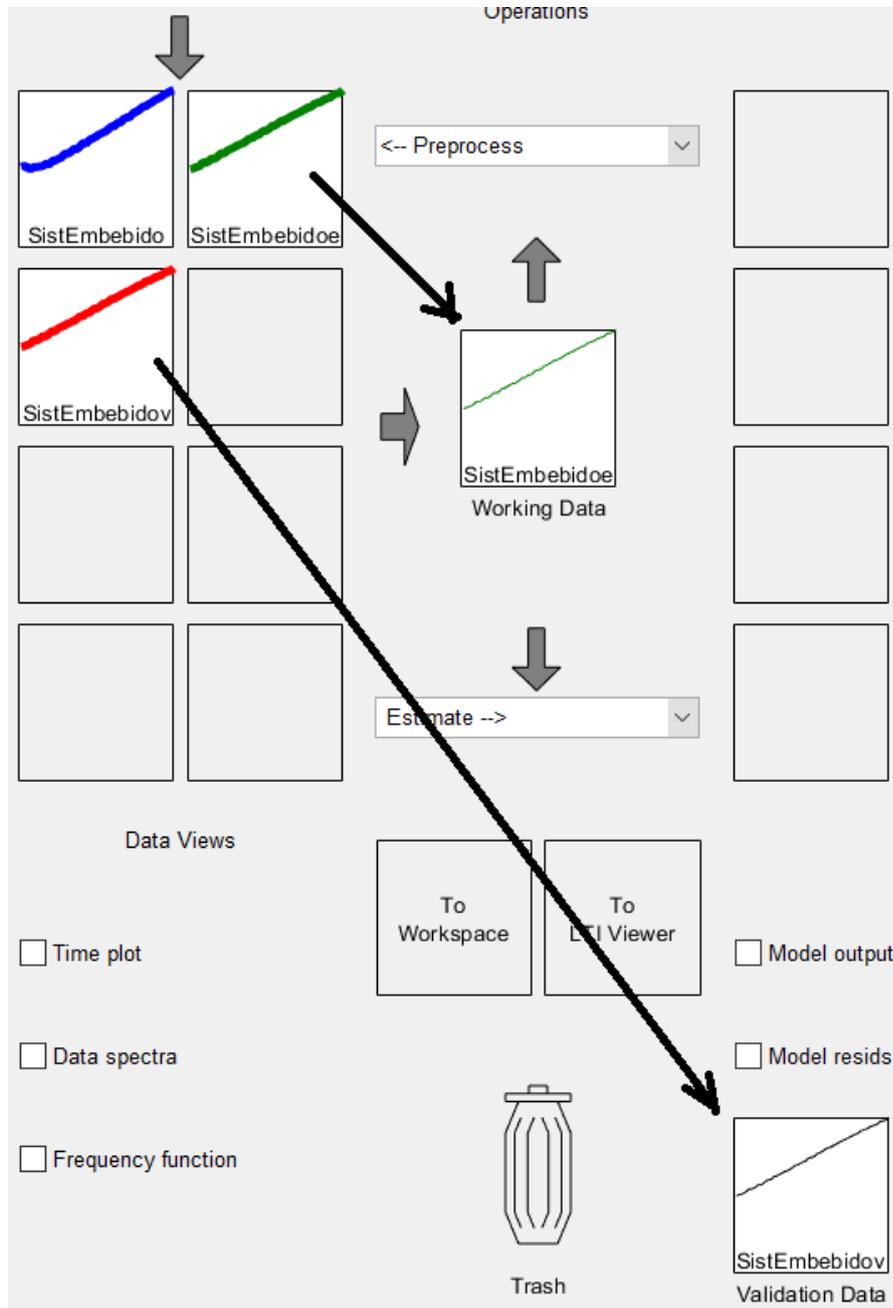


Fig. 60. Selección del área de datos (continuación 2).

Seguidamente se estima el modelo, para lo cual en el menú Estimate se navega hacia el comando Process Models y se lo selecciona.

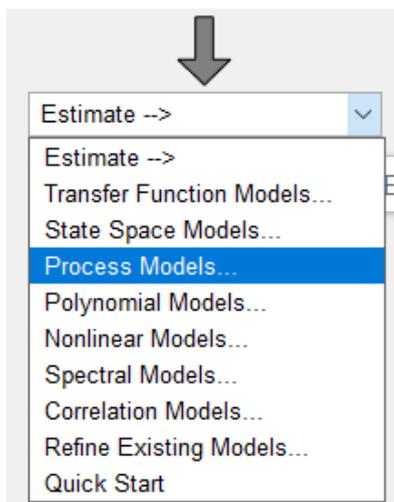


Fig. 61. Selección del área de datos (continuación 3).

Se observa en la Fig. 62 que se despliegan modelos de procesos, por defecto se presenta el modelo para función de transferencia; sin embargo, se puede comparar dicha señal con el tipo de respuesta obtenido, para validar el número de polos y ceros.

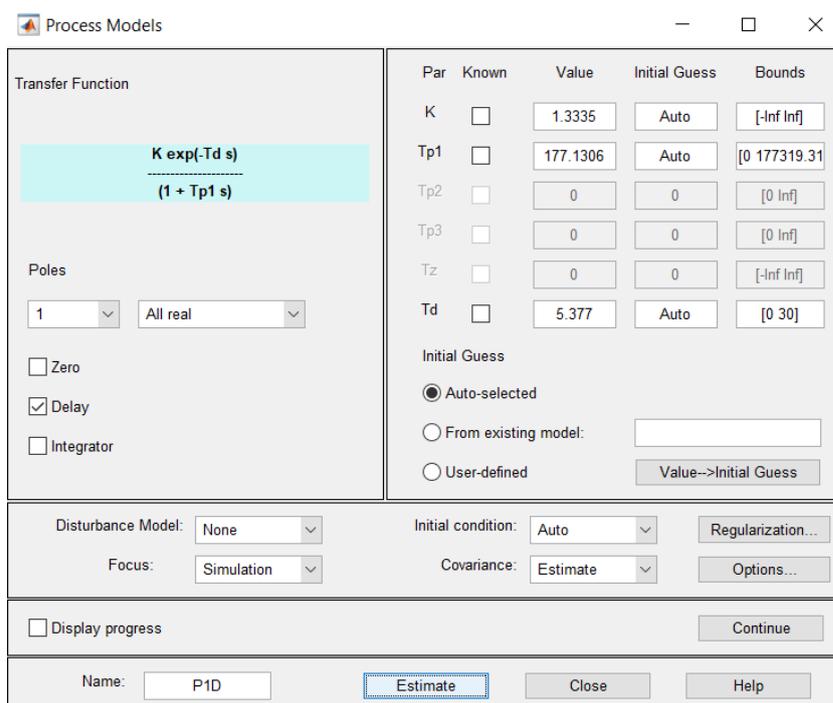


Fig. 62. Selección del área de datos (continuación 4).

Estando en el menú Process Models se selecciona Estimate y observa que se genera un modelo en el primer recuadro de Import models, ver Fig. 63.

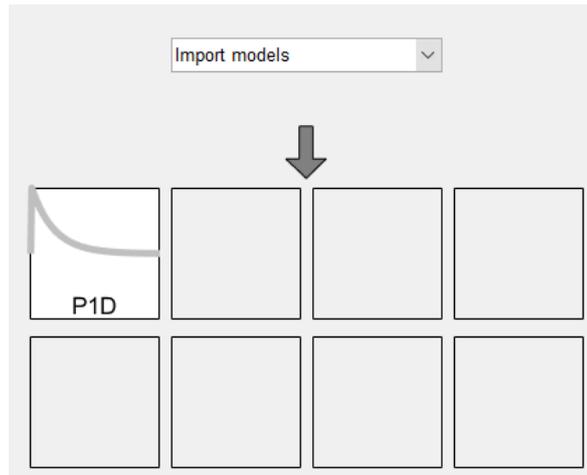


Fig. 63. Selección del área de datos (continuación 5).

Y dando click sobre el recuadro se tiene las propiedades y parámetros del modelo.

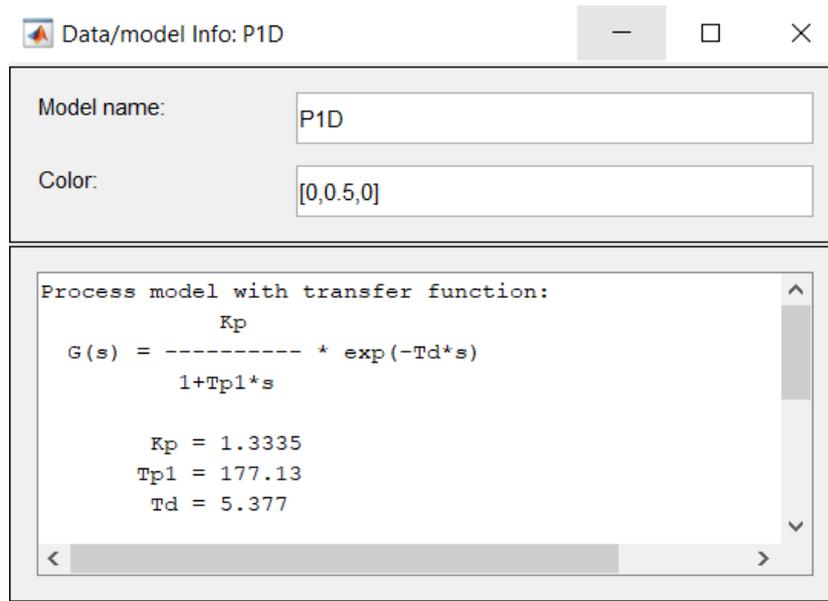


Fig. 64. Obtención del modelo matemático.

Finalmente, el modelo matemático obtenido de la planta en lazo abierto es:

$$G(s) = \frac{1.3335}{1 + 177.13 * s} * e^{-5.377*s}$$

Si se aplica un click en Model output, se valida el porcentaje de similitud del modelo con la respuesta real del sistema.

Model output

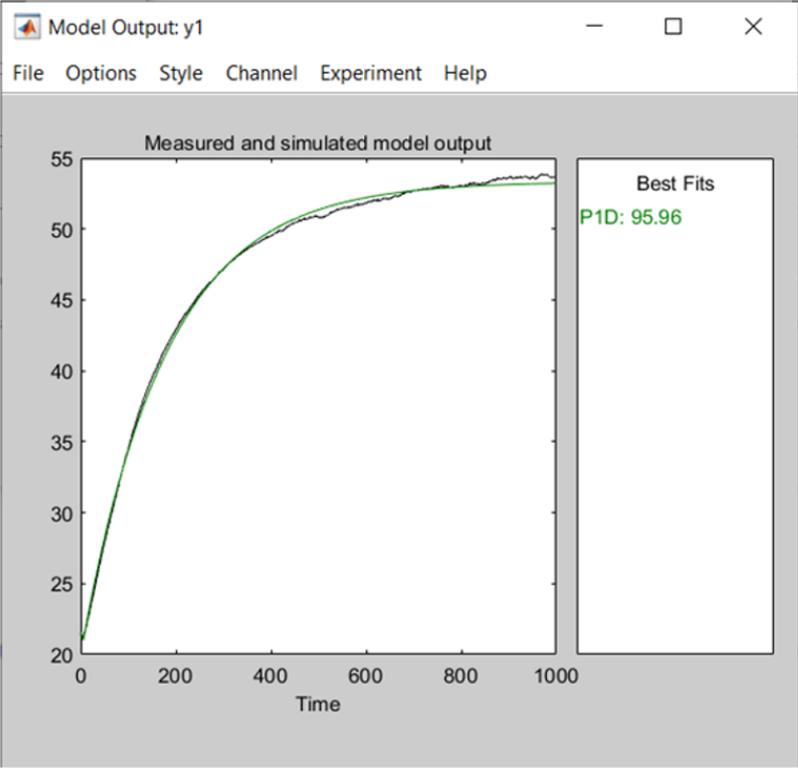


Fig. 65. Validación del modelo.

ANEXO 3. IMPLEMENTACIÓN DE LA PANTALLA DE CRISTAL LIQUIDO LCD CON I2C

A3.1 Implementación física

Se utiliza un display de cristal líquido LCD comunicándose con dos pines hacia Arduino, esto es posible mediante el protocolo I2C, el cual es un expander de entradas salidas. El módulo adaptador de LCD a protocolo I2C, se trata del circuito integrado PCF8574T, cuya dirección por defecto es 0x27.

La disposición de su estructura se muestra en la Fig. 63:



Fig. 66. Estructura del adaptador LCD a I2C, PCF8574T. Fuente:[35].

Para su utilización se descarga la librería LiquidCrystal_I2C.zip de la página de código libre Github [32]. En la aplicación de Arduino IDE se incluye el archivo .zip usando el comando de "Incluir Librería" y "Añadir biblioteca .ZIP" como se indica en la Fig. 64, a continuación, es necesario cerrar la aplicación Arduino IDE y volver a abrir, con ello se habilita la librería deseada.

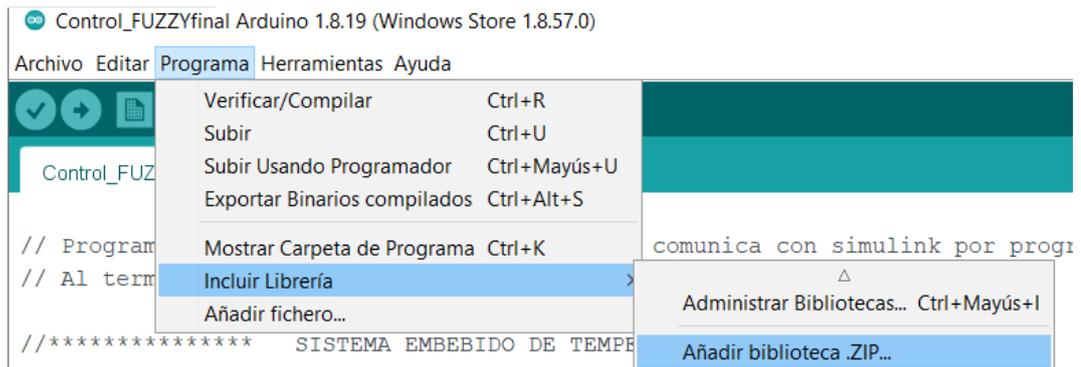


Fig. 67. Comando para incluir librerías en Arduino IDE.

Físicamente se conecta el cableado de acuerdo con Fig. 65:

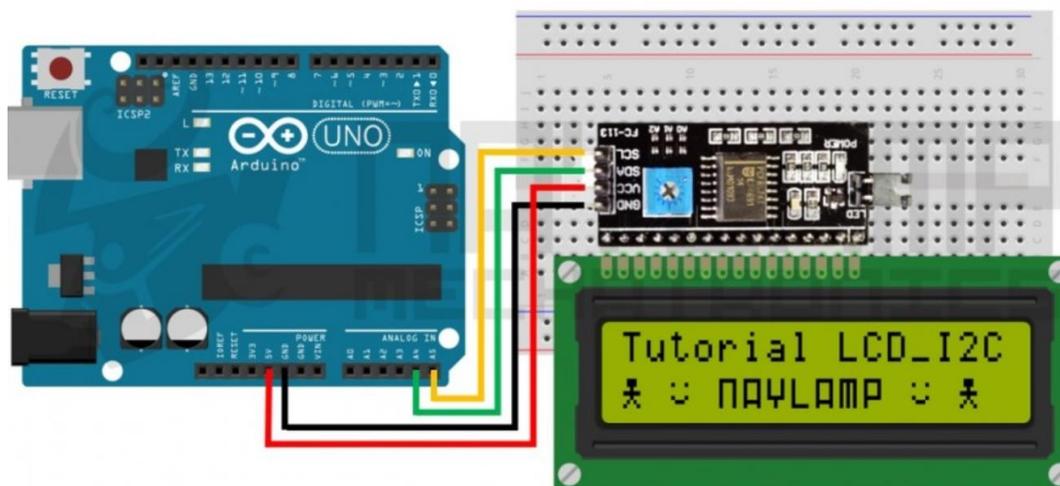


Fig. 68. Cableado display LCD con Arduino usando protocolo I2C. Fuente [35].

La tabla de conexiones es la siguiente:

Tabla XXII. Conexionado LCD a I2C en Arduino UNO.

Adaptador LCD - I2C	Arduino UNO
GND	GND
VCC	5V
SDA	A4
SCL	A5

Fuente [35].

A3.1 Implementación lógica

En la programación de Arduino, en su cabecera se incluye la librería `LiquidCrystal_I2C.h`, a continuación, en las líneas de configuración se coloca la dirección, el número de dígitos y las filas que posee el display. Finalmente, en el `void setup` se programa el lugar y los valores que se desea aparezcan en el LCD.

```
// CONFIGURACION DEL LCD
LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{

    //CONFIGURACION DEL LCD
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("SP");
    lcd.setCursor(4,0);
    lcd.print(":");
    lcd.setCursor(12,0);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("PV:");
    lcd.setCursor(7,1);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(10,1);
    lcd.print("CV:");
    lcd.setCursor(15,1);
    lcd.print("%");
}
```

ANEXO 4. CONSTRUCCIÓN FÍSICA

A4.1 Listado de material

En base a los diagramas de diseño obtenidos tanto para la planta Fig. 31 y para el control en Fig. 34, se elabora en la tabla XXI el listado de material y el precio comercial. El valor total es relativamente bajo comparado con tarjetas de controladores industriales clásicos. No incluye valor de computador personal y el software Matlab, ya que sólo se usan cuando se requiere análisis de datos.

Tabla XXIII. Listado de material y precio del sistema embebido.

Ítem	Tag	Descripción	Precio US \$
1	Rb	Potenciómetro de precisión 1k Ω , set 820 Ω	0.45
2	Re	Resistencia 10 Ω , 1/2 W	0.10
3	Q1	Transistor bipolar TIP31C	3.00
4	U1	Sensor de temperatura LM35	2.50
5	R1	Resistencia 10k Ω , 1/4 W	0.10
6	R2	Resistencia 30k Ω , 1/4 W	0.10
7	R3	Resistencia 470 Ω , 1/4 W	0.10
8	R4	Resistencia 470 Ω , 1/4 W	0.10
9	R5	Potenciómetro de precisión 1k Ω , set 600 Ω	0.45
10	R6	Potenciómetro 20k Ω , de perilla	0.90
11	R7	Potenciómetro precisión 20k Ω , set 11k Ω	0.45
12	R8	Resistencia 1k Ω , 1/4 W	0.10
13	R9	Resistencia 150 Ω , 1/4 W	0.10
14	R10	Resistencia 10k Ω , 1/4 W	0.10
15	U2	CI LM358N	0.40
16	ARD1	Controlador Arduino UNO	16.00
17	LCD	Display LCD 16 x2	4.00
18	U3	Adaptador LCD-I2C, PCF8574T	3.50
19	BAT1	Fuente 120 VAC - 12 VDC, 3A	20.00
20	SW1	Switch ON-OFF, luz incorporada	2.00
21	SW2	Switch ON-OFF	0.75
22	FUSE	Porta fusible	1.00
23	FU1	Fusible de vidrio 2A	0.10
24	L1	Luz indicador 12VDC	3.00
25	--	Conectores hembra tipo banana, panel	4.00
26	--	Conectores macho tipo banana	4.00
27	--	Cable 18 AWG, 3 metros	2.00
28	--	Diagrama PCB y quemado baquelita	22.00
29	--	Rotulación	12.00
30	--	2 Cajas de conexiones	32.00
Precio total			135.30

A4.2 Diagramas de cajas y tarjetas

En base al diagrama de Fig. 31 se elabora la tarjeta para colocar los elementos electrónicos asociados a la planta. La placa elaborada se muestra en Fig. 69.

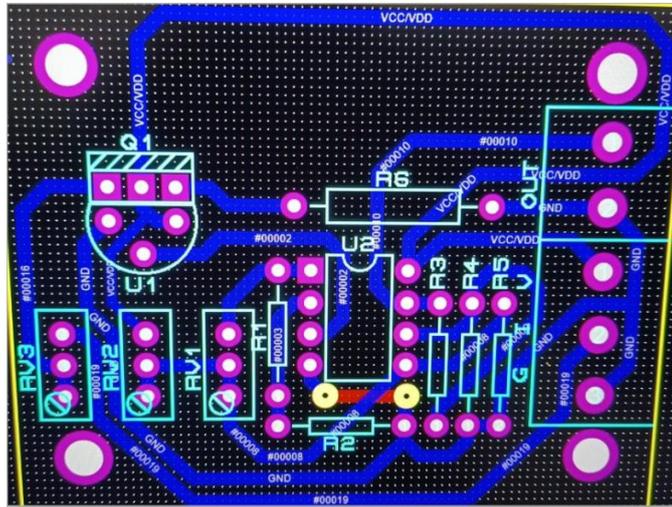


Fig. 69. Placa base del circuito de la planta.

Se elaboran las carátulas para las cajas de conexiones de la planta y el control como se indica a continuación, se tienen separadamente la planta y el control porque cada una se puede acoplar a cualquier sistema en caso de requerirse. En la Fig. 70 se presentan la fachada frontal de diseño.

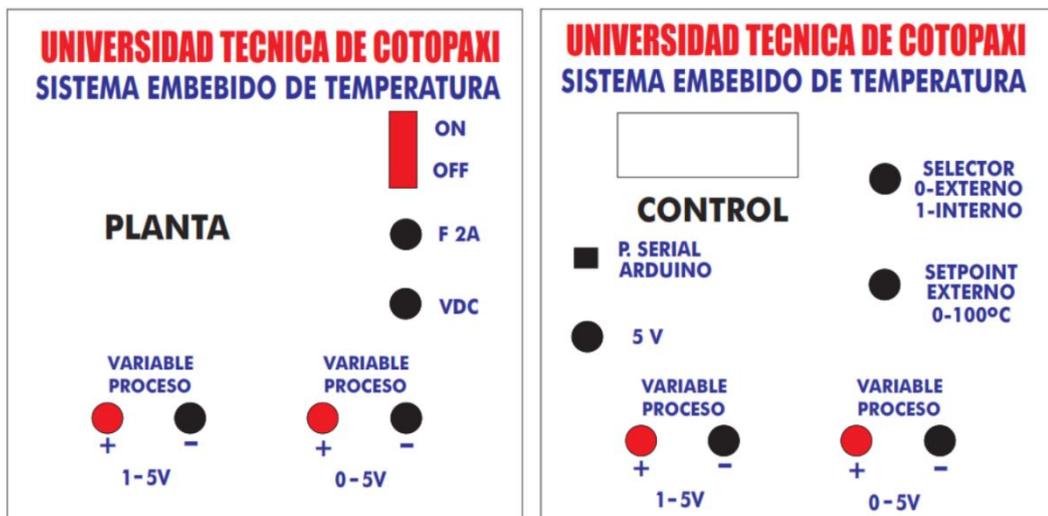


Fig. 70. Diseño de las carátulas de las cajas de conexiones.

La implementación se realiza en cajas comerciales de 20 cm x 20 cm x 10 cm, el resultado final se muestra en Fig. 71 y Fig. 72.

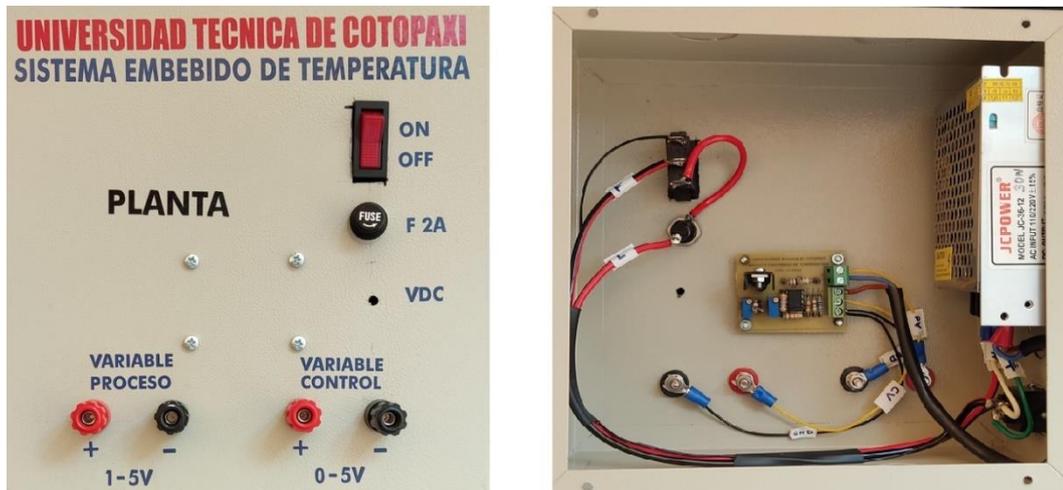


Fig. 71. Vista frontal y posterior de la caja para la planta



Fig. 72. Vista frontal y posterior de la caja para el control

Las dimensiones de las cajas se muestran en Fig. 73 son las siguientes:

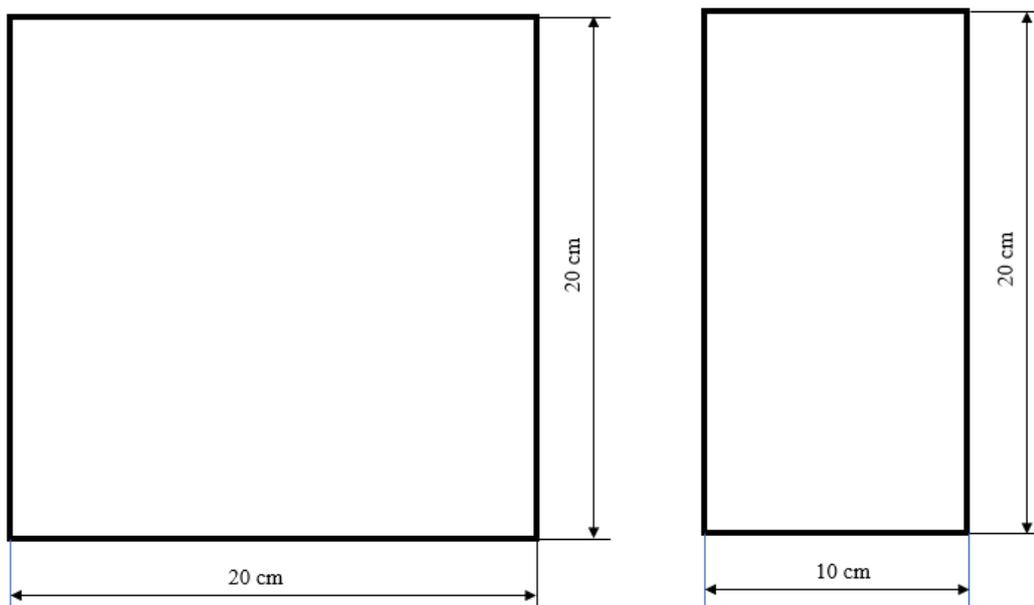


Fig. 73. Dimensiones de cajas de la planta y control.

ANEXO 5. DETERMINACIÓN DEL PUNTO DE TRABAJO DEL TIP31C

A5.1 Cálculo de la recta de carga y el punto Q del TIP31C

Los puntos de cálculo para la recta de carga son:

- La corriente de colector máxima $I_{c\ max}$ cuando el voltaje colector emisor V_{ce} sea nulo.

$$I_{c\ max} = \frac{V_{cc}}{R_e}$$

- El voltaje máximo presente en colector emisor V_{ce} , cuando la corriente del colector I_c es nula.

$$V_{ce} \approx V_{cc}$$

Para los valores de diseño $V_{cc}= 12V$, $R_e=10\ \Omega$, $R_b=820\ \Omega$ y con el voltaje V_{bb} variable se obtienen los datos tabulados en la tabla XXIV.

Tabla XXIV. Valores calculados de I_c vs. V_{CE} del transistor TIP31C.

V_{bb} [V]	V_{CE} [V]	I_C [mA]
	0	1200
5	9.0882	289.5
4	9.7654	222.2
3	10.443	154.8
2	11.12	87.52
1	11.79	20.19
0	12	0

La gráfica de la recta de carga y la región de operación se observa en la Fig. 74 y se observa que la zona de operación se encuentra junto a la región de corte del transistor, por lo que el voltaje colector emisor se halla cercano al valor de polarización V_{cc} , la potencia disipada y luego transformada en calor es aprovechada, sin embargo, cualquier valor de diseño que se cambie debe ser analizado cuidadosamente porque es muy susceptible de entrar en la región de corte.

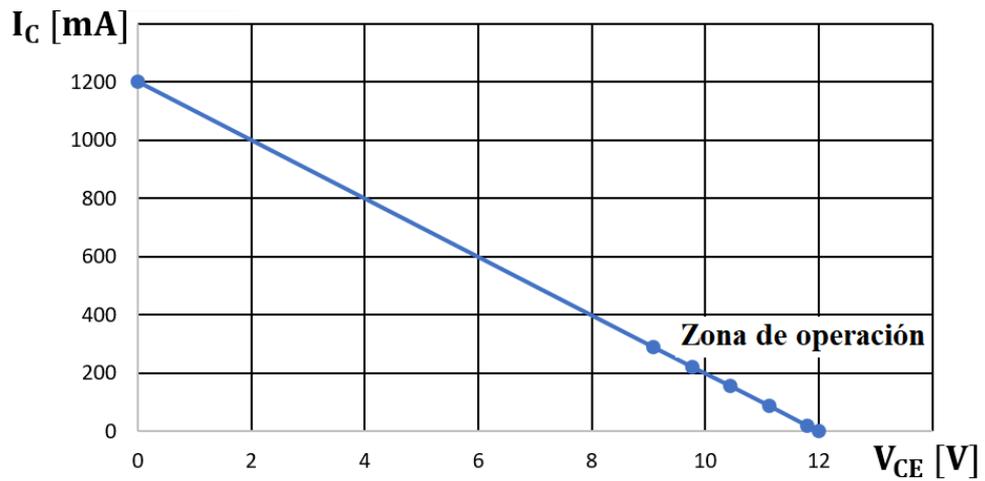


Fig. 74. Recta de carga y zona de operación del TIP31C

Finalmente, en Fig. 75 se coloca la gráfica de la zona de operación en la curva característica del “Área de Operación Segura” tomada del Datasheet del TIP31C.

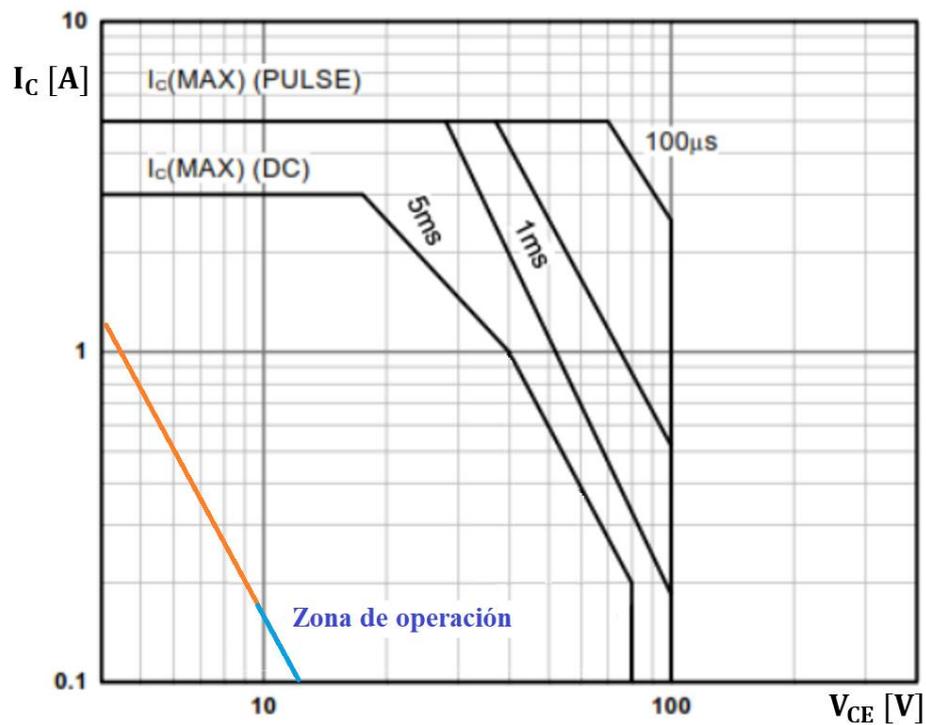


Fig. 75. Zona de operación de la planta del sistema embebido vs Área de Operación Segura TIP31C