



**Diseño e implementación de estrategias de control avanzado embebidas en un controlador lógico programable y aplicadas a un entorno industrial de escala real**

Herrera Yela, Michelle Carolina y Hurtado Freire, Stefano Gonzalo

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica, Automatización y Control

Ing. Gordillo Orquera, Rodolfo Xavier, Ph.D.

31 de julio del 2022



TESIS\_Herrera\_Hurtado\_FINAL.pdf

Scanned on: 15:12 July 31, 2022 UTC



Escaneado electrónicamente por:  
RODOLFO XAVIER  
GORDILLO ORQUERA



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	281
Words with Minor Changes	46
Paraphrased Words	92
Omitted Words	0



**Departamento de Eléctrica, Electrónica y Telecomunicaciones**

**Carrera de Ingeniería en Electrónica, Automatización y Control**

### **Certificación**

Certifico que el trabajo de titulación: **“Diseño e implementación de estrategias de control avanzado embebidas en un controlador lógico programable y aplicadas a un entorno industrial de escala real”** fue realizado por los señores **Herrera Yela, Michelle Carolina y Hurtado Freire, Stefano Gonzalo**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

**Sangolquí, 31 de julio 2022**

Firma:



**Ing. Gordillo Orquera, Rodolfo Xavier, Ph.D.**

C. C. : 1001510203



**Departamento de Eléctrica, Electrónica y Telecomunicaciones**  
**Carrera de Ingeniería en Electrónica, Automatización y Control**

**Responsabilidad de Autoría**

Nosotros, **Herrera Yela, Michelle Carolina y Hurtado Freire, Stefano Gonzalo**, con cédulas de ciudadanía n° 1718319641 y n° 1725616849, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Diseño e implementación de estrategias de control avanzado embebidas en un controlador lógico programable y aplicadas a un entorno industrial de escala real** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Sangolquí, 26 de julio 2022**

Firma

...Michelle.H.Y. ....

**Herrera Yela, Michelle Carolina**

C.C.: 1718319641

Firma

...Stefano.G. ....

**Hurtado Freire, Stefano Gonzalo**

C.C.: 1725616849



**Departamento de Eléctrica, Electrónica y Telecomunicaciones**

**Carrera de Ingeniería en Electrónica, Automatización y Control**

**Autorización de Publicación**

Nosotros **Herrera Yela, Michelle Carolina** y **Hurtado Freire, Stefano Gonzalo**, con cédulas de ciudadanía n° 1718319641 y n° 1725616849, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño e implementación de estrategias de control avanzado embebidas en un controlador lógico programable y aplicadas a un entorno industrial de escala real** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

**Sangolquí, 26 de julio 2022**

Firma

.....*Michelle H.Y.*.....

**Herrera Yela, Michelle Carolina**

C.C.: 1718319641

Firma

.....*Stefano*.....

**Hurtado Freire, Stefano Gonzalo**

C.C.: 1725616849

## Dedicatoria

Dedicado a mis padres Manolo y Ondina, a mi hermano Andrés, quienes están siempre para apoyarme incondicionalmente sin ellos nada de esto sería posible.

A mis abuelos Manuel e Inés (QEPD) quienes en estos momentos estarían muy orgullosos de verme cumplir mis objetivos.

A cada integrante de mi familia Herrera – Yela, que siempre han estado pendientes de mí durante este proceso y ahora finalmente lo he concluido.

A cada niña, adolescente y mujer que consideren entrar al mundo de la ingeniería o STEM del país, que sepan que es un mundo lleno de sorpresas y emociones, la ciencia e ingeniería son un lugar interesante en el cual refugiarse y todos somos capaces de desarrollarnos profesionalmente en estos campos.

“El amor es lo único que trasciende las dimensiones del tiempo y del espacio.”

*Interstellar.*

Michelle Carolina Herrera Yela

## **Agradecimiento**

Mi agradecimiento hacia a todas las personas que a lo largo de este camino me supieron brindar su apoyo en cualquier momento. A mi padre y a mi madre quienes siempre saben brindarme las mejores palabras de aliento para seguir adelante cuando estoy a punto de rendirme. A mi familia que siempre estuvieron pendientes de cómo iba en la universidad y cualquier duda o problema que surgiera en el camino supieron extenderme su mano de ayuda.

A mis docentes universitarios quienes desde un inicio supieron compartirnos sus conocimientos y guiarnos en este valioso camino de la ingeniería, como alguna vez un docente nos supo decir “La ingeniería se compone de vitamina D: Dedicación, disciplina y determinación”

A mis amigas de colegio quienes también supieron brindarme su apoyo emocional en los momentos difíciles que han transcurrido a lo largo de estos años y que su amistad ha sido un pilar importante en mi crecimiento como persona, gracias por confiar en mis capacidades intelectuales y brindarme siempre una palabra de aliento para hacerme saber que soy capaz de cumplir con todas mis metas.

A mis compañeros de universidad quienes, entre risas, diversión y mucho estrés también siempre estuvieron para apoyarme en lo que se pueda para salir todos adelante en el camino que se nos presentaba durante cada semestre. A la Rama Estudiantil IEEE ESPE en donde encontré personas y momentos inolvidables que me permitieron crecer como persona y profesional. Finalmente, gracias a la Ing. Andrea López e Ing. Cristina Aguirre, que han sabido motivarme durante este camino y demostrarme que las mujeres también somos muy capaces de llegar lejos y demostrar liderazgo en el campo STEM. Esperando ser inspiración y motivación para más mujeres en ingeniería.

## Dedicatoria

Dedicado a mi padre Gonzalo quien ha sido un apoyo incondicional durante esta etapa de estudios y una motivación en mi desarrollo personal, quien pese a que llegó a ser padre cuando era solo un adolescente supo cómo dar lo mejor de él para que juntos salgamos adelante, cada logro mío es un logro para ambos. A mis tíos Carly y Hugo quienes han estado siempre dispuestos a brindarme su apoyo, consejos y preocupación por mi futuro. A mi abuela Magdalena quien me ha brindado su amor y comprensión en todo momento.

A mi familia y amigos que siempre creyeron en mí y estuvieron a mi lado en todo este proceso.

Stefano Gonzalo Hurtado Freire



## Agradecimiento

Agradezco la confianza sin medida de mi padre en lo que soy capaz de lograr, por su apoyo sin condiciones para que yo pueda lograr mis metas y alcanzar mis sueños, por cada consejo y cada palabra de aliento que recibí desde que era un niño.

A mi abuela Magoly quien siempre se preocupó por mi bienestar, quien sin condiciones me dio su afecto y me acogió como una madre.

A mi tía Carly por su preocupación y su apoyo en todo mi desarrollo universitario, por sus consejos, regaños y palabras de ánimo que han sido un empujón extra en el proceso de aprendizaje y desarrollo personal.

Quiero agradecer a mi mejor amigo David, cuya amistad ha sido un constante apoyo en mi crecimiento personal, por la confianza y seguridad que me ha dado para creer en mis capacidades.

A mis compañeros, que como un equipo hemos logrado escalar cada una de las dificultades durante la carrera para poder juntos llegar a este momento.

A mis profesores más cercanos quienes supieron como transmitir todo su conocimiento sobre las materias y también brindarnos su experiencia en la Ingeniería para darnos una idea sobre la vida profesional.

A mi tutor, quien desde las materias previas dictadas ha sido un ejemplo en el ámbito académico, sorprendiéndonos con la simplicidad que tienen los diferentes temas de la ingeniería cuando se tiene vocación y gusto por lo que se hace.

Al Ing. Alexis Bracero y la empresa INGUIBRA, quien durante el desarrollo de este proyecto y el inicio de mi vida profesional me han extendido su mano y su confianza, exigiéndome únicamente el compromiso y trabajo duro.

## Índice de Contenidos

Análisis de similitud de contenidos.....	2
Certificación del tutor .....	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Agradecimiento.....	7
Índice de Contenidos .....	10
Índice de Tablas .....	13
Índice de Figuras .....	14
Resumen .....	17
Abstract.....	18
Capítulo I. Introducción.....	19
Antecedentes .....	19
Estado del arte .....	22
Justificación e Importancia .....	24
Alcance del proyecto.....	27
Objetivos .....	28
<i>Objetivo general</i> .....	28
<i>Objetivos específicos</i> .....	28

	11
Metodología de la investigación .....	28
Capítulo II. Marco conceptual.....	31
Identificación de sistemas.....	31
<i>Métodos para la identificación de sistemas.....</i>	<i>32</i>
Estrategias de control avanzado .....	38
<i>Controlador PID.....</i>	<i>39</i>
<i>Controlador en espacio de estados .....</i>	<i>47</i>
<i>Controlador autoajustable difuso .....</i>	<i>53</i>
Generación de código para controladores mediante la herramienta Matlab/Simulink .....	58
PLC de la marca TURCK .....	60
<i>Plataforma de software CODESYS.....</i>	<i>61</i>
Capítulo III. Diseño de estrategias de control avanzado.....	62
Identificación de la planta de temperatura. ....	62
<i>Método gráfico de estimación por respuesta al escalón. ....</i>	<i>65</i>
<i>Función ARX. IDENT (System Identification toolbox)-Matlab .....</i>	<i>68</i>
<i>Función PROCEST P1D. ....</i>	<i>68</i>
Diseño del Controlador PI 2DOF .....	69
<i>PID en forma digital directa. ....</i>	<i>70</i>
<i>Sintonización de PID por Ziegler - Nichols.....</i>	<i>72</i>
<i>Sintonización de controladores mediante la herramienta PID Tuner Free Web Site .....</i>	<i>73</i>

	12
Diseño del Controlador en espacio de estados con acción integral SSIA .....	75
Diseño del Controlador autoajutable difuso (AFuzzy).....	77
Capítulo IV. Generación de código para la plataforma de software CODESYS.....	84
<i>PI 2DOF</i> .....	84
<i>SSIA</i> .....	85
<i>AFuzzy</i> .....	87
Capítulo V. Simulaciones y escenarios de prueba .....	88
Análisis de desempeño de las estrategias de control avanzado mediante simulación .....	88
<i>PI 2DOF</i> .....	88
<i>SSIA</i> .....	91
<i>AFuzzy</i> .....	93
Análisis de desempeño de las estrategias de control avanzado en tiempo real .....	99
<i>PI 2DOF</i> .....	99
<i>SSIA</i> .....	102
<i>AFuzzy</i> .....	104
Capítulo VI. Conclusiones y recomendaciones .....	110
Conclusiones .....	110
Recomendaciones.....	111
Bibliografía .....	111
Apéndices.....	115

## Índice de Tablas

<b>Tabla 1.</b> Marcas de PLC's que implementan controladores a nivel industrial.....	22
<b>Tabla 2.</b> Reglas de sintonización por Ziegler-Nichols para un sistema de primer orden con tiempo muerto .....	44
<b>Tabla 3.</b> Reglas del controlador difuso para sintonización de $k_p$ .....	57
<b>Tabla 4.</b> Reglas del controlador difuso para sintonización de $k_i$ .....	57
<b>Tabla 5.</b> Definición de variables para generar entrada escalón al sistema .....	62
<b>Tabla 6.</b> Definición de constantes para generar entrada escalón al sistema .....	63
<b>Tabla 7.</b> Comparación de resultados en simulación de los controladores PI2DOF, SSIA Y AFuzzy .....	97
<b>Tabla 8.</b> Comparación de resultados de los controladores PI2DOF, SSIA Y AFuzzy implementados. ....	109

## Índice de Figuras

<b>Figura 1.</b> Metodología aplicada al proyecto de investigación .....	29
<b>Figura 2.</b> Clasificación de la obtención de modelos.....	32
<b>Figura 3.</b> Métodos para la identificación de sistemas .....	33
<b>Figura 4.</b> Pasos generales a seguir para la identificación de sistemas .....	34
<b>Figura 5.</b> Parámetros de la respuesta al escalón de una planta de temperatura con retardo.....	36
<b>Figura 6.</b> Diagrama de bloques de un controlador PID ideal.....	40
<b>Figura 7.</b> Estructura de un controlador PID en forma digital directa .....	42
<b>Figura 8.</b> Controlador PID digital Tipo A .....	45
<b>Figura 9.</b> Controlador PID 2DOF digital.....	46
<b>Figura 10.</b> Diagrama de bloques modelamiento de tiempo discreto en espacio de estados .....	51
<b>Figura 11.</b> Diagrama de bloques de un sistema de control con realimentación de estado .....	52
<b>Figura 12.</b> Procesos del controlador Fuzzy Logic .....	55
<b>Figura 13.</b> Estructura de un controlador difuso autoajutable .....	56
<b>Figura 14.</b> Diagrama de bloques para un diseño basado en modelos.....	59
<b>Figura 15.</b> Relación del análisis matemático y los procedimientos de diseño para sistemas físicos reales	60
<b>Figura 16.</b> PLC BL20 (IP20) marca Turck .....	61
<b>Figura 17.</b> Diagrama de flujo para generar entrada escalón al sistema .....	64
<b>Figura 18.</b> Sistema térmico implementado físicamente.....	65
<b>Figura 19.</b> Método gráfico de estimación por respuesta al escalón .....	66
<b>Figura 20.</b> Respuesta al escalón del sistema, datos originales de la planta y método de identificación gráfico .....	67
<b>Figura 21.</b> Respuesta al escalón del sistema función Procest P1D y función ARX.....	69

<b>Figura 22.</b> Estructura de un controlador PID en forma digital directa aplicado a un sistema térmico con retraso.....	70
<b>Figura 23.</b> Respuesta del sistema ante el controlador PID en forma digital directa .....	72
<b>Figura 24 .</b> Respuesta del sistema ante el controlador PI diseñado por Ziegler-Nichols.....	73
<b>Figura 25.</b> Respuesta del sistema controlador PI y PI 2DOF.....	75
<b>Figura 26.</b> Respuesta del Sistema controlador SSIA .....	77
<b>Figura 27.</b> Funciones de membresía respecto al error .....	78
<b>Figura 28.</b> Funciones de membresía respecto a la derivada del error .....	78
<b>Figura 29.</b> Funciones de membresía respecto al valor de $k_p$ .....	79
<b>Figura 30.</b> Funciones de membresía respecto al valor de $k_i$ .....	80
<b>Figura 31.</b> Reglas del FLC para estimar valores de $k_p$ y $k_i$ .....	80
<b>Figura 32.</b> Diagrama de bloques del controlador FLC.....	81
<b>Figura 33.</b> Superficie de control para el valor sintonizado de $k_p$ .....	82
<b>Figura 34.</b> Superficie de control para el valor sintonizado de $k_i$ .....	82
<b>Figura 35.</b> Respuesta del Sistema controlador AFuzzy .....	83
<b>Figura 36.</b> Subsistema para la generación de código PID 2DOF .....	84
<b>Figura 37.</b> Subsistema para la generación de código SSIA .....	86
<b>Figura 38.</b> Subsistema para la generación de código AFuzzy. ....	87
<b>Figura 39.</b> Diseño controlador PI 2DOF para simulación.....	89
<b>Figura 40.</b> Respuesta del Sistema térmico con un controlador PI 2DOF en simulación.....	90
<b>Figura 41.</b> Simulación del controlador PI2DOF con rechazo a perturbaciones y seguimiento a referencia. .....	91
<b>Figura 42.</b> Diseño controlador SSIA para simulación.....	91
<b>Figura 43.</b> Respuesta del Sistema térmico con un controlador SSIA en simulación.....	92

<b>Figura 44.</b> Simulación del controlador SSIA con rechazo a perturbaciones y seguimiento a referencia. ..	93
<b>Figura 45.</b> Diseño controlador AFuzzy para simulación.....	93
<b>Figura 46.</b> Respuesta del Sistema térmico con un controlador AFuzzy en simulación .....	94
<b>Figura 47.</b> Valores de $k_p$ en simulación del controlador AFuzzy .....	95
<b>Figura 48.</b> Valores de $k_i$ en simulación del controlador AFuzzy.....	95
<b>Figura 49.</b> Simulación del controlador AFuzzy con rechazo a perturbaciones y seguimiento a referencia .....	96
<b>Figura 50.</b> Bloque de comparación de controladores PI2DOF, SSIA y AFuzzy en simulación.....	97
<b>Figura 51.</b> Respuesta del Sistema ante los controladores PI2DOF, SSIA y AFuzzy en simulación .....	98
<b>Figura 52.</b> Respuesta del Sistema térmico con un controlador PI 2DOF en tiempo real .....	100
<b>Figura 53.</b> Respuesta del Sistema térmico ante una perturbación con el controlador PI 2DOF en tiempo real .....	100
<b>Figura 54.</b> Respuesta del Sistema térmico con un controlador PI 1DOF en tiempo real .....	101
<b>Figura 55.</b> Respuesta del Sistema térmico ante una perturbación con el controlador PI 1DOF en tiempo real .....	102
<b>Figura 56.</b> Respuesta del Sistema térmico con un controlador SSIA en tiempo real .....	103
<b>Figura 57.</b> Respuesta del Sistema térmico ante una perturbación con el controlador SSIA en tiempo real .....	104
<b>Figura 58.</b> Respuesta del Sistema térmico con un controlador AFuzzy en tiempo real .....	105
<b>Figura 59.</b> Respuesta del Sistema térmico ante una perturbación con el controlador AFuzzy en tiempo real .....	106
<b>Figura 60.</b> Valores de $k_p$ y $k_i$ en tiempo real del controlador AFuzzy .....	107
<b>Figura 61.</b> Valores de $k_p$ y $k_i$ en tiempo real del controlador AFuzzy ante una perturbación .....	108
<b>Figura 62.</b> Respuesta del Sistema ante los controladores PI2DOF, SSIA y AFuzzy implementados.....	109



## Resumen

A lo largo de los últimos años, la realización de estrategias de control avanzado a nivel industrial ha representado un proceso complejo y las herramientas de generación o conversión de código no han sido ampliamente difundidas. Así mismo, la literatura al respecto resulta ser relativamente escasa y usualmente se pueden requerir de recursos adicionales para trasladar el código generado directamente en los controladores lógicos programables. Este proyecto pretende ser un aporte metodológico en la realización de estrategias de control avanzado a través de la generación de código y su implementación en dispositivos industriales. La metodología propuesta para este proyecto fue, inicialmente la construcción de una planta de temperatura física a escala real y la identificación del modelo de la misma, a partir del modelo identificado se diseñaron dos de las tres estrategias de control propuestas: un controlador dos grados de libertad y un controlador en espacio de estados con capacidad de rechazo a las perturbaciones; la tercera estrategia de control corresponde a un controlador autoajustable difuso con un diseño libre de modelo. Se realizó la simulación de los controladores diseñados y la generación de código a través de Matlab/Simulink para trasladarlo a un PLC de la marca Turck mediante la plataforma de software CODESYS. Finalmente se efectuó la validación y el análisis de desempeño de las estrategias de control avanzado tanto en simulación como en tiempo real. Los resultados obtenidos muestran que es factible la realización de las estrategias de control propuestas a través de la generación de código para la implementación directa de las mismas en un entorno industrial de escala real.

*Palabras clave:* estrategias de control avanzado, generación de código, implementación en tiempo real, código embebido, PLC.

### **Abstract**

Over the last few years, the realization of advanced control strategies at the industrial level has been a complex process and code generation or conversion tools have not been widely available. Also, the literature on the subject is relatively scarce and usually additional resources may be required to transfer the generated code directly into programmable logic controllers. This project aims to be a methodological contribution in the realization of advanced control strategies through the generation of code and its implementation in industrial devices. The methodology proposed for this project was, initially the construction of a full-scale physical temperature plant and the identification of its model, from the identified model two of the three proposed control strategies were designed: a two degrees of freedom controller and a state space controller with disturbance rejection capability; the third control strategy corresponds to a fuzzy self-adjusting controller with a model-free design. The simulation of the designed controllers and the generation of code through Matlab/Simulink was carried out in order to transfer it to a Turck PLC through the CODESYS software platform. Finally, the validation and performance analysis of the advanced control strategies was carried out both in simulation and in real time. The results obtained show that it is feasible to implement the proposed control strategies through the generation of code for their direct implementation in a full-scale industrial environment.

*Keywords:* advanced control strategies, code generation, real-time implementation, embedded code, PLC.

## Capítulo I. Introducción

En este capítulo se desarrolla una visión general sobre el trabajo de investigación. En primera instancia se muestran los antecedentes, seguido de un breve estudio sobre estado del arte y la justificación, se expone el alcance y los objetivos que engloban la principal motivación para realización del mismo. Además, se presenta la estructura y organización, así como la metodología aplicada a la realización del proyecto.

### Antecedentes

La necesidad de controlar variables que cambian constantemente y son susceptibles a perturbaciones dentro de los procesos industriales ha logrado generar una línea concreta de investigación en el área de la automatización industrial, por lo tanto, se busca que los sistemas que controlan dichas variables sean robustos, eficientes, adaptables y posean un alto desempeño. Es decir que la necesidad de implementar este tipo de sistemas nace de la mano con los paradigmas que necesitaba resolver la industria para sus sistemas de producción como lo plantea en el prefacio del libro de Marlin, donde se afirma que hasta aproximadamente 1960 el control de procesos se realizaba en base a sistemas físicos basados en circuitos analógicos, los cuales realizaban las tareas de filtrado, amplificación y decisión, estos sistemas eran poco flexibles y sus cálculos no podían ser demasiado complejos (Marlin, 2015).

Tras la rigurosa investigación desarrollada a lo largo de la historia en la rama de control de procesos, se han logrado desarrollar varios métodos, técnicas y leyes de control que permiten un desempeño aceptable de los controladores automáticos teóricamente, sin embargo, a nivel de aplicación la mayoría de los equipos Controladores Lógicos Programables (PLC's) comerciales, donde se embeben los controladores automáticos poseen únicamente la opción del controlador de tipo PID, en el cual se permite realizar pocas variaciones en su estructura y desempeño en base a los parámetros disponibles que poseen de configuración. También vale la pena mencionar como se recalca en (Aström

& Hägglund, Control PID avanzado, 2009) que a lo largo de los años, para el control de procesos, siempre ha sido común implementar controladores clásicos tales como el PID ya que estos poseen la cualidad de ser simples y efectivos, para ciertos casos, pero cuando se trata de procesos multi variables o no lineales, los PID pierden su efectividad y es ahí cuando se debe apuntar hacia técnicas de control avanzado que envuelven algoritmos con inteligencia computacional. Debido a estas cualidades que se sugiere que tengan los controladores automáticos, varios PLC hoy en día, cuentan con la capacidad de cómputo y procesamiento necesarios, para poder embeber en ellos técnicas de control inteligente. Un punto más a considerar sobre los controladores clásicos, específicamente sobre el controlador PID, como se menciona en (Yang & Bian, 2012) acerca de implementar el controlador en plantas de temperatura, estas plantas presentan características tales como retardos e inercia, lo que hace que se vuelva una tarea compleja el modelamiento matemático de la planta de forma precisa. Siendo así que un controlador convencional como el PID no logre una precisión en el control del sistema, por tal razón, en varias investigaciones, se aplican controladores inteligentes como el controlador difuso basados en el funcionamiento de un controlador clásico PID, para mejorar el rendimiento dinámico de la planta.

Aproximadamente el 95% de controladores implementados en la industria son de tipo PID, sin embargo, más de la mitad de ese porcentaje no tiene un buen desempeño y al proceso que controlan no le corresponde ese tipo de controlador (Aström & Wittenmark, Computer Controlled Systems, 1997). Además, Eriksson asegura que entre el 5% y 10% de los procesos que requieren un control automático no pueden ser controlados de ninguna manera con un controlador PID (Eriksson & Johansson, 2007). Es decir que un control clásico no siempre es suficiente ya que no se llegan a cubrir las necesidades, características y niveles de control que requiere un proceso industrial. También como se indica en investigaciones sobre diseño, implementación y comparación de controladores para aplicaciones industriales en (Shakya, Rajanwal, Patel, & Dinkar, 2014) generalmente en las industrias, sobre todo para el control de procesos tales como el control de flujo, se utilizan controladores automáticos PID o

PD, sin embargo, se ha demostrado con el pasar del tiempo, que muchas de las veces, este tipo de controladores no son suficientes para obtener un control rápido y presentan oscilaciones en sus respuestas, dañando el sistema. Por lo que, para mejorar la respuesta del sistema ante perturbaciones sin causar daños, también se diseña un controlador difuso y se compara su desempeño con los controladores clásicos, proponiendo aplicar estas técnicas de control automático inteligente en distintas aplicaciones a nivel industrial.

Por lo tanto, para resumir de mejor manera los antecedentes planteados, se presenta a continuación en la tabla 1, las marcas más conocidas de controladores lógicos programables que implementan a nivel industrial estrategias de control. De manera que se puede observar, en su mayoría el tipo de controladores que son implementados en la industria son los PID tipo A, en el mejor de los casos son de tipo B (PID Controller Tuning, 2006). Por lo que en el presente trabajo de titulación se propone principalmente desarrollar estrategias de control avanzado que brinden un mejor desempeño a nivel industrial, mencionando también que el beneficio del tipo de PID no solo se encuentra en la reducción del sobre impulso sino a nivel de una menor demanda de energía al permitir trabajar al actuador con menores niveles de la misma. Una de las marcas usualmente más utilizadas en la industria como lo es Siemens actualmente cuenta con herramientas específicas para desarrollar controladores inteligentes tales como un controlador en lógica difusa, sin embargo, el uso de estas herramientas tiene un costo adicional para poder hacer uso de ellas. De igual manera cuenta con funciones para diseñar controladores con estrategias de control avanzado tales como los PID de tipo A, B o C basadas en puntos de ajustes ponderados.

**Tabla 1**

*Marcas de PLC's que implementan controladores a nivel industrial*

Marca	Tipo de PID	Ecuación
<b>Allen Bradley Logix5550</b>	Independiente tipo B	$CO = K_p e + K_i \int e dt + K_d \frac{d(-PV)}{dt}$
<b>GE Fanuc Series 90-30 y 90-70</b>	Independiente tipo A	$CO = K_p e + K_i \int e dt + K_d \frac{de}{dt}$
<b>Modicon 984</b>	Dependiente tipo A	$CO = \frac{100}{PB} \left( e + K2 \int e dt + K3 \frac{de}{dt} \right)$

Nota: Tomado de *PID Controller Auto-Tuning Software*, por BESTune, 2006.

Actualmente se encuentran desarrollando alternativas para implementar técnicas de control avanzado como la marca Siemens que cuenta con funciones respecto al diseño de controladores PID, esto se puede corroborar en la fuente de información (Siemens, 2018). Se toma a consideración que, en sí, estas marcas de PLC poseen bloques de función en los cuales permite embeber un controlador PID. Para utilizarlas se requiere de tiempo y entrenamiento sobre la interfaz de programación del PLC y así comprender su correcto funcionamiento. Estas funciones para la implementación de estrategias de control avanzado, en ciertas marcas de PLC se las debe adquirir por un costo adicional determinado. Por lo que a continuación se detalla un breve estado del arte que, en base a nuestro conocimiento, se presentan investigaciones en las cuales se han embebido distintas estrategias de control avanzado en PLC's.

### Estado del arte

En investigaciones tales como en (Portillo & Ordoñez, 2020) se diseña e implementa un controlador difuso para realizar el control de un motor trifásico mediante frecuencia en un PLC de la marca Siemens S7-1200. El diseño del controlador difuso se realizó en el software Matlab y para poder implementarlo se tuvo que desarrollar en un bloque de funciones propio del PLC en SCL (lenguaje

estructurado de Siemens) para el desarrollo del algoritmo inteligente. También respecto a la implementación de controladores difusos se presenta en (Ponce & Ramírez, 2010) metodologías basadas en LabVIEW, puesto que esta herramienta contiene librerías en donde se pueden generar las funciones de membresía correspondientes y realizar las operaciones de fusificación tales como intersección, unión, complemento y producto. Por lo que esta metodología propone a través de sus herramientas específicas para control difuso implementar el diseño del controlador mediante operaciones y bloques de función en ofrecidas por LabVIEW.

En trabajos de investigación tales como en (Ferri, 2019) se diseña el software correspondiente a un controlador predictivo por modelo para la implementación del mismo en un PLC de la marca Schneider del tipo Modicon 241. El diseño y generación de código de este controlador se lo realiza en el software computacional Matlab, mismo que exporta el diseño del controlador en texto estructurado para así poder implementarlo en el entorno de SoMachine. Una vez que se ha generado el código en texto estructurado, se deben definir las distintas funciones que permiten hacer uso del controlador generado en la plataforma de SoMachine, para esto en el lenguaje de CodeSys se debe realizar la generación de las funciones correspondientes para que el código generado funcione en el PLC. Este proyecto presenta similitud con el presente proyecto de investigación desarrollado en este documento ya que, se realiza la generación de código mediante Matlab / Simulink hacia el lenguaje CodeSys, pero su diferencia es que no se implementa directamente hacia el controlador si no que debe pasar por ciertas modificaciones en bloques de funciones específicas para la plataforma de SoMachine; mientras que en este proyecto la implementación es directamente sobre el PLC Turck una vez generado el código en CodeSys sin necesidad de modificar a la estructura general del código generado.

Otro proyecto basado en la implementación de los controladores diseñados mediante el entorno de desarrollo LabView es el que se presenta en (Johnson, 2010) donde se diseñan e implementan tres diferentes tipos de control tales como control PI tradicional, control PI fraccionario y

control adaptativo por referencia a modelo. Estos controladores fueron diseñados en base a bloques de función de LabView específicamente para el control de temperatura sobre una plataforma de automatización industrial. Las simulaciones del control PI fraccionario fueron llevadas a cabo en el software de Matlab, sin embargo, para su debida implementación hacia el controlador industrial NI compact FieldPoint la estrategia de control fue también realizada mediante bloques de implementación en LabView.

El desarrollo de estrategias de control avanzado tales como el control con redes neuronales, controlador en la estructura RST (dos grados de libertad) y un controlador PI estimado por el método de asignación de polos, son diseñados e implementados en un PLC de la marca Siemens para el control de una planta de flujo y nivel en (García & Giraldo, 2012). En este proyecto de investigación citado se diseñan las estrategias control mencionadas en lenguaje SCL para ser implementadas mediante bloques de función y datos propios del PLC. Se utiliza en total 10 bloques, entre bloques de función y bloques de datos para poder implementar el diseño de los controladores avanzados. A diferencia del presente proyecto descrito en este documento, donde se genera código para implementarlo directamente al PLC. La única similitud que se encuentra es la metodología aplicada para realizar la identificación del sistema ya que se utiliza la herramienta de identificación IDENT de Matlab.

### **Justificación e Importancia**

El desarrollo del área de control digital en procesos industriales que manejan variables continuas como temperatura, flujo o nivel ha desatado una rama de investigación muy firme y concreta. Esta línea de investigación posee mucho desarrollo teórico respecto a algoritmos de control inteligente, los cuales mejoran el desempeño de los procesos controlados, sin embargo, actualmente los beneficios que se plantean en las investigaciones acerca de estos algoritmos no son llevados a producción en entornos industriales reales, es decir que el desarrollo en esta línea a nivel teórico es muy amplio, sin embargo en el aspecto de implementación, en esta rama no se ha alcanzado un desarrollo significativo.



La literatura respecto a la aplicación e implementación de controladores inteligentes en la industria ecuatoriana es limitada. En vista de que tradicionalmente se aplican en su mayoría controladores automáticos como el controlador proporcional integral PID y sus variaciones; es importante comparar el desempeño de estos controles automáticos clásicos, con controladores inteligentes.

La baja implementación de nuevas estrategias de control de procesos, tales como en plantas de temperatura muy comunes en la industria, limita la mejora de desempeño del proceso productivo general de la industria. Como se menciona en (Feijoo, Chanchay, Llanos, & Ortiz, 2021) las técnicas de control avanzado implementados en los ambientes industriales representan un mejor desempeño, incluso permitiendo incrementar la vida útil de los actuadores. En investigaciones tales como en (Ferri, 2019) se hace mención a la importancia de adaptar los procesos productivos hacia los requerimientos de la industria 4.0, por lo que estos debidos avances y desarrollos tecnológicos tienen la necesidad de poder implementar estrategias de control más avanzadas, de características no lineales a las cuales pueden acoplarse las técnicas de control predictivo o inteligente, en los PLC's.

Las limitaciones de implementación de los controladores inteligentes radican en el costo de programación que poseen estos modelos, ya que, la mayoría de los equipos PLC's comerciales, donde se embeben los controladores automáticos poseen únicamente la opción del controlador PID. Otra razón crítica es que las funciones matemáticas que se requieren para lograr un controlador inteligente son complejas y las opciones a nivel de funciones que nos brindan la mayoría de controladores son limitadas.

Como se muestra en el artículo de Jae-Sub Ko (Ko, Huh, & Kim, 2019) donde se implementa un controlador de temperatura difuso PI, el cual posee características elevadas de desempeño a nivel de diseño, el artículo propone el uso de hardware de desarrollo que embebe al controlador difuso con una

tarjeta de adquisición Arduino, donde las limitaciones son evidentes, ya que requiere un módulo para trabajar con las salidas analógicas y módulos para recibir los valores de temperatura, la electrónica utilizada en este caso no posee características para entornos industriales, fiabilidad ni robustez. Además, como se detalla en el artículo el costo de programación de algoritmo es elevado, ya que se desarrolló desde cero, tomando en cuenta que las librerías de Arduino son limitadas y la capacidad de procesamiento, respecto a los resultados, no se obtuvieron las características iniciales debido a los retrasos y desfases de tiempo que genera el Arduino (Chabni, Rachid, & Benbouali, 2016).

Actualmente en la carrera de Ingeniería en Electrónica, Automatización y Control, en las materias de sistemas de control, control digital y control inteligente se diseñan controladores y modelos de simulación con el apoyo de Matlab, donde se analiza la respuesta y desempeño de los controladores diseñados, en las versiones nuevas de Matlab se han implementado compiladores de código automático para PLC, donde a través de Simulink inicialmente el diseñador plantea el controlador inteligente y Matlab genera un código automático compatible con ciertos modelos de PLC's en lenguaje de código estructurado ST. Por lo tanto, se recalca una de las principales razones por la que se desea explorar la calidad y desempeño de los controladores que se realizan digitalmente a través de la generación de código en vista de que convencionalmente no han sido utilizados.

De igual manera el uso de marcas no comerciales de PLC's en el país distintas a Siemens o Schneider, permiten expandir los conocimientos acerca del diseño e implementación de controladores inteligentes embebidos en PLC's y su aplicabilidad en entornos industriales. Por lo tanto, en el presente proyecto de grado se propone el diseño e implementación de estrategias de control avanzado basadas en controladores inteligentes, los cuales serán embebidos en un PLC de marca TURCK, el cual es poco conocido en el país, sin embargo, posee ventajas frente a otros equipos ya que utiliza una plataforma de programación gratuita denominada CODESYS. Los controladores serán aplicados a una planta de temperatura la cual esta desarrollada en base a un entorno industrial a escala real.

## **Alcance del proyecto**

El presente proyecto de investigación plantea realizar el diseño e implementación de estrategias de control avanzado embebidas en un controlador lógico programable y aplicadas a un entorno industrial de escala real. El control avanzado, entre otras, se centra en el diseño e implementación de distintas estrategias como el control Multivariable, modelos de control Predictivo o Lógica Difusa (Ferri, 2019).

Se desarrollará el proceso de identificación de la planta y el proceso de puesta en marcha de los controladores automáticos a diseñar. Para las etapas de diseño, identificación de la planta de temperatura e implementación de los controladores se trabajará con un troceador Triac con entrada analógica cuya función es administrar energía controlada al sistema de calefacción el cual poseerá una PT100 para sensar la temperatura, un PLC de marca TURCK, que se programa y se configura con el software CODESYS, el cual es gratuito.

De esta manera, se propuso realizar la identificación de una planta de temperatura, para consiguiente, realizar el control de esta a partir del diseño de un controlador PID dos grados de libertad, un controlador en espacio de estados con capacidad de rechazo a las perturbaciones, ambos basados en modelo y un controlador autoajustable difuso con un diseño libre de modelo. Los controladores a diseñarse operarán en tiempo discreto, los diseños se realizarán de forma digital directa los cuales permiten que los controladores alcancen características y niveles de desempeño y robustez superiores a los algoritmos de aproximación que poseen los PLC's comerciales actualmente.

Posteriormente a través del uso de las herramientas de Matlab y Simulink para la generación de código automático para PLC's se implementaron dichos controladores en la planta de temperatura, tanto para la variable controlada como para la salida del controlador.

## **Objetivos**

### *Objetivo general*

Diseñar e implementar estrategias de control avanzado embebidas en un controlador lógico programable y aplicadas a un entorno industrial de escala real.

### *Objetivos específicos*

- Realizar una correcta identificación de la planta de temperatura con el uso del PLC TURCK, para obtener un gemelo digital con mínimo error.
- Diseñar estrategias de control avanzado para la planta de temperatura identificada.
- Generar el código correspondiente a los controladores diseñados mediante la herramienta de Matlab y Simulink para la plataforma de software CODESYS.
- Análisis de desempeño de las estrategias de control avanzado tanto en simulación como en tiempo real.

## **Metodología de la investigación**

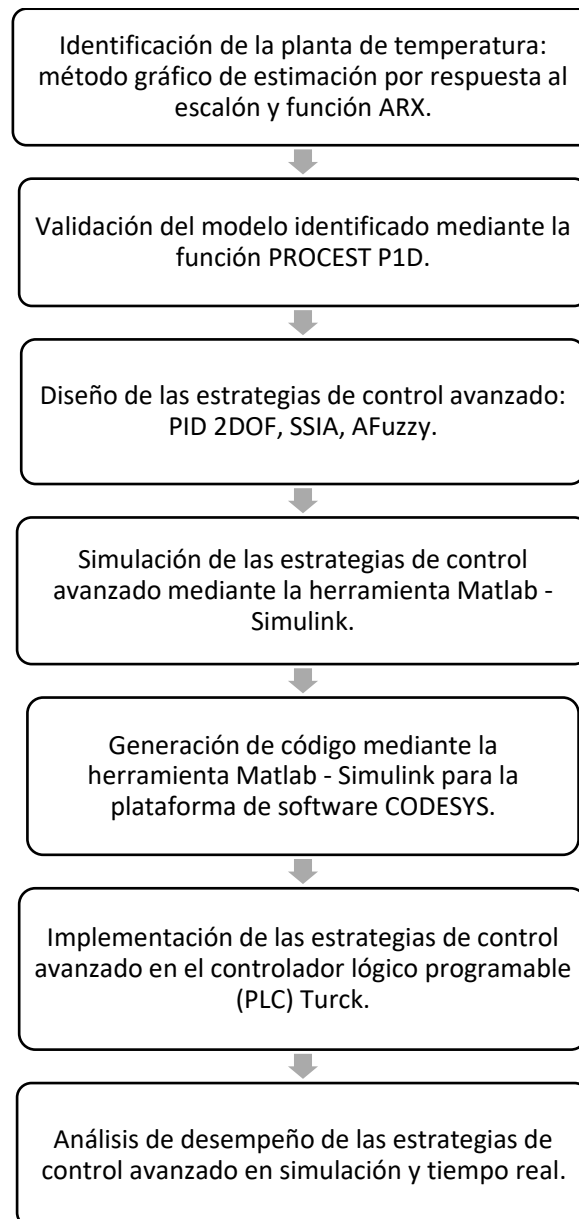
La metodología llevada a cabo para el presente proyecto de investigación se muestra en la figura 1. Se detalla el orden de la metodología propuesta como: la identificación de la planta de temperatura mediante el método gráfico de estimación por respuesta al escalón y la función ARX de la herramienta IDENT de Matlab para su posterior validación a través de la función PROCEST P1D, el diseño de las estrategias de control avanzado partiendo de un PI 2DOF (PI dos grados de libertad), un SSIA (controlador por espacio de estados con acción integral) y un AFuzzy (controlador PI difuso autoajutable), la simulación de los controladores diseñados y la generación de código a través de Matlab – Simulink para embeberlo en el PLC a través de la plataforma de software CODESYS.

El elemento final de la metodología del proyecto es el análisis de desempeño de las estrategias de control avanzado tanto en simulación como en la implementación de los mismos en tiempo real. Se

toma como métricas de desempeño el seguimiento a referencia, rechazo a las perturbaciones, sobre impulso generado, tiempo de establecimiento, tiempo de subida y el consumo de energía de cada controlador.

### Figura 1

*Metodología aplicada al proyecto de investigación*



Acorde a la metodología expuesta la organización del proyecto de investigación se desglosa de la siguiente manera. En el capítulo I se desarrolla una breve introducción al proyecto donde se detallan los antecedentes, estado del arte, justificación e importancia, alcance, objetivos y la metodología a emplear en el desarrollo de la investigación.

En el capítulo II se describe el marco conceptual del proyecto abarcando los temas de identificación de sistemas y los métodos a aplicar tales como: el método gráfico de estimación por respuesta al escalón, función ARX y PROCEST P1D de la herramienta IDENT de Matlab, las estrategias de control avanzado para el diseño del controlador PI 2DOF (PI dos grados de libertad), SSIA (controlador por espacio de estados con acción integral) y un controlador AFuzzy (PI difuso autoajutable), la generación de código para controladores mediante la herramienta Matlab/Simulink y el PLC de la marca TURCK tanto como la plataforma de software CODESYS.

En el capítulo III se abarca el diseño de las estrategias de control avanzado correspondientes al controlador PI 2DOF, SSIA y un controlador AFuzzy.

En el capítulo IV se presenta la respectiva generación de código para las estrategias de control diseñadas.

En el capítulo V se ejecuta la simulación de la respuesta de cada controlador y la implementación en la planta industrial a escala real, con lo que se interpretó el desempeño de las estrategias de control avanzado tanto en simulación como en tiempo real.

Finalmente, en el capítulo VI se plantea las conclusiones y recomendaciones recopiladas a lo largo del trabajo de investigación propuesto.

## Capítulo II. Marco conceptual

En este capítulo se describe el marco conceptual acerca de: la identificación de sistemas y los diferentes métodos utilizados en el proyecto, tales como, el método gráfico de estimación por respuesta al escalón, la función ARX y PROCEST P1D de la herramienta IDENT de Matlab, los controladores PI 2DOF, SSIA y un PI AFuzzy, la generación de código mediante la herramienta Simulink, el PLC a utilizar en el proyecto y el software utilizado para embeber los controladores.

### Identificación de sistemas

La identificación de sistemas representa el proceso de usar datos en lugar de usar leyes físicas para generar el modelo de un sistema dinámico. Un modelo es la representación real de un sistema y la idea principal de la identificación de sistemas es implementar una copia con las características esenciales del sistema a tratar. La importancia de obtener un modelo real del sistema desemboca en que se pueden diseñar estrategias de control, realizar análisis y simulaciones, todo esto forma parte de lo que se conoce como diseño basado en modelos (Douglas, 2021).

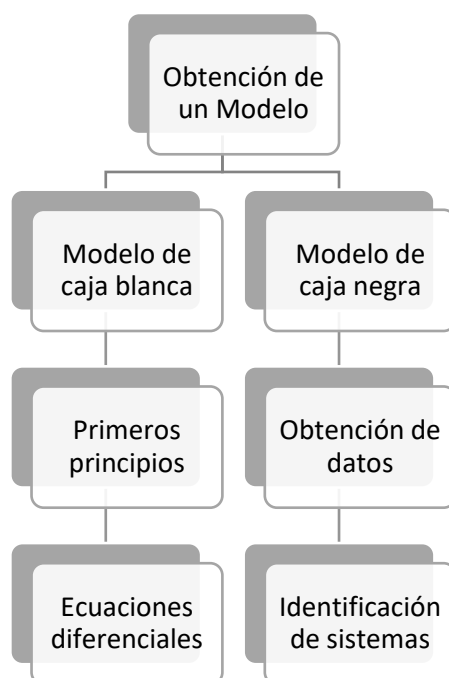
De manera que lo menciona el autor (Garrido, 1999) se puede definir a la identificación de sistemas, como la estimación de modelos de sistemas dinámicos mediante datos observados. Siendo así la identificación de sistemas, permite imitar lo más cercano a la realidad el sistema desconocido de forma experimental, a través de modelos matemáticos según sea el caso. También se tiene una definición desde hace varios años ofrecida por el autor Lofti Zadeh (Zadeh, 1962) que describe “Identificación es la determinación, en base a la entrada y la salida, de un sistema, dentro de una clase de sistemas identificada, al cual el sistema probado es equivalente”.

La identificación de sistemas permite realizar un análisis espectral del modelo, como, por ejemplo, en el área de las comunicaciones; permite la detección de fallos, reconocimiento de patrones, control adaptativo, predicciones lineales, entre otras (Söderström & Stoica, 1989).

Por lo tanto, la identificación del sistema o la planta, es el paso fundamental para poder realizar un adecuado control sobre el sistema. En la figura 2 se muestra de manera global la obtención de modelos, en donde en base a lo que se conozca sobre el mismo existen dos métodos de obtención de modelos. El modelo de caja blanca es cuando se conoce a ciencia cierta el comportamiento del sistema a través de los primeros principios que gobiernan sobre el mismo, por ende, se puede obtener de manera directa el modelo mediante ecuaciones diferenciales. Por otro lado, cuando no se conoce del todo al sistema, se conoce a esto como el modelo de caja negra en donde a través de la identificación de sistemas mediante la obtención de datos se puede generar el modelo de la planta.

**Figura 2**

*Clasificación de la obtención de modelos*



### ***Métodos para la identificación de sistemas***

Para los diferentes métodos de identificación de sistemas, se puede considerar en principio dos tipos de identificación de sistemas tales como la identificación en línea y fuera de línea. La identificación en línea se la realiza mediante técnicas de identificación paramétrica como lo son el método de mínimos

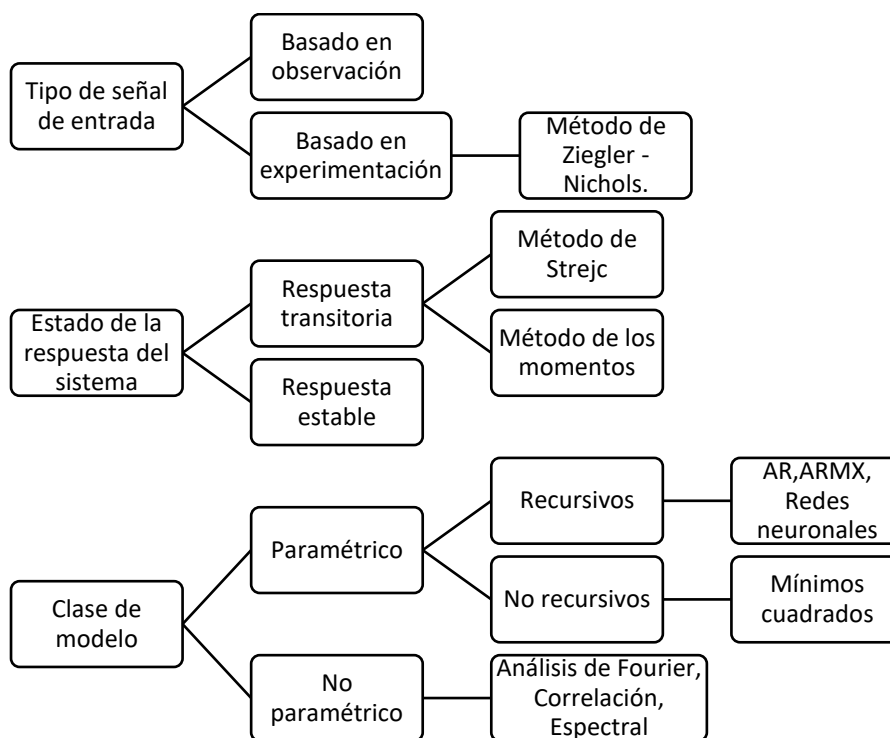


cuadrados, este tipo de método permite identificar el sistema mediante la actualización de los parámetros calculados según el tiempo de muestreo de un modelo que ya haya sido definido previamente. Por otro lado, la identificación fuera de línea se basa en el tratamiento de datos que se han obtenido mediante la experimentación de la planta (Aluisa, 2014).

En la figura 3, se detallan ciertos métodos correspondientes a modelos de tipo lineal, según el tipo de señal de entrada o el estado de la respuesta del sistema.

**Figura 3**

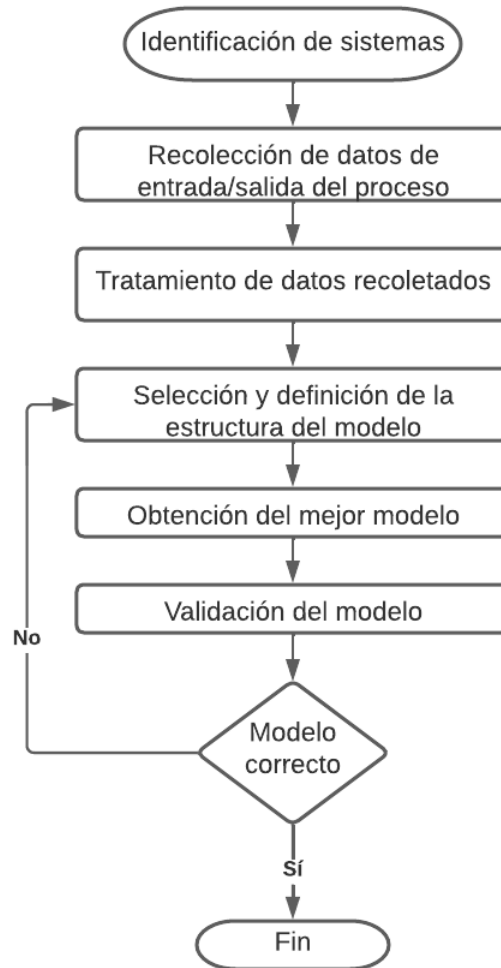
*Métodos para la identificación de sistemas*



Es posible sintetizar de manera general los pasos que se deben llevar a cabo cuando se utiliza uno de los métodos descritos anteriormente, en base al siguiente diagrama de flujo mostrado en la figura 4.

**Figura 4**

*Pasos generales a seguir para la identificación de sistemas*



A continuación, se describe a mayor detalle los métodos de identificación utilizados en el presente proyecto de investigación, como lo son el método gráfico de estimación por respuesta al escalón, la función ARX y PROCEST P1D de la herramienta IDENT de Matlab.

#### **Método gráfico de estimación por respuesta al escalón**

Partiendo de los antecedentes sobre sistemas térmicos se conoce que el modelo a obtener es un modelo continuo correspondiente a un sistema de primer orden con retardo o tiempo muerto como se muestra en la ecuación 1.1.

$$G(s) = \frac{K * e^{-s*t_d}}{(s + a)} \quad (1.1)$$

El método gráfico de estimación por respuesta al escalón permite determinar los parámetros de comportamiento de la planta, obtenidos por cualquier método de registro de datos (Interiano, 2013). Por lo tanto, el método consiste en observar la respuesta del sistema ante una entrada tipo escalón y realizar un debido análisis sobre la misma según los parámetros que se muestran a continuación.

Constante de tiempo:

$$\tau = 2(t_{63\%} - t_{39\%}) \quad (1.2)$$

$$a = \frac{1}{\tau} \quad (1.3)$$

Ganancia del sistema:

$$K = \frac{\lim_{t \rightarrow \infty} y(t) - T_i}{A} * a \quad (1.4)$$

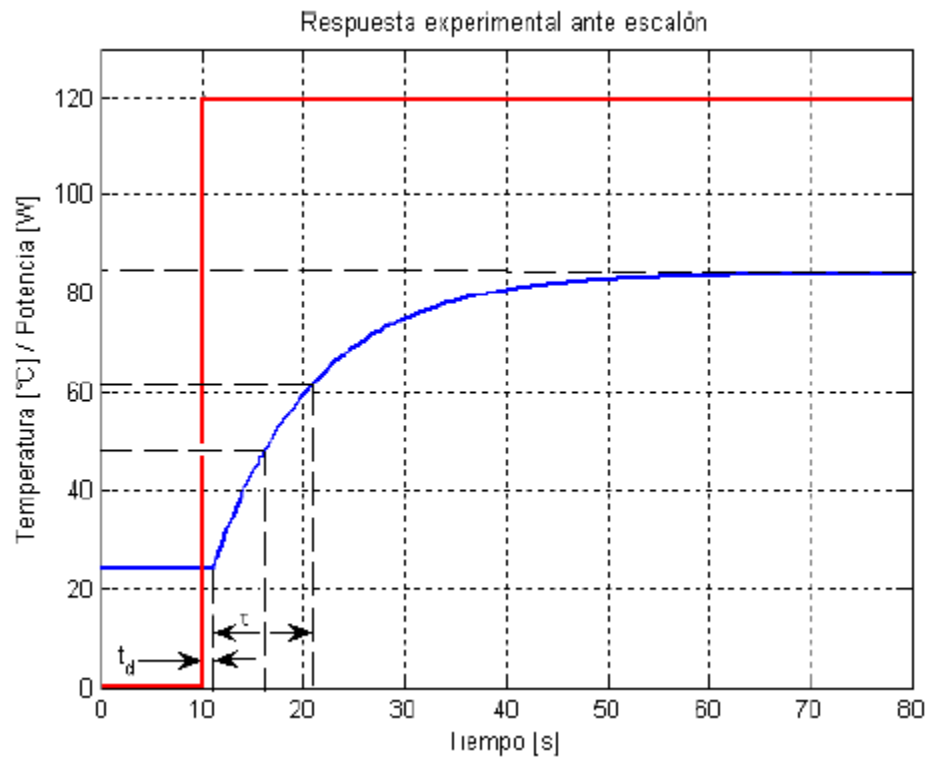
Retardo:

$$t_d = t_{63\%} - (T_i + \tau) \quad (1.5)$$

Para entender de mejor manera los parámetros mostrados en las ecuaciones 1.2-1.5 se presenta la figura 5, donde se puede observar cómo obtener estos parámetros a partir del gráfico de la respuesta al escalón. En esta figura se representa un ejemplo de la respuesta al escalón de una planta de temperatura con retardo.

**Figura 5**

*Parámetros de la respuesta al escalón de una planta de temperatura con retardo*



Nota: Tomado de *Control Automático. Ejemplo de control de un sistema térmico*, por Eduardo Interiano, 2013.

### **Función ARX. IDENT (System Identification toolbox)- Matlab**

Esta herramienta de identificación ofrecida por el software computacional de Matlab permite crear modelos de sistemas dinámicos lineales discretos a partir de datos tomados de entrada y salida. Esta aplicación utiliza datos de entrada y salida en el dominio del tiempo y también en el dominio de la frecuencia para identificar funciones de transferencia en tiempo discreto, modelos de procesos y modelos en espacio de estados (Matlab, 2022). System Identification toolbox realiza la identificación de sistemas mediante el término de caja negra para poder estimar los parámetros de un modelo en específico.

Los modelos de caja negra hacen referencia a los casos en donde previamente se conoce que el proceso puede ser descrito como un sistema lineal en un rango operacional particular (Aström & Wittenmark, Computer Controlled Systems, 1997) como por ejemplo un modelo representado por la siguiente ecuación a diferencias:

$$\mathbf{A}(z)\mathbf{y}(k) = \mathbf{B}(z)\mathbf{u}(k) + \mathbf{C}(z)\mathbf{e}(k) \quad (1.6)$$

Donde  $u$  representa la entrada,  $y$  la salida y  $e$  una perturbación de tipo ruido blanco. Tanto los parámetros como el orden de los modelos se consideran desconocidos, por tal razón hacen mención a un modelo de caja negra. De esta manera se realiza la identificación de sistemas mediante el método de los mínimos cuadrados en el cual la estructura del modelo es representada por el tipo de función ARX (Martínez, 2011). Partiendo de este método se obtiene la función discreta del sistema representada en la ecuación 1.7:

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (1.7)$$

De donde:

$$\Delta y_k + a_1 \Delta y_{k-1} + a_2 \Delta y_{k-2} = b_1 \Delta u_{k-1} + b_2 \Delta u_{k-2} \quad (1.8)$$

Por lo tanto, este método de identificación se basa principalmente en obtener los valores de  $a_1, a_2, b_1, b_2$ . Para esto se utiliza tanto el tipo de función como lo es ARX y el método de los mínimos cuadrados para estimar los valores antes mencionado, obteniéndose así la siguiente representación matemática.

$$\Delta y_k + a_1 \Delta y_{k-1} + a_2 \Delta y_{k-2} = b_1 \Delta u_{k-1} + b_2 \Delta u_{k-2} + e_k \quad (1.9)$$

$$\Delta y(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \Delta u(z) + \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} e(z) = \frac{B}{A} \Delta u + \frac{1}{A} e \quad (1.10)$$

Estos algoritmos son uno de los varios que se encuentran disponibles en la herramienta de Matlab. De esta manera en este apartado se pretende mostrar las bases sobre las cuales se fundamenta la herramienta IDENT de Matlab para realizar la identificación de sistemas, misma que fue utilizada para la identificación de la planta de temperatura del presente proyecto.

### **Función PROCEST P1D**

La función PROCEST para la identificación de sistemas corresponde al modelado de procesos, se basa en la estimación del modelo con datos basados en tiempo o frecuencia. El modelo de un proceso de una sola entrada y una sola salida con una ganancia, constante de tiempo y retardo mediante esta función se representa como se muestra en la ecuación 1.11, la cual es equivalente a la ecuación 1.1 para un sistema de primer orden con retardo.

$$sys = \frac{k_p * e^{-s * t_d}}{1 + T_{p1}s} \quad (1.11)$$

Para generar la función de identificación se utiliza la forma  $sys = procest(data, type)$ , donde se debe especificar los datos tomados sobre la planta de entrada y salida detallando el tipo de estructura que definirá al sistema  $sys$ , para el presente proyecto se utiliza la estructura P1D ( donde "P" es los polos del sistema, 1 el número de polos y "D" el tiempo de retardo del sistema), obteniendo una función de una sola entrada y una sola salida con un polo y tiempo de retardo.

### **Estrategias de control avanzado**

Las estrategias de control avanzado son óptimas y robustas a su aplicación. El objetivo principal de un control óptimo es maximizar el retorno de un sistema al mínimo costo. Existen diferentes escenarios para los cuales es recomendable diseñar e implementar estrategias de control avanzado que sean óptimas tales como el control de mínima energía mismo que forma parte del alcance del presente

trabajo. El control de mínima energía permite transferir el sistema desde un estado inicial hacia un estado final con el mínimo consumo de energía del controlador.

Dentro de las estrategias de control avanzado también se encuentra el diseño de sistemas de control inteligente. Los controladores inteligentes tienen la capacidad de conocimiento, razonamiento y entendimiento, por lo tanto, se tiene la habilidad de razonar, planificar y aprender. Junto a la inteligencia artificial y el control avanzado, se pueden diseñar sistemas autónomos que actúen de manera inteligente y sean capaces de lograr el comportamiento deseado bajo condiciones de incertidumbre (Burns, 2001). En el presente trabajo de investigación el controlador inteligente con el que se trabajará es un controlador autoajustable difuso, mismo que se abordará más adelante en las secciones correspondientes al diseño e implementación del controlador AFuzzy.

Se debe tomar en consideración que las estrategias de control avanzado, permite tratar con sistemas complejos que pueden tener tanto múltiples entradas como múltiples salidas variantes en el tiempo. Esto representa el caso de trabajar con metodologías en espacio de estados.

### ***Controlador PID***

Para construir un controlador PID 2DOF en primera instancia se debe partir de un PID un grado de libertad. Es importante conocer las bases de cómo se conforma un controlador PID tal como su estructura en el dominio del tiempo, el dominio de la frecuencia, su representación en diagrama de bloques, su forma estándar o paralela, al igual que los diferentes tipos de PID tales como el A, B o C en donde el PID tipo C puede representar un controlador 2DOF debido a su estructura. En este apartado se abarcan conceptos básicos e importantes sobre los controladores PID y también se plantea el diseño de un controlador PID en forma digital directa tal como el PID discreto con filtro de la acción derivativa y la sintonización de controladores por el método de Ziegler-Nichols.

El controlador PID ideal en el dominio del tiempo se puede representar como se muestra en la ecuación 1.12:

$$u(t) = k_p e(t) + k_i \int_0^t e(t) dt + k_d \frac{d}{dt} e(t) \quad (1.12)$$

Si se transforma al dominio de la frecuencia S se tiene:

$$k_{PID}(s) = k_p + \frac{k_i}{s} + k_d s \quad (1.12.1)$$

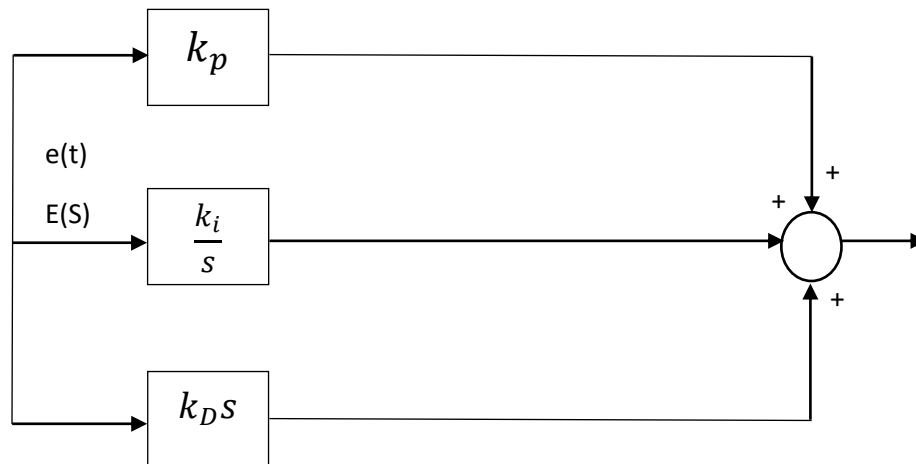
Factorizando el término  $k_p$ :

$$k_{PID}(s) = k_p \left( 1 + \frac{1}{t_i s} + t_d s \right) \quad (1.12.2)$$

Representando en diagrama de bloques la ecuación de un controlador PID ideal factorizando el término  $k_p$ , en donde se puede observar que el valor proporcional del controlador  $k_p$  depende del error actual  $e(t)$ , el valor integral de errores pasados  $E(s)$ , mientras la parte derivativa representa una predicción de los errores futuros, por lo que realizando la suma de estos tres parámetros se puede obtener un control sobre el proceso en el que se esté trabajando, como se puede observar a continuación en la figura 6.

**Figura 6**

*Diagrama de bloques de un controlador PID ideal*





Por lo tanto, se tiene la representación de la clasificación de los controladores PID como se muestra en las ecuaciones 1.13 y 1.14:

Forma estándar o dependiente:

$$C(s) = k \left[ 1 + \frac{1}{st_i} + st_d \right] \quad (1.13)$$

Forma paralela o independiente:

$$C(s) = k_c + \frac{ki}{s} + sk_d \quad (1.14)$$

Donde:

$$k = k_c \quad (1.14.1)$$

$$ki = \frac{k}{t_i} \quad (1.14.2)$$

$$k_d = k * t_d \quad (1.14.3)$$

### **PID Tipo A.**

La clasificación de los PID según la señal con la que se trabaja parte desde la forma paralela o independiente. De esta manera se tiene que el PID tipo A solo procesa la señal de control, por lo que se tiene la versión posición del controlador:

$$u(t) = k_c e(t) + k_i \int_0^t e(t) dt + k_d \frac{d}{dt} e(t) \quad (1.15)$$

### **PID Tipo B.**

Si se deriva la ecuación del controlador PID tipo A se tiene la versión velocidad del controlador que representa a un PID tipo B:

$$u(t) = k_c e(t) + k_i \int_0^t e(t) dt - k_d \frac{d}{dt} y(t) \quad (1.16)$$

### PID Tipo C.

El controlador PID tipo C es también conocido como la arquitectura dos grados de libertad de un PID, el cual parte inicialmente de un controlador PID tipo A, por lo tanto, se tiene:

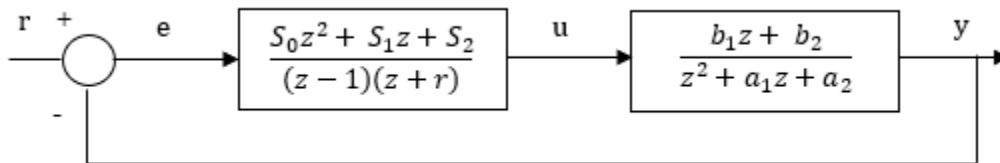
$$u(t) = k_i \int_0^t e(t) dt - \left[ k_c y(t) + k_d \frac{d}{dt} y(t) \right] \quad (1.17)$$

### Controlador PID en forma digital directa.

El diseño de un controlador PID en forma digital directa presenta la siguiente estructura mostrada en la figura 7, en donde se muestra la estructura del controlador digital mismo que ingresa a la planta con una estructura digital determinada. A continuación, se presenta el desarrollo del diseño de este controlador aterrizando en las matrices correspondientes a P, C y D, importantes al momento de construir este diseño.

**Figura 7**

*Estructura de un controlador PID en forma digital directa*



$$\Delta(z) = (z^2 + (r-1)z - r)(z^2 + a_1z + a_2) + (S_0z^2 + S_1z + S_2)(b_1z + b_2) \quad (1.18)$$

$$\Delta d(z) = (z^2 + d_1z + d_2)(z + z_1)(z + z_2) = z^4 + d_1z^3 + d_2z^2 + d_3z + d_4 \quad (1.18.1)$$

$$\Delta(z) = \Delta d(z)$$

$$z^4: 1 = 1$$

$$z^3: a_1 + (r-1) + S_0b_1 = d_1$$

$$z^2: -r + a_2 + a_1(r-1) + S_0b_2 + S_1b_1 = d_2$$

$$z: -ra_1 + a_2(r-1) + S_1b_2 + S_2b_1 = d_3$$

$$\mathbf{z}^0: -a_2 r + S_2 b_2 = d_4$$

$$\begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ r \end{bmatrix} = \begin{bmatrix} d_1 - a_1 + 1 \\ d_2 + a_1 - a_2 \\ d_3 + a_2 \\ d_4 \end{bmatrix}$$

$$P = \begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix}$$

$$C = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ r \end{bmatrix}$$

$$D = \begin{bmatrix} d_1 - a_1 + 1 \\ d_2 + a_1 - a_2 \\ d_3 + a_2 \\ d_4 \end{bmatrix}$$

### Sintonización de controladores por el método de Ziegler-Nichols.

Las reglas de sintonización por el método de Ziegler-Nichols es uno de los métodos más populares. Su objetivo principal es proporcionar un buen rechazo a las perturbaciones. Estas reglas suelen ser empíricas, sin embargo, son consistentes con el significado físico respecto a los parámetros correspondientes a un PID, como lo son los valores de  $k_p$ ,  $T_i$ ,  $T_D$ . Estos valores son los que hacen referencia a los diferentes efectos que se tiene sobre la acción derivativa, integral y proporcional. Así se sabe que la acción derivativa presta una amortiguación adicional al sistema aumentando su estabilidad relativa por lo que se puede aumentar la acción proporcional tanto como la integral para mantener un tiempo de respuesta estable (Fadali & Visioli, 2013). Por lo tanto, se muestra en la tabla 2 las reglas de sintonización por Ziegler-Nichols para sistemas de primer orden con tiempo muerto.

**Tabla 2**

*Reglas de sintonización por Ziegler-Nichols para un sistema de primer orden con tiempo muerto*

Tipo de controlador	$K_P$	$T_i$	$T_d$
<b>P</b>	$\frac{\tau}{KL}$	-	-
<b>PI</b>	$0.9 \frac{\tau}{KL}$	$3L$	-
<b>PID</b>	$1.2 \frac{\tau}{KL}$	$2L$	$0.5L$

*Nota.* Tomado de *Digital Control Engineering Analysis and Design*, por Fadali y Visioli, 2013.

#### **Sintonización de controladores mediante la herramienta PID Tuner Free Web Site.**

La herramienta de sintonización de controladores PID Tuner Free Web Site permite estimar los valores para diferentes tipos de controles P, PI, PD y PID basados en la identificación del modelo con el que se va a trabajar. La herramienta utiliza los datos adquiridos mediante la respuesta al escalón del sistema y permite realizar un paso a paso del proceso de sintonización para mejorar la afinación del mismo (PID Tuner Controller, 2022). La herramienta recomienda conocer el paso a paso que se sigue para realizar la sintonización de los controladores, el paso inicial consiste en que el diseñador conozca las especificaciones de diseño requeridas para un comportamiento óptimo del sistema tales como el sobre impulso, seguimiento a referencia y rechazo a perturbaciones. Se debe obtener los datos correspondientes a la respuesta en lazo cerrado del sistema ante una entrada tipo escalón para cumplir con el primer paso de la herramienta. El segundo paso consiste en la recolección de los datos adquiridos en el paso anterior tomando en cuenta que el tiempo de muestreo seleccionado debe ser suficiente para capturar la dinámica del sistema, estos datos se deben guardar en un archivo CSV. Una vez importados los datos se procede a elegir el mejor modelo que represente al sistema y finalmente se estiman los valores correspondientes al controlador PID.

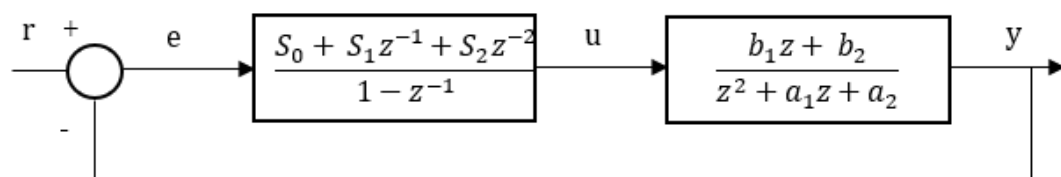
### Estructura controlador PID 2DOF.

Los controladores PID 2DOF representa un tipo de estructura de control que permite enfrentar a las perturbaciones del sistema y a los cambios de señal en el set point (Alfaro, Arrieta, & Vilanova, 2009). Según lo menciona el autor (Ogata, 2010) más de la mitad de los controladores industriales usados en la actualidad son esquemas de control PID o PID's modificados tales como el PID dos grados de libertad. De igual manera se recalca la utilización de los controladores PID en los casos en donde no se tiene conocimiento sobre el modelo matemático de la planta y no es posible utilizar métodos de diseño analítico. Sin embargo, en sistemas de control de procesos, los PID básicos no suelen ser los más óptimos a pesar de aportar con un control satisfactorio. Los controladores PID con distintos grados de libertad tienen la oportunidad de satisfacer especificaciones cuando se tienen perturbaciones o conflictos en el diseño. Al aumentar los grados de libertad del controlador, se puede cumplir con requerimientos referentes a perturbaciones de entrada y seguimiento a la referencia.

Para poder comprender el funcionamiento de un controlador PID 2DOF, se parte del conocimiento de un controlador PID simple, como por ejemplo un PID tipo A el cual se describió anteriormente y se detalló la ley de control correspondiente en la ecuación 1.15. A partir de esta ecuación se realiza una digitalización del controlador obteniéndose así la estructura correspondiente a un PID tipo A digital el cual se puede observar a continuación en la figura 8.

**Figura 8**

*Controlador PID digital Tipo A*



En la figura 8 se puede apreciar que el controlador se encuentra digitalizado al igual que la planta, entonces la ley de control del PID tipo A en forma digital corresponde a:

$$u(n) = u(k-1) + s_0 e(k) + s_1 e(k-1) + s_2 e(k-2) \quad (1.19)$$

Donde:

$$s_0 = k_p + k_i T + \frac{k_d}{T} \quad (1.19.1)$$

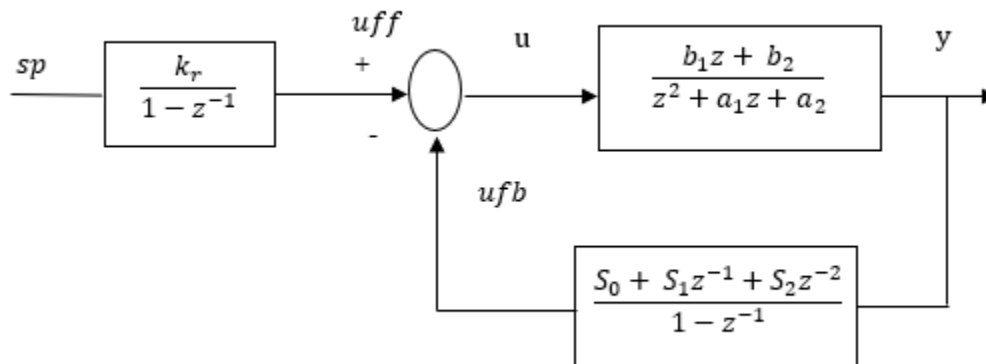
$$s_1 = -k_p - 2 \frac{k_d}{T} \quad (1.19.2)$$

$$s_2 = \frac{k_d}{T} \quad (1.19.3)$$

En la figura 9, se detalla el diagrama de bloques de un controlador PID 2DOF mismo que corresponde a la arquitectura de un controlador PID tipo C, este tipo de estructura en tiempo discreto puede equivaler a tener dos controladores en el sistema.

**Figura 9**

*Controlador PID 2DOF digital*



En la figura 9 la abreviación *uff* representa al seguimiento a referencia, mientras que *ufb* representa el rechazo a las perturbaciones del sistema. Por lo que se puede representar a la ley de control de este controlador de manera digital como se muestra en las ecuaciones a continuación.

$$u(k) = u(k-1) - k_p [y(k) - y(k-1)] + k_i T e(k) - \frac{k_d}{T} [y(k) - 2y(k-1) + y(k-2)] \quad (1.20)$$

$$e(k) = sp - y(k) \quad (1.20.1)$$

$$u(k) = u(k-1) - s_0 y(k) - s_1 y(k-1) - s_2 y(k-2) + k_i T sp \quad (1.20.2)$$

$$k_i T = s_0 + s_1 + s_2 = kr \quad (1.20.3)$$

$$u(z) = \left[ \frac{kr}{1-z^{-1}} \right] sp - \left[ \frac{s_0 + s_1 z^{-1} + s_2 z^{-2}}{1-z^{-1}} \right] y(z) \quad (1.20.4)$$

Realizando un análisis respectivo sobre dichas funciones, se puede concluir que dos de las tres funciones de transferencia en lazo cerrado son independientes. Consiguientemente, el controlador dos grados de libertad permite tomar control sobre el error a seguimiento de referencia y el rechazo a perturbaciones. El diseño de un controlador PID dos grados de libertad, también se los conoce como PID en estructura RST cuando se ha pasado de modelo continuo a discreto.

### **Controlador en espacio de estados**

Un controlador en espacio de estados forma parte de la teoría moderna de control y permite trabajar sobre sistemas dinámicos variantes o invariantes en el tiempo con múltiples entradas y salidas. Es importante definir cada concepto que interviene en la teoría del espacio de estados, tal como se describe a continuación.

#### **Estado.**

Se puede definir el concepto de estado de un sistema según (Burns, 2001) como el conjunto de variables conocidas también como variables de estado que dan a conocer el sistema desde su conjunto

de variables más pequeñas en un tiempo  $t = t_0$  que determinan el comportamiento del sistema en todo  $t > t_0$ . La representación matemática del estado de un sistema se puede simbolizar como:

$$x(t) = \Psi(t, t_0, x(t_0), u(\tau)), \quad t_0 \leq \tau \leq t \quad (1.21)$$

### **Variables de estado.**

Citando a (Calderón, 2008) las variables de estado pueden definirse como aquellas variables que muestran la mínima cantidad de información extra del comportamiento del sistema, esto representa información de comportamientos pasados sobre la dinámica del mismo. Esto permite predecir los comportamientos futuros del sistema ante cualquier señal de entrada.

### **Vector de estado.**

Las variables de estado representan el conjunto de variables de estado necesarias para representar de forma completa el comportamiento del sistema. Cuando se tiene una señal de entrada al sistema se determina el estado del mismo para cualquier tiempo (Calderón, 2008). La representación de un vector de estado es como se muestra a continuación:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

### **Espacio de estados.**

Representa un espacio vectorial n dimensional donde los ejes del espacio son n variables de estado y se representa como un punto en el espacio de estado (Ogata, 2010)

### **Ecuaciones del modelo de estado.**

Las ecuaciones diferenciales del vector de estado tales como:



$$\frac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_{11}u_1 + \cdots + b_{1m}u_m$$

$$\frac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_{21}u_1 + \cdots + b_{2m}u_m$$

$$\frac{dx_n}{dt} = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_{n1}u_1 + \cdots + b_{nm}u_m$$

Pueden expresarse también de forma matricial como:

$$\dot{x} = Ax + Bu \quad (1.22)$$

Donde  $x$  representa el vector de estados,  $u$  el vector de entradas,  $A$  la matriz del sistema y  $B$  la matriz de control, siendo esta la representación de la ecuación de estados.

La salida del sistema también puede representarse como:

$$y = Cx + Du \quad (1.23)$$

Donde  $y$  representa el vector de salida,  $C$  la matriz de salida y  $D$  la ganancia directa de entrada/salida que usualmente suele ser nula.

#### **Solución del vector de estados.**

Partiendo desde la ecuación 1.22, si se toma la transformada de Laplace en el vector de estado:

$$s x(s) - x(0) = A x(s) + B u(s)$$

$$s x(s) - A x(s) = x(0) + B u(s)$$

$$(sI - A) x(s) = x(0) + B u(s)$$

Multiplicando por  $(sI - A)^{-1}$

$$x(s) = (sI - A)^{-1} x(0) + (sI - A)^{-1} B u(s)$$

Aplicando la transformada inversa de la Laplace

$$x(t) = \mathcal{L}^{-1}\{x(s)\}$$

$$x(t) = \mathcal{L}^{-1}\{(sI - A)^{-1}\} x(0) + \mathcal{L}^{-1}\{(sI - A)^{-1} B u(s)\}$$

De donde la matriz exponencial  $e^{At}$  de la ecuación obtenida tras la inversa de la Laplace, se la conoce también como matriz de transición de estado  $\Phi(t)$  la cual representa la respuesta natural del sistema, entonces se tiene el vector resolvente como:

$$\Phi(t) = \mathcal{L}^{-1}\{(sI - A)^{-1}\} = e^{At} \quad (1.24)$$

Entonces se tiene:

$$x(t) = \Phi(t) x(0) + \int_0^t \Phi(t) B u(t - \tau) d\tau \quad (1.25)$$

Donde el primer término hace referencia a la respuesta de las condiciones iniciales y el término de integración es la respuesta a la señal de entrada. La ecuación 1.25 se puede representar también de la siguiente forma:

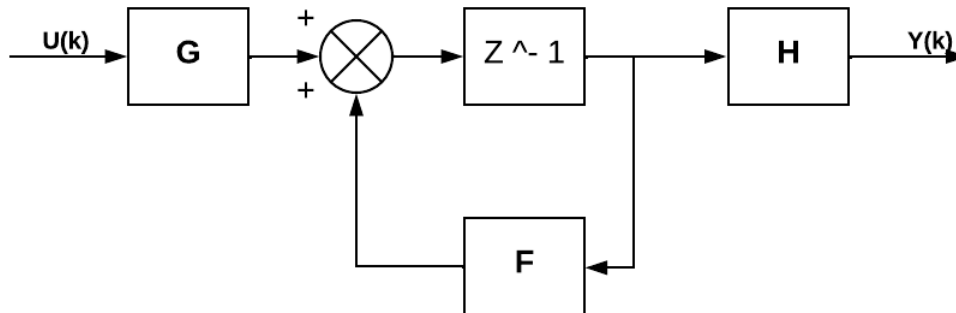
$$x(k+1) = \Phi(t) x(k) + \left[ I \int_0^t \Phi(t) B d\tau \right] u(k) \quad (1.26)$$

$$x(k+1) = F x(k) + G u(k) \quad (1.26.1)$$

La representación en diagrama de bloques se muestra a continuación en la figura 10.

Figura 10

Diagrama de bloques modelamiento de tiempo discreto en espacio de estados



De digitalizar la ecuación 1.26.1 se puede obtener que:

$$z x(z) = F x(z) + G u(z) \quad (1.27)$$

$$x(z) = (zI - F)^{-1} G u(z) \quad (1.27.1)$$

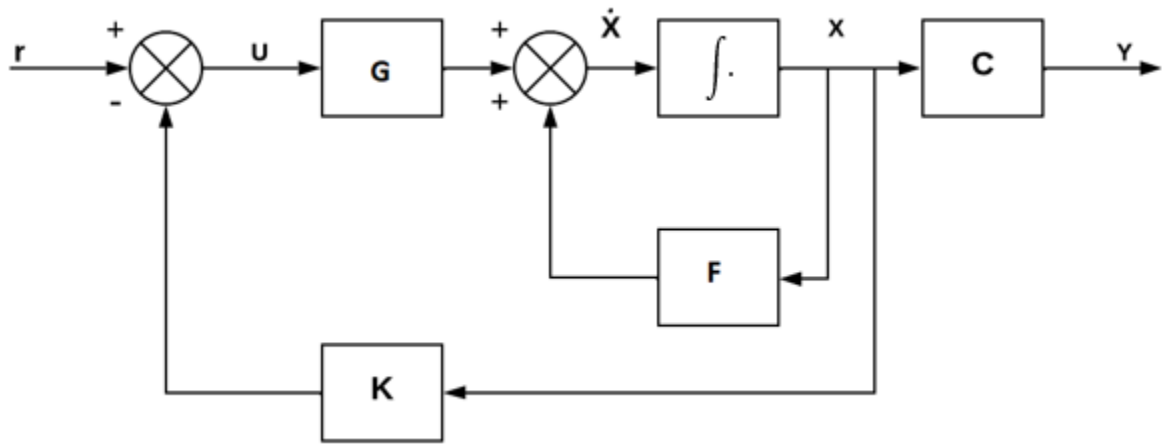
$$y(z) = H (zI - F)^{-1} G u(z) \quad (1.27.2)$$

#### Realimentación de estados.

El objetivo de la realimentación de estados es regular la salida del sistema de manera que se obtenga un seguimiento a referencia a pesar de existir perturbaciones (Villota, 2009). En la figura 11 se puede observar el diagrama de bloques correspondiente a un sistema de control con realimentación de estado, donde  $\mathbf{K}$  representa la matriz de realimentación, misma que debe ser diseñada para cumplir con las condiciones de diseño. En este tipo de esquemas se conjetura que en todo momento se conoce el valor de todos los estados de la planta, determinando la señal de control como una combinación lineal de los estados (Calderón, 2008).

Figura 11

Diagrama de bloques de un sistema de control con realimentación de estado



El diseño de un controlador por espacio de estados para un sistema térmico con retardo se representa en la ecuación 1.1 en la cual se detalla la función de transferencia para sistemas en tiempo continuo. Se representa al sistema en espacio de estados como se muestra a continuación en la ecuación 1.28

$$y(n+1) = \varphi y(n) + b_1 u(n) + b_2 u(n-1) \quad (1.28)$$

Obteniéndose así los siguientes estados:

$$x_1(n) = y(n)$$

$$x_2(n) = u(n-1)$$

$$x_2(n+1) = u(n)$$

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} \varphi & b_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} b_1 \\ 1 \end{bmatrix} u(n) \quad (1.29)$$

Donde:

$$F = \begin{bmatrix} \varphi & b_2 \\ 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} b_1 \\ 1 \end{bmatrix}$$

### **Acción integral por retroalimentación de estados.**

Para el diseño de un controlador por espacio de estados con acción integral, se parte de las ecuaciones 1.28 y 1.29, las cuales para agregar la acción integral al controlador es necesario obtener las matrices aumentadas del sistema. Estas matrices aumentadas se representan a continuación en las ecuaciones 1.30 y 1.31

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} F & \mathbf{0} \\ C & I \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} G \\ \mathbf{1} \end{bmatrix} u(n) \quad (1.30)$$

$$y(k) = [C \quad \mathbf{0}] \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} \quad (1.31)$$

### **Controlador autoajustable difuso**

Los controladores difusos pertenecen al grupo de controladores inteligentes. Al hablar de inteligencia en los controladores, hace referencia a términos tales como razonar, aprender, percepción, resolución de problemas e inteligencia lingüística. Haciendo hincapié sobre la definición de inteligencia se puede decir que consiste en las distintas habilidades para aprender de la experiencia, adaptarse a nuevas situaciones, entender conceptos abstractos y utilizar el conocimiento para poder manipular un entorno en base a lo aprendido.

Tras haber definido el concepto de inteligencia, se puede entonces hablar sobre inteligencia artificial. La inteligencia artificial representa la habilidad que posee un computador para realizar tareas que comúnmente se asocian al concepto de inteligencia. La IA es una de las resultantes de la revolución de la información tecnológica (IT). Por lo tanto, un sistema de control inteligente combina las técnicas de la ingeniería de control con la inteligencia artificial para diseñar sistemas autónomos que puedan sensor, razonar, planear, aprender y actuar de una forma inteligente (Burns, 2001).

El concepto y la metodología de la lógica difusa fue propuesta en el año 1965 por el matemático Lotfi A. Zadeh. En los años posteriores se siguió investigando y desarrollando la metodología tanto en Europa como en países asiáticos tales como Japón. En 1975 el científico computacional Ebrahim

Mamdani realizó también valiosos aportes al desarrollo de la lógica difusa. Siendo así que para el año 2000 la metodología de la lógica difusa se convierte en una técnica estándar para el control multivariable aplicado en datos y el análisis de señales adquirida en sensores.

El comportamiento dinámico de un sistema difuso se caracteriza por la descripción de reglas lingüísticas basadas en conocimientos o sistemas expertos, donde este conocimiento experto es de la forma **IF** (si una serie de condiciones son satisfactorias) **THEN** (una serie de consecuencias pueden inferirse) esto se conoce como sentencias condicionales difusas (Chennakesava, 2007). La teoría de conjuntos es una base fundamental de la teoría difusa ya que distingue los elementos que forman parte de un conjunto y aquellos que no, presentando nitidez sobre los límites establecidos. Por lo tanto, para un conjunto difuso es fundamental la función de membresía la cual toma valores entre 0 y 1. Esta característica se puede desarrollar como se muestra en la ecuación 1.32 y 1.33.

$$\mathbf{u}_A(x) = \begin{cases} \mathbf{1}, & x \in A \\ \mathbf{0}, & x \notin A \end{cases} \quad (1.32)$$

$$\mathbf{u}_A(x) = \begin{cases} \mathbf{1}, & x \in A \\ (\mathbf{0}, \mathbf{1}), & x \in A \textit{ parcialmente} \\ \mathbf{0}, & x \notin A \end{cases} \quad (1.33)$$

En las ecuaciones 1.32 y 1.33,  $A$  representa un conjunto difuso y tiene un rango existente entre 0 y 1, siendo así que la función de membresía o también conocida como función de pertenencia es la base de los conjuntos difusos. En la lógica difusa también se cuenta con operaciones básicas de conjuntos tales como la igualdad, la unión, intersección y complemento.

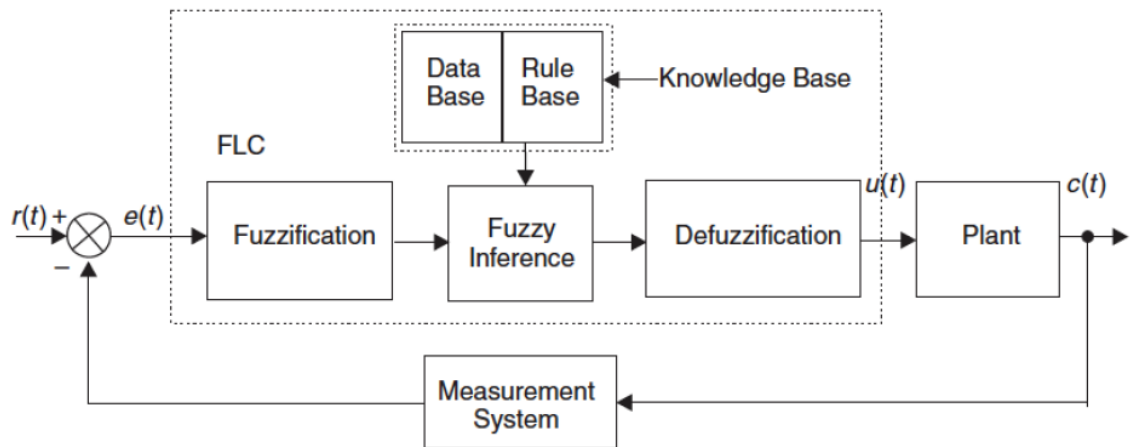
Generalmente un controlador difuso requiere de tres procesos: fusificación, inferencia difusa y la defusificación. En el proceso de fusificación se mapean las entradas respecto a valores de pertenencia de los conjuntos difusos en diversos universos de discurso. El proceso de inferencia difusa se encarga de obtener un vector de salidas difusas basado en un conjunto de reglas. Finalmente, en la defusificación se

mapea las señales de control difusas inferidas sobre una ventana de salida a una señal de control no difusa para ser utilizada de manera práctica (Acosta, 2022).

En la figura 12 a continuación, se muestran los procesos fundamentales que se llevan a cabo en un controlador de lógica difusa.

**Figura 12**

*Procesos del controlador Fuzzy Logic*

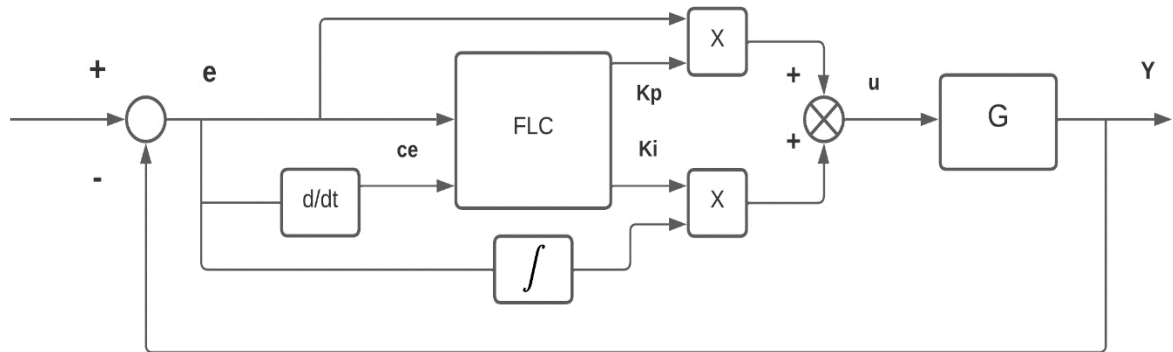


*Nota.* Tomado de *Advanced Control Engineering*, por Burns, 2001.

El diseño para llevar a cabo un controlador PI fuzzy autoajutable se lo puede observar en la figura 13 donde se tiene el diagrama de bloques correspondiente a un controlador PI difuso autoajutable. Se tiene el bloque principal donde se encuentra el sistema de inferencia generado tras la fusificación realizada, se tiene dos entradas y dos salidas de este bloque las cuales como entradas son el error y la derivada del error y salidas los valores de  $K_p$  y  $K_i$ . La construcción de la parte proporcional del controlador  $K_p$  la cual es multiplicada por el error y la parte integral  $K_i$  multiplicada por la integral del error. Ambos valores del controlador  $K_p$  y  $K_i$  son sumados y van directo como señal de control hacia la planta.

Figura 13

Estructura de un controlador difuso autoajustable



La ley de control correspondiente un controlador digital PI difuso autoajustable se muestra en la ecuación 1.34, donde se tiene  $T$  como el período de muestreo y  $e(k)$  el error cuando  $K_p, K_i > 0$ , entonces el error  $e(k)$  se representa en la ecuación 1.35 donde  $y_d(k)$  es el valor deseado que se desea alcanzar.

$$u(k) = k_p e(k) + k_i T \sum_{j=0}^k e(j) \quad (1.34)$$

$$e(k) = y_d(k) - y(k) \quad (1.35)$$

Para sintonizar el valor de  $K_p$  las reglas del controlador difuso son representadas como se describe a continuación. Cuando el error  $e(k)$  es positivo entonces  $e$  es P (positivo) para lo que se debería incrementar el valor de  $K_p$  y  $\Delta K_p$  debe ser positivo también. Cuando  $e(k)$  es negativo aparece un sobre impulso en la respuesta, entonces  $e$  es N, por lo que se debe bajar los valores de  $K_p$  y  $\Delta K_p$  debe ser negativo. Cuando el error es cercano a cero  $e$  es Z se tienen tres condiciones: si  $e_c$  es N el sobre impulso tiende a ser mayor entonces  $\Delta K_p$  es negativo; si  $e_c$  es Z entonces  $\Delta K_p$  debe ser positivo para bajar el error estático; si  $e_c$  es P entonces  $\Delta K_p$  debe ser positivo también ya que el error tiende a ser mayor. Estas reglas pueden resumirse como se muestra a continuación en la tabla 3.



**Tabla 3**

*Reglas del controlador difuso para sintonización de  $K_p$*

$e \setminus e_c$	<b>N</b>	<b>Z</b>	<b>P</b>
<b>N</b>	N	N	N
<b>Z</b>	N	P	P
<b>P</b>	P	P	P

De la misma manera se presentan las reglas de sintonización para el valor  $k_i$  del controlador difuso tomando en cuenta que cuando el error es demasiado pequeño se elige un valor de  $\Delta k_i$  grande, de lo contrario se elige un valor de  $\Delta k_i$  muy pequeño. Estas reglas de sintonización se presentan a continuación en la tabla 4.

**Tabla 4**

*Reglas del controlador difuso para sintonización de  $K_i$*

$e \setminus e_c$	<b>N</b>	<b>Z</b>	<b>P</b>
<b>N</b>	Z	Z	Z
<b>Z</b>	P	P	P
<b>P</b>	Z	Z	Z

Para el proceso de defusificación la técnica más utilizada es la del centroide, mismo que puede ser expresado como la división entre la suma de los primeros momentos del área y la suma de las áreas, obteniéndose así el valor crisp de la señal de control que puede ser utilizada. Esta técnica puede ser expresada para un sistema discreto como se muestra en la ecuación 1.36.

$$\mathbf{u}(kT) = \frac{\sum_{i=1}^n x_i \mathbf{u}(x_i)}{\sum_{i=1}^n \mathbf{u}(x_i)} \quad (1.36)$$

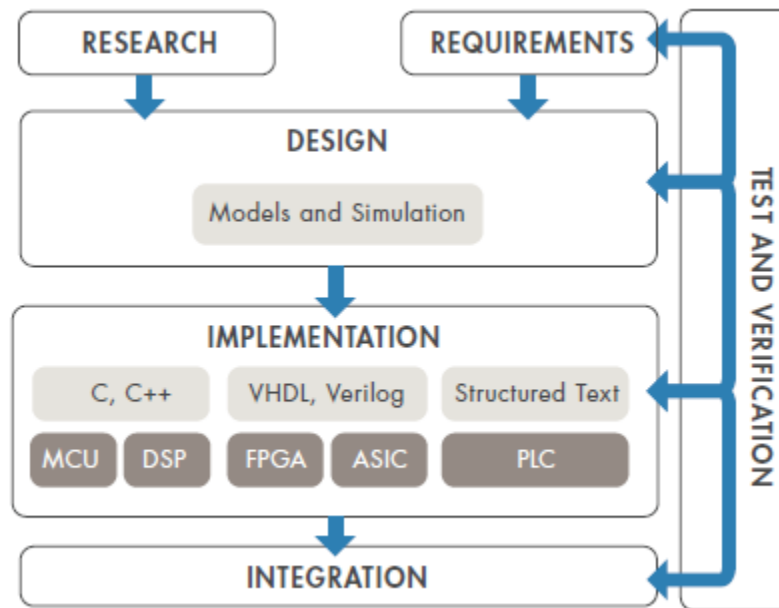
### **Generación de código para controladores mediante la herramienta Matlab/Simulink**

La herramienta de cómputo numérico Matlab ofrece y propone el diseño basado en modelos para sistemas de control embebidos. De esta forma se puede diseñar un sistema de control desde la misma identificación del sistema, el diseño de los algoritmos de control, simulación del diseño y generación de código para su posterior implementación y pruebas en el sistema real (Matlab, 2022). Por esta razón la generación de código que ofrece Matlab a través de Simulink es útil en la ingeniería de sistemas de control, ofertada en las últimas versiones del software.

Los modelos diseñados deben adquirir toda la información necesaria del entorno en el cual se desarrolla, para generar el código forma automática a partir del algoritmo de control diseñado. Se pueden realizar pruebas y verificaciones mediante software. Una vez generado el código se procede a realizar la metodología de hardware-in-the-loop para la implementación del diseño en el sistema físico (The MathWorks, Inc, 2020). En la figura 14 se presenta un diagrama de bloques respecto al diseño basado en modelos que parte de la investigación y los requerimientos del modelo para realizar el diseño y simulación correspondiente en software, para consiguiente realizar la implementación del sistema generando código del mismo que puede ser en lenguaje C, C++, VHDL, texto estructurado, etc. Estos distintos lenguajes pueden servir para microcontroladores, FPGA, PLC. Estos pasos en todo momento tienen una realimentación mediante pruebas y verificación de los requerimientos del modelo.

**Figura 14**

*Diagrama de bloques para un diseño basado en modelos*



*Nota.* Tomado de *Model Based Design for Embedded Control Systems*, por MathWorks, 2022.

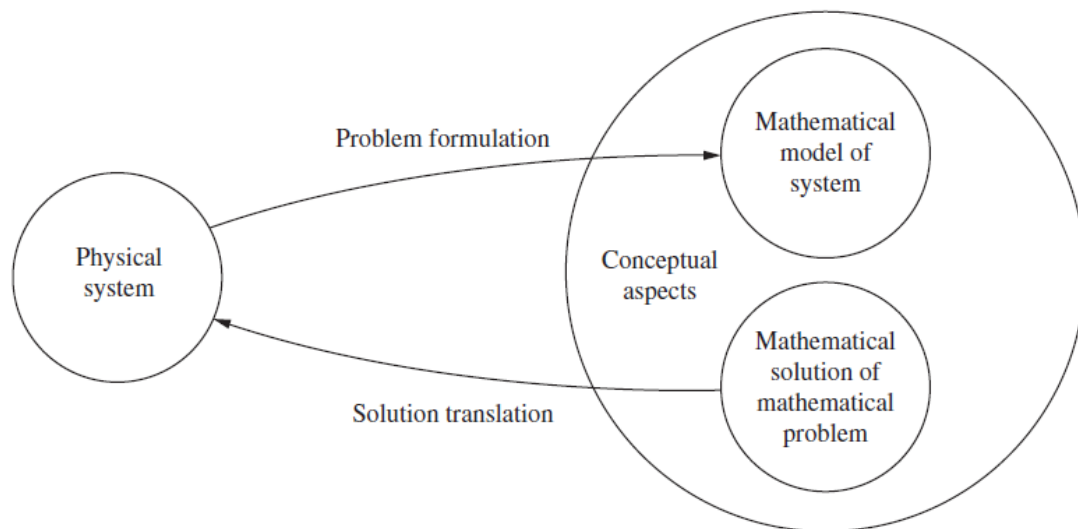
Tradicionalmente la generación de código del diseño de un sistema se lo realizaba de manera manual, los ingenieros de sistemas solían escribir el código en los controladores bajo las instrucciones de los ingenieros de control, quienes diseñaron los requerimientos del sistema. Cada paso del proceso debía ser codificado en el controlador de manera manual empezando por las especificaciones, el algoritmo de control y la depuración del código. Esto provoca errores y más tiempo para finalizar e implementar el diseño de un sistema.

La generación de código se vuelve un aspecto fundamental dentro del presente trabajo de investigación debido a que forma parte del flujo de trabajo que se lleva a cabo como parte del problema de control. Como mencionan los autores (Phillips, Nagle, & Chakraborty, 2015) parte del problema de control tanto en diseño como en análisis, es el poder garantizar que el diseño matemático se pase de forma eficiente hacia el sistema físico real y funcione de la misma forma que funciona en simulación. Para esto es necesario la generación eficiente de código del diseño propuesto. En la figura 15 se puede

observar de manera abstracta el objetivo al cual se pretende apuntar dentro del presente trabajo de investigación.

**Figura 15**

*Relación del análisis matemático y los procedimientos de diseño para sistemas físicos reales*



*Nota.* Tomado de *Digital Control System Analysis and Design*, por Phillips Charles, 2015.

### **PLC de la marca TURCK**

El Gateway BL20 de la Turck es un sistema modular con entradas/salidas digitales o análogas utilizado ampliamente en la automatización industrial. Este sistema cuenta con un Gateway que es el que permite la conexión hacia los diferentes módulos de entrada/salida, es el responsable del manejo de todos los datos del proceso y de generar cualquier información de diagnóstico al sistema en general (Turck, 2013). Por lo tanto, este sistema BL20 puede ser utilizado como un PLC autónomo para hacer tareas de control, comunicación y procesamiento de señales. Utiliza una interfaz de bus de campo para PROFINET, Modbus TCP, EtherNet IP o PROFIBUS-DP también soporta estándares como Ethernet y comunicación serial (Turck, 2022). En la figura 16 se puede observar una imagen referencial del dispositivo BL20 (IP20) de Turck.

**Figura 16**

*PLC BL20 (IP20) marca Turck*



### ***Plataforma de software CODESYS***

El entorno de desarrollo para la programación de controladores CODESYS se basa en el estándar industrial internacional IEC 61131-3. Este entorno permite la programación en (ST) texto estructurado, sin embargo, también ofrece otros lenguajes tales como Instruction List (IL), Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC) y Continuous Function Chart edition (CFC). En Europa este entorno de programación para controladores lógicos es muy demandado para tareas de automatización (3S-Smart Software Solutions GmbH, 2010).

### Capítulo III. Diseño de estrategias de control avanzado

En este capítulo se abarca el diseño de las estrategias de control avanzado como el PI 2DOF, SSIA y PI AFuzzy, la identificación del sistema térmico mediante el método gráfico de estimación por respuesta al escalón, función ARX y la validación del mismo a través de la función procest P1D de la herramienta Matlab.

#### Identificación de la planta de temperatura.

Para la identificación de la planta de temperatura, se programó en la plataforma CODESYS una rutina computacional para poder generar una entrada de tipo escalón al sistema. Esta rutina se detalla en el diagrama de flujo de la figura 17. Por lo tanto, se tiene la generación de una señal de entrada tipo escalón al 50%. Los datos obtenidos fueron normalizados, los valores de salida respecto al valor máximo de potencia entregada por el actuador y la temperatura respecto al valor máximo que puede alcanzar el sistema. Adicional se eliminó el offset de la temperatura ambiente para manejar rangos de operación normalizados y referenciados a cero. A continuación, se presentan las tablas 5 y 6 correspondientes a la definición de variables y constantes, utilizadas para desarrollar tanto el código y el diagrama de flujo mencionados anteriormente.

**Tabla 5**

*Definición de variables para generar entrada escalón al sistema*

<b>Variables</b>	<b>Definición</b>
<b>OUT</b>	Salida codificada de amplitud normalizada
<b>OUTC</b>	Salida decodificada de 15 bits
<b>COUNTER</b>	Contador de muestras digitales
<b>MI</b>	Cantidad de muestras iniciales
<b>SETPOINT</b>	Setpoint normalizado
<b>TEMP</b>	Temperatura normalizada adquirida por el sensor PT100

<b>ANALOG_I</b>	Entrada analógica de temperatura de 15 bits
<b>TEMPMIN</b>	Temperatura ambiente

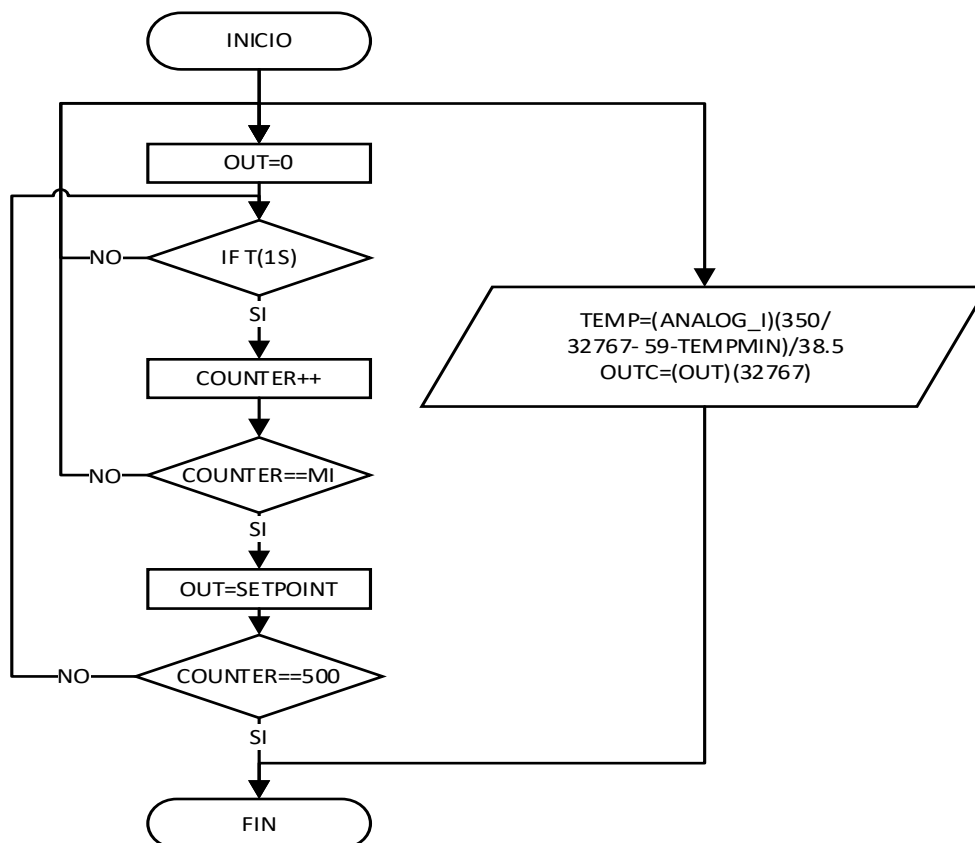
**Tabla 6**

*Definición de constantes para generar entrada escalón al sistema*

<b>Constantes</b>	<b>Definición</b>
<b>350</b>	Transmisor de temperatura de 0 a 350 °C.
<b>32767</b>	Valor máximo del conversor análogo digital $2^{15} = 32767$
<b>500</b>	Valor máximo de muestras del registro del PLC
<b>T = 1s</b>	Valor del timer que muestrea cada segundo
<b>38</b>	Valor máximo de Temperatura en °C

Figura 17

Diagrama de flujo para generar entrada escalón al sistema

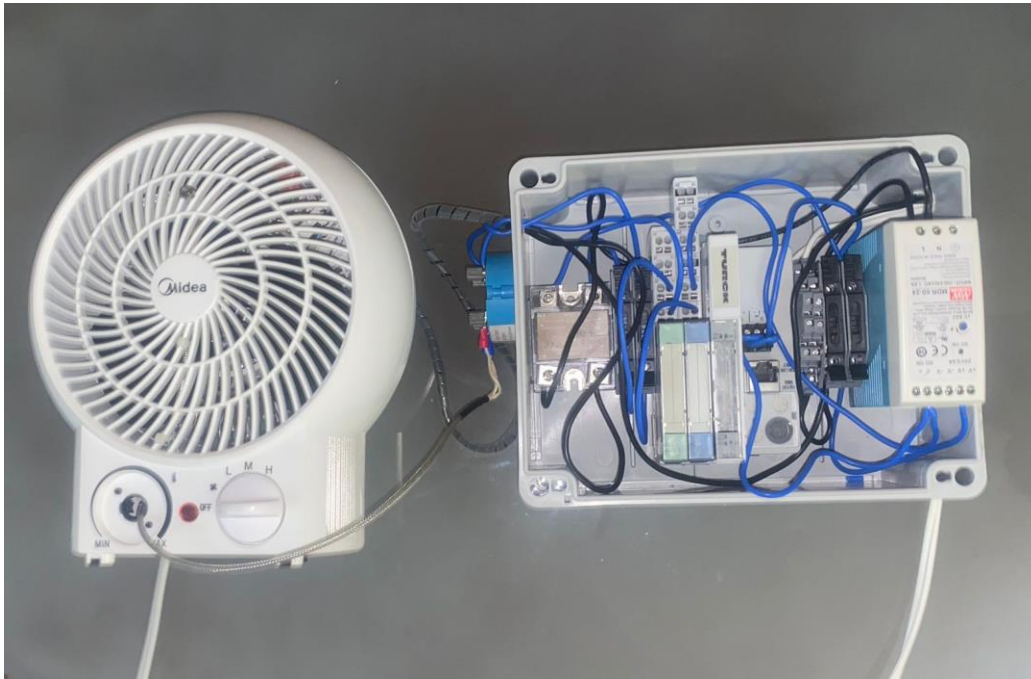


El sistema físico propuesto se detalla en la figura 18 donde puede observarse al lado izquierdo se encuentra un ventilador que forma parte de la planta de temperatura en conjunto con el actuador de la misma. El actuador del sistema es una resistencia térmica que se calienta en función de la potencia administrada. En la parte derecha se puede observar el controlador del sistema el cual es un PLC de la marca TURCK mismo que se encuentra con su respectivo módulo de entrada y salida analógica, también se tiene un interruptor de corriente tipo triac controlado por una señal de 4-20 mA y el transmisor de 4-20 mA correspondiente al sensor de temperatura PT100. Los planos del sistema se encuentran detallados en la sección de apéndices del presente documento.



**Figura 18**

*Sistema térmico implementado físicamente*

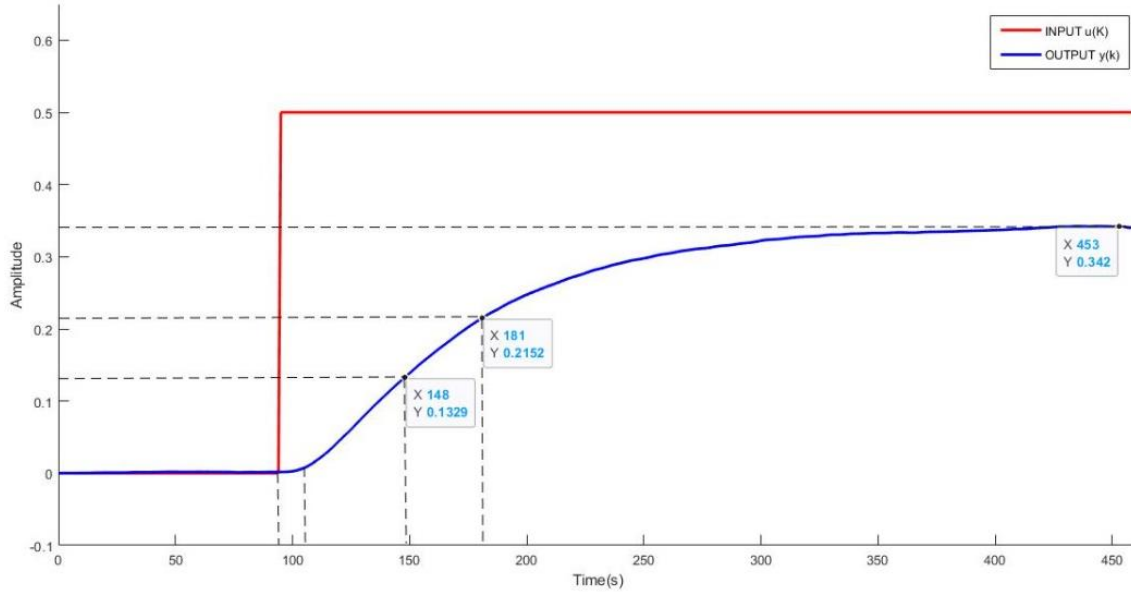


***Método gráfico de estimación por respuesta al escalón.***

Los datos muestreados fueron exportados a un archivo txt, los cuales fueron importados al sistema de cómputo numérico Matlab para obtener la gráfica de estos datos y realizar su posterior análisis. De esta manera mediante el método gráfico de estimación por respuesta al escalón se obtuvieron los siguientes parámetros detallados en la figura 19.

Figura 19

Método gráfico de estimación por respuesta al escalón



De donde se obtienen los siguientes parámetros de identificación del comportamiento del sistema partiendo de las ecuaciones 1.2-1.5 descritas en el marco conceptual anteriormente expuesto:

$$\tau = 2(t_{63\%} - t_{39\%})$$

$$\tau = 2(181 - 148) = 66$$

$$a = \frac{1}{\tau} = \frac{1}{66} = 0.015$$

$$k = \frac{\lim_{t \rightarrow \infty} y(t) - T_i}{A} * a$$

$$k = \frac{0.3429 - 0}{1} * 0.015 = 0.0051954$$

$$t_d = t_{63\%} - (T_i + \tau) = 21$$

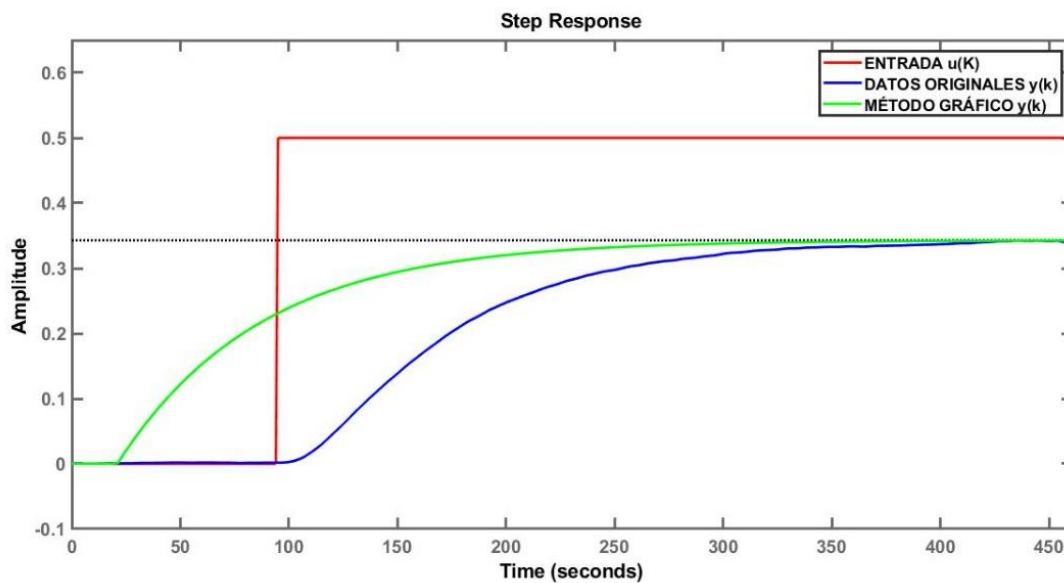
Obteniendo así la siguiente función de transferencia para el sistema térmico mediante el método de estimación por respuesta al escalón representada en la ecuación 2.1

$$G(s) = \frac{0.3429}{1 + 66s} * e^{-21s} \quad (2.1)$$

Con el resultado obtenido se concluyó que el método gráfico de respuesta al escalón no se ajusta a la respuesta esperada por la planta como se muestra en la figura 20, en la cual se observa que no cumple con las especificaciones de tiempo muerto, tiempo de retardo ni tiempo de estabilización respecto a los datos originales.

**Figura 20**

*Respuesta al escalón del sistema, datos originales de la planta y método de identificación gráfico*



### **Función ARX. IDENT (System Identification toolbox)-Matlab**

Debido a las limitaciones del método gráfico, se realizó la identificación de la planta a través de métodos computacionales que ofrecen una mejor estimación del sistema. Haciendo uso de la herramienta propia de Matlab de identificación de sistemas IDENT se obtiene la identificación del sistema tanto en estimación como validación del mismo.

A través de la estimación del tiempo muerto con la herramienta *delayest* se pudo determinar que el valor del tiempo muerto del sistema a identificar es de 4 segundos, el mismo que fue utilizado para la estimación del sistema a través de la función ARX de la herramienta Matlab. Obteniéndose así la función de transferencia representada en la ecuación 2.2, de esta forma se obtiene la función de transferencia correspondiente y una estimación del 93.42% sobre la raíz cuadrada media normalizada (NRMSE) del sistema.

$$G(s) = \frac{0.7080}{1 + 89.126s} * e^{-4s} \quad (2.2)$$

### **Función PROCEST P1D.**

Mediante la función PROCEST P1D de Matlab, se definió los datos previamente normalizados que fueron tomados experimentalmente, mismos que fueron tomados desde el primer método desarrollado. Se realizó la identificación de dichos datos y la estructura del sistema como P1D, es decir, un sistema con un polo y retardo. Obteniéndose así la función de transferencia validada del sistema, como se muestra en la ecuación 2.3.

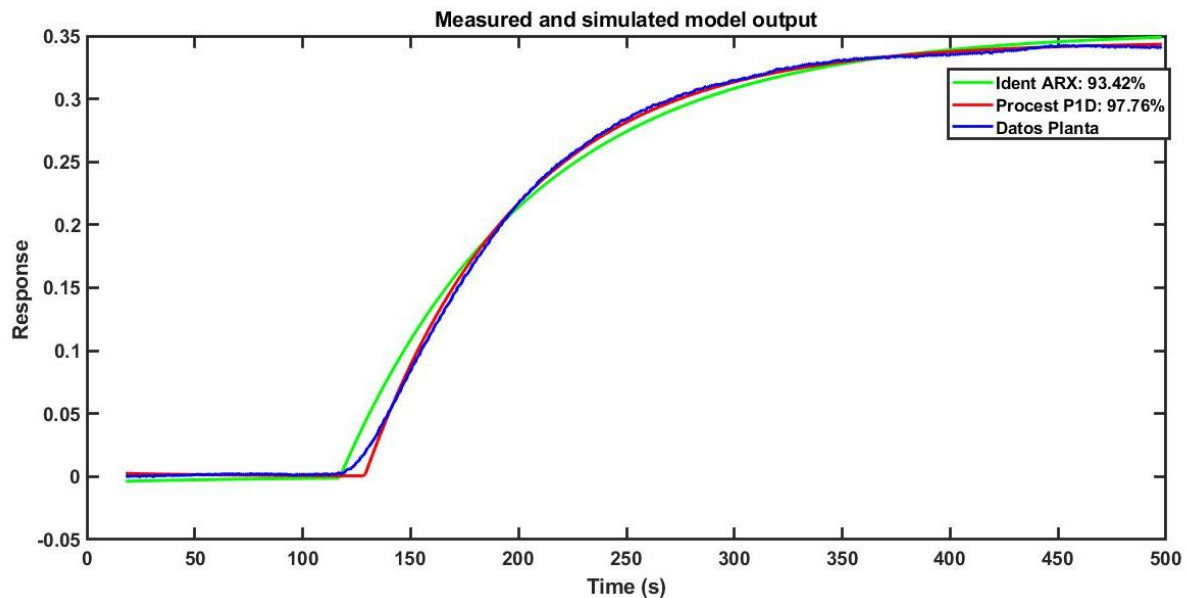
$$G(s) = \frac{0.69087}{1 + 71.982s} * e^{-15.727s} \quad (2.3)$$

Por lo tanto, se realiza una comparativa de las dos últimas funciones utilizadas para definir la mejor estimación y validación del sistema. En la figura 21 se muestra la estimación obtenida en la herramienta de identificación de sistemas IDENT mediante la función ARX y la validación del mismo a

través de la función PROCEST P1D. En donde la concordancia de los datos originales respecto a los obtenidos a través del sistema de identificación es de 97.76% para la función de identificación PROCEST P1D y del 93.42% de la función de identificación ARX. Por lo tanto, para el diseño de los controladores se utilizó la planta obtenida por la función PROCEST P1D.

**Figura 21**

*Respuesta al escalón del sistema función Procest P1D y función ARX*



### Diseño del Controlador PI 2DOF

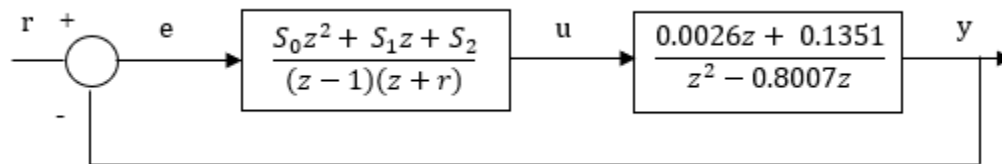
Para el diseño del controlador PI 2DOF es necesario partir de un controlador PID como base, para lo cual se utilizó dos métodos, el primero un PID diseñado de forma digital directa y el método de sintonización de Ziegler-Nichols para un modelo de primer orden con tiempo muerto detallado en la tabla 2. Para este método se escogió un controlador PI ya que el tiempo de retardo propio del sistema es mayor a 15 segundos por lo que se considera una planta de respuesta lenta. A continuación, se detalla el diseño de los controladores en base al modelo de la planta de temperatura identificada.

### PID en forma digital directa.

Se desarrolla el diseño de un controlador PID en base a la planta de temperatura previamente identificada, con lo que se obtiene la estructura del controlador del sistema térmico en la figura 22.

**Figura 22**

*Estructura de un controlador PID en forma digital directa aplicado a un sistema térmico con retraso*



Para el sistema térmico obtenido la variable  $a_2$  es 0, por lo tanto, a partir de las ecuaciones 1.18 y 1.18.1 se obtiene:

$$\Delta(z) = \Delta d(z)$$

$$z^4: 1 = 1$$

$$z^3: -0.8007 + (r - 1) + 0.0026S_0 = d_1$$

$$z^2: -r - 0.8007(r - 1) + 0.1351S_0 + 0.0026S_1 = d_2$$

$$z: 0.8007r + 0.1351S_1 + 0.0026S_2 = d_3$$

$$z^0: 0.1351S_2 = d_4$$

$$\begin{bmatrix} 0.0026 & 0 & 0 & 1 \\ 0.1351 & 0.0026 & 0 & -0.8007 \\ 0 & 0.1351 & 0.0026 & 0.8007 \\ 0 & 0 & 0.1351 & 0 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ r \end{bmatrix} = \begin{bmatrix} d_1 + 1.8007 \\ d_2 - 0.8007 \\ d_3 \\ d_4 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.0026 & 0 & 0 & 1 \\ 0.1351 & 0.0026 & 0 & -0.8007 \\ 0 & 0.1351 & 0.0026 & 0.8007 \\ 0 & 0 & 0.1351 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ r \end{bmatrix}$$

$$D = \begin{bmatrix} d_1 + 1.8007 \\ d_2 - 0.8007 \\ d_3 \\ d_4 \end{bmatrix}$$

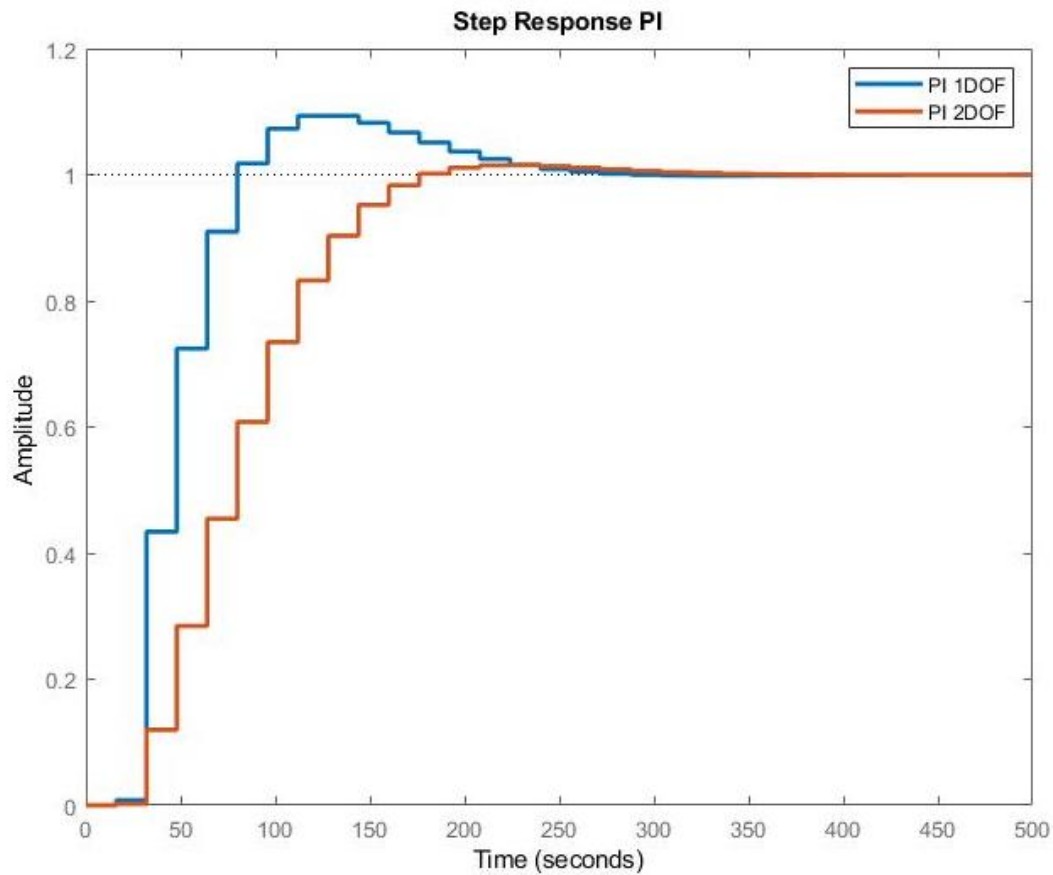
Obteniéndose el controlador PID 1DOF y el PI 2DOF mostrado en la figura 23. La respuesta del sistema cumple con los criterios de diseño esperados, como se puede observar el diseño del PI 2DOF ofrece una mejora sobre el overshoot del sistema presentado en el diseño del PID 1DOF. También se puede observar las funciones de transferencia obtenidas de los controladores diseñados mediante esta metodología.

$$cz1 = \frac{3.111 z^2 - 2.264 z}{z^2 - 0.6181 z - 0.3819} \quad (2.4)$$

$$cz2 = \frac{0.8469 z^2}{z^2 - 0.6181 z - 0.3819} \quad (2.5)$$

Figura 23

Respuesta del sistema ante el controlador PID en forma digital directa



### Sintonización de PID por Ziegler - Nichols

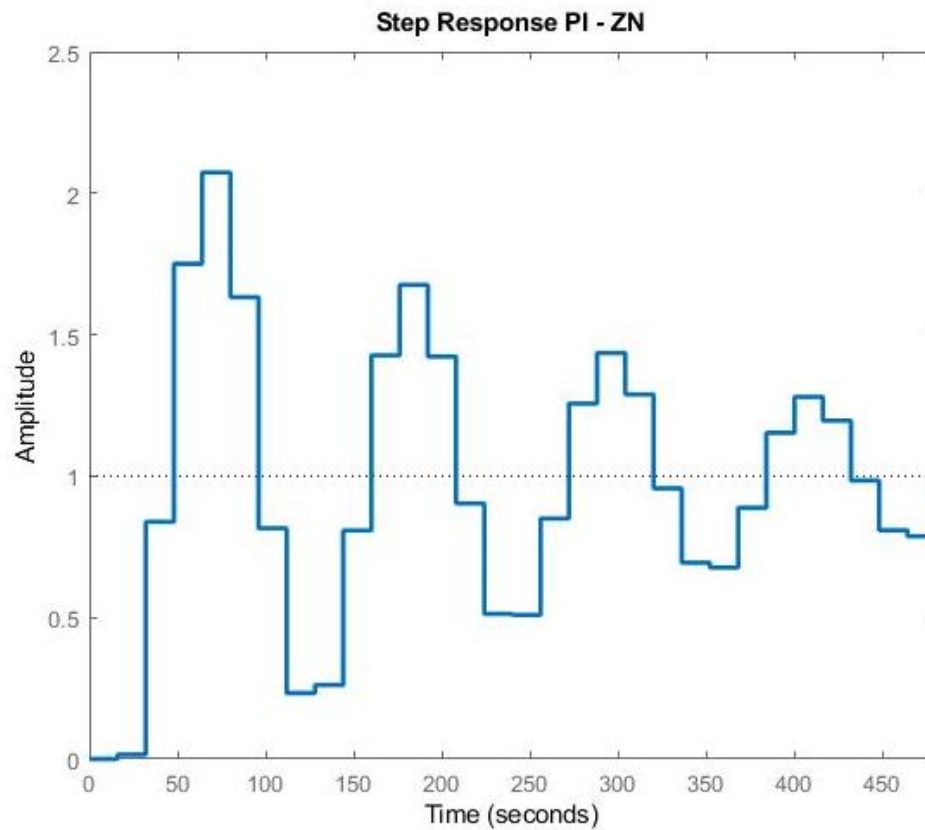
Adicional al diseño digital directo desarrollado en el apartado anterior, se realizó la sintonización de parámetros de un controlador PI mediante las reglas de Ziegler- Nichols. Con lo cual se obtuvo la siguiente función de transferencia en la ecuación 2.6 para el controlador PI base sintonizado. También se muestra en la figura 24 cómo reacciona el sistema con el diseño de este controlador.

$$C(s) = \frac{8.249s + 0.1309}{s} \quad (2.6)$$



**Figura 24**

*Respuesta del sistema ante el controlador PI diseñado por Ziegler-Nichols*



Como se puede observar en la figura 24 la respuesta es sub amortiguada y no cumple con los criterios de diseño respecto al tiempo de estabilización y sobre impulso, por lo tanto, se utilizó una herramienta diferente de sintonización como la descrita en el marco metodológico acerca de PID Tuner Free Web Site.

#### ***Sintonización de controladores mediante la herramienta PID Tuner Free Web Site***

Mediante la herramienta de sintonización PID tuner con una especificación de tiempo de estabilización de 250 segundos sin sobre impulso, se obtuvo la siguiente función de transferencia para el controlador PI 1DOF:

$$C(s) = \frac{1.9368s + 0.02707}{s} \quad (2.7)$$

Donde:

$$\begin{aligned}k_p &= 1.9368 \\k_i &= 71.5371 \\k_d &= 0\end{aligned}$$

La digitalización del controlador PI y de la planta se realizó con una frecuencia de muestreo de 16 segundos con lo que se garantizó que el tiempo de muestreo sea mayor al tiempo de retardo obteniendo así la siguiente planta y controlador digitalizadas. La digitalización se realizó utilizando el método ZOH.

Planta digitalizada:

$$G(z) = \frac{0.02615z + 0.1351}{z^2 - 0.807z} \quad (2.8)$$

Controlador PI base digitalizado:

$$C_1(z) = \frac{1.937z - 1.504}{z - 1} \quad (2.9)$$

Por lo tanto, se continuó con el diseño del controlador PI 2DOF basada en la ecuación 2.10, misma que parte de los fundamentos teóricos desarrollados previamente según la estructura de un controlador PID 2DOF.

$$C_2(z) = \frac{(s0 + s1 + s2)z^2}{z(z - 1)} \quad (2.10)$$

Donde:

$$s0 = k_p + k_i T + \frac{k_d}{T}$$

$$s1 = -k_p - 2 * \frac{k_d}{T}$$

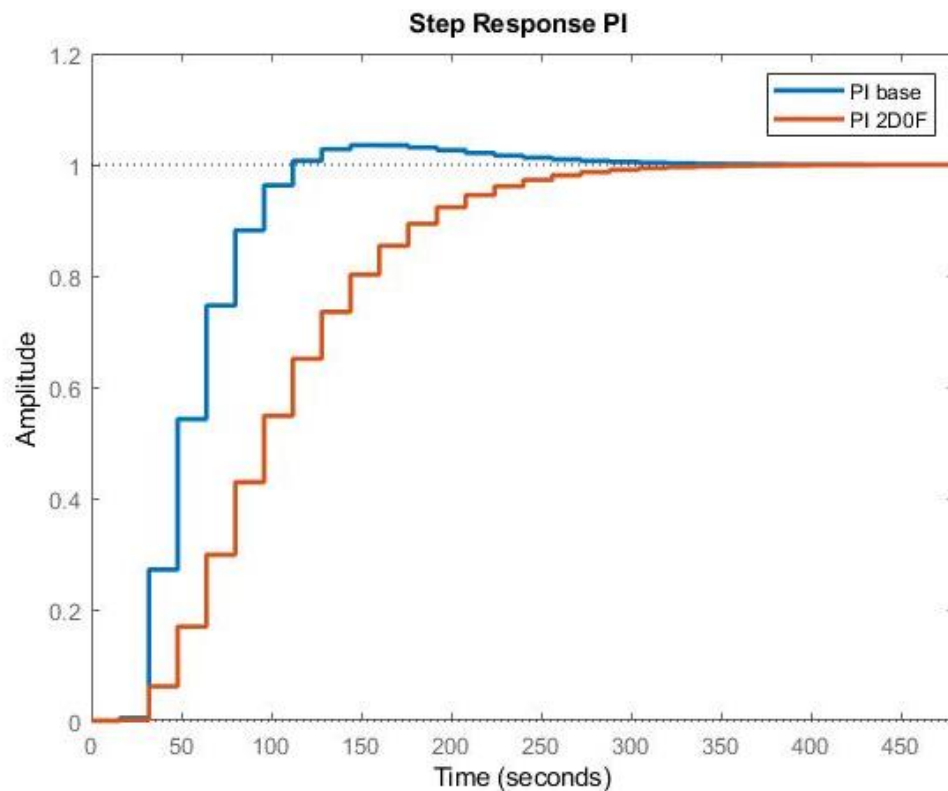
$$s2 = \frac{k_p}{T}$$

$$\therefore C_2(z) = \frac{0.4332z^2}{z^2 - z} \quad (2.10.1)$$

Con lo que se obtiene el diagrama de bloques del sistema con el controlador 2DOF, como se muestra en la figura 9, mismo que cumple con las especificaciones respecto a la estructura de un PID 2DOF o estructura RST. Como se puede observar en la figura 25 la respuesta es críticamente amortiguada y cumple con los criterios de diseño respecto al tiempo de estabilización y sobre impulso.

**Figura 25**

*Respuesta del sistema controlador PI y PI 2DOF*



### **Diseño del Controlador en espacio de estados con acción integral SSIA**

El diseño para el controlador SSIA para un sistema térmico con retardo parte de la ecuación 1.1 misma que representa la función de transferencia para este tipo de sistemas en tiempo continuo. Se

puede representar entonces al sistema en espacio de estados a partir de la ecuación 1.28 previamente desarrollada en el capítulo anterior. Por lo que aplicado a la planta de temperatura identificada se obtiene la ecuación 2.11.

$$\mathbf{y}(n + 1) = \mathbf{0.7574} \mathbf{y}(n) + \mathbf{0.0398} \mathbf{u}(n) + \mathbf{0.1278} \mathbf{u}(n - 1) \quad (2.11)$$

Obteniéndose los estados:

$$x_1(n) = y(n)$$

$$x_2(n) = u(n - 1)$$

$$x_2(n + 1) = u(n)$$

$$\begin{bmatrix} x_1(n + 1) \\ x_2(n + 1) \end{bmatrix} = \begin{bmatrix} -\mathbf{0.7574} & \mathbf{0.1278} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} \mathbf{0.0398} \\ \mathbf{1} \end{bmatrix} \mathbf{u}(n) \quad (2.12)$$

Donde:

$$F = \begin{bmatrix} -\mathbf{0.7574} & \mathbf{0.1278} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$G = \begin{bmatrix} \mathbf{0.0398} \\ \mathbf{1} \end{bmatrix}$$

El controlador actual realiza un seguimiento a referencia, sin embargo, no realiza un buen rechazo a perturbaciones, por lo que se utiliza el diseño de la acción integral. Para esto se debe realizar el diseño anterior mediante matrices aumentadas como se muestra en la ecuación 1.30 y 1.31. Por lo que aplicado al sistema térmico identificado se tendría las ecuaciones 2.13 y 2.14 sobre las matrices aumentadas  $Fa$  y  $Ga$  mostradas a continuación.

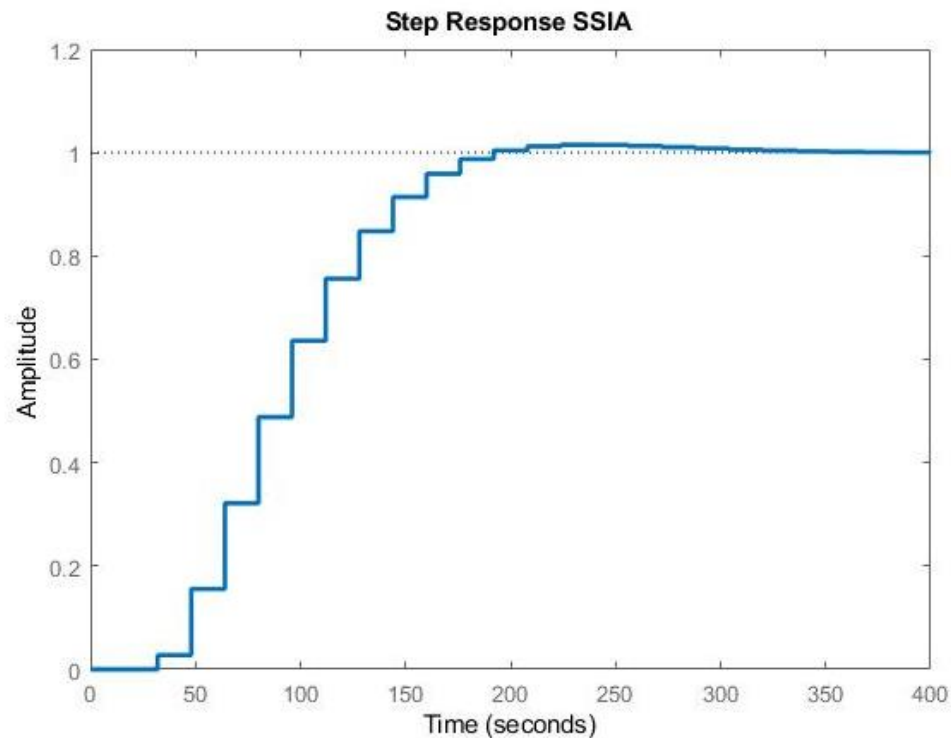
$$Fa = \begin{bmatrix} \mathbf{0.7574} & \mathbf{0.1278} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (2.13)$$

$$Ga = \begin{bmatrix} \mathbf{0.0398} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \quad (2.14)$$

Obteniéndose así la respuesta al sistema mediante el controlador SSIA mostrado en la figura 26. Donde se puede observar que el diseño cumple con los requerimientos planteados, no presenta sobre impulso y tiene un buen seguimiento a referencia al igual que rechazo a las perturbaciones.

**Figura 26**

*Respuesta del Sistema controlador SSIA*



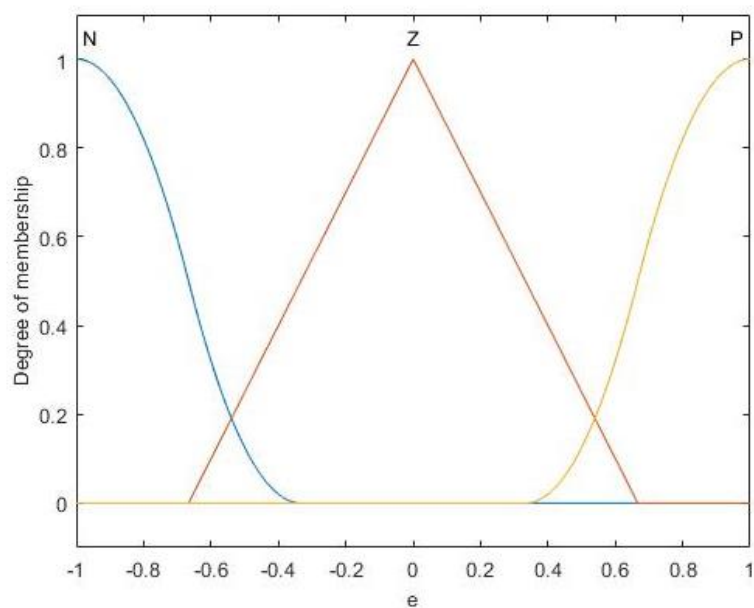
### **Diseño del Controlador autoajustable difuso (AFuzzy)**

Para estimar los valores de  $k_p$  y  $k_i$  del controlador PI, se parte de los valores obtenidos en el diseño del controlador PI 2DOF digital. Mediante la función *pid* de Matlab se obtienen los valores correspondientes a  $k_p$  y  $k_i$  para tener los rangos correspondientes en los que las funciones de membresía del controlador difuso deben trabajar. De igual manera las reglas de sintonización del controlador PI utilizadas se presentaron en el marco metodológico en la tabla 3 y 4. Obteniéndose así las

funciones de membresía correspondientes al error  $e$  y a la derivada del error  $e_c$  presentadas en las figuras 27 y 28 respectivamente.

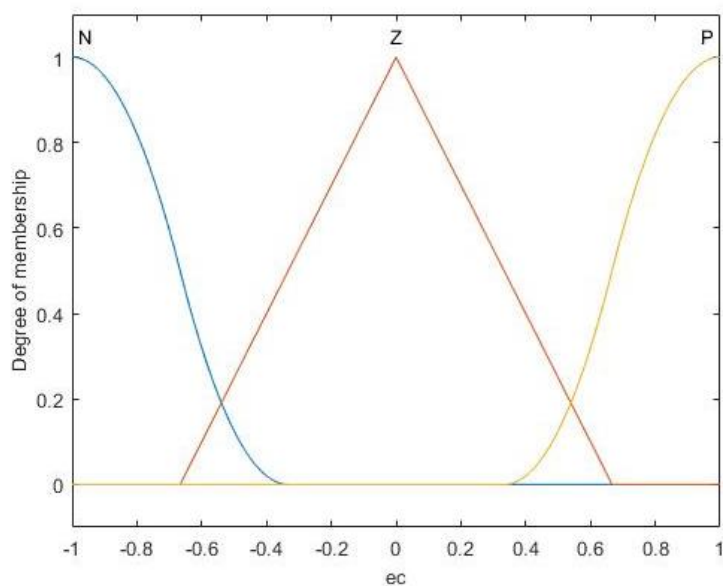
**Figura 27**

*Funciones de membresía respecto al error*



**Figura 28**

*Funciones de membresía respecto a la derivada del error*



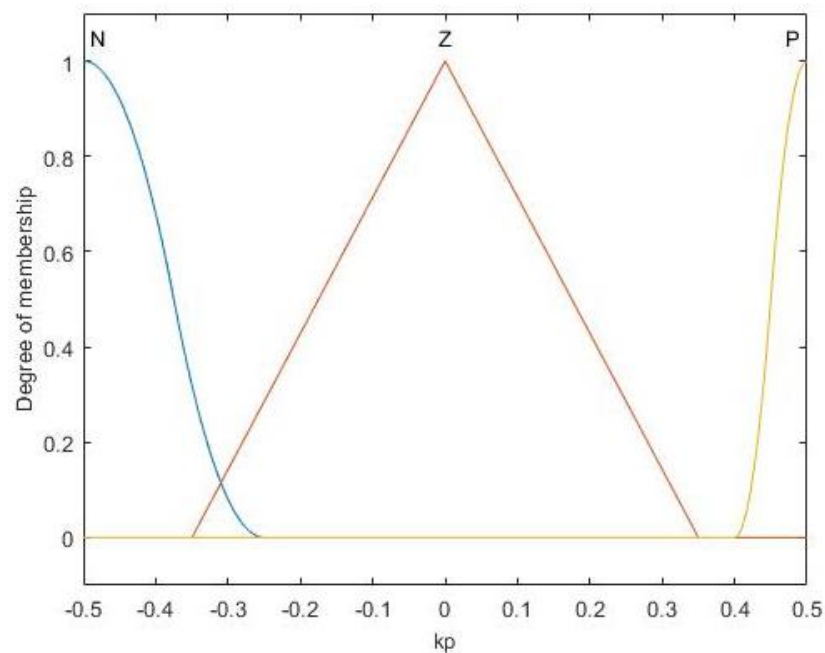
Como se puede observar en las figuras 27 y 28 los valores en los que se maneja el error son entre 1 y -1. Para el rango de los valores de  $k_p$  y  $k_i$  a través de la función *pid* se obtiene la forma paralela de un PI discreto con tiempo de muestreo igual a 16 segundos, como se muestra a continuación en la ecuación 2.15.

$$c(z) = K_p + K_i * \frac{Ts}{z - 1} \quad (2.15)$$

Donde los valores de  $k_p$  y  $k_i$  corresponden a 0.433 y 0.0271 respectivamente. Por lo tanto, para sintonizar estos valores del controlador PI se tiene en cuenta estos rangos en el diseño de las funciones de membresía, obteniéndose así las figuras 29 y 30 mostradas a continuación.

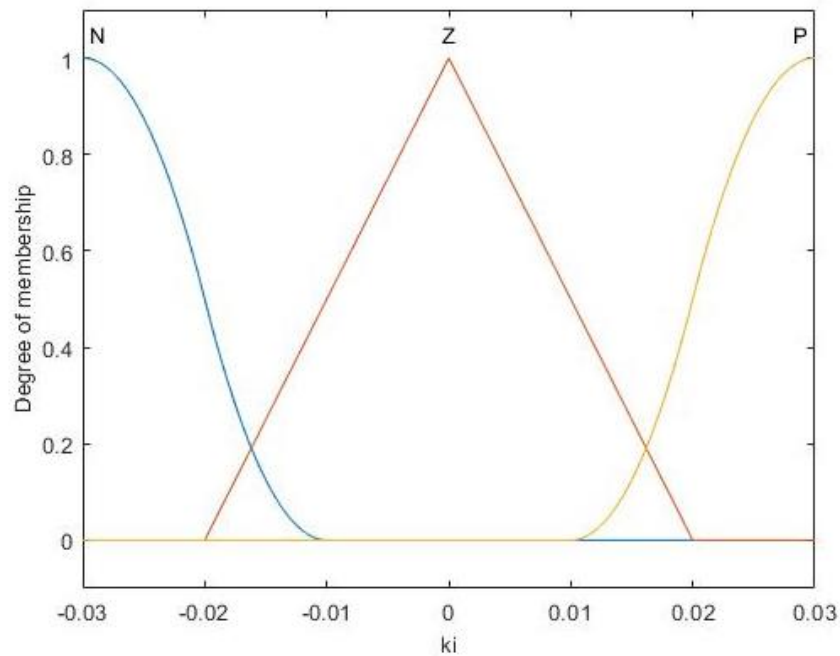
**Figura 29**

*Funciones de membresía respecto al valor de  $k_p$*



**Figura 30**

*Funciones de membresía respecto al valor de  $k_i$*



Las reglas en las que se basó el controlador autoajutable difuso para generar los valores correspondientes a  $k_p$  y  $k_i$ , descritas en las tablas 3 y 4, se muestra en la figura 31 donde se detallan de manera lingüística 9 reglas.

**Figura 31**

*Reglas del FLC para estimar valores de  $k_p$  y  $k_i$*

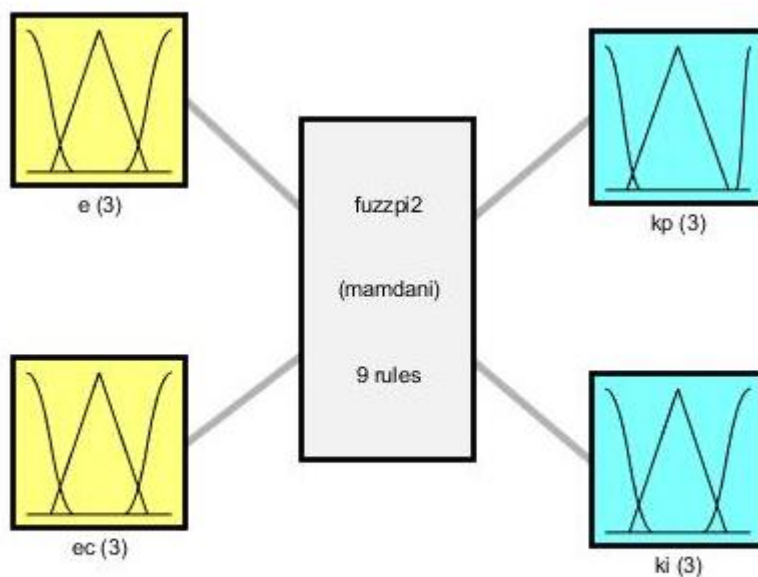
- ```
'1. If (e is N) and (ec is N) then (kp is N) (ki is Z) (1) '
'2. If (e is N) and (ec is Z) then (kp is N) (ki is Z) (1) '
'3. If (e is N) and (ec is P) then (kp is N) (ki is Z) (1) '
'4. If (e is Z) and (ec is N) then (kp is N) (ki is P) (1) '
'5. If (e is Z) and (ec is Z) then (kp is P) (ki is P) (1) '
'6. If (e is Z) and (ec is P) then (kp is P) (ki is P) (1) '
'7. If (e is P) and (ec is N) then (kp is P) (ki is Z) (1) '
'8. If (e is P) and (ec is Z) then (kp is P) (ki is Z) (1) '
'9. If (e is P) and (ec is P) then (kp is P) (ki is Z) (1) '
```



En la figura 32 se muestra un diagrama de bloques correspondiente a cómo se formó el controlador lógico difuso en el cual se tiene dos entradas, las cuales son el error y la derivada del error y dos salidas que vienen a ser los valores estimados de los parámetros correspondientes al controlador PI. La técnica utilizada para el diseño del controlador difuso fue Mamdani. También se puede observar en las figuras 33 y 34 la superficie de control que se genera al estimar los valores de  $k_p$  y  $k_i$  en base a las reglas definidas en la figura 31.

### Figura 32

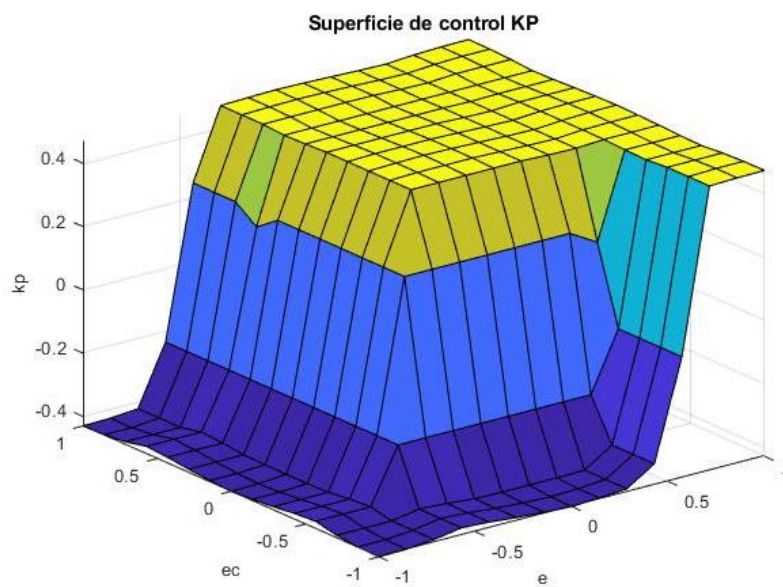
*Diagrama de bloques del controlador lógico difuso*



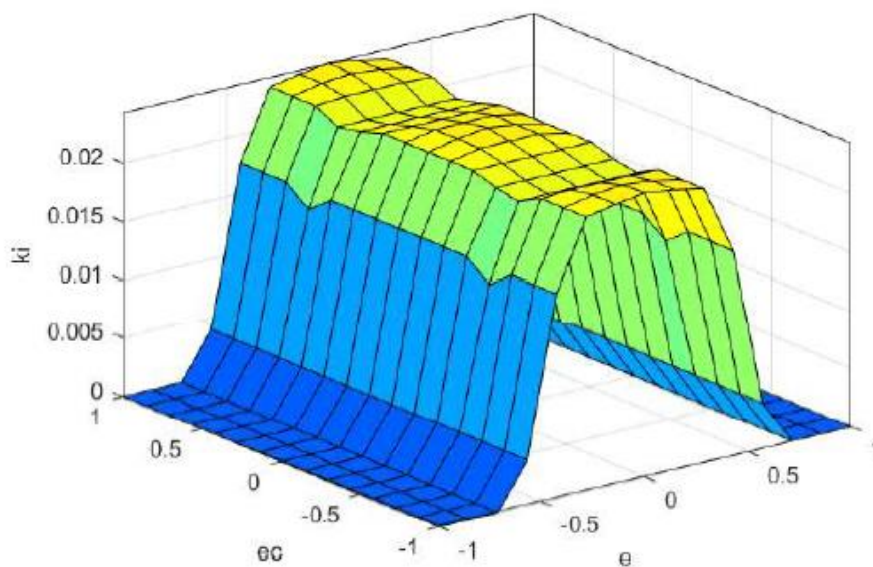
System fuzzypi2: 2 inputs, 2 outputs, 9 rules

**Figura 33**

*Superficie de control para el valor sintonizado de  $K_P$*

**Figura 34**

*Superficie de control para el valor sintonizado de  $K_I$*



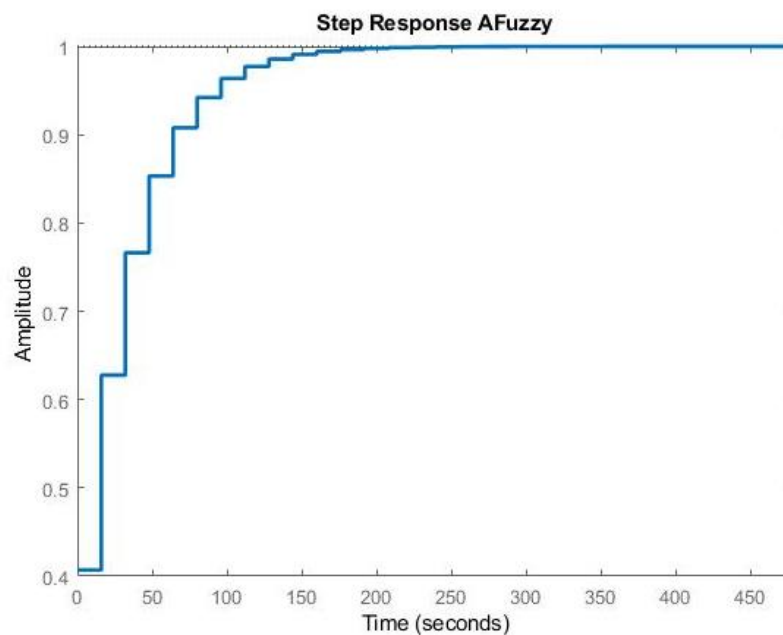
La estimación de los valores correspondientes a  $k_p$  y  $k_i$  demuestran mediante su superficie de control como van estimando el valor en función del error y la derivada del error. Así por ejemplo se

puede observar que para los valores de la parte integral el  $k_i$  tiende a valores de cero cuando el error también es cero.

En la figura 35, se puede observar la respuesta obtenida en la planta de temperatura tras el diseño del controlador lógico difuso para los valores estimados de  $k_p$  y  $k_i$ . Como se puede observar la respuesta cumple con los requerimientos de diseño dispuestos, no presenta sobre impulso y hace seguimiento a referencia por lo que demuestra que el diseño del controlador lógico difuso es óptimo.

**Figura 35**

*Respuesta del Sistema controlador AFuzzy*



A lo largo de este capítulo se ha realizado el diseño correspondiente a los controladores PI 2DOF, SSIA y AFuzzy. Se obtuvieron las primeras gráficas respecto al diseño realizado y en base a los criterios de desempeño especificados, se han utilizado las distintas metodologías hasta obtener una respuesta adecuada del sistema. Por lo que para el controlador PI 2DOF se trató con tres técnicas de sintonización, mientras que para el controlador SSIA y AFuzzy la primera técnica de diseño realizada demostró cumplir satisfactoriamente con los requerimientos de diseño. Así también el controlador

AFuzzy fue diseñado en base al criterio de ser autoajutable respecto a sus valores de ganancia proporcional e integral, este criterio adaptativo se lo corroborara más adelante en el capítulo V en simulación y tiempo real ante distintas entradas de perturbación al sistema.

#### Capítulo IV. Generación de código para la plataforma de software CODESYS

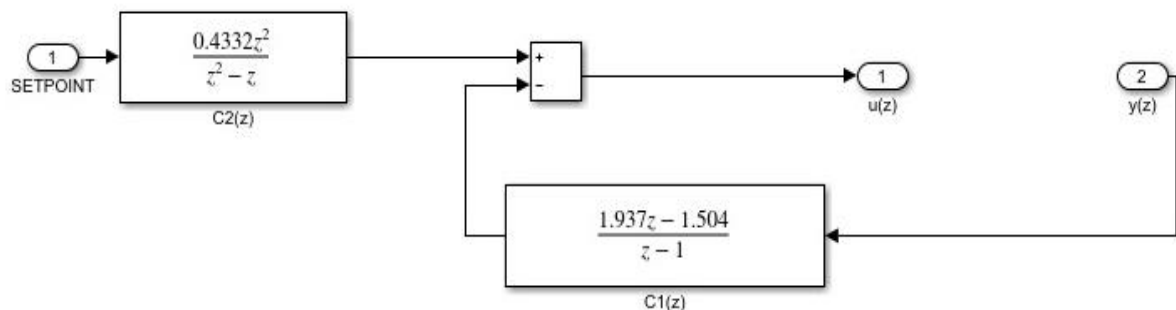
En este capítulo se presenta la generación de código de los controladores PI 2DOF, SSIA y AFuzzy diseñados. Se plantea un subsistema que parte del diseño en diagrama de bloques en la herramienta Simulink. Se visualiza el fragmento de la generación de código que contiene la ley de control correspondiente al controlador PI 2DOF y SSIA. Para el controlador AFuzzy se presenta la generación de código completa en la sección de apéndices debido a su extensión.

##### PI 2DOF

Para la generación de código del controlador PI 2DOF se planteó el siguiente subsistema el cual posee dos entradas y una salida, donde el primer puerto a la izquierda representa el punto de consigna del sistema, el segundo puerto representa la ley de control correspondiente a  $u(z)$  y  $y(z)$  la salida del sistema como se encuentra representando a continuación en la figura 36.

**Figura 36**

*Subsistema para la generación de código PI 2DOF*



Con lo que se genera un banco de pruebas el cual simula los datos de temperatura para observar la respuesta del controlador sin necesidad de tener conectada la planta al PLC. Por lo que genera 32

ciclos sin importar la frecuencia de muestreo tomando los datos de forma continua. Genera la función principal del controlador, en donde el subsistema contiene la ley de control el cual detalla las constantes, variables y retardos de las señales de entrada y salida como se muestra en el fragmento de código generado a continuación:

```

CASE ssMethodType OF
  SS_INITIALIZE:
    (Inicialización de variables)
    DiscreteTransferFcn3_states[0] := 0.0;
    DiscreteTransferFcn3_states[1] := 0.0;
    DiscreteTransferFcn2_states := 0.0;
  SS_STEP:
    (Ley de Control)
    DiscreteTransferFcn3_tmp := In1 - (-DiscreteTransferFcn3_states[0]);
    DiscreteTransferFcn2_tmp := In2 - (-DiscreteTransferFcn2_states);
    rtb_DiscreteTransferFcn2 := (1.937 * DiscreteTransferFcn2_tmp) + (-1.504 *
DiscreteTransferFcn2_states);
    (Actualización de variables retrasadas)
    DiscreteTransferFcn3_states[1] := DiscreteTransferFcn3_states[0];
    DiscreteTransferFcn3_states[0] := DiscreteTransferFcn3_tmp;
    DiscreteTransferFcn2_states := DiscreteTransferFcn2_tmp;
    (Cálculo de la salida en función a los retardos)
    Out1 := (0.4332 * DiscreteTransferFcn3_tmp) - rtb_DiscreteTransferFcn2;
END_CASE;

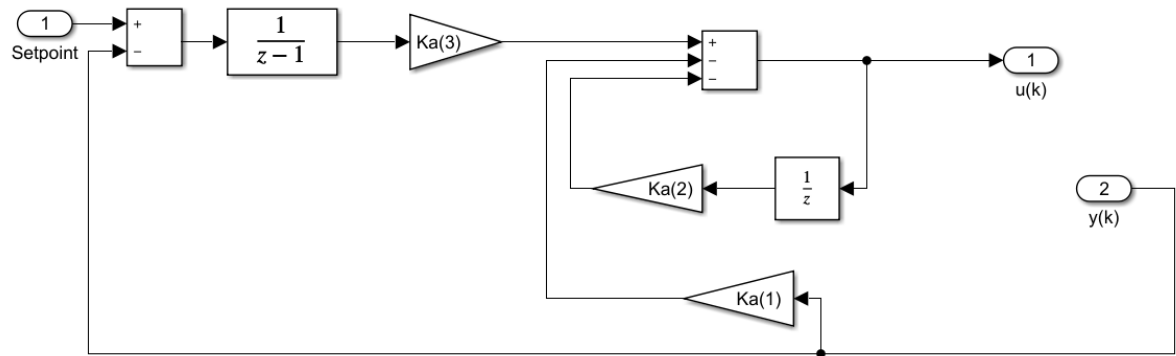
```

### SSIA

Para la generación de código del controlador SSIA se planteó el subsistema que se compone de dos entradas y una salida, donde la primera entrada representa el punto de consigna y la segunda entrada es el valor actual del sensor de temperatura y la salida para el actuador, también se puede observar las tres ganancias generadas por el diseño planteado para el controlador de espacio de estados con acción integral. Este subsistema se puede observar representado a continuación en la figura 37.

Figura 37

Subsistema para la generación de código SSIA



De igual manera que en el controlador PI 2DOF a partir del subsistema diseñado en Simulink, se genera un banco de pruebas que simula los datos de temperatura para observar la respuesta del controlador sin necesidad de tener conectada la planta al PLC. También genera 32 ciclos sin importar la frecuencia de muestreo tomando los datos de forma continua. Genera la función principal del controlador, mismo que contiene la ley de control donde se detallan las constantes, variables y retardos de las señales de entrada y salida como se muestra en el fragmento de código generado continuación:

```

CASE ssMethodType OF
SS_INITIALIZE:
  (Inicialización de variables)
  DiscreteTransferFcn_states := 0.0;
  UnitDelay_DSTATE := 0.0;
  SS_STEP:
  (Ley de Control)
  uk := ((0.84694366121736908 * DiscreteTransferFcn_states) - (3.1105878633490236 * yk)) -
  (0.38188569934547445 * UnitDelay_DSTATE);
  DiscreteTransferFcn_states := (Setpoint - yk) - (-DiscreteTransferFcn_states);
  (Actualización de variables retrasadas)
  UnitDelay_DSTATE := uk;
END_CASE;

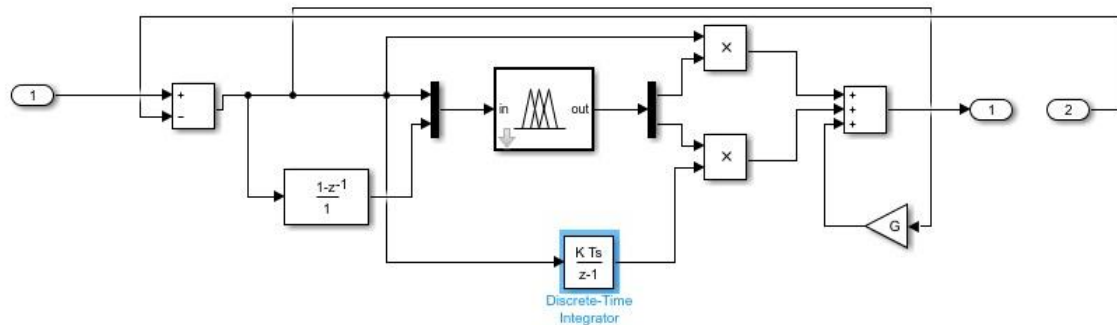
```

## AFuzzy

La generación de código para el controlador AFuzzy se realiza mediante la implementación del diseño del controlador en Simulink para así poder identificar los puertos de entrada y salida correspondientes. De esta manera se muestra a continuación en la figura 38 el subsistema realizado para la generación de código del controlador. Donde se puede observar que se presentan los mismos puertos de entrada y salida como lo son el punto de consigna a la entrada y los otros dos puertos correspondientes a la salida de la ley de control y la salida del sistema.

**Figura 38**

*Subsistema para la generación de código AFuzzy*



Se genera de igual manera un banco de pruebas que simula los datos de temperatura para observar la respuesta del controlador sin necesidad de tener conectada la planta al PLC. Genera igualmente 32 ciclos sin importar la frecuencia de muestreo tomando los datos de forma continua y la función principal del controlador, por lo que este subsistema contiene la ley de control donde se detallan las constantes, variables y retardos de las señales entrada y salida. Tomando en consideración que para este controlador la generación de código es extensa, el código completo generado del controlador se lo puede encontrar en la sección de apéndices del presente documento.

## **Capítulo V. Simulaciones y escenarios de prueba**

En el presente capítulo se ejecutan las respectivas simulaciones y escenarios de prueba con la aplicación de los controladores PI 2DOF, SSIA y AFuzzy. Las simulaciones se realizan mediante la herramienta de software Simulink. Se implementa el código generado en el PLC Turck para observar el comportamiento del sistema respecto a los criterios de diseño considerados tanto en seguimiento a referencia como en rechazo a perturbaciones. Se realiza un análisis de desempeño de las estrategias control diseñadas en simulación e implementándolas en la planta real.

### **Análisis de desempeño de las estrategias de control avanzado mediante simulación**

En el presente apartado se desarrolla un análisis de desempeño de las estrategias de control previamente diseñadas PI 2DOF, SSIA y AFuzzy mediante simulación. Para esto se realizaron los correspondientes diagramas de bloques para cada estrategia de control con la respectiva planta digitalizada. Estas simulaciones se las llevó a cabo mediante la herramienta Simulink y cabe recalcar que todas las estrategias de control fueron implementadas en simulación de manera digital.

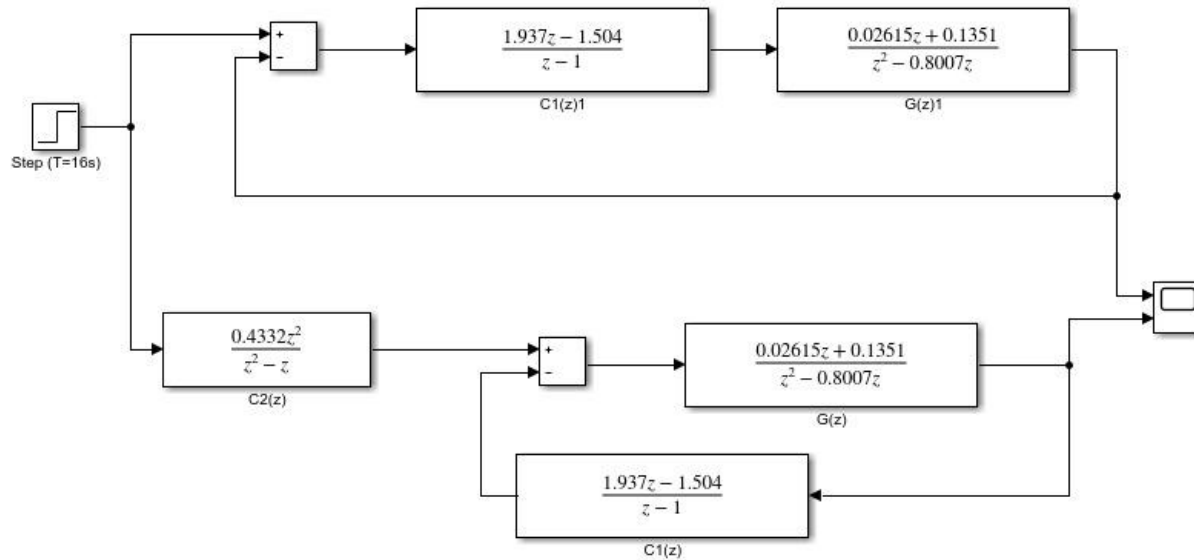
#### ***PI 2DOF***

En la figura 39 se puede observar el diseño del controlador PI 2DOF implementado en simulink con todos los parámetros de diseño propuestos de donde se obtiene la respuesta del sistema en simulación. Se realiza la simulación del controlador PI 1DOF para poder compararlo con el diseño del PI 2DOF, por tal razón en la figura 39 se observa en la parte superior del diagrama de bloques la técnica de control de un PI 1DOF y en la parte inferior se tiene el diseño de un controlador PI 2DOF.



Figura 39

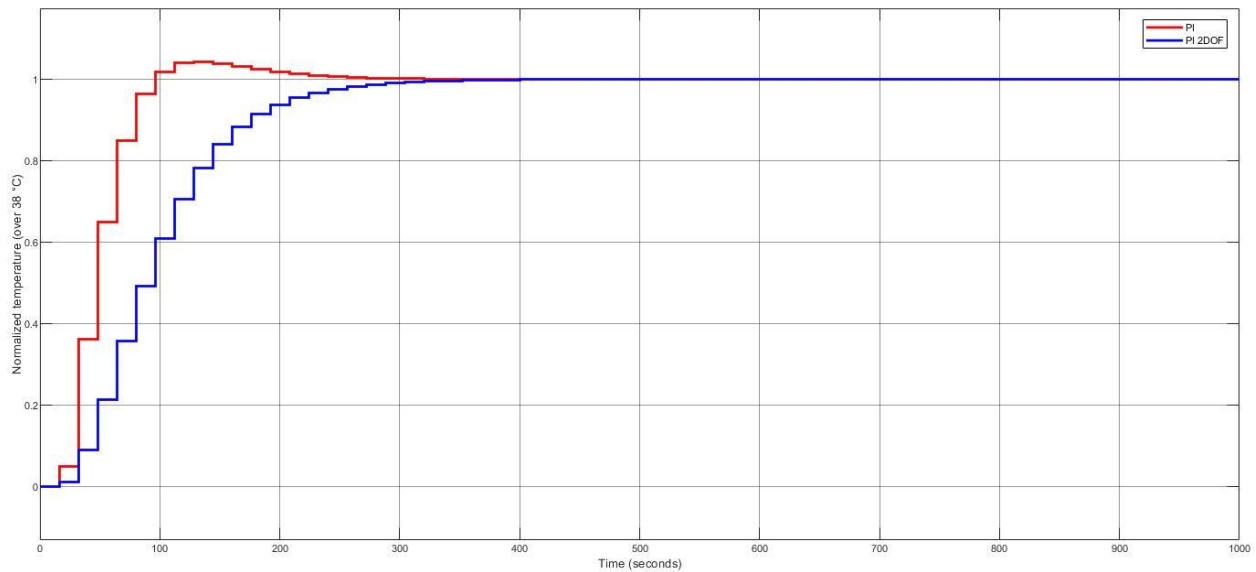
Diseño controlador PI 2DOF para simulación



La respuesta del sistema ante una entrada escalón y el controlador PI 2DOF diseñado se puede observar en la figura 40, donde se puede analizar claramente que este controlador reduce el sobre impulso en la respuesta al escalón manteniendo el tiempo de estabilización y el seguimiento a referencia. La gráfica del controlador correspondiente al PI 1DOF se encuentra representado de color rojo, distinguiéndose, así como con este controlador se tiene más sobre impulso que a comparación del controlador PI 2DOF representado de color azul en la gráfica. También es importante señalar que la simulación se la realizó con el mismo período de muestreo que se trabaja para la identificación de la planta y el diseño de los tres controladores, llevando así un tiempo de simulación de 1000 segundos para poder apreciar de mejor manera el comportamiento del controlador.

**Figura 40**

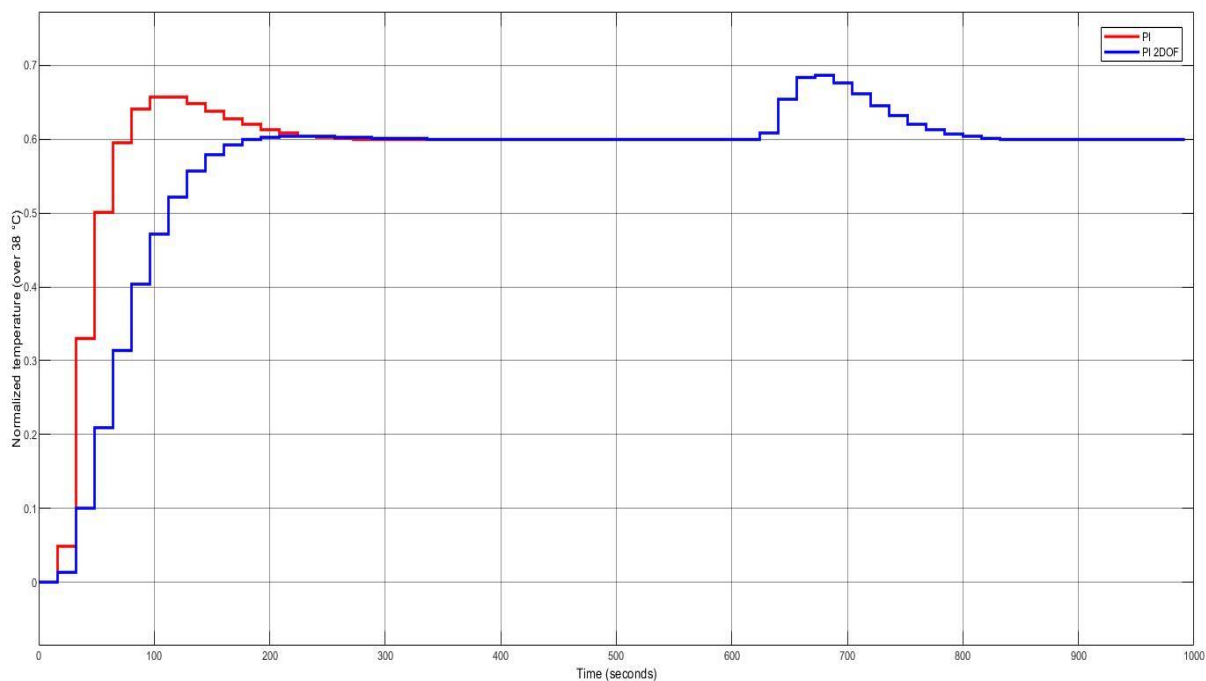
*Respuesta del Sistema térmico con un controlador PI 2DOF en simulación*



En los escenarios de simulación de los controladores se realizan pruebas con cambio de punto de consigna e introducción de perturbaciones. Al introducir una perturbación se obtiene el resultado de la figura 41 en donde la respuesta del controlador PI 2DOF con un punto de consigna de temperatura normalizada sobre 38 °C de valor 0.6; se introduce una perturbación a los 600 segundos de un aumento de temperatura representado su valor normalizado en 0.3, con la cual se evidencia el desempeño de este controlador respecto al rechazo de perturbaciones y el seguimiento a referencia.

Figura 41

Simulación del controlador PI2DOF con rechazo a perturbaciones y seguimiento a referencia

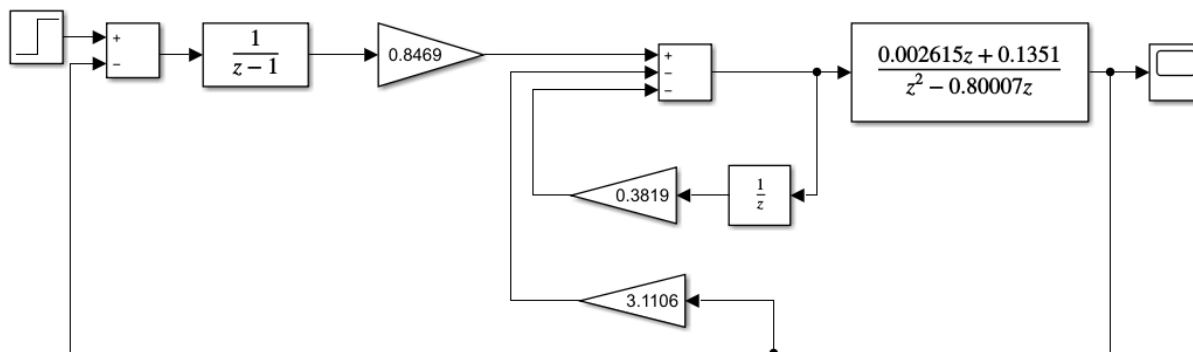


### SSIA

El diagrama de bloques de simulación para el controlador SSIA se presenta a continuación en la figura 42 implementado en Simulink con todos los parámetros de diseño propuestos.

Figura 42

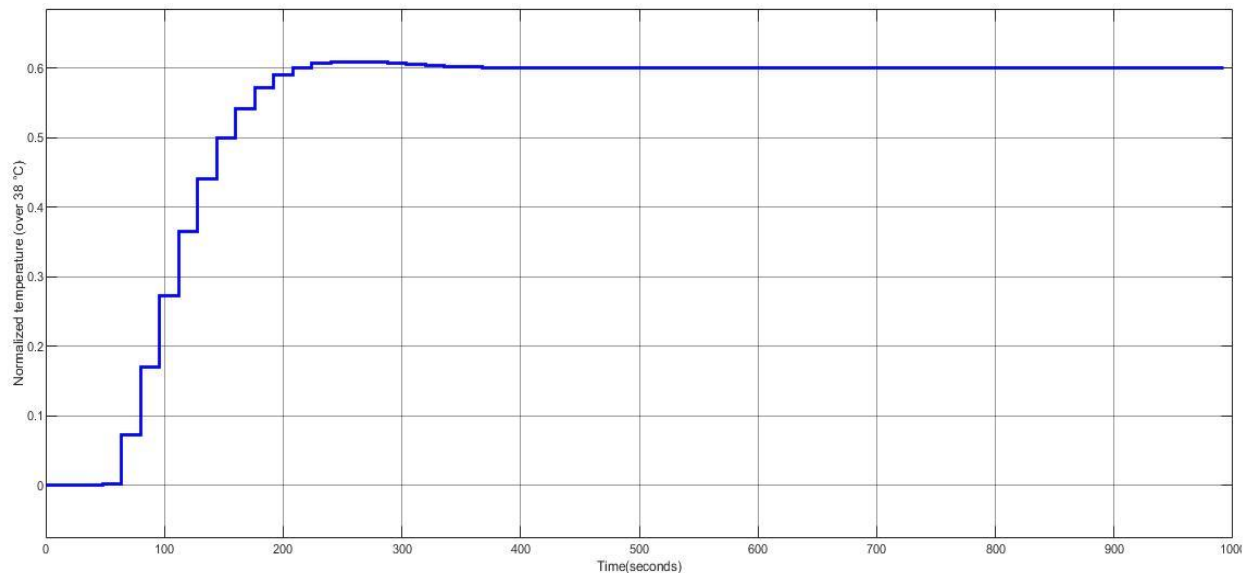
Diseño controlador SSIA para simulación



Donde se obtiene el resultado de simulación en la figura 43. Se puede observar que el controlador SSIA cumple con las especificaciones de diseño esperado, con un tiempo de estabilización, seguimiento a referencia y rechazo a perturbaciones, sin embargo, se tiene un sobre impulso mínimo en comparación con el controlador PI 2DOF simulado anteriormente.

**Figura 43**

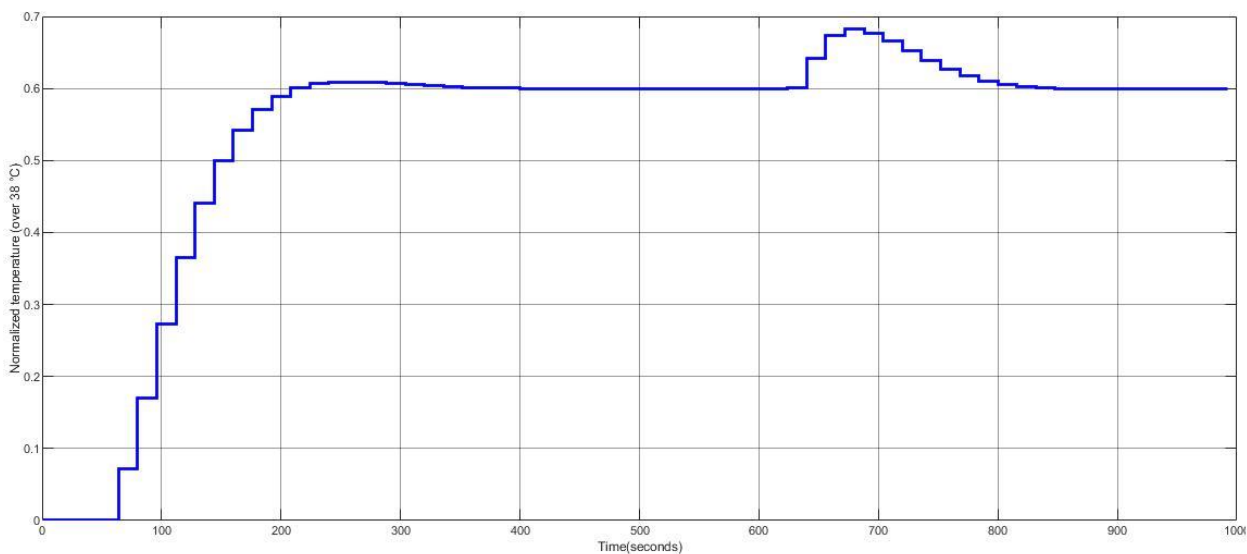
*Respuesta del Sistema térmico con un controlador SSIA en simulación*



Para la simulación del controlador SSIA con perturbaciones en el sistema, se muestra la respuesta en la figura 44, donde se tiene un punto de consigna respecto a la temperatura normalizada sobre 38 °C de un valor de 0.6 y con una perturbación generada a los 600 segundos de un valor de 0.3 que representa un aumento de temperatura con la cual se evidencia el desempeño del controlador respecto al rechazo de perturbaciones y el seguimiento a referencia.

**Figura 44**

*Simulación del controlador SSIA con rechazo a perturbaciones y seguimiento a referencia*

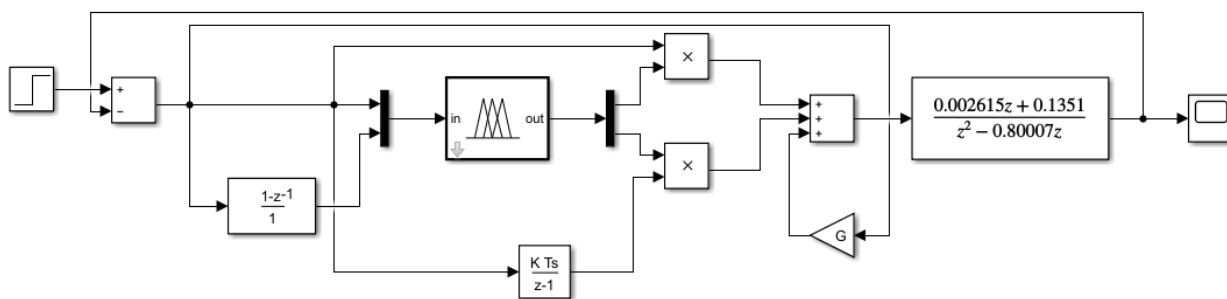


### **AFuzzy**

Para la simulación del sistema con el controlador diseñado AFuzzy se muestra en la figura 45 el controlador implementado en Simulink. Se puede observar que cumple con el diagrama de bloques presentado en el marco conceptual en la figura 13 sobre el diseño del controlador PI difuso autoajutable.

**Figura 45**

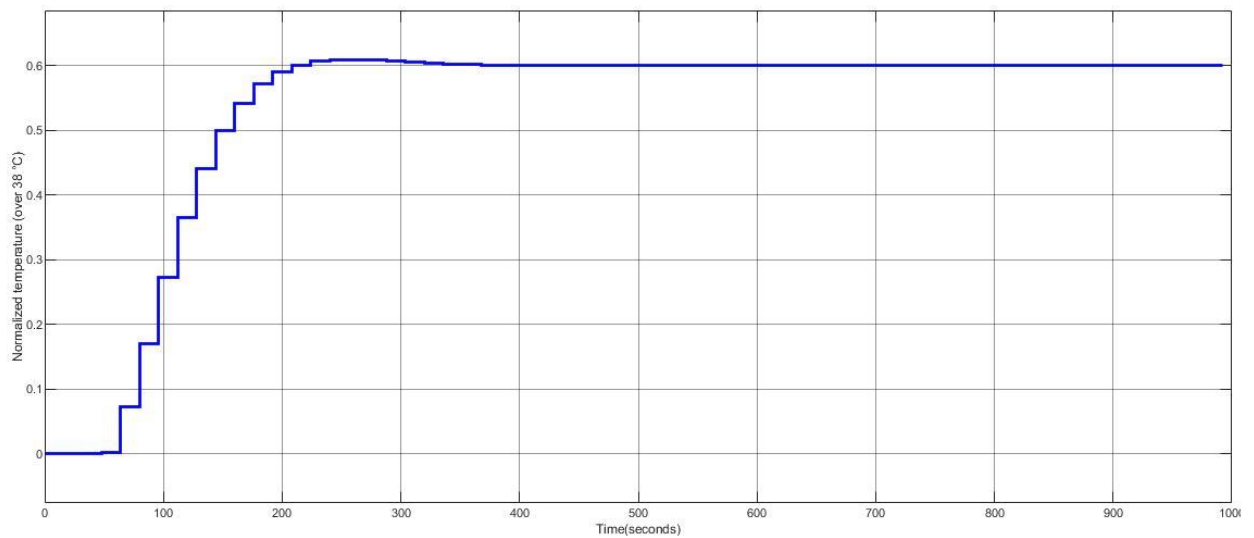
*Diseño controlador AFuzzy para simulación*



De esta manera se obtiene el siguiente resultado de simulación para el controlador AFuzzy mostrado en la figura 46, donde se puede observar que cumple con las especificaciones de diseño esperado, como el tiempo de estabilización, sin embargo, se tiene un sobre impulso mínimo, pero este sobre impulso resulta ser menor que el sobre impulso obtenido en los dos anteriores controladores el PI 2DOF y SSIA.

**Figura 46**

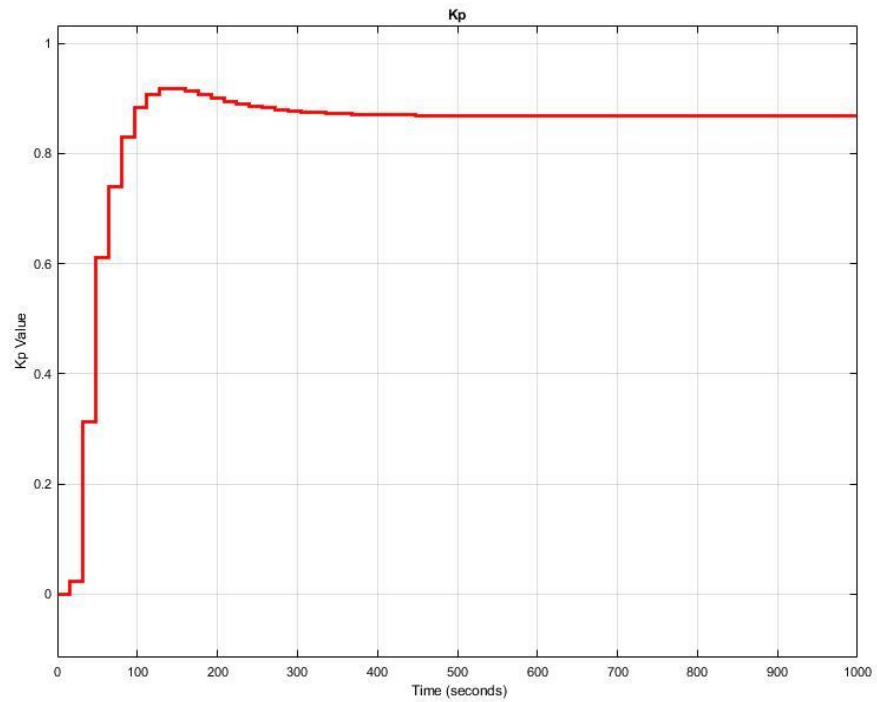
*Respuesta del Sistema térmico con un controlador AFuzzy en simulación*



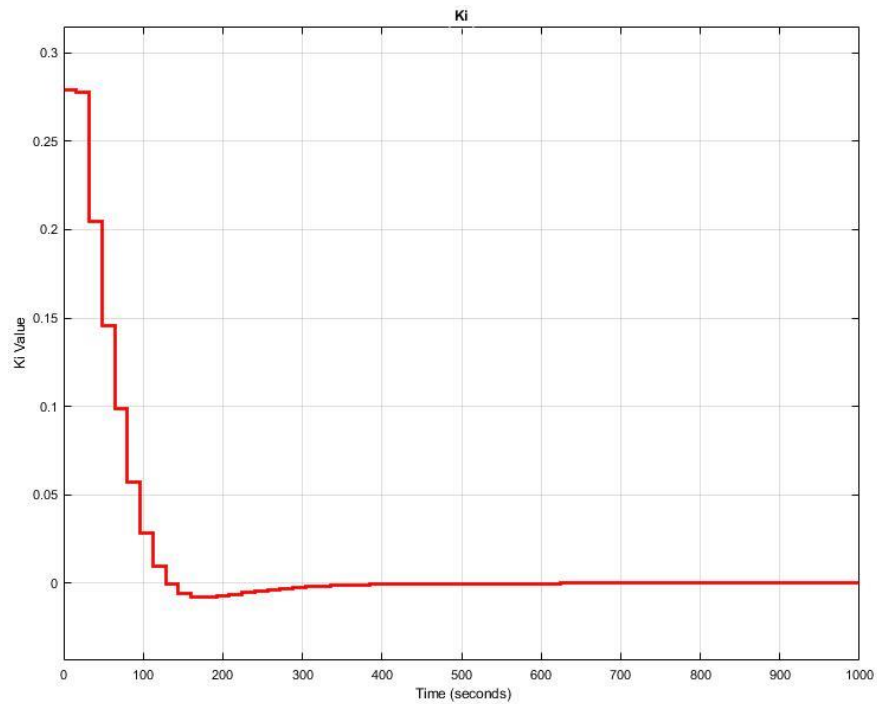
Se obtuvieron las gráficas respecto a los valores de  $k_p$  y  $k_i$  para constatar que el controlador diseñado cumpla con la especificación de ser un controlador difuso autoajutable. Por lo que en las figuras 47 y 48 a continuación se puede visualizar los valores obtenidos en simulación para el control proporcional e integral respectivamente.

**Figura 47**

Valores de  $k_p$  en simulación del controlador AFuzzy

**Figura 48**

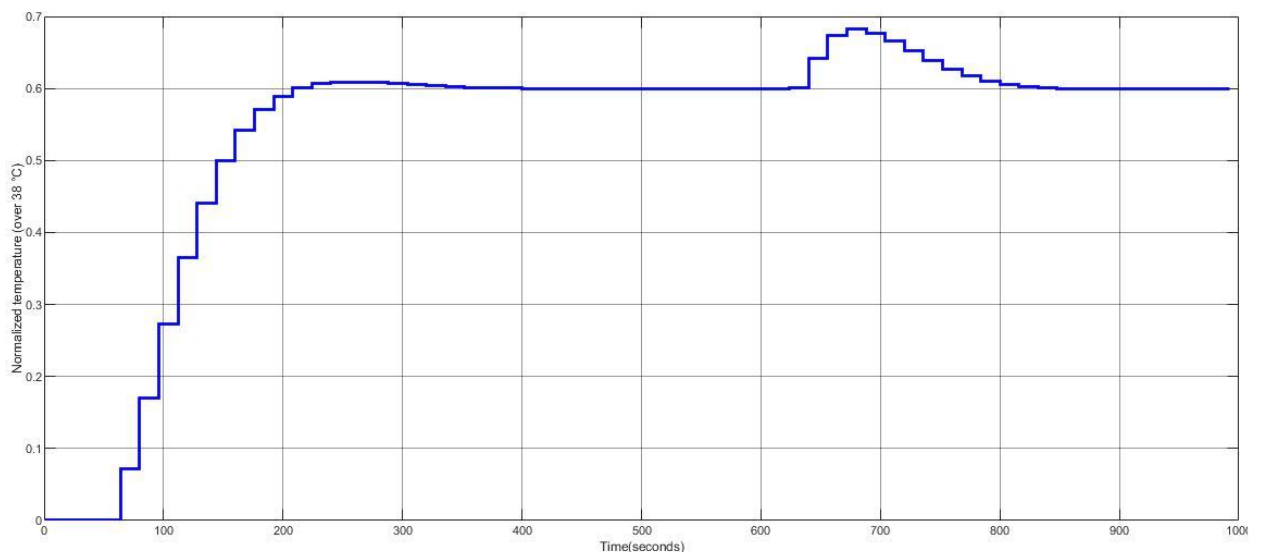
Valores de  $k_i$  en simulación del controlador AFuzzy



Siguiendo con la metodología aplicada a los dos anteriores controladores, también se presenta un escenario de prueba respecto al rechazo de perturbaciones y seguimiento a referencia del controlador en simulación. Los resultados de este escenario de prueba se muestran en la figura 49 en donde se observa la respuesta del controlador con un punto de consigna de temperatura normalizada sobre 38 °C de un valor de 0.6 y con una perturbación introducida a los 600 segundos de 0.3 representado como un aumento de temperatura, con la cual se evidencia el desempeño del controlador respecto al rechazo de perturbaciones y el seguimiento a referencia.

**Figura 49**

*Simulación del controlador AFuzzy con rechazo a perturbaciones y seguimiento a referencia*



Debido a lo cual se demuestra mediante simulación el desempeño de los tres controladores diseñados, PI 2DOF, SSIA y AFuzzy tanto en condiciones normales del sistema como ante el rechazo de perturbaciones y el seguimiento a referencia. Cada controlador presenta un buen desempeño, con sus respectivas diferencias. Este análisis respecto a la respuesta del sistema con los tres controladores diseñados se puede resumir a continuación en la tabla 7.



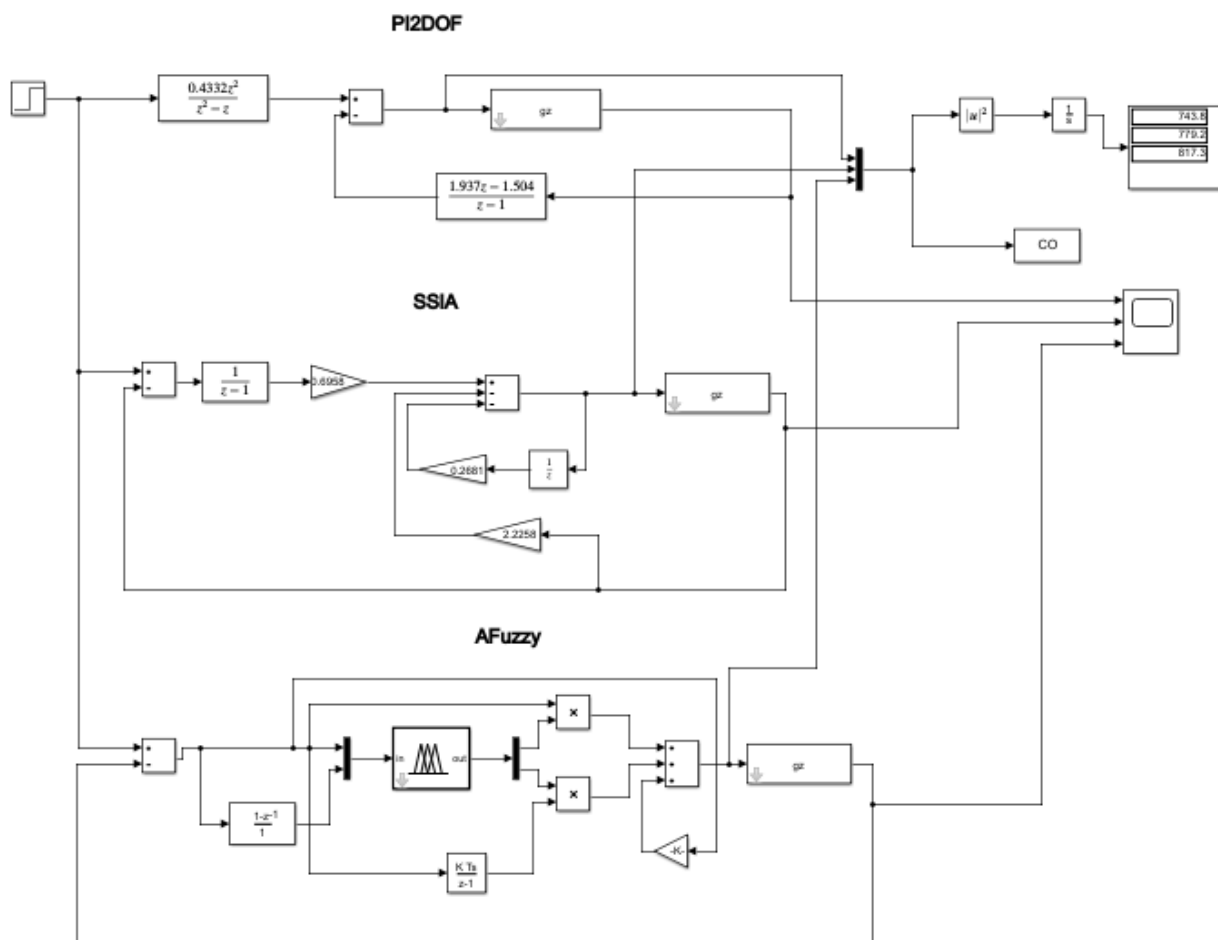
Tabla 7

Comparación de resultados en simulación de los controladores PI2DOF, SSIA Y AFuzzy

| Controladores | Sobre impulso (%) | Tiempo de establecimiento(s) | Tiempo de subida(s) | Consumo de energía |
|---------------|-------------------|------------------------------|---------------------|--------------------|
| PI2DOF        | 0.5               | 417.802                      | 138.982             | 743.8              |
| SSIA          | 5.85              | 528.696                      | 92.62               | 779.2              |
| AFuzzy        | 2.57              | 381.809                      | 74.206              | 817.3              |

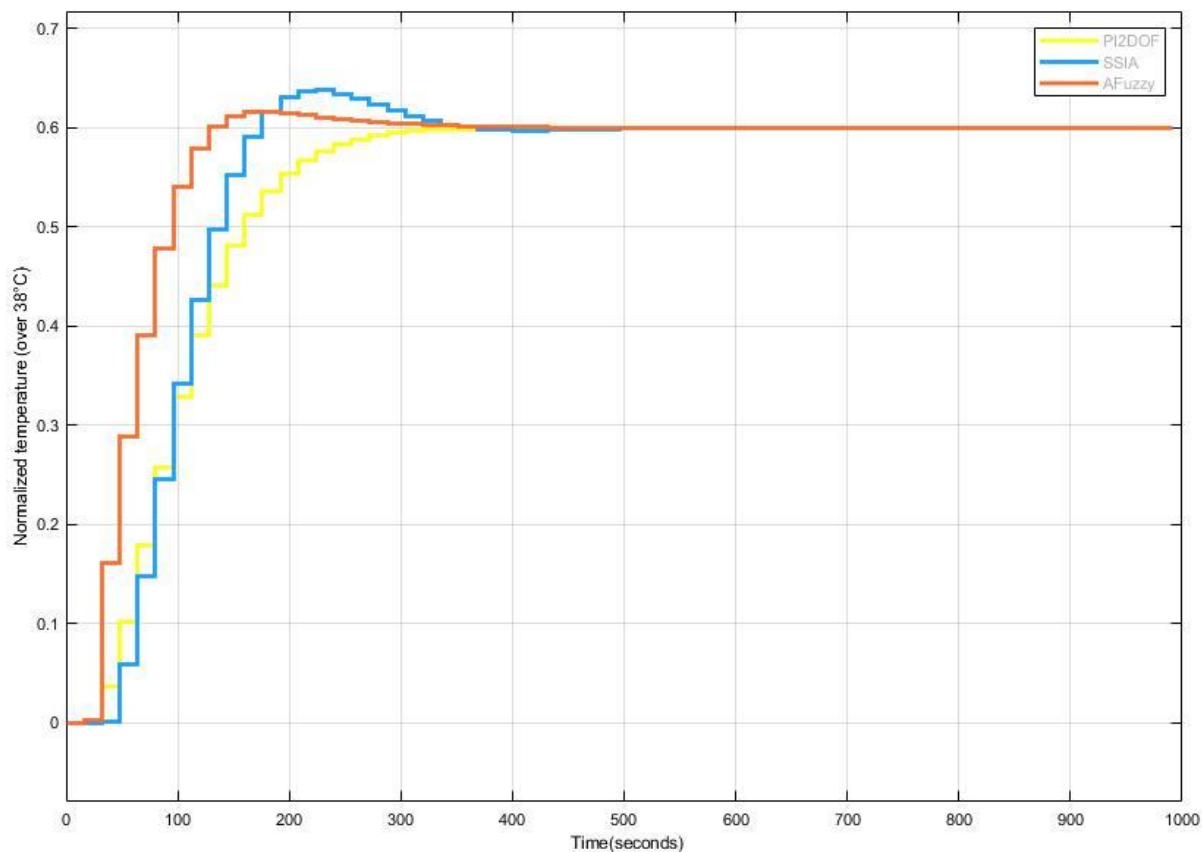
Figura 50

Bloque de comparación de controladores PI2DOF, SSIA y AFuzzy en simulación



**Figura 51**

*Respuesta del Sistema ante los controladores PI2DOF, SSIA y AFuzzy en simulación*



En la figura 51 se detalla la comparación de los tres controladores diseñados en donde respecto a la tabla 7, el controlador que presenta menos porcentaje de sobre impulso es el controlador PI 2DOF seguido del controlador AFuzzy. Respecto al tiempo de establecimiento el controlador que presenta mejor desempeño en este parámetro es el AFuzzy. En el consumo de energía de los controladores el que menor consumo representa es el controlador PI 2DOF. También respecto al tiempo de subida de cada controlador el controlador AFuzzy presenta un mejor desempeño ante los otros dos controladores con un valor de 74.206s.

## **Análisis de desempeño de las estrategias de control avanzado en tiempo real**

En el presente apartado se realiza el análisis de desempeño correspondiente a las estrategias de control diseñadas PI 2DOF, SSIA y AFuzzy implementadas en tiempo real sobre la planta de temperatura física. Se muestran las gráficas obtenidas sobre la respuesta del sistema y su reacción ante perturbaciones. Así también se puede observar el comportamiento de cada controlador ante el seguimiento a referencia.

### ***PI 2DOF***

En la figura 52 se tiene la implementación del controlador PI 2DOF en el PLC obteniéndose la siguiente respuesta que cumple con las especificaciones de diseño de tiempo de estabilización y seguimiento a referencia mientras que el sobre impulso no se pudo evitar en la implementación del controlador a escala real. El punto de consigna de temperatura normalizada sobre 38 °C es del valor de 0.78 como se observa en el gráfico la línea de color verde. Es importante señalar que para todos los gráficos respecto a la implementación de los controladores en tiempo real la leyenda del gráfico representa como entrada  $u(k)$  representa la señal de control hacia la planta mientras que la salida  $y(k)$  representa el valor actual de temperatura.

En la figura 53 se muestra un escenario de prueba respecto al rechazo de perturbaciones en donde a los 735 segundos se generó una perturbación externa que redujo la temperatura alrededor de 5°C de forma variable. Con lo que se obtuvo un correcto seguimiento a referencia tras la perturbación generada. Se observa también un cambio de punto de consigna de 0.78 a 0.65, con lo que se obtuvo un correcto seguimiento a referencia. De igual manera entre los 800 y 1000 segundos se generó una perturbación externa que elevó la temperatura, con lo que se obtuvo un correcto rechazo a la perturbación tras la intervención en el sistema de control.

Figura 52

Respuesta del Sistema térmico con un controlador PI 2DOF en tiempo real

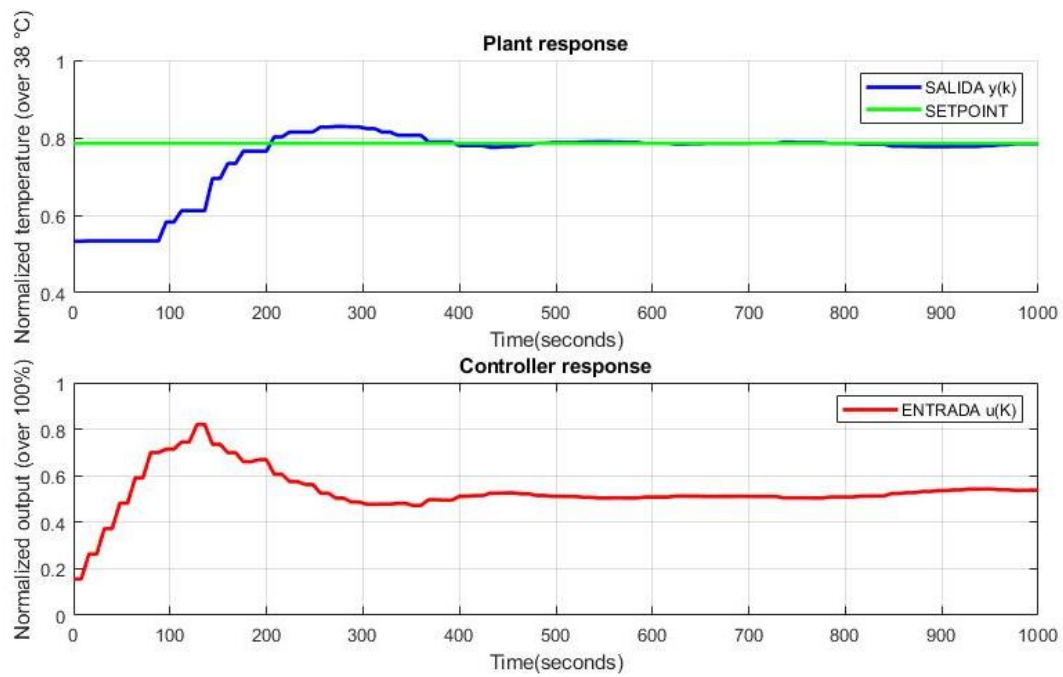
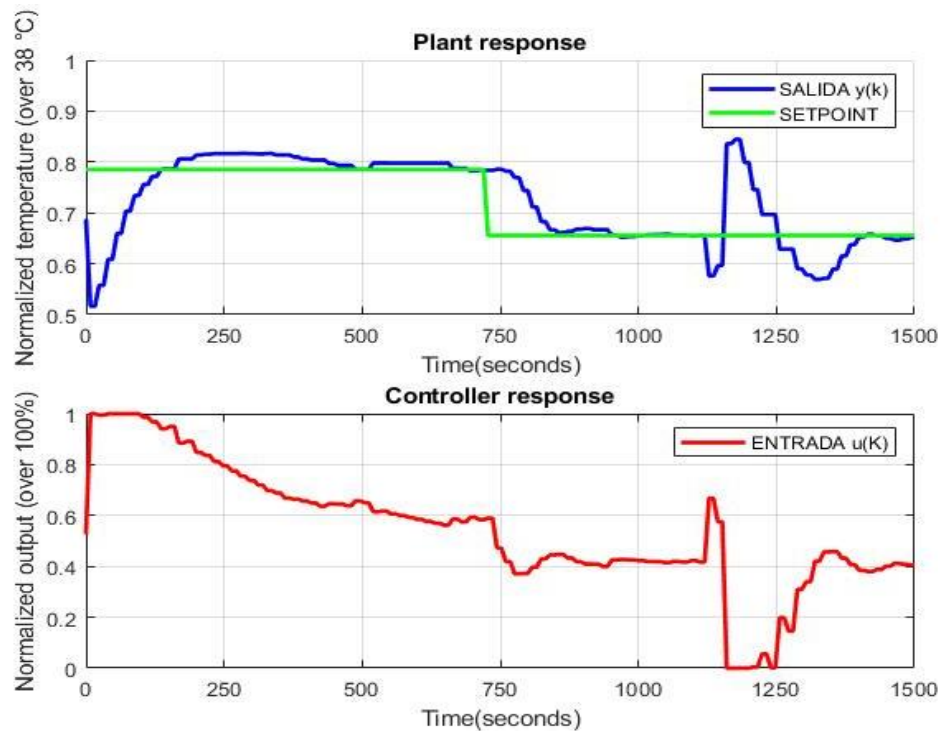


Figura 53

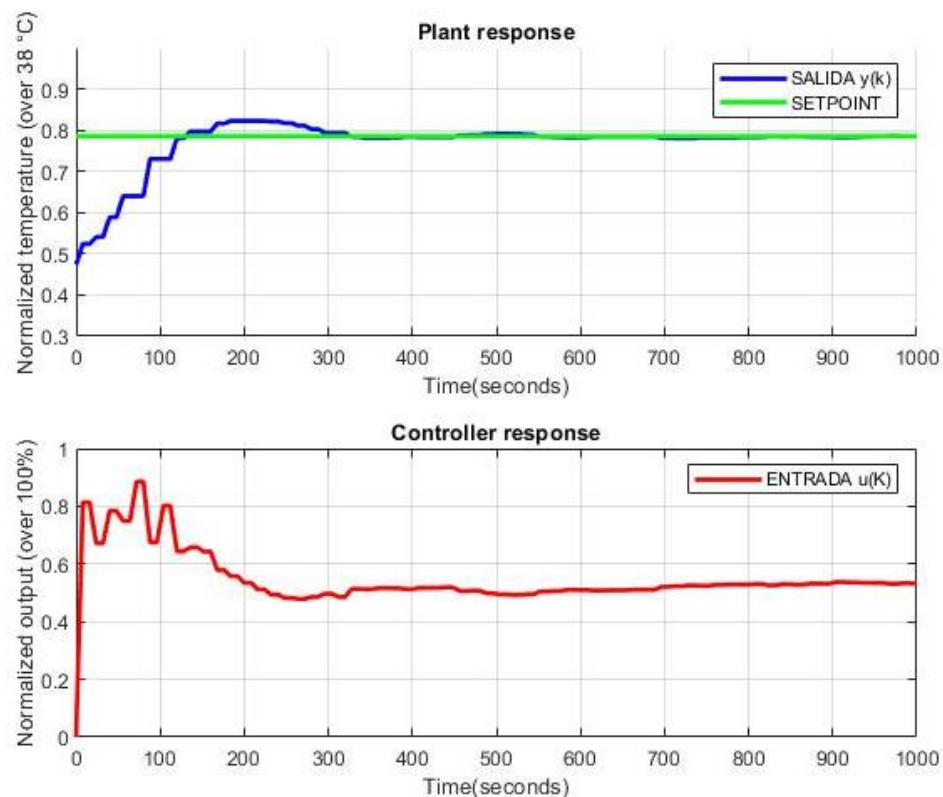
Respuesta del Sistema térmico ante una perturbación con el controlador PI 2DOF en tiempo real



Además, se realizó un ensayo de implementación sobre el sistema solo con el controlador PI 1DOF para visualizar el comportamiento de este controlador en la planta y obtener una comparación adicional sobre este controlador respecto al PI 2DOF. El resultado se puede visualizar en la figura 54 en donde el controlador PI 1DOF también presenta un sobre impulso al ser implementado en tiempo real, concluyendo así que tanto el controlador PI 1DOF y el PI 2DOF presentan sobre impulso en tiempo real.

**Figura 54.**

*Respuesta del Sistema térmico con un controlador PI 1DOF en tiempo real*

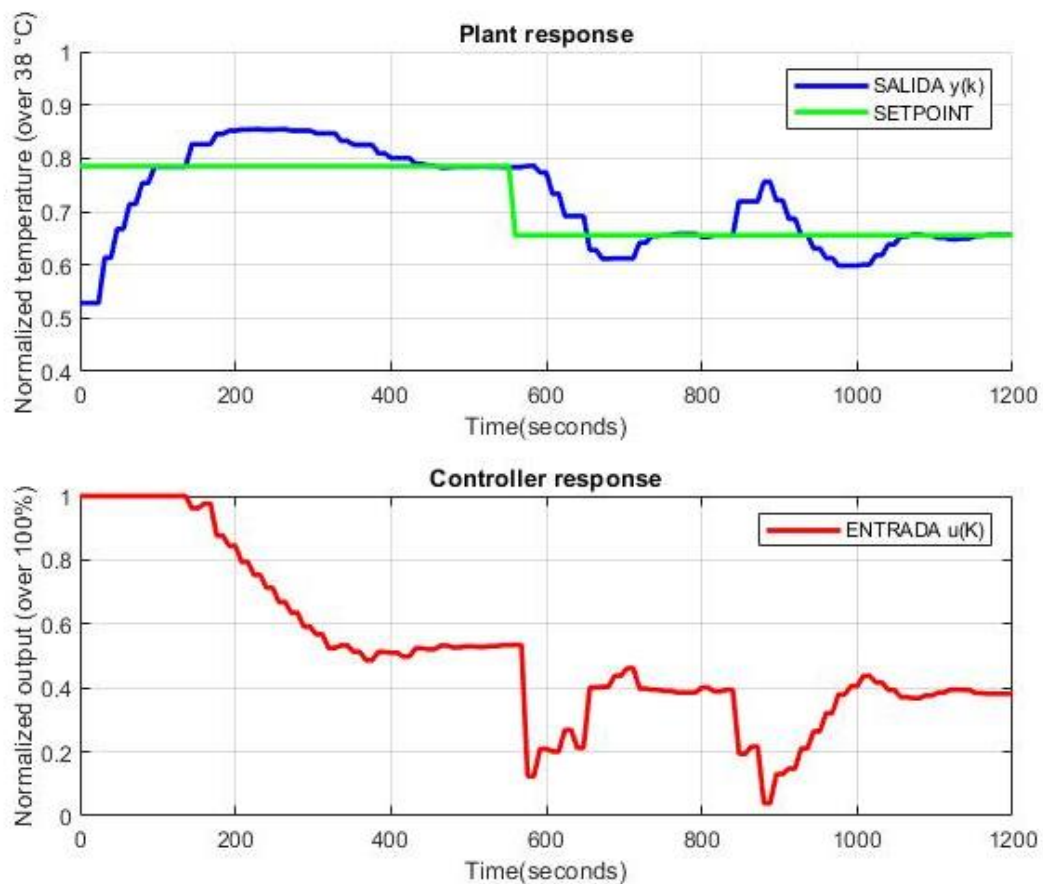


Para la implementación del controlador PI 1DOF en el sistema, se realizó un escenario de prueba en el que se incluyen perturbaciones para observar cómo reacciona el sistema ante el rechazo de perturbaciones y seguimiento a referencia. Donde aproximadamente a los 600 segundos se realizó un cambio de punto de consigna de un valor normalizado de 0.78 a 0.65, con lo que se obtuvo un correcto

seguimiento a referencia. De igual manera entre los 800 y 1000 segundos se generó una perturbación externa que misma que elevó la temperatura, con lo que se obtuvo un correcto rechazo a la perturbación tras la intervención en el sistema de control. Esto se puede observar a continuación en la figura 55.

**Figura 55**

*Respuesta del Sistema térmico ante una perturbación con el controlador PI 1DOF en tiempo real*



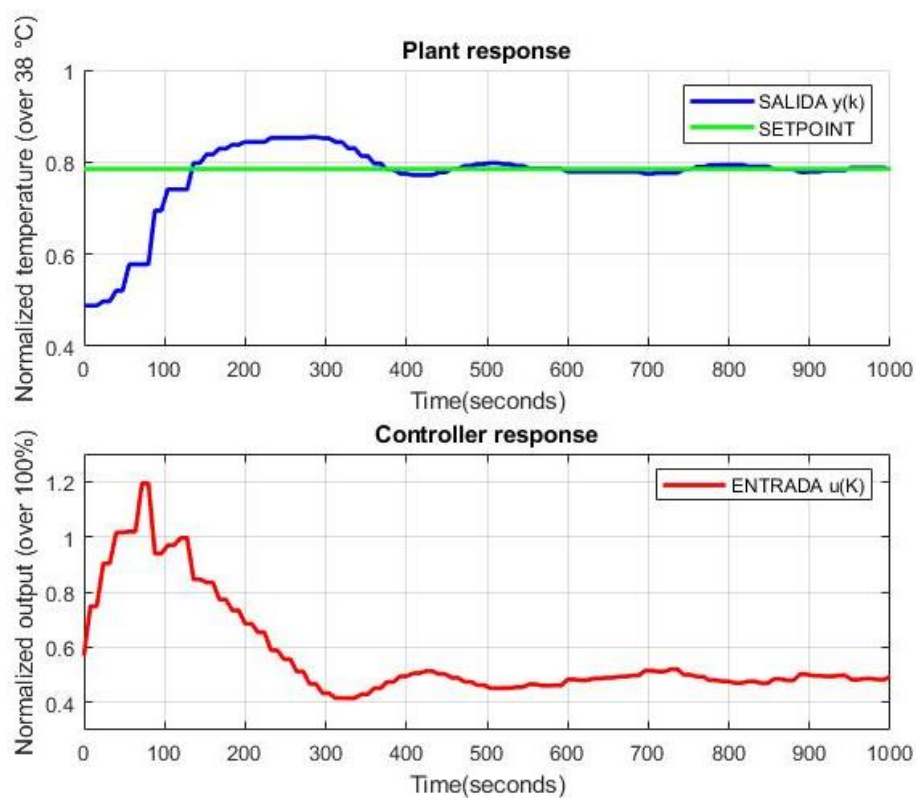
## SSIA

En la figura 56 se visualiza la implementación del controlador SSIA en el PLC obteniéndose la siguiente respuesta la cual cumple con las especificaciones de diseño de tiempo de estabilización y seguimiento a referencia mientras que el sobre impulso no se pudo evitar en la implementación del

controlador a escala real. El sobre impulso de este controlador implementado se presentó en los resultados de simulación por lo que se esperaba esta respuesta. Además, a los 800 segundos se generó una perturbación externa que redujo la temperatura alrededor de  $3^{\circ}\text{C}$  de forma variable. Con lo que se obtuvo un correcto seguimiento a referencia y rechazo a la perturbación tras la perturbación generada, representada en la figura 57.

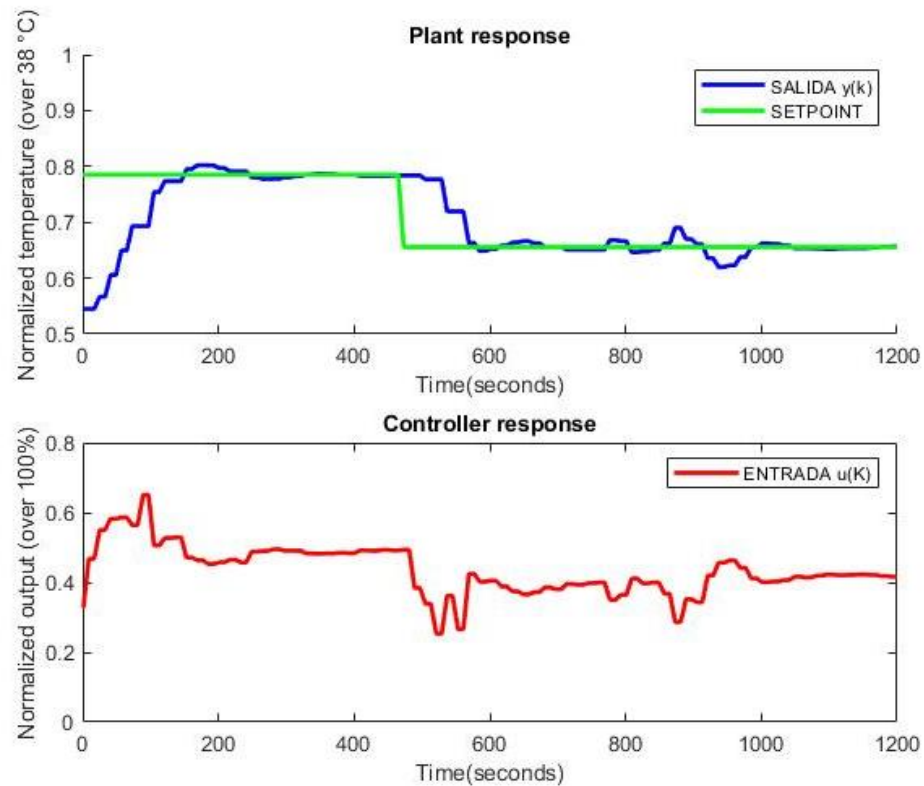
**Figura 56**

*Respuesta del Sistema térmico con un controlador SSIA en tiempo real*



**Figura 57**

*Respuesta del Sistema térmico ante una perturbación con el controlador SSIA en tiempo real*



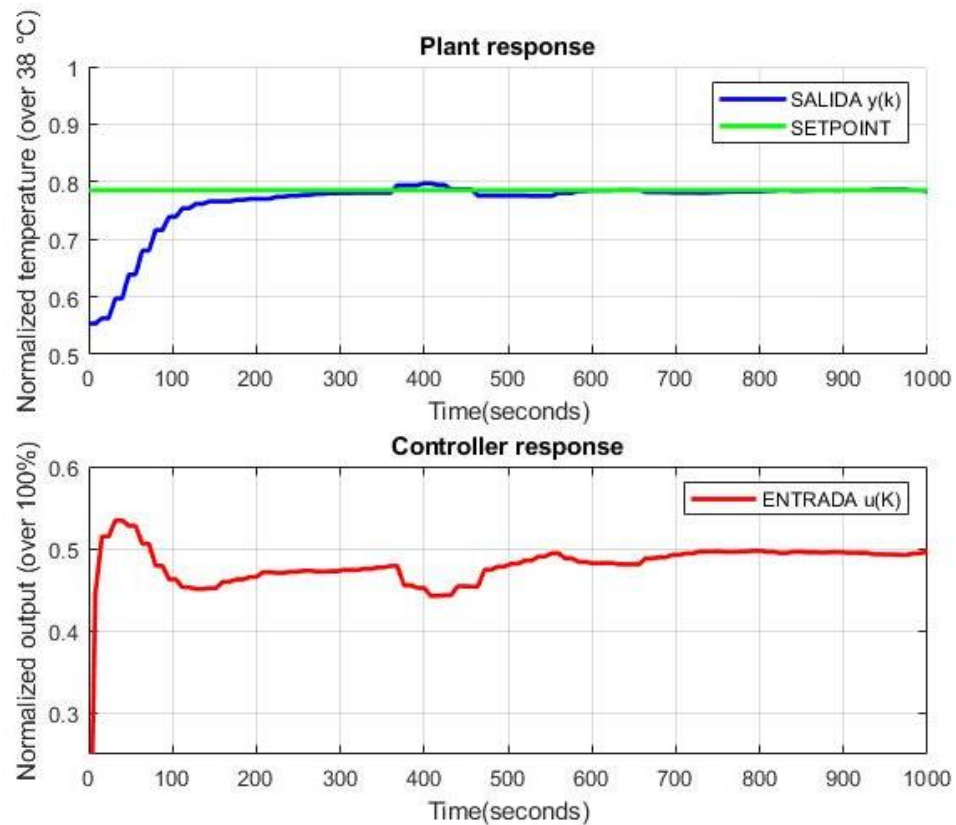
### **AFuzzy**

Tras la generación de código para el controlador AFuzzy se procedió a realizar la debida implementación del mismo en el PLC obteniéndose así un excelente resultado de desempeño por parte de este controlador. Se presentó un menor sobre impulso en comparación con los otros dos controladores anteriormente implementados el PI 2DOF y SSIA. Esto se puede visualizar en la figura 58 cómo reacciona el sistema con este controlador.



Figura 58

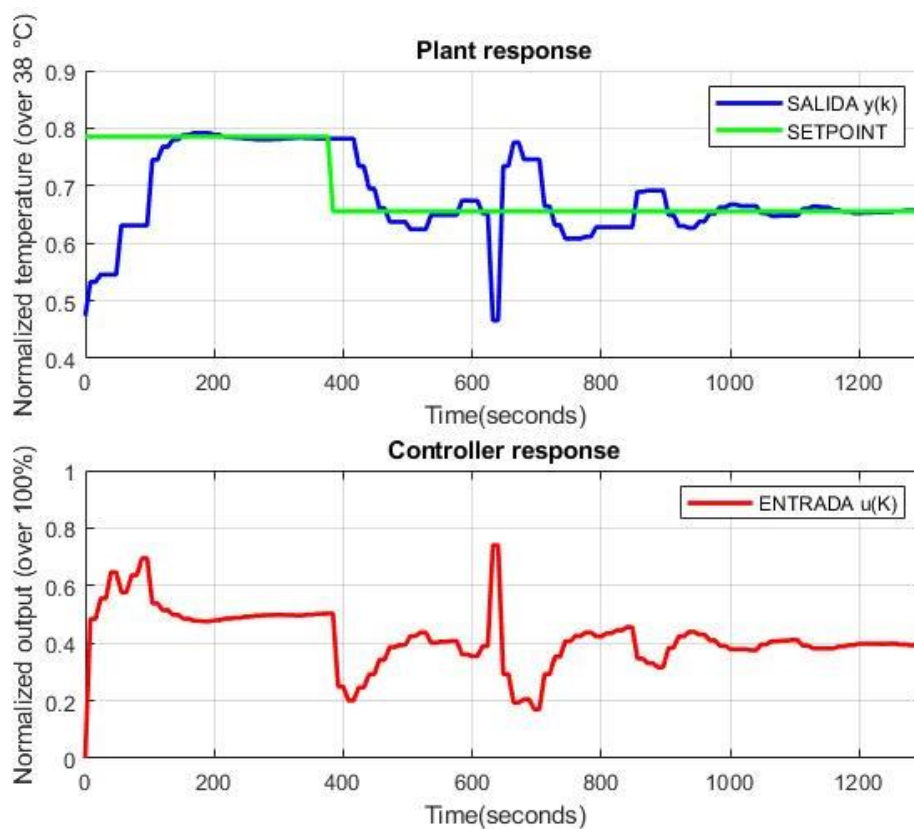
Respuesta del Sistema térmico con un controlador AFuzzy en tiempo real



Se realizó un escenario de prueba respecto al rechazo de perturbaciones y seguimiento a referencia con el controlador AFuzzy en donde con un punto de consigna de temperatura normalizada sobre 38 °C de 0.78 y en donde aproximadamente a los 400 segundos, se realizó un cambio del punto de consigna de 0.78 a 0.65, obteniéndose un correcto seguimiento a referencia. De igual manera entre los 600 y 800 segundos se generó una perturbación externa que elevó la temperatura del sistema, con lo que se obtuvo un correcto rechazo a la perturbación tras la intervención en el sistema de control. Esto se puede observar a continuación en la figura 59.

Figura 59

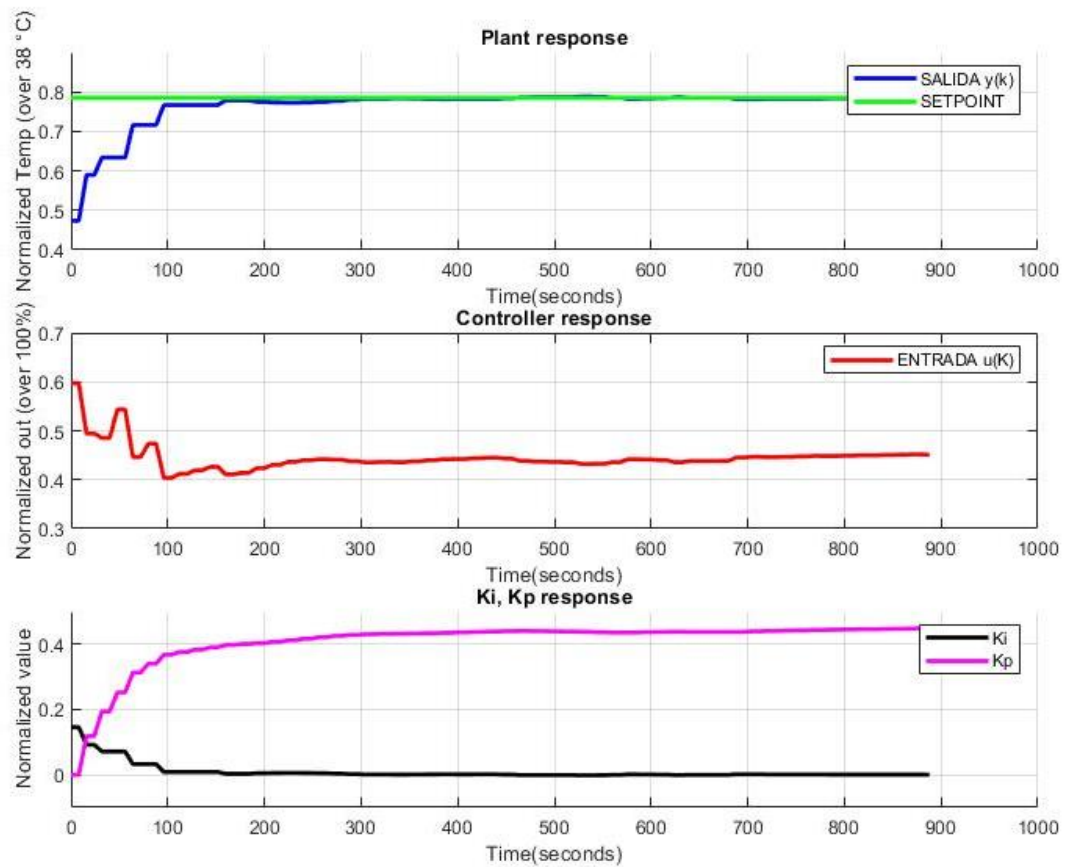
*Respuesta del Sistema térmico ante una perturbación con el controlador AFuzzy en tiempo real*



Para comprobar que el controlador AFuzzy cumple con el criterio de ser autoajutable, se generaron las gráficas de los valores de  $k_p$  y  $k_i$  en tiempo real para notar cómo estas se comportan en el tiempo en conjunto con la ley de control y la salida del sistema. De esta manera se presenta a continuación en la figura 60 el comportamiento de los valores correspondientes al control proporcional e integral.

Figura 60

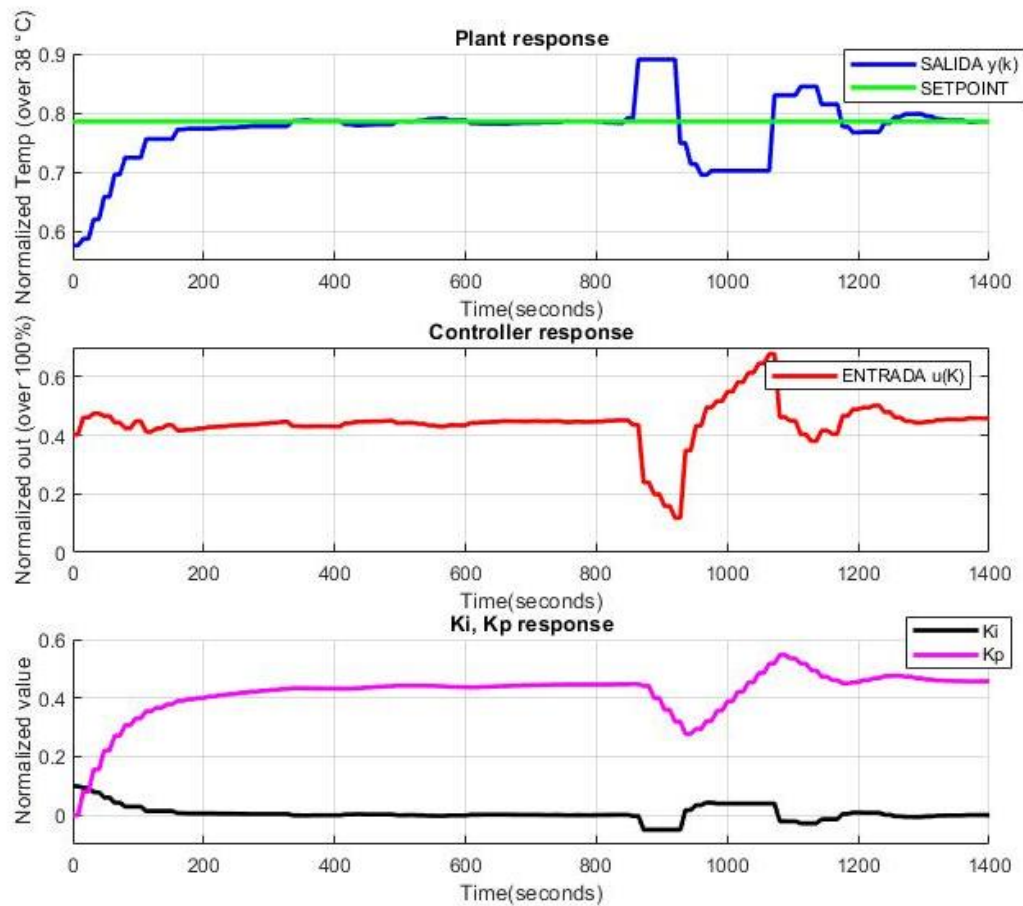
Valores de  $k_p$  y  $k_i$  en tiempo real del controlador AFuzzy



Para comprobar que el controlador AFuzzy cumple con el criterio de ser autoajutable ante perturbaciones en la planta y analizar cómo se comportan los valores de  $k_p$  y  $k_i$  se implementó un escenario de prueba, que a los 900 segundos eleva la temperatura debido a que se procedió a apagar el ventilador de la planta para generar esta perturbación. Los resultados obtenidos se pueden visualizar a continuación en la figura 61.

Figura 61

Valores de  $k_p$  y  $k_i$  en tiempo real del controlador AFuzzy ante una perturbación



Los controladores implementados reflejaron un distinto desempeño entre ellos, es decir que su comportamiento varió para cada configuración. Este análisis respecto a la respuesta del sistema con los tres controladores implementados se resume a continuación en la tabla 8. En la figura 62 se tiene la comparación de los tres controladores implementados, donde el controlador que presenta menos porcentaje de sobre impulso es el controlador Afuzzy, seguido del controlador PI2DOF. Respecto al tiempo de establecimiento el controlador, el consumo de energía y tiempo de subida el controlador AFuzzy presenta un mejor desempeño ante los otros dos controladores.

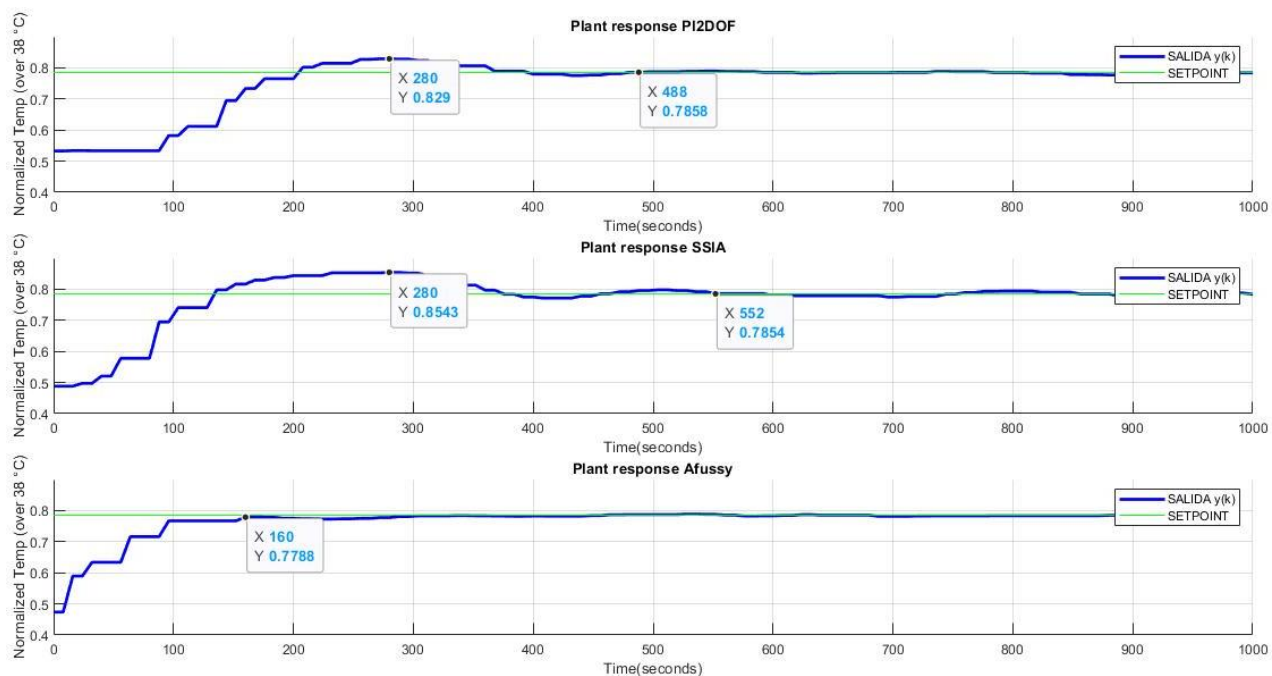
Tabla 8

Comparación de resultados de los controladores PI2DOF, SSIA Y AFuzzy implementados.

| Controladores | Sobre impulso (%) | Tiempo de establecimiento(s) | Tiempo de subida(s) | Consumo de energía |
|---------------|-------------------|------------------------------|---------------------|--------------------|
| PI2DOF        | 5,5               | 488                          | 136                 | 334.9              |
| SSIA          | 0.7               | 552                          | 80                  | 342.2              |
| AFuzzy        | 0                 | 160                          | 56                  | 252.4              |

Figura 62

Respuesta del Sistema ante los controladores PI2DOF, SSIA y AFuzzy implementados



## Capítulo VI. Conclusiones y recomendaciones

### Conclusiones

El desarrollo metodológico propuesto en este proyecto resultó ser satisfactorio, se mostró la factibilidad de la realización de estrategias de control avanzado a través de la generación de código y su implementación en dispositivos industriales de escala real.

La obtención del modelo matemático del sistema térmico fue eficiente, ya que en el proceso de validación se obtuvo un NRMSE del 97.76%.

El controlador PI 2DOF fue ajustado utilizando varios métodos de los cuales la sintonización a través de la herramienta PID Tunner cumplió con los parámetros de diseño esperados, mostrando mejor desempeño respecto a los otros métodos de sintonización.

De las estrategias de control avanzado desarrolladas, se observaron distorsiones entre simulación y tiempo en real en las dos primeras, sin embargo, el controlador AFuzzy mantuvo una correlación más directa entre simulación e implementación.

La generación de código mediante la herramienta de Matlab/Simulink resultó ser efectiva para la traslación de código hacia la plataforma de software CODESYS.

En el análisis de desempeño de las estrategias de control avanzado en tiempo real el controlador Afuzzy presentó el mejor desempeño vs las otras estrategias de control, tanto en seguimiento a referencia como en rechazo a las perturbaciones.

Respecto al consumo de energía el controlador PI 2DOF presenta menor consumo en simulación, sin embargo, en el entorno de tiempo real el controlador AFuzzy presenta un consumo menor en comparación a las otras estrategias de control implementadas.

## Recomendaciones

Durante el desarrollo de la metodología del diseño del controlador, no se debe perder de vista los aspectos conceptuales a tomar en cuenta para lograr una traslación de la solución matemática hacia el sistema físico real de manera efectiva.

Es importante para la implementación de los controladores, tomar en cuenta el tiempo de muestreo con el que se diseñaron, ya que inicialmente al escoger un valor menor al tiempo de retardo del sistema, se genera una respuesta inestable que no cumple con los criterios de desempeño esperada.

Para lograr un desempeño adecuado de la implementación de los controladores respecto al diseño, es necesario trabajar con las variables bajo una misma unidad, ya que si existe una diferencia dimensional importante se generan errores en el cálculo del error y saturaciones o carencias en el valor de la salida del controlador.

Se recomienda trabajar con valores normalizados en las variables del lazo de control, con ello se obtiene una respuesta que corresponde al comportamiento estimado en el diseño.

Para lograr una identificación adecuada del sistema y obtener un modelo con un error mínimo se recomienda restar la temperatura ambiente.

Dependiendo del tipo de controlador lógico programable en el que se pretenda embeber el código generado, se debe revisar los lenguajes de programación con el que trabaja el PLC para que la realización de la estrategia de control diseñada sea eficiente.

## Bibliografía

3S-Smart Software Solutions GmbH. (2010). *User Manual for PLC Programming with CodeSys 2.3*.

Kempen: 3S - Smart Software Solutions GmbH.

- Acosta, J. (2022). *Diseño y simulación de un sistema de generación y almacenamiento híbrido para el abastecimiento de energía a una comunidad aislada del Ecuador*. Sangolquí: Universidad de las Fuerzas Armadas ESPE.
- Alfaro, V., Arrieta, O., & Vilanova, R. (2009). Control de dos grados de libertad (2GdL) aplicados al "Benchmark" de sistemas para controladores PID. *Revista Iberoamericana de Automática e Informática Industrial*, 59-67.
- Aluisa, P. (2014). *Desarrollo de un sistema de identificación de procesos industriales en línea usando la plataforma Open-Source Arduino y Matlab/Simulink*. Sangolquí: Universidad de las Fuerzas Armadas ESPE.
- Aström, K., & Hägglund, T. (2009). *Control PID avanzado*. Madrid: Pearson Educación.
- Aström, K., & Wittenmark, B. (1997). *Computer Controlled Systems*. Mainland: Prentice Hall.
- Burns, R. (2001). *Advanced Control Engineering*. Oxford: Butterworth Heinemann.
- Calderón, J. C. (2008). *Análisis de desempeño de controladores adaptativos discretos en el espacio de estados*. Sangolquí: Universidad de las Fuerzas Armadas ESPE.
- Chabni, F., Rachid, T., & Benbouali, A. (2016). The Application of Fuzzy Control in Water Tank Level Using Arduino. *International Journal of Advanced Computer Science and Applications* , 261–265.
- Chennakesava, A. (2007). *Fuzzy Logic and Neural Networks, basic concepts and applications*. Andhra Pradesh: New Age.
- Douglas, B. (2021, Noviembre 30). *System Identification: Engineering Media*. Retrieved from Engineering Media: <https://engineeringmedia.com/videos>



- Eriksson, L., & Johansson, M. (2007). ID Controller Tuning Rules for Varying Time-Delay Systems. *Proceedings of the 2007 American Control Conference*, 619-622.
- Fadali, S., & Visioli, A. (2013). *Digital Control Engineering. Analysis and Design*. Oxford: Elsevier Inc.
- Feijoo, J., Chanchay, D., Llanos, J., & Ortiz, D. (2021). Advanced Controllers for Level and Temperature Process Applied to Virtual Festo MPS® PA Workstation. *Lect. Notes Electr. Eng.*, vol. 762 LNEE, 164-180.
- Ferri, D. (2019). *Implementación de algoritmos de control avanzados MPC en PLC's industriales*. Valencia: Universidad Politécnica de Valencia - UPV.
- García, L., & Giraldo, M. (2012). Controladores avanzados en PLC. *Revista Politécnica*, 57-66.
- Garrido, S. (1999). *Identificación, estimación y control de sistemas no lineales mediante RGO*. Leganés: Universidad Carlos III de Madrid.
- Interiano, E. (2013). *Control Automático: Ejemplo de control de un sistema térmico*. Tecnológico de Costa Rica.
- Johnson, A. (2010). *Diseño e implementación de estrategias de control avanzado sobre plataforma LabView para un sistema de calentamiento inductivo de fluidos*. Santiago de Chile: Universidad de Chile.
- Ko, J.-S., Huh, J.-H., & Kim, J.-C. (2019). Improvement of Temperature Control Performance of Thermoelectric Dehumidifier Used Industry 4.0 by the SF-PI Controller. *Processes* vol. 7, no. 2,, 1-23.
- Marlin, T. (2015). *Process Control, Designing Processes and Control Systems for Dynamic Performance*. Hamilton: McGraw-Hill.

- Martínez, A. (2011). *Sistemas de control automático. Identificación experimental de sistemas*.  
Universidad de Alicante.
- Matlab. (2022, Abril 16). *MathWorks*. Retrieved from <https://la.mathworks.com/products/sysid.html>
- Ogata, K. (2010). *Ingeniería de Control Moderna*. Madrid: Pearson.
- Phillips, C., Nagle, T., & Chakraborty, A. (2015). *Digital Control System Analysis and Design*. Edinburgh:  
Pearson.
- PID Controller Tuning. (2006, Junio 11). *PID Controller Tuning*. Retrieved from PID Controller Tuning:  
<http://bestune.50megs.com/>
- PID Tuner Controller. (2022, Junio 11). *PID Tuner Controller*. Retrieved from PID Tuner Controller:  
<https://pidtuner.com>
- Ponce, P., & Ramírez, F. (2010). *Intelligent Control Systems with LabVIEW*. New York: Springer.
- Portillo, E., & Ordoñez, J. (2020). Diseño e Implementación de un Controlador Difuso para Control de Frecuencia de un Motor en un PLC S7-1200. *18th LACCEI International Multi-Conference for Engineering, Education, and Technology*.
- Reyes, H., & Montaña, F. (2010). *Modelamiento y control digital de temperatura para horno eléctrico*.  
Bogotá: Pontificia Universidad Javeriana.
- Shakya, R., Rajanwal, K., Patel, S., & Dinkar, S. (2014). Design and Simulation of PD, PID and Fuzzy Logic Controller for Industrial Application . *International Journal of Information and Computation Technology* . , 363-368.
- Siemens. (2018, Agosto 24). *Support industry Siemens*. Retrieved from  
<https://support.industry.siemens.com/cs/mdm/109760968?c=113376438923&lc=el-GR>

Söderström, T., & Stoica, P. (1989). *System Identification*. Cambridge: Prentice Hall International.

The MathWorks, Inc. (2020, Abril 4). *MathWorks*. Retrieved from <https://la.mathworks.com/>

Turck. (2013). *Turck*. Retrieved from [www.turck.us/attachment/D301263.pdf](http://www.turck.us/attachment/D301263.pdf)

Turck. (2022, Junio 6). *Turck - Programmable Gateways*. Retrieved from

<https://www.turck.us/en/productgroup/Industrial%20Controls/Programmable%20Gateways>

Villota, E. (2009). *Control moderno y óptimo*. Lima: Universidad Nacional de Ingeniería.

Yang, Y., & Bian, H. (2012). Design and realization of fuzzy self-tuning PID water temperature controller based on PLC. *International Conference on Intelligent Human-Machine Systems and Cybernetics*, 3-6.

Zadeh, L. (1962). From circuit theory to system theory. *IRE 50*, 856-865.

## Apéndices