



UNIVERSIDAD DE LA FRONTERA
FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

**“DESARROLLO DE UN SISTEMA DE CONTROL BASADO EN LÓGICA DIFUSA EN UN ORDENADOR
DE PLACA REDUCIDA”**

**ACTIVIDAD DE TITULACIÓN, MODALIDAD PROYECTO DE TITULACIÓN
PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO**

PROFESOR GUÍA: ARTURO ANDRÉS PADILLA BERNEDO

GUSTAVO ENRIQUE VALENZUELA FUENTEALBA

2020

**DESARROLLO DE UN SISTEMA DE CONTROL BASADO EN LÓGICA DIFUSA EN UN ORDENADOR
DE PLACA REDUCIDA
GUSTAVO ENRIQUE VALENZUELA FUENTEALBA**

COMISIÓN EVALUADORA

ARTURO ANDRÉS PADILLA BERNEDO
Profesor Guía

EDUARDO IGNACIO DIEZ CIFUENTES
Académico Evaluador

GLORIA MILLARAY JULIA CURILEM SALDIAS
Académico Evaluador

**Calificación informe escrito :
Calificación presentación oral :
Calificación final :**

*Este trabajo se lo dedico a mis padres.
Todo lo que he logrado no sería posible
sin el amor, apoyo y consejos que me
han entregado durante toda mi vida.*

AGRADECIMIENTOS

Este trabajo de título es la culminación del camino que he recorrido durante mi vida universitaria, la cual estuvo llena de alegrías, aprendizaje, crecimiento personal, éxitos y también fracasos. Sin lugar a duda en este recorrido he vivido las experiencias más enriquecedoras en lo que va de mi vida.

Quiero agradecer a todas esas personas que estuvieron desde un principio y a los que conocí en el camino, pude aprender mucho de todos ustedes. A mi familia Enrique, Norma, Francisco y Paula, por estar siempre presente y por ser un pilar fundamental en mi vida. A mis amigos más cercanos Bernardino, Fernando y Germán que siempre estuvieron ahí en los momentos en que los necesitaba. A mi profesor guía Arturo Padilla por aceptar guiarme cuando le presenté este proyecto y por estar siempre dispuesto a ayudarme. En fin, a todas las personas que creyeron y confiaron en mí.

Muchas gracias a todos y cada uno de ustedes.

RESUMEN

La evolución tecnológica que se ha desarrollado en las últimas décadas ha dado lugar a una complejidad cada vez mayor en la automatización de procesos tanto en la industria como en las distintas áreas de investigación científica. Debido a esto, investigadores de diversas áreas han concentrado sus esfuerzos en proporcionar algoritmos simples y sencillos para hacer frente a esta creciente complejidad. El control basado en lógica difusa es una alternativa práctica para el control de sistemas complejos, ya que proporciona un método conveniente para construir controladores no-lineales emulando la toma de decisiones que realiza un experto humano.

En este trabajo de título se estudia el desarrollo de sistemas de control difuso que puedan ser aplicados a problemas reales complejos con un enfoque en sistemas térmicos. El desarrollo de este proyecto involucra la búsqueda de un modelo de un sistema térmico adecuado para el estudio, el diseño del controlador difuso para dicho modelo utilizando un algoritmo genético para su sintonización, y la simulación del proceso controlado en un ordenador de placa reducida Raspberry Pi 3 modelo B+. El controlador PID difuso desarrollado se comparó con un controlador PID convencional demostrando tener un desempeño superior para el control de temperatura de una planta no-lineal.

Índice de contenidos

Capítulo 1. Introducción	1
1.1. Objetivo general	2
1.2. Objetivos específicos	2
Capítulo 2. Antecedentes generales	3
2.1. Conceptos generales sobre sistemas de control	4
2.1.1 Controlador PID	5
2.1.2 Índices de desempeño	6
2.2. Algoritmos genéticos	8
2.2.1 Descripción general de los algoritmos genéticos	8
2.2.2 Conceptos fundamentales de los algoritmos genéticos	9
2.3. Lógica difusa	11
2.3.1 Conjuntos difusos	12
2.3.2 Funciones de pertenencia	14
2.3.3 Operaciones en conjuntos difusos	17
2.3.4 T -normas y T -conormas	19
2.3.5 Variables lingüísticas	21
2.3.6 Sistemas difusos basados en reglas	22
2.3.7 Modelo difuso lingüístico	23
2.4. Sistemas de control difuso	24
2.4.1 Inferencia en un controlador difuso	26
2.4.2 Fuzzificación	27
2.4.3 Agregación	27
2.4.4 Activación	28
2.4.5 Acumulación	28
2.4.6 Defuzzificación	28
Capítulo 3. Metodología	29
3.1. Búsqueda de modelos	30
3.2. Desarrollo de sistema de control utilizando un controlador PID	30
3.2.1 Discretización del controlador PID	30
3.3. Desarrollo de sistema de control utilizando lógica difusa	32
3.3.1 Estructura del controlador PID difuso	33
3.4. Optimización de los controladores	34
3.5. Comparación del controlador de lógica difusa con el controlador PID	34
3.6. Implementación en simulación en una Raspberry Pi	35
Capítulo 4. Resultados y discusión	36
4.1. Modelos de sistemas térmicos	37
4.1.1 Sistema térmico seleccionado	38
4.1.2 Modelo matemático del sistema térmico seleccionado	39
4.2. Formulación del problema de control	42
4.3. Funciones objetivo e hiperparámetros del algoritmo genético	42
4.4. Sintonización del controlador PID	43
4.5. Desarrollo del controlador difuso	44
4.5.1 PID difuso lineal	45
4.5.2 PID difuso no-lineal	48

4.5.3 Sintonización del controlador difuso	50
4.6. Comparación de los controladores	51
4.7. Implementación de sistemas de control difuso	53
4.7.1 Implementación de controlador difuso mediante <i>Lookup table</i>	53
4.7.2 Discretización e interpolación de <i>Lookup table</i>	54
4.7.3 Implementación en Raspberry Pi	56
Capítulo 5. Conclusiones	57
Nomenclatura	60
Bibliografía	61
Anexo A. Derivación de modelo térmico	63

Índice de tablas

Tabla 3.1 Características principales Raspberry Pi modelo 3 B+	35
Tabla 4.1 Hiperparámetros seleccionados para el algoritmo genético	43
Tabla 4.2 Parámetros óptimos del controlador PID	44
Tabla 4.3 Parámetros óptimos del controlador PID difuso	50
Tabla 4.4 Comparación PID y PID difuso	51
Tabla 4.5 Comparación de tiempo de respuesta	52

Índice de figuras

Figura 2.1 Diagrama de bloques de un proceso con un controlador en lazo cerrado	4
Figura 2.2 Operaciones Genéticas (Brunton and Kutz, 2019).	9
Figura 2.3 Diversidad de población en AGs	10
Figura 2.4. Ejemplo de representación de pertenencia clásica y difusa (Jantzen, 2013).	13
Figura 2.5 Función de pertenencia trapezoidal	14
Figura 2.6 Función de pertenencia en forma de campana	15
Figura 2.7 Función de pertenencia singleton	16
Figura 2.8 Función de pertenencia curva S-shaped	16
Figura 2.9 Función de pertenencia curva Z-shaped	17
Figura 2.10 Complemento de un conjunto difuso	18
Figura 2.11 Intersección de conjuntos difusos	18
Figura 2.12 Unión de conjuntos difusos	19
Figura 2.13. Controlador difuso en lazo cerrado (Passino and Yurkovich, 1998).	24
Figura 2.14 Proceso de inferencia de un controlador difuso (Jantzen, 2013)	26
Figura 3.1 Controlador difuso PD+I (Jantzen, 2013)	34
Figura 3.2. Raspberry Pi modelo 3 B+.	35
Figura 4.1 Ejemplo de un baño de agua de laboratorio comercial	38
Figura 4.2. Diagrama esquemático proceso térmico (Melba Mary and Marimuthu, 2009).	39
Figura 4.3 Comportamiento del coeficiente $c(y_k)$	41
Figura 4.4 Diferencia de la sintonización del controlador PID basado en la optimización de la función objetivo J_1 y J_2	44
Figura 4.5 Funciones de pertenencias para variables premisa.	46
Figura 4.6 Singletons de salida.	46
Figura 4.7 Superficie de control lineal.	47
Figura 4.8 Respuesta controlador difuso PD+I lineal	48
Figura 4.9 Funciones de pertenencia no-lineales para variables premisa.	49
Figura 4.10 Superficie de control no-lineal	50
Figura 4.11 Diferencia de la sintonización del controlador PID difuso basado en la optimización de la función objetivo J_1 y J_2	51
Figura 4.12 Comparación PID y PID difuso basado en J_1	52
Figura 4.13 Comparación controlador PID difuso LUT	55
Figura 4.14 Simulador de sistema de control PID difuso LUT en Python	56

Capítulo 1. Introducción

La evolución tecnológica que se ha desarrollado en las últimas décadas ha dado lugar a una complejidad cada vez mayor en la automatización de procesos tanto en la industria como en las distintas áreas de investigación científica. Esta complejidad se atribuye principalmente a la no-linealidad, a una dinámica insuficientemente definida y a la ausencia de información previa sobre los sistemas. La imprecisión, la incertidumbre y la ambigüedad de la información sobre los sistemas desempeñan un papel fundamental en el aumento de la complejidad de estas aplicaciones, esta alta complejidad provoca que las técnicas de control convencional sean difíciles de aplicar y tengan resultados indeseados (Azeem, 2012; Jantzen, 2013).

Un ejemplo de un proceso automatizado complejo es el secado de madera con un sistema deshumidificador, la complejidad de este sistema se manifiesta de la siguiente manera: (i) el proceso de secado es complejo, variable en el tiempo y no-lineal; (ii) no es posible representar adecuadamente todos los parámetros que exhiben un comportamiento de interacción; (iii) algunas variables de secado (por ejemplo, calidad del producto y color) no se pueden medir directamente; otras mediciones (por ejemplo, contenido de humedad) pueden ser inconsistentes, imprecisas, incompletas o no totalmente confiables; (iv) según la búsqueda exhaustiva realizada por el autor, el único modelo que existe actualmente disponible para caracterizar el proceso de secado por deshumidificación (Sun and Carrington, 1999) es sumamente complejo y no es viable utilizarlo para el desarrollo de un controlador. En la industria del secado de madera, la estrategia de control generalmente se basa en la anticipación, el conocimiento práctico del proceso de secado y la experiencia personal de los operadores. Debido a lo mencionado anteriormente, para poder desarrollar un sistema de control para este proceso es necesario poseer conocimientos de control automático avanzados.

Otro tipo de sistemas con una complejidad un poco menor al secado de madera son los sistemas térmicos. Por sistema térmico entenderemos un sistema cerrado que contiene un calefactor que permite aumentar o disminuir la temperatura de dicho sistema. Este tipo de sistema provoca problemas para la sintonización de controladores PID debido que exhiben un comportamiento dinámico no simétrico (Babuska and Kober, 2018).

El control basado en lógica difusa es una alternativa práctica para este tipo de aplicaciones, ya que proporciona un método conveniente para construir controladores no-lineales emulando la toma de decisiones que realiza un experto humano en base a su conocimiento de cómo

controlar de mejor forma el proceso. Esta técnica de control es relativamente reciente y está basada en la teoría de conjuntos difusos, disciplina matemática que fue introducida por Lotfi Zadeh (Zadeh, 1965). En 1974 se reportó la primera aplicación exitosa de la lógica difusa en control (Mamdani, 1974), causando un gran interés y desarrollo desde entonces, teniendo un gran éxito en múltiples aplicaciones.

En este proyecto se estudia el desarrollo de sistemas de control difuso que puedan ser aplicados a problemas reales complejos como por ejemplo el secado de madera y control de temperatura. El desarrollo del sistema de control involucra el análisis y diseño en el software MATLAB; además, involucra la implementación en un ordenador de placa reducida que es validada mediante simulación. El sistema de control difuso desarrollado será comparado con un controlador PID.

Con esto se busca incentivar el estudio de este tipo de técnicas para facilitar el análisis y control de sistemas complejos, además de mostrarlo como una alternativa útil de aplicar en aplicaciones industriales, de emprendimiento y de investigación.

1.1. Objetivo general

- Implementar en simulación un sistema de control basado en la técnica de control de lógica difusa en una Raspberry Pi.

1.2. Objetivos específicos

- Desarrollar un sistema de control utilizando un controlador PID.
- Desarrollar un sistema de control utilizando lógica difusa.
- Comparar el controlador de lógica difusa con el controlador PID.
- Implementar en simulación en una Raspberry Pi el sistema de controlador basado en lógica difusa.

Capítulo 2. Antecedentes generales

2.1. Conceptos generales sobre sistemas de control

El control automático ha desempeñado un papel vital en el avance de la ingeniería y la ciencia. El control automático se ha convertido en una parte importante e integral en los sistemas de vehículos espaciales, en los sistemas robóticos, en los procesos modernos de fabricación y en cualquier operación industrial que requiera el control de temperatura, presión, humedad, flujo, etc. (Ogata, 2010).

Un sistema que mantiene una relación determinada entre la salida y la entrada de referencia, comparándolas y usando la diferencia como medio de control, se denomina sistema de control realimentado, denominado también sistema de control en lazo cerrado. En la **Figura 2.1** se muestra un sistema de realimentación sencillo mediante un diagrama de bloques. El sistema tiene dos componentes principales, el proceso y el controlador, representados como cajas con flechas que denotan la relación causal entre entradas y salidas. El proceso tiene una entrada, la variable manipulada, también llamada variable de control denotada por $u(t)$. La variable de control influye sobre el proceso vía un actuador. La salida del proceso se representa por $y(t)$. Esta variable se mide con un sensor; en la figura el actuador y el sensor se consideran parte del bloque “proceso”. El valor deseado de la variable de proceso se llama el punto de consigna o valor de referencia, en la figura denotada por $r(t)$. El error de control $e(t)$ es la diferencia entre valor de referencia y la variable de proceso, es decir, $e(t) = r(t) - y(t)$ (Aström and Hägglund, 2009).

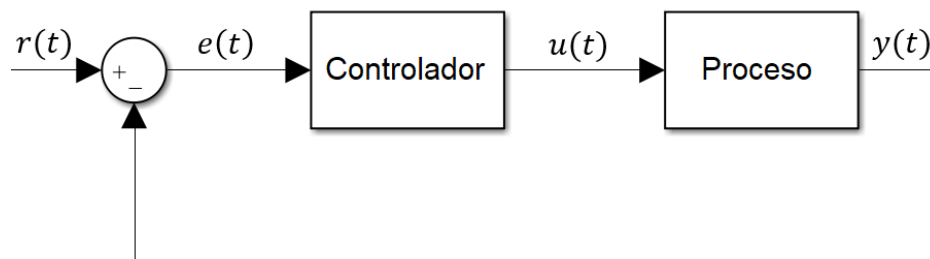


Figura 2.1 Diagrama de bloques de un proceso con un controlador en lazo cerrado

2.1.1 Controlador PID

El controlador PID es una implementación simple de la idea de realimentación. Tiene la capacidad de eliminar errores en estado estacionario mediante la acción integral, y puede anticipar el futuro con la acción derivativa. Los controladores PID, o incluso los controladores PI, son suficientes para muchos problemas de control, particularmente cuando las dinámicas del proceso son benignas y los requisitos de comportamiento son modestos. Los controladores PID se encuentran en un gran número en todas las industrias y se presentan de muchas formas diferentes. Es interesante señalar que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID o PID modificado (Ogata, 2010).

La ley de control de un controlador PID se representa por la siguiente ecuación:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.1)$$

donde los parámetros K_p , K_i y K_d corresponden a la ganancia proporcional, integral y derivativa respectivamente.

La señal de control es la suma de los términos:

- Proporcional (P): proporciona una acción de control proporcional al error.
- Integral (I): proporciona una acción de control que es proporcional a la integral del error. Este término permite eventualmente garantizar cero error en estado estacionario.
- Derivativo (D): proporciona una acción de control proporcional a la derivada en el tiempo del error. Este término permite adelantarse en la acción de control.

2.1.2 Índices de desempeño

Un índice de desempeño es una medida cuantitativa del desempeño de un sistema y se elige de manera que se haga hincapié en las especificaciones importantes del sistema (Dorf, Richard C and Bishop, 2011). Para ser útil, un índice de desempeño debe ser un número que siempre es positivo o cero. Usualmente están en función del error $e(t) = r(t) - y(t)$, donde $r(t)$ es la señal de referencia e $y(t)$ la salida medida. A continuación, se describen algunos de los índices de desempeño más utilizados.

➤ **ISE-Integral Squared Error**

Es el valor de la integral del error al cuadrado en un periodo de tiempo determinado $t \in (t_1, t_2)$.

$$ISE = \int_{t_1}^{t_2} e(t)^2 dt \quad (2.2)$$

El índice ISE penaliza altamente los errores con valores grandes en oposición a los errores pequeños. Los valores grandes del error de control suelen producirse inmediatamente después de una perturbación y pueden observarse como sobrepaso; es por esto que este índice se utiliza principalmente para indicar el sobrepaso y el control agresivo.

➤ **IAE-Integral Absolute Error**

Es el valor de la integral del error absoluto en un periodo de tiempo determinado $t \in (t_1, t_2)$.

$$IAE = \int_{t_1}^{t_2} |e(t)| dt \quad (2.3)$$

El índice IAE no distingue entre las contribuciones positivas y negativas del error de control.

➤ **ITAE-Integral of Time multiplied Absolute Error**

Es el valor de la integral del error de control absoluto multiplicado por el tiempo en el período de tiempo determinado $t \in (t_1, t_2)$.

$$ITAE = \int_{t_1}^{t_2} t|e(t)|dt \quad (2.4)$$

La multiplicación por el tiempo refuerza los errores que persisten durante un tiempo prolongado. Pondera fuertemente los errores más grandes que ocurren tarde y hace menos énfasis en los errores de control iniciales.

➤ **ITSE-Integral of Time multiplied Squared Error**

Es el valor de la integral del error al cuadrado multiplicado por el tiempo en el periodo de tiempo determinado $t \in (t_1, t_2)$.

$$ITSE = \int_{t_1}^{t_2} te(t)^2 dt \quad (2.5)$$

El índice ITSE al igual que el ITAE refuerza los errores que persisten durante un tiempo prolongado y además penaliza los errores con valores grandes en oposición a los errores pequeños.

➤ **ISCO-Integral Squared Control Output**

Los índices de desempeño descritos anteriormente están en función de la señal de error $e(t)$, sin embargo en algunos casos es útil considerar algún índice de desempeño que se pueda relacionar con la señal de control $u(t)$. El índice ISCO es el valor de la integral de la señal de control al cuadrado en un periodo de tiempo determinado $t \in (t_1, t_2)$.

$$ISCO = \int_{t_1}^{t_2} u(t)^2 dt \quad (2.6)$$

2.2. Algoritmos genéticos

Los algoritmos genéticos (AGs) son una clase particular de algoritmos evolutivos que utilizan técnicas basadas en la selección natural, proceso que impulsa la evolución biológica según la teoría de la evolución de Darwin. Los AGs se utilizan en el campo de la robótica, la planificación de estrategias, los sistemas dinámicos no lineales, el análisis de datos, el arte, la música y muchas otras aplicaciones en el mundo real. Los AGs proporcionan una flexibilidad y robustez únicas para la optimización de procesos (Jesus and Barbosa, 2015). Usualmente los AGs son utilizados para resolver una variedad de problemas de optimización que no son adecuados para los algoritmos de optimización estándar, incluyendo problemas en los que la función objetivo es discontinua, no diferenciable, estocástica o altamente no lineal. Los AGs pueden ser utilizados para ajustar y adaptar los parámetros de un controlador (Brunton and Kutz, 2019).

2.2.1 Descripción general de los algoritmos genéticos

En los AGs una población de individuos, llamada generación, compiten en una tarea determinada con una función de costo bien definida. El AG modifica repetidamente una población de soluciones individuales. En cada paso, se selecciona al azar individuos de la población actual para que sean padres y se utilizan para producir los hijos de la próxima generación. A lo largo de las sucesivas generaciones, la población "evoluciona" hacia una solución óptima. Los individuos exitosos tienen una mayor probabilidad de ser seleccionados para avanzar a la siguiente generación, de acuerdo con las siguientes operaciones genéticas:

- **Elitismo:** Un número determinado de los individuos más aptos con el mejor desempeño se adelantan directamente a la siguiente generación.
- **Replicación:** Un individuo es seleccionado para avanzar a la siguiente generación.
- **Cruce:** Dos individuos son seleccionados para intercambiar una porción de su código y luego avanzar a la siguiente generación; el cruce sirve para aprovechar y mejorar las estrategias exitosas existentes.
- **Mutación:** Un individuo es seleccionado para que se modifique una porción de su código con nuevos valores; la mutación promueve la diversidad y sirve para aumentar la exploración del espacio de los parámetros.

La **Figura 2.2** muestra un esquema que ilustra la evolución en un algoritmo genético. Los individuos de la generación k son evaluados y clasificados en orden ascendente según su función de costo, que es inversamente proporcional a su probabilidad de selección para las operaciones genéticas. Luego, los individuos son escogidos en base a esta probabilidad ponderada para avanzar a la generación $k + 1$ utilizando las cuatro operaciones: elitismo, replicación, cruce y mutación. De esta manera se forma la generación $k + 1$, y la secuencia se repite hasta que las cifras de la población convergen o se alcanza otro criterio de parada adecuado.

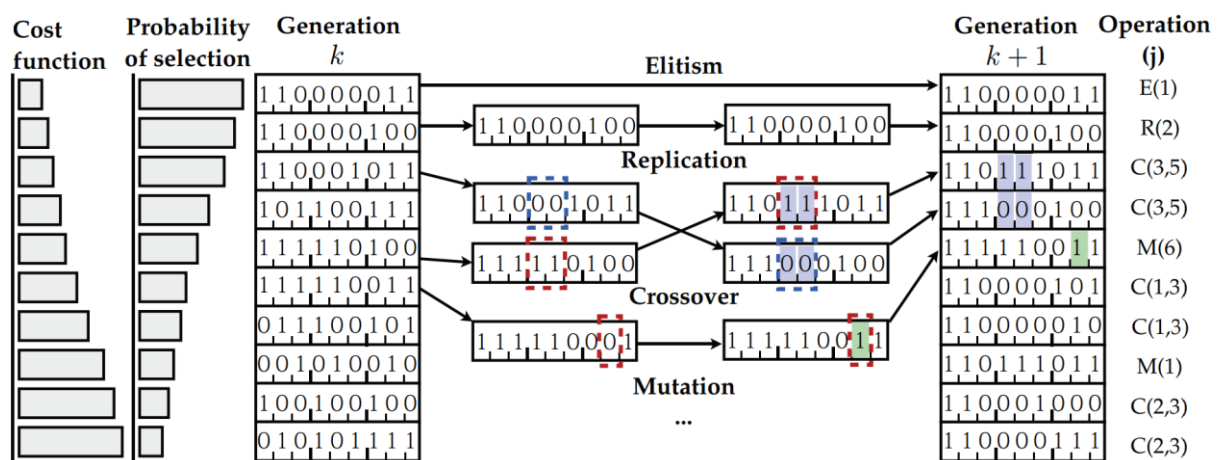


Figura 2.2 Operaciones Genéticas (Brunton and Kutz, 2019).

2.2.2 Conceptos fundamentales de los algoritmos genéticos

A continuación, se describen algunos de los conceptos fundamentales de los algoritmos genéticos (MathWorks, 2020):

➤ Función de aptitud (*fitness*)

La función de aptitud, o función de costo, es la función que se quiere optimizar. En los algoritmos de optimización estándar, la función de aptitud se conoce como función objetivo.

➤ Individuos

Un individuo es cualquier punto al que se puede aplicar la función de aptitud. El valor de la función para un individuo es su puntuación. Por ejemplo, si la función objetivo es

$$J(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - 2)^2 \quad (2.7)$$

el vector $[2, -3, 1]$, cuya longitud es el número de variables del problema, es un individuo. La puntuación del individuo $[2, -3, 1]$ es $J(2, -3, 1) = 51$.

➤ Poblaciones y generaciones

Una población es un conjunto de individuos. Por ejemplo, si el tamaño de la población es 100 y el número de variables en la función de aptitud es 3, se representa la población por una matriz de 100×3 . En cada iteración, el algoritmo genético realiza una serie de cálculos sobre la población actual para producir una nueva población. Cada población sucesiva se llama una nueva generación.

➤ Diversidad

La diversidad se refiere a la distancia media entre los individuos de una población. La diversidad es esencial para el AG porque permite al algoritmo buscar en una región más grande del espacio. En la **Figura 2.3**, la población de la izquierda tiene una gran diversidad, mientras que la población de la derecha tiene una baja diversidad.

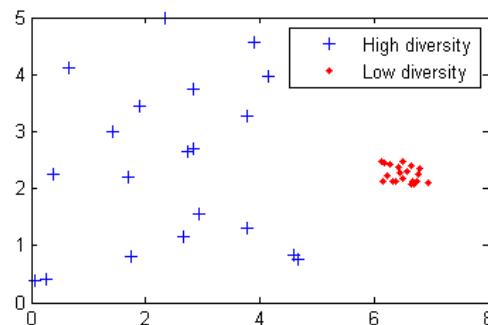


Figura 2.3 Diversidad de población en AGs

➤ Padres e hijos

Para crear la próxima generación, el algoritmo genético selecciona ciertos individuos de la población actual, llamados padres, y los usa para crear individuos en la próxima generación, llamados hijos. Típicamente, es más probable que el algoritmo seleccione a los padres que tienen mejores valores de aptitud.

2.3. Lógica difusa

Para comprender que es el control difuso, primero es necesario definir algunos aspectos generales sobre la lógica difusa, ya que este es el principio matemático en que se basa este tipo de controladores.

En las últimas décadas se han desarrollado diferentes técnicas que buscan facilitar la construcción de máquinas y sistemas basados en los mecanismos de razonamiento humano, algunos ejemplos de ello son los sistemas expertos, las redes neuronales artificiales y los sistemas basados en lógica difusa.

La lógica difusa, al igual que los sistemas expertos y las redes neuronales, pueden obtener la salida de un sistema en función de sus entradas sin necesidad de recurrir a la utilización de un modelo analítico. La lógica difusa es similar a los sistemas expertos en la forma que usan para representar el conocimiento de forma estructurada mediante reglas, y también se puede comparar a las redes neuronales por la facilidad para manipular la información de forma numérica. Esto convierte a la lógica difusa en una alternativa muy atractiva para la implementación de sistemas de control en distintas áreas.

La teoría de conjuntos difusos desarrollada por L. A. Zadeh a mediados de la década de los 60, proporciona una herramienta adecuada para modelar la incertidumbre presente en el lenguaje del ser humano y emular los mecanismos de razonamiento aproximado utilizados por nuestro cerebro.

En esta sección se presentan los aspectos matemáticos fundamentales para el desarrollo de un controlador basado en lógica difusa.

2.3.1 Conjuntos difusos

El concepto fundamental en el que se basa la lógica difusa es el conjunto difuso, un tipo de conjunto en el cual la pertenencia de sus elementos está definida por valores en el intervalo unitario $[0,1]$. Los conjuntos difusos pueden ser considerados una generalización de los conjuntos clásicos (Zimmermann, 1996).

En la teoría de conjuntos clásica, los elementos pertenecen plenamente a un conjunto o están totalmente excluidos de él. La pertenencia $\mu_A(x)$ de x de un conjunto clásico A , como un subconjunto del universo X , se define por:

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (2.8)$$

esto significa que un elemento x es un miembro del conjunto A ($\mu_A(x) = 1$) o no ($\mu_A(x) = 0$). Esta estricta clasificación es útil en las matemáticas y otras ciencias que se basan en definiciones precisas.

Un conjunto difuso es un conjunto con una pertenencia en el intervalo real $\mu_A(x) \in [0,1]$. Es decir, los elementos pueden pertenecer a un conjunto difuso en un cierto grado. Como tal, los conjuntos difusos pueden utilizarse para representaciones matemáticas de conceptos imprecisos. Un conjunto difuso A en el universo X es un conjunto definido por la función de pertenencia $\mu_A(x)$ que es una asignación del universo X en el intervalo unitario:

$$\mu_A(x): X \rightarrow [0,1] \quad (2.9)$$

Si el valor de la función de pertenencia, llamado grado de pertenencia, es igual a 1, x pertenece completamente al conjunto difuso. Si es igual a 0, x no pertenece al conjunto. Si el grado de pertenencia está entre 0 y 1, x es un miembro parcial del conjunto difuso:

$$\mu_A(x) = \begin{cases} = 1 & x \text{ pertenece completamente a } A \\ \in (0,1) & x \text{ pertenece parcialmente a } A \\ = 0 & x \text{ no pertenece a } A \end{cases} \quad (2.10)$$

Para comprender de mejor forma la diferencia entre la lógica clásica y difusa para resolver problemas que impliquen imprecisión, es mejor visualizarlo con un ejemplo sencillo.

Consideremos un aire acondicionado con un controlador termostático que reconoce sólo dos estados: por encima de la temperatura deseada (21°C , por ejemplo), o por debajo de la temperatura deseada. Utilizando lógica clásica el aire acondicionado considera que cualquier temperatura por encima de 21°C es cálida. La lógica difusa, por otro lado, permite valores de veracidad intermedios entre verdadero y falso, el aire acondicionado en este caso considera que las temperaturas son gradualmente más cálidas. En la **Figura 2.4** se representan las funciones de pertenencia utilizando ambos enfoques. La línea segmentada simboliza la pertenencia de la lógica clásica, donde se asignan valores nítidos (*crisp*) que pueden ser 0 ó 1. Por otra parte, la pertenencia difusa (*fuzzy*) se representa con la línea continua, donde la función toma valores en el intervalo $[0,1]$.

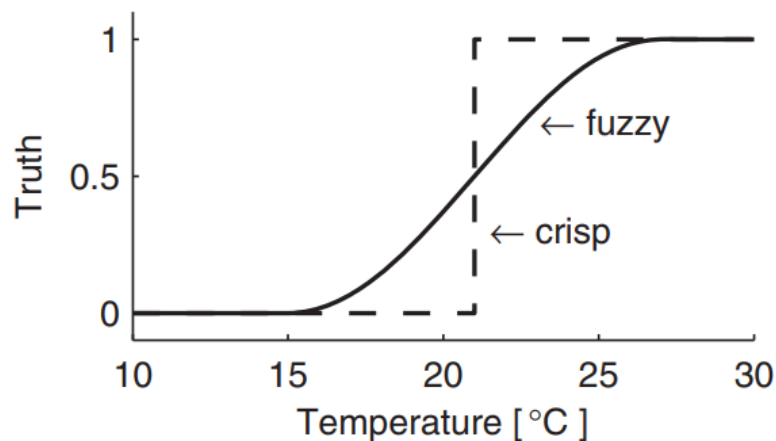


Figura 2.4. Ejemplo de representación de pertenencia clásica y difusa (Jantzen, 2013).

Muchas clases o conjuntos tienen límites difusos en lugar de nítidos, y esta es la base matemática de la lógica difusa: el conjunto de medidas de temperatura "cálida" es un ejemplo de un conjunto difuso (Jantzen, 2013).

2.3.2 Funciones de pertenencia

Existen varias maneras de definir un conjunto difuso. El método más común e intuitivo es por medio de una descripción analítica de su función de pertenencia (Babuska and Kober, 2018). Estas definiciones analíticas permiten describir la forma de estas funciones. A continuación, se presentan algunos tipos de funciones utilizadas frecuentemente:

➤ **Función de pertenencia trapezoidal:**

$$\mu(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (2.11)$$

donde a , b , c y d son las coordenadas de los vértices del trapecioide. Cuando $b = c$, se obtiene una función de pertenencia triangular. La función de pertenencia trapezoidal se muestra en la **Figura 2.5**.

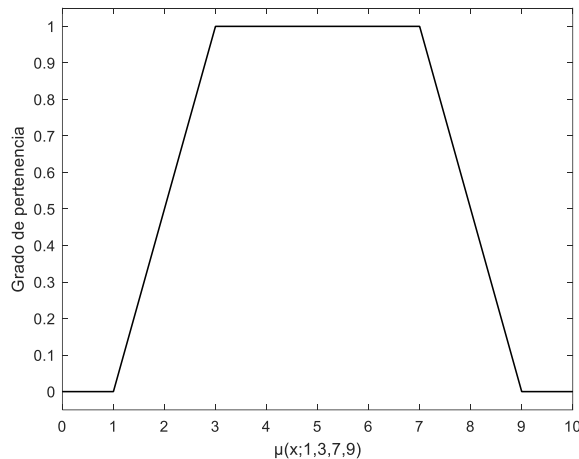


Figura 2.5 Función de pertenencia trapezoidal

➤ **Función de pertenencia en forma de campana**

$$\mu(x; a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (2.12)$$

donde los parámetros a y b suelen ser positivos y el parámetro c localiza el centro de la curva. La función de pertenencia en forma de campana se muestra la **Figura 2.6**.

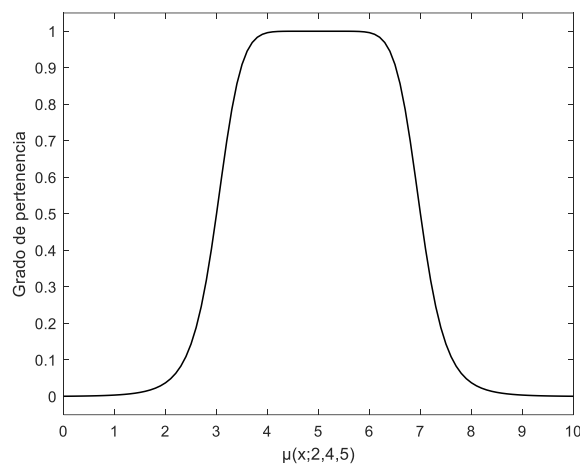


Figura 2.6 Función de pertenencia en forma de campana

➤ **Función de pertenencia Singleton:**

$$\mu(x) = \begin{cases} 1, & x = x_0 \\ 0, & \text{en otro caso.} \end{cases} \quad (2.13)$$

La función de pertenencia singleton es la representación difusa de un número real. El valor x_0 representa su ubicación en el universo. La representación gráfica del singleton se muestra en la **Figura 2.7**.

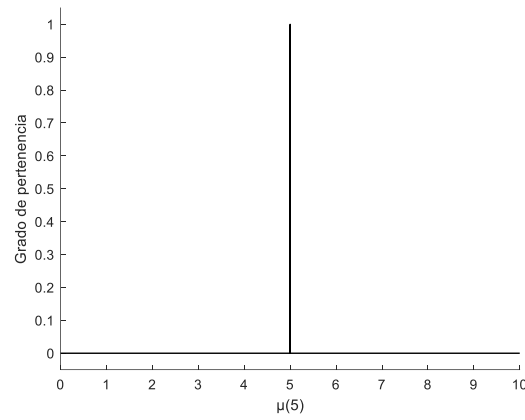


Figura 2.7 Función de pertenencia singleton

➤ **Función de pertenencia curva “S-shaped”:**

$$\mu(x; a, b) = \begin{cases} 0, & x \leq a \\ 2 \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2 \left(\frac{x-a}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases} \quad (2.14)$$

La curva *S-shaped* depende de los parámetros a y b que localizan los extremos del tramo inclinado de la curva. La representación gráfica de esta función de pertenencia se muestra en la **Figura 2.8**.

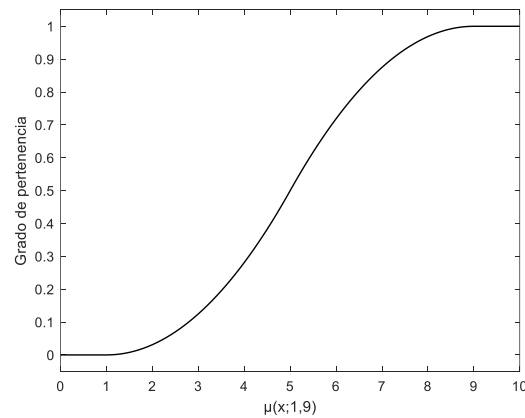


Figura 2.8 Función de pertenencia curva *S-shaped*

➤ **Función de pertenencia curva “Z-shaped”:**

$$\mu(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2 \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2 \left(\frac{x-b}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b \end{cases} \quad (2.15)$$

La curva *Z-shaped* es la versión decreciente de la curva *S-shaped*, depende de los parámetros a y b que localizan los extremos del tramo inclinado de la curva. La representación gráfica de esta función de pertenencia se muestra en la **Figura 2.9**.

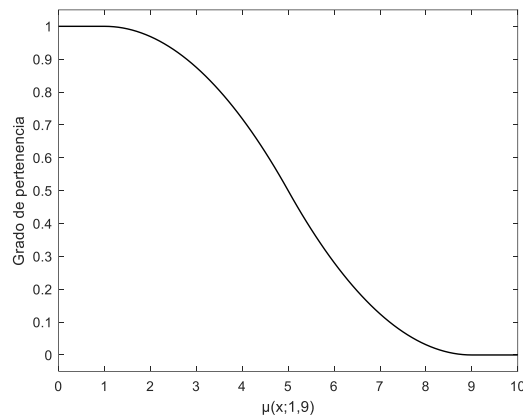


Figura 2.9 Función de pertenencia curva *Z-shaped*

2.3.3 Operaciones en conjuntos difusos

Todas las operaciones de los conjuntos convencionales han sido generalizadas a los conjuntos difusos. De hecho, cuando se realiza una operación de conjunto difuso en un conjunto convencional, se obtiene el resultado correspondiente al conjunto convencional. En esta sección se presentan las tres operaciones básicas presentadas por Zadeh (Zadeh, 1965): complemento, unión e intersección.

- **Complemento de un conjunto difuso:** Sea A un conjunto difuso en X . El complemento de A es un conjunto difuso, denotado \bar{A} , tal que para cada $x \in X$:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.16)$$

En la **Figura 2.10** se muestra la representación gráfica del complemento difuso desde el punto de vista de una función de pertenencia.

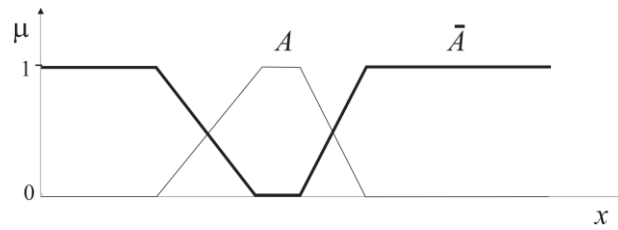


Figura 2.10 Complemento de un conjunto difuso

- **Intersección de conjuntos difusos:** Sea A y B dos conjuntos difusos en X . La intersección de A y B es un conjunto difuso C , denotado $A \cap B$, tal que para cada $x \in X$:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.17)$$

El operador mínimo también se puede denotar por ' \wedge ', i.e., $\mu_C(x) = \mu_A(x) \wedge \mu_B(x)$. En la **Figura 2.11** se muestra su representación gráfica.

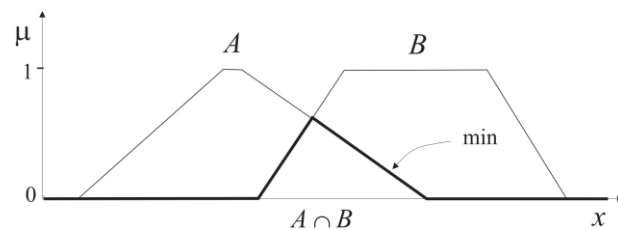


Figura 2.11 Intersección de conjuntos difusos

- **Unión de conjuntos difusos:** Sea A y B dos conjuntos difusos en X . La unión de A y B es un conjunto difuso C , denotado $A \cup B$, tal que para cada $x \in X$:

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.18)$$

El operador máximo también se puede denotar por ' \vee ', i.e., $\mu_C(x) = \mu_A(x) \vee \mu_B(x)$. En la **Figura 2.12** se muestra su representación gráfica.

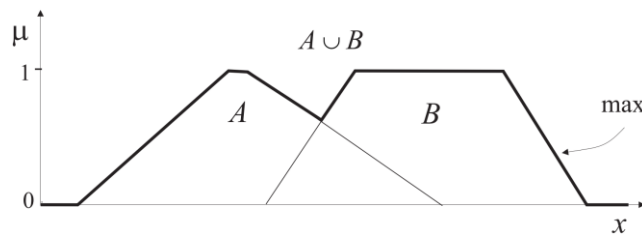


Figura 2.12 Unión de conjuntos difusos

2.3.4 T -normas y T -conormas

La intersección de dos conjuntos difusos puede especificarse de manera más general mediante una operación binaria en un intervalo unitario, es decir, una función de la forma:

$$T: [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (2.19)$$

Para que una función T pueda calificarse como una intersección difusa, debe poseer propiedades apropiadas. Las funciones conocidas como t -normas (normas triangulares) poseen las propiedades necesarias para la intersección. Del mismo modo, las funciones denominadas t -conormas pueden utilizarse para la unión difusa.

Una t -norma T es una operación binaria en el intervalo unitario que satisface al menos los siguientes axiomas para todo $a, b, c \in [0, 1]$ (Klir and Yuan, 1995):

1. Condición de contorno: $T(a, 1) = a$
2. Monotonicidad: $b \leq c \Rightarrow T(a, b) \leq T(a, c)$
3. Conmutatividad: $T(a, b) = T(b, a)$
4. Asociatividad: $T(a, T(b, c)) = T(T(a, b), c)$

Algunas de las t -normas más utilizadas son:

➤ **Intersección standard (Zadeh):**

$$T(a, b) = \min(a, b) \quad (2.20)$$

➤ **Producto algebraico (intersección probabilística):**

$$T(a, b) = ab \quad (2.21)$$

➤ **Intersección de Łukasiewicz:**

$$T(a, b) = \max(0, a + b - 1) \quad (2.22)$$

Una t -conorma S es una operación binaria en el intervalo unitario que satisface al menos los siguientes axiomas para todo $a, b, c \in [0, 1]$ (Klir and Yuan, 1995).

1. Condición de contorno: $S(a, 0) = a$
2. Monotonicidad: $b \leq c \Rightarrow S(a, b) \leq S(a, c)$
3. Conmutatividad: $S(a, b) = S(b, a)$
4. Asociatividad: $S(a, S(b, c)) = S(S(a, b), c)$

Algunas de las t -conormas más utilizadas son:

➤ **Unión standard (Zadeh):**

$$S(a, b) = \max(a, b) \quad (2.23)$$

➤ **Suma algebraica (unión probabilística):**

$$S(a, b) = a + b - ab \quad (2.24)$$

➤ **Unión de Łukasiewicz:**

$$S(a, b) = \min(1, a + b) \quad (2.25)$$

2.3.5 Variables lingüísticas

Una variable lingüística es una variable cuyo valor se representa con palabras en lugar de números. En las aplicaciones de control, las variables medidas se consideran variables lingüísticas. Por ejemplo, en un enunciado como "la temperatura es caliente", estamos diciendo que la variable lingüística temperatura tiene el valor lingüístico caliente. El término o valor lingüístico caliente es un conjunto difuso en el universo del discurso de las temperaturas. En general, una variable lingüística con el universo de discurso X puede tomar varios valores lingüísticos.

Los términos lingüísticos pueden considerarse como valores cualitativos que se utilizan para describir una relación particular mediante reglas lingüísticas. Típicamente, un conjunto de N términos lingüísticos $A = \{A_1, A_2, \dots, A_N\}$ se define en el dominio de una variable x dada. Debido a que esta variable asume valores lingüísticos, se llama variable lingüística. Para distinguir entre la variable lingüística y la variable numérica original, a esta última se le llama variable base.

Una variable lingüística L se define como la quintupla (5-tupla) (Klir and Yuan, 1995):

$$L = (x, A, X, g, m) \quad (2.26)$$

donde x es la variable base (al mismo tiempo el nombre de la variable lingüística), $A = \{A_1, A_2, \dots, A_N\}$ es el conjunto de términos lingüísticos, X es el dominio (universo del discurso) de x , g es una regla sintáctica para la generación de términos lingüísticos y m es una regla semántica que asigna a cada término lingüístico su significado (un conjunto difuso en X).

2.3.6 Sistemas difusos basados en reglas

Una vez definidas las variables y valores lingüísticos, se pueden formular las reglas del sistema de inferencia difusa. Se denomina sistema difuso a un sistema estático o dinámico que utiliza conjuntos difusos y su correspondiente estructura matemática. Los conjuntos difusos pueden estar involucrados en un sistema de varias maneras, tales como:

- En la descripción del sistema: Un sistema puede definirse, por ejemplo, como una serie de reglas del tipo **IF – THEN** (SI-ENTONCES).
- En la especificación de los parámetros del sistema: El sistema puede definirse mediante una ecuación algebraica o diferencial, en la que los parámetros son números difusos en lugar de números reales.
- Las variables de entrada, salida y estados de un sistema pueden ser conjuntos difusos. Las entradas difusas pueden ser lecturas de sensores poco fiables (datos "ruidosos"), o cantidades relacionadas con la percepción humana, como la comodidad, la belleza, etc.

Los sistemas difusos pueden servir para diferentes propósitos, como la modelación, el análisis de datos, la predicción o el control.

El sistema difuso más común es el basado en reglas del tipo **IF – THEN**, este tipo de sistema se denomina sistema difuso basado en reglas y tiene la forma general:

"SI (IF) proposición antecedente ENTONCES (THEN) proposición consecuente"

Las proposiciones difusas son afirmaciones como " x es grande", donde "grande" es un valor lingüístico definido por un conjunto difuso en el universo de una variable x . La proposición antecedente es siempre una proposición difusa del tipo " x es A ", donde x es una variable

lingüística y A es un valor lingüístico. Dependiendo de la estructura particular de la proposición consecuente, existen tres tipos principales de modelos (Babuska and Kober, 2018):

- Modelo difuso lingüístico, donde tanto el antecedente como el consecuente son proposiciones difusas.
- Modelo relacional difuso, que puede considerarse como una generalización del modelo lingüístico, que permite asociar una proposición antecedente particular con varias proposiciones consecuentes diferentes mediante una relación difusa.
- El modelo difuso de Takagi-Sugeno (TS), donde el consecuente es una función nítida de las variables antecedentes en lugar de una proposición difusa.

2.3.7 Modelo difuso lingüístico

El modelo lingüístico (Zadeh, 1973; Mamdani, 1977) fue presentado como una forma de captar el conocimiento cualitativo en forma de reglas del tipo **IF – THEN**. Una regla R_i de un modelo difuso lingüístico tiene la forma:

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } y \text{ is } B_i \quad i = 1, 2, \dots, K. \quad (2.27)$$

aquí x es la variable lingüística de entrada (antecedente), y A_i son los términos lingüísticos de entrada. Del mismo modo, y es la variable lingüística de salida (consecuente) y B_i son los términos lingüísticos consecuentes. Los valores de x e y son generalmente conjuntos difusos, pero dado que un número real es un caso especial de un conjunto difuso (conjunto singleton), estas variables también pueden tener un valor real. Los términos lingüísticos A_i y B_i son siempre conjuntos difusos.

Un caso especial del modelo difuso lingüístico se obtiene cuando los conjuntos difusos consecuentes B_i son conjuntos del tipo singleton. Estos conjuntos pueden ser representados como números reales b_i , dando lugar a las siguientes reglas:

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } y = b_i \quad i = 1, 2, \dots, K. \quad (2.28)$$

2.4. Sistemas de control difuso

La principal motivación para recurrir a un sistema de control difuso es encontrarse con un sistema el cual es sumamente complejo de modelar, e incluso si se llega a desarrollar un modelo relativamente preciso de la dinámica del sistema, este suele ser demasiado complejo para ser utilizado en el desarrollo del controlador, principalmente porque muchos procedimientos de diseño de control convencional requieren supuestos restrictivos para la planta, por ejemplo, la linealidad del modelo (Passino and Yurkovich, 1998).

El diagrama de bloques del controlador difuso se presenta en la **Figura 2.13**, donde se tiene un controlador difuso incorporado en un sistema de control de lazo cerrado. La salida de la planta se denota con $y(t)$, su entrada se denota con $u(t)$, y la entrada de referencia al controlador difuso se denota con $r(t)$.

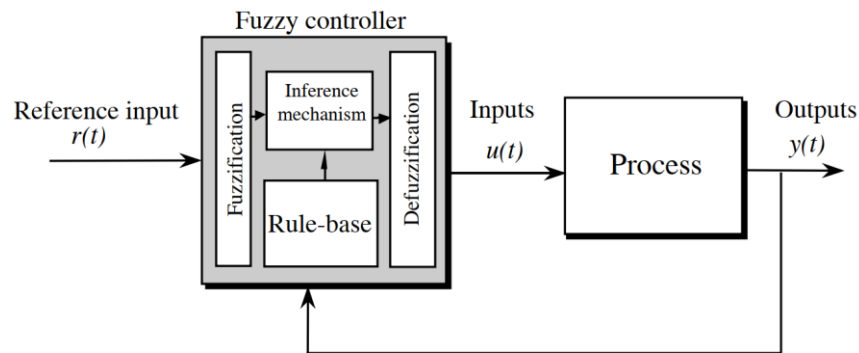


Figura 2.13. Controlador difuso en lazo cerrado (Passino and Yurkovich, 1998).

El controlador difuso tiene cuatro componentes principales:

1. La "base de reglas" sostiene el conocimiento, en forma de un conjunto de reglas que resumen la mejor manera de controlar el sistema.
2. El mecanismo de inferencia evalúa qué reglas de control son relevantes en el momento actual y luego decide cuál debe ser la entrada de la planta.
3. La interfaz de fuzzificación simplemente modifica las entradas para que puedan ser interpretadas y comparadas con las reglas de la base de reglas.

4. La interfaz de defuzzificación convierte las conclusiones obtenidas por el mecanismo de inferencia en entradas para la planta.

Para diseñar el controlador difuso, el ingeniero de control debe recopilar información sobre cómo debe actuar una persona que toma las decisiones en el sistema de lazo cerrado. A veces esta información puede provenir de un humano que toma decisiones y que realiza la tarea de control, mientras que en otras ocasiones el ingeniero de control puede llegar a comprender la dinámica de la planta y escribir un conjunto de reglas sobre cómo controlar el sistema sin ayuda externa. Estas "reglas" dicen básicamente:

*"SI la salida de la planta y la entrada de referencia se comportan de cierta manera, **ENTONCES** la entrada de la planta debería tener algún valor".*

Un conjunto completo de tales reglas **"IF – THEN"** se introducen en la base de reglas, y se elige una estrategia de inferencia, entonces el sistema está listo para ser probado para ver si se cumplen las especificaciones de circuito cerrado.

Las variables del sistema son de dos tipos, las variables de entrada medidas desde el proceso de control y las variables de salida utilizadas por el sistema difuso para controlar el proceso. Dependiendo de los objetivos del diseño, se pueden construir diferentes tipos de controladores. Por ejemplo, se puede tener un número fijo de reglas de control difuso (una base de conocimientos difusos estática) o puede tener capacidad de aprendizaje mediante la modificación de la base de conocimientos (una base de conocimientos difusos dinámica).

Una regla permite utilizar varias variables tanto en la premisa como en la conclusión. Por lo tanto, un controlador puede ser del tipo MIMO (*Multiple-input Multiple-output*) o SISO (*Single-input Single-output*). El controlador SISO típico regula una señal de control según una señal de error. Un controlador puede utilizar el error, el cambio de error y el error integral, pero aun así se considera un controlador SISO, ya que el cambio del error y su integral se derivan del error, y el bucle es un lazo de retroalimentación único.

2.4.1 Inferencia en un controlador difuso

Para realizar una explicación sencilla del proceso de inferencia que ejecuta internamente un sistema basado en lógica difusa se utilizará un ejemplo. La **Figura 2.14** muestra una representación gráfica del proceso de inferencia difusa, donde cada una de las cuatro filas superiores representa una regla.

Consideremos la siguiente base de reglas:

- **IF** error is *NEG* and *change in error* is *NEG* **THEN** control is *NEG*
- **IF** error is *NEG* and *change in error* is *POS* **THEN** control is *ZERO*
- **IF** error is *POS* and *change in error* is *NEG* **THEN** control is *ZERO*
- **IF** error is *POS* and *change in error* is *POS* **THEN** control is *POS*

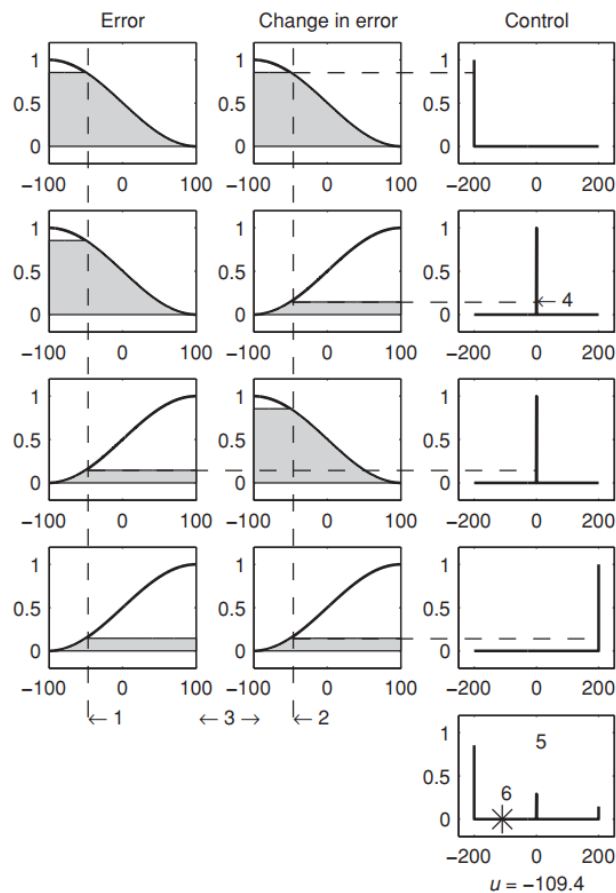


Figura 2.14 Proceso de inferencia de un controlador difuso (Jantzen, 2013)

Notar que en este caso se utilizaron singletons como conclusión, sin embargo, es posible utilizar cualquier otro tipo de función de pertenencia. Hay por lo menos tres ventajas en el uso de singletons en lugar de conjuntos difusos completos en el lado de la conclusión:

1. Los cálculos son más sencillos.
2. Es posible llevar la señal de control a los extremos del universo.
3. Es natural para un operador pensar en términos de la magnitud de la señal.

El mecanismo de inferencia sigue un procedimiento de cálculo por pasos. A continuación, se realiza la descripción de cada uno de estos pasos.

2.4.2 Fuzzificación

Las mediciones del error y el cambio de error se indican mediante las líneas segmentadas verticales que cruzan la primera y la segunda columna del gráfico a partir de los puntos 1 y 2 marcados en la **Figura 2.14**. Para cada regla, el mecanismo de inferencia busca el valor de pertenencia difusa, que es donde la línea vertical intersecta una función de pertenencia. Esto se denomina fuzzificación, ya que asocia un valor de pertenencia difuso a partir de una medida nítida.

2.4.3 Agregación

El grado de cumplimiento α_i de la regla i se produce por el valor de pertenencia difuso $\mu_{i,1}$ correspondiente a la medición del error, y un valor de pertenencia $\mu_{i,2}$ correspondiente al cambio en la medición del error. Su agregación, indicada en el punto 3 de la **Figura 2.14**, es la intersección de ambos.

$$\alpha_i = \mu_{i,1} \wedge \mu_{i,2} \quad (2.29)$$

El operador \wedge combina las dos proposiciones en la base de reglas; en general, podría ser una combinación de muchas proposiciones conectadas por \wedge o \vee . En la **Figura 2.14**, \wedge se implementa como la operación *min*.

2.4.4 Activación

La activación de una regla es la derivación de una conclusión en función del grado de cumplimiento α_i , tal como se indica en el punto 4 de la **Figura 2.14**. Sólo se activa una porción de cada singleton. Generalmente, se utiliza como operador de activación el mínimo o la multiplicación. Cuando se utilizan singletons como conclusión el resultado es el mismo, pero en general, la multiplicación mantiene la forma de la función de pertenencia de salida, mientras que el mínimo la recorta.

2.4.5 Acumulación

El último gráfico de la parte inferior derecha, indicado por el punto 5 de la **Figura 2.14**, es la acumulación de todas las conclusiones activadas. En la **Figura 2.14** se utiliza la acumulación-suma, pero alternativamente se puede utilizar una acumulación-máx. La activación por multiplicación seguida de la acumulación-suma da como resultado la función de pertenencia:

$$\mu_c(b_i) = [(b_1, \alpha_1), (b_2, \alpha_2 + \alpha_3), (b_4, \alpha_4)] \quad (2.30)$$

donde b_i es el singleton de la proposición consecuente de la regla R_i (ver (2.28)).

2.4.6 Defuzzificación

El conjunto difuso μ_c resultante debe convertirse en un número nítido para obtener una señal de control para la planta. Este proceso es conocido como defuzzificación. En la figura, la señal de control defuzzificada está marcada con un asterisco. Existen diversos métodos de defuzzificación; dos de los métodos de defuzzificación más utilizados son: el centro de gravedad (COG) y la media de los máximos (habitualmente llamado MOM). En el caso de que las conclusiones sean singletons la media de los máximos y el centro de gravedad producen el mismo resultado. Su valor es un promedio ponderado:

$$u = \frac{\alpha_1 b_1 + \alpha_2 b_2 + \alpha_3 b_3 + \alpha_4 b_4}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4} \quad (2.31)$$

Capítulo 3. Metodología

3.1. Búsqueda de modelos

Para el diseño de un controlador se puede trabajar directamente con el sistema real o bien en simulación mediante un modelo del sistema. En este trabajo se considera la segunda opción. La idea central de este proyecto es estudiar el desarrollo de sistemas de control difuso que puedan ser aplicados a problemas reales complejos con un enfoque en sistemas térmicos. Es por esto que en primer lugar se realizará una búsqueda en la literatura de un modelo de un sistema térmico apropiado para este estudio. Además, se estudiarán las técnicas de modelado existentes con el fin de seleccionar un modelo de complejidad adecuada para poder alcanzar exitosamente los objetivos que han sido planteados.

3.2. Desarrollo de sistema de control utilizando un controlador PID

En la metodología del diseño de controladores difusos usualmente se recomienda comenzar abordando el problema utilizando las herramientas de control clásicas, como lo es el controlador de tipo PID (Passino and Yurkovich, 1998). El controlador PID es bien conocido, fácil de implementar tanto en su forma digital como analógica y es ampliamente utilizado. Sin embargo, puede presentar problemas cuando se trabaja con modelos no-lineales.

En esta etapa del proyecto el enfoque principal es diseñar y sintonizar un controlador PID clásico, con el objetivo de visualizar bajo qué condiciones y en qué medida este tipo de control muestra un mal comportamiento. La herramienta de diseño utilizada en esta etapa fue el software MATLAB.

3.2.1 Discretización del controlador PID

Dado que se desea realizar una implementación digital, será necesario trabajar con un PID discreto. A continuación, se muestra el procedimiento utilizado para realizar la discretización del controlador PID.

La función de transferencia en tiempo continuo del controlador PID es:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (3.1)$$

En presencia de ruido de alta frecuencia, la acción derivativa causa incrementos no deseados en la señal de control. En este trabajo no se consideró la presencia de ruido. Comúnmente se utilizan filtros pasa bajas para disminuir este efecto, por lo que, para implementaciones reales se recomienda tener en cuenta este aspecto.

La discretización de la ecuación (3.1) puede realizarse reemplazando el término s por su equivalencia en el dominio z , ésta equivalencia dependerá del tipo de aproximación que se utilice para los términos integral y derivativo. En este caso, para el término integral la aproximación se realizó utilizando el método Euler hacia adelante. Para el término derivativo la aproximación utilizada fue mediante el método Euler hacia atrás. Ambas equivalencias se muestran a continuación:

➤ **Euler hacia adelante:**

$$s = \frac{z - 1}{T_s} \quad (3.2)$$

➤ **Euler hacia atrás:**

$$s = \frac{z - 1}{zT_s} \quad (3.3)$$

sustituyendo (3.2) y (3.3) en (3.1):

$$G_c(z) = K_p + \frac{K_i}{\left(\frac{z-1}{T_s}\right)} + K_d \left(\frac{z-1}{zT_s}\right) \quad (3.4)$$

$$G_c(z) = K_p + \frac{K_i T_s}{z-1} + \frac{K_d}{T_s} \left(\frac{z-1}{z}\right) \quad (3.5)$$

$$G_c(z) = \frac{\left[K_p(z^2 - z) + K_i T_s z + \frac{K_d}{T_s} (z^2 - 2z + 1) \right]}{z^2 - z} \quad (3.6)$$

$$G_c(z) = \frac{\left[\left(K_p + \frac{K_d}{T_s}\right)z^2 + \left(-K_p + K_i T_s - \frac{2K_d}{T_s}\right)z + \frac{K_d}{T_s}\right]}{z^2 - z} \quad (3.7)$$

$$\frac{u(z)}{e(z)} = \frac{\left[\left(K_p + \frac{K_d}{T_s}\right) + \left(-K_p + K_i T_s - \frac{2K_d}{T_s}\right)z^{-1} + \frac{K_d}{T_s}z^{-2}\right]}{1 - z^{-1}} \quad (3.8)$$

despejando $u(z)$:

$$u(z) = z^{-1}u(z) + ae(z) + bz^{-1}e(z) + cz^{-2}e(z) \quad (3.9)$$

donde:

$$a = K_p + \frac{K_d}{T_s} \quad (3.10)$$

$$b = -K_p + K_i T_s - \frac{2K_d}{T_s} \quad (3.11)$$

$$c = \frac{K_d}{T_s} \quad (3.12)$$

aplicando la transformada z inversa obtenemos la ecuación en diferencias:

$$u_k = u_{k-1} + ae_k + be_{k-1} + ce_{k-2} \quad (3.13)$$

3.3. Desarrollo de sistema de control utilizando lógica difusa

Una vez acabado el diseño del controlador PID, el siguiente paso a seguir es realizar el diseño del sistema de control difuso. Al igual que para el controlador PID, la herramienta computacional de diseño fue MATLAB.

Desde que se comenzó a utilizar la lógica difusa en sistemas de control se han desarrollado diversas estructuras y procedimientos para el diseño de este tipo de controladores, cada una con sus pros y contras. Para este trabajo se utilizó una metodología de diseño descrita en (Jantzen, 2013), donde se plantea comenzar diseñando un PID convencional para posteriormente realizar la transición hacia un controlador PID difuso.

Una gran ventaja de esta metodología es que se asegura que el funcionamiento del controlador difuso es como mínimo igual a un PID. Este procedimiento de diseño se constituye de los siguientes pasos:

1. Construir y sintonizar un controlador PID convencional.
2. Sustituirlo por un controlador difuso lineal equivalente.
3. Convertir el controlador difuso en uno no-lineal.
4. Sintonizar.

A su vez, el diseño de un controlador difuso sigue la siguiente estructura general:

1. Determinar las entradas y salidas del controlador difuso.
2. Definir las funciones de pertenencia.
3. Diseñar la base de reglas.
4. Sintonizar el controlador.
5. Comprobar su funcionamiento.

3.3.1 Estructura del controlador PID difuso

Un controlador PID difuso puede realizarse con tres entradas: error, integral del error y derivada del error. Con estas tres variables de entrada y, por ejemplo, tres términos lingüísticos para cada una de ellos, la base de reglas completa constaría de $3^3 = 27$ reglas, lo que hace que su diseño sea engorroso, además es difícil establecer reglas relativas a la acción integral. Sin embargo, un diseño simple y frecuentemente utilizado es combinar la acción integral del PID clásico y una base de reglas PD difusa lo que se conoce como controlador difuso PD+I como se muestra en la **Figura 3.1**.

De acuerdo a la **Figura 3.1**, las entradas a la base de reglas difusa PD son el error (e) y su derivada $\left(\frac{de}{dt}\right)$. Ambas entradas se multiplican por las ganancias GE y GCE de manera de asegurar entradas normalizadas (E y CE) en el universo del discurso que se haya definido para cada variable. Se trabaja de manera similar con la integral $\int e \, dt$. Ésta se multiplica por GIE lo que resulta en la señal IE , valor que se suma a la salida del controlador difuso obteniendo la salida total u , este valor es multiplicado por GU para obtener U que es finalmente la señal de control que se aplica al actuador. Esta estructura permite que GE , GCE , GIE y GU puedan ser tratados como parámetros de sintonización del controlador, que es el enfoque que se utilizó en este trabajo.

Otra opción es sintonizar las funciones de pertenencia, es decir, modificar la cantidad, la forma y la posición en el universo de tal manera de poder obtener mejores resultados. Sin embargo, con ese enfoque también se vuelve necesario modificar la base de reglas. Por lo que comúnmente se utilizan algoritmos avanzados para realizar este trabajo.

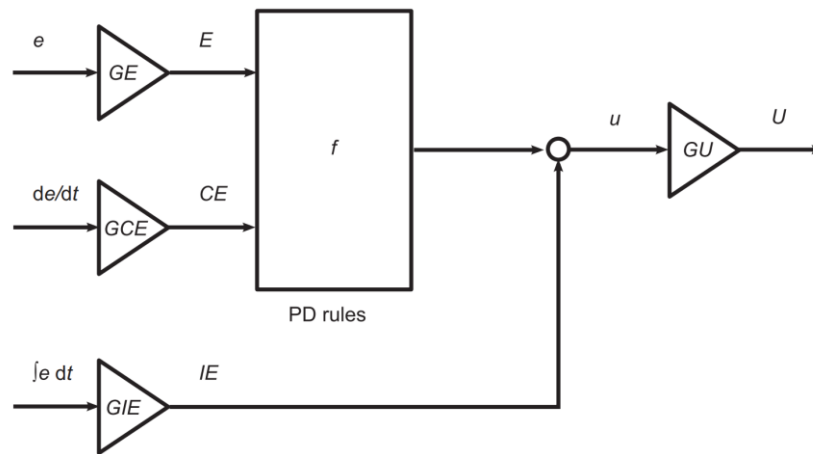


Figura 3.1 Controlador difuso PD+I (Jantzen, 2013)

3.4. Optimización de los controladores

Para poder realizar una comparación cuantitativa entre los sistemas de control PID y PID difuso se realizará una optimización de los parámetros de los controladores utilizando algoritmos genéticos, obteniéndose así controladores óptimos. Se considera que un sistema de control es óptimo cuando los parámetros del sistema se ajustan de manera que una cierta función de costo o índice alcance un extremo, comúnmente un valor mínimo. De esta forma, el sistema de control óptimo se define como el sistema que minimiza este índice.

3.5. Comparación del controlador de lógica difusa con el controlador PID

Una vez desarrollado de forma exitosa ambos controladores, se procederá a realizar una comparación mediante MATLAB utilizando distintos criterios de desempeño y de esta forma observar el comportamiento de los controladores PID y PID difuso mediante simulaciones. Este paso es clave, y aquí se verá que tan adecuado es el uso de la lógica difusa para el control de sistemas no-lineales.

3.6. Implementación en simulación en una Raspberry Pi

Ya teniendo el diseño final del sistema de control difuso, se implementará el sistema de control difuso en una Raspberry Pi. Una Raspberry Pi es un pequeño ordenador que posee pines IO que permiten conectar hardware personalizado. El modelo utilizado para realizar este trabajo corresponde a una Raspberry Pi 3 B+ (**Figura 3.2**) que será programada en lenguaje Python. Las principales especificaciones de este dispositivo se muestran en la **Tabla 3.1**.

Tabla 3.1 Características principales Raspberry Pi modelo 3 B+

Característica	Detalle
Procesador	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
Frecuencia de reloj	1,4 GHz
Memoria	1GB LPDDR2 SDRAM
Conectividad inalámbrica	2.4GHz / 5GHz IEEE 802.11.b/g/n/ac Bluetooth 4.2, BLE
Puertos	GPIO 40 pines HDMI 4 x USB 2.0 Micro SD Micro USB (alimentación) Power-over-Ethernet (PoE)

En esta etapa se estudiarán las alternativas existentes para implementar de forma exitosa el sistema de control difuso. La implementación será validada considerando un proceso simulado.



Figura 3.2. Raspberry Pi modelo 3 B+.

Capítulo 4. Resultados y discusión

4.1. Modelos de sistemas térmicos

Se realizó una búsqueda bibliográfica para seleccionar un modelo de sistema térmico adecuado para el estudio de un controlador difuso. Un tipo de sistema térmico complejo en el cual se han realizado una gran cantidad de investigaciones son los sistemas HVAC (*Heating, Ventilation, and Air-conditioning*), esto debido a que se busca que estos sistemas sean lo más eficiente posible, siendo el sistema de control un elemento clave para cumplir dicho objetivo.

Los sistemas HVAC poseen una estructura compleja, que consiste en una serie de equipos que involucran procesos altamente no-lineales de transferencia de calor y masa. Se pudo observar que una de las grandes dificultades que conlleva el control de estos sistemas es el proceso de modelación matemática. Debido a esta dificultad se han realizado diversos estudios para evaluar distintas técnicas de modelación, con el fin de poder encontrar modelos que sean adecuados para el desarrollo de sistemas de control.

Las técnicas de modelización de un sistema pueden dividirse en tres categorías: basadas en la física (caja blanca), basadas en datos (caja negra) e híbrida (caja gris). En el ámbito de la transferencia de calor, los modelos caja blanca se desarrollan a partir de leyes fundamentales de balance de masa, energía y momentum, de donde se pueden derivar y resolver un conjunto de ecuaciones matemáticas. Los modelos caja negra se elaboran mediante un proceso de recolección de datos sobre el comportamiento del sistema a partir de pruebas experimentales y luego se establece una relación entre las variables de entrada y de salida utilizando diferentes técnicas matemáticas. Los modelos de caja gris pueden considerarse como la combinación de los modelos de caja blanca y caja negra. La estructura básica del modelo se forma a partir de métodos basados en la física, mientras que los parámetros del modelo se determinan mediante el uso de algoritmos de estimación de parámetros sobre los datos medidos del sistema (Afroz *et al.*, 2018).

4.1.1 Sistema térmico seleccionado

Por la complejidad descrita en la Sección 4.1 se decidió buscar un sistema térmico más sencillo para poder lograr los objetivos de este trabajo. El sistema seleccionado para realizar el diseño del controlador corresponde a un dispositivo denominado baño de agua de laboratorio, también conocido como “baño maría” (**Figura 4.1**). Algunas de sus aplicaciones consisten en incubar muestras celulares o calentar productos químicos inflamables que pueden entrar en combustión si se exponen a una llama directa. En estas aplicaciones se requiere trabajar con precisión a temperaturas constantes durante largos períodos de tiempo. Este sistema se ha utilizado como referencia para el desarrollo de distintos tipos de controladores (Tanomaru and Omatu, 1992; Chin-Teng Lin, Chia-Feng Juang and Chung-Ping Li, 1999; Li and Lee, 2003; Melba Mary and Marimuthu, 2009).



Figura 4.1 Ejemplo de un baño de agua de laboratorio comercial

Este trabajo está basado en un baño de agua experimental utilizado en (Chin-Teng Lin, Chia-Feng Juang and Chung-Ping Li, 1999). Éste consta de cinco componentes principales: un recipiente de 6 litros, un agitador, un controlador de temperatura, un módulo de sensor termopar y un calefactor de 1000 W basado en un relé de estado sólido (SSR). En la **Figura 4.2** se muestra un diagrama esquemático del proceso.

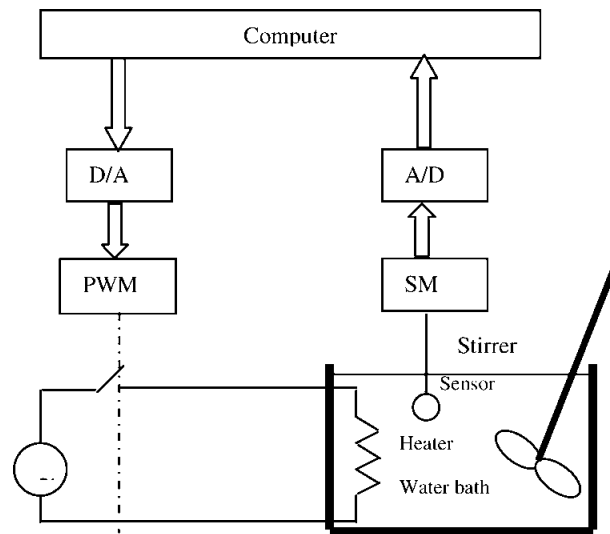


Figura 4.2. Diagrama esquemático proceso térmico (Melba Mary and Marimuthu, 2009).

4.1.2 Modelo matemático del sistema térmico seleccionado

El modelo del sistema térmico en tiempo continuo se describe mediante la siguiente expresión (Melba Mary and Marimuthu, 2009):

$$\frac{dy(t)}{dt} = \frac{f(t)}{C} + \frac{Y_0 - y(t)}{RC} \quad (4.1)$$

La Ecuación (4.1) se obtiene a partir de un balance de energía, por lo que corresponde a un modelo del tipo caja blanca. Aquí t denota el tiempo en s , $y(t)$ la temperatura del agua en $^{\circ}C$, $f(t)$ el calor en W que fluye hacia el interior del sistema, Y_0 la temperatura ambiente en $^{\circ}C$ que se asume constante para simplificar el modelo, C la capacidad térmica del sistema en $J \cdot ^{\circ}C^{-1}$ y R la resistencia térmica en $^{\circ}C \cdot W^{-1}$ entre los límites del sistema y sus alrededores. Se supone además que R y C son constantes.

Considerando un retentor de orden cero, se puede demostrar (ver **Anexo A. Derivación de modelo térmico**) que el sistema en tiempo discreto es:

$$y_{k+1} = a(T_S)y_k + b(T_S)f_k + [1 - a(T_S)]Y_0 \quad (4.2)$$

donde k es el nr. de muestra, f_k e y_k denotan la entrada y la salida del sistema, respectivamente, y T_s el tiempo de muestreo.

Los parámetros están dados por:

$$a(T_s) = e^{-\alpha T_s} \quad (4.3)$$

y

$$b(T_s) = \frac{\beta}{\alpha} (1 - e^{-\alpha T_s}) \quad (4.4)$$

donde α y β son valores constantes que dependen de R y C .

Para que el modelo físico de la Ecuación (4.2) pueda utilizarse en un sistema de control, el calor f_k se reemplaza por el voltaje u_k que corresponderá a la variable manipulada que afectará la variable de control y_k . Por otra parte, se desea considerar en el modelo un límite para la temperatura máxima; en la práctica, la temperatura máxima está limitada pues la capacidad del elemento calefactor es finita. Debido a lo anteriormente expuesto, el modelo de la Ecuación (4.2) se modifica de la siguiente manera:

$$y_{k+1} = a(T_s)y_k + b(T_s)c(y_k)u_k + [1 - a(T_s)]Y_0 \quad (4.5)$$

donde

$$c(y_k) = \frac{1}{1 + e^{0.5y_k - \gamma}} \quad (4.6)$$

La presencia del coeficiente $c(y_k)$ constituye una no-linealidad en el modelo. Notar que la representación (4.5) corresponde a un modelo *quasi-LPV* (*LPV: linear parameter varying*), donde la variable de programación es y_k (Tóth, 2010).

En el sistema real considerado en (Tanomaru and Omatu, 1992) el voltaje varía de 0 a 5 V. De esta forma el calefactor se activa de manera proporcional al voltaje u_k ; por ejemplo, si se requiere que el calefactor trabaje al 60%, la señal u_k debe ser un 60% de 5 V, es decir, 3 V.

Los valores de los parámetros del modelo (4.5) determinados en (Tanomaru and Omatu, 1992; Li and Lee, 2003) son: $\alpha = 1.00151E - 4$, $\beta = 8.67973E - 3$ y $\gamma = 40$. En la **Figura 4.3** se observa la variación $c(y_k)$ en función de y_k . Considerando la estructura del modelo (4.5) y el comportamiento del parámetro $c(y_k)$, se aprecia que para temperaturas mayores a $\approx 70^\circ\text{C}$, el efecto del calefactor se hace menos importante a medida que la temperatura aumenta. En efecto, para temperaturas sobre 100°C , el aumento del voltaje no genera un aumento de la temperatura.

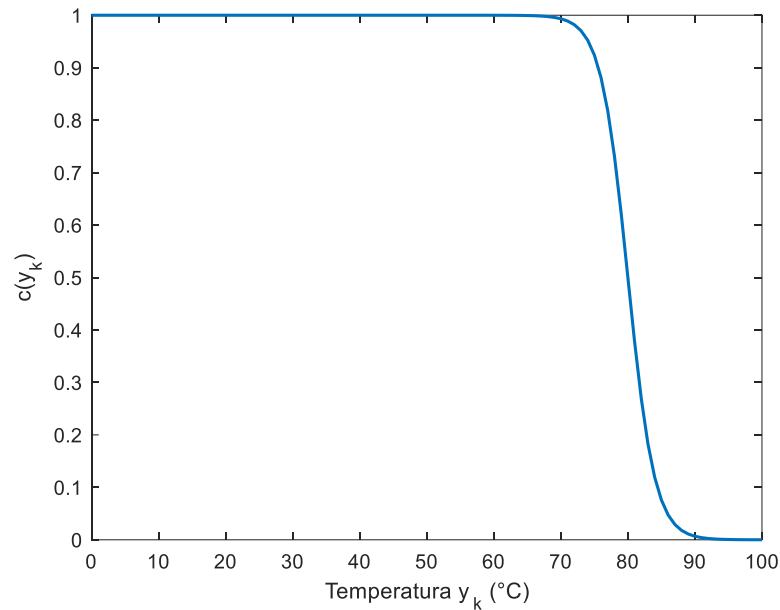


Figura 4.3 Comportamiento del coeficiente $c(y_k)$

El tiempo de muestreo debe elegirse adecuadamente, como se indica en (Tanomaru and Omatu, 1992; Li and Lee, 2003) debe ser $T_s \geq 10$ s para esta aplicación en particular. En (Melba Mary and Marimuthu, 2009) se encontró que para $T_s = 25$ s se logra un excelente desempeño, mostrando una respuesta ante una entrada escalón que se comporta de una manera casi ideal, por lo que en este trabajo se considera ese valor.

4.2. Formulación del problema de control

Como se comentó en la Sección 4.1.2, el modelo del sistema térmico (4.5) corresponde a un modelo lineal para temperaturas inferiores a $\approx 70^\circ\text{C}$; para temperaturas mayores a $\approx 70^\circ\text{C}$ el modelo es no-lineal. Por ello se propone realizar el diseño del sistema de control considerando los siguientes dos cambios de referencia:

1. Para el régimen de operación lineal, cambio de referencia escalón de 50 a 65 $^\circ\text{C}$
2. Para el régimen de operación no-lineal, cambio de referencia escalón de 65 a 80 $^\circ\text{C}$

Se considera una situación donde la temperatura inicial es 50 $^\circ\text{C}$ y la temperatura ambiente es $Y_0 = 25^\circ\text{C}$. Finalmente, la señal de referencia utilizada para la simulación es:

$$r(t) = \begin{cases} 65^\circ\text{C}, & \text{para } 0\text{ h} < t \leq 1\text{ h} \\ 80^\circ\text{C}, & \text{para } 1\text{ h} < t \leq 2\text{ h} \end{cases} \quad (4.7)$$

De esta forma la sintonización del sistema de control se realiza solo para el cambio de referencia propuesto.

4.3. Funciones objetivo e hiperparámetros del algoritmo genético

Para efectuar la optimización es necesario contar con una función objetivo que debe ser formulada de manera tal que se pueda lograr el desempeño deseado. Esta función puede ser directamente alguno de los índices que se introdujeron en la Sección 2.1.2, o bien pueden ser una combinación de estos. Se formularon las siguientes funciones objetivo para el ajuste de los controladores, esto para poder comparar la sintonización que otorga la minimización de cada una de estas funciones:

$$J_1(\theta) = \int_{t_1}^{t_2} [w_1 t |e(t)| + w_2 u(t)^2] dt = (w_1 \times ITAE) + (w_2 \times ISCO) \quad (4.8)$$

$$J_2(\theta) = \int_{t_1}^{t_2} [w_1 t e(t)^2 + w_2 u(t)^2] dt = (w_1 \times ITSE) + (w_2 \times ISCO) \quad (4.9)$$

donde θ corresponde al vector de parámetros a optimizar. Las ponderaciones $\{w_1, w_2\}$ en las funciones objetivo dan una flexibilidad extra al diseñador, dependiendo de la naturaleza de la aplicación y la relevancia de tener un bajo índice de error y una baja señal de control (Das *et al.*, 2012).

Debido a la complejidad que trae consigo poder obtener parámetros que sean adecuados para todo el rango de operación de la planta, la optimización mediante algoritmos genéticos fue realizada por tramos, es decir, se buscaron los parámetros óptimos para la referencia de 65 °C en $0\text{ h} < t \leq 1\text{ h}$ y posteriormente para 80°C en $1\text{ h} < t \leq 2\text{ h}$, tanto para el PID como para el PID difuso.

Los hiperparámetros utilizados en el algoritmo genético para las optimizaciones realizadas en este trabajo se muestran en la **Tabla 4.1**. Estos valores fueron seleccionados después de varias pruebas; y basándose en estudios similares en donde se utilizaron AGs para la optimización de parámetros de un controlador difuso (Das *et al.*, 2012; Jesus and Barbosa, 2015).

Tabla 4.1 Hiperparámetros seleccionados para el algoritmo genético

Hiperparámetro	Valor
Número de generaciones	100
Población	25
Individuos elite	2
Factor de cruzamiento	0.8
Factor de mutación	0.2

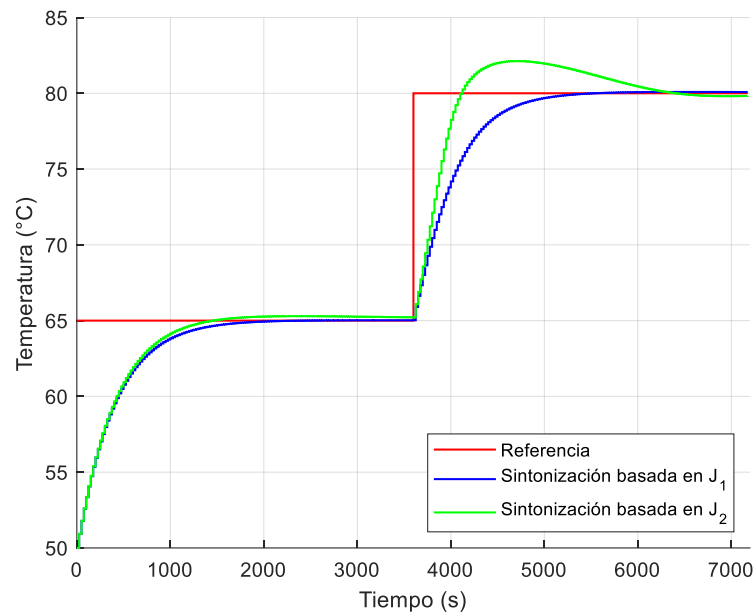
4.4. Sintonización del controlador PID

Como se mencionó en la Sección 3.4 la búsqueda de los parámetros óptimos del controlador PID se realizó mediante algoritmos genéticos. Para las funciones objetivo J_1 (4.8) y J_2 (4.9) se consideraron las ponderaciones $\{w_1, w_2\}$ como $w_1 = w_2 = 1$. El vector de parámetros para la optimización del controlador es $\theta = [K_p, K_i, K_d]^T$; el intervalo del espacio de búsqueda para estas 3 variables fue $\theta \in [0,1]$. En la Tabla 4.2 se muestra la sintonización obtenida basada en las funciones objetivo J_1 y J_2 .

Tabla 4.2 Parámetros óptimos del controlador PID

Función objetivo	$0\text{ h} < t \leq 1\text{ h}$				$1\text{ h} < t \leq 2\text{ h}$			
	$J_i(\hat{\theta})$	Parámetros controlador			$J_i(\hat{\theta})$	Parámetros controlador		
		K_p	K_i	K_d		K_p	K_i	K_d
J_1	2.2892e+06	0.3014	7.7720e-05	0.0177	2.4990e+07	0.2378	1.4088e-04	0.8192
J_2	8.3162e+06	0.3125	1.0174e-04	0.1226	1.5888e+08	0.2385	1.0e-03	0.8209

donde $J_i(\hat{\theta})$ es la función objetivo J_1 o J_2 evaluada en el vector de parámetros estimado $\hat{\theta}$. En la **Figura 4.4**. Se puede observar que en $0\text{ h} < t \leq 1\text{ h}$ la sintonización basada en J_2 es un poco más rápida, pero muestra un leve error en estado estacionario. En $1\text{ h} < t \leq 2\text{ h}$ la sintonización basada en J_2 muestra un gran sobrepaso en comparación a la sintonización de J_1 .

**Figura 4.4** Diferencia de la sintonización del controlador PID basado en la optimización de la función objetivo J_1 y J_2

4.5. Desarrollo del controlador difuso

Continuando con la metodología que se planteó en la Sección 3.3, el siguiente paso es construir un controlador PID difuso lineal que sea equivalente al PID desarrollado anteriormente.

4.5.1 PID difuso lineal

Los controladores PID difusos son equivalentes a los controladores PID bajo ciertas condiciones sobre la forma de las funciones de pertenencia y el método de inferencia (Siler and Ying, 1989; Mizumoto, 1995; Wu and Mizumoto, 1996; Tso and Fung, 1997). Las condiciones que se deben cumplir son:

1. En las variables premisa utilizar funciones de pertenencia triangulares que se superpongan en el grado de pertenencia $\mu = 0.5$.
2. Construir una base de reglas que contenga todas las posibles combinaciones \wedge (“y”) de los términos de la premisa.
3. Usar la conjunción producto (*) para el conector \wedge .
4. Usar singletons de conclusión, posicionados en la suma de las posiciones máximas de los conjuntos de premisas.
5. Utilizar el promedio ponderado para encontrar la señal de control resultante.

Con estas elecciones de diseño, la superficie de control es un plano diagonal, y el universo de salida es la suma de los universos de entrada.

Siguiendo la estructura general de diseño descrita en la Sección 3.3, los pasos a seguir se muestran a continuación:

➤ **Determinar las entradas y salidas del controlador difuso.**

Siguiendo la estructura del controlador PID difuso descrita en la Sección 3.3.1, las entradas para la base de reglas PD son E y CE . Debemos recordar que cada una de estas entradas corresponde a una variable lingüística, por lo tanto, debe definirse el universo de discurso de cada una de estas variables. Usualmente se trabaja con universos normalizados en el rango $[-1,1]$, que fue la elección que se usó en este caso. Sin embargo, se debe tener en cuenta que la elección del rango del universo del discurso es arbitraria.

Para la variable de salida u , el universo de salida debe ser la suma de los universos de entrada, es decir $[-2,2]$.

➤ **Definir las funciones de pertenencia.**

En las variables de entrada se definieron 2 funciones de pertenencias para cada variable (E y CE). Las funciones son “negativo” (NEG) y “positivo” (POS) del tipo triangular que se superponen en $\mu = 0.5$, como se muestra en la **Figura 4.5**.

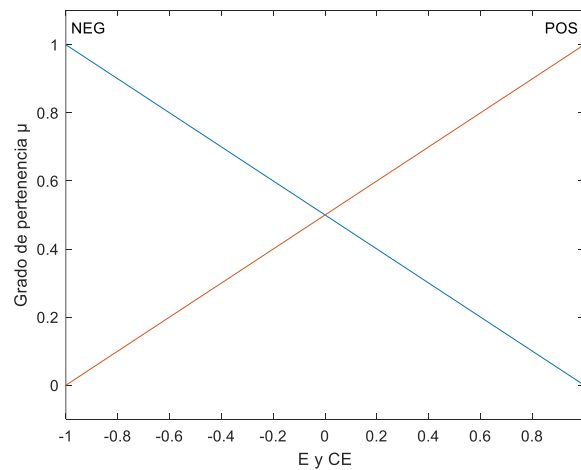


Figura 4.5 Funciones de pertenencias para variables premisa.

Para la variable de salida u , se utilizaron singletons en $u=-2$, $u=0$ y $u=2$ posicionados en la suma de las posiciones máximas de los conjuntos de premisas, como se muestra en la **Figura 4.6**.

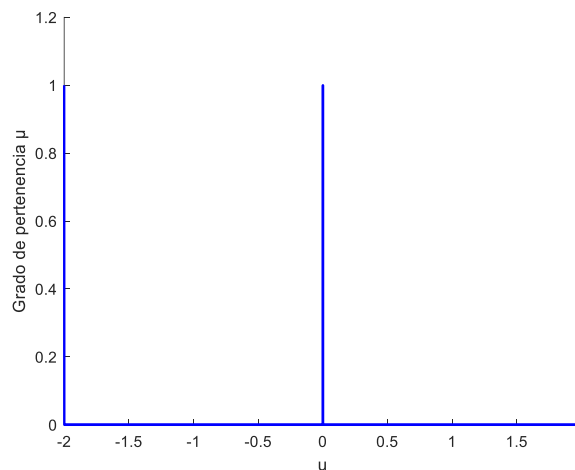


Figura 4.6 Singletons de salida.

➤ **Diseñar la base de reglas.**

La base de reglas construida fue:

- IF E is *NEG* and CE is *NEG* then u is *NB*
- IF E is *NEG* and CE is *POS* then u is *ZE*
- IF E is *POS* and CE is *NEG* then u is *ZE*
- IF E is *POS* and CE is *POS* then u is *PB*

Bajo esta base de reglas, y utilizando las demás consideraciones de diseño descritas al inicio de esta sección; se obtiene la superficie de control de la **Figura 4.7**.

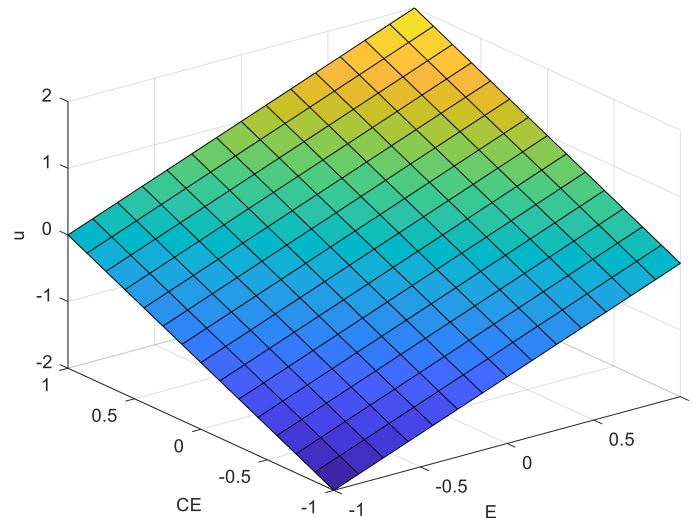


Figura 4.7 Superficie de control lineal.

➤ **Sintonizar el controlador.**

Para obtener una sintonización equivalente a un PID convencional utilizando la estructura del controlador difuso PD+I deben cumplirse además las siguientes relaciones.

$$K_p = GE * GU \quad (4.10)$$

$$K_i = GIE * GU \quad (4.11)$$

$$K_d = GCE * GU \quad (4.12)$$

➤ **Comprobar su funcionamiento.**

En la **Figura 4.8** se muestra la respuesta que se tiene con el controlador difuso PD+I lineal con las consideraciones anteriores. Para ilustrar la equivalencia se utilizó como referencia la sintonización realizada para los controladores PID (ver **Figura 4.4**). Como era de esperar ambos controladores son equivalentes.

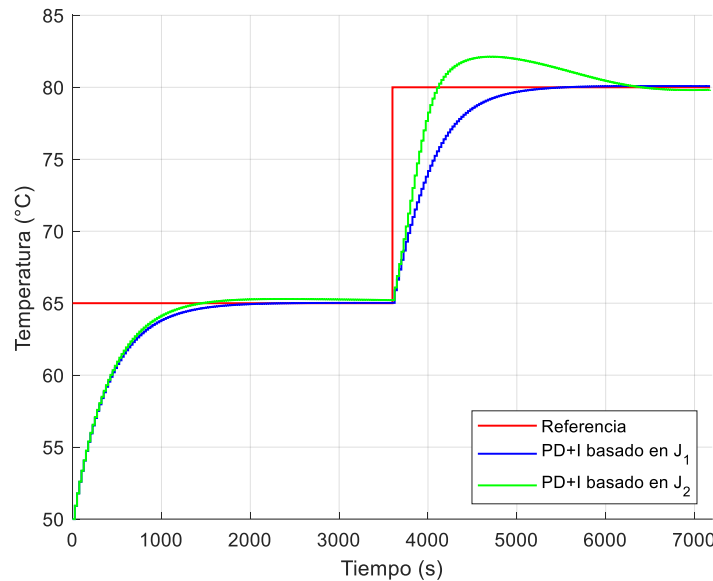


Figura 4.8 Respuesta controlador difuso PD+I lineal

4.5.2 PID difuso no-lineal

Habiendo diseñado el controlador PID difuso lineal, es momento de convertirlo en uno no-lineal. Esto se puede realizar de diversas maneras, por ejemplo, se pueden agregar más variables lingüísticas en las variables de entrada y salida. Sin embargo, esto provoca que aumente el número máximo de reglas posibles en la base de reglas, lo que trae problemas al momento de seleccionar las reglas adecuadas que aseguren un correcto desempeño del control.

Otra alternativa es modificar la forma de las funciones de pertenencia de las variables lingüísticas ya existentes, de tal manera de ir alterando la superficie de control lineal de la **Figura 4.7**. Esta es una estrategia sencilla pero efectiva ya que se mantienen las mismas variables y base

de reglas que el sistema de control PID difuso lineal diseñado anteriormente, razón por la cual esta fue la estrategia seleccionada.

Las funciones de pertenencia no-lineales seleccionadas fueron la función “Z-shaped” $\mu(x; -0.5, 0.5)$ para el término “NEG” y “S-shaped” $\mu(x; -0.5, 0.5)$ para el término “POS”, para ambas variables lingüísticas de entrada E y CE como se muestra en la **Figura 4.9**.

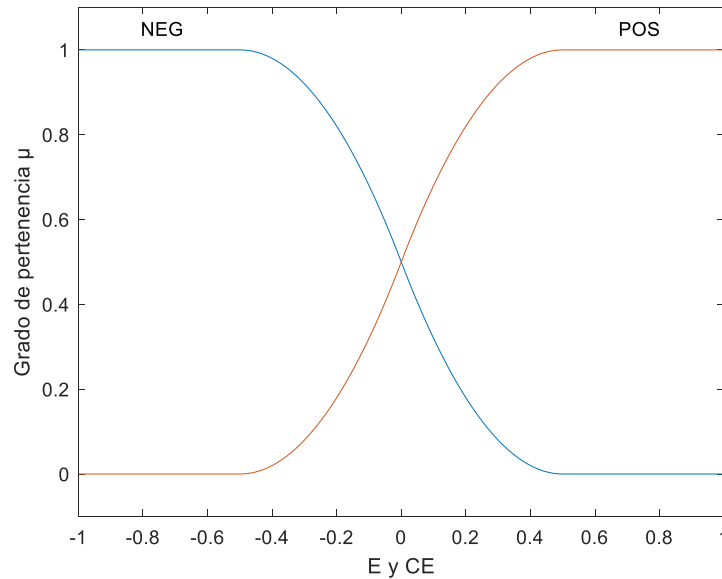


Figura 4.9 Funciones de pertenencia no-lineales para variables premisa.

Para la variable de salida u , se utilizaron singletons ubicados en la misma posición que se muestra en la **Figura 4.6** y la base de reglas construida fue equivalente a la que se utilizó para el PID difuso lineal. Bajo esta base de reglas, y utilizando las consideraciones de diseño descritas anteriormente se obtiene finalmente la superficie de control no-lineal que se muestra en la **Figura 4.10**.

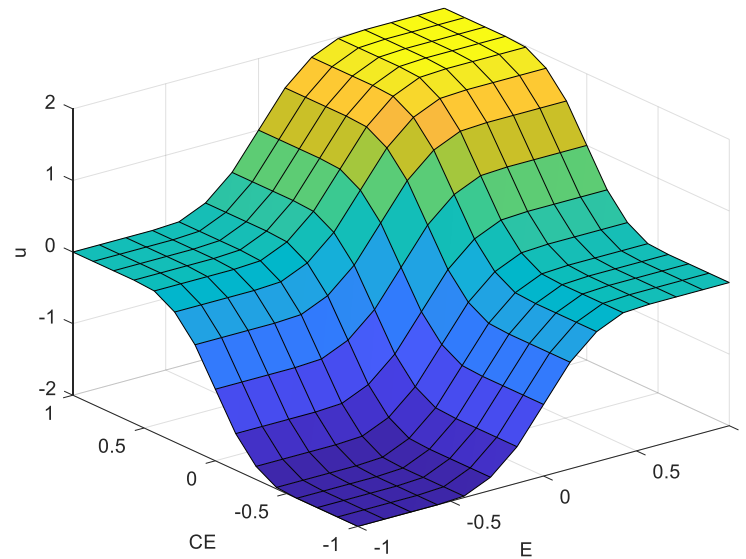


Figura 4.10 Superficie de control no-lineal

4.5.3 Sintonización del controlador difuso

Al igual que con el controlador PID la búsqueda de los parámetros óptimos del controlador PID difuso fue realizada mediante algoritmos genéticos. Para las funciones objetivo J_1 (4.8) y J_2 (4.9) se consideraron las ponderaciones $\{w_1, w_2\}$ como $w_1 = w_2 = 1$. El vector de parámetros para la optimización del controlador es $\theta = [GE, GU, GIE, GCE]^T$; los intervalos del espacio de búsqueda de estas variables fueron $\theta_1 \in [0, 0.066]$, $\theta_2 \in [0, 4.5]$, $\theta_3 \in [0, 0.01]$ y $\theta_4 \in [0, 0.4]$. En la **Tabla 4.3** se muestra la sintonización obtenida basada en las funciones objetivo J_1 y J_2 .

Tabla 4.3 Parámetros óptimos del controlador PID difuso

Función objetivo	$0 h < t \leq 1 h$					$1 h < t \leq 2 h$				
	$J_i(\hat{\theta})$	Parámetros controlador				$J_i(\hat{\theta})$	Parámetros controlador			
		GE	GU	GIE	GCE		GE	GU	GIE	GCE
J_1	5.0029e+05	0.0658	4.5	3.0178e-05	0.01	7.1194e+06	0.066	3.885	4.994e-05	0.01
J_2	3.0673e+06	0.0653	4.49	3.9745e-05	0.04	6.5347e+07	0.065	3.436	7.9796e-05	0.03

$J_i(\hat{\theta})$ es la función objetivo J_1 o J_2 evaluada en el vector de parámetros estimado $\hat{\theta}$. En la **Figura 4.11** se muestran las respuestas obtenidas. Se puede observar que tanto para $0 h < t \leq 1 h$ y $1 h < t \leq 2 h$ la sintonización basada en J_1 muestra la mejor respuesta.

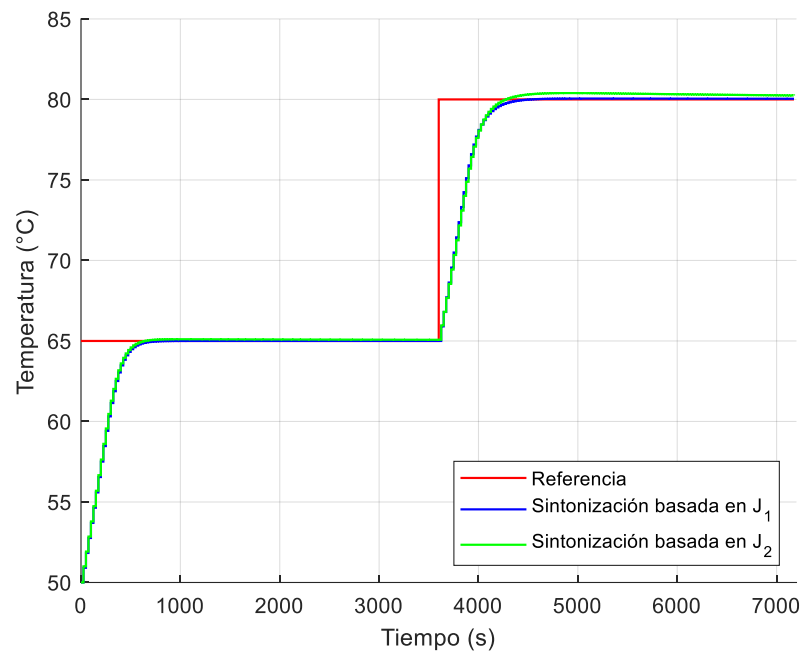


Figura 4.11 Diferencia de la sintonización del controlador PID difuso basado en la optimización de la función objetivo J_1 y J_2

4.6. Comparación de los controladores

Con los resultados obtenidos en las Secciones 4.4 y 4.5.3 se pudo observar que la sintonización basada en la optimización de la función objetivo J_1 otorga un mejor comportamiento del sistema. Debido a esto, a continuación se realiza la comparación de la respuesta del sistema utilizando el controlador PID y PID difuso, ambos sintonizados mediante la optimización de la función J_1 .

En la Tabla 4.4 se muestra la comparación del valor $J_1(\hat{\theta})$ obtenido para cada controlador junto a sus parámetros óptimos de sintonización.

Tabla 4.4 Comparación PID y PID difuso

t	PID				PID difuso				
	J ₁ (θ̂)	Parámetros controlador			J ₁ (θ̂)	Parámetros controlador			
		K _p	K _i	K _d		GE	GU	GIE	GCE
0 h < t ≤ 1 h	2.2892e+06	0.3014	7.7720e-05	0.0177	5.0029e+05	0.0658	4.5	3.0178e-05	0.01
1 h < t ≤ 2 h	2.4990e+07	0.2378	1.4088e-04	0.8192	7.1194e+06	0.066	3.885	4.994e-05	0.01

En la **Figura 4.12** se muestra gráficamente la comparación de las repuestas del sistema con los controladores PID y PID difuso.

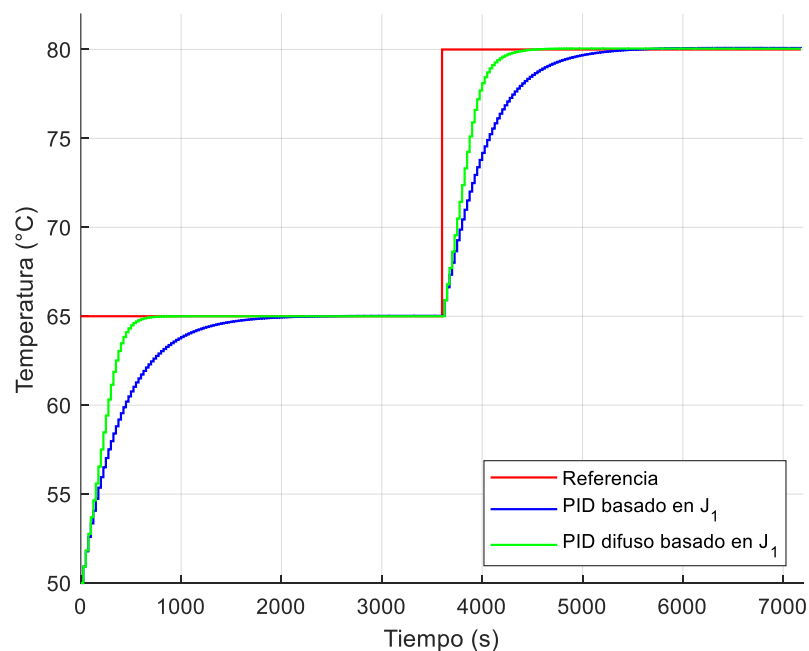


Figura 4.12 Comparación PID y PID difuso basado en J_1

En la **Tabla 4.5** se muestra la comparación del tiempo que tarda cada controlador en alcanzar la señal de referencia

Tabla 4.5 Comparación de tiempo de respuesta

Referencia	Tiempo de respuesta (minutos)	
	PID	PID difuso
65°C	39.58	15
80°C	32.1	15.42

4.7. Implementación de sistemas de control difuso

Dado que el desarrollo de los controladores difusos se basa en una intensa interacción con el diseñador, varios proveedores de software y hardware como Omron, Siemens, Apronix, Inform, National Semiconductors, etc., han introducido herramientas de software especiales. La mayoría de los programas se ejecutan en un PC, bajo Windows, algunos de ellos están disponibles también para sistemas UNIX.

Una vez que el controlador difuso se prueba con alguna herramienta de análisis de software, puede utilizarse para controlar la planta, ya sea directamente desde el entorno (a través de puertos de computadora o entradas/salidas analógicas), o mediante la generación de un código para su ejecución. La mayoría de los programas generan un código C estándar y también un código de máquina para un hardware específico, como microcontroladores o controladores lógicos programables (PLC). De esta manera, para implementar un controlador difuso no es necesario contar con hardware especializado, ya que el hardware comercial que se utiliza típicamente para tareas de control puede ser utilizado también para implementar un controlador difuso (Babuska and Kober, 2018).

El objetivo fundamental de un controlador difuso es proporcionar relaciones de entrada y salida en términos de reglas difusas. Para que esto sea posible deben realizarse las 3 operaciones esenciales: fuzzificación, inferencia difusa y defuzzificación. La implementación de un controlador difuso puede ser realizada mediante el procesamiento en tiempo real de estas 3 operaciones, lo que se conoce como implementación *on-line*. Otra opción que se utiliza comúnmente consiste en procesar estas operaciones previamente y calcular los valores de salida del controlador para cada combinación posible de valores de entrada, lo que se conoce como implementación *off-line*. La implementación que se utilizó en este trabajo corresponde a la segunda opción.

4.7.1 Implementación de controlador difuso mediante *Lookup table*

Una de las formas típicas para realizar una implementación *off-line* de un controlador difuso es mediante una tabla de consulta, conocida comúnmente por su nombre en inglés *Lookup table* (LUT). Una LUT es en informática una estructura de datos, que se usa para sustituir una rutina de computación con una simple indexación de vectores.

A continuación, se muestran los pros y contras de este enfoque:

Pros:

- Mayor velocidad: sacar un valor de memoria es mucho más rápido que hacer una serie de cálculos complejos.
- Programación sencilla.

Contras:

- Sólo funciona para sistemas con entradas y salidas limitadas.
- No hay flexibilidad en el tiempo de ejecución: nada puede ser ajustado o configurado en tiempo real, a excepción de los factores de escala.

Para construir un controlador difuso basado en LUT los universos premisa deben ser discretos, de esta forma es posible pre-calcular todas las posibles acciones de control antes de poner en funcionamiento el controlador.

En un controlador basado en LUT, la relación entre todas las combinaciones de los puntos del universo de premisas y sus correspondientes acciones de control están dispuestas en una tabla. Con dos entradas en el controlador y una acción de control, la tabla es bidimensional (2D-LUT). Con tres entradas, la tabla se convierte en un conjunto tridimensional (3D-LUT). El cálculo previo de los posibles resultados permite mejorar la velocidad de ejecución en el tiempo de funcionamiento, ya que la inferencia se reduce a una consulta de tabla, que normalmente es más rápida que una inferencia basada en reglas (Bai and Roth, 2018).

4.7.2 Discretización e interpolación de *Lookup table*

Si la resolución de la tabla es demasiado baja, se puede manifestar un efecto conocido como ciclo límite, que provoca oscilaciones sobre el punto referencia. Este efecto puede eliminarse mediante interpolación entre las celdas en lugar de redondear a la celda con el valor más cercano. Existen diversos métodos para realizar una interpolación de una 2D-LUT siendo los más utilizados los de tipo lineal y *spline*.

Para visualizar el comportamiento del controlador PID difuso basado en *Lookup Table* (PID difuso LUT) en la **Figura 4.13** se muestra el gráfico de respuesta del sistema utilizando el controlador PID difuso basado en J_1 con una LUT de 21x21 celdas bajo 3 condiciones: sin interpolación, interpolación lineal e interpolación *spline* grado 3.

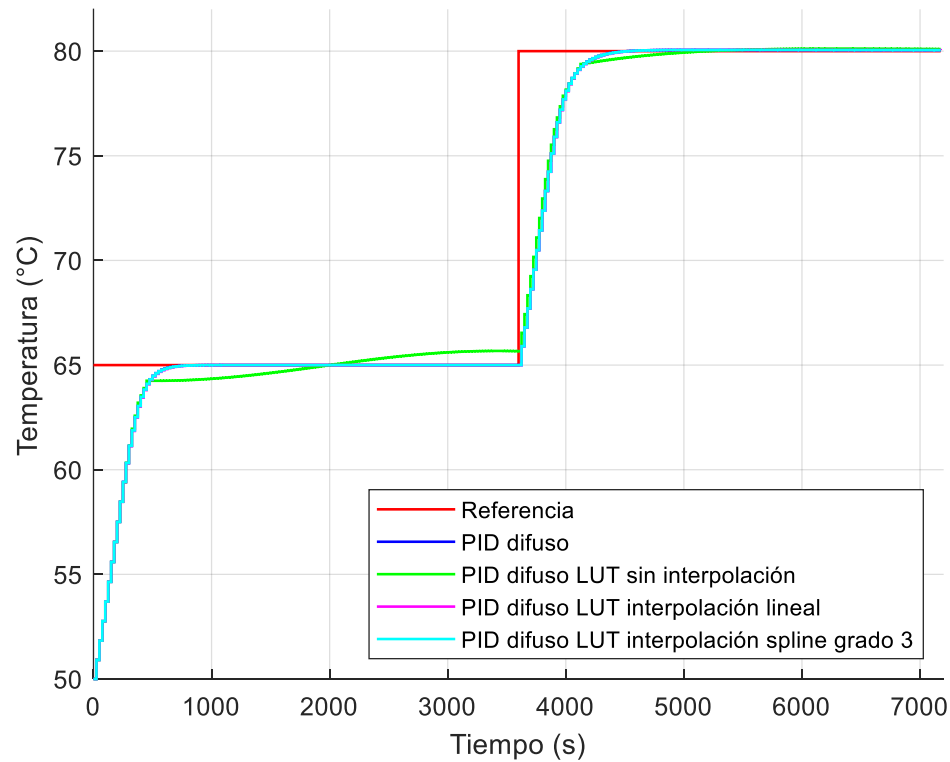


Figura 4.13 Comparación controlador PID difuso LUT

En la **Figura 4.13** se observa que el PID difuso LUT sin interpolación presenta problemas a medida que el sistema se acerca al punto de referencia, esto se puede notar de manera más clara en el tramo $0 h < t \leq 1 h$.

Para el PID difuso LUT con interpolación se observa que tanto con interpolación lineal y con *spline* grado 3 los resultados son bastantes buenos. Sin embargo, el método de interpolación *spline* grado 3 entrega los resultados más cercanos, pero hay que tener en consideración que es un método computacionalmente más costoso que la interpolación lineal.

4.7.3 Implementación en Raspberry Pi

Finalmente, se desarrolló una rutina que permite realizar la simulación del sistema de control PID difuso LUT para el control de temperatura. El programa fue realizado en Python 3.7.3 corriendo bajo la distribución Linux Raspbian 10 Buster (**Figura 4.14**).

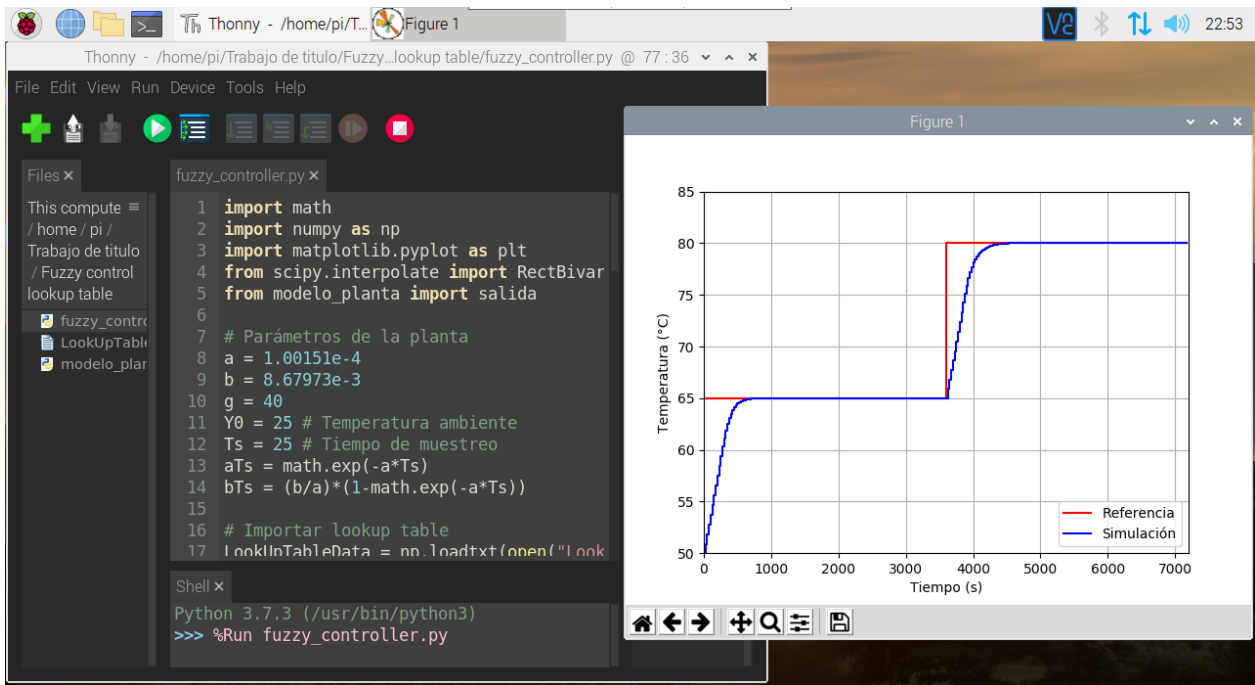


Figura 4.14 Simulador de sistema de control PID difuso LUT en Python

El programa se elaboró de tal forma que sea sencillo realizar modificaciones. Las ganancias *GE*, *GU*, *GIE* y *GCE* se pueden ingresar como vectores. La LUT puede ser cargada como un archivo `.csv` y se puede seleccionar el tipo de interpolación que se desee implementar.

Para que el programa funcione correctamente es necesario contar con las librerías *Scipy* que permite interpolar y *Numpy* que permite trabajar fácilmente con vectores y matrices.

Capítulo 5. Conclusiones

En este trabajo de titulación se realizó exitosamente el desarrollo de un sistema de control de temperatura basado en lógica difusa. El trabajo consideró la búsqueda de un modelo de sistema térmico adecuado para el estudio, el diseño del controlador difuso para dicho modelo utilizando un algoritmo genético para su sintonización, y la simulación del proceso controlado en un ordenador de placa reducida Raspberry Pi 3 modelo B+. En términos más específicos, se puede concluir lo siguiente:

- El controlador basado en lógica difusa que se desarrolló corresponde a uno del tipo PID difuso. La implementación del controlador difuso se realizó utilizando un procedimiento propuesto en la literatura, relativamente sencillo, que garantiza un desempeño del controlador difuso igual o superior al de un PID convencional.
- El controlador PID difuso desarrollado se comparó con un controlador PID convencional. Con los resultados obtenidos se puede concluir que el controlador PID difuso es superior al PID convencional para el control de temperatura de una planta no-lineal. Esta mejora se manifiesta en una gran disminución en el tiempo de respuesta respecto al controlador PID.
- El uso de algoritmos genéticos para la sintonización de los parámetros de los controladores PID y PID difuso demostró ser una herramienta bastante útil, ya que de no haberse utilizado se hubiese tenido que realizar una sintonización manual que conlleva una tarea de prueba y error. El uso de estos algoritmos permitió disminuir significativamente el tiempo necesario para realizar esta tarea.
- Hay que tener en cuenta que el desempeño del algoritmo genético depende en gran medida de la función objetivo que se plantee. En general, la elección de la mejor opción de función objetivo dependerá de la aplicación. En este estudio se evaluaron dos funciones objetivo. La función J_2 presentó la ventaja de resultar en una optimización más rápida que la función J_1 ; sin embargo, tiene la desventaja de entregar un error en estado estacionario mayor.
- La implementación del controlador difuso en la Raspberry Pi se realizó a través de un proceso de interpolación mediante una *Lookup-Table*. Aunque dicha implementación fue evaluada solo en simulación, ésta sirve como base para poder ser utilizada en aplicaciones reales en el futuro.

- La estructura de controlador utilizada es sencilla, pero puede ser de bastante utilidad en muchas aplicaciones reales. A medida que se aumenta complejidad del proceso a controlar, pueden utilizarse otras opciones más sofisticadas como por ejemplo estructuras adaptativas en donde la superficie de control es dinámica, en estos casos se suelen utilizar otras herramientas de inteligencia artificial, por ejemplo, redes neuronales artificiales en conjunto a un sistema difuso para realizar controladores conocidos como controladores neuro difusos.
- Si bien el uso de este tipo de controladores se ve prometedor, hay que tener en cuenta que su uso debe realizarse sólo cuando la dificultad del problema justifique su uso. Siempre hay que establecer con claridad cuáles serán los objetivos que se quieren lograr con el controlador y además tener en cuenta los recursos disponibles para resolver el problema. Pueden existir restricciones en los recursos, que involucren presupuesto, equipo de trabajo y tiempo disponible. Un enfoque errado es pretender utilizar este tipo de controlador para solucionar todos los problemas de automatización. Es por esto que al abordar un problema primero hay que examinar si éste puede ser resuelto mediante una estructura sencilla, como por ejemplo un controlador PID. Si se observa que el PID no responde bien quizás el problema sea una sintonización inadecuada; si este es el caso se puede optar por sintonizar el controlador PID mediante algoritmos genéticos, que es una solución que puede ser útil en muchos casos.

Bajo la experiencia adquirida se recomienda que este tipo de controladores se utilice cuando, por ejemplo:

- Se desea realizar el control de procesos que sean nuevos o de los que no se cuente con suficiente información previa.
- El mejoramiento de la tarea de control traiga consigo otros beneficios como mejora de calidad de producto, aumento de la productividad del proceso, disminución de pérdidas, etc.
- El costo de adquisición de un controlador comercial específico para el proceso sea muy alto.

Nomenclatura

Símbolo	Descripción	Unidades
t	: Tiempo	s
$y(t)$: Temperatura en el instante t	$^{\circ}C$
$f(t)$: Flujo de calor en el instante t	W
Y_0	: Temperatura ambiente	$^{\circ}C$
C	: Capacidad térmica	$J \cdot ^{\circ}C^{-1}$
R	: Resistencia térmica	$^{\circ}C \cdot W^{-1}$
T_S	: Tiempo de muestreo	s
u_k	: Voltaje aplicado en el instante k	V
y_k	: Temperatura del agua en el instante k	$^{\circ}C$
μ	: Grado de pertenencia	
R_i	: Regla i de una base de reglas	
α_i	: Grado de cumplimiento de la regla i	
GE	: Ganancia de la señal de error del controlador difuso	
GCE	: Ganancia de la señal de la derivada del error del controlador difuso	
GIE	: Ganancia de la señal de la integral del error del controlador difuso	
GU	: Ganancia de la señal de salida del controlador difuso	

Bibliografía

- Afroz, Z. *et al.* (2018) 'Modeling techniques used in building HVAC control systems: A review', *Renewable and Sustainable Energy Reviews*, **83**, pp. 64–84. doi: 10.1016/j.rser.2017.10.044.
- Aström, K. J. and Hägglund, T. (2009) *Control PID avanzado*. Pearson Prentice Hall.
- Azeem, M. F. (2012) *Fuzzy Inference System - Theory and Applications*. InTech.
- Babuska, R. and Kober, J. (2018) 'Knowledge-Based Control Systems, Lecture Notes of SC42050'. Delft University of Technology.
- Bai, Y. and Roth, Z. S. (2018) *Classical and modern controls with microcontrollers: design, implementation and applications*. Springer.
- Brunton, S. L. and Kutz, J. N. (2019) *Data-Driven Science and Engineering*. Cambridge University Press. doi: 10.1017/9781108380690.
- Chin-Teng Lin, Chia-Feng Juang and Chung-Ping Li (1999) 'Temperature control with a neural fuzzy inference network', *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, **29**(3), pp. 440–451. doi: 10.1109/5326.777078.
- Das, Saptarshi *et al.* (2012) 'A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices', *Engineering Applications of Artificial Intelligence*, **25**(2), pp. 430–442. doi: 10.1016/j.engappai.2011.10.004.
- Dorf, Richard C and Bishop, R. H. (2011) *Modern Control Systems, 12th Edition*, Pearson.
- Jantzen, J. (2013) *Foundations of Fuzzy Control*. 2nd edn. Chichester, UK: John Wiley & Sons, Ltd. doi: 10.1002/9781118535608.
- Jesus, I. S. and Barbosa, R. S. (2015) 'Genetic optimization of fuzzy fractional PD+I controllers', *ISA Transactions*, **57**, pp. 220–230. doi: 10.1016/j.isatra.2015.01.006.
- Klir, G. J. and Yuan, B. (1995) *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Pearson.
- Li, C. and Lee, C. Y. (2003) 'Self-organizing neuro-fuzzy system for control of unknown plants', *IEEE Transactions on Fuzzy Systems*, **11**(1), pp. 135–150. doi: 10.1109/TFUZZ.2002.805898.
- Mamdani (1977) 'Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis', *IEEE Transactions on Computers*, **C-26**(12), pp. 1182–1191. doi: 10.1109/TC.1977.1674779.
- Mamdani, E. H. (1974) 'Application of fuzzy algorithms for control of simple dynamic plant', *Proceedings of the Institution of Electrical Engineers*, **121**(12), p. 1585. doi: 10.1049/piee.1974.0328.
- MathWorks, I. (2020) 'Global Optimization Toolbox User's Guide'.
- Melba Mary, P. and Marimuthu, N. S. (2009) 'Design of self-tuning fuzzy logic controller for the control of an unknown industrial process', *IET Control Theory and Applications*, **3**(4), pp. 428–436. doi: 10.1049/iet-cta.2007.0334.

- Mizumoto, M. (1995) 'Realization of PID controls by fuzzy control methods', *Fuzzy Sets and Systems*, **70**(2–3), pp. 171–182. doi: 10.1016/0165-0114(94)00215-S.
- Ogata, K. (2010) *Ingeniería de control moderna*. 5ª edición. Pearson Prentice Hall.
- Passino, K. and Yurkovich, S. (1998) *Fuzzy Control*. Addison-Wesley.
- Siler, W. and Ying, H. (1989) 'Fuzzy control theory: The linear case', *Fuzzy Sets and Systems*, **33**(3), pp. 275–290. doi: 10.1016/0165-0114(89)90118-8.
- Sun, Z. F. and Carrington, C. G. (1999) 'Dynamic modelling of a dehumidifier wood drying kiln', *Drying Technology*, **17**(4–5), pp. 711–729. doi: 10.1080/07373939908917565.
- Tanomaru, J. and Omatu, S. (1992) 'Process Control by On-Line Trained Neural Controllers', *IEEE Transactions on Industrial Electronics*, **39**(6), pp. 511–521. doi: 10.1109/41.170970.
- Tóth, R. (2010) *Modeling and Identification of Linear Parameter-Varying Systems, Lecture Notes in Control and Information Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Control and Information Sciences). doi: 10.1007/978-3-642-13812-6.
- Tso, S. K. and Fung, Y. H. (1997) 'Methodological development of fuzzy-logic controllers from multivariable linear control', *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, **27**(3), pp. 566–572. doi: 10.1109/3477.584965.
- Wu, Z. Q. and Mizumoto, M. (1996) 'PID type fuzzy controller and parameters adaptive method', *Fuzzy Sets and Systems*, **78**(1), pp. 23–35. doi: 10.1016/0165-0114(95)00115-8.
- Zadeh, L. A. (1965) 'Fuzzy sets', *Information and Control*, **8**(3), pp. 338–353. doi: 10.1016/S0019-9958(65)90241-X.
- Zadeh, L. A. (1973) 'Outline of a New Approach to the Analysis of Complex Systems and Decision Processes', *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-3**(1), pp. 28–44. doi: 10.1109/TSMC.1973.5408575.
- Zimmermann, H.-J. (1996) *Fuzzy Set Theory—and Its Applications*. 3rd edn. Dordrecht: Springer Netherlands. doi: 10.1007/978-94-015-8702-0.

Anexo A. Derivación de modelo térmico

Se desea discretizar la siguiente ecuación diferencial:

$$\frac{dy(t)}{dt} = \frac{f(t)}{C} + \frac{Y_0 - y(t)}{RC} \quad (\text{A.1})$$

Consideremos que Y_0 es una segunda entrada dependiente del tiempo, i.e $Y_0 = Y_0(t)$. Se discretiza primero para $f(t)$, entonces

$$\frac{dy(t)}{dt} = -\frac{1}{RC}y(t) + \frac{1}{C}f(t) \quad (\text{A.2})$$

$$RC \frac{dy(t)}{dt} + y(t) = Rf(t) \quad (\text{A.3})$$

la función de transferencia asociada es:

$$G_p(s) = \frac{y(s)}{f(s)} = \frac{R}{RCs + 1} \quad (\text{A.4})$$

reescribiendo $G_p(s)$,

$$G_p(s) = \frac{K_p}{\tau_p s + 1} \quad (\text{A.5})$$

donde $K_p = R$ es la ganancia y $\tau_p = RC$ es la constante de tiempo. Considerando un retentor de orden cero,

$$H(s) = \frac{1 - e^{-sT_s}}{s} \quad (\text{A.6})$$

se desea obtener la transformada z de

$$G(s) = H(s)G_p(s) \quad (\text{A.7})$$

el resultado es

$$Z\{G(s)\} = G(z) = \frac{y(z)}{f(z)} = \frac{K_p(1 - e^{-T_s/\tau_p})z^{-1}}{1 - e^{-T_s/\tau_p}z^{-1}} \quad (\text{A.8})$$

$$= \frac{K_p(1 - e^{-T_s/\tau_p})}{z - e^{-T_s/\tau_p}} = \frac{b}{z - a} \quad (\text{A.9})$$

con $b = 1 - a$. Por lo tanto

$$zy(z) - ay(z) = bf(z) \quad (\text{A.10})$$

aplicando la inversa de la propiedad de desplazamiento en el tiempo se tiene

$$y_{k+1} = ay_k + bf_k \quad (\text{A.11})$$

Si consideramos la segunda entrada se tiene

$$\frac{dy(t)}{dt} = -\frac{1}{RC}y(t) + \frac{1}{RC}Y_0(t) \quad (\text{A.12})$$

$$RC \frac{dy(t)}{dt} + y(t) = Y_0(t) \quad (\text{A.13})$$

La FT asociada es

$$G_p(s) = \frac{y(s)}{Y_0(s)} = \frac{1}{RCs + 1} \quad (\text{A.14})$$

Reescribiendo $G_p(s)$

$$G_p(s) = \frac{y(s)}{Y_0(s)} = \frac{K_p}{\tau_p s + 1} \quad (\text{A.15})$$

donde $K_p = 1$ y $\tau_p = RC$.

Procediendo análogamente se tiene

$$y_{k+1} = a_2 y_k + b_2 Y_{0,k} \quad (\text{A.16})$$

Notar que $a_2 = a$ y $b_2 = 1 - a$. Reescribimos (A.11) y (A.16) para diferenciar las salidas

$$y_{k+1}^1 = a y_k^1 + b f_k \quad (\text{A.17})$$

$$y_{k+1}^2 = a y_k^2 + (1 - a) Y_{0,k} \quad (\text{A.18})$$

Aplicando el principio de superposición, se tiene que $y_k = y_k^1 + y_k^2$. Luego,

$$y_{k+1} = a y_k + b f_k + (1 - a) Y_{0,k} \quad (\text{A.19})$$

Como Y_0 es constante

$$y_{k+1} = a y_k + b f_k + (1 - a) Y_0 \quad (\text{A.20})$$

donde

$$a = e^{-T_s/RC} \quad (\text{A.21})$$

$$b = R(1 - e^{-T_s/RC}) \quad (\text{A.22})$$

Para que Ecuación (A.20) pueda utilizarse en un sistema de control, el calor f_k se reemplaza por el voltaje u_k que corresponderá a la variable manipulada que afectará la variable de control y_k . Por otra parte, se desea considerar en el modelo un límite para la temperatura máxima. Debido a lo anteriormente expuesto, el modelo de la Ecuación (A.20) se modifica de la siguiente manera:

$$y_{k+1} = ay_k + bcu_k + (1 - a)Y_0 \quad (\text{A.23})$$

donde

$$c = \frac{1}{1 + e^{0.5y_k - \gamma}} \quad (\text{A.24})$$

La presencia del coeficiente c constituye una no-linealidad en el modelo. Notar que la representación (A.23) corresponde a un modelo *quasi-LPV* (*LPV: linear parameter varying*), donde la variable de programación es y_k .