



Universidad Nacional Experimental del Táchira

Vicerrectorado Académico

Decanato de Docencia

Departamento de Ingeniería Electrónica

Trabajo de Aplicación Profesional

Proyecto Especial de Grado

## **Laboratorio virtual de sistemas de control clásicos y difusos utilizando software libre**

**Autor:** Kleiver J. Carrasco M.

C.I: V-24.743.884

[kleiver.carrasco@unet.edu.ve](mailto:kleiver.carrasco@unet.edu.ve)

**Tutor:** Juan R. Vizcaya R.

[jvizcaya@unet.edu.ve](mailto:jvizcaya@unet.edu.ve)

San Cristóbal, junio de 2019

San Cristóbal, 23 de junio de 2019

**Señores:**

Miembros de la Comisión del Trabajo de Aplicación Profesional  
Departamento de Ingeniería Electrónica

Yo, Kleiver Jesús Carrasco Márquez, titular de la cédula de identidad No. V-24.743.884, inscrito en el periodo académico 2019-1, estudiante del décimo semestre de la carrera de Ingeniería Electrónica, por medio de la presente someto a consideración de la Comisión de Trabajo de Aplicación Profesional de este departamento, la propuesta de Proyecto Especial de Grado titulada “**Laboratorio virtual de sistemas de control clásicos y difusos utilizando software libre**”, la cual se desarrollará bajo la tutoría del profesor Juan Rafael Vizcaya Rojas.

Se anexan los recaudos exigidos, de acuerdo a lo dispuesto en la normativa para el Trabajo de Aplicación Profesional de la Universidad Nacional Experimental del Táchira. La fecha estimada de culminación del proyecto es el **!FECHA!**.

Sin otro particular a que hacer referencia y en espera de su respuesta,

Atentamente,

---

Kleiver Jesús Carrasco Márquez  
C.I. V-24.743.884

San Cristóbal, 23 de junio de 2019

**Señores:**

Miembros de la Comisión del Trabajo de Aplicación Profesional  
Departamento de Ingeniería Electrónica

Yo, Juan Rafael Vizcaya Rojas, titular de la cédula de identidad No. V-7.348.191, de profesión ingeniero en electrónica, adscrito al departamento de Ingeniería Electrónica de la Universidad Nacional Experimental del Táchira en la categoría de Asociado y con dedicación exclusiva, por medio de la presente expreso mi conformidad y acepto la tutoría de la propuesta de Proyecto Especial de Grado titulada “**Laboratorio virtual de sistemas de control clásicos y difusos utilizando software libre**”, presentada por el bachiller Kleiver Jesús Carrasco Márquez, titular de la cédula de identidad No. V-24.743.884, para optar al título de Ingeniero en Electrónica en la Universidad Nacional Experimental del Táchira.

Sin otro particular a que hacer referencia, me suscribo de usted,

Atentamente,

---

Juan Rafael Vizcaya Rojas  
C.I. V-7.348.191



Universidad Nacional Experimental del Táchira

Vicerrectorado Académico

Decanato de Docencia

Departamento de Ingeniería Electrónica

Trabajo de Aplicación Profesional

Proyecto Especial de Grado

**Aprobación del Tutor para presentación de la Propuesta del Proyecto Especial de Grado**

Yo, Juan Rafael Vizcaya Rojas, en mi carácter de Tutor del Proyecto Especial de Grado titulado “**Laboratorio virtual de sistemas de control clásicos y difusos utilizando software libre**”, presentado por el bachiller Kleiver Jesús Carrasco Márquez titular de la cédula de identidad No. V-24.743.884 por medio de la presente autorizo la presentación de la Propuesta del Proyecto Especial de Grado, ante la Comisión del Trabajo de Aplicación Profesional del Departamento de Ingeniería Electrónica, en virtud de considerar que reúne los requisitos establecidos en el artículo 16 de las Normas para el Trabajo de Aplicación Profesional de la UNET.

---

Tutor  
Juan Rafael Vizcaya Rojas  
C.I. V-7.348.191

# ÍNDICE

	Pág.
<b>ÍNDICE</b> . . . . .	v
<b>ÍNDICE DE FIGURAS</b> . . . . .	vii
<b>ÍNDICE DE TABLAS</b> . . . . .	viii
 <b>CAPÍTULO I EL PROBLEMA</b>	 1
1.1 Planteamiento del problema. . . . .	1
1.2 Objetivos de la investigación . . . . .	3
1.2.1 Objetivo general . . . . .	3
1.2.2 Objetivos específicos . . . . .	3
1.3 Justificación e importancia . . . . .	3
1.4 Alcance y limitaciones . . . . .	4
 <b>CAPÍTULO II MARCO TEÓRICO</b>	 5
2.1 Antecedentes . . . . .	5
2.2 Bases teóricas . . . . .	6
2.2.1 Procesos . . . . .	7
2.2.2 Modelado de procesos en tiempo continuo . . . . .	7
2.2.2.1 Transformada de Laplace. . . . .	7
2.2.2.2 Ecuaciones de espacio de estado . . . . .	8
2.2.3 Modelado de procesos en tiempo discreto. . . . .	9
2.2.3.1 Transformada z . . . . .	9
2.2.3.2 Ecuaciones de espacio de estado discretas. . . . .	9
2.2.4 Sistemas de control . . . . .	10
2.2.4.1 Sistema en lazo abierto . . . . .	10
2.2.4.2 Sistema en lazo cerrado . . . . .	10

2.2.4.3	Estabilidad de los sistemas . . . . .	11
2.2.4.3.1	Análisis de estabilidad en el plano complejo . . . . .	11
2.2.4.3.2	Criterio de estabilidad de Nyquist. . . . .	11
2.2.4.3.3	Análisis de estabilidad con las trazas de Bode . . . . .	11
2.2.4.4	Controlador PID . . . . .	12
2.2.4.4.1	Componente proporcional . . . . .	13
2.2.4.4.2	Componente integral . . . . .	13
2.2.4.4.3	Componente derivativa . . . . .	13
2.2.4.5	Entonación por método de Ziegler-Nichols en lazo abierto . . . . .	14
2.2.5	Lógica difusa . . . . .	16
2.2.5.1	Conjuntos difusos . . . . .	16
2.2.5.2	Reglas de inferencia. . . . .	17
2.2.5.3	Defuzzificación . . . . .	17
2.2.5.3.1	Método de centro de área. . . . .	18
2.2.5.4	Sistemas de control difusos . . . . .	19
2.2.6	Python . . . . .	20
2.2.6.1	Biblioteca NumPy . . . . .	20
2.2.6.2	Biblioteca SciPy . . . . .	20
2.2.6.3	Biblioteca Matplotlib . . . . .	21
2.2.6.4	Biblioteca de control . . . . .	21
2.2.6.5	Biblioteca Scikit-Fuzzy. . . . .	21
<b>CAPÍTULO III MARCO METODOLÓGICO</b>		<b>22</b>
3.1	Tipo de investigación. . . . .	22
3.2	Fases de la investigación. . . . .	23
3.2.1	Fase 1: Estudio de los sistemas de control clásicos y difusos . . . . .	23
3.2.2	Fase 2: Codificación de rutinas . . . . .	23
3.2.3	Fase 3: Interfaz gráfica y enlace con rutinas. . . . .	23
3.2.4	Fase 4: Comparación de resultados. . . . .	23
<b>REFERENCIAS</b> . . . . .		<b>24</b>

## ÍNDICE DE FIGURAS

FIGURA	Pág.
2.1 Ejemplo de un sistema en lazo cerrado . . . . .	10
2.2 Ejemplo de un sistema en lazo cerrado con controlador . . . . .	12
2.3 Modelado por curva de reacción . . . . .	15
2.4 Ejemplo de un conjunto difuso . . . . .	17
2.5 Esquemas de control difuso . . . . .	19

## ÍNDICE DE TABLAS

TABLA	Pág.
2.1 Regla de entonación de Zigler-Nichols . . . . .	15
2.2 Tautología para modus ponens . . . . .	18
2.3 Tautología para modus tollens . . . . .	18



# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1 Planteamiento del problema**

Desde hace muchos años que el hombre dedico parte de sus esfuerzos a ofrecer servicios y producir bienes para el consumo de las personas, antaño, los procesos de producción eran fáciles de implementar y de complejidad reducida, por tanto, se podían controlar de forma manual utilizando instrumentos y herramientas simples, pero Creus (2010) afirma que: “[...] la gradual complejidad con qué éstos se han ido desarrollando ha exigido su automatización progresiva por medio de los instrumentos de medición y control”(p. 1).

Así mismo, la calidad de vida de las personas ha mejorado gracias a que ahora la producción de bienes y servicios se realizan de forma más eficiente, parte de este incremento de eficiencia se debe a la incorporación de nuevas tecnologías que traen consigo ventajas como rapidez, precisión y mejoras en la automatización. La automatización por medio de controladores analógicos y electrónicos ha desempeñado un papel importante en esta mejora, tanto así, que se ha convertido en parte integra en los sistemas de vehículos espaciales, robóticos, procesos modernos de fabricación y en cualquier operación industrial (Ogata, 2003).

Siguiendo este orden de ideas, es necesario mencionar que analizar los sistemas de control puede llegar a ser, en ocasiones, una tarea difícil de realizar si no se tienen los conocimientos necesarios o si no se utilizan las herramientas adecuadas, uno de los motivos es que “[...] los antecedentes matemáticos requeridos incluyen temas tales como la teoría de la variable compleja, ecuaciones diferenciales y en diferencias, transformada de Laplace y transformada  $z$  [...]”(Kuo, 1996, p. 21).

Considerando esto, es natural pensar que un modo de abarcar el análisis, diseño y simulación de sistemas de control es por medio de las computadoras, Ogata (2003) sugiere que gran parte del tiempo dedicado será verificando el comportamiento del sistema mediante un análisis, es por esto que recomienda utilizar un programa de computadora como MATLAB para que realice gran parte del cálculo matemático necesario en los estudios de sistemas de control, no obstante, se debe aclarar que MATLAB es un software que, aunque potente, permanece cerrado y de pago.

En adición a lo anterior, se puede pensar en utilizar herramientas libres como Octave y Scilab, que proporcionan un número elevado de funciones matemáticas, pero se requiere de saber programar, además, no ofrecen un entorno gráfico para la entonación de controladores, lo cual puede llegar a ser problemático para algunos ingenieros, por otro lado, suelen ofrecer soluciones aisladas entre sí e integrarlas suele ser tedioso y problemático, Suárez (2014) afirma que: “Si lograr el dominio de la herramienta computacional es un reto en sí mismo deja de ser una herramienta práctica [...]”(p. 6).

Finalmente, se debe tener en cuenta que las herramientas libres y gratuitas no tienen la posibilidad de diseñar controladores a base de lógica difusa de forma intuitiva y sencilla, lo cual puede conllevar un desperdicio de tiempo considerable en comparación con el uso de una interfaz gráfica para el diseño del controlador, es por esto que la mayoría de las herramientas gratuitas están limitadas a usarse, de manera práctica, solo en teoría clásica de control.

En base a la problemática expuesta, surgen las siguientes preguntas: ¿Es posible realizar un laboratorio para el análisis de sistemas de control con software libre?, ¿Cumpliría con los requisitos para analizar, diseñar y simular sistemas de control? y ¿Cómo se desempeñaría en comparación con otras herramientas?, preguntas que se responderán con el desarrollo de esta investigación y que se utilizarán para guiar el rumbo de la misma.

## **1.2 Objetivos de la investigación**

### **1.2.1 Objetivo general**

Desarrollar un laboratorio virtual de sistemas de control clásicos y difusos utilizando software libre.

### **1.2.2 Objetivos específicos**

1. Estudiar los sistemas de control clásicos.
2. Estudiar el diseño de controladores difusos tipo Mamdani.
3. Codificar las rutinas de análisis, diseño y simulación de sistemas de control necesarias.
4. Realizar la interfaz gráfica de un laboratorio de sistemas de control virtual.
5. Comparar los resultados obtenidos con otras herramientas de corte similar.

## **1.3 Justificación e importancia**

Actualmente hay una dependencia muy alta de MATLAB a la hora de trabajar con calculo numérico, así mismo, es el software más usado en la UNET para analizar, diseñar y simular sistemas de control, con el desarrollo de un laboratorio de control utilizando software libre se puede eliminar parcialmente dicha dependencia, logrando así que herramientas externas sean usadas solo cuando se den casos más particulares o complejos.

Con el desarrollo del laboratorio de control se quiere tener una herramienta que cumpla con los requisitos actuales para el análisis, diseño y simulación de sistemas de control de forma libre, gratuita, rápida y sencilla sin tener que invertir demasiado tiempo en aprender la herramienta, y si en lo que importa, diseñar un sistema de control preciso y confiable.

Por otro lado, se espera desarrollar la herramienta de forma que su interfaz gráfica acepte otros módulos i.e., permitirá a aquel que lo desee expandir las funcionalidades

del laboratorio de sistemas de control sin que se tenga que rehacer todo, esto permitirá mantener actualizado y útil la herramienta, además, se estará generando la posibilidad de realizar otras investigaciones alrededor de la misma para su expansión.

#### **1.4 Alcance y limitaciones**

Con esta investigación se realizará una interfaz gráfica que permita realizar cuatro funciones principales, la primera, análisis de procesos en el dominio del tiempo y en el dominio de la frecuencia como: respuesta al escalón, respuesta al impulso, bode, entre otras, la segunda, entonación de controladores PID de forma manual y con auto tuning, la tercera, será el diseño de controladores difusos tipo Mamdani generales y para esquemas específicos de sistemas de control, y finalmente, la simulación de sistemas de control utilizando controladores PID y controladores difusos para esquemas específicos.

Hay que aclarar que, para modelar los sistemas de control en cada una de las funcionalidades mencionadas se le dará la opción al usuario para representarlos como funciones de transferencia o ecuaciones de espacio de estado y se podrá especificar si el modelo está en tiempo continuo o en tiempo discreto. El análisis de sistemas de control es un tema muy amplio y requiere de una gran cantidad de funciones que se pueden realizar utilizando software libre, no obstante, se considera que con este alcance se estaría logrando cumplir con las necesidades fundamentales.

El software que se utilizará para realizar la interfaz gráfica y los cálculos correspondientes será el lenguaje de programación Python junto con su set de bibliotecas externas. Aunque la biblioteca de Python para análisis de sistemas de control es potente y brindan varias herramientas de forma directa, se debe dejar claro que no cumplen con todas las funciones requeridas, por tanto, algunas rutinas de simulación se deberán codificar de cero con ayuda de bibliotecas de cálculo numérico, y otras, harán uso de la biblioteca de control con código complementario.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1 Antecedentes**

Para esta investigación se buscaron antecedentes de alcance nacional e internacional que ayuden a guiar, sustentar y validar esta investigación.

Casallas, Chacón y Rivera (2005), escribieron para la revista *Acción Pedagógica* de la Universidad de los Andes un artículo titulado “Desarrollo básico de un Laboratorio Virtual de Control de Procesos basado en Internet”, fue desarrollado conjuntamente entre la UNET y la ULA para ser usado a través de Internet y cuyo objetivo fue desarrollar un laboratorio virtual de control de procesos para la enseñanza a distancia, permitiendo ser utilizado solo por usuarios registrados y en determinados horarios. Aunque la finalidad de la investigación acá propuesta difiere del objetivo de Casallas et al., esta servirá como ayuda para determinar las funciones más utilizadas en el área de control, así como las necesidades típicas de un laboratorio de control virtual.

Adicionalmente, Salazar (2019), realizó en la Universidad Técnica del Norte, Ecuador, una tesis titulada “Diseño de un sistema de riego inteligente para cultivos de hortalizas basado en Fuzzy Logic en la granja la pradera de la Universidad Técnica del Norte.” para la implementación del controlador basado en la lógica difusa utilizó, entre otros equipos, un Raspberry PI para la ejecución de Python, específicamente, hizo uso de la biblioteca Scikit-Fuzzy, la cual le permitió definir las funciones de membresía, establecer las reglas difusas, realizar los procesos de fuzzificación y defuzzificación, todo esto para una arquitectura de controlador difuso tipo Mamdani, esta tesis servirá como referencia para establecer el diseño de controladores difusos utilizando la biblioteca Scikit-Fuzzy.

Por otro lado, Congo (2018) realizo en la Universidad Tecnológica Israel, Ecuador, una tesis titulada “Aplicaciones del software libre Python para prácticas de laboratorio aplicado a la asignatura de tratamiento digital de señales de la Universidad Tecnológica Israel” cuyo objetivo fue desarrollar por medio del software Python y sus diferentes librerías científicas, la realización de 3 Prácticas de Laboratorio para la asignatura Procesamiento Digital de Señales de la Universidad Tecnológica Israel, al igual que la intención de este trabajo, se realizó una interfaz gráfica con el objetivo de facilitar su uso, en este caso como herramienta para el procesamiento digital de señales. Este trabajo sustenta la viabilidad de realizar una interfaz gráfica con enfoque similar en el campo de la electrónica y se utilizara como guía parcial para establecer su uso en sistemas de control.

Finalmente, Gómez (2009) redacto para la revista Educación en Ingeniería de Colombia, un artículo titulado “Toolbox didáctico para el diseño y análisis de sistemas de control lineal”, cuyo objetivo es describir un toolbox realizado en MATLAB para el análisis de sistemas de control lineales, este toolbox consiste de una interfaz gráfica que permite realizar pruebas a sistemas de control como respuesta escalón, respuesta en frecuencia, análisis de estabilidad, diseño de controladores, entre otras funciones. Este articulo será de utilidad para establecer la interfaz gráfica que se pretende realizar, además, sirvió como punto de partida para determinar las funciones que debería tener un laboratorio virtual de sistemas de control, como punto adicional se puede resaltar que reafirma la utilidad de esta investigación.

## **2.2 Bases teóricas**

Para poder realizar el laboratorio virtual de sistemas de control clásicos y difusos será necesario abarcar conocimientos de análisis de sistemas de control, diseño de controladores PID y controladores difusos, a continuación, se presentan los conceptos necesarios para el desarrollo de esta investigación.

### 2.2.1 Procesos

Es importante partir de la base, es por ellos que empezaremos con los procesos, Sánchez (2003) define los procesos como “[...]un bloque que se identifica porque tiene una o más variables de salida de las cuales es importante conocer y mantener sus valores”(p. 153). Así, un proceso se caracteriza por tener una entrada y una salida (para procesos SISO), dicha salida deberá mantenerse alrededor de un punto de referencia dado, es decir, se debe controlar su salida.

### 2.2.2 Modelado de procesos en tiempo continuo

Un proceso puede ser representado por ecuaciones diferenciales que determinen su comportamiento dinámico en el dominio del tiempo, no obstante, trabajar con ecuaciones diferenciales es tedioso a nivel matemático, por tanto, se prefiere modelar los procesos utilizando la transformada de Laplace para llevar de una representación dinámica a una algebraica, esta ecuación algebraica se operara para llevar a una función de transferencia que represente al proceso en el dominio de la frecuencia compleja (Smith y Corripio, 1985).

#### 2.2.2.1 Transformada de Laplace

La transformada de Laplace de una función dependiente del tiempo  $f(t)$  viene dada por la ecuación:

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)e^{-st} dt \quad (2.1)$$

Si suponemos que  $f(t)$  es un proceso cuya representación dinámica viene dada por una ecuación diferencial de la forma:

$$a_2 \frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_0 x(t) \quad (2.2)$$

Aplicando (2.1) a (2.2) y despejando la relación  $Y(s)/X(s)$  para el proceso con condiciones iniciales igual a cero (debido a que el proceso debe ser lineal) se obtiene la correspondiente función de transferencia del proceso  $H(S)$ :

$$H(s) = \frac{Y(s)}{X(s)} = \frac{b_0}{a_2 s^2 + a_1 s + a_0} \quad (2.3)$$

La demostración completa (Smith y Corripio, 1985, pp. 21-22) no es de interés para este trabajo, pero si su resultado, la función de transferencia puede ser analizada para determinar las características del proceso, como su ganancia, constante de tiempo, tiempo muerto, entre otras.

### ***2.2.2.2 Ecuaciones de espacio de estado***

Las ecuaciones de espacio de estado son un método más moderno para modelar todo tipo de sistemas, no solo físicos, sino también biológicos, económicos, sociales y otros. Así como una ecuación diferencial puede ser representada como una función de transferencia también se puede representar como una ecuación de espacio de estado, por tanto, una función de transferencia también puede ser representada en el espacio de estados y viceversa. Las ecuaciones linealizadas alrededor de un estado de operación (Ogata, 2003, p. 31) son:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2.4)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (2.5)$$

con:

$A(t)$ : como matriz de estado.

$B(t)$ : como matriz de entrada.

$C(t)$ : como matriz de salida.

$D(t)$ : como matriz de transmisión directa.

Para un modelo continuo e invariante en el tiempo las matrices A, B, C y D deben ser independientes del tiempo.



### 2.2.3 Modelado de procesos en tiempo discreto

Así como para tiempo continuo se utilizan ecuaciones diferenciales, en el tiempo discreto se hace uso de ecuaciones en diferencias, las ecuaciones en diferencias se pueden utilizar para aproximar a las ecuaciones diferenciales, estas primeras se suelen utilizar porque son más fáciles de programar (Kuo, 1996).

#### 2.2.3.1 Transformada $z$

En tiempo discreto se puede modelar un proceso utilizando la transformada  $z$  sobre la ecuación en diferencias del proceso para obtener la función de transferencia discreta  $H(z)$ , la ecuación general para la transformada  $z$  es:

$$F(z) = \sum_{k=0}^{\infty} f(k)z^{-k} \quad (2.6)$$

De este modo aplicando (2.6) sobre la entrada y salida de la ecuación en diferencias del proceso se obtiene  $H(z)$  y viene dada por:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{\infty} y(k)z^{-k}}{\sum_{k=0}^{\infty} x(k)z^{-k}} \quad (2.7)$$

#### 2.2.3.2 Ecuaciones de espacio de estado discretas

Similar al tiempo continuo, un proceso puede ser modelado utilizando ecuaciones de espacio de estado discreta, un proceso invariante en el tiempo viene dado por las siguientes ecuaciones:

$$\dot{x}(k+1) = Ax(k) + Bu(k) \quad (2.8)$$

$$y(k) = Cx(k) + Du(k) \quad (2.9)$$

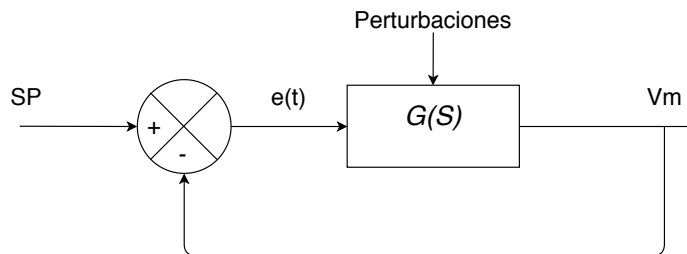
## 2.2.4 Sistemas de control

### 2.2.4.1 Sistema en lazo abierto

Un sistema en lazo abierto es aquel cuya salida no es medida en ningún momento en orden de ajustar la entrada, esto provoca que “las perturbaciones que se presentan en el proceso ocasionen que sus efectos se sientan en la salida del proceso, es decir, en el valor de la variable controlada”(Maloney, 2006, p. 350). La mayoría de los sistemas en lazo abierto son estables, pero carecen de utilidad al no poder mantenerlos en un punto de referencia debido a las perturbaciones.

### 2.2.4.2 Sistema en lazo cerrado

Los sistemas de lazo cerrado son aquellos cuya salida es realimentada a la entrada del sistema, esta realimentación es de tipo negativa para que sea estable, no obstante, existen sistemas que se vuelven inestables con el simple hecho de cerrar el lazo, un ejemplo de sistema realimentado se puede observar en la Figura 2.1.



**Figura 2.1. Sistema en lazo cerrado.** Se observan las principales señales de un sistema en lazo cerrado. Fuente: Elaboración propia.

En la Figura 2.1 se denotan algunas de las señales que componen a un sistema de control,  $SP$ , que corresponde al set point o valor de referencia,  $V_m$ , que corresponde a la variable medida y es la variable a controlar, las perturbaciones, las cuales son magnitudes físicas que pueden afectar al proceso como la temperatura ambiente, presión, vibraciones, entre otras, y finalmente,  $e(t)$ , que corresponde a la señal de error la cual viene dada por la diferencia entre el valor de referencia y la variable medida (Maloney, 2006).

### **2.2.4.3 Estabilidad de los sistemas**

La estabilidad se ve afectada por la estructura del sistema, por tanto, se debe tomar en cuenta cuando se cierra el lazo del sistema. Ogata (2003) afirma que, desde el punto de vista de estabilidad, el sistema de control en lazo abierto no es un problema importante. Por otra parte, la estabilidad es un gran problema en el sistema de control en lazo cerrado debido a la posibilidad de que se generen oscilaciones. Esto quiere decir que los análisis de estabilidad se realizan para sistemas en lazo cerrado, esto es principalmente porque los sistemas en lazo abierto tienden a ser estables, además, para realizar el control de un proceso se suele utilizar los sistemas en lazo cerrado.

**2.2.4.3.1 Análisis de estabilidad en el plano complejo.** La estabilidad de un sistema lineal viene dada por la ubicación de sus polos a lo largo del plano complejo  $s$ , tomando esto en cuenta se puede determinar si un sistema en lazo cerrado es estable si no posee ninguno polo en el semiplano derecho del plano complejo  $s$  incluyendo el eje coordenado  $j\omega$  (Ogata, 2003). Esto permite que el análisis de estabilidad se pueda realizar de forma analítica y gráfica.

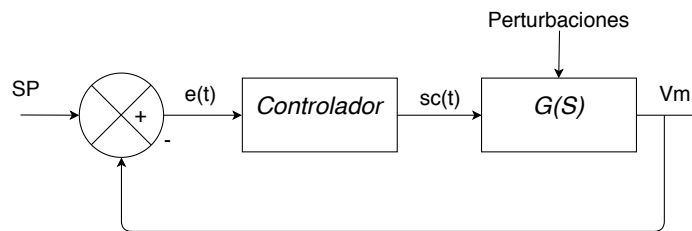
**2.2.4.3.2 Criterio de estabilidad de Nyquist.** El criterio de estabilidad de Nyquist determina el estado de estabilidad utilizando los polos en lazo abierto y la respuesta en frecuencia del sistema en lazo abierto, este criterio “es útil en la ingeniería de control, debido a que permite determinar gráficamente la estabilidad absoluta del sistema en lazo cerrado a partir de las curvas de respuesta en frecuencia en lazo abierto, sin que sea necesario determinar los polos en lazo cerrado”(Ogata, 2003, p. 446). Este criterio tiene la ventaja de poder realizarse tanto con cálculos analíticos como con datos experimentales.

**2.2.4.3.3 Análisis de estabilidad con las trazas de Bode.** Los diagramas de bode son una forma de representar la respuesta en frecuencia de un sistema tomando en cuenta los cambios de amplitud de  $H(j\omega)$  y del ángulo de fase de  $H(j\omega)$  respecto a la frecuencia (Nilsson y Riedel, 2005). Para analizar la estabilidad se utilizan el margen

de ganancia y el margen de fase del sistema en un diagrama de Bode a modo de trazas, creando así, zonas claramente delimitadas por estas trazas que determinan la estabilidad relativa del sistema dependiendo de la zona en donde se ubique en función de  $\omega$ .

#### 2.2.4.4 Controlador PID

El controlador será el elemento dentro del sistema de control que se encargara de llevar al proceso a un valor de referencia deseado, existen varios esquemas de control para el control de procesos, cada uno con sus ventajas y desventajas, el más común es un sistema en lazo cerrado con controlador, este se puede observar en la Figura 2.2.



**Figura 2.2. Sistema en lazo cerrado con controlador.** El controlador recibe la señal de error y genera una señal de control. Fuente: Elaboración propia.

En este esquema se observa que el controlador recibe la señal de error producto del comparador entre el setpoint y la variable controlada, este error es utilizado por el controlador para generar una señal de control  $sc(t)$ . El controlador más usado en la industria es el controlador PID, así lo afirma Kuo (1996): “[...] uno de los controladores más ampliamente empleados es el controlador PID [...] donde las letras son las iniciales de proporcional, integral y derivativo”(p. 671). La ecuación que define a un PID en el dominio del tiempo es la siguiente:

$$sc(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (2.10)$$

Y su forma en función de transferencia:

$$G_c(s) = \frac{sc(s)}{e(s)} = \frac{K_d s^2 + K_p s + K_i}{s} \quad (2.11)$$

**2.2.4.4.1 Componente proporcional.** La componente proporcional se obtiene de multiplicar la señal de error por la ganancia proporcional, esto provoca que la señal de control sea proporcional a la señal de error con una relación igual a la ganancia proporcional, si el controlador solo constase de componente proporcional el error en estado estable nunca se eliminaría, dicho de otro modo, el error podría ser muy bajo, pero jamás cero. “Este error se denomina la desviación permanente u ‘offset’. El mismo disminuye si se aumenta el valor de  $K$ ”(Mata, 1999, p. 54).

**2.2.4.4.2 Componente integral.** La componente integral viene dada por la sumatoria de la señal del error en el tiempo multiplicada por la ganancia integral, por tanto, se obtiene una acumulación en un periodo determinado, esto implica que mientras exista error, la componente integral seguirá aumentando o disminuyendo dependiendo del signo de la señal de error hasta que el error sea cero, la componente integral se suma con la componente proporcional formando una única señal de control.

Debido a que la componente integral depende del tiempo, es posible corregir desviaciones de la variable controlada generadas por perturbaciones externas, por otro lado, debido a que eventualmente se conseguirá que el error sea igual a cero la componente proporcional no hará ningún aporte al sistema, y la señal de control pasara a ser totalmente generada por la componente integral hasta que exista un cambio en el setpoint o una alteración en la variable medida.

**2.2.4.4.3 Componente derivativa.** La componente derivativa viene dada por el ritmo de cambio de la señal de error multiplicado por la ganancia derivativa, por tanto, cuando el error permanece constante la componente derivativa es igual a cero, la componente derivativa será tan grande como la velocidad con la que cambie el error, esto ayuda a evitar que se generen sobre pasos en la variable medida respecto a la variable de referencia.

La componente derivativa es muy sensible al ruido, es por esto que se debe utilizar solo cuando se requiera un cierto grado de anticipación y no exista ruido (Smith

y Corripio, 1985). Es recomendable utilizar filtros en orden de disminuir el posible ruido, a su vez, cuando el proceso a controlar es de respuesta rápida se recomienda no agregar componente derivativa o que su ganancia sea muy pequeña.

#### **2.2.4.5 Entonación por método de Ziegler-Nichols en lazo abierto**

El método de Ziegler-Nichols en lazo abierto requiere primero aproximar el modelo del proceso como una función de transferencia de primer orden con tiempo muerto.

$$G(s) = \frac{K}{\tau s + 1} e^{-\alpha s} \quad (2.12)$$

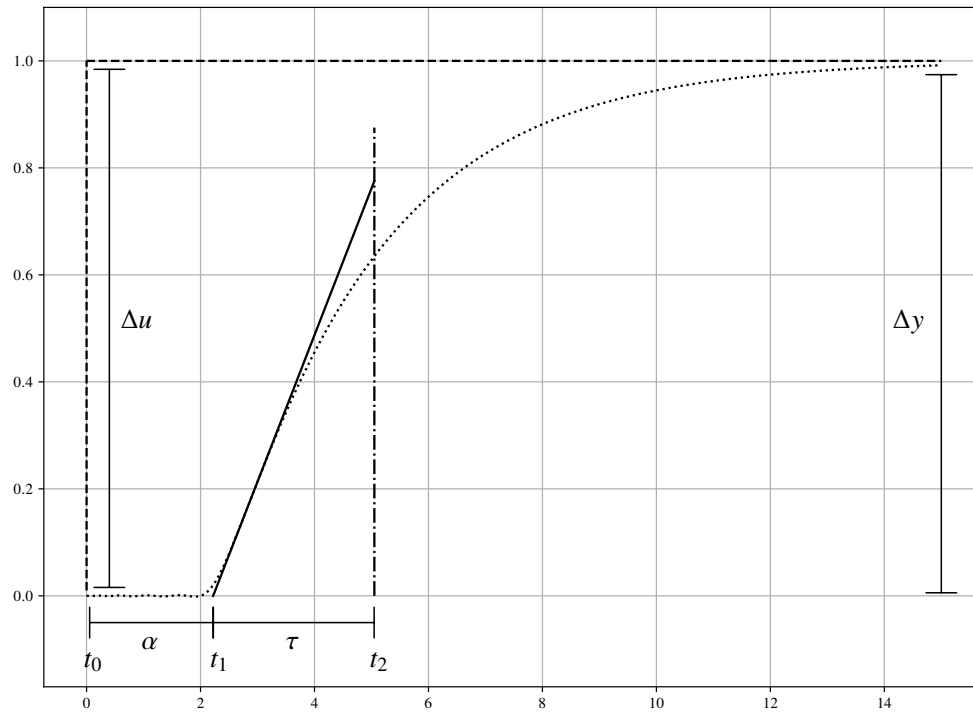
Los parámetros correspondientes pueden ser obtenidos realizando una prueba al escalón y extrayendo los datos correspondientes de la gráfica de la respuesta, un ejemplo se puede observar en la Figura 2.3. Acá se observa: (a)  $t_0$  que corresponde al inicio del escalón, (b)  $t_1$  inicio de la respuesta por parte del proceso al escalón, (c)  $t_2$  cuando el proceso alcanza el 63 % del cambio total, (d)  $\Delta y$  cambio total del proceso ante el escalón y (e)  $\Delta u$  magnitud de cambio de la entrada. Los parámetros para aproximar el proceso a un modelo de primer orden con tiempo muerto se calculan como sigue:

$$K = \frac{\Delta y}{\Delta u} \quad (2.13)$$

$$\tau = t_2 - t_1 \quad (2.14)$$

$$\alpha = t_1 - t_0 \quad (2.15)$$

Con estos valores se puede obtener la entonación de un controlador P, PI o PID, los valores de  $K_p$ ,  $K_i$  y  $K_d$  se pueden obtener sustituyendo los valores en la Tabla 2.1 correspondiente a la regla de entonación de Ziegler-Nichols. Ogata (2003) comenta que: “Las reglas de sintonía de Ziegler-Nichols [...] se han usado ampliamente para sintonizar controladores PID en sistemas de control de procesos en los que no se conoce con precisión la dinámica de la planta”(p. 572).



**Figura 2.3. Modelado por curva de reacción.** Para realizar este modelado se estudia la respuesta del sistema ante la entrada de la función de heaviside  $u(t)$ , también llamada función escalón unitario, el modelo obtenido a través de este método es una aproximación del sistema. Fuente: Elaboración propia.

**Tabla 2.1.**  
**Regla de entonación de Zigler-Nichols**

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$\frac{\tau}{\alpha}$	$\infty$	0
PI	$0.9 \frac{\tau}{\alpha}$	$\frac{\alpha}{0.3}$	0
PID	$1.2 \frac{\tau}{\alpha}$	$2\alpha$	$0.5\alpha$

**Nota.** Los valores de  $K_i$  y  $K_d$  se pueden obtener de la siguiente forma:  $K_i = K_p/T_i$  ;  $K_d = K_p T_d$ . Fuente: Ogata (2003).

### **2.2.5 Lógica difusa**

La lógica difusa es una extensión matemática de la lógica convencional en donde se asigna un grado de pertenencia a un hecho entre un valor verdadero y uno falso, “consiste en que los valores verdaderos (en lógica difusa) o valores de pertenencia (en conjuntos difusos) se indican en un número entre  $[0.0, 1.0]$ , donde 0.0 representa falsedad total y 1.0 significa verdad absoluta”(Ponce, 2010, p. 4).

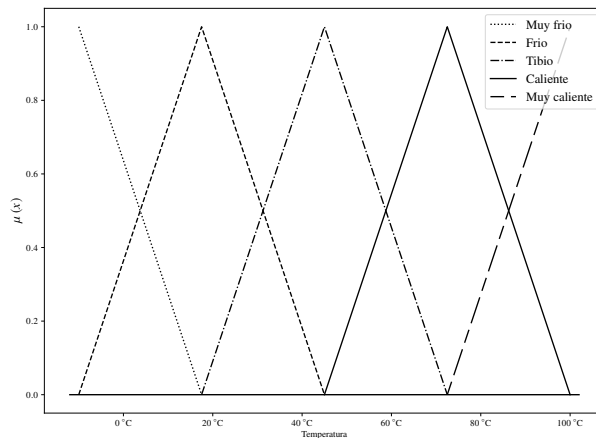
La lógica se puede estructurar en tres etapas, la primera etapa consiste en establecer cada una de las variables con etiquetas lingüística que determinen en conjunto el universo de discurso de cada una de ellas, es decir, se crean los conjuntos difusos, en la segunda etapa se definen las reglas de inferencia difusa que determinaran el comportamiento del sistema difuso, y tercera, se obtienen los valores de salida utilizando las reglas de inferencia y realizando el proceso de defuzzificación que consiste en llevar los grados de pertenencia a un valor nítido o real (Ponce, 2010).

#### **2.2.5.1 Conjuntos difusos**

Un conjunto difuso está compuesto de funciones de membresía, dichas funciones de membresía representan valores para asignar el grado de pertenencia de una variable a un estado en particular, estos estados son identificados con etiquetas lingüísticas descriptivas, como alto, caliente, bajo, negativo, entre otras, en la Figura 2.4 se puede observar un ejemplo de conjunto difuso.

Las funciones de membresía pueden tomar varias formas y rangos en orden de poder representar de mejor forma los grados de pertenencia de la variable a una etiqueta en particular, sin embargo, las formas triangulares suelen ser las menos pesadas en memoria y en poder de computo (Riid, 2002). La cantidad de etiquetas que se pueden asignar a un conjunto difuso es ilimitada, pero se debe tener en cuenta el nivel de complejidad que se genera a medida que se incrementa su número.





**Figura 2.4. Ejemplo de un conjunto difuso.** Este conjunto difuso corresponde a una variable de temperatura que va desde -10°C hasta 100°C. Fuente: Elaboración propia.

### 2.2.5.2 Reglas de inferencia

Las reglas de inferencia, al igual que en la lógica convencional, se encargan de realizar una conexión entre los antecedentes y los consecuentes con el fin de obtener una salida, en este caso, entre los conjuntos difusos de entrada y la salida, la salida puede ser un conjunto difuso o una función matemática. Como sugiere Ponce (2010), las reglas pueden formularse de dos formas denominadas modus ponens y modus tollens, modus ponens utiliza los antecedentes para obtener el consecuente, por otro lado, el modus tollens utiliza los consecuentes recíprocos para obtener el antecedente. La tautología para modus ponens y modus tollens se puede observar a modo de tablas de la verdad en la Tabla 2.2 y Tabla 2.3 respectivamente.

### 2.2.5.3 Defuzzificación

En esta última etapa también llamada desfusificación, se obtienen los valores reales de salida, Ponce (2010) lo describe como el “[...] mapeo a escala que convierte el rango de valores de las variables de salida a sus universos de discurso correspondientes. La desfusificación es la herramienta para obtener la acción de control nítida a partir de una acción de control difusa” (p. 73). Este proceso es matemáticamente costoso y existen distintos métodos para realizarlo, el más común es el método de centro de área.

**Tabla 2.2.**  
**Tautología para modus ponens**

A	B	$A \rightarrow B$	$(A \wedge (A \rightarrow B))$	$(A \wedge (A \rightarrow B)) \rightarrow B$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

**Tabla 2.3.**  
**Tautología para modus tollens**

A	B	$A \rightarrow B$	$(A \wedge (A \rightarrow B))$	$(A \wedge (A \rightarrow B)) \rightarrow B$
0	0	1	0	1
0	1	0	0	1
1	0	1	0	1
1	1	1	1	1

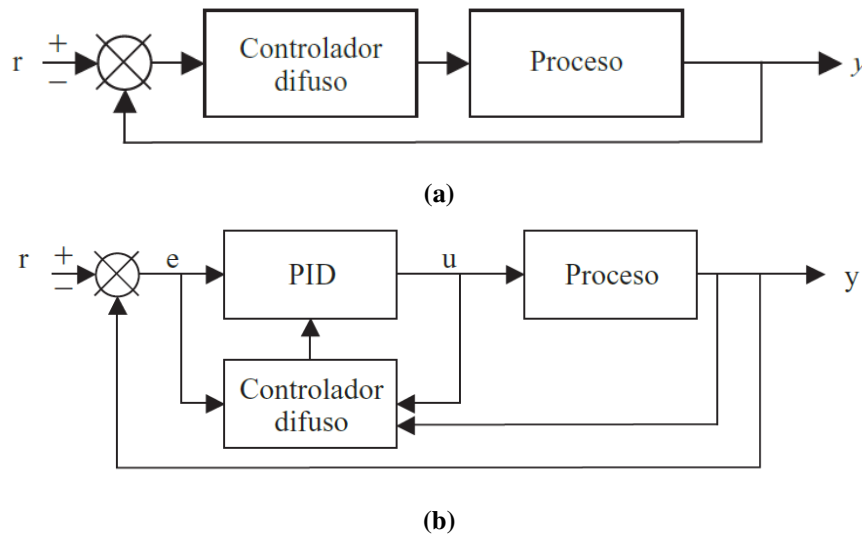
**Nota.** La tautología para modus ponens y modus tollens es presentada utilizando tablas de la verdad.  
Fuente: Ponce (2010), p. 70

**2.2.5.3.1 Método de centro de área.** Este método se utiliza para realizar la defuzzificación y obtener las salidas nítidas, consiste en cortar las funciones de membresía  $\mu_n(x)$  con los valores de entrada  $x$  para determinar un área, el área inferior será la que se tome en cuenta para realizar el cálculo, el método de centro de área se puede representar de modo general en su forma discreta de la siguiente forma:

$$Salida = \frac{\sum_{x=a}^b \mu_n(x) \cdot x}{\sum_{x=a}^b \mu_n(x)} \quad (2.16)$$

#### 2.2.5.4 Sistemas de control difusos

Un controlador difuso es cualquier tipo de controlador que utilice en su interior alguna forma de lógica difusa, al igual que con los controladores clásicos el esquema de control más común es en lazo cerrado tal como se observa en la Figura 2.5a. El controlador difuso a su vez puede contener todo tipo de metodologías de control, de hecho, el uso más común es realizar un controlador P, PI, PD o PID, todos difusos. Para realizar estos controladores se suele utilizar la estructura Mamdani, la cual asigna conjuntos difusos para las entradas y para las salidas, por tanto, es necesario el proceso de defuzzificación, comúnmente con el método de centro de área.



**Figura 2.5. Esquemas de control difuso.** (a) Esquema de control difuso clásico, (b) Esquema de control difuso autosintonizable. Fuente: Ponce (2010), p. 86 y p. 104 respectivamente.

Adicionalmente existe el esquema de control difuso con PID autosintonizable, el cual consiste en dos controladores, uno difuso y un PID clásico, el controlador difuso se encargará de generar las ganancias del PID de forma dinámica en función de las entradas que reciba, a su vez, el PID será quien gobierne al proceso, esto permite adaptar el controlador PID a los cambios de carga y a los cambios del setpoint, el esquema de control se puede observar en la Figura 2.5b.

No existen métodos analíticos claros para la entonación de controladores difusos, por tanto, la entonación se debe realizar bajo la experiencia del diseñador, tomando en cuenta las necesidades de control y las dinámicas del proceso, esto es debido a que el controlador difuso depende enteramente de la estructura de los conjuntos difusos y de la asignación de las reglas de inferencia (Ponce, 2010).

### **2.2.6 Python**

Python es un lenguaje de programación interpretado y multiparadigma ya que soporta programación orientada a objetos, imperativa y en menor medida funcional. Van Rossum (2017) afirma que Python es un lenguaje de programación potente que puede aprenderse fácilmente. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su escritura dinámica, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Adicionalmente, Python posee bibliotecas externas para realizar cálculos numéricos complejos, también existen bibliotecas para el análisis de sistemas de control y para el diseño de controladores difusos, por otro lado, existen bibliotecas para salidas gráficas con calidad de publicación, gráficas en tiempo real y en 3D.

#### **2.2.6.1 Biblioteca NumPy**

NumPy es una biblioteca para realizar cálculos computacionales científicos y matemáticos, algunas de sus funciones son: cálculo de vectores n-dimensionales, transformada de Fourier, álgebra lineal, entre otras (Oliphant, 2006)

#### **2.2.6.2 Biblioteca SciPy**

El paquete SciPy es un conjunto de bibliotecas matemáticas, científicas y de ingeniería compuestas por: NumPy, SciPy (biblioteca), SymPy, Matplotlib, IPython y Pandas, la biblioteca SciPy es uno de los núcleos del paquete SciPy. Provee varias rutinas

eficientes de uso amigable para el usuario de tipo numérica para realizar integración, interpolación, optimización, álgebra lineal, estadísticas, entre otras (Jones, Oliphant, Peterson et al., 2001), toda la documentación puede ser conseguida en la página oficial de SciPy.

#### ***2.2.6.3 Biblioteca Matplotlib***

Matplotlib es una biblioteca para la graficación 2D que produce figuras con calidad de publicación en varios formatos y a través de múltiples ambientes interactivos. Matplotlib puede ser usado en scripting, consolas de comando, IPython, jupyter notebooks y en aplicaciones web (Hunter, 2007). La sintaxis para usar Matplotlib es muy similar al sistema de plots de MATLAB, del mismo modo, configurar parámetros de estilos, fuentes, anchos de línea, entre otros, se realiza de manera similar.

#### ***2.2.6.4 Biblioteca de control***

La biblioteca control de python es un conjunto de clases y funciones que implementan las operaciones más comunes en sistemas de control, además, posee un módulo de compatibilidad para usuarios de MATLAB que emula las funciones y sintaxis de este lenguaje (“Control Systems Library for Python”, 2017). Para crear el modelo de un sistema podemos usar ecuaciones de espacio de estado o funciones de transferencia.

#### ***2.2.6.5 Biblioteca Scikit-Fuzzy***

Scikit-Fuzzy es una colección de algoritmos de lógica difusa con la intención de pertenecer al paquete de herramientas científicas SciPy, esta biblioteca permite diseñar y simular controladores difusos con estructura tipo Mamdani (Warner et al., 2016), no posee un modo para realizar lazos de control o compatibilidad con la biblioteca de control de forma directa, de modo que se tendrían que codificar aparte las rutinas que conformen el lazo cerrado de control.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

En este capítulo se abarcará la metodología a emplear en el desarrollo de esta investigación, se definirá el tipo de investigación y las fases de la misma.

#### **3.1 Tipo de investigación**

Tomando en cuenta los objetivos de la investigación y las bases teóricas que la componen se considera que esta investigación es de tipo descriptiva, esto es debido a que se pretende realizar una propuesta concreta para solventar una problemática, Hurtado de Barrera (2010) afirma que:

La investigación proyectiva tiene como objetivo diseñar o crear propuestas dirigidas a resolver determinadas situaciones. Los proyectos de arquitectura e ingeniería, el diseño de maquinarias, la creación de programas de intervención social, el diseño de programas de estudio, los inventos, la elaboración de programas informáticos, entre otros, siempre que estén sustentados en un proceso de investigación, son ejemplos de investigación proyectiva. (p. 133)

Sin embargo, también se considera que esta investigación es un proyecto factible, debido a que tiene objetivos para atender una necesidad por medio de unas acciones claramente definidas, Dubs de Moya (2002) afirma que:

un proyecto factible consiste en un conjunto de actividades vinculadas entre sí, cuya ejecución permitirá el logro de objetivos previamente definidos en atención a las necesidades que pueda tener una institución o un grupo social en un momento determinado. Es decir, la finalidad del proyecto factible radica en el diseño de una propuesta de acción dirigida a resolver un problema o necesidad previamente detectada en el medio. (pp. 6-7)

## **3.2 Fases de la investigación**

### **3.2.1 Fase 1: Estudio de los sistemas de control clásicos y difusos**

En esta fase se procederá a realizar los estudios necesarios en el área de los sistemas de control, esto con la idea de abarcarlos en profundidad y tener un entendimiento claro de su funcionamiento y de la matemática implicada, además, se realizará de forma similar un estudio de controladores difusos con estructura Mamdani y de los esquemas de control difuso.

### **3.2.2 Fase 2: Codificación de rutinas**

Para esta fase con los conocimientos adquiridos de la fase 1, se determinarán que rutinas pueden ser ejecutadas solo con las bibliotecas de python y cuales se deberán codificar de cero, además, se codificaran todas las rutinas necesarias para el funcionamiento del laboratorio virtual, para esto, se hará uso de las bibliotecas externas de cálculo numérico, control, diseño de controladores difusos y salidas gráficas junto con las que se consideren necesarias.

### **3.2.3 Fase 3: Interfaz gráfica y enlace con rutinas**

En esta fase se realizará la interfaz gráfica para el usuario final, esta interfaz gráfica deberá conectarse y adaptarse a las rutinas previamente codificadas en la fase 2 para su funcionamiento adecuado, en orden de tener diseño acorde se tomarán en cuenta los antecedentes presentados en esta propuesta.

### **3.2.4 Fase 4: Comparación de resultados**

En esta última fase y con el laboratorio de sistemas de control ya en funcionamiento se procederá a analizar los resultados obtenidos y a compararlos con otras herramientas, la evaluación se realizará en función del resultado esperado, facilidad de implementación, velocidad de ejecución, las ventajas y desventajas de cada una de las herramientas.

## REFERENCIAS

- Casallas, R., Chacón, R. y Rivera, F. P. (2005). Desarrollo básico de un Laboratorio Virtual de Control de Procesos basado en Internet. *Acción pedagógica*, 14(1), 58-65. Recuperado el 16 de junio de 2019, desde <https://dialnet.unirioja.es/descarga/articulo/2969002.pdf>
- Congo, J. W. (2018). *Aplicaciones del software libre Python para prácticas de laboratorio aplicado a la asignatura de tratamiento digital de señales de la Universidad Tecnológica Israel* (Tesis de pregrado, Universidad Tecnológica Israel, Ecuador). Recuperado el 9 de junio de 2019, desde <http://157.100.241.244/bitstream/47000/1626/1/UISRAEL-EC-ELDT-378.242-2018-042.pdf>
- Control Systems Library for Python. GitHub. Recuperado el 1 de junio de 2019, desde <https://python-control.readthedocs.io/en/0.8.2/>
- Creus, A. (2010). *Instrumentación Industrial* (8.<sup>a</sup> ed.). Barcelona, España: Alfaomega.
- Dubs de Moya, R. (2002). El Proyecto Factible: una modalidad de investigación. *Sapiens. Revista Universitaria de Investigación*, 3(2). Recuperado el 15 de junio de 2019, desde <http://www.redalyc.org/pdf/410/41030203.pdf>
- Gómez, J. C. (2009). Toolbox didáctico para el diseño y análisis de sistemas de control lineal. *Revista Educación en Ingeniería*, 4(8), 155-169. Recuperado el 13 de junio de 2019, desde <http://www.educacioneningenieria.org/index.php/edi/article/download/96/86>
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. Recuperado el 1 de junio de 2019, desde <https://ieeexplore.ieee.org/document/4160265>
- Hurtado de Barrera, J. (Ed.). (2010). *Guía para la comprensión holística de la ciencia*. Caracas, Venezuela: Fundación Sypal.
- Jones, E., Oliphant, T., Peterson, P. et al. (2001). SciPy: Open source scientific tools for Python. Recuperado el 1 de junio de 2019, desde <https://www.scipy.org/scipylib/index.html>



- Kuo, B. C. (1996). *Sistemas de control automático* (7.<sup>a</sup> ed.). México: Prentice Hall Hispanoamericana.
- Maloney, T. J. (2006). *Electrónica industrial moderna* (5.<sup>a</sup> ed.). México: Pearson Educación.
- Mata, N. C. (1999). *Fundamentos prácticos para el control de procesos*. Caracas, Venezuela: MCL CONTROL.
- Nilsson, J. W. y Riedel, S. A. (2005). *Circuitos eléctricos* (7.<sup>a</sup> ed.). Madrid, España: Pearson Educación.
- Ogata, K. (2003). *Ingeniería de control moderna* (5.<sup>a</sup> ed.). Madrid, España: Pearson Educación.
- Oliphant, T. (2006). NumPy: A guide to NumPy. USA: Trelgol Publishing. Recuperado el 31 de mayo de 2019, desde <http://www.numpy.org/>
- Ponce, P. C. (2010). *Inteligencia artificial con aplicaciones a la ingeniería*. México: Alfaomega.
- Riid, A. (2002). *Transparent Fuzzy Systems: modelling and control* (Tesis doctoral, Tallinn Technical University, Tallinn, Estonia). Recuperado el 14 de junio de 2019, desde [https://www.researchgate.net/publication/34752921\\_Transparent\\_Fuzzy\\_Systems\\_Modeling\\_and\\_Control](https://www.researchgate.net/publication/34752921_Transparent_Fuzzy_Systems_Modeling_and_Control)
- Salazar, L. J. (2019). *Diseño de un sistema de riego inteligente para cultivos de hortalizas basado en Fuzzy Logic en la granja la pradera de la Universidad Técnica del Norte* (Tesis de pregrado, Universidad Técnica del Norte, Ecuador). Recuperado el 8 de junio de 2019, desde <http://repositorio.utn.edu.ec/jspui/bitstream/123456789/9137/1/04%5C%20RED%5C%20219%5C%20TRABAJO%5C%20DE%5C%20GRADO.pdf>
- Sánchez, J. A. (2003). *Control avanzado de procesos:(teoría y práctica)*. Madrid, España: Ediciones Díaz de Santos.
- Smith, C. A. y Corripio, A. B. (1985). *Principles and practice of automatic process control* (2.<sup>a</sup> ed.). New York, United States: Wiley New York.

- Suárez, M. Á. (2014). *Ciencias e Ingeniería para estudiantes de Python*. Recuperado el 7 de junio de 2019, desde [https://www.researchgate.net/publication/313664762\\_Ciencias\\_e\\_Ingenieria\\_para\\_estudiantes\\_de\\_Python](https://www.researchgate.net/publication/313664762_Ciencias_e_Ingenieria_para_estudiantes_de_Python)
- Van Rossum, G. (2017). El tutorial de Python. Recuperado el 15 de junio de 2019, desde <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
- Warner, J., Sexauer, J., twmeggs, S., A. M., Unnikrishnan, A., Castelão, G., . . . Mishra, H. (2016). Scikit-Fuzzy: fuzzy logic library for python. Github. Recuperado el 16 de junio de 2019, desde <https://github.com/scikit-fuzzy/scikit-fuzzy>