



espol

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

**Facultad de Ingeniería en Mecánica y Ciencias de la
Producción**

**“Análisis comparativo de la estabilidad de un Péndulo Triple
Invertido sobre un carro mediante la implementación de un
control optimizado Fuzzy – PSO y un regulador lineal cuadrático”**

TESIS DE POSTGRADO

Previo a la obtención del Título de:

**MAGÍSTER EN CIENCIAS DE LA INGENIERÍA MECÁNICA CON
MENCION EN SISTEMAS MECÁNICOS**

Presentada por:

José Darwin Alvarado Bravo

GUAYAQUIL – ECUADOR

Año: 2022

AGRADECIMIENTO

Agradezco a Dios por la vida, a mi familia y enamorada por la confianza y apoyo en este trayecto, a mi director de tesis, el Dr. Jorge Hurel Ezeta, y a las personas que colaboraron de una u otra forma para la realización de este trabajo.

DEDICATORIA

Este trabajo realizado con
esfuerzo por varios meses
está dedicado a mis padres,
novia, abuelos, familiares y
amigos.

TRIBUNAL DE TITULACIÓN

Ángel Ramírez M., Ph.D.
DECANO DE FIMCP
PRESIDENTE

Jorge Hurel E., Ph.D.
DIRECTOR DE TESIS

Jorge Amaya R., Ph.D.
CO-DIRECTOR

Christian Silva S., Ph.D.
VOCAL 1

Carlos Helguero A., Ph.D.
VOCAL 2

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta tesis de titulación, me corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

José Darwin Alvarado Bravo

RESUMEN

El péndulo invertido en dinámica se ha convertido en uno de los sistemas más importantes para determinar leyes o estrategias de control debido a su simplicidad de construcción y por sus interesantes propiedades, como la no linealidad, y por ser un sistema subactuado.

A lo largo de los años, el péndulo invertido ha presentado diversas configuraciones en su construcción de acuerdo con el número de eslabones (simple, doble, triple). En este trabajo el sistema de péndulo triple invertido es empleado para desarrollar las estrategias de control propuestas, debido a su aumento considerable en la inestabilidad.

Los controladores difusos han mostrado excelentes resultados en la interpretación de decisiones para el control de sistemas no lineales, asemejándose al pensamiento humano. Sin embargo, el ajuste de los factores de escala no siempre resulta sencillo y eficiente, y los métodos de ensayo y error pueden requerir mucho tiempo.

Por lo tanto, los algoritmos evolutivos como la optimización de enjambre de partículas se han aplicado con éxito en los últimos años para ajustar los factores de escala de los sistemas lineales y no lineales, sin embargo, existe una falta de investigación en la literatura para estabilizar un péndulo triple invertido utilizando la lógica difusa, en donde los factores de escala se ajustan por medio del algoritmo de optimización de enjambre de partículas.

Este trabajo presenta un control de la estabilidad de un péndulo triple invertido utilizando un controlador difuso, donde los factores de escala son sintonizados mediante el algoritmo de optimización de enjambre de partículas. Además, los resultados fueron comparados con un regulador lineal cuadrático y un sintonizador de ganancias PSO.

El sistema triple péndulo invertido fue linealizado y representado en espacio de estados para ser introducido dentro del entorno Matlab – Simulink junto con los controladores y el algoritmo de optimización PSO.

Los resultados mostraron un mejor control del sistema al reducir el tiempo de estabilización del péndulo triple invertido utilizando el controlador borroso con respecto al controlador LQR y al sintonizador de ganancias PSO. Esto demuestra que el controlador borroso es un 39,13% y un 20% más eficiente en la reducción del tiempo de estabilización del sistema triple péndulo invertido que el controlador LQR y el sintonizador de ganancias PSO respectivamente.

ÍNDICE GENERAL

	Pág.
RESUMEN.....	I
ÍNDICE GENERAL	II
ABREVIATURAS	IV
SIMBOLOGÍA	V
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	VII
CAPÍTULO 1	
1. INTRODUCCIÓN.....	1
1.1. Antecedentes	1
1.2. Planteamiento del problema	2
1.3. Justificación.....	2
1.4. Objetivo general	3
1.4.1. Objetivos específicos.....	3
CAPÍTULO 2	
2. REVISIÓN LITERARIA.....	4
2.1. El triple péndulo invertido sobre un carro.....	4
2.2. Sistemas de control y su clasificación	5
2.2.1. Sistemas de control en lazo abierto.....	5
2.2.2. Sistemas de control en lazo cerrado.....	6
2.3. Elementos de un sistema de control.....	6
2.4. Modelado de sistemas de control.....	6
2.4.1. Representación en espacios de estados.....	8
2.4.1.1. Controlabilidad.....	9
2.4.1.2. Observabilidad.....	9
2.5. Teoría del control óptima.....	9
2.6. Controladores basados en la teoría de control óptimo.....	9
2.6.1. Regulador Óptimo no lineal.....	10
2.6.2. Regulador lineal cuadrático (LQR).....	11
2.6.3. Controlador LQR con seguimiento de referencia	12
2.7. Lógica difusa.....	13
2.7.1. Función de pertenencia	14
2.7.2. Características de la función de pertenencia.....	14
2.7.3. Operaciones con conjuntos difusos.....	15

2.7.4. Fuzzificación.....	16
2.7.5. Defuzzificación.....	16
2.7.6. Métodos de defuzzificación.....	16
2.7.7. Sistema de Inferencia difusa.....	17
2.7.7.1. Construcción y funcionamiento del sistema de inferencia.....	17
2.7.7.2. Métodos de inferencia difusa.....	18
2.8. Optimización por enjambre de partículas (PSO).....	19
2.8.1. Implementación del factor social, cognitivo y peso de inercia.....	19
2.8.1.1. Factor cognitivo y social.....	19
2.8.1.2. Factor de Inercia.....	20
2.8.2. Procedimiento del algoritmo.....	20
CAPÍTULO 3	
3. METODOLOGÍA.....	22
3.1. Modelo Matemático no lineal del triple péndulo.....	23
3.2. Linealización y representación en espacio de estado.....	34
3.3. Diseño del controlador LQR.....	35
3.4. Diseño del sintonizador de ganancias mediante el algoritmo PSO.....	38
3.5. Diseño del controlador difuso sintonizado mediante el algoritmo PSO.....	42
3.6. Diseño del controlador difuso.....	42
3.7. Desarrollo del algoritmo de optimización.....	46
CAPÍTULO 4	
4. RESULTADOS.....	50
CAPÍTULO 5	
5. CONCLUSIONES Y RECOMENDACIONES.....	54
5.1. Conclusiones.....	54
5.2. Recomendaciones.....	55
BIBLIOGRAFÍA	
ANEXOS	

ABREVIATURAS

IP	Péndulo invertido
DIP	Péndulo invertido doble
TIP	Péndulo invertido triple
AI	Inteligencia artificial
PSO	Optimización por enjambre de partículas
LQR	Regulador lineal cuadrático
SCA	Sistemas de control en lazo abierto
SCC	Sistemas de control en lazo cerrado
Gain-PSO	Ganancias sintonizadas por algoritmo de optimización de enjambre
Fuzzy-PSO	Controlador difuso sintonizado por algoritmo de optimización de enjambre
$x(t)$	Vector de estados
$y(t)$	Vector de salidas
y_{ref}	Señal de referencia
$u(t)$	Vector de entradas
$A(t)$	Matriz de estados
$B(t)$	Matriz de entrada
$C(t)$	Matriz de salida
$D(t)$	Matriz de transmisión directa
X_0	Estado inicial
X_{tf}	Estado final
T_0	Tiempo inicial
T_f	Tiempo final
X_{tf}	Estado final
p	Ecuación de coestado
J	Índice de rendimiento
Q	Error del estado
R	Esfuerzo del controlador
K	Ganancias del controlador LQR
FIS	Sistema de inferencia difusa
T-S	Takagi-Sugeno
W	Inercia

SIMBOLOGÍA

Kg	Kilogramos
m	Metros
N	Newton
s	Segundo
t	Tiempo
rad	Radianes
g	Gravedad
d	Fricción viscosa
j	Momento de inercia
c	Centro de masa
Nms ²	Newton metro segundo al cuadrado

ÍNDICE DE FIGURAS

	Pág.
Figura 2. 1. Esquema del péndulo triple invertido -----	4
Figura 2. 2. Esquema del sistema de control en lazo abierto-----	5
Figura 2. 3. Esquema del sistema de control en lazo cerrado-----	6
Figura 2. 4. Esquema de la representación matemática de un sistema -----	7
Figura 2. 5. Representación en espacio de estados mediante diagrama de bloques -----	8
Figura 2. 6. Esquema del controlador LQR en lazo cerrado-----	12
Figura 2. 7. Esquema del controlador LQR con seguimiento de referencia-----	13
Figura 2. 8. Sistema difuso que acepta datos imprecisos y afirmaciones vagas -----	13
Figura 2. 9. Formas comunes de las funciones de pertenencias-----	14
Figura 2. 10. Características de la función de pertenencia-----	15
Figura 2. 11. a) Unión. b) Intersección. c) Complemento-----	15
Figura 2. 12. Métodos: -----	17
Figura 2. 13. Sistema de inferencia difusa -----	18
Figura 2. 14. Representación gráfica de sistema de inferencia difusa -----	18
Figura 2. 15. Diagrama de flujo del algoritmo PSO -----	21
Figura 3. 1. Esquema metodológico para el análisis comparativo del sistema TIP.....	22
Figura 3. 2. Diagrama de bloques de la planta no lineal.....	32
Figura 3. 3. Respuesta del sistema no lineal sin acción de control: a) Péndulo 1,.....	33
Figura 3. 4. Sistema linealizado	35
Figura 3. 5. Respuesta del sistema linealizado sin acción de control: a) Péndulo 1,	36
Figura 3. 6. Diagrama de bloques del controlador LQR	37
Figura 3. 7. Respuesta del controlador LQR en el sistema TIP.....	37
Figura 3. 8. Señal de control de la estrategia LQR.....	38
Figura 3. 9. Esquema del sintonizador de ganancias PSO	39
Figura 3. 10. Diagrama de boques del sintonizador de ganancias PSO.....	40
Figura 3. 11. Respuesta del sintonizador de ganancias PSO.....	41
Figura 3. 12. Señal de control del sintonizador de ganancias PSO	41
Figura 3. 13. Esquema del controlador propuesto Fuzzy-PSO.....	42
Figura 3. 14. Caja de herramientas de Matlab: "Fuzzy Logic Designer".	43
Figura 3. 15. Funciones de pertenencia para variables normalizadas.....	44
Figura 3. 16. Visualización de las reglas difusas dentro de la herramienta de Matlab".....	45
Figura 3. 17. Visualización gráfica de las reglas difusas.	45
Figura 3. 18. Evaluación de la partícula fitness en el enjambre inicial.....	47
Figura 3. 19. Núcleo del algoritmo	48
Figura 3. 20. Diagrama de bloques en simulink de la optimización Fuzzy - PSO	49
Figura 3. 21. Respuesta del controlador Fuzzy – PSO.....	49
Figura 4. 1. Diagrama de bloques en simulink para el análisis comparativo.....	50
Figura 4. 2. Comparación de las estrategias de control empleadas para la estabilización del sistema TIP.....	52
Figura 4. 3. Comparación de las señales de control.....	53

ÍNDICE DE TABLAS

	Pág.
Tabla 1. Parámetros de diseño del Péndulo Triple Invertido	23
Tabla 2. Parámetros iniciales del algoritmo PSO (Caso1).....	39
Tabla 3. Velocidad y posición inicial del enjambre	40
Tabla 4. Reglas del controlador difuso	43
Tabla 5. Variables lingüísticas.....	44
Tabla 6. Parámetros iniciales del algoritmo PSO (Caso 2).....	46
Tabla 7. Velocidad y posición inicial del enjambre	47
Tabla 8. Resultados obtenidos en las ganancias	51

CAPÍTULO 1

1. Introducción

1.1. Antecedentes

El sistema de péndulo invertido en un carro es uno de los problemas de referencia con mayor interés, en científicos e investigadores, para su estudio debido a su simplicidad de construcción, alto grado de no linealidad y por ser un sistema sub-actuado (El-Hawwary et al., 2006). Por lo que dicho sistema es usado para determinar nuevas leyes y estrategias de control que permitan mejorar la respuesta y rendimientos deseados de procesos o sistemas automatizados (Prasad et al., 2014).

El estudio del péndulo invertido tiene su historia desde los años 1960 cuando J. K. Robergeen desarrolló la primera solución a la estabilización del Péndulo invertido utilizando un controlador lineal (Roca García, 2015), siendo aquel aporte, una puerta para nuevos estudios y contribuciones a este tipo de sistemas no lineales.

En la literatura, se pueden encontrar distintas configuraciones del sistema péndulo invertido y el estudio de cada método de control (Glück et al., 2013), como por ejemplo, el péndulo invertido simple (Prasad et al., 2014), el péndulo invertido doble (Graichen et al., 2007) y el péndulo invertido triple (Eltohamy & Kuo, 1998; Glück et al., 2013; Medrano-Cerda, 1997; Tsachouridis & Medrano-Cerda, 1999). Sin embargo, este trabajo sólo se basa en la última configuración presentada anteriormente por ser el sistema con mayor grado de dificultad debido a su considerable aumento de la inestabilidad.

Las técnicas de control empleadas habitualmente para este tipo de sistemas constan de controladores LQR (Regulador Lineal cuadrático), autoajustables, y controladores robustos. Sin embargo, en las últimas dos décadas se han incorporado técnicas de control empleando inteligencia Artificial (AI), como redes neuronales, controladores difusos, basado en reglas inciertas semejantes al comportamiento humano y técnicas de optimización.

A lo largo de los años se han presentado diferentes soluciones a la estabilización del Triple Péndulo Invertido (TIP) ya sea empleando un esquema de control de dos grados de libertad con la incorporación de un controlador de retroalimentación no lineal y un controlador de retroalimentación óptima (Glück et al., 2013), o mediante un controlador basado en la teoría del regulador lineal de tiempo discreto introducido por medio de un observador de orden reducido (Medrano-Cerda, 1997). Un controlador de retroalimentación de una sola entrada y un controlador de tiempo discreto basado en la teoría H utilizando un observador dinámico robusto también tuvieron su participación en la estabilización del sistema TIP. (Eltohamy & Kuo, 1998; Tsachouridis & Medrano-Cerda, 1999).

1.2. Planteamiento del problema

El sistema de Péndulo Triple Invertido es un dispositivo físico que consta de tres eslabones unidos entre sí por medio de pivotes fijos, oscilando libremente alrededor de las juntas. El primer eslabón se encuentra montado por su parte inferior sobre un dispositivo comúnmente denominado "carro", el cual se mueve por medio de una señal de control de forma horizontal. El propósito de la señal de control es mover el carro hasta lograr mantener los tres péndulos en una posición vertical.

A pesar de que ya existen soluciones para estabilizar el sistema TIP como se ha descrito anteriormente empleando diferentes estrategias de control, en la literatura existe una falta de investigación en la estabilización del péndulo triple invertido usando lógica difusa, ya que lograr la sintonización de los factores de escalas del controlador difuso resulta muy tedioso y puede llevar mucho tiempo.

1.3. Justificación

Hoy en día existen muchas aplicaciones en las que los controladores basados en la estabilización del péndulo invertido han sido empleados, como por ejemplo, en la simulación del equilibrio humano (Chang et al., 2020), lanzamiento y control de misiles (Krzysztofik & Koruba, 2012), entre otros.

Por tal motivo, debido al rápido crecimiento en la automatización y sistemas inteligentes, es necesario mejorar y buscar nuevas estrategias de control, que permitan aumentar la eficiencia de estos controladores. Una técnica de control que ha permitido en los últimos años ofrecer una excelente respuesta a sistemas lineales y no lineales es la técnica de lógica borrosa o difusa (Belyadi & Haghghat, 2021), ganando un amplio interés en la teoría de control y en particular en la estabilización del péndulo invertido simple (El-Hawwary et al., 2006; Roose et al., 2017), doble (Cheng et al., 1996) y triple (Chen & Theodomile, 2016) logrando incluso combinarse con otras técnicas de control, desarrollando controladores más eficientes y precisos (Hazem et al., 2020).

Sin embargo, lograr el ajuste de los parámetros de un controlador difuso no siempre es sencillo y eficiente, los métodos de prueba y error pueden requerir mucho tiempo (Hurel et al., 2012). Por ello, en los últimos años se han desarrollado y aplicado con éxito métodos de inteligencia bioinspirada para sintonizar los controladores de un péndulo invertido, como por ejemplo, la optimización por enjambre de partículas (PSO) (Çakan et al., 2017; Hamza et al., 2015; Shima & Bashir, 2021).

1.4. Objetivo general

Modelar un sistema de Péndulo Triple Invertido sobre un carro, para un análisis comparativo de la estabilidad mediante el uso de Lógica borrosa y PSO.

1.4.1. Objetivos específicos.

- Desarrollar el modelo dinámico del sistema a partir de las ecuaciones de movimiento del Péndulo triple invertido haciendo uso de la herramienta Matlab – Simulink.
- Desarrollar los controladores LQR y Borroso - PSO para el sistema TIP.
- Analizar y comparar la estabilización del péndulo triple invertido mediante un controlador LQR, un sintonizador de ganancias PSO y un controlador difuso optimizado mediante PSO.

CAPÍTULO 2

2. Revisión literaria

Para desarrollar el modelado del sistema Péndulo Triple Invertido es necesario definir la base teórica que sustenta el trabajo realizado, exponiendo documentación e investigaciones existentes. Los tipos de configuraciones del péndulo invertido que existen, las bases teóricas de las estrategias del control LQR, la técnica de control difusa y del algoritmo de optimización de enjambre de partículas (PSO).

2.1. El triple péndulo invertido sobre un carro.

El péndulo invertido triple en un carro dentro de la teoría de control es un sistema de gran relevancia para su estudio por ser un sistema inestable, de alto grado de no-linealidad, subactuado y por su simplicidad en la construcción, usado para determinar nuevas leyes y estrategias de control que permiten mejorar la respuesta y rendimientos deseados de procesos o sistemas complejos (Prasad et al., 2014).

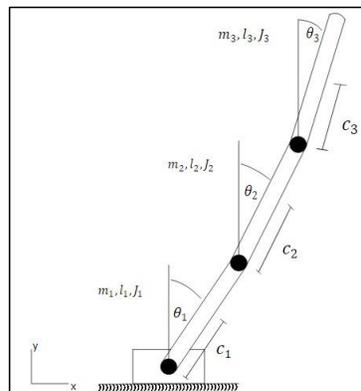


Figura 2. 1 Esquema del péndulo triple invertido

Fuente: Autor

El sistema de Triple Péndulo Invertido se muestra en la figura 1, y se compone de tres eslabones cilíndricos rígidos cuya longitud está representada por l_1, l_2, l_3 respectivamente. Estos eslabones se encuentran conectados por medio de juntas de revoluta. El primer eslabón, en su articulación inferior, está conectada con un carro que se mueve a lo largo del eje horizontal debido a la señal de control que se ejerce para la acción programada. La masa de los eslabones es representada por m_1, m_2, m_3 respectivamente. Los ángulos que forman los eslabones con respecto al eje vertical son representados por $\theta_1, \theta_2, \theta_3$, donde el objetivo es conseguir que estos ángulos sean iguales a cero para que el sistema se encuentre en equilibrio, es decir, los eslabones alineados de forma vertical superior.

Dentro de este grupo de sistemas se encuentran 2 configuraciones comúnmente estudiadas como son:

- Péndulo Invertido Simple (IP).
- Péndulo Invertido Doble (DIP).

Donde su estructura y funcionamiento es igual al sistema TIP con la única diferencia que el número de eslabones disminuye de acuerdo con su configuración.

2.2. Sistemas de control y su clasificación

Sistemas de control es un conjunto de dispositivos con la finalidad de regular, coordinar o dirigir el comportamiento de otro sistema, con el objetivo de obtener resultados deseados con el menor porcentaje de error posible (Ogata, 2010). Comúnmente existen dos tipos de sistemas de control que son:

- Sistemas de control en lazo abierto (SCA)
- Sistemas de control en lazo cerrado (SCC)

2.2.1. Sistemas de control en lazo abierto.

Son aquellos sistemas o esquemas de control en los que la salida no presenta ningún efecto sobre la señal de control que se produce, por tal motivo en este tipo de sistemas no es necesario medir la salida y compararla con la entrada (Gutiérrez & Iturralde, 2017).

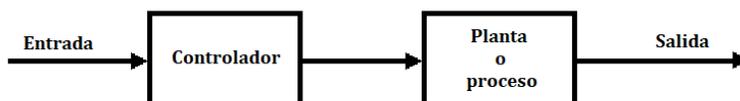


Figura 2. 2 Esquema del sistema de control en lazo abierto

Fuente: Ogata, 2010

2.2.2. Sistemas de control en lazo cerrado

Comúnmente son llamados sistemas con realimentación, y la señal de salida influye plenamente en la señal de control del sistema que es la señal de error. Básicamente es la diferencia entre la señal de entrada y la señal de salida (Gutiérrez & Iturralde, 2017).

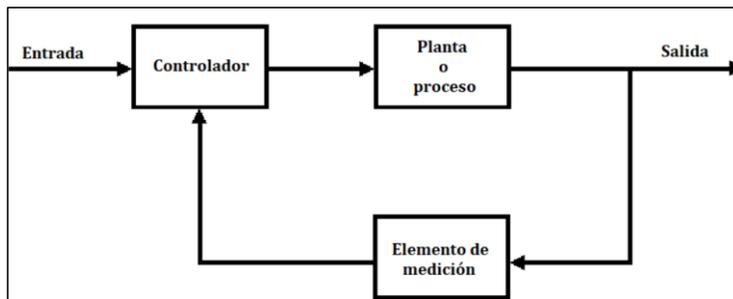


Figura 2. 3 Esquema del sistema de control en lazo cerrado

Fuente: Ogata, 2010

2.3. Elementos de un sistema de control.

Dentro de un sistema de control intervienen una serie de elementos que a continuación serán definidos de forma clara y breve.

- **Planta:** Es cualquier objeto físico que pueda realizar una operación o proceso. Por ejemplo, un reactor químico, un horno eléctrico, un péndulo invertido, entre otros.
- **Proceso:** Conjunto de operaciones secuenciales para obtener un cambio de estado o fase. Dicho proceso deberá ser controlado. Por ejemplo, un proceso químico, biológico, entre otros.
- **Señal de entrada:** Señal que ingresa al esquema del control y que es utilizada para excitar el sistema.
- **Señal de Salida:** Es la respuesta que se obtiene del sistema.

2.4. Modelado de sistemas de control.

Un modelo de un sistema es capaz de describir el comportamiento de un sistema dado. El modelar un sistema puede tener múltiples representaciones, ya sea empleando leyes numéricas, gráficas o analógicas (Llata García et al., 2017)

Para un mismo sistema físico existen varios modelos apropiados. Los modelos más útiles y comunes pueden clasificarse en las siguientes categorías:

- **Analogía directa:** Son modelos en los que se aplica una réplica del sistema, ya sea a escala o no.
- **Representación gráfica:** Son modelos que se representan mediante diagrama de bloques, diagrama de flujos de señal y ayudan a visualizar la relación que existe entre la entrada y la salida.
- **Representación Matemática:** Mediante la aplicación de leyes numéricas se puede describir el comportamiento de un sistema. Por ejemplo, ecuaciones diferenciales, ecuaciones de estado, funciones de transferencias, representaciones matriciales entre otros. Por lo general todos los sistemas físicos en la vida real son no lineales y en muchas ocasiones obtener la representación no lineal resulta una tarea muy compleja, por tal motivo, es necesario recurrir a la linealización del sistema dentro de un dominio de funcionamiento siempre y cuando esté permitido, para facilitar el análisis matemático (Pérez et al., 2008).

El esquema de la figura 2.4 muestra el proceso para encontrar la representación matemática de un sistema lineal y no lineal.

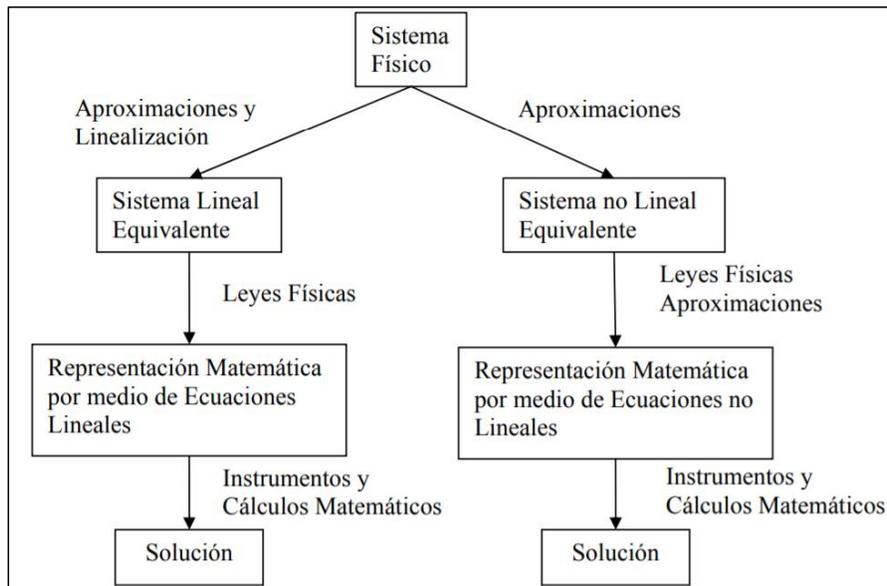


Figura 2. 4 Esquema de la representación matemática de un sistema

Fuente: Pérez, 2008

2.4.1. Representación en el espacio de estados.

La representación en el espacio de estados es un modelo matemático que describe un sistema físico que se combina con ecuaciones diferenciales matriciales de primer orden. Nos brinda un esquema compacto, claro y conveniente de modelar y analizar con múltiples entradas y salidas.

El subconjunto más pequeño de variables de un sistema que puede representar un estado dinámico en un instante determinado se conoce como variables de estado. Las variables de estado contienen suficiente información como para predecir el comportamiento de un sistema en ausencia de excitaciones externas (Ramos et al., 2016).

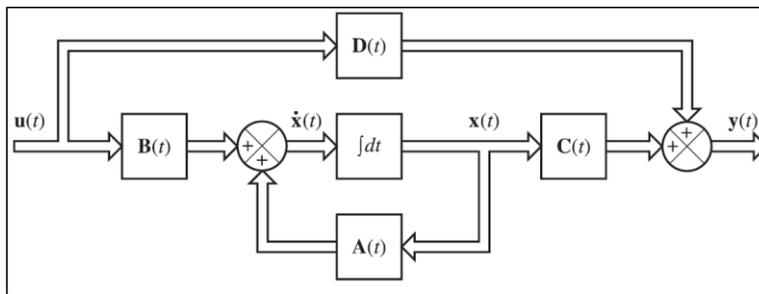


Figura 2. 5 Representación en espacio de estados mediante diagrama de bloques

Fuente: Ogata, 2010

La forma general para representar los espacios de estados de un sistema lineal está dada por la ecuación (1) y (2):

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

$$y(t) = Cx(t) + Du(t) \quad (2)$$

En donde:

$x(t)$ = es el vector de estados

$y(t)$ = es el vector de salidas

$u(t)$ = es el vector de entradas

$A(t)$ = es la matriz de estados

$B(t)$ = es la matriz de entrada

$C(t)$ = es la matriz de salida

$D(t)$ = es la matriz de transmisión directa

2.4.1.1. Controlabilidad

Se dice que un sistema es completamente controlable si existe una señal $u(t)$ que permite transferir los estados iniciales del sistema $x_0 = x(t_0)$ a cualquier otro estado $x_{t_f} = x(t_f)$ en un tiempo finito $T = t_f - t_0$ (Carrillo Paz, 2011).

$$\text{rank}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n \quad (3)$$

2.4.1.2. Observabilidad

Se dice que un sistema es completamente observable si el estado $x(t_0)$ se determina a partir de la observación de $y(t)$ durante un intervalo finito $t_0 \leq t \leq t_1$, el sistema es completamente observable si todas las transiciones de estado afectan eventualmente a todos y cada una de las variables de salidas o al vector de salida (Carrillo Paz, 2011).

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4)$$

2.5. Teoría del control óptima

Los trabajos presentados por Bellman, Kalman y Pontryagin fomentaron las bases de la Teoría de Control Moderno u óptimo. La formulación del control óptimo es basada en la utilización de variables de control que logran maximizar o minimizar una función objetivo sujeta a restricciones dadas (Serrano Malagón & Álvarez León, 2009).

2.6. Controladores basados en la teoría de control óptimo

En los años setenta, los controladores de horizonte finito empiezan a ser tomados en cuenta y en los años ochenta se empiezan a realizar investigaciones con especial detalle con capacidades de predicción. Actualmente, se pueden combinar estos controladores con inteligencia artificial, lógica difusa, y minería de datos (Alvarado Anangón & Gonzales Zurita, 2018)

2.6.1. Regulador Optimo no lineal.

Es considerado aún como un problema abierto en la matemática donde se busca desarrollar un algoritmo preciso para obtener un controlador no lineal óptimo, donde el problema más importante es tratar de resolver la ecuación diferencial parcial no lineal de Hamilton – Jacobi Bellman (Holzhüter, 2004).

El regulador no lineal óptimo considera una ecuación diferencial del tipo:

$$\dot{x} = f(x) + g(x) \cdot u \quad (5)$$

Donde el control óptimo es representado por u . La minimización del funcional dado por el criterio puramente cuadrático viene dado por:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (6)$$

El problema es de horizonte infinito, es decir que no está limitado en el tiempo, en donde, el control óptimo puede obtenerse en la teoría de cálculo variacional con la siguiente expresión (Alvarado Anangón & Gonzales Zurita, 2018):

$$u = -R^{-1} g^T(x) p \quad (7)$$

El co-estado p puede obtenerse a partir de la derivada del costo óptimo V . Donde la ecuación de co-estado se representa de la siguiente manera:

$$\dot{p} = - \left(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial x} u \right)^T p - Q x \quad (8)$$

Es preciso mencionar que este controlador no lineal en teoría es el mejor, ya que considera todas las variables dinámicas del sistema de estudio, con sus comportamientos completos y no restringidos a un conjunto de estados (Bryson et al., 2008).

2.6.2. Regulador lineal cuadrático (LQR).

El regulador cuadrático lineal (LQR) es un método de control que como su nombre lo indica, actúa en sistemas linealizados y está representado por las ecuaciones (1) y (2).

En donde A es la dinámica de los estados del sistema representada en forma matricial, B es la relación de la entrada con los estados del sistema, y u es la variable de control.

En este método de control también se utiliza el índice de rendimiento y las variables de estado para calcular la entrada de control óptima. El índice de rendimiento J , que incluye los errores de estado y la entrada del sistema, se indica en la siguiente ecuación:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt + \frac{1}{2} x^T(T) P_1 x(T) \quad (9)$$

Donde Q es el error del estado, mientras que R es el esfuerzo del controlador, y el término $\frac{1}{2} x^T(T) P_1 x(T)$ representa una penalización en el controlador debido a que en la práctica es imposible tener un controlador ideal, este término puede ser despreciado dependiendo del autor (Ávila Ramos & Suarez Cárdenas, 2012).

Las matrices Q y R son matrices reales, simétricas, constantes y definidas positivas que en la mayoría de los casos son escogidas de forma heurísticas.

$$Q = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & q_n \end{bmatrix} \quad R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & r_m \end{bmatrix}$$

Para escoger estas matrices hay que tener en cuenta lo siguiente: Si Q es más grande que R , la regulación de los estados es más rápida. Mientras que si R es más grande que Q , la regulación de los estados es más lenta. Para determinar el cambio óptimo de control, la función de Hamilton se deriva y se iguala a cero para obtener:

$$u(t) = -Kx(t) = R^{-1}B^T P x(t) \quad (10)$$

Donde el valor de P se determina mediante la ecuación algebraica de Riccati, dada como:

$$(PA - PBR^{-1}B^T P + Q + A^T P) = 0 \quad (11)$$

En la figura 2.6 se muestra el esquema en lazo cerrado del controlador LQR.

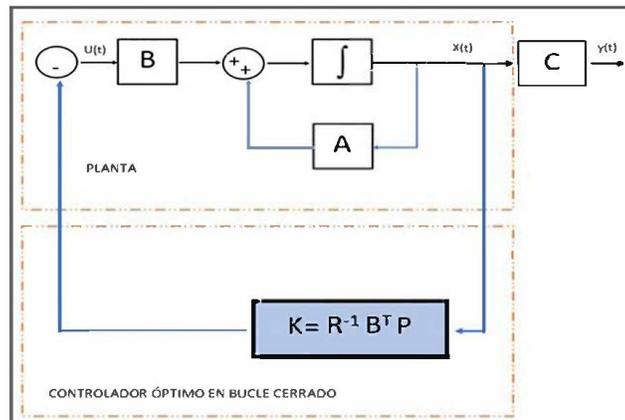


Figura 2. 6 Esquema del controlador LQR en lazo cerrado

Fuente: Autor

2.6.3. Controlador LQR con seguimiento de referencia

El principio es el mismo en comparación al caso anterior, con la única diferencia que se tiene una señal de referencia a la cual se desea llegar, de tal modo que el índice de rendimiento a minimizar toma la siguiente forma:

$$J = \int_0^{\infty} (y_{ref} - y(t))^T Q (y_{ref} - y(t)) + u(t)^T R u(t) dt \quad (12)$$

Donde la entrada del sistema se expresa como:

$$u(t) = K(ref - x) \quad (13)$$

En la figura 2.7 se muestra el esquema en lazo cerrado del controlador LQR con seguimiento de referencia.

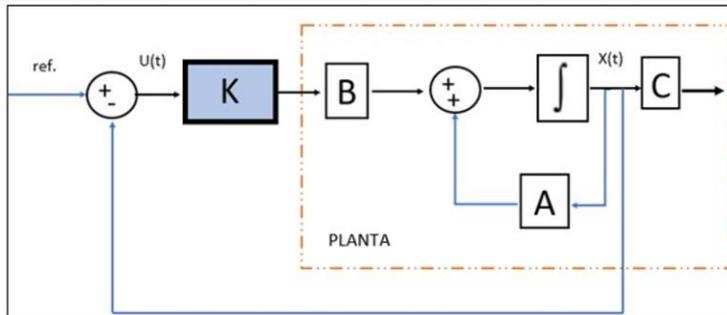


Figura 2. 7 Esquema del controlador LQR con seguimiento de referencia

Fuente: Autor

2.7. Lógica difusa.

L.A. Zadeh inició la teoría de control con la publicación de su trabajo "Fuzzy Sets" en 1965. La lógica difusa es una herramienta matemática llena de incertidumbres ya que proporciona una técnica para tratar la imprecisión y la granularidad de la información. En otras palabras, esta técnica matemática permite una capacidad de razonamiento humano adecuada basándose en la noción de pertenencia relativa (Liu, 2018).

Los conjuntos difusos ofrecen recursos capaces de modelar la incertidumbre asociada a la vaguedad, la imprecisión y la poca información sobre una planta, etc.

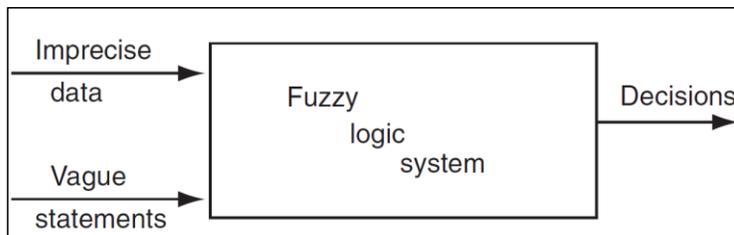


Figura 2. 8 Sistema difuso que acepta datos imprecisos y afirmaciones vagas

Fuente: Sivanandam et al., 2006

2.7.1. Función de pertenencia

En lógica difusa, la función de pertenencia de un conjunto nos indica el grado que cada elemento de un universo dado pertenece a dicho conjunto. Las funciones de pertenencia son una forma de representar gráficamente un conjunto borroso sobre un universo. Lotfi Zadeh propuso la idea de la pertenencia a un conjunto para tomar decisiones adecuadas cuando se produce la incertidumbre. Si el conjunto es nítido, su función de pertenencia tomará valores en $[0, 1]$. Mientras que, si es un conjunto borroso, tomará valores dentro del intervalo de $[0, 1]$ (Olmo Castillo, 2008).

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

Donde $\chi_A(x)$ es la pertenencia del elemento x al conjunto A y por su parte A es el conjunto completo del universo. Existe una gran variedad de formas para las funciones de membresía, las más comunes son del tipo trapezoidal, triangular, singleton, forma S (LÓGICA DIFUSA Y SISTEMAS DE CONTROL, n.d.).

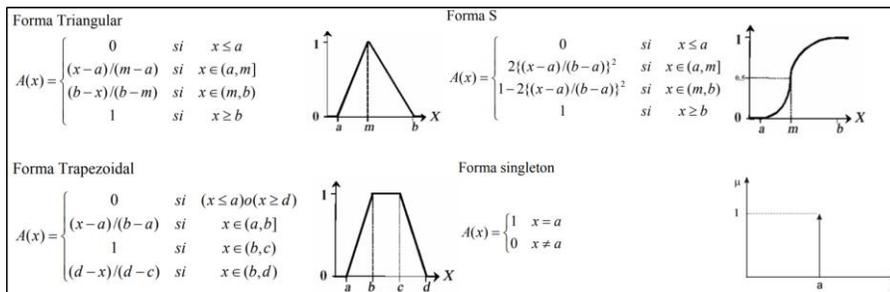


Figura 2. 9 Formas comunes de las funciones de pertenencias

Fuente: LÓGICA DIFUSA Y SISTEMAS DE CONTROL, n.d.

2.7.2. Características de la función de pertenencia.

Una función de pertenencia está definida por tres propiedades básicas que son:

- **Núcleo:** Los elementos que tienen la función de pertenencia como 1, son los elementos del núcleo, es decir, aquí $\mu_A(x) = 1$.
- **Soporte:** Si la región del universo se caracteriza por la pertenencia no nula al conjunto A , esto define el soporte de una función de pertenencia para el conjunto difuso A . El soporte tiene los elementos cuya membresía es mayor que 0 es decir $\mu_A(x) > 0$.

- **Límite:** Si la región del universo tiene una membresía no nula pero no una membresía completa, esto define el límite de una membresía para el conjunto difuso A. La frontera tiene los elementos cuya membresía está entre 0 y 1, es decir $0 < \mu_A(x) < 1$.

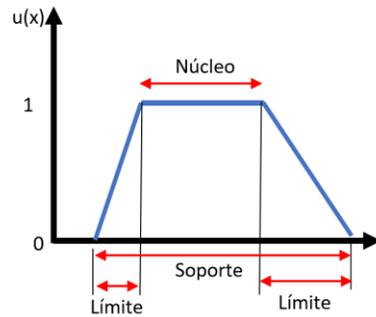


Figura 2. 10 Características de la función de pertenencia

Fuente: Autor

2.7.3. Operaciones con conjuntos difusos.

Las operaciones básicas de la lógica difusa se definen como:

- Unión: $\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$
- Intersección: $\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$
- Complemento: $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

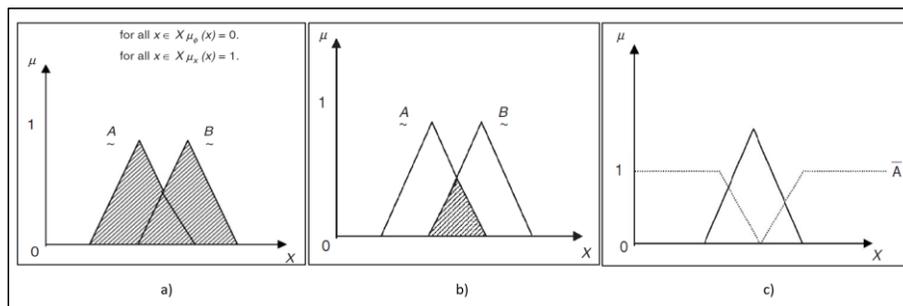


Figura 2. 11 a) Unión. b) Intersección. c) Complemento

Fuente: Sivanandam et al., 2006

2.7.4. Fuzzificación.

Este término es un concepto muy relevante en la teoría difusa, ya que la fuzzificación es el proceso por el cual se logra “traducir” una variable numérica, en una concepción difusa de dicho valor. Así, el proceso de fuzzificación puede implicar la asignación de valores de pertenencia para las cuantificaciones de los críticos dados (Bouza & Alimenti, 1997).

2.7.5. Defuzzificación.

La defuzzificación se encuentra en la etapa final del proceso de aplicación de la lógica difusa y se refiere a las conversiones de difuso a nítido. Los resultados difusos generados no pueden utilizarse como tales en las aplicaciones, por lo que es necesario convertir las cantidades difusas en salidas con valor numérico único (Lara-Valencia et al., 2015).

2.7.6. Métodos de defuzzificación.

Existen métodos de defuzzificación que se emplean para convertir las cantidades difusas en cantidades nítidas. La salida de un proceso difuso completo puede ser la unión de dos o más funciones de membresía difusas. A continuación, se presentan los métodos más utilizados para defuzzificar (Sivanandam et al., 2006). Estos son:

- **Principio de Max-membresía o método de la altura:** Este método está dado por la expresión $\mu_c(z^*) \geq \mu_c(Z)$ para todo $Z \in Z$.
- **Método del centroide:** Este es el método más utilizado y comúnmente se lo conoce como método del centro de gravedad o del centro de área y se lo define mediante la ecuación (14)

$$z^* = \frac{\int \mu_c(Z)ZdZ}{\int \mu_c(Z)dZ} \quad (14)$$

- **Método de la media ponderada:** Este método no puede utilizarse para funciones de membresía de salida asimétrica, solo puede utilizarse para funciones de membresía de salida simétrica. Este método se representa por la ecuación (15).

$$z^* = \frac{\sum \mu_c(\bar{z})\bar{z}}{\sum \mu_c(\bar{z})} \quad (15)$$

- **Pertenencia media máxima:** Este método está relacionado con el principio de pertenencia máxima. Este método también se denomina método del medio de los máximos y su expresión es la siguiente

$$z^* = \frac{a + b}{2} \quad (16)$$

La figura 2.11 muestra la representación gráfica de cada método mencionado.

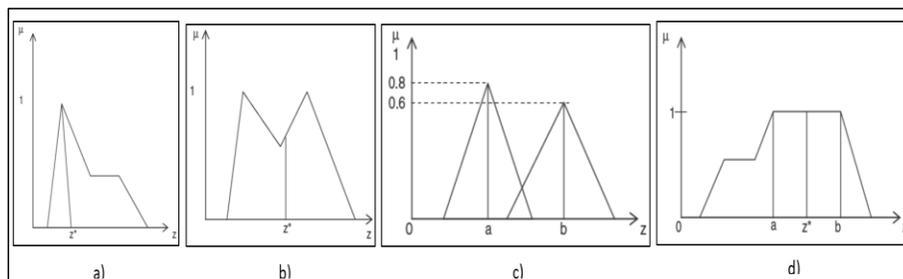


Figura 2. 12 Métodos: a) altura. B) centroide. C) media ponderada. d) media máxima

Fuente: Sivanandam et al., 2006

2.7.7. Sistema de Inferencia difusa

Los sistemas de inferencia difusa (FIS) son conocidos también como sistemas expertos difusos basados en reglas difusas. Se trata de una unidad principal de un sistema de lógica difusa

2.7.7.1. Construcción y funcionamiento del sistema de inferencia

Este sistema consta de una etapa inicial de fuzzificación donde los valores nítidos ingresan a esta interfaz para ser transformados en valores difusos, luego ingresan al proceso de creación de base de reglas y base de datos, para obtener una toma de decisiones donde una base de reglas contiene un número de reglas difusas IF-THEN; y una base de datos define las funciones de pertenencia de los conjuntos difusos utilizados en las reglas difusas. Por último estos valores ingresan a la etapa de defuzzificación para obtener valores numéricos nítidos (Sivanandam et al., 2006).

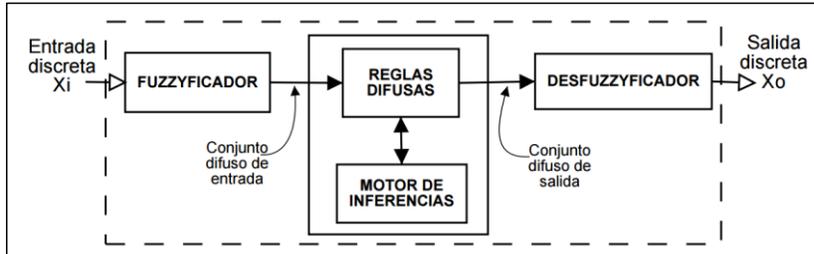


Figura 2. 13 Sistema de inferencia difusa

Fuente: Sancho Caparrini, 2019

2.7.7.2. Métodos de inferencia difusa.

Existen algunos métodos de inferencia difusa sin embargo se presentarán dos métodos que son considerados los más importantes dentro de la teoría difusa.

El método de Inferencia difusa Mandani es tal vez el más utilizado. Fue introducido en el año 1975 por Mandani y Assilian y utiliza un conjunto de reglas difusas "SI-ENTONCES" (IF-THEN). Toma como entrada los valores de la fuzificación y son aplicadas a los antecedentes de las reglas difusas. Si una regla tiene múltiples antecedentes, se utiliza el operador AND u OR para obtener un único número que represente el resultado de la evaluación.

El método Takagi-Sugeno-Kang o T-S fue introducido en el año 1985 por Sugeno y se basa en reglas difusas, pero el consecuente no nos da un conjunto difuso sino una serie de funciones lineales. Este modelo es útil para sistemas complejos y de dimensiones mayores de los que podemos resolver por el método de Mamdani. La principal diferencia que presenta el método TSK respecto al de Mamdani es que no es necesario realizar un proceso de defuzzificación.

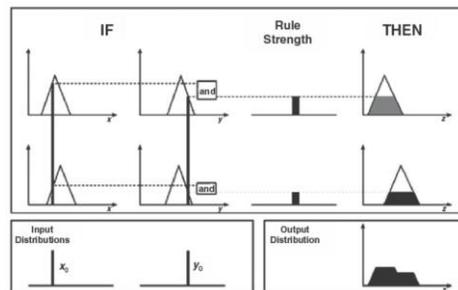


Figura 2. 14 Representación gráfica de sistema de inferencia difusa

Fuente: Sivanandam et al., 2006

2.8. Optimización por enjambre de partículas (PSO)

La optimización de enjambres de partículas (PSO) es un algoritmo de optimización inteligente metaheurístico desarrollado en el año 1995 el cual permite imitar el comportamiento de los enjambres biológicos, como bandadas de pájaros y enjambres de abejas (Kennedy et al., 1995).

Una característica importante de estos algoritmos es que no son determinísticas, es decir que a pesar de tener la misma función objetivo siempre vamos a tener resultados diferentes. Además, la calidad que llegan a ofrecer en sus resultados es de un 99% con un costo computacional muy bajo en comparación a los algoritmos genéticos (Velázquez Dodge & Mejía Lavalle, 2016).

2.8.1. Implementación del factor social, cognitivo y peso de inercia.

En 2006 se trató de definir un algoritmo estándar. Sin embargo, a lo largo de los años éste ha sufrido varias modificaciones de acuerdo con el problema que se desea optimizar, por ejemplo, diseño de controladores e identificación de sistemas, incorporando el factor social, cognitivo el peso de inercia y el modo en que las partículas se comunicaban por medio de topologías definidas (Velázquez Dodge & Mejía Lavalle, 2016).

2.8.1.1. Factor cognitivo y social

La componente cognitiva ($c_1 r_1$) contribuye para que la partícula tenga una especie de memoria y es la responsable de mover la partícula hacia la posición donde ha obtenido mejores resultados hasta el momento (Benítez et al., 2013).

El valor que puede tomar el coeficiente cognitivo c_1 se encuentra entre [0.2], siendo el último el que se recomienda comúnmente, mientras que r_1 es un número aleatorio que se encuentra dentro del rango de [0.1] aportando cierto comportamiento estocástico, de tal manera que ofrece la oportunidad de no caer en mínimos locales (Rodrigo, 2019).

Mientras que la componente social ($c_2 r_2$), es la que provoca que la partícula se sienta atraída por la mejor posición encontrada por el enjambre hasta ese momento (Benítez et al., 2013). Como en el caso anterior los valores que toman c_2 y r_2 se encuentran dentro del rango [0.2] y [0.1].

La relación entre la componente social y cognitiva dentro del algoritmo PSO es la que determina el comportamiento exploratorio del enjambre. Ya que si c_1 es mayor respecto a c_2 , las partículas tendrán mayor independencia en su movimiento, lo que permite un mayor rango de exploración pero el tiempo de convergencia será mayor. Mientras que si c_2 es mayor que c_1 , las partículas pierden la capacidad de moverse independiente, lo que conduce a una reducción del espacio a explorar pero su convergencia es mucho más rápida (Rodrigo, 2019).

2.8.1.2. Factor de Inercia

Este componente de inercia (w) fue introducido para mantener a las partículas moviéndose en la dirección actual. Dentro de las investigaciones previas, el valor recomendado para el coeficiente de inercia está comprendido entre 0.8 y 1.2. Sin embargo, si $w < 1$, la convergencia es mucho más rápida pero el espacio de exploración es menor debido a que las partículas van desacelerando mientras las iteraciones avanzan. Por otra parte, si $w > 1$, ocurre lo contrario (Rodrigo, 2019).

2.8.2. Procedimiento del algoritmo

En esta sección se describe la secuencia del algoritmo PSO, presentado en el diagrama de flujo mostrado en la figura 2.15.

En el proceso de iniciación se establecen los parámetros del algoritmo, como tamaño del enjambre número de partículas, posición inicial, valores de los factores social y cognitivo. Luego el algoritmo realiza el proceso de evaluación de la población inicial, y la mejor partícula global ($Gbest$) es seleccionada.

Para realizar la actualización de la posición y velocidad del enjambre se utilizan las ecuaciones (17) y (18) respectivamente. Luego la posición actualizada es evaluada de acuerdo con la función objetivo establecida, de tal manera que las partículas $Gbest$ y $L_i best$ son actualizadas si es necesario.

Luego el algoritmo pasa por el proceso de verificación de cumplimiento de la condición de parada, de tal manera que, si la condición establecida ya sea por el número de iteraciones o por el valor mínimo o máximo de la función objetivo ha sido cumplida, el algoritmo se da por finalizado, caso contrario el algoritmo entra de nuevo en el proceso de actualización de velocidad y posición del enjambre.

$$v_i(k + 1) = w * v_i(k) + c_1 * r_1(k) * (L_i best - p_i(k)) + c_2 * r_2(k) * (Gbest - p_i(k)) \quad (17)$$

$$p_i(k + 1) = p_i(k) + v_i(k) \quad (18)$$

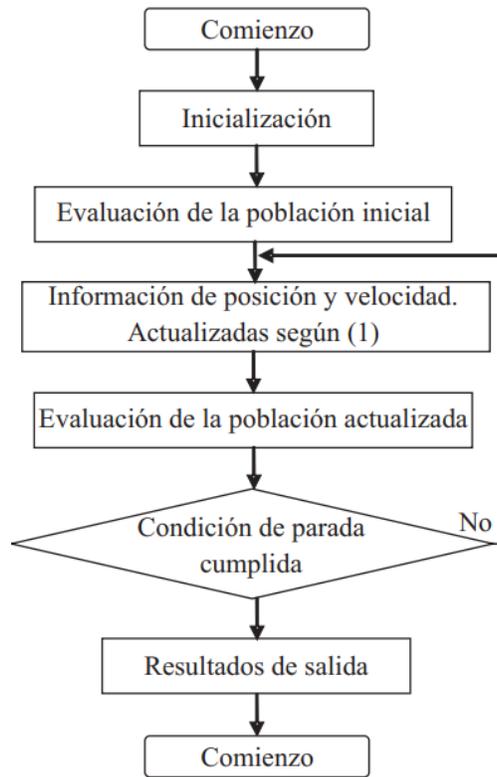


Figura 2. 15 Diagrama de flujo del algoritmo PSO

Fuente: Duarte & Quiroga, 2010

CAPÍTULO 3

3. Metodología

En favor de alcanzar los objetivos de la presente investigación, la figura 3.1 muestra los pasos para realizar el análisis comparativo de las estrategias de control aplicadas para la estabilización del Triple Péndulo Invertido.

Como punto inicial se establecen los parámetros de diseño del TPI mostrados en la Tabla 1, para ser introducidos en las ecuaciones del modelo matemático. El modelo del sistema es definido mediante las ecuaciones de Lagrange obteniendo de esta manera las ecuaciones de movimiento no lineales para cada uno de los estados (péndulos 1, 2, 3 y carro).

Una vez obtenido el modelo no lineal del sistema, es implementado en Matlab/Simulink, para realizar la linealización de sistema y luego ser representado en espacio de estados en Simulink. De tal manera, se procede a diseñar e implementar las estrategias de control por separado para al final obtener su respectivo análisis y comparación con respecto al control de la estabilidad del sistema TIP.

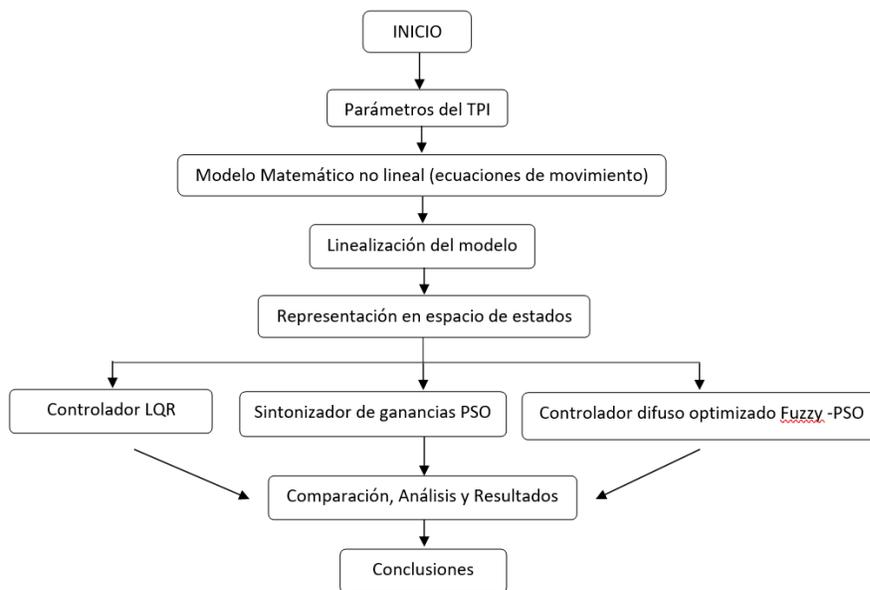


Figura 3. 1 Esquema metodológico para el análisis comparativo del sistema TIP

Fuente: Autor.

Tabla 1.
Parámetros de diseño del Péndulo Triple Invertido

Descripción	Valores	Parámetros	Unidades
Masa del Carrito	1	m_c	kg
Masa del Péndulo 1	0.876	m_1	kg
Masa del Péndulo 2	0.938	m_2	kg
Masa del Péndulo 3	0.553	m_3	kg
Longitud del Péndulo 1	0.323	l_1	m
Longitud del Péndulo 2	0.419	l_2	m
Longitud del Péndulo 3	0.484	l_3	m
Centro de Masa Péndulo 1	0.215	c_1	m
Centro de Masa Péndulo 2	0.269	c_2	m
Centro de Masa Péndulo 3	0.226	c_3	m
Momento de Inercia Péndulo 1	0.013	J_1	Nms^2
Momento de Inercia Péndulo 2	0.024	J_2	Nms^2
Momento de Inercia Péndulo 3	0.018	J_3	Nms^2
Fricción de carro con superficie	0	d_0	Nm
Fricción Viscosa Junta 1	0.215	d_1	Nm
Fricción Viscosa Junta 2	0.002	d_2	Nm
Fricción Viscosa Junta 3	0.002	d_3	Nm
Gravedad	9.81	g	m/s^2

Fuente: Glück et al., 2013

3.1. Modelo Matemático no lineal del triple péndulo.

El modelado matemático del sistema se establece mediante el principio de Lagrange y no por la segunda ley de Newton como comúnmente se determinan los modelos de los sistemas. La mecánica Lagrangiana utiliza un conjunto de parámetros que sirven para determinar la configuración de un sistema mecánico con ciertos grados de libertad. En este trabajo, la fricción del carro con la superficie es despreciada.

Comentado [UdW2]: No se comenta la tabla 1

Comentado [JDAB3R2]: Está comentada en la página anterior, al final de la primera línea del segundo párrafo

Por lo tanto, el primer paso para encontrar el modelo matemático del triple péndulo empleando mecánica lagrangiana, es estableciendo las coordenadas generalizadas de acuerdo con los grados de libertad. El sistema presenta cuatro grados de libertad, el desplazamiento del carro de manera horizontal y el movimiento de cada uno de los péndulos $(x_c, \theta_1, \theta_2, \theta_3)$, de esta manera se tendrán cuatro coordenadas generalizadas (q_0, q_1, q_2, q_3) . La formulación de Lagrange se representa por la ecuación (19).

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_{\delta i} \quad (19)$$

Sin embargo, para el desarrollo del modelo matemático del péndulo triple, se incorpora una función de disipación conocida como "Función de disipación de Rayleigh", que incorpora las fuerzas viscosas o de rozamiento en este caso de las juntas o rotulas de los péndulos. Por lo tanto, la ecuación de Lagrange a utilizar en esta investigación queda definida como:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} + \frac{\partial \mathcal{R}}{\partial \dot{q}_i} = \tau_{\delta i} \quad i = 0, 1, 2, 3 \quad (20)$$

Donde:

\mathcal{L} = Lagrangiano del sistema

\mathcal{R} = Función de disipación de Rayleigh

q_i = Coordenadas generalizadas

$\tau_{\delta i}$ = Fuerzas aplicadas al sistema

El Lagrangiano está dado por la diferencia entre la energía cinética (T) y la energía potencial del sistema (V), como:

$$\mathcal{L} = T - V \quad (21)$$

La energía cinética del carro es:

$$T_0 = \frac{1}{2} m_c \dot{q}_0^2 \quad (22)$$

La energía cinética de los péndulos está formada por una parte traslacional y otra rotacional por el movimiento que generan. De esta manera, la energía cinética en los péndulos está representada por la ecuación (23).

$$T_i = \frac{1}{2} \sum_i^3 m_i \dot{p}_i^2 + \frac{1}{2} \sum_i^3 J_i \dot{q}_i^2 \quad i = 1,2,3 \quad (23)$$

De aquí P es la posición del centro de masa con respecto al marco de referencia [x,y]. La energía cinética del péndulo 1 está representada por:

$$T_i = \frac{1}{2} m_1 \dot{p}_1^2 + \frac{1}{2} J_1 \dot{q}_i^2 \quad (24)$$

La posición del centro de masa del péndulo 1 con respecto al marco de referencia [x,y] es: $P_1 = [q_0 + c_1 \text{sen} q_1, c_1 \text{cos} q_1]$. Por tanto, derivando esta ecuación, se obtiene:

$$\dot{P}_1 = [\dot{q}_0 + \dot{q}_1 c_1 \text{sen} q_1, -\dot{q}_1 c_1 \text{cos} q_1] \quad (25)$$

Reemplazando la ecuación (25) en la ecuación (24) se tiene que:

$$T_1 = \frac{1}{2} m_1 [\dot{q}_0 + \dot{q}_1 c_1 \text{sen} q_1, -\dot{q}_1 c_1 \text{cos} q_1]^2 + \frac{1}{2} J_1 \dot{q}_i^2 \quad (26)$$

Resolviendo, la energía cinética del péndulo 1 es:

$$T_1 = \frac{1}{2} m_1 \dot{q}_0^2 + \frac{1}{2} m_1 \dot{q}_1^2 c_1^2 + m_1 \dot{q}_0 \dot{q}_1 c_1 \text{cos} q_1 + \frac{1}{2} J_1 \dot{q}_i^2 \quad (27)$$

La energía cinética del péndulo 2 está representada por:

$$T_2 = \frac{1}{2}m_2\dot{P}_2^2 + \frac{1}{2}J_2\dot{q}_2^2 \quad (28)$$

La posición del centro de masa del péndulo 2 con respecto al marco de referencia $[x, y]$ es: $P_2 = [q_0 + l_1\text{sen}q_1 + c_2\text{sen}q_2, l_1\text{cos}q_1 + c_2\text{cos}q_2]$. Derivando se obtiene:

$$\dot{P}_2 = [\dot{q}_0 + l_1\dot{q}_1\text{cos}q_1 + c_2\dot{q}_2\text{cos}q_2, -l_1\dot{q}_1\text{sen}q_1 - c_2\dot{q}_2\text{sen}q_2] \quad (29)$$

Reemplazando la ecuación (29) en la ecuación (28) y resolviendo, la energía cinética del péndulo 2 es:

$$T_2 = \frac{1}{2}m_2\dot{q}_0^2 + \frac{1}{2}m_2l_1^2\dot{q}_1^2 + \frac{1}{2}m_2c_2^2\dot{q}_2^2 + m_2l_1\dot{q}_0\dot{q}_1\text{cos}q_1 + m_2c_2\dot{q}_0\dot{q}_2\text{cos}q_2 + m_2l_1c_2\dot{q}_1\dot{q}_2\text{cos}(q_2 - q_1) + \frac{1}{2}J_2\dot{q}_2^2 \quad (30)$$

La energía cinética del péndulo 3 está representada por:

$$T_3 = \frac{1}{2}m_3\dot{P}_3^2 + \frac{1}{2}J_3\dot{q}_3^2 \quad (31)$$

La posición del centro de masa del péndulo 3 con respecto al marco de referencia es: $P_3 = [q_0 + l_1\text{sen}q_1 + l_2\text{sen}q_2 + c_3\text{sen}q_3, l_1\text{cos}q_1 + l_2\text{cos}q_2 + c_3\text{cos}q_3]$. Derivando se obtiene:

$$\dot{P}_3 = [\dot{q}_0 + l_1\dot{q}_1\text{cos}q_1 + l_2\dot{q}_2\text{cos}q_2 + c_3\dot{q}_3\text{cos}q_3, -l_1\dot{q}_1\text{sen}q_1 - l_2\dot{q}_2\text{sen}q_2 - c_3\dot{q}_3\text{sen}q_3] \quad (32)$$

Reemplazando la ecuación (32) en la ecuación (31) y resolviendo, se tiene que la energía cinética del péndulo 3 es:

$$\begin{aligned}
T_3 = & \frac{1}{2}m_3\dot{q}_0^2 + \frac{1}{2}m_3l_1^2\dot{q}_1^2 + \frac{1}{2}m_3l_2^2\dot{q}_2^2 + \frac{1}{2}m_3c_3^2\dot{q}_3^2 + m_3l_1\dot{q}_0\dot{q}_1\cos q_1 + m_3l_2\dot{q}_0\dot{q}_2\cos q_2 \\
& + m_3c_3\dot{q}_0\dot{q}_3\cos q_3 + m_3l_1l_2\dot{q}_1\dot{q}_2\cos(q_2 - q_1) + m_3l_1c_3\dot{q}_1\dot{q}_3\cos(q_3 - q_1) \\
& + m_3l_2c_3\dot{q}_2\dot{q}_3\cos(q_3 - q_2) + \frac{1}{2}J_3\dot{q}_3^2
\end{aligned} \quad (33)$$

La energía cinética total del sistema viene dada por la suma de la energía cinética en cada uno de sus estados, es decir, carro y péndulos 1, 2, 3. Por lo tanto, reemplazando las ecuaciones (22), (27), (30) y (33) en $T = T_0 + T_1 + T_2 + T_3$, resolviendo y simplificando, la energía cinética total del sistema es:

$$\begin{aligned}
T = & \frac{1}{2}\dot{q}_0^2(m_c + m_1 + m_2 + m_3) + \frac{1}{2}\dot{q}_1^2(m_1c_1^2 + m_2l_1^2 + m_3l_1^2) \\
& + \dot{q}_0\dot{q}_1\cos q_1(m_1c_1 + m_2l_1 + m_3l_1) + \frac{1}{2}\dot{q}_2^2(m_2c_2^2 + m_3l_2^2) \\
& + \dot{q}_0\dot{q}_2\cos q_2(m_2c_2 + m_3l_2) + \dot{q}_1\dot{q}_2\cos(q_2 - q_1)(m_2l_1c_2 + m_3l_1l_2) \\
& + \frac{1}{2}m_3c_3^2\dot{q}_3^2 + m_3c_3\dot{q}_0\dot{q}_3\cos q_3 + m_3l_1c_3\dot{q}_1\dot{q}_3\cos(q_3 - q_1) \\
& + m_3l_2c_3\dot{q}_2\dot{q}_3\cos(q_3 - q_2) + \frac{1}{2}J_1\dot{q}_1^2 + \frac{1}{2}J_2\dot{q}_2^2 + \frac{1}{2}J_3\dot{q}_3^2
\end{aligned} \quad (34)$$

Para una simplificación de la ecuación, se realiza un cambio en las constantes:

$$\begin{aligned}
a_0 &= m_c + m_1 + m_2 + m_3 & b_1 &= m_1c_1^2 + m_2l_1^2 + m_3l_1^2 \\
a_1 &= m_1c_1 + m_2l_1 + m_3l_1 & b_2 &= m_2c_2^2 + m_3l_2^2 \\
a_2 &= m_2c_2 + m_3l_2 & b_3 &= m_3c_3^2 \\
a_3 &= m_3c_3 \\
a_4 &= a_2l_1 \\
a_5 &= a_3l_1 \\
a_6 &= a_3l_2
\end{aligned}$$

Realizando el cambio de las constantes en la ecuación (34), resolviendo y simplificando la energía cinética total del sistema es:

$$\begin{aligned}
T = & \frac{1}{2}a_0\dot{q}_0^2 + \frac{1}{2}b_1\dot{q}_1^2 + \frac{1}{2}b_2\dot{q}_2^2 + \frac{1}{2}b_3\dot{q}_3^2 + a_1\dot{q}_0\dot{q}_1\cos q_1 + a_2\dot{q}_0\dot{q}_2\cos q_2 + a_3\dot{q}_0\dot{q}_3\cos q_3 \\
& + a_4\dot{q}_1\dot{q}_2\cos(q_2 - q_1) + a_5\dot{q}_1\dot{q}_3\cos(q_3 - q_1) + a_6\dot{q}_2\dot{q}_3\cos(q_3 - q_2) + \frac{1}{2}J_1\dot{q}_1^2 \\
& + \frac{1}{2}J_2\dot{q}_2^2 + \frac{1}{2}J_3\dot{q}_3^2
\end{aligned} \quad (35)$$

La energía potencial total del sistema viene representada por la suma de las energías potenciales como se muestra en la ecuación (36)

$$V = V_0 + V_1 + V_2 + V_3 \quad (36)$$

El carro se encuentra en la misma altura de la línea de referencia, por lo tanto, la energía potencial del carro es igual a cero. Mientras que la energía potencial del péndulo 1 es:

$$V_1 = gm_1(c_1 \cos q_1) \quad (37)$$

La energía potencial del péndulo 2 es:

$$V_2 = gm_2(l_1 \cos q_1 + c_2 \cos q_2) \quad (38)$$

La energía potencial del péndulo 3 es:

$$V_3 = gm_3(l_1 \cos q_1 + l_2 \cos q_2 + c_3 \cos q_3) \quad (39)$$

Reemplazando las ecuaciones (37), (38), (39) en la ecuación (36), simplificando y reagrupando se obtiene la energía potencial total del sistema dada como:

$$V = (m_1gc_1 + m_2gl_1 + m_3gl_1)\cos q_1 + (m_2gc_2 + m_3gl_2)\cos q_2 + (m_3gc_3)\cos q_3 \quad (40)$$

Para una simplificación de la ecuación, se realiza un cambio en las constantes:

$$k_1 = m_1gc_1 + m_2gl_1 + m_3gl_1$$

$$k_2 = m_2gc_2 + m_3gl_2$$

$$k_3 = m_3gc_3$$

Por lo tanto, realizando el cambio de las constantes, reagrupando y simplificando, la energía potencial total del sistema viene dada por:

$$V = k_1 \cos q_1 + k_2 \cos q_2 + k_3 \cos q_3 \quad (41)$$

Reemplazando las ecuaciones (35) y (41) en (21), se obtiene el lagrangiano del sistema como se presenta en la ecuación (42).

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} a_0 \dot{q}_0^2 + \frac{1}{2} b_1 \dot{q}_1^2 + \frac{1}{2} b_2 \dot{q}_2^2 + \frac{1}{2} b_3 \dot{q}_3^2 + a_1 \dot{q}_0 \dot{q}_1 \cos q_1 + a_2 \dot{q}_0 \dot{q}_2 \cos q_2 + a_3 \dot{q}_0 \dot{q}_3 \cos q_3 \\ & + a_4 \dot{q}_1 \dot{q}_2 \cos(q_2 - q_1) + a_5 \dot{q}_1 \dot{q}_3 \cos(q_3 - q_1) + a_6 \dot{q}_2 \dot{q}_3 \cos(q_3 - q_2) \\ & + \frac{1}{2} J_1 \dot{q}_1^2 + \frac{1}{2} J_2 \dot{q}_2^2 + \frac{1}{2} J_3 \dot{q}_3^2 - k_1 \cos q_1 - k_2 \cos q_2 - k_3 \cos q_3 \end{aligned} \quad (42)$$

La función de disipación Rayleigh's para el sistema TIP, se define como:

$$\mathcal{R} = \frac{1}{2} d_1 \dot{q}_1^2 + \frac{1}{2} d_2 (\dot{q}_2 - \dot{q}_1)^2 + \frac{1}{2} d_3 (\dot{q}_3 - \dot{q}_2)^2 \quad (43)$$

Para el carro, la ecuación de movimiento se representa mediante la ecuación (44), donde τ es la fuerza que se le aplica al carro para estabilizar el sistema.

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_0} \right) - \frac{\partial \mathcal{L}}{\partial q_0} + \frac{\partial \mathcal{R}}{\partial \dot{q}_0} = \tau \quad (44)$$

Reemplazando el lagrangiano y la función de disipación Rayleigh's en la ecuación (44), derivando respecto a cada variable y resolviendo, se obtiene que la ecuación de movimiento del carro es:

$$a_0 \ddot{q}_0 + a_1 (\ddot{q}_1 \cos q_1 - \dot{q}_1^2 \sin q_1) + a_2 (\ddot{q}_2 \cos q_2 - \dot{q}_2^2 \sin q_2) + a_3 (\ddot{q}_3 \cos q_3 - \dot{q}_3^2 \sin q_3) = \tau \quad (45)$$

Para el primer péndulo, la ecuación de movimiento se representa mediante:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) - \frac{\partial \mathcal{L}}{\partial q_1} + \frac{\partial \mathcal{R}}{\partial \dot{q}_1} = 0 \quad (46)$$

Reemplazando el lagrangiano y la función de disipación Rayleigh's en la ecuación (46), derivando respecto a cada variable y resolviendo, se obtiene que la ecuación de movimiento del primer péndulo es:

$$\begin{aligned} & a_1(\ddot{q}_0 \cos q_1 - g \operatorname{sen} q_1) + a_4(\ddot{q}_2 \cos(q_2 - q_1) - \dot{q}_2^2 \operatorname{sen}(q_2 - q_1)) \\ & + a_5(\ddot{q}_3 \cos(q_3 - q_1) - \dot{q}_3^2 \operatorname{sen}(q_3 - q_1)) + b_1 \ddot{q}_1 + J_1 \dot{q}_1 + d_1 \dot{q}_1 \\ & - d_2(\dot{q}_2 - \dot{q}_1) = 0 \end{aligned} \quad (47)$$

Para el segundo péndulo, la ecuación de movimiento se representa mediante:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) - \frac{\partial \mathcal{L}}{\partial q_2} + \frac{\partial \mathcal{R}}{\partial \dot{q}_2} = 0 \quad (48)$$

Reemplazando el lagrangiano y la función de disipación Rayleigh's en la ecuación (48), derivando respecto a cada variable y resolviendo, se obtiene que la ecuación de movimiento del segundo péndulo es:

$$\begin{aligned} & a_2(\ddot{q}_0 \cos q_2 - g \operatorname{sen} q_2) + a_4(\ddot{q}_1 \cos(q_2 - q_1) + \dot{q}_1^2 \operatorname{sen}(q_2 - q_1)) + a_6(\ddot{q}_3 \cos(q_3 - q_2) \\ & - \dot{q}_3^2 \operatorname{sen}(q_3 - q_2)) + b_2 \ddot{q}_2 + J_2 \dot{q}_2 + d_2(\dot{q}_2 - \dot{q}_1) - d_3(\dot{q}_3 - \dot{q}_2) = 0 \end{aligned} \quad (49)$$

Para el tercer péndulo, la ecuación de movimiento se representa mediante:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_3} \right) - \frac{\partial \mathcal{L}}{\partial q_3} + \frac{\partial \mathcal{R}}{\partial \dot{q}_3} = 0 \quad (50)$$

De la misma manera reemplazando el lagrangiano y la función de disipación Rayleigh´s en la ecuación (50), derivando respecto a cada variable y resolviendo, se tiene que la ecuación de movimiento del tercer péndulo es:

$$\begin{aligned} & a_3(\ddot{q}_0 \cos q_3 - g \sin q_3) + a_5(\ddot{q}_1 \cos(q_3 - q_1) + \dot{q}_1^2 \sin(q_3 - q_1)) \\ & + a_6(\ddot{q}_2 \cos(q_3 - q_2) + \dot{q}_2^2 \sin(q_3 - q_2)) + b_3 \ddot{q}_3 + J_3 \ddot{q}_3 \\ & + d_3(\dot{q}_3 - \dot{q}_2) = 0 \end{aligned} \quad (51)$$

Las ecuaciones de movimiento expresadas en notación matricial pueden representarse como:

$$M(q)\ddot{q} + C_o(q, \dot{q})\dot{q} + g(q) = \tau \quad (52)$$

En donde:

M= Matriz de masa.

C_o= Matriz de Coriolis y amortiguamiento

g= Matriz de gravedad.

Por lo tanto, agrupando las ecuaciones (47), (49), (51) y rescribiendo en la forma matricial dada por la ecuación (52), se obtiene la representación no lineal del sistema triple péndulo invertido dado como:

$$M(q) = \begin{bmatrix} a_0 & a_1 \cos q_1 & a_2 \cos q_2 & a_3 \cos q_3 \\ a_1 \cos q_1 & b_1 + J_1 & a_4 \cos(q_2 - q_1) & a_5 \cos(q_3 - q_1) \\ a_2 \cos q_2 & a_4 \cos(q_2 - q_1) & b_2 + J_2 & a_6 \cos(q_3 - q_2) \\ a_3 \cos q_3 & a_5 \cos(q_3 - q_1) & a_6 \cos(q_3 - q_2) & b_3 + J_3 \end{bmatrix}$$

$$C_o(q, \dot{q}) = \begin{bmatrix} d_0 & -a_1 \dot{q}_1 \sin q_1 & -a_2 \dot{q}_2 \sin q_2 & -a_3 \dot{q}_3 \sin q_3 \\ 0 & d_1 + d_2 & -d_2 - a_4 \dot{q}_2 \sin(q_2 - q_1) & -a_5 \dot{q}_3 \sin(q_3 - q_1) \\ 0 & -d_2 + a_4 \dot{q}_1 \sin(q_2 - q_1) & d_2 + d_3 & -d_3 - a_6 \dot{q}_3 \sin(q_3 - q_2) \\ 0 & a_5 \dot{q}_1 \sin(q_3 - q_1) & -d_3 + a_6 \dot{q}_2 \sin(q_3 - q_2) & d_3 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} 0 \\ -k_1 \sin q_1 \\ -k_2 \sin q_2 \\ -k_3 \sin q_3 \end{bmatrix} \quad \tau = \begin{bmatrix} u \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Para obtener la representación del sistema en el espacio de estados, es necesario despejar la variable de mayor grado en la ecuación (52), es decir:

$$\ddot{q} = M^{-1}[\tau - C_o(q, \dot{q})\dot{q} - g(q)] \quad (53)$$

En donde, las variables de estado son definidas como:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

La figura 3.2, presenta el diagrama de bloques implementado en Simulink para obtener la respuesta no lineal en cada uno de los péndulos del sistema.

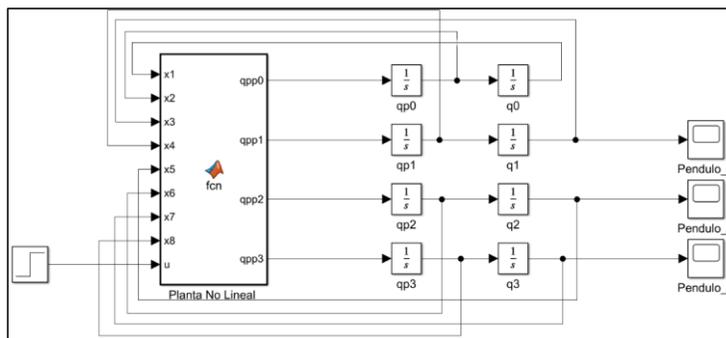


Figura 3. 2 Diagrama de bloques de la planta no lineal

Fuente: Autor

En la figura 3.3. se muestra la respuesta del sistema no lineal cuando éste en estado inicial se encuentra en su posición vertical superior y se deja caer libremente, es decir, no se encuentra bajo ninguna acción de control.

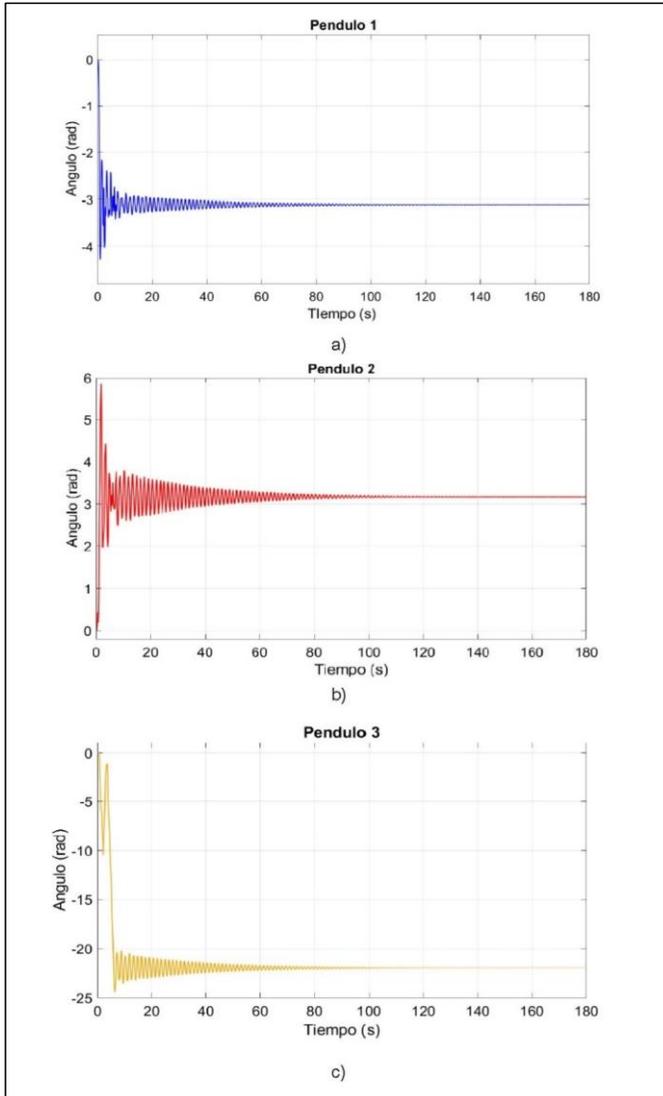


Figura 3. 3. Respuesta del sistema no lineal sin acción de control: a) Péndulo 1, b) Péndulo 2, c) Péndulo 3

Fuente: Autor

3.2. Linealización y representación en el espacio de estados.

Para el desarrollo de las estrategias de control propuestas, el sistema del triple péndulo invertido es linealizado alrededor de los puntos de equilibrio del sistema dados como: $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, u] = [0, 0, 0, 0, 0, 0, 0, 0, 0]$, con el objetivo de facilitar su aplicación. Este proceso conduce al sistema de espacio de estado presentado en la ecuación (1) y (2).

Donde A, B y C representan las matrices de estado, de entrada y de salida del sistema respectivamente, $x(t)$ es el vector de estados, $u(t)$ es el vector de entradas y $y(t)$ es el vector de salida. Las matrices A y B están formadas de la siguiente manera:

$$A = \begin{bmatrix} 0_{4 \times 4} & I_{4 \times 4} \\ -D(0)^{-1} * G & -D(0)^{-1} * C(0,0) \end{bmatrix}_{8 \times 8} \quad B = \begin{bmatrix} 0_{4 \times 1} \\ D(0)^{-1} * H \end{bmatrix}_{8 \times 1} \quad (54)$$

Realizando la linealización del modelo no lineal y representando en espacio de estados, se obtienen las siguientes matrices:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 18.2787 & -0.9534 & 0.0269 & 0 & 0.6039 & -0.0075 & 0.0005 \\ 0 & -144.5732 & 54.2372 & -1.5302 & 0 & -4.7965 & 0.1561 & -0.0288 \\ 0 & 75.0684 & -80.3780 & 11.9416 & 0 & 2.5125 & -0.2048 & 0.0806 \\ 0 & -8.2026 & 46.2486 & -38.7707 & 0 & -0.2903 & 0.1592 & -0.1713 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.8223 \\ -2.7813 \\ 0.2008 \\ -0.0219 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

En la figura 3.4. se muestra el diagrama de bloques implementado en simulink para obtener la respuesta del sistema linealizado en donde se utiliza el bloque de espacios de estados y se ingresan las matrices A, B, C y D.

La respuesta del sistema linealizado se presenta en la figura 3.5, en donde se observa una pequeña variación con respecto al sistema no lineal, debido a la simplificación realizada alrededor de los puntos de equilibrio. A pesar de estos cambios el sistema lineal sigue ofreciendo una buena representación del sistema real.

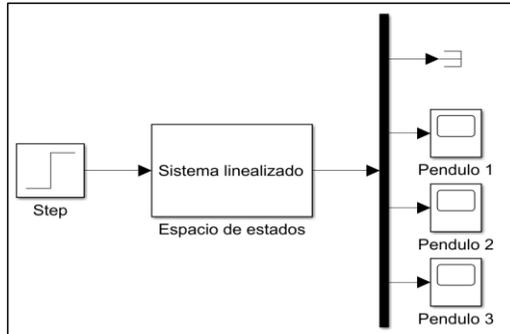


Figura 3. 4. Sistema linealizado

Fuente: Autor

3.3. Diseño del controlador LQR

Obtenidas las matrices de estado, de entrada y salida del sistema, se diseña el controlador LQR bajo un esquema en lazo cerrado, donde el sistema se encuentra en una posición inicial de 0.35 rad en el tercer péndulo y el objetivo es estabilizar el sistema. En primer lugar se definen las matrices de ponderación para el controlador. En este caso, la matriz Q y R son definidas como:

$$Q = \text{diag}[10, 100, 100, 100] \quad R = 1 \quad (55)$$

Para encontrar las ganancias del controlador LQR en Matlab se utilizan comandos predeterminados para el diseño de este. Al utilizar la ecuación algebraica de Riccati, las líneas de códigos establecidas en Matlab para el diseño del controlador LQR, se presentan como:

$$P = \text{care}(A, B, Q, R) \quad (56)$$

$$K = \text{inv}(R) * B' * P \quad (57)$$

Mientras que una forma más directa, puede ser empleando el comando lqr en Matlab, dado como:

$$K = \text{lqr}(A, B, Q, R) \quad (58)$$

En cualquier caso, las ganancias generadas para el controlador LQR deben ser las mismas. La figura 3.6. presenta el esquema desarrollado en Simulink en donde el controlador LQR en lazo cerrado es implementado.

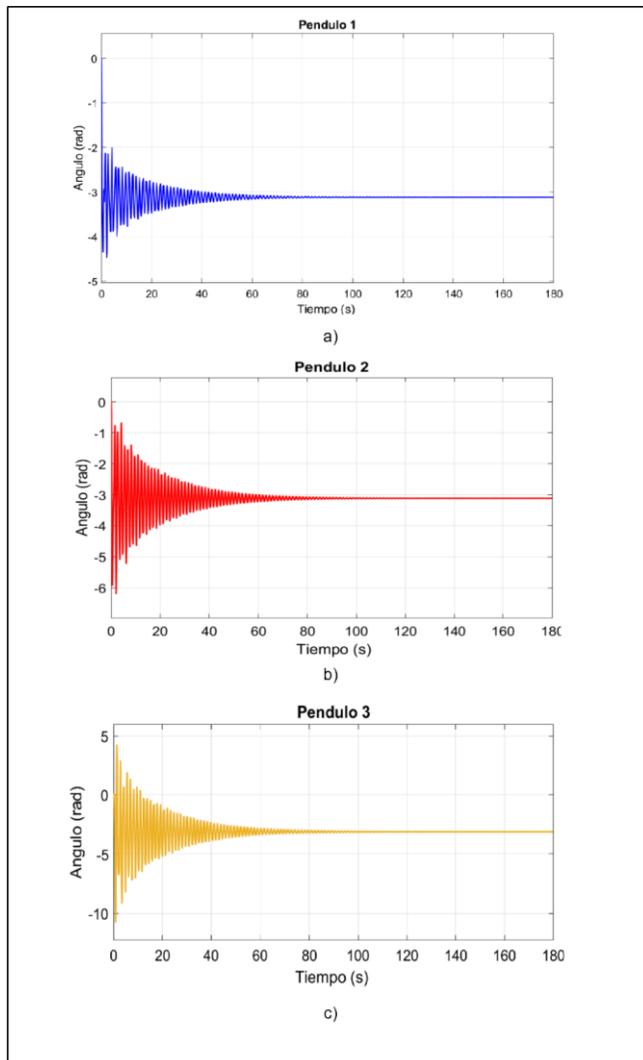


Figura 3. 5 Respuesta del sistema linealizado sin acción de control: a) Péndulo 1, b) Péndulo 2, c) Péndulo 3

Fuente: Autor

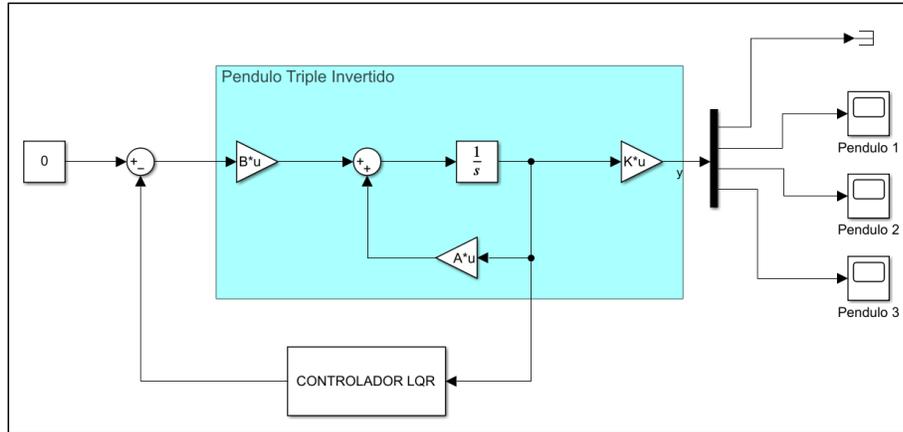


Figura 3. 6 Diagrama de bloques del controlador LQR

Fuente: Autor

Al implementar el controlador LQR con los parámetros establecidos anteriormente en el sistema TIP, se observa un control relativamente lento donde la estabilización es alcanzada entorno a los 5 segundos como se muestra en la figura 3.7.

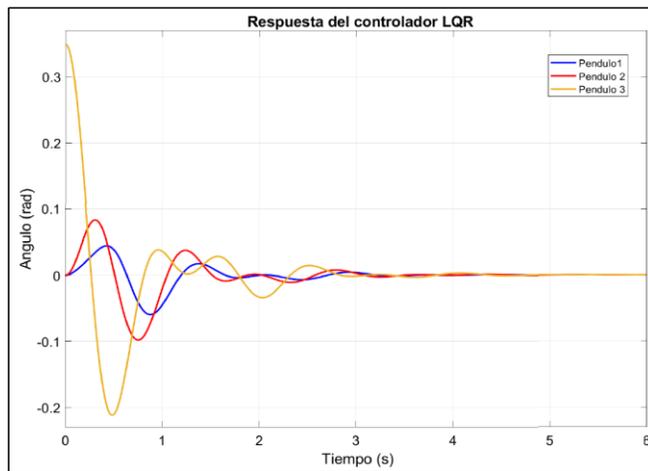


Figura 3. 7 Respuesta del controlador LQR en el sistema TIP

Fuente: Autor

La señal de control que emite la estrategia LQR para estabilizar el sistema TIP es presentada en la figura 3.8, en donde se observa que la fuerza máxima que emite es de 1,224 Newton y acorde la estabilización se va logrando, la fuerza va disminuyendo hasta llegar prácticamente a cero.

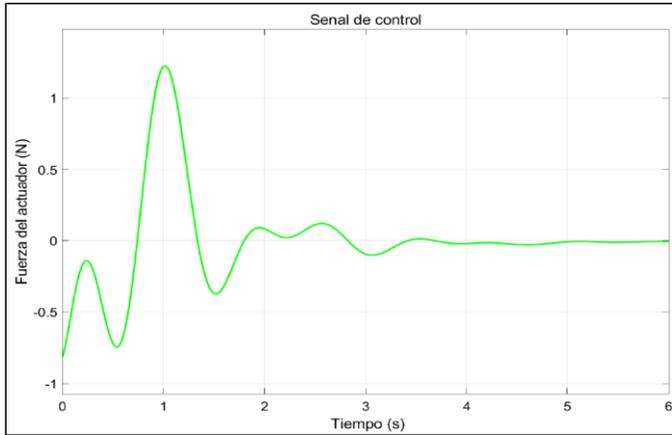


Figura 3. 8 Señal de control de la estrategia LQR

Fuente: Autor

3.4. Diseño del sintonizador de ganancias mediante el algoritmo PSO

Para el diseño del sintonizador de ganancias mediante el algoritmo PSO, se emplea un esquema con una señal de referencia dada como cero y una posición inicial del tercer péndulo de 0.35 rad, como en el caso anterior.

El esquema empleado se muestra en la figura 3.9 en donde el propósito es encontrar las ganancias adecuadas para estabilizar el sistema en el menor tiempo posible mediante la minimización de la función objetivo presentada en la ecuación (59) donde únicamente intervienen los errores de las posiciones en cada estado del sistema.

$$F_1 = \int [(e_c)^2 + (e_1)^2 + (e_2)^2 + (e_3)^2] \quad (59)$$

Donde e_c representa el error en la posición del carro con respecto a la referencia dada, y e_1, e_2, e_3 representan los errores en la posición del péndulo 1, 2 y 3 con respecto a la referencia dada.

Dado que el controlador LQR proporciona una cantidad de 8 ganancias (una para cada estado, con su respectiva derivada), la estrategia presentada en esta sección posee el mismo número de ganancias, con la particularidad que su sintonización se lo realiza por medio del algoritmo de optimización de enjambre de partícula (PSO). Por lo tanto, la dimensión del problema es de 8. En la tabla 2 se presentan los parámetros iniciales que se utiliza para la ejecución del algoritmo.

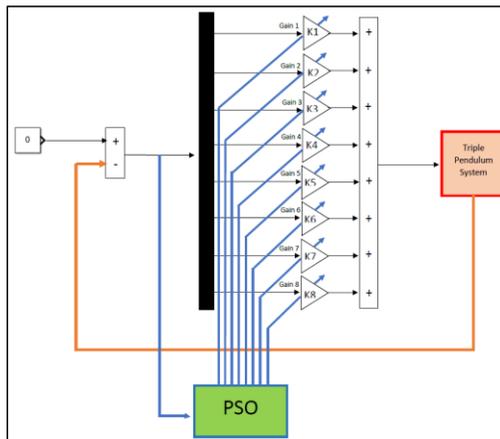


Figura 3. 9 Esquema del sintonizador de ganancias PSO

Fuente: Autor

Tabla 2.
Parámetros iniciales del algoritmo PSO (Caso1)

Size = 100	Tamaño del enjambre
Max. Iter = 1000	Número máximo de iteraciones
Dim = 8	Dimensión del problema
$C_2 = 2.4$	Aceleración cognitiva
$C_1 = 2.4$	Aceleración social
$W = 0.8$	Momento o inercia "peso de la velocidad"

Fuente: Autor

La posición y la velocidad inicial del enjambre se establecen aleatoriamente en función de la dimensión y tamaño del enjambre, y su forma de obtención dentro del algoritmo es presentado en la tabla 3.

Tabla 3.
Velocidad y posición inicial del enjambre

Posición Inicial del enjambre	$P_{\text{current}} = \text{diag}(1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1) * \text{rand}(\text{dim}, \text{Size})$
Velocidad inicial del enjambre	$V = 5 * \text{rand}(\text{dim}, \text{Size})$

Fuente: Autor

La actualización de estas velocidades y posiciones del enjambre se lo realiza utilizando las ecuaciones (17) y (18) hasta cumplir con el número de iteraciones establecidas satisfaciendo la función objetivo implementada.

El diagrama de bloques realizado en simulink para la sintonización de las ganancias es presentado en la figura 3.10.

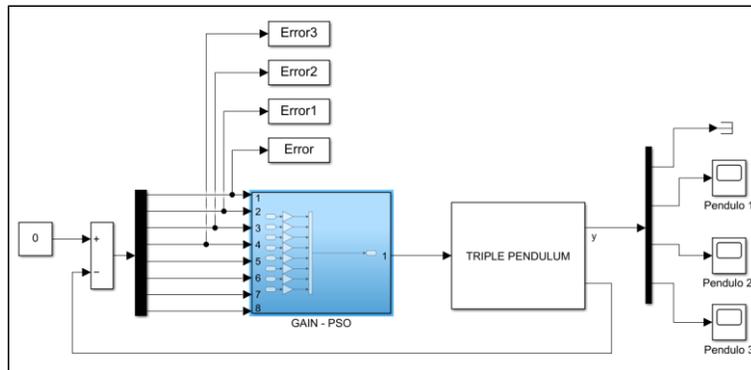


Figura 3. 10 Diagrama de boques del sintonizador de ganancias PSO

Fuente: Autor

En la figura 3.11 se muestra la respuesta de dicha estrategia, donde se obtiene una mínima mejora en cuanto a la estabilización en comparación al caso anterior.

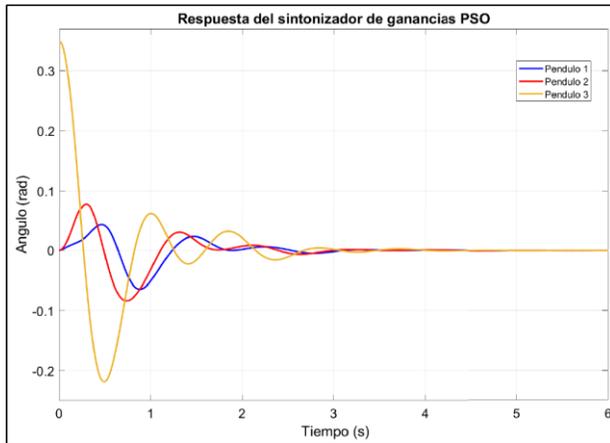


Figura 3. 11 Respuesta del sintonizador de ganancias PSO

Fuente: Autor

La señal de control que emite el sintonizador de ganancias PSO es presentada en la figura 3.12, en donde se observa que la fuerza máxima es de 1,372 Newton y como en el caso anterior acorde la estabilización se va logrando, la fuerza va disminuyendo hasta llegar prácticamente a cero.

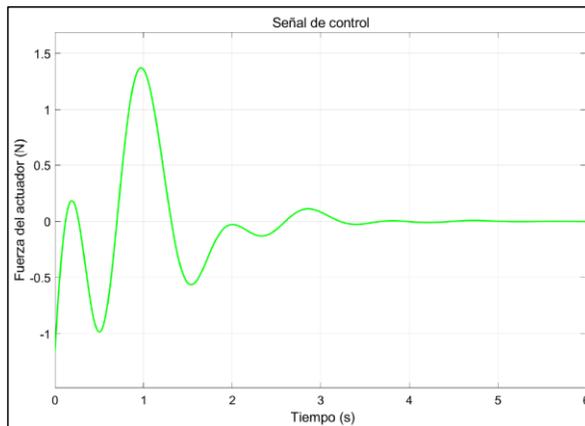


Figura 3. 12 Señal de control del sintonizador de ganancias PSO

Fuente: Autor

3.5. Diseño del controlador difuso sintonizado mediante el algoritmo PSO

En esta sección se presenta el diseño de un controlador difuso optimizado por el algoritmo PSO, el cual se describe en la figura 3.13. En donde, en este caso el objetivo del algoritmo de optimización es sintonizar los factores de escala de un controlador difuso para estabilizar el sistema en el mínimo tiempo posible.

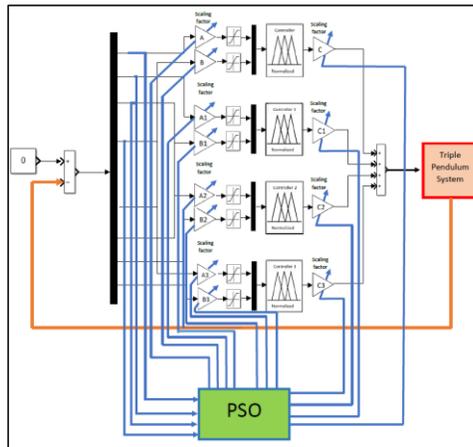


Figura 3. 13 Esquema del controlador propuesto Fuzzy-PSO

Fuente: Autor

Para implementar la estrategia de control presentada es necesario primero desarrollar el diagrama de bloques en simulink, utilizando el sistema TIP implementado anteriormente. Luego los pasos por seguir son los siguientes:

- Diseño del controlador difuso
- Desarrollo del algoritmo de optimización

3.6. Diseño del controlador difuso

La estrategia de control utiliza 4 controladores difusos (uno para cada estado), en donde estos son iguales entre sí. Por lo tanto, esta sección se centra en la presentación del diseño de un solo controlador difuso. Para el diseño del controlador se utilizó la caja de herramientas de Matlab "fuzzy logic designer" como se presenta en la figura 3.14, en donde el tipo de controlador utilizado es Mandani con defuzzificación por el método del centroide basándose en dos entradas: el error "e" y el cambio en el error "er" del estado a controlar (la posición del carro, o la posición en cada uno de los péndulos).

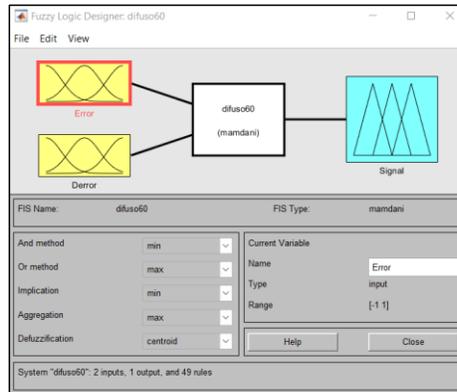


Figura 3. 14 Caja de herramientas de Matlab: “Fuzzy Logic Designer”.

Fuente: Matlab 2021

Para el desarrollo de la base de reglas difusas del controlador, fue necesario realizar una investigación exhaustiva de distintos artículos, en donde la mayoría de los controladores que utilizaban el error y el cambio de error como entradas del controlador, para brindar una salida, empleaban reglas difusas del tipo "si X e Y entonces Z". Por lo tanto, la base de reglas utilizada en este trabajo es basada en investigaciones previas y son presentadas en la tabla 4 con un número total de 49 reglas difusas (Chen & Theodomile, 2016; Hurel et al., 2012).

Tabla 4.
Reglas del controlador difuso

		Error						
		NB	NM	NS	ZE	PS	PM	PB
Tasa del error	NB	NB	NB	NB	NM	NM	NS	ZE
	NM	NB	NB	NM	NM	NS	ZE	PS
	NS	NB	NM	NM	NS	ZE	PS	PM
	ZE	NM	NM	NS	ZE	PS	PM	PM
	PS	NM	NS	ZE	PS	PM	PM	PB
	PM	NS	ZE	PS	PM	PB	PB	PB
	PB	ZE	PS	PM	PB	PB	PB	PB

Fuente: Chen & Theodomile, 2016; Hurel et al., 2012.

El significado de los términos lingüísticos es presentado en la tabla 5. El controlador difuso propuesto presenta cinco funciones de pertenencia triangulares y dos trapezoidales para cada variable normalizada es decir cuyo universo de discurso se encuentra dentro del rango $[-1,1]$, como se muestra en la figura 3.15. Así, los valores reales del controlador se convierten utilizando los factores de escala A, B y C en cada uno.

Tabla 5.
Variables lingüísticas

VARIABLES LINGÜÍSTICAS	SIGNIFICADO
NB	Negativo grande
NM	Negativo medio
NS	Negativo pequeño
ZE	Error cero
PS	Positivo pequeño
PM	Positivo medio
PB	Positivo grande

Fuente: Chen & Theodomile, 2016; Hurel et al., 2012.

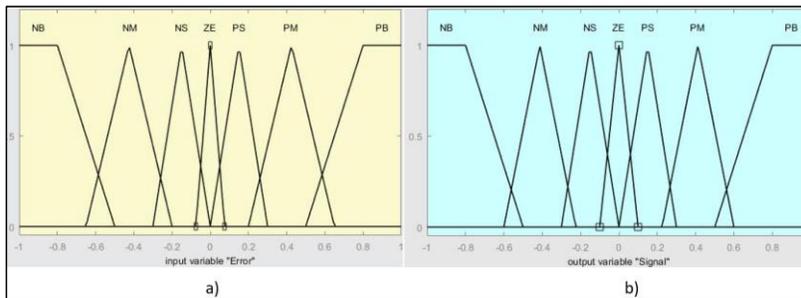


Figura 3. 15 Funciones de pertenencia para variables normalizadas: (a) variable de entrada “Error”; (b) variable de salida “Señal”

Fuente: Autor

En la figura 3.16 se visualizan las 49 reglas difusas ingresadas por el usuario que ofrecen la acción de control que permite estabilizar el sistema. La caja de herramientas de Matlab también permite visualizar de manera gráfica el resultado de la defuzzificación, mientras

que de manera numérica se lo visualiza en la parte superior de la columna como se muestra en la figura 3.17.

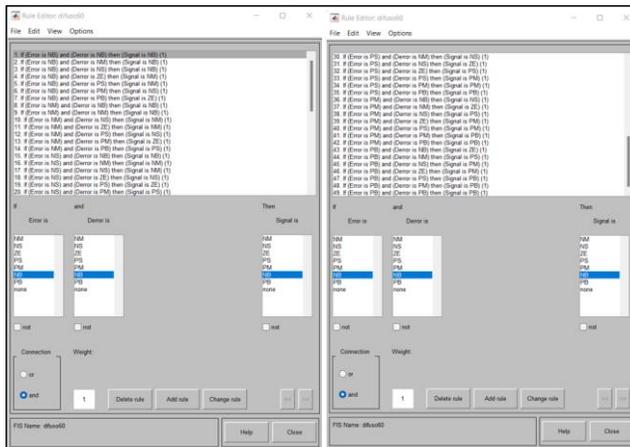


Figura 3. 16 Visualización de las reglas difusas dentro de la herramienta de Matlab”.

Fuente: Autor

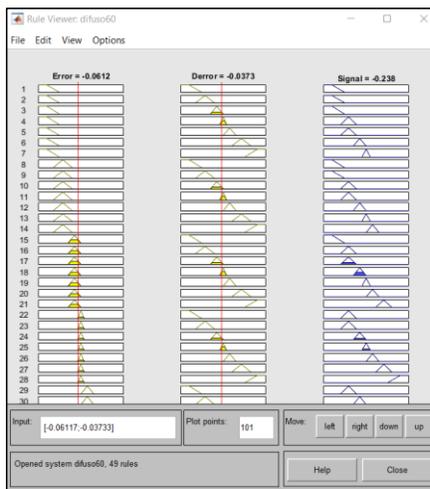


Figura 3. 17 Visualización gráfica de las reglas difusas.

Fuente: Autor

3.7. Desarrollo del algoritmo de optimización

Es importante conocer primero el problema a optimizar para comenzar con el desarrollo del algoritmo. En este caso, el objetivo es sintonizar los factores de escala (A, B, C) de los 4 controladores (uno para cada estado) como se muestra en la figura 3.13. Luego se establece la función objetivo a minimizar, en este caso la función establecida es:

$$F_2 = \int [300(ec)^2 + 320(e_1)^2 + 360(e_2)^2 + 400(e_3)^2 + 0.3(\max(ec)) + 0.45(\max(e_1)) + 0.6(\max(e_2)) + 0.8(\max(e_3))] \quad (60)$$

En donde se establecen pesos en cada variable escogidos por el diseñador. En esta función objetivo intervienen los errores en cada estado, y el sobre - impulso (max). En la tabla 6 se muestran los parámetros iniciales establecidos para el algoritmo PSO.

Tabla 6.
Parámetros iniciales del algoritmo PSO (Caso 2)

Tamaño = 24	Tamaño del enjambre
Max. Iter = 100	Número máximo de iteraciones
Dim = 12	Dimensión del problema
C ₂ = 2.4	Aceleración cognitiva
C ₁ = 2.4	Aceleración social
W = 0.9	Momento o inercia "peso de la velocidad"

Fuente: Autor

Se crean las variables r1 y r2 (números aleatorios) y se establecen las velocidades y posiciones iniciales del enjambre como se muestra en la tabla 7.

Luego se evalúa la mejor partícula del enjambre inicial, de modo que se crea la variable "fitness" como una matriz de ceros de una columna y el número de filas correspondiente al tamaño del enjambre. Para luego crear un bucle "for" e ir guardando el resultado de la mejor partícula evaluada mediante la función objetivo (Fval70) como se muestra en la figura 3.18.

Comentado [UdW4]: Tabla en español

Tabla 7.
Velocidad y posición inicial del enjambre

Posición Inicial del enjambre	$P_{current} = \text{diag}([1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]) * \text{rand}(\text{dim}, \text{Size})$
Velocidad inicial del enjambre	$V = 5 * \text{rand}(\text{dim}, \text{Size})$

Fuente: Autor

```

Evaluar la aptitud del enjambre inicial
fitness = zeros(1, size)
for i = 1:Size
    fitness(i) = Fval70(Pcurrent(:, i))
end

inicialmente, LBfitness es igual a la posición
LBfitness = fitness

Determinar la aptitud y el índice de la mejor partícula.
[bestFitness, bestIndex] = min (LBfitness)

Las evaluaciones de aptitud anteriores cuentan como iteraciones
iter = Size
  
```

Figura 3. 18 Evaluación de la partícula fitness en el enjambre inicial

Fuente: Autor

Inicialmente se define a la variable “LBfitness” igual que la posición luego se determina la mejor partícula y el índice de la mejor partícula. Las evaluaciones de aptitud anteriores cuentan como iteraciones por tanto “iter = Size”.

El núcleo del algoritmo, donde se realizan las actualizaciones de las posiciones y velocidades de las partículas mediante la implementación de las ecuaciones (17) y (18) se presenta en la figura 3.19 en donde mediante un bucle while se establecen ciertas condiciones para la búsqueda de los mejores resultados en el enjambre.

Por último, es necesario crear las variables donde se guardan los mejores resultados para ser cargados en el modelo simulink presentado en la figura 3.20. Siendo aquello el último paso para la obtención de los resultados para la estabilización del sistema TIP.

```

while iter < max.iter

    % Elección de índice de una partícula aleatoria del enjambre
    i = ceil(Size * rand())

    % Nueva velocidad de la partícula
    V(:,i) = w * (V(:,i)) + c1 * R1 * (Pbest_local(:,i) - Pcurrent(:,i)) + c2 * R2 *
    (Pbest_local(:,bestIndex) - Pcurrent(:,i))

    % Nueva posición de la partícula
    Pcurrent(:,i) = Pcurrent(:,i) + V(:,i)

    if Pcurrent(12,i) < 100

        %Evaluar la aptitud de un nuevo enjambre
        newFitness = Feval70(Pcurrent(:,i))

        %Actualizar las posiciones y la aptitud de las partículas
        if newFitness < fitness(i)
            fitness(i) = newFitness

            Actualizar la mejor posición de las partículas
            Pbest_local(:,i) = Pcurrent(:,i)

            Actualizar la mejor posición del enjambre
            if newFitness < bestFitness

                % Índice en la matriz Pbest_local
                bestIndex = i
                bestFitness = newFitness

            end

        end

        iter = iter + 1 % Incremento del contador
    end

end

```

Figura 3. 19 Núcleo del algoritmo

Fuente: Autor

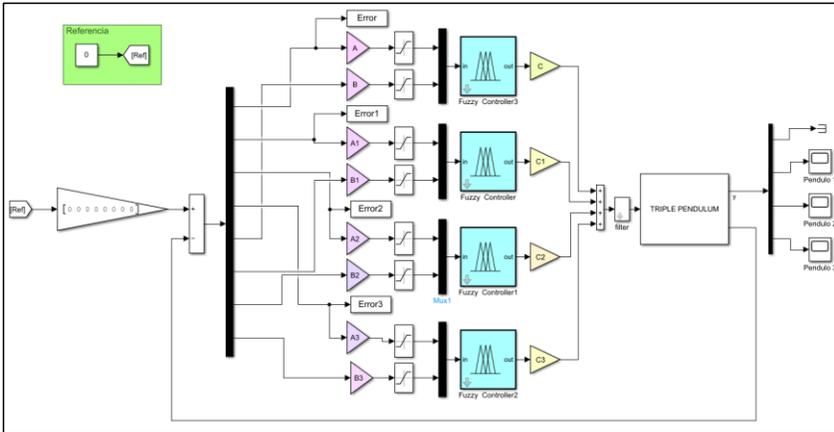


Figura 3. 20 Diagrama de bloques en simulink de la optimización Fuzzy - PSO

Fuente: Autor

La respuesta del controlador Fuzzy-PSO se muestra en la figura 3.21, donde se observa una mayor rapidez en la estabilización del sistema en comparación a las estrategias presentadas anteriormente.

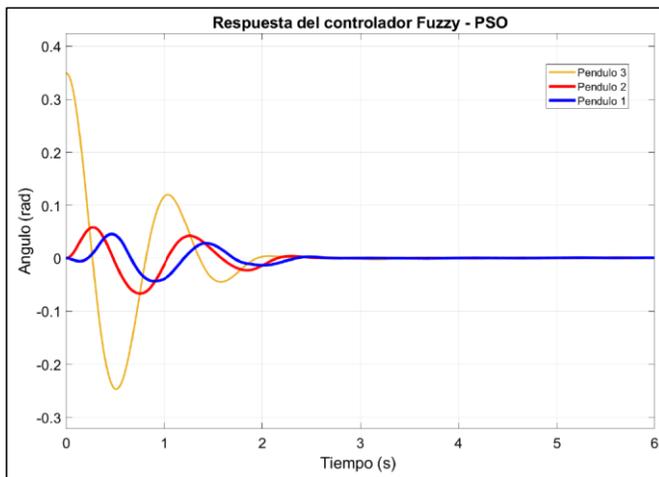


Figura 3. 21 Respuesta del controlador Fuzzy – PSO.

Fuente: Autor

CAPÍTULO 4

4. Resultados

En este capítulo se muestran los resultados obtenidos mediante los tres esquemas de control (LQR, sintonizador de ganancias Gain-PSO y Fuzzy-PSO) desarrollados en el entorno de Matlab/Simulink y se realiza un análisis comparativo con el fin de indicar cuál de ellos presenta un mejor desempeño según la respuesta del sistema y el tiempo de estabilización.

En la figura 4.1 se presenta el modelo Simulink empleado para realizar el análisis comparativo de los controladores descritos anteriormente. Estos modelos de control se encuentran bajo la misma señal de referencia dada como cero para obtener su equilibrio en la posición vertical superior y el sistema en su posición inicial dada como: péndulo 1 = 0 rad, péndulo 2 = 0 rad y péndulo 3 = 0.35 rad.

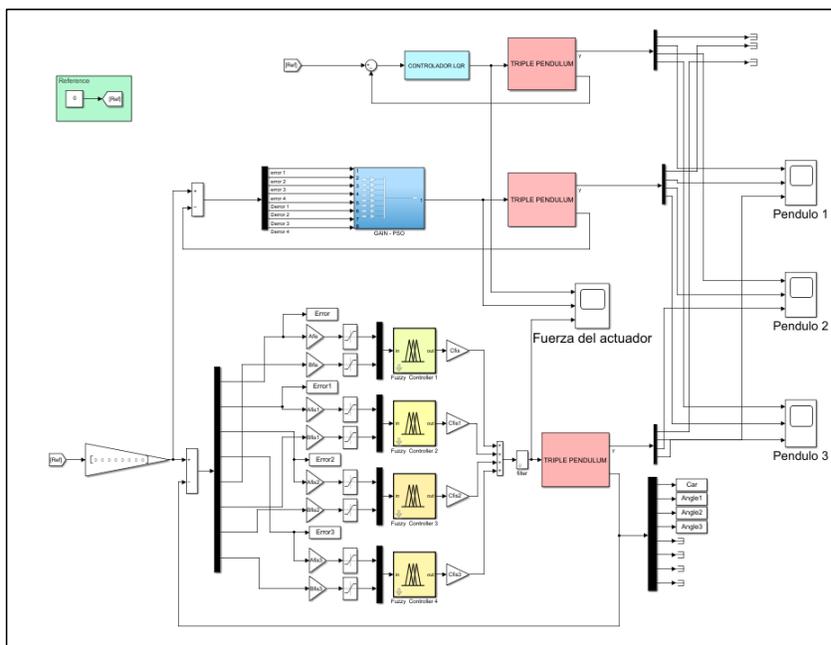


Figura 4. 1 Diagrama de bloques en simulink para el análisis comparativo

Fuente: Autor

Los resultados de las ganancias generadas por el controlador LQR (primer caso), la sintonización Gain-PSO (segundo caso) y el controlador Fuzzy-PSO (tercer caso) para estabilizar el sistema se presentan en la Tabla 8.

Tabla 8.
Resultados obtenidos en las ganancias

LQR (1 ^{er} caso)		Gain – PSO (2 ^{do} caso)		Fuzzy – PSO (3 ^{er} caso)	
Ganancias	Resultado	Ganancias	Resultado	Ganancias	Resultado
	3.1623	K1	6.6845	A	-0.7192
	-7.4107	K2	1.7739	B	-0.5438
	-4.5701	K3	-0.6779	C	-0.7040
K	2.3414	K4	3.3407	A1	-0.1772
	5.2449	K5	6.6972	B1	1.4340
	-0.6742	K6	-0.9444	C1	-0.9123
	-1.8765	K7	-2.0023	A2	-0.4516
	-0.7389	K8	-0.1778	B2	1.2779
				C2	-1.0911
				A3	0.8128
				B3	0.2294
				C3	-1.3621

Fuente: Autor

El desarrollo y ejecución del algoritmo PSO se implementó en Matlab en una CPU Intel(R) Core (TM) i7-10750H a 2,60 GHz, con un tiempo de simulación de 1857,23 segundos para el segundo caso y 7396 segundos para el tercer caso.

La figura 4.2 muestra la respuesta angular en el sistema bajo las tres acciones de control para la estabilización del sistema. Se observa que el tiempo para estabilizar los tres péndulos utilizando el controlador LQR es de 4,6 segundos, mientras que el tiempo de estabilización utilizando el esquema de control Gain - PSO es de 3,5 segundos. Sin embargo, empleando el controlador Fuzzy-PSO propuesto, el tiempo de estabilización se reduce a 2,85s.

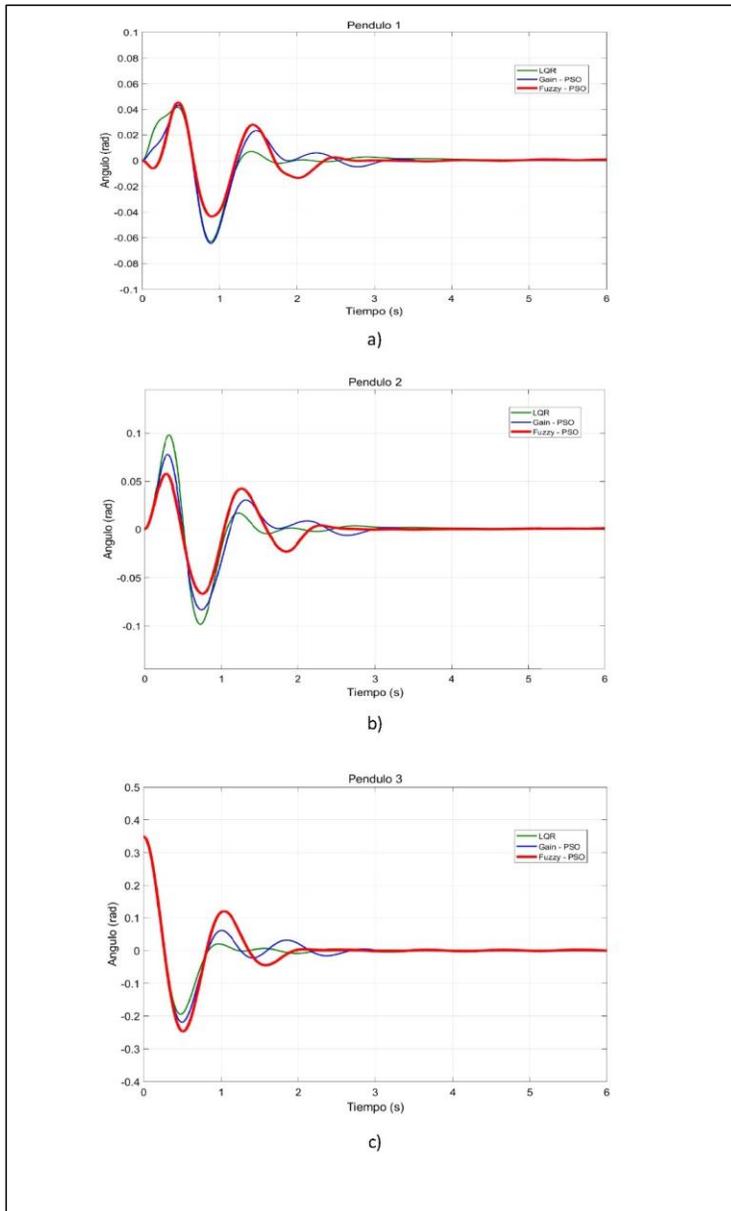


Figura 4. 2 Comparación de las estrategias de control empleadas para la estabilización del sistema TIP: a) Péndulo 1, b) Péndulo 2, c) Péndulo 3.

Fuente: Autor

Por tanto, el controlador Fuzzy-PSO propuesto resulta ser un 39,13% más eficiente que el controlador LQR y un 20% más eficiente que el esquema Gain-PSO de acuerdo con el tiempo de estabilización obtenido.

La figura 4.2 (a) muestra la respuesta angular en el primer péndulo, donde se observa que el controlador Fuzzy - PSO logra reducir el sobre impulso en comparación con las otras estrategias de control.

En la figura 4.2 (b) correspondiente al segundo péndulo, se observa de igual manera una reducción del sobre impulso utilizando la estrategia Fuzzy - PSO en comparación con el controlador LQR y Gain - PSO.

Mientras que en la figura 4.2 (c) correspondiente al tercer péndulo donde la posición inicial es de 0.35 rad el sobre impulso generado es ligeramente superior al de las otras estrategias de control sin embargo la estabilización es más rápida.

Por su parte la señal de control generada por la estrategia Fuzzy – PSO presenta sobre impulsos menores como se muestra en la figura 4.3, lo que conduce a un menor gasto de energía en el actuador.

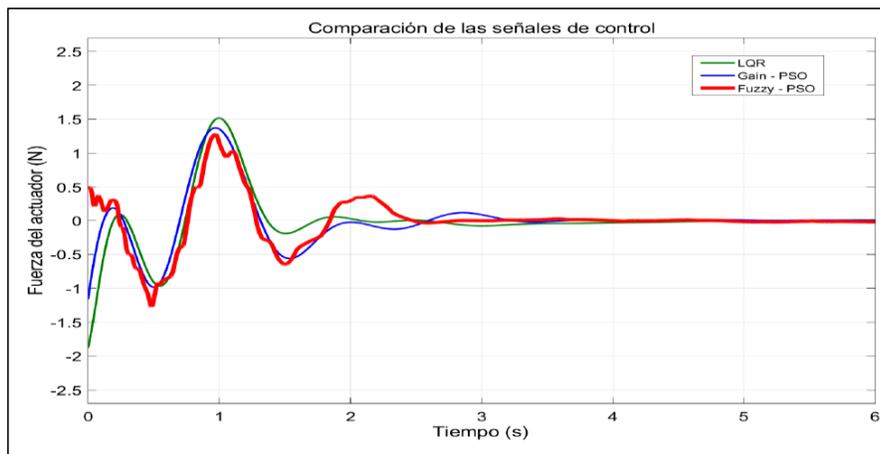


Figura 4. 3 Comparación de las señales de control

Fuente: Autor

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

1. Se logró modelar un sistema péndulo triple invertido a partir de las ecuaciones de movimiento del sistema linealizado dentro del entorno Matlab/Simulink.
2. Se desarrollaron tres estrategias de control: regulador lineal cuadrático (LQR), sintonizador de ganancias PSO (Gain – PSO), y optimización del controlador difuso (Fuzzy – PSO), y se implementaron con éxito para lograr la estabilización de sistema TIP.
3. El algoritmo PSO se implementó para disminuir los errores de estado en el segundo caso y para ajustar los factores de escala del controlador difuso en el tercer caso bajo una señal de referencia de cero y una posición inicial del sistema de 0,35 radianes en el tercer péndulo.
4. Mediante el análisis comparativo realizado se obtuvo que el controlador Fuzzy – PSO propuesto es un 39.13% y un 20% más eficiente en reducir el tiempo de estabilización del sistema en comparación a la estrategia LQR y la sintonización de ganancias Gain – PSO respectivamente, en donde el tiempo de estabilización de cada estrategia fue la siguiente: LQR = 4.6 segundos, Gain – PSO = 3.5 segundos y Fuzzy – PSO = 2.85 segundos
5. De este modo, se logró obtener una estrategia de control mucho más eficiente y así ampliar las investigaciones en el área de controladores difusos para el sistema TIP.

5.2.Recomendaciones

1. Generar alteraciones en el algoritmo de optimización de enjambre de partículas como en el factor social, cognitivo o en la inercia podría mejorar los resultados, aunque también se podría caer en mínimos locales y dar resultados erróneos.
2. Se recomienda en trabajos futuros mejorar la función objetivo en el algoritmo PSO para obtener mejores resultados en el sistema.
3. Generar el mismo análisis utilizando otras herramientas de cálculo y simulación como LabView y comprobar si existen diferencias significativas en los resultados presentados en esta investigación.
4. Desarrollar un algoritmo que permita encontrar las matrices Q y R mediante una optimización PSO para obtener un controlador LQR – Fuzzy
5. Si en trabajos futuros se desea construir el sistema físico, se recomienda tener en consideración las variaciones en las señales que producen los sensores diseñando filtros para aislar estas perturbaciones.

BIBLIOGRAFÍA

- Alvarado Anagonó, G., & Gonzales Zurita, O. (2018). *El control optimo y su desarrollo a través de la historia*.
- Ávila Ramos, C. A., & Suarez Cardenas, F. A. (2012). DISEÑO DE CONTROLADORES BASADOS EN TÉCNICAS DE CONTROL ÓPTIMO LQR+i Y H² PARA UN PROTOTIPO DEL PÉNDULO INVERTIDO SOBRE RUEDAS. *Revista Politécnica*, 8(15), 45–51.
- Belyadi, H., & Haghighat, A. (2021). Fuzzy logic. *Machine Learning Guide for Oil and Gas Using Python*, 381–418. <https://doi.org/10.1016/B978-0-12-821929-4.00003-2>
- Benítez, R., Escudero, G., & Kanaan, S. (2013). *Inteligencia artificial avanzada*.
- Bouza, A., & Alimenti, O. (1997). *Microcontroladores Estándar en el desarrollo de Sistemas de lógica difusa*.
- Bryson, A. E., Ho, Y.-C., & Siouris, G. M. (2008). Applied Optimal Control: Optimization, Estimation, and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(6), 366–367. <https://doi.org/10.1109/tsmc.1979.4310229>
- Çakan, A., Önen, Ü., & İlhan, İ. (2017). *Particle Swarm Optimization Based LQR Control of an Inverted Pendulum*. <https://doi.org/10.18535/etj/v2i5.01>
- Carrillo Paz, A. J. (2011). *Sistemas Automaticos de Control, fundamentos básicos de análisis y modelado* (2da edicion). UNERMB.
- Chang, L., Piao, S., Leng, X., He, Z., & Zhu, Z. (2020). Inverted pendulum model for turn-planning for biped robot. *Physical Communication*, 42, 1–7. <https://doi.org/10.1016/j.phycom.2020.101168>
- Chen, W., & Theodomile, N. (2016). Simulation of a Triple Inverted Pendulum Based on Fuzzy Control. *World Journal of Engineering and Technology*, 04(02), 267–272. <https://doi.org/10.4236/wjet.2016.42026>
- Cheng, F., Zhong, G., Li, Y., & Xu, Z. (1996). Fuzzy control of a double-inverted pendulum. *Fuzzy Sets and Systems*, 79(3), 315–321. [https://doi.org/10.1016/0165-0114\(95\)00156-5](https://doi.org/10.1016/0165-0114(95)00156-5)
- Duarte, C., & Quiroga, J. (2010). Algoritmo PSO para identificación de parámetros en un motor DC. *Rev. Fac. Ing. Univ. Antioquia*, 55, 116–124.
- El-Hawwary, M. I., Elshafei, A. L., Emara, H. M., & Fattah, H. A. A. (2006). Adaptive fuzzy control of the inverted pendulum problem. *IEEE Transactions on Control Systems Technology*, 14(6), 1135–1144. <https://doi.org/10.1109/TCST.2006.880217>
- Eltohamy, K. G., & Kuo, C. Y. (1998). Nonlinear optimal control of a triple link inverted pendulum with single control input. *International Journal of Control*, 69(2), 239–256. <https://doi.org/10.1080/002071798222811>

- Glück, T., Eder, A., & Kugi, A. (2013). Swing-up control of a triple pendulum on a cart with experimental validation. *Automatica*, 49(3), 801–808. <https://doi.org/10.1016/j.automatica.2012.12.006>
- Graichen, K., Treuer, M., & Zeitz, M. (2007). Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica*, 43(1), 63–71. <https://doi.org/10.1016/j.automatica.2006.07.023>
- Gutierrez, M., & Iturralde, S. (2017). *Fundamentos básicos de instrumentación y control* (Universidad Estatal Península de Santa Elena, Ed.; Primera edición). Editorial UPSE.
- Hamza, M. F., Yap, H. J., & Choudhury, I. A. (2015). Genetic Algorithm and Particle Swarm Optimization Based Cascade Interval Type 2 Fuzzy PD Controller for Rotary Inverted Pendulum System. *Mathematical Problems in Engineering*, 2015. <https://doi.org/10.1155/2015/695965>
- Hazem, Z. ben, Fotuhi, M. J., & Bingül, Z. (2020). Development of a Fuzzy-LQR and Fuzzy-LQG stability control for a double link rotary inverted pendulum. *Journal of the Franklin Institute*, 357(15), 10529–10556. <https://doi.org/10.1016/j.jfranklin.2020.08.030>
- Holzhüter, T. (2004). Optimal regulator for the inverted pendulum via Euler-Lagrange backward integration. *Automatica*, 40(9), 1613–1620. <https://doi.org/10.1016/j.automatica.2004.04.012>
- Hurel, J., Mandow, A., & Garcia-Cerezo, A. (2012). Tuning a fuzzy controller by particle swarm optimization for an active suspension system. *IECON Proceedings (Industrial Electronics Conference)*, 2524–2529. <https://doi.org/10.1109/IECON.2012.6388697>
- Kennedy, J., Eberhart, R., & gov, bls. (1995). *Particle Swarm Optimization*.
- Krzysztofik, I., & Koruba, Z. (2012). Adaptive control of anti-aircraft missile launcher mounted on a mobile base. *Theoretical and Applied Mechanics Letters*, 2(4), 043008–043008–4. <https://doi.org/10.1063/2.1204308>
- Lara-Valencia, L. A., Valencia-Gonzalez, Y., & Vital De Brito, J. L. (2015). Uso de lógica difusa para la administración de un sistema disipador de energía en estructuras compuesto por amortiguadores magnetoreológicos. *Revista Facultad de Ingeniería Universidad de Antioquia*, 74, 151–164. <https://www.redalyc.org/articulo.oa?id=43038629014>
- Liu, J. (2018). *Intelligent Control Design and MATLAB Simulation*. Springer. <https://doi.org/https://doi.org/10.1007/978-981-10-5263-7>
- Llata García, J. R., Gonzáles Sarabia, E., Fernández Pérez, D., Torre Ferrero, C., & Robla Gómez, M. S. (2017). Modelado de sistemas de control. In Universidad de Cantabria (Ed.), *Automática*. Creative Commons.
- LÓGICA DIFUSA Y SISTEMAS DE CONTROL. (n.d.). Retrieved June 28, 2022, from <https://cayetanoguerra.github.io/ia/logicadifusa/Logica%20difusa%20y%20sistemas%20de%20control.pdf>

- Medrano-Cerda, G. A. (1997). Robust stabilization of a triple inverted pendulum-cart. *International Journal of Control*, 68(4), 849–866. <https://doi.org/10.1080/002071797223361>
- Ogata, Katsuhiko. (2010). *Ingeniería de control moderna*. Pearson Educación.
- Olmo Castillo, Á. M. (2008). *Tutorial de Introducción de Lógica Borrosa*. Universidad Politécnica de Madrid.
- Perez, M. A., Perez Hidalgo, A., & Perez Berenguer, E. (2008). *INTRODUCCION A LOS SISTEMAS DE CONTROL Y MODELO MATEMÁTICO PARA SISTEMAS LINEALES INVARIANTES EN EL TIEMPO.*
- Prasad, L. B., Tyagi, B., & Gupta, H. O. (2014). Optimal control of nonlinear inverted pendulum system using PID controller and LQR: Performance analysis without and with disturbance input. *International Journal of Automation and Computing*, 11(6), 661–670. <https://doi.org/10.1007/s11633-014-0818-1>
- Ramos, M. G., Vallés Defendine, L., Olaya Flórez, J. J., & Palomino Naranjo, C. (2016). Aplicación del modelo de control en espacios de estado a partir de las pérdidas totales obtenidas del porcentaje de carbono de la chapa de acero al silicio. *Iteckne*, 13(2), 127–136.
- Roca García, J. (2015). *DISEÑO E IMPLEMENTACION DE UN PÉNDULO INVERTIDO SOBRE UN CARRO MOVIL PARA EL LABORATORIO DE MECATRONICA DE LA FIMCP APLICANDO ESTRATEGIAS DE CONTROL PROPORCIONAL INTEGRAL DERIVATIVO (PID)*. Escuela Superior Politécnica del Litoral.
- Rodrigo, J. A. (2019). *Optimización con enjambre de partículas (Particle Swarm Optimization)*. https://www.cienciadedatos.net/documentos/Py02_optimizacion_pso.
- Roose, A. I., Yahya, S., & Al-Rizzo, H. (2017). Fuzzy-logic control of an inverted pendulum on a cart. *Computers and Electrical Engineering*, 61, 31–47. <https://doi.org/10.1016/j.compeleceng.2017.05.016>
- Sancho Caparrini, F. (2019). *Introducción a la Lógica Difusa*. <https://www.cs.us.es/~fsancho/?E=97>.
- Serrano Malagón, L. M., & Alvarez León, Ó. A. (2009). *Introducción al Control Óptimo y aplicación del Principio de Pontryagin al comportamiento de un motor de Corriente Continua*. Universidad Pontificia Bolivariana.
- Shima, T. J., & Bashir, H. A. (2021). Pso-based integral sliding mode controller for optimal swing-up and stabilization of the cart inverted pendulum system. *Nigerian Journal of Technological Development*, 18(2), 88–97. <https://doi.org/10.4314/njtd.v18i2.2>
- Sivanandam, S. N., Sumathi, S., & Deepa, S. N. (2006). *Introduction to Fuzzy Logic using MATLAB*. Springer.
- Tsachouridis, V. A., & Medrano-Cerda, G. A. (1999). Discrete-time H^∞ control of a triple inverted pendulum with single control input. *IEE Proceedings: Control Theory and Applications*, 146(6), 567–576. <https://doi.org/10.1049/ip-cta:19990588>

Velázquez Dodge, C. F., & Mejía Lavalle, M. (2016). Análisis del algoritmo de optimización por enjambre de partículas por medio de una aplicación gráfica 3D. *Research in Computing Science*, 116, 135–140.

ANEXOS

ANEXO A

Código de la planta no lineal en Matlab.

El código presentado a continuación es aplicado en el bloque fcn de la planta no lineal presentado en la figura 3.2.

```
function [qpp0, qpp1, qpp2, qpp3] = fcn(x1, x2, x3, x4, x5, x6, x7, x8, u)
```

```
% masa del carrito
```

```
m0=1;
```

```
% masas de los péndulos
```

```
m1=0.876;
```

```
m2=0.938;
```

```
m3=0.553;
```

```
% longitud de los péndulos
```

```
l1=0.323;
```

```
l2=0.419;
```

```
l3=0.484;
```

```
%Momento de inercia de los péndulos
```

```
J1=0.013;
```

```
J2=0.024;
```

```
J3=0.018;
```

```
% longitud al centro de masas
```

```
c1=0.215;
```

```
c2=0.269;
```

```
c3=0.226;
```

```
% Coeficiente de gravitación
```

```
g=9.81;
```

```
% constantes para simplificación
```

```
a0=m0+m1+m2+m3;
```

```
a1=m2*l1+m3*l1+m1*c1;
```

```
a2=m2*c2+m3*l2;
```

```
a3=m3*c3;
```

```
a4=a2*l1;
```

```
a5=a3*l1;
```

```
a6=a3*l2;
```

```
b1=J1+(m1*c1*c1)+(m2*I1*I1)+(m3*I1*I1);
b2=J2+(m2*c2*c2)+(m3*I2*I2);
b3=J3+m3*c3*c3;
```

```
k1=g*a1;
k2=g*a2;
k3=g*a3;
```

```
% Fricción Viscosa de las juntas
```

```
d0=0; d1=0.215; d2=0.002; d3=0.002;
```

```
% Matriz de masa
```

```
D11=a0;
D12=a1*cos(x3);
D13=a2*cos(x5);
D14=a3*cos(x7);
```

```
D21=a1*cos(x3);
D22=b1;
D23=a4*cos(x5-x3);
D24=a5*cos(x7-x3);
```

```
D31=a2*cos(x5);
D32=a4*cos(x5-x3);
D33=b2;
D34=a6*cos(x7-x5);
```

```
D41=a3*cos(x7);
D42=a5*cos(x7-x3);
D43=a6*cos(x7-x5);
D44=b3;
```

```
% Vector de Fuerzas de Coriolis y de amortiguamiento
```

```
C11=d0;
C12=-a1*sin(x3)*x4;
C13=-a2*sin(x5)*x6;
C14=-a3*sin(x7)*x8;
```

```
C21=0;
C22=d1+d2;
C23=-d2-a4*sin(x5-x3)*x6;
C24=-a5*sin(x7-x3)*x8;
```

```
C31=0;
C32=-d2+a4*sin(x5-x3)*x4;
C33=d2+d3;
C34=-d3-a6*sin(x7-x5)*x8;
```

```
C41=0;
C42=a5*sin(x7-x3)*x4;
C43=-d3+a6*sin(x7-x5)*x6;
C44=d3;
```

%Matriz de Gravedad

```
g1=0;
g2=-k1*sin(x3);
g3=-k2*sin(x5);
g4=-k3*sin(x7);
```

% Variables

```
q=[x1;x3;x5;x7];
qp=[x2;x4;x6;x8];
```

% Matrices del robot

```
D=[D11,D12,D13,D14;D21,D22,D23,D24;D31,D32,D33,D34;D41,D42,D43,D44];
C=[C11,C12,C13,C14; C21,C22,C23,C24; C31,C32,C33,C34; C41,C42,C43,C44];
g=[g1; g2; g3; g4];
```

```
tao=u;
T=[tao;0;0;0];
```

```
N=(C*qp)+g;
```

% Ecuaciones de Estado

```
qpp=inv(D)*(T-N);
qpp0=qpp(1);
qpp1=qpp(2);
qpp2=qpp(3);
qpp3=qpp(4);
```

ANEXO B

Código en Matlab para la linealización y representación en espacios de estados

A continuación se presenta el código generado para realizar la linealización del sistema y obtener las matrices en espacios de estados aplicado en la figura 3.4.

```
clc
clear all
syms xp u m0 m1 m2 m3 l1 l2 l3;
syms x1 x2 x3 x4 x5 x6 x7 x8;

%% Parámetros Péndulo Triple

% masa del carrito
m0=1;

% masas de los péndulos
m1=0.876;
m2=0.938;
m3=0.553;

% longitud de los péndulos
l1=0.323;
l2=0.419;
l3=0.484;

% Momento de inercia de los péndulos
J1=0.013;
J2=0.024;
J3=0.018;

% longitud al centro de masas
c1=0.215;
c2=0.269;
c3=0.226;

% Coeficiente de gravitación
g=9.81;

% constantes para simplificación
a0=m0+m1+m2+m3;
```

```
a1=m2*l1+m3*l1+m1*c1;
a2=m2*c2+m3*l2;
a3=m3*c3;
a4=a2*l1;
a5=a3*l1;
a6=a3*l2;
```

```
b1=J1+(m1*c1*c1)+(m2*l1*l1)+(m3*l1*l1);
b2=J2+(m2*c2*c2)+(m3*l2*l2);
b3=J3+m3*c3*c3;
```

```
k1=g*a1;
k2=g*a2;
k3=g*a3;
```

```
% Fricción Viscosa
```

```
d0=0; d1=0.215; d2=0.002; d3=0.002;
```

```
%% Linealización
```

```
D11=a0;
D12=a1*cos(x3);
D13=a2*cos(x5);
D14=a3*cos(x7);
```

```
D21=a1*cos(x3);
D22=b1;
D23=a4*cos(x5-x3);
D24=a5*cos(x7-x3);
```

```
D31=a2*cos(x5);
D32=a4*cos(x5-x3);
D33=b2;
D34=a6*cos(x7-x5);
```

```
D41=a3*cos(x7);
D42=a5*cos(x7-x3);
D43=a6*cos(x7-x5);
D44=b3;
```

```
D=subs([D11,D12,D13,D14;D21,D22,D23,D24;D31,D32,D33,D34;D41,D42,D43,D44],...
```

```
{x1,x2,x3,x4,x5,x6,x7,x8,u,m0,m1,m2,m3,l1,l2,l3},{0,0,0,0,0,0,0,0,2.4,1.323,1.389,0.8655,0.402,0.332,0.72});
```

```
D=double(D);
Din=inv(D);
```

```

C11=d0;
C12=-a1*sin(x3)*x4;
C13=-a2*sin(x5)*x6;
C14=-a3*sin(x7)*x8;

C21=0;
C22=d1+d2;
C23=-d2-a4*sin(x5-x3)*x6;
C24=-a5*sin(x7-x3)*x8;

C31=0;
C32=-d2+a4*sin(x5-x3)*x4;
C33=d2+d3;
C34=-d3-a6*sin(x7-x5)*x8;

C41=0;
C42=a5*sin(x7-x3)*x4;
C43=-d3+a6*sin(x7-x5)*x6;
C44=d3;

C=subs([C11,C12,C13,C14;C21,C22,C23,C24;C31,C32,C33,C34;C41,C42,C43,C44],...

{x1,x2,x3,x4,x5,x6,x7,x8,u,m0,m1,m2,m3,l1,l2,l3},{0,0,0,0,0,0,0,0,2.4,1.323,1.389,0.8655,0.402,0.332,0.72});
C=double(C);

G=[0,0,0,0;0,k1,0,0;0,0,k2,0;0,0,0,k3];

%% Matriz A
A11=zeros(4);
A12=eye(4);
A21=-Din*G;
A22=-Din*C;
A=[A11 A12;A21 A22]

%% Matriz B
ho=[1;0;0;0];
B=[0;0;0;0;Din*ho]

%% Matriz C
C0=[1 0 0 0 0 0 0; 0 1 0 0 0 0 0; 0 0 1 0 0 0 0; 0 0 0 1 0 0 0]

%% Matriz D
D=[0 0 0 0]'

```

ANEXO C

Código en Matlab para diseñar el controlador LQR

El código presentado es continuación del código anterior ya que se encuentra dentro del mismo script.

```
%% Matrices de Ponderación Q y R
```

```
Q=[10 0 0 0 0 0 0 0; 0 100 0 0 0 0 0 0; 0 0 100 0 0 0 0 0; 0 0 0 100 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0];
```

```
R=1;
```

```
%% CONTROLADOR LQR
```

```
P=care(A,B,Q,R);
```

```
K=inv(R)*B'*P % usando la solución P
```

```
Ac = [(A-B*K)];
```

```
Bc = [B];
```

```
Cc = [C0];
```

```
Dc = [D];
```

```
%% Representación gráfica dentro del mismo script
```

```
states = {'xc' 'x1' 'x2' 'x3' 'xc_p' 'x1_p' 'x2_p' 'x3_p'};
```

```
inputs = {'u'};
```

```
outputs = {'xc' 'x1' 'x2' 'x3'};
```

```
sys = ss(A,B,C0,D,'statername',states,'inputname',inputs,'outputname',outputs);
```

```
step(sys)
```

```
eig_Ac = eig(Ac);
```

```
sys_cl =
```

```
ss(Ac,Bc,Cc,Dc,'statername',states,'inputname',inputs,'outputname',outputs);
```

```
Nc = 1/dcgain(sys_cl(1));
```

```
sys_cl =
```

```
ss(Ac,Bc*Nc,Cc,Dc,'statername',states,'inputname',inputs,'outputname',outputs);
```

```
figure; step(sys_cl)
```

ANEXO D

Código en Matlab para el script de la función objetivo del esquema Gain – PSO.

```
function F = Fval100(Gain)

    K1 = Gain(1);
    K2 = Gain(2);
    K3 = Gain(3);
    K4 = Gain(4);
    K5 = Gain(5);
    K6 = Gain(6);
    K7 = Gain(7);
    K8 = Gain(8);

    % Modify simulation settings
    simopt =
simset('solver','ode5','SrcWorkspace','Current','DstWorkspace','Current');

    % Initialize sim options
    [tout,xout,yout] = sim('PSO',[0 6],simopt);%Simulate dynamic system

    e=sqrt(sum((Error).^2));
    e1=sqrt(sum((Error1).^2));
    e2=sqrt(sum((Error2).^2));
    e3=sqrt(sum((Error3).^2));

    % Objctive function
    F=e+e1+e2+e3

end
```

ANEXO E

Código en Matlab para el esquema Gain – PSO.

```
%% Tuning of fuzzy controller using PSO for a triple pendulum inverted
```

```
%% Initialization
```

```
clear all
```

```
clc
```

```
Size = 100; % Size of the swarm
```

```
max_iter = 1000; % Maximum number of iterations
```

```
dim = 8; % Dimension of the problem
```

```
c2 = 2.4; % PSO parameter C1 "cognitive acceleration"
```

```
c1 = 2.4; % PSO parameter C2 "social acceleration"
```

```
w = 0.8; % PSO momentum or inertia "velocity weight"
```

```
tic;
```

```
%Generate Random Numbers
```

```
R1 = rand();
```

```
R2 = rand();
```

```
% Initializing swarm, velocities, and position
```

```
Pcurrent = diag([1 1 1 1 1 1 1 1])*rand(dim,Size);
```

```
V = 5*rand(dim, Size);
```

```
Pbest_local = Pcurrent ; % initial Pbest_local position vals
```

```
%Evaluate fitness of initial swarm
```

```
fitness = zeros(1, Size);
```

```
for i = 1:Size
```

```
    fitness(i) = Fval100(Pcurrent(:,i));
```

```
end
```

```
LBfitness = fitness ; %initially, LBfitness is same as position
```

```
% Determine fitness and index of best particle.
```

```
[bestFitness,bestIndex] = min(LBfitness) ;
```

```
iter = Size; % Fitness evaluations above count as iterations.
```

```
while iter < max_iter
```

```
    % Pick index for a random particle from the swarm.
```

```
    i = ceil(Size * rand());
```

```

%New velocity and position for particle
V(:,i) = w*(V(:,i)) + c1*R1*(Pbest_local(:,i)-Pcurrent(:,i)) +
c2*R2*(Pbest_local(:,bestIndex)-Pcurrent(:,i));
Pcurrent(:,i) = Pcurrent(:,i) + V(:,i);

%z=max(Pcurrent(:,i));
if Pcurrent(8,i)<100
%evaluate fitness of a new swarm
newFitness = Fval100(Pcurrent(:,i));

% Update positions and fitness of particles.
if newFitness < fitness(i)
fitness(i) = newFitness;

% Update particle's best-known position.
Pbest_local(:,i) = Pcurrent(:,i);

% Update swarm's best-known position.
if newFitness < bestFitness

% Index into the Pbest_local array.
bestIndex = i;
bestFitness = newFitness;
end
end
iter = iter + 1 %Increment of counter.
end

end

% Return best found solution, fitness already set.
bestX = Pbest_local(:,bestIndex)
fitnes=bestFitness
K1 =bestX(1,1);
K2 =bestX(2,1);
K3 =bestX(3,1);
K4 =bestX(4,1);
K5 =bestX(5,1);
K6 =bestX(6,1);
K7 =bestX(7,1);
K8 =bestX(8,1);

toc

```

ANEXO F

Código en Matlab para el script de la función objetivo del esquema Fuzzy – PSO.

```
function F = Fval70(Gain)

    Afis = Gain(1);
    Bfis = Gain(2);
    Cfis = Gain(3);
    Afis1= Gain(4);
    Bfis1= Gain(5);
    Cfis1= Gain(6);
    Afis2= Gain(7);
    Bfis2= Gain(8);
    Cfis2= Gain(9);
    Afis3= Gain(10);
    Bfis3= Gain(11);
    Cfis3= Gain(12);

    % Modify simulation settings
    simopt =
simset('solver','ode5','SrcWorkspace','Current','DstWorkspace','Current');

    % Initialize sim options
    [tout,xout,yout] = sim('Simulink70',[0 6],simopt);%Simulate dynamic system

    e=sqrt(sum((Error).^2)); % compute the error of cart
    OverA=sqrt(max((Car).^2)); % compute the overshoot of cart
    e1=sqrt(sum((Error1).^2)); % compute the error of pendulum 1
    OverA1=sqrt(max((Angle1).^2)); % compute the overshoot of pendulum 1
    e2=sqrt(sum((Error2).^2)); % compute the error of pendulum 2
    OverA2=sqrt(max((Angle2).^2)); % compute the overshoot of pendulum 2
    e3=sqrt(sum((Error3).^2)); % compute the error of pendulum 3
    OverA3=sqrt(max((Angle3).^2)); % compute the overshoot of pendulum 3

    % Objective function
    F=300*e+e1*320+e2*360+e3*400+OverA*0.3+OverA1*0.45+OverA2*0.6+OverA3*0.8

end
```

ANEXO G

Código en Matlab para el esquema Fuzzy – PSO.

```
%% Tuning of fuzzy controller using PSO for a triple pendulum inverted

%% Initialization
clear all
clc
Size = 24;    % Size of the swarm
max_iter = 100; % Maximum number of iterations
dim = 12;    % Dimension of the problem
c2 = 2.4;    % PSO parameter C1 "cognitive acceleration"
c1 = 2.4;    % PSO parameter C2 "social acceleration"
w = 0.9;     % PSO momentum or inertia "velocity weight"

tic;

%Generate Random Numbers
R1 = rand();
R2 = rand();

% Initializing swarm, velocities and position
Pcurrent = diag([1 1 1 1 1 1 1 1 1 1 1 1])*rand(dim,Size);
V = 5*rand(dim, Size);
Pbest_local = Pcurrent ; % initial Pbest_local position vals

%Evaluate fitness of initial swarm
fitness = zeros(1, Size);
for i = 1:Size
    fitness(i) = Fval70(Pcurrent(:,i));
end

LBfitness = fitness ; %initially, LBfitness is same as position

% Determine fitness and index of best particle.
[bestFitness,bestIndex] = min(LBfitness) ;
iter = Size; % Fitness evaluations above count as iterations.

while iter < max_iter

    % Pick index for a random particle from the swarm.
    i = ceil(Size * rand());
```

```

%New velocity and position for particle
V(:,i) = w*(V(:,i)) + c1*R1*(Pbest_local(:,i)-Pcurrent(:,i)) +
c2*R2*(Pbest_local(:,bestIndex)-Pcurrent(:,i));
Pcurrent(:,i) = Pcurrent(:,i) + V(:,i);

if Pcurrent(12,i)<10
%evaluate fitness of a new swarm
newFitness = Fval70(Pcurrent(:,i));

% Update positions and fitness of particles.
if newFitness < fitness(i)
fitness(i) = newFitness;

% Update particle's best-known position.
Pbest_local(:,i) = Pcurrent(:,i);

% Update swarm's best-known position.
if newFitness < bestFitness

% Index into the Pbest_local array.
bestIndex = i;
bestFitness = newFitness;

end
end
iter = iter + 1 %Increment of counter.
end
end

% Return best found solution, fitness already set.
bestX = Pbest_local(:,bestIndex)
fitnes=bestFitness

Afis = bestX(1,1);
Bfis = bestX(2,1);
Cfis = bestX(3,1);
Afis1= bestX(4,1);
Bfis1= bestX(5,1);
Cfis1= bestX(6,1);
Afis2= bestX(7,1);
Bfis2= bestX(8,1);
Cfis2= bestX(9,1);
Afis3= bestX(10,1);
Bfis3= bestX(11,1);
Cfis3= bestX(12,1);

toc

```