

UNIVERSIDAD SANTO TOMÁS

IMPLEMENTACIÓN DE UN ALGORITMO DE APRENDIZAJE DE MÁQUINA PARA LA SINTONIZACIÓN DE UN CONTROLADOR APLICADO A UN PÉNDULO INVERTIDO

Realizado por

Wilson Augusto Guarnizo Hernández

Proyecto de Grado presentado en cumplimiento del requisito
para optar por el grado de Ingeniería Electrónica



Grupo de Estudio y Desarrollo en Robótica
Facultad de Ingeniería Electrónica
División de Ingenierías

Julio de 2022

IMPLEMENTACIÓN DE UN ALGORITMO DE APRENDIZAJE DE MÁQUINA PARA LA SINTONIZACIÓN DE UN CONTROLADOR APLICADO A UN PÉNDULO INVERTIDO

Realizado por

Wilson Augusto Guarnizo Hernández

Proyecto de Grado presentado en cumplimiento del requisito
para optar por el grado de Ingeniería Electrónica

Dirigido por

José Guillermo Guarnizo Marín

Co-Dirigido por

Armando Mateus Rojas

Grupo de Estudio y Desarrollo en Robótica
Facultad de Ingeniería Electrónica
División de Ingenierías

Julio de 2022

DEDICATORIA

Dedico este proyecto a mis padres, los cuales siempre me dieron motivación y soporte durante todo el desarrollo de mi carrera y a los cuales les debo el agradecimiento de haberme guiado como profesional, y mas importante como una persona de principios y valores católicos.

Wilson Guarnizo

AGRADECIMIENTOS

Agradezco primeramente a Dios por darme la guía y permitirme concluir este capítulo de mi vida educativa como profesional, a toda mi familia por ser el mecanismo que me impulsa cada día a mejorar y aprender nuevas cosas para poder devolverles todo lo que me han dado hasta la fecha, a mis directores de grado José Guillermo Guarnizo y Armando Mateus por aportarme los conocimientos necesarios para la realización de este proyecto, y por último agradezco a todas las personas que pude conocer y aprender de ellas en el transcurso de esta carrera.

Wilson Guarnizo

Índice general

Índice de Figuras	VI
Índice de Cuadros	VIII
Índice de Abreviaturas	IX
1. Introducción	3
2. Planteamiento del Problema	5
3. Justificación	8
4. Impacto Social	10
5. Objetivos	11
5.1. Objetivo General:	11
5.2. Objetivos Específicos:	11
6. Estado del Arte	12
6.1. Breve historia del control de los péndulos invertidos:	12
6.2. Modelado y control	13
6.3. Técnicas de Inteligencia Artificial:	14
6.4. Aplicaciones de la técnicas de control en la industria:	15
6.5. Selección de técnica de aprendizaje de maquina	15
7. Marco Teórico	18
7.1. Modelamiento Péndulo Rotatorio Invertido	18
7.1.1. Linealización	20
7.2. Sistemas Bio-inspirados	24
7.2.1. Algoritmos Evolutivos	24
7.2.2. Teoría de selección clonal	26
7.2.3. Algoritmo de selección clonal	29
7.2.3.1. Paso 1: Inicialización	29
7.2.3.2. Paso 2: Selección	29
7.2.3.3. Paso 3: Clonación y Variación Genética	30

7.2.3.4.	Paso 4: Evaluación de la afinidad	30
7.2.3.5.	Paso 5: Autorregulación	30
7.2.3.6.	Paso 6: Repetición en ciclo	30
8.	Desarrollo Metodológico	34
8.0.1.	Modelo métrico a utilizar	34
8.0.2.	Metodología	35
9.	Diseño y validación de controladores	38
9.1.	Simulación	38
9.1.1.	Modelo físico del péndulo invertido en Simulink	38
9.1.2.	Diseño del controlador clásico	39
9.1.2.1.	Diseño de controlador por LGR	43
9.1.2.2.	Algoritmo de sintonización CLONAL	48
9.1.2.3.	Paso 1: Definición de población de anticuerpos	50
9.1.2.4.	Paso 2: Cálculo de la función de aptitud u objetivo	50
9.1.2.5.	Paso 3: Selección de los mejores 2 anticuerpos	51
9.1.2.6.	Paso 4: Clonación de los 2 mejores anticuerpos	52
9.1.2.7.	Paso 5: Mutación de los 2 mejores anticuerpos	52
9.1.2.8.	Paso 6: Cálculo de la función de aptitud u objetivo con los anticuerpos clonados y mutados	52
9.1.2.9.	Paso 7: Selección y almacenamiento del mejor anticuerpo del conjunto clonado y mutado	52
9.1.2.10.	Paso 8: Eliminación y reemplazo de los anticuerpos con menor afinidad para reinicio del ciclo	53
9.1.3.	Modelo implementado en Simulink	53
9.1.4.	Resultados obtenidos en simulación	54
9.1.4.1.	Sintonización CLONAL con un 20 % de mutación en Simulink	54
9.1.4.2.	Sintonización CLONAL con un 50 % de mutación en Simulink	61
9.1.4.3.	Sintonización CLONAL con un 80 % de mutación en Simulink	66
9.2.	Implementación física del Péndulo Rotatorio Invertido	71
9.2.1.	Implementación del algoritmo CLONAL al péndulo rotatorio invertido	71
9.2.1.1.	Reconocimiento del controlador por defecto propiedad de QUANSER	72
9.2.1.2.	Asignación de parámetros usando algoritmo de sintonización clonal sobre el modelo de control del péndulo rotatorio invertido	73
9.2.2.	Resultados obtenidos en la implementación real del Péndulo Rotatorio Invertido	75
9.2.2.1.	Sintonización CLONAL con un 70 % de mutación sobre el modelo físico en QUARC- Simulink	75
9.2.3.	Pruebas del controlador base y controlador sintonizado con diferentes valores de perturbación	80
9.3.	Discusión	83
10.	Conclusiones	86

11. Trabajos Futuros	89
Bibliografía	90

Índice de Figuras

1.	Esquemático del Péndulo Invertido [49]	18
2.	Diagrama de bloques del algoritmo evolutivo [Autoría propia].	26
3.	Identificación de los antígenos por medio de anticuerpos [52]	27
4.	Principio de selección clonal [55]	28
5.	Diagrama de flujo que se utiliza para aplicar un proceso de optimización CLONAL [Autoría propia]	31
6.	Curva de respuesta a escalón unitario de un sistema [56]	34
7.	Descripción y implementación de la metodología en el desarrollo del proyecto de grado [Autoría propia]	37
8.	Modelo físico implementado en simulink [Autoría propia]	38
9.	Modelo físico Implementado en simulink [Autoría propia]	39
10.	Representación para transformación de un sistema SIMO a un sistema SISO [Autoría propia]	42
11.	Respuesta de la planta en lazo abierto [Autoría propia]	43
12.	Ubicación de los polos del sistema en lazo abierto dentro del plano complejo [Autoría propia]	43
13.	Plano complejo con asignación de región de diseño para el controlador [Autoría propia]	44
14.	Plano complejo con asignación de región de diseño para el controlador [Autoría propia]	44
15.	Respuesta al escalón con el controlador diseñado [Autoría propia]	45
16.	Esquema de control para la planta del péndulo rotatorio invertido para el ángulo alpha en Simulink [Autoría propia]	46
17.	Respuesta del sistema de control para la planta del péndulo rotatorio invertido para el ángulo Alpha [Autoría propia]	47
18.	Respuesta del sistema de control para la planta del péndulo rotatorio invertido para el ángulo Theta [Autoría propia]	47
19.	Diagrama de bloques del sistema a implementar [Autoría propia]	48
20.	Diagrama de flujo que se utiliza para aplicar un proceso de optimización CLONAL [Autoría propia]	49
21.	Diagrama de control implementado en Simulink [Autoría propia]	53
22.	Respuesta del controlador de (0-7) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]	56

23.	Respuesta del actuador de (0-7) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]	57
24.	Respuesta del controlador de (7-13) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]	58
25.	Respuesta del actuador de (7-13) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]	59
26.	Gráfica de afinidad por ciclo de iteración del algoritmo CLONAL [Autoría propia]	60
27.	Respuesta del controlador de (0-7) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]	62
28.	Respuesta del actuador de (0-7) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]	63
29.	Respuesta del controlador de (7-10) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]	64
30.	Respuesta del actuador de (7-10) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]	64
31.	Gráfica de afinidad por ciclo de iteración del algoritmo CLONAL [Autoría propia]	66
32.	Respuesta del controlador de (0-7) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]	67
33.	Respuesta del actuador de (0-7) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]	68
34.	Respuesta del controlador de (7-10) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]	69
35.	Respuesta del actuador de (7-10) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]	69
36.	Gráfica de afinidad por ciclo de iteración del algoritmo CLONAL [Autoría propia]	71
37.	Conexión y implementación física de la planta QUANSER [Autoría propia] . . .	71
38.	Diagrama de control por RVE base que viene por defecto con planta QUANSER [Autoría propia]	72
39.	Respuestas del controlador por defecto y control resultado del algoritmo con factor de mutación del 70 % [Autoría propia]	77
40.	Respuestas del actuador del controlador por defecto y control resultado del algoritmo CLONAL con factor de mutación del 70 % [Autoría propia]	79
41.	Respuestas del controlador base y el control sintonizado resultado del algoritmo CLONAL con 20 % de perturbación [Autoría propia]	80
42.	Respuestas del actuador base y el actuador sintonizado resultado del algoritmo CLONAL con 20 % de perturbación [Autoría propia]	81
43.	Respuestas del controlador base y el control sintonizado resultado del algoritmo CLONAL con 50 % de perturbación [Autoría propia]	81
44.	Respuestas del actuador base y el actuador sintonizado resultado del algoritmo CLONAL con 50 % de perturbación [Autoría propia]	82
45.	Respuestas del controlador base y el control sintonizado resultado del algoritmo CLONAL con 80 % de perturbación [Autoría propia]	82
46.	Respuestas del actuador base y el actuador sintonizado resultado del algoritmo CLONAL con 80 % de perturbación [Autoría propia]	83

Índice de Tablas

2.	Tabla de jerarquía para selección de técnica de aprendizaje de máquina.	16
3.	Tabla de selección del algoritmo Bioinspirado para la sintonización de controladores	16
4.	Parámetros físicos de la planta del péndulo rotatorio invertido	19
5.	Resultados obtenidos de la sintonización durante 13 iteraciones del algoritmo CLONAL con 20 % de mutacion	55
6.	Resultados de función de afinidad obtenidos de la sintonización durante 13 iteraciones del algoritmo CLONAL con 20 % de mutacion	60
7.	Resultados obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 50 % de mutación	61
8.	Resultados de función de afinidad obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 50 % de mutación	65
9.	Resultados obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 80 % de mutacion	66
10.	Resultados de función de afinidad obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 80 % de mutación	70
11.	Resultados obtenidos de la sintonización de la planta durante 1 iteración del algoritmo CLONAL con 70 % de mutación	76
12.	Comparación de resultados del error cuadrático medio de estado estacionario del controlador base y el controlador sintonizado con algoritmo Clonal	78
13.	Resultados de función de afinidad obtenidos de la sintonización durante 1 iteración del algoritmo CLONAL con 70 % de mutación en planta física QUANSER	78
14.	Comparación de resultados del valor medio del voltaje del actuador proveniente del controlador base y el controlador sintonizado con un factor de mutación del 70 %	80

Índice de Abreviaturas

CRAI	Centro de R ecursos para el A prendizaje y la I ntestigación
FAC	Fully-Autonomous C ontrol
IEEE	Institute of Electrical and Electronics E ngineer
LQR	Linear Quadratic R egulator
PID	Proportional Integral D erivative
PSO	Particle Swarm O ptimization
USTA	Universidad S anto T omás
CS	Cukoo S earch
VTOL	Vertical Take-Off and L anding
LMI	Linear M atriz- I nequality
LGR	Lugar G eométrico de las R aices
RMSE	R aiz del E rror C uadrático M edio
Mp	Sobrepaso M áximo
Ts	Tiempo de E stablecimiento
Ess	Error de Estado E stacionario

Resumen

En este proyecto de grado se presenta el diseño e implementación de una sintonización de un controlador de posición angular para un péndulo rotatorio invertido, utilizando un algoritmo Bio-inspirado de selección clonal para el ajuste de los parámetros de un controlador PID (Proporcional, Integral y Derivativo). Para esto se detalla el proceso de creación del controlador PID, el modelo simulado en MATLAB del péndulo rotatorio invertido y la programación del algoritmo de selección clonal para la sintonización de las ganancias que requiere el controlador para su acción de control. Así mismo, se realizan comparaciones de desempeño de controladores en su proceso transitorio en donde se mide que tanto mejora el sobrepaso máximo, tiempo de establecimiento y error de estado estacionarios de los controladores base, y el controlador sintonizado con el algoritmo clonal. Por último, se verifica el desempeño de la sintonización del algoritmo de selección clonal sobre un controlador RVE (Realimentación de variables de estado) aplicado a una planta física QUANSER de un Péndulo Rotatorio Invertido.

Abstract

This degree project presents the design and implementation of a tuning of an angular position controller for an inverted rotary pendulum, using a Bio-inspired clonal selection algorithm for adjusting the parameters of a PID controller. This details the process of creating the PID controller, the simulated model in MATLAB of the inverted rotary pendulum, and the programming clonal selection algorithm for tuning the gains required by the controller for its control action. Likewise, comparisons of the performance of controllers in their transitive process are made, where they measure how much improves the overshoot, time setting and

steady-state error of the base controllers and the controller tuned to the clonal algorithm. Finally, the tuning performance of the clonal selection algorithm on an FSF (Full State Feedback) controller applied to a QUANSER Rotary Inverted Pendulum physical plant is verified.

Capítulo 1

Introducción

Los controladores que se utilizan en las plantas industriales, vienen por defecto con ganancias en los parámetros del controlador con valores unitarios, esto debido a que, por lo general en los procesos industriales, el sistema cumple correctamente la función que requiere el operario. Pero a largo plazo, el sistema de control para este tipo de plantas industriales puede llegar a acarrear costos elevados a la empresa que los utilice, esto debido a que no se ha realizado una sintonización previa de las ganancias del controlador lo que permitiría mejorar el consumo y el rendimiento en comparación a los valores de fábrica de la planta [1].

Una forma de solución a lo anteriormente mencionado es la implementación de algoritmos inteligentes, los cuales se caracterizan por utilizar procesamiento de máquina para resolver problemas, en donde cada uno de estos tipos de algoritmos posee diversas características para la aplicación que se desee sintonizar.

En este caso se desea implementar un algoritmo evolutivo el cual se inspire en un modelo de selección clonal, este algoritmo se enfoca en la neutralización de bacterias para la solución de problemas de tipo de reconocimiento de patrones u optimización de funciones. Para el uso de este algoritmo se plantea utilizar un controlador PID en donde el algoritmo asigne los valores de los parámetros del controlador para ajustar la posición angular de un péndulo rotatorio invertido utilizando la planta SRV02 Quanser.

Como contenidos, el documento propone varios capítulos donde se estudia y desarrolla el péndulo rotatorio invertido, la diversidad de controladores y algoritmos que existen los cuales

se desarrollan actualmente para realizar una creación de un controlador, como también una sintonización del mismo. En el capítulo 2 se menciona la problemática actual que tienen los controladores que vienen por defecto en los sistemas industriales y como los algoritmos evolutivos tienen la capacidad de afrontarlos. En el capítulo 3 se dan los motivos principales que inspiraron el desarrollo del proyecto. En el capítulo 4 se mencionan los aportes sociales que busca conseguir el proyecto permitiendo ser un caso de estudio que traiga desarrollo en la materia de control en Colombia. En el capítulo 5 se hace una breve contextualización histórica que viene desarrollando el péndulo invertido a través de los años, como también varios controladores clásicos y modernos usados para su funcionamiento y como estos vienen generando desarrollo al momento de unirlos con técnicas de inteligencia artificial. En el capítulo 6 se plantean los objetivos del proyecto. En el capítulo 7 se hace una investigación profunda de los conceptos y fuentes que requiere el desarrollo del proyecto. En el capítulo 8 se explica el desarrollo del trabajo en donde se hace registro de cada proceso que se requirió dentro del proyecto, utilizando los conceptos y fuentes agregados en el marco teórico, y como estos permiten la obtención de los resultados propuestos en los objetivos. En el capítulo 9 se registran los resultados e imágenes de desempeño que se obtuvieron aplicando el algoritmo de selección clonal sobre un controlador PID propuesto, junto también a un controlador desarrollado por QUANSER para su implementación física. Por último, en el capítulo 10 se hacen las correspondientes conclusiones del proyecto en donde se destaca el beneficio del uso de las metodologías de inteligencia artificial, tomando como base los resultados obtenidos durante el desarrollo del proyecto y proponer diferentes trabajos futuros que puedan aportar al campo del control junto con la inteligencia artificial.

Capítulo 2

Planteamiento del Problema

Una problemática que se presenta en las industrias de procesos, las cuales utilizan dispositivos con controladores es que casi el 30 % de estos dispositivos, funcionan con las ganancias que vienen por defecto por el fabricante en el sistema. Esto provoca en la mayoría de casos que los operarios de estas máquinas, no consideren la posible importancia a largo plazo que tiene la sintonización en un proceso de control, esto debido a que los parámetros instaurados por el fabricante en el controlador, permiten al sistema ser estable. También se encuentra los casos en que los operarios hacen la sintonización en el control de manera empírica, lo cual puede llegar a resolver la problemática a corto plazo, pero puede presentarse problemas a medio y largo plazo, ya que las necesidades de control dependiendo de un caso en particular que se presente cambian, lo cual hace que se requiera de un nuevo ajuste de los parámetros del controlador [2] . Un tipo de planta en que se puede presentar esto es la que se trata en este proyecto la cual es el péndulo invertido.

Un ejemplo que evidencia la problemática de la falta de sintonización en los controladores , se encuentra en este caso al oeste de Argentina, en donde en un proceso de un sistema de molienda por impacto con clasificación, se presentaban paradas frecuentes en su producción debido a sobrepasos de corriente en donde el motor central se atascaba en las cámaras del molino por material mineral. Estos errores hacían que se presentaran problemas en la secuencia de arranque que compone al sistema clasificador. La forma parcial de corrección para este problema era la aplicación de un control manual por parte del operario encargado, lo cual provocaba desgaste a corto plazo y generaba retrasos en la producción. También el proceso preventivo para evitar que la maquina llegara a valores altos de corriente , se

supervisaba a partir de un operario el cual manipulaba el controlador de corriente manualmente para evitar los sobrepicos [1].

El péndulo rotatorio invertido como sistema representa un reto para los controladores convencionales debido a su alta complejidad al momento de implementarlos. Esto se debe a que el péndulo rotatorio invertido (RIP) es un sistema altamente no lineal el cual depende de una linealización que acarrea una elevada dificultad matemática, pero aun así han sido considerados como objeto de estudio e investigación a lo largo de los años en donde se han implementado varios sistemas de control con diferentes técnicas [3, 4].

La aplicación del péndulo invertido se encuentra asociada a aplicaciones en tiempo real como puede ser el Vehículo Segway, el cual es un mecanismo de transporte que permite moverse, pararse y equilibrarse a partir de su mismo movimiento [5]. También el péndulo invertido se encuentra asociado en el control del balance de robots humanoides, esto debido a que las perturbaciones que los robots experimentan están relacionadas a un modelo de péndulo invertido con alta no linealidad [6].

La dinámica del péndulo invertido se basa en un modelamiento de ecuaciones no lineales el cual relaciona variables de posición de los ángulos y sus velocidades respectivamente. También dentro del modelo se expresan las constantes que interactúan directamente y afectan las ecuaciones de movimiento, y posición no lineales, las cuales vienen asignadas por los factores físicos del motor, brazo y péndulo [7]. Para aplicar una técnica convencional de control, primero se debe realizar un modelo lineal del péndulo invertido. Existen métodos como: Euler-Lagrange, Euler-Newton y Series de Taylor, los cuales se encargan de linealizar el sistema en un punto de equilibrio que permita aplicar un control efectivo atado a ciertas condiciones [8]. El problema surge al momento de utilizar las anteriores técnicas mencionadas, las cuales acarrearán un proceso matemático arduo en donde puede llegar a ser muy fácil equivocarse en su ejecución, y donde respectivamente están restringidas a ciertas condiciones de operación como es depender de un punto de trabajo.

Existen varios tipos de controladores lineales que se pueden implementar a un modelo linealizado del péndulo invertido como son los reguladores cuadráticos lineales (LQR) [9], controladores proporcional, integral derivativo (PID) y por realimentación de variables de estado (RVE) [10]. El diseño de los controladores anteriormente mencionados, se pueden ajustar para lograr una sintonización de los parámetros del controlador, aplicando diferentes

métodos clásicos como puede ser Ziegler-Nichols [11], o integrando técnicas más avanzadas de control como puede ser controladores LMI [12].

El problema de los anteriores métodos al momento de sintonizar los parámetros del controlador, es que requieren de un dominio de control amplio, conocimiento profundo de los parámetros que componen a la planta para su implementación e intervención directa del diseñador para el ajuste de los parámetros. Por este motivo se propone a partir de algoritmos de aprendizaje de máquina, encontrar la mejor respuesta de control posible a partir de pruebas reiterativas en donde se compara el rendimiento por medio de figuras de mérito seleccionadas con el fin de mejorar el desempeño del controlador.

De acuerdo a lo anterior mencionado , se plantea la siguiente pregunta problema:

¿Cómo integrar técnicas de aprendizaje de máquina que permita la sintonización de los parámetros de un controlador seleccionado, aplicado a un péndulo invertido para un punto de trabajo específico?

Capítulo 3

Justificación

Los motivos que llevaron al desarrollo de este proyecto se encuentran fundamentados en la aplicación de estrategias de aprendizaje de máquina, que buscan ofrecer una alternativa de control a los sistemas de péndulo rotatorio invertido, los cuales permitan una versatilidad y sintonización adecuada de los parámetros del controlador. Así mismo, el propósito del método que se implementó en el presente trabajo es contribuir en la aplicación de metodologías de control, que ofrezcan un desarrollo para plantas que presenten no linealidades en su dinámica y que requieran de una adaptabilidad para ajustar sus parámetros, y obtener respuestas de control, que cumplan con los requisitos de diseño que el usuario asigne. También se busca contribuir en los proyectos de grado en las líneas de investigación de inteligencia computacional del grupo GED (Grupo de Estudio y Desarrollo en Robótica) de la Universidad Santo Tomás, en donde este proyecto permita vincular las áreas de algoritmos Bio-inspirados y las técnicas de control clásicas para un desarrollo práctico de laboratorios dentro de un espacio compartido, para lograr incentivar el uso del procesamiento de máquina para resolver problemas de control.

Como se detalla más adelante en el estado del arte, se han reportado en la literatura, diferentes trabajos que implementan técnicas de aprendizaje de máquina que se utilizan para estabilizar el péndulo invertido. Como por ejemplo algoritmos de lógica difusa [13, 14], o implementación de algoritmos de aprendizaje por refuerzo [15, 16], pero estos sistemas de control terminan dando una respuesta única, la cual si se desea mejorar, requiere una intervención directa en el algoritmo por algún operario humano; es por esto que para poder obtener una respuesta con un mejor desempeño en la planta, se busca ajustar alguna variable

como se aplica en el controlador por búsqueda Cuckoo [17] .

Al poder sintonizar el controlador con el fin de mejorar la respuesta de la posición angular del sistema del péndulo invertido, es posible cambiar el desempeño del actuador, lo cual hace que se pueda reducir el sobre paso máximo, el tiempo de establecimiento o el error de estado estacionario dependiendo del parámetro del controlador a modificar. Esta sintonización automática va a ir variando los parámetros del controlador por medio del algoritmo de aprendizaje con el objetivo de obtener la mejor respuesta posible sin intervención humana; todo esto sin tener que requerir volver a rediseñar un controlador distinto para la planta [17, 18].

Capítulo 4

Impacto Social

Con este proyecto se propuso aportar una base con respecto al desarrollo de sistemas de control, utilizando sintonización de controladores para plantas no lineales como el péndulo invertido incorporando técnicas de aprendizaje de máquina. El desarrollo planteado podría emplearse para mejorar el rendimiento de procesos en varias aplicaciones donde los péndulos invertidos intervienen, como en la industria de la robótica [19] , lanzamiento de cohetes, transporte [20],[21] y cualquier otro enfoque donde se requiera un control para un dispositivo que presente características similares al péndulo invertido.

También el estudio de estos métodos para dispositivos como el péndulo invertido en Colombia, permitiría en las instituciones educativas superiores adoptar nuevos métodos de aprendizaje, y mejorar el espectro de posibilidades de uso de controladores los cuales integren estas técnicas y proporcionen soluciones más versátiles para la comodidad del usuario. También el proyecto podría llegar a ser aplicado dentro del marco de los objetivos de desarrollo sostenible, más específicamente el objetivo 9. Este objetivo busca el desarrollo económico bajo la inversión en infraestructura e innovación de nuevas tecnologías que permitan el crecimiento en las industrias. Esto busca promover un desarrollo al acceso a la tecnología computacional para aumentar la capacidad tecnológica de los sectores industriales en Colombia, fomentando la innovación y aumentando de aquí a 2030, el número de personas que trabajan en investigación y desarrollo[22].

Capítulo 5

Objetivos

5.1. Objetivo General:

Implementar un sistema de control de posición angular de un péndulo invertido , por medio de la sintonización de parámetros de un controlador clásico basada en técnicas de aprendizaje de máquina.

5.2. Objetivos Específicos:

- Realizar la selección de una técnica de aprendizaje de máquina, para la sintonización de los parámetros de un controlador de posición angular, aplicado a un péndulo invertido, basado en la revisión del estado del arte.
- Establecer un modelo métrico por el cual se pueda evaluar el desempeño del algoritmo propuesto del controlador sintonizado y del controlador clásico, comparando sobrepaso máximo, tiempo de establecimiento y error de estado estacionario de la respuesta de la salida del sistema.
- Validar el desempeño del sistema de control alrededor de una referencia de posición angular asignada, por medio de una comparación con un método de control clásico ajustado previamente y simulado en la plataforma MATLAB.
- Comparar los resultados obtenidos del controlador sin sintonizar y el controlador sintonizado en una planta Quanser.

Capítulo 6

Estado del Arte

6.1. Breve historia del control de los péndulos invertidos:

La teoría de control la cual se aplica a péndulos invertidos se puede encontrar desde la década de 1950, específicamente la versión de control del carro en movimiento, en donde se utilizó para enseñar la teoría lineal de control por realimentación de variables de estado, a partir de un punto de equilibrio para estabilizar un sistema en lazo abierto [23]. En 1960 fue descrita la primera solución a este problema del péndulo invertido, por J. K. Roberge en su tesis “The Mechanical Seal”; En la cual presenta un diseño de un servomecanismo que busca equilibrar una escoba. El autor utiliza técnicas analíticas que determinan la compensación necesaria para mantener una escoba en una posición vertical invertida [24]. Después en 1966, J. F. Schaefer y R. H. Cannon aplicaron otro controlador al mismo sistema del péndulo invertido en su publicación “On the control of unstable mechanical systems” [25]. Desde este punto el péndulo invertido ha sido considerado como ejemplo en varios libros en donde se estudia y se analiza sus comportamiento tanto lineales como no lineales [26]. También el péndulo invertido ha sido objeto de estudio para el diseño de sistemas de control lineal con optimización [27], y para sistemas de control no lineales que presenten inestabilidades [28].

Acorde con el anterior párrafo se puede destacar que el péndulo invertido resulta ser un sistema no lineal típico el cual termina siendo un objeto de estudio desde la década de los 50, en donde a partir de un punto de equilibrio inicial estable, el péndulo podía acomodar su posición vertical invertida a partir de controladores lineales, los cuales ejercían la acción de control en actuadores; siendo en este caso a un sistema mecánico vehicular en donde a partir de su desplazamiento ajustaba la escoba a una posición completamente vertical. Un problema presente en esa época en la aplicación de estos controladores al péndulo invertido, es que

dependían de una condición inicial o punto de equilibrio para su correcto funcionamiento; es por esto que a medida de los años se han creado nuevas teorías de control aplicable a la mecánica del péndulo invertido, en donde se da resolución a los problemas que no tenían la capacidad de resolver los controladores de los años 50-60.

6.2. Modelado y control

Como se presentó en la anterior sección el principal problema que tiene el sistema del péndulo invertido es su alta no linealidad y la inestabilidad lo que dificulta el control. Por esto se han propuesto diferentes modelos lineales los cuales dependen de una intervención matemática compleja en donde hay diferentes técnicas de linealización, como es a partir Euler-Lagrange. Esta técnica utiliza las energías cinéticas y potenciales para formular ecuaciones de movimiento no lineales del sistema, luego a partir de estas ecuaciones se ajusta un punto de equilibrio el cual se iguala cada ecuación y se aplica una derivada con respecto a la variable característica del sistema; en este caso del péndulo, es el ángulo de giro del brazo y el ángulo de giro del motor [7, 29]. Estas técnicas permiten desarrollar sistemas de control con métodos clásicos lineales y de control óptimo. Entre estos, se encuentra el método de control por PID el cual a partir de distintos coeficientes proporcional, integral y derivativo se obtiene una respuesta de control [9, 30, 31], en la cual se puede ajustar las propiedades de la respuesta a partir de un método de sintonización como es el método de Ziegler Nichols [11]. Se pueden encontrar otros métodos lineales más complejos como reguladores lineales cuadráticos (LQR) [9, 29, 30, 32] o por realimentación de variables de estados [10] e implementación de métodos basado en el control de la energía [33, 34], los cuales han conseguido adoptar capacidades básicas para el control del péndulo invertido, incluso en ambientes sometidos a perturbaciones de velocidad o posición.

Sin embargo, estos enfoques lineales presentan problemas en el momento en que la acción de control se aleja de su punto de equilibrio. Para esto existen distintas soluciones con enfoques con control robusto y no lineal, como los basados en modelos no lineales para un control predictivo [35], control por realimentación de variables de estado adaptativo basado en múltiples variables [36, 37], control robusto a partir de desigualdades matriciales o LMI [12], control de tiempo óptimo [38], entre otros [38]. Estos tipos de desarrollos tienen cierta robustez al ser capaces de operar con varios puntos de operación, los cuales consideran cambios en los parámetros de la planta y los hacen ser más fuertes frente a las condiciones

exteriores a la cual son sometidos, lo que les permiten ser más fiables ante perturbaciones. A pesar de esto, los diseños de los mismo requieren de tener un modelo matemático preciso con sistema de control de orden superior, lo cual acarrea un proceso matemático más complejo para el cálculo de los coeficientes.

6.3. Técnicas de Inteligencia Artificial:

Enfoques más recientes aplican una combinación de métodos clásicos de control con métodos de aprendizaje de máquina de manera que se pueda evitar una dependencia fuerte al modelo matemático, y más bien aplicar procesamiento de máquina para encontrar una solución a partir de diferentes metodologías, como es por medio de algoritmos de enjambre de partículas o (PSO) ,el cual dependiendo de una asignación de un parámetro para su maximización o minimización acorde al modelo matemático al cual se aplique, puede optimizar la respuesta de control de un PI vectorial [39], obteniendo un tiempo de estabilización de 40 segundos, después de un entrenamiento de 4 horas, por lo mismo tiene la capacidad de ser aplicado para la identificación del péndulo invertido [40] o la sintonización de otras plantas como un convertido Boost [40]. También con la implementación de lógica difusa, en donde las decisiones de control no se toman a partir de valores booleanos, si no que se evalúan en una función de pertenencia que tiene las variables en sus universos para así, ajustar mejor las decisiones del control aplicado [13, 14, 41, 42]. También por medio de la integración de agentes de aprendizaje por refuerzo los cuales parten de iteraciones y condiciones de soluciones para ajustar una salida de control [15, 16]. Otro algoritmo de búsqueda el cual tiene la capacidad de optimizar la respuesta por medio de técnicas meta heurísticas es el algoritmo Cuckoo (CS), este algoritmo busca los parámetros óptimos de K_p , K_i y K_d de un controlador PID con el objetivo de minimizar la función objetivo mientras se respetan un conjunto de restricciones como pueden ser el tiempo de subida, el sobre pico de la respuesta , el tiempo de estabilización y el error de estado estacionario [17]. También se puede aplicar algoritmos genéticos de búsqueda metaheurística para ajustar los parámetros a un controlador por inversión aplicado a la planta del péndulo invertido [43]. Otro método que integra sistema inmunes artificiales para sintonizaciones de controladores y que es referente importante en este proyecto, es un algoritmo el cual se comporta como respuesta de un sistema inmune adaptativo, el cual un organismo identifica y crea anticuerpos que van formando varias generaciones las cuales van mejorando sus aptitudes hasta un punto en que sintonizan los

parámetros del controlador, para una respuesta más eficiente de la planta [44, 45].

6.4. Aplicaciones de la técnicas de control en la industria:

Una aplicación frecuente de las técnicas de inteligencia artificial en péndulos invertidos se puede encontrar en el despegue y aterrizaje vertical de aeronaves (VTOL) en donde requieren de mantener una posición vertical basado en un modelo no lineal del péndulo, para esto se necesita de la implementación de controladores que tengan la capacidad de atenuar o eliminar el efecto de las perturbaciones como puede ser la implementación de un controlador por feedforward networks el cual trata de anticiparse a las perturbaciones que puedan afectar el modelo [20, 21, 46]. Otra aplicación frecuente del péndulo invertido se encuentra en el control del equilibrio de robots apoyados en dos ruedas, en los cuales va variando su posición angular vertical a medida en que el robot opera y se mueve; Así las ruedas pueden accionarse de tal manera que permanezca por debajo del centro de gravedad del robot [19, 47] .

6.5. Selección de técnica de aprendizaje de maquina

Para realizar la selección de la técnica de aprendizaje, primero se realiza una revisión a los documentos y artículos anexados previamente en las secciones anteriores. Para esto se realiza una tabla de jerarquía en donde se clasifican los artículos que utilizan el procesamiento de máquina para ajustar controladores de péndulos invertidos. A partir de la tabla 2 se hace una clasificación de los algoritmos y métodos que se utilizan para controlar el péndulo invertido, en función a la importancia que estos presentan para la aplicación de este proyecto.

Nivel	Importancia	Tipo	Descripción	Referencia
1	Alta	Bioinspirados	Algoritmos que utilizan técnicas heurísticas las cuales se obtienen por medio de teorías de la evolución. Muy usados para optimización.	[17, 39, 44, 48]
2	Media	Redes Neuronales	Modelo computacional inspirado en el funcionamiento de la neuronas. Muy utilizados para la clasificación y regresión de sistemas	[15, 16]
3	Baja	Lógica Difusa	Técnicas que permite manejar datos los cuales tienen transiciones suaves entre categorías, estando parcialmente en un grupo o en otro.	[13, 14, 41, 42]
4	Complementario	Ziegler Nichols	Método de sintonización clásico que permite definir las ganancias proporcional, integral y derivativa a partir de la respuesta en lazo cerrado	[11]

TABLA 2: Tabla de jerarquía para selección de técnica de aprendizaje de máquina.

Debido a que se busca una sintonización de parámetros de un controlador aplicando una técnica de aprendizaje de máquina, el rango mayor de búsqueda se tiene para algoritmo de naturaleza bioinspirada, los cuales integran diferentes metodologías adaptables para realizar sintonización de controladores a partir de teorías de la evolución. Posteriormente se procede a elegir el método de sintonización que se encuentra dentro del tipo de algoritmos bioinspirados, para esto se extrae los datos de desempeño de los algoritmos en función al modelo métrico a considerar en este proyecto los cuales son: Sobrepaso máximo, Tiempo de establecimiento y Error de estado estacionario. Para la selección también se considera las iteraciones o épocas que el algoritmo requiere para su convergencia.

Técnica	Mp(Deg)	Ts(seg)	Ess(Deg)	Iteraciones	Referencias
PSO	28.59	40	0	20	[39]
Cuckoo	5.50	0.4	0.0006	30	[17]
PSO	26.9	2.34	0.0099	145	[48]
Algoritmo Genético (GA)	32.8	2.83	0.0028	145	[48]
Algoritmo CLONAL	4.4354	2.17	0.0053	25	[44]

TABLA 3: Tabla de selección del algoritmo Bioinspirado para la sintonización de controladores

A partir de la tabla 3 se anexan los documentos los cuales utilizan algoritmos de procesamiento de máquina para la sintonización de controladores. Estos algoritmos son de naturaleza bioinspirada, ya que estos permiten una buena versatilidad para ser usados en casos

de optimización por su metodología heurística, proporcionando buenos resultados de desempeño, como se puede observar en la tabla 3. Lo que caracteriza principalmente a estos algoritmos es la función objetivo o de aptitud utilizada para poder ajustar las partículas en el caso del PSO, o en los algoritmos genéticos los anticuerpos asociados al antígeno en cuestión que se desea solucionar. De esta forma a medida que la función objetivo disminuya, se va ir mejorando el desempeño de las respuesta del controlador reduciendo el sobrepaso máximo, tiempo de establecimiento o error de estado estacionario. Para el desarrollo del proyecto se hace la selección del algoritmo de selección CLONAL, ya que comparando las métricas del sobrepaso máximo, tiempo de establecimiento y error de estado estacionario de los artículos anexados, y considerando el numero de iteraciones que utiliza conseguir estos resultados, representa la mejor opción para implementar. Otra razón para seleccionar este algoritmo en comparación a los otros, es la metodología presentada en el documento de selección CLONAL consultado [44], en donde utilizan la misma planta QUANSER para su implementación.

Capítulo 7

Marco Teórico

7.1. Modelamiento Péndulo Rotatorio Invertido

El Péndulo Rotatorio Invertido es un sistema el cual ha sido actualmente utilizado en muchos casos de estudio para la creación de controladores debido a sus características no lineales. El objetivo que se busca en un controlador del péndulo invertido, es que este mantenga su brazo de manera vertical, el cual representa 180 grados de traslación desde su punto de inicio. Para observar de mejor manera esto se utiliza la figura 1 a continuación.

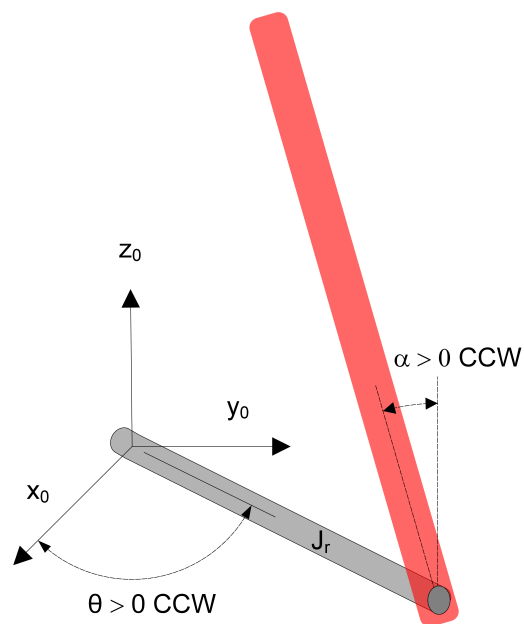


FIGURA 1: Esquemático del Péndulo Invertido [49]

Como se observa en el esquemático del péndulo invertido, este opera en función a dos variables las cuales son θ y α . En este proyecto se diseñó un sistema de control para ajustar la posición de α de manera vertical. Para esto se hace necesario conocer que parámetros influyen en la dinámica del sistema. Estos parámetros se anexan en la tabla 4 agregada a continuación.

Numero	Símbolo	Descripción	Valor
1	K_t	Contante del torque del motor	0.042 N-m/A
2	K_m	Constante trasera EMF	0.042 N-m/A
3	R_m	Resistencia de la armadura	8.4 Ω
4	J_r	Momento de inercia rotatorio del brazo	$m_r * L_r^2/12$
5	L_r	Longitud del brazo rotatorio	0.085m
6	m_r	Masa del brazo rotatorio	0.095kg
7	J_p	Momento de inercia del péndulo	$m * L_p^2/12$
8	L_p	Longitud del péndulo	0.129m
9	m_p	Masa del péndulo	0.024kg

TABLA 4: Parámetros físicos de la planta del péndulo rotatorio invertido

En primer lugar para poder implementar algún controlador que permita establecer un ángulo de referencia al péndulo invertido, es necesario aplicar un proceso de modelamiento de la planta; esto con el objetivo de tener una base de modelo lineal por el medio del cual se va a implementar el control base del sistema y el control con el algoritmo para la sintonización de los parámetros del controlador. Para lograr esto, se procede a utilizar las siguientes ecuaciones no lineales, las cuales representan la cinemática del péndulo invertido con el motor que tiene la planta y la energía potencial del sistema [50].

$$\begin{aligned}
 & (m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r) \ddot{\Theta} - (\frac{1}{2} m_p L_p L_r \cos(\alpha)) \ddot{\alpha} \\
 & + (\frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha)) \dot{\Theta} \dot{\alpha} + (\frac{1}{2} m_p L_p L_r \sin(\alpha)) \dot{\alpha}^2 = \tau - D_r \dot{\Theta}
 \end{aligned} \tag{7.1}$$

$$\frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\Theta} + (J_p + \frac{1}{2} m_p L_p^2) \ddot{\alpha} - \frac{1}{4} m_p L_p^2 \cos(\alpha) \cos(\alpha) \dot{\Theta}^2 + \frac{1}{2} m_p L_p g \sin(\alpha) = -D_p \dot{\alpha} \tag{7.2}$$

7.1.1. Linealización

A continuación se muestra un ejemplo de cómo linealizar una variable de dos funciones no lineales llamada $f(z)$. La variable z está definida como.

$$z^T = [z_1 z_2] \quad (7.3)$$

y $f(z)$ debe linealizarse alrededor del punto de operación.

$$z_0^T = [ab] \quad (7.4)$$

La función linealizada es.

$$f_{lin} = f(z_0) + \frac{\partial f(z)}{\partial z_1} \Big|_{z=z_0} (z_1 - a) + \frac{\partial f(z)}{\partial z_2} \Big|_{z=z_0} (z_2 - b) \quad (7.5)$$

Se asigna la matriz de z .

$$Z^T = [\Theta, \alpha, \dot{\Theta}, \dot{\alpha}, \ddot{\Theta}, \ddot{\alpha}] \quad (7.6)$$

Donde se define el valor inicial del $z_0^T = [000000]$, para la siguiente ecuación no lineal de movimiento.

$$\begin{aligned} f(z) = & (m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r) \ddot{\Theta} - (\frac{1}{2} m_p L_p L_r \cos(\alpha)) \ddot{\alpha} \\ & + (\frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha)) \dot{\Theta} \dot{\alpha} + (\frac{1}{2} m_p L_p L_r \sin(\alpha)) \dot{\alpha}^2 = \tau - D_r \dot{\Theta} \end{aligned} \quad (7.7)$$

Luego desde la anterior ecuación se aplica la siguiente evaluación.

$$f_{lin} = f(z_0) + \frac{\partial f(z)}{\partial \ddot{\Theta}} \Big|_{z=z_0} \ddot{\Theta} + \frac{\partial f(z)}{\partial \ddot{\alpha}} \Big|_{z=z_0} \ddot{\alpha} + \frac{\partial f(z)}{\partial \dot{\Theta}} \Big|_{z=z_0} \dot{\Theta} + \frac{\partial f(z)}{\partial \dot{\alpha}} \Big|_{z=z_0} \dot{\alpha} + \frac{\partial f(z)}{\partial \theta} \Big|_{z=z_0} \theta + \frac{\partial f(z)}{\partial \alpha} \Big|_{z=z_0} \alpha \quad (7.8)$$

Se linealiza $F(z)$ con respecto a $\ddot{\Theta}$

$$\frac{\partial f(z)}{\partial \ddot{\Theta}}|_{z=z_0} = m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(0) + J_r = m_p L_r^2 + J_r \quad (7.9)$$

Luego se linealiza $F(z)$ con respecto a $\ddot{\alpha}$

$$\frac{\partial f(z)}{\partial \ddot{\alpha}}|_{z=z_0} = -\frac{1}{2} m_p L_p L_r \cos(0) = -\frac{1}{2} m_p L_p L_r \quad (7.10)$$

Después se procede a evaluar las otras variables de estado en cero y se obtiene lo siguiente.

$$\frac{\partial f(z)}{\partial \dot{\alpha}}|_{z=z_0} = 0 \quad (7.11)$$

$$\frac{\partial f(z)}{\partial \dot{\theta}}|_{z=z_0} = 0 \quad (7.12)$$

$$\frac{\partial f(z)}{\partial \alpha}|_{z=z_0} = 0 \quad (7.13)$$

$$\frac{\partial f(z)}{\partial \theta}|_{z=z_0} = 0 \quad (7.14)$$

Reemplazando los valores anteriormente calculados se obtiene la siguiente expresión de la primera ecuación linealizada.

$$(m_p L_r^2 + J_r) \ddot{\theta} - \frac{1}{2} m_p L_p L_r \ddot{\alpha} = \tau - B_r \dot{\Theta} \quad (7.15)$$

Ahora se procede a linealizar la segunda ecuación no lineal del péndulo rotatorio invertido asignando las mismas condiciones iniciales.

$$f(z) = \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\Theta} + (J_p + \frac{1}{2} m_p L_p^2) \ddot{\alpha} - \frac{1}{4} m_p L_p^2 \cos(\alpha) \cos(\alpha) \dot{\Theta}^2 + \frac{1}{2} m_p L_p g \sin(\alpha) = -D_p \dot{\alpha} \quad (7.16)$$

Luego desde la anterior ecuación se aplica la siguiente evaluación.

$$f_{lin} = f(z_0) + \frac{\partial f(z)}{\partial \ddot{\Theta}}|_{z=z_0} \ddot{\Theta} + \frac{\partial f(z)}{\partial \ddot{\alpha}}|_{z=z_0} \ddot{\alpha} + \frac{\partial f(z)}{\partial \dot{\theta}}|_{z=z_0} \dot{\theta} + \frac{\partial f(z)}{\partial \dot{\alpha}}|_{z=z_0} \dot{\alpha} + \frac{\partial f(z)}{\partial \theta}|_{z=z_0} \theta + \frac{\partial f(z)}{\partial \alpha}|_{z=z_0} \alpha \quad (7.17)$$

A partir de la segunda ecuación no lineal del sistema del péndulo invertido se procede a calcular las derivadas parciales con respecto a cada variable de estado.

$$\frac{\partial f(z)}{\partial \ddot{\Theta}}|_{z=z_0} = \frac{1}{2} m_p L_p L_r \ddot{\alpha} \quad (7.18)$$

$$\frac{\partial f(z)}{\partial \ddot{\alpha}}|_{z=z_0} = J_p + \frac{1}{4} m_p L_p^2 \quad (7.19)$$

$$\frac{\partial f(z)}{\partial \alpha}|_{z=z_0} = -\frac{1}{2} m_p L_p g \quad (7.20)$$

Los otros valores de las derivadas de las variables de estado tienen un valor de $f(z_0) = 0$. Al evaluar se obtiene la siguiente expresión linealizada de la segunda ecuación no lineal.

$$\frac{1}{2} m_p L_p L_r \ddot{\Theta} + (J_p + \frac{1}{4} m_p L_p^2) \ddot{\alpha} + \frac{1}{2} m_p L_p g \alpha = -B_p \dot{\alpha} \quad (7.21)$$

Ahora se procede a dejar las ecuaciones (7.15) y (7.21) de movimiento lineales del péndulo invertido de forma matricial. Para esto se considera la siguiente expresión.

$$\begin{bmatrix} m_p L_r^2 + J_r & -\frac{1}{2} m_p L_p L_r \\ \frac{1}{2} m_p L_p L_r & J_p + \frac{1}{4} m_p L_p^2 \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} D_r + k_t \frac{k_m}{R} & 0 \\ 0 & D_p \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} m_p L_p g \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \quad (7.22)$$

Como se observa en la ecuación (7.22), se organizan los parámetros en relación al orden que tiene las variables de estado. Después se organizan las expresiones en una suma matricial en donde se tiene organizado los términos según el orden que tenga alpha y theta. Ya con esto, se procede a dejar la expresión anterior de la siguiente forma de matrices de variable de estado.

$$\dot{x} = Ax + Bu \quad (7.23)$$

Para esto se considera la siguiente forma donde se tienen los parámetros físicos que tiene el péndulo invertido en su dinámica.

$$\dot{x} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}G \end{bmatrix} x + \begin{bmatrix} 0 \\ -M^{-1}F \end{bmatrix} V_m \quad (7.24)$$

Ya definida la estructura matricial del modelo en variables de estado, se procede a asignar los parámetros del péndulo invertido que contienen M,K,G y F.

$$M = \begin{bmatrix} m_p L_r^2 + J_r & -\frac{1}{2} m_p L_p L_r \\ \frac{1}{2} m_p L_p L_r & J_p + \frac{1}{4} m_p L_p^2 \end{bmatrix} \quad (7.25)$$

$$G = \begin{bmatrix} D_r + k_t \frac{k_m}{R} & 0 \\ 0 & D_p \end{bmatrix} \quad (7.26)$$

$$K = \begin{bmatrix} 0 \\ \frac{1}{2} m_p L_p g \end{bmatrix} \quad (7.27)$$

$$F = \begin{bmatrix} \frac{k_t}{R} \\ 0 \end{bmatrix} \quad (7.28)$$

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Theta \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \Theta \\ \alpha \end{bmatrix} V_m \quad (7.29)$$

Como se tienen dos salidas en la matriz (7.29), para el control del péndulo invertido se utiliza únicamente la salida de α . Tomando como base el modelo de variables de estado de la ecuación (7.24), el cual representa el modelo linealizado de la planta del péndulo rotatorio invertido sobre el punto de trabajo [0,0,0,0], se asignan los valores de las constantes que influyen dentro del modelo las cuales buscan representar lo más aproximadamente el modelo real de péndulo invertido y que se pueden ver en la tabla 4. A partir de tabla 4, se hace el reemplazo de los parámetros dentro de las ecuaciones (7.24) hasta la (7.28) y se obtiene el siguiente modelo de variables de estado a utilizar para el desarrollo del control.

$$\begin{bmatrix} \ddot{\Theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -17,85 & -2,0055 & -0,2942 \\ 0 & -54,89 & 1,12 & -0,9046 \end{bmatrix} \begin{bmatrix} \Theta \\ \alpha \\ \dot{\Theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 10,44 \\ -5,88 \end{bmatrix} V_m \quad (7.30)$$

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Theta \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \Theta \\ \alpha \end{bmatrix} V_m \quad (7.31)$$

7.2. Sistemas Bio-inspirados

Los sistemas Bio-inspirados o sistemas que utilizan técnicas heurísticas con inspiración biológica son procedimientos que buscan, aprenden y optimizan modelos los cuales se obtienen por medio de teorías de evolución basadas en el esquema del Neo-Darwinismo que se aplica sobre el contexto de poblaciones. El Neo-Darwinismo se define como el conjunto de teorías que explican los procesos de reproducción, mutación, selección y competencia que tienen las especies en su proceso de evolución [51]. Cada algoritmo bio-inspirado tiene como característica principal utilizar una población, la cual contiene un conjunto de soluciones posibles al problema que se desee solucionar, a partir de la generación aleatoria de poblaciones y al sometimiento de cada población a un proceso iterativo en donde en función de un algoritmo, se busca la mejor solución posible bajo ciertas características que se ajusten de mayor manera al resultado deseado .

7.2.1. Algoritmos Evolutivos

Los algoritmos evolutivos hacen parte de las ramas que conforman el área de la inteligencia artificial la cual es la encargada de hacer el diseño de entidades capaces de resolver problemas basándose principalmente de la inteligencia humana, y en donde utilizando recursos computacionales junto con lógica programada que permiten obtener soluciones a problemas complejos que la mente humana le costaría resolver. Dentro de los algoritmos de computación evolutiva se pueden encontrar tres diferentes corrientes de investigación como son: programación evolutiva, estrategias evolutivas y algoritmos genéticos. Las anteriores corrientes mencionadas fueron principalmente impulsadas dentro de la década de los años 60 a 70 de las manos de científicos como Lawrence J.Fogel, John H. Holland y Ingo Rechenberg

[52, 53].

La principal ventaja que tienen estos algoritmos evolutivos los cuales se utilizan para dar solución a problemas es que no necesitan de ninguna especificación de un entorno lo que permite ser aplicados en áreas como la robótica, física, química, marketing e inclusive en campos como el arte. Debido a la complejidad computacional que tienen estos algoritmos es que se evita su uso en aplicaciones de producción. La razón de este alto procesamiento computacional, se debe a la evaluación de la función de costo que utiliza el algoritmo, la cual se encarga de optimizar la solución obtenida asignando un alcance máximo o mínimo a los parámetros de salida que el algoritmo calcula para su resultado.

Un algoritmo se puede entender como un conjunto de instrucciones ordenadas y definidas para la descripción de un proceso, el cual se debe de seguir para así lograr llegar a una solución de un problema específico [52]. El nombre de algoritmos evolutivos vienen debido a postulados publicados en los temas de evolución biológica y toma como base las ideas de evolución propuestas por Charles Darwin. En base, el procedimiento que involucra estos algoritmos consiste en hacer evolucionar una población la cual mediante la aplicación de una serie de operadores que se repiten varias veces, logran obtener una solución óptima del problema planteado. Cada población se compone de varios candidatos o individuos, de los cuales, mediante valores dados por una función de costo, se determina la calidad de cada solución.

Como se puede observar en la figura 2, un algoritmo evolutivo inicia con una población inicial P , la cual representa los datos del problema del que se parte. Esta población se compone principalmente de varios individuos los cuales son candidatos que forman parte de una solución final. Posteriormente, los individuos ingresan a un bloque de función de costo el cual se encarga de hacer la optimización de los individuos. Cada individuo en la población P tiene un valor característico dentro de la función de costo y mediante el análisis de este valor se puede determinar cuando un candidato es apto o no para formar parte de la solución. La operación anteriormente mencionada se hace en el bloque selección, en donde sus resultados pasan a un consecuente proceso PI , el cual consigue obtener a los candidatos mas propensos a llegar a la solución deseada. Por ultimo se aplica un proceso que varia con respecto al tipo de algoritmo evolutivo que se aplique, en donde se implementan una serie de mecanismos de variación, como la clonación, mutación o cruce, entre otros. Ya que se haya hecho la modificación de la población mediante los mecanismos que el tipo de algoritmo aplique, se

vuelve a iniciar la cadena de procesos hasta que se cumpla la función de parada, la cual puede ser un valor determinado o un numero de iteraciones del proceso en especifico [52].

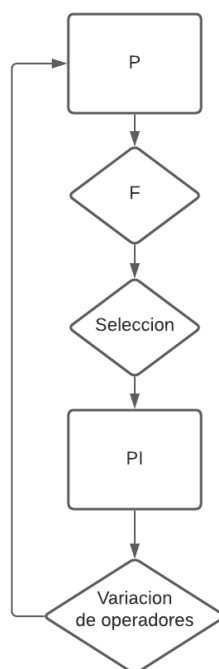


FIGURA 2: Diagrama de bloques del algoritmo evolutivo [Autoría propia].

En este documento se aplica un algoritmo el cual se basa en la teoría de la selección clonal presente en los sistemas inmunológicos de los seres vivos, mas específicamente en los mamíferos.

7.2.2. Teoría de selección clonal

El sistema inmunológico adaptativo de los mamíferos ha ido evolucionando hasta un punto que brinda una respuesta inmunitaria mayor debido principalmente a la memoria inmunológica que estos poseen. El sistema inmunitario que tiene la capacidad de adaptarse, cuenta con dos tipo de células B y T llamadas linfocitos la cuales realizan un proceso complejo. Primeramente se tiene los procesos de diferenciación y proliferación de elementos celulares que son capaces de reconocer los cuerpos invasivos o antígeno para así obtener una selección de estas. Posteriormente, se hace un proceso de maduración en varios patrones de los anticuerpos, donde se logran crear nuevos cambios de naturaleza genética aleatoria que se responsabilizan de tratar con el cuerpo antígeno identificado. Por ultimo, se hace una selección de los cuerpos celulares B en donde células con mayor afinidad con respecto al

cuerpo invasivo, entrarán en un conjunto de candidatos de memoria, y las que presenten menor afinidad serán ignoradas y eliminadas. Una respuesta inmune tiene tres tipos: Respuesta de tipo primaria, respuesta de tipo secundaria y reacción inmunológica cruzada las cuales ocurren de forma secuencial[54].

Al principio el antígeno empieza con un bajo número de células B que tienen baja afinidad pero posteriormente se producirán un tipo de anticuerpo con una afinidad diferente relacionada al antígeno. Es por esto, que el primer paso que realiza el sistema inmunológico es reconocer un patógeno infeccioso que no es propio del organismo. Debido a esto una molécula no perteneciente del organismo puede ser identificada como un antígeno por el sistema inmunológico. Este procedimiento se consigue por las células T las cuales pueden reconocer un elemento no propio del organismo, como un patógeno. Por ultimo, cada patógeno se recuerda por un elemento no propio del organismo que desembarca una formación de anticuerpos que son capaces de tratar con este elemento invasivo como se muestra en la siguiente figura 3.

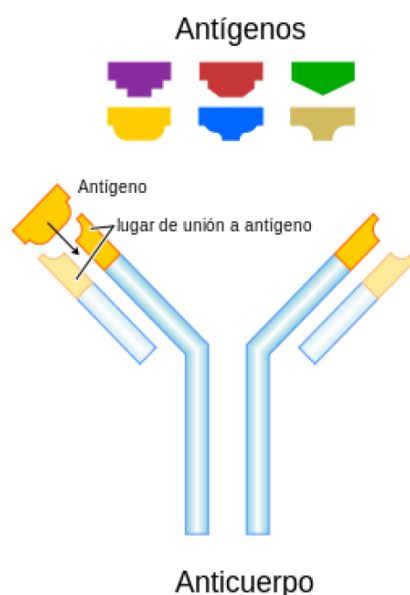


FIGURA 3: Identificación de los antígenos por medio de anticuerpos [52]

Los antígenos se generan por medio de las células B, donde individualmente cada receptor de los antígenos se contiene dentro de la célula. Cada linfocito B logra identificar a un ente invasivo cuando los anticuerpos se unen a otros antígenos no propios. Ya cuando se obtiene el antígeno ligado al anticuerpo y a su vez está realizando un ataque a los patógenos, se comienza a dividir las células T y B, creando miles de copias de ellas denominándose

respuesta primaria. Algunas de las copias creadas logran ser candidatos a un conjunto de anticuerpos los cuales se reservan en memoria del mamífero, debido a que representan una mejor respuesta al patógeno y permite al sistema inmunológico hacer memoria al momento de que se vuelva a encontrar el mismo patógeno en un futuro en el sistema. Un ejemplo de este proceso se puede ver en la figura 4, en donde el conjunto de anticuerpos de memoria establece respuestas únicas para infecciones sucesivas con el objetivo de eliminar a un patógeno infeccioso. Es así que la efectividad que va a tener la respuesta inmunológica, dependerá de si los anticuerpos que están en memoria son capaces de combatir con los antígenos desde la primera infección y en función a esto puedan crear nuevos anticuerpos que tengan una mayor afinidad al momento de volver a combatir con el cuerpo invasivo del sistema. La respuesta que prosigue a la primaria se le conoce como respuesta secundaria, la cual tiene un periodo de tiempo más corto que la primera respuesta, pero una concentración mayor de anticuerpos con alta afinidad y velocidad. Posterior a la respuesta secundaria, el sistema de defensa inmunológico se va ir preparando a futuros ataques externos con similares características.

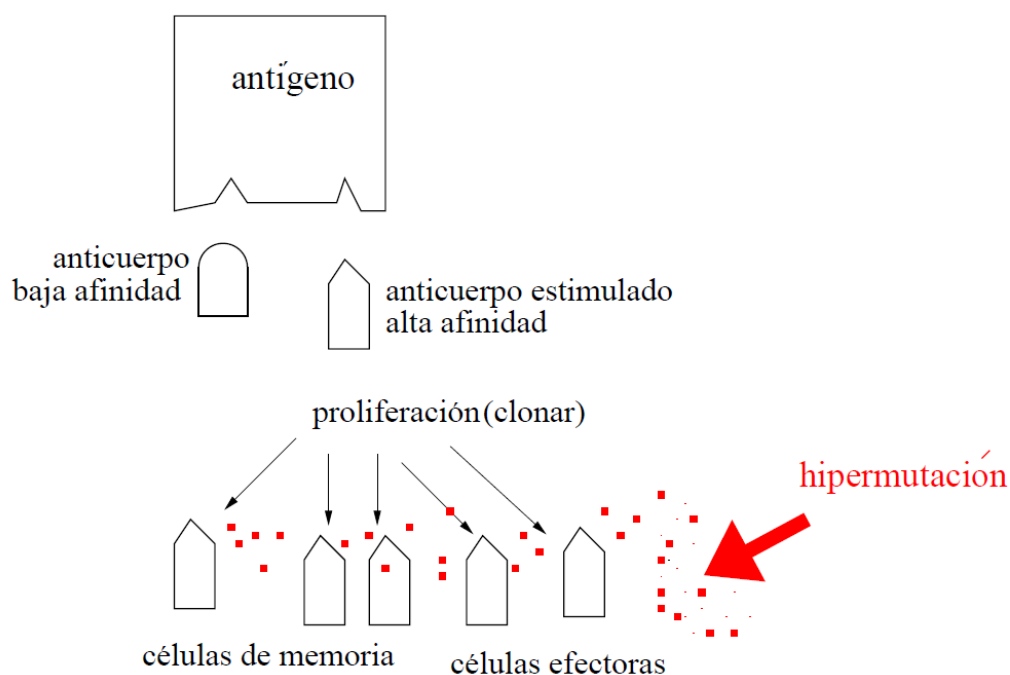


FIGURA 4: Principio de selección clonal [55]

La selección clonal nace por una estimulación la cual proviene de un patógeno en el sistema al momento de que aparece un linfocito. Esto hace que se cree un proceso de creación de agentes defensivos o anticuerpos los cuales presentan diferentes grados de afinidad con el antígeno, o agente invasor para hacer cierta unión del paratope con el epitope. Al momento de hacerse

esta unión se puede generar tres casos: Se realiza una unión perfecta, lo que significa que el anticuerpo consiguió identificar al antígeno; La detección es parcial, lo que significa se tiene solamente una parte de la información del antígeno y se obtiene un indicativo de que tan cerca se estuvo de identificar el antígeno por medio de la función de aptitud; Por último no se encuentra similitudes o compatibilidad por tener una afinidad muy baja o casi nula. En el momento de que ocurra el caso de una detección parcial del antígeno, el sistema artificial debe crear un proceso de clonación aplicando mutación y selección de los candidatos que más elevado tengan la función de aptitud para así guardar los mejores anticuerpos en memoria y volver a repetir el ciclo de incremento de la afinidad. Si al final el sistema inmune consigue encontrar los mejores grupos de anticuerpos del proceso los otros candidatos con menor afinidad se eliminan para aplicar una autorregulación en el sistema inmune [52]

7.2.3. Algoritmo de selección clonal

El algoritmo de selección clonal se basa enteramente del principio del sistema inmune el cual se va adaptando a los patrones que amenazan los cuerpos de mamíferos. Para este reconocimiento de patrones el algoritmo emplea la analogía del antígeno como un problema que se tiene que resolver y los anticuerpos representan las posibles soluciones que respondan al antigénico. Para esto se hace una secuencia genética la cual se va ir iterando a medida de que se ejecute el algoritmo. A continuación se muestra los pasos que componen una iteración del algoritmo de selección clonal

7.2.3.1. Paso 1: Inicialización

En esta etapa inicial del algoritmo se genera una población inicial (anticuerpos) de manera aleatoria o se asigna un espacio de búsqueda específico dando valores a los anticuerpos iniciales.

7.2.3.2. Paso 2: Selección

Posteriormente a tener una población inicial de candidatos o anticuerpos para resolver el problema, se hace una selección de anticuerpos para cada antígeno escogiendo a los

anticuerpos que tengan mayor afinidad con respecto al antígeno. Para este caso las soluciones representarían a los anticuerpos y el antígeno como la función objetivo que se busca resolver.

7.2.3.3. Paso 3: Clonación y Variación Genética

Después de hacer una selección de los anticuerpos se aplica una clonación en forma directamente proporcional a la afinidad. Así los anticuerpos los cuales tengan mas afinidad mutaran menos en relación a los anticuerpos con menos afinidad. Es decir, que la mutación para los anticuerpos con menos afinidad va a ser mayor para así hacer mas probable una convergencia a la solución buscada.

7.2.3.4. Paso 4: Evaluación de la afinidad

Se evalúa cuanta afinidad existe entre cada anticuerpo mutado en relación al antígeno para asignar un valor de afinidad a cada posible solución.

7.2.3.5. Paso 5: Autorregulación

En este paso se descartan los anticuerpos los cuales no tuvieron una buena afinidad con relación al antígeno.

7.2.3.6. Paso 6: Repetición en ciclo

En este último paso el algoritmo regresa al paso 2 de selección haciendo un bucle el cual va ir repitiéndose hasta que se llegue a un criterio de convergencia.

Siguiendo los anteriores pasos se logra la maduración, adaptación y aprendizaje con la continuación del proceso de evaluación y mutación que los anticuerpos con mejor desempeño consigan. Estos anticuerpos se almacenan en la memoria la cual reconocerá a antígenos futuros que se presenten dentro del sistema artificial inmune. Debido a la facilidad que tienen

estos algoritmos para el reconocimiento de patrones los cuales obtienen información por medio de características relevantes, es que se pueden adaptar para problemas de optimización en donde actúa como una técnica heurística la cual permite buscar una solución que represente un resultado óptimo para una función objetivo determinada. Un diagrama de flujo que se utiliza para aplicar un proceso de optimización implementando el algoritmo de selección clonal se ilustra con la siguiente figura 5 .

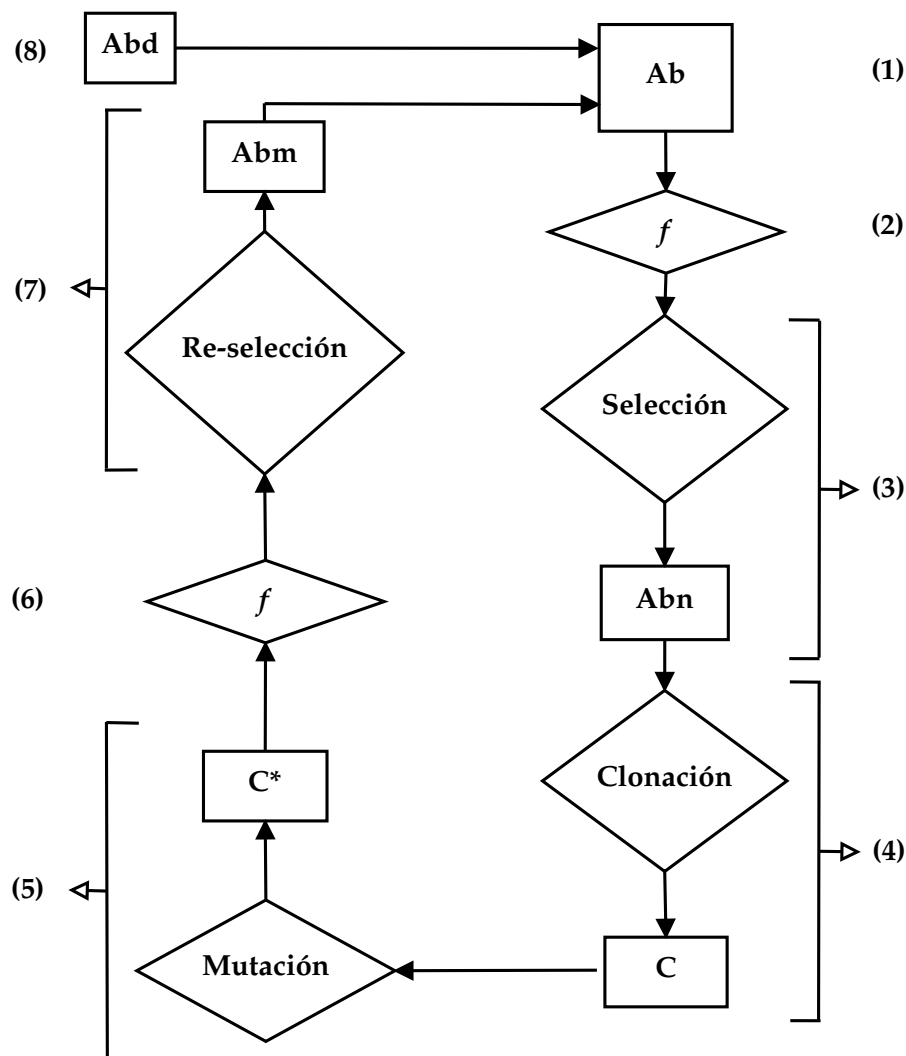


FIGURA 5: Diagrama de flujo que se utiliza para aplicar un proceso de optimización CLONAL [Autoría propia]

Como se observa en la anterior figura 5 el algoritmo de selección clonal se ajusta a un diagrama de optimización en donde los antígenos hacen referencia a una función objetivo mediante el cual se busca un máximo o mínimo que represente un valor óptimo. La función objetivo puede contener información del problema que se busca solucionar de la mejor forma posible, en donde actúa en su proceso de solución los anticuerpos que se generen de forma

inicial, y que en cada iteración vayan ajustando sus valores hasta llegar a una respuesta que se adapte de mejor forma al antígeno. A continuación se detalla que hace cada numeral dentro del diagrama de secuencia para un problema de optimización.

1. Se define la población inicial de anticuerpos Ab con un tamaño N asignado.
2. Se calcula un valor que representa la función objetivo para cada anticuerpo presente y se guarda en un vector f .
3. Se hace la selección de los anticuerpos Abn con tamaño N los cuales presenten mayor afinidad.

$$Abn = \min(f(N)) \quad (7.32)$$

A partir de la anterior ecuación se hace la selección de los anticuerpos con mayor afinidad; es decir, con menor valor de función de aptitud u objetivo. Se guarda en un espacio de memoria Abn

4. Se realiza la clonación de los elementos pertenecientes al grupo de los anticuerpos de alta afinidad Abn . La clonación se realiza de manera proporcional a la afinidad, esto significa que los elementos con mayor afinidad generarán más clones que los anticuerpos con menor afinidad. También se puede clonar los anticuerpos en la misma medida. Al final se forma un conjunto de clones C . Una función matemática que se aplique en este paso se asocia de la siguiente forma.

$$C = -0,5 + \text{rand}(N) \quad (7.33)$$

En la anterior ecuación para la clonación se tiene una función interna rand , la cual genera valores aleatorios del vector entrada de N . Se guarda en un espacio de memoria C

5. Los elementos clonados se mutan para aumentar la diversidad. Para esto el proceso de mutación deber realizarse de manera inversamente proporcional a la afinidad que tenga el anticuerpo. Esto quiere decir que con mayor afinidad tenga un anticuerpo la mutación será menor, y los anticuerpos que tengan menor afinidad mutarán de mayor medida para lograr que la población converga y se obtenga el conjunto C^* de clones mutados.

Una función matemática que se aplique en este paso se asocia de la siguiente forma.

$$C^* = -C * (\text{Porcentaje de mutación}) * Abn \quad (7.34)$$

Como se observa en la anterior ecuación, los elementos clonado C se multiplican con un factor de mutación y la población inicial para hacer la variación de los anticuerpos. Se guarda en un espacio de memoria C^*

6. Se calcula de nuevo el valor de la función objetivo f con los elementos clonados y mutados.
7. Se hace la selección de los anticuerpos que optimizan la función objetivo para formar parte de una memoria. También se puede seleccionar el elemento que tenga una mayor afinidad para ser un candidato a entrar en el conjunto de elementos de memoria Abm .

$$Abm = \min(f(C^*)) \quad (7.35)$$

Como se observa en la anterior ecuación, se vuelve a tomar el valor mínimo asociado al nuevo vector de la función de aptitud y se guarda en un espacio de memoria Abm .

8. Por último el resto de anticuerpos con baja afinidad se reemplazarán por otros nuevos candidatos Abd .

Debido a la versatilidad que tiene el algoritmo de selección clonal al momento de aplicar una optimización de una función de afinidad u objetivo, la cual representa un valor característico que contiene datos de referencia, sobre que tanto se aproxima una solución al problema planteado, o en este caso al anticuerpo que se busca adaptar el sistema inmune. De esta forma es posible asignar aplicaciones con controladores en los cuales, los anticuerpos podrían representar las ganancias que requiere el controlador para mantener su referencia. El anticuerpo podría representar los valores de desempeño que tenga la respuesta que se desee mejorar, a partir de la sintonización de los parámetros. La función de aptitud podría representar un valor característico general que considere las imágenes de desempeño que el diseñador del controlador busque mejorar. El algoritmo se encargaría de generar y clonar candidatos que representen un mejor desempeño con ganancias que varíen de mayor a menor medida, en función de la mutación asignada por el programador.

Capítulo 8

Desarrollo Metodológico

8.0.1. Modelo métrico a utilizar

Como primer paso se asigna un modelo métrico por medio el cual se pueda evaluar el desempeño que tiene el algoritmo de selección CLONAL. Para esto se extraen los valores resultantes de la respuesta del sistema con la siguiente figura 6.

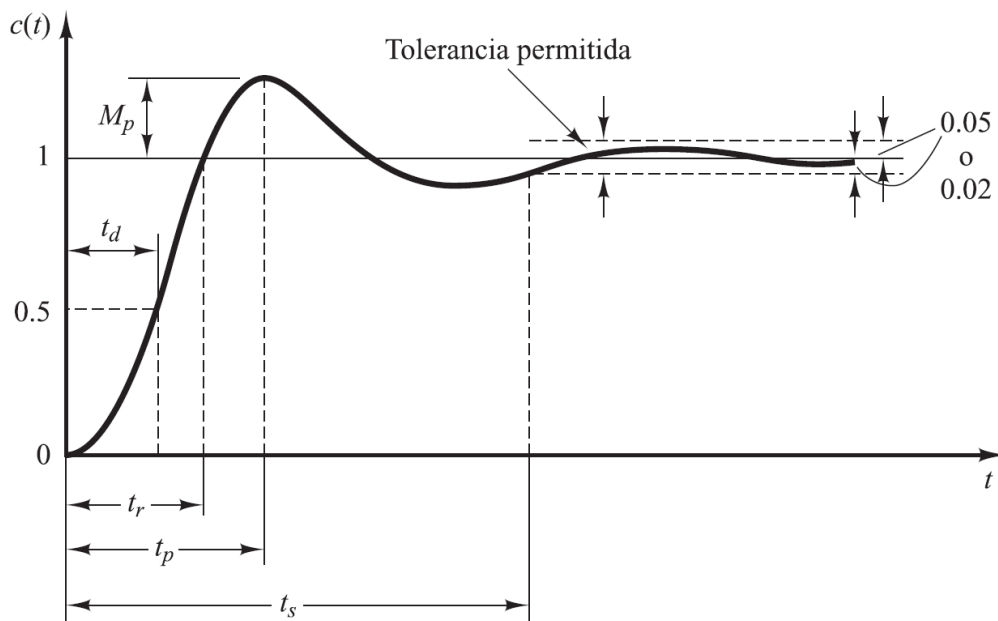


FIGURA 6: Curva de respuesta a escalón unitario de un sistema [56]

Sobrepaso Máximo (M_p): Es el valor pico máximo de la curva de respuesta, medido a partir de la unidad. Si el valor final en estado estacionario de la respuesta es diferente a la unidad, es frecuente utilizar el porcentaje de sobrepaso máximo el cual se define como la resta del sobrepaso máximo entre el valor de estado estacionario dividido por el valor de estado estacionario[56]. El modelo en ecuación que representa el sobrepaso máximo se puede observar con la siguiente expresión.

$$M_P = \frac{C(t_p) - C(\infty)}{C(\infty)} \quad (8.1)$$

Tiempo de estabilización (t_s): El tiempo de estabilización es el tiempo que se requiere para que la curva de respuesta alcance un rango alrededor del valor final del tamaño especificado por el porcentaje absoluto del valor final (por lo general, de 2 % o 5 %), esto se puede observar en la figura 6. El tiempo de estabilización se relaciona con la mayor constante de tiempo del sistema de control. Los objetivos del diseño del sistema en cuestión determinan qué criterio de error en porcentaje utilizar[56].

Error de estado estacionario (ess): El error de estado estacionario es el error que existe de la respuesta obtenida a la salida del sistema de control y la señal de referencia a la cual el sistema de control trata de aproximarse. En la figura 6, se tiene una referencia unitaria y una respuesta de estado estable del sistema muy cercana al valor unitario, dando como resultado un error pequeño. Este error se considera generalmente en un porcentaje con respecto a la unidad [56].

Función objetivo o aptitud: Esta métrica a considerar es una función representativa dentro de los algoritmos bioinspirados, la cual permite optimizar una expresión que contenga varias métricas, ya sea maximizando o minimizando los resultados. De esta forma, se le asocia un valor representativo a cada posible solución, para conocer que tanto se aproxima a resolver el problema que plantea la función.

A partir de las anteriores métricas se realiza el proceso de sintonización de los controladores utilizados en este proyecto.

8.0.2. Metodología

En este capítulo se abordan todos los procesos realizados durante el desarrollo del proyecto, en donde se define las diferentes etapas que conforman en primera medida la sección de

simulación, en donde se detallan las metodologías utilizadas para la asignación del entorno de simulación y el diseño del controlador clásico PID. Posteriormente dentro de la misma sección se realiza la programación y ajuste del algoritmo de selección clonal en donde se realiza la programación del algoritmo en función a la figura 5, para acomodarlo en relación al problema de sintonización del controlador PID. Por último en la sección de simulación se implementa el algoritmo de selección clonal con el controlador PID para realizar una sintonización de los parámetros K_p , K_i , K_d y P , los cuales inician con los valores iniciales del controlador PID desarrollado por el método de LGR dentro de la misma sección de simulación. El proceso metodológico de la simulación se contiene dentro de las etapas I,II y III de la figura 7.

Después de la sección de la simulación se realiza la sección de implementación física con la planta QUANSER. En esta sección se realiza el desarrollo para la implementación de una simulación en tiempo real en un entorno QUARC, el cual es eficiente de diseñar, desarrollar, implementar y validar aplicaciones en tiempo real en hardware utilizando Simulink. QUARC genera código en tiempo real directamente desde los controladores diseñados por Simulink y lo ejecuta en tiempo real en Windows, todo sin procesamiento de señal digital o sin escribir una sola línea de código. En este entorno de implementación se utiliza un controlador RVE (Realimentación de Variables de Estado), el cual funciona a partir de un punto de trabajo específico ajustado mediante la linealización de un modelo no lineal; Todo esto viene por defecto y lo proporciona QUANSER en su planta de péndulo invertido. Posteriormente dentro de la misma sección se realiza el ajuste de los valores iniciales del algoritmo de selección clonal previamente programado en la etapa II, los cuales ahora representan las ganancias de retroalimentación que venia por defecto en la QUANSER, para comenzar desde ese punto la búsqueda en la sintonización. Por ultimo en la sección de implementación, se realiza la evaluación de la sintonización del controlador RVE el cual ajusta las ganancias asociadas a cada variable de estado que hace parte del péndulo rotatorio invertido las cuales son: $[Gain_\alpha, Gain_\theta, Gain_{V_{el\alpha}}, Gain_{V_{el\theta}}]$. El proceso metodológico de la implementación en la QUANSER se contiene dentro de las etapas IV,V y VI de la figura 7.

Por ultimo, en la sección de análisis de resultados se realiza una discusión sobre el desempeño obtenido de ambos controladores en sus respectivos entornos, evaluando y comparando qué tanto mejoró el sobrepaso máximo, tiempo de establecimiento y error de estado estacionario después de haber aplicado el algoritmo de selección clonal y compararlo con los controladores bases de los cuales deriva la sintonización. Lo anteriormente mencionado se ilustra de mejor forma con la siguiente figura .

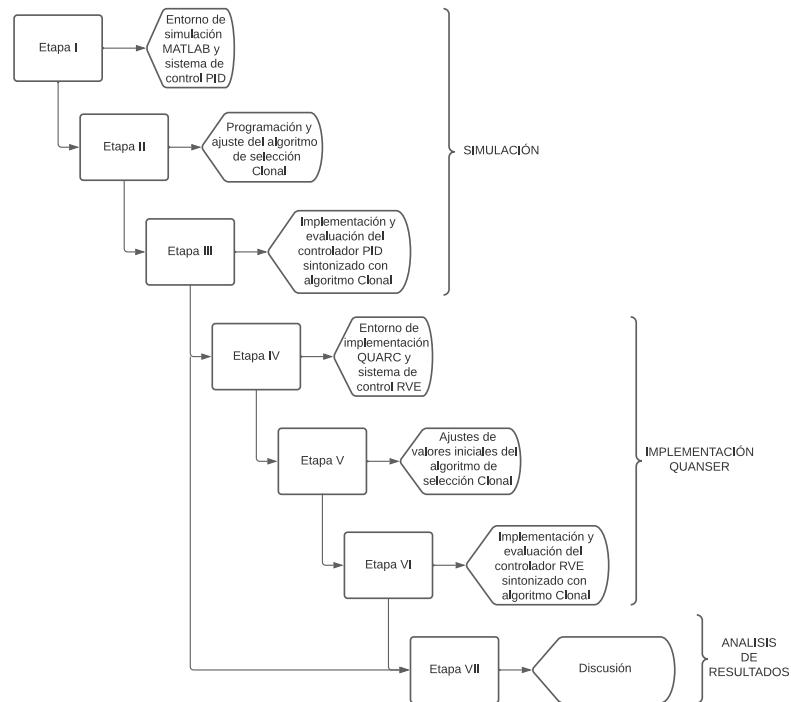


FIGURA 7: Descripción y implementación de la metodología en el desarrollo del proyecto de grado [Autoría propia]

En la anterior figura 7 se puede observar todas las etapas que conforman la metodología implementada en el proyecto. En el siguiente capítulo se abordará más profundamente dentro los diseños, como validaciones de los controladores y como a estos se les asigna una sintonización por medio del algoritmo clonal de la figura 5.

Capítulo 9

Diseño y validación de controladores

9.1. Simulación

9.1.1. Modelo físico del péndulo invertido en Simulink

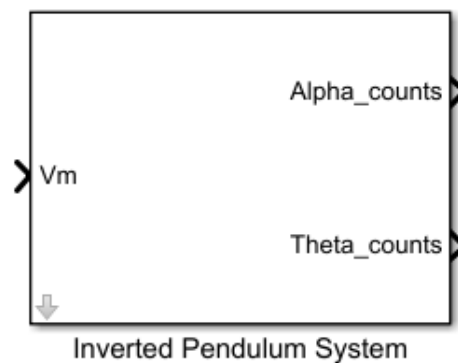


FIGURA 8: Modelo físico implementado en simulink [Autoría propia]

Para obtener resultados mas aproximados a la realidad se utiliza un modelo físico el cual considera los parámetros que tiene la planta del péndulo rotatorio invertido y que están anexados en la anterior tabla 4 . También en este modelo se introduce el factor de las resistencia del motor y la inductancia que este tiene al mecanismo del péndulo rotatorio invertido, utilizando una función de transferencia que representa a un motor DC. Esto se observa de mejor manera con la siguiente figura 9.

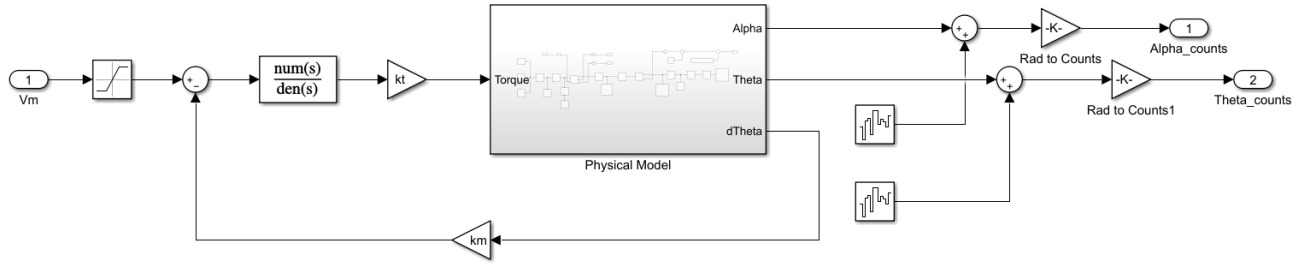


FIGURA 9: Modelo físico Implementado en simulink [Autoría propia]

Partiendo del modelo de la figura 9 se propone a hacer un controlador clásico PID el cual proporcione parámetros base para el ajuste automático del algoritmo bioinspirado dentro del entorno de simulink

9.1.2. Diseño del controlador clásico

Ya teniendo el modelo de variables de estado linealizado en un punto de operación $z_0^T = [000000]$ se procede a determinar controlabilidad y observabilidad. Esto permite determinar si el modelo linealizado tiene la capacidad de aplicar observadores de estados y también si se puede aplicar controladores a cada variable de estado que hace parte del modelo.

Para determinar la observabilidad del modelo se aplica la siguiente expresión.

$$O = \begin{bmatrix} C^T & A^T C^T & \dots & (A^T)^{n-1} C^T \end{bmatrix} \quad (9.1)$$

Reemplazando los valores del modelo de variable de estados de la ecuación (7.30) y (7.31), se obtiene la siguiente matriz de observabilidad.

$$Ob = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -17,8529 & -2,0055 & -0,2942 \\ 0 & 51,9530 & 3,6895 & -16,9967 \end{bmatrix} \quad (9.2)$$

Ya con la matriz de observabilidad, se determina si el sistema es observable, para esto se evalúa si existen autovalores dentro de la matriz distintos a cero, para lo cual se utiliza la

siguiente expresión.

$$\det(A - \lambda I) = 0 \quad (9.3)$$

Al reemplazar los valores correspondientes de las matrices de estado y identidad , se obtiene los siguientes auto valores

$$\begin{aligned} \lambda_0 &= -16,98 \\ \lambda_1 &= -1,00 + 4,21i \\ \lambda_2 &= -1,00 + 4,21i \\ \lambda_3 &= 1,00 \end{aligned} \quad (9.4)$$

Como se observa en la anterior expresión, cada autovalor de la matriz de observabilidad es distinto de cero, lo cual representa que cada valor solución es linealmente independiente entre si, haciendo posible utilizar observadores de estado a cada variable del sistema.

Ahora se procede a determinar la controlabilidad del sistema. Para esto se utiliza la siguiente expresión.

$$C_o = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} \quad (9.5)$$

Reemplazando los valores del modelo de variable de estados de la ecuación (7.30) y (7.31), se obtiene la siguiente matriz de controlabilidad.

$$C_o = \begin{bmatrix} 0 & 10,4 & -19,2 & 138,5 \\ 0 & -5,9 & 17,1 & 285,8 \\ 10,4 & -19,2 & 138,5 & -667,6 \\ -5,9 & 17,1 & 285,8 & -1041,0 \end{bmatrix} \quad (9.6)$$

Ya con la matriz de controlabilidad, se determina si el sistema es controlable, para esto se evalúa si existen autovalores dentro de la matriz distintos a cero, para lo cual se utiliza la siguiente expresión.

$$\det(A - \lambda I) = 0 \quad (9.7)$$

Al reemplazar los valores correspondientes de las matrices de estado y identidad , se obtiene los siguientes auto valores

$$\begin{aligned} \lambda_0 &= -856 \\ \lambda_1 &= -28,75 + 52,94i \\ \lambda_2 &= -28,75 - 52,94i \\ \lambda_3 &= 5,12 \end{aligned} \quad (9.8)$$

Como se observa en la anterior expresión, cada autovalor de la matriz de controlabilidad es distinto de cero, lo cual representa que cada valor solución es linealmente independiente entre si, haciendo posible controlar todas las variables de estado.

Ya conociendo que es posible controlar el sistema linealizado del péndulo invertido, se procede a diseñar el controlador clásico. Para esto como se observa en la ecuación (7.31), se debe hacer la transición de un sistema SIMO (Unica Entradas Múltiples Salidas) a un sistema SISO (Única Entrada Única Salida). Para esto se analiza la figura 10.

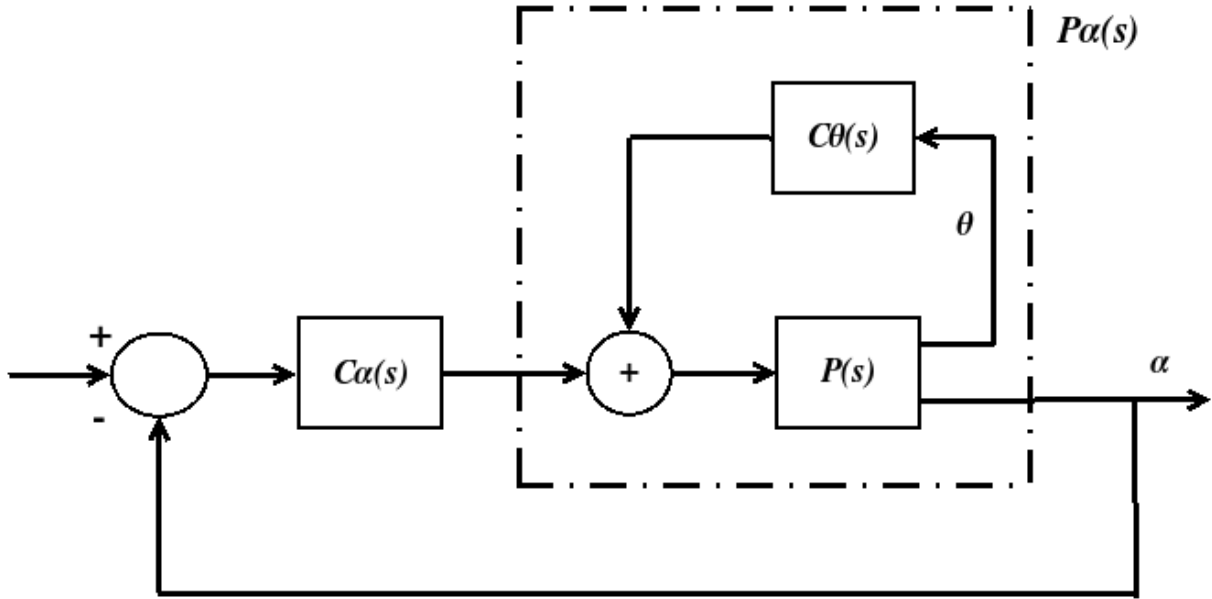


FIGURA 10: Representación para transformación de un sistema SIMO a un sistema SISO [Autoría propia]

Como se observa, para poder simplificar el diseño del controlador y convertir un sistema SIMO a un sistema SISO, se cierra un lazo de retroalimentación para asignar una posición inicial del ángulo de giro del motor θ . $P(s)$ representa el modelo en variables de estado de las ecuaciones (7.30) y (7.31). $C\alpha$ representa el controlador clásico que busca establecer la posición vertical del péndulo invertido y por medio el cual el algoritmo de sintonización bioinspirado encontrará los mejores parámetros para el control.

La función de transferencia $C\theta$ se define directamente por el autor del toolbox base que se utiliza en este documento el cual tiene la siguiente forma.

$$C\theta = \frac{102s + 100}{s + 50} \quad (9.9)$$

Al cerrar el lazo de control de $P\alpha$ y hacer la simplificación se obtiene la siguiente función de transferencia.

$$P\alpha = \frac{10,45s^3 + 533,4s^2 + 1237s + 33920}{s^5 + 52,91s^4 - 862,9s^3 + 797,3s^2 - 63800s - 67840} \quad (9.10)$$

$P\alpha$ representa la nueva función de transferencia la cual contiene el controlador del ángulo θ en la posición cero y actúa como nueva planta para el sistema a controlar del péndulo rotatorio invertido.

9.1.2.1. Diseño de controlador por LGR

Se aclara que el diseño del controlador se hace en función del modelo linealizado en el punto de operación $[0,0,0,0]$, el cual contiene la función de transferencia de la ecuación (9.10) y representa el modelo a utilizar en esta sección de simulación del péndulo rotatorio invertido. Para el diseño del controlador clásico PID se procede a utilizar la herramienta de Control System Designer de Matlab, la cual permite ajustar un diseño de controlador en función a unos parámetros de diseño. Para esta planta se asignó un diseño de control de $Sp < 20\%$, $Ts < 2s$ y $Ess = 0\%$. A continuación se observa la respuesta de la planta en lazo abierto.

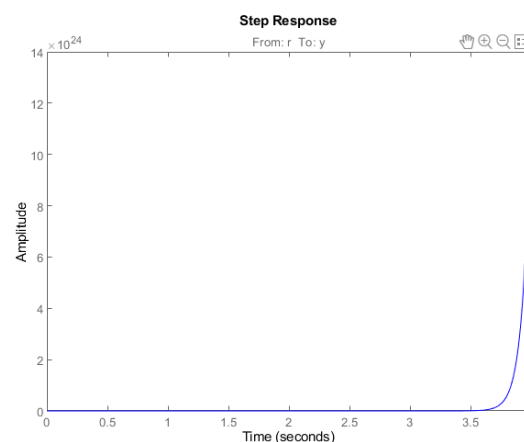


FIGURA 11: Respuesta de la planta en lazo abierto [Autoría propia]

Como se observa en la anterior figura 11, la respuesta que tiene el sistema en lazo abierto es inestable, esto se evidencia al momento de analizar el comportamiento de los polos del sistema.

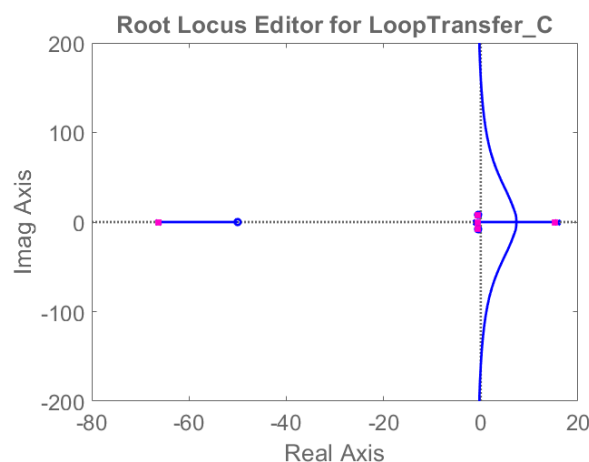


FIGURA 12: Ubicación de los polos del sistema en lazo abierto dentro del plano complejo [Autoría propia]

Se puede analizar a partir de la anterior figura 12, que la trayectoria que tienen los polos de la planta se encuentran en el plano complejo. A continuación se procede a asignar las condiciones de diseño dentro del plano complejo, para así conocer en que posición deben estar los polos del sistema en lazo cerrado para cumplir con las condiciones de diseño previamente asignadas.

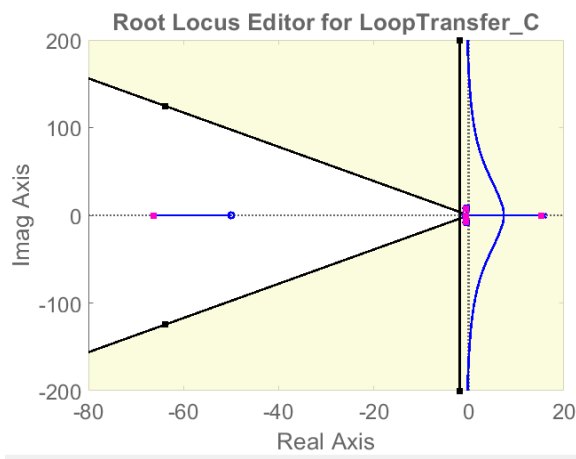


FIGURA 13: Plano complejo con asignación de región de diseño para el controlador [Autoría propia]

Se puede observar a partir de la anterior figura 13 la región de diseño en donde se pueden ubicar los polos para poder cumplir con las condiciones de diseño del controlador en lazo cerrado. Debido a que las trayectorias se encuentran por defecto en lazo abierto en la zona inestable, se procede a asignar un cero y un polo al controlador, ajustando una ganancia para así obtener la siguiente ubicación de los polos dentro del plano complejo.

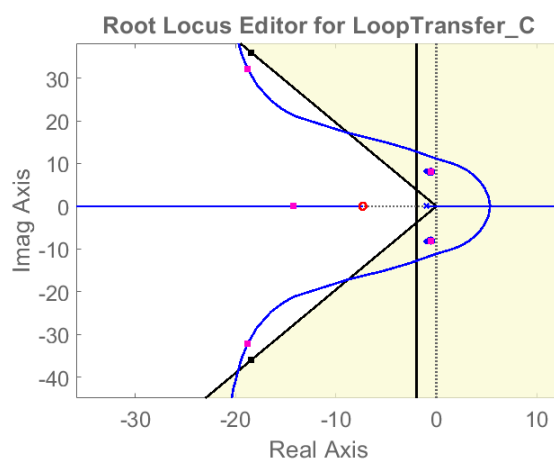


FIGURA 14: Plano complejo con asignación de región de diseño para el controlador [Autoría propia]

Al asignar los polos de la anterior figura 14 con el controlador en lazo cerrado se puede observar que todos los polos del sistema de la planta se encuentran en la zona estable del plano complejo, esto permite obtener la siguiente respuesta al escalón en lazo cerrado.

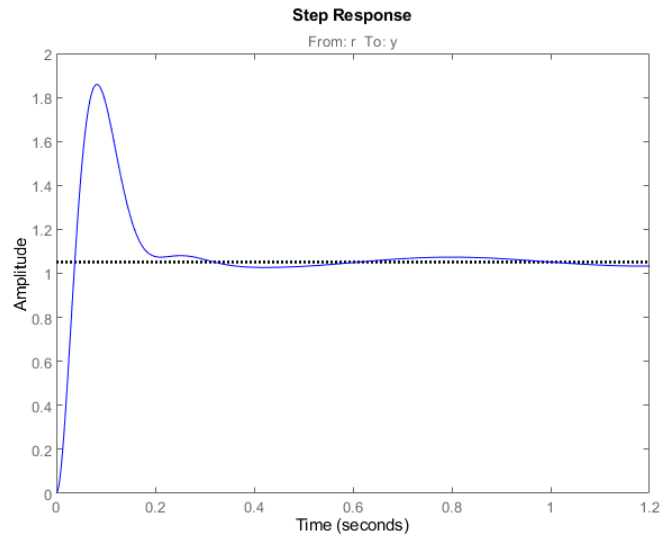


FIGURA 15: Respuesta al escalón con el controlador diseñado [Autoría propia]

Como se puede observar en la anterior figura 15, la planta en lazo cerrado del modelo no lineal con el controlador mantiene un sobre pico del 80 %, pero con un tiempo de establecimiento menor al estipulado en las condiciones de diseño. Esta diferencia del sobre pico con respecto a la asignada por la condición de diseño se debe a que el sistema a controlar es de un orden mayor a 2, lo cual hace que los controladores diseñados distingan en su resultados al momento de evaluarlos. La función de transferencia que permite la anterior respuesta se observa en la siguiente expresión.

$$C(\alpha) = \frac{278,78(s + 7,329)}{s + 50} \quad (9.11)$$

Se ajusta la anterior ecuación a una expresión donde se evidencien las ganancias proporcional, derivativa y la posición del polo de manera en la que se pueda hacer variación de estos parámetros.

$$C(\alpha) = 40,38 + 4,7688 * \left(\frac{50}{1 + 50 * \frac{1}{s}} \right) \quad (9.12)$$

Ya con la función de transferencia se procede a evaluar el desempeño del controlador en la plataforma simulink, en donde se utiliza un bloque el cual contiene un modelo aproximado a

la planta del péndulo invertido. En simulink se implementa el siguiente bloque, el cual contiene el mismo esquema de la figura 10.

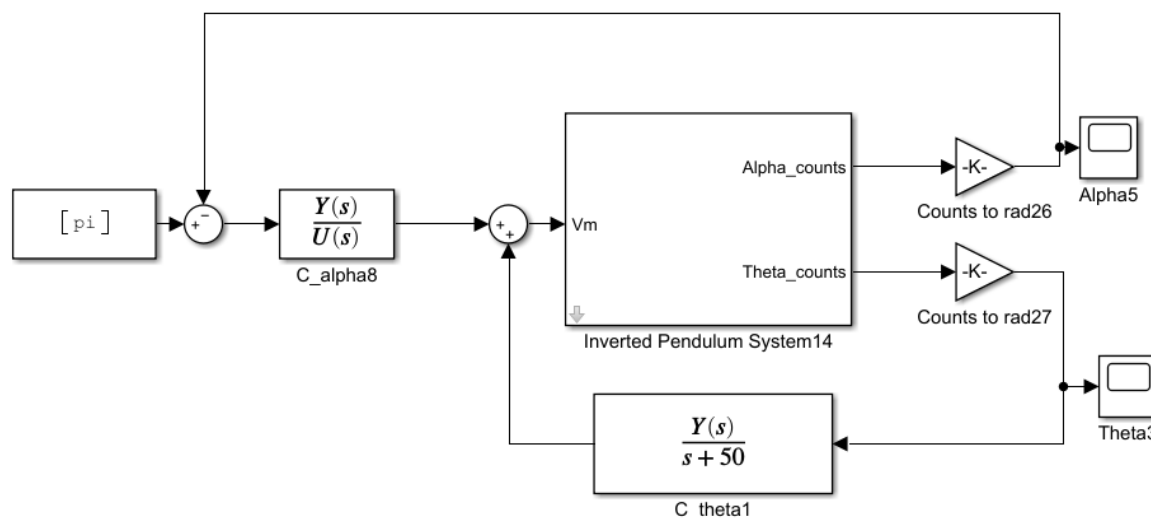


FIGURA 16: Esquema de control para la planta del péndulo rotatorio invertido para el ángulo alpha en Simulink [Autoría propia]

Acorde al anterior esquema de la figura 16 del sistema, el cual representa el modelo no lineal en simulación del péndulo invertido, se puede observar dos controladores base los cuales actúan sobre las variables alpha y theta. El controlador C_{theta1} , contiene la función de transferencia de la ecuación (9.9). Esto permite ajustar la condición inicial del ángulo de giro del motor que contiene el brazo que sostiene al péndulo invertido en cero, esto se hace con el objetivo de que a medida en que varíe el ángulo alpha y este afecte la condición inicial del ángulo theta, este controlador mantenga esa condición al momento de ejecutarse el tiempo de transición en que el péndulo se estabiliza en su punto de operación de 180 grados con respecto al eje de giro del brazo. Posteriormente, se tiene la función de transferencia del controlador C_{alpha8} , la cual se representa en la ecuación (9.11). Esta permite controlar la posición del ángulo alpha, el cual dependiendo de el ángulo inicial en el que se encuentre el péndulo invertido, este va ir ajustando su posición a 180 grados o si se considera en radianes al valor de π .

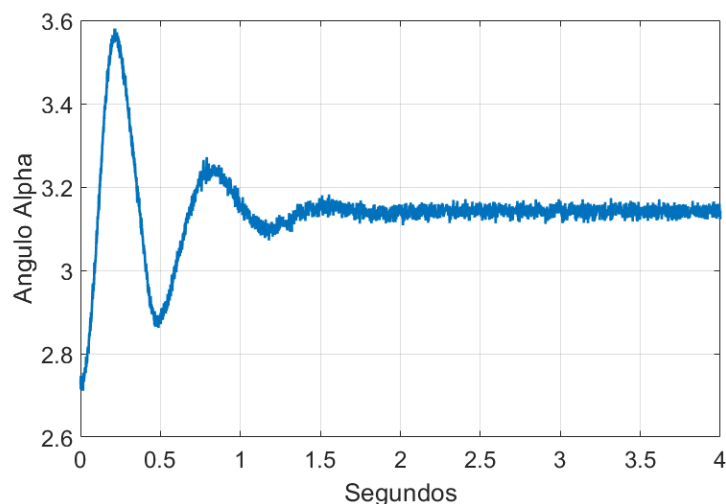


FIGURA 17: Respuesta del sistema de control para la planta del péndulo rotatorio invertido para el ángulo Alpha [Autoría propia]

En la figura 17 se puede observar la respuesta que tiene el sistema del péndulo rotatorio invertido al accionar el controlador y asignarle una posición inicial de 156 grados. El controlador permite estabilizar el ángulo alpha del péndulo invertido en su posición de 180 grados y es coherente con la condición de diseño del tiempo de estabilización de menos de dos segundos. Esta condición inicial del péndulo invertido permite conocer el máximo límite en el cual el sistema puede operar correctamente con el controlador diseñado, debido a que si se excede del rango de 24 grados, el modelado del sistema se saldría de su punto de operación y el controlador diseñado en base a este no operaría de forma correcta.

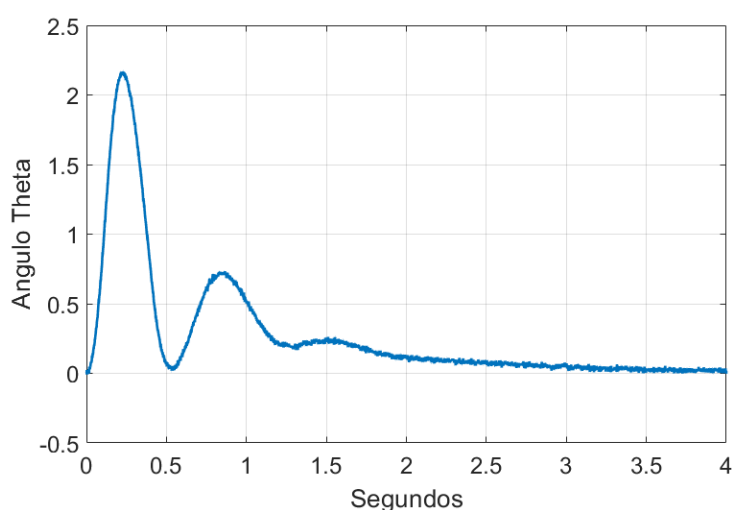


FIGURA 18: Respuesta del sistema de control para la planta del péndulo rotatorio invertido para el ángulo Theta [Autoría propia]

En la figura 18 se puede observar la respuesta que tiene el ángulo Theta del motor de giro de la planta del péndulo invertido, la cual logra estabilizarse en el ángulo cero en el momento que el péndulo invertido se encuentra en su posición de equilibrio de 180 grados. Debido a que el control está basado en un modelo linealizado en $z_0^T = [000000]$ las otras variables de estado de velocidad del ángulo alpha y theta deben inicializarse en cero.

9.1.2.2. Algoritmo de sintonización CLONAL

Se ajusta un diagrama de bloques del sistema a simular en donde en función a los datos de error del sobrepaso máximo, tiempo de establecimiento y error de estado estacionario, se ajusta de manera heurística por medio del algoritmo las ganancias de un controlador PID el cual efectúa la acción de control sobre la planta del péndulo rotatorio invertido. Esto se observa de mejor forma en la siguiente figura 19.

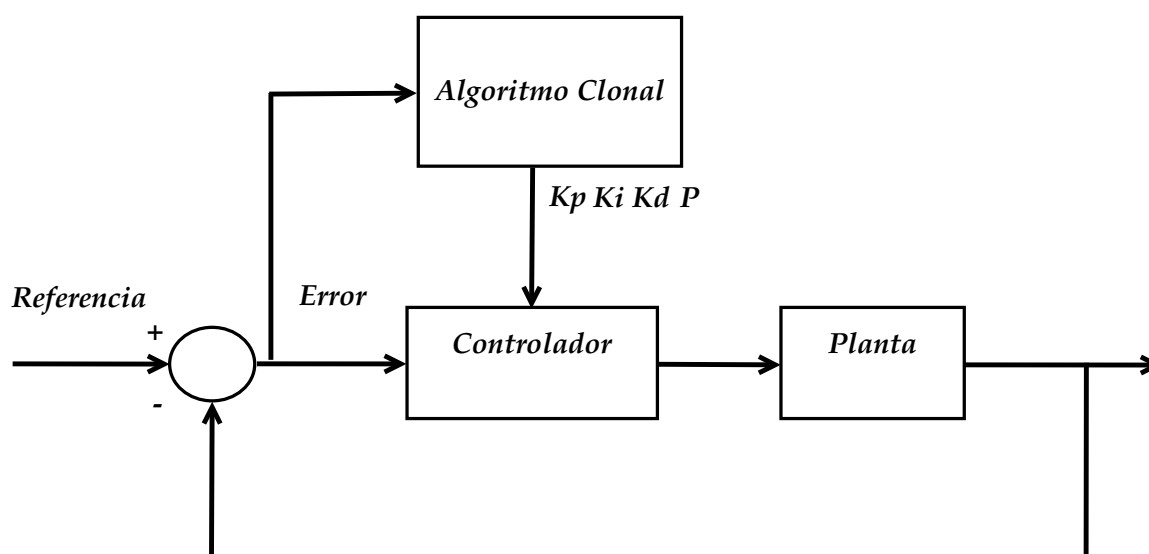


FIGURA 19: Diagrama de bloques del sistema a implementar [Autoría propia]

Para la programación del algoritmo de sintonización CLONAL, se asigna un número de anticuerpos los cuales representan a la población inicial en un espacio de búsqueda. Debido a que el problema que se busca resolver es un controlador que sintonice parámetros para mejorar el desempeño de sobrepaso máximo, tiempo de establecimiento y error de estado estacionario de una respuesta de control en su estado transitorio, se utiliza el esquema presentado en la figura 20.

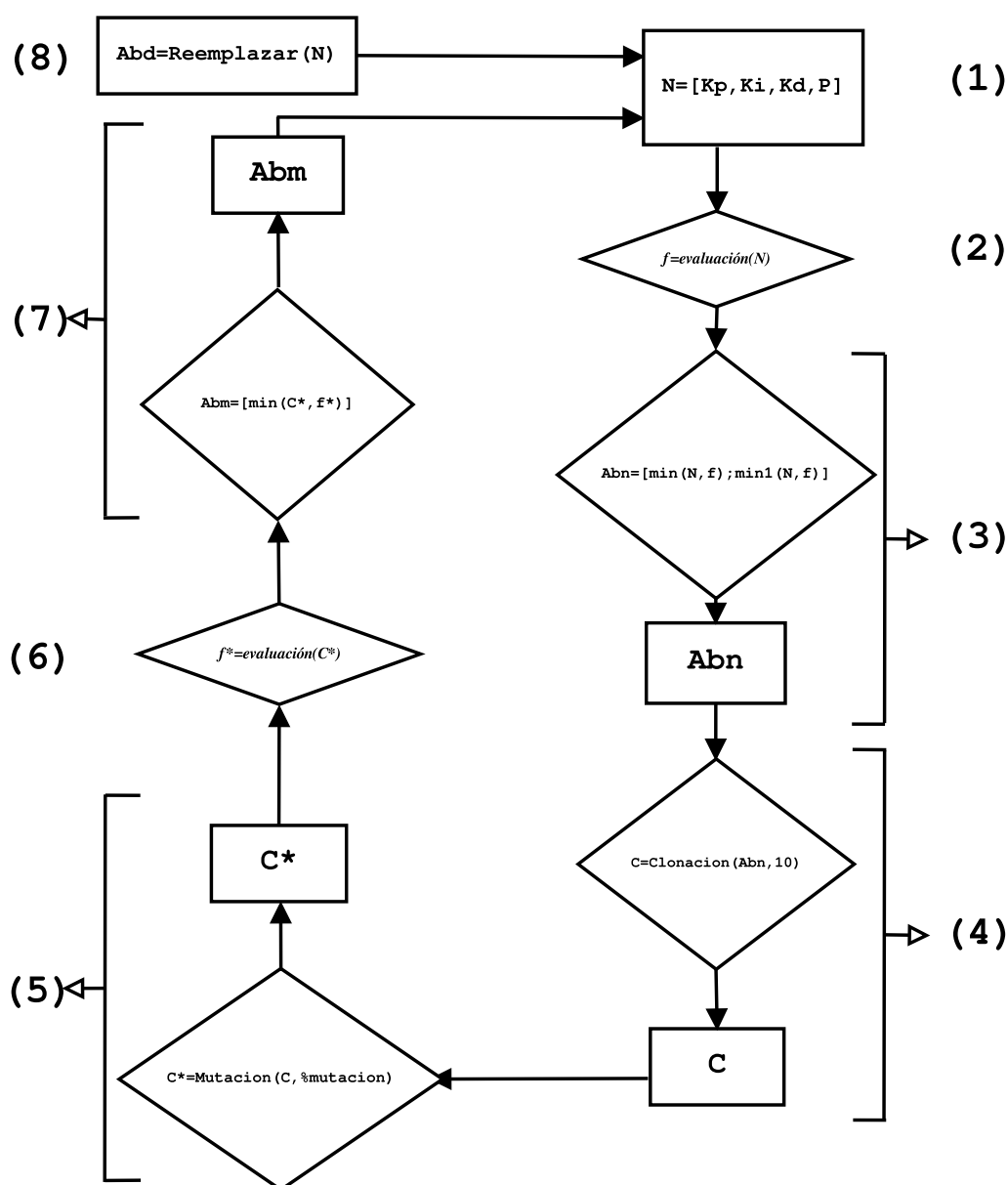


FIGURA 20: Diagrama de flujo que se utiliza para aplicar un proceso de optimización CLONAL [Autoría propia]

Como se puede observar, se tiene un diagrama de flujo el cual sigue una secuencia iterativa en donde se adaptan 8 pasos, los cuales representan un ciclo del algoritmo para lograr una sintonización o mejora de las imágenes de desempeño de la respuesta de control al péndulo rotatorio invertido. A continuación se detalla que hace cada numeral y como se ajusta para hacer una sintonización a los parámetros del controlador.

9.1.2.3. Paso 1: Definición de población de anticuerpos

En esta etapa inicial del algoritmo se genera una población inicial de anticuerpos, aplicado a este caso se genera un anticuerpo el cual representa las ganancias y ubicación del polo del controlador de la ecuación (9.12) $K_p=40.38$, $K_d=4.7688$ y $P=50$. La ganancia K_p representa el valor proporcional que actúa en el controlador, K_d representa el valor derivativo y el valor de P representa la posición inicial del polo que se requiere para que el sistema sea estable. Debido a que se propone un controlador PID, se agrega una ganancia integral con un valor pequeño de $K_i=0.1$. En este paso se hace la clonación y mutación de este primer anticuerpo reservado en memoria para crear 10 poblaciones y así proseguir con el siguiente paso.

9.1.2.4. Paso 2: Cálculo de la función de aptitud u objetivo

Al tener las 10 poblaciones de candidatos se hace una evaluación de que tanta afinidad tiene cada candidato. Para esto se utiliza la siguiente expresión que representa la función de aptitud.

$$f = \sqrt{(e_{M_p})^2 + (e_{t_s})^2 + (e_{e_{ss}})^2} \quad (9.13)$$

Donde e_{M_p} , e_{t_s} , $e_{e_{ss}}$ representan los valores de precisión del anticuerpo normalizados para cumplir con las condiciones de diseño y ajustarse con el antígeno. Cada valor que compone la función objetivo representa el error cuadrático obtenido del valor deseado del sobrepaso máximo , tiempo de establecimiento y error de estado estacionario menos el valor extraído anteriormente o valor real medido. Esto se representa con las siguientes expresiones.

$$e_{M_{p_N}} = \frac{M_p - M_{p_D}}{M_{p_D}}$$

$$e_{t_{s_N}} = \frac{t_{s_D} - t_s}{t_s} \quad (9.14)$$

$$e_{e_{ss_N}} = \frac{e_{ss_D} - e_{ss}}{e_{ss_D}}$$

M_p , t_s , e_{ss} representan los valores medidos actualmente que se obtienen cada vez que se evalúa un anticuerpo sobre la planta del péndulo rotatorio invertido. M_{p_D} , t_{s_D} , e_{ss_D} representan los

valores a donde se desea llegar y lo cuales sirven como referencia para que a medida que vaya iterando el algoritmo los parámetros del sobrepaso máximo, tiempo de establecimiento y error de estado estacionario vayan sintonizando a un valor deseado . Posteriormente al obtener los anteriores $e_{M_{pN}}$, $e_{t_{sN}}$, $e_{e_{ssN}}$ se debe aplicar un proceso de normalización utilizando el método mínimo-máximo, esto se hace con el objetivo de asignarle un mismo peso a los tres valores que componen la función objetivo para que así, no halla una mayor influencia de un valor por encima de otro y se maneje un mismo sistema de unidades. Para esto se aplica las siguientes expresiones.

$$N_{max} = \max(e_{M_{pN}} e_{t_{sN}} e_{e_{ssN}})$$

$$N_{min} = \min(e_{M_{pN}} e_{t_{sN}} e_{e_{ssN}})$$

Al obtener el valor máximo y mínimo de los tres errores se procede a hacer el siguiente calculo.

$$\begin{aligned} e_{M_p} &= \frac{e_{M_{pN}} - N_{min}}{N_{max} - N_{min}} \\ e_{t_s} &= \frac{e_{t_{sN}} - N_{min}}{N_{max} - N_{min}} \\ e_{e_{ss}} &= \frac{e_{e_{ssN}} - N_{min}}{N_{max} - N_{min}} \end{aligned} \tag{9.15}$$

Con los valores de los errores normalizados calculados de la ecuación (9.15) se asignan estos valores al calculo de la función de aptitud de la expresión (9.13). De esta manera si el anticuerpo cumple con la condición de tener mas afinidad; es decir, un menor valor con respecto hacia los otros anticuerpos, este ultimo pasara al siguiente proceso.

9.1.2.5. Paso 3: Selección de los mejores 2 anticuerpos

En este paso se procede a hacer una selección de los dos mejores candidatos los cuales tienen una mayor afinidad con respecto hacia los otros. Para esto, se toman los anticuerpos que hayan tenido un menor valor en la función de aptitud u objetivo, y se guardan en un espacio de memoria Abn. Se hace de esta forma para obtener mas posibilidades de conseguir

candidatos que converjan a una reducción en la función de aptitud posterior.

9.1.2.6. Paso 4: Clonación de los 2 mejores anticuerpos

En este paso se cogen los dos mejores candidatos o anticuerpos y se aplica un procesos de clonación, en donde cada anticuerpo se clona diez veces; Es decir, se obtendrían 20 nuevos anticuerpos los cuales se almacenan en un espacio de memoria C.

9.1.2.7. Paso 5: Mutación de los 2 mejores anticuerpos

En este paso se asigna un porcentaje de mutación con el propósito de expandir o reducir el espacio de búsqueda en donde los anticuerpos se ubiquen. Para esta aplicación se utiliza un mismo porcentaje de mutación a los 20 anticuerpos creados anteriormente en el proceso de clonación. En total en memoria C* se almacenan 30 anticuerpos, los 20 anticuerpos sacados a partir de los dos mejores anticuerpos de la primera población y los diez primeros anticuerpos creados en el paso 1.

9.1.2.8. Paso 6: Cálculo de la función de aptitud u objetivo con los anticuerpos clonados y mutados

En este paso se le asigna un valor de afinidad a todos los anticuerpos creados anteriormente con el propósito de conocer cual de todos presenta un menor valor de aptitud que se interpreta como mejor desempeño y se guarda en la estructura f .

9.1.2.9. Paso 7: Selección y almacenamiento del mejor anticuerpo del conjunto clonado y mutado

En este paso se hace la selección del mejor anticuerpo de los 30 candidatos, escogiendo el candidato con menor valor de aptitud en relación a la población general, donde se incluyen los anticuerpos creados en el primer paso y los anticuerpos creados a partir de los dos mejores candidatos mutados y clonados de la primera población. Al final este mejor candidato se

almacena en un espacio de memoria Abm, el cual representa el mejor anticuerpo obtenido durante la iteración. Este espacio de memoria Abm se almacena en un espacio no volátil, ya que en este campo se almacenará todos los mejores candidatos futuros que se obtengan, cada vez que se repita el ciclo iterativo. Por último para iniciar el siguiente ciclo el valor de N, inicia con las ganancias halladas en la anterior iteración y se vuelve a repetir el proceso.

9.1.2.10. Paso 8: Eliminación y reemplazo de los anticuerpos con menor afinidad para reinicio del ciclo

En este paso, se borran y reemplazan los diez primeros candidatos en el espacio de memoria N, clonando y mutando el mejor anticuerpo hallado en el ciclo anterior, para así aumentar la posibilidad de convergencia a un resultado que sintonice el controlador del péndulo rotatorio invertido

9.1.3. Modelo implementado en Simulink

Primero se requiere de implementar un diagrama de control de un sistema SISO el cual contenga un esquema similar al obtenido en la figura 19. Este diagrama de control debe permitir ajustar los parámetros Kp, Ki, Kd y P a medida de que el sistema se estabilice en su punto de equilibrio de 180 grados, que es cuando se encuentra verticalmente. Para esto se utiliza el diagrama presentado en la figura 21 .

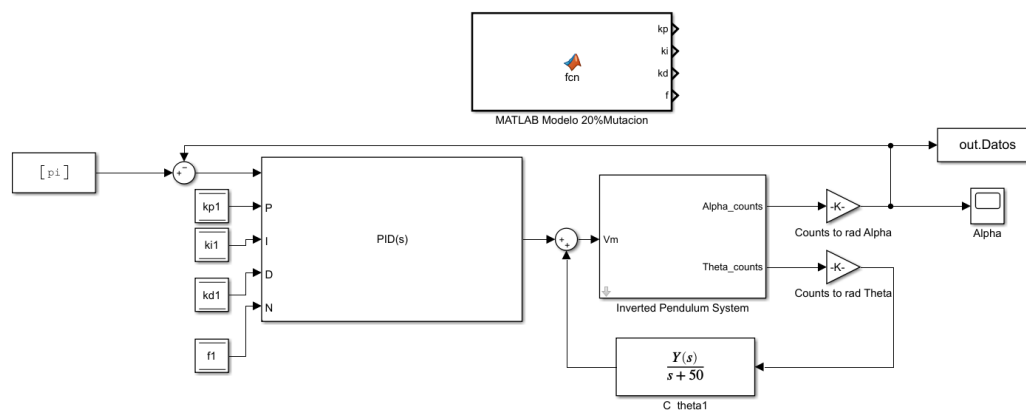


FIGURA 21: Diagrama de control implementado en Simulink [Autoría propia]

Se puede observar que se utiliza un bloque de control PID en donde como valores de entrada se pueden ingresar los parámetros K_p, K_i, K_d y F . En este caso la representación del control expresada en una ecuación matemática en función a los parámetros se puede observar de la siguiente forma.

$$K_p + K_i \frac{1}{s} + K_d \frac{F}{1 + F * \frac{1}{s}} \quad (9.16)$$

Posteriormente se agrega un bloque matlab function por medio el cual se realiza la programación del algoritmo de sintonización CLONAL utilizando bloques de Data Store Read, los cuales permiten seleccionar datos de salida que se almacenan en bloque Function. Debido a que se utiliza una lectura de datos los cuales provienen de bloque function, se debe crear un Bloque de Memoria de Dato o Data Store Memory, este permite definir una región de memoria dentro de la simulación para poder guardar los datos provenientes del bloque function y poder actualizarlos paralelamente cuando se esta ejecutando la simulación dentro de Simulink. Por ultimo se agrega un bloque de To Workspace o Zona de Trabajo, el cual registra el tiempo y la amplitud de la señal de salida del ángulo del péndulo para así poder extraer los valores de interés que requiere el algoritmo de sintonización, como es el sobrepaso máximo, tiempo de establecimiento y error de estado estacionario para así obtener los parámetros k_p, k_i, k_d y P que requiere el controlador para ajustar la posición de α del péndulo rotatorio invertido

9.1.4. Resultados obtenidos en simulación

En esta sección se registran los resultados obtenidos en simulación utilizando la plataforma MATLAB y aplicando el esquema de control de la figura 19 para la sintonización de parámetros de un controlador PID aplicado a un modelo aproximado del péndulo rotatorio invertido.

9.1.4.1. Sintonización CLONAL con un 20 % de mutación en Simulink

En esta sección se registra los resultados obtenidos de la simulación programando el diagrama de flujo del algoritmo de sintonización CLONAL de la figura 20 y implementándolo en un bloque function de Matlab, el cual ajusta los parámetros de un controlador PID sobre la planta del péndulo rotatorio invertido. Debido a que el algoritmo para hallar la función de aptitud necesita de unas condiciones de diseño las cuales se requieren para calcular el error para el

sobrepaso máximo, tiempo de establecimiento y error de estado estacionario de la ecuación (9.17), se asignan las siguientes condiciones.

$$e_{M_{pN}} = \frac{M_p - 3,14}{3,14}$$

$$e_{t_{sN}} = \frac{0,1 - t_s}{t_s} \quad (9.17)$$

$$e_{e_{ssN}} = \frac{0,01 - e_{ss}}{0,01}$$

Ya conociendo los errores que requiere la función de aptitud el algoritmo se encarga de encontrar las ganancias que requiere el controlador para su estabilización. Al realizar 13 iteraciones del algoritmo de selección CLONAL siguiendo la lógica de programación de la figura 20, se obtienen los resultados presentados en la tabla 5.

Iteración	Kp	Ki	Kd	P	Mp (rad)	ts (s)	ess (rad)
0	40.38	0.1	4.7688	50	3.567	1.2	0.00833
1	48.1017	0.0994	5.35879	43.0942	3.468	0.806	0.00833
2	57.3000	0.0988	6.0217	37.1422	3.399	0.8749	0.00833
3	68.2573	0.0982	6.7667	32.0123	3.369	0.528	0.00833
4	81.3151	0.1070	6.6819	32.5166	3.346	0.5702	0.00833
5	91.5144	0.0968	7.5778	32.8079	3.323	0.605	0.00833
6	109.0144	0.0962	8.5153	28.2766	3.305	0.667	0.00833
7	129.9303	0.1018	8.0675	29.2290	3.291	0.769	0.00833
8	154.7765	0.1012	9.0656	25.1921	3.278	0.789	0.00833
9	184.3854	0.1103	8.9520	25.5889	3.266	0.829	0.00833
10	219.6585	0.1306	8.8398	25.9921	3.254	0.984	0.00833
11	261.6794	0.1424	8.7290	26.4016	3.246	0.7894	0.00833
12	280.7223	0.1574	9.5985	25.2757	3.239	0.968	0.00833
13	317.3422	0.1650	10.6698	22.9596	3.232	0.882	0.00833

TABLA 5: Resultados obtenidos de la sintonización durante 13 iteraciones del algoritmo CLONAL con 20 % de mutacion

Como se puede observar el algoritmo consigue disminuir considerablemente el sobrepaso máximo y el tiempo de establecimiento en comparación al controlador base diseñado por LGR. Debido a que el sistema se logra estabilizar siempre a un valor final muy aproximado a 3.14 (rad), se utiliza un mismo error de estado estacionario para cada iteración. Para visualizar el desempeño de cada salida de control obtenida por iteración se anexan la siguientes figuras 22 y 24 .

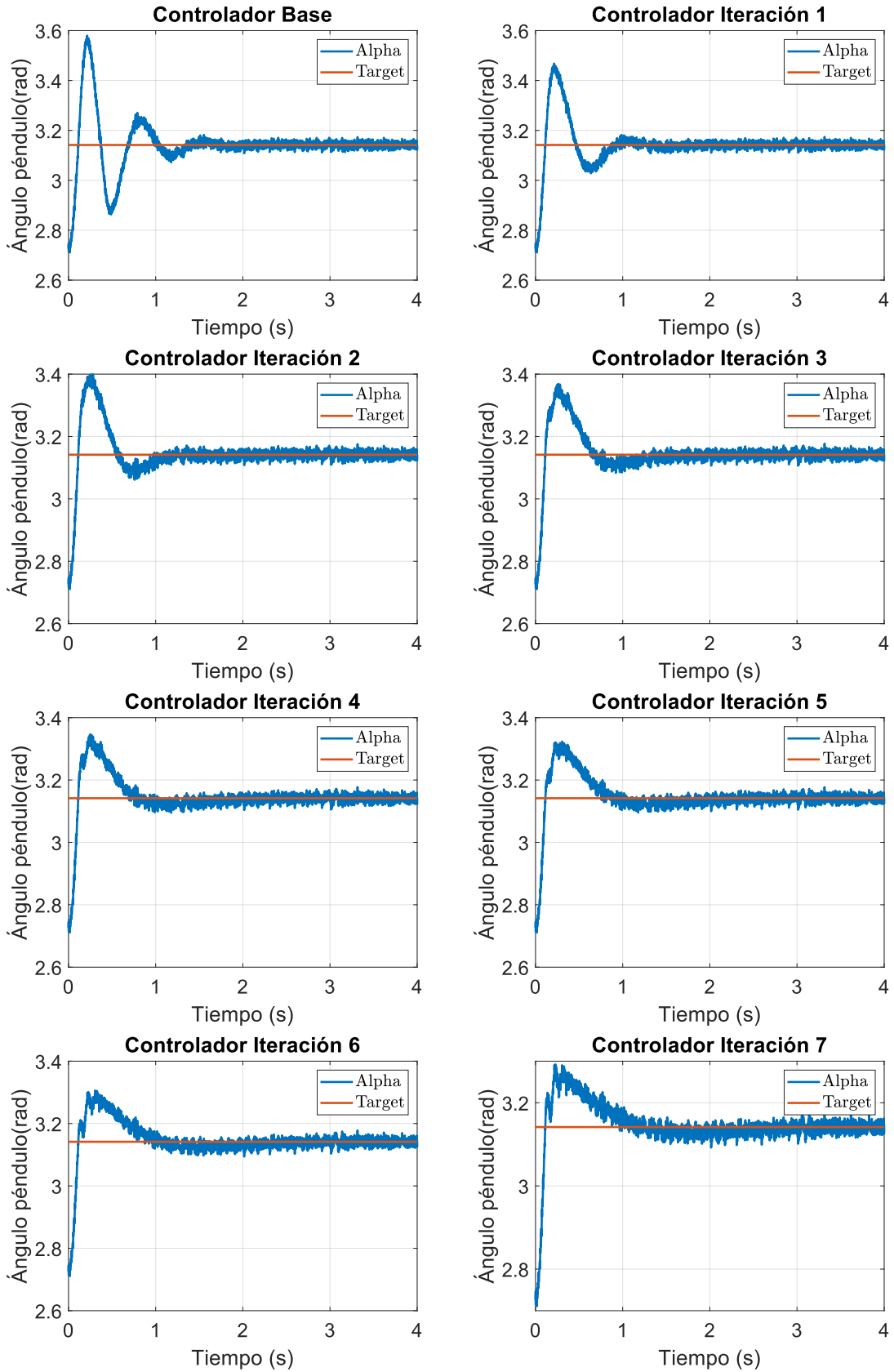


FIGURA 22: Respuesta del controlador de (0-7) iteraciones del algoritmo CLONAL con 20% de mutación [Autoría propia]

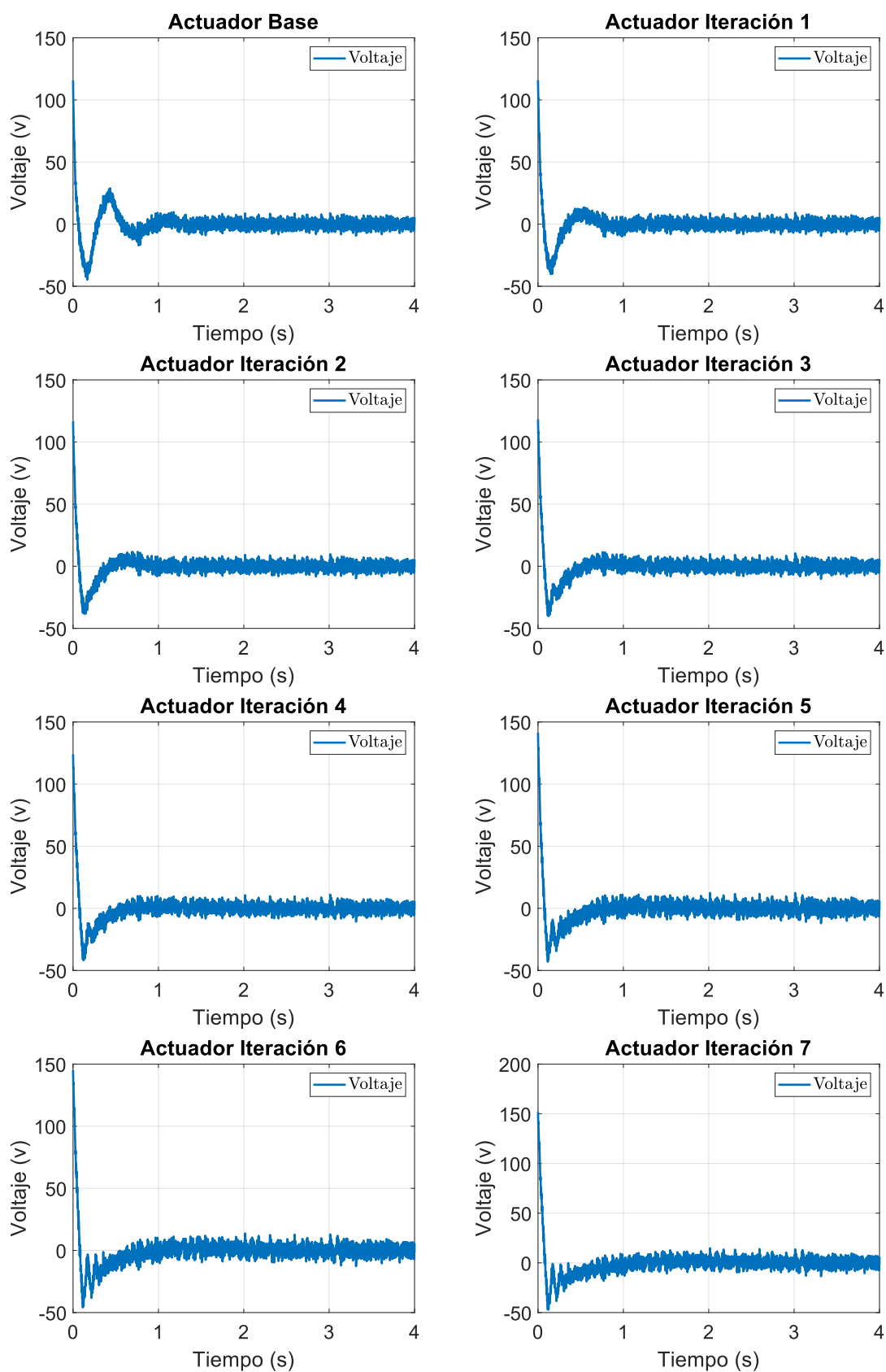


FIGURA 23: Respuesta del actuador de (0-7) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]

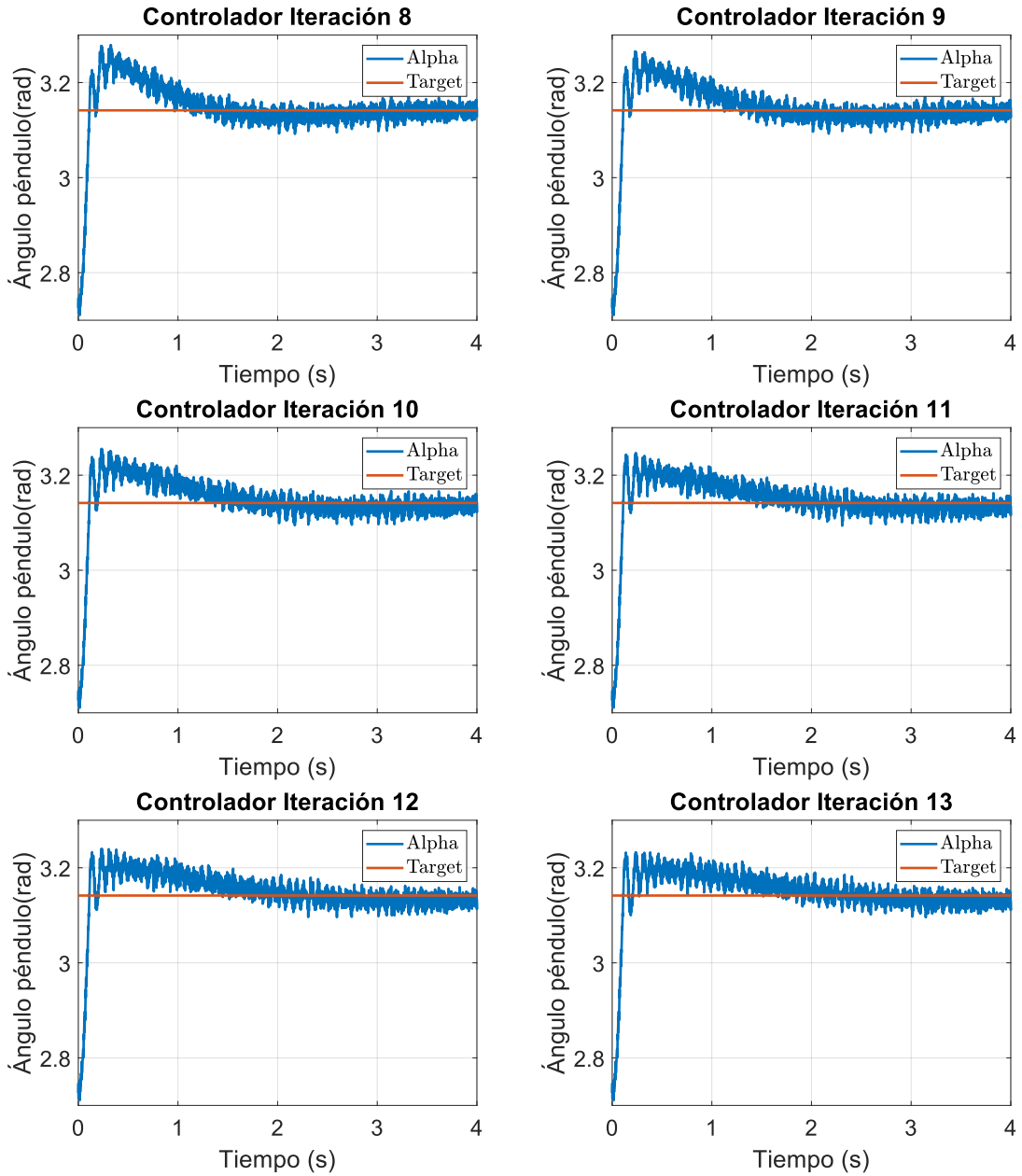


FIGURA 24: Respuesta del controlador de (7-13) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]

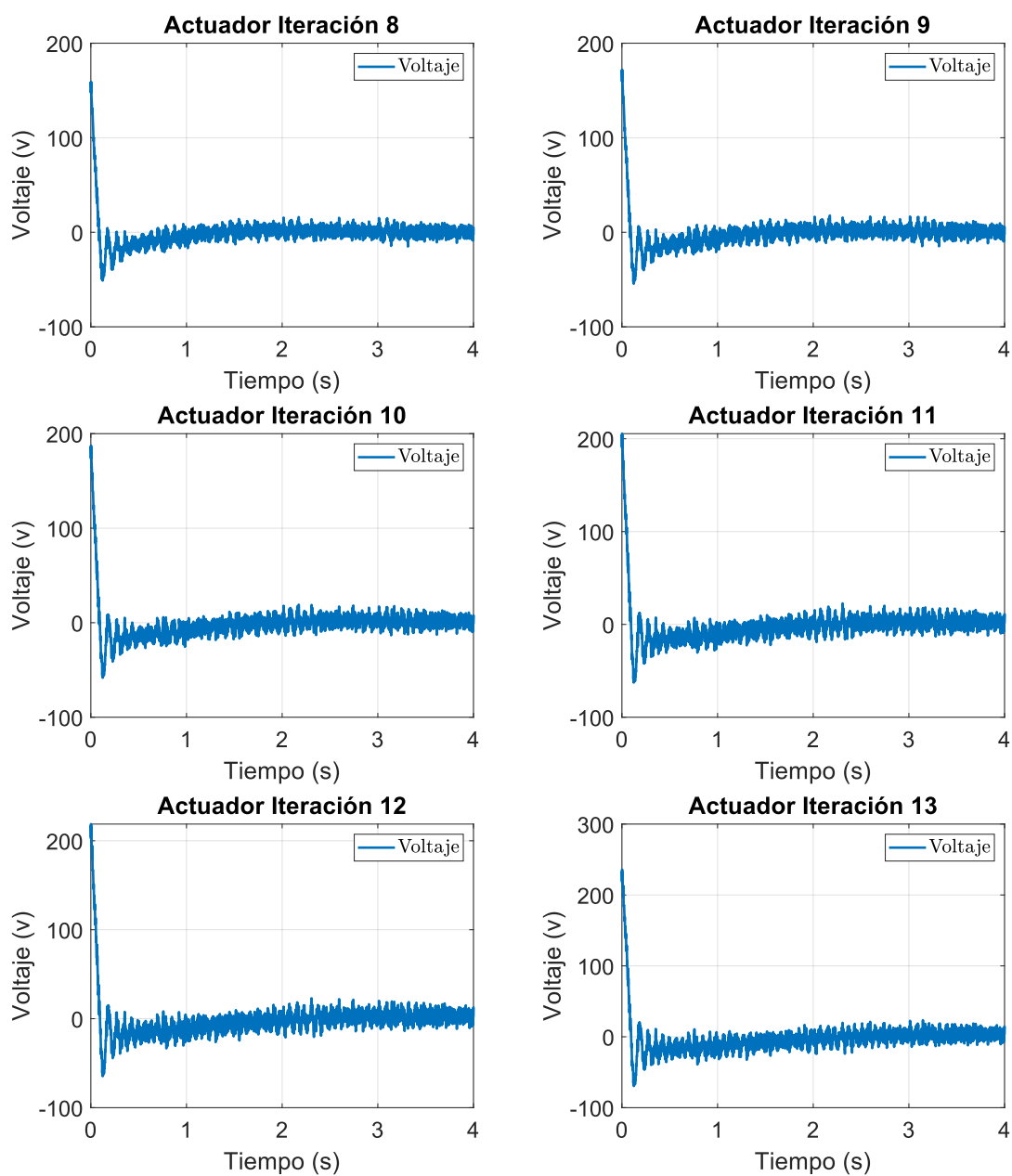


FIGURA 25: Respuesta del actuador de (7-13) iteraciones del algoritmo CLONAL con 20 % de mutación [Autoría propia]

Por último en la tabla 6 se presenta el comportamiento de la función de aptitud u objetivo por cada iteración. Para esto se toman los valores de aptitud que hallan tenido un menor valor en comparación a los 29 anticuerpos creados por cada iteración.

Iteración	Valor de aptitud
0	1.3808
1	1.3389
2	1.3147
3	1.2496
4	1.2305
5	1.2187
6	1.2098
7	1.2030
8	1.1964
9	1.1854
10	1.1566
11	1.1074
12	1.1040
13	1.10018

TABLA 6: Resultados de función de afinidad obtenidos de la sintonización durante 13 iteraciones del algoritmo CLONAL con 20 % de mutacion

El resultado de la tabla 6 se muestra de mejor manera realizando una gráfica la cual denote su transición en función al número de iteraciones que realiza el algoritmo. Esto se puede ver con la figura 26 a continuación.

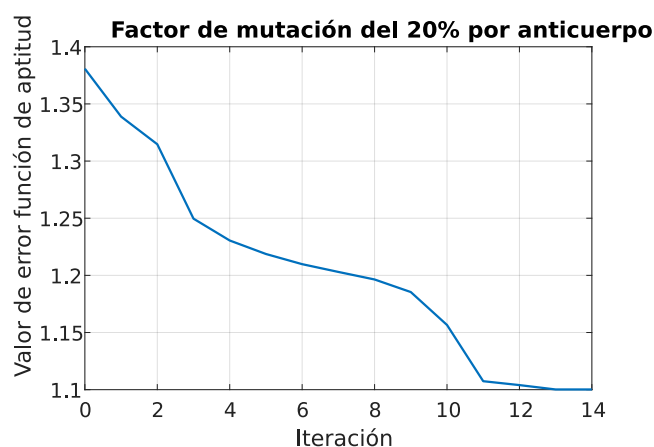


FIGURA 26: Gráfica de afinidad por ciclo de iteración del algoritmo CLONAL [Autoría propia]

Al analizar los resultados obtenidos, se observa un proceso de sintonización mas suave, en donde al momento de alcanzar la tercera iteración, el tiempo de establecimiento va aumentando

mientras que el sobrepaso va disminuyendo. Al llegar a la última iteración se observa que a pesar de que se aumento el tiempo de establecimiento a 0.882 segundos en comparación a la iteración 3 que tenia 0.528 segundos , el algoritmo pudo reducir el sobrepaso máximo hasta un valor de 3.232 radianes. También en la figuras 23 y 25 se observa un tiempo de consumo de voltaje menor a medida de que se va aumentando las iteraciones del algoritmo. Por otra parte al analizar las figuras obtenidas 22 y 24 se observa un ruido asociado a la señal de salida de sistema de control. Este ruido asociado representa a los posibles ruidos que se pueden obtener desde una implementación real y que son asignado directamente en el modelo simulado de la planta como se puede observar en la figura 9.

9.1.4.2. Sintonización CLONAL con un 50 % de mutación en Simulink

En esta sección se aplica las mismas condiciones de diseño dadas por el ecuación (9.17) para encontrar los errores que requiere la función de aptitud, pero cambiando el factor de mutación a un 50 % para obtener los resultados presentados en la tabla 7

Iteración	Kp	Ki	Kd	P	Mp (rad)	ts (s)	ess (rad)
0	40.38	0.1	4.7688	50	3.567	1.2	0.00833
1	62.09313	0.1093	5.2167	42.9116	3.406	0.480	0.00833
2	75.5167	0.1033	6.1754	36.4891	3.399	0.521	0.00833
3	92.5635	0.1269	8.9034	23.3901	3.322	0.6671	0.00833
4	134.0737	0.08872	7.6154	32.5166	3.293	0.769	0.00833
5	190.5253	0.07193	8.9568	25.4836	3.262	0.896	0.00833
6	253.5476	0.0838	10.327	22.9923	3.245	0.969	0.00833
7	378.2708	0.09102	14.7419	27.0773	3.227	0.789	0.00833
8	576.2967	0.0763	18.9691	26.7701	3.217	0.174	0.00833
9	703.027	0.09614	13.7596	36.2095	3.208	0.107	0.00833
10	807.05718	0.0671	13.722	45.2072	3.199	0.107	0.00833

TABLA 7: Resultados obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 50 % de mutación

Como se pudo observar, el algoritmo consigue disminuir considerablemente el sobrepaso máximo y el tiempo de establecimiento en comparación al controlador base diseñado por LGR. Al comparar el desempeño con los resultados obtenidos del algoritmo con un 20 %, se observa que al aumentar la mutación, la sintonización converge mas rápido y con menos iteraciones. Para visualizar el desempeño de cada salida de control obtenida por iteración se anexan las siguientes figuras 27 y 29 .

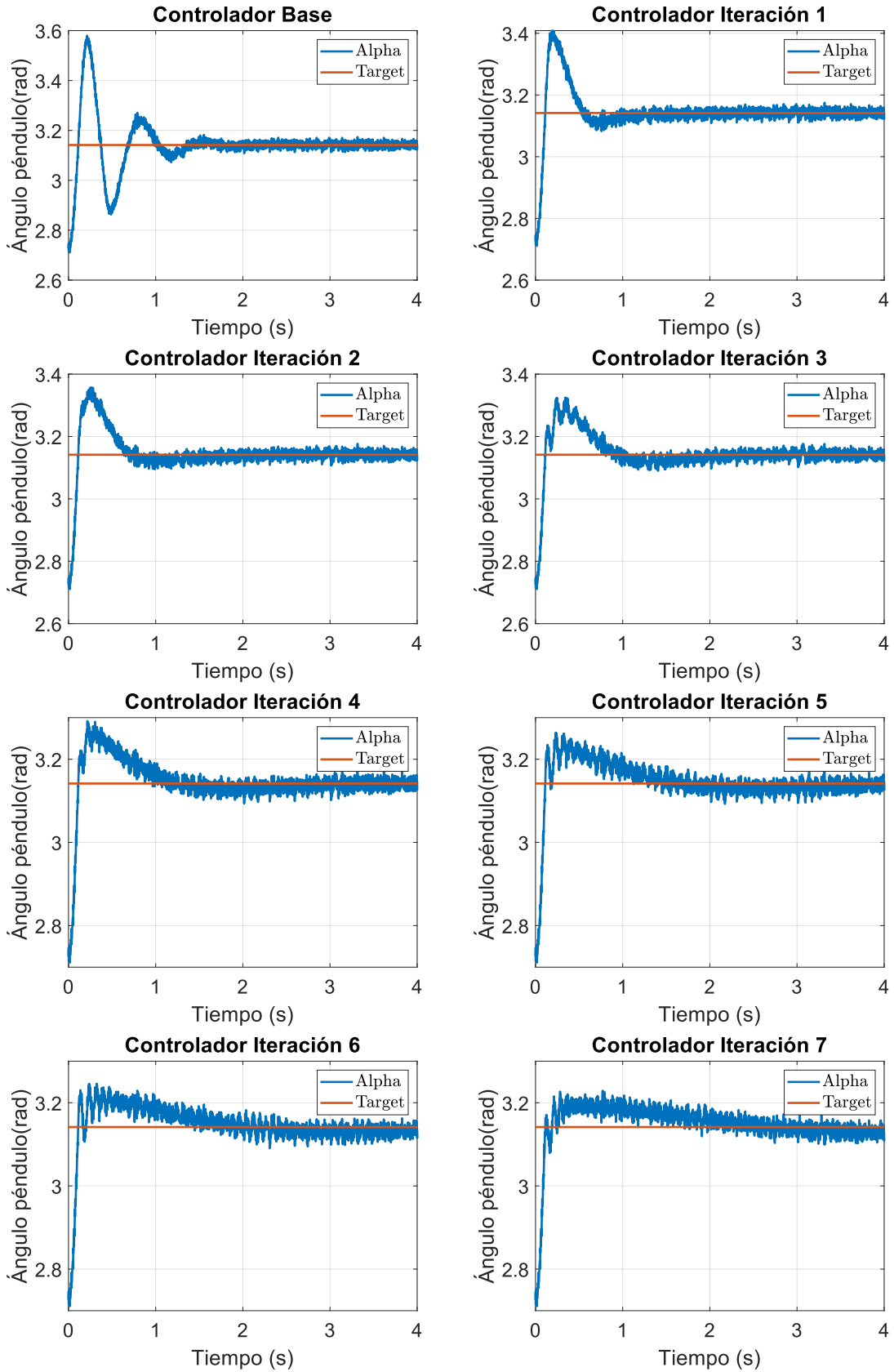


FIGURA 27: Respuesta del controlador de (0-7) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]

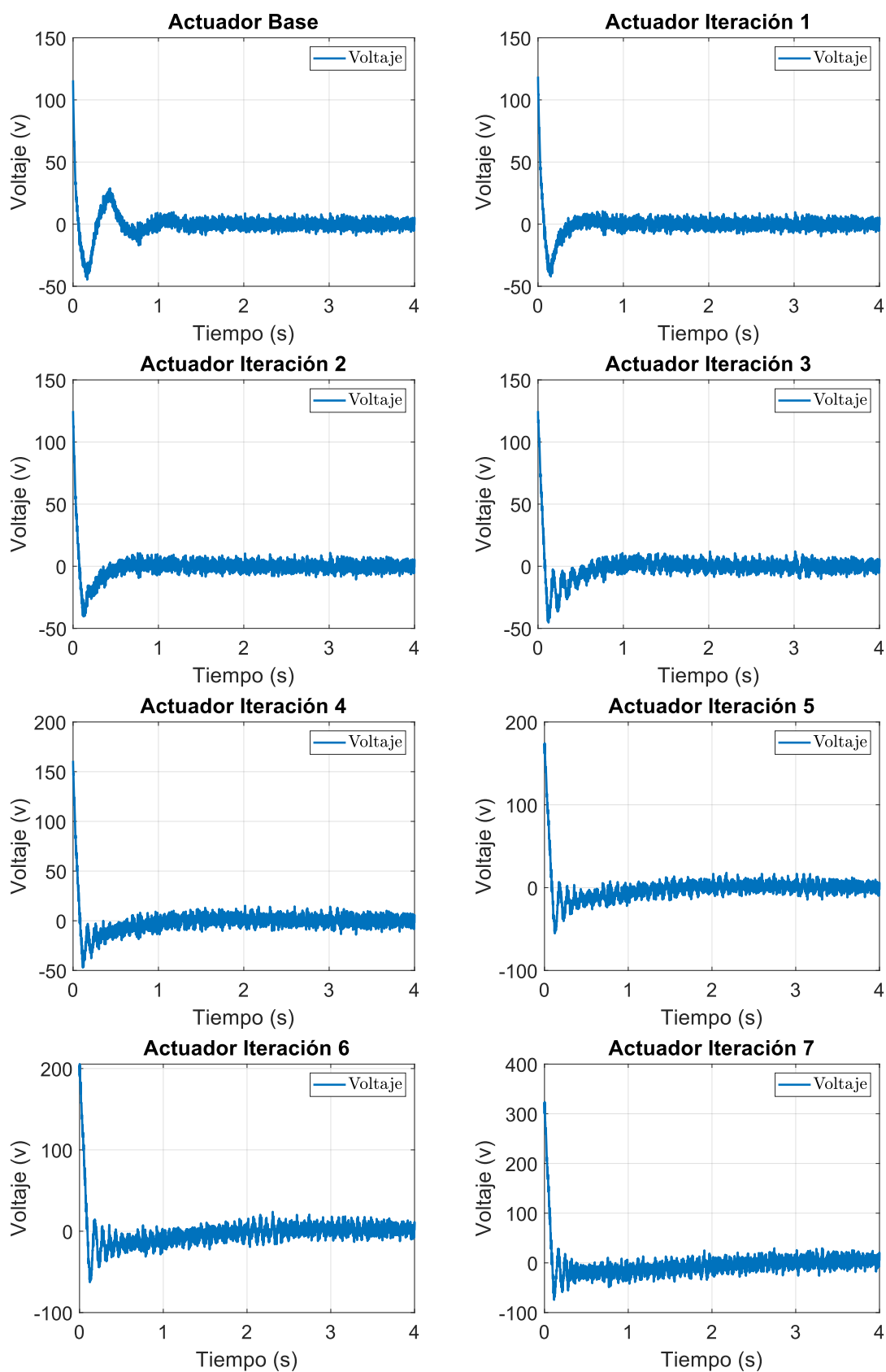


FIGURA 28: Respuesta del actuador de (0-7) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]

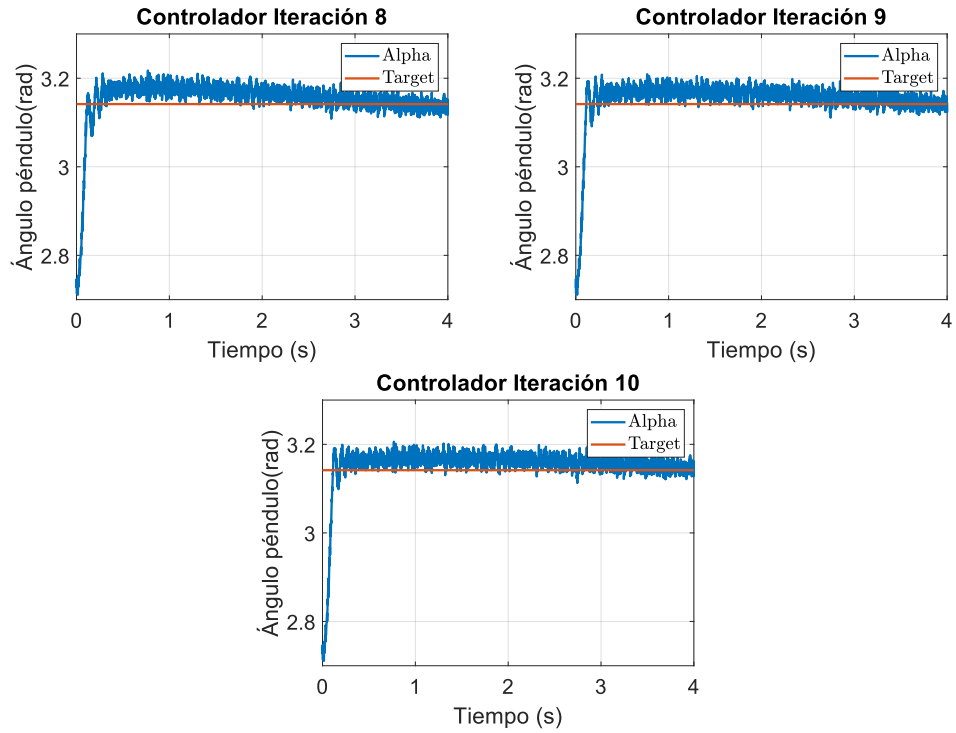


FIGURA 29: Respuesta del controlador de (7-10) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]

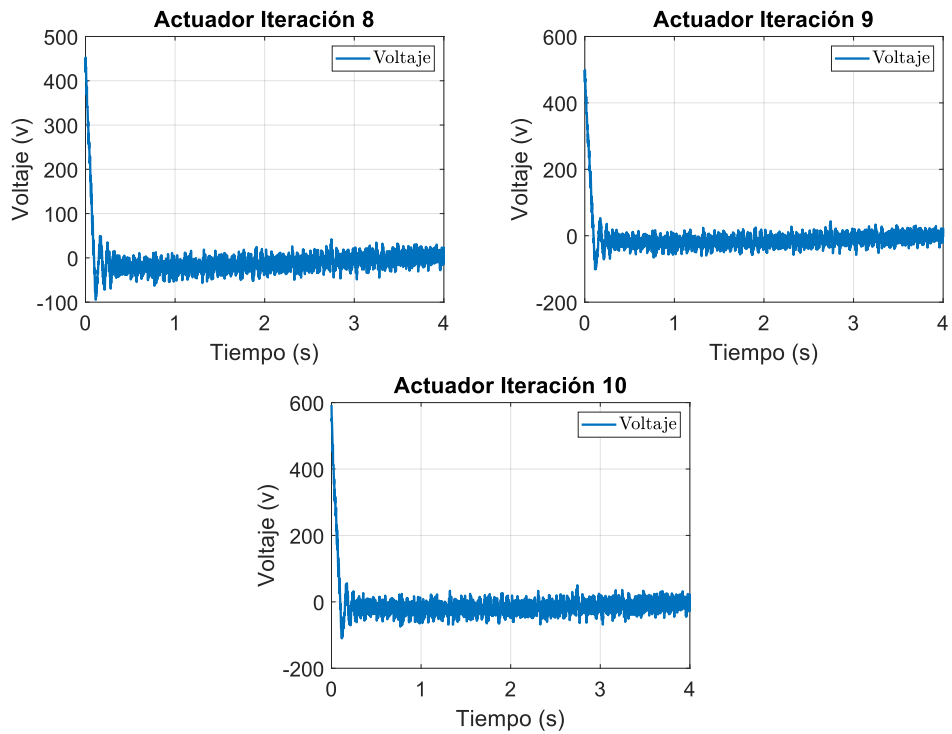


FIGURA 30: Respuesta del actuador de (7-10) iteraciones del algoritmo CLONAL con 50 % de mutación [Autoría propia]

Al observar los resultados obtenidos de las anteriores figuras, se puede denotar que al aumentar la mutación a un 50 por ciento, las imágenes de desempeño convergen mucho mas rápido desde la novena iteración. Esto permite obtener en la última iteración un sobrepaso máximo de 3.199 radianes, pero con un tiempo de establecimiento muy cercano al deseado en las condiciones de diseño de la ecuación (9.17), con un valor de 0.107 segundos. También al observar las figuras 28 y 30 de los actuadores obtenidos se analiza que a medida que se incrementa las iteraciones, la voltaje que requiere el sistema de control se estabiliza mas rápido y permite reducir el consumo promedio que requiere el sistema de control; Por otro lado, al comparar los resultados con la sintonización con un factor de mutación del 20 %, se puede observar un aumento del punto máximo de voltaje a medida en que se aumenta el numero de iteración. En las figuras 27 y 29 se puede observar que los controladores con mas iteraciones se estabilizan mas rápido al valor objetivo o target del sistema de referencia para el control.

Por último en la tabla 8 se muestra el comportamiento de la función de aptitud u objetivo por cada iteración. Para esto se toman los valores de aptitud que hallan tenido un menor valor en comparación a los 29 anticuerpos creados por cada iteración.

Iteración	Valor de aptitud
0	1.3808
1	1.3544
2	1.3446
3	1.3401
4	1.3331
5	1.3269
6	1.3195
7	1.3089
8	1.2733
9	1.0686
10	1.0411

TABLA 8: Resultados de función de afinidad obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 50 % de mutación

El resultado obtenido se muestra de mejor manera realizando una gráfica la cual denote su transición en función al número de iteraciones que realiza el algoritmo. Esto se puede ver con la figura 31 a continuación.

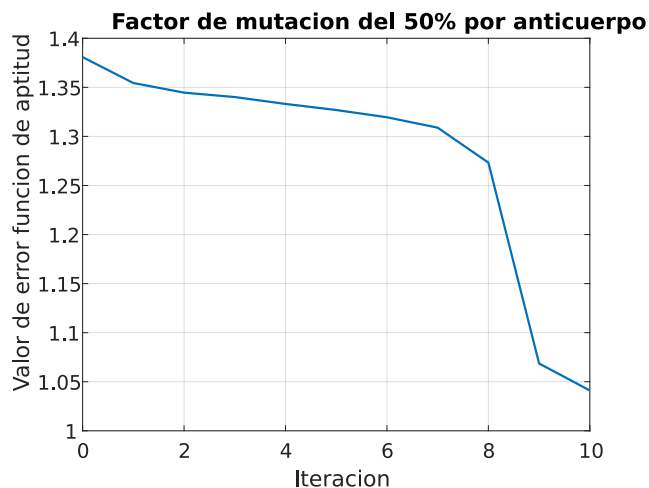


FIGURA 31: Gráfica de afinidad por ciclo de iteración del algoritmo CLONAL
[Autoría propia]

9.1.4.3. Sintonización CLONAL con un 80 % de mutación en Simulink

En esta sección se aplica las mismas condiciones de diseño dadas por el ecuación (9.17) para encontrar los errores que requiere la función de aptitud, pero cambiando el factor de mutación a un 80 % para así obtener los resultados presentados en la tabla 9 .

Iteración	Kp	Ki	Kd	P	Mp (rad)	ts (s)	ess (rad)
0	40.38	0.1	4.7688	50	3.567	1.2	0.00833
1	58.4951	0.04087	5.4473	37.8015	3.398	0.786	0.00833
2	102.4289	0.03494	5.8747	39.7558	3.318	0.605	0.00833
3	186.6575	0.02812	6.0218	55.8180	3.267	0.790	0.00833
4	297.7138	0.0389	6.9475	83.2484	3.228	0.7894	0.00833
5	399.7329	0.0244	9.4675	99.073	3.216	0.769	0.00833
6	547.2380	0.01745	8.8811	86.8860	3.209	0.750	0.00833
7	911.0780	0.0139	15.9914	136.2246	3.190	0.1057	0.00833
8	1626.645	0.01069	21.140	109.7121	3.181	0.1055	0.00833
9	2022.6713	0.0153	26.878	131.237	3.177	0.1057	0.00833
10	2600.206	0.01714	32.2866	137.6418	3.181	0.0994	0.00833

TABLA 9: Resultados obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 80 % de mutacion

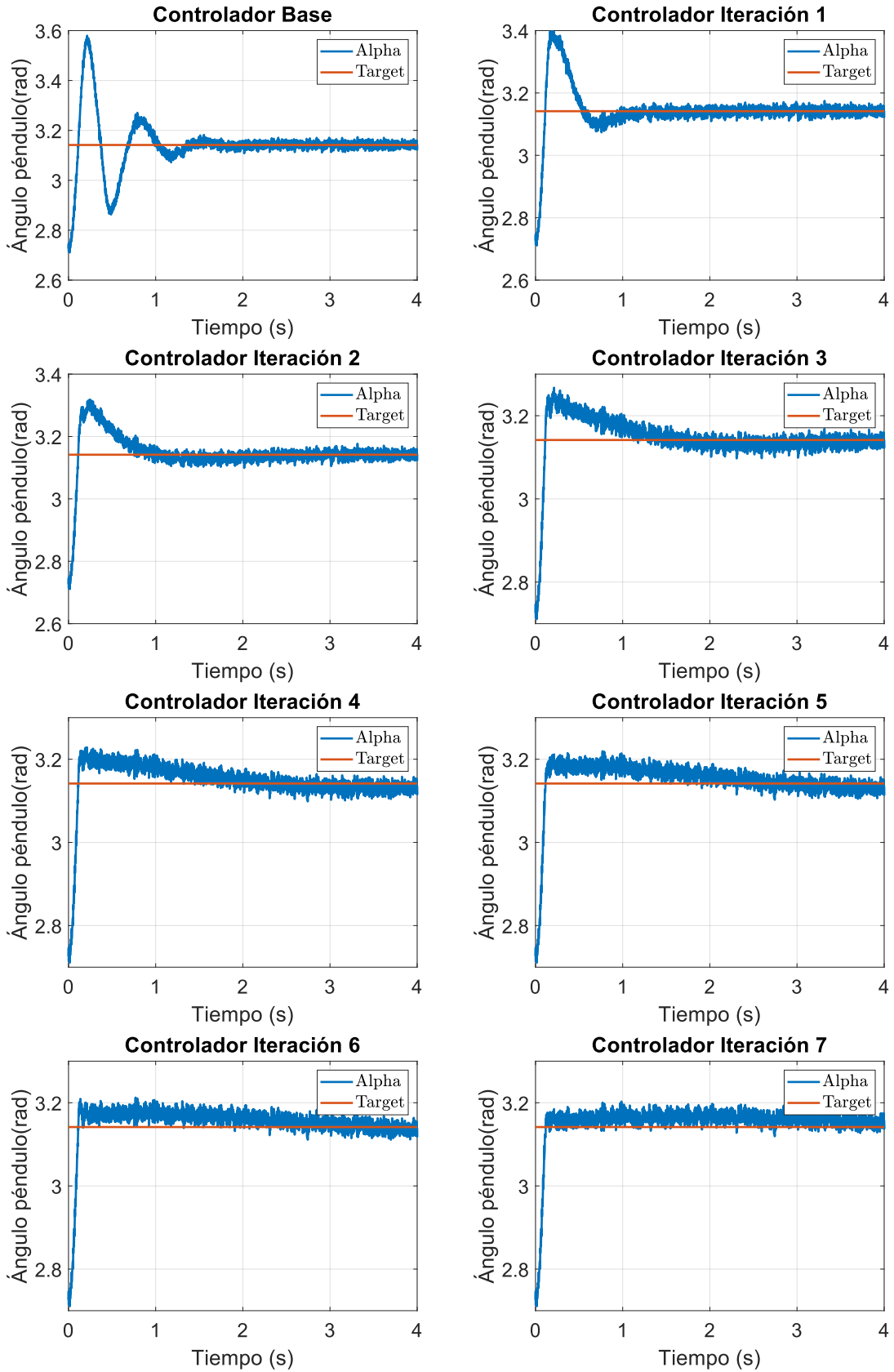


FIGURA 32: Respuesta del controlador de (0-7) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]

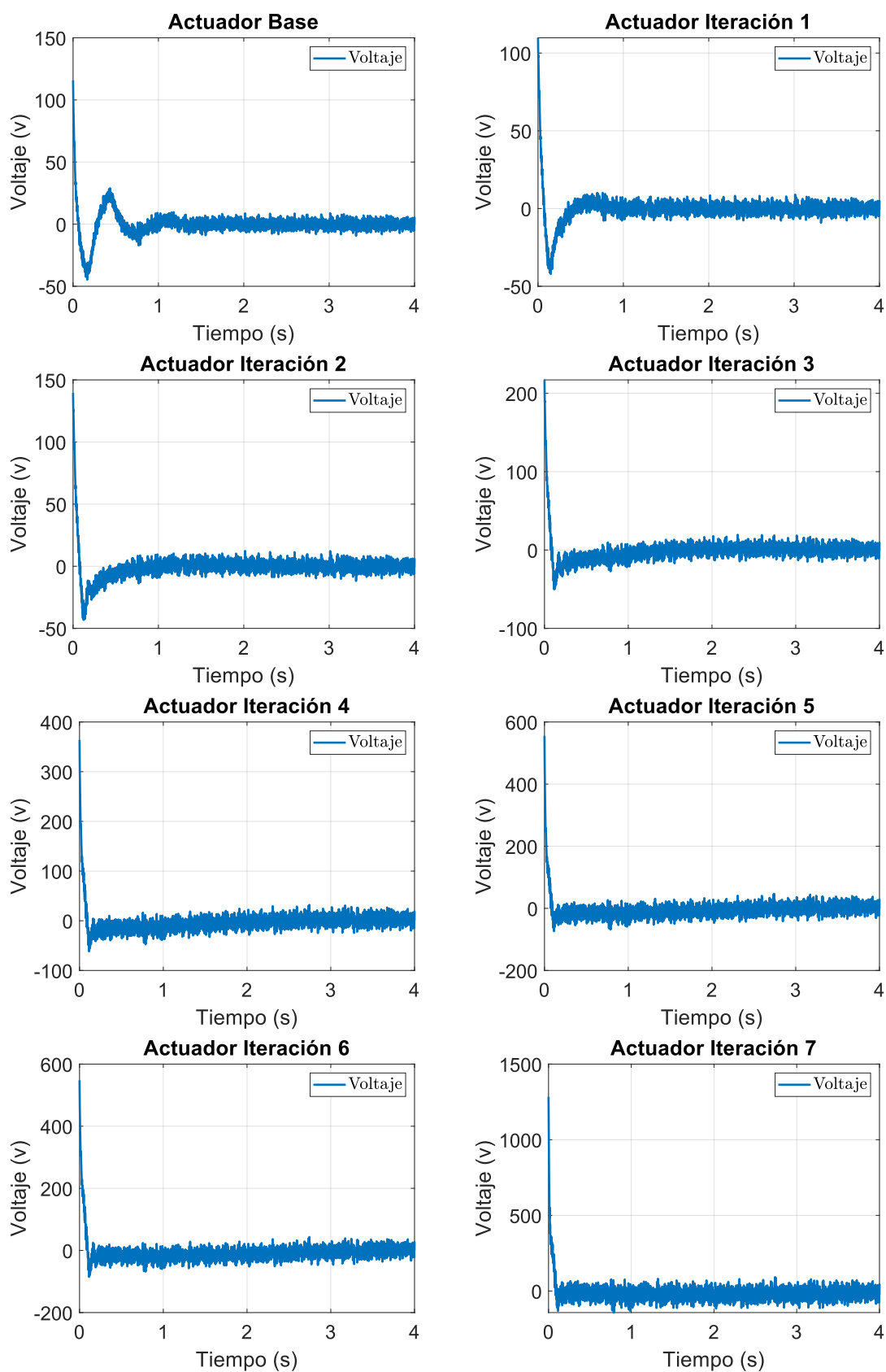


FIGURA 33: Respuesta del actuador de (0-7) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]

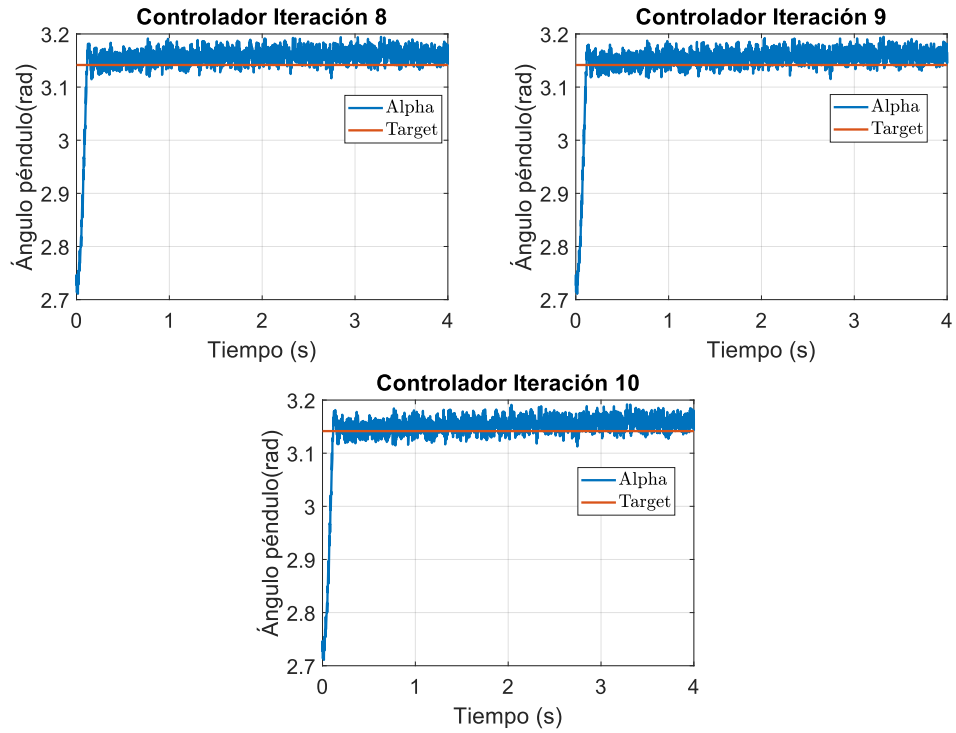


FIGURA 34: Respuesta del controlador de (7-10) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]

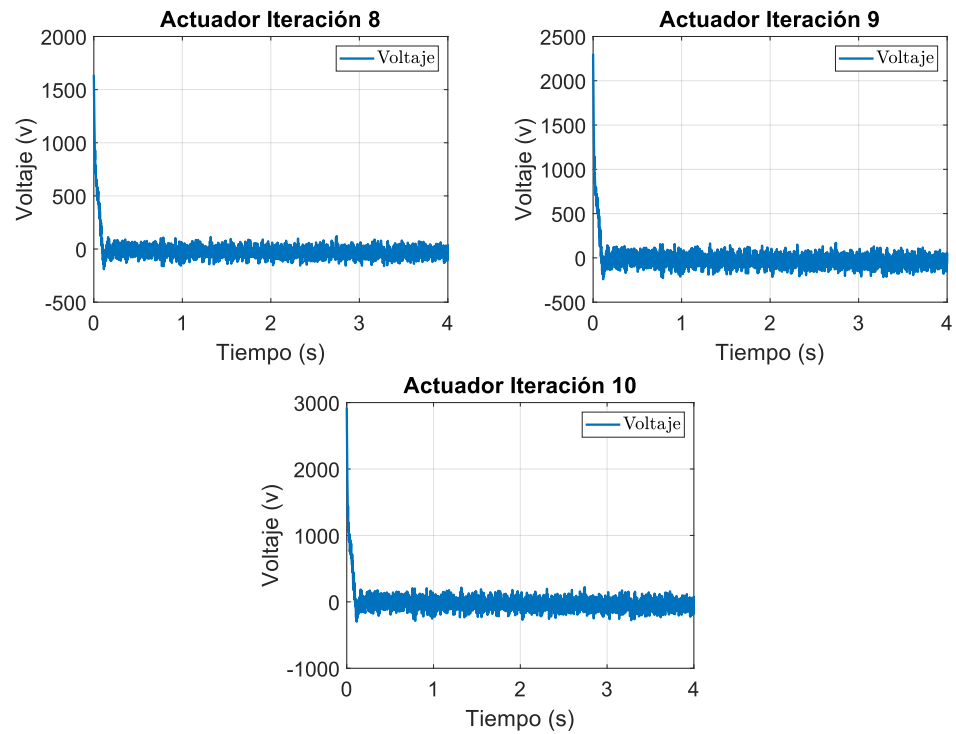


FIGURA 35: Respuesta del actuador de (7-10) iteraciones del algoritmo CLONAL con 80 % de mutación [Autoría propia]

Como se puede observar el algoritmo consigue disminuir considerablemente el sobrepaso máximo y el tiempo de establecimiento en comparación al controlador base diseñado por LGR. Al comparar el desempeño con los resultados obtenidos del algoritmo con un 50 %, se observa que al aumentar la mutación, la sintonización converge con el mismo numero de iteraciones pero mejora el desempeño del tiempo de establecimiento y sobrepaso máximo. Para analizar el desempeño de cada salida de control obtenida por iteración se anexan las anteriores figuras .

Al analizar los resultados obtenidos de las anteriores figuras 32 y 34, se observa que al tener un factor de mutación del 80 por ciento el algoritmo consigue converger mas rápidamente que en los casos anteriores, logrando mejorar considerablemente en la iteración 6 obteniendo valores de sobrepaso máximo de 3.181 radianes, con un tiempo de establecimiento de 0.0994 segundos. Este resultado de tiempo de establecimiento mejora las condiciones de diseño que se asignaron en la ecuación 9.17, donde se asigno un tiempo de establecimiento deseado de 0.1 segundos.

Por ultimo en la tabla 10, se muestra el comportamiento de la función de aptitud u objetivo por cada iteración. Para esto se toman los valores de aptitud que hallan tenido un menor valor en comparación a los 29 anticuerpos creados por cada iteración.

Iteración	Valor de aptitud
0	1.3808
1	1.3577
2	1.3368
3	1.3258
4	1.3093
5	1.293
6	1.0702
7	1.0656
8	1.0351
9	1.0335
10	1.0322

TABLA 10: Resultados de función de afinidad obtenidos de la sintonización durante 10 iteraciones del algoritmo CLONAL con 80 % de mutación

El resultado se muestra de mejor manera realizando una gráfica la cual denote su transición en función al numero de iteraciones que realiza el algoritmo. Esto se puede ver con la figura 36 a continuación.

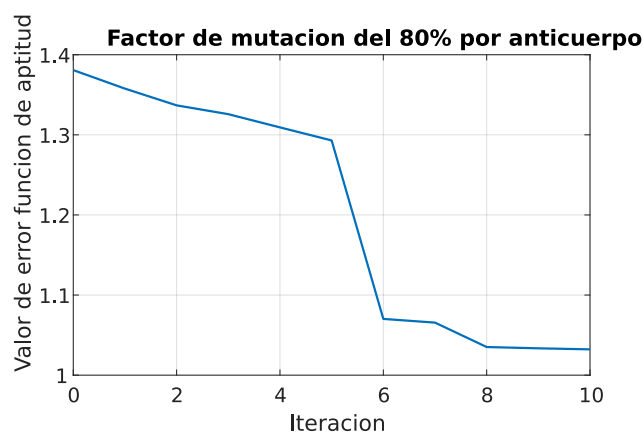


FIGURA 36: Gráfica de afinidad por ciclo de iteración del algoritmo CLONAL [Autoría propia]

9.2. Implementación física del Péndulo Rotatorio Invertido

9.2.1. Implementación del algoritmo CLONAL al péndulo rotatorio invertido

El dispositivo rotativo con péndulo invertido de la marca QUANSER, se constituye principalmente de un motor de corriente directa (C), dos codificadores de alta resolución óptico o encoder (A) y una estación de trabajo que permite la adquisición de datos y la amplificación de potencia con la planta a experimentar (B). El sistema físico implementado se puede observar de mejor forma con la siguiente figura 37.



FIGURA 37: Conexión y implementación física de la planta QUANSER [Autoría propia]

9.2.1.1. Reconocimiento del controlador por defecto propiedad de QUANSER

Se hace un reconocimiento sobre el controlador base que viene por defecto en el software QUARC de simulink con la planta, la cual esta ajustada para realizar una acción de control sobre el péndulo rotatorio invertido, manteniendo una posición vertical o de 180 grados con respecto a su eje. Este control se hace en tiempo real, como se puede evidenciar con la figura 38 a continuación.

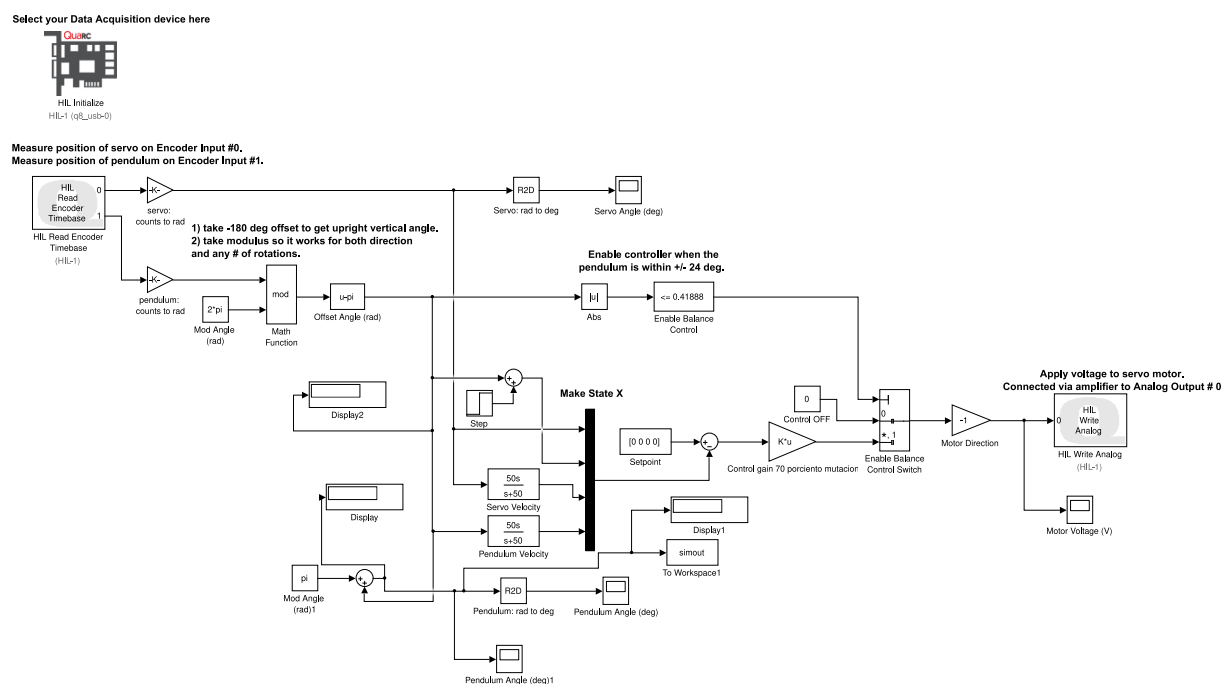


FIGURA 38: Diagrama de control por RVE base que viene por defecto con planta QUANSER [Autoria propia]

Se puede observar un esquema de control por realimentación de variables de estado en donde por medio del lector del encoder HLL se obtiene la medida en tiempo real del ángulo de giro del motor y del péndulo invertido. Posteriormente se hace una conversión del contador que viene en el encoder a una medida coherente en radianes, restando la salida de la multiplexación de las variables de estado y el valor de referencia $[0,0,0,0]$ para obtener el error el cual va ir siendo ajustado por las ganancias del controlador asignadas de base por QUANSER. Por ultimo la matriz resultante se conecta con un interruptor el cual actúa como limitador del ángulo de activación de control ajustando un rango de activación de 24 grados; Es decir, que si el péndulo invertido se encuentra fuera del rango de 156 grados y 236 grados, la acción de control no se activa y no escribe nada sobre el Encoder que controla el motor de giro. Esto se hace con el objetivo de que el péndulo invertido se mantenga en un punto de operación similar a lo simulado en la sección de simulación anterior. Ya teniendo

contextualizado como se ejerce la acción de control sobre el péndulo rotatorio invertido se procede a aplicar el algoritmo de selección clonal sobre las ganancias base que vienen por defecto en el esquema de control propuesto por el fabricante para el presente sistema de péndulo invertido. Para medir el desempeño del algoritmo se le agrega una perturbación utilizando un bloque Step de -0.215 rad a la variable de estado del giro del péndulo α y la cual se activa en el momento en que la simulación física implementada llegue a 24 segundos. Esta respuesta obtenida de la perturbación se utilizará para medir las imágenes de desempeño del sobrepaso máximo, tiempo de establecimiento y error de estado estacionario.

9.2.1.2. Asignación de parámetros usando algoritmo de sintonización clonal sobre el modelo de control del péndulo rotatorio invertido

Primeramente se hace un ajuste al modelo del diagrama de la figura 38 en donde se toman los valores iniciales que tiene el controlador base, para asignar una posición inicial de búsqueda de nuevos parámetros en los cuales el algoritmo clonal pueda basarse. Para calcular las ganancias, primero el manual que proporciona la compañía QUANSER, especifica las condiciones de diseño que tiene el controlador base. En donde asigna un factor de amortiguamiento de 0.7 y una frecuencia natural no amortiguada de 4.

$$\zeta = 0,7$$

$$W_n = 4 \tag{9.18}$$

Posteriormente se hace el cálculo para la ubicación de los polos dominantes sobre el eje real.

$$\sigma = \zeta * W_n \tag{9.19}$$

Luego se realiza el cálculo para los polos dominantes sobre el eje imaginario utilizando la siguiente expresión.

$$W_d = W_n * \sqrt{1 - \zeta^2} \tag{9.20}$$

Después se hace la asignación de los polos deseados en -30 y -40 para el comportamiento del sistema en lazo cerrado.

$$D_P = \begin{bmatrix} -\sigma + j * W_d \\ -\sigma - j * W_d \\ -30 \\ -40 \end{bmatrix} \quad (9.21)$$

Debido a que se tiene un modelo de péndulo invertido con diferentes parámetros físicos en comparación ala anterior sección de simulación, el modelo de variables de estado para este caso de implementación es el siguiente.

$$\begin{bmatrix} \ddot{\Theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 80,3 & -45,80 & -0,930 \\ 0 & 122 & -44,1 & -1,40 \end{bmatrix} \begin{bmatrix} \Theta \\ \alpha \\ \dot{\Theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 83,40 \\ 80,30 \end{bmatrix} V_m \quad (9.22)$$

Por último para el cálculo de las ganancias base se utiliza el comando acker de matlab, el cual permite hacer el calculo automático de un vector de ganancia K. Esta función utiliza el método de Ackerman para trasladar los polos en lazo cerrado a la posición D_p , según los valores obtenidos del vector K [57].

$$K = acker(A, B, D_p) = [-5,2612, 30,0353, -2,6407, 3,5405] \quad (9.23)$$

La ganancias que vienen de base se aplican a cada variable de estado que tiene el péndulo rotatorio invertido las cuales son:

$$Gain_{\alpha} = -5,2612$$

$$Gain_{\theta} = 30,0353 \quad (9.24)$$

$$Gain_{Vel\alpha} = -2,6407$$

$$Gain_{Vel\theta} = 3,5405$$

Con las ganancias iniciales se procede a aplicar el mismo esquema de la figura 20 con los consecuentes pasos que se realizan en el esquema, variando solamente las ganancias iniciales de la variable $N = [Gain_{\alpha}, Gain_{\theta}, Gain_{Vel_{\alpha}}, Gain_{Vel_{\theta}}]$ con las cuales se asigna el espacio de búsqueda inicial que requiere el algoritmo.

9.2.2. Resultados obtenidos en la implementación real del Péndulo Rotatorio Invertido

En esta sección se agregan los resultados obtenidos de la implementación real de un controlador RVE sintonizado en un planta de péndulo invertido en hardware, utilizando el algoritmo de selección clonal

9.2.2.1. Sintonización CLONAL con un 70 % de mutación sobre el modelo físico en QUARC- Simulink

En esta sub-sección se registra los resultados obtenidos de la implementación física sobre la planta de péndulo invertido usando el diagrama de flujo del algoritmo de sintonización CLONAL de la figura 20 y implementándolo en un bloque function de Matlab, el cual ajusta los parámetros de un controlador RVE sobre la planta del péndulo rotatorio invertido. Debido a que el algoritmo para hallar la función de aptitud necesita de unas condiciones de diseño las cuales se requieren para calcular el error para el sobrepaso máximo, tiempo de establecimiento y error de estado estacionario de la ecuación (9.17), se asignan las siguientes condiciones.

$$e_{M_{pN}} = \frac{M_p - Valor_{estable}}{Valor_{estable}}$$

$$e_{t_{sN}} = \frac{0,1 - t_s}{t_s} \quad (9.25)$$

$$e_{e_{ssN}} = abs(\frac{Rango_{e_{ss}} - e_{ss}}{e_{ss}})$$

Como se observa en las ecuaciones (9.25) se hace un cambio de asignación de condiciones de diseño con respecto a la sintonización de la anterior sección, en esta debido a que los valores finales que reconoce el encoder de la planta para el ángulo del péndulo invertido, varían en función a las condiciones iniciales que el usuario físicamente asigne para su control, lo cual lo

hace impreciso y no permite replicar los mismos resultados a medida de que se prueba cada controlador o antígeno que asigne el algoritmo para la solución. Por esto para conocer el sobrepaso máximo $e_{M_{PN}}$, se utiliza el valor de estado estable final al cual se llegue después de que ocurra la perturbación. También para obtener el e_{ess_N} se realiza otro cambio para obtener el error de estado estacionario, ya que el rango de error varia también debido al cambio del valor final de establecimiento, para esto se utiliza el rango de error de estado estacionario variable, en donde en función al valor final obtenido, este vaya cambiando y ajustándose automáticamente. Debido a que se llegan a producir cambios que pueden dar un valor de error estado estacionario negativo, se toma solamente el valor absoluto que es la magnitud del rango del error que se requiere. Ya conociendo los errores que requiere la función de aptitud, el algoritmo se encarga de encontrar las ganancias que necesita el controlador para su estabilización. Al realizar 1 iteración del algoritmo de selección CLONAL siguiendo la lógica de programación de la figura 20, se obtiene la tabla 11 a continuación .

Iteración	Gain $_{\alpha}$	Gain $_{\theta}$	Gain $_{Vel\alpha}$	Gain $_{Vel\theta}$	Mp (rad)	ts (s)	ess (rad)
0	-5.2612	30.0353	-2.6407	3.5405	3.2336	2.4	0.01386
1	-3.9168	32.01964	-3.4392	4.5917	3.1952	1.15	0.0061945

TABLA 11: Resultados obtenidos de la sintonización de la planta durante 1 iteración del algoritmo CLONAL con 70 % de mutación

A partir de la tabla 11 se puede observar los resultados obtenidos de la respuesta del sistema de control RVE del péndulo invertido de la planta QUANSER. Se analiza que con solo con una iteración del algoritmo, se consigue mejorar la métrica del sobrepaso máximo con un valor de 3.19 radianes y una reducción del tiempo de estabilización de 1.15 segundos. Por ultimo y como característica mas notable, se consigue reducir el error de estado estacionario de la respuesta del sistema con el algoritmo sintonizado, reduciéndose a 0.00619 radianes con respecto a la señal de referencia de 3.14 radianes que representa el valor estabilización de 180 grados. Para visualizar estos resultados de mejor forma se anexa la siguiente figura 39 de la respuesta del controlador por defecto de la QUANSER y el sintonizado por el algoritmo CLONAL.

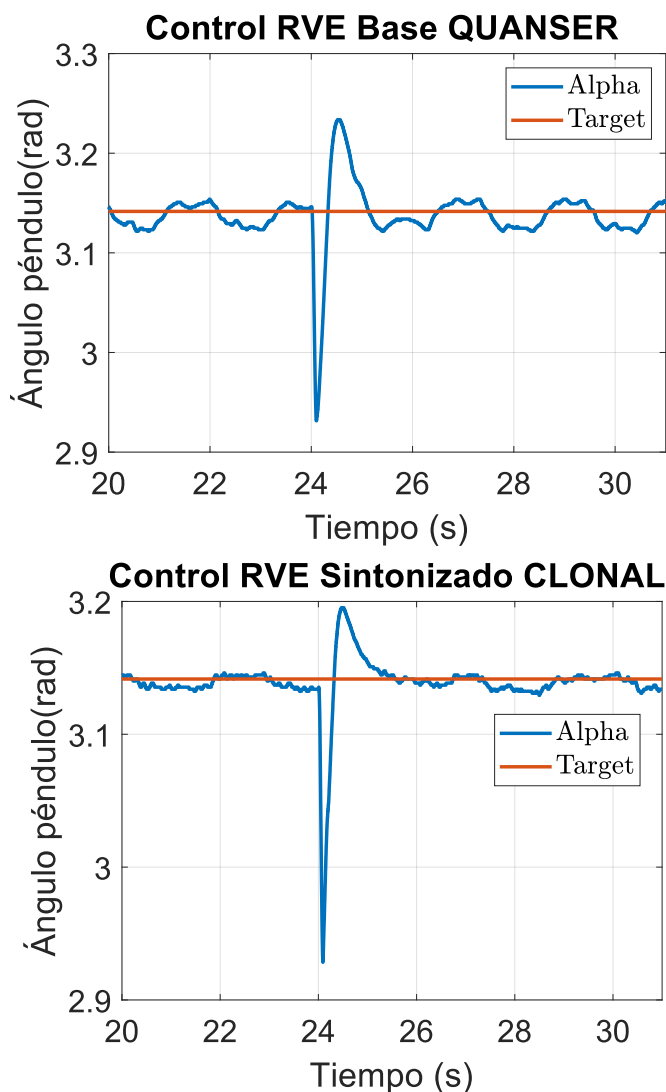


FIGURA 39: Respuestas del controlador por defecto y control resultado del algoritmo con factor de mutación del 70 % [Autoría propia]

Como se puede observar, el algoritmo consigue disminuir el sobrepaso máximo y el tiempo de establecimiento en comparación al controlador base diseñado por RVE por la compañía creadora. También el controlador consigue mejorar el error de estado estacionario en comparación al que viene por defecto en el dispositivo, aproximándose mas 3.14 , que es el equivalente en radianes a 180 o posición vertical del péndulo invertido. Para comprobar lo anteriormente mencionado se procede a hacer el calculo del error medio de los datos obtenidos del control con ganancias por defecto de la planta y el control resultado obtenido del algoritmo con un factor de mutación del 70 %. Para esto se hace uso de la siguiente ecuación.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Valor_{predicho} - Valor_{actual})^2}{N}} \quad (9.26)$$

Ya con la anterior ecuación se procede a hacer el calculo del error medio sobre los datos de amplitud que tienen ambas señales. Esto se anexa en la tabla 12 de a continuación .

Tipo	Numero de datos (N)	RMSE
Controlador Base	5501	0.02936
Controlador Sintonizado	5501	0.02202

TABLA 12: Comparación de resultados del error cuadrático medio de estado estacionario del controlador base y el controlador sintonizado con algoritmo Clonal

Se procede a agregar una tabla que muestra el comportamiento de la función de aptitud u objetivo por cada iteración. Para esto se toman los valores de aptitud que hallan tenido un menor valor en comparación a los 29 anticuerpos creados por cada iteración.

Iteración	Valor de aptitud
0	1.1426
1	1.1148

TABLA 13: Resultados de función de afinidad obtenidos de la sintonización durante 1 iteración del algoritmo CLONAL con 70 % de mutación en planta física QUANSER

Por último se comprueba el voltaje del actuador empleada para ambos controladores; Esto con el objetivo de identificar si existe una mejora en relación al voltaje consumido por la planta. Esto se observa en la siguiente figura 40.

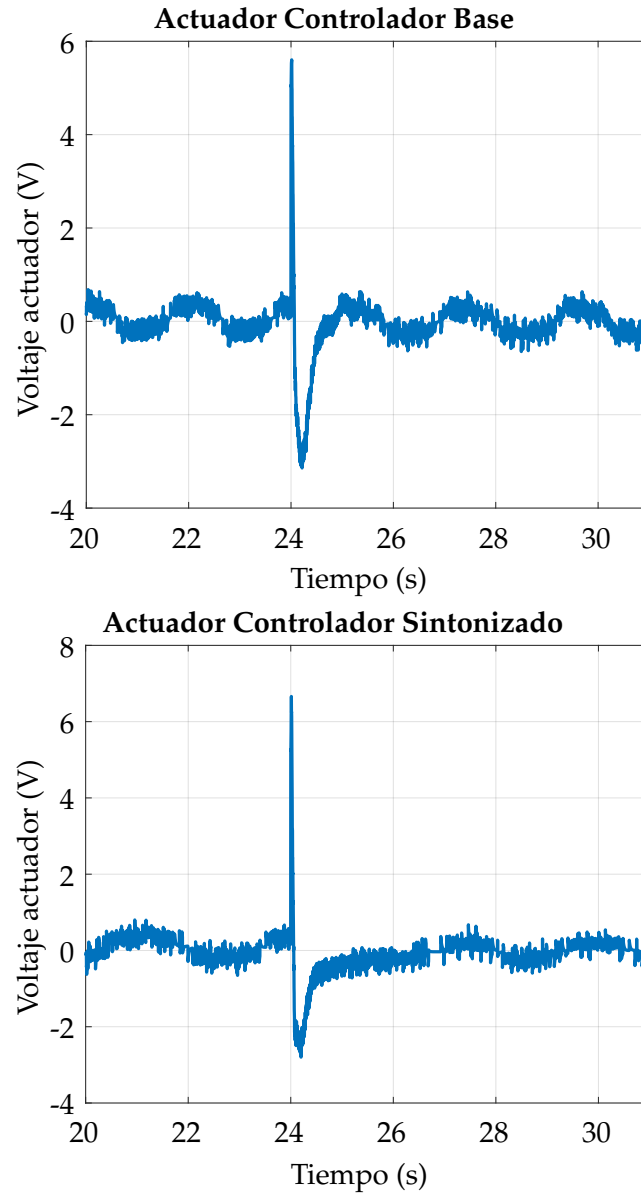


FIGURA 40: Respuestas del actuador del controlador por defecto y control resultado del algoritmo CLONAL con factor de mutación del 70 % [Autoría propia]

Para evaluar cuanto voltaje de media consume ambos controladores se aplica la siguiente ecuación para calcular la tensión del actuador.

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (9.27)$$

Como se puede observar, el cálculo considera cada valor de tensión el cual va ir sumando hasta el número límite de datos que se tenga, luego ya teniendo la suma de todos los datos se divide el resultado por el número total de datos sumados. El resultado obtenido se presenta en la tabla 14 .

Tipo	Numero de datos (N)	Valor medio
Controlador Base	5501	0.3027V
Controlador Sintonizado	5501	0.2820V

TABLA 14: Comparación de resultados del valor medio del voltaje del actuador proveniente del controlador base y el controlador sintonizado con un factor de mutación del 70 %

Como se puede observar, el controlador sintonizado con el algoritmo presenta un menor valor medio total de tensión del actuador, lo cual denota que la sintonización también permite reducir el consumo de voltaje en el sistema de control.

9.2.3. Pruebas del controlador base y controlador sintonizado con diferentes valores de perturbación

En esta sección se hacen varios experimentos con el controlador base y el controlador sintonizado variando la perturbación realizada en la figura 39 en un 20, 50 y 80 por ciento del valor utilizado en la misma. Para esto, primero se agregan los resultados obtenidos con un 20 % de la perturbación para el controlador base y el controlador sintonizado.

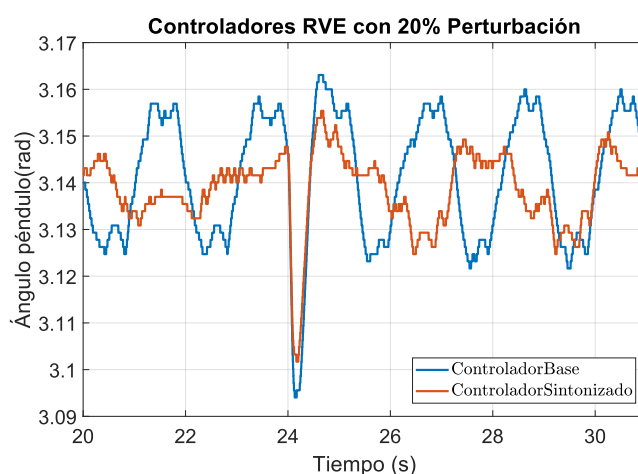


FIGURA 41: Respuestas del controlador base y el control sintonizado resultado del algoritmo CLONAL con 20 % de perturbación [Autoría propia]

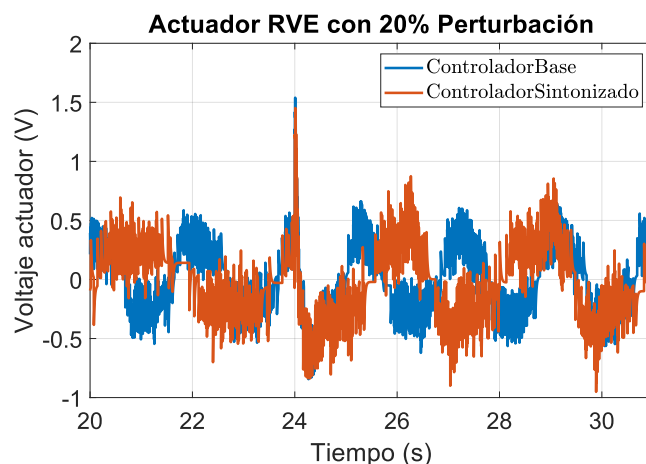


FIGURA 42: Respuestas del actuador base y el actuador sintonizado resultado del algoritmo CLONAL con 20 % de perturbación [Autoría propia]

Al observar los resultados obtenidos de las figuras 41 y 42 se analiza que con una perturbación del 20 por ciento ambos controladores presentan desempeños similares dentro del sistema de control, como en la respuesta del actuador. Al observar los resultados obtenidos de las figuras 41 se detalla una leve mejora del controlador sintonizado al momento de evaluar el sobrepaso máximo obteniendo un valor menor que 3.16 radianes.

Posteriormente se realiza un cambio a la perturbación agregada al sistema de un 50 por ciento y se obtiene los siguientes resultados.

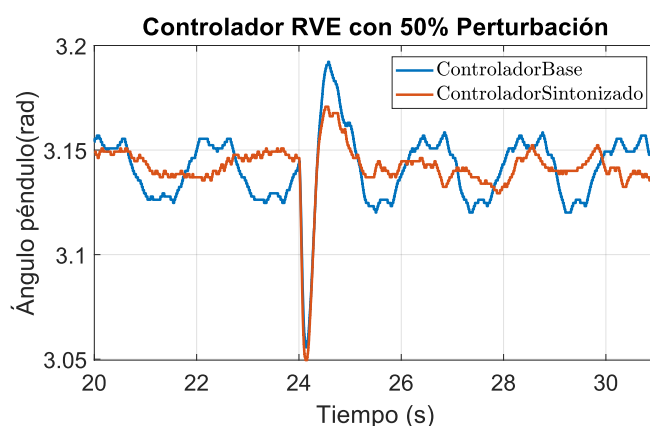


FIGURA 43: Respuestas del controlador base y el control sintonizado resultado del algoritmo CLONAL con 50 % de perturbación [Autoria propia]

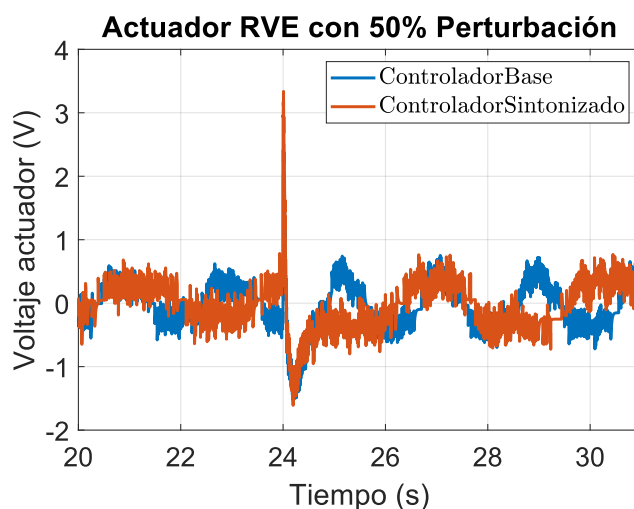


FIGURA 44: Respuestas del actuador base y el actuador sintonizado resultado del algoritmo CLONAL con 50 % de perturbación [Autoría propia]

Al observar los resultados obtenidos de las figuras 43 y 44 se analiza que con una perturbación del 50 por ciento el controlador sintonizado presenta mejor desempeño que el controlador base. El controlador base presenta un sobrepaso máximo de 3.19 radianes, siendo mayor que en comparación al controlador sintonizado con un sobrepaso máximo de 3.17 radianes. También se observa que el controlador sintonizado presenta menor error de estado estacionario que el controlador base RVE.

Posteriormente se realiza un cambio a la perturbación agregada al sistema de un 80 por ciento y se obtiene los siguientes resultados.

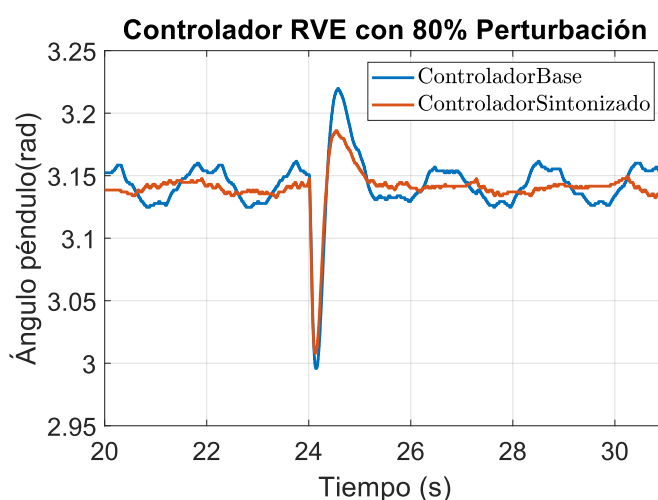


FIGURA 45: Respuestas del controlador base y el control sintonizado resultado del algoritmo CLONAL con 80 % de perturbación [Autoría propia]

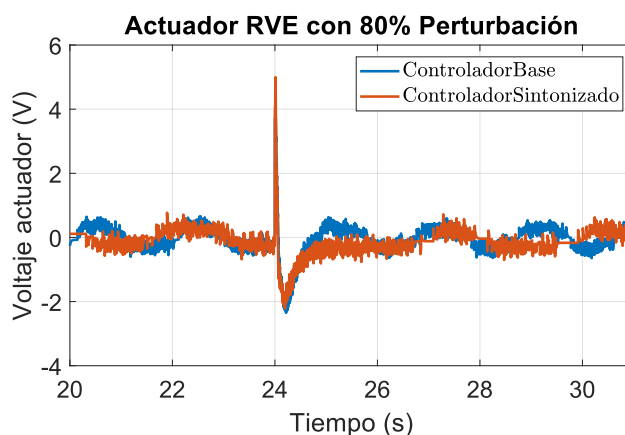


FIGURA 46: Respuestas del actuador base y el actuador sintonizado resultado del algoritmo CLONAL con 80 % de perturbación [Autoría propia]

Al observar los resultados obtenidos de las figuras 45 y 46 se analiza que con una perturbación del 80 por ciento el controlador sintonizado presenta mejor desempeño que el controlador base. El controlador base presenta un sobrepaso máximo de 3.21 radianes, siendo mayor que en comparación al controlador sintonizado con un sobrepaso máximo de 3.18 radianes. También se observa que el controlador sintonizado presenta menor error de estado estacionario que el controlador base RVE.

9.3. Discusión

Se simuló un controlador PID basado en un algoritmo bio-inspirado (Selección CLONAL) que permite hacer la sintonización de los parámetros del Sobrepaso Máximo (M_p), Tiempo de Establecimiento (t_s) y Error de Estado Estacionario (e_{ss}) del ángulo del péndulo rotatorio invertido (α). Esto se hace a partir de un modelo linealizado en donde se realizó un sistema de control propio por el método de LGR y se ajustó particularmente utilizando el algoritmo de selección CLONAL en un entorno de simulación en Matlab-Simulink.

El controlador diseñado por LGR presentó un sobrepaso máximo del 11.87 % , con un tiempo de establecimiento de 1.2 segundos y error de estado estacionario de 0.0083 radianes. Al momento de aplicar el algoritmo de sintonización variando el porcentaje de mutación, se concluye que el que mejor desempeño obtuvo fue el controlador obtenido con un porcentaje de mutación del 80 % con unas imágenes de merito de sobrepaso máximo del 1.28 % , tiempo de establecimiento de 0.0994 segundos y error de estado estacionario de 0.0083 radianes. Se

resalta que las condiciones de diseño que se asignaron fueron: un sobrepaso máximo cercano al 0 %, un tiempo de establecimiento cercano al 0.1 segundos y un error de estado estacionario 0.01 y que se evidencia en la ecuación (9.17). Estos resultados se obtuvieron por medio de un modelo lineal conformado por las ecuaciones (7.15) y (7.21), esto significa que mientras se cumpla el rango de operación inicial para las variables de estado de $[\alpha = 0, \theta = 0, \dot{\alpha} = 0, \dot{\theta} = 0]$ el control funcionaria correctamente.

Al comparar los resultados simulados obtenidos con otros algoritmo que integran técnicas de inteligencia artificial como es el algoritmo de búsqueda Cuckoo [17], se observa que en este se obtuvo un resultado de sobrepaso máximo del 5.50 % , tiempo de establecimiento de 0.4 segundos y error de estado estacionario de 0.006 radianes, después de que el algoritmo converja en 20 generaciones o iteraciones. Acá se observa que algoritmo sintonizado por la técnica de selección clonal con un porcentaje de mutación del 80 %, presentó un mejor comportamiento transitivo que el controlador PID desarrollado con el algoritmo de búsqueda Cuckoo, ya que como se puede observar en la figura 36 , donde se evalúa la afinidad de el algoritmo, se consigue converger en 10 iteraciones o generaciones menos que el algoritmo Cuckoo. También el sobrepaso máximo y tiempo de establecimiento que se consigue con la técnica CLONAL presentan mejor desempeño.

Por ultimo, para comprobar que tan practico es el algoritmo de sintonización CLONAL en un entorno físico en donde no se tienen condiciones iniciales ideales, se evalúan los resultados obtenidos de la sintonización de un controlador por RVE desarrollado por la empresa QUANSER . Al asignar al algoritmo un porcentaje de mutación del 70 % , este consigue converger en solo una iteración logrando disminuir la función de aptitud u objetivo de la tabla 11 para conseguir los resultados de la figura (39). Estos resultados se evidencia que al aplicar el proceso de sintonización clonal se consiguió disminuir el sobrepaso máximo a un valor pico del 1.72 % , tiempo de establecimiento de 1.15 segundos y error de estado estacionario de 0.0061 radianes. Por otro lado, el controlador que viene por defecto en la QUANSER presentó un desempeño de sobrepaso máximo del 2.89 %, tiempo de establecimiento de 2.4 segundos y un error mayor de 0.01386 radianes. Para comprobar el desempeño que tiene el error de estado estacionario en su estado estable, se realiza un calculo del error cuadrático medio de todos los datos de amplitud en su estado estacionario para así tener un indicativo de cuanto en media, varia el error de estado estacionario en el tiempo para ambos controladores. En la tabla 12 se evidencia que el controlador sintonizado obtuvo un valor RMSE menor que el controlador base RVE (Realimentación de Variables de Estado) de propiedad QUANSER. Esto significa que el péndulo rotatorio invertido en su estado estable vertical oscila menor con el

controlador sintonizado y lo hace mas preciso al valor de referencia asignado que es el equivalente en radianes de 180 grados, es decir 3.14159 radianes. Paralelamente, al analizar los resultados obtenidos en la figura 40, se observa que el controlador sintonizado presenta un pico de voltaje mayor, en comparación al controlador base. Pero, al hacer el cálculo del valor medio de voltaje anexado en la tabla 14, el controlador sintonizado consume en promedio menos voltaje que el controlador base; Esto permite agregar a los beneficios de la sintonización del controlador sintonizado, una reducción de la energía consumida en la planta a medida que ejerce la acción de control sobre el péndulo invertido.

Capítulo 10

Conclusiones

Se puede concluir a partir de la tabla 2 que los tipos de algoritmos Bioinspirados dentro de la documentación consultada en el estado de arte, presentan mas metodologías de sintonización con controladores para péndulos invertidos que las demás técnicas consultadas. Para la selección de la técnica dentro de los tipos de algoritmos bioinspirados, se realiza una comparación de las métricas resultantes de la tabla 3, y se concluye que el algoritmo clonal obtiene mejores métricas de desempeño en un número de iteraciones menor que las demás técnicas en la tabla.

Se puede concluir a partir de los resultados obtenidos en las tablas 6, 8 y 10 que el controlador sintonizado con mejor desempeño para la sección simulada en MATLAB, resultó ser el controlador con un factor de mutación del 80 %, ya que este obtuvo un valor asociado de función objetivo o de aptitud menor que los otros resultados obtenidos con porcentajes de mutación del 50% y 20 %. Debido a que la función de aptitud contiene los valores de la métricas del sobrepaso máximo, tiempo de establecimiento y error de estado estacionario permite intuir que entre mas pequeño resulte el valor de la función de aptitud, menor valor tendrán estas métricas asociadas.

En el desarrollo de este proyecto se exploró las grandes posibilidades que tiene la implementación de algoritmos de aprendizaje de máquina en procesos de control sobre plantas no lineales. Debido a que estos algoritmos permiten aplicar acciones de control sin la necesidad de basarse enteramente en el modelamiento de la planta; Si no que utilizan información particular que pueda representar una métrica de desempeño, siendo en este caso del proyecto la respuesta de control que tiene el péndulo invertido al momento de evaluar el sobrepaso máximo, tiempo de establecimiento y error de estado estacionario. Al utilizar estas

métricas, un algoritmo como puede ser el aplicado en este proyecto de selección CLONAL alrededor de un punto de operación, se puede ajustar a la problemática que se desee solucionar, siendo en este caso la sintonización del controlador que actúa sobre el péndulo rotatorio invertido.

Al considerar los resultados obtenidos en la aplicación del algoritmo de selección CLONAL sobre el modelo del péndulo rotatorio invertido en un entorno de simulación, se puede concluir que al expandir el espacio de búsqueda de los anticuerpos y aumentar el porcentaje de mutación, se aumenta la posibilidad de caer en una respuesta que represente un mínimo local; Es decir, que alcanza a converger mas rápidamente, pero los anticuerpos se estancan en un punto en el cual ya no van a mejorar más. Esto se observa de mejor forma con los resultados obtenidos en la figuras 32 y 34 , en donde el algoritmo cae en un mínimo local después de 10 iteraciones como se puede ver en la figura 36. Una ventaja que tiene el algoritmo de selección CLONAL con respecto a este tema, es que tiene un factor de mutación variable en función a la afinidad que tiene cada anticuerpo; Es decir, que a medida que un anticuerpo tenga una baja afinidad, el algoritmo automáticamente asignará un valor mayor de mutación a este antígeno, para que tenga mas posibilidades de encontrar un punto que aumente la afinidad. En este proyecto se manejó un factor de mutación constante para poder realizar varias pruebas y obtener diferentes resultados a medida de que se aumentaba la mutación.

Al considerar los resultados obtenidos en la aplicación del algoritmo de selección CLONAL sobre el modelo del péndulo rotatorio invertido en un entorno de implementación física, se puede concluir que el algoritmo de selección CLONAL permite obtener ganancias del controlador que se desempeñen de mejor forma, al obtener mejores resultados de sobrepaso máximo , tiempo de establecimiento y error de estado estacionario que el controlador por defecto que venga en un dispositivo. Esto se puede destacar en los resultados obtenidos en la figura 39, en donde el algoritmo CLONAL presenta un menor sobrepaso máximo, tiempo de establecimiento y error de estado estacionario que el controlador base creado por QUANSER. También al evaluar el error cuadrático medio de ambos resultados, se puede observar un menor error por parte del controlador por algoritmo CLONAL, el cual se evidencia en la tabla 12. Aun así, los resultados pueden llegar a ser falibles al momento de volver a evaluar los mismos anticuerpos obtenidos del proceso clonación. Esto debido a que es muy complicado asignar las mismas condiciones iniciales que tenía un anticuerpo al momento de probarlo, ya que puede haber variado algún parámetro físico dentro de la planta por elementos externos, como puede ser la temperatura, elementos desgastados dentro de la planta

o fallos en la manipulación del operario que maneje la planta. Por esta razón se recomienda realizar el proceso de sintonización en un ambiente controlado de laboratorio, en donde no se presenten incertidumbres que puedan afectar los resultados de las imágenes de merito del controlador asignado.

Capítulo 11

Trabajos Futuros

Como trabajos futuros se plantea mejorar el algoritmo de selección clonal, asignando un factor de mutación variable en función a la afinidad resultante después de cada iteración. Esto permitiría que los anticuerpos que presenten menor afinidad tengan la posibilidad de aumentar su espacio de búsqueda y tener mas posibilidades de converger a un mejor resultado. También se plantea desarrollar diferentes aplicaciones del algoritmo sobre otro tipo de plantas que presenten comportamientos no lineales y que se requiera de alternativas de control moderno utilizando técnicas de Inteligencia Artificial como la utilizada en este proyecto de grado.

Bibliografía

- [1] O. Chuk y C Medina. «Transferencia tecnológica de un sistema de control automático basado en PC para un molino de martillos industrial». En: *Facultad de ingeniería, Universidad Nacional de San Juan. San Juan, Argentina*. 2014.
- [2] Marco Paz Ramos y Suselle Esquivel. «Un controlador bien sintonizado es dinero en su bolsillo». En: *InTech México Automatización*. Vol. 9. 2010, págs. 30-34.
- [3] Ishan Chawla y Ashish Singla. «Real-Time Control of a Rotary Inverted Pendulum using Robust LQR-based ANFIS Controller». En: *International Journal of Nonlinear Sciences and Numerical Simulation* 19.3-4 (2018), págs. 379-389. DOI: doi:10.1515/ijnsns-2017-0139. URL: <https://doi.org/10.1515/ijnsns-2017-0139>.
- [4] Ishan Chawla y Ashish Singla. «ANFIS based system identification of underactuated systems». En: *International Journal of Nonlinear Sciences and Numerical Simulation* 21.7-8 (2020), págs. 649-660. DOI: 10.1515/ijnsns-2018-0005. URL: <https://doi.org/10.1515/ijnsns-2018-0005>.
- [5] Wael Younis y Mohammed Abdelati. «Design and implementation of an experimental segway model». En: *AIP Conference Proceedings*. Vol. 1107. American Institute of Physics, 2009. Cap. 1, págs. 350-354.
- [6] Ahmed Elhasairi y Alexandre Pechev. «Humanoid Robot Balance Control Using the Spherical Inverted Pendulum Mode». En: *Frontiers in Robotics and AI* 2 (2015). DOI: 10.3389/frobt.2015.00021.
- [7] Ishan Chawla y Ashish Singla. «Real-Time Stabilization Control of a Rotary Inverted Pendulum Using LQR-Based Sliding Mode Controller». En: *Arabian Journal for Science and Engineering* 46.3 (2021), págs. 2589-2596. DOI: 10.1007/s13369-020-05161-7. URL: <https://doi.org/10.1007/s13369-020-05161-7>.

-
- [8] M Akhtaruzzaman y A A Shafie. «Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis». En: - *2010 IEEE International Conference on Mechatronics and Automation*. 2010, págs. 1342-1347. ISBN: 2152-7431. DOI: 10.1109/ICMA.2010.5589450.
 - [9] G Sainzaya y col. «LQR control with refined PID to balance rotary inverted pendulum with time-varying uncertainty». En: - *2017 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*. 2017, págs. 1-6. ISBN: 2377-5831. DOI: 10.1109/iFUZZY.2017.8311812.
 - [10] Javier Gonzalo González Fontanet, Ania Lussón Cervantes e Irina Bausa Ortiz. «Alternativas de control para un Péndulo de Furuta». En: *Revista Iberoamericana de Automática e Informática industrial* 13, Núm. 4 (). DOI: 10.1016/j.riai.2016.05.008. URL: <https://polipapers.upv.es/index.php/RIAI/article/view/9255>.
 - [11] S Howimanporn, S Chookaew y C Silawatchananai. «Comparison between PID and Sliding Mode Controllers for Rotary Inverted Pendulum Using PLC». En: - *2020 4th International Conference on Automation, Control and Robots (ICACR)*. 2020, págs. 122-126. ISBN: NULL-. DOI: 10.1109/ICACR51161.2020.9265510.
 - [12] Gisela Pujol-Vazquez y col. «Rotary inverted pendulum with magnetically external perturbations as a source of the pendulum's base navigation commands». En: *Journal of the Franklin Institute* 355.10 (2018), págs. 4077-4096. URL: <https://www-sciencedirect-com.crai-ustadigital.usantotomas.edu.co/science/article/pii/S0016003218302151>.
 - [13] V Stanovov, S Akhmedova y E Semenkin. «Automatic Design of Fuzzy Controller for Rotary Inverted Pendulum with Success-History Adaptive Genetic Algorithm». En: - *2019 International Conference on Information Technologies (InfoTech)*. 2019, págs. 1-4. ISBN: NULL-. DOI: 10.1109/InfoTech.2019.8860874.
 - [14] G Yuntao y col. «The Design and Realization of a Rotary Inverted Pendulum Based on STM32». En: - *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*. 2015, págs. 185-188. ISBN: NULL-. DOI: 10.1109/IIKI.2015.46.
 - [15] J Kim y col. «Imitation Reinforcement Learning-Based Remote Rotary Inverted Pendulum Control in OpenFlow Network». En: *IEEE Access*. Vol. 7. 2019, págs. 36682-36690. ISBN: 2169-3536. DOI: 10.1109/ACCESS.2019.2905621.

-
- [16] H Lim y col. «Federated Reinforcement Learning for Controlling Multiple Rotary Inverted Pendulums in Edge Computing Environments». En: - 2020 *International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. 2020, págs. 463-464. ISBN: NULL-. DOI: 10.1109/ICAIIIC48513.2020.9065233.
- [17] C Sompracha y S Rukkaphan. «Optimal PIAD μ Controller Design for Rotary Inverted Pendulum System via Cuckoo Search Algorithm». En: - 2020 *17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. 2020, págs. 490-493. ISBN: NULL-. DOI: 10.1109/ECTI-CON49241.2020.9158293.
- [18] David Mateo Rojas Vallejo. «Desarrollo de una estrategia de control predictivo aplicado a un péndulo invertido, que permita la estabilización de imágenes en robots para exploración agrícola». En: *Universidad de La Salle. Facultad de Ingeniería. Ingeniería en Automatización*. 2019.
- [19] Mr Kulkarni. «Self Balancing Robot». En: *International Journal for Research in Applied Science and Engineering Technology* 7 (2019), págs. 814-818. DOI: 10.22214/ijraset.2019.6141.
- [20] Philippe Martin, Santosh Devasia y Brad Paden. «A different look at output tracking: control of a vtol aircraft». En: *Automatica* 32.1 (1996), págs. 101-107. DOI: [https://doi.org/10.1016/0005-1098\(95\)00099-2](https://doi.org/10.1016/0005-1098(95)00099-2). URL: <https://www.sciencedirect.com/science/article/pii/0005109895000992>.
- [21] Daniel Jerome Block, Karl J Åström y Mark W Spong. *Reaction wheel pendulum*. San Rafael, Calif.]: Morgan & Claypool, 2007. ISBN: 1598291947 9781598291940 9781598294439 1598294431.
- [22] EMANUEL OCHOA HENAO. «ODS número 9 industria, innovación e infraestructura y número 17 alianzas para lograr los objetivos y su efecto en las PYMES para Colombia». En: *UNIVERSITARIA ESUMER*. 2019, págs. 1-56. URL: <http://repositorio.esumer.edu.co/jspui/handle/esumer/2025>.
- [23] Thomas Kailath. *Linear systems*. Englewood Cliffs, N.J.: Prentice-Hall, 1980. ISBN: 0135369614 9780135369616.
- [24] J K Roberge. «The Mechanical Seal, Bachelor's thesis». En: *Massachusetts Institute Of Technology*. Vol. 1. 1960, págs. 0-48. URL: <http://web.mit.edu/klund/www/papers/Roberge1960.pdf>.

-
- [25] J F Schaefer y col. *On the bounded control of some unstable mechanical systems*. Stanford, Calif.; Virginia: Dept. of Aeronautics y Astronautics, Stanford University ; Distributed by the Defense Documentation Center, 1965.
- [26] William Siebert. *Circuits, signals, and systems*. Cambridge, Mass.; New York: MIT Press ; McGraw-Hill, 1985. ISBN: 0070572909 9780070572904 0262192292 9780262192293.
- [27] Huibert Kwakernaak y Raphael Sivan. *Linear optimal control systems*. Vol. 1. Wiley-interscience New York, 1972.
- [28] H K Khalil. *Nonlinear Systems*. Prentice Hall, 2002. ISBN: 9780131227408. URL: https://books.google.com.co/books?id=v_BjPQAACAAJ.
- [29] Navin John Mathew, K Koteswara Rao y N Sivakumaran. «Swing Up and Stabilization Control of a Rotary Inverted Pendulum». En: *IFAC Proceedings Volumes* 46.32 (2013), págs. 654-659. DOI: <https://doi.org/10.3182/20131218-3-IN-2045.00128>. URL: <https://www.sciencedirect.com/science/article/pii/S1474667015383324>.
- [30] Sandeep Pandey, Prakash Dwivedi y Deepak Mishra. «Performance Evaluation : Anti-windup Two-loop PID Controller for Rotary Single Inverted Pendulum». En: *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. 2021, págs. 1-6. DOI: 10.1109/UPCON52273.2021.9667579.
- [31] Adharsh Lal M y col. «Stabilization of Rotary Inverted Pendulum using PID Controller». En: *2021 8th International Conference on Smart Computing and Communications (ICSCC)*. 2021, págs. 376-380. DOI: 10.1109/ICSCC51209.2021.9528290.
- [32] M Siva Kumar, B Dasu y G Ramesh. «Design of LQR Based Stabilizer for Rotary Inverted Pendulum System». En: *IJCTA* 9 (2016), pág. 29.
- [33] X Zhang y col. «Study on Swing-up Control of Rotary Inverted Pendulum Based on Energy Feedback». En: - *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*. 2018, págs. 994-998. ISBN: NULL-. DOI: 10.1109/ICISCE.2018.00205.
- [34] Montoya Giraldo Oscar, Juan Valenzuela Hernández y Giraldo Buitrago Didier. «Control global del Péndulo Rotacional Invertido empleando modelos de energía». En: *Scientia et Technica* 1.52 (2012). DOI: 10.22517/23447214.7831. URL: <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/7831>.

-
- [35] S P Diwan y S S Deshpande. «Nonlinear Model Predictive Controller for the Real-Time control of Fast Dynamic System». En: - *2019 International Conference on Communication and Electronics Systems (ICCES)*. 2019, págs. 289-294. ISBN: NULL-. DOI: 10.1109/ICCES45898.2019.9002380.
 - [36] O Saleem y K Mahmood-Ul-Hasan. «Indirect Adaptive State-Feedback Control of Rotary Inverted Pendulum Using Self-Mutating Hyperbolic-Functions for Online Cost Variation». En: - *IEEE Access*. Vol. 8. 2020, págs. 91236-91247. ISBN: 2169-3536. DOI: 10.1109/ACCESS.2020.2994830.
 - [37] M U Soydemir y col. «Learning Feedback Linearization Based Stable Robust Adaptive NARMA Controller Design for Rotary Inverted Pendulum». En: - *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. 2019, págs. 795-799. ISBN: NULL-. DOI: 10.23919/ELECO47770.2019.8990417.
 - [38] S Chatterjee y S K Das. «Sampled-Data Control for Optimal Gain Margin of Cart Inverted Pendulum System: Comparison with Continuous-Time Control». En: - *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2018, págs. 1153-1157. DOI: 10.1109/ICARCV.2018.8581364.
 - [39] A Rahimi y col. «Controller design for rotary inverted pendulum system using particle swarm optimization algorithm». En: - *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2013, págs. 1-5. ISBN: 0840-7789. DOI: 10.1109/CCECE.2013.6567710.
 - [40] Morteza Harati, Amir Aminzadeh Ghavifekr y Amir Rikhtehgar Ghiasi. «Model Identification of Single Rotary Inverted Pendulum Using Modified Practical Swarm Optimization Algorithm». En: *2020 28th Iranian Conference on Electrical Engineering (ICEE)*. 2020, págs. 1-5. DOI: 10.1109/ICEE50131.2020.9261035.
 - [41] J.G. Guarnizo y J. Avendano. «Liquid level system as a pedagogical tool to teach fuzzy control». En: *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)*. 2017, págs. 1-5. DOI: 10.1109/CONIELECOMP.2017.7891820.
 - [42] Ngo Phong Nguyen y col. «Fuzzy-Based Super-Twisting Sliding Mode Stabilization Control for Under-Actuated Rotary Inverted Pendulum Systems». En: *IEEE Access* 8 (2020), págs. 185079-185092. DOI: 10.1109/ACCESS.2020.3029095.
 - [43] C Alarcón y C Muñoz. «Minimum time swing-up controller applied to a rotary inverted pendulum». En: - *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. 2017, págs. 1-6. ISBN: NULL-. DOI: 10.1109/CHILECON.2017.8229611.

-
- [44] Gloria Catalina Cifuentes Duarte. «Diseño de un algoritmo automático de sintonización de parámetros de un controlador PID empleando técnicas Bio-inspiradas de aprendizaje de máquinas». En: *Repositorio Universidad Santo Tomás*. 2019, págs. 15-49. DOI: <http://hdl.handle.net/11634/16570>.
 - [45] Luisa Fernanda Alarcón Sánchez. «Sintonización de un controlador óptimo cuadrático para sistemas multivariables usando sistemas inmunes». En: *Universidad Tecnológica de Pereira. Facultad de Tecnologías e Ingeniería*. 2015.
 - [46] Brian Noriega y col. «Design and Simulation of a Voltage Control Based on Neural Networks». En: *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*. 2021, págs. 133-138. DOI: 10.1109/CCAC51819.2021.9633294.
 - [47] Jose Guillermo Guarnizo, Cesar Leonardo Trujillo y Javier Antonio Guacaneme. «Modeling and control of a two DOF helicopter using a robust control design based on DK iteration». En: *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*. 2010, págs. 162-167. DOI: 10.1109/IECON.2010.5675183.
 - [48] Mukhtar Fatihu Hamza, Hwa Jen Yap e Imtiaz Ahmed Choudhury. «Genetic Algorithm and Particle Swarm Optimization Based Cascade Interval Type 2 Fuzzy PD Controller for Rotary Inverted Pendulum System». En: *Mathematical Problems in Engineering* 2015 (2015), págs. 1-15.
 - [49] Michael Levis. «Péndulo rotatorio: explorando el desafío del control clásico con QUBE-Servo 2». En: *Modelo Péndulo Rotatorio Invertido QUBE-Servo 2*. 2020. URL: <https://www.quanser.com/blog/rotary-pendulum-control-challenge-with-qube-servo/>.
 - [50] Martin J Corless y Arthur E. Frazho. «Linear systems and control: an operator perspective». En: *CRC Press*. 2003, pág. 390. URL: <https://www.worldcat.org/title/linear-systems-and-control-an-operator-perspective/oclc/053441587>.
 - [51] Juan Carlos Herrera Lozada. «Sistema inmune artificial con población reducida para optimización numérica». En: *Tesis doct, Repositorio Dspace*. 2017, pág. 82. URL: <http://tesis.ipn.mx/handle/123456789/21995>.
 - [52] Laura Carrasco Payo. «Implementación de algoritmo de scan-matching basado en Clonalg». En: *B.S. thesis*. Vol. 1. 2015, pág. 78. URL: <http://hdl.handle.net/10016/23213>.
 - [53] Thomas Bäck, David B Fogel y Zbigniew Michalewicz. «Evolutionary computation 1: Basic algorithms and operators». En: *CRC press*. 2018.

-
- [54] Jose Guarnizo y col. «Applications of Artificial Immune Systems in Agents». En: *IGI Global*. 2009, págs. 99-122. DOI: 10.4018/978-1-60566-310-4.ch005.
- [55] N Cruz Cortés. «Sistema inmune artificial para solucionar problemas de optimización». En: *PhD thesis, Centro de Investigación y de Estudios Avanzados del Instituto*. 2004.
- [56] Katsuhiko Ogata. «Ingeniería de control moderna». En: *Pearson Educación 1* (2003), págs. 1-908. URL: https://www.academia.edu/9814191/Ingenieria_de_Control_Moderna_Ogata_5ed.
- [57] Michel Lévis Jacob Apkarian Paul Karam. «Instructor Workbook». En: *Inverted Pendulum Experiment for MatLab®/simulink® users*. 2011, pág. 61. URL: <https://www.quanser.com/products/rotary-inverted-pendulum/>.