



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TEMA: DISEÑO E IMPLEMENTACION DE CONTROLADORES DE
TEMPERATURA APLICANDO LOS ALGORITMOS BIO INSPIRADOS COLONIA
DE HORMIGAS Y CUCKOO SEARCH

AUTOR: CAIZA QUINTE, CRISTIAN DANIEL

DIRECTOR: ING. IBARRA JÁCOME, OSWALDO ALEXANDER

SANGOLQUÍ

2019



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
CARRERA DE INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, **DISEÑO E IMPLEMENTACION DE CONTROLADORES DE TEMPERATURA APLICANDO LOS ALGORITMOS BIO INSPIRADOS COLONIA DE HORMIGAS Y CUCKOO SEARCH.** fue realizado por el señor **CAIZA QUINTE, CRISTIAN DANIEL** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 22 de enero del 2019

A handwritten signature in black ink, enclosed in a large, irregular oval shape. The signature is stylized and appears to read "Oswaldo Alexander Ibarra Jácome".

Ing. Oswaldo Alexander Ibarra Jácome

C. C. 1719535427



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERA DE INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, **CAIZA QUINTE, CRISTIAN DANIEL**, declaro que el contenido, ideas y criterios del trabajo de titulación: **DISEÑO E IMPLEMENTACION DE CONTROLADORES DE TEMPERATURA APLICANDO LOS ALGORITMOS BIO INSPIRADOS COLONIA DE HORMIGAS Y CUCKOO SEARCH**, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 22 de enero del 2019

Cristian Daniel Caiza Quinte

C. C. 1722637178



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERA DE INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL

AUTORIZACIÓN

Yo, **CAIZA QUINTE, CRISTIAN DANIEL** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **DISEÑO E IMPLEMENTACION DE CONTROLADORES DE TEMPERATURA APLICANDO LOS ALGORITMOS BIO INSPIRADOS COLONIA DE HORMIGAS Y CUCKOO SEARCH.** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 22 de enero del 2019

A handwritten signature in black ink, appearing to read 'Cristian Daniel Caiza Quinte', is written over a horizontal dotted line.

Cristian Daniel Caiza Quinte

C. C. 1722637178

DEDICATORIA

El presente trabajo de investigación está dedicado a mis padres Rodrigo y María, por su amor, trabajo, sacrificio y por los valores que me han inculcado en todos estos años para hacerle frente a las adversidades de mi vida.

A mis hermanos Henry y Maritza, por su apoyo incondicional durante toda mi trayectoria estudiantil, su cariño y su confianza durante toda mi vida.

Cristian D. Caiza Q.

AGRADECIMIENTO

A mi familia, por el apoyo y la confianza que me han brindado durante toda mi vida.

A mis compañeros y amigos de universidad, por tantas experiencias vividas.

A los docentes de la Universidad, por su formación académica y personal a lo largo de mi carrera universitaria.

Cristian D. Caiza Q.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS.....	xii
RESUMEN.....	xvii
ABSTRACT.....	xviii
CAPÍTULO 1	1
INTRODUCCIÓN	1
1.1. Antecedentes.	1
1.2. Justificación e Importancia.	4
1.3. Objetivos	7
1.3.1. General.....	7
1.3.2. Específicos	8
CAPITULO 2.....	9
ESTADO DEL ARTE.....	9

2.1. Estado Del Arte	9
2.2. Computación Inspirada En La Naturaleza	13
2.2.1. Algoritmos evolutivos.....	15
2.2.2. Inteligencia colectiva	16
2.2.3. Redes neuronales	17
2.2.4. Sistema inmunológico artificial	17
2.3. Algoritmos Bio Inspirados	18
2.3.1. Algoritmos Genéticos	19
2.3.2. Algoritmo Colonia De Hormigas.....	29
2.3.3. Algoritmo Cuckoo Search.....	41
CAPITULO 3	52
DESARROLLO DE CONTROLADORES BIO INSPIRADOS.....	52
3.1. Modelado de la Planta.....	52
3.1.1.1. Introducción	52
3.1.1.2. Componentes del sistema.....	54
3.1.1.3. Obtención de la función de transferencia de la planta	54
3.2. Objetivo de Control.....	61
3.3. Sintonización del controlador PID estándar.....	62
3.4. Diseño del controlador optimizado por el método de Algoritmos Genéticos	63
3.4.1. Definición de conceptos.....	63
3.4.2. Consideraciones iniciales	64
3.4.3. Esquema general y desarrollo del algoritmo.....	65

3.4.4. Resultados obtenidos de la aplicación del algoritmo	83
3.5. Diseño del controlador optimizado por el algoritmo Colonia de Hormigas	85
3.5.1. Definición de conceptos	85
3.5.2. Consideraciones iniciales	85
3.5.3. Esquema general y desarrollo del algoritmo.....	86
3.5.4. Resultados obtenidos de la aplicación del algoritmo	99
3.6. Diseño del controlador optimizado por el algoritmo Cuckoo Search	100
3.6.1. Definición de conceptos	100
3.6.2. Consideraciones iniciales	101
3.6.3. Esquema general y desarrollo del algoritmo.....	102
3.6.4. Resultados obtenidos de la aplicación del algoritmo	112
CAPITULO 4.....	114
ANÁLISIS DE RESULTADOS	114
4.1. Especificaciones del sistema:	114
4.2. Resultados obtenidos.....	115
4.2.1. Sintonización de controladores	115
4.3. Análisis de resultados.....	143
4.3.1. Análisis de resultados simulados	143
4.3.2. Análisis de resultados medidos	147
4.3.3. Análisis comparativo de resultados simulados con medidos	149
CAPITULO 5	151
CONCLUSIONES Y RECOMENDACIONES.....	151

5.1. Conclusiones 151

5.2. Recomendaciones..... 155

BIBLIOGRAFÍA..... 156

ÍNDICE DE TABLAS

Tabla 1 <i>Respuesta del sensor IC a las variaciones de temperatura</i>	55
Tabla 2 <i>Exactitud medid en las funciones generadas para 3 distintos casos</i>	57
Tabla 3 <i>Funciones de transferencia generadas para cada muestreo realizado</i>	59
Tabla 4 <i>Especificaciones a cumplir por el sistema de control.....</i>	61
Tabla 5 <i>Parámetros del controlador.....</i>	63
Tabla 6 <i>Desempeño y robustez.....</i>	63
Tabla 7 <i>Resultados sintonizados con la aplicación del método Algoritmos Genéticos implementado.....</i>	84
Tabla 8 <i>Resultados sintonizados con la aplicación del método Colonia de Hormigas implementado.....</i>	100
Tabla 9 <i>Resultados sintonizados con la aplicación del método Cuckoo Search implementado.....</i>	113
Tabla 10 <i>Parámetros iniciales para el desarrollo de los algoritmos</i>	114
Tabla 11 <i>Tiempos de establecimiento medidos por los controladores optimizados por Algoritmos Genéticos</i>	119
Tabla 12 <i>Controladores calculados por algoritmo colonia de hormigas.....</i>	119
Tabla 13 <i>Tiempos de establecimiento medidos por los controladores optimizados por Colonia de hormigas</i>	124
Tabla 14 <i>Controladores calculados por algoritmo colonia de hormigas</i>	124

Tabla 15 <i>Tiempos de establecimiento medidos por los controladores optimizados por Cuckoo Search</i>	130
Tabla 16 <i>Controladores calculados por algoritmo Cuckoo Search</i>	130
Tabla 17 <i>Resultados obtenidos en sobrelongación por los controladores sintonizados ...</i>	132
Tabla 18 <i>Resultados obtenidos en tiempo de establecimiento por los controladores sintonizados</i>	132
Tabla 19 <i>Resultados medidos en controladores ante perturbación de -0.1 [V]</i>	134
Tabla 20 <i>Resultados medidos en controladores ante perturbación de -0.2 [V]</i>	136
Tabla 21 <i>Mediciones tomadas del porcentaje de sobrelongación en los controladores aplicados sobre la planta PCT-2</i>	139
Tabla 22 <i>Mediciones tomadas del tiempo de establecimiento en los controladores aplicados sobre la planta PCT-2</i>	140
Tabla 23 <i>Ganancias encontradas por los algoritmos aplicados</i>	143
Tabla 24 <i>Resumen de características tomadas en la sintonización de controladores</i> ...	143
Tabla 25 <i>Resumen de características tomadas en la sintonización de controladores por algoritmos bio-inspirados y PID Tunner</i>	145
Tabla 26 <i>Resumen de resultados obtenido en rechazo a las perturbaciones por los controladores sintonizados</i>	146
Tabla 27 <i>Detalle de mediciones tomada en la aplicación de los controladores sintonizados</i>	147
Tabla 28 <i>Comparación de resultados obtenidos en tiempo de establecimiento.</i>	150
Tabla 29 <i>Comparación de resultados obtenidos en sobrelongación</i>	150

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Diagrama general de Algoritmos Genéticos.....	21
<i>Figura 2.</i> Ejemplo de Cruce	23
<i>Figura 3.</i> Ejemplo de Mutación.....	24
<i>Figura 4.</i> Ejemplo algoritmo colonia de hormigas.....	31
<i>Figura 5.</i> Algoritmo de optimización por colonia de hormigas	34
<i>Figura 6.</i> Ejemplo CH: Diagrama general de posibles nodos y rutas	37
<i>Figura 7.</i> Ejemplo CH: Costos y feromonas disponibles en cada ruta.....	38
<i>Figura 8.</i> Ejemplo CH: Probabilidades calculadas para cada ruta a partir del nodo 1	38
<i>Figura 9.</i> Ejemplo CH: Probabilidades calculadas para cada ruta a partir del nodo 2.....	39
<i>Figura 10.</i> Ejemplo CH: Probabilidades calculadas para cada ruta a partir del nodo 3.....	40
<i>Figura 11.</i> Ejemplo CH: Costo y feromonas actualizadas	41
<i>Figura 12.</i> Diagrama general del Algoritmo Cuckoo Search.....	45
<i>Figura 13.</i> Proceso térmico del modelo PCT-2 Fuente (Miniguano Miniguano, 2008)	53
<i>Figura 14.</i> Módulo PCT-2 disponible en el laboratorio de instrumentación de la Universidad de las Fuerzas Armadas ESPE.....	53
<i>Figura 15.</i> Respuesta del sensor IC a las variaciones de temperatura.....	56
<i>Figura 16.</i> Comparación de muestras realizadas ante entrada escalón de 3.3V.....	56
<i>Figura 17.</i> Generación de funciones de transferencia con la herramienta System Identification	57

Figura 18. Comparación sistema real con modelos matemáticos generados (Muestra real en negro, caso 1 en azul, caso 2 en verde, caso 3 en rojo).....	58
Figura 19. Comparación sistema real - modelo calculada (En rojo sistema real, verde modelo 1, azul modelo 2, negro modelo 3).....	59
Figura 20. Comparación de las funciones generadas par cada modelo con la función promedio calculada. (En rojo la TF promedio calculada, en azul la TF del modelo 1, en verde la TF del modelo 2, en negro la TF del modelo 3)	60
Figura 21. Sintonización del controlador PID por medio de la herramienta PID Tunner de	62
Figura 22. Esquema simplificado de simulación para el método de Algoritmos Genéticos	66
Figura 23. Método de la ruleta para dos selecciones Fuente: (Sedano García, 2013)	70
Figura 24. Esquema general del controlador	73
Figura 25. Ejemplo de solución generada para la medición de su sobre-elongación	74
Figura 26. Ejemplo de solución generada para la medición de su tiempo de establecimiento.....	76
Figura 27. Ejemplo de resultado: desempeño de controladores.....	80
Figura 28. Ejemplo de resultado: costo de controladores	81
Figura 29. Ejemplo de resultado: Desempeño global de soluciones.....	82
Figura 30. Ejemplo de resultado: Respuesta de los controladores.....	83
Figura 31. Resultados obtenidos de la aplicación de Algoritmos genéticos en la sintonización de ganancias	84

Figura 32. Esquema simplificado de simulación para el método de Colonia de Hormigas	87
Figura 33. Ejemplo recorrido de hormigas durante el algoritmo.....	98
Figura 34. Resultados obtenidos de la aplicación de Colonia de Hormigas en la sintonización de ganancias	100
Figura 35. Esquema simplificado de simulación para el método de Cuckoo Search	102
Figura 36. Vuelos de Lévy aplicados en la búsqueda del mejor controlador	112
Figura 37. Resultados obtenidos de la aplicación de Cuckoo Search en la sintonización de ganancias	113
Figura 38. Desempeño medido por el recorrido de las hormigas en el proceso de búsqueda.....	116
Figura 39. Costo medido por el recorrido de las hormigas en el proceso de búsqueda.....	117
Figura 40. Modelo tridimensional de desempeños alcanzados para el método de Algoritmos Genéticos.....	118
Figura 41. Respuesta de los mejores controladores ante entrada escalón unitario	118
Figura 42. Recorrido de hormigas realizado a lo largo del algoritmo.	120
Figura 43. Desempeño medido por el recorrido de las hormigas en el proceso de búsqueda.....	121
Figura 44. Costo medido por el recorrido de las hormigas en el proceso de búsqueda.....	122
Figura 45. Modelo tridimensional de desempeños alcanzados para el algoritmo Colonia de Hormigas	123
Figura 46. Respuesta de los mejores controladores ante entrada escalón unitario	123

Figura 47. Recorrido de los pájaros Cuco realizado a lo largo del algoritmo.....	126
Figura 48. Desempeño medido por la búsqueda de los Cuco	127
Figura 49. Costo medido por la búsqueda de los cuco.....	128
Figura 50. Modelo tridimensional de desempeños alcanzados para el algoritmo Cuckoo Search	129
Figura 51. Respuesta de los mejores controladores ante entrada escalón unitario	129
Figura 52. Resultados simulados obtenidos con las mejores soluciones de cada caso	131
Figura 53. Caso 1: Respuesta de los controladores ante una entrada escalón de $-0.1[V]$...	133
Figura 54. Medición de las respuestas de los controladore ante una entrada escalón de $-$ $0.1[V]$	133
Figura 55. Caso 2: Respuesta de los controladores ante una entrada escalón de $-0.2[V]$...	135
Figura 56. Medición de las respuestas de los controladores ante una entrada escalón de $-$ $0.2[V]$	135
Figura 57. Respuesta medida de los controladores en régimen transitorio.....	137
Figura 58. Medición de porcentaje de sobrelongación en los controladores aplicados sobre la planta PCT-2.....	138
Figura 59. Medición del tiempo de establecimiento en los controladores aplicados sobre la planta PCT-2.....	139
Figura 60. Rechazo ante perturbaciones medido para controlador sintonizado por Algoritmos Genéticos.....	140
Figura 61. Rechazo ante perturbaciones medido para controlador sintonizado por Colonia de Hormigas.....	141

Figura 62. Rechazo ante perturbaciones medido para controlador sintonizado por Cuckoo

Search 141

Figura 63. Rechazo ante perturbaciones medido para controlador sintonizado por PID

tunner..... 142

RESUMEN

El presente trabajo de investigación se basa en el diseño e implementación de controladores PID optimizados en base a los algoritmos bio-inspirados *Colonia de hormigas* y *Cuckoo search*, en comparación con un controlador optimizado aplicando algoritmos genéticos. Para ello se ha realizado la investigación pertinente acerca de trabajos previamente realizados, el comportamiento de hormigas en búsqueda de alimento, el proceso de búsqueda de nidos por las aves parasitarias cuco para depositar sus huevos, el proceso evolutivo planteado por Charles Darwin y del método de búsqueda aleatoria llamado Vuelos de Lévy, con el fin de aplicar algoritmos que simulen los comportamientos estudiados en búsqueda de las mejores soluciones al problema planteado. El desarrollo del trabajo continua con la programación de los algoritmos investigados en software dedicado para el efecto, el modelado de la planta de temperatura PCT-2 disponible en la Universidad de las Fuerzas Armadas ESPE sobre la cual se va a trabajar, y la sintonización de los controladores optimizados tanto por la herramienta PID Tunner como por los algoritmos diseñados, otorgando resultados gráficos del proceso de sintonización de controladores con su respectivo análisis tanto en régimen permanente como transitorio, para los controladores simulados y las mediciones reales de los controladores implementados en la planta de temperatura.

PALABRAS CLAVE:

- **CROMOSOMA**
- **FEROMONA**
- **AVES PARASITARIAS**

ABSTRACT

The present research work is based on the design and implementation of optimized PID controllers based on the bio-inspired algorithms of Ant colony and Cuckoo search, in comparison with an optimized controller applying genetic algorithms. To this end, relevant research has been carried out on previously performed work, the behavior of ants in search of food, the process of searching for nests by the parasitic birds cuckoo to deposit their eggs, the evolutionary process proposed by Charles Darwin and the method of random search called Lévy flights, in order to apply algorithms that simulate the behaviors studied in search of the best solutions to the problem. The development of the work continues with the programming of the algorithms investigated in dedicated software for the effect, the modeling of the temperature plant PCT-2 available in the University of the Armed Forces ESPE on which it is going to work, and the tuning of the controllers optimized both by the PID Tunner tool and by the designed algorithms, providing graphical results of the tuning process of controllers with their respective analysis both in permanent and transitory regime, for the simulated controllers and the real measurements of the controllers implemented in the plant of temperature.

KEY WORDS:

- **CHROMOSOME**
- **PHEROMONE**
- **PARASITIC BIRDS**

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Antecedentes.

En la actualidad, los controladores más utilizados a nivel industrial son los llamados: proporcional (P), proporcional integral (PI) y proporcional integral derivativo (PID), ya que estos cuentan con una estructura simple y relativamente fácil de diseñar; sin embargo, esta misma simpleza suele convertirse en su debilidad, ya que limita el rango de plantas en las cuales se los pueda utilizar de forma satisfactoria. Ante esta problemática, el área de investigación desarrolla un sinnúmero de métodos de optimización para el diseño de controladores.

Los algoritmos bio-inspirados son un método novedoso que se ha desarrollado para dar solución a múltiples problemáticas en el ámbito científico, ya que estos se basan en emular el comportamiento de sistemas biológicos. Estos algoritmos pueden basarse en la inteligencia de enjambres, sistemas biológicos y sistemas físico-químicos, de acuerdo a las fuentes en que se enfoquen (Zarzosa Gómez, 2017). No todos los algoritmos pueden resultar útiles, sin embargo, varios de ellos han demostrado ser una poderosa herramienta para la resolución de problemas en el mundo real.

Dentro de este ámbito, los algoritmos evolutivos han logrado convertirse en un tema altamente investigado y utilizado para la resolución de distintos problemas científicos, ya que estos presentan la ventaja de ser robustos ante cambios dinámicos en el entorno, pueden utilizarse para complementar otros métodos y se los puede aplicar a cualquier escenario que pueda ser planteado como un problema de optimización de funciones (Ruge & Alvis, 2010).

El método de Algoritmos Genéticos actualmente es el más utilizado dentro del ámbito de la computación evolutiva. Este método de optimización, se ha convertido en uno de los más populares y desarrollados en los últimos años, debido a que está inspirado en la variación genética y la selección natural de las especies, en el cual los individuos más adaptados tienen mayores probabilidades de sobrevivir y transferir su material genético a las nuevas generaciones (Ruge & Alvis, 2010).

Sin embargo, existen métodos aún no explorados en su totalidad dentro del ámbito de los algoritmos bio-inspirados, que cuentan con un increíble potencial y cuyo campo de investigación aún se encuentra joven. Un ejemplo de estos es el método de Colonia de Hormigas, inspirado en la capacidad de las hormigas para encontrar la ruta más corta entre la fuente de alimentación y su nido, en base a la cooperación de toda la colonia, cuya técnica de optimización ha sido abarcada por algunos autores en múltiples áreas de investigación. Otro método de optimización igualmente utilizado es el Cuckoo Search, el cual es una técnica inspirada en el estilo de vida de los pájaros cuco, conocidos por ser un tipo de ave “parasita” que deposita sus huevos en nidos ajenos, y que, en base a las características ambientales y a la

migración de sus sociedades, buscan el mejor hábitat para la cría y reproducción de su especie. (Ortiz, Perusquía, & Montes, 2003)

Las ventajas de utilizar estos métodos, han sido abarcadas en múltiples áreas dentro de la ingeniería. Por ejemplo, en (Obando, Zamora, & Girakdo) se utilizó algoritmos genéticos y un sistema basado en colonia de hormigas para la obtención de patrones de barras de control en reactores de agua en ebullición, donde se logró dar una mejora al desempeño de un ciclo de equilibrio de 18 meses, y cuyos resultados concluyeron que el método de colonia de hormigas logró disminuir la potencia al final del ciclo, lo cual representó un ahorro anual de 100 000\$.

Por otro lado, en (Trejo, López, & Ruiz, 2013) se utilizó el método de Optimización de Colonia de Hormigas en el ámbito de ingeniería en producción, en la Universidad Distrital Francisco José de Caldas, para dar solución al problema de distribución en planta, el cual consiste en buscar la configuración más óptima para máquinas, equipos y recursos en un espacio disponible, donde se obtuvo resultados favorables en comparación a otros métodos empleados con anterioridad.

En (Cárdenas Dobson) se realiza una comparativa entre el método de búsqueda Cuckoo Search, Modified Cuckoo Search (MCS) y un método de evolución diferencial, aplicándolos a un modelo de crecimiento de cultivos, en el cual se utilizó los métodos como un sistema de optimización para la predicción de los parámetros del modelo. Los resultados obtenidos permitieron estimar los parámetros del modelo SUCROS para el crecimiento potencial de los cultivos, y aunque los tres métodos presentaron desempeños similares, el algoritmo de

evolución diferencial resultó un poco mejor, llegando a la conclusión de que los algoritmos bio-inspirados mencionados, arrojaron una solución confiable y pueden ser aplicados en el proceso de estimación de parámetros de modelos de cultivos.

La Universidad de las Fuerzas Armadas ESPE cuenta con laboratorios y equipos especializados para el desarrollo académico de sus estudiantes. Dentro del área de Electrónica, el laboratorio de robótica se encuentra equipado con varias plantas de temperatura, caracterizadas por otorgar una salida lineal analógica, y capaz de ser manejada desde una tarjeta controladora. Este equipo es de gran utilidad para la implementación y el análisis de distintas técnicas de control, debido a su simplicidad de uso y la posibilidad de agregar perturbaciones externas, tal como sucedería en un entorno industrial.

1.2. Justificación e Importancia.

En el año de 2017, en Ecuador se presentó el Plan Nacional de Desarrollo 2017-2021 “Toda una vida” (PNBV), el cual incorpora las propuestas presentadas en más de 380 mesas de diálogo a través del país (Desarrollo, 2017). Uno de los objetivos marcados en dicho plan incluye *“Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria”*, para lo cual se plantea *“Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las*

universidades” (CNP); En base a esto, el proyecto planteado encaja dentro de los lineamientos presentados por el PNBV.

Dentro del ámbito de los sistemas de control, los modelos de controladores tradicionales como el controlador PID implementan las acciones proporcional, integral y derivativa para realizar la acción de control, con lo cual se enfoca en eliminar el error en estado estacionario gracias a su acción integral, y anticiparse o aplicar una acción predictiva sobre la salida del controlador gracias a su acción derivativa. Estos controladores PID son una respuesta satisfactoria para la resolución de problemas de control en muchas aplicaciones dentro de la industria, siempre y cuando la dinámica del sistema lo permita; sin embargo, al momento de requerir de controladores más robustos se suele optar por otras alternativas que brinden mejores resultados. (López Hernandez, 2014)

Los algoritmos bio-inspirados son, en la actualidad, uno de los campos más prometedores de investigación en el diseño de algoritmos para la resolución de problemas de optimización, que puede abarcar múltiples áreas de investigación gracias a que presentan robustez a los cambios en su entorno y pueden ser adaptativos o complementarios con otros métodos populares.

El área de investigación dentro del campo de la electrónica en control, desarrolla a diario nuevos y novedosos métodos para el diseño de controladores, por lo cual es importante contribuir con el desarrollo de nuevas técnicas de control que brinden solución a las distintas situaciones en que los controladores clásicos no responden de manera adecuada, aportando en

la investigación de nuevas técnicas aun no profundizadas como los algoritmos bio-inspirados orientados a los sistemas de control. Es importante recalcar que, al desarrollar técnicas de control modernas para el diseño de sistemas de control, es factible mejorar el desempeño alcanzado con las técnicas convencionales (Ruge & Alvis, 2010).

Actualmente el mayor exponente utilizado en el campo de los algoritmos evolutivos, son los algoritmos genéticos, el cual ha sido utilizada en los sistemas de control como un proceso de optimización complementario para los controladores PID, y permite converger a la solución óptima basándose únicamente en la función de error. Sin embargo, existen múltiples métodos aún no explorados en su totalidad que pueden ser igualmente factibles para el diseño de controladores. En este proyecto se pretende analizar la funcionalidad de los algoritmos bio-inspirados Colonia de Hormigas y Cucko Search, orientados al diseño de controladores en base a la optimización de sus parámetros, y el posterior análisis comparativo con diseño realizado en base a la optimización de parámetros mediante Algoritmos Genéticos, con el fin de brindar nuevas alternativas de solución al momento de seleccionar técnicas de optimización para el diseño de controladores.

Como resultado, este proyecto contribuirá al análisis y aplicación de nuevas alternativas de optimización en el diseño de controladores y a la utilización de algoritmos bio-inspirados de manera general para la resolución de problemas de control basados en optimización. Esto conlleva al desarrollo investigativo sobre novedosas alternativas de controladores robustos y

veloces, lo cual resulta beneficioso tanto para la industria como para la comunidad científica y estudiantil.

La investigación presentada será una de las pioneras dentro del área del desarrollo de técnicas de control basadas en algoritmos bio-inspirados, y servirá como base para futuras investigaciones dentro del área de los sistemas de control inteligente y de los problemas de ingeniería basados en optimización, tanto a nivel de pregrado como de posgrado.

El área de investigación de este trabajo se encuentra dentro del proyecto “Ajuste de los parámetros del controlador Fuzzy Logic del sistema de gestión energética de una microrred doméstica conectada a red mediante algoritmos de optimización inspirados en la naturaleza”. (2018-REV-003).

1.3. Objetivos

1.3.1. General

- Realizar un análisis comparativo entre el desempeño alcanzado por un controlador PID optimizado mediante Algoritmos Genéticos y dos controladores PID optimizados mediante la aplicación de los algoritmos bio-inspirados Colonia de hormigas y Cuckoo Search, aplicados a una planta de temperatura.

1.3.2. Específicos

- Determinar los parámetros a optimizar en el diseño de los controladores
- Desarrollar el controlador PID optimizado mediante la aplicación del algoritmo bio-inspirado Colonia de Hormigas.
- Desarrollar el controlador PID optimizado mediante la aplicación del algoritmo bio-inspirado Cuckoo Search.
- Determinar los criterios de comparación para los controladores diseñados.
- Comparar el comportamiento de los controladores PID optimizados mediante los algoritmos señalados con un modelo PID convencional optimizado mediante Algoritmos genéticos
- Analizar el desempeño de los controladores implementados en régimen transitorio y permanente

CAPITULO 2

2. ESTADO DEL ARTE

2.1. Estado Del Arte

Son varias las aplicaciones que se han dado para el método de optimización por algoritmos genéticos, dada su robustez y capacidad de adaptarse y complementarse a distintos métodos de búsqueda y optimización.

Un ejemplo de esto es la investigación realizada por (Agustín, Enero 1998) , donde se implementa algoritmos genéticos con el objetivo de comparar su desempeño con los clásicos algoritmos de ramificación y acotamiento, aplicado en programas que frecuentemente se emplean en el diseño óptimo de redes de distribución de energía eléctrica. Estudiando las ventajas de los algoritmos genéticos en cuanto a tiempos de cálculo y soluciones obtenidas, donde se obtuvo resultados favorables en comparación a otros métodos empleados con anterioridad.

Por otro lado, en (Hoyos Pineda & Pérez Castillo, 2007) se analiza la importancia de mejorar los servicios de inteligencia Web y se proponen técnicas de computación flexible, como la computación evolucionaria, tomando como ejemplo los Algoritmos Genéticos para el problema de la clasificación de imágenes de satélite, obteniendo una resultados favorables en el procesamiento de imágenes de percepción remota y proponiendo un integración de distintos

métodos de clasificación existentes con el método de optimización por Algoritmos Genéticos para futuras investigaciones.

Dentro del ámbito del diseño de controladores, (Martinez, Sanchis, & Blasco, 2010) implementan un diseño de control robusto de un proceso de masa-muelle planteado como un problema de optimización multi-objetivo y cómo se puede utilizar *Physical Programming (PP)* con *Algoritmos Genéticos* para dar solución a los problemas que esta técnica presenta frente a funciones con numerosos mínimos locales. Los resultados obtenidos son comparados con las soluciones obtenidas por otros autores frente al mismo problema tipo, concluyendo que el método de *Algoritmos Genéticos* utilizado como técnica de optimización no lineal presenta una gran eficacia en cuando a la obtención de soluciones que mejoró las presentadas originalmente.

En (Parisi, Pairisi, & David, 2006), se realizó un análisis de la capacidad de los modelos construidos a partir de algoritmos genéticos recursivos y una red neuronal Ward, orientados a la predicción del signo de las variaciones semanales de los índices bursátiles *Nikkei 225*, *Hang Seng*, *Shanghai Composite*, *Seoul Composite* y *Taiwan Weighted*. Los resultados fueron comparados con los de un modelo ingenuo o AR(1) y una estrategia buy and hold, concluyendo que, a pesar de que el modelo de red neuronal Ward tuvo una mayor capacidad predictiva, el modelo obtenido a partir de algoritmos genéticos obtuvo mejores desempeños en términos de rentabilidad corregida por riesgo, medida por los índices de Sharpe y Treynor

El algoritmo de optimización por colonia de hormigas es una técnica de optimización computacional que se utiliza generalmente para dar solución a los problemas de búsqueda de

los mejores caminos en grafos, aunque puede ser planteado para diversos problemas de optimización en múltiples áreas de investigación.

Un experimento realizado en (Barán & Almirón) se basa la afirmación de que en una colonia, las hormigas son agentes autónomos que trabajan en paralelo, para presentar una implementación paralela asíncrona en una red de computadoras personales, con el fin de analizar la viabilidad de dicha implementación e identificar las ventajas del paralelismo, sin ánimo de comparar los resultados con otras investigaciones.

Por otro lado, (Insfrán, Pinto, & Barán) propone un novedoso enfoque de resolución del Routing and Wavelength Assignment (RWA) basado en colonia de hormigas, proponiendo una conversión del problema RWA original a un problema de enrutamiento tradicional. Esto se logró gracias a la modificación del grafo G original de la red óptica, donde los enlaces con varias longitudes de onda son reemplazados por nodos y enlaces independientes representando diferentes caminos virtuales.

Dentro del ámbito militar, (Mora García, 2009) presentó un trabajo orientado a la resolución de un problema de búsqueda del camino más óptimo atendiendo a varios criterios dentro de un entorno que modela un campo de batalla militar. Para ello plantea el algoritmo de optimización por colonia de Hormigas para la resolución de problemas *multi objetivo*, por lo que el trabajo está enfocado a la resolución de problemas con múltiples objetivos a optimizar y lo compara con métodos convencionales adaptados para resolver la misma problemática. El algoritmo presentado concluyó de manera satisfactoria con las pruebas realizadas, en su

mayoría con mejores soluciones que los métodos convencionales como el enfoque de voraz o greedy.

De igual forma, en (Coy Calixto, 2005) se desarrolla un aplicativo para la calibración de cuencas hidrológicas usando el modelo de lluvia-escorrentía formulado por Thomas, para lo cual se implementó tres casos de estudio: uno teórico y dos prácticos en las cuencas de Subachoque y Curubital en la Sabana de Bogotá, demostrando las fortalezas de los modelos basados en colonia de hormigas y del aplicativo del modelo de Thomas.

Cuckoo Search es un algoritmo de optimización inspirado en el comportamiento de las aves parasitarias cucos. Este tipo de algoritmo (con algunas variantes) ha sido estudiado por varios autores y utilizado en múltiples áreas de ingeniería, brindando resultados satisfactorios frente a las técnicas convencionales.

Una ejemplo de esto es la investigación realizada en (Zheng & Zhou, 2012), donde se analiza el problema de convergencia del algoritmo Cuckoo Search convencional (CS), y se plantea un novedoso actualización basada en una distribución de Gauss (GCS), aplicados a la resolución de funciones de prueba estándar y problemas de optimización de ingeniería, donde concluyeron que la técnica de GCS presentó mejores resultados a los obtenidos por CS, y que presenta una alta tasa de convergencia.

De igual forma, el algoritmo Cuckoo Search es utilizado en (Ortiz, Perusquía, & Montes, 2003), para la optimización de parámetros en el diseño de un controlador para un

sistema MIMO, donde los resultados obtenidos fueron comparados con otros dos controladores diseñados mediante la aplicación de algoritmos genéticos y enjambre de partículas para la estimación de sus parámetros, obteniendo como resultado una superioridad en la convergencia de la solución en un número menor de iteraciones, bajo las condiciones aplicadas en los ejemplos utilizados.

Por otro lado en (Chandrasekaran & Simon, 2012) se plantea un sistema híbrido de *Cuckoo Search* integrado con un *Fuzzy System*, para dar solución al problema de resolución multi-objetivo. El documento considera tres funciones principales: el costo del combustible, las emisiones y el nivel de confiabilidad del sistema para su resolución. El algoritmo *CS* se encarga de buscar la mejor solución dentro del dominio de búsqueda difuso, ajustando simultáneamente las variables del diseño planteado. El ajuste de variables del diseño difuso elimina la necesidad de experiencia para poder plantear una solución acertada. El diseño se probó tanto en sistemas de un único objetivo como en los multi-objetivos, midiendo su efectividad en comparación a otros métodos convencionales descritos en la bibliografía del artículo.

2.2. Computación Inspirada En La Naturaleza

A lo largo de la historia, la naturaleza ha sido una insaciable fuente de inspiración para el hombre. Distintas áreas como el arte, la arquitectura y la ingeniería, han explorado los métodos y técnicas de la inteligencia colectiva, los sistemas físico químicos o de distintos factores biológicos. Dentro del ámbito ingenieril, se ha desarrollado la denominada

Computación inspirada en la naturaleza o también conocida como *Computación natural*, la cual busca brindar soluciones efectivas a las distintas problemáticas con los que cuentan los modelos convencionales.

La computación natural abarca un conjunto de modelos que se caracterizan por simular el modo en que la naturaleza actúa, es decir, estudia la forma en que las leyes de la naturaleza actúan sobre los distintos sistemas. La computación inspirada en la naturaleza abarca un conjunto de algoritmos que basan su comportamiento en fenómenos naturales definidos; A estos algoritmos, cuyo comportamiento se basa en la simulación de los procesos biológicos naturales o sociales, se los ha denominado como *Algoritmos Bio-inspirados*.

Los algoritmos bio-inspirados simulan el comportamiento de sistemas naturales, aplicados al diseño de métodos heurísticos no determinísticos, enfocados a la búsqueda, aprendizaje, comportamiento, entre otras cualidades; por este motivo, se suele utilizar a los algoritmos bio-inspirados para la resolución de problemas de optimización en sistemas no lineales, donde los modelos convencionales tienen dificultad para operar. Estos algoritmos se han convertido actualmente en uno de los temas de investigación más prometedores y llamativos para brindar solución a las distintas problemáticas que suelen suscitarse en el mundo real.

El interés por esta novedosa mecánica de solución se basa en que dichos algoritmos recrean varios comportamientos inteligentes en ambientes cambiantes y complejos, los mismos

que son capaces de aprender, adaptarse, generalizar, abstraer, descubrir, asociar, entre otras cualidades que podemos encontrar dentro de la naturaleza. (Mezura, 2009)

De manera general, podemos abarcar a estos bio-algoritmos en cuatro grandes grupos: Algoritmos evolucionarios, inteligencia colectiva, Redes neuronales y algoritmos basados en sistemas inmunológicos. Cabe recalcar que esta clasificación no es exclusiva y que algunos algoritmos pueden abarcar más de un área de interés, dependiendo del objetivo de los mismos. (Pappa, 2012)

2.2.1. Algoritmos evolutivos

Los *algoritmos evolutivos* son métodos de optimización y búsqueda de soluciones basados en las leyes de la evolución biológica y permiten brindar solución a problemas no lineales que involucren varias variables de búsqueda en problemas complejos. Estos algoritmos se modelan simulando el proceso de selección natural postulado por Darwin.

En la naturaleza, el objetivo de la optimización es la supervivencia, lo cual requiere de la maximización de ciertos factores específicos y la minimización de otros, para lograr dicho objetivo; por lo tanto, un algoritmo evolutivo será utilizado para optimizar una sola función. Estos han sido aplicados satisfactoriamente en problemas de aprendizaje, optimización, simulación de sistemas dinámicos y resolución de problemas no lineales donde los métodos más comunes no son capaces de encontrar soluciones en tiempos aceptables.

El ejemplo más utilizado dentro de los algoritmos evolutivos son los llamados *Algoritmos Genéticos*, establecido por Holland en 1975 y cuya investigación ha sido abarcada por varios autores a lo largo de los años. De igual forma existen otros ejemplos como: *Programación genética*, *Evolución gramatical*, *Estrategias evolucionarias*, *programación evolucionaria*.

2.2.2. Inteligencia colectiva

La *inteligencia colectiva* o también denominada *Inteligencia de enjambre*, hace referencia al comportamiento combinado de algunos sistemas naturales, en el cual varios agentes interactúan y se guían en base a reglas naturales simples. Cada individuo del sistema puede ser considerado como inteligente, sin embargo, es la recopilación de varios individuos la que presenta un comportamiento organizado e inteligente para cumplir con un objetivo definido.

Estos algoritmos están inspirados en el comportamiento colectivo de algunas especies animales, tales como insectos (hormigas, avispas, termitas, abejas), aves e incluso peces. Dentro de este ámbito podemos encontrar algunos tipos de algoritmos desarrollados en los últimos años, tales como: *Colonia de hormigas*, *algoritmo murciélago*, *cuckoo search*, *algoritmo luciérnaga*, *monkey search*, entre otros.

2.2.3. Redes neuronales

Las redes neuronales son modelos matemáticos basados en el comportamiento de las neuronas y en la estructura del cerebro mismo. Gracias a su flexibilidad, una red neuronal es capaz de “aprender” acerca de un sistema definido con el fin de adaptarse a las posibles problemáticas que puedan presentarse bajo condiciones similares.

La importancia de las redes neuronales radica principalmente en su aplicación sobre los sistemas no lineales, sin embargo, por su naturaleza experimental, la base teórica con la que se cuenta actualmente es en gran mayoría heurística, por lo que su aplicación se sustenta principalmente en métodos basados en la experiencia. (Tablada & Torres)

Algunos ejemplos de algoritmos dentro de las redes neuronales son: *Multilayer perception (MLP)*, *radial basis function (RBF)*, *Self organizing maps (SOM)*.

2.2.4. Sistema inmunológico artificial

Los algoritmos basados en el sistema inmunológico artificial se basan en imitar el sistema inmunológico humano, encargado de proteger al cuerpo de infecciones, tanto externas como internas, y cuya función principal es la de reconocer las células del cuerpo como propias o no propias. De manera general, el proceso de este algoritmo consiste en generar una población inicial de posibles soluciones, seleccionar un porcentaje de los mejores individuos (los cuales

serán clonados), y a los cuales se les aplica una hipermuta, para finalmente llegar a una solución óptima.

Algunos ejemplos de estos algoritmos son: *Algoritmos de selección negativa*, *algoritmos de expansión clonal*, *algoritmos de redes*.

2.3. Algoritmos Bio Inspirados

Como una categoría de la inteligencia artificial, los algoritmos bio-inspirados son técnicas de optimización cuyo diseño se basa en emplear analogías con sistemas naturales o sociales, orientados a la resolución de problemas. Este tipo de algoritmos simulan el comportamiento de un sistema natural para el diseño de métodos heurísticos no determinísticos.

Actualmente, los algoritmos bio-inspirados, son uno de los campos más prometedores dentro de la investigación orientada al diseño de algoritmos optimizados; Estos son caracterizados por ser no determinísticos, con frecuencia presentan una estructura paralela, y son adaptativos.

En presente trabajo investigativo abarca la comprensión, aplicación e implementación de los algoritmos bio-inspirados *Colonia de Hormigas* y *Cuckoo Search* orientados al diseño de controladores PID optimizados, en comparación a un controlador PID optimizado mediante la técnica de *Algoritmos Genéticos*.

2.3.1. Algoritmos Genéticos

2.3.1.1. Principio de evolución biológica

Podemos describir a la evolución biológica como el proceso en que las poblaciones y las especies de organismos cambian con el tiempo. Esta idea se la atribuye al naturalista británico Charles Darwin, cuyos postulados, controversiales para su época, fueron publicados en su libro *El origen de las especies* en el año de 1859 en base a observaciones durante sus viajes por el mundo.

El mecanismo para la evolución es descrito como “selección natural”, término que se basa en que los rasgos heredables de un individuo u organismo, que lo ayudan a sobrevivir y reproducirse, se vuelven más comunes dentro de una población a lo largo del tiempo. El proceso mediante el cual los organismos cambian sus características a lo largo de generaciones con el fin de sobrevivir, era descrito por Darwin como “descendencia con modificaciones”, aunque hoy en día es más conocido como evolución.

Por imitación de este proceso, se ha desarrollado el método de Algoritmos genéticos para la resolución de problemas, donde la población de organismos es conocida como un conjunto de soluciones, las cuales “evolucionan” hasta conseguir una única solución óptima.

2.3.1.2. Descripción del algoritmo

Los algoritmos genéticos encajan dentro de la categoría de *Algoritmos evolutivos*, y fueron desarrollados por J. Hollan, en los años 70, con el fin de emular y entender el proceso adaptativo natural de las especies, para luego aplicarlo en procesos de optimización en los años 80.

Este método se caracteriza por ser de tipo adaptativo y robusto, por lo que pueden ser utilizados para resolver problemas de búsqueda y optimización, provenientes de distintas áreas. Si bien, este método no garantiza que la solución obtenida sea la más óptima, mediante la evidencia empírica se ha podido comprobar que entregan una solución de nivel aceptable, en un tiempo capaz de competir con el resto de algoritmos de optimización combinatoria.

En caso de que exista una técnica especializada para la resolución de un problema definido, es probable que esta supere al desempeño de los algoritmos genéticos, por lo cual, este método es utilizado principalmente para la resolución de problemas de optimización que no cuentan con una técnica especializada; aunque diversas investigaciones han logrado mejorar varias técnicas convencionales mediante la hibridación del método común con la aplicación de los algoritmos genéticos.

El diagrama general del proceso de los algoritmos genéticos, se puede apreciar en la siguiente figura:

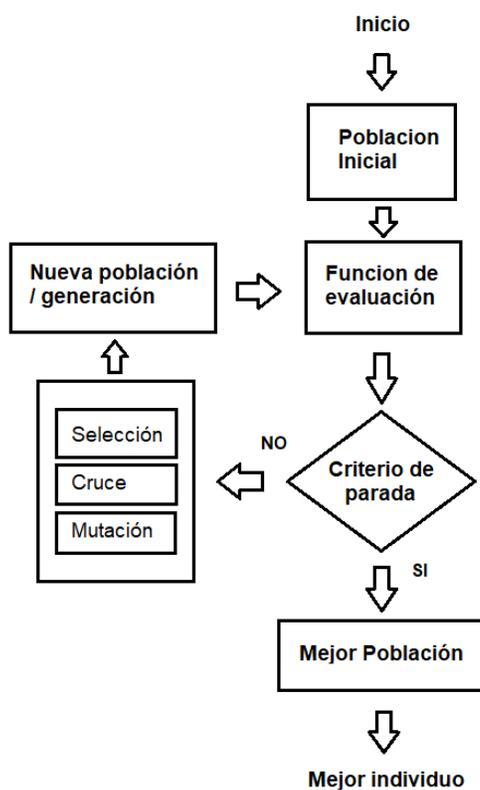


Figura 1. Diagrama general de Algoritmos Genéticos

El detalle de cada sección se presenta a continuación:

a. Población inicial

Generalmente la población inicial se escoge generando un conjunto de soluciones o individuos al azar. En algunas investigaciones se suele optar por inicializar la población con los resultados obtenidos de alguna otra técnica heurística o de optimización; como resultado se suele mejorar la velocidad de la convergencia del algoritmo, sin embargo, en algunos casos

presenta la desventaja de que el resultado converge hacia uno de los individuos iniciales sin presentar mejoras con la población inicial.

b. Función de evaluación

Una vez obtenida la población inicial, la función de evaluación es el proceso mediante el cual se aplica una *función de aptitud*, con el fin de determinar “que tan buena” es la población actual de posibles soluciones.

c. Selección

El proceso de selección se encarga de determinar que individuos son aptos para ser padres. Para ello se suele utilizar el *método de la ruleta*, con el fin de determinar con cierta probabilidad en base a la aptitud de cada padre, quienes serán los seleccionados para los próximos procesos de cruce o mutación dentro del algoritmo

d. Cruce

Luego de pasar por el proceso de selección, el algoritmo emplea un *cruce basado en un punto*, el cual consiste en que dos individuos encargados de jugar el papel de padres, son recombinados tomando como referencia el punto de cruce seleccionado para dar origen a una nueva generación. De manera gráfica, se puede observar proceso de cruce en la siguiente figura:

e. Mutación

La mutación representa un operador básico que proporciona un elemento aleatorio dentro del conjunto de soluciones. Se basa en el principio de la evolución primitiva, la cual no requiere de cruce, y se enfoca en generar una pequeña variación o mutación de un individuo hacia su descendencia. Esta variación depende de un punto aleatorio de mutación, en el cual, la información será alterada para la descendencia del individuo, como se muestra en la siguiente figura:



Figura 3. Ejemplo de Mutación

Donde podemos observar el punto aleatorio de mutación en el 4to bit del vector, el mismo que será alterado para dar origen a la descendencia.

Para ello es necesario definir una probabilidad, generalmente estática y menor a la probabilidad de cruce, la cual se efectúa luego del proceso de selección; a pesar de que no existe un valor ideal para la probabilidad de mutación en el algoritmo, existe diversas

investigaciones que se basan en el tamaño del vector solución, o bien del número de individuos de la población.

Este proceso es importante dentro del proceso ya que evita que el algoritmo se estanque en alguna solución local y permite abarcar nuevas soluciones con el fin de analizar todas las posibles ramas dentro del proceso de búsqueda.

f. Nueva población/generación

Una vez que se ha realizado el proceso de selección y cruce o mutación, la nueva población corresponde a la descendencia obtenida de la iteración anterior, es decir, una vez que se ha conseguido una población de descendientes, estos pasarán a ser los padres en la siguiente iteración. Este proceso se repite mientras no se cumpla el criterio de parada, el cual determina que el algoritmo ha convergido a una población de soluciones aceptables.

g. Criterio de parada

El criterio de parada es el encargado de permitir o rechazar que se produzca una nueva generación de soluciones en base al cumplimiento de una o varias condiciones que dichas soluciones deben cumplir. Esto suele estar representado mediante el cálculo del error obtenido en la evaluación de las soluciones presentadas en cada generación, definiendo un valor mínimo aceptable para las soluciones buscadas.

h. Mejor población

La mejor población corresponde al conjunto de las mejores soluciones obtenidas una vez que el algoritmo ha convergido a un óptimo global. Esto se obtiene a través de varias iteraciones o generaciones y se consigue una vez cumplidos los criterios de parada.

i. Mejor individuo

De la mejor población obtenida, se procede a evaluar cada uno de las soluciones encontradas a fin de seleccionar la mejor de ellas; esta solución es conocida como el mejor individuo.

2.3.1.3. Ejemplo de aplicación de Algoritmos Genéticos

a. Paso 1: Población inicial

Partimos con la generación aleatoria de una población inicial de posibles soluciones al problema planteado, como se muestra en el siguiente ejemplo

0101 0011 1100 1101 0101

b. Paso 2: Función de evaluación

Evaluamos el desempeño de cada uno de los individuos planteados con el fin de definir la probabilidad de que estos sean seleccionados en el proceso de selección.

Desempeño:

1010_{24%} 0101_{75%} 0011_{33%} 1000_{67%} 1101_{40%} 0101_{12%}

Probabilidad de ser seleccionados:

9.56% 29.88% 13.14% 26.7% 15.93% 4.78%

c. Paso 3: Criterio de parada

Definimos un criterio para que nuestro algoritmo se detenga, el cual puede ser alcanzar cierto objetivo o completar un número de repeticiones definido.

5 iteraciones

d. Paso 4: Selección

Se seleccionan los mejores individuos de cada generación en base a la probabilidad calculada en función de su desempeño

0101_{75%}0011_{33%}1000_{67%}1101_{40%}

e. Paso 5: Cruce

El proceso de cruce se realiza en parejas de individuos, tomando como referencia un punto de cruce aleatorio y combinando las partes de cada padre, como se muestra a continuación:

Suponiendo un punto de cruce entre el 2do y 3er bit

Padre1: 0101 Padre2: 0011

Padre3: 1000 Padre4: 1101

Hijo1: 0111 Hijo2: 0001

Hijo3: 1001 Hijo4: 1100

f. Paso 6: Mutación

La mutación se efectúa sobre un bit aleatorio dentro del individuo, y cuenta con su propia probabilidad de ser o no realizada, como se muestra a continuación

Suponiendo un punto de mutación en el tercer bit para el Hijo2

Hijo2: 0001 → 0011

g. Paso 7: Nueva generación

La nueva población será conformada por la descendencia resultante de los procesos de selección, cruce y mutación ejecutados. Estos vendrán a conformar los nuevos individuos que darán origen a las próximas descendencias en busca de las mejores soluciones para la resolución del problema.

0111

0011

1001

1100

h. Paso 8: Volver al paso 2

Una vez realizados todos los pasos anteriores, repetimos el proceso desde el paso 2 hasta cumplir el criterio de parada definido.

2.3.2. Algoritmo Colonia De Hormigas

2.3.2.1. Inteligencia basada en colonia de hormigas

Las hormigas son uno de los principales ejemplos de inteligencia colectiva que podemos encontrar en la naturaleza. Estos insectos se basan en la colaboración mutua para llevar acabo múltiples tareas que serían imposibles de realizar desde el punto de vista de una hormiga individual.

Una habilidad muy importante con que cuentan muchas especies de hormigas, radica en la habilidad de encontrar la ruta más corta entre su hormiguero y la fuente de comida más cercana. Es importante recalcar que muchas de las especies de hormigas que existen son prácticamente ciegas o tienen una visión sumamente limitada, por lo que llama aún más la atención la capacidad que tiene de localizar las rutas más cortas hacia una fuente de alimento; esta habilidad ha sido desarrollada en base a la secreción de feromonas por parte de las hormigas

El proceso para lograr este objetivo se puede describir de manera general de la siguiente forma: En un primer momento, la hormiga exploradora se mueve de manera casi aleatoria por el terreno de exploración hasta encontrar una fuente de alimento; si lo logra, retorna al hormiguero de una manera casi directa, dejando tras de sí un rastro de feromonas.

Estas feromonas atraen a las hormigas más cercanas y, de una manera casi probabilística, pueden seguir o no el rastro dejado por la primera exploradora. De seguir el rastro de feromonas marcado y regresar con alimento a la colonia, estas nuevas hormigas depositaran un rastro de feromonas, lo cual fortalece el anterior y, por consecuencia, atraerá más hormigas al mismo camino.

Si existe más de un camino marcado hacia la fuente de alimentación al mismo tiempo, la ruta más corta será recorrida por un mayor número de hormigas, y en consecuencia aumentará la intensidad de las feromonas depositadas, aumentando la probabilidad de que el

resto de hormigas sigan el mismo rastro. De a poco, las rutas más largas irán desapareciendo mientras prevalece la más corta, escogida por todas las hormigas.

Este proceso se puede observar de manera simplificada en la siguiente ilustración:

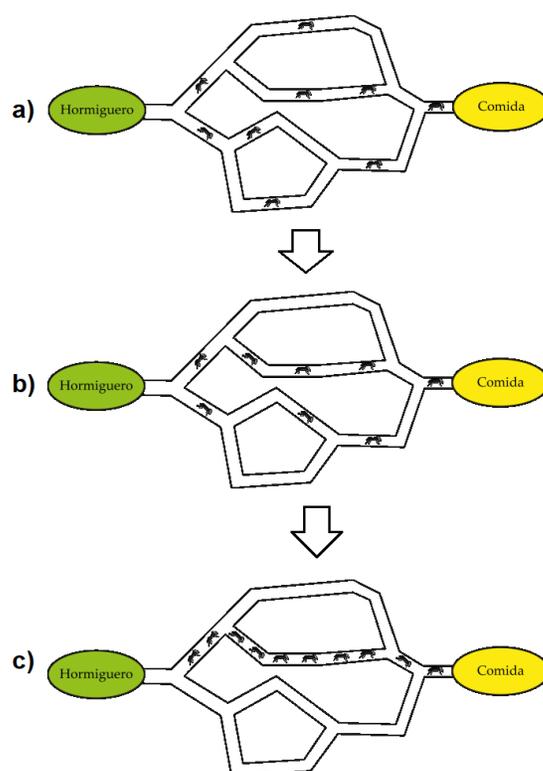


Figura 4. Ejemplo algoritmo colonia de hormigas

En la Figura 4.a podemos observar el esquema inicial de búsqueda de alimento de las hormigas, en el cual estas se dispersan por distintos caminos hasta encontrar la fuente de alimento de mejor calidad, dejando un rastro de feromonas a su paso. La figura 4.b presenta la

tendencia de las hormigas a seguir los caminos con una mayor concentración de feromonas, descartando los caminos más largos y concentrándose en los más rápidos a la fuente de alimentación. La figura 4.c muestra la culminación del algoritmo, donde la población entera de hormigas recorre un único camino más viable y rápido hacia su fuente de alimentación.

2.3.2.2. Descripción del algoritmo

Inspirados en el comportamiento natural de las hormigas de los entomólogos Pierre-Paul Grassé, Bert Holldobler y Edward Osbome Wilson, *Ant System (AS)* fue una técnica propuesta por Marco Dorigo para brindar solución al conocido paradigma del cajero viajante. Posteriores trabajos de Dorigo extendieron el alcance de AS, como por ejemplo un híbrido denominado *ANT-Q*, formado de la unión entre el AS y un algoritmo de aprendizaje conocido como Q-learning. Posteriormente fue desarrollado *Ant Colony System (ACS)*, como una extensión de *Ant-Q*, que presenta mejoras respecto al procedimiento de selección de la siguiente ciudad a visitar, la actualización global de la matriz de feromonas en base al mejor camino recorrido y la regla de actualización local, que permite a las hormigas actualizar la matriz de feromonas al terminar cada recorrido. Desde aquel entonces, ha existido un gran interés dentro de la comunidad científica por aplicar estos algoritmos a una gran variedad de problemas de optimización. (Márquez Vázquez, 2012)

Este algoritmo es de tipo iterativo, y en cada iteración se coloca una hormiga representando una posible solución. Las hormigas se desplazan a lo largo de distintos *odos*, hasta encontrar su fuente de alimentación, y liberan cierta cantidad de feromonas artificiales

para influenciar sobre las siguientes en base al desempeño obtenido por la fuente de alimentación y el costo de la ruta seguida.

A lo largo de las iteraciones, las hormigas construyen la mejor solución de una manera probabilística, en base al costo de cada camino y a la cantidad de feromonas depositadas en cada ruta de solución. Para cada iteración es necesario actualizar la cantidad de feromonas para las rutas exploradas, y es importante aplicar un tipo de *desvanecimiento de feromonas* para que, con cierto tiempo, la cantidad de feromonas de caminos alternos a la solución buscada vaya reduciéndose hasta desaparecer y facilitar la toma de decisiones de las próximas hormigas hacia la solución más óptima.

Los resultados computacionales obtenidos a lo largo de varias investigaciones han logrado calificar a este método como uno de los mejores posibles y han atraído a una gran cantidad de nuevos investigadores para usarlo en novedosas áreas de interés, tales como el diseño de circuitos lógicos combinatorios, redes neuronales, procesamiento de imágenes, enrutamiento de vehículos, control inteligente entre otros. (Robles Algarín, 2010)

La siguiente ilustración presenta el esquema general de funcionamiento del algoritmo de optimización por colonia de hormigas:

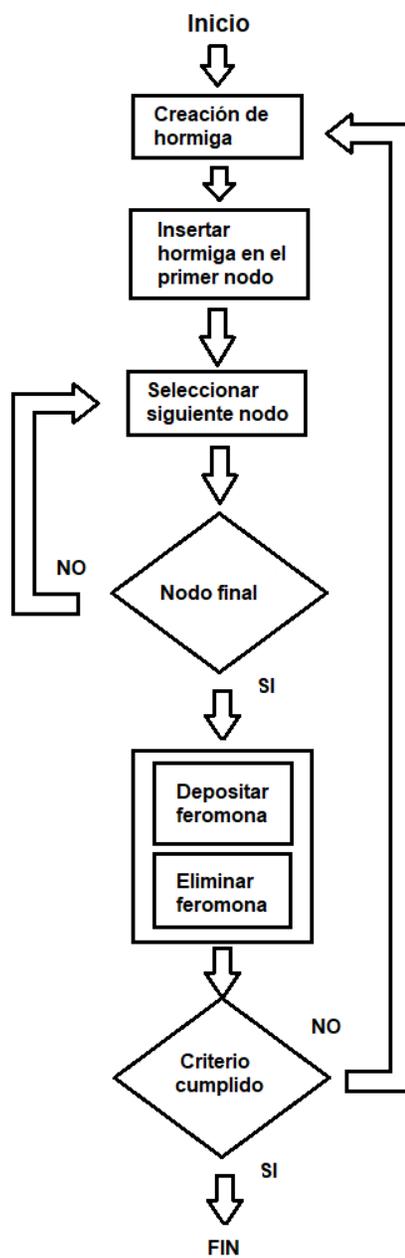


Figura 5. Algoritmo de optimización por colonia de hormigas

El detalle de cada sección se muestra a continuación:

a. Creación de hormiga

La creación de la hormiga hace referencia de la población inicial de posibles soluciones con las que vamos a trabajar. De manera general esta se escoge generando un conjunto de soluciones u hormigas al azar, aunque esta también puede ser resultado de algún método previo para la obtención de los parámetros.

b. Insertar hormiga en el primer nodo

Una vez que tenemos nuestra población inicial lista, procedemos al cálculo probabilístico de que cada hormiga se dirija hacia uno de los nodos a su alcance. Este valor se consigue en base a la cantidad de feromonas en cada nodo y al costo que tiene la ruta hacia el mismo nodo.

c. Seleccionar el siguiente nodo

El proceso de selección del primer nodo se repite para cada ruta que tomen las hormigas, por lo que el cálculo probabilístico deberá realizarse para todas las rutas posibles y dependerá del criterio del diseñador el permitir o no que las hormigas regresen a un nodo ya recorrido.

d. Depósito de feromonas

Una vez alcanzado el nodo final, se procede a realizar una actualización de feromonas en cada una de las rutas recorridas por la hormiga hasta llegar a su fuente de comida. Para ello se realiza un cálculo en base al desempeño de cada solución obtenida, y este define la cantidad de feromonas que son agregadas a las rutas involucradas.

e. Eliminación de feromonas

Con el fin de evitar una saturación de feromonas en las rutas recorridas por cada hormiga, es necesario definir una constante de desvanecimiento de feromonas, la cual controla la velocidad con la que estas van disminuyendo en las rutas menos transitadas y permiten una acumulación de feromonas únicamente en las rutas con el mejor desempeño calculado

f. Criterio de parada

El criterio de parada será el encargado de decidir si más hormigas saldrán en busca de alimento o si el algoritmo se detendrá para analizar los resultados más comunes hacia los cuales ha convergido la búsqueda de soluciones.

2.3.2.3. Ejemplo de aplicación de Colonia de hormigas

a. Paso 1: Creación de hormiga

Para comenzar es necesario definir un conjunto de posibles soluciones representadas con hormigas. En este caso se utilizará una sola hormiga encargada de trasladarse del nodo 1 al nodo 5 en la siguiente configuración:

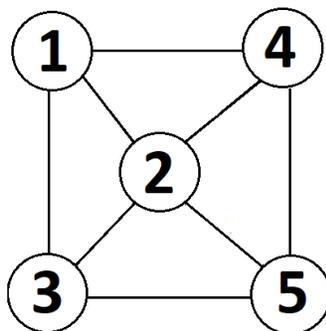


Figura 6. Ejemplo CH: Diagrama general de posibles nodos y rutas

b. Paso 2: Insertar hormiga en el primer nodo

Para continuar ubicamos a la primera hormiga en el nodo 1 en color naranja e identificamos las posibles rutas que esta puede recorrer en color celeste. De igual forma podemos ver el costo de cada ruta en color verde, y la cantidad de feromonas depositada en cada una en color rosa:

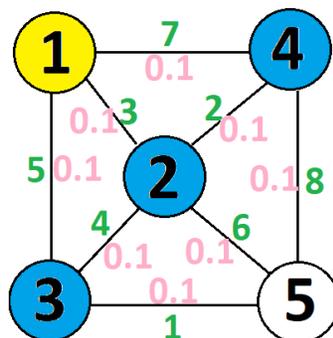


Figura 7. Ejemplo CH: Costos y feromonas disponibles en cada ruta.

El cálculo de probabilidad para cada ruta a seguir se realiza al multiplicar el inverso del costo por la cantidad de feromonas, dividido entre la sumatoria de los costos de cada posible ruta a seguir multiplicada por su respectiva feromona. En el ejemplo planteado obtenemos las siguientes probabilidades en color rojo:

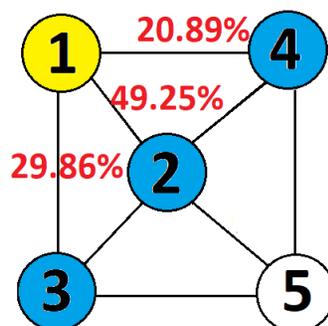


Figura 8. Ejemplo CH: Probabilidades calculadas para cada ruta a partir del nodo 1

En base a las probabilidades calculadas, el camino seleccionado por la hormiga para este ejemplo es hacia el nodo 2.

c. Paso 3: Seleccionar el siguiente nodo

Una vez que la hormiga se ha trasladado al nodo 2, procedemos a realizar el cálculo de probabilidades nuevamente. En este caso se asumirá que la hormiga no puede regresar a un nodo ya visitado. Cabe recalcar que estas consideraciones quedan a criterio del diseñador.

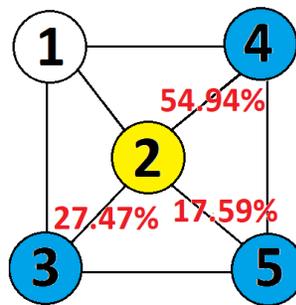


Figura 9. Ejemplo CH: Probabilidades calculadas para cada ruta a partir del nodo 2

Al igual que en el caso anterior, trasladamos la hormiga hacia el siguiente nodo y repetimos el proceso hasta conseguir llegar al nodo final. El siguiente camino a recorrer, al no tener más opciones tendrá una probabilidad de 100% de llegar al nodo final.

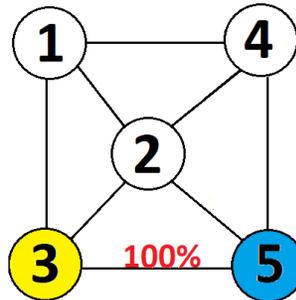


Figura 10. Ejemplo CH: Probabilidades calculadas para cada ruta a partir del nodo 3

d. Paso 4: Depósito y eliminación de feromonas

Para realizar el desvanecimiento de feromonas es necesario multiplicar a las feromonas actuales por un factor de desvanecimiento. Esto permitirá que solo los caminos con un mejor desempeño obtengan las probabilidades más altas de ser seleccionados por las siguientes hormigas.

Una vez alcanzado el objetivo, se calcula el costo total de la ruta recorrida, y se convierte a un valor de feromonas con la ayuda de una constante de aprendizaje, este valor de es agregado a la cantidad de feromonas con las que contaba dicha ruta inicialmente. En el ejemplo planteado los valores resultantes para cada ruta se pueden apreciar a continuación

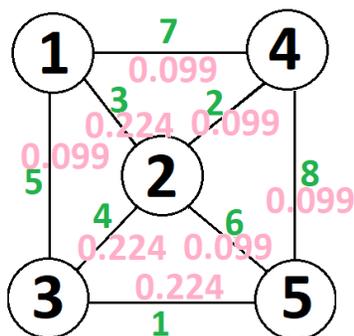


Figura 11. Ejemplo CH: Costo y feromonas actualizadas

e. Paso 5: Criterio de parada

Para este caso se ha definido un criterio de parada de 5 iteraciones, por lo cual el algoritmo repetirá los pasos del 2 al 4 durante 5 repeticiones, finalizando con un camino cargado de feromonas en comparación a los demás. Cabe recalcar que la concentración de feromonas al finalizar el algoritmo dependerá del número de iteraciones utilizadas.

2.3.3. Algoritmo Cuckoo Search

2.3.3.1. Pájaros Cuco

Los pájaros cucos son una familia de aves de mediano tamaño, cuyo nombre proviene de la onomatopeya de su canto, y que se caracterizan por que varias de estas especies que practican el parasitismo de puesta, ya que depositan sus huevos en nidos de otras especies y dejan que estos sean criados por las mismas, eliminando así la necesidad de alimentar y cuidar de sus propias crías.

Para ello, la hembra Cuco deposita su huevo en el nido ocupado, y picotea o tira del nido los huevos del anfitrión, por lo que este no suele darse cuenta del infiltrado en su nido. Los huevos de la hembra Cuco tienen diversos patrones de color, por lo que esta escoge los nidos de las especies de pájaro cuyo patrón de color mimeticen mejor. Al poco tiempo de nacida, la cría de cuco suele arrojar del nido al resto de crías y huevos empujándolos con su espalda, a fin de recibir todo el alimento y cuidado de sus padres adoptivos, hasta que este alcanza un tamaño suficiente para abandonar el nido. Muchas especies de aves aprenden a reconocer los huevos intrusos de los cucos y arrojan a este fuera del nido o lo abandonan para comenzar de nuevo. Así que el cuco intenta constantemente mejorar su mimetismo de los huevos de sus huéspedes, para mantener su estilo de reproducción mediante parasitismo de puesta. (Mascotas, 2015)

El éxito reproductivo de los pájaros cuco se basa en el mimetismo de sus huevos con las especies hospedadoras, respecto al tamaño y fenotipo de colores de sus huevos. Según análisis de ADN, se piensa que esto ocurre gracias a factores genéticos heredados de la madre y que pasan de generación en generación a sus crías, gracias a ello, las hembras se especializan en una sola especie, por lo que suelen poner huevos que se mimetizan en el nido de la especie hospedadora en la cual su línea materna ha logrado sobrevivir. Los machos, por otro lado, pueden fertilizar a las hembras de todas las líneas, manteniendo suficiente flujo genético entre las distintas líneas maternas.

El proceso de reproducción del pájaro cuco se lo puede resumir de la siguiente manera: las hembras cuco pueden producir hasta 25 huevos que serán colocados de uno en uno en distintos nidos previamente seleccionados. Los huevos parásitos tienen un diseño y colores similares a los propios del ave propietaria del nido. Si esta especie no detecta la intrusión, incubará el huevo durante 12 días, por lo que el polluelo cuco generalmente nace anticipado por varios días a los polluelos legítimos; esto le permite al polluelo cuco, deshacerse de los huevos o polluelos recién nacidos, a fin de obtener todo el cuidado de sus padres adoptivos.

La estancia del cuco en el nido tiene una duración de 13 a 20 días, antes de que este pueda emprender una vida independiente. Finalmente, cada hembra cuco se especializa en parasitar una especie de ave en concreto, generalmente la misma que lo crio, por lo que sus huevos tienen un parecido extraordinario con los del anfitrión (Life & BBVA, 2008)

2.3.3.2. Vuelo de Lévy

Los vuelos de Lévy es una técnica capaz de representar la forma en que varios animales e insectos exploran una determinada área mediante desplazamientos en línea recta, marcados por abruptos cambios de dirección. Esta técnica presenta un recorrido aleatorio, cuya longitud de desplazamiento se extrae de una distribución de Lévy con varianza y media infinita. (Flores Vilches, 2015)

El éxito de esta técnica se basa en que un porcentaje de los nuevos movimientos dependen del desempeño de las mejores soluciones obtenidas hasta el momento, acelerando el

proceso de búsqueda, pero sin descuidar el porcentaje restante encargado de generar movimientos al azar y lejos de la solución actual, con el fin de asegurar una exploración eficiente sin estancarse en soluciones locales. (Flores Vilches, 2015)

2.3.3.3. Descripción del Algoritmo

En base a este peculiar comportamiento, Xin-she Yang, profesor de la universidad de Cambridge, y Suash Deb, desarrollaron el algoritmo de optimización Cuckoo Search en el año 2009, inspirados en el parasitismo de puesta que utilizan los pájaros cucos para su reproducción. Dentro de este algoritmo, cada huevo anfitrión en un nido representa una solución, mientras que cada huevo de cuco representa una nueva y mejor solución, por lo cual, el objetivo se basa en utilizar las nuevas y generalmente mejores soluciones (cucos), para reemplazar a las soluciones no tan buenas de los nidos

El algoritmo puede describirse bajo tres ideas fundamentales:

- Cada Cuckoo deposita un huevo en un nido seleccionado al azar
- Los mejores nidos y huevos sobrevivirán y darán paso a la siguiente generación.
- Existe una cantidad definida de nidos disponibles, y cada huevo tiene una probabilidad de ser descubierto. En este caso, la madre abandonará el nido y creará uno nuevo.

(Flores Vilches, 2015)

La siguiente ilustración presenta el diagrama general del algoritmo Cuckoo Search

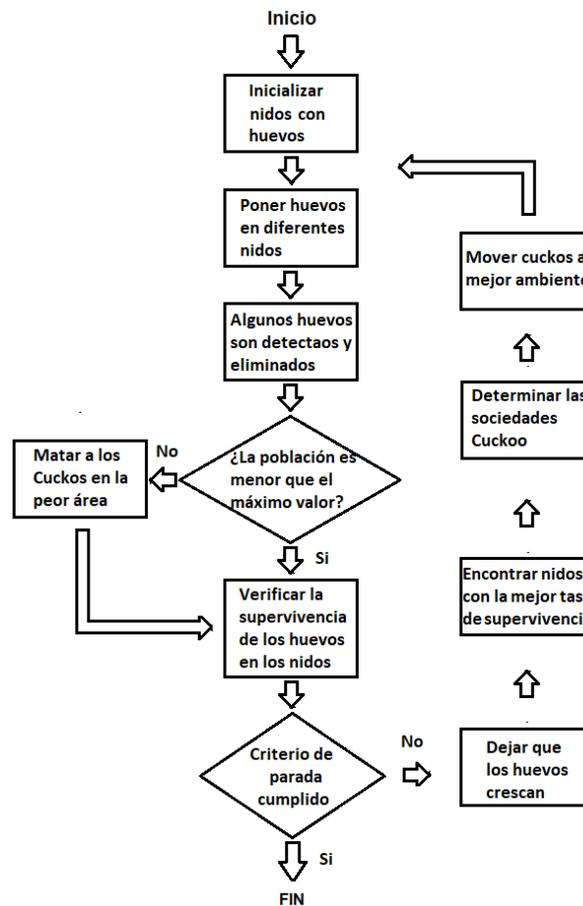


Figura 12. Diagrama general del Algoritmo *Cuckoo Search*

El detalle de cada sección se muestra a continuación:

a. Inicializar nidos con huevos.

La inicialización nidos con huevos hace referencia de la población inicial de posibles soluciones con las que vamos a trabajar, en la cual cada huevo anfitrión representa una

solución. De manera general, esta población se escoge generando un conjunto de soluciones o huevos aleatorios para cada nido.

b. Poner huevos en diferentes nidos

Podemos aplicar la técnica de vuelo de Lévy para abarcar cada nido dentro del área de interés, obteniendo un nuevo cuckoo encargado de colocar una nueva solución en los nidos seleccionados. Cabe recalcar que cada cuckoo pondrá un solo huevo o nueva solución en cada nido.

c. Detección y eliminación de huevos

Una fracción de huevos con el menor desempeño cuenta con una mayor probabilidad de ser rechazados. En base a este valor, cierta cantidad de huevos serán detectados y sus nidos serán abandonados para dar origen a la construcción de nuevos nidos.

d. Máximo valor de población

Para evitar que el conjunto de soluciones crezca de manera indefinida, es necesario mantener un máximo valor de población con el cual, en base a la evaluación del desempeño de cada nido, los sobrevivientes serán los mejores nidos de su generación, evitando el aumento descontrolado de posibles soluciones y mejorando la calidad de cada huevo y nido.

e. Verificación de supervivencia de los huevos

Para verificar la supervivencia de los huevos cuckoo en cada nido, se procede a la comparación del desempeño de los huevos anfitriones con los huevos cuckoo. Si el huevo Cuckoo presenta un desempeño mayor al obtenido por los huevos anfitriones del nido, este irá reemplazandolos hasta convertirse en la única solución del nido.

f. Criterio de cumplimiento

Al igual que los anteriores algoritmos presentados, el criterio de cumplimiento se puede plantear como un objetivo a alcanzar con las soluciones encontradas, o bien como un numero de iteraciones, en cuyo fin se analizará el desempeño alcanzado por cada iteración y se tomará al mejor de estos como solución al problema.

g. Detección de los mejores nidos

Con la ayuda del desempeño calculado en cada nido, podemos seleccionar los más adecuados para la reproducción del cuckoo y definir las áreas donde el algoritmo presenta las mejores soluciones al problema.

h. Traslado de Cuckos al mejor ambiente

Una vez identificados los mejores nidos, podemos trasladar a los siguientes cuckoo a las áreas contenedoras de las mejores soluciones, con el fin de que el algoritmo converja en cada iteración hacia la mejor solución posible.

2.3.3.4. Ejemplo de aplicación de algoritmo Cuckoo Search

a. Paso 1: Inicializar nidos con huevos.

Empezamos el algoritmo con la población inicial de posibles soluciones de manera aleatoria, generando un conjunto de Nidos con una solución en cada uno de ellos;

<i>Sol1</i>	<i>Sol2</i>	<i>Sol3</i>	<i>Sol4</i>
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 3</i>	<i>Nido 4</i>

b. Paso 2: Poner huevos en diferentes nidos

A continuación, obtenemos un nuevo cuckoo y mediante el vuelo de Lévy lo colocamos en uno de los nidos generados como se muestra a continuación:

	<i>Cuckoo1</i>		
<i>Sol1</i>	<i>Sol2</i>	<i>Sol3</i>	<i>Sol4</i>
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 3</i>	<i>Nido 4</i>

c. Paso 3: Detección y eliminación de huevos

Realizamos la evaluación del desempeño de cada una de las soluciones obtenidas en los nidos y del Cuckoo depositado. En caso de que el cuckoo consiga un desempeño mayor, este se encarga de eliminar a las demás soluciones del nido:

	<i>Cuckoo1</i> _{75%}			
<i>Sol1</i> _{43%}	<i>Sol2</i> _{30%}	<i>Sol3</i> _{21%}	<i>Sol4</i> _{53%}	
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 3</i>	<i>Nido 4</i>	

En este caso, el desempeño alcanzado por los el Cuckoo1 es mayor a la solución original en el nido 2, por lo cual el huevo de Cuckoo reemplazará al huevo anfitrión:

<i>Sol1</i> _{43%}	<i>Cuckoo1</i> _{75%}	<i>Sol3</i> _{21%}	<i>Sol4</i> _{53%}	
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 3</i>	<i>Nido 4</i>	

d. Paso 4: Máximo valor de población

Con la creación de nuevos nidos en cada iteración en base a cierta probabilidad definida, es factible pensar que en ocasiones el número de nidos sobrepase el máximo establecido, por lo cual se procede a eliminar aquellos que presenten un menor desempeño en comparación a los demás, asegurando que el algoritmo converja hacia las mejores soluciones.

<i>Sol1</i> _{43%}	<i>Cuckoo1</i> _{75%}	<i>Sol3</i> _{21%}	<i>Sol4</i> _{53%}	<i>Sol5</i> _{37%}
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 3</i>	<i>Nido 4</i>	<i>Nido 5</i>

En este caso asumimos una población máxima de 4 nidos, por lo que al agregar un nido 5, uno de los anteriores deberá ser eliminado:

<i>Sol1</i> _{43%}	<i>Cuckoo1</i> _{75%}	<i>Sol4</i> _{53%}	<i>Sol5</i> _{37%}
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 4</i>	<i>Nido 5</i>

e. Paso 5: Criterio de cumplimiento

Al ser un algoritmo iterativo, criterio de cumplimiento puede definirse como un objetivo a alcanzar, o bien un numero de iteraciones establecido. En el segundo caso, las mejores soluciones las hallaremos en el conjunto de nidos donde el desempeño alcanzado por los huevos cuckoo han sido mayores. En este ejemplo se ha utilizado un criterio de cumplimiento de 5 iteraciones.

5 iteraciones

f. Paso 6: Detección de los mejores nidos

En base al desempeño medido en cada nido, podemos definir las áreas en las que se encuentran los mejores nidos, con el fin de aumentar la probabilidad de que los cuckoos de las próximas generaciones generen sus respuestas en estas áreas. No olvidemos que los vuelos de Lévy evitan que la concentración de soluciones se estanque en un área local y permiten abarcar el espacio de trabajo completo de una mejor manera

<i>Sol1</i> _{43%}	<i>Cuckoo1</i> _{75%}	<i>Sol4</i> _{53%}	<i>Sol5</i> _{37%}
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 4</i>	<i>Nido 5</i>

En este ejemplo podemos definir el área conformada por los nidos 2 y 4 como las mejores de la generación actual.

g. Paso 7: Traslado de Cuckos al mejor ambiente

Una vez identificadas las mejores áreas dentro del espacio, variamos las probabilidades de ser seleccionadas en la siguiente generación. Esta probabilidad influye en los vuelos de Lévy para que permita que los próximos huevos Cuckoo sean depositados en dichas áreas.

<i>Prob: 20.67%</i>	<i>Prob: 36.06%</i>	<i>Prob: 25.48%</i>	<i>Prob: 17.79%</i>
<i>Sol1</i> _{43%}	<i>Cuckoo1</i> _{75%}	<i>Sol4</i> _{53%}	<i>Sol5</i> _{37%}
<i>Nido 1</i>	<i>Nido 2</i>	<i>Nido 4</i>	<i>Nido 5</i>

El algoritmo repetirá cada uno de los pasos presentados, actualizando en cada iteración los nidos actuales y agregando nuevos hasta satisfacer el criterio de cumplimiento

CAPITULO 3

3. DESARROLLO DE CONTROLADORES BIO INSPIRADOS

3.1. Modelado de la Planta

3.1.1.1. Introducción

Con el fin de aplicar el diseño de controladores optimizados a un caso real, se ha planteado la implementación de dichos controladores en el módulo PCT-2 (módulo de temperatura de flujo de aire), disponible en el laboratorio de instrumentación de la Universidad de las Fuerzas Armadas ESPE, para lo cual es necesario obtener un modelo matemático que permita a los algoritmos planteados encontrar el mejor controlador.

La figura 13 representa el modelo físico del sistema a controlar, el cual consta de un sensor IC de estado sólido como elemento de medición, un conducto de flujo de aire controlado mediante un ventilador de aire frío cuyo actuador consiste en una resistencia o niquelina a la entrada del conducto de aire alimentada con una entrada de voltaje. En conclusión, el proceso de control se basa en modificar el voltaje de entrada $V(t)$ en la planta, con el fin de modificar temperatura de salida $T_o(t)$.

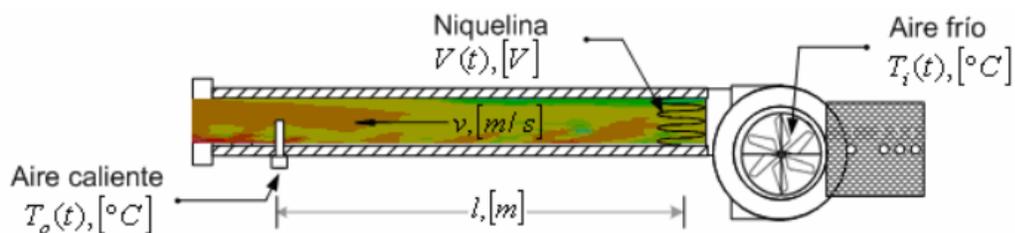


Figura 13. Proceso térmico del modelo PCT-2
Fuente (Miniguano Miniguano, 2008)

La siguiente figura presenta al módulo de temperatura PCT-2 disponible en la Universidad De Las Fuerzas Armadas ESPE:

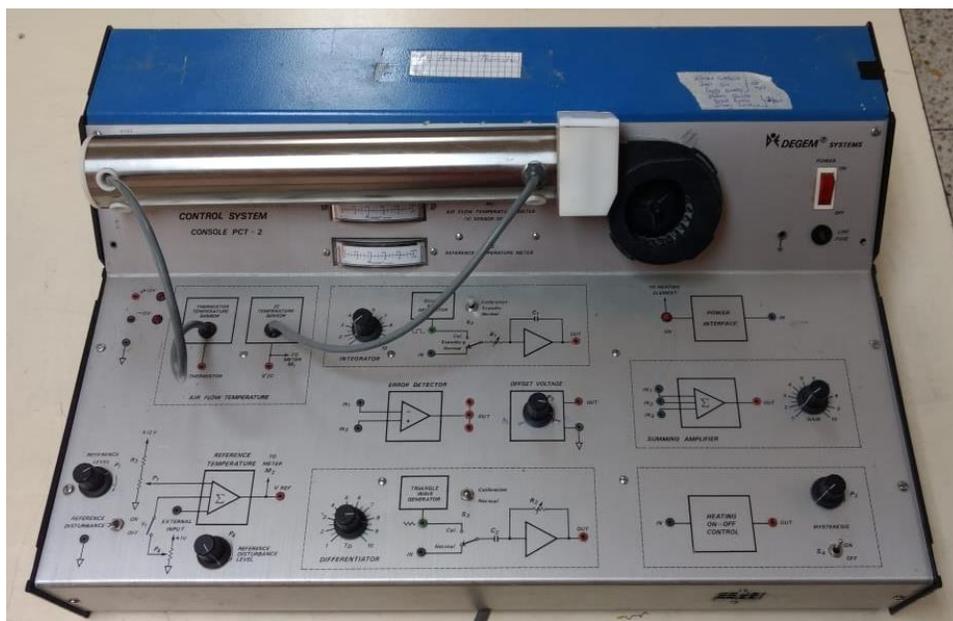


Figura 14. Módulo PCT-2 disponible en el laboratorio de instrumentación de la Universidad de las Fuerzas Armadas ESPE

3.1.1.2. Componentes del sistema

Los componentes principales utilizados para el sistema de adquisición de datos del módulo PCT-2 se describen a continuación:

- **Elemento primario y de medición:** Sensor-trasmisor de temperatura IC de estado sólido. Este componente realiza la medición de temperatura (20 – 70 [°C]) y entrega un voltaje de salida lineal (0 – 5 [V]) hacia el controlador.
- **Controlador:** Arduino UNO. Este componente cuenta con una serie de entradas y salidas analógicas, encargadas de recibir la medida otorgada por el elemento primario y entregar una salida de control en escala de (0 – 5 [V]).
- **Actuador:** Módulo Degem PCT-2. Corresponde al calentador eléctrico de temperatura de acuerdo a la entrada otorgada por el controlador en un rango de 0 – 5 [V].

3.1.1.3. Obtención de la función de transferencia de la planta

El modelado de un sistema consiste en la construcción de modelos matemáticos de un proceso a partir de los datos de salida medidos al aplicar una entrada conocida sobre este. En la búsqueda del modelo más acertado o que mejor represente el comportamiento del sistema a controlar, es necesario determinar un conjunto de parámetros que determinen la estructura del modelo matemático a calcular, tales como número de polos y zeros o retardos que pueda tener la respuesta medida.

En primera instancia es necesario conocer la respuesta del sensor transductor que servirá para la medición de datos. Como ya se mencionó, se ha utilizado un sensor IC de estado sólido con respuesta lineal a las variaciones de temperatura, cuyo voltaje de salida responde de acuerdo a la siguiente tabla

Tabla 1

Respuesta del sensor IC a las variaciones de temperatura

Temperatura [°C]	20	30	40	50	60	70
Voltaje de Salida [V]	0	1	2	3	4	5

De manera gráfica, en la figura 14 podemos verificar la linealidad de la lectura y salida del sensor, así como obtener una ecuación característica que represente el comportamiento del sensor para las próximas interpretaciones de resultados, determinada por:

$$y = 10x + 20 \quad (1)$$

Siendo y la temperatura medida y x la lectura de voltaje a la salida del sensor IC

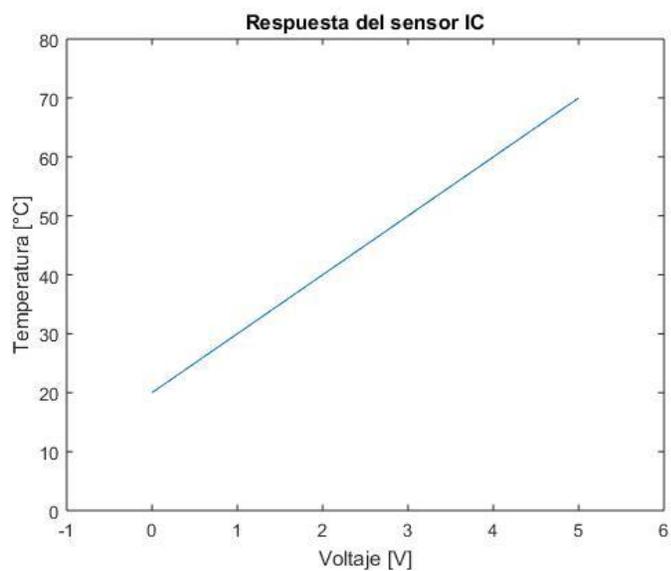


Figura 15. Respuesta del sensor IC a las variaciones de temperatura.

Para continuar con el modelado del sistema, se ha realizado 3 muestreos por un tiempo de 7 minutos con medidas tomadas cada 0.5 segundos y con el sistema sometido a una entrada escalón de 3.3V, obteniendo los siguientes resultados:

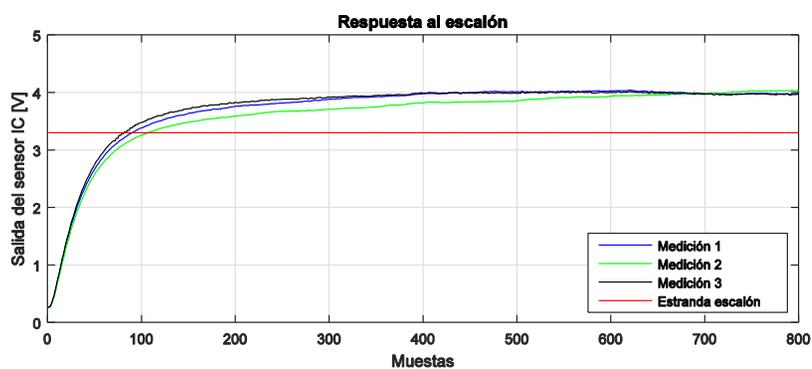


Figura 16. Comparación de muestras realizadas ante entrada escalón de 3.3V

A continuación, se procedió a la selección del grado de la función de transferencia con la cual se representará al sistema. Para ello se utilizó la interfaz *System Identification* de la herramienta computacional utilizada

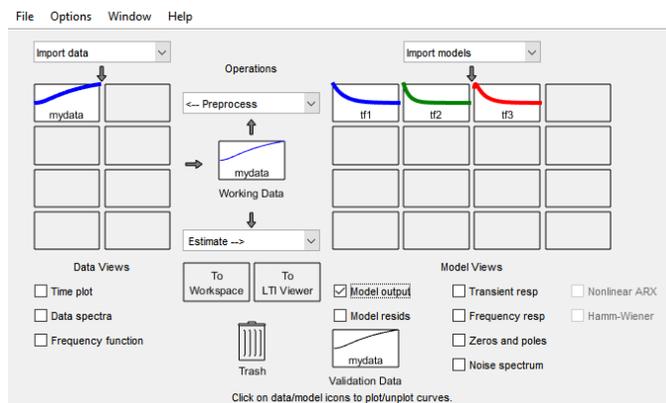


Figura 17. Generación de funciones de transferencia con la herramienta *System Identification*

Con esta herramienta se realizó la estimación del modelo matemático para 3 distintos casos con los siguientes resultados:

Tabla 2

Exactitud medida en las funciones generadas para 3 distintos casos

Caso	Polos	Zeros	Exactitud
1	1	0	98.12
2	2	1	96.89
3	3	1	81.01

La figura 18 presenta una comparación entre la muestra real utilizada y los tres casos de modelado:

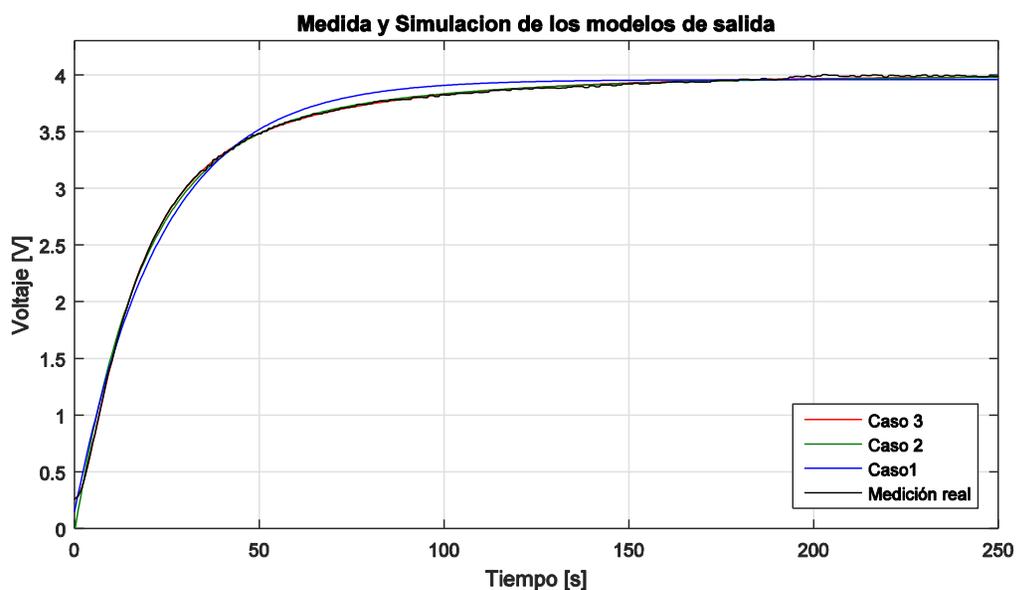


Figura 18. Comparación sistema real con modelos matemáticos generados (Muestra real en negro, caso 1 en azul, caso 2 en verde, caso 3 en rojo).

Ya que el porcentaje de exactitud de los casos 2 y 3 tienen una exactitud de 96% y 97% respectivamente, se ha seleccionado el caso 2 como referencia para los próximos modelados, con el fin de reducir el grado de la función de transferencia sin alejarse del comportamiento real de la planta.

Los modelos calculados para cada una de las señales medidas se pueden observar en la siguiente tabla:

Tabla 3

Funciones de transferencia generadas para cada muestreo realizado

Muestreo	Función de transferencia
1	$\frac{0.06022 s + 0.001021}{s^2 + 0.07464 s + 0.0008394}$
2	$\frac{0.05634 s + 0.002382}{s^2 + 0.05806 s + 0.0001839}$
3	$\frac{0.06106 s + 0.001128}{s^2 + 0.007435 s + 0.0009324}$

Los resultados medidos se pueden observar en la figura 19, mediante una comparación de cada muestreo realizado con su respectiva función de transferencia calculada.

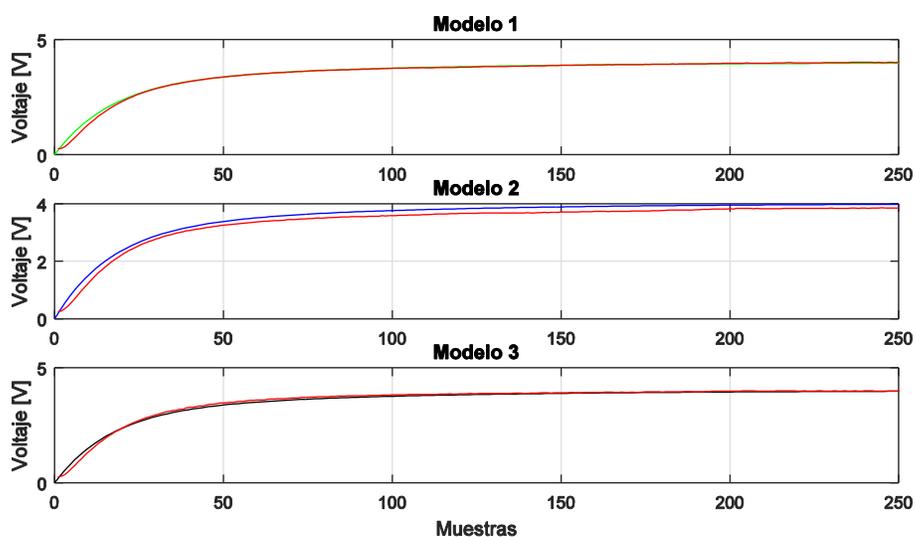


Figura 19. Comparación sistema real - modelo calculada
(En rojo sistema real, verde modelo 1, azul modelo 2, negro modelo 3)

Ya que se requiere utilizar una única función de transferencia que englobe las características de cada una de las funciones generadas, se ha realizado un promedio de los parámetros encontrados en cada una. La función resultante se describe de la siguiente forma:

$$TF = \frac{0.055921 s + 0.007954}{s^2 + 0.06902 s + 0.0006519}$$

La siguiente figura presenta una comparación de la función de transferencia resultante en comparación a las generadas para cada muestreo:

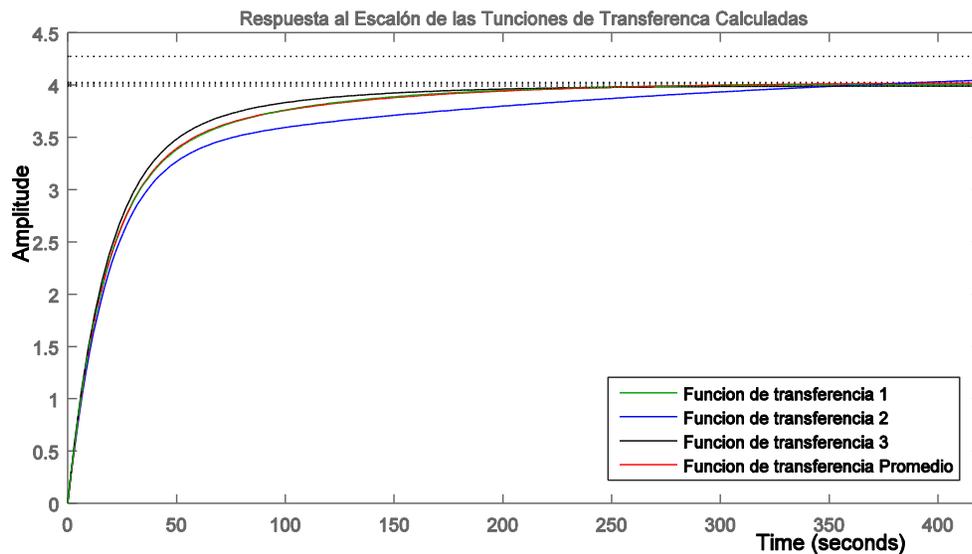


Figura 20. Comparación de las funciones generadas par cada modelo con la función promedio calculada. (En rojo la TF promedio calculada, en azul la TF del modelo 1, en verde la TF del modelo 2, en negro la TF del modelo 3)

3.2. Objetivo de Control

Con el objetivo de realizar la búsqueda de un controlador aceptable para la planta medida, se ha planteado el objetivo de control como un conjunto de condiciones mínimas que el sistema debe cumplir, determinadas en la siguiente tabla:

Tabla 4

Especificaciones a cumplir por el sistema de control

Parámetro	Especificación
Salida	30 °C
Sobrelongación	<15%
Tiempo de establecimiento	<20 segundos
Error de estado estable	2%

Ya que la salida del sensor IC nos entrega una respuesta lineal en escala de 0 a 5 V, es necesario realizar la respectiva conversión de temperatura a voltaje, ya que esta será la lectura a ser manejada por nuestro controlador:

Partiendo de la fórmula (1):

$$x = \frac{y - 20}{10} \quad (2)$$

$$x = \frac{30 - 20}{10} = 1V$$

Por lo tanto, el voltaje de referencia hacia el cual nuestro controlador va a trabajar es de 1V medido desde el sensor IC.

3.3. Sintonización del controlador PID estándar

A continuación, se ha realizado una sintonización estándar de un controlador PID con la ayuda de la herramienta *PID Tuner*, para lo cual se ha ingresado la función de transferencia definida en la sección 3.1 y el tipo de controlador a calcular (en este caso un PID estándar). Los parámetros se han ajustado con el fin de que la salida del sistema obedezca a las especificaciones detalladas en la tabla 4, como se muestra en la siguiente figura

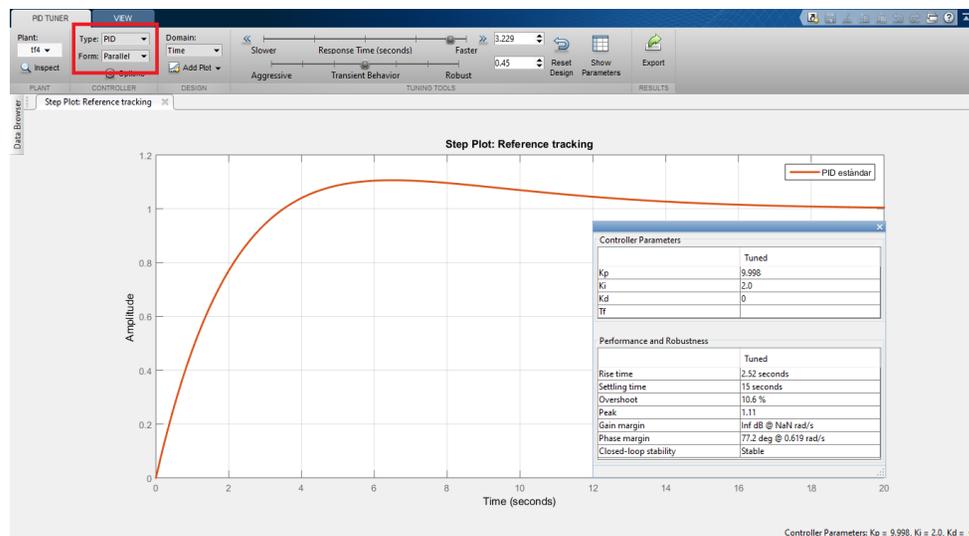


Figura 21. Sintonización del controlador PID por medio de la herramienta PID Tuner de

Los resultados medidos con el controlador encontrado se presentan en la siguiente tabla:

Tabla 5*Parámetros del controlador*

Parámetro	Sintonización
Kp	9.998
Ki	2
Kd	0

Tabla 6*Desempeño y robustez*

	Sintonización
Tiempo de levantamiento	2.52 segundos
Tiempo de estabilización	14.9 segundos
Sobrelongación	10.5 %
Pico	1.1

Con lo cual se ha determinado un controlador PID con la siguiente estructura

$$PID_{estandar} = 9.998 + 2 \frac{1}{s}$$

Podemos observar que la herramienta *PID Tunner* presenta a un controlador PI como la mejor solución para el sistema a controlar. Cabe recalcar que los parámetros calculados por la sintonización realizada serán tomados como base para definir el rango de búsqueda en los algoritmos de optimización.

3.4. Diseño del controlador optimizado por el método de Algoritmos Genéticos

3.4.1. Definición de conceptos

Cromosoma: corresponde la longitud de la solución escrita en código binario

Gen: representa un bit del cromosoma escrito en código binario

Generación: se encarga de controlar el número de veces que el algoritmo es ejecutado en busca de solución

Individuo: representa una posible solución dentro de la población actual.

Población: engloba el conjunto de todos los individuos creados en la generación actual

3.4.2. Consideraciones iniciales

Para dar inicio al algoritmo de optimización, es necesario definir un conjunto de consideraciones iniciales bajo las cuales se va a regir el algoritmo a lo largo de su ejecución, entre las cuales podemos destacar:

Rango de las ganancias (K_p , K_i , K_d): Se encarga de definir un rango de valores dentro del cual el algoritmo buscará la mejor solución.

Porcentaje de cruce: Encargado de definir el porcentaje de ocasiones en que el algoritmo va a ejecutar el procedimiento de cruce

Porcentaje de Mutación: Encargado de definir el porcentaje de ocasiones en que el algoritmo va a ejecutar el procedimiento de cruce

Máximo coeficiente de sobreelongación: define el máximo valor de sobreelongación permitido dentro de la solución, sobre el cual se evaluará a las soluciones encontradas

Máximo tiempo de establecimiento: define el máximo tiempo de establecimiento permitido dentro de la solución, sobre el cual se evaluará a las soluciones encontradas

Error de estado estable: define el máximo valor de error que la señal se debe mantener para considerarse estable.

Tamaño de la población: define el máximo número de individuos contenidos dentro del conjunto de soluciones

3.4.3. Esquema general y desarrollo del algoritmo

Para realizar la simulación del proceso de selección por Algoritmos genéticos se ha determinado un esquema general que engloba todos los pasos presentados en la figura 1, con el fin de simplificar la programación del mismo en los siguientes bloques:

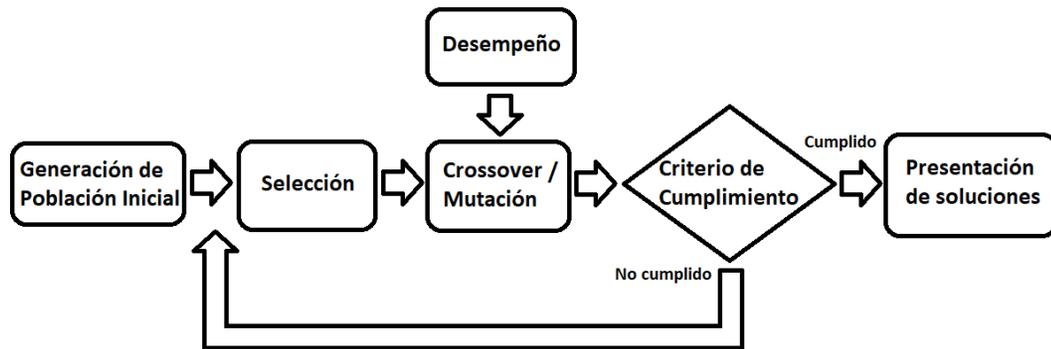


Figura 22. Esquema simplificado de simulación para el método de Algoritmos Genéticos

El detalle de cada bloque será presentado a continuación:

a. Generación de población inicial

El algoritmo inicia con la creación de una población inicial de posibles soluciones generadas de manera aleatoria y contenidas en una matriz; cada solución se presenta en conjunto, con información adicional para facilitar el posterior tratamiento de la misma. El formato de la matriz de soluciones generada se presenta a continuación:

La primera parte de la solución está conformada por los valores generados para las ganancias del controlador (K_p , K_i , K_d), cuya resolución se puede definir por la longitud del cromosoma contenedor de cada ganancia, como se muestra en la siguiente representación:

$$Kp_{bin} = \{b_0 \ b_1 \ b_2 \ \dots \ b_n\} \quad (3)$$

Donde Kp_{bin} representa un cromosoma con una cantidad n de genes b_n que representa el valor de Kp escrito en código binario.

Por otro lado, la solución general está conformada por la concatenación de los valores de las constantes Kp , Ki y Kd , representado tanto en código binario como en su equivalente decimal.

$$Sol_{bin} = [Kp_{bin} Ki_{bin} Kd_{bin}] \quad (4)$$

Donde Sol_{bin} representa una solución escrita en código binario, conformada por los cromosomas de Kp , Ki y Kd concatenados.

Una vez generada la solución aleatoria en forma binaria, se procede a calcular su equivalente decimal para poder trabajar sobre la simulación del controlador. Cabe recalcar que el proceso de conversión binario – decimal se ha resuelto mediante el método de notación posicional, y la solución generada ha sido escalada dentro de los límites deseados para cada constante del controlador.

$$Sol_{dec} = [Kp_{dec} Ki_{dec} Kd_{dec}] \quad (5)$$

Sol_{dec} representa a la solución contenedora de las ganancias Kp, Ki y Kd , escritos en notación decimal.

Para cada solución se ha calculado el desempeño alcanzado con los valores generados mediante una simulación de su respuesta sobre el sistema a controlar, con el fin de asignar esta información a la matriz de poblaciones iniciales.

El vector contenedor de toda la información recopilada se ha representado bajo la siguiente configuración:

$$Solución = [Sol_{bin} Sol_{dec} Desempeño] \quad (6)$$

Donde $Solución$ representa la concatenación de toda la información presentada con anterioridad.

Finalmente, la matriz de poblaciones iniciales está conformada por el conjunto de las soluciones generadas, como se muestra a continuación

$$Población Inicial = \begin{bmatrix} Solución_1 \\ Solución_2 \\ Solución_3 \\ \vdots \\ Solución_m \end{bmatrix} \quad (7)$$

Donde m representa el tamaño definido de la población con la cual se va a trabajar en cada generación.

b. Selección

El proceso de selección de padres para dar origen a la nueva población en cada generación se realiza bajo un criterio probabilístico, determinado en base al desempeño alcanzado por cada solución en comparación a las demás. Podemos calcular este valor en base a la siguiente fórmula:

$$P_i = \frac{J_i}{\sum_j^n J_j} \quad (8)$$

Donde J_i representa el desempeño alcanzado por la solución i

Con la probabilidad de selección de cada solución definida, se ha recurrido al conocido *método de la ruleta* para determinar aquellas soluciones que serán tomadas para dar origen a la siguiente población de soluciones, el cual consiste en simular el giro de una ruleta sobre un espacio formado por las posibles soluciones, y seleccionar aquella en donde la ruleta se detenga, como se puede observar en la siguiente ilustración:

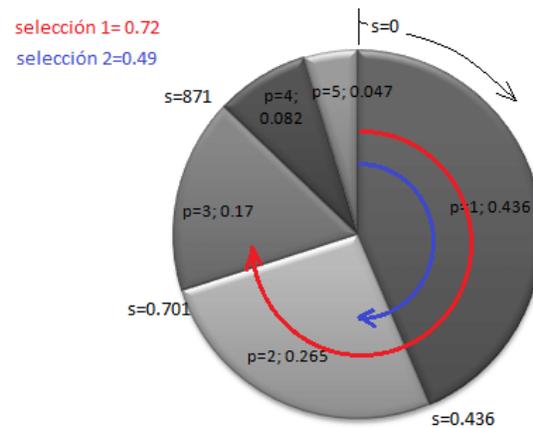


Figura 23. Método de la ruleta para dos selecciones
 Fuente: (Sedano García, 2013)

Se puede asegurar de manera intuitiva, que aquellas soluciones que presentan un mejor desempeño (y por tanto una mayor probabilidad de selección), serán las más escogidas para dar origen a la nueva población de soluciones en la siguiente generación.

c. Crossover / Mutación

Los procesos de crossover y Mutación son los encargados de dar origen a las nuevas soluciones de cada generación, tomando como padres a las soluciones previamente escogidas en el proceso de selección. Ambas cuentan con una probabilidad definida de ser ejecutadas, pero cada una se rige bajo sus propias reglas, como se presenta a continuación:

i. Crossover

Tal como se ha explicado en la sección 2.3.1, el proceso de crossover requiere de la identificación de un punto de cruce para su ejecución, el cual es generado de manera aleatoria entre los genes de los individuos “*Padres*” sobre los cuales se va a dar origen a las nuevas soluciones. Es importante recalcar que el proceso de cruce únicamente se aplica sobre la parte de la solución representada en forma binaria.

El proceso de crossover se ha ejecutado siguiendo las reglas establecidas en dicha sección, y utiliza el bloque adicional *Desempeño* de la figura 22, para agregar la información referente al comportamiento de cada solución sobre el sistema.

En (Holland, 1975) se recomienda una probabilidad de cruce entre 0.6 y 0.9, en cuyas pruebas se ha determinado que una probabilidad en este rango determina mejores soluciones para el algoritmo, por lo que se ha seleccionado una probabilidad de cruce de 0.8 para el desarrollo de este trabajo. Así mismo, se ha utilizado el *cruce basado en un punto* en base a los resultados obtenidos en (Holland, 1975), donde se concluye que utilizar más de un punto de cruce puede provocar una mayor cantidad de rupturas sobre las mejores soluciones que con un solo punto de cruce.

ii. Mutación

Al igual que el proceso de crossover, la ejecución del proceso de mutación depende de una probabilidad definida, aunque mucho menor que la probabilidad de crossover. Son varios los estudios realizados para definir la mejor probabilidad de mutación, sin embargo en (Holland, 1975) se ha definido como una de las mejores opciones a un valor de l^{-1} , siendo l la longitud de la solución representada en código binario.

El detalle de ejecución del proceso de mutación se ha regido de acuerdo a las especificaciones señaladas en la sección 2.3.1.

d. Desempeño

Los procesos de Crossover y Mutación requieren de un medidor de desempeño para adicionar a la solución obtenida, con el fin de que la próxima generación sea capaz de calcular sus nuevas probabilidades de selección en base al desempeño alcanzado por estas.

Para ello, se ha hecho uso de la herramienta *Simulink* para medir el comportamiento del controlador con las soluciones encontradas sobre el sistema a controlar, y extraer la respuesta del sistema para su posterior análisis. El esquema utilizado para medir la respuesta del sistema a cada individuo se puede ver a continuación:

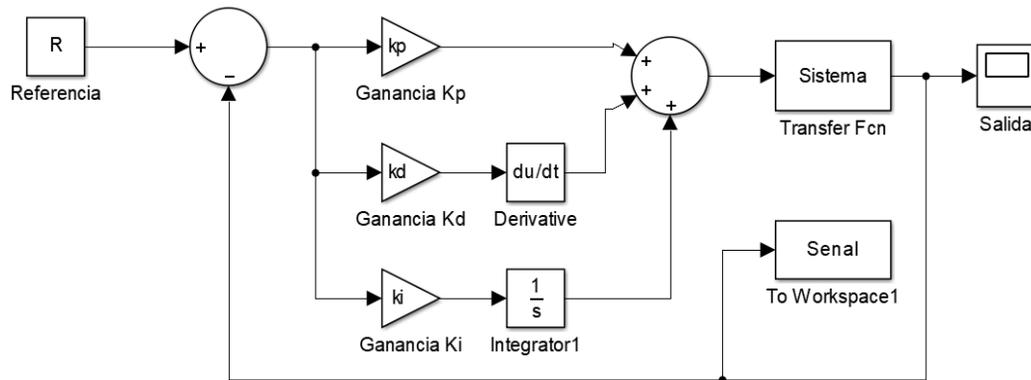


Figura 24. Esquema general del controlador

De acuerdo al esquema planteado, se extrae la respuesta del sistema al controlador aplicado en la variable *Senal* y se trabaja sobre la misma en base a los siguientes aspectos. Los valores deseados para el controlador son definidos al empezar el algoritmo, y en base a estos se ha calculado el desempeño individual y global sobre un controlador, en base a los siguientes parámetros:

- i. Costo en base al porcentaje de sobreelongación

Tomando la respuesta medida del controlador se ha extraído el máximo valor de la señal, la misma que será trabajada con un valor porcentual respecto a la salida deseada.

A continuación, se trabaja con la sobreelongación medida y se compara con la máxima sobreelongación máxima determinada inicialmente, otorgando un valor porcentual que busca reducirse en cada generación. La siguiente ilustración presenta un ejemplo del comportamiento ante una entrada escalón unitario de una solución generada para su posterior tratamiento:

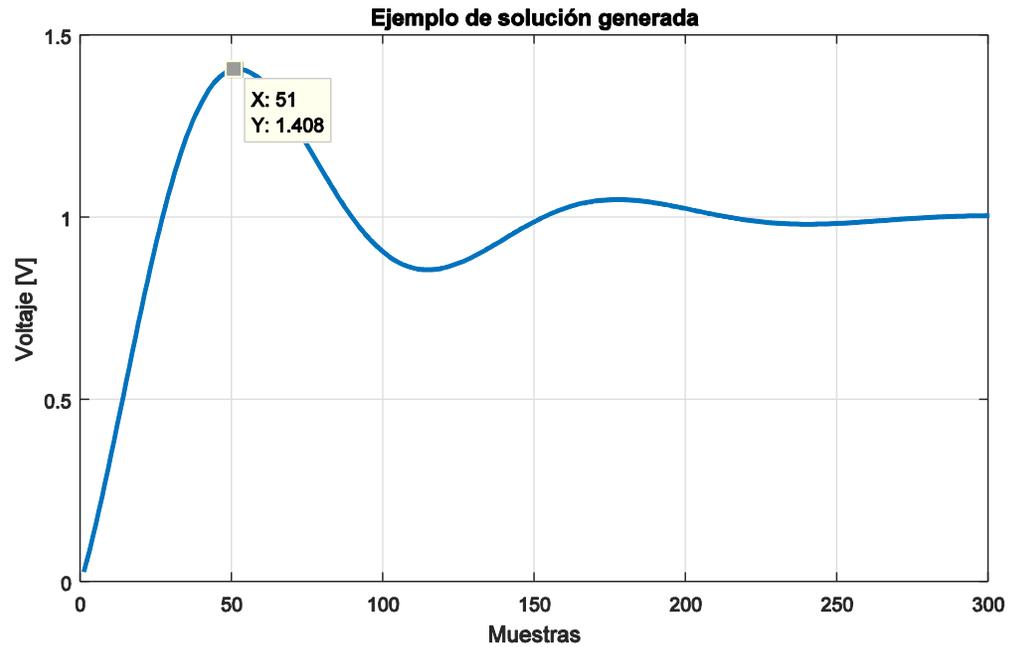


Figura 25. Ejemplo de solución generada para la medición de su sobre-elongación

En la figura 10 podemos observar un máximo valor de 1.393 en la señal medida. Tomando en cuenta que el controlador fue sometido a una entrada escalón unitario el porcentaje de sobre-elongación de la señal es:

$$Mp(\%) = \left(\frac{Mp_{max} - Ref}{Ref} \right) \times 100\% \quad (9)$$

$$Mp(\%) = (1.408 - 1) \times 100\% = 40.8\%$$

Asumiendo que la máxima sobrelongación deseada es de 25%, la señal encontrada tendrá un costo para su sobre-elongación de:

$$J_{Mp} = \frac{Mp(\%)}{Mp_{deseado}(\%)} \quad (10)$$

$$J_{Mp} = \frac{40.8}{25} = 1.632$$

Bajo esta consideración, podemos afirmar que a medida que el coeficiente de sobre-elongación disminuya, el costo del controlador en base a su sobre-elongación también disminuirá.

ii. Costo en base al tiempo de establecimiento

Para calcular el costo de la solución en base al máximo tiempo de establecimiento es necesario definir inicialmente un criterio de error de estado estable. En base a lo establecido en (Ogata, 1997), se ha determinado un error de 2%. A partir del momento en que la señal se mantenga con un error menor al establecido, esta podrá considerarse estable y se procede al cálculo del tiempo que demora en hacerlo. Para el ejemplo presentado con anterioridad podemos observar el momento de estabilización después de 309 muestras, como se presenta a continuación:

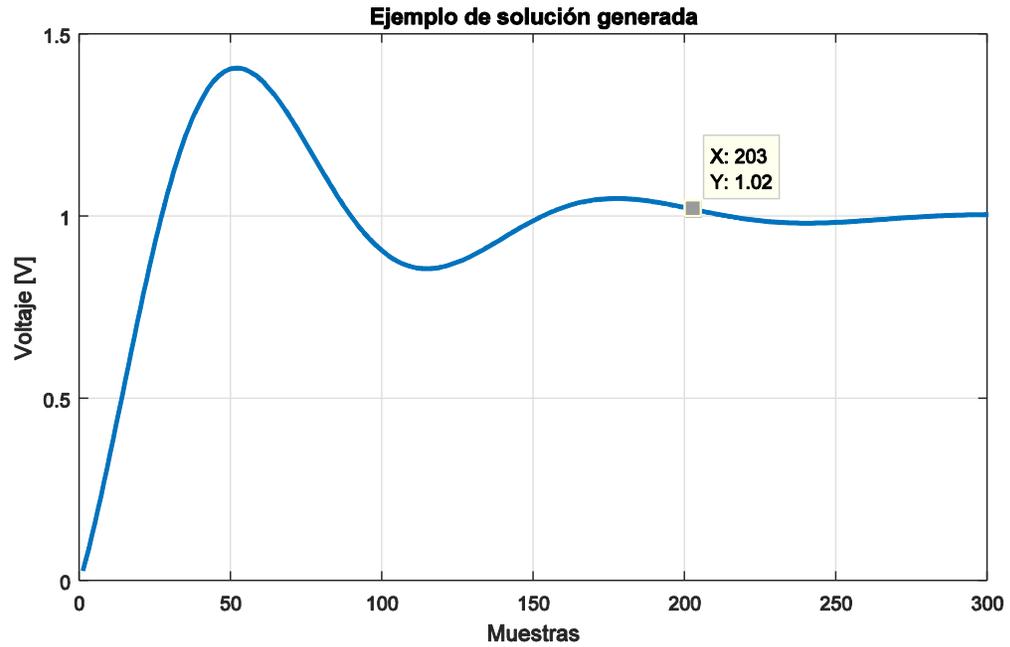


Figura 26. Ejemplo de solución generada para la medición de su tiempo de establecimiento

En base a este punto identificado podemos calcular su tiempo de establecimiento ts a través de la siguiente formula:

$$ts = \frac{\text{punto}}{\text{muestras}} t \quad (11)$$

Donde *punto* representa a la posición de la muestra a partir de la cual la señal se considera estable; *muestras* es el número total de muestras tomadas de la señal obtenida; y t representa el tiempo de ejecución de la simulación.

Bajo esta consideración, podemos definir un tiempo de establecimiento para este ejemplo de:

$$t_s = \frac{203}{400} \times 40 = 20.3 \text{ segundos}$$

Tomando en cuenta un máximo tiempo de establecimiento deseado de 25 segundos, podemos calcular el costo del controlador obtenido en base a la siguiente fórmula

$$J_{ts} = \frac{t_{s_{medido}}}{t_{s_{deseado}}} \quad (12)$$

$$= \frac{20.3}{25} = 0.812$$

Al igual que en el caso anterior, podemos afirmar que a medida que el tiempo de estabilización disminuya, el costo del controlador en base a su tiempo de estabilización también disminuirá.

iii. Desempeño global

Finalmente, se puede agrupar ambos costos bajo un criterio de importancia de cada uno, en base a la siguiente ecuación:

$$J = \frac{1}{J_{Mp} \times C1 + J_{ts} \times C2} \quad (13)$$

Donde C1 define un porcentaje de importancia para J_{Mp} y C2 es el complemento de C1.

Para el ejemplo presentado se ha definido que tanto el coeficiente de sobrelongación como el tiempo de establecimiento tienen el mismo valor de importancia, y tomando en cuenta que el desempeño está definido como el inverso del costo, podemos definir un desempeño global de la solución analizada como:

$$J = \frac{1}{(1.632 \times 0.5) + (0.812 \times 0.5)} = 1.222$$

Finalmente, la optimización del algoritmo se basará en conseguir el mejor desempeño global de cada solución.

e. Criterio de cumplimiento

Como ya se ha señalado en la sección 2.3.1, el criterio de cumplimiento se puede definir en base a un número de iteraciones realizadas o a un objetivo por alcanzar. En este caso el objetivo de control se ha detallado como valores máximos de sobrelongación y tiempo de establecimiento, por lo que no existe una única solución que cumpla los requisitos marcados. En base a esto se ha seleccionado un criterio de cumplimiento basado en 50 iteraciones, sobre las cuales los resultados obtenidos dejan de presentar cambios significativos y no aportan mejoras al sistema.

f. Presentación de soluciones

De acuerdo a las consideraciones tomadas a lo largo de la ejecución del algoritmo, los resultados pertinentes para la selección del mejor controlador que se presentan son:

i. Matriz de soluciones

Engloba el conjunto de las mejores soluciones encontradas al finalizar el algoritmo, almacenando la solución con el mejor desempeño global de cada generación.

ii. Desempeño de los controladores

Presenta una gráfica general del desempeño obtenido por las mejores soluciones de cada iteración. Esta grafica se presenta dividida en 3 secciones que permiten analizar los máximos locales y el máximo global obtenido al finalizar el algoritmo para 3 casos de interés: Desempeño global, desempeño por sobrelongación y desempeño por tiempo de establecimiento.

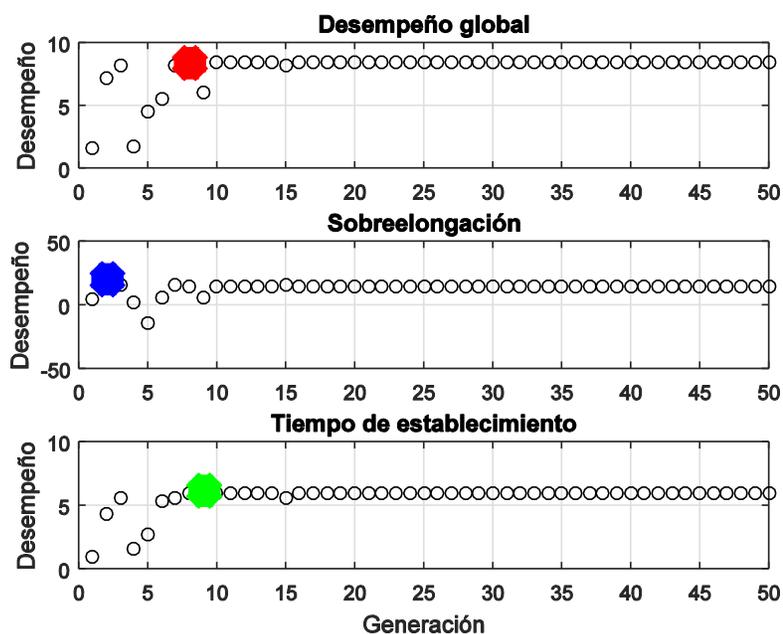


Figura 27. Ejemplo de resultado: desempeño de controladores

El punto rojo representa la mejor solución global localizada durante la ejecución del algoritmo, el punto azul representa la mejor solución encontrada respecto a la sobreelongación medida, mientras que el punto verde indica la mejor solución encontrada respecto al tiempo de establecimiento. Cabe recalcar que esta configuración de colores se mantendrá para las siguientes gráficas tanto en el presente algoritmo, como en las soluciones presentadas por los algoritmos *Colonia de Hormigas* y *Cuckoo Search*.

iii. Costo de los controladores

Presenta una gráfica similar a la mostrada con el desempeño de los controladores, pero mostrando el costo, definido como el inverso del desempeño.

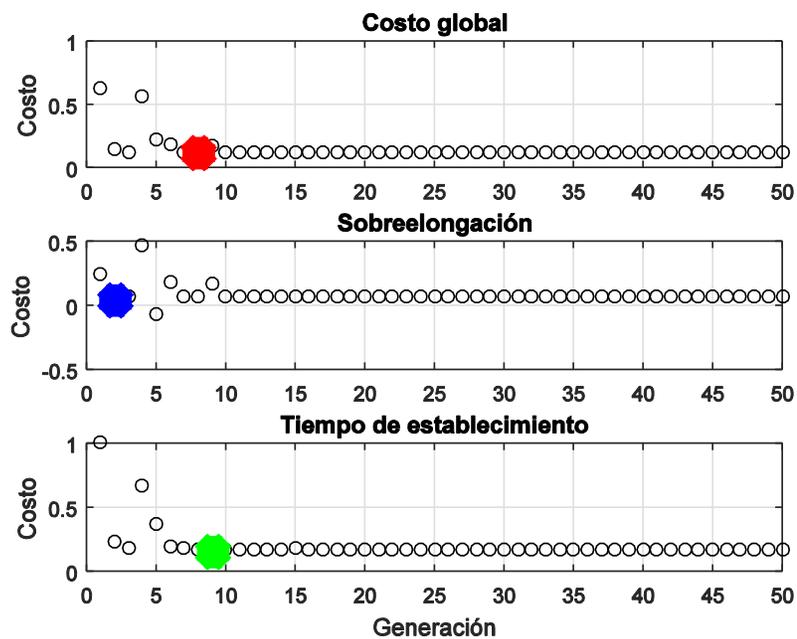


Figura 28. Ejemplo de resultado: costo de controladores

iv. Desempeño global de las soluciones

Presenta una gráfica en 3 dimensiones que localiza la posición en el espacio del desempeño para cada una de las soluciones encontradas.

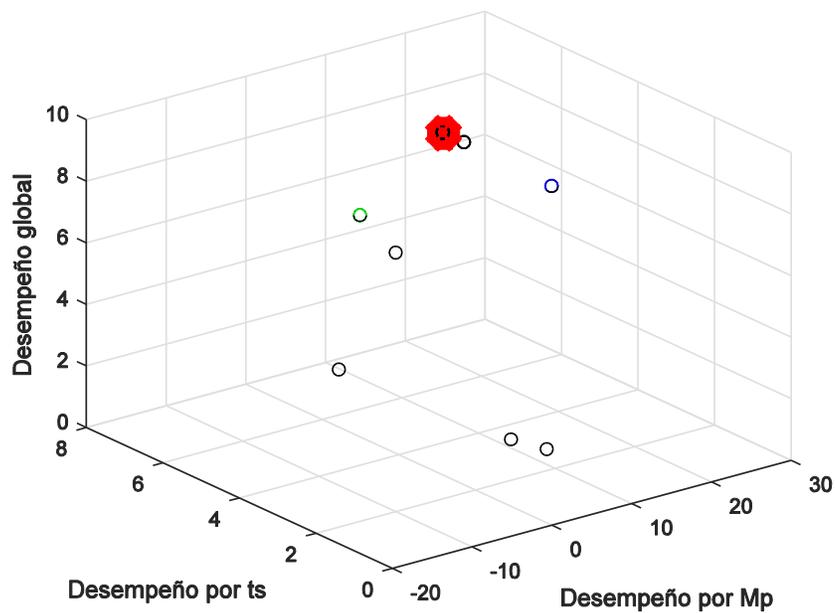


Figura 29. Ejemplo de resultado: Desempeño global de soluciones

v. Comparación de controladores

Presenta un gráfico comparativo de las mejores soluciones en cuanto al desempeño global, desempeño por sobre-elongación y desempeño por tiempo de establecimiento.

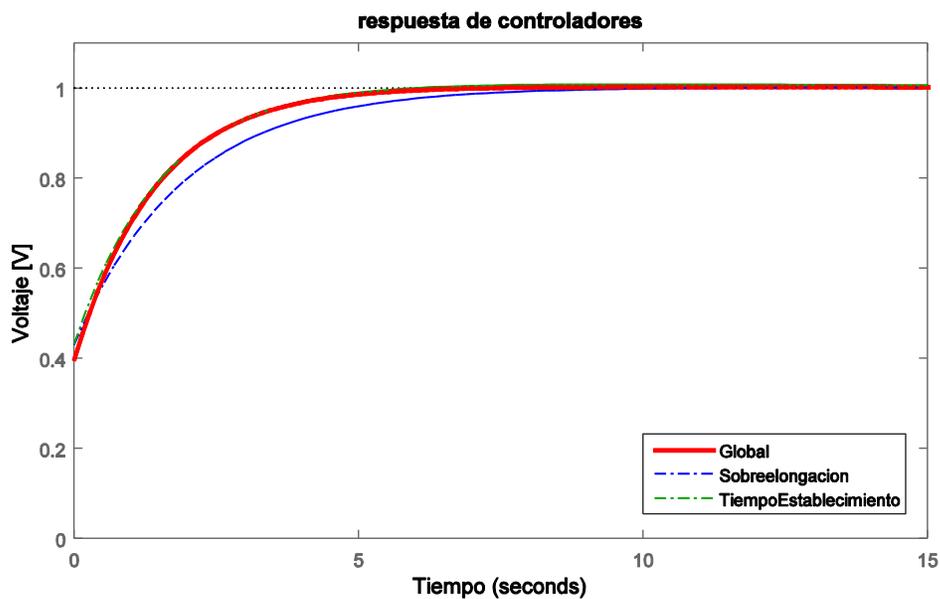


Figura 30. Ejemplo de resultado: Respuesta de los controladores

3.4.4. Resultados obtenidos de la aplicación del algoritmo

El detalle de la aplicación del algoritmo planteado sobre el software dedicado para el efecto se puede apreciar en el **Anexo A**

Los resultados obtenidos luego de la ejecución del algoritmo se pueden observar en la siguiente tabla:

Tabla 7

Resultados sintonizados con la aplicación del método Algoritmos Genéticos implementado

	Kp	Ki	Kd
Con mejor desempeño global	11.7176	0.4706	1.5216
Con mejor desempeño por sobre-elongación	9.7412	0.4706	1.2549
Con mejor desempeño por tiempo de establecimiento	11.9059	0.5686	0.5176

La respuesta al escalón de la solución encontrada se puede observar en la siguiente figura:

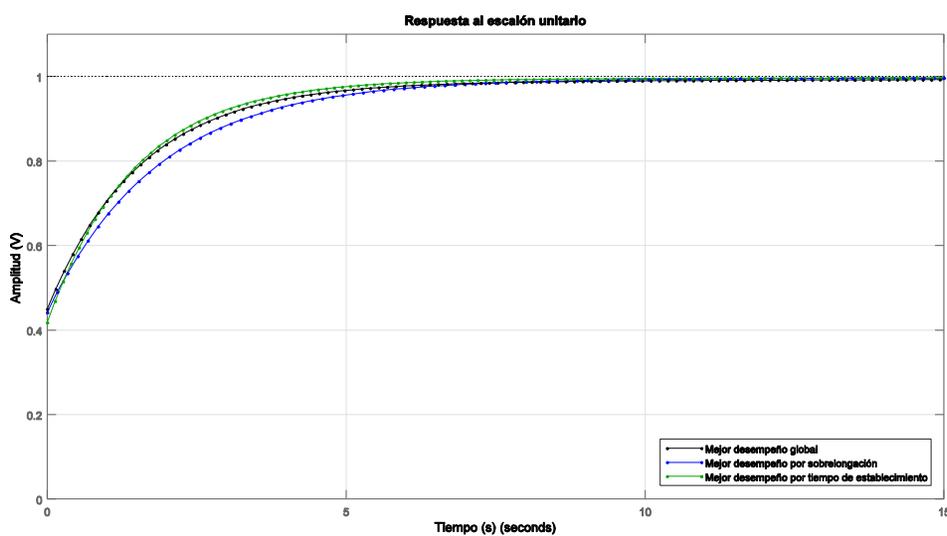


Figura 31. Resultados obtenidos de la aplicación de Algoritmos genéticos en la sintonización de ganancias

3.5. Diseño del controlador optimizado por el algoritmo Colonia de Hormigas

3.5.1. Definición de conceptos

Hormiga: representa una posible solución dentro de la población actual

Factor de desvanecimiento: coeficiente que controla la velocidad de desvanecimiento en cada ruta

Ruta: Conjunto de nodos recorridos para llegar a una solución

Longitud: Distancia o costo medido entre dos nodos

Visibilidad: inverso de la longitud

Feromonas: Factor que aumenta la probabilidad de selección de una ruta

3.5.2. Consideraciones iniciales

Para dar inicio al algoritmo de optimización, es necesario definir un conjunto de consideraciones iniciales bajo las cuales se va a regir el algoritmo a lo largo de su ejecución, entre las cuales podemos destacar:

Rango de las ganancias (K_p , K_i , K_d): Se encarga de definir un rango de valores dentro del cual el algoritmo buscará la mejor solución.

Máximo coeficiente de sobrelongación: define el máximo valor de sobrelongación permitido dentro de la solución, sobre el cual se evaluará a las soluciones encontradas

Máximo tiempo de establecimiento: define el máximo tiempo de establecimiento permitido dentro de la solución, sobre el cual se evaluará a las soluciones encontradas

Error de estado estable: define el máximo valor de error que la señal se debe mantener para considerarse estable.

Tamaño de la población: define el máximo número de individuos contenidos dentro del conjunto de soluciones

Índice de aprendizaje (Q): influye sobre la cantidad de feromonas depositadas por una ruta recorrida

Feromonas iniciales: define la cantidad de feromonas iniciales que tiene cada ruta

3.5.3. Esquema general y desarrollo del algoritmo

Para realizar la simulación del proceso de optimización por Colonia de hormigas se ha determinado un esquema general que engloba todos los pasos presentados en la figura 5, con el fin de simplificar la programación del mismo en los siguientes bloques:

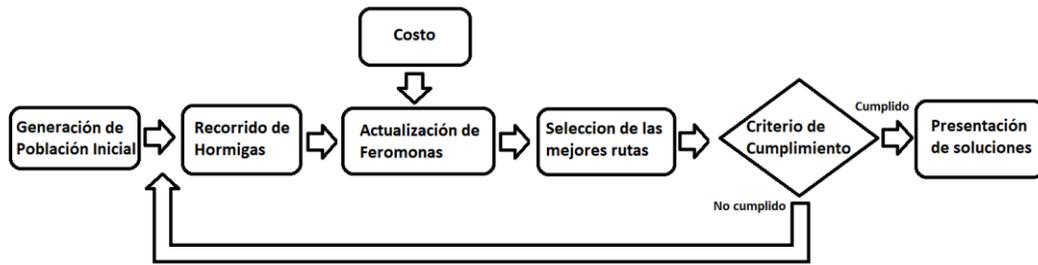


Figura 32. Esquema simplificado de simulación para el método de Colonia de Hormigas

El detalle de cada bloque será presentado a continuación:

a. Generación de población inicial

El algoritmo da inicio con la creación de una población inicial de hormigas, generadas de manera aleatoria y contenidas en una matriz en conjunto con el desempeño alcanzado por cada una.

La primera parte del algoritmo está conformada por los valores de los nodos generados de manera aleatoria para el recorrido de cada hormiga hasta localizar una solución:

$$nodos = \begin{Bmatrix} Kp_1 & Ki_1 & Kd_1 \\ Kp_2 & Ki_2 & Kd_2 \\ Kp_3 & Ki_3 & Kd_3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ Kp_n & Ki_n & Kd_n \end{Bmatrix} \quad (14)$$

Donde n representa la cantidad de nodos que serán generados para cada ganancia del controlador. Cabe recalcar que la configuración presentada está determinada para que cada hormiga recorra un nodo de cada columna hasta la solución final.

A continuación, se utiliza una matriz encargada de almacenar las feromonas localizadas en cada posible ruta a recorrer, en base a la siguiente configuración:

$$feromonas = \left\{ \begin{array}{ccc} f_{11} & \dots & f_{1n} \\ f_{21} & \dots & f_{2n} \\ f_{31} & \dots & f_{3n} \\ \vdots & \dots & \vdots \\ \cdot & \dots & \cdot \\ f_{n1} & \dots & f_{nn} \end{array} \right\} \quad (15)$$

Donde f_{11} representa las feromonas en las ruta $Kp_1 - Ki_1$

Para abarcar todas las rutas disponibles dentro de configuración presentada se ha realizado tres matrices de feromonas con la siguiente configuración:

Desde el origen hasta el nodo Kp

$$feromonas_1 = [f_1 \quad \dots \quad f_n]_{1n} \quad (16)$$

Desde el nodo Kp hasta el nodo Kd

$$feromonas_2 = \begin{bmatrix} f_{11} & \dots & f_{1n} \\ \vdots & \vdots & \vdots \\ f_{n1} & \dots & f_{nn} \end{bmatrix}_{nn} \quad (17)$$

Desde el nodo Kd hasta el nodo Ki

$$feromonas_3 = \begin{bmatrix} f_{11} & \dots & f_{1n} \\ \vdots & \vdots & \vdots \\ f_{n1} & \dots & f_{nn} \end{bmatrix}_{nn} \quad (18)$$

Cada una de las matrices contiene un valor de feromonas inicial del mismo valor para cada una de las rutas, con lo cual se asegura que la generación inicial de soluciones sea aleatoria.

b. Recorrido de hormigas

A continuación, se procede a soltar un conjunto de hormigas encargadas de buscar las soluciones iniciales sobre las cuales partirá el algoritmo. Cada una de estas trabaja de forma independiente y sus recorridos son almacenados en una matriz de soluciones iniciales en conjunto con el desempeño alcanzado por cada una de ellas:

$$soluciones_actuales = \begin{bmatrix} Kp_1 & Ki_1 & Kd_1 & desempeño_1 \\ Kp_2 & Ki_2 & Kd_3 & desempeño_2 \\ \vdots & \vdots & \vdots & \vdots \\ Kp_m & Ki_m & Kd_m & desempeño_m \end{bmatrix} \quad (19)$$

Donde m es la cantidad de hormigas utilizadas en el algoritmo

La selección de cada nodo dentro del recorrido se rige de manera probabilística en base a la cantidad de feromonas depositadas en la ruta, de acuerdo a las siguientes formulas:

$$P_{xy} = \frac{f_{xy}n_{xy}}{\sum_i^n f_{xy}(i) n_{xy}(i)} \quad (20)$$

Donde P_{xy} representa la probabilidad de selección de la ruta entre el nodo x y el nodo y , f_{xy} representa la cantidad de feromonas depositadas en la ruta xy y n_{xy} representa la visibilidad entre los nodos x y y . La visibilidad puede ser calculada por:

$$n = \frac{1}{l} \quad (21)$$

Donde l representa la distancia entre los nodos involucrados.

Dentro del planteamiento del algoritmo se ha definido a cada nodo por recorrer como una ganancia del controlador, sin embargo, al no mantenerse dentro de un mismo espacio, no se puede medir una distancia entre ellos, por lo que se ha definido una distancia unitaria en todos los casos para el respectivo cálculo de visibilidad.

Al igual que en el caso de las feromonas, cada recorrido cuenta con su probabilidad de selección por la siguiente hormiga, por lo cual se han definido 3 matrices encargadas de almacenar las probabilidades de cada una de las rutas disponibles durante el recorrido de las hormigas

Desde el origen hasta el nodo Kp

$$probabilidad_1 = [P_1 \quad \dots \quad P_n]_{1n} \quad (22)$$

Donde P_i es la probabilidad P de seleccionar el camino i

Desde el nodo Kp hasta el nodo Kd

$$probabilidad_2 = \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{n1} & \dots & P_{nn} \end{bmatrix}_{nn} \quad (23)$$

Desde el nodo Kd hasta el nodo Ki

$$probabilidad_3 = \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{n1} & \dots & P_{nn} \end{bmatrix}_{nn} \quad (24)$$

Una vez obtenidas las probabilidades de cada ruta, se ha realizado un proceso de selección de rutas basadas en dichas probabilidades para cada una de las hormigas,

c. Desempeño

Al igual que en el diseño del método de optimización por algoritmos genéticos, se ha utilizado la herramienta *Simulink* para la evaluación de la respuesta del controlador seleccionado y se ha definido los criterios de optimización en base al costo individual de cada parámetro de interés, tales como:

- i. Costo en base al porcentaje de sobreelongación

Se realizó el cálculo del porcentaje de sobre elongación en base a la señal obtenida de la simulación, a través de la (9). Con el porcentaje de sobre elongación obtenido, se procede a calcular su relación en comparación al máximo porcentaje de sobre elongación deseado, haciendo uso de la formula (10).

De manera intuitiva, podemos afirmar que a medida que el coeficiente de sobre-elongación disminuya, el costo del controlador en base a su sobre-elongación también disminuirá.

ii. Costo en base al tiempo de establecimiento

Para calcular el costo de cada hormiga en base al máximo tiempo de establecimiento deseado, es necesario definir inicialmente un criterio de error de estado estable; en este caso se ha definido un error de estado estable de 0.5%. A partir del momento en que la señal se mantenga con un error menor al establecido, esta podrá considerarse estable y se procede al cálculo del tiempo que demora en hacerlo.

El cálculo del tiempo de establecimiento medido se realiza en base a un punto identificado, a partir del cual la señal se considera estable por medio de la fórmula (11). Tomando en cuenta el máximo tiempo de establecimiento deseado, podemos calcular el costo del controlador obtenido en base a la fórmula (12).

Al igual que en el caso anterior, podemos afirmar que a medida que el tiempo de estabilización disminuya, el costo del controlador en base a su tiempo de estabilización también disminuirá.

iii. Costo global

Finalmente, se puede agrupar ambos costos bajo un criterio de importancia de cada uno, en base a la siguiente ecuación:

$$J = J_{Mp} * C1 + J_{ts} * C2 \quad (25)$$

Donde C1 define un porcentaje de importancia para J_{Mp} y C2 es el complemento de C1.

Finalmente, la optimización del algoritmo se basará en conseguir el costo global más bajo de cada solución, y por lo tanto el mejor desempeño de los controladores.

d. Actualización de feromonas

Una vez obtenido el costo o el desempeño de cada hormiga, se procede a la actualización de feromonas en cada una de las rutas utilizadas por la población de hormigas.

Este procedimiento consta de dos partes fundamentales:

i. Evaporación de feromonas

En cada iteración, se realiza la evaporación de feromonas en todas las rutas disponibles, con el fin de evitar una concentración masiva de estas y retrasar el proceso de selección de la mejor ruta en el algoritmo. Para ello se hace uso de la siguiente fórmula

$$f_{xy} = (1 - \rho)f_{xy} \quad (26)$$

Donde f_{xy} representa la cantidad de feromonas en la ruta del nodo x al nodo y , y ρ representa un factor de desvanecimiento (entre 0 y 1), encargado de controlar el porcentaje de feromonas que se desvanece en cada iteración

ii. Adición de feromonas

Una vez finalizado el recorrido de cada hormiga, estas depositan cierta cantidad de feromonas en la ruta recorrida en base al costo obtenido por cada solución, haciendo uso de la siguiente fórmula:

$$f_{xy} = f_{xy} + \sum_k^m \Delta f_{xy}(k) \quad (27)$$

Donde f_{xy} representa la cantidad de feromonas actuales en la ruta xy , m es la cantidad de hormigas de la población, y además

$$\Delta f_{xy} = \begin{cases} Q/L_k & \text{si la hormiga } k \text{ recorrió la ruta } xy \\ 0 & \text{si no} \end{cases} \quad (28)$$

Donde Q es el factor de aprendizaje que controla la influencia del costo calculado sobre la adición de feromonas, y L_k es el costo obtenido de la hormiga k sobre la ruta xy

Con este procedimiento se actualiza la cantidad de feromonas en las rutas disponibles para cada iteración realizada. Se puede intuir que aquellas soluciones con un menor costo (o mayor desempeño) agregarán una mayor cantidad de feromonas a sus caminos recorridos y, por consiguiente, mejorarán la probabilidad de selección de dichas rutas para las siguientes hormigas.

e. Selección de las mejores rutas

Adicionalmente a la matriz de soluciones actuales creada al inicio del algoritmo, se ha determinado una matriz de las soluciones locales en cada iteración, con el fin de almacenar los resultados obtenidos por las hormigas con un mayor desempeño sobre el sistema, para su posterior selección al finalizar el algoritmo.

$$soluciones_globales = \begin{bmatrix} Kp_1 & Ki_1 & Kd_1 & desempeño_1 \\ Kp_2 & Ki_2 & Kd_3 & desempeño_2 \\ \vdots & \vdots & \vdots & \vdots \\ Kp_s & Ki_s & Kd_s & desempeño_s \end{bmatrix} \quad (29)$$

Donde s representa el número de iteraciones realizadas a lo largo del algoritmo.

f. Criterio de cumplimiento

Para este diseño se ha determinado al objetivo de control como como los valores máximos de sobre elongación y tiempo de establecimiento, por lo que no existe una única solución que cumpla los requisitos marcados. En base a esto se ha seleccionado un criterio de cumplimiento basado en 50 iteraciones.

g. Presentación de soluciones

Los siguientes resultados presentan un estilo similar a los presentados en el diseño de controladores aplicando Algoritmos Genéticos, con el fin de mejorar su comparación en futuros análisis de resultados. De igual forma, en base a las consideraciones tomadas a lo largo de la ejecución del algoritmo, los resultados pertinentes para la selección del mejor controlador que se presentan son:

i. Matriz de soluciones

Engloba el conjunto de las mejores soluciones encontradas al finalizar el algoritmo, almacenando la solución con el mejor desempeño global de cada generación.

ii. Recorrido de las hormigas

Presenta un conjunto de gráficas acerca del recorrido de las hormigas a lo largo de la ejecución del algoritmo, como se muestra a continuación:

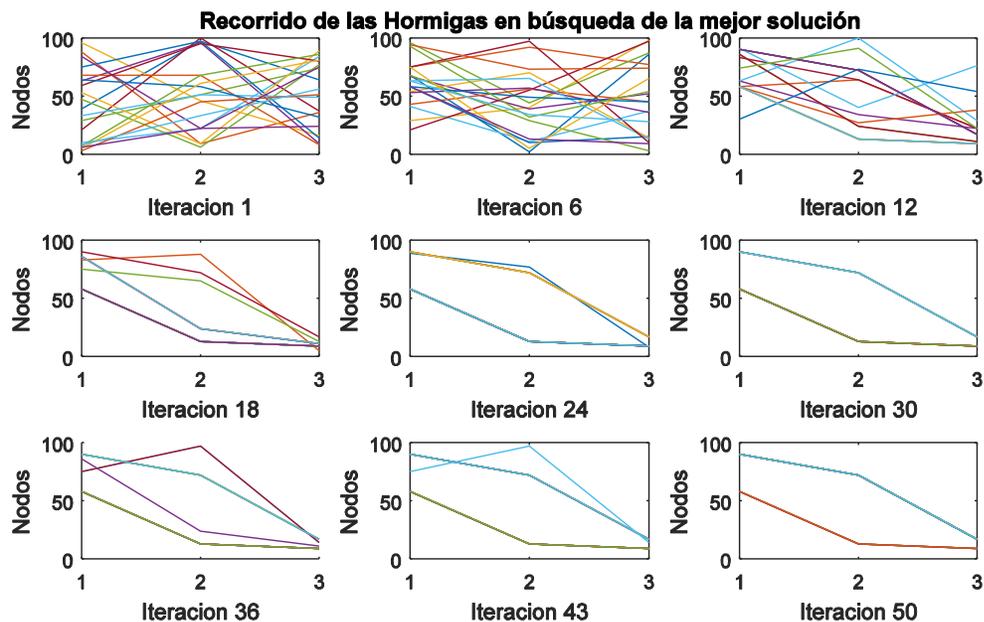


Figura 33. Ejemplo recorrido de hormigas durante el algoritmo.

iii. Desempeño de los controladores

Presenta una gráfica general del desempeño obtenido por las mejores soluciones de cada iteración. Esta grafica se presenta dividida en 3 secciones que permiten analizar los máximos locales y el máximo global obtenido al finalizar el algoritmo para 3 casos de interés: Desempeño global, desempeño por sobreelongación y desempeño por tiempo de establecimiento.

iv. Costo de los controladores

Presenta una gráfica similar a la mostrada con el desempeño de los controladores, pero mostrando el costo, definido como el inverso del desempeño.

v. Desempeño global de las soluciones

Presenta una gráfica en 3 dimensiones que localiza la posición en el espacio del desempeño para cada una de las soluciones encontradas.

vi. Comparación de controladores

Presenta un gráfico comparativo de las mejores soluciones en cuanto al desempeño global, desempeño por sobre elongación y desempeño por tiempo de establecimiento.

3.5.4. Resultados obtenidos de la aplicación del algoritmo

El detalle de la aplicación del algoritmo planteado sobre el software dedicado para el efecto se puede apreciar en el **Anexo B**

Los resultados obtenidos luego de la ejecución del algoritmo se pueden observar en la siguiente tabla:

Tabla 8

Resultados sintonizados con la aplicación del método Colonia de Hormigas implementado

	Kp	Ki	Kd
Con mejor desempeño global	10.3652	0.5683	0.5265
Con mejor desempeño por sobre-elongación	7.3847	0.3929	0.0697
Con mejor desempeño por tiempo de establecimiento	10.4244	0.5690	1.2370

La respuesta al escalón de la solución encontrada se puede observar en la siguiente figura:

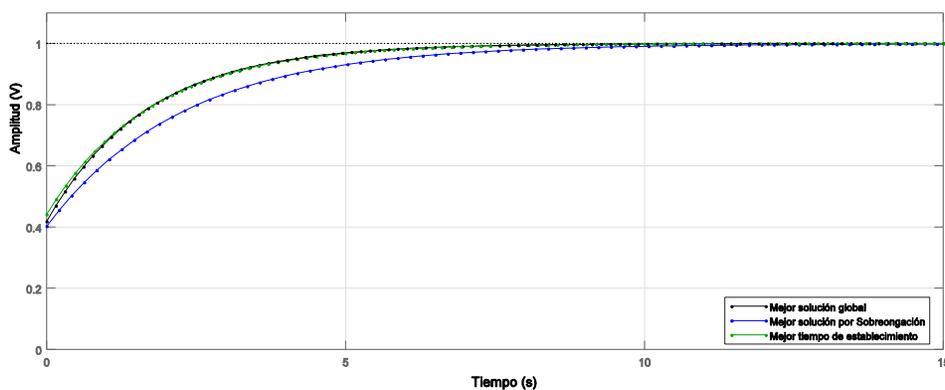


Figura 34. Resultados obtenidos de la aplicación de Colonia de Hormigas en la sintonización de ganancias

3.6. Diseño del controlador optimizado por el algoritmo Cuckoo Search

3.6.1. Definición de conceptos

Nido con huevo: Corresponde a una posible solución al problema planteado

Huevo Cuckoo: Representa una nueva y mejor solución

Población: Engloba el conjunto de nidos disponibles en el espacio de búsqueda

Factor de descubrimiento: Controla el porcentaje de huevos cuckoo que tienden a ser descubiertos

3.6.2. Consideraciones iniciales

Para dar inicio al algoritmo de optimización, es necesario definir un conjunto de consideraciones iniciales bajo las cuales se va a regir el algoritmo a lo largo de su ejecución, entre las cuales podemos destacar:

Rango de las ganancias (K_p , K_i , K_d): Se encarga de definir un rango de valores dentro del cual el algoritmo buscará la mejor solución.

Máximo coeficiente de sobreelongación: define el máximo valor de sobreelongación permitido dentro de la solución, sobre el cual se evaluará a las soluciones encontradas

Máximo tiempo de establecimiento: define el máximo tiempo de establecimiento permitido dentro de la solución, sobre el cual se evaluará a las soluciones encontradas

Error de estado estable: define el máximo valor de error que la señal se debe mantener para considerarse estable.

Numero de nidos: define el máximo número de nidos dentro de los cuales se buscará las mejores soluciones

3.6.3. Esquema general y desarrollo del algoritmo

Para realizar la simulación del proceso de optimización por el método *Cuckoo Search* se ha determinado un esquema simplificado que engloba todos los pasos presentados en la figura 12, con el fin de facilitar la programación del mismo en los siguientes bloques:

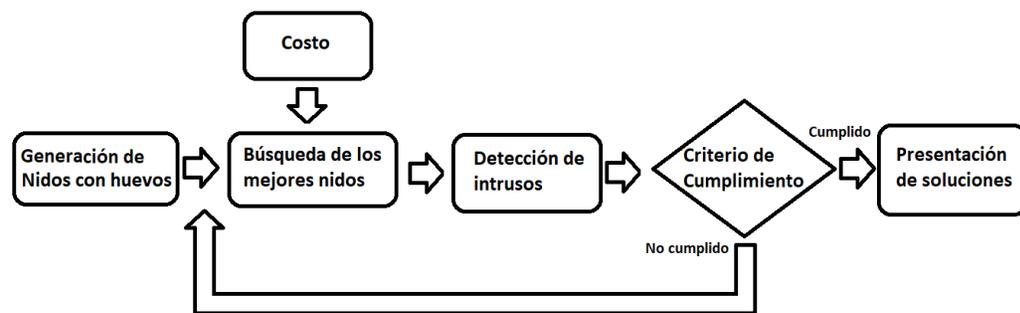


Figura 35. Esquema simplificado de simulación para el método de *Cuckoo Search*

a. Generación de Nidos con huevos

La primera fase del algoritmo corresponde a la generación de soluciones iniciales, es decir, al primer conjunto de nidos contendores de huevos *cuckoo*. Para ello se ha planteado una matriz de soluciones bajo la siguiente configuración:

La primera parte de la matriz engloba la información correspondiente a los huevos *cuckoo* depositados, es decir las soluciones generadas.

$$\text{soluciones} = \begin{bmatrix} Kp_1 & Ki_1 & Kd_1 \\ Kp_2 & Ki_2 & Kd_2 \\ Kp_3 & Ki_3 & Kd_3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ Kp_n & Ki_n & Kd_n \end{bmatrix} = \begin{bmatrix} \text{Solución}_1 \\ \text{Solución}_2 \\ \text{Solución}_3 \\ \cdot \\ \cdot \\ \text{Solución}_n \end{bmatrix} \quad (30)$$

Adicionalmente se ha incluido el desempeño alcanzado por cada una de estas soluciones. El conjunto de información almacenada en cada fila representa un huevo dentro de cada nido.

$$\text{Nidos} = \begin{bmatrix} \text{Solución}_1 & \text{desempeño}_1 \\ \text{Solución}_2 & \text{desempeño}_2 \\ \text{Solución}_3 & \text{desempeño}_3 \\ \cdot & \cdot \\ \cdot & \cdot \\ \text{Solución}_n & \text{desempeño}_n \end{bmatrix} = \begin{bmatrix} \text{Huevo}_1 \\ \text{Huevo}_2 \\ \text{Huevo}_3 \\ \cdot \\ \cdot \\ \text{Huevo}_n \end{bmatrix} \quad (31)$$

Donde n representa la cantidad de nidos que se van a utilizar para la optimización.

b. Búsqueda de mejores Nidos

El siguiente paso dentro del algoritmo corresponde al momento en que los huevos *cuckoo* depositados en la primera generación de nidos han crecido y están listos para dar origen a una nueva generación de huevos. Para ello el algoritmo de optimización por

Cuckoo Search requiere de la aplicación de técnicas de exploración de los mejores nidos dentro del espacio de búsqueda. Este proceso ha sido realizado mediante la técnica de búsqueda *Vuelos de Lévy* descrito en la sección 2.3.3.

i. Vuelos de Lévy

En (Flores Vilches, 2015) se plantea un procedimiento general para la implementación de la técnica de búsqueda *Vuelos de Lévy* dentro del algoritmo. En primer lugar, es necesario identificar aquellos nidos en que el crecimiento de los cucos tuvo menor dificultad o menor costo, y por tanto un mayor desempeño. El nido seleccionado por tener un menor costo dentro de la población actual se utiliza como referencia para el traslado de los cucos en búsqueda de mejores condiciones, mediante las siguientes formulas:

$$Solución_d = Solución_d + ss_d \times Rand(norm) \quad (32)$$

Donde d corresponde al número elementos dentro de *Solución*, es decir, las ganancias Kp , Kd y Ki tratadas de manera individual, y $Rand(norm)$ corresponde a un valor aleatorio generado con una distribución normal.

Además ss_d se puede calcular a partir de:

$$ss_d = st_d \times (Solucion_d - S_d) \quad (33)$$

Siendo S_d los elementos del nido con mejor desempeño identificado en primer lugar, y st_d se puede obtener de la siguiente ecuación:

$$st_d = (u_d \times [v_d])^{\frac{2}{3}} \quad (34)$$

Siendo u_d y v_d calculados como:

$$u_d = Rand(norm) \times sigma \quad (35)$$

$$v_d = Rand(norm) \quad (36)$$

Donde $sigma$ controla el tamaño de los pasos recorridos durante el proceso de búsqueda.

En resumen, este procedimiento genera un conjunto de valores aleatorios (con una distribución normal) interactuando entre sí, que se relacionan con la distancia medida entre cada elemento de la solución y los elementos del mejor nido identificado, y se agregan a los

valores actuales de cada nido. Como resultado se obtiene un conjunto de nuevos nidos disponibles cuya localización dentro del espacio de búsqueda ha sido encontrada en base a valores aleatorios e influenciada en cierta medida por el mejor nido de la generación previa.

c. Costo

El cálculo del costo que tiene cada cuco para crecer, o en su defecto el desempeño de los nidos utilizados, se ha realizado mediante la herramienta *Simulink* para la evaluación de la respuesta del controlador seleccionado y se ha definido los criterios de optimización en base al costo individual de cada parámetro de interés, tales como:

i. Costo en base al porcentaje de sobre elongación

Se realizó el cálculo del porcentaje de sobre elongación en base a la señal obtenida de la simulación, a través de la fórmula (9). Con el porcentaje de sobre elongación obtenido, se procede a calcular su relación en comparación al máximo porcentaje de sobre elongación deseado, haciendo uso de la formula (10).

ii. Costo en base al tiempo de establecimiento

Para calcular el costo de cada hormiga en base al máximo tiempo de establecimiento deseado, es necesario definir inicialmente un criterio de error de estado estable; en este caso se ha definido un error de estado estable de 2%. A partir del momento en que la señal se mantenga

con un error menor al establecido, esta podrá considerarse estable y se procede al cálculo del tiempo que demora en hacerlo.

El cálculo del tiempo de establecimiento medido se realiza en base a un punto identificado, a partir del cual la señal se considera estable por medio de la fórmula (11).

Tomando en cuenta el máximo tiempo de establecimiento deseado, podemos calcular el costo del controlador obtenido en base a la siguiente fórmula (12)

iii. Costo global

Finalmente, se puede agrupar ambos costos bajo un criterio de importancia de cada uno, en base a la fórmula (25). Finalmente, la optimización del algoritmo se basará en conseguir el costo global más bajo de cada solución, y por lo tanto el mejor desempeño de los controladores.

d. Detección de intrusos

El proceso de detección de intrusos se basa en un factor de descubrimiento determinado al inicio del algoritmo. Este factor determina una probabilidad de que el huevo *cuckoo* sea descubierto y eliminado del nido en la generación actual.

En (Flores Vilches, 2015) se plantea una técnica de descubrimiento de intrusos basado en una probabilidad Pa , con la cual se realiza un proceso de alteración de los nidos

seleccionados y se verifica su desempeño para saber si el *cuckoo* es descubierto o no. El proceso mencionado se ejecuta dentro del algoritmo en base a las siguientes fórmulas:

$$\text{nueva población} = \text{Matriz de nidos} + (ss \times mpa)$$

Donde ss se calcula a partir de:

$$ss = \text{rand}(0 \text{ a } 1) \times (N1 - N2) \quad (37)$$

$$N1 = \text{permutación}(\text{Matriz de nidos}) \quad (38)$$

$$N2 = \text{permutación}(\text{Matriz de nidos}) \quad (39)$$

Siendo $\text{rand}(0 \text{ a } 1)$ un valor aleatorio generado entre 0 y 1, $N1$ y $N2$ una matriz resultante de la permutación de la Matriz de nidos actuales, y mpa es una matriz con elementos binarios determinada por:

$$mpa = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ \cdot & \cdot & \cdot \\ b_{n1} & b_{n2} & b_{n3} \end{bmatrix}_{n \times 3} \quad \text{donde } b_{ij} = \begin{cases} 1 & \text{con una probabilidad } P_a \\ 0 & \text{con una probabilidad } 1 - P_a \end{cases} \quad (40)$$

De manera general, el procedimiento descrito genera las matrices N1 y N2 como una permutación de la matriz de nidos actual y realiza una diferencia de sus parámetros para obtener un valor relacionado con las soluciones de la población. La magnitud de este resultado es disminuida de manera aleatoria y adicionada a la matriz de nidos actuales únicamente en los casos en que la probabilidad P_a lo permite, dando como resultado una nueva población cuyos nidos serán comparados con la matriz de nidos originales.

Si los nuevos nidos presentan mejores resultados, el huevo *cuckoo* originalmente depositado en el nido es eliminado y el nido se mantiene con la nueva solución. Este procedimiento permite que la búsqueda de soluciones no se estanque en una zona específica y da la posibilidad de que la exploración abarque otros puntos aleatorios dentro del espacio de búsqueda.

e. Criterio de cumplimiento

Para este diseño se ha determinado al objetivo de control como como los valores máximos de sobre elongación y tiempo de establecimiento, por lo que no existe una única solución que cumpla los requisitos marcados. En base a esto se ha seleccionado un criterio de cumplimiento basado en 50 iteraciones con el objetivo de localizar el mejor resultado de entre todas las soluciones planteadas a lo largo de la ejecución del algoritmo de búsqueda.

f. Presentación de soluciones

Los siguientes resultados presentan un estilo similar a los presentados en los diseños anteriores, con el fin de facilitar su comparación en futuros análisis de resultados. De igual forma, en base a las consideraciones tomadas a lo largo de la ejecución del algoritmo, los resultados pertinentes para la selección del mejor controlador que se presentan son:

i. Matriz de soluciones

Engloba el conjunto de las mejores soluciones encontradas al finalizar el algoritmo, almacenando la solución con el mejor desempeño global de cada generación.

ii. Desempeño de los controladores

Presenta una gráfica general del desempeño obtenido por las mejores soluciones de cada iteración. Esta grafica se presenta dividida en 3 secciones que permiten analizar los máximos locales y el máximo global obtenido al finalizar el algoritmo para 3 casos de interés: Desempeño global, desempeño por sobrelongación y desempeño por tiempo de establecimiento.

iii. Costo de los controladores

Presenta una gráfica similar a la mostrada con el desempeño delos controladores, pero mostrando el costo, definido como el inverso del desempeño.

iv. Desempeño global de las soluciones

Presenta una gráfica en 3 dimensiones que localiza la posición en el espacio del desempeño para cada una de las soluciones encontradas.

v. Comparación de controladores

Presenta un gráfico comparativo de las mejores soluciones en cuanto al desempeño global, desempeño por sobre elongación y desempeño por tiempo de establecimiento.

vi. Vuelos de Levy

Presenta un gráfico tridimensional del recorrido de los cucos aplicando los vuelos de Lévy en búsqueda de los mejores resultados, cuyos ejes corresponden a las ganancias K_p K_i K_d localizadas. Para el gráfico se ha tomado el recorrido de 7 cucos representados cada uno por un color y se presenta su recorrido a lo largo del espacio de búsqueda

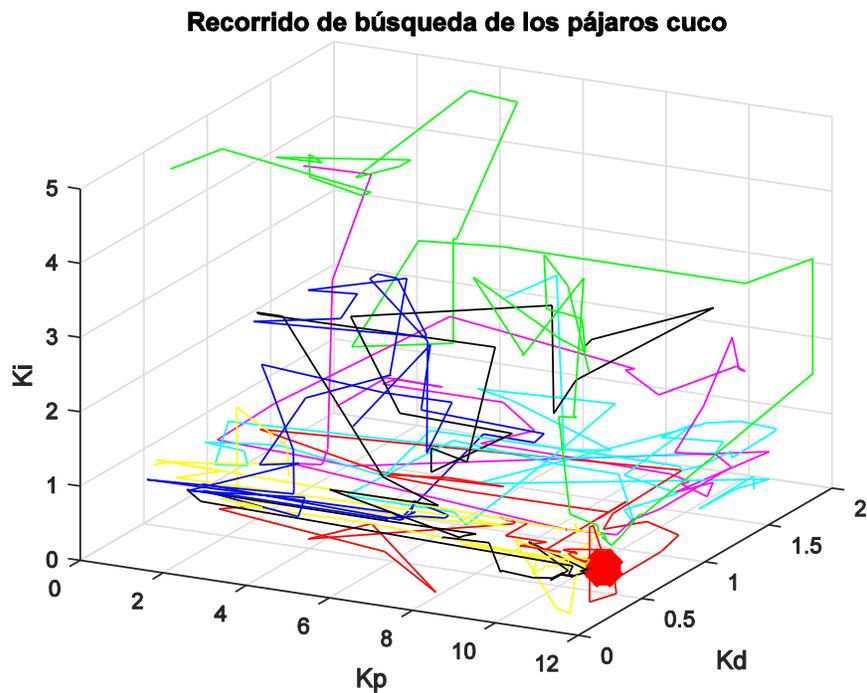


Figura 36. Vuelos de Lévy aplicados en la búsqueda del mejor controlador

3.6.4. Resultados obtenidos de la aplicación del algoritmo

El detalle de la aplicación del algoritmo planteado sobre el software dedicado para el efecto se puede apreciar en el **Anexo C**

Los resultados obtenidos luego de la ejecución del algoritmo se pueden observar en la siguiente tabla:

Tabla 9

Resultados sintonizados con la aplicación del método *Cuckoo Search* implementado

	Kp	Ki	Kd
Con mejor desempeño global	11.8192	0.6485	0.2712
Con mejor desempeño por sobrelongación	11.6760	0.6021	0.4102
Con mejor desempeño por tiempo de establecimiento	11.9131	0.7417	1.3884

La respuesta al escalón de la solución encontrada se puede observar en la siguiente figura:

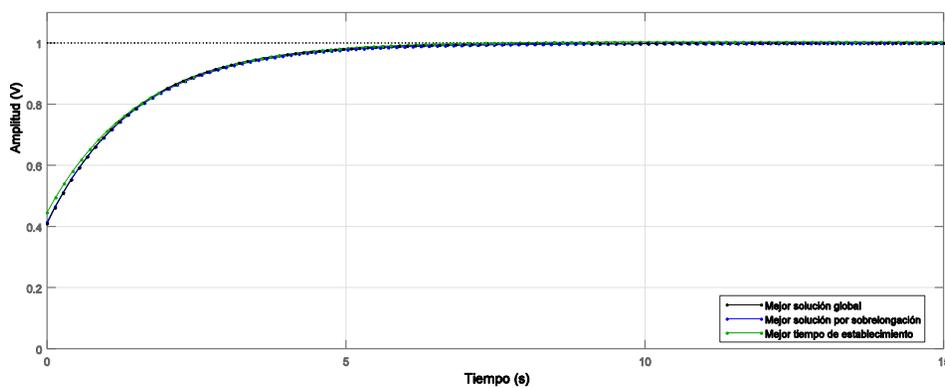


Figura 37. Resultados obtenidos de la aplicación de *Cuckoo Search* en la sintonización de ganancias

CAPITULO 4

4. ANÁLISIS DE RESULTADOS

4.1. Especificaciones del sistema:

Como ya se ha mencionado en la sección 3.2, el objetivo de control que se requiere cumplir corresponde al diseño de controladores que alcancen un tiempo de establecimiento menor a 20s con una sobreelongación menor a 15% tomando en cuenta un criterio de estabilización de 2% de error de estado estable.

En la sección 3.3 se ha realizado la sintonización del controlador por medio de la herramienta *PID Tunner* cuyos resultados se muestran en la tabla 5. Tomado estos como base, se ha determinado los siguientes rangos de búsqueda para la sintonización de controladores por los tres métodos de algoritmos bio-inspirados diseñados, como se muestra en la siguiente tabla:

Tabla 10

Parámetros iniciales para el desarrollo de los algoritmos

Ganancia Kp	0 – 12
Ganancia Ki	0 – 5
Ganancia Kd	0 – 2
Máximo porcentaje de sobreelongación	15%

CONTINÚA

Máximo tiempo de establecimiento	20 s
Error de estado estable	2%
Tamaño de la población	20
Número de iteraciones	50
Voltaje de referencia	1V

4.2. Resultados obtenidos

4.2.1. Sintonización de controladores

a. Algoritmos Genéticos

Con los parámetros de diseño definidos en la tabla 7, se procedió a ejecutar el programa realizado con el fin de encontrar el mejor conjunto de ganancias para el problema planteado por el método de Algoritmos Genéticos. En este caso particular, el tamaño de la población representa en número de individuos con los cuales se va a trabajar en cada generación. Adicionalmente se ha determinado una longitud de cromosoma de 8 bits para cada ganancia calculada, con el fin de tener una mejor resolución entre los rangos de ganancias definidos.

La figura 28 presenta el conjunto de las mejores soluciones de cada iteración y el desempeño alcanzado por las mismas bajo 3 distintos parámetros detallados en la sección 3.4.

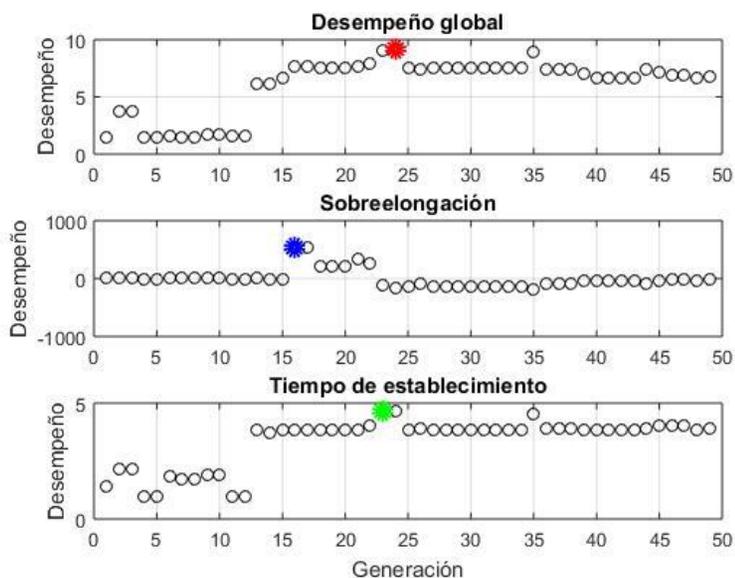


Figura 38. Desempeño medido por el recorrido de las hormigas en el proceso de búsqueda

La figura 39 presenta un modelo tridimensional del desempeño alcanzado por cada una de las soluciones medidas, haciendo énfasis en las mejores soluciones respecto al porcentaje de sobreelongación medido (en azul), el menor tiempo de establecimiento alcanzado (en verde) y el mejor desempeño global logrado (en rojo).

La figura nos permite observar que, a pesar de que la solución con el mejor desempeño en cuanto a sobreelongación se encuentra en la generación 16 y la solución con el mejor desempeño en base al tiempo de establecimiento se encuentra en la generación 23, la solución

encontrada por el algoritmo con el mejor desempeño global se ha localizado en la generación 24 del proceso de búsqueda.

Por otro lado, la siguiente figura presenta los resultados obtenidos en función del costo alcanzado por cada solución. Cada una de las medidas de costo presentada representa al mínimo local obtenido en su respectiva iteración, sin embargo, es importante destacar al mínimo global del proceso de búsqueda que representa la mejor solución encontrada por el algoritmo.

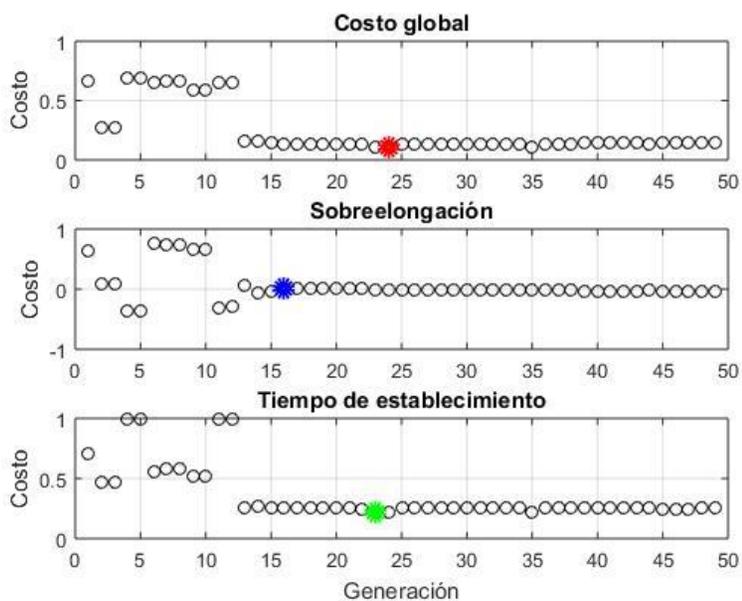


Figura 39. Costo medido por el recorrido de las hormigas en el proceso de búsqueda

Los resultados presentados mantienen el mismo esquema de colores mencionado

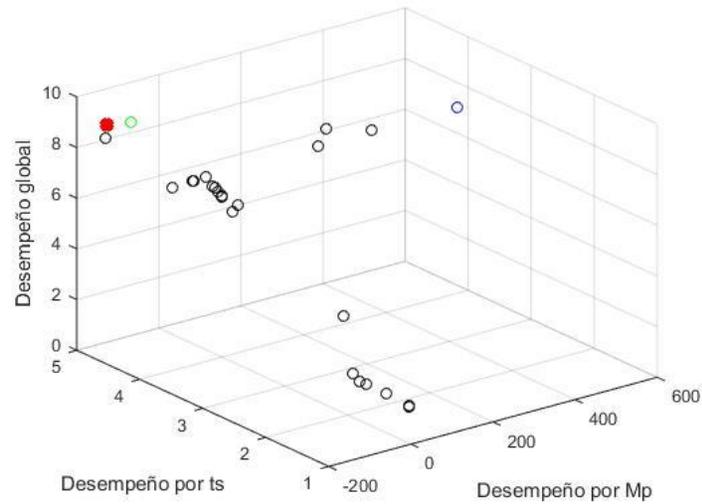


Figura 40. Modelo tridimensional de desempeños alcanzados para el método de Algoritmos Genéticos

Para comparar la respuesta obtenida entre los tres mejores controladores identificados, se los ha sometido a una entrada escalón unitario correspondiente a la medida de 30°C en la planta real, obteniendo los siguientes resultados.

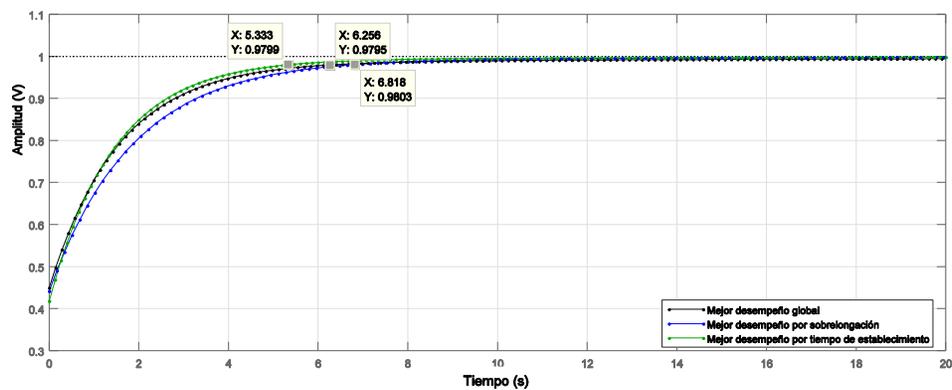


Figura 41. Respuesta de los mejores controladores ante entrada escalón unitario

Se puede observar que ninguna de las respuestas obtenidas presenta un porcentaje de sobrelongación antes de estabilizarse, por lo cual se tomará en cuenta únicamente el tiempo de establecimiento medido para el posterior análisis.

Tabla 11

Tiempos de establecimiento medidos por los controladores optimizados por Algoritmos Genéticos

Sistema	Medida
Sistema con mejor desempeño global	6.26 segundos
Sistema con mejor desempeño por sobrelongación	6.82 segundos
Sistema con mejor desempeño por tiempo de establecimiento	5.33 segundos

Los controladores obtenidos por el algoritmo de búsqueda basado en Algoritmos Genéticos se presentan a continuación

Tabla 12

Controladores calculados por algoritmo colonia de hormigas

	Controlador
Con mejor desempeño global	$1.5216 s + 11.7176 + \frac{0.4706}{s}$
Con mejor desempeño por sobrelongación	$1.2549 s + 9.7412 + \frac{0.4706}{s}$
Con mejor desempeño por tiempo de establecimiento	$0.5176 s + 11.9059 + \frac{0.5686}{s}$

Adicionalmente se ha medido el tiempo de ejecución del algoritmo en 3 ocasiones con un promedio de 157.885508 para ejecutarse de acuerdo a los parámetros iniciales establecidos.

b. Colonia de Hormigas

Al igual que en caso anterior, una vez definidos los parámetros de diseño, se procedió a correr el programa realizado con el fin de encontrar el mejor conjunto de ganancias para el problema planteado. En este caso particular, el tamaño de la población representa en número de hormigas que se utilizarán para buscar el camino con mayor desempeño dentro de los rangos planteados y se ha utilizado un conjunto de 100 nodos dentro de los rangos establecidos para cada ganancia.

La siguiente figura muestra los caminos de búsqueda que recorren las hormigas en determinadas iteraciones a lo largo de la ejecución del algoritmo. Se puede apreciar que a medida que avanzan las iteraciones, el recorrido de las hormigas converge hacia un único camino contenedor de la mayor cantidad de feromonas depositadas.

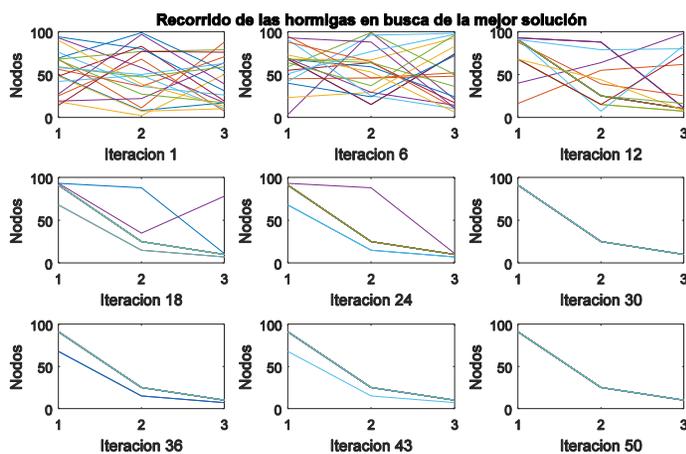


Figura 42. Recorrido de hormigas realizado a lo largo del algoritmo.

La figura 43 presenta el conjunto de las mejores soluciones de cada iteración y el desempeño alcanzado por las mismas bajo 3 distintos parámetros detallados en la sección 3.5.

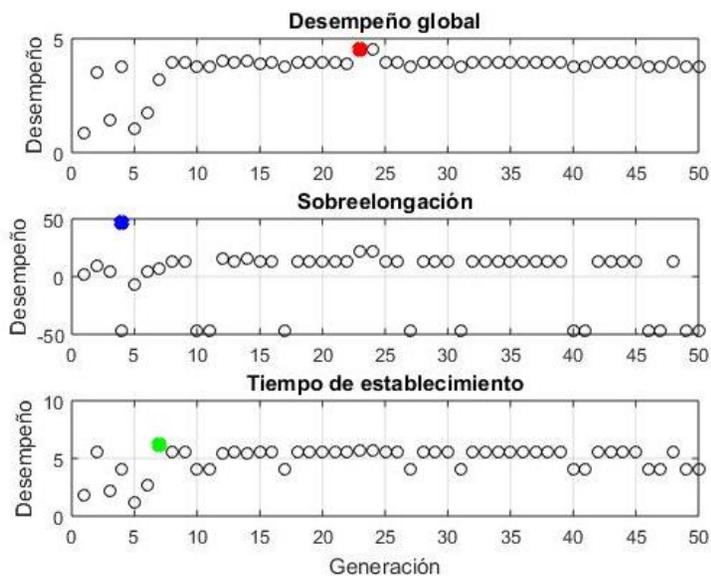


Figura 43. Desempeño medido por el recorrido de las hormigas en el proceso de búsqueda

Podemos observar que, a pesar de que la solución con el mejor desempeño en cuanto a sobreelongación se encuentra en la 4ta iteración y la solución con el mejor desempeño en base al tiempo de establecimiento se encuentra en la 7ma generación, la solución encontrada por las hormigas con el mejor desempeño global se ha localizado en la iteración 23 del proceso de búsqueda.

De igual forma, la siguiente gráfica presenta los resultados obtenidos en función del costo alcanzado por cada solución. Cada una de las medidas de costo presentada representa al mínimo local obtenido en su respectiva iteración, sin embargo, es importante destacar al

mínimo global del proceso de búsqueda que representa la mejor solución encontrada por el algoritmo.

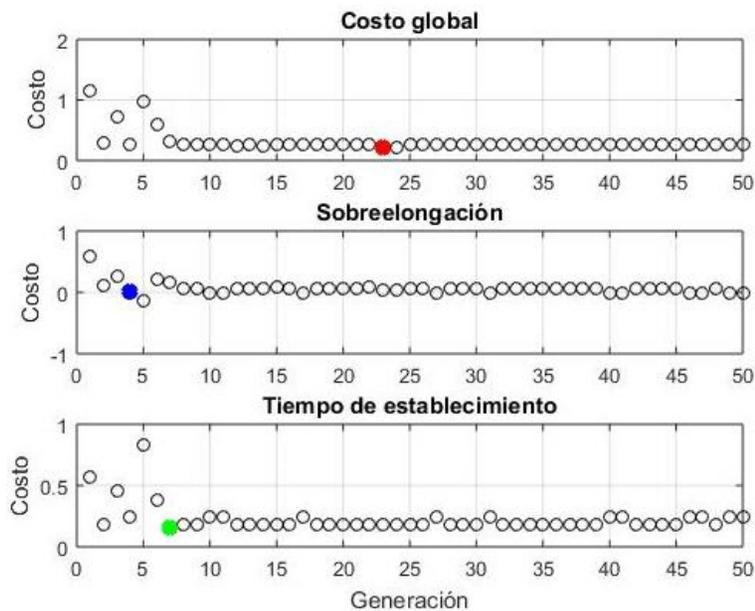


Figura 44. Costo medido por el recorrido de las hormigas en el proceso de búsqueda

La figura 45 nos presenta un modelo tridimensional del desempeño alcanzado por cada una de las soluciones medidas, haciendo énfasis en las mejores soluciones respecto al porcentaje de sobreelongación medido (en azul), el menor tiempo de establecimiento alcanzado (en verde) y el mejor desempeño global logrado (en rojo).

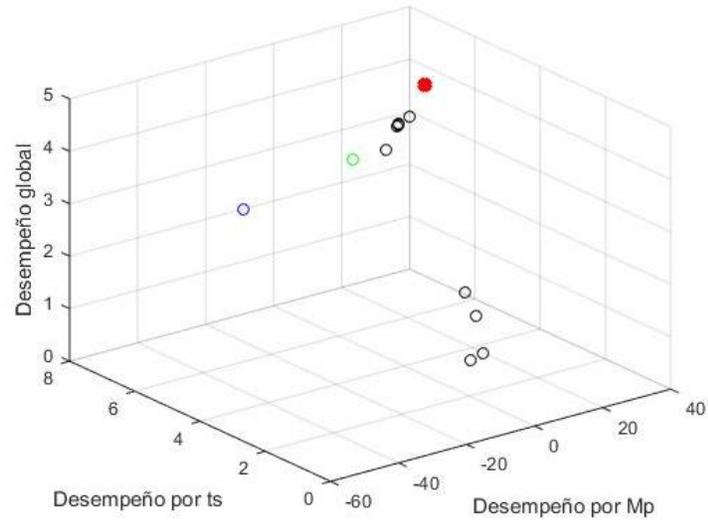


Figura 45. Modelo tridimensional de desempeños alcanzados para el algoritmo Colonia de Hormigas

Para comparar la respuesta obtenida entre los tres mejores controladores identificados, se los ha sometido a una entrada escalón unitario correspondiente a la medida de 30°C en la planta real, obteniendo los siguientes resultados.

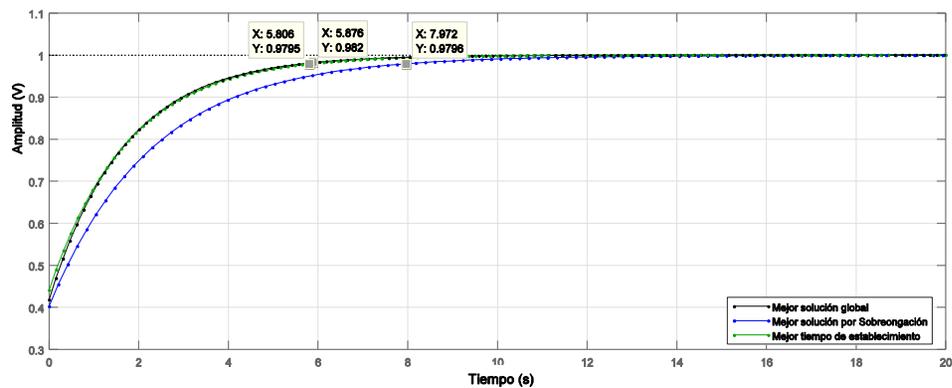


Figura 46. Respuesta de los mejores controladores ante entrada escalón unitario

Se puede observar que ninguna de las respuestas obtenidas presenta un porcentaje de sobrelongación antes de estabilizarse, por lo cual se tomará en cuenta únicamente el tiempo de establecimiento medido.

Tabla 13

Tiempos de establecimiento medidos por los controladores optimizados por Colonia de hormigas

Sistema	Medida
Sistema con mejor desempeño global	5.88 segundos
Sistema con mejor desempeño por sobrelongación	7.97 segundos
Sistema con mejor desempeño por tiempo de establecimiento	5.81 segundos

Los controladores obtenidos por el algoritmo de búsqueda basado en colonia de hormigas se presentan a continuación

Tabla 14

Controladores calculados por algoritmo colonia de hormigas

	Controlador
Con mejor desempeño global	$0.5265 s + 10.3652 + \frac{0.5683}{s}$
Con mejor desempeño por sobrelongación	$0.0697 s + 7.3847 + \frac{0.3929}{s}$
Con mejor desempeño por tiempo de establecimiento	$1.2370 s + 10.4244 + \frac{0.5690}{s}$

Adicionalmente se ha medido el tiempo de ejecución del algoritmo en 3 ocasiones con un promedio de 59.928055 para ejecutarse de acuerdo a los parámetros iniciales establecidos.

c. Cuckoo Search

Tomando en cuenta los parámetros de diseño definidos en la tabla 7, se procedió a ejecutar el programa de optimización por el método de Cuckoo Search con el fin de encontrar el mejor conjunto de ganancias para el controlador requerido. En este caso particular, el tamaño de la población representa en número de nidos en los que se depositarán los huevos cuco o soluciones que se utilizarán para buscar el camino con mayor desempeño dentro de los rangos planteados.

La siguiente figura muestra los caminos recorridos por los cucos en la búsqueda de mejores nidos para sobrevivir a lo largo de la ejecución del algoritmo. Se puede apreciar que el método de vuelos de Lévy utilizado para la exploración en el espacio de trabajo permite realizar la búsqueda de los mejores controladores de forma aleatoria, pero tendiendo hacia un único sector de soluciones a lo largo de cada generación.

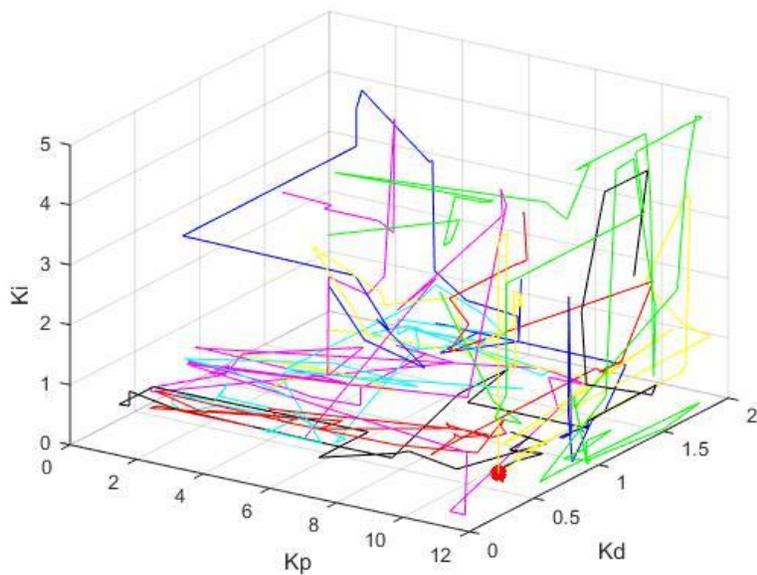


Figura 47. Recorrido de los pájaros Cuco realizado a lo largo del algoritmo.

La figura 48 presenta el conjunto de las mejores soluciones de cada iteración y el desempeño alcanzado por las mismas bajo 3 distintos parámetros detallados en la sección 3.6

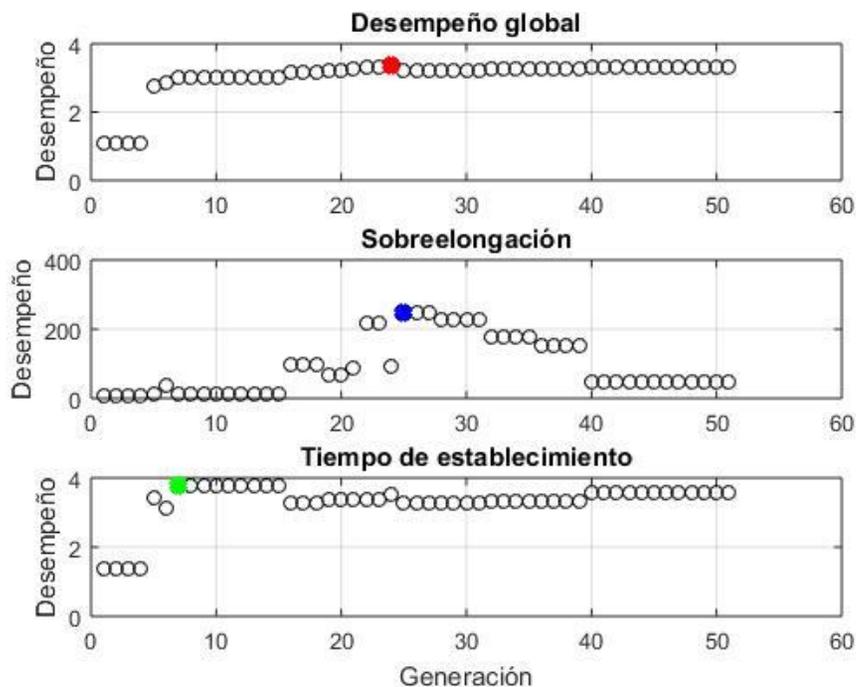


Figura 48. Desempeño medido por la búsqueda de los Cucu

De la gráfica podemos observar que a pesar de que la solución con el mejor desempeño en cuanto a sobreelongación se encuentra en la generación 25 y la solución con el mejor desempeño en base al tiempo de establecimiento se encuentra en la generación 7, la solución encontrada por los cucos con el mejor desempeño global se ha localizado en la iteración 24 del proceso de búsqueda.

En la siguiente figura podemos observar las medidas de costo obtenidas a lo largo de la ejecución del algoritmo. Cabe recalcar que cada medida presentada representa al mínimo local obtenido en su respectiva generación, sin embargo, es importante destacar al mínimo global del proceso de búsqueda que representa la mejor solución encontrada por el algoritmo.

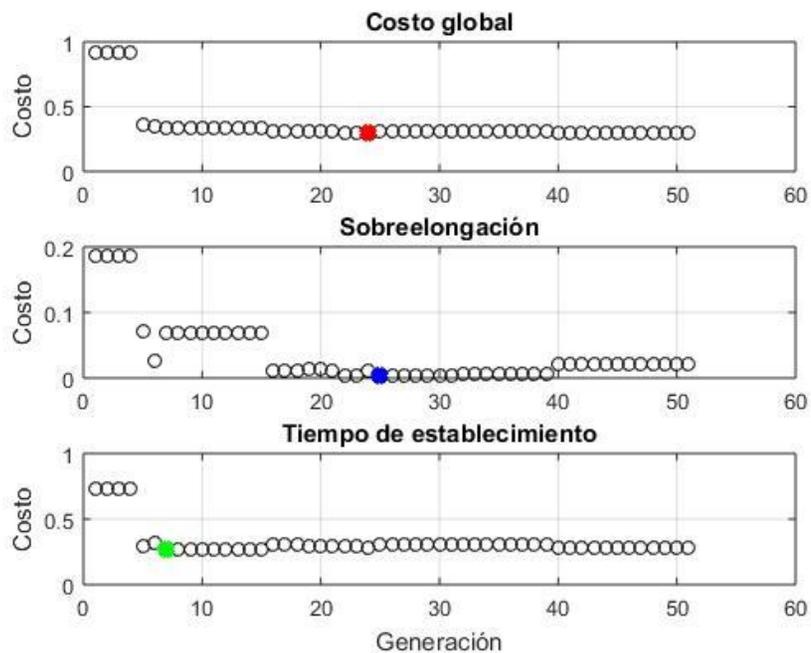


Figura 49. Costo medido por la búsqueda de los cuco

La figura 50 nos presenta un modelo tridimensional del desempeño alcanzado por cada una de las soluciones medidas, haciendo énfasis en las mejores soluciones respecto al porcentaje de sobreelongación medido (en azul), el menor tiempo de establecimiento alcanzado (en verde) y el mejor desempeño global logrado (en rojo).

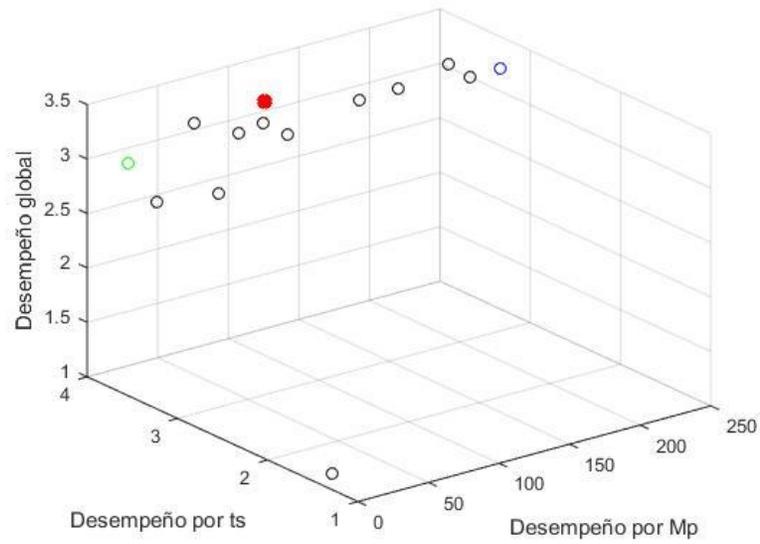


Figura 50. Modelo tridimensional de desempeños alcanzados para el algoritmo Cuckoo Search

Para comparar la respuesta obtenida entre los tres mejores controladores identificados, se los ha sometido a una entrada escalón unitario correspondiente a la medida de 30°C en la planta real, obteniendo los siguientes resultados.

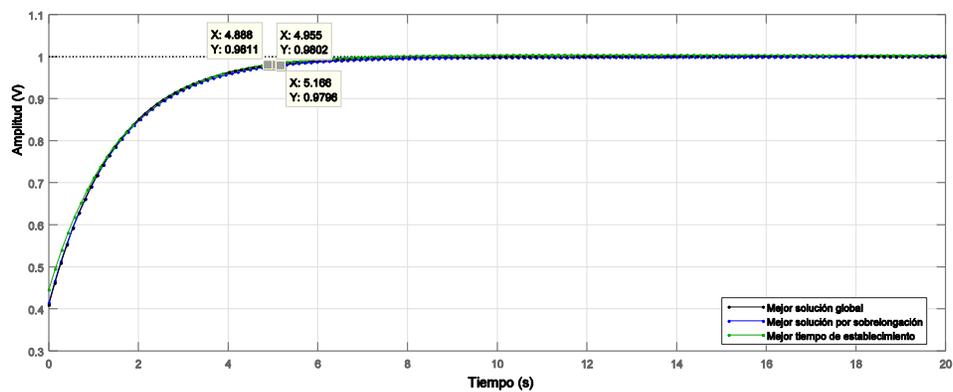


Figura 51. Respuesta de los mejores controladores ante entrada escalón unitario

Se puede observar que ninguna de las respuestas obtenidas presenta un porcentaje de sobrelongación antes de estabilizarse, por lo cual se tomará en cuenta únicamente el tiempo de establecimiento medido.

Tabla 15

Tiempos de establecimiento medidos por los controladores optimizados por Cuckoo Search

Sistema	Medida
Sistema con mejor desempeño global	4.96 segundos
Sistema con mejor desempeño por sobrelongación	5.17 segundos
Sistema con mejor desempeño por tiempo de establecimiento	4.89 segundos

Los controladores obtenidos por el algoritmo de búsqueda basado en colonia de hormigas se presentan a continuación

Tabla 16

Controladores calculados por algoritmo Cuckoo Search

Controlador	
Con mejor desempeño global	$0.2712 s + 11.8192 + \frac{0.6485}{s}$
Con mejor desempeño por sobrelongación	$0.4102 s + 11.6760 + \frac{0.6021}{s}$
Con mejor desempeño por tiempo de establecimiento	$1.3884 s + 11.9131 + \frac{0.7417}{s}$

Adicionalmente se ha medido el tiempo de ejecución del algoritmo en 3 ocasiones con un promedio de 104.921250 para ejecutarse de acuerdo a los parámetros iniciales establecidos.

4.2.2. Resultados Simulados

a. Resultados simulados en régimen transitorio

La figura 42 presenta una comparación gráfica de los resultados obtenidos de los mejores controladores sintonizados por cada algoritmo diseñado y por el controlador PID sintonizado mediante la herramienta *PID Tunner*. En color negro podemos observar la respuesta del controlador sintonizado por Algoritmos genéticos; de igual forma, en color azul la respuesta obtenida del controlador sintonizado por Colonia de hormigas; en color verde la respuesta obtenida del controlador sintonizado por Cuckoo Search y en color rojo la respuesta obtenida del controlador sintonizado con la herramienta *PID Tunner*.

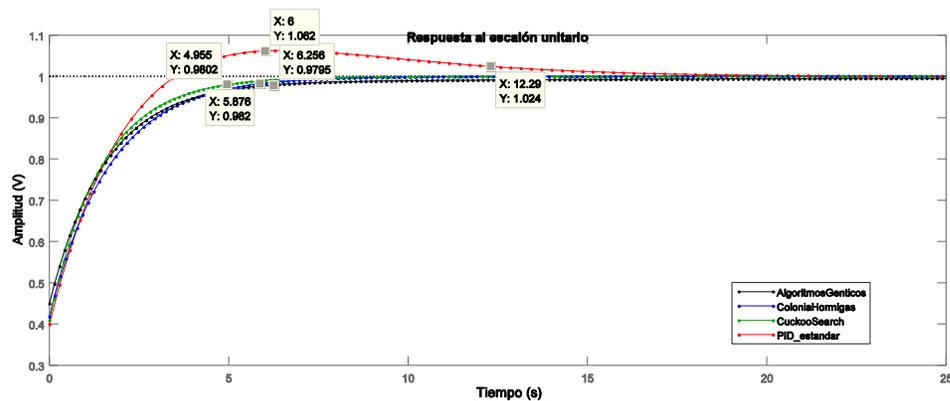


Figura 52. Resultados simulados obtenidos con las mejores soluciones de cada caso

El resumen de los resultados obtenidos en por los controladores señalados régimen transitorio se pueden observar en las siguientes tablas:

Tabla 17*Resultados obtenidos en sobrelongación por los controladores sintonizados*

Controlador	Sobrelongación
PID Estándar	6%
Algoritmos Genéticos	-
Colonia de Hormigas	-
Cuckoo Search	-

Tabla 18*Resultados obtenidos en tiempo de establecimiento por los controladores sintonizados*

Controlador	Tiempo de establecimiento
PID Estándar	12.3 s
Algoritmos Genéticos	6.26 s
Colonia de Hormigas	5.88 s
Cuckoo Search	4.97 s

b. Resultados simulados en régimen permanente

Con el fin de analizar el comportamiento de los controladores sintonizados en rechazo a las perturbaciones se ha realizado dos experimentos ingresando perturbaciones a las señales estabilizadas; en el primer caso se ha representado a la perturbación por una entrada escalón de amplitud -0.1, y en el siguiente caso la perturbación será representada por una entrada escalón de -0.2% -0.2 [V], obteniendo los siguientes resultados:

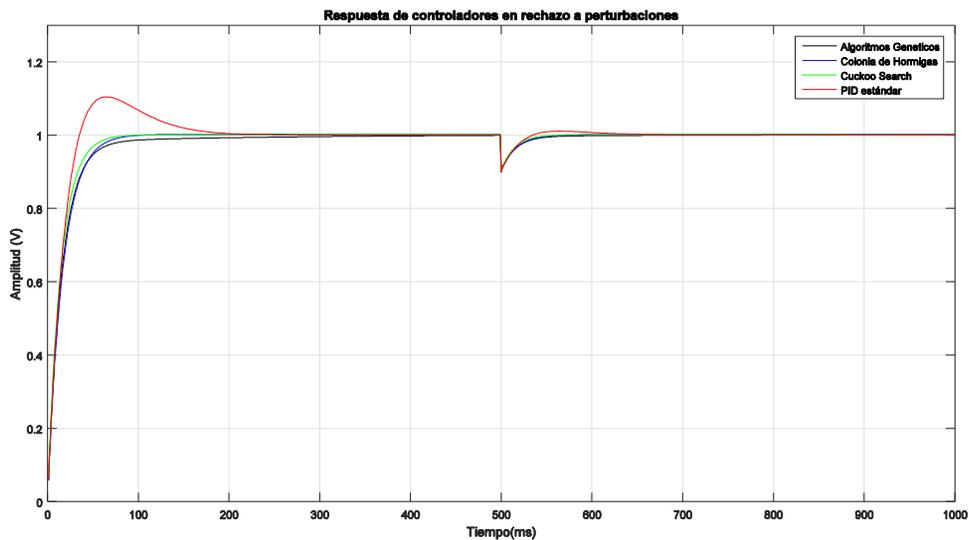


Figura 53. Caso 1: Respuesta de los controladores ante una entrada escalón de $-0.1[V]$

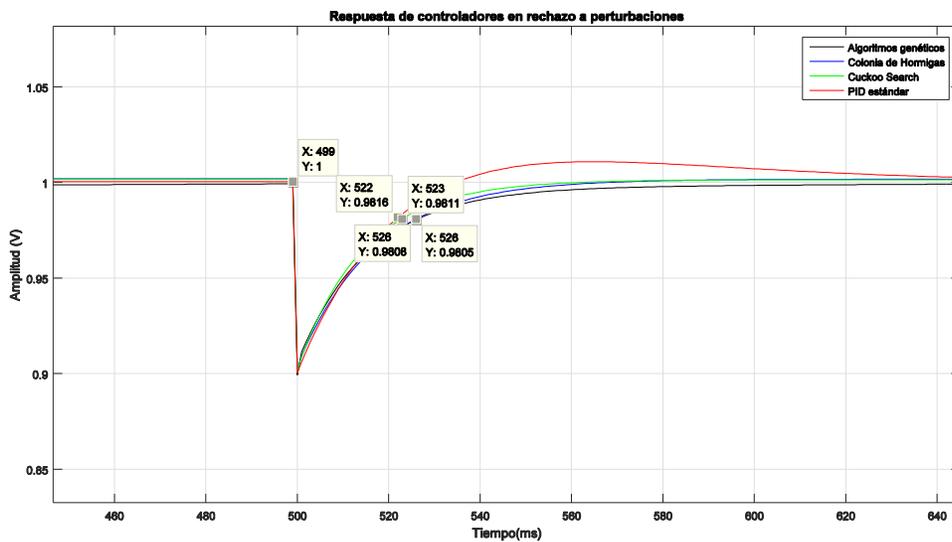


Figura 54. Medición de las respuestas de los controladores ante una entrada escalón de $-0.1[V]$

Cabe recalcar que la simulación ha sido ejecutada con una tasa de muestreo de 10 mediciones por segundo. El tiempo de recuperación medido luego de que se ha producido la perturbación se puede observar en la siguiente tabla:

Tabla 19

Resultados medidos en controladores ante perturbación de -0.1 [V]

Controlador	Muestras	Tiempo de establecimiento
PID Estándar	23	2.3 s
Algoritmos Genéticos	27	2.7 s
Colonia de Hormigas	27	2.7 s
Cuckoo Search	24	2.4 s

Se puede observar que, a pesar de que el controlador PID estándar tiene una repuesta ligeramente más rápida en rechazo a las perturbaciones, esta también genera una pequeña oscilación antes de estabilizarse nuevamente.

Para el siguiente caso se ha implementado una entrada escalón de -0.2 [V] después de 50 segundos de estabilización.

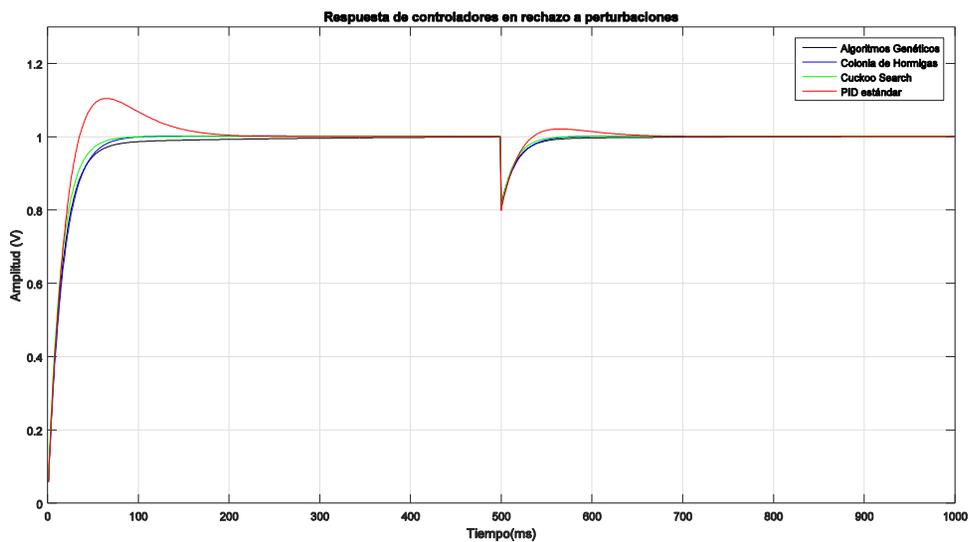


Figura 55. Caso 2: Respuesta de los controladores ante una entrada escalón de $-0.2[V]$

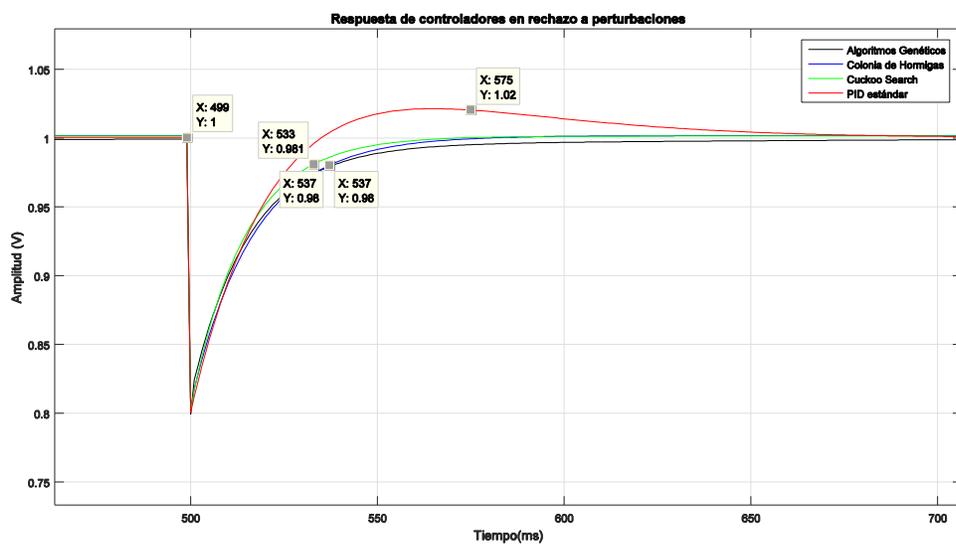


Figura 56. Medición de las respuestas de los controladores ante una entrada escalón de $-0.2[V]$

El tiempo de recuperación medido luego de que se ha producido la perturbación se puede observar en la siguiente tabla:

Tabla 20

Resultados medidos en controladores ante perturbación de -0.2 [V]

Controlador	Muestras	Tiempo de establecimiento
PID Estándar	76	7.6 s
Algoritmos Genéticos	38	3.8 s
Colonia de Hormigas	38	3.8 s
Cuckoo Search	34	3.4 s

En este caso podemos observar que la oscilación que presenta el controlador PID estándar ha provocado que el sistema no se recupere inmediatamente. Por otro lado, los controladores sintonizados con algoritmos bio-inspirados mantienen respuestas similares en los tres casos, destacando el controlador sintonizado por Cuckoo Search, que se recupera un 12% más rápido.

4.2.3. Mediciones reales

a. Mediciones reales en régimen transitorio

Las figuras muestran el comportamiento de los controladores aplicados a la planta de temperatura PCT-2. Cabe recalcar que las medidas presentadas se han tomado con un tiempo de muestreo de 50ms.

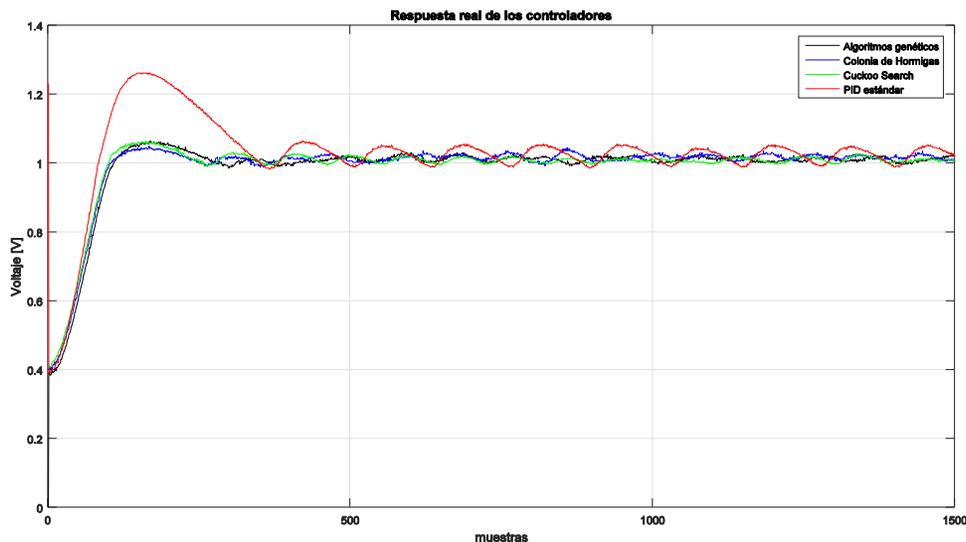


Figura 57. Respuesta medida de los controladores en régimen transitorio

En la figura 57 podemos observar la comparación de las mediciones realizadas a los distintos controladores aplicados. En negro se muestra la respuesta del controlador sintonizado por algoritmos genéticos, en azul la respuesta del controlador sintonizado por Colonia de hormigas, en verde la respuesta del controlador sintonizado por Cuckoo Search y en rojo la respuesta del PID estándar sintonizado con PID Tunner.

En esta figura se puede apreciar un rizo excesivo en las señales estabilizadas. Esto sucede en consecuencia de la inestabilidad de la planta y la respuesta errónea del sensor IC de estado sólido, que luego de un determinado tiempo de uso, presenta fluctuaciones en su señal de salida, misma que será medida por el controlador. Por otro lado, cabe mencionar que las medidas en los puntos extremos de las oscilaciones producidas en las señales correspondientes a los controladores optimizados por Algoritmos genéticos, Colonia de hormigas y Cuckoo

Search, no sobrepasan el 2% de la señal medida, por lo cual, se puede considerar a estas señales como estables bajo el criterio de 2% de error establecido en el objetivo de control.

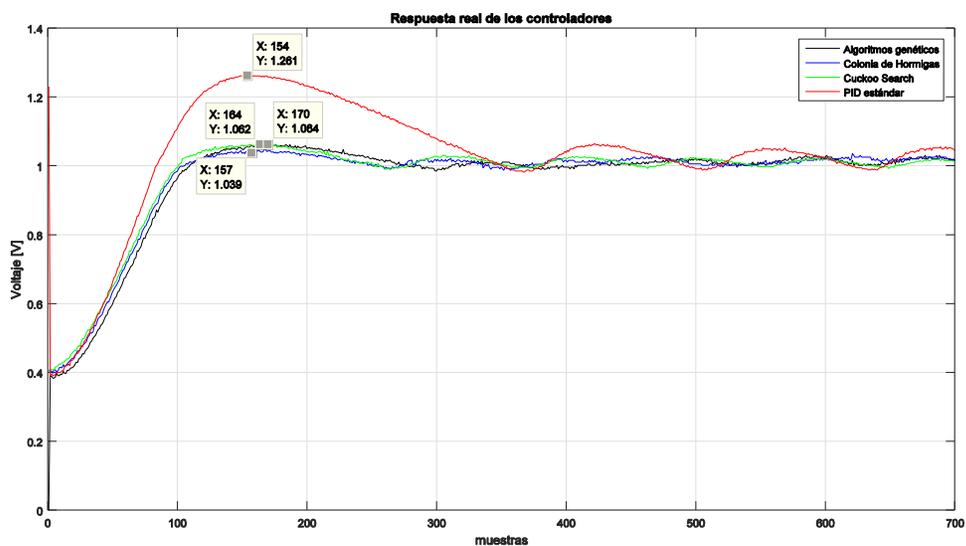


Figura 58. Medición de porcentaje de sobrelongación en los controladores aplicados sobre la planta PCT-2

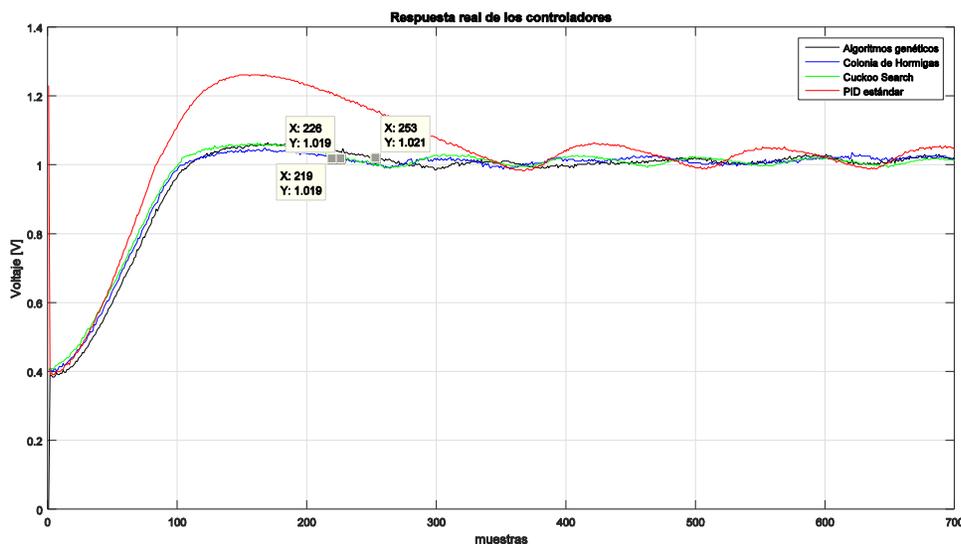


Figura 59 Medición del tiempo de establecimiento en los controladores aplicados sobre la planta PCT-2.

La siguiente tabla presenta un resumen de las medidas tomadas en las figuras 57 58 y 59. Cabe recalcar que las medidas fueron tomadas con un tiempo de muestreo de 50ms.

Tabla 21

Mediciones tomadas del porcentaje de sobrelongación en los controladores aplicados sobre la planta PCT-2

Controlador	Voltaje máximo [v]	Porcentaje de sobrelongación
PID	1.261	26.1%
Algoritmos Genéticos	1.064	6.4%
Colonia de hormigas	1.061	6.1%
Cuckoo Search	1.039	3.9%

Tabla 22

Mediciones tomadas del tiempo de establecimiento en los controladores aplicados sobre la planta PCT-2

Controlador	# Muestra	Tiempo de establecimiento [s]
PID	-	-
Algoritmos Genéticos	253	12.65
Colonia de hormigas	219	10.95
Cuckoo Search	226	11.3

b. Mediciones reales en régimen permanente

Para realizar la medición de respuesta en estado transitorio de los controladores aplicados en la planta PCT-2, se ha aplicado una perturbación sobre el ventilador encargado de proporcionar el flujo de aire frío detallado en la figura 13 de la sección 3.1 con una duración de 10 segundos. Los resultados medidos en comparación a su respectiva señal de control medida se pueden observar a continuación:

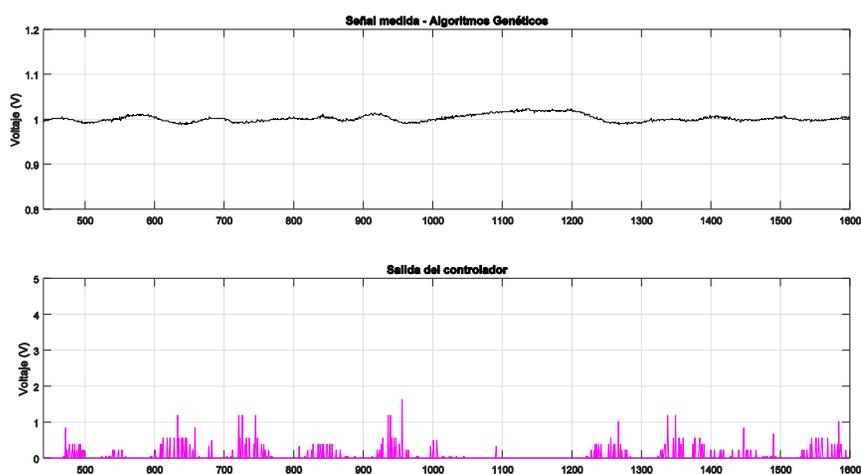


Figura 60. Rechazo ante perturbaciones medido para controlador sintonizado por Algoritmos Genéticos

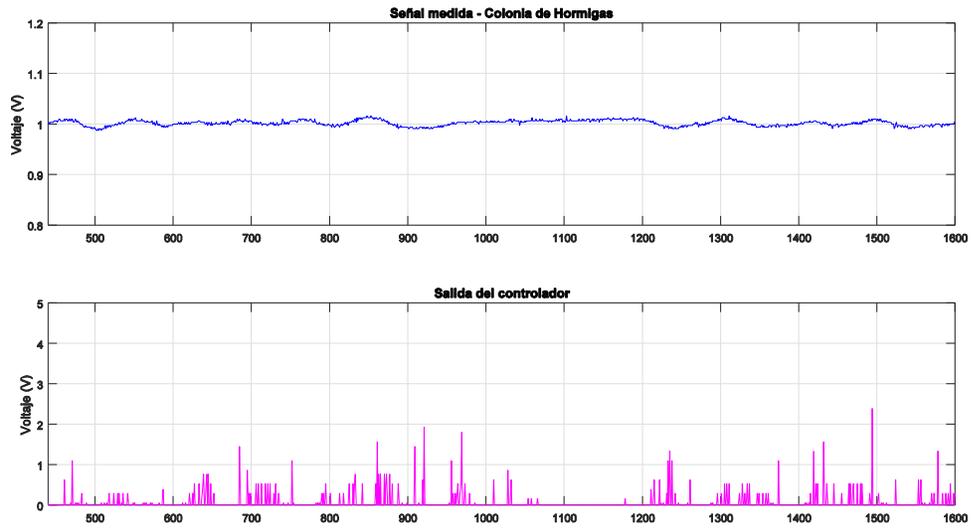


Figura 61. Rechazo ante perturbaciones medido para controlador sintonizado por Colonia de Hormigas

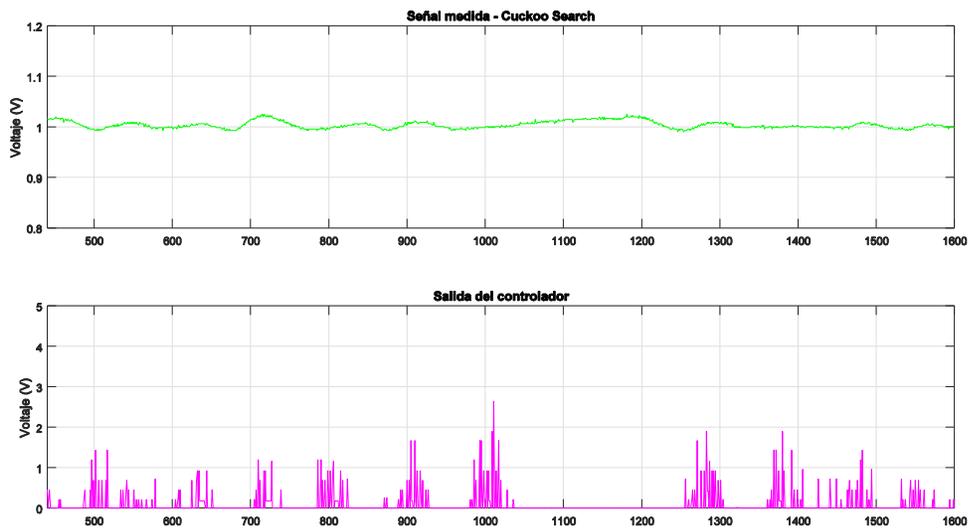


Figura 62. Rechazo ante perturbaciones medido para controlador sintonizado por *Cuckoo Search*

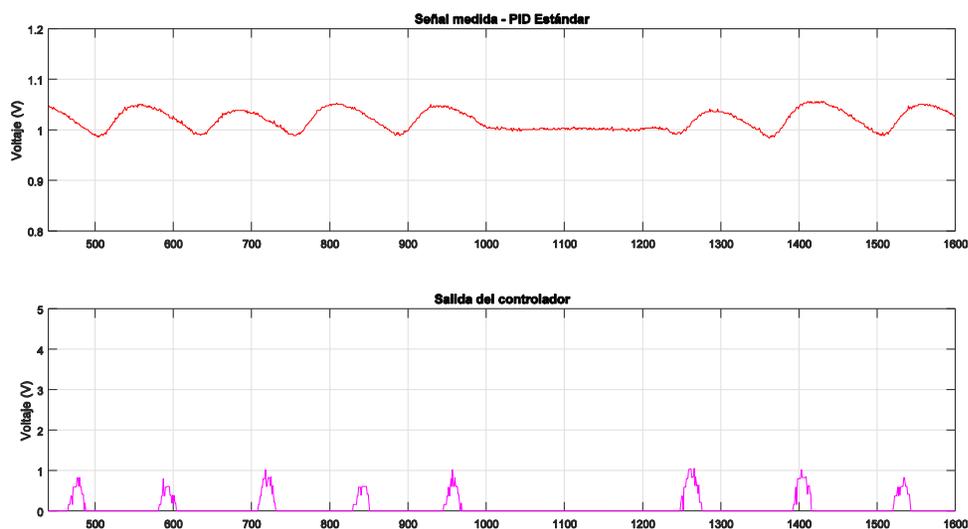


Figura 63. Rechazo ante perturbaciones medido para controlador sintonizado por PID tuner

De las figuras 60, 61 y 62 podemos observar una respuesta rápida del controlador ante las fluctuaciones producidas por la salida del sensor IC, lo cual permite que la señal se mantenga dentro del rango de error aceptable de 2%. En contraste, el algoritmo PID estándar presenta una respuesta lenta a las fluctuaciones de la señal de salida, por lo cual el controlador no logra devolver la señal valor de estabilización antes de que esta supere el rango de 2% de la referencia, y en consecuencia, provoca que la señal no se pueda considerar estable bajo las especificaciones dadas en el objetivo de control.

4.3. Análisis de resultados

4.3.1. Análisis de resultados simulados

La sintonización controladores realizado por los algoritmos bio-inspirados descritos se ha realizado bajo los mismos requisitos de funcionamiento, con el fin de realizar un análisis comparativo lo más justo posible. La tabla 20 nos presenta las ganancias encontradas por cada uno de los métodos de sintonización aplicados

Tabla 23

Ganancias encontradas por los algoritmos aplicados

Método de sintonización	Ganancia Kp	Ganancia Ki	Ganancia Kd
Algoritmos genéticos	11.7176	0.4706	1.5216
Colonia de hormigas	10.3652	0.5683	0.5265
Cuckoo Search	11.8192	0.6485	0.2712

La siguiente tabla presenta el detalle de los resultados medidos en los tres casos de sintonización:

Tabla 24

Resumen de características tomadas en la sintonización de controladores

Método de sintonización	Tiempo de establecimiento medido [s]	Porcentaje de sobrelongación	Tamaño de la población	Generaciones para alcanzar el mejor resultado	Número máximo de iteraciones realizadas	Tiempo de simulación medido
Algoritmos genéticos	6.26	0%	20	24	50	157.88

CONTINÚA

Colonia de hormigas	5.88	0%	20	23	50	59.93
Cuckoo Search	4.96	0%	20	24	50	104.92

De los datos presentados en la Tabla 20 podemos observar que el mejor controlador encontrado en términos de desempeño para los casos planteados se ha sintonizado por el método de cuckoo search, teniendo una diferencia aproximada de 1 y 2 segundos con respecto a los otros dos controladores. Adicionalmente se puede observar que ninguno de los controladores presenta sobrelongación antes de estabilizarse por lo que el análisis tomara en cuenta únicamente el parámetro de tiempo de estabilización.

Por otro lado, se puede notar que los tres controladores en igualdad de condiciones han alcanzado sus objetivos entre las 23 y 24 iteraciones realizadas, sin embargo, el costo computacional que estos han tenido se puede ver reflejado en el tiempo que le ha tomado a cada algoritmo el ser ejecutado, siendo el algoritmo Colonia de Hormigas el que cuenta con un menor costo computacional, seguido por el algoritmo Cuckoo Search. Cabe recalcar que los tres algoritmos han sido diseñados siguiendo estructuras similares con el fin de realizar un análisis comparativo justo entre ellos.

Es importante mencionar que los algoritmos diseñados parten de una población inicial generada de forma aleatoria para realizar la búsqueda del mejor controlador, por lo que no es factible asegurar que uno de ellos entregará mejores resultados en todas las ocasiones. Por otro lado, el tiempo de simulación medido en la ejecución del algoritmo si es un factor constante que tiene cada uno de los algoritmos diseñados, por lo que es recomendable tomar en cuenta

el costo computacional del algoritmo acompañado de la complejidad del mismo para la selección de uno de estos métodos.

Adicionalmente, el desempeño alcanzado por estos algoritmos depende del rango de las ganancias que planteamos y del tamaño de la población con la que trabajemos, por lo que si no conocemos al sistema en su totalidad, es una buena ayuda realizar una sintonización estándar por cualquier método y a partir de los resultados obtenidos plantear los rangos de búsqueda para optimizar los parámetros calculados con los métodos de optimización por algoritmos bio-inspirados.

La siguiente tabla muestra una comparación entre los resultados alcanzados por los algoritmos bio-inspirados y el PID estándar sintonizado por la herramienta PID Tunner.

Tabla 25

Resumen de características tomadas en la sintonización de controladores por algoritmos bio-inspirados y PID Tunner

Controlador	Ganancia Kp	Ganancia Ki	Ganancia Kd	Tiempo de establecimiento	de Sobrelongación
PID Estándar	9.9977	2	0	12.3 s	6%
Algoritmos Genéticos	11.7176	0.4706	1.5216	6.26 s	-
Colonia de Hormigas	10.3652	0.5683	0.5265	5.88 s	-
Cuckoo Search	11.8192	0.6485	0.2712	4.97 s	-

Respecto a los resultados obtenidos por los algoritmos bio-inspirados en comparación al controlador PID estándar sintonizado con la herramienta PID Tunner podemos notar una notable ventaja tanto en tiempo de estabilización como en sobrelongación medida. De manera

gráfica la figura 42 de la sección 4.2.2 nos permite verificar el comportamiento de cada controlador, donde podemos observar una respuesta similar en los tres casos de sintonización por algoritmos bio-inspirados que destacan en comparación al comportamiento del controlador sintonizado por la herramienta *PID Tunner*. Ya que los rangos utilizados en la búsqueda de los mejores controladores en los algoritmos bio-inspirados fue definido en base a las ganancias calculadas por la sintonización realizada por *PID Tunner*, es factible asegurar que los algoritmos de optimización mejorarán o mantendrán el desempeño obtenido por el controlador original.

Los resultados obtenidos por las pruebas de rechazo a las perturbaciones aplicados a los 4 controladores sintonizados se pueden apreciar en la siguiente tabla

Tabla 26

Resumen de resultados obtenido en rechazo a las perturbaciones por los controladores sintonizados

Controlador	ts ante perturbación de - 0.1V	ts ante perturbación de - 0.2V
PID Estándar	2.3 s	7.6 s
Algoritmos Genéticos	2.7 s	3.9 s
Colonia de Hormigas	2.7 s	3.8 s
Cuckoo Search	2.4 s	3.4 s

Como se puede observar en las figuras 54 y 56 de la sección 4.2.2, el comportamiento obtenido por los controladores sintonizados por algoritmos bio-inspirados mantienen un comportamiento y valores de reacción similares en respuesta a las perturbaciones aplicadas, manteniendo tiempos similares de reacción y sin presentar sobrelongaciones en la señal. Por otro lado, el controlador PID estándar presenta una respuesta ligeramente más rápida que los demás controladores, sin embargo, esto genera una pequeña sobrelongación en la señal. Para

el caso con una perturbación de $-0.1[V]$ el controlador PID mantiene su sobreelongación dentro del rango aceptable de error (2%), por lo que se le puede considerar con una respuesta ligeramente más rápida en contraste a los algoritmos bio-inspirados; sin embargo, para el segundo caso, con una perturbación de $-0.2[V]$, la sobreelongación producida por el controlador PID estándar

De esto podemos concluir que la respuesta del controlador PID estándar presenta un porcentaje de sobreelongación que depende de la magnitud de la perturbación, por lo cual es más recomendable el comportamiento presentado por los controladores sintonizados por algoritmos bio-inspirados, ya que estos mantienen un tiempo de respuesta similar al PID convencional sin la desventaja de presentar sobreelongaciones en su respuesta.

4.3.2. Análisis de resultados medidos

La siguiente tabla nos presenta los resultados que fueron tomados de las mediciones realizadas durante la aplicación de los controladores sintonizados en la planta de temperatura PCT-2:

Tabla 27

Detalle de mediciones tomada en la aplicación de los controladores sintonizados

Controlador	Tiempo de establecimiento [s]	Porcentaje de sobreelongación
PID	-	26.1%
Algoritmos Geneticos	12.65	6.4%
Colonia de hormigas	10.95	6.1%
Cuckoo Search	11.45	3.9%

A diferencia de los casos simulados, podemos observar que en todos los casos medidos el porcentaje de sobrelongación y el tiempo de establecimiento medido aumenta en comparación de los resultados obtenidos en las simulaciones. Sin embargo, el patrón de comportamiento de las señales se mantiene, como se puede observar en la figura 57 de la sección 4.2.3, presentando un mínimo porcentaje de sobrelongación y un menor tiempo de establecimiento en los tres casos de controladores sintonizados por algoritmos bio-inspirados en comparación a las medidas tomadas del controlador PID estándar. De acuerdo a las medidas tomadas podemos notar un mejor desempeño respecto al tiempo de establecimiento en el controlador sintonizado por Colonia de hormigas (al igual que en caso simulado) con 10.95 segundos, sin embargo, ya que en esta ocasión las señales medidas si presentan sobrelongación, se puede observar que la respuesta obtenida por el controlador sintonizado por Cuckoo Search entrega la menor sobrelongación de todas con 3.9%.

Las figuras 60 61 y 62 de la sección 4.2.3 nos muestran el comportamiento de las señales ante una perturbación en contraste con el comportamiento de la señal de control. En estas imágenes se puede observar un cambio brusco y una reacción rápida a los cambios de temperatura en la señal, por lo cual los controladores sintonizados por algoritmo bio-inspirados presentan una mayor estabilidad que el controlador PID estándar mostrado en la figura 63.

En cuanto al rechazo a las perturbaciones, ninguno de los 3 controladores salió del rango de error aceptable durante los 10 segundos de ejecución de las perturbaciones aplicadas, por lo que no existe un tiempo de recuperación medido, sin embargo, en las mismas imágenes

se puede evidenciar la pronta respuesta del controlador al cambio de temperatura producido por la perturbación.

Por otro lado, la respuesta medida por el controlador PID estándar presenta una sobreelongación excesiva de 26.1% que sale de los rangos definidos en el objetivo de control, y no posee un tiempo de establecimiento medido durante las fases de prueba ya que en ningún momento obtuvo un error menor al 2% establecido. Esto producido por la respuesta lenta del controlador que se puede apreciar en la figura 63 de la sección 4.2.3.

La respuesta del controlador PID estándar respecto al rechazo a las perturbaciones se mantuvo dentro de los rangos de error establecidos, y su comportamiento es similar al de los controladores sintonizados por algoritmos bio-inspirados, en contraste a su equivalente simulado que presentó sobreelongaciones en la respuesta del controlador ante ciertas perturbaciones.

4.3.3. Análisis comparativo de resultados simulados con medidos

En base a las mediciones realizadas en la simulación del comportamiento de los controladores sintonizados y el comportamiento real de los controladores aplicados a la planta de temperatura PCT-2, se ha detallado los resultados más significativos en las siguientes tablas:

Tabla 28

Comparación de resultados obtenidos en tiempo de establecimiento.

Controlador	Tiempo de establecimiento simulado	Tiempo de establecimiento medido
PID	12.3 s	-
Algoritmos Genéticos	6.26 s	12.65
Colonia de hormigas	5.88 s	10.95
Cuckoo Search	4.97 s	11.45

Tabla 29

Comparación de resultados obtenidos en sobrelongación

Controlador	Tiempo de establecimiento simulado	Porcentaje de sobrelongación medido
PID	6%	26.1%
Algoritmos Genéticos	0	6.4%
Colonia de hormigas	0	6.1%
Cuckoo Search	0	3.9%

Como se puede observar en las tablas 25 y 26, los tiempos y sobrelongaciones medidas con respecto a las simuladas se aproxima al doble de cada medición. Esto producido por el porcentaje de sobrelongación que se produce en las mediciones reales, ya que esto evita que la señal se estabilice en su primera elevación como en el caso simulado, y tiene que esperar a que la sobrelongación disminuya para estabilizarse.

Es importante mencionar que, a pesar de las diferencias medidas en comparación a los controladores simulados, en los tres casos los controladores diseñados cumplen con el objetivo de control planteado en la sección 3.2. En el caso del controlador PID estándar, su lenta respuesta a las variaciones de temperatura provoca que el sistema no logre estabilizarse en un rango de erro menor al 2% definido.

CAPITULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Dentro del desarrollo de los algoritmos de optimización es de vital importancia definir el o los criterios de optimización que rijan el funcionamiento del algoritmo, en consecuencia, se debe definir las características principales sobre las cuales queremos que actúe nuestro sistema. Estos parámetros pueden ser medidos e interpretados en términos de costo o desempeño para su tratamiento dentro del algoritmo seleccionado. En este trabajo se ha definido los parámetros de sobrelongación y tiempo de establecimiento para determinar el desempeño global de cada solución explorada, asignando un criterio de importancia de 50% a cada parámetro para asegurar que el proceso de búsqueda cumpla con todos los requisitos propuestos en el objetivo de control.

Se desarrolló con éxito el diseño de controladores optimizados por medio de algoritmos bio-inspirados Colonia de hormigas y Cuckoo search en base a los criterios de optimización planteados. El algoritmo colonia de hormigas basa su funcionamiento en la búsqueda de las mejores rutas o soluciones para alcanzar el objetivo de control planteado, basándose en la concentración de feromonas sobre los caminos recorridos y logrando que a cierto número de iteraciones las soluciones encontradas converjan hacia una única ruta con el mejor desempeño. Por otro lado, el algoritmo Cuckoo Search se enfoca en imitar el comportamiento de las aves

parasitarias cuco y su capacidad de localizar los mejores nidos mediante una técnica de búsqueda aleatoria representado por los vuelos de Lévy.

Luego de realizar el respectivo análisis de resultados se ha determinado 4 parámetros fundamentales para realizar la comparación entre los algoritmos diseñados: Por un lado, el tiempo de establecimiento medido en el caso simulado determina que el controlador optimizado mediante el algoritmo Cuckoo Search (CS) se estabiliza un 14.7% más rápido que el controlador optimizado mediante Algoritmos Genéticos (AG) y un 11.76% más rápido que el controlador optimizado por Colonia de Hormigas (CH), mientras que en el caso del controlador implementado sobre el módulo de temperatura se pudo medir una mayor velocidad de establecimiento en el controlador optimizado por CH, siendo este un 15.52% más rápido que el optimizado por AG y un 3.19% más rápido que el controlador optimizado por CS. En el segundo parámetro se logró medir el mejor rendimiento en cuando al porcentaje de sobrelongación en el controlador diseñado por CS, siendo este un 56.41% más pequeño que el controlador optimizado por CH y un 64.1% más pequeño que el controlador optimizado por AG. De igual forma, se ha medido el tiempo de recuperación en rechazo a las perturbaciones en los tres controladores optimizados, concluyendo que el algoritmo cuckoo search se recupera un 12.5% más rápido que los otros dos algoritmos implementados. Y finalmente, se ha medido un tiempo de ejecución de los algoritmos implementados, observando una mayor rapidez de ejecución y un menor costo computacional en el algoritmo CH, el cual se ejecuta un 75.07% más rápido que el algoritmo CS y un 163.34% que AG.

En base a la información extraída, se puede observar que los mejores resultados los entrega las respuestas otorgadas por el algoritmo Cuckoo search, sin embargo, el algoritmo Colonia de hormigas entrega resultados bastante similares en la mitad del tiempo de ejecución.

Al ser algoritmos de búsqueda y optimización de parámetros que parten de la generación aleatoria de individuos, es factible suponer que cada uno logrará un objetivo similar siempre y cuando estén sometido a los mismos parámetros y diseñados bajo la misma estructura. Ya que los tres casos dependen de valores aleatorios iniciales, sería inadecuado asegurar que un algoritmo será mejor que los demás en todos los casos, ya que estos se basan en modelos probabilísticos para llegar a su resolución. Por otro lado, en base a los tiempos de ejecución presentados con anterioridad, podemos afirmar que, para los métodos de desarrollo de algoritmos aplicados en este trabajo, el algoritmo colonia de hormigas presentó un menor costo computacional medido en base al tiempo de necesario para su ejecución, alcanzando su mejor resultado en la generación 24, al igual que los otros dos algoritmos planteados.

El método de optimización por algoritmos genéticos ha sido altamente utilizado por la comunidad científica en comparación a otros métodos de optimización menos explorados, tales como los presentados en el presente trabajo de investigación. Luego de las pruebas realizadas, se puede afirmar que los algoritmos de optimización Colonia de hormigas y Cuckoo search son igualmente válidos en un proceso de optimización en cuanto a los resultados obtenidos, e incluso pueden representar una menor complejidad de ejecución o un menor costo

computacional, por lo cual se pueden considerar alternativas factibles e igualmente válidas que el método de algoritmos genéticos en un proceso de optimización.

Luego de realizar el análisis de los controladores optimizados en comparación con un controlador sintonizado mediante la herramienta PID Tunner, se puede afirmar que los tres controladores optimizados presentan comportamientos muy similares entre sí y una gran ventaja respecto al controlador estándar en términos de tiempo de establecimiento, sobrelongación y rechazo a las perturbaciones. En cuanto a la implementación en la planta de temperatura PCT-2, el comportamiento real de los controladores diseñados presentó variaciones respecto a las simulaciones realizadas, siendo la más destacable el aumento del porcentaje de sobrelongación producido por la inestabilidad de la planta y el rizo presente en las señales estabilizadas, provocado por las fluctuaciones generadas a la salida del sensor IC. Por otro lado, se puede evidenciar la rápida acción de control ante los cambios de temperatura medidos en los controladores diseñados en base a algoritmos bio-inspirados, a diferencia del comportamiento lento del controlador PID estándar que dio como resultado una señal incapaz de estabilizarse en un rango menor a 2%.

5.2. Recomendaciones

Debido a la complejidad que pueden tener algunos algoritmos de optimización, es recomendable plantear un modelo estándar que simplifique los procedimientos planteados en el modelo original del algoritmo, sin descartar las partes fundamentales de este.

Los algoritmos bio-inspirados cuentan con factores de aprendizaje o probabilidades de ocurrencia que modifican el comportamiento de los mismos, cuyos valores suelen permanecer a criterio del diseñador. Para esto es recomendable fundamentarse en investigaciones previas que se enfocan en conseguir los mejores valores para estos parámetros con el fin de mejorar el desempeño general del algoritmo.

Para futuras implementaciones de controladores en la planta de temperatura PCT-2 de la Universidad de las Fuerzas Armadas Espe, se recomienda verificar la linealidad de la señal obtenida por el sensor IC y el correcto funcionamiento de este al momento de realizar la medición de temperatura, ya que el mal funcionamiento de estos puede desencadenar en un modelado inestable y por consecuencia un diseño erróneo de los controladores aplicados.

6. BIBLIOGRAFÍA

Agustín, J. L. (Enero 1998). *Aplicación de algoritmos genéticos al diseño óptimo de sistemas de distribución de energía eléctrica*. Zaragoza.

Barán, B., & Almirón, M. (s.f.). *Colonia de hormigas en un ambiente paralelo asíncrono*. San Lorenzo, Paraguay: Universidad Nacional de Asunción, Centro Nacional de Computación.

Cárdenas Dobson, R. (s.f.). *Diseño de controladores*. Región de Magallanes y de la Antártica Chilena, Chile: Universidad de Magallanes.

Chandrasekaran, K., & Simon, S. P. (2012). *Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm*. *ELSIEVER*, 1-16. Obtenido de Science Direct.

CNP, C. N. (s.f.). *"Plan Nacional de Desarrollo 2017-2021 Toda una Vida"* . Obtenido de http://www.planificacion.gob.ec/wp-content/uploads/downloads/2017/10/PNBV-26-OCT-FINAL_0K.compressed1.pdf

Coy Calixto, C. A. (2005). *Implementación en Hidroinformática de un método de optimización matemática basada en la colonia de hormigas*. Pontificia Universidad Javeriana.

Desarrollo, S. N. (28 de Noviembre de 2017). *El Plan Nacional de Desarrollo 2017-2021 “Toda una Vida”*. Obtenido de <http://www.planificacion.gob.ec/el-plan-nacional-de-desarrollo-2017-2021-toda-una-vida-se-presento-en-riobamba/>

Flores Vilches, J. E. (2015). *Diseño e implementación de algoritmo de búsqueda cuckoo para ajuste de funciones de selección*. Pontificia Universidad Católica de Valparaíso.

Holland, J. (1975). *Adaptation in Natural and Artificial System*. Ann Arbor: University of Michigan Press. Obtenido de <https://elvex.ugr.es/decsai/iaio/slides/G2%20Genetic%20Algorithms.pdf>

Hoyos Pineda, J. G., & Pérez Castillo, J. N. (2007). *Aplicación de algoritmos genéticos a la clasificación de imágenes de satélite en el marco de los servicios Grid Inteligentes Estado del Arte*. Obtenido de Ciencia e Ingeniería Neogradina: <http://www.redalyc.org/articulo.oa?id=91117207>

Insfrán, C., Pinto, D., & Barán, B. (s.f.). *Diseño de Topologías Virtuales en Redes Ópticas. Un enfoque basado en Colonia de Hormigas*. San Lorenzo, Paraguay.

Life, S. B., & BBVA, F. (2008). *SEO Bird Life*. Obtenido de Cuco Común: <https://www.seo.org/ave/cuco-comun/>

López Hernández, J. C. (2014). *Sistemas de control basados en algoritmos genéticos y lógica difusa para la aplicación en procesos industriales*. Bogotá, Colombia: Pontificia Universidad Javeriana.

Márquez Vázquez, J. M. (04 de 2012). *Cálculo de rutas en sistemas de E-Learning utilizando un algoritmo de optimización por colonia de hormigas*. Obtenido de Universidad de Sevilla: https://idus.us.es/xmlui/bitstream/handle/11441/15687/2012marqucalcu_1.pdf?sequence=1

Martínez, M. A., Sanchis, J., & Blasco, X. (2010). *Algoritmos Genéticos Aplicados al Diseño de Controladores Robustos*. Revista Iberoamericana de Automática e Informática industrial, 39-51. Obtenido de <https://polipapers.upv.es/index.php/RIAI/article/view/8107>

Mascotas, L. W. (2015). *Web sobre el cuidado de las mascotas y el mundo de los animales salvajes*. Obtenido de <http://www.webdelasmascotas.com/aves/por-algo-le-dicen-cuco.html>

Mezura, E. (Junio de 2009). *Optimización inspirada en la naturaleza*. Xalapa, Veracruz, Mexico.

Miniguano Miniguano, H. S. (2008). *Diseño e implementación de un controlador robusto para el módulo experimental de control de temperatura de un flujo de aire PCT-2*. Sangolquí .

Mora García, A. M. (2009). *Resolución del Problema Militar de Búsqueda de Camino Óptimo Multiobjetivo Mediante el Uso de Algoritmos de Optimización Basados en Colonias de*

Hormigas . Universidad de Granada, Departamento de arquitectura y Tecnología de Computadoras.

Obando, J., Zamora, J., & Girakdo, F. (s.f.). *Algoritmo bioinspirado en inteligencia de enjambres de optimizacion de colonias de hormigas para el caso del problema de distribucion en planta*. Bobotá-Colombia: Universidad Distrital Francisco José de Caldas-Facultad Tecnológica. Obtenido de <http://repository.udistrital.edu.co/bitstream/11349/3138/1/ObandoSolanoJhinethPaola2015.pdf>

Ogata, K. (1997). *Modern Control Engineering, Third Edition*. Minnesota: Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©1997.

Ortiz, J. J., Perusquía, R., & Montes, J. L. (2003). *Algoritmos bio-inspirados para la obtencion de patrones de barras de control en reactores BWR*. Estado de Mexico: XIII Congreso Tecnico Científico ININ-SUTIN. Obtenido de https://inis.iaea.org/collection/NCLCollectionStore/_Public/36/095/36095239.pdf

Pappa, G. (2012). *Algoritmos Bio-inspirados*. Obtenido de Conceitos e Aplicações em Aprendizado de Máquina: <https://homepages.dcc.ufmg.br/~glpappa/cverao/CursoVerao-Parte1.pdf>

- Parisi, A., Pairisi, F., & David, D. (2006). *Modelos de Algoritmos Genéticos y Redes Neuronales en la Predicción de Índices Bursátiles Asiáticos*. Cuadernos de Economía v.43, 251-284.
- Robles Algarín, C. A. (2010). *Optimización por colonia de hormigas: Aplicaciones y Tendencias*. Revista Ingeniería Solidaria, vol. 6, núm. 10, pp. 83-89.
- Ruge, I. A., & Alvis, M. A. (2010). *Aplicación de los algoritmos genéticos para el diseño de un controlador PID adaptativo*. con-ciencias.
- Sedano García, Á. (9 de 2013). *Metodología De Síntesis Óptima Dimensional De Mecanismos Mediante Algoritmos De Optimización Híbridos*. Obtenido de https://www.researchgate.net/publication/270860970_metodología_de_sintesis_optima_dimensional_de_mecanismos_mediante_algoritmos_de_optimización_híbridos
- Tablada, C., & Torres, G. (s.f.). *Redes neuronales artificiales*. Obtenido de <http://www.famaf.unc.edu.ar/~revm/digital24-3/redes.pdf>
- Trejo, C., López, L., & Ruiz, A. (2013). *Estimacion de parametros para un modelo de crecimiento de cultivos usando algoritmos evolutivos y bio-inspirados*. Chapingo, Estado de México: Universidad Autónoma Chapingo.
- Zarzosa Gómez, M. A. (2017). *artificial, Descripción de comportamientos animales que se utilizan para algoritmos de optimización e inteligencia*. Santa Cruz de Tenerife, España: Universidad de La Laguna.

Zheng, H., & Zhou, Y. (2012). *A Novel Cuckoo Search Optimization Algorithm Base on Gauss Distribution*. *Journal of Computational Information Systems* 8, 4193–4200.