



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO



“REGULACIÓN DE POSICIÓN DE UN HELICÓPTERO PIVOTADO USANDO
HEURÍSTICAS REDUCIDAS BASADAS EN INTELIGENCIA DE ENJAMBRES”

T E S I S

QUE PARA OBTENER EL GRADO ACADÉMICO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

JHOVANI GARCÍA JAIME

DIRECTOR DE TESIS:

DR. JUAN FERNANDO GARCÍA MEJÍA

CODIRECTORES:

DR. EVERARDO EFRÉN GRANDA GUTIÉRREZ

DR. ALLAN ANTONIO FLORES FUENTES

OCTUBRE, 2025

RESUMEN

Una de las áreas de aplicación de la Inteligencia Artificial es el diseño de reguladores de fenómenos físicos como la posición de helicópteros pivotados, los cuales son modelos experimentales montados sobre una base fija que permiten estudiar el control de movimiento en varios ejes. Esta posición es comúnmente controlada por medio de técnicas de control óptimo como los Reguladores Lineales Cuadráticos (LQR), susceptibles a ser diseñados por métodos heurísticos que permiten la búsqueda eficiente de soluciones en el espacio de las matrices de ponderación de los reguladores para sintonizarlos de manera efectiva. Las versiones canónicas de estos métodos suelen ser computacionalmente costosas, por lo que una alternativa es el desarrollo de versiones reducidas que disminuyan el costo computacional sin sacrificar la calidad de las soluciones obtenidas. En esta investigación se evaluaron diversos algoritmos basados en Inteligencia de Enjambres para la sintonización de un LQR en la regulación de la posición de un helicóptero pivotado. Se desarrollaron versiones reducidas de heurísticas bioinspiradas, comparando su desempeño frente a las versiones originales. Los resultados mostraron que las versiones reducidas alcanzan convergencia hacia soluciones óptimas similares a las obtenidas por las heurísticas canónicas, realizando un número menor de operaciones totales debido a la reducción en generaciones y tamaño de población, lo que permite disminuir significativamente el consumo de recursos y los tiempos de cómputo. El análisis estadístico indicó que la mediana y el rango intercuartílico (IQR) de los resultados confirman la estabilidad y reproducibilidad de la optimización, mostrando que ambas versiones tienden a minimizar hacia el mismo punto óptimo. Asimismo, los criterios de desempeño del helicóptero pivotado, como los errores de seguimiento y métricas de respuesta transitoria, evidenciaron que las soluciones obtenidas con heurísticas reducidas mantienen un desempeño equivalente al de los métodos canónicos. Este trabajo contribuye al estudio de la aplicación de heurísticas reducidas basadas en Inteligencia de Enjambres en el diseño de controladores LQR, un campo en el que aún existen escasos desarrollos documentados en la literatura técnica especializada, especialmente en el contexto del control de sistemas físicos no lineales como los helicópteros pivotados.

Palabras clave: Optimización, helicóptero pivotado, Regulador Lineal Cuadrático, heurísticas reducidas.

ABSTRACT

One of the application areas of Artificial Intelligence is the design of regulators for physical phenomena, such as the position of pivoted helicopters, which are experimental models mounted on a fixed base that allow the study of multi-axis motion control. This position is commonly regulated using optimal control techniques, such as Linear Quadratic Regulators (LQR), which can be designed using heuristic methods that enable an efficient search for solutions within the weighting matrices of the controllers for effective tuning. Canonical versions of these methods are usually computationally expensive, so an alternative is the development of reduced versions that lower computational cost without compromising solution quality. In this research, various swarm intelligence-based algorithms were evaluated for the tuning of an LQR in the regulation of a pivoted helicopter's position. Reduced versions of bio-inspired heuristics were developed and compared with the original versions. The results showed that the reduced versions converge toward optimal solutions similar to those obtained by canonical heuristics, performing a smaller total number of operations due to reductions in the number of generations and population size, which significantly decreases resource consumption and computational time. Statistical analysis indicated that the median and interquartile range (IQR) of the results confirm the stability and reproducibility of the optimization, showing that both versions tend to minimize toward the same optimal point. Furthermore, the performance criteria of the pivoted helicopter, including tracking errors and transient response metrics, demonstrated that the solutions obtained with the reduced heuristics maintain performance equivalent to that of the canonical methods. This work contributes to the study of applying reduced swarm intelligence-based heuristics in LQR controller design, a field with limited documented developments in technical literature, particularly in the context of controlling nonlinear physical systems such as pivoted helicopters.

Keywords: Optimization, pivoted helicopter, Linear Quadratic Regulator, reduced heuristics.

ÍNDICE

RESUMEN.....	2
ABSTRACT	3
ÍNDICE DE FIGURAS	7
ÍNDICE DE TABLAS.....	9
1 INTRODUCCIÓN.....	10
2 PLANTEAMIENTO DEL PROBLEMA.....	13
2.1 Definición del problema	13
2.2 Preguntas de investigación	14
2.3 Objetivos de la investigación.....	14
2.3.1 Objetivo general:	14
2.3.2 Objetivos específicos:.....	15
2.4 Hipótesis.....	15
2.5 Justificación	15
2.6 Impactos	16
3 FUNDAMENTOS TEÓRICOS	17
3.1 Helicóptero pivotado	17
3.1.1 Descripción del helicóptero	17
3.1.2 Modelo matemático de un helicóptero pivotado	18
3.2 Regulador Lineal Cuadrático.....	20
3.2.1 Fundamentos del control LQR y función de costo	21
3.2.2 Ajuste de las matrices Q y R	22
3.2.3 Criterios de desempeño	23
3.3 Heurísticas de Inteligencia de Enjambres.....	25
3.3.1 Principios bioinspirados	26

3.3.2	Particle Swarm Optimization	27
3.3.3	Whale Optimization Algorithm	28
3.3.4	White Shark Optimizer	31
3.3.5	SailFish Optimizer	35
3.3.6	Beluga Whale Optimization	38
3.3.7	Orca Predation Algorithm	41
3.4	Heurísticas reducidas	44
3.4.1	Concepto y motivación	45
3.4.2	Estrategias de reducción	45
4	ESTADO DEL ARTE	47
4.1	Métodos de control de posición de un helicóptero pivotado	47
4.2	Sintonización de controladores mediante Inteligencia Computacional	50
4.3	Matriz de referencias del estado del arte	52
5	METODOLOGÍA	54
5.1	Método	54
5.2	Modelado y simulación del helicóptero	56
5.3	Diseño del controlador LQR	57
5.4	Definición de la función objetivo	59
5.5	Implementación de algoritmos heurísticos canónicos	60
5.6	Diseño de heurísticas reducidas	63
5.7	Análisis estadístico inferencial	65
6	RESULTADOS	66
6.1	Desempeño de las heurísticas canónicas	66
6.2	Sintonización del LQR con heurísticas canónicas	74
6.3	Desempeño de las heurísticas reducidas	77

6.4	Sintonización del LQR con heurísticas reducidas	85
6.5	Comparativa de desempeño de las heurísticas canónicas frente a las reducidas..	87
7	CONCLUSIONES.....	99
	BIBLIOGRAFÍA	101

ÍNDICE DE FIGURAS

Figura 1.1 Modelo de helicóptero pivotado.	10
Figura 3.1 Diagrama esquemático del helicóptero pivotado.	17
Figura 3.2 Diagrama del espacio de estados del helicóptero pivotado de 2-DOF.	20
Figura 3.3 Diagrama de bloques del Regulador Lineal Cuadrático.	21
Figura 3.4 Respuesta típica de un sistema y sus criterios de desempeño temporal.	24
Figura 3.5 Diagrama de flujo del algoritmo PSO.	27
Figura 3.6 Caza por red de burbujas de una ballena jorobada.	29
Figura 3.7 Diagrama de flujo del algoritmo WOA.	31
Figura 3.8 Diagrama de flujo del algoritmo WSO.	35
Figura 3.9 Diagrama de flujo del algoritmo SFO.	38
Figura 3.10 Diagrama de flujo del algoritmo BWO.	41
Figura 3.11 Diagrama de flujo del algoritmo OPA.	44
Figura 5.1 Esquema metodológico propuesto.	55
Figura 5.2 Respuesta inestable del helicóptero pivotado sin regulación.	57
Figura 5.3 Respuesta del helicóptero pivotado con LQR analítico.	59
Figura 5.4. Estructura general de implementación de las heurísticas canónicas.	62
Figura 5.5. Estructura general de implementación de las heurísticas reducidas.	64
Figura 6.1 Curva de convergencia del algoritmo PSO.	67
Figura 6.2 Curva de convergencia del algoritmo BWO.	68
Figura 6.3 Curva de convergencia del algoritmo WSO.	69
Figura 6.4 Curva de convergencia del algoritmo WOA.	70
Figura 6.5 Curva de convergencia del algoritmo SFO.	71
Figura 6.6 Curva de convergencia del algoritmo OPA.	72
Figura 6.7 Comparativa de convergencia de los algoritmos canónicos.	73
Figura 6.8 Respuesta al escalón del helicóptero con el LQR sintonizado mediante la heurística de mejor desempeño (WOA) y la de menor desempeño (SFO).	74
Figura 6.9 Curva de convergencia del algoritmo PSO reducido.	78
Figura 6.10 Curva de convergencia del algoritmo BWO reducido.	79
Figura 6.11 Curva de convergencia del algoritmo WOA reducido.	80

Figura 6.12 Curva de convergencia del algoritmo WSO reducido.....	81
Figura 6.13 Curva de convergencia del algoritmo SFO reducido.	82
Figura 6.14 Curva de convergencia del algoritmo OPA reducido.	83
Figura 6.15 Comparativa de convergencia de los algoritmos reducidos.....	84
Figura 6.16 Respuesta al escalón del helicóptero con el LQR sintonizado mediante la heurística reducida de mejor desempeño (μ WSO) y la de menor desempeño (μ BWO).	85
Figura 6.17 Comparativa de curvas de convergencia del PSO vs μ PSO.	88
Figura 6.18 Comparativa de curvas de convergencia del BWO vs μ BWO.	89
Figura 6.19 Comparativa de curvas de convergencia del WSO vs μ WSO.	90
Figura 6.20 Comparativa de curvas de convergencia del WOA vs μ WOA.	91
Figura 6.21 Comparativa de curvas de convergencia del SFO vs μ SFO.	92
Figura 6.22 Comparativa de curvas de convergencia del OPA vs μ OPA.	93
Figura 6.23 Respuesta al escalón del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO).	95
Figura 6.24 Respuesta del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO) frente a una referencia tipo pulso.	96
Figura 6.25 Respuesta del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO) frente a una referencia tipo escalera.	97
Figura 6.26 Respuesta del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO) ante perturbaciones.....	98

ÍNDICE DE TABLAS

Tabla 3.1 Parámetros del modelo dinámico del helicóptero pivotado.	19
Tabla 3.2 Estrategias de reducción en metaheurísticas.	46
Tabla 4.1 Síntesis de trabajos relacionados.	52
Tabla 5.1. Matrices de estado del helicóptero pivotado.	56
Tabla 5.2. Matrices Q y R del Regulador Lineal Cuadrático sintonizado analíticamente. ..	58
Tabla 5.3 Parámetros homogéneos de implementación de los algoritmos heurísticos.	61
Tabla 5.4 Parámetros de configuración de las versiones reducidas.	64
Tabla 6.1 Resultados de la prueba de repetibilidad de los algoritmos canónicos.	66
Tabla 6.2 Criterios de desempeño del LQR sintonizado con heurísticas canónicas.	75
Tabla 6.3 Matrices solución sintonizadas con heurísticas canónicas.	76
Tabla 6.4 Resultados de la prueba de repetibilidad de los algoritmos reducidos.	77
Tabla 6.5 Criterios de desempeño del LQR sintonizado con heurísticas reducidas.	86
Tabla 6.6 Matrices solución sintonizadas con heurísticas reducidas.	87
Tabla 6.7 Comparación entre heurísticas canónicas y reducidas.	94

1 INTRODUCCIÓN

Un helicóptero es un vehículo aéreo que necesita para su vuelo de fuerzas de sustentación y propulsión, las cuales son generadas por dos rotores principales. Estos permiten que la aeronave pueda controlar su posición y orientación en el espacio, generando movimiento en cuatro grados de libertad (4-DOF, por las siglas en inglés de Degrees Of Freedom) [1]. El eje de alabeo, le permite inclinarse hacia la derecha o izquierda; el eje de traslación le permite moverse hacia adelante o atrás, el eje de cabeceo controla la inclinación de la nariz hacia arriba o abajo; y el eje de guiñada, le permite girar sobre su propio eje hacia la derecha o izquierda para cambiar de dirección sin alterar su posición horizontal [2].

Una manera de realizar actividades de investigación y pruebas sobre un helicóptero, en un entorno controlado, es limitando los grados de libertad de este. Esto se logra a través de un helicóptero pivotado (véase Figura 1.1) definido como una plataforma a escala que emula el comportamiento de un helicóptero real con solo dos grados de libertad (2-DOF) [3]. Para ello se ancla el modelo a un eje vertical que limita el movimiento de la aeronave para lograr exclusivamente la elevación y el control de giros. Este tipo de plataforma de emulación permite la implementación y prueba de técnicas de control de posición, las cuales afectan directamente la dinámica de vuelo, incluyendo la estabilidad, la rapidez de respuesta y la precisión en el control de la elevación y el giro del helicóptero [4].

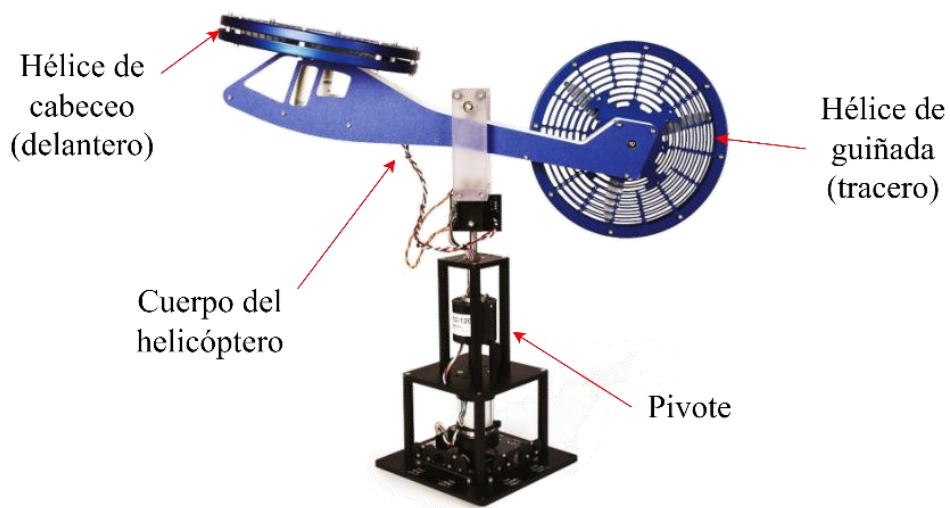


Figura 1.1 Modelo de helicóptero pivotado.
(Tomado de [5]).

La investigación y desarrollo experimental sobre los helicópteros pivotados tiene como objetivos, el análisis y modelado matemático de la aeronave, la simulación de sistemas aeronáuticos, la detección y corrección de fallos, y la evaluación de métodos avanzados de regulación tales como el uso de distintas técnicas de control moderno y de Inteligencia Artificial (*Artificial Intelligence, AI, por sus siglas en inglés*) [6-8]. Esto dado que los enfoques clásicos no suelen regular sistemas cuya respuesta en el dominio del tiempo tiende a la inestabilidad, como el helicóptero de 2-DOF, cuyo modelo dinámico es no lineal, acoplado y subactuado [9].

La Inteligencia Computacional (*Computing Intelligence, CI, por sus siglas en inglés*) se ha consolidado como un área de la AI para el desarrollo de soluciones en sistemas complejos y dinámicos, como los sistemas de control y los de posición. A diferencia de los enfoques clásicos, la CI emplea técnicas capaces de adaptarse y aprender en entornos no lineales al no depender de modelos matemáticos exactos [10]. En la literatura especializada, varios autores reportan el uso de técnicas de CI en el ajuste de controladores para helicópteros de 2-DOF. Entre los paradigmas documentados, se encuentran las Redes Neuronales Artificiales (*Artificial Neural Networks, ANN, por sus siglas en inglés*), los algoritmos heurísticos, la Lógica Difusa (*Fuzzy Logic, FL, por sus siglas en inglés*), así como el desarrollo de enfoques híbridos de las técnicas antes mencionadas [3, 5, 11-15].

En la actualidad, la mayoría de los trabajos del área de control automático han documentado el diseño de controladores para helicópteros pivotados usando diversas plataformas de prueba para evaluar la efectividad de varias estrategias de control. Entre estas, se destacan el control robusto, el control inteligente y el control óptimo [3]. En particular, este último, ha ganado creciente atención en el diseño de controladores basados en el Regulador Lineal Cuadrático (LQR, por sus siglas en inglés) debido a su simplicidad de implementación y alta precisión [16].

El LQR es una técnica de control óptimo que busca minimizar una función de costo que compensa la regulación de los estados del sistema y el esfuerzo requerido en las señales de control. El diseño del LQR se basa en encontrar la ley de control que minimice la función de costo J , la cual se define como

$$J = \int_0^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \quad 1.1$$

donde Q es una matriz simétrica semidefinida positiva que pondera el vector transpuesto (T) de estados del sistema x , R es una matriz simétrica definida positiva que pondera las entradas de control u , para cada instante de tiempo t .

La sintonización de los parámetros del controlador LQR, es decir, la elección adecuada de las matrices Q y R , es crucial para el desempeño del sistema controlado [17] puesto que estas capturan las dinámicas del sistema para cumplir con los objetivos de control. De acuerdo con el estado del arte, los algoritmos heurísticos, presentan una alternativa eficiente para la optimización de estos parámetros al permitir explorar un espacio de soluciones posibles y converger hacia una solución óptima. En este sentido, es posible definir como objetivo de esta propuesta de investigación el uso de técnicas de Inteligencia Computacional para sintonizar los parámetros de un LQR aplicado al control de posición de un helicóptero pivotado.

2 PLANTEAMIENTO DEL PROBLEMA

2.1 Definición del problema

Los helicópteros pivotados tienen aplicaciones en la investigación y control de sistemas no lineales, dado que permiten estudiar la dinámica de vuelo de esta y otras aeronaves no tripuladas con características similares. Sin embargo, debido a su naturaleza inestable y a la presencia de no linealidades en su modelo dinámico, la regulación precisa de su posición requiere estrategias de control que garanticen su desempeño y estabilidad [18]. Uno de los enfoques más utilizados para la regulación de este tipo de sistemas es el control LQR, el cual permite minimizar una función de costo basada en los errores de estado y el esfuerzo de control. Las implementaciones del LQR reportadas en [19-22] presentan tiempos de estabilización en un intervalo de 20 a 30 segundos, y errores estacionarios que alcanzan el 5%.

La efectividad del LQR depende de la adecuada selección de los parámetros de ponderación Q y R . En la literatura [23-25] han explorado técnicas para la sintonización de estos parámetros, un ejemplo de esto son los algoritmos heurísticos basados en Inteligencia de Enjambres. No obstante, el costo computacional de algoritmos en su versión canónica comúnmente es elevado, derivado de que se requieren ajustes iterativos para evaluar el impacto de las distintas configuraciones de los parámetros propios de cada algoritmo, así como una evaluación final de repetibilidad para verificar que este mantengan la eficacia en la regulación de posición del helicóptero.

Dado el contexto descrito, surge la necesidad de explorar versiones reducidas de los algoritmos heurísticos, que mantengan la capacidad de exploración y explotación de estos métodos sin incurrir en elevados costos computacionales. La reducción de estos algoritmos permitiría mejorar la eficiencia del proceso de sintonización, facilitando su aplicación en estudios de simulación extensivos o en escenarios donde se requiera ajustar el regulador de manera más ágil.

El presente trabajo de investigación se enfoca en evaluar el desempeño de versiones reducidas de algoritmos heurísticos basados en Inteligencia de Enjambres aplicados a la

sintonización del LQR para la regulación de la dinámica de un helicóptero pivotado a través de un modelo matemático representativo. Esta evaluación se realiza mediante análisis estadístico inferencial, con el fin de determinar si existen diferencias estadísticamente significativas entre los algoritmos heurísticos en su versión canónica y sus variantes reducidas, tanto en términos de calidad de la solución como en reducción del costo computacional.

2.2 Preguntas de investigación

De la problemática anteriormente descrita, se formula la siguiente interrogante:

- ¿Cómo se determinan las ventajas de algoritmos heurísticos reducidos con relación a sus versiones canónicas aplicados al diseño de un Regulador Lineal Cuadrático para la regulación de la posición del helicóptero pivotado?

A partir de la pregunta de investigación, es posible plantear las siguientes preguntas auxiliares:

- ¿Cuáles son las técnicas de optimización inteligente aplicadas al control de helicópteros pivotados en la literatura existente?
- ¿Cómo se regula la posición de un helicóptero pivotado?
- ¿Cómo se puede mejorar el funcionamiento de un controlador LQR aplicado a la regulación de posición del helicóptero pivotado?
- ¿Cómo se determina cuál método heurístico mejora el funcionamiento de un controlador LQR aplicado a la regulación de posición del helicóptero pivotado?

2.3 Objetivos de la investigación

Las preguntas antes descritas permiten la construcción de los objetivos de esta propuesta investigativa.

2.3.1 Objetivo general:

Evaluar algoritmos soportados en Inteligencia de Enjambres en el diseño de un Regulador Lineal Cuadrático para la regulación de la posición de un helicóptero pivotado simulado, por medio de heurísticas reducidas.

2.3.2 Objetivos específicos:

- Revisar la literatura que aborde la optimización inteligente aplicada al control de helicópteros pivotados.
- Diseñar un controlador LQR para la posición de los ejes de guiñada y cabeceo del helicóptero pivotado.
- Sintonizar un regulador LQR por medio de heurísticas basadas en Inteligencia de Enjambres.
- Determinar diferencias estadísticas en la respuesta de un controlador LQR con respecto al ajuste de diferentes heurísticas basadas en Inteligencia de Enjambres.

2.4 Hipótesis

Es posible regular la posición de un helicóptero pivotado mediante la sintonización de un Regulador Lineal Cuadrático por medio de heurísticas reducidas basadas en Inteligencia de Enjambres, las cuales disminuyen el costo computacional respecto a las versiones canónicas y permiten alcanzar un error en estado estable menor al 5% y un tiempo de establecimiento de 10 segundos.

2.5 Justificación

Existen diversas técnicas de control clasificadas en control clásico y control moderno. Las técnicas clásicas se basan en el análisis de sistemas de una entrada y una salida (SISO, por sus siglas en inglés) en el dominio de la frecuencia, adecuadas para manipular sistemas lineales y de orden bajo. En contraste, las técnicas modernas utilizadas para manejar dinámicas complejas, no lineales e inestables en lazo abierto, como las del helicóptero pivotado, y pueden operar con sistemas de múltiples entradas y múltiples salidas (MIMO, por sus siglas en inglés) en el dominio del tiempo [3], [26].

Un ejemplo de técnicas de control moderno es el Regulador Lineal Cuadrático o LQR, el cual aplicado al control de posición de un helicóptero pivotado, requiere el ajuste preciso de sus parámetros Q y R fundamentales para asegurar un desempeño adecuado del sistema. Esto permite transformar un problema de control en uno de optimización donde el objetivo es encontrar los valores de Q y R que minimicen una determinada función de costo que, en

consecuencia, mejore la respuesta del sistema, un aspecto estudiado en la literatura especializada.

Los algoritmos heurísticos, y en particular aquellos basados en Inteligencia de Enjambres, explorados en la literatura especializada de esta investigación, se presentan como una alternativa para abordar la sintonización del LQR. Estos enfoques, a diferencia de los métodos deterministas tradicionales, los cuales pueden requerir una comprensión profunda del sistema y dependen de modelos matemáticos específicos, permiten explorar el espacio de soluciones de manera eficiente y flexible, sin la necesidad de conocer la dinámica exacta del sistema.

La aplicación directa de algoritmos canónicos basados en Inteligencia de Enjambres puede resultar computacionalmente costosa. La naturaleza iterativa y la cantidad de parámetros involucrados en estos algoritmos pueden llevar a tiempos de procesamiento elevados y a una complejidad innecesaria. Por ello, es pertinente explorar el desarrollo de microheurísticas, las cuales buscan reducir el costo computacional sin sacrificar la calidad de las soluciones obtenidas. Las heurísticas reducidas, también llamadas micro (μ) permiten disminuir el número de iteraciones requeridas, la cantidad de agentes en el enjambre o simplificar las reglas de interacción de los elementos de la población denominados soluciones, traduciéndose en una implementación más rápida y eficiente.

2.6 Impactos

El desarrollo de esta propuesta de investigación tiene un impacto científico, puesto que se desarrollaron las versiones reducidas de los algoritmos heurísticos basados en la Inteligencia de Enjambres aplicados a la sintonización del LQR para la regulación de la posición de un helicóptero pivotado, que no se documentan en la literatura especializada revisada en este proceso investigativo.

3 FUNDAMENTOS TEÓRICOS

3.1 Helicóptero pivotado

3.1.1 Descripción del helicóptero

Un helicóptero pivotado de dos grados de libertad (2-DOF) se define como una plataforma experimental diseñada para la implementación, validación y análisis de algoritmos de control aplicados a sistemas dinámicos aeroespaciales. La configuración pivotada, como la que se muestra en la Figura 3.1, consiste en una estructura mecánica articulada que emula el cuerpo de la aeronave acoplada a un eje vertical fijo, lo cual elimina los grados de libertad asociados a las traslaciones lineales y al alabeo, centrándose así en los movimientos de cabeceo (pitch, θ) y guiñada (yaw, ψ). Esta simplificación permite el estudio de fenómenos asociados a la dinámica de vuelo de la plataforma, dentro de un entorno controlado y seguro [27].

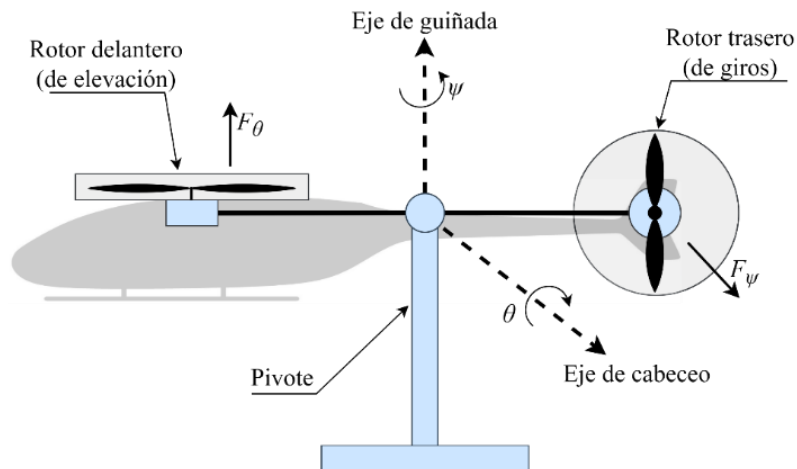


Figura 3.1 Diagrama esquemático del helicóptero pivotado.
(Adaptado de [5]).

El helicóptero pivotado consta de dos motores de corriente directa independientes que accionan hélices montadas en extremos opuestos de una estructura rígida. Cada hélice produce una fuerza de empuje (F_θ y F_ψ) proporcional al voltaje aplicado al motor en el que se encuentra acoplada. El empuje generado en la hélice frontal provoca un momento respecto al eje de rotación lateral que modifica el ángulo de elevación, mientras que el empuje del motor trasero induce una rotación sobre el eje vertical, modificando el ángulo de guiñada. La disposición de las hélices genera una interacción entre los dos grados de libertad, obligan a considerar un acoplamiento en el modelado y control de la plataforma [28].

3.1.2 Modelo matemático de un helicóptero pivotado

El análisis dinámico del helicóptero pivotado puede representarse mediante un modelo lineal en espacio de estados que facilita el diseño de estrategias de control. El sistema posee dos grados de libertad, correspondientes a los movimientos angulares de cabeceo (θ) y guiñada (ψ). Para cada instante de tiempo t , se define el vector de estado $x(t)$, que incluye los ángulos y sus derivadas temporales para capturar las velocidades angulares asociadas tal como se presenta en la ecuación 3.1, donde el superíndice T indica la transpuesta del vector [29].

$$x(t) = [\theta(t) \ \psi(t) \ \dot{\theta}(t) \ \dot{\psi}(t)]^T \quad 3.1$$

El sistema es controlado por dos señales de entrada que corresponden a los voltajes aplicados a los motores DC que hacen girar las hélices. Estas entradas se agrupan en el vector definido por la ecuación 3.2 [29],

$$u(t) = [V_f(t) \ V_b(t)]^T \quad 3.2$$

donde $V_f(t)$ controla el motor frontal (cabeceo) y $V_b(t)$ el motor trasero (guiñada). La descripción matemática del comportamiento del sistema, considerando una linealización alrededor del punto de equilibrio ($\theta = 0, \psi = 0, \dot{\theta} = 0, \dot{\psi} = 0$), se expresa mediante el modelo de espacio de estados mostrado en la ecuación 3.3 [30].

$$\dot{x}(t) = Ax(t) + Bu(t) \quad 3.3$$

La matriz A contiene los coeficientes que determinan la dinámica inherente del sistema, mientras que B relaciona las señales de entrada con la evolución de los estados. En la Tabla 3.1 se presentan los parámetros utilizados en la formulación del modelo, donde se incluyen constantes de fricción, inercia, masa y coeficientes de empuje y torque inducido por los motores en sus respectivos ejes de acción [29]. Estas matrices se definen como se muestra en las ecuaciones 3.4 y 3.5.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{B_p}{J_{eq_p} + m_h l_{cm}^2} & 0 \\ 0 & 0 & 0 & -\frac{B_y}{J_{eq_y} + m_h l_{cm}^2} \end{bmatrix} \quad 3.4$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{pp}}{J_{eq_p} + m_h l_{cm}^2} & \frac{K_{py}}{J_{eq_p} + m_h l_{cm}^2} \\ \frac{K_{yp}}{J_{eq_y} + m_h l_{cm}^2} & \frac{K_{yy}}{J_{eq_y} + m_h l_{cm}^2} \end{bmatrix} \quad 3.5$$

Tabla 3.1 Parámetros del modelo dinámico del helicóptero pivotado.
(Adaptado de [29]).

Variable	Descripción	Valor	Unidad
B_p	Amortiguamiento viscoso equivalente en el eje de cabeceo.	0.8000	N/V
B_y	Amortiguamiento viscoso equivalente en el eje de guiñada.	0.3180	N/V
J_{eq_p}	Momento de inercia total respecto al eje de cabeceo.	0.0384	kg·m ²
J_{eq_y}	Momento de inercia total respecto al eje de guiñada.	0.0432	kg·m ²
m_h	Masa total móvil del helicóptero.	1.3872	kg
l_{cm}	Distancia del centro de masa al eje de cabeceo.	0.1860	m
K_{pp}	Constante de empuje generada por el motor trasero (eje de guiñada).	0.2040	N·m/V
K_{py}	Constante de torque inducido sobre el eje de cabeceo por el motor trasero.	0.0068	N·m/V
K_{yp}	Constante de torque inducido sobre el eje de guiñada por el motor frontal.	0.0219	N·m/V
K_{yy}	Constante de torque directo sobre el eje de guiñada producido por el motor trasero.	0.0720	N·m/V

La salida del sistema está compuesta por los ángulos de cabeceo y guiñada [30]. Esta relación se expresa mediante la ecuación 3.6,

$$y(t) = Cx(t) + Du(t) \quad 3.6$$

donde la matriz C selecciona los primeros dos estados y la matriz D representa el efecto directo de la entrada sobre la salida. Dado que en este sistema no existe tal efecto (no hay

3.2.1 Fundamentos del control LQR y función de costo

El LQR es una técnica de control óptimo que busca determinar una ley de realimentación de estados de la forma en la ecuación 3.9, asumiendo que la referencia $r(t)$ actúa como una entrada simple (por ejemplo, una señal de escalón) [11],

$$u(t) = -Kx(t) \quad 3.9$$

donde $x(t) \in \mathbb{R}^n$ representa el vector de n estados del sistema, $u(t) \in \mathbb{R}^m$ es el vector m de entradas de control, y $K \in \mathbb{R}^{m \times n}$ es la matriz de ganancia que debe ser calculada. Esta ley de control se aplica a sistemas lineales e invariantes en el tiempo, descritos mediante el modelo en espacio de estados como el de la ecuación 3.2, permitiendo obtener el esquema de bloques de la Figura 3.3 [11], [30].

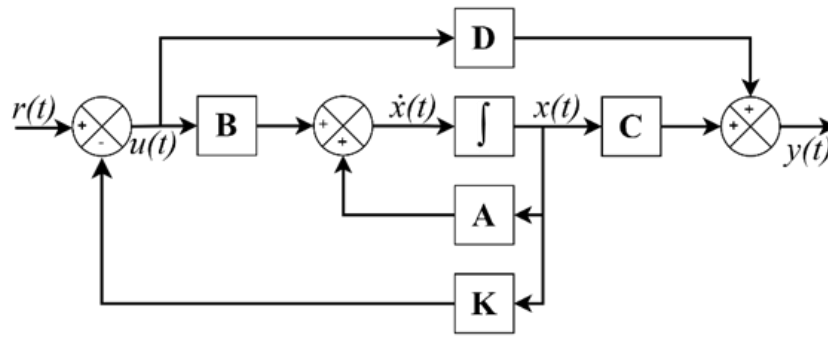


Figura 3.3 Diagrama de bloques del Regulador Lineal Cuadrático. (Adaptado de [30]).

El objetivo del LQR es determinar una matriz K que minimice una función de costo, la cual evalúa simultáneamente el desempeño del sistema y el esfuerzo de control aplicado. Esta función de costo está definida por la expresión 3.10 [11].

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad 3.10$$

En la expresión mostrada en 3.10 se realiza una penalización sobre los estados del sistema a través de $Q \in \mathbb{R}^{n \times n}$ así como la ponderación del esfuerzo aplicado por las señales de control por medio de $R \in \mathbb{R}^{m \times m}$. La matriz Q debe ser simétrica y semidefinida positiva, mientras que R debe ser simétrica y definida positiva, asegurando así la existencia de una solución óptima [11].

El término $x^T Q x$, dentro de la función de costo J , caracteriza las desviaciones del sistema respecto al equilibrio deseado, promoviendo una respuesta cercana a la referencia. Por su parte, el término $u^T R u$ impone restricciones al uso energético, limitando la magnitud de las señales de control y , con ello, protegiendo los actuadores frente a saturaciones o esfuerzos innecesarios [26], [31].

Para determinar la matriz K , es necesario resolver la ecuación algebraica de Riccati, una ecuación matricial no lineal [31], que se expresa como en la ecuación 3.11. En esta expresión, $P \in \mathbb{R}^{n \times n}$ representa la solución que garantiza la estabilidad del sistema en lazo cerrado. Una vez obtenida la matriz P , la ganancia de control óptima se calcula directamente con la relación establecida en la ecuación 3.12. Esta solución asegura que la dinámica del sistema en lazo cerrado, definida por $\dot{x}(t) = (A - B K)x(t)$, sea estable en términos asintóticos [26], [31].

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad 3.11$$

$$K = R^{-1} B^T P \quad 3.12$$

3.2.2 Ajuste de las matrices Q y R

El desempeño del controlador LQR depende fundamentalmente de la adecuada elección de las matrices de ponderación Q y R , dado que estas determinan la forma en que se penalizan las desviaciones del sistema respecto a su referencia y el esfuerzo de control requerido para corregirlas. Aunque la formulación matemática del problema garantiza la existencia de una solución óptima mediante la resolución de la ecuación algebraica de Riccati (3.11), dicha solución está condicionada a los valores de estas matrices, los cuales deben ser definidos por el diseñador [26].

En la práctica, una de las estrategias más comunes para ajustar los valores de las matrices de ponderación consiste en asignar de forma empírica valores positivos a los elementos diagonales de Q y R . Además del enfoque empírico anterior, existe una alternativa sistemática para seleccionar las matrices Q y R que parte directamente de la estructura del sistema, donde la matriz Q se define como el producto matricial de la matriz de estados $C^T \cdot C$ y asumir una matriz identidad para R [26]. Si bien ambas metodologías pueden resultar efectivas, su aplicación requiere de validación experimental y evaluación de resultados.

3.2.3 Criterios de desempeño

La validación del desempeño de un sistema controlado mediante estrategias como el LQR no se limita únicamente a garantizar la estabilidad en lazo cerrado, sino que requiere una evaluación cuantitativa del comportamiento dinámico mediante métricas objetivas. Para ello, se recurre a indicadores clásicos e integrales que permiten caracterizar la calidad de la respuesta temporal ante entradas de referencia o perturbaciones.

3.2.3.1 Indicadores temporales clásicos

Uno de los criterios más empleados es el *tiempo de establecimiento* (t_s , *settling time*), el cual se define como el menor tiempo después del cual la respuesta del sistema permanece, de forma continua, dentro de una banda de tolerancia δ alrededor del valor final y_{ref} , es decir, cuando se cumple que $|y(t) - y_{ref}| \leq \delta \cdot |y_{ref}|$ para todo $t \geq t_s$. Típicamente, se consideran bandas de tolerancia δ del 2% o 5%, según la exigencia del sistema. Este parámetro mide cuán rápido el sistema alcanza una región aceptable de desempeño [32], [33].

El *tiempo de subida* (t_r , *rise time*), corresponde al intervalo necesario para que la salida del sistema pase por primera vez desde un porcentaje bajo hasta un porcentaje alto del valor final, típicamente entre el 10% y el 90% del valor final y_{ref} . Esta métrica cuantifica la velocidad inicial de respuesta, sin considerar el sobreimpulso [32], [33].

El *sobreimpulso máximo* (M_p) corresponde a la máxima desviación de la respuesta respecto al valor de referencia, normalizada como porcentaje, es decir, $\frac{\max_t(y(t)) - y_{ref}}{y_{ref}} \cdot 100$. Este parámetro permite identificar la presencia de oscilaciones iniciales indeseadas o posibles picos peligrosos para la operación segura del sistema [32].

Por su parte, el *error en estado estacionario* (e_{ss}) representa la diferencia entre la salida del sistema y la referencia una vez que el sistema ha alcanzado su régimen permanente. Se define como el límite del error conforme $t \rightarrow \infty$, dado por $e_{ss} = \lim_{t \rightarrow \infty} (y_{ref} - y(t))$, y permite evaluar la capacidad del sistema para mantener su salida en la referencia deseada una vez concluida la dinámica transitoria [32], [33].

La representación gráfica de estos tres parámetros puede observarse en la Figura 3.4, donde se muestra la respuesta típica de un sistema ante una entrada escalón unitario.

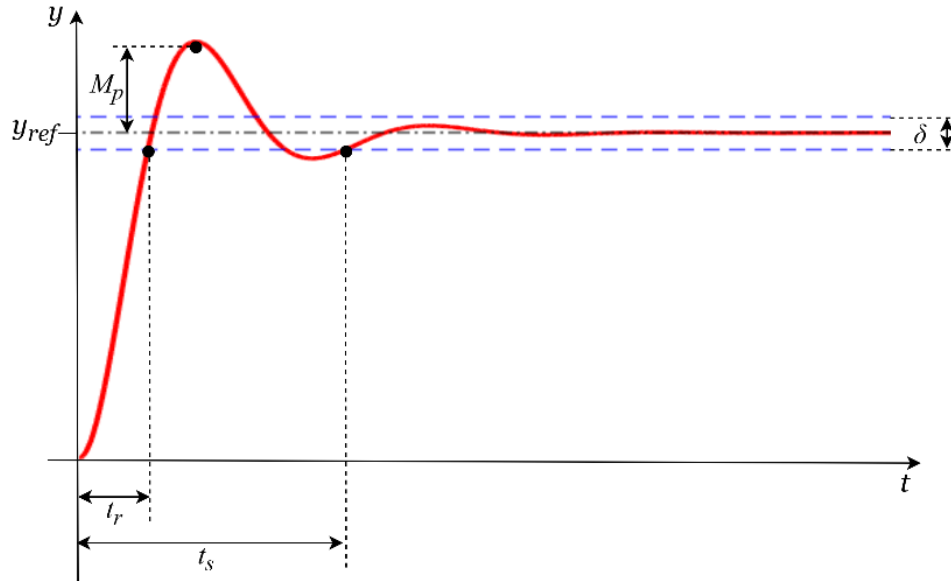


Figura 3.4 Respuesta típica de un sistema y sus criterios de desempeño temporal. (Adaptado de [33]).

3.2.3.2 Índices de error

Además de los indicadores puntuales anteriores, existen índices integrales que cuantifican el desempeño considerando simultáneamente la magnitud y la duración del error. Estos criterios permiten evaluar la respuesta global del sistema a lo largo del tiempo y son ampliamente utilizados como funciones objetivo en procesos de optimización. De acuerdo con [11], estos criterios son:

El Error Absoluto Integral (*IAE, por sus siglas en inglés*) caracteriza el error de manera lineal, otorgando igual importancia a todos los instantes de tiempo, sin distinguir entre errores tempranos o tardíos. Es útil en sistemas donde se busca un buen desempeño promedio, sin enfatizar excesivamente los transitorios. Se define como en la expresión 3.13.

$$IAE = \int_0^T |e(t)| dt \quad 3.13$$

El Error Cuadrático Integral (*ISE, por sus siglas en inglés*) caracteriza con mayor severidad los errores grandes, ya que estos se elevan al cuadrado. Es recomendable cuando se desea minimizar oscilaciones significativas o comportamientos extremos no deseados. Se expresa como,

$$ISE = \int_0^T e^2(t) dt \quad 3.14$$

El Error Absoluto Ponderado por el Tiempo (*ITAE*, por sus siglas en inglés), definido en la expresión 3.15, introduce un factor de caracterización creciente en el tiempo. Este índice prioriza la reducción del error en etapas tardías del proceso, lo cual es deseable cuando se busca una buena precisión en régimen permanente, aún a costa de transitorios iniciales más agresivos.

$$ITAE = \int_0^T t \cdot |e(t)| dt \quad 3.15$$

El Error Cuadrático Ponderado por el Tiempo (*ITSE*, por sus siglas en inglés) se define como en 3.16, y al igual que el *ITAE*, este índice caracteriza errores tardíos, pero también amplifica el efecto de errores grandes. Su uso es adecuado cuando se desea lograr una respuesta suave y precisa, con transitorios controlados y errores residuales mínimos.

$$ITSE = \int_0^T t \cdot e^2(t) dt \quad 3.16$$

En todos los casos anteriores, la variable e representa el error en el instante t , calculado como la diferencia entre la referencia y la salida del sistema, como se muestra en la expresión 3.17.

$$e(t) = y_{ref}(t) - y(t) \quad 3.17$$

3.3 Heurísticas de Inteligencia de Enjambres

En el diseño de sistemas de control óptimo, la sintonización de sus parámetros, particularmente las matrices de ponderación del LQR, es fundamental en el desempeño del sistema. Si bien el método clásico de ajuste se basa en estrategias empíricas, este enfoque no siempre garantiza una configuración óptima, especialmente cuando se trata de sistemas complejos. La naturaleza multivariable del helicóptero pivotado y el acoplamiento de sus ejes hacen que el ajuste tradicional de estas matrices resulte ineficiente, propensa a errores y con frecuencia subóptima.

Como alternativa a los métodos clásicos, han surgido técnicas de optimización computacional que permiten automatizar este proceso, explorando el espacio de soluciones posibles para encontrar aquellas combinaciones de Q y R que optimizan un conjunto de indicadores de desempeño predefinidos. Dentro de este enfoque, los algoritmos bioinspirados, y en particular aquellos basados en Inteligencia de Enjambres, se han consolidado como herramientas eficientes.

3.3.1 Principios bioinspirados

Los algoritmos de Inteligencia de Enjambres son técnicas de optimización estocástica inspiradas en los comportamientos colectivos observados en la naturaleza, como las migraciones de aves, los cardúmenes de peces, las colonias de insectos o las estrategias de caza cooperativa en mamíferos o especies marinas. Estos métodos parten del principio de que agentes simples, actuando localmente bajo reglas de interacción, pueden generar soluciones globales eficientes mediante cooperación, adaptación y exploración del entorno. Este principio ha sido trasladado al ámbito computacional mediante algoritmos que simulan la dinámica de interacción entre los múltiples agentes de un grupo [34].

En los algoritmos de Inteligencia de Enjambres, cada agente representa una posible solución al problema de optimización, y el conjunto de agentes conforma una población o enjambre. La calidad de cada solución se evalúa mediante una función objetivo, típicamente definida en términos de uno o varios criterios de desempeño del sistema controlado, como los indicadores descritos en la sección 3.2.3. Durante el proceso iterativo, los agentes exploran el espacio de búsqueda ajustando sus posiciones y velocidades basándose en información propia (memoria individual) y compartida (mejor solución colectiva) [35].

En este trabajo se consideran algoritmos de optimización fundamentadas en Inteligencia de Enjambres, en especial, estrategias están basadas en el comportamiento colectivo de animales marinos, tales como, Optimización de la Ballena Beluga (*Beluga Whale Optimization, BWO*), Optimizador de Tiburón Blanco (*White Shark Optimizer, WSO*) Optimizador de Pez Vela (*SailFish Optimizer, SFO*), Algoritmo de Optimización de Ballenas (*Whale Optimization Algorithm, WOA*) y Algoritmo de Depredación de Orcas (*Orca Predation Algorithm, OPA*).

Además, se suma el algoritmo de Optimización por Enjambre de Partículas (*Particle Swarm Optimization, PSO*), considerado uno de los algoritmos pioneros en este campo, el cual constituye una de las bases conceptuales de las heurísticas de enjambre. En las secciones siguientes se presenta la formulación matemática, estructura poblacional, estrategia de actualización y adaptaciones particulares de cada una de estas heurísticas.

3.3.2 Particle Swarm Optimization

El algoritmo de Optimización por Enjambre de Partículas (*PSO*, por sus siglas en inglés) es una técnica de optimización poblacional propuesta por Kennedy y Eberhart en 1995, basada en el modelado matemático del comportamiento colectivo observado en bandadas de aves y bancos de peces. Su objetivo es encontrar la mejor solución posible dentro de un espacio de búsqueda multidimensional mediante la cooperación de múltiples agentes que comparten información durante el proceso [36]. El diagrama de flujo de la Figura 3.5 muestra las principales etapas del algoritmo.

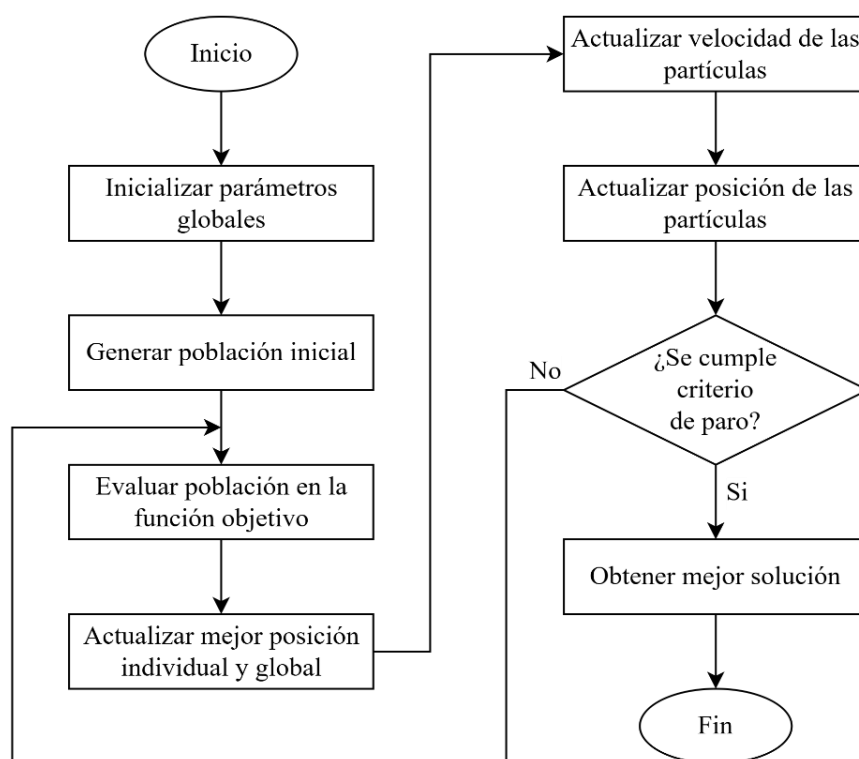


Figura 3.5 Diagrama de flujo del algoritmo PSO.
(Adaptado de [37]).

El algoritmo comienza con la generación de una población inicial X , compuesta por N partículas, donde cada i -ésima partícula posee una posición inicial $x_i(t) \in \mathbb{R}^n$, correspondiente a la iteración $t = 0$, generada aleatoriamente dentro de un dominio de búsqueda definido por límites inferiores x^{min} y superiores x^{max} [36], de modo que,

$$x_i(0) \in [x^{min}, x^{max}], \forall i \in \{1, 2, \dots, N\} \quad 3.18$$

Simultáneamente, se asigna a cada partícula una velocidad inicial $v_i(0) \in \mathbb{R}^n$, también generada de manera aleatoria dentro de un rango predefinido. Cada partícula evalúa su desempeño inicial mediante una función objetivo $f(x)$, registrando su mejor posición histórica p_i y comparándola con la mejor posición global del enjambre g . En cada iteración t , la evolución dinámica del enjambre se rige por un sistema de ecuaciones que actualiza las velocidades y posiciones de cada partícula [37]. La velocidad $v_i(t) \in \mathbb{R}^n$ se actualiza de acuerdo con la expresión 3.19, mientras que la ecuación 3.20 permite calcular la nueva posición de la partícula.

$$v_i(t + 1) = wv_i(t) + c_1r_1(t)[p_i - x_i(t)] + c_2r_2(t)[g - x_i(t)] \quad 3.19$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad 3.20$$

Donde, w es el factor de inercia, que modula la persistencia del movimiento previo; c_1, c_2 son los coeficientes de aceleración cognitivo y social, respectivamente y $r_1, r_2 \in [0, 1]$ son vectores aleatorios generados en cada iteración que introducen variabilidad estocástica. Estas iteraciones continúan hasta que se cumple un criterio de paro, típicamente un número máximo de iteraciones T_{max} [36].

3.3.3 Whale Optimization Algorithm

El Algoritmo de Optimización de Ballenas, conocido como Whale Optimization Algorithm (WOA), es una heurística estocástica desarrollada por Seyedali Mirjalili y Andrew Lewis en 2016. Este método se inspira en la estrategia de alimentación de las ballenas jorobadas, particularmente en el comportamiento conocido como bubble-net hunting (caza con red de burbujas), mediante el cual estos cetáceos generan espirales de burbujas para rodear y capturar a sus presas [38], como se ilustra en la Figura 3.6. Este fenómeno se modela

computacionalmente mediante tres fases principales: el encerramiento de la presa, el ataque en espiral, y la exploración aleatoria del entorno.

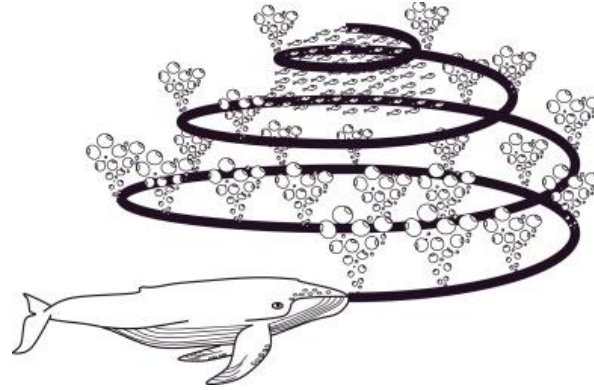


Figura 3.6 Caza por red de burbujas de una ballena jorobada. (Adaptado de [38]).

El algoritmo comienza generando una población inicial de i agentes de búsqueda, cada uno de los cuales representa una solución candidata al problema de optimización. Esta población se denota como se expresa en la ecuación 3.21, donde $x_{i,j}$ representa la posición del agente i -ésimo en la dimensión j , y n es la dimensionalidad del espacio de búsqueda. Cada $X_i(t) \in \mathbb{R}^n$, corresponde al vector de posición de un agente de búsqueda en la iteración t . Esta población inicial se genera aleatoriamente dentro de los límites inferiores y superiores del dominio del problema x^{min} y x^{max} [39].

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{in} \end{bmatrix} \quad 3.21$$

Posteriormente, cada agente X_i es evaluado mediante una función objetivo $f(X_i)$, y se selecciona la mejor solución global conocida hasta el instante (t) actual, denotada como $X^*(t)$. En la primera fase del algoritmo, que modela el comportamiento de encerramiento de la presa, las ballenas intentan acercarse al mejor agente de la población $X^*(t)$ [39]. Esta fase se basa en el cálculo de un vector de distancia expresado en la ecuación 3.22 y una actualización proporcional a dicha distancia, dada por la expresión 3.23 .

$$D_i(t) = |C \cdot X^*(t) - X_i(t)| \quad 3.22$$

$$X_i(t + 1) = X^*(t) - A \cdot D_i(t) \quad 3.23$$

donde, los coeficientes A y C se definen a partir de un vector aleatorio $r \in [0, 1]$, según las expresiones 3.24 y 3.25.

$$A = 2a \cdot r - a \quad 3.24$$

$$C = 2 \cdot r \quad 3.25$$

El parámetro vectorial a decrece linealmente en el tiempo desde 2 hasta 0, para permitir una transición controlada entre exploración y explotación [38], lo cual se modela como en 3.26.

$$a(t) = 2 \cdot \left(1 - \frac{t}{t_{max}}\right) \quad 3.26$$

Una vez activado el mecanismo de aproximación, el algoritmo incorpora una segunda fase denominada ataque en espiral, en la cual los agentes simulan una trayectoria logarítmica helicoidal alrededor de la presa. Esta actualización está dada por la expresión 3.27, donde b es una constante para definir la forma de la espiral y $l \in [-1, 1]$ es un escalar aleatorio [38].

$$X_i(t + 1) = D_i(t) \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \quad 3.27$$

Las ballenas jorobadas nadan alrededor de la presa dentro de un círculo en contracción y a lo largo de una trayectoria de translación espiral hacia ella. Para modelar computacionalmente este comportamiento dual, se recurre a un enfoque estocástico mediante una variable aleatoria p , la cual asigna una probabilidad del 50 % a la elección entre el mecanismo de reducción radial (movimiento circular de radio decreciente) y el patrón espiral, determinando con ello la forma en que se actualiza la posición de cada ballena [39]. Así, el comportamiento del agente se define de forma condicional como en la expresión 3.28.

$$X_i(t + 1) = \begin{cases} X^*(t) - A \cdot D_i(t); & p < 0.5 \\ D_i(t) \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t); & p \geq 0.5 \end{cases} \quad 3.28$$

Además de los mecanismos de acercamiento y ataque en espiral, el algoritmo WOA incorpora una tercera fase correspondiente al proceso de exploración aleatoria del espacio de búsqueda. Esta fase se activa cuando el módulo del vector de coeficiente de dirección A es mayor o igual a 1, es decir, cuando $|A| \geq 1$, lo cual indica que el agente debe buscar fuera de la región de explotación local. Cada agente actualiza su posición con base en la ubicación

de otro individuo seleccionado aleatoriamente de la población, denotado como X_{rand} [40]. El vector de distancia se redefine como en 3.29 y la posición del agente de búsqueda se ajusta conforme a la ecuación 3.30.

$$D_i(t) = |C \cdot X_{rand}(t) - X_i(t)| \quad 3.29$$

$$X_i(t + 1) = X_{rand}(t) - A \cdot D_i(t) \quad 3.30$$

La Figura 3.7 presenta el diagrama de flujo del WOA, en el cual se esquematizan las principales fases del proceso de optimización bioinspirada antes descritas. Estas fases se repiten durante un número determinado de iteraciones t_{max} . Al finalizar, la mejor posición $X^*(t_{max})$ considera la solución óptima estimada del problema [41].

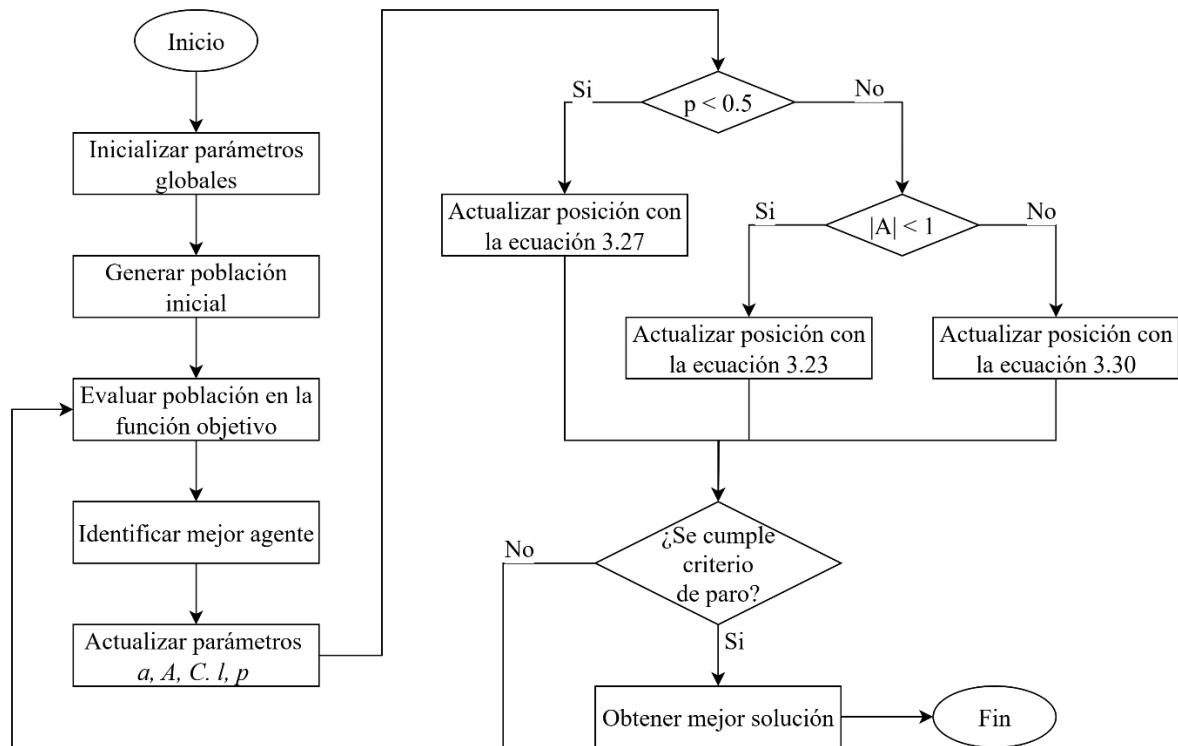


Figura 3.7 Diagrama de flujo del algoritmo WOA. (Adaptado de [42]).

3.3.4 White Shark Optimizer

El algoritmo White Shark Optimizer (WSO) es una heurística bioinspirada en el comportamiento de caza del tiburón blanco, desarrollada para resolver problemas de optimización numérica. Su diseño se fundamenta particularmente en las estrategias de exploración y explotación apoyadas en sus sentidos de audición y olfato altamente

desarrollados de estos depredadores marinos. Este algoritmo modela tres patrones principales de búsqueda: el movimiento hacia la presa a partir de ondas percibidas (tracking), la búsqueda aleatoria (exploración) y la cooperación mediante comportamiento escolar o de cardumen (fish school behavior), todo lo cual permite una actualización eficiente y adaptativa de la población hacia soluciones óptimas [43].

El WSO inicia generando una población de soluciones candidatas, modeladas como una matriz de dimensión $n \times d$, donde n es el número de tiburones blancos (agentes de búsqueda) y d es la cantidad de variables de decisión del problema [43]. Esta matriz se define como en la ecuación 3.31.

$$W = \begin{bmatrix} w_1^1 & w_1^2 & \cdots & w_1^d \\ w_2^1 & w_2^2 & \cdots & w_2^d \\ \vdots & \vdots & \ddots & \vdots \\ w_n^1 & w_n^2 & \cdots & w_n^d \end{bmatrix} \quad 3.31$$

Cada componente w_i^j se inicializa aleatoriamente dentro de los límites del dominio de búsqueda mediante la expresión 3.32, donde l_j y u_j son los límites inferior y superior, respectivamente, para cada dimensión, y $r \in [0, 1]$ es un número aleatorio generado con distribución uniforme, dando lugar a la posición w_i de cada tiburón. La calidad de cada solución candidata para cada nueva posición de un tiburón blanco w_i se evalúa en una función objetivo $f(w_i)$, permitiendo identificar las mejores posiciones [44].

$$w_i^j = l_j + r \cdot (u_j - l_j), \quad \forall i \in \{1, \dots, n\}, \quad j \in \{1, \dots, d\} \quad 3.32$$

Una vez identificadas las presas más prometedoras, los tiburones ajustan su ubicación en función de la mejor posición global conocida w_{gbest} y la mejor posición individual $w_{v_i}^{best}$. La velocidad de este movimiento se calcula mediante la ecuación 3.33; donde $\mu = \frac{2}{|2 - \tau - \sqrt{\tau^2 - 4\tau}|}$; $\tau = 4.125$, es un parámetro de aceleración que puede ser ajustado empíricamente, $c_1, c_2 \in [0, 1]$ son coeficientes aleatorios y p_1, p_2 son parámetros que modulan la influencia de las soluciones líderes w_{gbest} y $w_{v_i}^{best}$, se calculan como en 3.34 y 3.35. K y k representan el número máximo y actual de iteraciones, respectivamente [43].

$$v_i^{k+1} = \mu \left[v_i^k + p_1 (w_{g_{best}}^k - w_i^k) \cdot c_1 + p_2 (w_{v_i^k}^{best} - w_i^k) \cdot c_2 \right] \quad 3.33$$

$$p_1 = p_{max} + (p_{max} - p_{min}) \cdot e^{-\left(\frac{4k}{K}\right)^2}; \quad \begin{cases} p_{min} = 0.5 \\ p_{max} = 1.5 \end{cases} \quad 3.34$$

$$p_2 = p_{min} + (p_{max} - p_{min}) \cdot e^{-\left(\frac{4k}{K}\right)^2}; \quad \begin{cases} p_{min} = 0.5 \\ p_{max} = 1.5 \end{cases} \quad 3.35$$

Una vez actualizada la velocidad de cada agente de búsqueda, el algoritmo procede a modificar su posición en el espacio de búsqueda. Esta modificación está inspirada en el comportamiento adaptativo de los tiburones blancos durante la persecución activa de presas, en la cual alternan entre trayectorias ondulantes y aproximaciones directas, dependiendo del entorno y la distancia a su objetivo [43]. Para capturar esta dinámica, el WSO implementa una regla de decisión estocástica basada en un umbral de movimiento ondulatorio mv , denotado como en 3.36.

$$mv = \frac{1}{a_0 + e^{\frac{K}{2-k}/a_1}}; \quad a_0 = 6.25, \quad a_1 = 100 \quad 3.36$$

Si un número aleatorio generado $rand \in [0, 1]$ cumple que $rand < mv$, se activa la estrategia de actualización no lineal, diseñada para simular trayectorias erráticas e impredecibles, como las que adopta un tiburón al detectar cambios repentinos en la trayectoria de su presa. Por otro lado, si el valor aleatorio no satisface la condición anterior, es decir, $rand \geq mv$, el algoritmo adopta una estrategia de avance ondulatorio directo basada en la velocidad previamente calculada [44]. Este comportamiento se describe mediante las expresiones 3.37 considerando sus dos intervalos.

$$w_i^{k+1} = \begin{cases} w_i^k \ominus \oplus w_0 + u \cdot a + l \cdot b; & rand < mv \\ w_i^k + \frac{v_i^k}{f}; & rand \geq mv \end{cases} \quad 3.37$$

Donde, w_i^k es la posición actual del tiburón i , $a = sgn(w_i^k - u) > 0$ y $b = sgn(w_i^k - l) < 0$ son vectores que delimitan el rango de búsqueda a partir de los valores superiores u e inferiores l , $w_0 = \oplus (a, b)$ denota un vector que representa la posición

óptima modificada mediante una operación lógica XOR y f representa la frecuencia del movimiento ondulante de un tiburón definida como en ecuación 3.38 [44].

$$f = f_{min} + \frac{f_{max} - f_{min}}{f_{max} + f_{min}} ; \quad f_{max} = 0.75, \quad f_{min} = 0.07 \quad 3.38$$

Adicionalmente, el WSO modela el uso de los órganos sensoriales especializados del tiburón blanco, como el sistema olfativo, que le permiten detectar con precisión la posición de su presa y de otros individuos cercanos. Este mecanismo simula el momento en que el tiburón detecta la estela de una presa en movimiento y ajusta su trayectoria en función de información sensorial. Para incorporar este principio biológico, se introduce una regla de actualización condicionada que desplaza al tiburón hacia la mejor solución global hallada w_{gbest} , aplicando una dirección estocástica determinada por la ecuación 3.39 [44].

$$w_i^{k+1} = w_{gbest}^k + r_1 \cdot \vec{D}_w \cdot sgn(r_2 - 0.5); \quad r_3 < s_s \quad 3.39$$

Donde $r_1, r_2, r_3 \in [0, 1]$ son valores aleatorios independientes, $s_s = \left| 1 - e^{-\frac{0.0005k}{K}} \right|$ es el umbral de sensibilidad sensorial, y $\vec{D}_w = |rand \cdot (w_{gbest} - w_i^k)|$ es la distancia vectorial entre el tiburón actual y el mejor agente conocido [44].

Por último, se incorpora una fase cooperativa conocida como comportamiento de cardumen (*fish school behavior*), que emula la tendencia natural de los tiburones a reagruparse, favoreciendo el intercambio de información entre individuos, especialmente cuando la presa se ha alejado. Este comportamiento se modela mediante la siguiente expresión 3.40; donde \acute{w}_i^{k+1} es la posición provisional obtenida mediante uno de los operadores definidos en las ecuaciones 3.37 o 3.39 y el denominador introduce una modulación aleatoria que ajusta la magnitud del movimiento [44].

$$w_i^{k+1} = \frac{w_i^k + \acute{w}_i^{k+1}}{2 \cdot rand} \quad 3.40$$

Todo el proceso descrito anteriormente se ejecuta de forma iterativa durante K iteraciones. En cada iteración, los agentes de búsqueda exploran y explotan el espacio de soluciones, adaptando su comportamiento según condiciones estocásticas y el desempeño histórico de la

población [45]. La Figura 3.8 resume gráficamente la estructura operativa del algoritmo WSO mediante un diagrama de flujo.

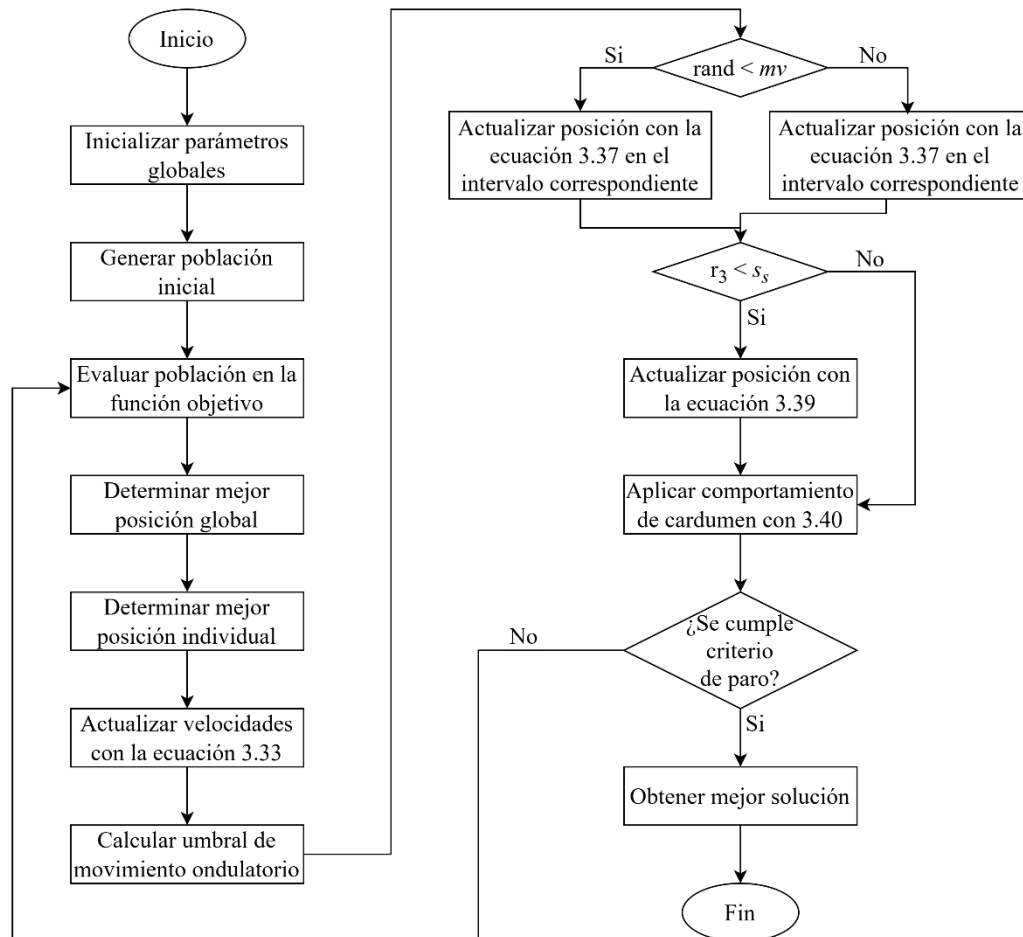


Figura 3.8 Diagrama de flujo del algoritmo WSO. (Adaptado de [45]).

3.3.5 SailFish Optimizer

El algoritmo Sailfish Optimizer (SFO) es una heurística de inteligencia de enjambre propuesta por Shadravan, Naji y Bardsiri en 2019. Este algoritmo está inspirado en la estrategia de caza grupal del pez vela. Su comportamiento colectivo durante la caza de cardúmenes de sardinas se distingue por la alternancia de ataques coordinados entre los individuos, lo que incrementa la probabilidad de captura al desestabilizar las estrategias de defensa colectiva características de sus presas [46]. Esta dinámica es modelada matemáticamente en el SFO mediante la interacción de dos poblaciones: los depredadores (sailfish) y las presas (sardinas), lo que permite enriquecer la exploración del espacio de

búsqueda al recorrer regiones diversas del espacio de soluciones posibles, evitando que el proceso de optimización se estanque en soluciones subóptimas [47].

El SFO comienza con la generación de las dos poblaciones distintas. La población de sailfish (peces vela) representa las soluciones candidatas y se modela como la matriz de la ecuación 3.41, donde m es el número de individuos y d la dimensión del problema. Cada vector $SF_i = (SF_{i,1}, SF_{i,2}, \dots, SF_{i,d})$ denota una posición en el espacio de búsqueda definido por límites inferiores L_j y superiores U_j en cada dimensión. De forma análoga, la población de n sardinas se representa mediante la matriz de la ecuación 3.42, donde cada fila corresponde a la posición de una sardina en el espacio de búsqueda [46].

$$SF = \begin{bmatrix} SF_{1,1} & SF_{1,2} & \cdots & SF_{1,d} \\ SF_{2,1} & SF_{2,2} & \cdots & SF_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ SF_{m,1} & SF_{m,2} & \cdots & SF_{m,d} \end{bmatrix} \quad 3.41$$

$$S = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,d} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n,1} & S_{n,2} & \cdots & S_{n,d} \end{bmatrix} \quad 3.42$$

La aptitud de cada individuo se evalúa mediante una función objetivo f [24], cuyos valores se almacenan en los vectores de aptitud definidos en 3.43 y 3.44.

$$SF_{Fitness} = \begin{bmatrix} f(SF_1) \\ f(SF_2) \\ \vdots \\ f(SF_m) \end{bmatrix} \quad 3.43$$

$$S_{Fitness} = \begin{bmatrix} f(S_1) \\ f(S_2) \\ \vdots \\ f(S_n) \end{bmatrix} \quad 3.44$$

Durante cada iteración, se determina el *sailfish elite* X_{SF}^{elite} como la mejor solución conocida hasta el momento, y la *sardina herida* X_S^{inj} , que representa la mejor presa capturada. La actualización de la posición de cada sailfish X_{SF} se realiza conforme a una estrategia de alternancia de ataque, definida por la ecuación 3.45. Donde $rand \in [0, 1]$ es una variable aleatoria y λ_i es un coeficiente dinámico que regula la distancia y dirección del

desplazamiento. El parámetro λ_i depende de la densidad de presas PD y se calcula como en 3.46. Siendo N_{SF} el número de sailfish y N_S el número de sardinas en la iteración actual [24].

$$X_{SF}^{new} = X_{SF}^{elite} - \lambda_i \left[rand \cdot \left(\frac{X_{SF}^{elite} + X_S^{inj}}{2} \right) - X_{SF}^{old} \right] \quad 3.45$$

$$\lambda_i = 2 \cdot rand \cdot PD - PD; \quad PD = 1 - \left(\frac{N_{SF}}{N_{SF} + N_S} \right) \quad 3.46$$

Las sardinas también actualizan su posición en respuesta a los ataques, empleando una estrategia de evasión adaptativa basada en dos parámetros: la intensidad del ataque AP y una variable aleatoria r . Esta dinámica se modela mediante la expresión 3.47, mientras que el poder del ataque del pez vela se calcula como en 3.48; donde itr es la iteración actual, A y ε son parámetros de control que reducen la agresividad del ataque conforme progresa la búsqueda [24].

$$X_S^{new} = rand \cdot (X_{SF}^{elite} - X_S^{old} + AP) \quad 3.47$$

$$AP = A \cdot (1 - 2 \cdot itr \cdot \varepsilon) \quad 3.48$$

En la etapa final de la caza, la sardina herida será capturada rápidamente. En el SFO, el proceso de caza se concreta cuando una sardina supera en aptitud a su correspondiente sailfish. En tal caso, la posición del sailfish se reemplaza por la de la sardina capturada como se denota en 3.49, esto garantiza la transmisión de soluciones prometedoras desde la población de presas hacia la de depredadores [24].

$$X_{SF} = X_S; \quad f(X_S) < f(X_{SF}) \quad 3.49$$

El proceso de optimización descrito se repite durante I iteraciones. La Figura 3.9 resume este procedimiento mediante un diagrama de flujo, el cual permite visualizar de forma estructurada la arquitectura iterativa del algoritmo y sus mecanismos clave de interacción biológica traducidos al dominio computacional [24].

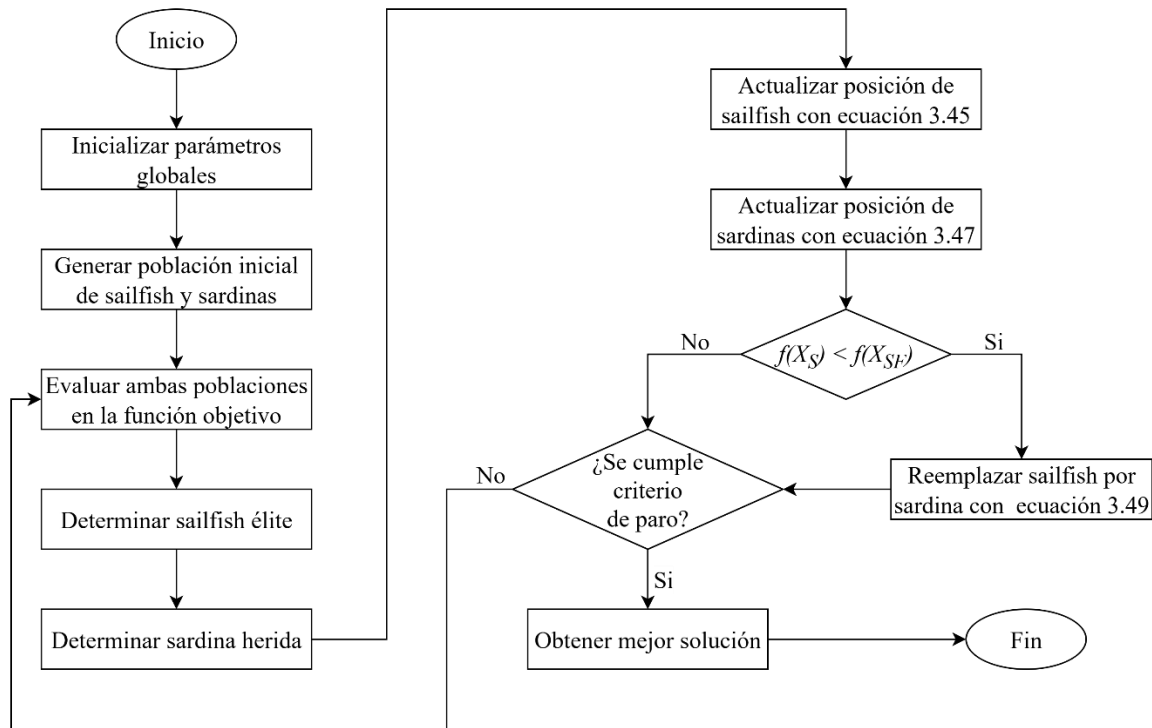


Figura 3.9 Diagrama de flujo del algoritmo SFO.
(Adaptado de [47]).

3.3.6 Beluga Whale Optimization

El algoritmo Beluga Whale Optimization (BWO) es una heurística de enjambres fue propuesta por Zhong et al. en 2022, está inspirada en el comportamiento de las ballenas beluga. Esta técnica modela tres patrones biológicos fundamentales: nado sincrónico (exploración), conducta de caza cooperativa (explotación) y el fenómeno conocido como caída de ballena (*whale fall*). Estas conductas se traducen computacionalmente en fases que permiten la exploración global de un espacio de búsqueda y la explotación local en regiones prometedoras [48].

El BWO inicia con la creación de una población representada por un conjunto de vectores de posición $X \in \mathbb{R}^{n \times d}$, donde n es el número de agentes (ballenas beluga) y d la dimensión del problema; esta matriz se define como en la expresión 3.50. Cada posición inicial $x_i \in \mathbb{R}^d$ se genera aleatoriamente en el dominio de búsqueda definido por los límites $[l_j, u_j]$ y se evalúa mediante una función objetivo $f(x_i)$ para obtener la aptitud de cada agente, el mejor agente conocido se identifica como x_{best} [49].

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \quad 3.50$$

A lo largo de las iteraciones, la decisión de activar la fase exploratoria o explotativa se toma con base en un factor de equilibrio B_f , calculado según la ecuación 3.51; donde T es la iteración actual, T_{max} el número máximo de iteraciones, y $B_0 \in [0, 1]$ es un valor aleatorio. La exploración se activa si $B_f > 0.5$; de lo contrario, se ejecuta la explotación [48].

$$B_f = B_0 \left(1 - \frac{T}{2T_{max}} \right) \quad 3.51$$

Cuando $B_f > 0.5$, se activa la fase de exploración, basada en el comportamiento de nado sincronizado de las belugas, específicamente el fenómeno de nado en pares. Esta estrategia permite actualizar las posiciones de los agentes de búsqueda a través de interacciones estocásticas entre individuos seleccionados aleatoriamente dentro de la población. Para cada agente i y dimensión j , la posición se actualiza como en la expresión 3.52; donde $x_{i,j}^{T+1}$ representa la nueva posición del agente i en la dimensión j en la iteración $T + 1$, x_{i,p_j}^T es la posición actual del agente i en una dimensión aleatoria p_j , x_{r,p_1}^T es la posición de un agente r seleccionado aleatoriamente en la dimensión p_1 y $r_1, r_2 \in [0, 1]$ son números aleatorios [49].

$$x_{i,j}^{T+1} = \begin{cases} x_{i,p_j}^T + (x_{r,p_1}^T - x_{i,p_j}^T) \cdot (1 + r_1) \cdot \sin(2\pi r_2), & \text{si } j \text{ es par} \\ x_{i,p_j}^T + (x_{r,p_1}^T - x_{i,p_j}^T) \cdot (1 + r_1) \cdot \cos(2\pi r_2), & \text{si } j \text{ es impar} \end{cases} \quad 3.52$$

En caso contrario, es decir, si $B_f \leq 0.5$, se activa la fase de explotación, inspirada en el comportamiento de caza cooperativa, que combina las trayectorias del individuo actual x_i^T , la mejor solución global x_{best}^T y un individuo aleatorio x_r^T , incorporando además una perturbación por vuelo de Lévy L_F . Esta dinámica se describe mediante la ecuación 3.53, donde $r_3, r_4 \in [0, 1]$ son parámetros aleatorios y $C_1 = 2 \cdot r_4 \cdot \left(1 - \frac{T}{T_{max}} \right)$ es el coeficiente de salto aleatorio que pondera la intensidad del vuelo de Levy, el cual es definido en las expresiones 3.54 y 3.55, donde u y v son variables aleatorias normalmente distribuidas y $\beta = 1.5$ [49].

$$x_i^{T+1} = r_3 \cdot x_{best}^T - r_4 x_i^T + C_1 \cdot L_F \cdot (x_r^T - x_i^T) \quad 3.53$$

$$L_F = 0.05 \cdot \frac{u \cdot \sigma}{|v|^{1/\beta}} \quad 3.54$$

$$\sigma = \left[\frac{\Gamma(1 + \beta) \cdot \sin(\pi\beta/2)}{\Gamma\left(\frac{1 + \beta}{2}\right) \cdot \beta \cdot 2^{(\beta-1)/2}} \right]^{1/\beta} \quad 3.55$$

Para incrementar la robustez del algoritmo y evitar convergencia prematura, el BWO incorpora una fase de dispersión inspirada en el proceso de descomposición de un cuerpo de ballena en el lecho marino. Esta etapa simula la pérdida de algunos agentes durante el proceso de optimización, permitiendo diversificación poblacional y la evasión de óptimos locales. Esta fase es activada aleatoriamente según una probabilidad decreciente W_f [48], definida en la ecuación 3.56.

$$W_f = 0.1 - 0.05 \cdot \frac{T}{T_{max}} \quad 3.56$$

Si un individuo x_i^T es seleccionado para esta fase (es decir, si $B_f < W_f$), su nueva posición se actualiza según la ecuación 3.57, donde se combinan de forma ponderada la posición actual x_i^T , una posición aleatoria x_r^T , y una perturbación adicional basada en un paso adaptativo x_{step} (ver ecuación 3.58), mientras que $r_5, r_6, r_7 \in [0, 1]$ son variables aleatorias uniformes [48].

$$x_i^{T+1} = r_5 x_i^T - r_6 x_r^T + r_7 x_{step} \quad 3.57$$

$$x_{step} = (u_b - l_b) \cdot e^{\frac{-C_2 T}{T_{max}}}; \quad C_2 = 2W_f \cdot n \quad 3.58$$

El algoritmo repite estas fases durante T_{max} iteraciones, y la secuencia completa se resume mediante el diagrama de flujo de la Figura 3.10, donde se muestran las operaciones de inicialización, evaluación, exploración, explotación y dispersión que estructuran el comportamiento dinámico del algoritmo BWO en su versión canónica [49].

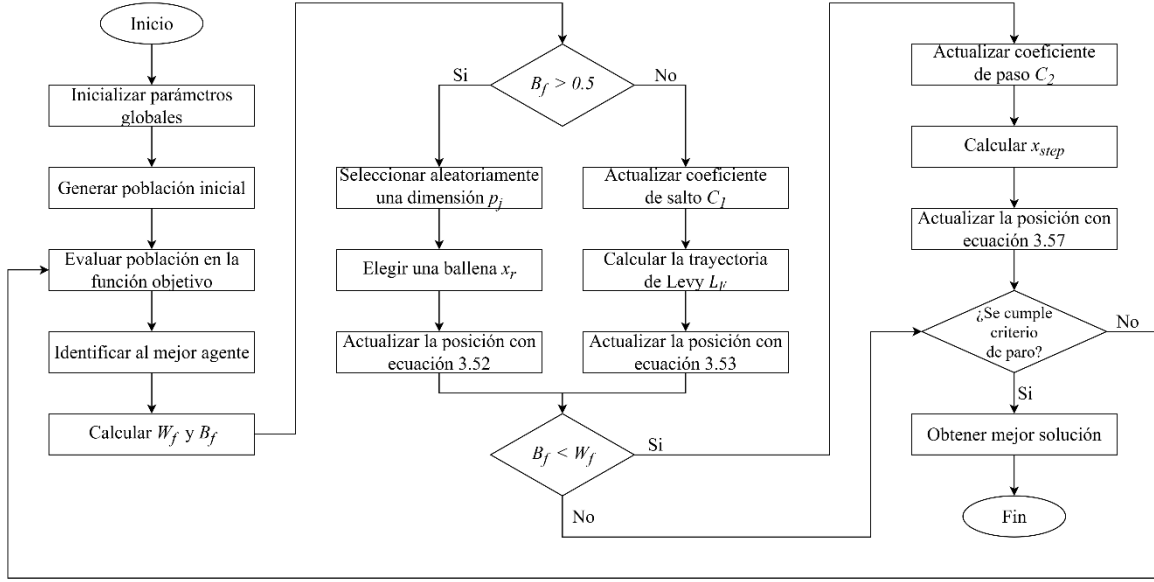


Figura 3.10 Diagrama de flujo del algoritmo BWO.
(Adaptado de [49]).

3.3.7 Orca Predation Algorithm

El Orca Predation Algorithm (OPA) es una metaheurística bioinspirada, desarrollada por Yuxin et al. en 2022, que emula el comportamiento de caza cooperativa de las orcas, conocidas por ser animales marinos altamente sociales que comparten presas y se comunican con su grupo para planificar tácticas de caza y alimentación. Este comportamiento colectivo ha sido modelado mediante tres fases secuenciales: persecución, ataque y evaluación adaptativa, las cuales se ejecutan iterativamente hasta alcanzar un criterio de paro [50].

Como toda heurística bioinspirada, el OPA comienza generando una población inicial $X \in \mathbb{R}^{n \times d}$ de n agentes de búsqueda (orcas) y d dimensiones del problema, donde cada individuo $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d}) \in \mathbb{R}^d$ representa una solución candidata al problema de optimización, de modo que, la población total se representa mediante la matriz de la expresión 3.59, donde cada posición $x_{i,j}$ se inicializa aleatoriamente dentro del dominio definido por los límites inferior l_j y superior u_j [51].

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \quad 3.59$$

Una vez generada la población inicial, el algoritmo inicia la fase de persecución, en la que cada agente decide entre dos comportamientos alternativos: conducción o encerramiento. La estrategia de conducción modela la tendencia de las orcas a guiar a sus presas hacia zonas desfavorables antes de ejecutar un ataque, mientras que el encerramiento simula cómo los individuos cercan a la presa mediante maniobras sincronizadas y envolventes. La selección entre ambas estrategias se controla mediante un parámetro probabilístico $p_1 \in [0,1]$, con $p_1 \geq 0.5$ para activar la conducción y $p_1 < 0.5$ para activar el encerramiento [50].

La persecución por conducción se implementa mediante un par de reglas de actualización (ver ecuaciones 3.60 y 3.61) que combinan la atracción hacia la mejor posición global x_{best} , la media de la población M y la posición actual del individuo x_i^t . Donde $a, b, d, e \in [0, 1]$ son coeficientes aleatorios, $c = 1 - b$ y $F = 2$ es un coeficiente de amplificación [51].

$$v_{1,i}^t = a \cdot [d \cdot x_{best}^t - F \cdot (b \cdot M^t + c \cdot x_i^t)] \quad 3.60$$

$$v_{2,i}^t = e \cdot (x_{best}^t - x_i^t) \quad 3.61$$

Dependiendo de un umbral probabilístico $q \in [0,1]$, se selecciona la trayectoria hacia la nueva posición en la iteración t [51], tal como se expresa en la ecuación 3.62.

$$x_{chase,i}^{t+1} = \begin{cases} x_i^t + v_{1,i}^t, & \text{si } r > q \\ x_i^t + v_{2,i}^t, & \text{si } r \leq q \end{cases} \quad 3.62$$

La maniobra de encierro en la que las posiciones se actualizan usando una combinación de tres orcas aleatorias, se calcula, para cada dimensión k , como en la expresión 3.63; donde $j_1, j_2, j_3 \in \{1, 2, \dots, n\}$ son índices distintos seleccionados aleatoriamente, y u es un coeficiente de amplitud decreciente [51], definido como en 3.64.

$$x_{chase,i,k}^{t+1} = x_{j_1,k}^t + u \cdot (x_{j_2,k}^t - x_{j_3,k}^t) \quad 3.63$$

$$u = 2 \cdot (rand - 0.5) \cdot \left(\frac{T_{max} - t}{T_{max}} \right) \quad 3.64$$

Una vez culminada la fase de persecución, y la orca ha alcanzado cercanía a la presa, se ejecuta un movimiento de ataque utilizando información de los individuos más exitosos de

la población y agentes aleatorios. Este proceso representa la coordinación social del ataque. La primera componente de velocidad se calcula como la desviación respecto al promedio de las cuatro mejores orcas, como se expresa en 3.65. La segunda componente, definida en la expresión 3.66, se basa en la interacción con tres orcas aleatorias [52].

$$v_{a1,i}^t = \frac{x_1^t + x_2^t + x_3^t + x_4^t}{4} - x_{chase,i}^t \quad 3.65$$

$$v_{a2,i}^t = \frac{x_{j_1}^t + x_{j_2}^t + x_{j_3}^t}{4} - x_i^t \quad 3.66$$

Finalmente, la posición del individuo se actualiza mediante una combinación lineal de ambas componentes, tal como lo muestra la ecuación 3.67. Donde $g_1 \in [0,2]$ y $g_2 \in [-2.5,2.5]$ son parámetros aleatorios que controlan la intensidad y dirección del ataque [52].

$$x_{attack,i}^t = x_{chase,i}^t + g_1 \cdot v_{a1,i}^t + g_2 \cdot v_{a2,i}^t \quad 3.67$$

Tras el ataque, se procede a la actualización adaptativa. Se evalúa si la nueva posición $x_{attack,i}^t$ resulta mejor respecto de la anterior $x_{chase,i}^t$, de ser así, se acepta como la nueva solución del individuo (ver ecuación 3.68) [52].

$$x_i^{t+1} = x_{attack,i}^t \quad \text{si } f(x_{attack,i}^t) < f(x_{chase,i}^t) \quad 3.68$$

En caso contrario, se activa un mecanismo de preservación adaptativa, que retiene la solución previa o realiza un ajuste compensado hacia el límite inferior del dominio (l_b), dependiendo de una probabilidad $p_2 \in [0,1]$ y una variable aleatoria r generada en $[0, 1]$ [50]. Este comportamiento se describe en la ecuación 3.69.

$$x_i^{t+1} = \begin{cases} x_{chase,i}^t & r \geq p_2 \\ l_b & r < p_2 \end{cases} \quad 3.69$$

El proceso completo del OPA se ejecuta de forma iterativa durante un número máximo de T iteraciones [50]. La lógica detallada del algoritmo se resume en el diagrama de flujo mostrado en la Figura 3.11, donde se visualiza de manera esquemática la evolución poblacional de los agentes y la interacción entre los operadores de comportamiento antes descritos.

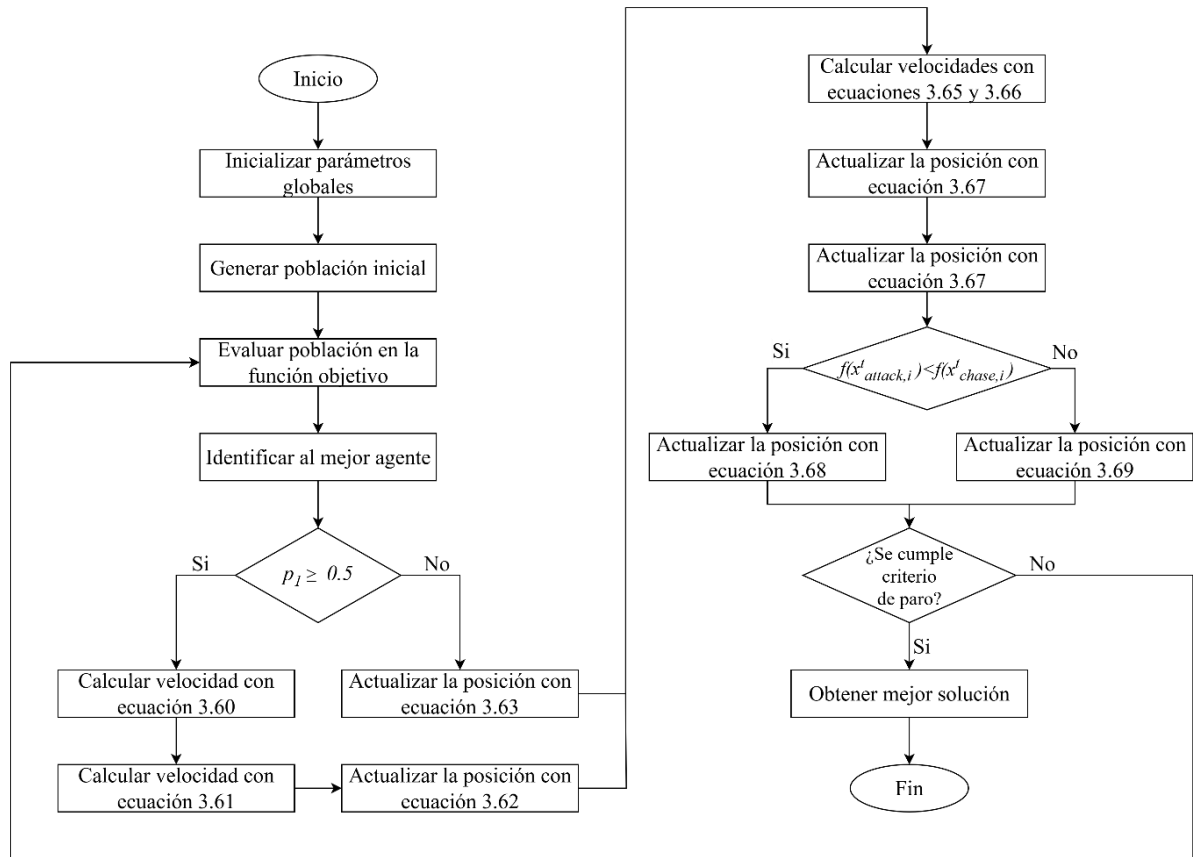


Figura 3.11 Diagrama de flujo del algoritmo OPA. (Adaptado de [52]).

3.4 Heurísticas reducidas

El uso de algoritmos metaheurísticos en problemas de optimización ha demostrado ser una alternativa eficaz frente a métodos deterministas clásicos. Sin embargo, la principal desventaja de muchas de estas técnicas radica en su alto costo computacional, derivado de la cantidad considerable de operaciones propias del algoritmo o del número evaluaciones de la función objetivo requeridas para alcanzar soluciones aceptables. Este problema se acentúa cuando los algoritmos se emplean en sistemas complejos, como la sintonización de parámetros del LQR. En este contexto, surge el enfoque de heurísticas reducidas, el cual busca preservar la capacidad exploratoria y la calidad de convergencia de los algoritmos bioinspirados originales, pero disminuyendo significativamente su carga computacional mediante mecanismos de simplificación.

3.4.1 Concepto y motivación

El término heurísticas reducidas se refiere a un conjunto de técnicas diseñadas para disminuir la carga computacional de los algoritmos metaheurísticos manteniendo, en lo posible, su eficacia en la búsqueda de soluciones óptimas. En contextos de optimización de problemas de alta complejidad computacional, los algoritmos metaheurísticos convencionales suelen volverse ineficientes o costosos debido al gran número de evaluaciones y operaciones requeridas. Ante este reto, la motivación central de las heurísticas reducidas es mantener un buen desempeño reduciendo los recursos utilizados [53].

Su origen puede rastrearse hasta los primeros trabajos en poblaciones dinámicas a inicios de la década de 1990, cuando autores como David Goldberg y Krishnakumar propusieron variar el tamaño de la población durante la ejecución para reducir el costo computacional sin degradar significativamente la calidad de las soluciones [54]. Con el tiempo, esta idea dio lugar a una amplia variedad de estrategias específicas que van desde reducir explícitamente la población de forma determinística, hasta ajustar dinámicamente parámetros y operadores; todas ellas englobadas bajo el concepto de heurísticas reducidas, por su objetivo común de aliviar la complejidad computacional de metaheurísticas canónicas [53], [55].

A través del tiempo, diferentes estudios [56-60] respaldan la efectividad de estas alternativas, puesto que, se ha demostrado que algoritmos con mecanismos de reducción bien diseñados pueden alcanzar resultados comparables a los de metaheurísticas canónicas, pero con ahorros computacionales significativos.

3.4.2 Estrategias de reducción

En la literatura se han documentado diversas estrategias de reducción dentro de distintas familias de metaheurísticas. En la Tabla 3.2 se describen algunas de las principales estrategias propuestas.

Tabla 3.2 Estrategias de reducción en metaheurísticas.
(Elaboración propia).

Estrategia	Descripción técnica	Objetivo
Reducción poblacional programada	Disminución controlada del número de agentes durante la ejecución, siguiendo un esquema de reducción predefinido, normalmente de forma lineal [61].	Optimizar el uso de recursos de evaluación en etapas donde la exploración amplia deja de ser eficiente [61], [62].
Tamaño poblacional adaptativo y poda de agentes	Ajuste dinámico de la población en función del rendimiento de los individuos, eliminando agentes de bajo desempeño y añadiendo nuevos cuando es necesario [63].	Mantener un equilibrio adaptativo entre exploración y explotación, evitando gasto innecesario de evaluaciones [63], [64].
Micro-poblaciones y reinicios periódicos	Uso deliberado de poblaciones muy pequeñas, combinadas con reinicios planificados para restaurar la diversidad [65], [66].	Concentrar recursos en zonas prometedoras y reactivar la búsqueda para evitar convergencia prematura [66], [67].
Elitismo selectivo y control de diversidad	Conservación de un subconjunto reducido de soluciones élite, seleccionadas por calidad y añadiendo nuevas para aportar diversidad [68].	Evitar evaluaciones redundantes y mantener diversidad poblacional [68], [69].
Ajuste dinámico de parámetros	Modificación de parámetros como tasas de mutación, factores de recombinación o pesos en respuesta al desempeño del algoritmo [70], [71].	Aumentar la eficiencia y la capacidad de adaptación a lo largo de la búsqueda [70], [71].
Selección adaptativa de operadores	Priorización del uso de operadores con mayor impacto, reduciendo la aplicación de operadores menos efectivos [72], [73], [74].	Maximizar el rendimiento global reduciendo el uso de recursos en procesos poco productivos [72], [73], [74].
Criterios de terminación adaptativos	Finalización anticipada de la búsqueda cuando los indicadores de mejora o diversidad muestran estancamiento [75], [76].	Evitar iteraciones sin beneficio y reducir tiempo total de cómputo [75], [76].

4 ESTADO DEL ARTE

En esta sección se identifican dos ejes que sustentan el desarrollo de la presente investigación. El primero se refiere a los métodos de control empleados en sistemas dinámicos con características similares a un helicóptero pivotado, incluyendo tanto enfoques clásicos, modernos y control inteligente. el segundo abarca la sintonización de controladores mediante técnicas de Inteligencia Computacional, enfocándose en la optimización de parámetros de distintos controladores a través de algoritmos heurísticos, con el objetivo de mejorar el desempeño del sistema controlado.

4.1 Métodos de control de posición de un helicóptero pivotado

En el campo de control óptimo, en [77] regulan la posición de los ángulos de cabeceo, balanceo y guiñada de un modelo simulado de un cuadricóptero de Quanser, denominado Qball-X4. Para ello, utilizan como técnica de control un LQR implementado en MATLAB®/Simulink. En consecuencia, el cuadricóptero alcanzó la posición deseada de 2 metros en aproximadamente 6 segundos. En el caso del control de los ángulos de cabeceo y balanceo, el sistema logra estabilizarse en 5 segundos para un ángulo de referencia de 0.5 radianes.

Por otra parte, en [78] se presenta el desarrollo de un controlador óptimo LQR para estabilizar un helicóptero con 3-DOF, bajo la presencia de incertidumbres estructuradas (como variaciones de parámetros) y no estructuradas (como perturbaciones externas). Al comparar el rendimiento del LQR con un controlador clásico PID, los resultados reportan que el controlador LQR es más eficiente, reduciendo el error de seguimiento de 0.24 rad a 0.06 rads y disminuyendo el esfuerzo de control de $1.46 \times 10^4 \text{ rad}$ a $1.18 \times 10^4 \text{ rad}$.

Así mismo, en [79], se desarrolla un LQR mejorado por un Filtro Kalman No Lineal (UKF, Nonlinear Kalman Filter, por sus siglas en inglés) para un modelo de helicóptero de 4 rotores y 6-DOF. Las simulaciones mostraron que el sistema seguía de manera precisa las trayectorias deseadas, incluyendo el seguimiento de velocidades y ángulos de rotación. El filtro UKF fue fundamental para estimar con rapidez los errores de modelo, permitiendo que el sistema se adaptara eficientemente a las perturbaciones.

En [80] se presenta el diseño de un Regulador Lineal Cuadrático con acción Integral (LQR-I) para un helicóptero de 2-DOF de la marca Quanser. Los resultados experimentales muestran que el sistema logró estabilidad en 12 segundos y un buen desempeño en la respuesta del sistema al escalón de referencia, aunque se puede apreciar la presencia de sobreimpulsos y algunas oscilaciones.

Finalmente, en el mismo ámbito de control óptimo, en [81] se presenta un controlador Integral Cuadrático Lineal (LQI) en conjunto con un marco de control adaptativo para un helicóptero de 2-DOF. Los resultados, producto de simulaciones y experimentos, demuestran que el esquema de control propuesto mejorará de manera significativa la respuesta del helicóptero tanto ante señales de referencia tipo escalón como frente a perturbaciones exógenas con errores en estado estacionario menores al 5%.

Por otro lado, en el área de control robusto, Falguni Gopmandal y Arun Ghosh en [41], abordan el diseño de un controlador Proporcional Integral Derivativo (PID) de tipo MIMO (Multiple Input, Multiple Output, por sus siglas en inglés), basado en un LQR aplicado a un helicóptero de dos grados de libertad (2-DOF). A partir de simulaciones demuestran que el controlador propuesto mejora la estabilidad de la planta significativamente, en términos de la función de sensibilidad, la función de sensibilidad complementaria y la función de sensibilidad de control, en comparación con el control nominal LQR.

Así mismo, en [82], se propone un nuevo enfoque de Control de Modo Deslizante continuo (SMC, Sliding Mode Control, por sus siglas en inglés) basado en un controlador de Retroalimentación No Lineal Compuesta (CNF) y un Control por Super-Twisting (STC), para regular las posiciones de cabeceo y guiñada de un helicóptero de 2-DOF. Los resultados reflejan que el controlador propuesto es más eficiente en minimizar los índices de desempeño (ISE, ITSE e ITAE) en comparación con otros controladores como el CNF y un SMC Integral.

De igual forma, en [83], se propone un esquema robusto de Control Adaptativo de Referencia de Modelo (MRAC, Model Reference Adaptive Control, por sus siglas en inglés) basado en un LQR para el ajuste de posición de un helicóptero 2-DOF. Los resultados experimentales

muestran que, con el controlador propuesto, el tiempo de establecimiento del sistema es de 15 segundos y el error de seguimiento es $\pm 0.9^\circ$.

Por otra parte, David Nodland, Hassan Zargarzadeh y Sarangapani Jagannathan en [12], proponen un enfoque de control inteligente mediante el diseño de un controlador adaptativo basado en Redes Neuronales para el seguimiento de trayectorias en un helicóptero no tripulado de 3-DOF. El análisis del sistema en lazo cerrado, realizado a través de simulaciones en Simulink, muestra que el helicóptero alcanza la estabilidad en un tiempo aproximado de 50 segundos para todos los ejes. No obstante, la respuesta transitoria presenta algunas variaciones con respecto a la trayectoria deseada.

Así mismo, en [84] se presenta el diseño de un control adaptativo de tipo SISO (*Single Input, Single Output*), para el vuelo vertical de un helicóptero, implementando Redes Neuronales. Al comparar este modelo con un controlador Proporcional-Derivativo (PD), los autores demostraron que el controlador diseñado mejora la precisión del seguimiento de la trayectoria logrando una mejor respuesta transitoria, sin sobreimpulsos, y reduciendo el error en estado estable de 0.02 a 0.001.

En [13] se describe el desarrollo de un sistema de control para el ajuste de posición del helicóptero Quanser de 2-DOF basado en Redes Neuronales Adaptativas de Base Radial. Los resultados obtenidos a través de simulaciones y experimentos indican que el controlador propuesto estabiliza el sistema en aproximadamente 15 segundos, alcanzando un error promedio de solo 0.0057° para el ángulo de cabeceo y 0.0133° para el ángulo de guiñada.

Alex Walker y Kelly Cohen en [85], presentan el diseño de un controlador inteligente de tipo MIMO para el helicóptero pivotado Quanser, utilizando Lógica Difusa. Las entradas del sistema incluyen los errores y las velocidades de los ángulos de cabeceo y guiñada, representados mediante 3 y 5 conjuntos difusos, respectivamente. Las simulaciones mostraron que el sistema difuso logra llevar al helicóptero a las posiciones deseadas; sin embargo, la respuesta es menos suave en comparación con la obtenida por un Regulador Cuadrático Lineal con acción Integral (*LQR+I*, por las siglas en inglés de *Linear Quadratic Regulator Integral*).

Finalmente, en [14] se presenta el desarrollo e implementación de un controlador de Lógica Difusa (*Fuzzy Logic Control, FLC, por sus siglas en inglés*) aplicado al ajuste de posición de un Tower-Copter, un helicóptero de 1-DOF que le permite solo volar verticalmente. El estudio compara el rendimiento del controlador FLC tipo Sugeno con el de un controlador PID, ambos implementados en un Arduino Uno, y muestra que el tiempo de establecimiento del FLC fue de 0.5 segundos frente a los 1.2 segundos del PID, representando una mejora del 58.33%.

4.2 Sintonización de controladores mediante Inteligencia Computacional

En la literatura se documentan trabajos que utilizan técnicas de optimización basadas en heurísticas para ajustar controladores de helicópteros de 2-DOF. Un primer ejemplo de esto es el trabajo de Ramalakshmi et al. [86], quienes mediante la sintonización de los parámetros de ganancia de un controlador PID utilizando un algoritmo de Optimización por Enjambre Partículas (PSO, por sus siglas en inglés), una versión modificada de este al que denominan MPSO y un Algoritmo Genético (GA) minimizan el Error Absoluto Integral en el Tiempo (ITAE) y el Error Cuadrático Integral en el Tiempo (ITSE). Los resultados muestran que el MPSO supera al PSO y al GA en términos de convergencia y reducción del error en estado estable del helicóptero.

Así mismo, en [11] se desarrolla un PSO, con un factor de inercia adaptable que ajusta dinámicamente las velocidades de las partículas, para sintonizar los pesos de las matrices de un LQR en un helicóptero de 2-DOF. Los experimentos muestran que el PSO adaptativo logra una reducción de la función objetivo ISE en los ejes de cabeceo y guiñada, obteniendo valores de 2.772 rad y 1.722 rad respectivamente, en comparación con los obtenidos por un PSO convencional de 5.446 rad y 5.377 rad .

En [87] se aborda el uso de un controlador LQR optimizado mediante un algoritmo PSO para un sistema aerodinámico de dos rotores dividido en dos subsistemas SISO. Las simulaciones muestran mejoras en el desempeño del sistema cuando se utilizó el controlador LQR optimizado en comparación con un controlador estándar. Para el eje de cabeceo, el tiempo de establecimiento se redujo de 7.58 segundos a 1.73 segundos, y el sobreimpulso disminuyó

de 27.73% a 1.91%. En el eje de guiñada, el tiempo de establecimiento se redujo de 3.47 segundos a 2.23 segundos y se eliminó el sobreimpulso.

En [88] se describe la implementación de un GA y un PSO para sintonizar un LQR aplicado a un helicóptero de 2-DOF. Se muestra que el uso de técnicas de Inteligencia Computacional para el ajuste de las matrices del LQR permite una respuesta más rápida y estable del sistema en comparación con la sintonización manual del controlador, reduciendo 30% los criterios de desempeño, los que se traduce en un tiempo de establecimiento de 6.09 y 6.34 segundos para el ángulo de cabeceo y guiñada respectivamente, mientras que el sobreimpulso fue de 0.03% en ambos ejes.

Por otra parte, Vishal Chugh y Jyoti Ohri en [15] presentan una comparativa entre un controlador LQR y un PID, ajustan de manera óptima los parámetros ambos controladores mediante un GA, para el control del ángulo de cabeceo de un avión. El GA minimiza el Error Absoluto Integral (IAE), el Error Cuadrático Integral (ISE), el ITAE y el ITSE. Las simulaciones muestran un mejor desempeño de la aeronave al utilizar el LQR optimizado, el cual presenta un tiempo de establecimiento de 0.366 segundos y un sobreimpulso del 4.35%. Comparativamente, el sistema controlado con el PID muestra un tiempo de establecimiento de 78 segundos y con un sobreimpulso del 8.69%.

En [89] se propone un esquema de control híbrido basado en un LQR y un controlador PD, optimizados mediante el Algoritmo de Polinización Floral (FPA, por sus siglas en inglés), aplicados al control de un helicóptero de 3-DOF. Los resultados indican que el control PD-LQR optimizado por FPA mejora el desempeño en comparación con el LQR optimizado por sí solo. Para ángulo de elevación, el tiempo de establecimiento se redujo de 4.64 a 0.83 segundos, eliminando el sobreimpulso del 4.2%. En el ángulo de desplazamiento, el tiempo de establecimiento disminuyó de 5.67 a 2.98 segundos, sin sobreimpulsos.

Finalmente, en [90] se presenta un controlador de modo deslizante de orden superior óptimo (OSOSMC) combinado con un LQR para un helicóptero de 2-DOF. El diseño incluye la sintonización simultánea de ambos controladores mediante un algoritmo de Optimización por Colonia de Hormigas (ACO, por sus siglas en inglés), un PSO y un GA utilizando el ISE como función objetivo. El controlador ajustado con el GA logró el menor índice de error, con

valores de 240.58 para el ángulo de cabeceo y 249.01 para el de guiñada, superando a los obtenidos con el ACO (256.75 para cabeceo y 367.75 para guiñada) y el PSO (265.50 para cabeceo y 516.34 para guiñada).

4.3 Matriz de referencias del estado del arte

De la revisión de la literatura especializada, es posible sintetizar los trabajos relacionados con la regulación de helicópteros pivotados en los ejes temáticos de la Tabla 4.1, donde se resumen las técnicas aplicadas y se identifican áreas de mejora.

Los estudios revisados constituyen un punto de partida para explorar la aplicación de técnicas de Inteligencia Computacional en el ajuste de controladores, evidenciando la eficacia de los algoritmos heurísticos para mejorar el desempeño de Reguladores Lineales Cuadráticos en helicópteros pivotados. No obstante, se identifica la ausencia de propuestas orientadas al desarrollo de heurísticas reducidas basadas en Inteligencia de Enjambres, lo que abre una línea de investigación en la literatura especializada. Con base en ello, el siguiente capítulo expone la metodología propuesta, en la cual se definen los fundamentos, procedimientos y herramientas que sustentan el desarrollo experimental de la presente investigación.

Tabla 4.1 Síntesis de trabajos relacionados.
(Elaboración propia)

Ejes	<p>A) Técnicas de control aplicadas a la regulación de posición de un helicóptero pivotado o aeronaves similares.</p> <p>B) Sintonización de controladores mediante técnicas de Inteligencia Computacional, aplicados a un helicóptero pivotado o aeronaves similares.</p>
Hechos	<p>Los trabajos de eje A se pueden clasificar en:</p> <ol style="list-style-type: none"> 1. Control óptimo (LQR, LQR-I y LQI). 2. Control robusto (Modo Deslizante e híbridos como PID-LQR). 3. Control inteligente (Redes Neuronales Artificiales y Lógica Difusa). <p>Los trabajos de eje B se pueden clasificar en:</p> <ol style="list-style-type: none"> 1. Algoritmos Evolutivos (Algoritmos Genéticos). 2. Inteligencia de Enjambres (Algoritmo de Optimización por Enjambre de Partículas, Algoritmo de Optimización por Colonia de Hormigas, Algoritmo de Polinización Floral).

Áreas de oportunidad Los trabajos revisados hacen uso de algoritmos heurísticos clásicos, sin explorar algoritmos recientes. Esto plantea la posibilidad de explorar nuevas alternativas basadas en Inteligencia de Enjambres, que podrían ofrecer beneficios en términos de eficiencia y/o convergencia.

La literatura existente no ha explorado las implicaciones computacionales de las diversas heurísticas. Se identifica una oportunidad para investigar la implementación de versiones reducidas de estos algoritmos que permitan reducir los recursos necesarios para su ejecución.

Propuesta Del análisis del estado del arte es posible plantear el uso de:

- Un Regulador Lineal Cuadrático para el control de posición de un helicóptero pivotado, como el planteado en [78], [79], [11] y [87].
 - Algoritmos heurísticos basados en Inteligencia de Enjambres, para sintonizar los parámetros del LQR, como los utilizados en [86], [88], [89] y [90].
 - Los índices ITAE, ISE, ITSE e IAE, como función objetivo a minimizar, tal como se plantea en [86], [11] y [15].
 - Versiones reducidas de los algoritmos heurísticos, para reducir el costo computacional respecto de sus versiones canónicas.
 - Un análisis estadístico inferencial, como el propuesto por [86] y [11], para comparar el desempeño de los algoritmos heurísticos reducidos.
-
-

5 METODOLOGÍA

5.1 Método

La Figura 5.1 presenta el esquema metodológico a seguir para la sintonización del LQR aplicado al helicóptero pivotado mediante heurísticas reducidas basadas en Inteligencia de Enjambres. El proceso inicia con una revisión del estado del arte orientada a identificar y recopilar los modelos matemáticos del sistema a regular reportados en la literatura, de modo que el comportamiento dinámico del mismo quede representado con la mayor fidelidad posible. A partir de este análisis, se procede a una segunda etapa de revisión enfocada en las técnicas de control aplicadas en sistemas dinámicos equivalentes, prestando especial atención a aquellas relacionadas con el control óptimo y el diseño de reguladores LQR.

En una tercera fase de la metodología se examinan los algoritmos de Inteligencia de Enjambres más recientes y relevantes en la literatura de optimización, lo que permite establecer una base sólida para la selección de las heurísticas a utilizar. Con esta información, se implementan en MATLAB® los algoritmos seleccionados en su versión canónica, codificándolos uno a uno y verificando su capacidad de convergencia y reducción de valor de la función objetivo, el vector solución encontrado corresponde a los valores de las matrices de ponderación LQR, con este valor se verifican los criterios de desempeño establecidos.

Después de verificar las versiones canónicas, se diseñan las variantes reducidas de cada algoritmo, orientadas a disminuir el costo computacional sin comprometer la calidad de las soluciones. Dichas versiones se validan en la misma tarea de sintonización, comparando su efectividad frente a las implementaciones originales. Los resultados obtenidos se analizan mediante estadística inferencial, esto permite determinar si existen diferencias significativas entre las versiones canónicas y reducidas.

Se realiza la evaluación de las heurísticas desarrolladas considerando su capacidad para minimizar la función objetivo, su velocidad de convergencia y la consistencia de sus resultados. Con base en estos criterios se seleccionan la heurística canónica y la heurística reducida más adecuadas, a fin de validar su desempeño en el mantenimiento de la estabilidad

del helicóptero ante distintas condiciones de operación, incluyendo la presencia de perturbaciones y variadas señales de referencia.

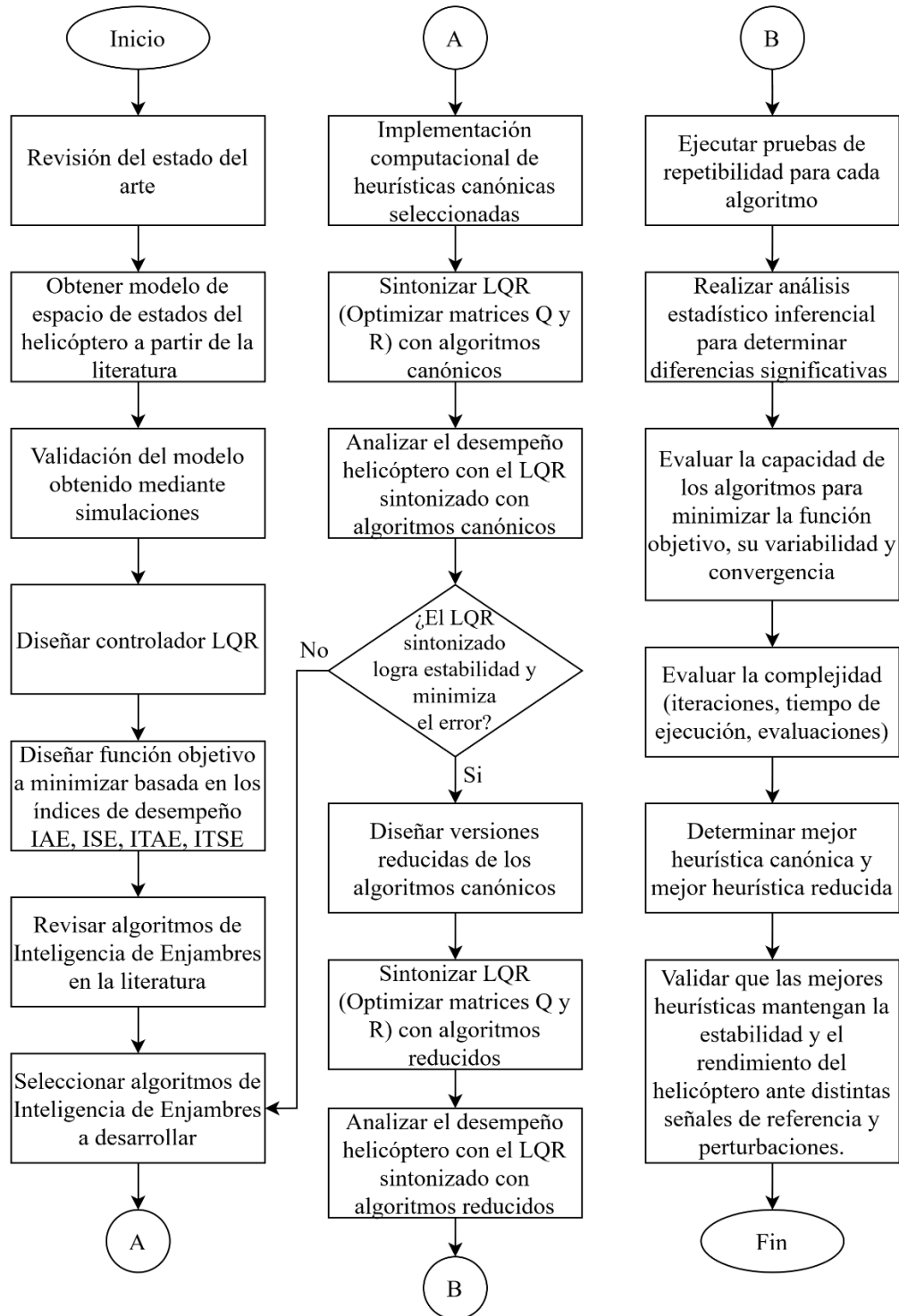


Figura 5.1 Esquema metodológico propuesto. (Elaboración propia).

5.2 Modelado y simulación del helicóptero

El modelo del helicóptero pivotado descrito en la sección 3.1.2 se implementó en el entorno MATLAB® mediante la representación en espacio de estados, usando las matrices A , B , C y D , cuyos valores se muestran en la Tabla 5.1.

Tabla 5.1. Matrices de estado del helicóptero pivotado.
(Adaptado de [29]).

Matriz	Valores
A	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -9.2768 & 0 \\ 0 & 0 & 0 & -3.4931 \end{bmatrix}$
B	$\begin{bmatrix} 0 & 0 \\ 2.3667 & 0.0789 \\ 0.2406 & 0.7909 \end{bmatrix}$
C	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
D	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Con el fin de analizar la dinámica en ausencia de regulador, se realizó una simulación a lazo abierto aplicando un escalón unitario en ambas entradas. El horizonte temporal se fijó en 40 segundos, con un intervalo de 0.01 segundos. La elección de estos valores obedece a dos razones: un tiempo de simulación suficientemente largo permite observar la tendencia del sistema sin limitaciones, y un paso de integración pequeño garantiza una resolución adecuada para captar la evolución continua del sistema. El proceso de simulación se describe en el Algoritmo 1.

Algoritmo 1. Simulación de la respuesta temporal del helicóptero sin regulación.

- 1 Definir matrices A , B , C , D del modelo del helicóptero.
 - 2 Construir el sistema en espacio de estados $G \leftarrow ss(A, B, C, D)$
 - 3 Definir el vector de tiempo T desde 0 hasta 40 s con paso 0.01 s.
 - 4 Definir la entrada U como escalón unitario en ambos ejes.
 - 5 Simular la respuesta temporal: $Y \leftarrow lsim(G, U, T)$.
 - 6 Graficar Y para ambos ejes en función de T .
-

La Figura 5.2 presenta la respuesta del helicóptero pivotado sin regular. Se observa que, al aplicarse la entrada de tipo escalón unitario, ambos ejes tienden al infinito, esto refleja el comportamiento inestable del sistema. Al no existir convergencia hacia un punto de equilibrio, no se aprecia un tiempo de establecimiento definido y, en consecuencia, el error respecto a la referencia tiende a aumentar con respecto al tiempo. Este comportamiento confirma la necesidad de un esquema de control que asegure estabilidad y un seguimiento eficiente de la señal de referencia deseada.

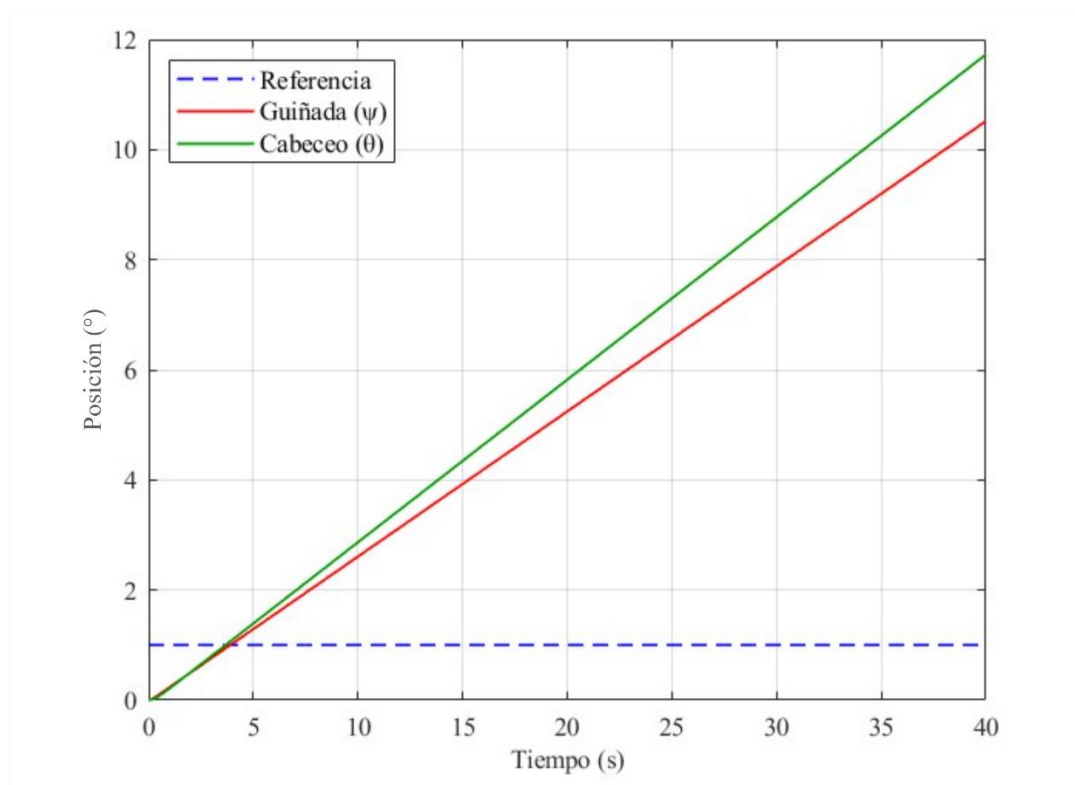


Figura 5.2 Respuesta inestable del helicóptero pivotado sin regulación. (Elaboración propia).

5.3 Diseño del controlador LQR

El diseño del Regulador Lineal Cuadrático (LQR) se realizó mediante la técnica analítica descrita en la sección 3.2.2. La implementación se efectuó en MATLAB® resolviendo la ecuación de Riccati y aplicando realimentación de estados $u(t) = -Kx(t)$ para el modelo del helicóptero pivotado. De esta manera, la dinámica del sistema en lazo cerrado queda descrita por la matriz modificada $A_c = A - BK$. Para evaluar el desempeño del regulador, se

simuló la respuesta temporal bajo las mismas condiciones que sin acción de control: una entrada de escalón unitario en ambos actuadores durante 40 s y paso de 0.01 s, tal como se muestra en el Algoritmo 2. Los valores de la matrices Q y R se muestran en la Tabla 5.2.

Algoritmo 2. Simulación de la respuesta temporal del helicóptero con LQR sintonizado analíticamente.

- 1 Definir matrices A, B, C, D del modelo del helicóptero.
- 2 Definir las matrices $Q \leftarrow C^T \cdot C$ y $R \leftarrow I_{2 \times 2}$
- 3 Calcular la matriz de ganancia $K \leftarrow lqr(A, B, Q, R)$
- 4 Calcular la matriz de retroalimentación $A_c \leftarrow A - B \cdot K$
- 5 Construir el sistema en espacio de estados $G \leftarrow ss(A_c, B, C, D)$
- 6 Definir el vector de tiempo T desde 0 hasta 40 s con paso 0.01 s.
- 7 Definir la entrada U como escalón unitario en ambos ejes.
- 8 Simular la respuesta temporal $Y \leftarrow lsim(G, U, T)$.
- 9 Graficar Y para ambos ejes en función de T .

Tabla 5.2. Matrices Q y R del Regulador Lineal Cuadrático sintonizado analíticamente. (Elaboración propia).

Matriz	Esquema generalizado	Valores
Q	$\begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & q_{33} & 0 \\ 0 & 0 & 0 & q_{44} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
R	$\begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

La Figura 5.3 muestra la respuesta con el controlador LQR. A diferencia de la simulación del helicóptero sin regulación, el sistema es estable y presenta comportamiento asintótico; alrededor de 22 segundos, ambos ejes se aproximan a la referencia. No obstante, persiste un error en régimen permanente: un eje queda por debajo y el otro se pasa, por lo que, aunque el error es mucho menor que sin control, no se anula. Este resultado es coherente con el diseño LQR de forma clásica, y sirve como línea base para la sintonización posterior mediante las heurísticas.

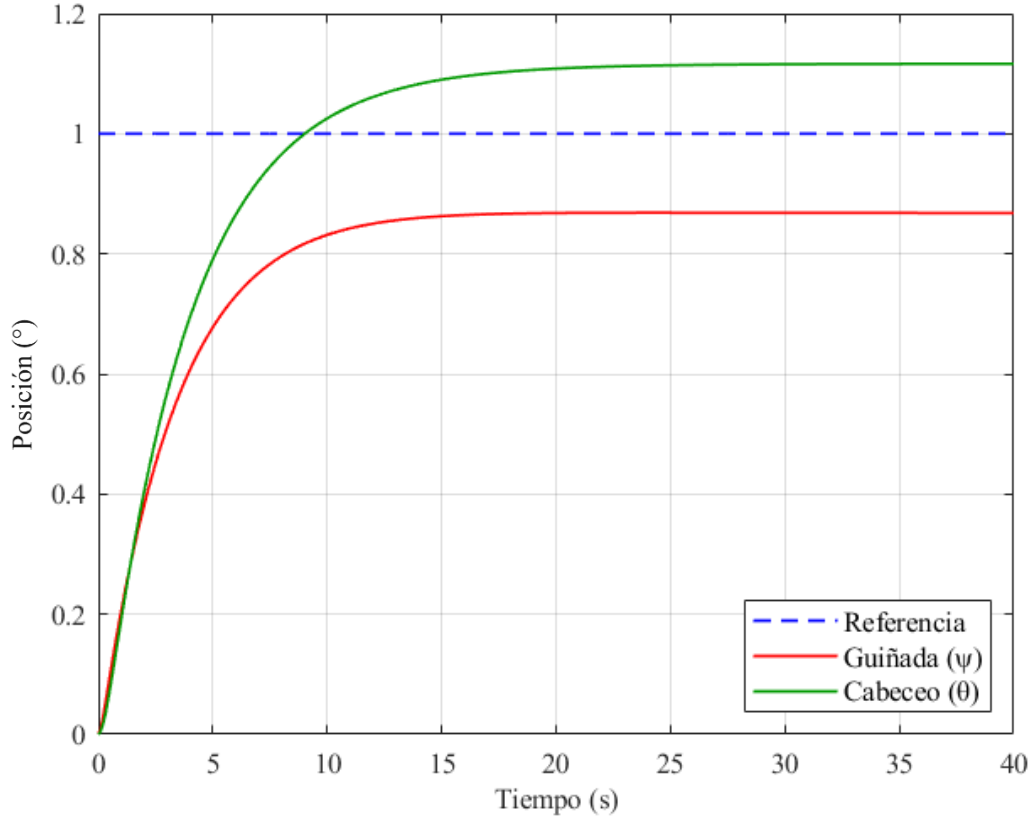


Figura 5.3 Respuesta del helicóptero pivotado con LQR analítico.
(Elaboración propia).

5.4 Definición de la función objetivo

La sintonización del LQR, se plantea como un problema de optimización que consiste en minimizar una función objetivo compuesta por los cuatro índices de desempeño que ya se han detallado en la sección 3.2.3.2: *IAE*, *ISE*, *ITAE* y *ITSE*. La suma de estas métricas, utilizadas en la literatura, captura tanto la magnitud del error como su persistencia temporal, de acuerdo con la ecuación 5.1.

$$F_{obj} = IAE + ISE + ITAE + ITSE \quad 5.1$$

La formulación completa se expresa en la ecuación 5.2. Sea $r(t)$ la referencia deseada (escalón unitario) y $y_j(t)$ las salidas de los ejes de guiñada y cabeceo del helicóptero; el error por eje es $e_j(t) = r(t) - y_j(t)$. La función objetivo utilizada por las heurísticas se evalúa sumando los índices de ambos ejes en el horizonte de simulación $t \in [0, T]$ con $T = 40$ s.

$$F_{obj} = \sum_{j=1}^2 \left[\int_0^T |e_j(t)| dt + \int_0^T e_j^2(t) dt + \int_0^T t|e_j(t)| dt + \int_0^T te_j^2(t) dt + \right] \quad 5.2$$

La evaluación de la función objetivo se implementó en MATLAB® mediante integración numérica con la regla trapezoidal. Para cada eje del helicóptero se calculó el error respecto a la señal de referencia y, a partir de éste, se obtuvieron sus versiones absolutas y cuadráticas, así como sus formas ponderadas en el tiempo. Posteriormente, se integraron estas señales a lo largo del tiempo de simulación, generando los valores del *IAE*, *ISE*, *ITAE* e *ITSE*. Finalmente, la suma de estos índices proporcionó un único valor escalar de desempeño, empleado como criterio de aptitud en los algoritmos heurísticos. Esta implementación se puede observar en el Algoritmo 3.

Algoritmo 3. Evaluación de la función objetivo en MATLAB®.

- 1 Calcular el error entre la referencia y la salida: $E \leftarrow U - Y$.
 - 2 Obtener el error absoluto: $AE \leftarrow |E|$.
 - 3 Obtener el error cuadrático: $SE \leftarrow E^2$.
 - 4 Calcular: $IAE \leftarrow \text{trapz}(T, AE)$, $ISE \leftarrow \text{trapz}(T, SE)$, $ITAE \leftarrow \text{trapz}(T, T \cdot AE)$, $ITSE \leftarrow \text{trapz}(T, T \cdot SE)$.
 - 5 Calcular la función objetivo total: $F_{obj} \leftarrow IAE + ISE + ITAE + ITSE$.
-

5.5 Implementación de algoritmos heurísticos canónicos

La implementación de los algoritmos heurísticos seleccionados se realizó en el entorno MATLAB®, a partir de sus formulaciones originales y con las adaptaciones necesarias para la sintonización del regulador LQR. La selección de los algoritmos se fundamenta en que todos ellos son metaheurísticos de enjambres marinos inspiradas en comportamientos de caza colectiva, con resultados competitivos reportados en la literatura reciente. Además, se incluyó al PSO, al ser una de las primeras técnicas basadas en inteligencia colectiva para establecer un punto de comparación directo con los otros métodos.

Con el fin de garantizar un equilibrio en la comparación, todos los algoritmos se ejecutaron bajo condiciones homogéneas de parametrización, determinadas con base en pruebas preliminares y un enfoque empírico. En cuanto a las variables de decisión, se consideraron únicamente las diagonales de las matrices Q y R , dado que en la práctica es común restringir la sintonización del LQR a dichos elementos. La Tabla 5.3 resume los parámetros de configuración empleados en todas las ejecuciones.

Tabla 5.3 Parámetros homogéneos de implementación de los algoritmos heurísticos.
(Elaboración propia).

Parámetro	Valor	Justificación
Número de agentes	100	Promueve una adecuada diversidad en la población inicial, reduciendo el riesgo de convergencia prematura.
Iteraciones máximas	60	Cantidad suficiente para observar la convergencia de las heurísticas sin incrementar excesivamente el costo computacional.
Espacio de búsqueda	[0, 10]	Intervalo amplio que garantiza la exploración del dominio de soluciones.
Dimensiones del problema	6	Cuatro variables correspondientes a la diagonal de Q y dos a la diagonal de R .
Variables de decisión	$[q_{11}, q_{22}, q_{33}, q_{44}, r_{11}, r_{22}]$	Elementos de las diagonales de Q y R seleccionados para la optimización.

Adicionalmente, todos los algoritmos fueron inicializados a partir de una misma población generada aleatoriamente, la cual se replicó de manera idéntica en cada ejecución. De esta forma se asegura la igualdad de condiciones iniciales para todos los métodos, lo que permite que las comparaciones posteriores reflejen únicamente las diferencias derivadas de la dinámica propia de cada heurística.

El diagrama de flujo de la Figura 5.4 define la estructura general bajo la cual se han implementado cada una de las heurísticas en su versión canónica.

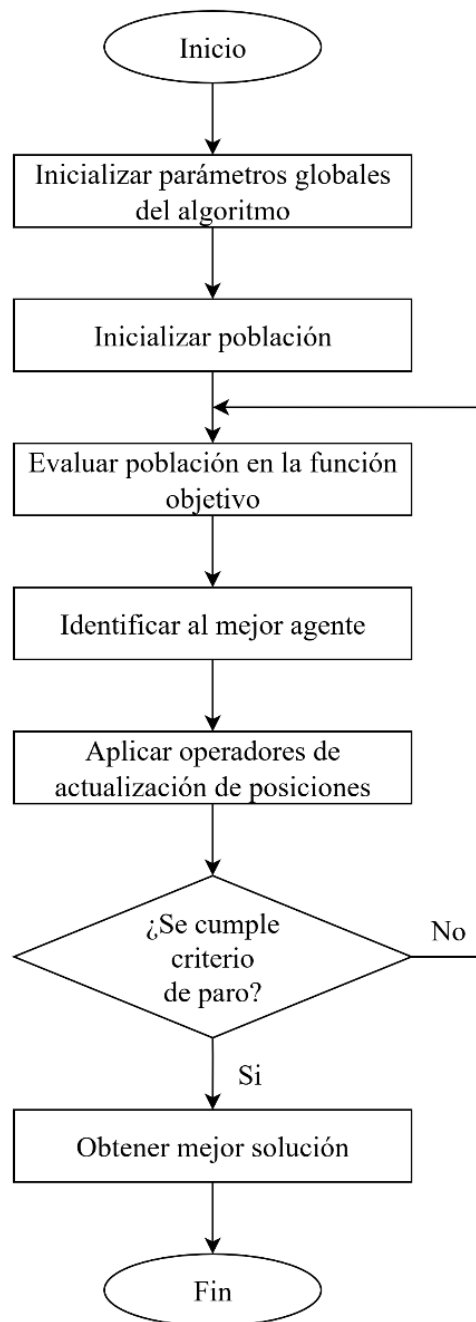


Figura 5.4. Estructura general de implementación de las heurísticas canónicas. (Elaboración propia).

5.6 Diseño de heurísticas reducidas

El diseño de heurísticas reducidas se planteó como una alternativa orientada a disminuir el costo computacional de los algoritmos metaheurísticos sin alterar sus principios esenciales de exploración y explotación. Para ello, se recurrió a tres estrategias principales: el uso de micro poblaciones, la incorporación de reinicios periódicos y la aplicación de un elitismo selectivo. En este esquema, las micro poblaciones se definieron con un tamaño fijo de 5 agentes, lo que permite reducir de manera significativa el número de evaluaciones en cada iteración y, por ende, el tiempo de cómputo, manteniendo al mismo tiempo un nivel mínimo de diversidad que evita la pérdida completa de variabilidad en el proceso de búsqueda.

En contrapeso, la exploración global se fortaleció mediante reinicios periódicos combinados con elitismo selectivo. Para ello, se estructuraron ciclos externos de 100 iteraciones, al final de los cuales se conserva el mejor agente encontrado y se generan nuevos individuos aleatorios para conformar la siguiente micro población. Este mecanismo garantiza la transferencia del mejor agente entre ciclos, evita el estancamiento en óptimos locales y renueva de forma constante la diversidad. A su vez, los ciclos internos de 5 iteraciones permiten una intensificación controlada alrededor de las soluciones más prometedoras, complementando la exploración global con un proceso de explotación local eficiente.

En conjunto, estas modificaciones dieron lugar a las versiones reducidas de los algoritmos canónicos que conservan su estructura conceptual, pero con un costo computacional considerablemente menor. Cabe señalar que todas las versiones reducidas también fueron desarrolladas bajo criterios de homogeneización de parámetros, de modo que comparten la misma población inicial, mismos límites de búsqueda, número de variables y representación de las matrices Q y R (véase Tabla 5.4). Esta uniformidad asegura que las comparaciones con sus contrapartes canónicas sean objetivas y que las diferencias en el desempeño puedan atribuirse exclusivamente al rediseño metodológico, y no a diferentes configuraciones experimentales.

Tabla 5.4 Parámetros de configuración de las versiones reducidas.
(Elaboración propia).

Parámetro	Valor	Justificación
Número de agentes	5	Disminuye el costo computacional manteniendo diversidad mínima.
Iteraciones internas	5	Permiten la intensificación local sin sobrecarga.
Iteraciones externas	100	Aseguran exploración global mediante reinicios controlados.
Mecanismo de reinicio	Elitismo aleatorio	Conservar el mejor agente y regenerar los restantes para evitar estancamiento.
Espacio de búsqueda	[0, 10]	Uniforme con las versiones canónicas para garantizar comparabilidad.
Dimensiones del problema	6	Cuatro variables correspondientes a la diagonal de Q y dos a la diagonal de R .
Variables de decisión	$[q_{11}, q_{22}, q_{33}, q_{44}, r_{11}, r_{22}]$	Elementos de las diagonales de Q y R seleccionados para la optimización.

El diagrama de flujo de la Figura 5.5 define la estructura general bajo la cual se han implementado cada una de las heurísticas en su versión reducida.

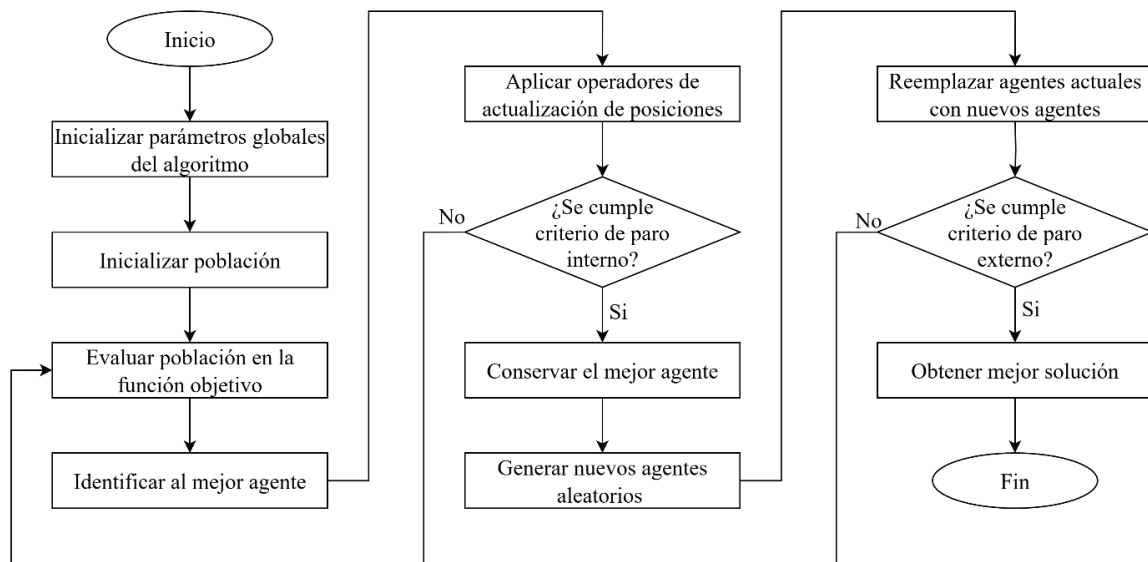


Figura 5.5. Estructura general de implementación de las heurísticas reducidas.
(Elaboración propia).

5.7 Análisis estadístico inferencial

Con el propósito de evaluar la consistencia y robustez de los algoritmos en sus versiones canónicas, se realiza una prueba de repetibilidad basada en 40 ejecuciones independientes por cada heurística. Este número de repeticiones se fundamenta en el Teorema del Límite Central, el cual establece que, al incrementar el número de muestras independientes de una población, la distribución de la media muestral tiende a aproximarse a una distribución normal, independientemente de la forma de la distribución original. De acuerdo con [91], un tamaño muestral de 30 o más observaciones suele considerarse suficiente para aplicar inferencias estadísticas con una precisión aceptable. Por lo tanto, se emplean 40 ejecuciones con el fin de garantizar una base estadística sólida para el análisis de variabilidad y estabilidad de los resultados.

Los valores obtenidos de la función objetivo se analizan mediante las pruebas de normalidad Shapiro–Wilk y Kolmogorov–Smirnov, con el fin de verificar si los datos se ajustan a una distribución normal. En todos los casos, ambas pruebas rechazan la hipótesis nula de normalidad, lo que indica que los resultados presentan una distribución no normal. Ante esta condición, se aplica la prueba no paramétrica de Wilcoxon para comparar los pares de algoritmos y determinar si existen diferencias estadísticamente significativas en su desempeño. Esta etapa es fundamental, ya que permite establecer si los algoritmos difieren de manera real y no solo por efecto aleatorio inherente al carácter estocástico de los métodos de optimización.

Dado que los datos no siguen una distribución normal, se emplean la mediana y el rango intercuartílico (IQR) como métricas de desempeño. La mediana representa una medida robusta de la tendencia central menos sensible a valores extremos, y el IQR cuantifica la dispersión de los resultados dentro del 50% central de las observaciones. Estas métricas ofrecen una descripción más precisa y estable del comportamiento de las heurísticas en distribuciones no normales. Este procedimiento permite realizar una evaluación objetiva del desempeño de los algoritmos, asegurando que la selección de las heurísticas más adecuadas se base en evidencia estadística confiable y no en observaciones particulares o resultados aislados.

6 RESULTADOS

En este capítulo se presentan los resultados obtenidos a partir de la aplicación de las heurísticas consideradas en esta investigación, tanto en sus versiones canónicas como en sus variantes reducidas. Primero, se evalúa el desempeño de los algoritmos a través de sus curvas de convergencia, pruebas de repetibilidad y análisis estadístico inferencial. Posteriormente, se examina su capacidad de sintonización del regulador LQR aplicado al modelo de helicóptero pivotado. Con base en estos resultados, se comparan las versiones canónicas y reducidas, y se realiza un análisis de las soluciones de las mejores heurísticas bajo distintas condiciones de operación del helicóptero.

6.1 Desempeño de las heurísticas canónicas

Con el propósito de analizar la efectividad de las versiones reducidas de las heurísticas, se repitió el procedimiento descrito en la sección 5.7. La Tabla 6.1 presenta, además de los valores iniciales y finales de f_{obj} , el IQR de los resultados de las 40 ejecuciones de cada heurística. El IQR indica la dispersión de los datos alrededor de la mediana y, por tanto, la consistencia del algoritmo: valores menores reflejan menor variabilidad y mayor estabilidad en las ejecuciones. En este caso, los IQR observados son muy reducidos en relación con el valor final de la f_{obj} ; incluso los algoritmos con IQR más altos, como SFO y PSO, muestran una variación que representa menos del 3% del valor final, lo que indica que todas las heurísticas son altamente repetibles.

Tabla 6.1 Resultados de la prueba de repetibilidad de los algoritmos canónicos.
(Elaboración propia).

Ranking	Algoritmo	Valor inicial f_{obj}	Valor final f_{obj}	Q_1	Q_3	IQR
1	WSO	56.1050	45.6320	45.6320	45.6335	0.0015
2	OPA	56.1050	45.6360	45.6360	45.6370	0.0010
3	WOA	56.1050	45.6825	45.6490	45.7220	0.0730
4	BWO	56.1050	45.6915	45.6610	45.9635	0.3025
5	PSO	56.1050	46.2050	45.6740	46.8755	1.2015
6	SFO	56.1050	48.3540	47.8630	49.2575	1.3945

Estas diferencias de dispersión también se reflejan en las curvas de convergencia mostradas en las figuras siguientes, el área sombreada bajo cada curva representa la variabilidad entre ejecuciones. Áreas más estrechas coinciden con *IQR* bajos y muestran que la heurística mantiene resultados consistentes a lo largo de las iteraciones. En algunos casos, como WSO, WOA y OPA, el *IQR* es tan pequeño que el área sombreada no es visible, indicando una estabilidad prácticamente uniforme. Por el contrario, áreas ligeramente más amplias, correspondientes a *IQR* mayores, indican mayor dispersión, aunque aún dentro de rangos aceptables en términos porcentuales.

La curva de convergencia del PSO, ilustrada en la Figura 6.1, desciende de manera acelerada durante las primeras diez iteraciones. Posteriormente, la reducción del valor objetivo se torna más gradual, hasta alcanzar la convergencia alrededor de la iteración 30, con una mejora cercana al 17.5 % respecto al valor inicial. El algoritmo presenta una variabilidad más marcada entre ejecuciones, la cual se representa por el área sombreada, evidenciada en un *IQR* de 1.2015; sin embargo, esta dispersión no compromete su capacidad para localizar soluciones óptimas.

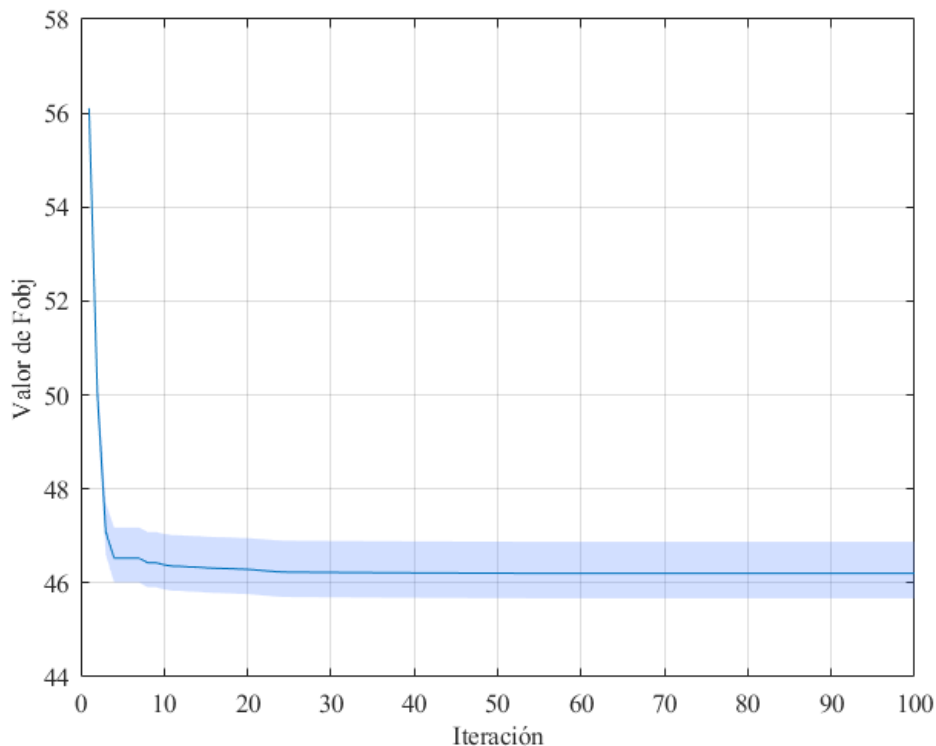


Figura 6.1 Curva de convergencia del algoritmo PSO.
(Elaboración propia).

La curva de convergencia del BWO presentada en la Figura 6.2, muestra una disminución progresiva del valor objetivo desde las primeras 20 iteraciones, logrando reducir más del 18% respecto al valor inicial. La convergencia se alcanza de manera gradual, estabilizándose alrededor de la iteración 60, momento en el que la curva entra en una zona con variaciones mínimas. Si bien el algoritmo logra una mejora considerable en términos de desempeño, la región sombreada refleja una variabilidad moderada entre ejecuciones de 0.3025.

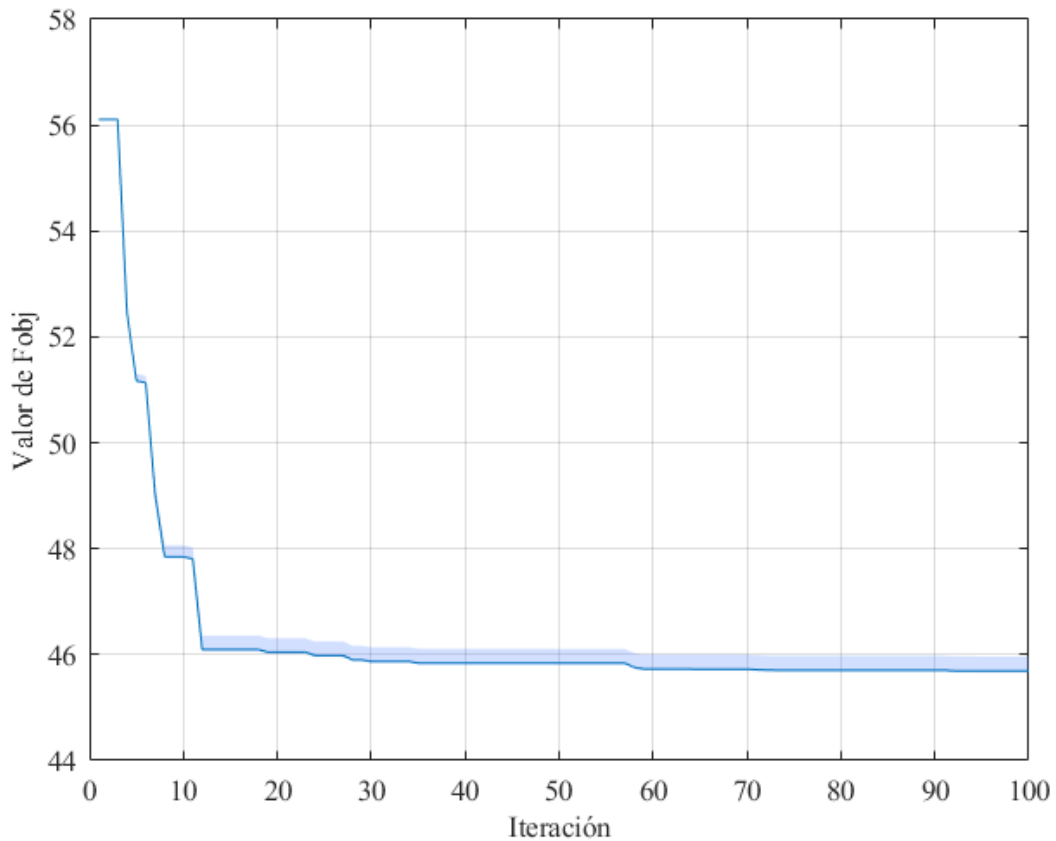


Figura 6.2 Curva de convergencia del algoritmo BWO.
(Elaboración propia).

La curva de convergencia del WSO, mostrada en la Figura 6.3, presenta un descenso muy pronunciado en las primeras cinco iteraciones, lo que podría sugerir un riesgo de convergencia prematura. Sin embargo, al alcanzar el valor final más bajo de todos los algoritmos evaluados, se descarta la posibilidad de estancamiento en óptimos locales. A partir de la iteración 10, la curva evidencia un comportamiento estable de convergencia, manteniéndose prácticamente constante en adelante. La variabilidad entre ejecuciones (0.0015) es tan reducida que resulta imperceptible en el gráfico, lo que demuestra la alta consistencia del algoritmo en sus resultados.

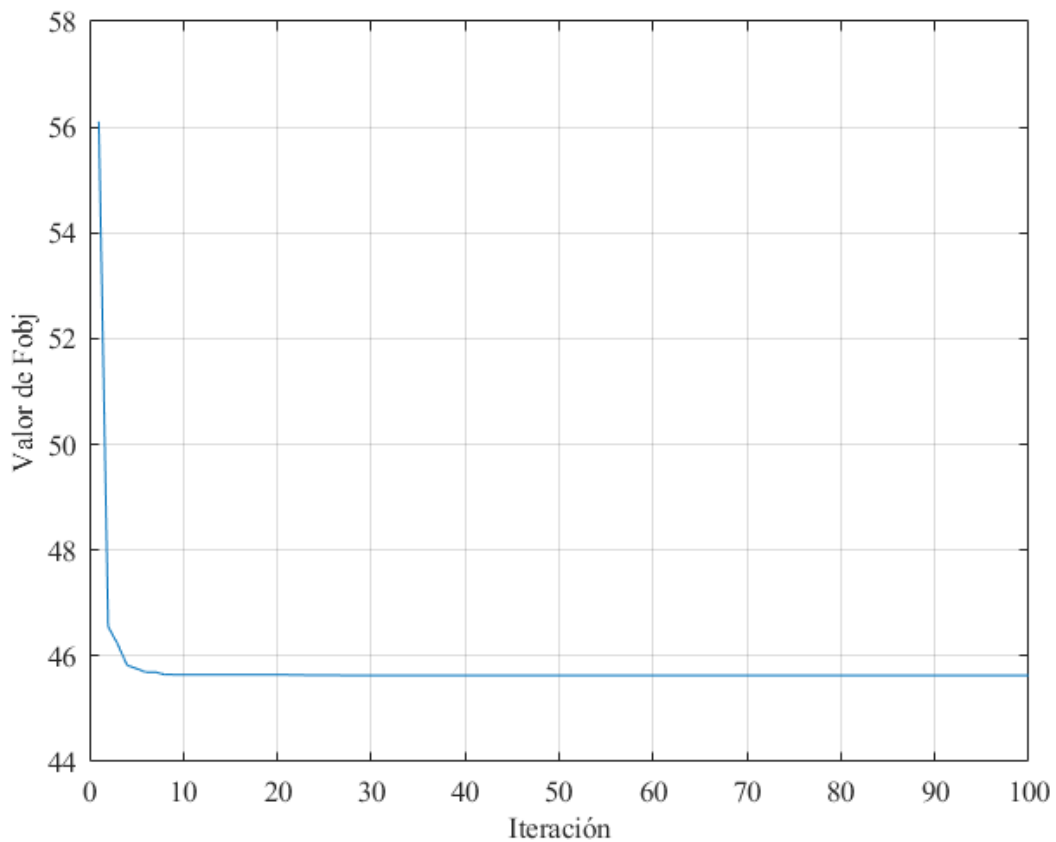


Figura 6.3 Curva de convergencia del algoritmo WSO.
(Elaboración propia).

La curva de convergencia del WOA, presentada en la Figura 6.4, exhibe el descenso más gradual entre todos los algoritmos analizados. La reducción del valor objetivo avanza de forma lenta y continua, comenzando a estabilizarse alrededor de la iteración 50. Incluso después de la iteración 80 se observan descensos adicionales hacia valores más pequeños. A pesar de este comportamiento, el WOA se ubica entre los métodos más efectivos al alcanzar de los valores finales más bajos de la función objetivo. La variabilidad entre ejecuciones es de apenas 0.0730, por lo que no se aprecia en la representación gráfica.

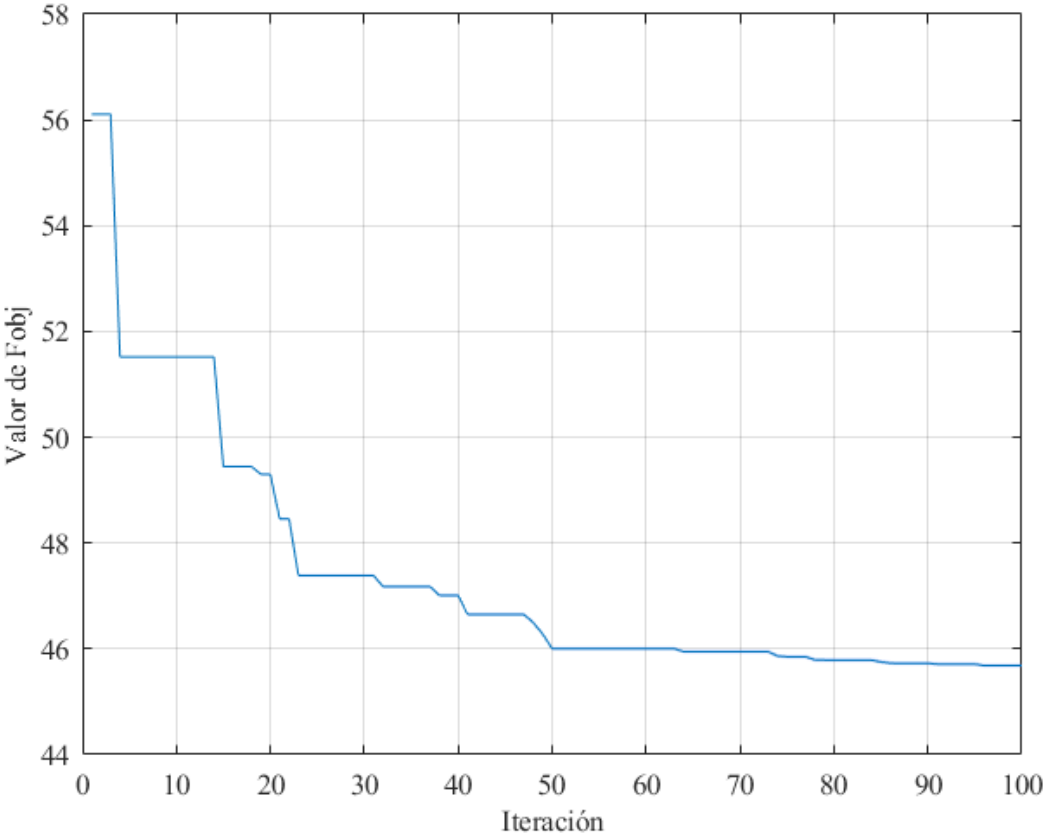


Figura 6.4 Curva de convergencia del algoritmo WOA.
(Elaboración propia).

La curva de convergencia del SFO, mostrada en la Figura 6.5, refleja una reducción temprana del valor objetivo, alcanzando la convergencia alrededor de la iteración 13. Sin embargo, en términos comparativos, se posiciona como el algoritmo con el desempeño menos favorable, ya que los valores alcanzados son superiores a los obtenidos por las demás heurísticas. Además, presenta la mayor variabilidad entre ejecuciones con un IQR de 1.3945, lo que se observa en la amplitud de la región sombreada. Cabe señalar que se realizaron pruebas adicionales ajustando los parámetros intrínsecos del algoritmo, pero estas modificaciones no lograron mejorar de manera significativa su rendimiento. Aun así, el SFO demuestra una capacidad aceptable de reducción del valor objetivo, aunque sin alcanzar la eficacia de los otros métodos evaluados.

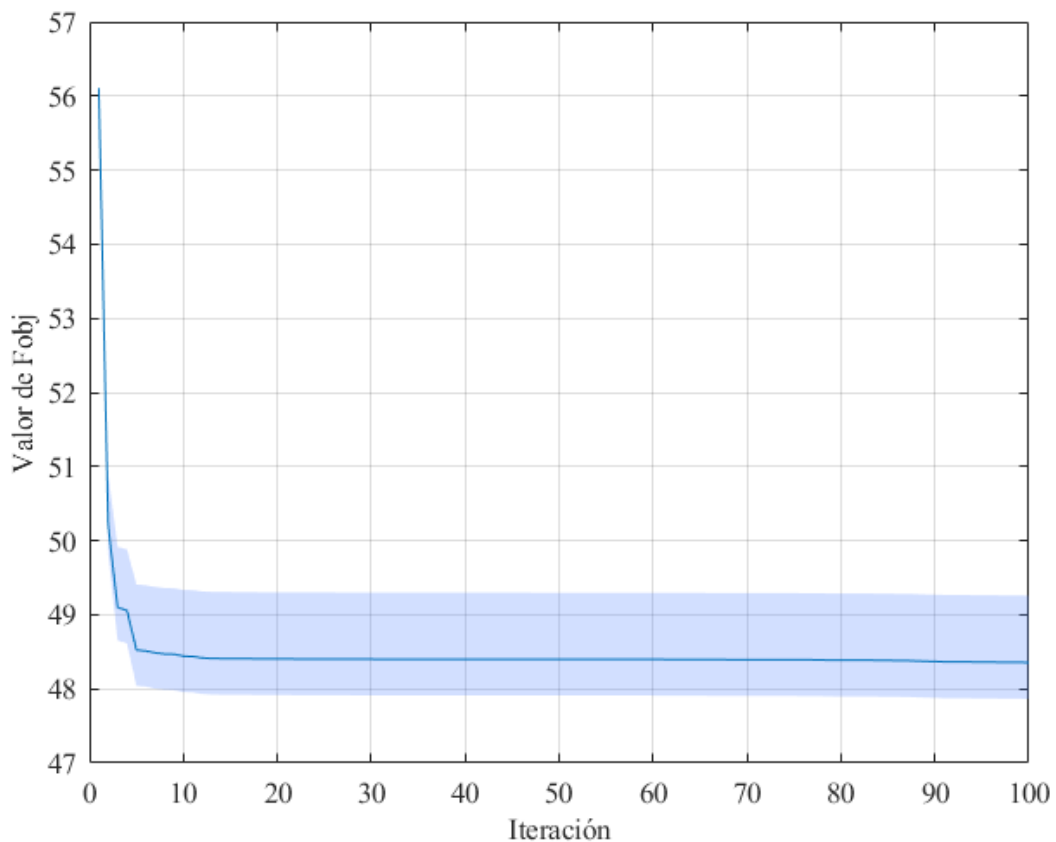


Figura 6.5 Curva de convergencia del algoritmo SFO.
(Elaboración propia).

La curva de convergencia del OPA, de la Figura 6.6, muestra un descenso acelerado del valor objetivo desde las primeras iteraciones, alcanzando la convergencia alrededor de la iteración 10. A primera vista, este comportamiento podría interpretarse como un indicio de convergencia prematura; sin embargo, al obtener consistentemente los mejores valores finales de la función objetivo entre todas las heurísticas, se descarta dicho supuesto. Además, el algoritmo se caracteriza por una variabilidad mínima entre ejecuciones con un IQR de 0.0010, siendo el de menor dispersión de todos los analizados. En conjunto, estos resultados evidencian que el OPA logra un equilibrio sobresaliente entre rapidez de convergencia y calidad de la solución alcanzada.

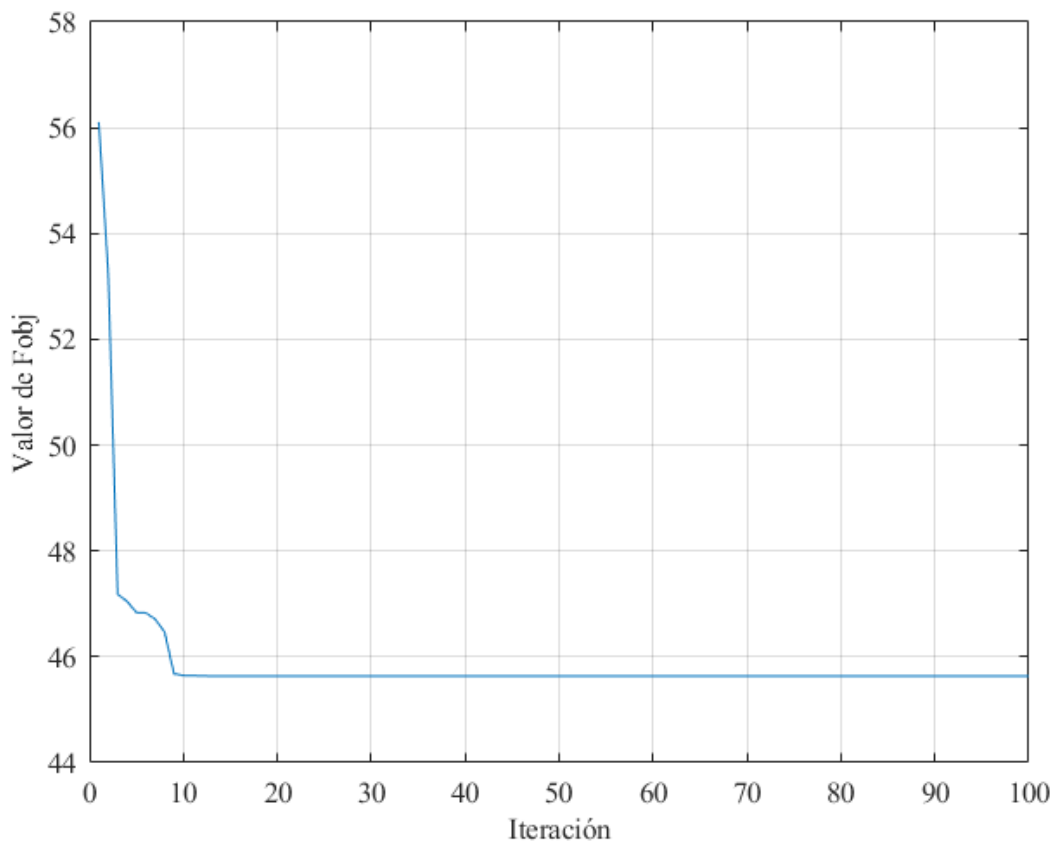


Figura 6.6 Curva de convergencia del algoritmo OPA.
(Elaboración propia).

La Figura 6.7 reúne en una misma gráfica las curvas de convergencia de todos los algoritmos en su versión canónica, lo que permite visualizar de manera directa sus diferencias de comportamiento. En términos de rapidez, tanto el WSO como el OPA alcanzan la convergencia en las primeras iteraciones, con resultados finales entre los más competitivos y prácticamente sin variabilidad. El WOA, aunque más lento en su descenso, logra valores muy próximos al óptimo y mantiene una estabilidad destacable en la parte final del proceso. Cabe señalar que el valor en torno a 45 corresponde al mínimo alcanzado de la función objetivo, la cual sin la sintonización del regulador LQR presenta un valor inicial de 339.35.

Por su parte, el PSO muestra un desempeño intermedio, con una reducción significativa y un nivel moderado de dispersión. En contraste, el BWO se caracteriza por una convergencia relativamente temprana y estable, aunque con mayor variabilidad que los anteriores. Finalmente, el SFO es el que presenta la menor efectividad, pues aunque reduce de forma rápida su valor objetivo, conserva la mayor dispersión entre ejecuciones y no logra alcanzar la calidad de solución de los demás algoritmos.

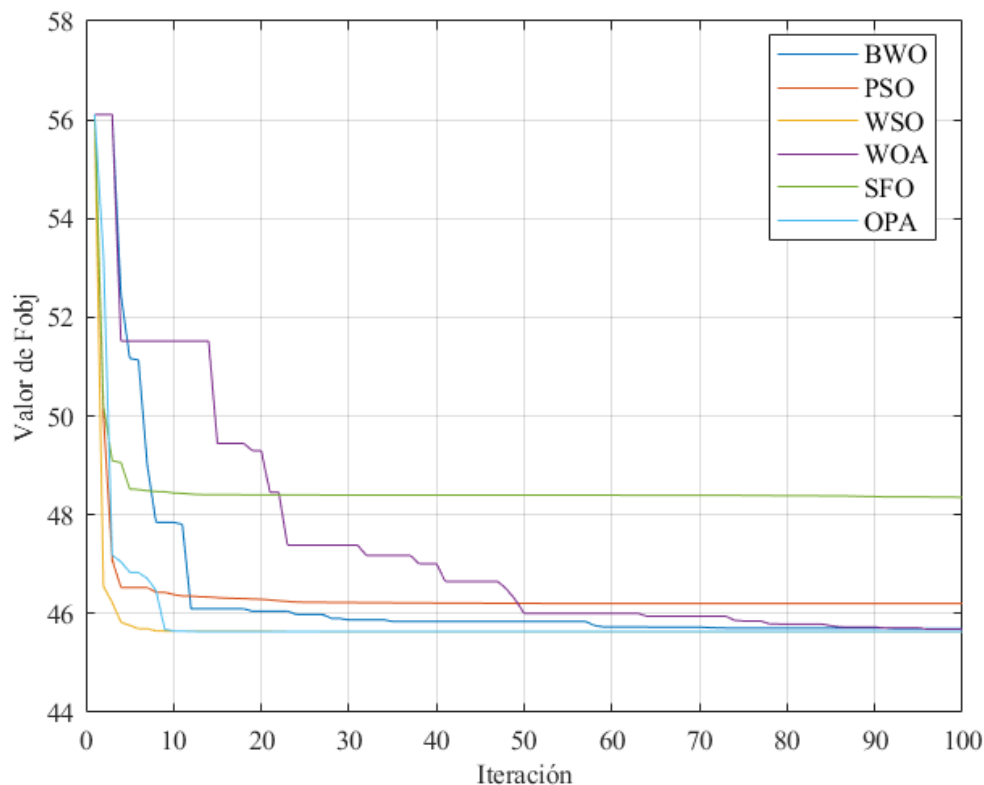


Figura 6.7 Comparativa de convergencia de los algoritmos canónicos. (Elaboración propia).

6.2 Sintonización del LQR con heurísticas canónicas

En esta sección se presentan los resultados obtenidos al emplear las versiones canónicas de las heurísticas en el ajuste de las matrices del LQR. Para cada algoritmo se utilizaron los valores de Q y R correspondientes a la mediana de las 40 ejecuciones de repetibilidad. Es importante destacar que estos resultados no deben confundirse con los de la sección anterior, al cual refleja el análisis estadístico global de las ejecuciones de optimización, mientras que aquí se ilustra el desempeño del helicóptero con el LQR diseñado mediante las soluciones más representativas de cada heurística.

La Figura 6.8 muestra un ejemplo de la respuesta del sistema al escalón unitario utilizando el LQR sintonizado con las heurísticas canónicas. Para evitar redundancia, solo se incluyen las trayectorias correspondientes al algoritmo que, con base en los valores de R y Q obtenidos a partir de la mediana de sus ejecuciones, alcanzó el menor valor de la función objetivo (WOA) así como al que registró el mayor valor en este mismo criterio (SFO). El resto de las respuestas resultan prácticamente indistinguibles entre sí, esta superposición confirma que, las heurísticas empleadas para sintonizar el LQR, logran estabilizar al helicóptero con características dinámicas muy similares.

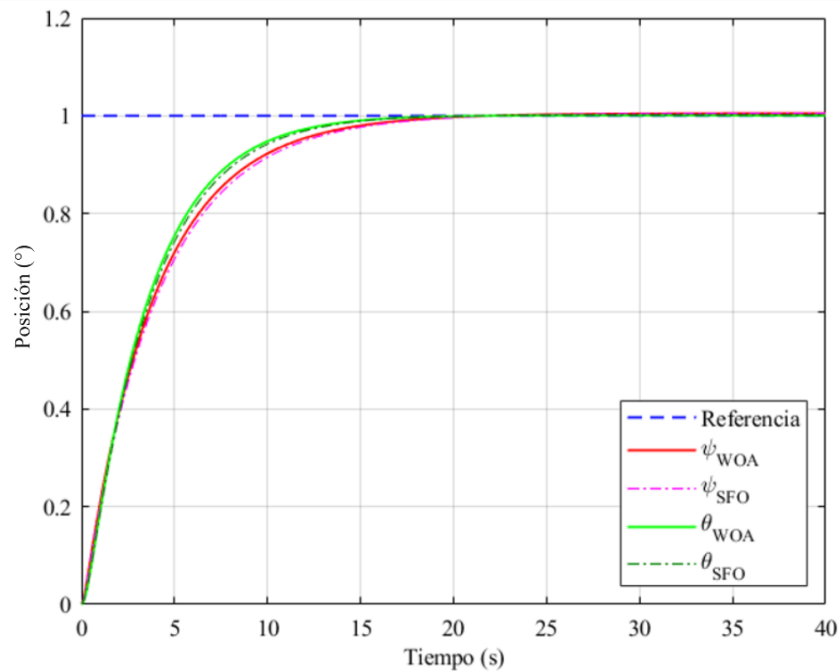


Figura 6.8 Respuesta al escalón del helicóptero con el LQR sintonizado mediante la heurística de mejor desempeño (WOA) y la de menor desempeño (SFO).
(Elaboración propia).

La Tabla 6.2 resume los principales indicadores de desempeño del LQR sintonizado con las heurísticas canónicas, considerando los ángulos ψ y θ . En todos los casos, el sistema diseñado con los valores de la mediana de Q y R mostró tiempos de establecimiento muy similares ($\approx 12s$ para ψ y $\approx 10s$ para θ), considerando una banda de tolerancia de $\pm 5\%$, salidas en estado estable muy próximas a la referencia y errores en estado estacionario prácticamente nulos, lo que confirma que las heurísticas conducen a respuestas dinámicas equivalentes.

No obstante, los criterios de desempeño permiten distinguir matices en la calidad de las soluciones. El WOA se posiciona como el algoritmo más efectivo al minimizar la f_{obj} , aunque este desempeño implica un mayor costo computacional debido al número de iteraciones requeridas para converger. En contraste, el SFO resultó ser el menos eficiente en la calidad de la solución, presentando los valores más altos en dichas métricas. Estos resultados permiten observar diferencias en el ajuste de posición del helicóptero alcanzado por cada heurística.

Tabla 6.2 Criterios de desempeño del LQR sintonizado con heurísticas canónicas.
(Elaboración propia).

Criterio		PSO	BWO	WSO	WOA	SFO	OPA	
t_s	ψ	12.1001	12.1010	12.1000	12.1007	12.5000	12.1000	
	θ	10.2000	10.2000	10.300	10.200	10.700	10.200	
e_{ss}	ψ	0.0057	0.0057	0.0054	0.0053	0.0053	0.0053	
	θ	0.0014	0.0008	0.0016	0.0009	0.0018	0.0009	
y_{ss}	ψ	1.0057	1.0057	1.0054	1.0053	1.0053	1.0053	
	θ	1.0014	1.0008	1.0016	1.0009	1.0018	1.0009	
f_{obj}	IAE	ψ	3.9883	3.9922	3.9971	3.9903	4.1280	3.9903
		θ	3.6101	3.6066	3.6178	3.6068	3.7311	3.6068
	ISE	ψ	2.0181	2.0172	2.0210	2.0177	2.0879	2.0177
		θ	1.9503	1.9511	1.9530	1.9511	2.0066	1.9511
	ITAE	ψ	16.8668	17.1310	17.0682	16.9904	17.9781	16.9917
		θ	12.2825	12.0425	12.4139	12.0527	13.2474	12.0540
	ITSE	ψ	3.8988	3.8945	3.9099	3.8971	4.1858	3.8971
		θ	3.3111	3.3159	3.3209	3.3158	3.5282	3.3158
	Σf_{obj}		47.9260	47.9510	48.3018	47.8219	50.8931	47.8245

La Tabla 6.3 presenta las matrices solución Q y R obtenidas a partir de la mediana de las 40 ejecuciones realizadas con cada heurística. Aunque la dinámica de la respuesta al escalón resultó prácticamente idéntica entre algoritmos, los valores de estas matrices ponen de manifiesto que cada técnica explora regiones distintas del espacio de búsqueda, proponiendo configuraciones particularmente diferentes. Estas diferencias confirman que, aun cuando el desempeño en el ajuste de posición del helicóptero sea similar, las heurísticas no convergen hacia la misma solución numérica, evidenciando las diferencias propias de sus procesos de exploración y explotación.

Tabla 6.3 Matrices solución sintonizadas con heurísticas canónicas.
(Elaboración propia).

Algoritmo	Matriz Q	Matriz R
PSO	$\begin{bmatrix} 4.0121 & 0 & 0 & 0 \\ 0 & 7.4367 & 0 & 0 \\ 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \end{bmatrix}$	$\begin{bmatrix} 5.7560 & 0 \\ 0 & 5.7421 \end{bmatrix}$
BWO	$\begin{bmatrix} 3.6360 & 0 & 0 & 0 \\ 0 & 6.7606 & 0 & 0 \\ 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \end{bmatrix}$	$\begin{bmatrix} 5.2255 & 0 \\ 0 & 5.2227 \end{bmatrix}$
WSO	$\begin{bmatrix} 4.8809 & 0 & 0 & 0 \\ 0 & 9.0496 & 0 & 0 \\ 0 & 0 & 0.3091 & 0 \\ 0 & 0 & 0 & 0.4201 \end{bmatrix}$	$\begin{bmatrix} 7.0077 & 0 \\ 0 & 7.0059 \end{bmatrix}$
WOA	$\begin{bmatrix} 5.3238 & 0 & 0 & 0 \\ 0 & 9.8893 & 0 & 0 \\ 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \end{bmatrix}$	$\begin{bmatrix} 7.6447 & 0 \\ 0 & 7.6299 \end{bmatrix}$
SFO	$\begin{bmatrix} 4.2086 & 0 & 0 & 0 \\ 0 & 7.7625 & 0 & 0 \\ 0 & 0 & 5.3479 & 0 \\ 0 & 0 & 0 & 6.7456 \end{bmatrix}$	$\begin{bmatrix} 6.0234 & 0 \\ 0 & 6.1304 \end{bmatrix}$
OPA	$\begin{bmatrix} 4.9468 & 0 & 0 & 0 \\ 0 & 9.1890 & 0 & 0 \\ 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \end{bmatrix}$	$\begin{bmatrix} 7.1034 & 0 \\ 0 & 7.1041 \end{bmatrix}$

6.3 Desempeño de las heurísticas reducidas

Con el propósito de analizar la efectividad de las versiones reducidas de las heurísticas, se repitió el procedimiento descrito en la sección 5.7. La Tabla 6.4 resume los resultados, incluyendo el valor inicial de la f_{obj} , la mediana del valor final alcanzado y IQR . El IQR refleja la dispersión de los resultados y, por tanto, la consistencia de cada algoritmo: valores menores indican mayor estabilidad entre ejecuciones. En las versiones reducidas, los IQR son ligeramente mayores que en las versiones canónicas, pero aun así representan variaciones relativamente pequeñas en términos porcentuales (de entre 2 y 7%) respecto al valor final de la f_{obj} .

Tabla 6.4 Resultados de la prueba de repetibilidad de los algoritmos reducidos.
(Elaboración propia).

Ranking	Algoritmo	Valor inicial f_{obj}	Valor final f_{obj}	$Q1$	$Q3$	IQR
1	μOPA	56.1050	46.4130	45.7290	47.5640	1.8350
2	μWOA	56.1050	46.4925	45.7630	47.0125	1.2495
3	μWSO	56.1050	47.1335	46.3345	47.8315	1.4970
4	μPSO	56.1050	47.8005	47.3770	48.4030	1.0260
5	μSFO	56.1050	47.9400	47.4735	48.5125	1.0390
6	μBWO	56.1050	49.6440	48.4095	51.9115	3.5020

Esta dispersión también se visualiza en las curvas de convergencia de las figuras siguientes, donde el área sombreada bajo cada curva indica la variabilidad entre ejecuciones. En heurísticas con IQR bajo, el área sombreada es perceptible pero restringida, mientras que en aquellas con mayor IQR , como μBWO , se observa un área más amplia, reflejando una dispersión mayor pero aún dentro de rangos aceptables. En general, las versiones reducidas mantienen una estabilidad suficiente para garantizar resultados confiables y repetibles, aunque con una variabilidad ligeramente superior a la de las versiones canónicas.

El μBWO presenta el IQR más amplio, reflejando variabilidad considerable entre ejecuciones, mientras que μPSO y μSFO muestran una dispersión más acotada. Por su parte, μWOA y μOPA alcanzan valores finales entre los más bajos, con una variabilidad moderada,

y μ WSO mantiene un buen equilibrio entre reducción del valor objetivo y estabilidad, aunque con *IQR* mayor al de μ PSO y μ SFO.

La curva de convergencia del μ PSO, de la Figura 6.9, presenta un descenso progresivo que se inicia de forma rápida en las primeras iteraciones y se estabiliza alrededor de la iteración 12, para comenzar una reducción más gradual hasta la iteración 91. Aunque la reducción del valor objetivo es menor que en su versión canónica, el algoritmo alcanza soluciones competitivas con una dispersión moderada, reflejada en un *IQR* relativamente estrecho. Este comportamiento indica que el μ PSO mantiene un buen equilibrio entre exploración y explotación, garantizando convergencia sin mostrar indicios de estancamiento prematuro.

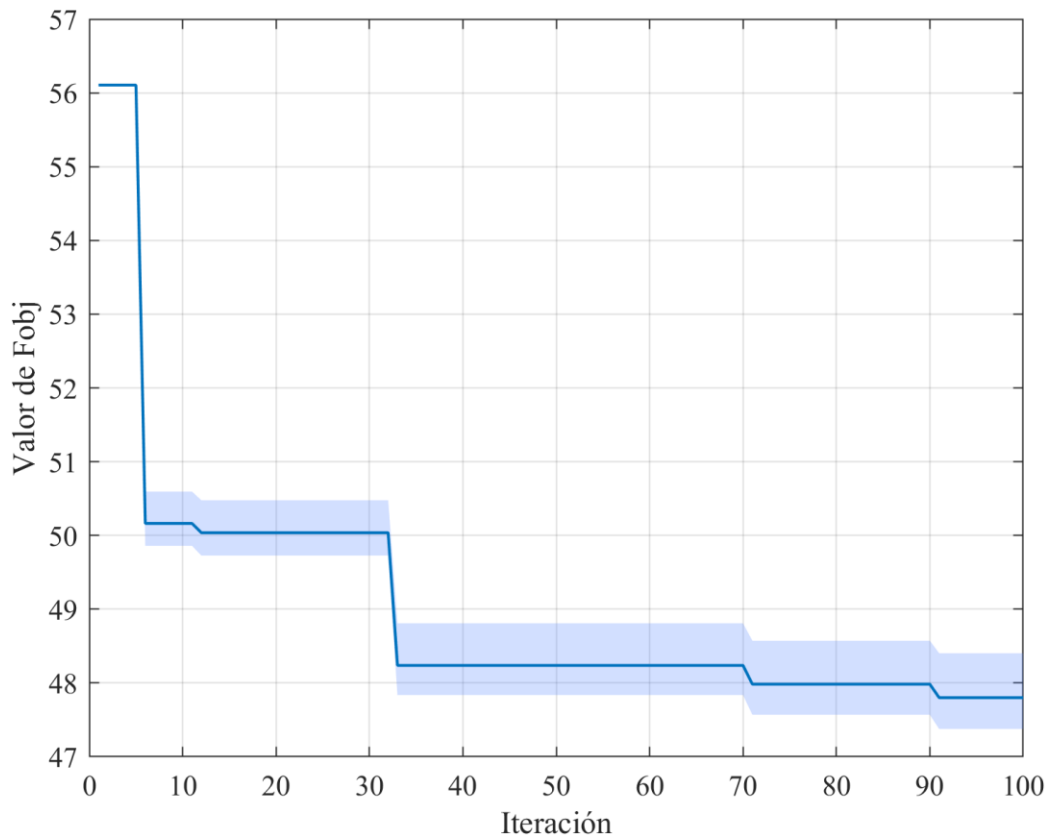


Figura 6.9 Curva de convergencia del algoritmo PSO reducido.
(Elaboración propia).

La curva de convergencia del μ BWO, de la Figura 6.10, muestra una caída inicial acelerada durante las primeras 15 iteraciones, seguida de una fase de reducción más lenta que se prolonga hasta la iteración 50, momento en que se estabiliza. En comparación con otros algoritmos reducidos, el μ BWO presenta la mayor dispersión entre ejecuciones, evidenciada en un IQR muy amplio. Aunque logra disminuir el valor objetivo de manera significativa, la variabilidad compromete su consistencia.

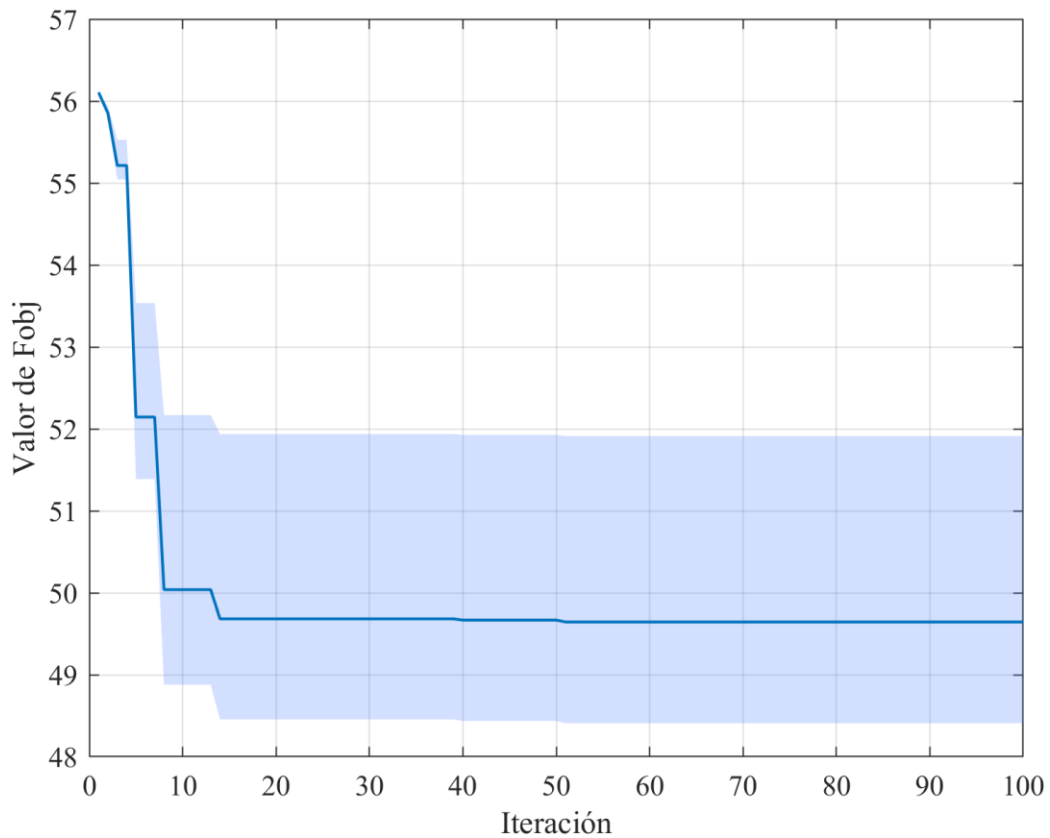


Figura 6.10 Curva de convergencia del algoritmo BWO reducido.
(Elaboración propia).

La curva de convergencia del μ WOA, de la Figura 6.11, se caracteriza por un descenso gradual que avanza de manera continua hasta estabilizarse alrededor de la iteración 37. En etapas tardías se observan pequeñas reducciones adicionales. Este algoritmo se ubica entre los más efectivos de su versión reducida, alcanzando valores finales bajos y con una dispersión intermedia, aunque superior a la de su versión canónica.

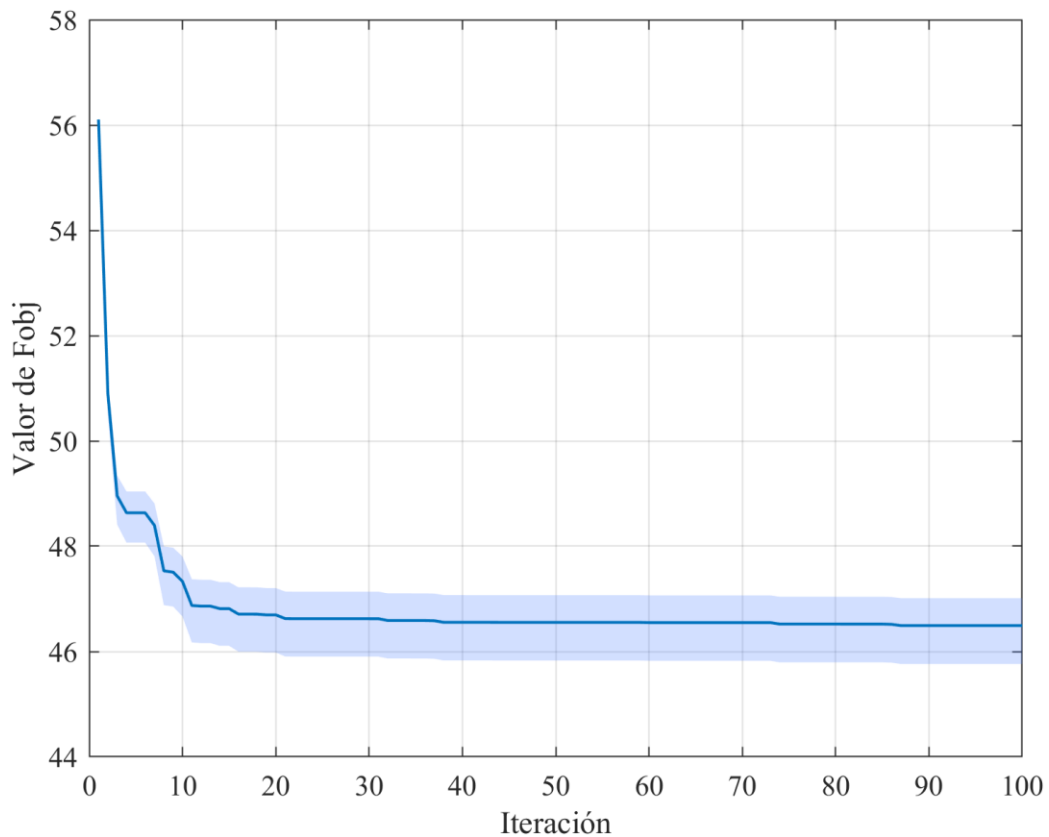


Figura 6.11 Curva de convergencia del algoritmo WOA reducido.
(Elaboración propia).

La curva de convergencia del μ WSO, de la Figura 6.12, desciende con gran rapidez durante las primeras diez iteraciones, alcanzando la convergencia poco después de la iteración 20. Si bien esta dinámica podría sugerir convergencia prematura, los valores finales obtenidos confirman que el algoritmo logra soluciones de buena calidad. La dispersión entre ejecuciones es moderada, con un IQR visible, aunque no excesivo, lo que refleja un desempeño robusto, aunque menos consistente que en su versión canónica.

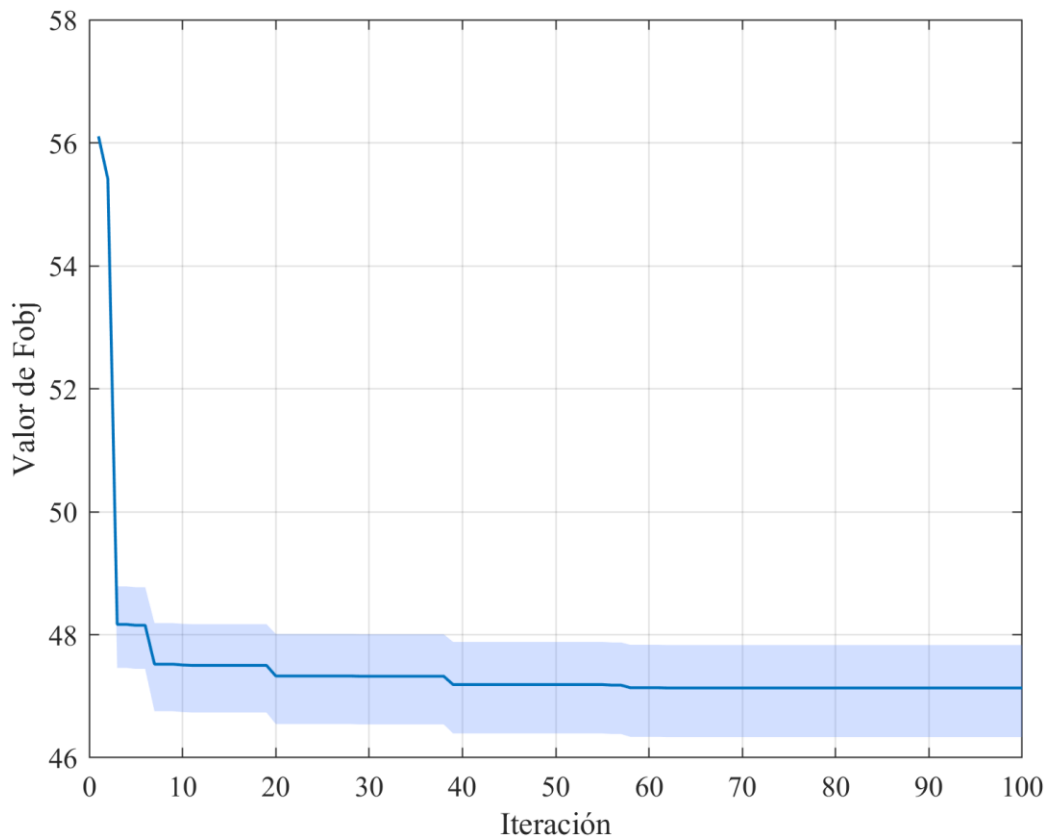


Figura 6.12 Curva de convergencia del algoritmo WSO reducido.
(Elaboración propia).

La curva del μ SFO, de la Figura 6.13, alcanza la convergencia más temprana entre las versiones reducidas, alrededor de la iteración 15. Aunque reduce el valor objetivo de manera efectiva, se posiciona entre los menos eficientes en comparación con los demás algoritmos, pues no alcanza los valores finales más bajos. No obstante, su variabilidad es reducida, con un IQR estrecho, lo que indica un comportamiento repetible y consistente entre ejecuciones, aunque limitado en cuanto a la calidad de la solución.

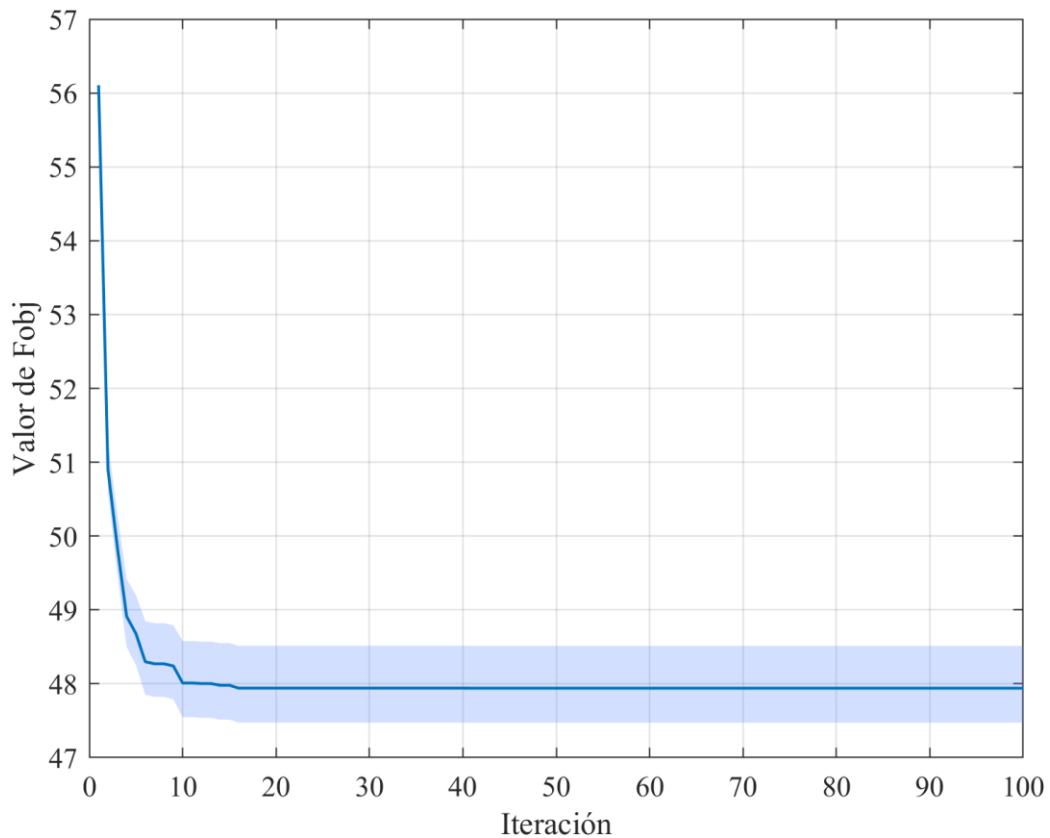


Figura 6.13 Curva de convergencia del algoritmo SFO reducido.
(Elaboración propia).

La curva de convergencia del μ OPA, de la Figura 6.14, presenta un descenso acelerado desde las primeras iteraciones y alcanza la convergencia antes de la iteración 10. Al igual que en su versión canónica, este comportamiento podría asociarse con una convergencia prematura; sin embargo, los valores finales obtenidos se encuentran entre los mejores de las versiones reducidas, lo que permite descartar este supuesto. El μ OPA presenta una variabilidad intermedia, mayor que en su versión original, pero mantiene un desempeño competitivo tanto en rapidez como en calidad de las soluciones alcanzadas.

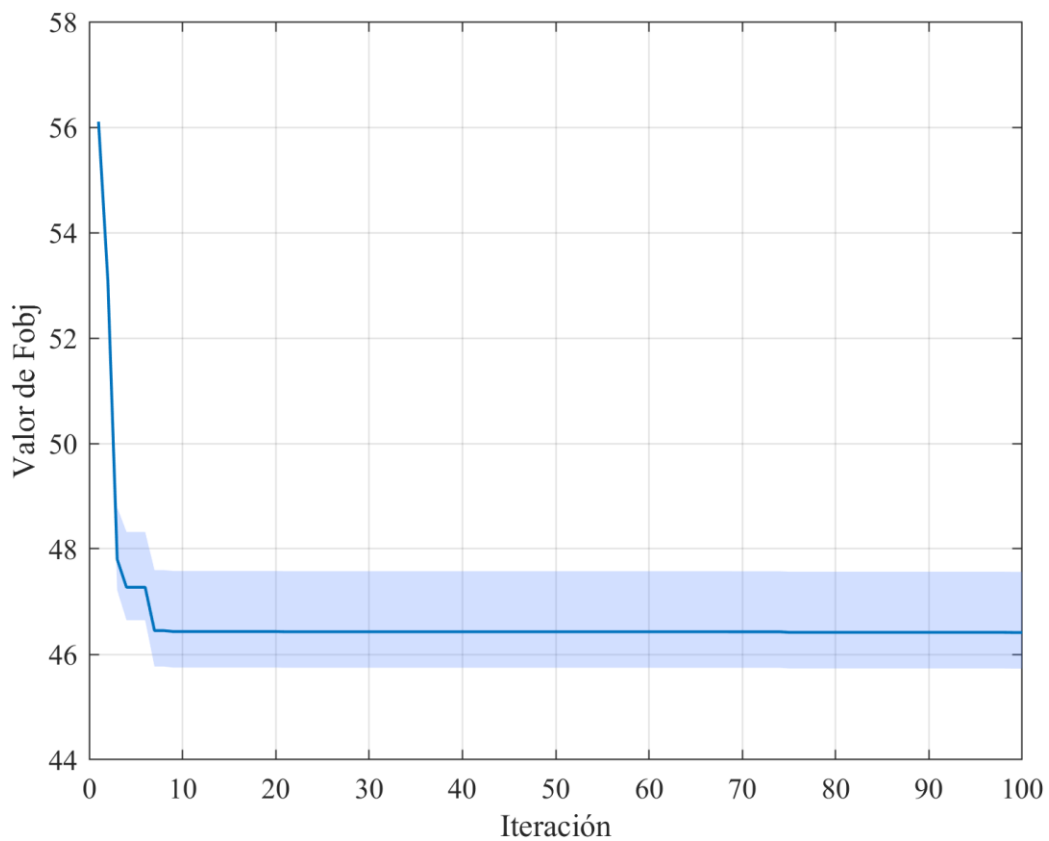


Figura 6.14 Curva de convergencia del algoritmo OPA reducido.
(Elaboración propia).

La Figura 6.15 reúne en una sola gráfica las curvas de convergencia de todas las heurísticas reducidas, permitiendo contrastar sus comportamientos. En términos de rapidez, el μ WSO y el μ OPA alcanzaron la convergencia en las primeras iteraciones, destacando por su eficiencia temporal, aunque con una dispersión mayor a la de sus versiones canónicas. El μ WOA se comportó de manera más gradual, estabilizándose de forma tardía, pero alcanzando uno de los valores finales más bajos. El μ PSO y el μ SFO mostraron un desempeño intermedio, pero con una variabilidad acotada que respalda su consistencia.

En contraste, el μ BWO presentó la mayor dispersión entre ejecuciones y el de menor eficacia en la minimización de la función objetivo. En conjunto, estos resultados permiten concluir que las versiones reducidas mantienen la capacidad de aproximarse a soluciones competitivas, aunque con un compromiso más equilibrado entre rapidez de convergencia, estabilidad y calidad de la solución alcanzada respecto de sus contrapartes canónicas.

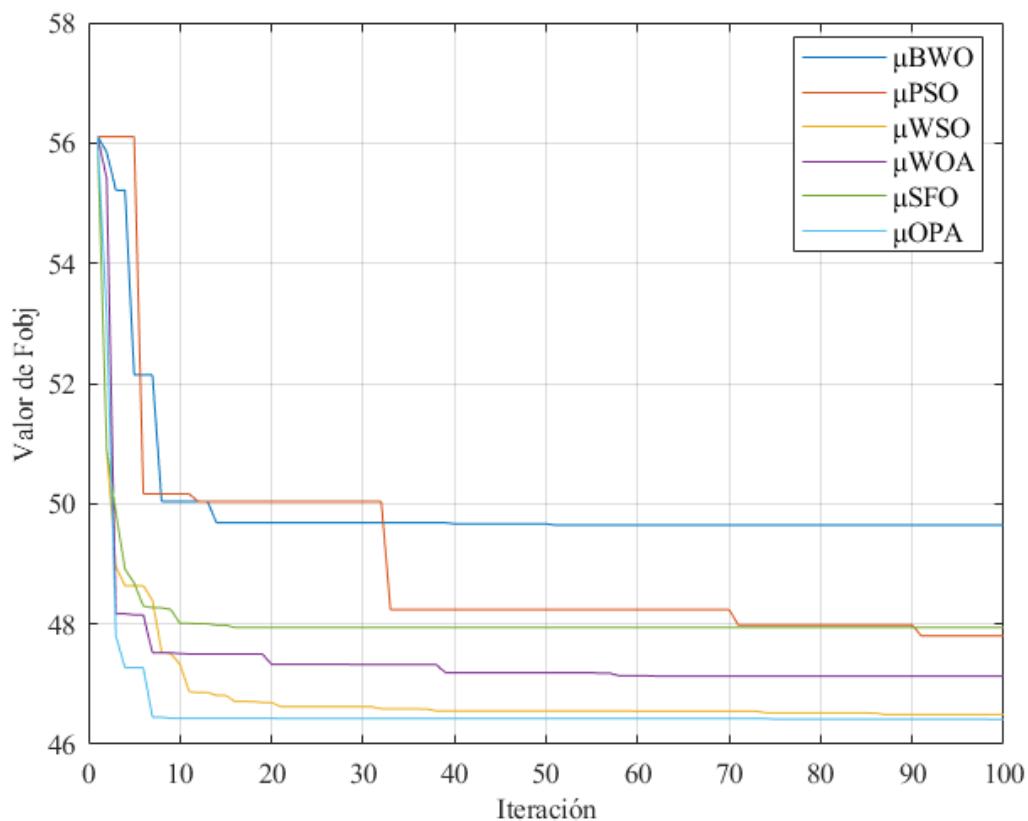


Figura 6.15 Comparativa de convergencia de los algoritmos reducidos. (Elaboración propia).

6.4 Sintonización del LQR con heurísticas reducidas

La Figura 6.16 muestra un ejemplo de la respuesta al escalón unitario del sistema cuando el LQR fue sintonizado con las heurísticas reducidas, con base en los valores de Q y R obtenidos a partir de la mediana de sus ejecuciones. Para evitar redundancia, solo se muestra el μ WSO, que alcanzó el mejor desempeño en términos de la función objetivo, y el μ BWO, que presentó uno de los valores más altos. Al igual que en el caso de las heurísticas canónicas, las trayectorias se superponen de manera casi total, evidenciando que las diferencias entre algoritmos son imperceptibles en la dinámica del sistema. Tanto en ψ como en θ , la salida sigue a la referencia con tiempos de establecimiento de alrededor de 12 y 10 segundos, respectivamente y errores estacionarios prácticamente nulos.

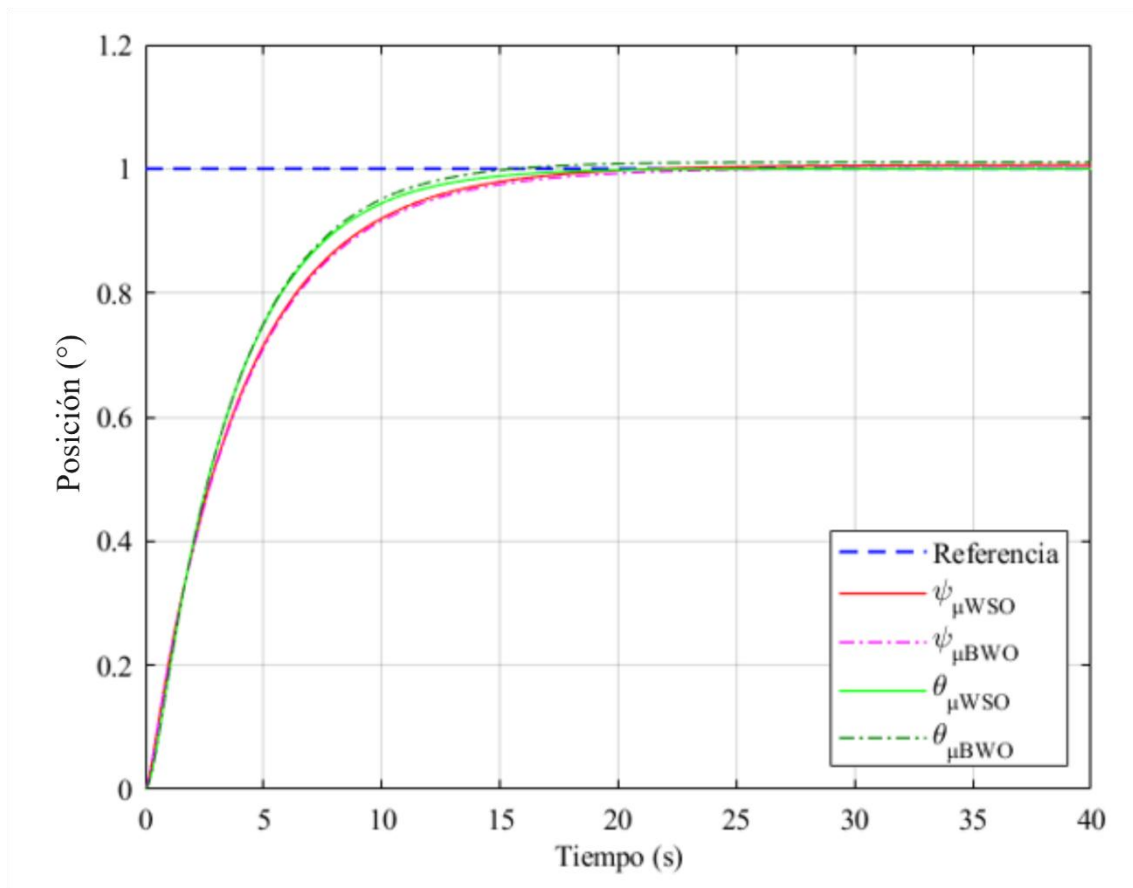


Figura 6.16 Respuesta al escalón del helicóptero con el LQR sintonizado mediante la heurística reducida de mejor desempeño (μ WSO) y la de menor desempeño (μ BWO). (Elaboración propia).

No obstante, al complementar este análisis con las métricas de la Tabla 6.5, se observa que el μ WSO logra índices integrales (IAE , ISE , $ITAE$ e $ITSE$) más favorables que el μ BWO, lo que confirma su mayor eficiencia en la sintonización. El μ BWO, aunque alcanza una respuesta estable y adecuada, muestra un valor de la función objetivo más elevado y una mayor variabilidad relativa entre ejecuciones.

Tabla 6.5 Criterios de desempeño del LQR sintonizado con heurísticas reducidas.
(Elaboración propia).

Criterio		μ PSO	μ BWO	μ WSO	μ WOA	μ SFO	μ OPA	
t_s	ψ	12.5000	12.3000	12.3000	12.1000	12.5000	12.2000	
	θ	10.500	10.6000	10.3000	10.2000	10.5000	10.3000	
e_{ss}	ψ	0.0066	0.0017	0.0006	0.0046	0.0065	0.0069	
	θ	0.0018	0.0085	0.0002	0.0014	0.0004	0.0005	
y_{ss}	ψ	1.0066	1.0017	1.0006	1.0046	1.0065	1.0069	
	θ	1.0018	1.0085	0.9998	1.0014	0.9995	1.0005	
f_{obj}	IAE	ψ	4.0965	4.0736	4.0457	3.9929	4.1216	4.0223
		θ	3.6913	3.7621	3.6603	3.6114	3.7198	3.6268
	ISE	ψ	2.0666	2.0676	2.0431	2.0216	2.0793	2.0270
		θ	1.9880	1.9753	1.9765	1.9509	2.0008	1.9617
	$ITAE$	ψ	18.1770	16.6303	17.6182	16.7692	17.3352	16.7775
		θ	13.9530	16.3973	12.2146	12.2962	12.7996	12.0813
	$ITSE$	ψ	4.0959	4.1132	3.9998	3.9145	4.1499	3.9339
		θ	3.4549	3.4180	3.4181	3.3132	3.5138	3.3573
	Σf_{obj}		51.5232	52.4374	46.9767	47.8699	49.7200	47.7881

La Tabla 6.6 presenta las matrices Q y R obtenidas a partir de la mediana de las 40 ejecuciones realizadas con cada heurística reducida. Aunque la respuesta del sistema al escalón unitario resultó prácticamente indistinguible entre algoritmos, los valores de estas matrices evidencian que cada técnica explora diferentes regiones del espacio de búsqueda y propone configuraciones de ponderación particulares. Estas diferencias numéricas permiten observar que, incluso cuando la dinámica del LQR es estable y muy similar, las heurísticas no convergen hacia la misma solución, lo cual refleja la naturaleza estocástica de estas técnicas de optimización.

Tabla 6.6 Matrices solución sintonizadas con heurísticas reducidas.
(Elaboración propia).

Algoritmo	Matriz Q	Matriz R
PSO	$\begin{bmatrix} 4.1671 & 0 & 0 & 0 \\ 0 & 7.6857 & 0 & 0 \\ 0 & 0 & 3.9237 & 0 \\ 0 & 0 & 0 & 4.3695 \end{bmatrix}$	$\begin{bmatrix} 5.9762 & 0 \\ 0 & 5.9612 \end{bmatrix}$
BWO	$\begin{bmatrix} 1.8104 & 0 & 0 & 0 \\ 0 & 3.2994 & 0 & 0 \\ 0 & 0 & 1.5146 & 0 \\ 0 & 0 & 0 & 1.6625 \end{bmatrix}$	$\begin{bmatrix} 2.5767 & 0 \\ 0 & 2.5970 \end{bmatrix}$
WSO	$\begin{bmatrix} 5.2343 & 0 & 0 & 0 \\ 0 & 9.7531 & 0 & 0 \\ 0 & 0 & 2.3481 & 0 \\ 0 & 0 & 0 & 3.6277 \end{bmatrix}$	$\begin{bmatrix} 7.5225 & 0 \\ 0 & 7.5751 \end{bmatrix}$
WOA	$\begin{bmatrix} 3.6707 & 0 & 0 & 0 \\ 0 & 6.7947 & 0 & 0 \\ 0 & 0 & 0.2440 & 0 \\ 0 & 0 & 0 & 0.0053 \end{bmatrix}$	$\begin{bmatrix} 5.2596 & 0 \\ 0 & 5.2609 \end{bmatrix}$
SFO	$\begin{bmatrix} 4.3518 & 0 & 0 & 0 \\ 0 & 8.0540 & 0 & 0 \\ 0 & 0 & 5.0817 & 0 \\ 0 & 0 & 0 & 5.6981 \end{bmatrix}$	$\begin{bmatrix} 6.2383 & 0 \\ 0 & 6.2193 \end{bmatrix}$
OPA	$\begin{bmatrix} 4.3563 & 0 & 0 & 0 \\ 0 & 8.1194 & 0 & 0 \\ 0 & 0 & 0.8719 & 0 \\ 0 & 0 & 0 & 1.2766 \end{bmatrix}$	$\begin{bmatrix} 6.2750 & 0 \\ 0 & 6.2920 \end{bmatrix}$

6.5 Comparativa de desempeño de las heurísticas canónicas frente a las reducidas

Para contrastar el desempeño de las versiones canónicas y reducidas de las heurísticas empleadas, se analizan de manera conjunta las curvas de convergencia, los valores de la función objetivo alcanzados y la dispersión entre ejecuciones. Adicionalmente, se consideran métricas de costo computacional, expresadas en términos del número de iteraciones y el tamaño de la población utilizada, con el fin de evaluar la relación entre eficiencia y calidad de la solución.

La Figura 6.17 muestra la comparación entre las curvas de convergencia del PSO canónico y su versión reducida. El algoritmo canónico alcanza la convergencia alrededor de la iteración 30, estabilizándose en un valor final cercano a 46.20. En contraste, el μ PSO requiere aproximadamente 91 iteraciones para converger y, aunque logra una reducción significativa, se mantiene en un valor final mayor (47.80). Este comportamiento refleja que, para esta heurística, la reducción en el tamaño de la población impacta en el proceso de búsqueda y afecta la calidad de la solución final.

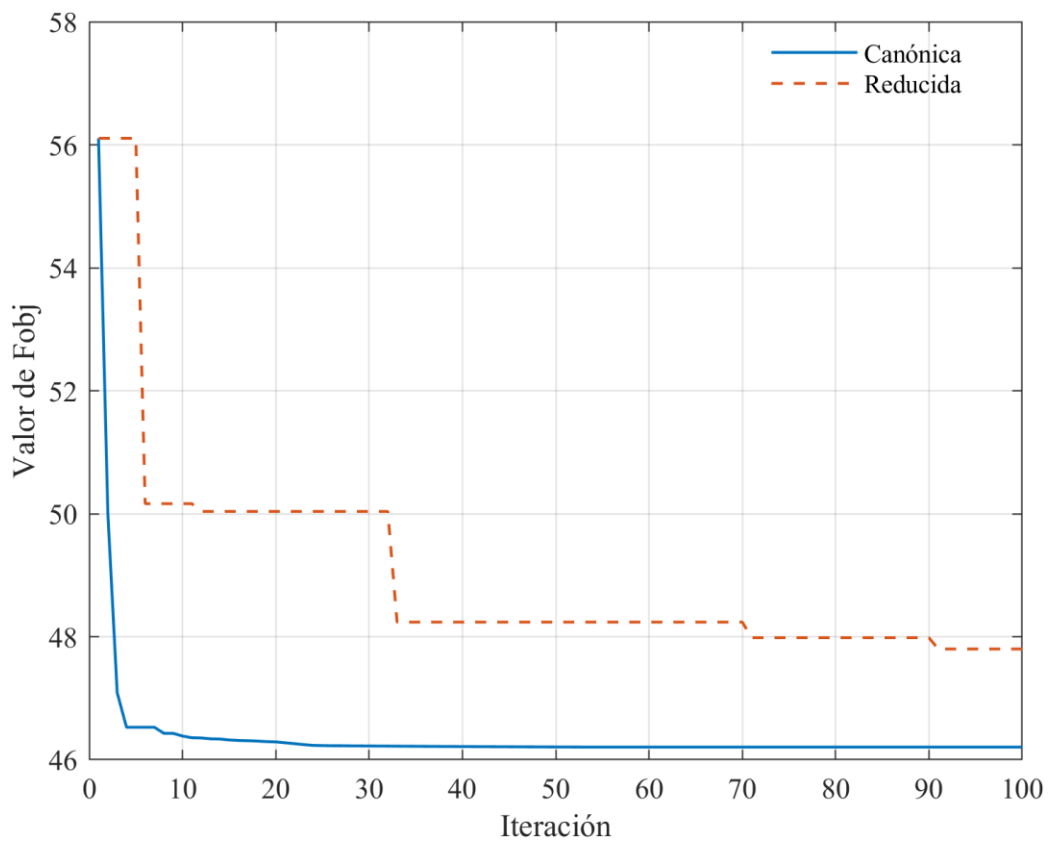


Figura 6.17 Comparativa de curvas de convergencia del PSO vs μ PSO.
(Elaboración propia).

En la Figura 6.18 se aprecia la comparación entre el BWO y su versión reducida. El BWO converge en torno a la iteración 60 con un valor final de 45.69, mientras que el μ BWO se estabiliza antes, aproximadamente en la iteración 50, aunque con un valor final mayor de 49.64. A pesar de esta diferencia, la curva del μ BWO refleja que incluso con un tamaño poblacional mínimo puede mantener la capacidad de encontrar soluciones viables en la sintonización de las matrices de ponderación del LQR.

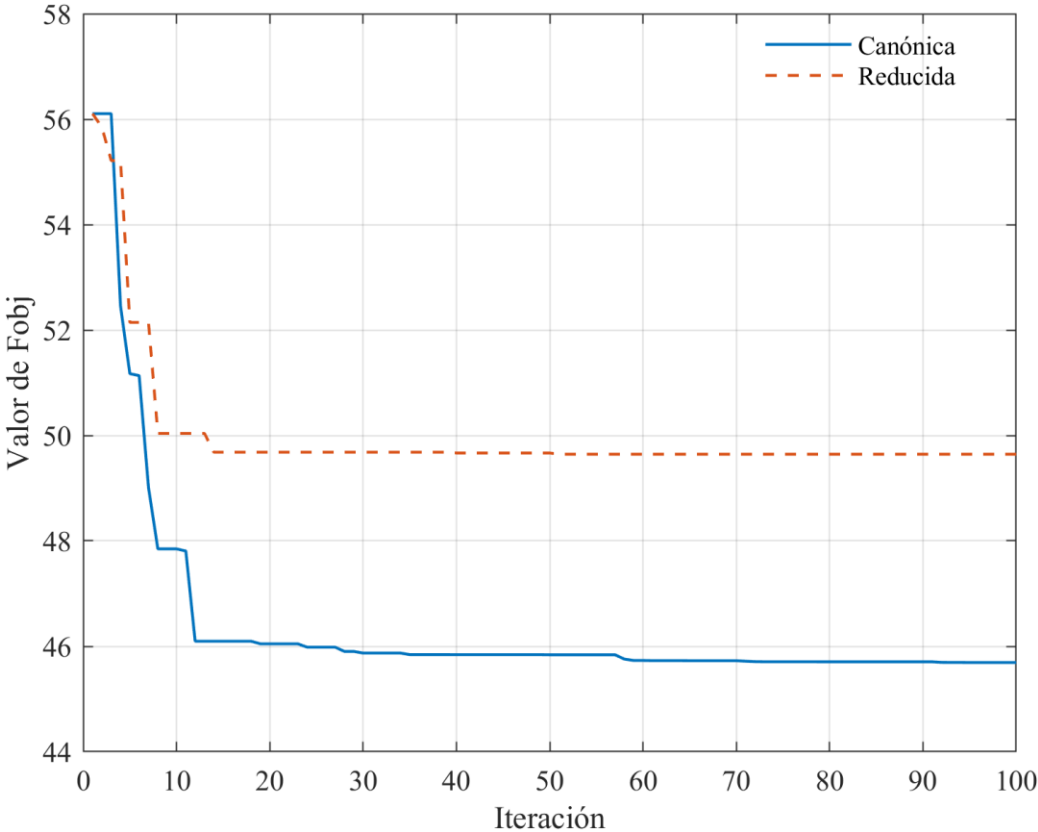


Figura 6.18 Comparativa de curvas de convergencia del BWO vs μ BWO. (Elaboración propia).

La Figura 6.19 muestra las curvas del WSO y el μ WSO. El WSO converge rápidamente hacia la iteración 10 alcanzando un valor final de 45.63, mientras que el μ WSO requiere alrededor de 37 iteraciones para estabilizarse en un valor final de 46.49. Aunque la versión reducida demanda más iteraciones y no logra igualar el desempeño de la versión completa, su trayectoria confirma que sigue siendo efectiva para aproximarse a soluciones de buena calidad con un costo computacional considerablemente menor debido a su número reducido de agentes de búsqueda.

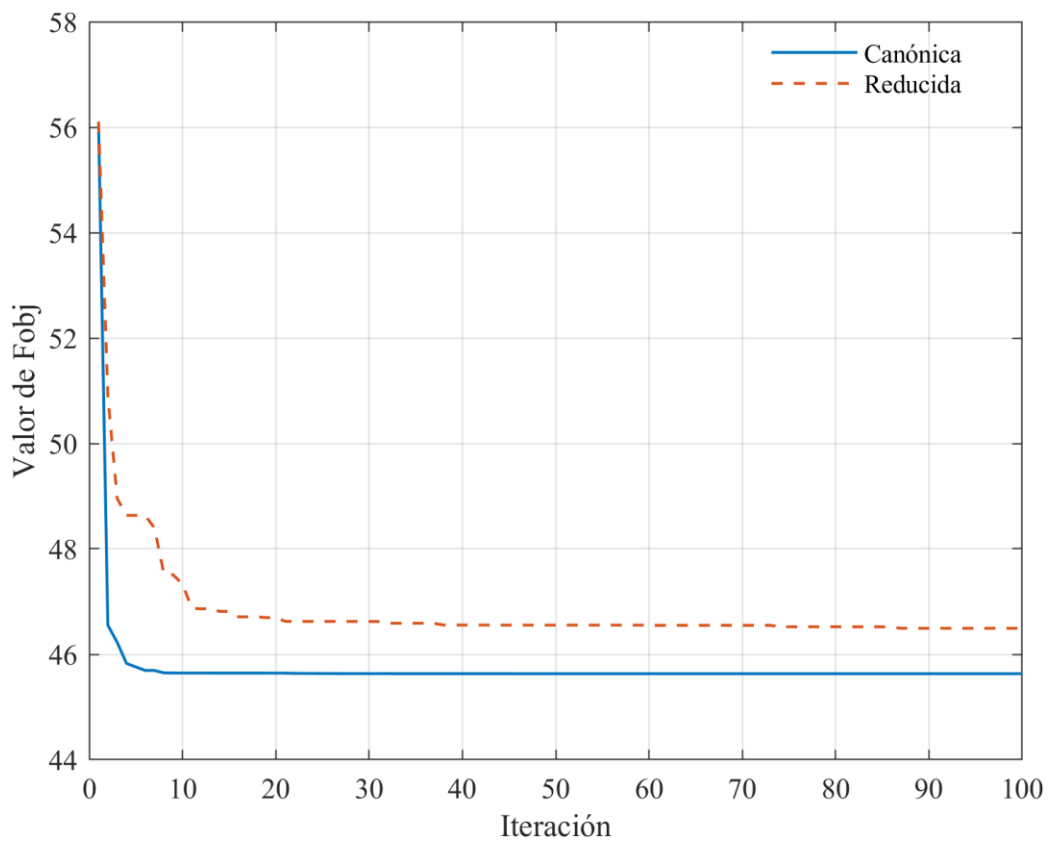


Figura 6.19 Comparativa de curvas de convergencia del WSO vs μ WSO.
(Elaboración propia).

En la Figura 6.20 se comparan las curvas de convergencia del WOA y el μ WOA. El WOA se caracteriza por un descenso gradual, alcanzando la convergencia cerca de la iteración 80 con un valor final de 45.68, mientras que el μ WOA logra estabilizarse mucho antes, alrededor de la iteración 21, con un valor final de 47.13. Aunque su desempeño en términos de función objetivo es menos preciso, la rapidez en la convergencia y el ahorro en recursos computacionales posicionan al μ WOA como una alternativa práctica en la sintonización del LQR.

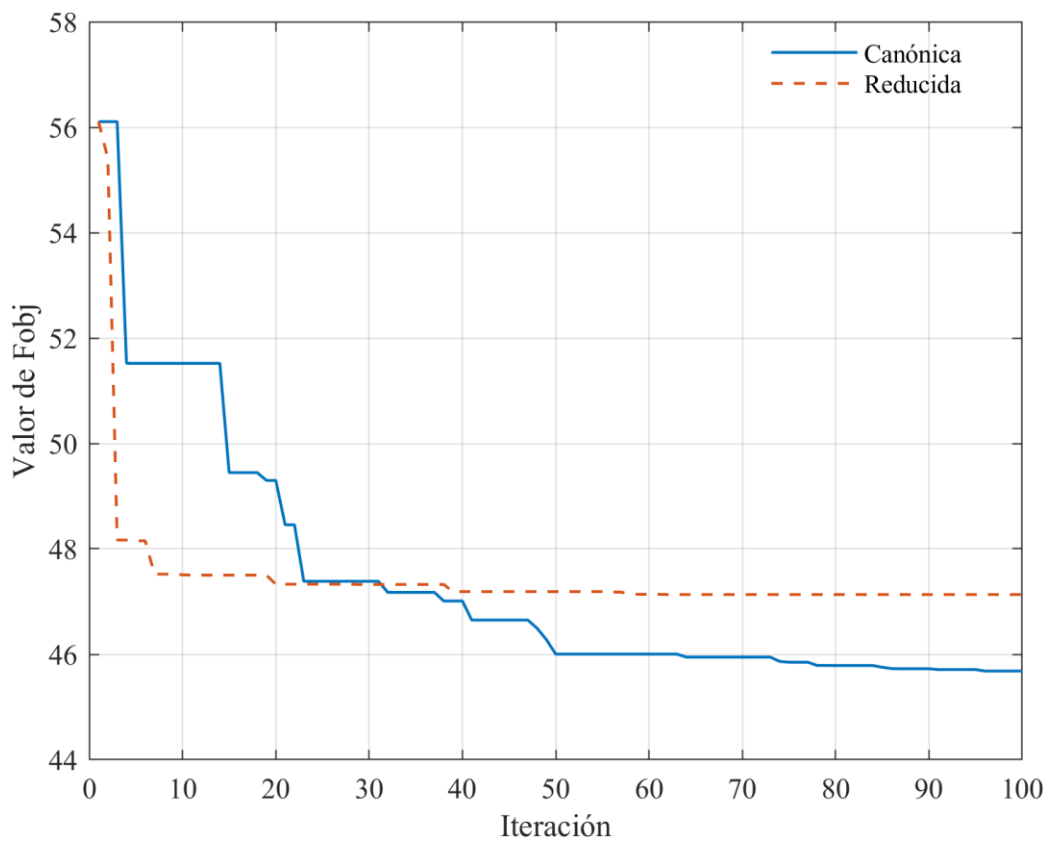


Figura 6.20 Comparativa de curvas de convergencia del WOA vs μ WOA.
(Elaboración propia).

La Figura 6.21 ilustra el comportamiento del SFO frente a su versión reducida. El SFO converge de manera temprana, en torno a la iteración 13, alcanzando un valor final de 48.35; en contraste, el μ SFO converge aún más rápido, en la iteración 10, con un valor final ligeramente mejor de 47.94. En este caso particular, la versión reducida no solo mantiene un buen nivel de desempeño, sino que incluso supera marginalmente al algoritmo original en el valor final de la función objetivo, confirmando el potencial de las micro poblaciones en ciertos escenarios.

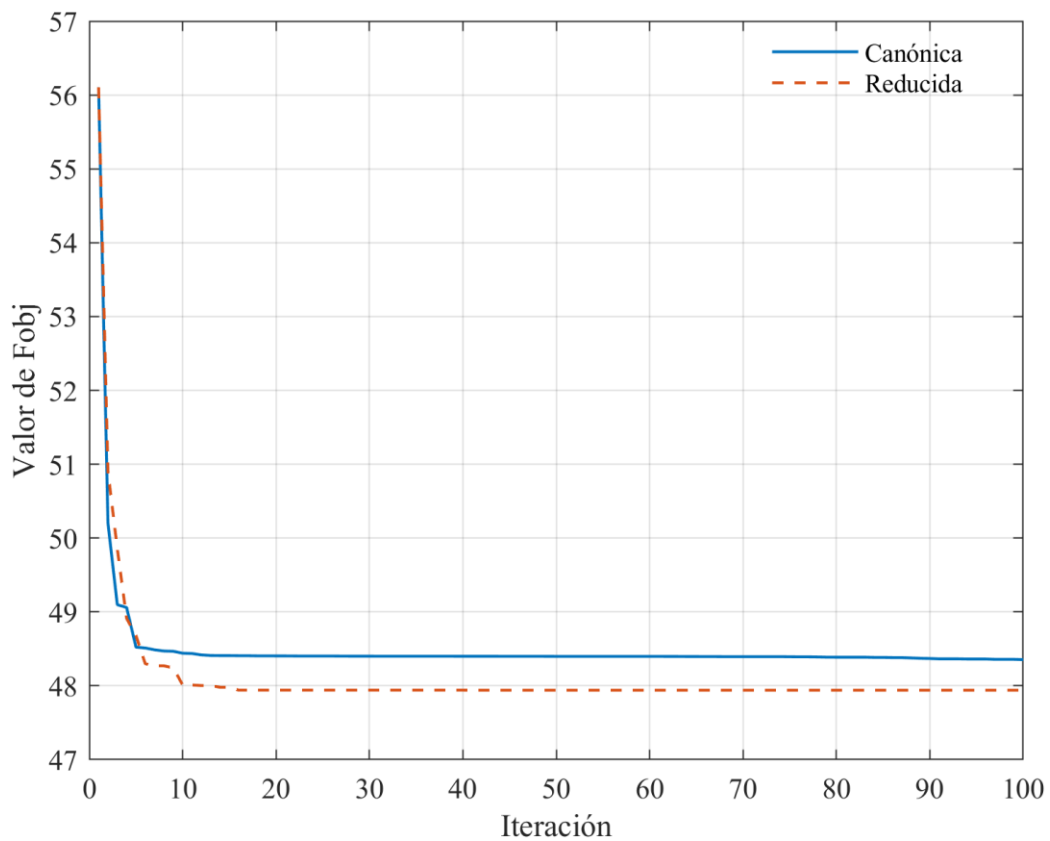


Figura 6.21 Comparativa de curvas de convergencia del SFO vs μ SFO.
(Elaboración propia).

Finalmente, la Figura 6.22 presenta la comparación entre el OPA y el μ OPA. El OPA alcanza convergencia muy rápida alrededor de la iteración 10 con un valor final de 45.63, mientras que el μ OPA lo hace todavía antes, en la iteración 7, aunque con un valor final ligeramente mayor de 46.41. Si bien la versión reducida no logra igualar del todo la precisión de la canónica, su rapidez en la convergencia y la simplicidad en el proceso de búsqueda confirman que constituye una alternativa eficiente en términos de costo computacional y tiempo de ejecución.

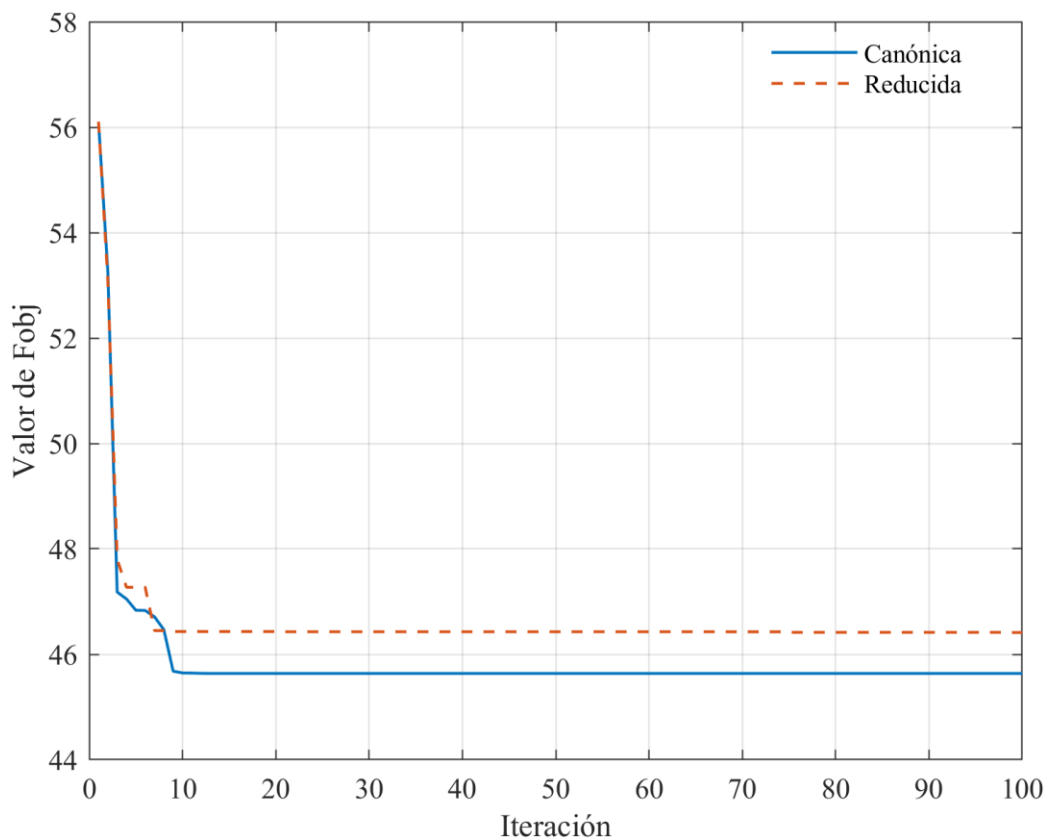


Figura 6.22 Comparativa de curvas de convergencia del OPA vs μ OPA.
(Elaboración propia).

Las métricas de la Tabla 6.7 muestran que, en términos de calidad de la solución, las versiones canónicas tienden a obtener valores de la función objetivo ligeramente mejores y con una variabilidad significativamente menor. En contraste, las versiones reducidas alcanzan valores finales competitivos, aunque con mayor dispersión y, en algunos casos, con un número superior de iteraciones para lograr la convergencia. Sin embargo, el costo computacional asociado a las versiones reducidas es mucho menor, pues emplean poblaciones de solo 5 agentes frente a los 100 agentes de las canónicas, lo que representa una reducción significativa en la cantidad de evaluaciones requeridas y los tiempos de ejecución referentes a la fase de sintonización del LQR.

Tabla 6.7 Comparación entre heurísticas canónicas y reducidas.
(Elaboración propia).

	PSO	BWO	WSO	WOA	SFO	OPA
<i>f_{obj}</i> (canónica)	46.2050	45.6915	45.6320	45.6825	48.3540	45.6360
<i>f_{obj}</i> (reducida)	49.6440	47.8005	46.4925	47.1335	47.9400	46.4130
<i>IQR</i> (canónica)	1.2015	0.3025	0.0015	0.0730	1.3945	0.0010
<i>IQR</i> (reducida)	3.5020	1.0260	1.2495	1.4970	1.0390	1.8350
No. iteraciones convergencia (canónica)	30	60	10	80	13	10
No. iteraciones convergencia (reducida)	91	50	37	21	10	7
No. de agentes (canónica)	100	100	100	100	100	100
No. de agentes (reducida)	5	5	5	5	5	5
No. evaluaciones realizadas (canónica)	3000	6000	1000	8000	1300	1000
No. evaluaciones realizadas (reducida)	455	250	185	105	50	35
% evaluaciones relativas	15.16%	4.16%	18.5%	1.31%	3.8%	3.5%
Tiempo ejecución (canónica)	30.29 s	53.67 s	58.60 s	51.94 s	43.05 s	54.36 s
Tiempo ejecución (reducida)	12.27 s	12.40 s	11.68 s	11.77 s	11.45 s	11.38 s

Para complementar la comparación entre las versiones canónicas y reducidas, la Figura 6.23 muestra la respuesta al escalón unitario obtenida al sintonizar el LQR con la mejor heurística canónica (WOA) y con la mejor reducida (μ WSO). La superposición de ambas curvas permite observar que las diferencias son mínimas, ya que en ambos casos el sistema alcanza la referencia con tiempos de establecimiento similares y con errores estacionarios prácticamente nulos.

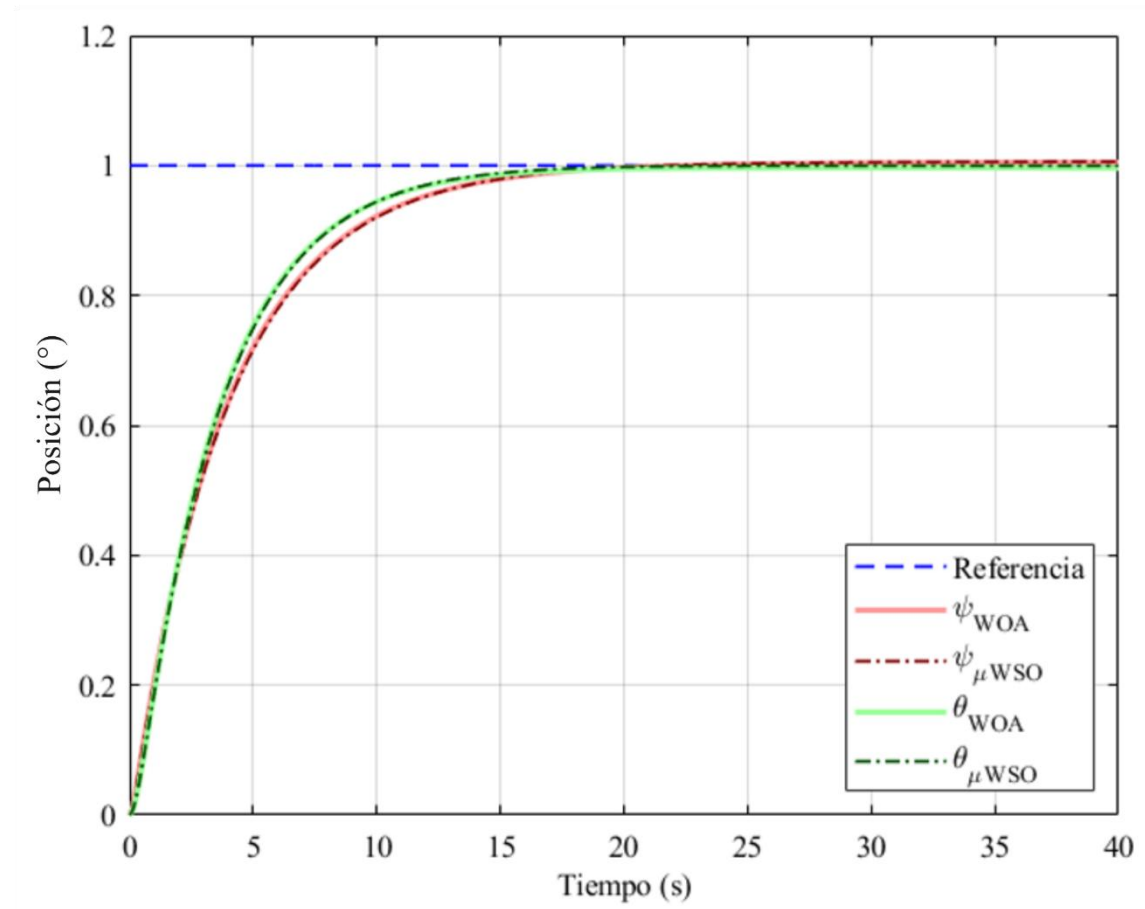


Figura 6.23 Respuesta al escalón del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO). (Elaboración propia).

La Figura 1.1Figura 6.24 muestra la respuesta del helicóptero con el LQR sintonizado mediante WOA y μ WSO frente a una señal de referencia tipo pulso que alterna entre 1 y 0. Se observa que, en ambos casos, el sistema es capaz de seguir adecuadamente los cambios abruptos, manteniendo estabilidad y sin presentar sobreimpulsos significativos, demostrando la capacidad del sistema para enfrentar entradas más complejas que el escalón unitario.

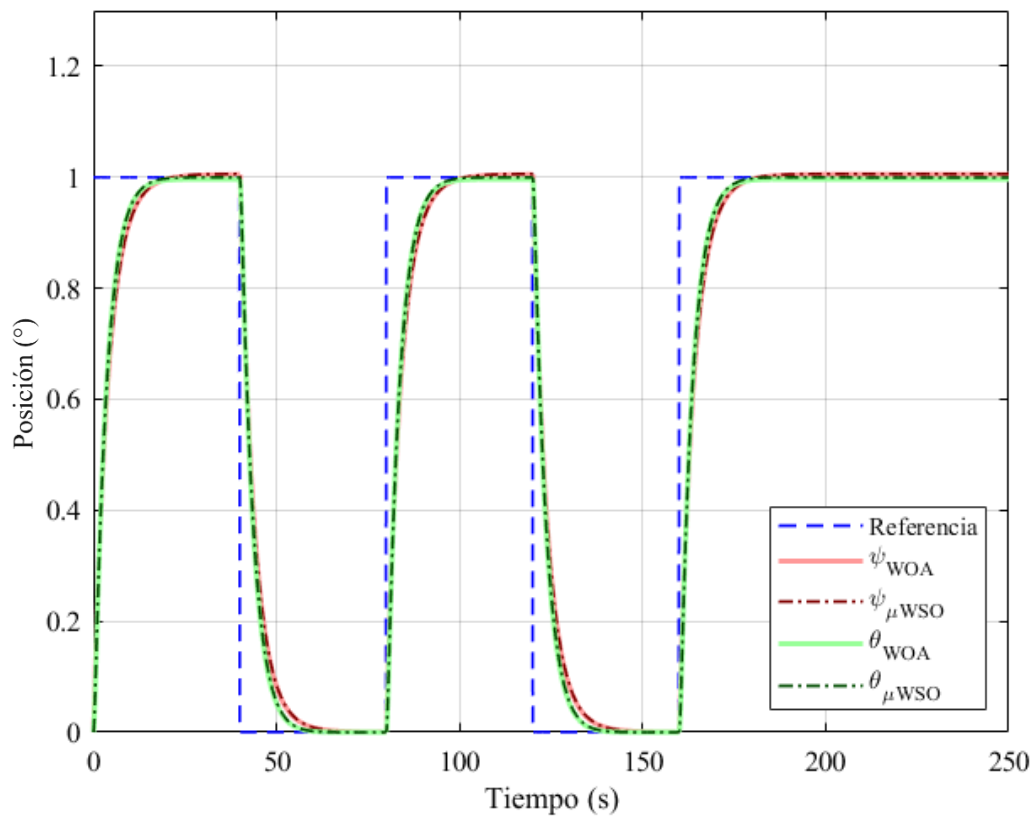


Figura 6.24 Respuesta del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO) frente a una referencia tipo pulso. (Elaboración propia).

La Figura 6.25 presenta la respuesta del helicóptero con el LQR bajo la misma configuración frente a una referencia en forma de escalera creciente y decreciente. Tanto en ψ como en θ , el sistema sigue de manera fiel la trayectoria de la referencia, evidenciando la robustez del diseño y la similitud en el desempeño de las versiones canónica y reducida de las heurísticas.

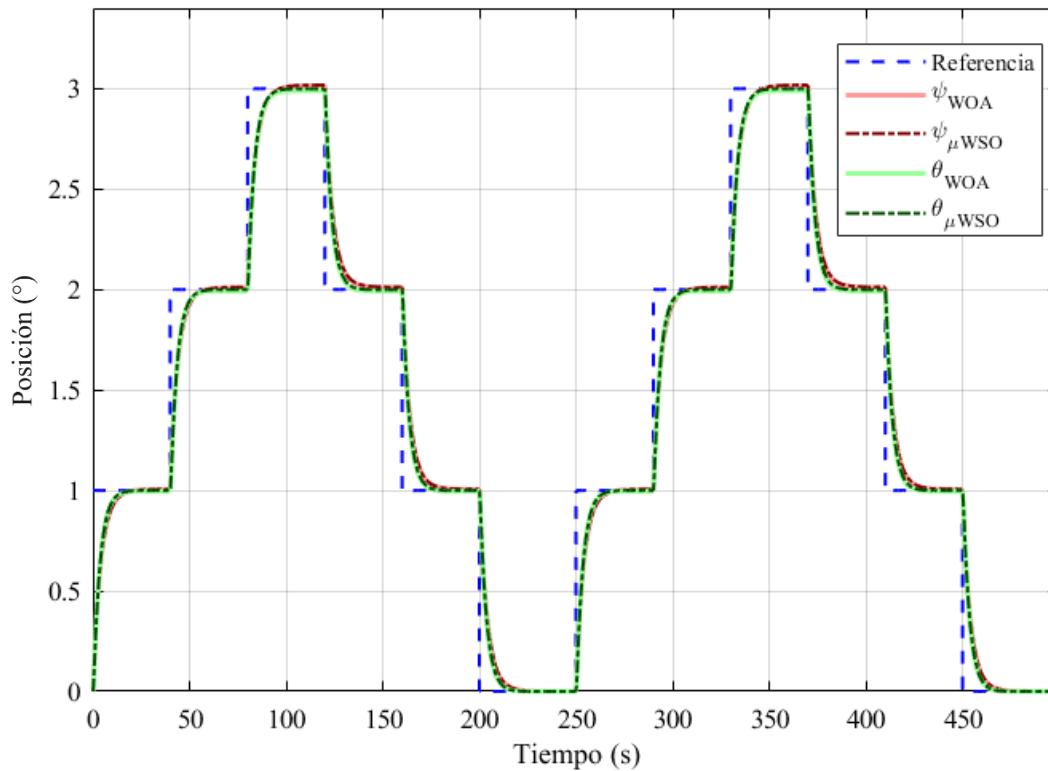


Figura 6.25 Respuesta del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO) frente a una referencia tipo escalera. (Elaboración propia).

La Figura 6.26 muestra la respuesta del helicóptero con el LQR sintonizado mediante WOA y μ WSO ante la aplicación de perturbaciones negativas y positivas en diferentes instantes de tiempo y de diferente magnitud. Se observa que, el regulador es capaz de compensar dichas perturbaciones de manera efectiva, logrando que el sistema regrese rápidamente a la referencia deseada. Este resultado evidencia la capacidad del controlador no solo para seguir referencias predeterminadas, sino también para mantener un comportamiento robusto ante señales inesperados.

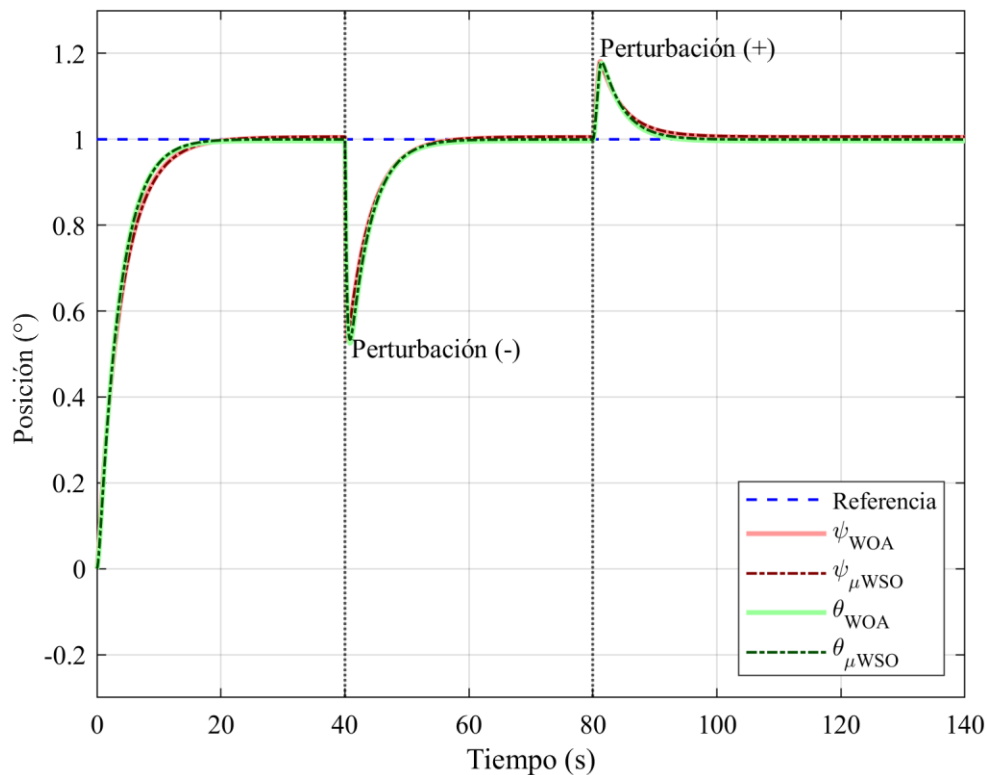


Figura 6.26 Respuesta del helicóptero con el LQR sintonizado mediante la mejor heurística canónica (WOA) y la mejor reducida (μ WSO) ante perturbaciones. (Elaboración propia).

Los resultados ponen de manifiesto que, si bien las versiones reducidas no alcanzan el mismo nivel de precisión en la optimización de la función objetivo que sus contrapartes canónicas, son capaces de mantener un desempeño satisfactorio en la dinámica del helicóptero. Esto adquiere especial relevancia al considerar que la simulación en tiempo continuo del sistema resulta computacionalmente costosa; por lo tanto, disponer de versiones reducidas que disminuyen de manera significativa el esfuerzo de cálculo sin afectar de forma perceptible la calidad del control constituye una alternativa práctica y eficiente para este tipo de problemas.

7 CONCLUSIONES

El trabajo desarrollado permitió establecer un modelo matemático del helicóptero pivotado y su correspondiente controlador LQR, sobre el cual se evaluaron distintos esquemas de optimización bioinspirada. Las simulaciones confirmaron que los algoritmos empleados fueron capaces de generar ganancias de control que aseguran estabilidad, seguimiento adecuado de la referencia y márgenes de error en estado estable prácticamente nulos. Con ello se demostró la pertinencia del enfoque de optimización heurística para la sintonización de reguladores óptimos.

La aplicación de seis algoritmos metaheurísticos en sus versiones canónicas y reducidas mostró que, en todos los casos, se obtuvieron soluciones viables para el problema de regulación. No obstante, las variantes reducidas presentaron una ventaja clara al alcanzar desempeños equiparables con un costo computacional significativamente menor en el proceso de sintonización del controlador. Esta eficiencia se evidenció en la reducción del número de agentes y del total de iteraciones requeridas para la convergencia, sin comprometer la calidad de la respuesta dinámica del sistema. Dicho resultado valida la utilidad de los mecanismos de reducción planteados como una alternativa sólida.

En términos de evaluación, las métricas basadas en la mediana, esto dado por el comportamiento sesgado de la distribución de los datos de la función objetivo, el rango intercuartil y la velocidad de convergencia permitieron caracterizar con precisión el comportamiento de cada algoritmo. Estas medidas confirmaron que las versiones reducidas no sólo mantienen la estabilidad de la solución, sino que también favorecen una convergencia eficiente aunque con mayor dispersión de resultados.

Un aspecto central de esta investigación radica en su relevancia científica, pues introduce el concepto de heurísticas reducidas de Inteligencia de Enjambres aplicadas a la sintonización de LQRs, un enfoque poco explorado en la literatura. Mientras que la mayoría de los trabajos previos se han centrado en la aplicación directa de versiones canónicas de metaheurísticas, aquí se demuestra que es posible mantener la calidad de las soluciones disminuyendo de manera sustancial el costo computacional.

Desde la perspectiva de las implicaciones prácticas, la reducción del costo computacional lograda mediante micropoblaciones y menor número de iteraciones abre la posibilidad de implementar este tipo de algoritmos en plataformas con limitaciones de hardware, tales como sistemas embebidos. En este sentido, la propuesta no sólo aporta en términos académicos, sino que también sienta bases para la transferencia tecnológica hacia sistemas de control prácticos y escalables.

Cabe señalar ciertas limitaciones del estudio que enmarcan el alcance de los resultados. La validación se realizó exclusivamente mediante simulaciones en MATLAB® sobre un modelo matemático del helicóptero pivotado, sin realizar experimentación física en un prototipo. Esto, principalmente debido al carácter computacional de la investigación, orientado a la solución de problemas de optimización numérica que a la Ingeniería de control práctica. Por lo tanto, aunque los hallazgos confirman la eficacia de las heurísticas reducidas en entornos simulados, futuras investigaciones deberán considerar pruebas físicas para fortalecer la aplicabilidad del enfoque.

En conjunto, los hallazgos corroboran que la regulación de la posición del helicóptero pivotado puede realizarse de forma satisfactoria a través de la sintonización de un LQR con heurísticas reducidas, alcanzando tiempos de respuesta aceptables y reduciendo sustancialmente el esfuerzo computacional frente a las implementaciones canónicas. De esta manera, se valida la premisa central de la investigación y se aporta evidencia de que las versiones reducidas constituyen una alternativa competitiva y eficiente para el diseño de controladores basados en inteligencia de enjambres.

Para continuar con la línea de generación y aplicación del conocimiento desarrollada durante este proceso investigativo se propone como área de oportunidad probar nuevas estrategias de reducción tales como el ajuste dinámico de poblaciones y parámetros o la selección adaptativa de operadores y de criterios de paro, también se puede plantear la aplicación de heurísticas emergentes como el Sand Kangaroo Optimization Algorithm, Cuckoo Catfish Optimizer, Blood-Sucking Leech Optimizer, Channa Argus Optimizer o el Ivy Algorithm; y la extensión hacia esquemas distintos al LQR como control en Modo Deslizante o Lógica Difusa.

BIBLIOGRAFÍA

- [1] H. Desai, S. Akash. y S. Kumar-Choudhary, «Modelling and Control of 3-DOF Helicopter,» *International Journal for Research in Applied Science & Engineering Technology*, vol. 8, nº 5, pp. 1325-1331, 2020.
- [2] J. Vishnupriyan, P. Manoharan y A. Ramalakshmi, «Uncertainty Modeling of Nonlinear 2-DOF Helicopter Model,» de *International Conference on Computer Communication and Informatics*, Coimbatore, India, 2014.
- [3] S. I. Abdelmaksoud, M. Mailah y A. M. Abdallah, «Control Strategies and Novel Techniques for Autonomous Rotorcraft Unmanned Aerial Vehicles: A Review,» *IEEE Access*, vol. 8, pp. 195142-195169, 2020.
- [4] E. Gonzalez-Galindo, E. Bugarin y A. Y. Aguilar-Bustos, «Control par-calculado para el cabeceo del helicóptero Quanser de 2 GDL,» *Revista Ingeniantes*, vol. 1, nº 1, pp. 84-89, 2015.
- [5] N. S. Al-Falooji y M. Abbod, «Helicopter Control Using Fuzzy Logic and Narma-L2 Techniques,» *International Journal of Intelligent Systems and Applications*, vol. 12, nº 5, pp. 1-14, 2020.
- [6] A. S. Dutka, A. W. Ordys y M. J. Grimble, «Non-linear predictive control of 2 DOF helicopter model,» de *IEEE International Conference on Decision and Control* , Maui, EUA, 2003.
- [7] K. K. Dandago, L. Zhang y W. Pan, «Fault Detection and Tolerant Control for Aero2 2D0F Two-rotor,» *Electrical Engineering and Systems Science*, 2024.
- [8] H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah y M. Guizani, «Unmanned Aerial Vehicles (UAVs): A Survey

- on Civil Applications and Key Research Challenges,» *IEEE Access*, vol. 7, pp. 48572-48634, 2019.
- [9] S. M. P. S. R. Bouabdallah, «Design and control of an indoor micro quadrotor,» de *IEEE International Conference on Robotics and Automation*, Los Angeles, USA, 2004.
- [10] G. Quaranta, W. Lacarbonara y S. Masri, «A review on computational intelligence for identification of nonlinear dynamical systems,» *Nonlinear Dynamics*, vol. 99, p. 1709–1761, 2020.
- [11] E. Vinodh Kumar, G. S. Raaja y J. Jerome, «Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter,» *Applied Soft Computing*, vol. 41, pp. 77-90, 2016.
- [12] D. Nodland, H. Zargarzadeh y S. Jagannathan, «Neural Network-Based Optimal Adaptive Output Feedback Control of a Helicopter UAV,» *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, nº 7, pp. 1061-1073, 2013.
- [13] G. Ma, H. Wu, Z. Zhao, Z. Tao y K. Hon, «Adaptive neural network control of a nonlinear two-degree-of-freedom helicopter system with prescribed performance,» *IET Control Theory & Applications*, vol. 17, nº 13, pp. 1789-1799, 2022.
- [14] Fahmizal, D. Y. Kharisma y S. Pramono, «Implementation of Fuzzy Logic Control on a Tower Copter,» *Journal of Fuzzy Systems and Control*, vol. 1, nº 1, pp. 14-17, 2023.
- [15] V. Chugh y J. Ohri, «GA tuned LQR and PID controller for aircraft pitch control,» de *India International Conference on Power Electronics*, Kurukshetra, India, 2015.
- [16] G. Beauchamp-Báez y R. Batista, «Aplicación de Técnicas de Control Óptimo a una plataforma estacionaria cuatrimotor,» *Revista Científica de Ingeniería Electrónica, Automática y Comunicaciones*, vol. 37, nº 3, pp. 34-49, 2016.

- [17] E. Okyere, A. Bousbaine, G. Poyi, A. Joseph y J. Andrade, «LQR controller design for quad-rotor helicopters,» *The Journal of Engineering*, vol. 2019, n° 17, pp. 4003-4007, 2019.
- [18] M. Aissa y V. Feliu-Batlle, «Reduction of the Disturbance Effects in a Twin Rotor Multi-Input Multi-Output System Based on a Modified Smith Predictor Control Scheme,» *Applied Sciences*, vol. 15, n° 10, 2025.
- [19] J. C. Murcia y L. F. Alfonso, «Modelling, Identification and Control of a 2 DOF Helicopter Prototype,» de *4th Colombian Conference on Automatic Control*, Medellín, Colombia, 2019.
- [20] T. Zhao y W. Li, «LQR-based attitude controllers design for a 3-DOF helicopter system with comparative experimental tests,» *International Journal of Dynamics and Control*, vol. 12, n° 4, pp. 1063-1072, 2024.
- [21] F. Dos Santos, F. Yukio y B. Augusto, «Sliding mode control with linear quadratic regulator augmented with integrators applied to a 2DOF helicopter,» de *Congresso Brasileiro de Automática*, Brasil, 2020.
- [22] N. Osmic, J. Velagic, S. Konjicija y A. Galijasevic, «Genetic algorithm based identification of a nonlinear 2DOF helicopter model,» de *18th Mediterranean Conference on Control and Automation*, Marruecos, 2010.
- [23] H. Suppachai, T. Sunphong, C. Sasithorn y S. Warin, «Design and implementation of PSO based LQR control for inverted pendulum through PLC,» de *IEEE/SICE International Symposium on System Integration*, Sapporo, Japan, 2016.
- [24] L. Oussama, A. Latifa, G. Imam Barket y D. Abdelhakim, «Robustness Analysis of LQR-PID Controller Based on PSO and GWO for Quadcopter Attitude Stabilization,» de *5th International Electronic Conference on Applied Sciences*, 2025.

- [25] L. Tang, N. L. Ren y F. Shawn, «Semi-active Suspension Control with PSO Tuned LQR Controller Based on MR Damper,» *International Journal of Automotive and Mechanical Engineering*, vol. 20, n° 2, pp. 10512-10522, 2023.
- [26] G. F. Franklin, J. D. Powell y A. Emami-Naeini, *Feedback control of dynamic systems*, Nueva York, Estados Unidos: Pearson, 2019.
- [27] J. Velagic y N. Osmic, «Identification and control of 2DOF nonlinear helicopter model using intelligent methods,» de *IEEE International Conference on Systems, Man and Cybernetics*, 2010.
- [28] T. Bresciani, «Modelling, Identification and Control of a Quadrotor Helicopter,» *MSc Theses*, 2008.
- [29] Quanser Inc., «Quanser 2-DOF Helicopter user and control manual,» Quanser, 2006.
- [30] E. C. González y D. M. G. E. J. Rivera, «Model and Observer-Based Controller Design for a Quanser Helicopter with Two DOF,» de *IEEE Ninth Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, México, 2012.
- [31] R. Büchi, *State Space Control, LQR and Observer: step by step introduction, with Matlab examples*, Zúrich, Suiza: ETH, 2010.
- [32] N. S. Nise, *Control systems engineering*, Hoboken, Nueva Jersey: Wiley, 2011.
- [33] G. C. Goodwin, S. F. Graebe, M. E. Salgado y G. C. Goodwin, *Control system design*, Upper Saddle River, Nueva Jersey: Prentice Hall, 2001.
- [34] A. Chakraborty y A. K. Kar, «Swarm Intelligence: A Review of Algorithms,» de *Nature-Inspired Computing and Optimization*, Springer, 2017, pp. 475-495.
- [35] A. E. Hassanien y E. Emary, *Swarm Intelligence: Principles, Advances, and Applications*, Boca Raton: CRC Press, 2018.

- [36] J. Kennedy y R. Eberhart, «Particle Swarm Optimization,» de *ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995.
- [37] A. Safari, «A PSO Procedure for a Coordinated Tuning of Power System Stabilizers for Multiple Operating Conditions,» *Journal of Applied Research and Technology*, vol. 11, n° 5, pp. 665-673, 2013.
- [38] S. Mirjalili y A. Lewis, «The Whale Optimization Algorithm,» *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.
- [39] M. Nadimi-Shahraki, H. Zamani y Z. M. ., S. Asghari Varzaneh, «A Systematic Review of the Whale Optimization Algorithm: Theoretical Foundation, Improvements, and Hybridizations,» *Archives of Computational Methods in Engineering*, vol. 30, pp. 4113-4159, 2023.
- [40] F. S. Gharehchopogh y H. Gholizadeh, «A comprehensive survey: Whale Optimization Algorithm and its applications,» *Swarm and Evolutionary Computation*, vol. 48, pp. 1-24, 2019.
- [41] F. Gopmandal y A. Ghosh, «LQR-Based MIMO PID Control of a 2-DOF Helicopter System with Uncertain Cross-Coupled Gain,» *IFAC-PapersOnLine*, vol. 55, n° 22, pp. 183-188, 2022.
- [42] M. H. Nasru, I. Musirin, M. K. M. Zamani, S. R. A. Rahim, M. H. Hussain y H. Suyono, «Loss Minimization in Optimal Reactive Power Planning (ORPP) Based on Whale Optimization Algorithm,» *International Journal of Engineering and Technology*, vol. 7, n° 3, pp. 192-196, 2018.
- [43] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar y M. Awadallah, «White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems,» *Knowledge-Based Systems*, vol. 243, n° 108457, pp. 1-29, 2022.

- [44] B. Guo, Y. Wei, Q. Luo y Y. Zhou, «White Shark Optimization for Solving Workshop Layout Optimization Problem,» *Biomimetics*, vol. 10, n° 5, pp. 1-21, 2024.
- [45] M. Farhat, S. Kamel, M. A. Elseify y A. Y. Abdelaziz, «A modified white shark optimizer for optimal power flow considering uncertainty of renewable energy sources,» *Scientific Reports*, vol. 14, n° 1, p. 3051, 2024.
- [46] S. Shadravan, H. R. Naji y V. K. Bardsiri, «The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems,» *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 20-34, 2019.
- [47] Y. Zhang y Y. Mo, «Dynamic Optimization of Chemical Processes Based on Modified Sailfish Optimizer Combined with an Equal Division Method,» *Dynamic Optimization of Chemical Processes Based on Modified Sailfish Optimizer Combined with an Equal Division Method*, vol. 9, n° 10, p. 1806, 2021.
- [48] C. Zhong, G. Li y Z. Meng, «Beluga whale optimization: A novel nature-inspired metaheuristic algorithm,» *Knowledge-Based Systems*, vol. 215, p. 109215, 2022.
- [49] Z. Fan, Z. Xiao, X. Li, Z. Huang y C. Zhang, «MSBWO: A Multi-Strategies Improved Beluga Whale Optimization Algorithm for Feature Selection,» *Biomimetics*, vol. 9, n° 9, p. 572, 2024.
- [50] Y. Jiang, Q. Wu, S. Zhu y L. Zhang, «Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems,» *Expert Systems with Applications*, vol. 188, p. 116026, 2022.
- [51] M. M. Emam, H. A. El-Sattar, E. H. Houssein y S. Kamel, «Modified orca predation algorithm: developments and perspectives on global optimization and hybrid energy systems,» *Neural Computing and Applications*, vol. 35, n° 20, pp. 15051-15073, 2023.

- [52] C. Ravelo, S. Medina y R. Olivares, «Orca Predator Algorithm for Feature Selection,» de *Software Engineering Methods in Systems and Network Systems*, Cham, Alemania, Springer, 2024, pp. 26-34.
- [53] C. A. Coello y G. Toscano Pulido, «A Micro-Genetic Algorithm for Multiobjective Optimization,» de *Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer, 2001, pp. 126-140.
- [54] D. E. Goldberg, «Sizing Populations for Serial and Parallel Genetic Algorithms,» de *International Conference on Genetic Algorithms*, San Francisco, Estados Unidos, 1989.
- [55] K. Krishnakumar y G. Rodriguez, «Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization,» de *Symposium on Visual Communications, Image Processing, and Intelligent Robotics Systems*, Filadelfia, Estados Unidos, 1990.
- [56] J. Lee, S.-M. Kim, H.-S. Park y B.-H. Woo, «Optimum design of cold-formed steel channel beams using micro Genetic Algorithm,» *Engineering Structures*, vol. 27, n° 1, pp. 17-24, 2005.
- [57] J. Mendoza, D. Morales, R. Lopez, E. Lopez, J.-C. Vannier y C. Coello Coello, «Multiobjective Location of Automatic Voltage Regulators in a Radial Distribution Network Using a Micro Genetic Algorithm,» *IEEE Transactions on Power Systems*, vol. 22, n° 1, pp. 404-412, 2007.
- [58] G. Bakare, G. Venayagamoorthy y U. Aliyu, «Reactive power and voltage control of the nigerian grid system using micro-genetic algorithm,» de *IEEE Power Engineering Society General Meeting*, San Francisco, Estados Unidos, 2005.
- [59] M. Radhika y P. Sivakumar, «Energy optimized micro genetic algorithm based LEACH protocol for WSN,» *Wireless Networks*, vol. 27, n° 1, pp. 27-40, 2021.

- [60] M. Bakhale, V. Hemalatha, S. Dhanalakshmi, R. Kumar y M. Siddharth Jain, «A Dynamic Inertial Weight Strategy in Micro PSO for Swarm Robots,» *Wireless Personal Communications*, vol. 110, n° 2, pp. 573-592, 2020.
- [61] R. Tanabe y A. S. Fukunaga, «Improving the search performance of SHADE using linear population size reduction,» de *IEEE Congress on Evolutionary Computation*, Beijing, China, 2014.
- [62] J. Brest y M. Sepesy Maučec, «Population size reduction for the differential evolution algorithm,» *Applied Intelligence*, vol. 29, n° 3, pp. 228-247, 2008.
- [63] A. E. Eiben, E. Marchiori y V. A. Valkó, «Evolutionary Algorithms with On-the-Fly Population Size Adjustment,» de *Parallel Problem Solving from Nature - PPSN VIII*, Berlin, Heidelberg, Springer, 2004, pp. 41-50.
- [64] D. Farinati y L. Vanneschi, «A survey on dynamic populations in bio-inspired algorithms,» *Genetic Programming and Evolvable Machines*, vol. 25, n° 2, p. 19, 2024.
- [65] A. Auger y N. Hansen, «A Restart CMA Evolution Strategy With Increasing Population Size,» de *IEEE Congress on Evolutionary Computation*, Edinburgh, Escocia, 2005.
- [66] K. S. Krishnakumar, «Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization,» de *SPIE*, 1990.
- [67] Y. E. Yildiz y A. O. Topal, «Large scale continuous global optimization based on micro differential evolution with local directional search,» *Information Sciences*, vol. 477, pp. 533-544, 2019.
- [68] B. Sareni y L. Krahenbuhl, «Fitness sharing and niching methods revisited,» *IEEE Transactions on Evolutionary Computation*, vol. 2, n° 3, pp. 97-106, 1998.

- [69] K. Deb, A. Pratap, S. Agarwal y T. Meyarivan, «A fast and elitist multiobjective genetic algorithm: NSGA-II,» *IEEE Transactions on Evolutionary Computation*, vol. 6, nº 2, pp. 182-197, 2002.
- [70] J. Brest, S. Greiner, B. Boskovic, M. Mernik y V. Zumer, «Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems,» *IEEE Transactions on Evolutionary Computation*, vol. 10, nº 6, pp. 646-657, 2006.
- [71] A. K. Qin y P. N. Suganthan, «Self-adaptive Differential Evolution Algorithm for Numerical Optimization,» de *IEEE Congress on Evolutionary Computation*, Edinburgh, Escocia, 2005.
- [72] W. Gong, Á. Fialho y Z. Cai, «Adaptive strategy selection in differential evolution,» de *Genetic and Evolutionary Computation Conference*, Portland, Estados Unidos, 2010.
- [73] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan y R. Qu, «Hyperheuristics: a survey of the state of the art,» *Journal of the Operational Research Society*, vol. 64, nº 12, pp. 1695-1724, 2013.
- [74] L. DaCosta, A. Fialho, M. Schoenauer y M. Sebag, «Adaptive operator selection with dynamic multi-armed bandits,» de *Genetic and Evolutionary Computation Conference*, Atlanta, Estados Unidos, 2008.
- [75] T. Wagner, H. Trautmann y B. Naujoks, «Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based on Statistical Testing,» de *Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer, 2009, pp. 198-215.
- [76] S. N. Ghoreishi, A. Clausen y B. N. Joergensen, «Termination Criteria in Evolutionary Algorithms: A Survey:,» de *9th International Joint Conference on Computational Intelligence*, Madeira, Portugal, 2017.

- [77] D. C. Tosun, Y. Işık y H. Korul, «LQR control of a quadrotor helicopter,» de *New Developments in Pure and Applied Mathematics*, Viena, Austria, 2015.
- [78] M. Nkemdirim, S. Dharan, H. Chaoui y S. Miah, «LQR Control of a 3-DOF Helicopter System,» *International Journal of Dynamics and Control*, vol. 10, n° 4, pp. 1084-1093, 2022.
- [79] Z. Jiang, J. Han, Y. Wang y Q. Song, «Enhanced LQR control for unmanned helicopter in hover,» de *1st International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, China, 2006.
- [80] E. C. Vivas, D. M. Rivera y E. J. Gómez, «Model and Observer-Based Controller Design for a Quanser Helicopter with Two DOF,» de *IEEE Ninth Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Mexico, 2012.
- [81] S. Karthick, S. Kanthalakshmi, E. V. Kumar, V. J. Kumar y A. E. Kumaran, «Experimental Validation of Adaptive Augmented LQI Control for a 2 DoF Helicopter,» *IETE Journal of Research*, vol. 70, n° 1, pp. 584-596, 2022.
- [82] S. Sadala y B. Patre, «A New Continuous Sliding Mode Control Approach with Actuator Saturation for Control of 2-DOF Helicopter System,» *ISA Transactions*, vol. 74, n° 1, pp. 165-174, 2018.
- [83] R. Ganapathy Subramanian y V. Kumar Elumalai, «Robust MRAC Augmented Baseline LQR for Tracking Control of 2 DoF Helicopter,» *Robotics and Autonomous Systems*, vol. 86, n° 1, pp. 70-77, 2016.
- [84] K. P. Tee, S. Sam Ge y F. E. H. Tay, «Adaptive Neural Network Control for Helicopters in Vertical Flight,» *IEEE Transactions on Control Systems Technology*, vol. 16, n° 4, pp. 753-762, 2008.

- [85] A. Walker y K. Cohen, «Fuzzy Logic Control of an Inverted Pendulum and a 2-DOF Helicopter,» de *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Grapevine, Texas, USA, 2013.*
- [86] A. P. Ramalakshmi, P. S. Manoharan y P. Deepamangai, «PID Tuning and Control for 2-DOF Helicopter Using Particle Swarm Optimization,» de *International Conference on Swarm, Evolutionary, and Memetic Computing, India, 2013.*
- [87] A. Al-Mahturi y H. Wahid, «Optimal Tuning of Linear Quadratic Regulator Controller Using a Particle Swarm Optimization for Two-Rotor Aerodynamical System,» *International Journal of Electronics and Communication Engineering*, vol. 11, n° 2, pp. 196-202, 2017.
- [88] P. Das, R. K. Mehta y O. Roy, «Comparative Performance Analysis for A MIMO System Based On Various Optimization Techniques,» de *International Conference on Systems Engineering, Sydney, Australia, 2018.*
- [89] I. B. Ghiloubi, L. Abdou y O. Lahmar, «PD-LQR control of 3DOF helicopter using FPA optimization,» de *International Conference on Control, Automation and Diagnosis, Roma, Italia, 2023.*
- [90] W. Boukadida, A. Benamora, H. Messaouda y P. Siarryb, «Multi-objective design of optimal higher order sliding mode control for robust tracking of 2-DoF helicopter system based on metaheuristics,» *Aerospace Science and Technology*, vol. 91, pp. 442-455, 2019.
- [91] D. Montgomery y G. Runger, *Applied Statistics and Probability for Engineers*, New Jersey: John Wiley & Sons, 2017.