



UNIVERSIDAD AUTÓNOMA DEL ESTADO
DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E
INGENIERÍA

ÁREA ACADÉMICA DE INGENIERÍA Y
ARQUITECTURA

Identificación de parámetros y optimización de
controladores no lineales por medio de algoritmos
metaheurísticos de una QUAV con carga
suspendida para mejorar su desempeño dinámico

T E S I S
PARA OBTENER EL TÍTULO DE
Doctor en Ciencias en Ingeniería, con Énfasis
en Análisis y Modelación de Sistemas
P R E S E N T A :
M. en M. Nadia Samantha Zúñiga Peña

DIRECTOR DE TESIS:
Dr. Norberto Hernández Romero

CODIRECTOR DE TESIS:
Dr. Juan Carlos Seck Tuoh Mora

Agosto de 2022



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Instituto de Ciencias Básicas e Ingeniería

School of Engineering and Basic Sciences

Área Académica de Ingeniería y Arquitectura

Department of Engineering and Architecture

Número de control: ICBI-AAIyA/3634/2023

Asunto: Autorización de impresión

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH
Presente.

El Comité Tutorial de Tesis del programa educativo de posgrado titulado “**Identificación de parámetros y optimización de controladores no lineales por medio de algoritmos metaheurísticos de una QUAV con carga suspendida para mejorar su desempeño dinámico**”, realizado por la sustentante **Nadia Samantha Zúñiga Peña** con número de cuenta **450520** perteneciente al programa de **Doctorado en Ciencias en Ingeniería, con Énfasis en Análisis y Modelación de Sistemas**; una vez que ha revisado, analizado y evaluado el documento recepcional de acuerdo a lo estipulado en el Artículo 110 del Reglamento de Estudios de Posgrado, tiene a bien extender la presente:

AUTORIZACIÓN DE IMPRESIÓN

Por lo que la sustentante deberá cumplir los requisitos del Reglamento de Estudios de Posgrado y con lo establecido en el proceso de grado vigente.

Atentamente

“Amor, Orden y Progreso”

Pachuca, Hidalgo a 08 de diciembre de 2023

El Comité Tutorial

Dr. Norberto Hernández Romero
Director

Dr. Irving Barragán Vite
Miembro del comité



Dr. Juan Carlos Seck Tzoh Mora
Codirector

Dr. Joselito Medina Marín
Miembro del comité

JCSTM/MABC

Ciudad del Conocimiento
Carretera Pachuca-Tulancingo km 4.5 Colonia
Carboneras, Mineral de la Reforma, Hidalgo,
México. C.P. 42184
Teléfono: +52 (771) 71 720 00 ext. 4000, 4001
Fax 2109
aai_icbi@uaeh.edu.mx



www.uaeh.edu.mx

Quiero expresar mi gratitud a mis asesores y revisores por todo el apoyo recibido. Por ser profesionales y tener una excelente calidad humana. Dr. Norberto Hernández Romero, Dr. Juan Carlos Seck Tuoh Mora, Dr. Joselito Medina Marín, Dr. Irving Barragán Vite, son ustedes una inspiración para seguir en el camino de la investigación.

A mi familia por su apoyo y amor incondicional.

Resumen

En este trabajo de tesis se realiza la identificación de parámetros y optimización de controladores no lineales por medio de algoritmos metaheurísticos de un Vehículo Aéreo no tripulado de tipo cuadirrotor (QUAV) con carga suspendida para mejorar su desempeño dinámico. Inicialmente, se describe a los sistemas QUAV y QUAV con carga suspendida, los cuales permiten tener conocimiento de la estructura del sistema, para posteriormente proponer la metodología de identificación y optimización, y llevar a cabo la implementación. La siguiente fase del trabajo es la elección del algoritmo metaheurístico que se emplea para la identificación de parámetros y optimización. Posteriormente, se hace la elección del controlador no lineal usando el sistema QUAV sin carga. Se valida el desempeño del algoritmo optimizado mediante simulación numérica usando el modelo dinámico tridimensional obtenido e implementando en una plataforma física. Además, se agrega una perturbación externa para verificar la robustez. Finalmente se hace la integración del algoritmo metaheurístico y el controlador no lineal seleccionados para implementarlo en el sistema QUAV con carga suspendida. Los resultados muestran un seguimiento de trayectoria con error mínimo y un buen desempeño dinámico al realizar el transporte de carga suspendida usando un QUAV. Adicionalmente, se propone una mejora para el algoritmo metaheurístico elegido, combinándolo con una Red Neuronal, con lo que se obtiene una mejora el desempeño dinámico, minimizando el error del QUAV al realizar el transporte de carga suspendida.

Abstract

In this work, the identification of parameters and optimization of nonlinear controllers applied to an unmanned aerial vehicle Quadrotor type (QUAV) with suspended load to improve its dynamic performance through metaheuristic algorithms is carried out. Initially, the QUAV and QUAV systems with suspended load are described, which allows knowledge of the system structure to propose the identification and optimization methodology and carry out the implementation. The next phase of the work is the choice of the metaheuristic algorithm used for parameter identification and optimization. The nonlinear controller is chosen and implemented using the QUAV system without load. The performance of the optimized algorithm is validated through numerical simulation using the obtained three-dimensional dynamic model and the physical platform. Additionally, an external perturbation is applied to verify the robustness. Finally, the metaheuristic algorithm and the selected controller are integrated to implement it in the QUAV system with a suspended load. The results show trajectory tracking with minimal error and good dynamic performance when a QUAV carries a suspended load. Furthermore, an improvement is proposed for the chosen metaheuristic algorithm. Combining it with a neural network improves dynamic performance and minimizes the QUAV error when transporting a suspended load.

Contenido

Introducción	1
Objetivos	3
Objetivo General	3
Objetivos Específicos	3
Justificación	3
Hipótesis	4
Organización de la Tesis	5
1 Descripción del sistema UAV con carga suspendida	7
1.1 Modelo matemático del QUAV	8
1.2 Modelo matemático del QUAV con carga suspendida	14
2 Optimización	22
2.1 Elección de algoritmo metaheurístico	22
2.1.1 Optimización por Enjambre de Partículas	24
2.1.2 Algoritmo de Optimización de Lobos	26
2.1.3 Algoritmo de Juegos del Hambre	28
2.1.4 Algoritmo L-SHADE	31
2.1.5 Algoritmo LSPACMA	31
2.1.6 Algoritmo de Predadores Marinos	34
2.1.7 Algoritmo SMA	35
2.1.8 Algoritmo de Optimización de Ballenas	37
2.2 Proceso de optimización de un controlador	38
2.3 Resultados del proceso de optimización de un controlador: comparación de los diferentes algoritmos Metaheurísticos	40

3	Control	51
3.1	Estructura de control	52
3.2	Diseño de Control PID	53
3.3	Diseño de Control ST - SMC	56
3.4	Resultados de la comparación de controles PID vs ST-SMC	61
3.4.1	Comparación en simulación	62
3.4.2	Comparación de los controladores en plataforma física	65
4	Transporte de carga suspendida mediante un QUAV con control ST-SMC optimizado implementando el algoritmo HGS	72
4.1	Implementación del Modelo dinámico	73
4.2	Implementación del controlador ST-SMC	74
4.3	Identificación de parámetros y optimización del controlador ST-SMC	76
4.3.1	Resultados	78
5	Hibridación NN-HGS	83
5.1	Hibridación del algoritmo NN-HGS en la optimización de controlador para el QUAV con carga suspendida	87
5.2	Resultados de la implementación	87
5.2.1	Comparación de optimización de control PID con HGS y NN-HGS	89
5.2.2	Comparación de optimización de control ST-SMC con HGS y NN-HGS	90
6	Conclusiones y trabajos futuros	95
6.1	Conclusiones	95
6.2	Aportaciones	97
6.3	Trabajos Futuros	97

Lista de Figuras

1.1	Marco de referencia, W , y marco del cuerpo del vehículo, B	9
1.2	Orden de rotación del marco W a B	11
1.3	QUAV con carga suspendida, moviéndose en en el espacio tridimensional $X - Y - Z$	15
2.1	Diagrama de flujo del PSO.	25
2.2	Diagrama de flujo del GWO.	27
2.3	Secuencia de acciones basadas en el Hambre. Algoritmo HGS.	32
2.4	Diagrama de flujo del L-SHADE.	33
2.5	Diagrama de flujo del LSPACMA.	33
2.6	Diagrama de flujo del MPA.	35
2.7	Diagrama de flujo del SMA.	37
2.8	Diagrama de flujo del WOA.	38
2.9	Esquema de sintonización de ganancias del controlador.	39
2.10	Desempeño de los algoritmos evolutivos en cada evolución.	46

2.11 Comparación del desempeño del QUAV en el eje X, evaluando los parámetros obtenidos con los diferentes algoritmos, empleando 20 individuos.	47
2.12 Comparación del desempeño del QUAV en el eje Y, evaluando los parámetros obtenidos con los diferentes algoritmos, empleando 20 individuos.	47
2.13 Comparación del desempeño del QUAV en el eje Z, evaluando los parámetros obtenidos con los diferentes algoritmos, empleando 20 individuos.	48
2.14 Desempeño de los ángulos ϕ y θ , contra los ángulos deseados calculados mediante el control virtual.	48
2.15 Desempeño de los ángulos ϕ y θ , contra los ángulos deseados calculados mediante el control virtual.	49
2.16 Desempeño del ángulo ψ , contra el ángulo deseado.	49
2.17 Seguimiento de trayectoria del QUAV en X - Y - Z.	50
2.18 Evaluación de diferente trayectoria usando las ganancias obtenidas mediante optimización.	50
3.1 Estructura de control propuesta para seguimiento de trayectoria. . . .	52
3.2 Método para la identificación de parámetros de los controles PID y ST-SMC.	62
3.3 Curvas de convergencia obtenidas por el HGS, con 20 iteraciones, 50 individuos y 30 repeticiones.	63

3.4	Simulaciones de seguimiento de trayectoria aplicando los controles PID y ST-SMC sin perturbaciones.	65
3.5	Comparación de los controles PID y ST-SMC para seguimiento de trayectoria aplicando una perturbación.	66
3.6	Estructura de la plataforma experimental.	68
3.7	Resultados de pruebas experimentales. Desempeño del QUAV con el controlador PID optimizado mediante HGS sin perturbaciones.	69
3.8	Resultados de pruebas experimentales. Desempeño del QUAV con el controlador ST-SMC optimizado mediante HGS sin perturbaciones.	70
3.9	Resultados de pruebas experimentales. Comparación del desempeño del QUAV con los controles PID y ST-SMC optimizados mediante HGS agregando una perturbación.	71
4.1	Estructura de control propuesta para la implementación del control ST-SMC al QUAV con carga.	74
4.2	Esquema de sintonización de ganancias del controlador ST-SMC implementando el algoritmo HGS.	77
4.3	Curva de convergencia obtenidas por el HGS, con 20 iteraciones, 50 individuos y 1 repetición.	78
4.4	Desempeño de las dinámicas traslacionales del QUAV con carga, controlado con ST-SMC optimizado mediante el algoritmo HGS.	80
4.5	Desempeño de las dinámicas rotaciones del QUAV con carga, controlado con ST-SMC optimizado mediante el algoritmo HGS.	81

4.6	Desempeño del QUAV con carga suspendida en seguimiento de trayectoria en el espacio $X - Y - Z$	82
5.1	Estructura aprendizaje no supervisado de la red neuronal Kohonen.	85
5.2	Secuencia de acciones basadas en el Hambre con ajuste de espacio de búsqueda mediante Red Neuronal. Algoritmo NN-HGS.	88
5.3	Diagrama de proceso para optimización de controlador mediante algoritmo NN-HGS.	89
5.4	Desempeño de las dinámicas de traslación del QUAV con carga suspendida.	92
5.5	Desempeño de las dinámicas de orientación del QUAV con carga suspendida.	93
5.6	Comparación del desempeño del QUAV con el control ST-SMC optimizados mediante HGS y NN-HGS.	94

Lista de Tablas

1.1	Tabla de notaciones	8
1.2	Tabla1	11
2.1	Parámetros del algoritmo HGS: approach food	29
2.2	Parámetros del algoritmo HGS: hunger role.	31
2.3	Parámetros físicos de modelo de QUAV.	42
2.4	Estadística de desempeño de cada algoritmo: RMSE con 20 individuos, 20 evoluciones and 30 repeticiones.	43
2.5	Parámetros obtenidos por cada algoritmo metaheurístico con 20 indi- viduos, 20 evoluciones y 30 repeticiones.	43
2.6	Estadística de desempeño de cada algoritmo: RMSE con 30 individuos, 20 evoluciones y 30 repeticiones.	44
2.7	Parámetros obtenidos por cada algoritmo metaheurístico con 30 indi- viduos, 20 evoluciones y 30 repeticiones.	44
2.8	Estadística de desempeño de cada algoritmo: RMSE con 40 individuos, 20 evoluciones y 30 repeticiones.	45

2.9	Parámetros obtenidos por cada algoritmo metaheurístico con 40 individuos, 20 evoluciones y 30 repeticiones.	45
3.1	Valores de RMSE alcanzados con 50 individuos, 20 evoluciones, y 30 repeticiones.	62
3.2	Parámetros obtenidos mediante el algoritmo HGS.	64
3.3	Parámetros implementados en simulación.	65
4.1	Parámetros físicos implementados.	73
4.2	Parámetros obtenidos mediante el algoritmo HGS.	79
5.1	Parámetros obtenidos mediante el algoritmo HGS y NN-HGS para el controlador PID.	90
5.2	Parámetros obtenidos mediante el algoritmo HGS y NN-HGS para el controlador ST-SMC.	91

Introducción

Los vehículos aéreos no tripulados (UAVs) son sistemas subactuados con dinámicas acopladas, característica que los vuelve una plataforma desafiante para el modelado y diseño de control. No obstante, el uso de UAVs se ha popularizado debido a su versatilidad de vuelo en entornos poco accesibles y peligrosos, que les permite llevar a cabo tareas complejas. Es común encontrar QUAVs (UAVs de cuatro rotores) realizando tareas de supervisión, rescate, transporte de cargas, recolección de datos e incluso tareas de reforestación [1]. Específicamente el transporte de carga ha sido tema de investigación de manera recurrente debido a la utilidad para aplicaciones relevantes como entrega de medicamentos y víveres en lugares alejados o poco accesibles para los humanos, el transporte de sustancias peligrosas y se espera que en un futuro cercano, la entrega comercial de paquetes sea de manera masiva, a través de UAVs [2].

Una problemática de los QUAVs es la inestabilidad debido a la no linealidad, por ello la importancia de asegurar robustez en el diseño de un buen controlador de orientación y traslación durante su trayectoria. En su mayoría, los controladores tienen parámetros que necesitan ajustarse para obtener un buen rendimiento, y es necesario realizar muchas pruebas para poder sintonizar dichos parámetros. El tiempo

que toma realizar esa sintonización es alto y no garantiza obtener los parámetros óptimos.

Debido a la complejidad de diseñar e implementar un controlador en los QUAVs, se han propuesto múltiples alternativas [3], entre las que se encuentran las soluciones de naturaleza estocástica [4], [5]. El proceso de sintonización de controladores puede ser abordado como un problema de optimización sujeto a restricciones, para el que se han propuesto diferentes tipos de algoritmos para darle solución. Entre los más populares se encuentran los algoritmos meta-heurísticos evolutivos y los basados en población, que han demostrado efectividad al implementarse en problemas complejos [6],[7], [8].

Los algoritmos metaheurísticos evolutivos están inspirados en las leyes de evolución biológica, no conservan ni comparten información de la población de una evolución a otra, sin embargo, los mejores individuos se combinan para formar a la nueva generación de la población, con lo que al final quedarán siempre los mejores individuos [9], [10], [11]. Los algoritmos meta-heurísticos basados en población están inspirados en el comportamiento colaborativo de grupos de individuos, con la habilidad de almacenar y compartir información entre los miembros de la población, lo que le permite crear conocimiento individual y colectivo, permitiendo encontrar una solución a partir de la mejora iterativa de las soluciones [12], [13], [14].

Objetivos

Objetivo General

Identificar, optimizar, implementar y evaluar controladores no lineales para un Vehículo Aéreo no Tripulado de tipo cuadirrotor con carga suspendida vía cable, por medio de algoritmos meta-heurísticos para mejorar el desempeño dinámico.

Objetivos Específicos

1. Identificar los parámetros de controladores no lineales por medio de algoritmos metaheurísticos para un Vehículo Aéreo no Tripulado de tipo cuadirrotor con carga suspendida vía cable.
2. Optimizar controladores no lineales mediante algoritmos metaheurísticos, para las dinámicas de altitud, traslacionales y de orientación de un Vehículo Aéreo no Tripulado de tipo cuadirrotor con carga suspendida vía cable.
3. Implementar los controladores no lineales optimizados para un Vehículo Aéreo no Tripulado de tipo cuadirrotor con carga suspendida vía cable.
4. Evaluar el desempeño dinámico del controlador no lineal optimizado para un Vehículo Aéreo no Tripulado de tipo cuadirrotor con carga suspendida vía cable.

Justificación

La investigación se realiza porque el uso de UAVs para el transporte de carga en la actualidad se ha popularizado y va en aumento, ya que permite llegar a zonas de difícil acceso o peligrosas para el humano, lo que los convierte en una herramienta de apoyo ideal en situaciones de búsqueda y rescate, entrega de víveres o medicamentos o inspección en situaciones de desastre, como pudimos comprobar en el sismo del 2017 [15], otro ejemplo de aplicación, es la agricultura de precisión, o en el área comercial para entrega de artículos, por mencionar algunas [1]. Al realizar transporte de carga suspendida de forma autónoma es necesario garantizar la seguridad del vehículo, la carga, así como del entorno. Lo anterior se logra mediante un controlador que estabilice el sistema manteniendo el ángulo entre la carga suspendida y el vehículo lo más cercano a cero. Los beneficiarios directos son las organizaciones que implementan el uso de UAVs para transporte de carga y la población en general, ya que permite realizarlo de forma autónoma y segura. Es factible realizar la investigación, ya que se cuenta con el Software y equipo necesario para las pruebas numéricas, así como el acceso para uso de la plataforma experimental que está integrada por un sistema de posicionamiento de Cámaras Vicon que permite conocer la posición y orientación de los cuerpos en tiempo real, el cual, se comunica mediante ROS (Robot Operating System) con un vehículo cuadrirrotor para implementación experimental.

Hipótesis

Mediante la identificación de parámetros por medio de algoritmos metaheurísticos, es posible la optimización de controles no lineales, para las dinámicas de altitud, traslacionales y de orientación de un QUAV con carga suspendida vía cable y mejorar así su desempeño dinámico.

Organización de la Tesis

El presente trabajo de tesis se ha organizado de la siguiente manera:

Capítulo 1. Modelo Matemático, se presentan las ecuaciones que describen las dinámicas del QUAV y QUAV con carga suspendida.

Capítulo 2. Optimización, se describe el procedimiento para la elección del algoritmo metaheurístico a implementar.

Capítulo 3. Control, se describe el procedimiento para la elección del controlador no lineal para el seguimiento de trayectoria del QUAV con carga suspendida.

Capítulo 4. Transporte de carga suspendida mediante un QUAV, se describe el procedimiento para la implementación del algoritmo metaheurístico HGS para la identificación de parámetros del controlador no lineal ST-SMC, para el seguimiento de trayectoria del QUAV con carga suspendida.

Capítulo 5. Hibridación NNHGS, se describe la Hibridación del Algoritmo Hunger Games Search y Red Neuronal de Kohonen para la identificación de parámetros y optimización de controles no lineales.

Capítulo 6. Conclusiones y Trabajos Futuros, se describen los resultados obtenidos de cada una de las fases del trabajo, así como las actividades que quedan por hacer para dar continuidad a la investigación.

Capítulo 1

Descripción del sistema UAV con carga suspendida

En este capítulo se obtiene el modelo dinámico del Vehículo Aéreo no Tripulado mediante la metodología de Newton-Euler y posteriormente el modelo que describe las dinámicas el UAV de cuatro rotores con carga suspendida vía cable.

1.1 Modelo matemático del QUAV

Tabla 1.1: Tabla de notaciones

Símbolo	Descripción
x, y, z	Coordenadas traslacionales respecto al marco inercial
$\dot{x}, \dot{y}, \dot{z}$	Velocidades traslacionales respecto al marco inercial
$\ddot{x}, \ddot{y}, \ddot{z}$	Aceleraciones traslacionales respecto al marco inercial
ϕ, θ, ψ	Coordenadas de orientación en el marco inercial
$\dot{\phi}, \dot{\theta}, \dot{\psi}$	Velocidad angular en el marco inercial
$\ddot{\phi}, \ddot{\theta}, \ddot{\psi}$	Aceleraciones angulares en el marco inercial
${}^W R_B$	Matriz de rotación del marco del cuerpo al inercial
p, q, r	Velocidad angular del marco del cuerpo
$\dot{p}, \dot{q}, \dot{r}$	Aceleración angular del marco del cuerpo
u, v, w	Velocidad traslacional en el marco del cuerpo
g	Gravedad en el marco inercial
m	Masa del cuerpo
\mathbf{M}	Momentos aplicados respecto al centro de masa del cuerpo
\mathbf{M}_a	Momentos resultantes de las fricciones aerodinámicas
\mathbf{M}_b	Momentos generados por los motores.
\mathbf{I}_B	Matriz de tensión de inercias
\mathbf{F}_B	Vector de fuerza de sustentación o <i>thrust</i>
\mathbf{F}_d	Vector de fuerzas de arrastre
\mathbf{F}_g	Vector de fuerzas debido a la gravedad
$k_{1,2,\dots,6}$	Coefficientes de arrastre
s, c	$\sin(\cdot), \cos(\cdot)$

Para describir el movimiento de un objeto en tres dimensiones, es necesario definir dos sistemas de referencia: el inercial o del mundo, W , y el sistema coordenado del cuerpo, B . El primero, es un punto fijo en la superficie de la tierra y está conformado por $x_W - y_W - z_W$, el segundo, es el punto donde se encuentra el centro de gravedad del objeto y se denota por $x_B - y_B - z_B$, como se muestra en la Fig. 1.1.

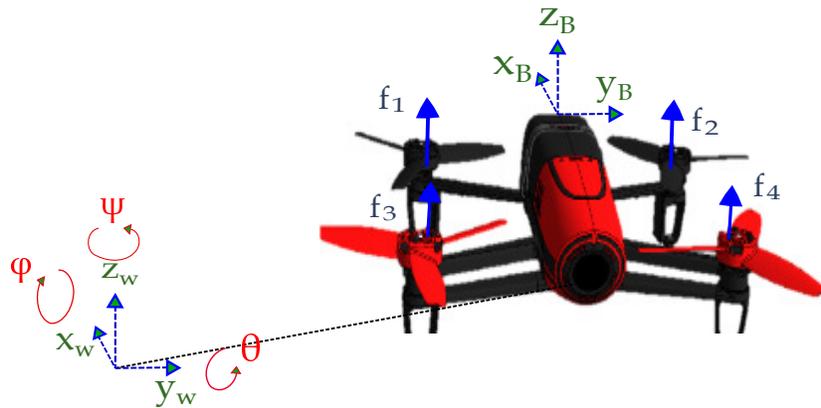


Figura 1.1: Marco de referencia, W , y marco del cuerpo del vehículo, B .

El modelo desarrollado en este trabajo se obtiene bajo las siguientes suposiciones: 1) La estructura es rígida, 2) la estructura es simétrica, 3) el centro de gravedad y el origen del marco del cuerpo coinciden con el centro geométrico del vehículo, y 4) las propelas son rígidas.

La posición absoluta del QUAUV, se define como las coordenadas desde el centro de gravedad del QUAUV (B), con respecto al sistema de referencia de la tierra (W), y está representada por el vector de posición $\mathbf{r} = [x, y, z]^T$. La posición angular, definida por los ángulos de *Euler* (ϕ, θ y ψ), representan la orientación del QUAUV con respecto al marco inercial, estos ángulos definen a las rotaciones de *yaw*, *roll* y *pitch*, es decir,

indican cómo el sistema coordenado, que inicialmente está alineado con el sistema inercial, debe ser rotado, para alinearse con el sistema coordenado del cuerpo. La matriz de rotación de W a B , está dada por ${}^W R_B = {}^c R_B {}^a R_c {}^W R_a$, donde ${}^c R_B$, representa la rotación de yaw, ${}^a R_c$, representa el efecto de pitch y ${}^W R_a$ representa el efecto de roll, es decir, el orden la rotación es $Z - Y - X$, como se muestra en la Fig 1.2.

$$\begin{aligned}
 {}^c R_B &= \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 {}^a R_c &= \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \\
 {}^W R_a &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \\
 {}^W R_B &= \\
 &\begin{pmatrix} \cos(\theta) \cos(\psi) & \cos(\psi) \sin(\theta) \sin(\phi) - \cos(\phi) \sin(\psi) & \cos(\psi) \cos(\phi) \sin(\theta) + \sin(\psi) \sin(\phi) \\ \cos(\theta) \sin(\psi) & \cos(\psi) \cos(\phi) + \sin(\theta) \sin(\psi) \sin(\phi) & \cos(\phi) \sin(\theta) \sin(\psi) - \cos(\psi) \sin(\phi) \\ -\sin(\theta) & \cos(\theta) \sin(\phi) & \cos(\theta) \cos(\phi) \end{pmatrix}
 \end{aligned}$$

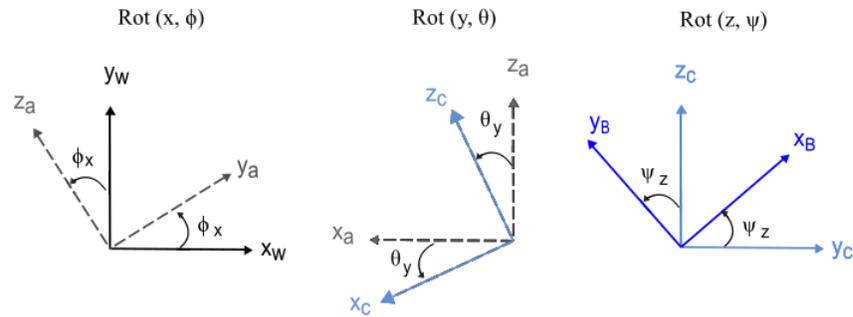


Figura 1.2: Orden de rotación del marco W a B .

Antes de pasar a las ecuaciones de movimiento, se debe definir la relación entre las velocidades en el marco inercial $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ y las velocidades en el marco del cuerpo (p, q, r) , como se muestra en la Tabla 1.2.

Tabla 1.2: Relación velocidades en W y B

Rotación	Marco inicial	Marco final	Vector de velocidad angular asociado con la rotación
$Rot(z, \psi)$	Marco inercial	Marco a	$\begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$ en W
$Rot(y, \theta)$	Marco a	Marco c	$\begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix}$ en a
$Rot(x, \phi)$	Marco c	Marco del cuerpo	$\begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix}$ en c

Entonces, para encontrar a la velocidad angular en el marco del cuerpo, $\omega_B = f(\dot{\phi}, \dot{\theta}, \dot{\psi})$, se utilizan los siguientes vectores de transformación, para $\dot{\psi}$:

$${}^W R_a {}^a R_c {}^c R_B [0 \ 0 \ \dot{\psi}]^T = \begin{pmatrix} -\sin(\theta)\dot{\psi} \\ \cos(\theta)\sin(\phi)\dot{\psi} \\ \cos(\theta)\cos(\phi)\dot{\psi} \end{pmatrix},$$

para $\dot{\theta}$:

$${}^W R_a {}^a R_c [0 \ 0 \ \dot{\phi}]^T = \begin{pmatrix} 0 \\ \sin(\phi)\dot{\theta} \\ -\sin(\phi)\dot{\theta} \end{pmatrix},$$

para $\dot{\phi}$:

$${}^W R_a [\dot{\phi} \ 0 \ 0]^T = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix}.$$

Entonces:

$$\begin{aligned} \omega_B &= \begin{pmatrix} -\sin(\theta)\dot{\psi} \\ \cos(\theta)\sin(\phi)\dot{\psi} \\ \cos(\theta)\cos(\phi)\dot{\psi} \end{pmatrix} + \begin{pmatrix} 0 \\ \sin(\phi)\dot{\theta} \\ -\sin(\phi)\dot{\theta} \end{pmatrix} + \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} = \\ &= \begin{pmatrix} \dot{\phi} - \dot{\psi}\sin(\theta) \\ \sin(\phi)\dot{\theta} + \dot{\psi}\cos(\theta)\sin(\phi) \\ \sin(\phi)\cos(\theta)\dot{\psi} - \dot{\theta}\sin(\phi) \end{pmatrix} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \end{aligned}$$

reescribiendo queda como:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}.$$

Asumiendo que los movimientos de orientación son pequeños, las coordenadas del cuerpo y del marco inercial son equivalentes. Aplicando la metodología de Newton-Euler, las ecuaciones de movimiento en el sistema coordinado del cuerpo, están dadas por las (1.1) y (1.2).

$$m\dot{\mathbf{v}} = \mathbf{F}_g + \mathbf{F}_d + \mathbf{F} \quad (1.1)$$

$$\mathbf{I}_B\dot{\boldsymbol{\omega}}_B + (\boldsymbol{\omega}_B \times \mathbf{I}_B\boldsymbol{\omega}_B) + \mathbf{M}_a = \mathbf{M}_B \quad (1.2)$$

En la Ecuación (1.1), la aceleración de la masa ($m\dot{\mathbf{v}} = m(\ddot{x} \ \ddot{y} \ \ddot{z})^T$), es equivalente a la fuerza de gravedad ($\mathbf{F}_g = (0, 0, -mg)^T$), fuerzas de arrastre ($\mathbf{F}_d = \text{diag}(k_1, k_2, k_3)\mathbf{v}^T$) y la fuerza de sustentación o *thrust* generada por los cuatro rotores ($\mathbf{F} = {}^W R_B[0, 0, u_1]^T$). En la Ecuación (1.2), la aceleración angular de los momentos de inercia ($\mathbf{I}_B\dot{\boldsymbol{\omega}}_B$), las fuerzas centrípetas ($(\boldsymbol{\omega}_B \times \mathbf{I}_B\boldsymbol{\omega}_B)$) y las fuerzas aerodinámicas ($\mathbf{M}_a = \text{diag}(k_4, k_5, k_6)\boldsymbol{\omega}_B^2$), son iguales a los torques externos ($\mathbf{M}_B = (u_2, u_3, u_4)^T$). Sustituyendo el vector de posición y las expresiones de las fuerzas, así como los momentos en (1.1) y (1.2) se obtienen las siguientes dinámicas de traslación y rotacionales del QUAV:

$$\begin{aligned}
 \ddot{x} &= \frac{1}{m} (\cos(\psi) \sin(\theta) \cos(\phi) + \sin(\phi) \sin(\psi)) u_1 - \frac{k_1}{m} \dot{x} \\
 \ddot{y} &= \frac{1}{m} (\sin(\psi) \sin(\theta) \cos(\phi) - \sin(\phi) \cos(\psi)) u_1 - \frac{k_2}{m} \dot{y} \\
 \ddot{z} &= \frac{1}{m} (\cos(\phi) \cos(\theta)) u_1 - g - \frac{k_3}{m} \dot{z} \\
 \ddot{\phi} &= \frac{(Iy - Iz)}{Ix} \dot{\theta} \dot{\psi} - \frac{k_4}{Ix} \dot{\phi}^2 + \frac{1}{Ix} u_2 \\
 \ddot{\theta} &= \frac{(Iz - Ix)}{Iy} \dot{\phi} \dot{\psi} - \frac{k_5}{Iy} \dot{\theta}^2 + \frac{1}{Iy} u_3 \\
 \ddot{\psi} &= \frac{(Ix - Iy)}{Iz} \dot{\phi} \dot{\theta} - \frac{k_6}{Iz} \dot{\psi}^2 + \frac{1}{Iz} u_4
 \end{aligned} \tag{1.3}$$

1.2 Modelo matemático del QUAV con carga suspendida

El modelo matemático de un QUAV con una carga suspendida vía cable, se obtiene bajo las siguientes dos suposiciones: 1) el cable es rígido, no posee masa y es inelástico y 2) la carga es considerada como un punto de masa. Para poder obtener el modelo matemático, es necesario definir tres marcos de referencia: 1) $\mathcal{I} = \{x_I, y_I, z_I\}$ que es el marco inercial, considerado fijo con respecto a la Tierra, 2) el marco fijo al cuerpo del cuadrirrotor $\mathcal{B} = \{x_B, y_B, z_B\}$, cuyo origen está situado en el centro de masa del vehículo y 3) $\mathcal{C} = \{x_C, y_C, z_C\}$ es el marco inercial fijo en la carga.

El sistema QUAV carga se muestra en la Figura 1.3, en el espacio tridimensional $X - Y - Z$.

El vector $\xi = [x \ y \ z]^T \in \mathbb{R}^3$ representa a la posición del centro de masa del QUAV

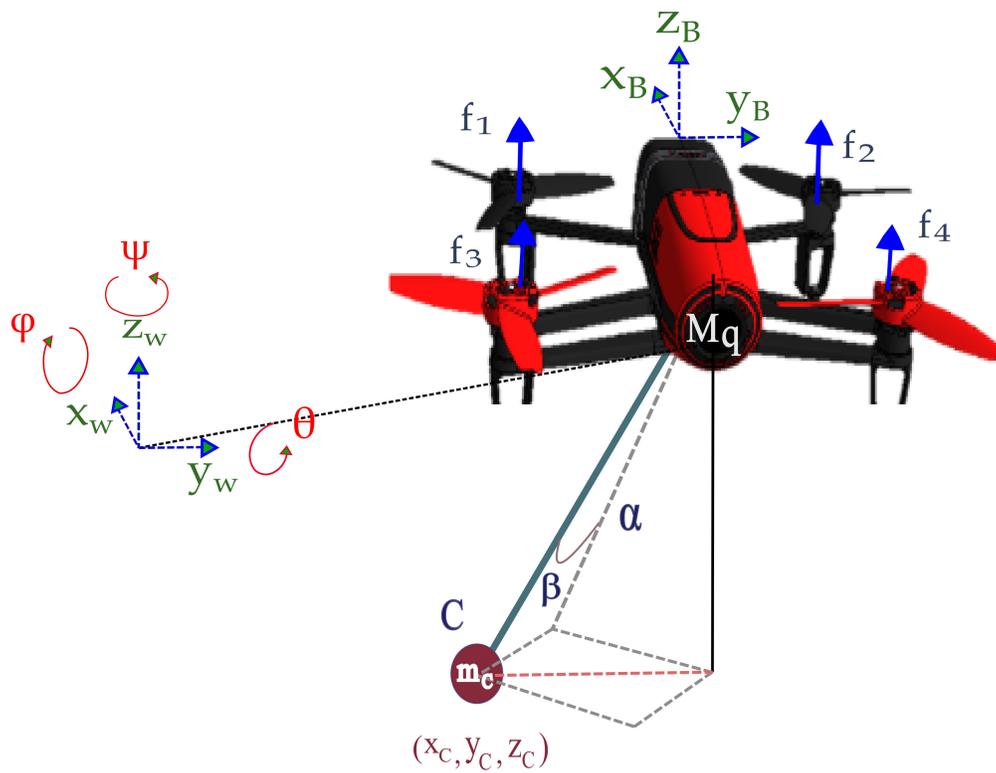


Figura 1.3: QUAUV con carga suspendida, moviéndose en en el espacio tridimensional $X - Y - Z$.

expresado en el marco inercial \mathcal{I} , $\xi_c = [x_c \ y_c \ z_c]^T \in \mathbb{R}^3$ denota a la posición de la carga, $\eta = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$ son los ángulos de Euler: roll, pitch y yaw, respectivamente, los cuales denotan a la posición angular del cuerpo con respecto al marco inercial, y $\eta_c = [\alpha \ \beta]^T$ representa los ángulos de la oscilación de la carga suspendida. M_q y m_c son la masa del QUAV y de la carga, respectivamente, g es la aceleración debido a la gravedad. Dadas las definiciones anteriores, el vector de coordenadas generalizadas del sistema QUAV-carga está dado por

$$q = (x \ y \ z \ \phi \ \theta \ \psi \ \alpha \ \beta)^T \in \mathbb{R}^8. \quad (1.4)$$

La derivación de las ecuaciones de movimiento se realiza aplicando la teoría de Euler-Lagrange, que es una expresión basada en la energía cinética y potencial. La energía total del sistema está expresada por la función Lagrangiano $\mathcal{L}(q, \dot{q})$ y se denota por

$$\mathcal{L}(q, \dot{q}) = K(q, \dot{q}) - U(q), \quad (1.5)$$

donde K representa a la energía cinética total y U a la energía potencial total, con las ecuaciones de Euler-Lagrange,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \begin{bmatrix} f \\ \tau \end{bmatrix}. \quad (1.6)$$

En particular, para el sistema QUAV-carga en 3D, la energía cinética y potencial están

dadas por

$$K = \frac{1}{2}M_q\dot{\xi}^T\dot{\xi} + \frac{1}{2}m_c\dot{\xi}_c^T\dot{\xi}_c + \frac{1}{2}\dot{\eta}^T\mathbb{J}\dot{\eta} + \frac{1}{2}I_c\dot{\eta}_c\dot{\eta}_c, \quad (1.7)$$

$$U = M_qgz + m_cgz_c, \quad (1.8)$$

donde,

$$\mathbb{J} = \mathbf{W}_\eta^T \mathbf{I} \mathbf{W}_\eta^T \in \mathbb{R}^3, \quad (1.9)$$

es la matriz de inercia total para la energía cinética rotacional, \mathbf{W}_η está dada por,

$$\mathbf{W}_\eta = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix},$$

$\mathbf{I} \in \mathbb{R}^{3 \times 3}$ es la matriz de inercia del cuadrirotor, compuesta por los valores,

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix},$$

I_c es la matriz identidad de inercia de la carga. La posición de la carga es expresada en términos de la posición de cuadrirotor, el cable de longitud L y la orientación de

la carga como,

$$\begin{aligned}x_c &= x - L \sin \alpha \\y_c &= y - L \sin \beta \\z_c &= z - L \cos \alpha \cos \beta\end{aligned}$$

Sustituyendo (1.7) y (1.8) en (1.5), el Lagrangiano está dado por,

$$\begin{aligned}L &= \frac{1}{2}(M_q)(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2}(m_c)[(\dot{x}^2 - 2l\dot{x} \cos \alpha \dot{\alpha} + l^2 \cos^2 \alpha \dot{\alpha}^2) + \\&(\dot{y}^2 - 2l\dot{y} \cos \beta \dot{\beta} + l^2 \cos^2 \beta \dot{\beta}^2) + (\dot{z}^2 + 2l \cos \beta \sin \alpha \dot{z} \dot{\alpha} + l^2 \cos^2 \beta \sin^2 \alpha \dot{\alpha}^2 + \\&2l \cos \alpha \sin \beta \dot{z} \dot{\beta} + 2l^2 \cos \alpha \cos \beta \sin \alpha \sin \beta \dot{\alpha} \dot{\beta} + l^2 \cos^2 \alpha \sin^2 \beta \dot{\beta}^2)] + \\&\frac{1}{2}[I_{xx} \dot{\phi}^2 - 2I_{xx} \sin \theta \dot{\phi} \dot{\psi} + \dot{\theta}^2 (I_{yy} \cos^2 \phi + I_{zz} \sin^2 \phi) + \\&\dot{\psi}^2 (I_{xx} \sin^2 \theta + I_{zz} \cos^2 \theta \cos^2 \phi + I_{yy} \cos^2 \theta \sin^2 \phi) + \dot{\theta} \dot{\psi} (I_{yy} - I_{zz}) \cos \theta \sin(2\phi)] + \\&\frac{1}{2}I_c(\dot{\alpha}^2 + \dot{\beta}^2) - M_q g z - m_c g (z - l \cos \alpha \cos \beta).\end{aligned}\tag{1.10}$$

Aplicando(1.6) a (1.10), las ecuaciones del modelo matemático del sistema pueden ser representadas en forma matricial como,

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) = \mathbf{b}(q)\mathbf{u},\tag{1.11}$$

donde $\mathbf{M}(q) \in \mathbb{R}^{8 \times 8}$ es la matriz de inercia, la cual es simétrica y definida positiva, $\mathbf{C}(q, \dot{q}) \in \mathbb{R}^{8 \times 8}$ es la matriz de Coriolis, $\mathbf{G}(q) \in \mathbb{R}^8$ es el vector de la gravedad, $\mathbf{b}(q) \in \mathbb{R}^{8 \times 4}$ es la matriz de rotación para las entradas y $\mathbf{u} \in \mathbb{R}^4$ es el vector de

entradas del sistema. Estas matrices están compuestas como sigue:

$$\mathbf{M}(q) = \begin{bmatrix} M_{11} & 0 & 0 & 0 & 0 & 0 & M_{17} & 0 \\ 0 & M_{22} & 0 & 0 & 0 & 0 & 0 & M_{28} \\ 0 & 0 & M_{33} & 0 & 0 & 0 & M_{37} & M_{38} \\ 0 & 0 & 0 & M_{44} & 0 & M_{46} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & M_{56} & 0 & 0 \\ 0 & 0 & 0 & M_{64} & M_{65} & M_{66} & 0 & 0 \\ M_{71} & 0 & M_{73} & 0 & 0 & 0 & M_{77} & M_{78} \\ 0 & M_{82} & M_{83} & 0 & 0 & 0 & M_{87} & M_{88} \end{bmatrix}, \quad (1.12)$$

$$M_{11} = M_{22} = M_{33} = (M + m),$$

$$M_{17} = M_{71} = -Lm \cos \alpha,$$

$$M_{28} = M_{82} = -Lm \cos \beta,$$

$$M_{37} = M_{73} = Lm \cos \beta \sin \alpha,$$

$$M_{38} = M_{83} = Lm \cos \alpha \sin \beta,$$

$$M_{44} = I_{xx},$$

$$M_{46} = M_{64} = -I_{xx} \sin \theta,$$

$$M_{55} = I_{yy} \cos^2 \phi + I_{zz} \sin^2 \phi,$$

$$M_{56} = (I_{yy} - I_{zz}) \cos \theta \cos \phi \sin \phi,$$

$$M_{65} = (I_{yy} - I_{zz}) \cos \theta \cos \phi \sin \phi,$$

$$M_{66} = I_{xx} \sin^2 \theta + I_{yy} \sin^2 \phi \cos^2 \theta + I_{zz} \cos^2 \phi \cos^2 \theta,$$

$$M_{77} = L^2 m (\cos^2 \alpha + \cos^2 \beta \sin^2 \alpha + Ic),$$

$$M_{88} = L^2 m (\cos^2 \beta + \cos^2 \alpha \sin^2 \beta + Ic).$$

$$M_{78} = M_{87} = L^2 \cos \alpha \cos \beta \sin \alpha \sin \beta,$$

$$\mathbf{C}(q, \dot{q}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & C_{17} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{28} \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{37} & C_{38} \\ 0 & 0 & 0 & 0 & C_{45} & C_{46} & 0 & 0 \\ 0 & 0 & 0 & C_{54} & C_{55} & C_{56} & 0 & 0 \\ 0 & 0 & 0 & C_{64} & C_{65} & C_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{77} & C_{78} \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{87} & C_{88} \end{bmatrix}, \quad (1.13)$$

donde,

$$C_{17} = Lm_c \sin \alpha \dot{\alpha}, \quad C_{28} = Lm_c \sin \beta \dot{\beta},$$

$$C_{37} = Lm_c \cos \beta \cos \alpha \dot{\alpha} - Lm_c \sin \beta \sin \alpha \dot{\beta},$$

$$C_{38} = Lm_c \cos \beta \cos \alpha \dot{\beta} - Lm_c \sin \beta \sin \alpha \dot{\alpha},$$

$$C_{45} = (I_{yy} - I_{zz})(\dot{\psi} \cos \theta \sin^2 \theta + \dot{\theta} \cos \phi \sin \phi) + (I_{zz} - I_{yy})\dot{\psi} \cos^2 \phi \cos \theta - I_{xx}\dot{\psi} \cos \theta,$$

$$C_{46} = (I_{zz} - I_{yy})\dot{\psi} \cos^2 \theta \cos \phi \sin \phi,$$

$$C_{54} = (I_{zz} - I_{yy})(\dot{\psi} \cos \theta \sin^2 \phi + \dot{\theta} \cos \phi \sin \phi) + (I_{yy} - I_{zz})\dot{\psi} \cos^2 \phi \cos \theta + I_{xx}\dot{\psi} \cos \theta,$$

$$C_{55} = (I_{zz} - I_{yy})\dot{\phi} \cos \phi \sin \phi,$$

$$C_{56} = -I_{xx}\dot{\psi} \sin \theta \cos \theta + I_{yy}\dot{\psi} \sin^2 \phi \sin \theta \cos \theta + I_{zz}\dot{\psi} \cos^2 \phi \sin \theta \cos \theta,$$

$$C_{64} = (I_{yy} - I_{zz})\dot{\psi} \cos^2 \theta \cos \phi \sin \phi - I_{xx}\dot{\theta} \cos \theta,$$

$$C_{65} = (I_{zz} - I_{yy})(\dot{\phi} \cos \theta \sin^2 \phi + \dot{\theta} \cos \phi \sin \phi \sin \theta) + (I_{yy} - I_{zz})\dot{\phi} \cos^2 \phi \cos \theta + I_{xx}\dot{\psi} \cos \theta \sin \theta - I_{yy}\dot{\psi} \sin^2 \phi \cos \theta \sin \theta - I_{zz}\dot{\psi} \cos^2 \phi \cos \theta \sin \theta,$$

$$C_{66} = (I_{yy} - I_{zz})\dot{\phi} \cos^2 \theta \sin \phi \cos \phi + I_{xx}\dot{\theta} \sin \theta \cos \theta + I_{yy}\dot{\theta} \sin^2 \phi \sin \theta \cos \theta + I_{zz}\dot{\theta} \cos^2 \phi \sin \theta \cos \theta,$$

$$C_{77} = L^2 m (-\sin(2\beta) \sin \alpha^2 \dot{\beta} - \sin(2\alpha) \dot{\alpha} + \cos \beta^2 \sin(2\alpha) \dot{\alpha} + \cos \alpha \sin \alpha \dot{\alpha}^2 - \cos \beta^2 \sin \alpha \cos \alpha \dot{\alpha}^2),$$

$$C_{78} = L^2 m (\cos \beta^2 \cos \alpha \sin \alpha \dot{\beta}),$$

$$C_{87} = L^2 m (\cos \alpha^2 \cos \beta \sin \beta \dot{\alpha}),$$

$$C_{88} = L^2 m (-\sin(2\alpha) \sin \beta^2 \dot{\alpha} - \sin(2\beta) \dot{\beta} + \cos \alpha^2 \sin(2\beta) \dot{\beta} + \cos \beta \sin \beta \dot{\beta}^2 - \cos \alpha^2 \sin \beta \cos \beta \dot{\beta}^2).$$

$$\mathbf{G}(q) = \begin{bmatrix} 0 & 0 & (M + m)g & 0 & 0 & 0 & mgL \sin \alpha \cos \beta & mgL \sin \beta \cos \alpha \end{bmatrix}^T, \quad (1.14)$$

$$\mathbf{b}(q) = \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & 0 & 0 & 0 \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & 0 & 0 & 0 \\ & \cos \phi \cos \theta & 0 & 0 & 0 \\ & & 0 & 1 & 0 & 0 \\ & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & & 0 & 0 & 0 & 0 \\ & & & & & & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.15)$$

El vector de entradas está dado por $\mathbf{u} = [f_q \ \tau]^T$, donde $f_q = f_1 + f_2 + f_3 + f_4$ es la fuerza de empuje total generada por los cuatro rotores. $\tau = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$, con τ_ϕ, τ_θ y τ_ψ son los torques de entrada para los ángulos roll, pitch and yaw , respectivamente.

Capítulo 2

Optimización

En este capítulo se detalla el procedimiento seguido para la identificación de parámetros y optimización de los controladores no lineales, así como la elección del algoritmo metaheurístico que será usado para el seguimiento de trayectoria del QUAV con carga suspendida.

2.1 Elección de algoritmo metaheurístico

Debido a la complejidad de diseñar e implementar un controlador en los QUAVs, se han propuesto múltiples alternativas, entre las que se encuentran las soluciones de naturaleza estocástica. El proceso de sintonización de controladores puede ser abordado como un problema de optimización sujeto a restricciones, para el que se han propuesto diferentes tipos de algoritmos para darle solución [16], [17]. Entre los más populares se encuentran los algoritmos metaheurísticos evolutivos y los basados en

población, que han demostrado efectividad al implementarse en problemas complejos [18].

En esta sección se presentan las ideas fundamentales detrás de cada uno de los algoritmos metaheurísticos a comparar para la sintonización de parámetros de un controlador. Los algoritmos implementados son: PSO, GWO, HGS, L-SHADE, LSPACMA, MPA, SMA y WOA.

Los algoritmos metaheurísticos evolutivos están inspirados en las leyes de evolución biológica, no conservan ni comparten información de la población de una evolución a otra, sin embargo, los mejores individuos se combinan para formar a la nueva generación de la población, con lo que al final quedarán siempre los mejores individuos. Entre los más populares se encuentran algoritmo genético (Genetic Algorithm, GA) [19], [20], [21] y el algoritmo de evolución diferencial (Differential Evolution, DE) [22].

Los algoritmos metaheurísticos basados en población están inspirados en el comportamiento colaborativo de grupos de individuos, con la habilidad de almacenar y compartir información entre los miembros de la población, lo que le permite crear conocimiento individual y colectivo, permitiendo encontrar una solución a partir de la mejora iterativa de las soluciones. Podemos mencionar al algoritmo de optimización por enjambre de partículas (Particle Swarm Optimization, PSO) [23], El Algoritmo de Optimización de Lobos (Grey Wolf Optimization, GWO) [24], el Algoritmo de Juegos del Hambre (Hunger Games Search, HGS) [25], los Algoritmos de Shade mejorados y extendidos (LSHADE y LSPACMA) [26, 27], Algoritmo de Predadores Marinos (Marine Predators Algorithm, MPA) [28], El Algoritmo de moho (Slime Mould Algorithm, SMA) [29], el Algoritmo de Optimización de Ballenas (Whale Optimization

Algorithm, WOA) [30].

2.1.1 Optimización por Enjambre de Partículas

El PSO es un método de optimización inspirado en el comportamiento que tienen las bandadas de pájaros o bancos de peces en los que, el movimiento de cada individuo (dirección, velocidad, aceleración), es el resultado de combinar las decisiones individuales de cada uno con el comportamiento del resto [23]. En el PSO, cada individuo es conocido como partícula que a su vez pertenece al enjambre o nube. La figura 2.1 describe el proceso que lleva a cabo el PSO, donde inicialmente se genera la población de partículas y se inicializan los valores de posición y velocidad; se evalúa la función costo o *fitness*, para dar una calificación a cada partícula, posteriormente se actualizan los vectores de velocidad y posición, mediante (2.1), (2.2),

$$v_{ij}(t+1) = wv_{ij}(t) + c_1R_1(P_{ij}(t) - x_{ij}(t)) + c_2R_2(P_{gj}(t) - x_{ij}(t)), \quad (2.1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (2.2)$$

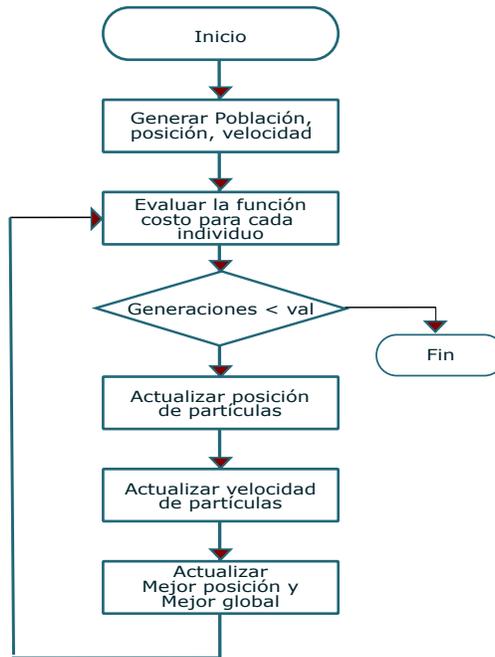


Figura 2.1: Diagrama de flujo del PSO.

donde:

- $v_{ij}(t+1)$, es la velocidad de la partícula i en el momento $t+1$, es decir, la nueva velocidad.
- $v_{ij}(t)$, es la velocidad de la partícula i en el momento (t) , es decir, la velocidad actual.
- w , es el coeficiente de inercia, reduce o aumenta la velocidad de la partícula.
- c_1 , es el coeficiente cognitivo.
- R_1 , es un vector de valores aleatorios.

- $P_{ij}(t)$, es la mejor posición en la que ha estado la partícula i hasta el momento.
- $x_{ij}(t)$, es la posición de la partícula en el momento t .
- c_2 , es el coeficiente social.
- R_2 , es un vector de valores aleatorios.
- $P_{gj}(t)$, es la mejor posición de todo el enjambre hasta el momento t , el mejor valor global.
- $x_{ij}(t + 1)$, es la posición de la partícula i en el momento $t + 1$, es decir, la nueva posición,

después de la actualización de posición y velocidad, se vuelve a evaluar el *fitness* y se repite el proceso hasta alcanzar el número de generaciones establecido.

2.1.2 Algoritmo de Optimización de Lobos

El Algoritmo GWO, está inspirado en el comportamiento de las manadas de lobos grises y su organización social para la caza de presas [24]. La manada se conforma de dos líderes llamados alfa (α) y el resto de la manada. En esta estructura social, los miembros alfa son los responsables de la toma de decisiones y el resto de los miembros debe seguir sus órdenes, dentro del algoritmo el alfa es la mejor solución. El segundo nivel en la jerarquía del grupo son los miembros beta (β), que deben colaborar con la toma de decisiones, seguidos por los delta (δ), estos son la segunda y tercera mejor solución, respectivamente. Existe una última clasificación, que se encuentra mas

abajo en la jerarquía de las manadas de lobos grises y se denominada omega (ω), la cual, está conformada por el resto de posibles soluciones. El diagrama de flujo de la Fig. 2.2, describe de manera gráfica el proceso de optimización del GWO.

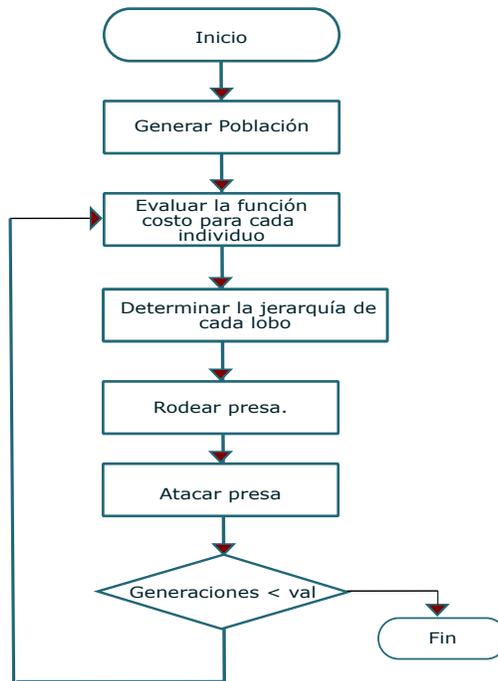


Figura 2.2: Diagrama de flujo del GWO.

El proceso de rodear a la presa se modela por las ecuaciones (2.3)- (2.4),

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \quad (2.3)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \quad \vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \quad \vec{C} = 2\vec{r}_2, \quad (2.4)$$

donde t indica la iteración actual, \vec{A} y \vec{D} son vectores coeficientes, $\vec{X}_p(t)$, es el vector

de posición de la presa, y $\vec{X}(t)$ es el vector de posición del lobo gris. El proceso de ataque a la presa, dirigido por el lobo alfa, se define por (2.5)- (2.7)

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (2.5)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_\beta = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_\delta = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta, \quad (2.6)$$

$$\vec{A} = 2\vec{a} \cdot r_1 - \vec{a}. \quad (2.7)$$

el proceso se repite hasta alcanzar el número de generaciones establecido.

2.1.3 Algoritmo de Juegos del Hambre

El algoritmo HGS, está inspirado en el comportamiento selectivo, así como las decisiones y acciones de los animales al momento de alimentarse. El HGS se diseña y emplea un peso adaptativo para simular el efecto del hambre en cada paso de búsqueda. Después, se modelan las reglas seguidas por la mayoría de animales al enfrentarse y adaptarse para aumentar las posibilidades de conseguir alimento y sobrevivir (games) [25].

- Approach food

Las ecuaciones mostradas en (2.8) representan los juegos de búsqueda de comida y de comunicación entre individuos y (2.9) describe la variación de posición entre los individuos, mientras que con (2.10) se va cerrando el radio de búsqueda disminuyendo gradualmente en cada iteración. La tabla 2.1, muestra las definiciones de cada una de las variables.

$$X_{ind}(t+1) = \begin{cases} \text{game 1 : } X_{ind}(t)(1 + randn(1)), & r_1 < l \\ \text{game 2 : } W_{hunger}X_{best} + RW_{hunger2}|X_{best} - X_{ind}(t)| & r_1 < l, r_2 > \Delta \\ \text{game 3 : } W_{hunger}X_{best} + RW_{hunger2}|X_{best} - X_{ind}(t)| & r_1 < l, r_2 < \Delta \end{cases} \quad (2.8)$$

$$\Delta = sech(|F(i) - F_{best}|), \quad i \in 1, 2, \dots, n \quad (2.9)$$

$$R = 2sc - s \quad (2.10)$$

Tabla 2.1: Parámetros del algoritmo HGS: aproach food

Parámetro	Significado
$X_{ind}(t)$	Ubicación de cada individuo
$randn(1)$	Número aleatorio que satisface la distribución normal
W_{hunger}	Influencia del peso de hambre en el rango de actividad: $ X_{best} - X_{ind}(t) $
$W_{hunger2}$	Influencia del peso de hambre 2 en el rango de actividad: $ X_{best} - X_{ind}(t) $
X_{best}	Ubicación del mejor individuo en la iteración actual
R	Es una ganancia dentro del rango de $[-a, a]$, este valor es el radio de búsqueda y va disminuyendo gradualmente hasta 0
r_1, r_2	Números aleatorios, dentro del rango de $[0, 1]$
l	Valor asignado para mejorar el algoritmo
Δ	Control de variación de posiciones
F	Valor resultado de evaluar la función costo de cada individuo (Fitness)
F_{best}	Mejor valor alcanzado hasta la iteración actual
$sech$	Función hiperbólica $\frac{2}{e^x - e^{-x}}$
s	Factor de reducción del espacio $2(1 - \frac{t}{t_{max}})$
t	Iteración actual
t_{max}	Número máximo de iteraciones
c	Número aleatorio en $[0,1]$

- Hunger role

La búsqueda de los individuos motivados por el hambre, se simula por el modelo propuesto matemáticamente mostrado en (2.11) y (2.12), mientras que el valor del

hambre se actualiza con la (2.13), asignandole 0 al individuo con el mejor valor de fitness y para el resto de individuos su valor actual mas H, que se actualiza mediante (2.14) y su límite superior se calcula mediante (2.15). La tabla 2.2, muestra las definiciones de cada una de las variables.

$$W_{hunger}(i) = \begin{cases} hungry(i) \frac{N}{\sum hungry} r_4 & , r_3 < l \\ 1 & , r_3 > l \end{cases} \quad (2.11)$$

$$W_{hunger2}(i) = (1 - e^{-|hungry(i) - \sum hungry|}) 2r_5, \quad (2.12)$$

$$hungry(i) = \begin{cases} 0, & F_{all} == F_{best} \\ hungry(i) + H, & F_{all} \neq F_{best}, \end{cases} \quad (2.13)$$

$$H = \begin{cases} H_{low}(1 + r), & H_{top} < H_{low} \\ H_{top}, & H_{top} \geq H_{low}, \end{cases} \quad (2.14)$$

$$H_{top} = \frac{F(i) - F_{best}}{F_{worst} - F_{est}} 2r_6 (B_{up} - B_{low}). \quad (2.15)$$

En la Figura 2.3, se muestra la secuencia en que el algoritmo HGS calcula cada una de las acciones basadas en el hambre, para la optimización. Podemos resaltar que después de la inicialización de parámetros y de calcular las posiciones o posibles soluciones, el algoritmo verifica en cada iteración, que los valores se encuentren dentro del los límites del espacio de búsqueda para acotarla. El proceso se repite hasta alcanzar el número de generaciones establecido.

Tabla 2.2: Parámetros del algoritmo HGS: hunger role.

Parámetro	Significado
N	Número de individuos
$\sum hungry$	Sumatoria del nivel de hambre de todos los individuos
r_3, r_4, r_5, r_6	Números aleatorios, dentro del rango de $[0, 1]$
l	Valor asignado para mejorar el algoritmo
$F(i)$	Valor resultado de evaluar la función costo de cada individuo (Fitness)
F_{best}	Mejor valor alcanzado hasta la iteración actual
F_{worst}	Peor valor alcanzado hasta la iteración actual
F_{all}	Vector de valor alcanzado hasta la iteración actual de todos los individuos
H_{low}	Valor límite inferior del hambre
H_{top}	Valor límite superior del hambre
B_{up}	Límite superior del espacio de búsqueda
B_{low}	Límite inferior del espacio de búsqueda

2.1.4 Algoritmo L-SHADE

El Algoritmo L-SHADE es una mejora del algoritmo SHADE, el cual a su vez, es una variante del Algoritmo evolutivo Diferencial (DE), que agrega un parámetro de adaptación basado en el historial de éxito. El Algoritmo L-SHADE, extiende al SHADE [22], mediante una reducción lineal del tamaño de la población (LPSR), con lo que la población decrece de manera continua, siguiendo el comportamiento de una función lineal [31]. EL Algoritmo L-SHADE sigue la secuencia mostrada en la Fig. 2.4 y ha sido implementado para resolver funciones de prueba y se ha comparado contra algoritmos como el SHADE y otras variantes del DE, reportando una mejora significativa en el desempeño [27].

2.1.5 Algoritmo LSPACMA

Este algoritmo es una mejora del L-SHADE, agregando un enfoque adaptativo para la selección de parámetros de control. El LSPACMA, utiliza la capacidad de adaptarse

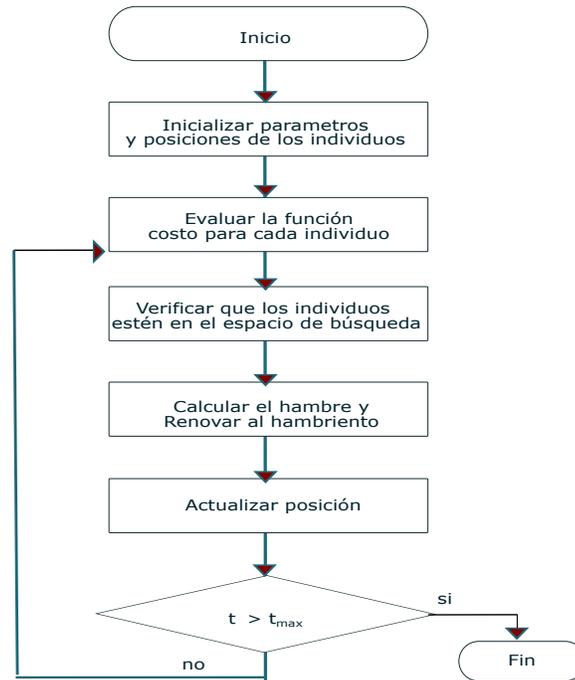


Figura 2.3: Secuencia de acciones basadas en el Hambre. Algoritmo HGS.

para ajustar la evolución diferencial del algoritmo. El enfoque propuesto consta de dos diferentes ajustes para los parámetros de control. En la primera mitad de la ejecución del algoritmo los parámetros de búsqueda se ajustan de manera adaptable, mientras que en la segunda mitad, lo hacen de manera aleatoria. El beneficio de esta semi-adaptación demuestra que el algoritmo supera a algoritmos aleatorios y algoritmos auto-adaptativos cuando se implementa para diferentes funciones prueba [32]. EL Algoritmo LSPACMA sigue la secuencia mostrada en la Fig. 2.5 .

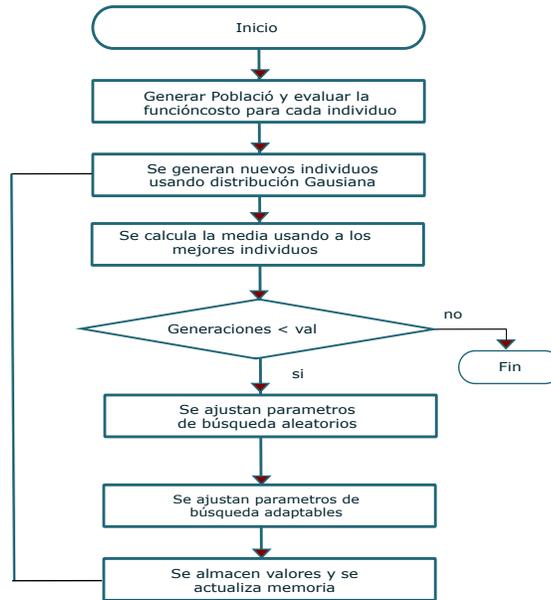


Figura 2.4: Diagrama de flujo del L-SHADE.

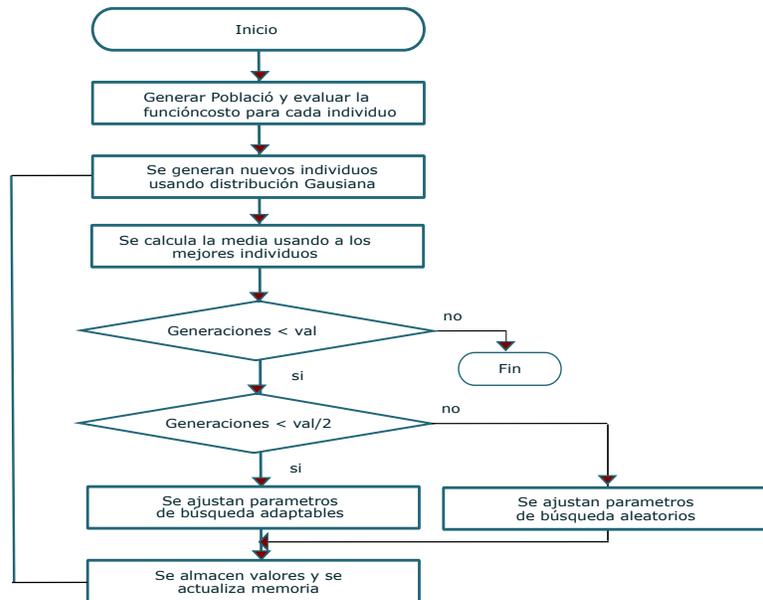


Figura 2.5: Diagrama de flujo del LSPACMA.

2.1.6 Algoritmo de Predadores Marinos

El algoritmo MPA está inspirado en la teoría de supervivencia del más apto, donde los depredadores deben elegir una estrategia para maximizar el encuentro con las presas en su entorno. Estas estrategias evolucionan en el ecosistema y son naturalmente seleccionadas por los depredadores para sobrevivir [28].

Existe un tipo de movimiento aleatorio que se denomina vuelo/caminata de Lévy, la cual se basa en una teoría de búsqueda óptima, los cuales, describen a movimientos de pasos pequeños con desplazamientos largos. Muchos animales y criaturas marinas siguen un patrón de vuelo/caminata de Lévy como su política de búsqueda óptima. El Algoritmo MPA sigue la secuencia mostrada en la Fig. 2.6, la cual se repite hasta alcanzar el número de iteraciones establecidas.

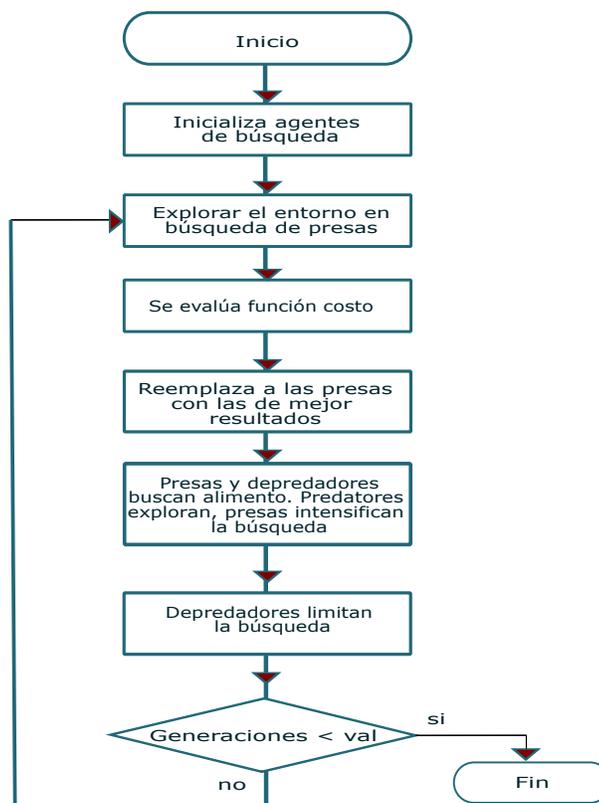


Figura 2.6: Diagrama de flujo del MPA.

2.1.7 Algoritmo SMA

El algoritmo del moho, SMA, imita el el proceso en que el moho produce la propagación positiva y negativa en forma de oscilaciones para formar una trayectoria óptima combinando la habilidad de exploración y explotación que lleva a cabo para obtener nutrientes o alimentarse [29]. Las tres fases para actualizar los parámetros del algoritmo son: 1) Aproximarse a la comida, 2) envolver comida, y 3) oscilación. Estas tres fases se repiten cada generación (ver Fig. 2.7) y se modelan mediante las ecuaciones

(2.16)-(2.21).

$$X_{ind}(t+1) = \begin{cases} Xb(t) + vb \cdot (W \cdot X_A(t) - X_B(t)), r < p \\ vc \cdot X(t), r \geq p \end{cases}, \quad (2.16)$$

$$p = \tanh|S(i) - DF|, \quad (2.17)$$

$$vb = [-a, a], \quad (2.18)$$

$$a = \arctan\left(-\left(\frac{t}{max_t}\right) + 1\right), \quad (2.19)$$

$$W(Smellindex(i)) = \begin{cases} Xb1 + r \cdot \log\left(\frac{b^F - S(i)}{b^F - w^F} + 1\right), \text{condicion} \\ 1 - r \cdot \log\left(\frac{b^F - S(i)}{b^F - w^F} + 1\right), \text{otro} \end{cases}, \quad (2.20)$$

$$Smellindex = \text{sort}(S), \quad (2.21)$$

donde vb es un parámetro en rango de y $[-a, a]$, vc decrece linealmente de 1 a 0. t representa la iteración actual, X_b representa la ubicación del individuo con la concentración de olor mas alta, X representa la ubicación del moho, X_A y X_B representan a dos individuos aleatorios, W representa el peso de los individuos. $S(i)$ califica a los individuos, r es un valor aleatorio entre $[0, 1]$, max_t muestra la iteración máxima o el valor de paro del algoritmo, b^F es el mejor valor obtenido en la iteración actual, w^F denota el peor valor obtenido. $Smellindex$ denota la secuencia de valores obtenidos ordenados de menos a mayor. EL proceso se repite hasta alcanzar el numero de iteraciones establecido.

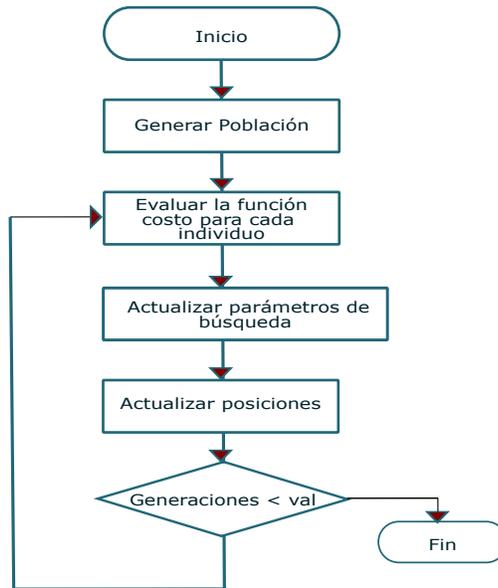


Figura 2.7: Diagrama de flujo del SMA.

2.1.8 Algoritmo de Optimización de Ballenas

El Algoritmo WOA, está inspirado en el comportamiento de cacería de las ballenas, conocido como estrategia de red de burbujas, que consiste en seleccionar una muestra aleatoria de posibles soluciones que circundan al área donde se encuentra el objetivo, repitiendo el proceso actualizando las posiciones en torno al mejor obtenido y así sucesivamente, hasta satisfacer el criterio de búsqueda [30]. El diagrama de flujo del WOA, se observa en la Fig. 2.8. Las ecuaciones que describen el comportamiento de rodear a la presa se describen en (2.22), (2.23)

$$\vec{D} = |\vec{C} \cdot \vec{X}_{best}(t) - \vec{X}(t)|, \quad \vec{D}' = |\vec{X}_{best}(t) - \vec{X}(t)| \quad (2.22)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \quad \vec{C} = 2 \cdot \vec{r}, \quad (2.23)$$

$$\vec{X}(t+1) = \vec{X}_{best}(t) - \vec{A} \cdot \vec{D}, \quad (2.24)$$

$$\vec{X}(t + 1) = \vec{D}' \cdot e^{bl} \cdot \cos 2\pi l + \vec{X}_{best}(t), \quad (2.25)$$

los movimientos de la estrategia de burbuja es descrito en (2.24) como mecanismo de cerco, y se obtiene reduciendo el valor de a de 2 a 0, ó mediante (2.25) que modela un movimiento en espiral, con b constante y l un número aleatorio. El proceso se repite hasta alcanzar el número de generaciones establecidas.

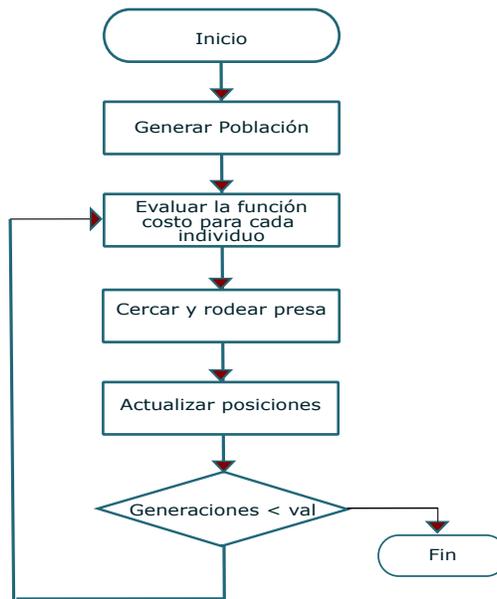


Figura 2.8: Diagrama de flujo del WOA.

2.2 Proceso de optimización de un controlador

La figura 2.9, muestra el esquema implementado para la sintonización de ganancias de un controlador PID con compensación de dinámicas, donde se evalúa cada uno de los algoritmos metaheurísticos. El mismo proceso es implementado para el controlador ST-SMC. Inicialmente el algoritmo es evaluado y entrega un conjunto de ganancias,

las cuales se aplican al controlador diseñado, quien a su vez, calcula las señales de control para el QUAV, posteriormente se mide el desempeño dinámico de cada uno de los ejes X, Y y Z , para finalmente calcular la raíz del error cuadrático medio (RMSE) entre las dinámicas deseadas y las obtenidas. Este ciclo se repite hasta que el número de iteraciones determinado inicialmente es alcanzado o hasta que el error o función costo alcanza un valor establecido.



Figura 2.9: Esquema de sintonización de ganancias del controlador.

Para obtener las ganancias k_p , k_i y k_d de cada una de las 6 dinámicas del QUAV, mediante los algoritmos evolutivos descritos en 2.1, la población a optimizar de un controlador PID es:

$$P = \{k_{p_x} \ k_{i_x} \ k_{d_x} \ k_{p_y} \ k_{i_y} \ k_{d_y} \ k_{p_z} \ k_{i_z} \ k_{d_z} \ k_{p_\phi} \ k_{i_\phi} \ k_{d_\phi} \ k_{p_\theta} \ k_{i_\theta} \ k_{d_\theta} \ k_{p_\psi} \ k_{i_\psi} \ k_{d_\psi}\}, \quad (2.26)$$

y la población para la optimización del controlador ST-SMC es:

$$P = \{\beta_x \ k_{1_x} \ k_{2_x} \ \beta_y \ k_{1_y} \ k_{2_y} \ \beta_z \ k_{1_z} \ k_{2_z} \ \beta_\phi \ k_{1_\phi} \ k_{2_\phi} \ \beta_\theta \ k_{1_\theta} \ k_{2_\theta} \ \beta_\psi \ k_{1_\psi} \ k_{2_\psi}\}, \quad (2.27)$$

siendo el espacio de búsqueda para cada uno de los individuos $S = [0, 10]$. La función

costo calculada mediante el error cuadrático medio:

$$ff = \sqrt{\frac{1}{N} \sum_{i=1}^G \sum_{K=0}^N (\chi_d^i(k) - \chi^i(k))^2}, \quad (2.28)$$

donde N es el número de datos de χ y G , representa a cada una de las dinámicas $x, y, z, \phi, \theta, \psi$ del QUAV.

2.3 Resultados del proceso de optimización de un controlador: comparación de los diferentes algoritmos Metaheurísticos

La implementación de los algoritmos se realizó en una Mac OS Big Sur (Intel core i9, RAM 24 GB, y 1 TB de disco duro), usando Matlab 2015a, para cada uno de los algoritmos.

Usando los parámetros de simulación que se muestran en la Tabla 2.3, se implementan los algoritmos para la sintonización de ganancias del controlador descrito en 3.1. Los algoritmos PSO, GWO, HGS, LSHADE, LSPACMA, MPA, SMA y WOA, no tienen parámetros adicionales que tengan que ser ajustados para la implementación. Los parámetros físicos de QUAV empleados para la evaluación de la función costo se muestran en la Tabla 2.3. Las posiciones y valores de ángulos iniciales se fijan como $[0, 0, 0]m$ y $[0, 0, 0]rad$.

El cambio constante en la referencia deseada $[x_d, y_d, z_d]$, provoca que el controlador

demande cambios igualmente en los ángulos deseados $[\phi_d, \theta_d]$ y aumenta la complejidad a los algoritmos de encontrar las ganancias del controlador.

En la figura 2.10, se muestra el desempeño de cada uno de los algoritmos al paso de las 20 iteraciones, con un número de individuos de 20, 30 y 40. Puede observarse que todos los algoritmos convergen de una manera similar, sin embargo el algoritmo HGS en los tres experimentos con diferente número de individuos, alcanza el menor valor de RMSE. En las tablas 2.4 y 2.5, se muestran los parámetros optimizados por cada algoritmo y la estadística de desempeño de cada uno, para el experimento con 20 individuos, 20 iteraciones y 30 repeticiones. En las tablas 2.6 y 2.7, se muestran los parámetros optimizados por cada algoritmo y la estadística de desempeño de cada uno, para el experimento con 30 individuos, 20 iteraciones y 30 repeticiones. De igual manera, en las tablas 2.8 y 2.9, se muestran los parámetros optimizados por cada algoritmo y la estadística de desempeño de cada uno, para el experimento con 40 individuos, 20 iteraciones y 30 repeticiones. Podemos observar que aumentar el número de individuos no representa una mejora significativa en la minimización del RMSE, por lo que para la demostración del desempeño de las dinámicas con los parámetros obtenidos, únicamente se evalúan los encontrados con 20 individuos, 20 iteraciones y 30 repeticiones.

En las figuras 2.11-2.16, se muestra la implementación a las dinámicas del QUAV de las diferentes ganancias obtenidas con cada uno de los algoritmos, se puede observar que el mejor desempeño lo obtiene el algoritmo HGS. Las dinámicas traslaciones pueden observarse en las Figuras 2.11 y 2.12, donde las dinámicas obtenidas mediante la implementación de las ganancias de cada uno de los algoritmos metaheurísticos al-

canzan a las trayectorias deseadas. La Fig. 2.13 muestra el desempeño de la dinámica de altitud, en donde los valores negativos al inicio del gráfico son debido a la compensación de la fuerza de gravedad, la dinámica presenta valores negativos hasta el momento en que la señal de control supera al valor de la gravedad. Las Fig. 2.14-2.16 muestra el comportamiento de los ángulos de orientación. Puede observarse que las dinámicas responden a los cambios de dirección de las trayectorias traslacionales deseadas.

La figura 2.17 muestra el desempeño del QUAV en seguimiento de trayectoria en tres dimensiones, donde se puede observar que el controlador lleva al vehículo por la trayectoria deseada. Finalmente en la figura 2.18 se implementó una trayectoria distinta a la que se empleó para la optimización y se observa que el controlador se desempeña correctamente llevando al QUAV a la referencia.

Tabla 2.3: Parámetros físicos de modelo de QUAV.

Parámetro	Valor
Masa	0.5 Kg
Tensor de Inercia	$I_x = 0.005, I_y = 0.005, I_z = 0.010$
Coefficiente de arrastre traslacional	$k_{1,2,3} = 5.56 \times 10^{-04}$
Coefficiente de fricción aerodinámicos	$k_{4,5,6} = 0.3729$
Aceleración de la gravedad	$9.81 m/s^2$

Tabla 2.4: Estadística de desempeño de cada algoritmo: RMSE con 20 individuos, 20 evoluciones and 30 repeticiones.

Statistics	PSO	GWO	HGS	LSHADE	LSPACMA	MPA	SMA	WOA
Best	0.037396168112890	0.0373629558560019	0.0372635355063979	0.038978058766757	0.0397411105315137	0.0372842398991813	0.037835061121072	0.037370940741738
Mean	0.054376357228900	0.043954323277539	0.045379957656500	0.052582989549809	0.050922002950839	0.038978646154916	0.051468628827586	0.068998262466992
SD	0.018193244089308	0.018827354037126	0.012805236930367	0.014038286466383	0.008042296173043	0.001885601376063	0.017924632527572	0.103765679125065
Worst	0.074792336225780	0.114069824231860	0.074527071012758	0.091325532418241	0.079482210630274	0.044105573338656	0.107284349261279	0.615170818395657

Tabla 2.5: Parámetros obtenidos por cada algoritmo metaheurístico con 20 individuos, 20 evoluciones y 30 repeticiones.

Parameters	PSO	GWO	HGS	LSHADE	LSPACMA	MPA	SMA	WOA
K_{ps}	5.408936203613887	5.213781792960648	5.412177273755431	5.715231175228559	6.620122525285044	5.481978926698987	5.711436859433793	5.437501487917403
K_{dx}	10.000000000000000	10.000000000000000	10.000000000000000	9.323716854587114	9.993641853644958	9.999863024805832	10.000000000000000	10.000000000000000
K_{pp}	5.096583444010690	4.891463309029316	5.36888612384497	4.877734569229675	3.772435488131499	4.896793864900809	4.548312153197413	6.315113281984011
K_{dp}	10.000000000000000	10.000000000000000	10.000000000000000	9.800086546316058	9.678338787974745	10.000000000000000	10.000000000000000	10.000000000000000
K_{ps}	10.000000000000000	10.000000000000000	9.950128906195692	9.590718062002178	6.771555264587359	9.95855577339498	9.771978015658293	10.000000000000000
K_{dx}	8.285393139094424	8.260159603944022	10.000000000000000	9.722104767452858	9.101622173737571	10.000000000000000	5.351259401838421	10.000000000000000
K_{pp}	7.058540895556090	4.822381113908009	4.709902665404981	1.42705042025854	9.962956330944161	6.546159756777394	2.246128817422761	10.000000000000000
K_{dp}	4.943026175684162	8.194215866646751	5.068174591175525	2.123783954799148	3.186248449818443	9.905810957096076	4.860624932515171	10.000000000000000
K_{pp}	4.844306519002986	4.7841384101376938	4.821175176721279	4.895525445351029	5.247032886854611	4.916081555171791	5.047194369837532	4.711197637806473
K_{dp}	0	0	0.000035339909776	0.000244462522990	0.060372557625467	0.000127362385910	0.000091054160333	0
K_{ps}	5.355372158301154	4.069504903320026	0.005334006148015	3.303332611822569	5.476771414363572	7.773619983483155	0.949746659092169	10.000000000000000
K_{dp}	3.808860161714160	0.175196233025389	0.043112681853079	0.317132270794582	7.891062763173645	9.976206665270322	0.361836747670447	8.120054392303674

Tabla 2.6: Estadística de desempeño de cada algoritmo: RMSE con 30 individuos, 20 evoluciones y 30 repeticiones.

Statistics	PSO	GWO	HGS	LSHADE	LSPACMA	MPA	SMA	WOA
Best	0.037298689699719	0.037386364140826	0.037250937439961	0.038703382913285	0.039391601057574	0.037805699738380	0.037610074434848	0.037526810582112
Median	0.049292829641084	0.0388231926452584	0.181468520773043	0.048098319555092	0.048447470240744	0.0384393858577506	0.040600753887036	0.106439604511052
SD	0.016922702608194	0.000917745219989	0.480087752312157	0.008671308416183	0.007749711491518	0.001368117884585	0.006458484603516	0.164512311240661
Worst	0.074597379399439	0.040928476103477	1.972825126787748	0.077406765826570	0.078783202115148	0.043410168070117	0.066852769977100	0.608263592411619

Tabla 2.7: Parámetros obtenidos por cada algoritmo metaheurístico con 30 individuos, 20 evoluciones y 30 repeticiones.

Parameters	PSO	GWO	HGS	LSHADE	LSPACMA	MPA	SMA	WOA
k_{ps}	5.321480404491486	5.316121848678594	5.506178152902620	5.573040492118754	6.456895021660555	5.597288673172770	5.357830158921898	5.985862339317787
k_{ds}	10.000000000000000	10.000000000000000	10.000000000000000	9.781398388783654	9.838517000280444	9.999480002823272	10.000000000000000	10.000000000000000
k_{pp}	5.360362063749348	5.022574636552078	5.575896169072749	7.045066726522021	6.373778110323427	5.777679946622201	5.492717168284171	5.831980752133919
k_{dp}	10.000000000000000	10.000000000000000	10.000000000000000	9.732603652525405	9.441276780545653	10.000000000000000	10.000000000000000	10.000000000000000
k_{ps}	10.000000000000000	9.427393279440687	10.000000000000000	8.640471589965724	8.362856784741542	10.000000000000000	9.982166969934834	10.000000000000000
k_{ds}	9.467313952657385	8.846939435077033	10.000000000000000	8.419482408166351	9.849967164971744	9.958303050280419	6.636230464386927	10.000000000000000
k_{pp}	6.814063438272202	3.228097695082624	4.541639604890506	6.863481745971876	8.796217915815690	9.847393024662287	10.000000000000000	5.796906949596673
k_{dp}	5.874995512822094	8.203547303875149	7.158872343937848	5.435527296456358	4.736415620880451	9.994682117338073	5.067661699504412	10.000000000000000
k_{ps}	4.819432444500308	4.774213565304394	4.861996886493411	5.066511011257896	4.508388434244966	4.900504250374575	4.697810473694974	4.624341847012862
k_{ds}	0	0	0	0.021053555495621	0.012780497660689	0	0	0
k_{pp}	6.781239359799283	0.961791842657223	0.814480987696878	6.972615636034867	3.964876907491253	9.875696545163565	6.784589697287003	4.717551680873766
k_{dp}	6.535849274794208	2.450515048981552	6.353764374969330	1.691008581444079	8.632031591890170	9.95088788897011	0	4.594521890201812

Tabla 2.8: Estadística de desempeño de cada algoritmo: RMSE con 40 individuos, 20 evoluciones y 30 repeticiones.

Statistics	PSO	GWO	HGS	LSHADE	LSPACMA	MPA	SMA	WOA
Best	0.037255373533488	0.037266015064895	0.037247252379126	0.039394303548209	0.038830921234619	0.037256592611999	0.037632310844408	0.037353819712401
Mean	0.045525283051302	0.041601797209679	0.066221944130588	0.043549315402296	0.045455785193484	0.038150710592382	0.039375241940074	0.0439285585596090
SD	0.012064390608393	0.010639844056084	0.154265972701073	0.003114729198649	0.005768092569169	0.0011526441167111	0.002176578772137	0.005084363577577
Worst	0.074392122335153	0.084154455562805	0.882698952672554	0.051704880477973	0.062767121112513	0.043218862736804	0.045909805573245	0.059239414200638

Tabla 2.9: Parámetros obtenidos por cada algoritmo metaheurístico con 40 individuos, 20 evoluciones y 30 repeticiones.

Parámetros	PSO	GWO	HGS	LSHADE	LSPACMA	MPA	SMA	WOA
k_{ps}	5.387775921956180	5.355284009147565	5.381403859043313	5.513337891603976	5.234732646639856	5.425915525332709	4.965902929744495	5.821726954392787
k_{ds}	10.000000000000000	10.000000000000000	10.000000000000000	9.933746539383339	9.733042357851929	10.000000000000000	10.000000000000000	10.000000000000000
k_{pp}	5.503169448530415	5.15765253295196	5.385842705691601	7.630413711770860	5.066585315120156	5.431085259751310	6.231850241856680	5.339150052042956
k_{dp}	10.000000000000000	10.000000000000000	10.000000000000000	9.112575074225544	9.601563597114865	9.99805203324547	10.000000000000000	10.000000000000000
k_{ps}	10.000000000000000	10.000000000000000	10.000000000000000	8.452040812542801	8.276025623946069	10.000000000000000	9.757836996055245	10.000000000000000
k_{ds}	10.000000000000000	10.000000000000000	9.999937779326700	7.857066737354142	8.533283889078021	9.989924454194904	9.979985496334406	10.000000000000000
k_{pp}	5.509435611867958	6.302627444914116	6.386994221918801	7.928849397796594	7.473150998878030	8.009573766497542	4.558718104204019	10.000000000000000
k_{dp}	9.432035226553529	8.636335189288980	9.788907844769838	8.628408426770227	9.597213064062426	9.678365436395564	1.593956521679671	10.000000000000000
k_{pp}	4.794923996582154	4.910674065557497	4.826014474662918	5.075458800622620	4.630441901688464	4.82227320881963	4.753962189265832	4.763044802438290
k_{dp}	0	0	0	0.008017685879173	0.027765559630236	0.000003518624809	0	0
k_{ps}	7.5116333082929468	3.018192575071655	1.76632827277126	1.903434392477181	8.539208296636906	9.998471291768176	9.814218990526523	9.362558661978090
k_{ps}	7.963122860297110	3.360333706177101	0.191715360093252	4.441115912390526	6.058651260609002	8.920397598071631	3.400723395395384	8.585227416173138

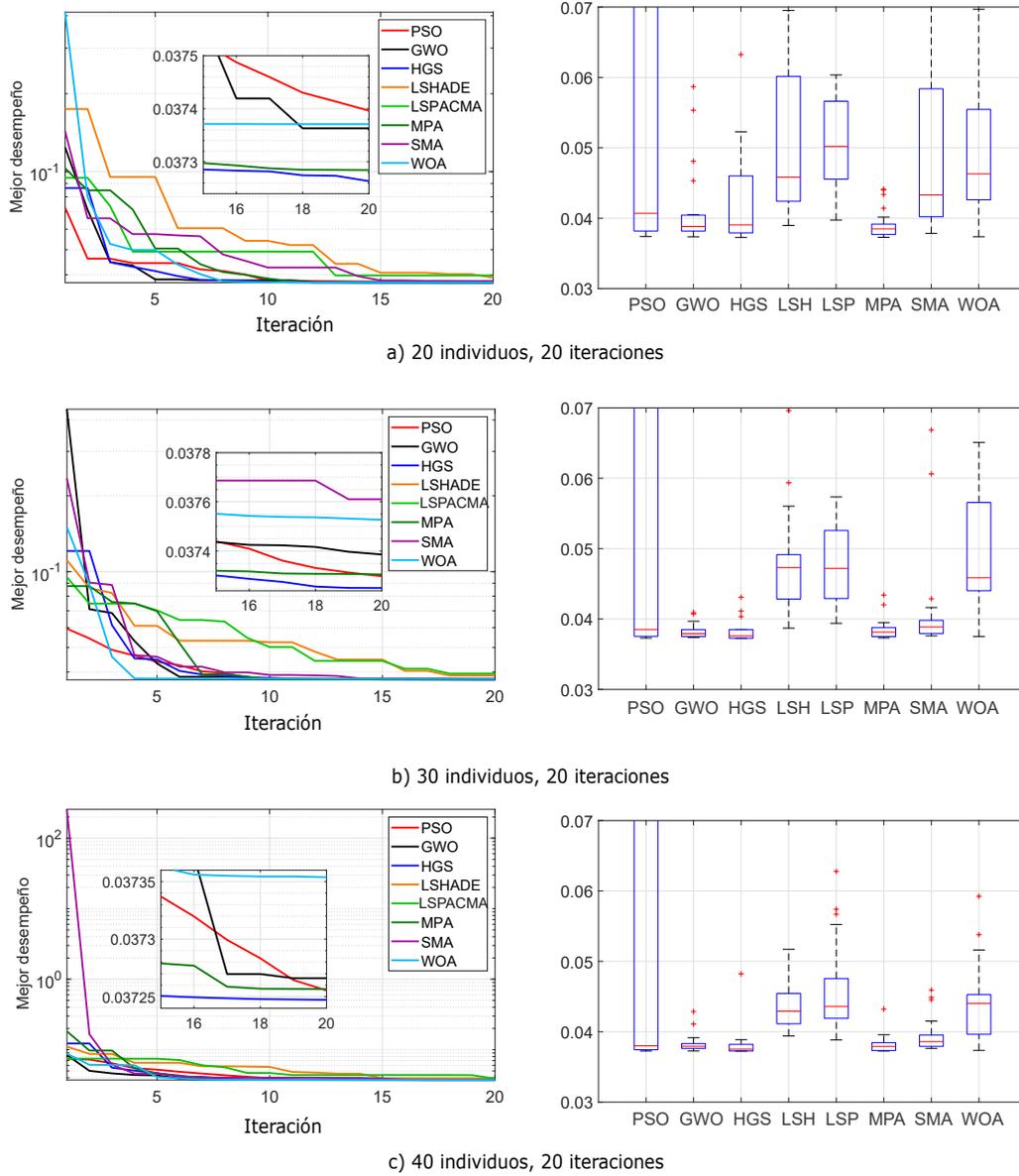


Figura 2.10: Desempeño de los algoritmos evolutivos en cada evolución.

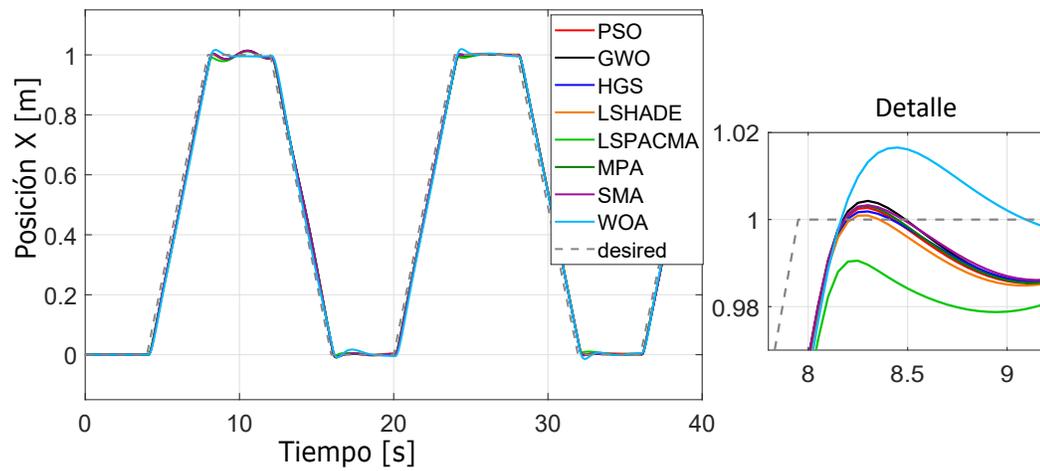


Figura 2.11: Comparación del desempeño del QUAV en el eje X, evaluando los parámetros obtenidos con los diferentes algoritmos, empleando 20 individuos.

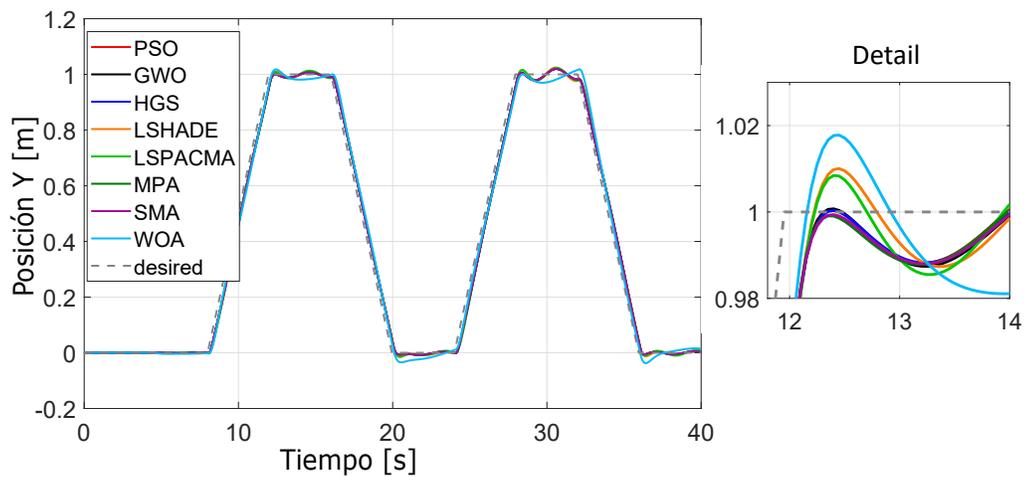


Figura 2.12: Comparación del desempeño del QUAV en el eje Y, evaluando los parámetros obtenidos con los diferentes algoritmos, empleando 20 individuos.

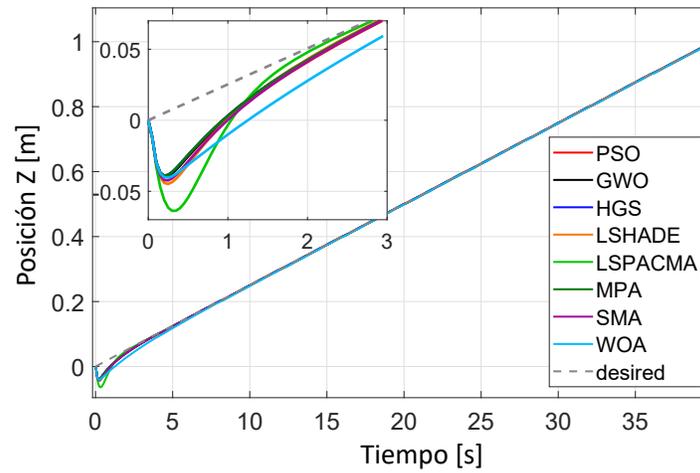


Figura 2.13: Comparación del desempeño del QUAUV en el eje Z, evaluando los parámetros obtenidos con los diferentes algoritmos, empleando 20 individuos.

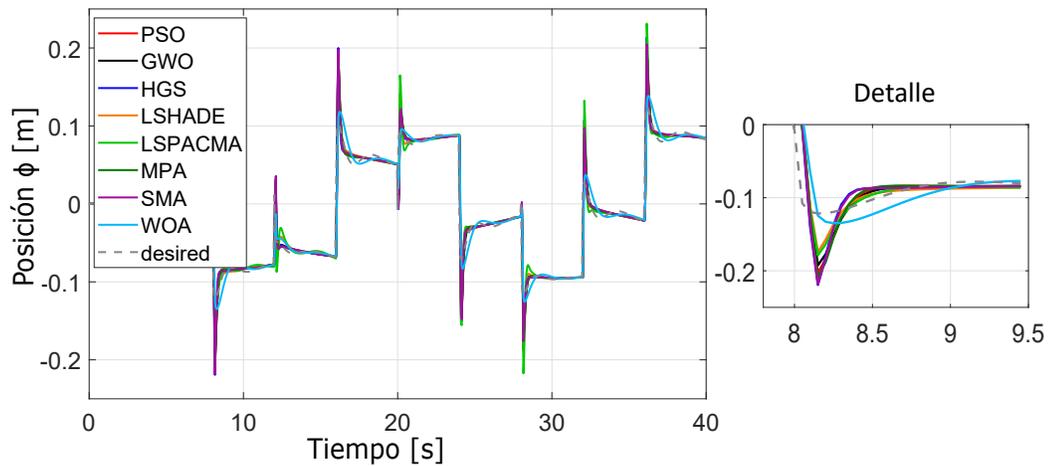


Figura 2.14: Desempeño de los ángulos ϕ y θ , contra los ángulos deseados calculados mediante el control virtual.

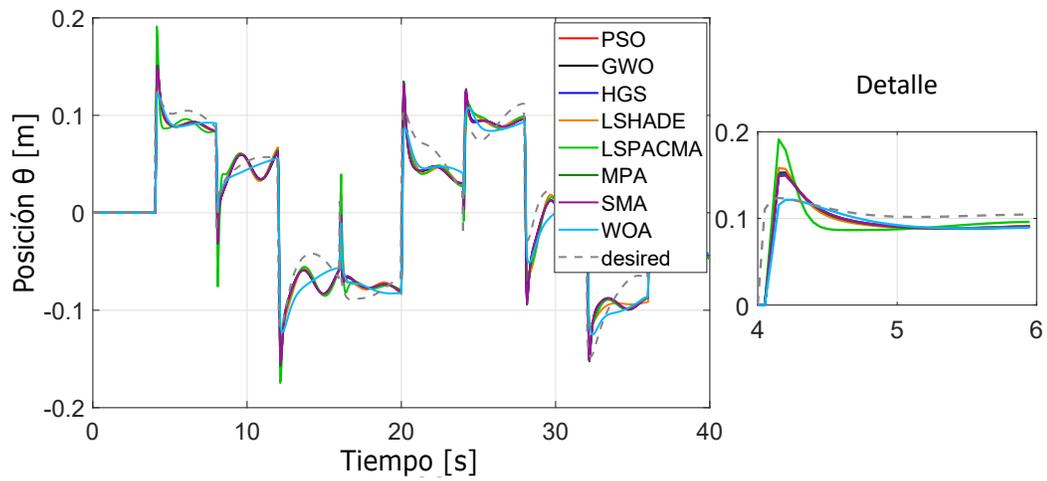


Figura 2.15: Desempeño de los ángulos ϕ y θ , contra los ángulos deseados calculados mediante el control virtual.

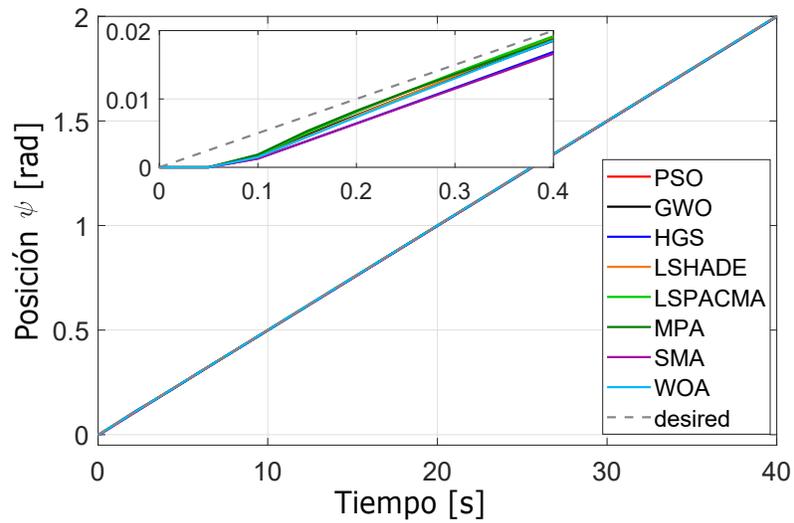


Figura 2.16: Desempeño del ángulo ψ , contra el ángulo deseado.

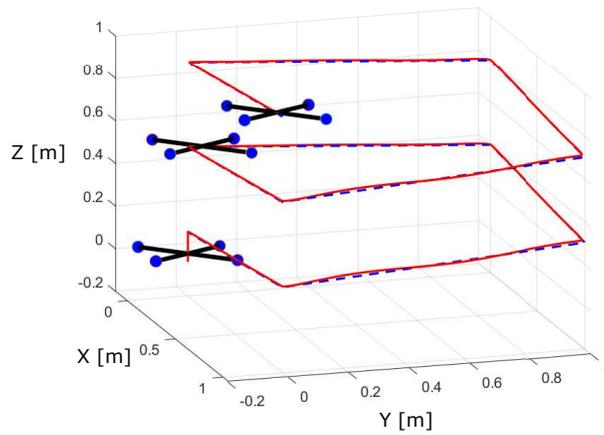


Figura 2.17: Seguimiento de trayectoria del QUAV en X - Y - Z.

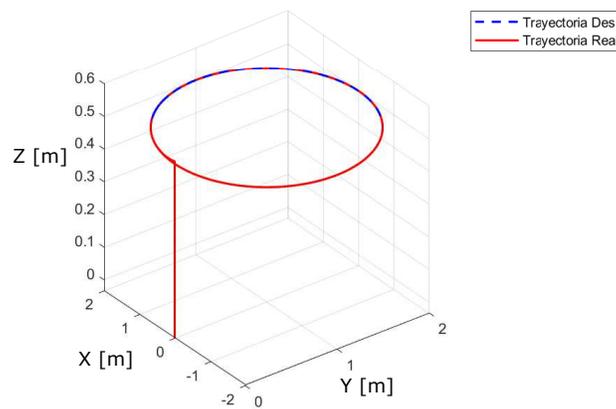


Figura 2.18: Evaluación de diferente trayectoria usando las ganancias obtenidas mediante optimización.

Capítulo 3

Control

En este capítulo se lleva a cabo la elección del controlador que lleve al QUAV a la trayectoria deseada en el espacio tridimensional. Primero, se explica la estructura propuesta y se detalla el diseño de cada uno de los controladores PID y ST-SMC. Se implementa el algoritmo metaheurístico HGS con la finalidad de identificar de parámetros óptimos de cada controlador. Se emplea únicamente el algoritmo HGS, ya que en el Capítulo 2, se encontró que es el que obtiene un valor menor al minimizar el error RMSE entre la trayectoria deseada y la real. Para la elección del controlador, se compara la implementación en simulación y física de ambos algoritmos optimizados mediante HGS. Adicionalmente, para evaluar la robustez de los algoritmos se agrega una perturbación externa al QUAV.

3.1 Estructura de control

El objetivo del control es llevar al QUAUV a la referencia deseada en X - Y - Z. Como se muestra en la Figura 3.1, las posiciones deseadas x_d, y_d, z_d se introducen al Control Traslacional y de Altura que calcula la señal de control u_1 , y genera los ángulos deseados ϕ_d, θ_d , los cuales son las señales de referencia para el Control de Orientación. Las señales de control son aplicadas a las dinámicas del QUAUV, las cuales a su vez, se realimentan a los controles de traslación y altura y de orientación. Este proceso se repite, hasta que la trayectoria deseada se ejecute en su totalidad.

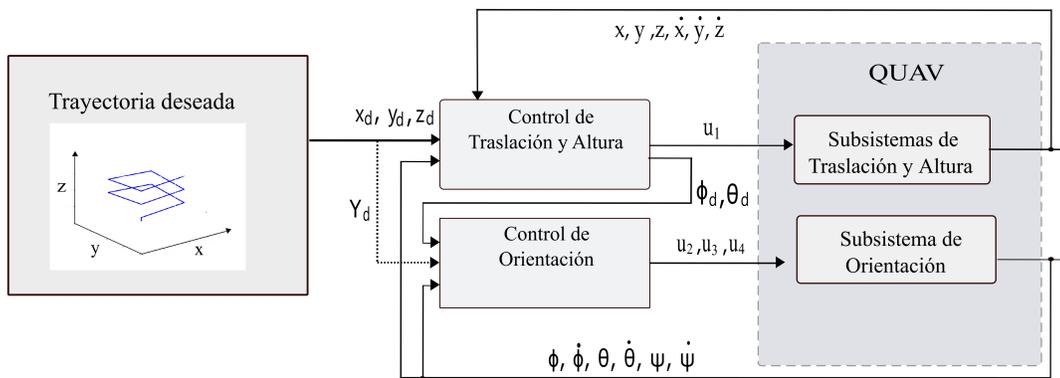


Figura 3.1: Estructura de control propuesta para seguimiento de trayectoria.

Debido a la naturaleza no lineal del sistema QUAUV, se proponen un controlador PID compensado [33] y un controlador por modos deslizantes ST-SMC, para seguimiento de trayectoria [34].

3.2 Diseño de Control PID

Las dinámicas del QUAV, mostradas en (1.3) se pueden representar como un sistema de segundo orden, de la forma

$$\ddot{\chi} = f(\boldsymbol{\chi}) + g(\boldsymbol{\chi})u, \quad (3.1)$$

donde $f(\boldsymbol{\chi})$, en este trabajo, se asume que es únicamente una función no lineal debido a las fuerzas de arrastre del QUAV, $\boldsymbol{\chi} = [\chi, \dot{\chi}]$, $g(\boldsymbol{\chi}) \neq 0$ es una función conocida u es la señal de control. La ley de control se diseña como

$$u = \frac{1}{g(\boldsymbol{\chi})} (-f(\boldsymbol{\chi}) + \mathbf{K}^T \mathbf{E} + \ddot{\chi}_d), \quad (3.2)$$

donde, $\ddot{\chi}_d$ es la segunda derivada de la trayectoria deseada, $\mathbf{E} = [e \int e \dot{e}]^T$, con $e = \chi_d - \chi$ sustituyendo (3.2) en (3.1), se tiene que la dinámica en lazo cerrado del sistema es

$$\ddot{e} + k_p e + k_i \int e + k_d \dot{e} = 0,$$

Control Traslacional y de Altura

La dinámica de la altitud mostrada en (1.3), puede representarse como (3.1),

$$\ddot{z} = \frac{1}{m} (\cos \phi \cos \theta) u_1 - g - \frac{k_3}{m} \dot{z}, \quad (3.3)$$

de esta manera si se diseña la señal de control como en (3.2) se obtiene

$$u_1 = \frac{m}{\cos \theta \cos \phi} \left[\frac{k_3}{m} \dot{z} + K_z^T E_z + \ddot{z}_d \right] + g, \quad (3.4)$$

la ecuación (3.4) es válida para $-\pi/2 < \phi < \pi/2$ y $-\pi/2 < \theta < \pi/2$.

Para la dinámica traslación en el eje X , se elige a una señal de control virtual como

$\mu_x = \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi$, obteniendo la dinámica

$$m\ddot{x} = f(\mathbf{x}) + \mu_x u_1, \quad (3.5)$$

y la señal de control se elige como en (3.2)

$$\mu_x = \frac{m}{u_1} \left[\frac{k_1}{m} \dot{x} + \mathbf{K}_x^T E_x + \ddot{x}_d \right]. \quad (3.6)$$

De manera similar para la dinámica en el eje Y , se elige a la señal de control virtual

como $\mu_y = \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi$, obteniendo la dinámica

$$m\ddot{y} = \frac{k_2}{m} \dot{y} + \mu_y u_1, \quad (3.7)$$

y la señal de control se elige como en (3.2)

$$\mu_y = \frac{m}{u_1} \left[\frac{k_2}{m} \dot{y} + K_y^T E_y + \ddot{y}_d \right]. \quad (3.8)$$

Las ecuaciones (3.6) y (3.8) son válidas para $u_1 > 0$, lo cual se cumple, ya que u_1 representa la fuerza de sustentación generada por los rotores del vehículo.

A partir de las señales de control virtual μ_x y μ_y se obtienen los ángulos deseados para el control de orientación en *roll*, ϕ_d y en *pitch*, θ_d :

$$\begin{bmatrix} \cos \phi \sin \theta \\ \sin \phi \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$$

$$\begin{aligned} \phi_d &= a \sin(\sin \psi \mu_x - \cos \psi \mu_y) \\ \theta_d &= a \sin\left(\frac{\cos \psi \mu_x + \sin \psi \mu_y}{\cos \phi}\right). \end{aligned} \quad (3.9)$$

Control de Orientación

Para las dinámicas de orientación, se tiene que pueden ser representadas como en (3.1) y diseñarse la señal de control como en (3.2).

Para el movimiento de *roll*, se tiene la señal de control u_2 como:

$$u_2 = I_x \left(\frac{k_4}{I_x} \dot{\phi}^2 - \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} + \mathbf{K}_\phi^T E_\phi + \ddot{\phi}_d \right), \quad (3.10)$$

Para el movimiento de *pitch*, se tiene la señal de control u_3 como:

$$u_3 = I_y \left(\frac{k_5}{I_y} \dot{\theta}^2 - \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} + \mathbf{K}_\theta^T E_\theta + \ddot{\theta}_d \right), \quad (3.11)$$

Finalmente, para el movimiento de *yaw*, se tiene la señal de control u_4 como:

$$u_4 = I_z \left(\frac{k_6}{I_z} \dot{\psi}^2 - \frac{(I_x - I_y)}{I_z} \dot{\phi} \dot{\theta} + \mathbf{K}_\psi^T E_\psi + \ddot{\psi}_d \right). \quad (3.12)$$

3.3 Diseño de Control ST - SMC

Para implementar el algoritmo ST-SMC, se propuso la siguiente estructura: primero, se tiene la trayectoria deseada en cada uno de los ejes traslacionales x , y , z y el movimiento de orientación de yaw . La trayectoria deseada se introduce a los controladores de orientación. El controlador Traslacional calcula la señal de control u_1 , al mismo tiempo que los ángulos deseados, los cuales junto a ψ_d son las referencias para el control de Orientación. Las señales de control u_2 , u_3 y u_4 son calculadas en el control de Orientación. Adicional a las señales de control, las perturbaciones afectan el comportamiento del QUAV, como se puede ver en la Fig. 3.1. Finalmente, las dinámicas de QUAV son nuevamente retroalimentadas a los Controladores hasta que la trayectoria deseada se completa.

Se diseñó el controlador ST-SMC, asumiendo que cada dinámica en (1.3), puede representarse como

$$\ddot{\chi} = f(\boldsymbol{\chi}) + g(\boldsymbol{\chi})u + w(t) \quad (3.13)$$

donde $f(\boldsymbol{\chi})$ y $g(\boldsymbol{\chi}) \neq 0$, son funciones no lineales de parámetros físicos no modelados y fuerzas aerodinámicas del QUAV, las cuales pueden ser total o parcialmente desconocidas, $\boldsymbol{\chi} = [\chi, \dot{\chi}]$, u es la señal de control y $w(t)$ representa las perturbaciones externas y parámetros desconocidos.

La superficie de deslizamiento se define por

$$s = \dot{e} + \beta e, \quad (3.14)$$

e y \dot{e} , son el error de seguimiento ($e = \chi_d - \chi$) y su derivada, β , es el parámetro de velocidad de convergencia, eligiendo a $\beta > 0$, se puede garantizar que el error tiende a cero cuando el tiempo tiende a infinito [35]. Derivando (3.14) se obtiene

$$\dot{s} = \ddot{e} + \beta \dot{e} = \ddot{\chi}_d - \ddot{\chi} + \beta \dot{e}, \quad (3.15)$$

sustituyendo (3.13) en (3.15) se obtiene

$$\dot{s} = \ddot{\chi}_d - [f(\boldsymbol{\chi}) + g(\boldsymbol{\chi})u + w(t)] + \beta \dot{e}. \quad (3.16)$$

La ley de control se compone por dos términos, los cuales son: una señal de control equivalente (u_{eq}) y la señal de control de deslizamiento (u_{st}) [36]

$$u = u_{eq} + u_{st} \quad (3.17)$$

donde

$$u_{eq} = \ddot{\chi}_d - [f(\boldsymbol{\chi}) + g(\boldsymbol{\chi})u + w(t)] + \beta \dot{e},$$

$$u_{st} = k_1 \sqrt{|s|} \text{sign}(s) + k_2 \int \text{sign}(s) dt.$$

Control Traslacional y de Altitud

La dinámica de altitud mostrada en(1.3), puede representarse en la forma de (3.13),

$$\ddot{z} = \frac{1}{m}(\cos \phi \cos \theta)u_1 - \frac{k_3}{m}\dot{z} - g, \quad (3.18)$$

se puede entonces diseñar una ley de control de la misma forma que en (3.17), obteniendo

$$u_1 = \frac{m}{\cos \theta \cos \phi} \left(\ddot{z}_d + \beta_z \dot{e}_z + k_1 \sqrt{|s_z|} \text{sign}(s_z) + k_2 \int \text{sign}(s_z) dt + w(t) \right) + g, \quad (3.19)$$

la ecuación (3.19) es válida para $-\pi/2 < \phi < \pi/2$ y $-\pi/2 < \theta < \pi/2$.

Para la dinámica de traslación de eje X , es necesario elegir una señal de control virtual $\mu_x = \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi$, que resulta en

$$m\ddot{x} = f(\mathbf{x}) + \mu_x u_1, \quad (3.20)$$

dando como resultado la ley de control (3.17)

$$\mu_x = \frac{m}{u_1} \left(\ddot{x}_d + \beta_x \dot{e}_x + k_1 \sqrt{|s_x|} \text{sign}(s_x) + k_2 \int \text{sign}(s_x) dt + w(t) \right). \quad (3.21)$$

De la misma manera para la dinámica de Y , la señal de control virtual es $\mu_y = \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi$, resultando en

$$m\ddot{y} = \frac{k_2}{m}\dot{y} + \mu_y u_1, \quad (3.22)$$

con la señal de control de la misma forma que en (3.17)

$$\mu_y = \frac{m}{u_1} \left(\ddot{y}_d + \beta_y \dot{e}_y + k_1 \sqrt{|s_y|} \text{sign}(s_y) + k_2 \int \text{sign}(s_y) dt + w(t) \right), \quad (3.23)$$

las expresiones (3.21) y (3.23) son válidas si $u_1 > 0$, lo cual se cumple, ya que u_1 es la fuerza de sustentamiento producida por los rotores.

Con las señales de control virtual μ_x y μ_y pueden calcularse los ángulos deseados *roll*, y *pitch* de la siguiente manera:

$$\begin{bmatrix} \cos \phi \sin \theta \\ \sin \phi \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$$

$$\begin{aligned} \phi_d &= a \sin(\sin \psi \mu_x - \cos \psi \mu_y) \\ \theta_d &= a \sin \left(\frac{\cos \psi \mu_x + \sin \psi \mu_y}{\cos \phi} \right), \end{aligned} \quad (3.24)$$

los cuales serán las referencias para el Control de Orientación.

Control de Orientación

Las dinámicas de orientación también pueden ser representadas como (3.13), entonces se diseña una señal de control con la estructura mostrada en la ecuación (3.17).

Para el movimiento de *roll*, corresponde la señal de control u_2 como

$$u_2 = I_x \left(\ddot{\phi}_d + \beta_\phi \dot{e}_\phi + k_1 \sqrt{|s_\phi|} \text{sign}(s_\phi) + k_2 \int \text{sign}(s_\phi) dt - \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} \right), \quad (3.25)$$

de la misma manera para la dinámica de *pitch*, la señal de control u_3 se calcula como

$$u_3 = I_y \left(\ddot{\theta}_d + \beta_\theta \dot{e}_\theta + k_1 \sqrt{|s_\theta|} \text{sign}(s_\theta) + k_2 \int \text{sign}(s_\theta) dt - \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} \right), \quad (3.26)$$

Finalmente, para el movimiento de *yaw*, u_4 se calcula de la siguiente manera:

$$u_4 = I_z \left(\ddot{\psi}_d + \beta_\psi \dot{e}_\psi + k_1 \sqrt{|s_\psi|} \text{sign}(s_\psi) + k_2 \int \text{sign}(s_\psi) dt - \frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\psi} \right). \quad (3.27)$$

3.4 Resultados de la comparación de controles PID vs ST-SMC

Para llevar a cabo la comparación de los controladores PID y ST-SMC, y encontrar el que presenta el mejor desempeño al aplicarse a las dinámicas del QUAV, se lleva a cabo la identificación de parámetros y optimización. Con este propósito, se implementa la estructura mostrada en la Figura 3.2. Puede observarse que inicialmente el algoritmo HGS entrega una serie de parámetros que el controlador implementa y con los cuales calcula las señales de control que se le envían al QUAV; del QUAV se obtiene el desempeño de cada una de las dinámicas $(x, y, z, \phi, \theta, \psi)$ con lo que se calcula el error RMSE, el cual se convierte en la función costo para el algoritmo HGS. Todo el proceso se repite un número de iteraciones establecido.

En la Figura 3.3, se puede observar el comportamiento de los controles PID y ST-SMC durante la optimización y se muestran los valores mínimos alcanzados por el algoritmo HGS en cada iteración. En 3.3a, puede verse que el PID converge más rápido y en 3.3b, se observa una convergencia mas lenta pero que alcanza un valor más pequeño al minimizar el error RMSE.

En la Tabla 3.1, se reporta el desempeño de cada uno de los controles PID y ST-SMC. Los valores estadísticos: mejor valor, media, desviación estándar y el peor valor alcanzados, cuando se aplica el algoritmo HGS. Puede observarse que el valor menor es alcanzado por el algoritmo ST-SMC.

Tabla 3.1: Valores de RMSE alcanzados con 50 individuos, 20 evoluciones, y 30 repeticiones.

Parámetro	PID	ST-SMC
Mejor valor	0.036290839363412	0.017650649172323
Media	0.627945953121468	0.018197329174221
Desviación estándar	0.623996714866044	0.000564758140820
Peor valor	1.620029205574020	0.020239081645909

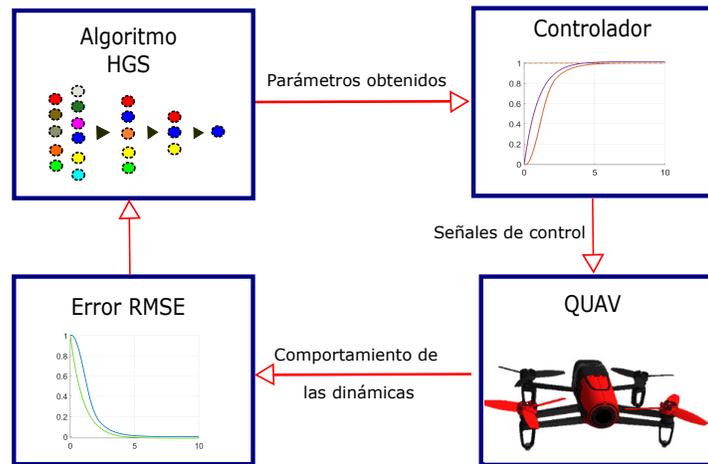
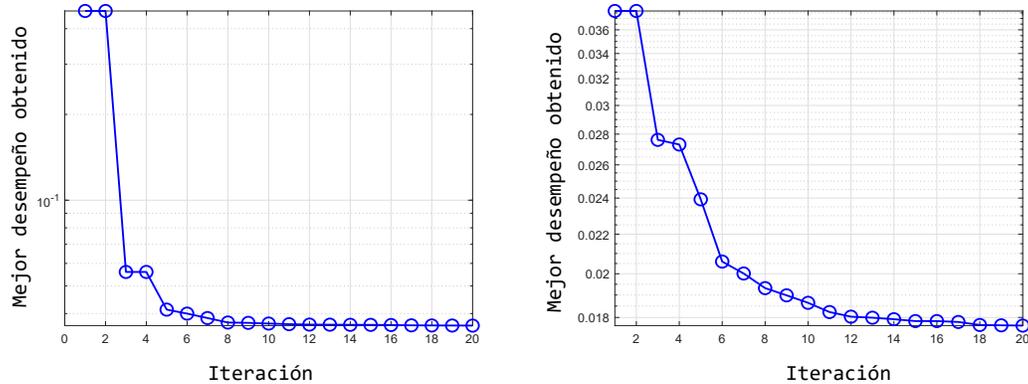


Figura 3.2: Método para la identificación de parámetros de los controles PID y ST-SMC.

3.4.1 Comparación en simulación

Primero se lleva a cabo la comparación en simulaciones numéricas de los controladores PID y ST-SMC. Las especificaciones del equipo de cómputo y software usados, son las mostradas en 2.2. Los parámetros físicos en (1.3), usados para la implementación se encuentran en la Tabla 3.3, obtenidos de [37]. Las ganancias para cada uno de los controladores son los mostrados en 3.2.

Las trayectorias deseadas son funciones trapezoidales para las dinámicas de traslacionales (x y y) y una rampa para la dinámica de altitud (z) y de orientación (ψ).



(a) Valores mínimos del error RMSE mientras se optimiza el control PID. (b) Valores mínimos del error RMSE mientras se optimiza el control ST-SMC

Figura 3.3: Curvas de convergencia obtenidas por el HGS, con 20 iteraciones, 50 individuos y 30 repeticiones.

En la figura 3.4 se puede observar el desempeño del QUAV cuando se implementan ambos algoritmos de control. La Figura 3.4a muestra el resultado de la simulación del controlador PID. Puede verse que la trayectoria se realiza con un sobre impulso en los ejes X y Y cuando hay cambios en la dirección.

Por otra parte, la Figura 3.4b muestra una trayectoria con un sobre impulso mínimo cuando el controlador ST-SMC es implementado. Incluso cuando hay un cambio de dirección, las dinámicas x y y presentan un error menor comparado con el experimento del controlador PID.

Comparación del seguimiento de trayectoria agregando una perturbación externa.

Para validar el desempeño de los controladores propuestos PID y ST-SMC, se agrega una perturbación $\omega = \frac{1}{2} \cos t$, de la misma manera que la reportada en el

Tabla 3.2: Parámetros obtenidos mediante el algoritmo HGS.

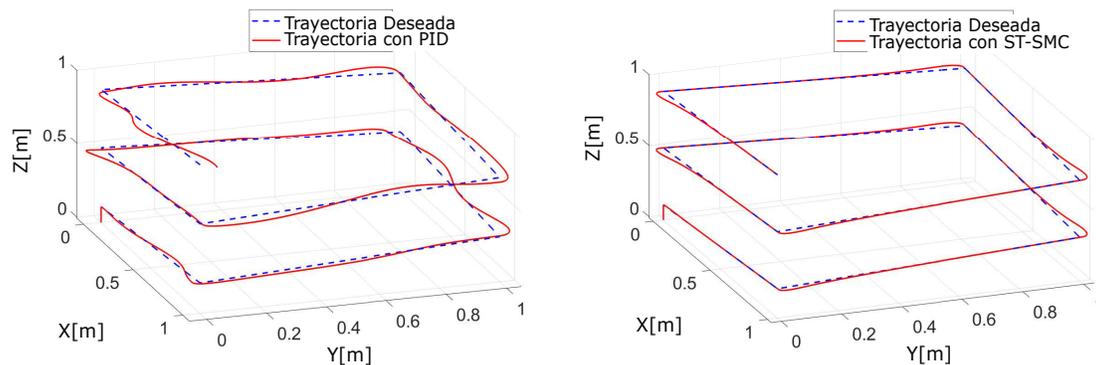
Dinámica	Algoritmo			
	PID		ST-SMC	
	Parámetro	Valor	Parámetro	Valor
x	k_p	9.999	β	2.0
	k_i	4.319	k_1	2.0
	k_d	9.144	k_2	1.997
y	k_p	10.0	β	2.0
	k_i	4.215	k_1	2.0
	k_d	10.0	k_2	1.999
z	k_p	9.99	β	1.401
	k_i	10.0	k_1	1.223
	k_d	0	k_2	1.898
ϕ	k_p	9.882	β	0.066
	k_i	4.386	k_1	0.0469.
	k_d	0	k_2	1.9846
θ	k_p	4.618	β	0.075
	k_i	0	k_1	0.039
	k_d	0	k_2	2.0
ψ	k_p	10.0	β	0.089
	k_i	0.701	k_1	0.005
	k_d	1.029	k_2	0.005

trabajo [38]. Se evalúa la respuesta de las dinámicas del QUAUV, al implementar cada uno de los controladores optimizados mediante el algoritmo HGS.

La figura 3.5 muestra el desempeño tridimensional del QUAUV al seguir la trayectoria deseada, mientras que la perturbación es aplicada. En la Figura 3.5a, se puede observar la respuesta dinámica del QUAUV cuando se implementa el controlador PID, puede verse que la trayectoria presenta oscilaciones de magnitud considerable en los movimientos. Por el contrario, cuando se implementa el controlador ST-SMC la trayectoria mantiene un error mínimo, lo que muestra que el controlador es capaz de contrarrestar a la perturbación (ver Figura 3.5b).

Tabla 3.3: Parámetros implementados en simulación.

Parámetro	Valor	Unidades
Masa	0.4	kg
Momento de inercia	$I_x = 0.000906 I_y = 0.001242, I_z = 0.002054$	kgm ²
Coefficientes de arrastre traslacionales	$k_{1,2,3} = 5.56 \times 10^{-04}$	--
Coefficiente de fricción aerodinámica	$k_{4,5,6} = 0.3729$	--
Aceleración de la gravedad	9.81	m/s ²



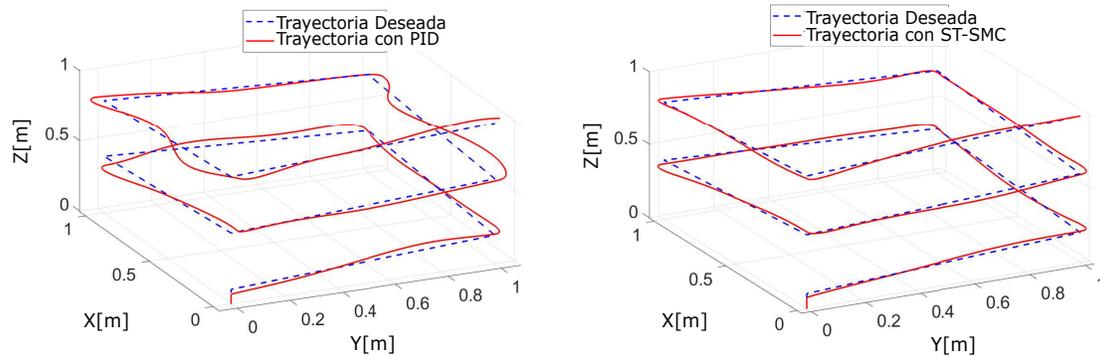
(a) Desempeño del QUAV en los ejes X-Y-Z con el control PID. (b) Desempeño del QUAV en los ejes X-Y-Z con el control ST-SMC.

Figura 3.4: Simulaciones de seguimiento de trayectoria aplicando los controles PID y ST-SMC sin perturbaciones.

3.4.2 Comparación de los controladores en plataforma física

Adicional a la comparación desarrollada en simulaciones, se lleva a cabo la comparación de los controladores PID y ST-SMC optimizados con el algoritmo HGS en una plataforma experimental. La plataforma está compuesta por un sistema de posicionamiento conformado por 8 cámaras Vicon® Bonita, comunicadas mediante el sistema operativo ROS que permite conocer la posición y orientación de un vehículo QUAV Parrot® bebop 1 (ver Figura 3.6a).

ROS es un sistema que permite la comunicación mediante nodos, a través de



(a) Desempeño del QUAV en los ejes X-Y-Z con el control PID. (b) Desempeño del QUAV en los ejes X-Y-Z con el control ST-SMC.

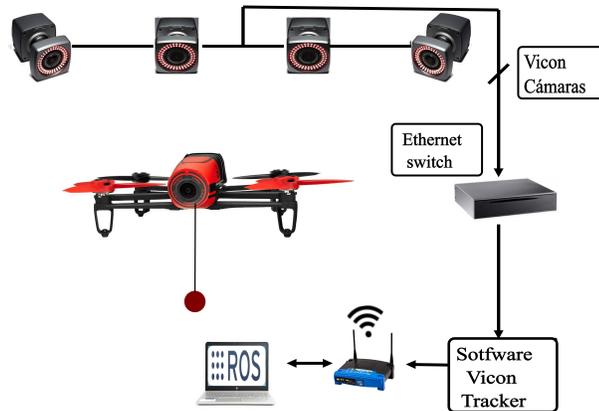
Figura 3.5: Comparación de los controles PID y ST-SMC para seguimiento de trayectoria aplicando una perturbación.

tópicos es posible enviar información con publicadores, y recibir o leer mediante suscriptores. Para esta plataforma se usan cuatro nodos: 1) Vicon, es un publicador que permite acceder a la información de posición del QUAV en cada instante; 2) Bebop, es un suscriptor, el cual recibe comandos como despegue y aterrizaje, así como las señales de control que van calculando los controladores; 3) Joystick, permite activar el nodo de control a través de los botones de un joystick; y 4) Nodo de Control, este nodo enlaza todos los nodos anteriores y administra las tareas del sistema, también controla el paso de modo manual a automático (ver Figura 3.6b).

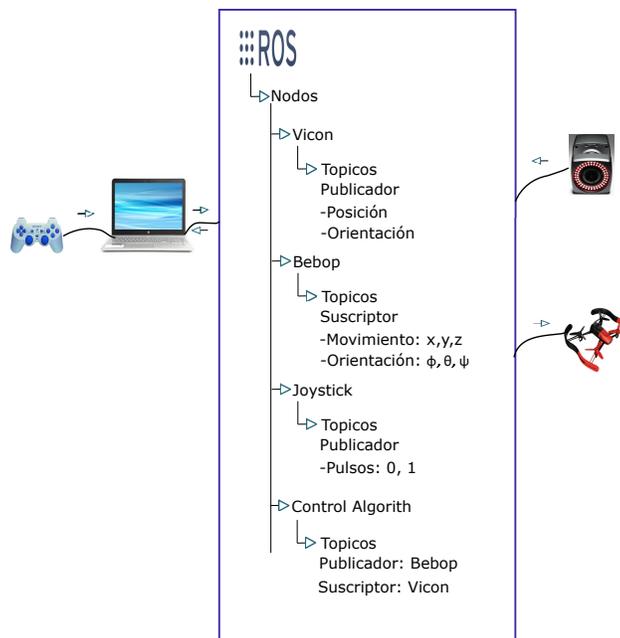
Las trayectorias deseadas son las mismas que las implementadas en simulación. Los parámetros físicos y ganancias de los controladores se muestran en las Tablas 3.3 y 3.2, donde todos los parámetros son multiplicados por $1/2300$ debido a que el QUAV bebop opera con señales de control alrededor de $10e - 3$. Esta relación fue encontrada mediante la experimentación.

El desempeño obtenido por los controles optimizados con el algoritmo HGS al implementar en plataforma física es muy similar que el obtenido en simulaciones por cada controlador, como puede observarse en ambos experimentos: PID, Figura 3.7, y ST-SMC, Figura 3.8. El QUAV lleva a cabo de manera satisfactoria el seguimiento de trayectoria con los controles PID y ST-SMC cuando no existen perturbaciones externas. Las dinámicas traslacionales (ejes X y Y) presentan un sobre impulso y oscilación como resultado de los cambios de dirección en la trayectoria deseada (Figuras 3.7a, 3.7b, 3.8a, 3.8b). La dinámica de altitud presenta un error máximo al inicio de la trayectoria. Este error, debido al efecto de rebote de aire que el QUAV experimenta cuando está cercano al piso (Figuras 3.7c y 3.8c).

En la Figura 3.9, se muestran los resultados de los experimentos agregando una perturbación externa $\frac{1}{2} \sin(t)$. La Figura 3.9a muestra el desempeño de las dinámicas del QUAV en los ejes $X - Y - Z$ cuando el controlador PID es implementado. Puede observarse que la trayectoria real presenta oscilaciones con un error de considerable magnitud. Por el contrario, cuando se implementa el controlador ST-SMC, el QUAV mantiene un seguimiento de trayectoria con un mínimo error con respecto a la trayectoria deseada. Esta respuesta demuestra que el controlador ST-SMC es robusto en presencia de perturbaciones (ver Figura 3.9b). Los datos de los experimentos realizados están disponibles en este repositorio: <https://github.com/nasamzp/Optimization-of-a-ST-SMC-control-for-path-tracking-3D-with-perturbations-of-a-QUAV-using-the-Huger-G>.



(a) Diagrama de la plataforma física usada para realizar los experimentos y comparación de los controladores optimizados.



(b) Estructura de comunicación implementada en la plataforma experimental usando ROS.

Figura 3.6: Estructura de la plataforma experimental.

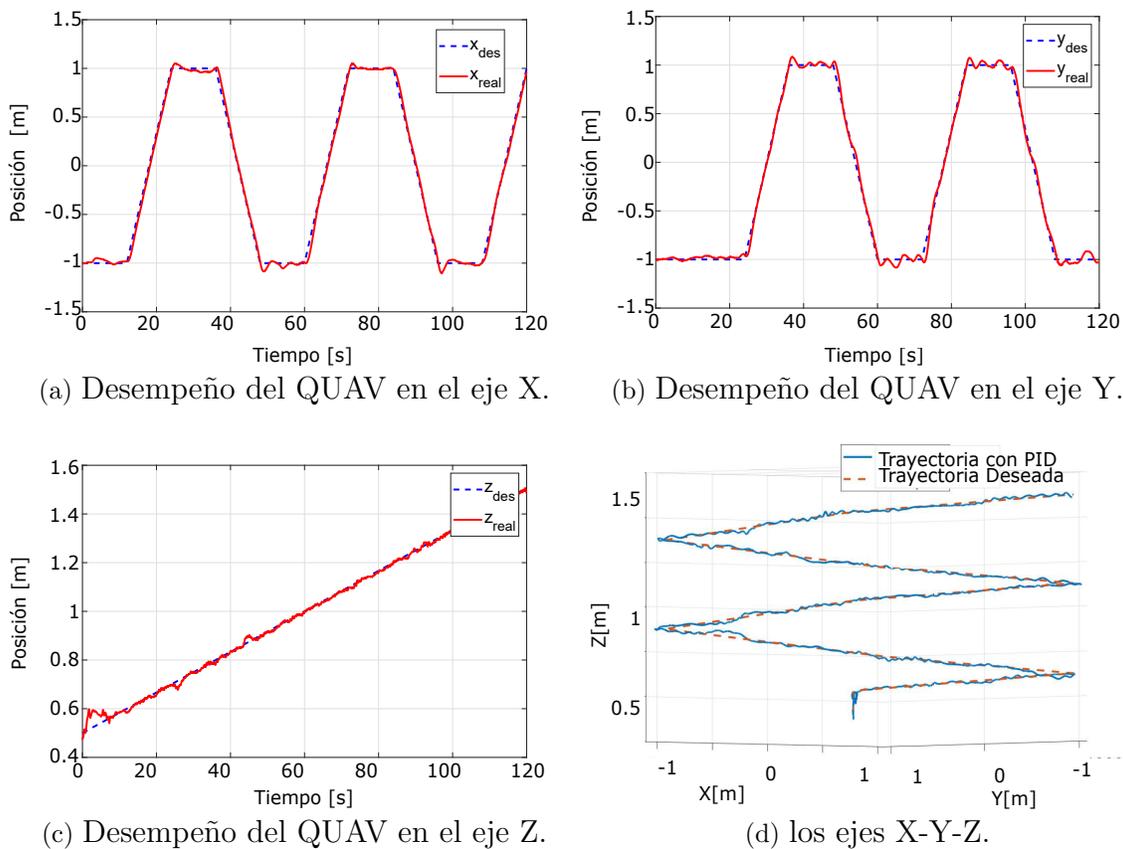
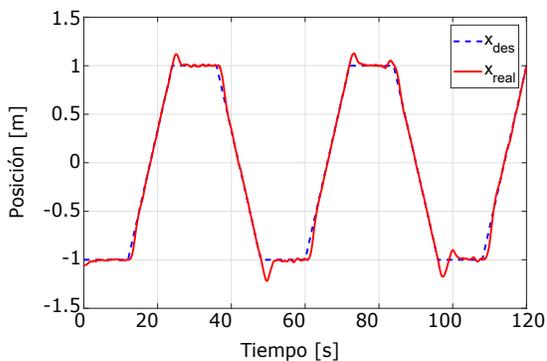
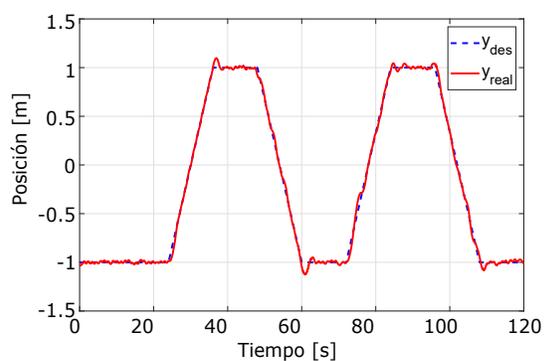


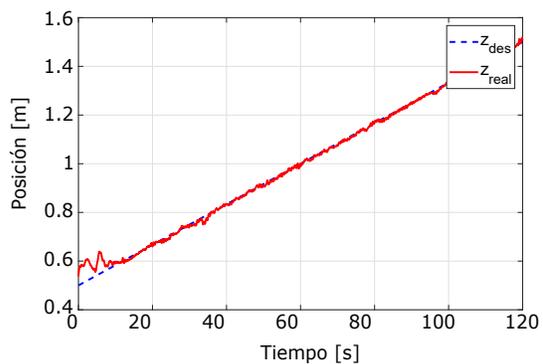
Figura 3.7: Resultados de pruebas experimentales. Desempeño del QUAV con el controlador PID optimizado mediante HGS sin perturbaciones.



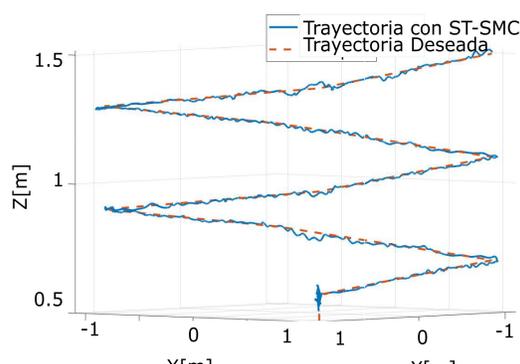
(a) Desempeño del QAV en el eje X.



(b) Desempeño del QAV en el eje Y.

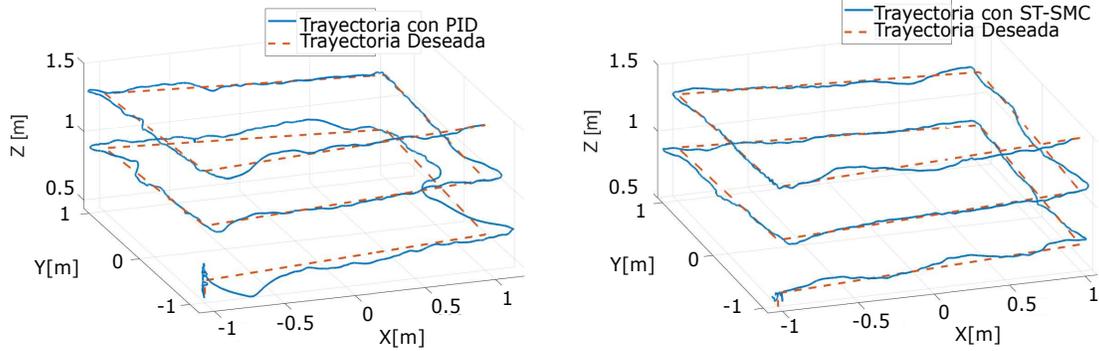


(c) Desempeño del QAV en el eje Z.



(d) Desempeño del QAV en los ejes X-Y-Z.

Figura 3.8: Resultados de pruebas experimentales. Desempeño del QAV con el controlador ST-SMC optimizado mediante HGS sin perturbaciones.



(a) Desempeño del QUAV en seguimiento de trayectoria con el controlador PID. (b) Desempeño del QUAV en seguimiento de trayectoria con el controlador ST-SMC.

Figura 3.9: Resultados de pruebas experimentales. Comparación del desempeño del QUAV con los controles PID y ST-SMC optimizados mediante HGS agregando una perturbación.

Capítulo 4

Transporte de carga suspendida mediante un QUAV con control ST-SMC optimizado implementando el algoritmo HGS

En este capítulo se lleva a cabo la implementación del algoritmo metaheurístico HGS para identificación de parámetros y optimización del controlador no lineal ST-SMC para seguimiento de trayectoria en espacio tridimensional de un QUAV con carga suspendida vía cable.

4.1 Implementación del Modelo dinámico

Para llevar a cabo la implementación del algoritmo metaheurístico HGS para identificación de parámetros y optimización del controlador no lineal ST-SMC para seguimiento de trayectoria en espacio tridimensional de un QUAUV con carga suspendida vía cable, es necesario hacer uso del modelo matemático mostrado en la ecuación 1.11, desarrollado en la sección 1.2:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) = \mathbf{b}(q)\mathbf{u},$$

esta ecuación describe las 8 dinámicas presentes en el sistema QUAUV con carga suspendida vía cable: las seis dinámicas del QUAUV $(x, y, z, \phi, \theta, \psi)$, y las dos dinámicas agregadas por la carga suspendida que se comporta como un péndulo (ángulos α y β , ver Figura 1.3).

Tabla 4.1: Parámetros físicos implementados.

Parámetro	Valor	Unidades
Masa	0.4	<i>kg</i>
Momento de inercia	$I_x = 0.000906 I_y = 0.001242, I_z = 0.002054$	<i>kgm</i> ²
Momento de inercia de carga	$I_c = 0.1$	--
Aceleración de la gravedad	9.81	<i>m/s</i> ²

4.2 Implementación del controlador ST-SMC

La estructura del controlador es la mostrada en la sección 3.1 y el diseño del controlador no lineal ST-SMC es la mostrada en la sección 3.3. Se emplea esta misma estructura, debido a que las dinámicas de la carga suspendida, se consideran como parte de la perturbaciones externas que afectan al QUAUV, como puede observarse en la figura 4.1. Se ha elegido el controlador No lineal ST-SMC, ya que en el capítulo 3 se encontró que es el algoritmo capaz de contrarrestar las perturbaciones externas.

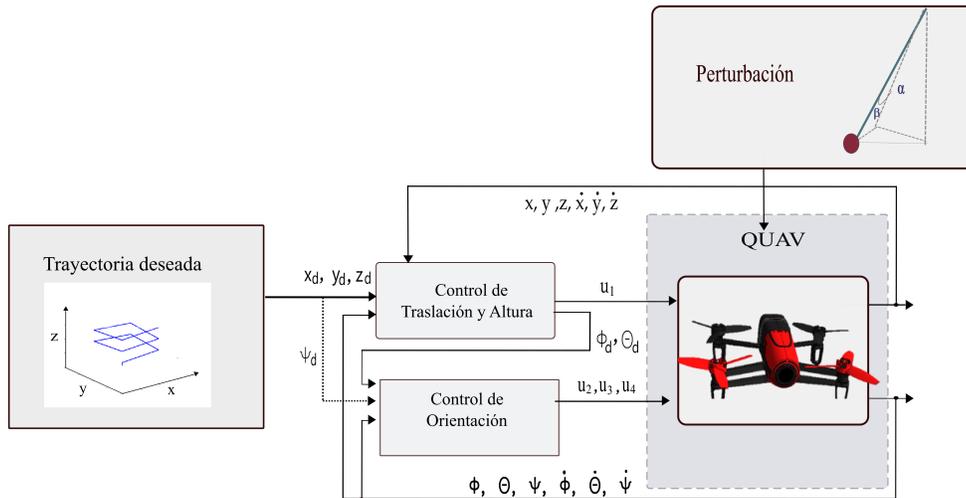


Figura 4.1: Estructura de control propuesta para la implementación del control ST-SMC al QUAUV con carga.

Las ecuaciones de las señales de control toman la siguiente estructura

$$u = \frac{1}{b} [M(q) (\dot{s} + \ddot{q}_d \beta \dot{e}) + C(q) \dot{q} + w(t) + g]. \quad (4.1)$$

Entonces, para las dinámicas de altura y traslación tenemos:

$$u_1 = \frac{1}{b_3} \left[M_{33} + M_{37} + M_{38} \left(\ddot{z}_d + \beta_z \dot{e}_z + k_{1z} \sqrt{|s_z|} \text{sign}(s_z) + k_{2z} \int \text{sign}(s_z) dt \right) + (C_{37} + C_{38}) \dot{z} + (M_{33})g \right], \quad (4.2)$$

$$\mu_x = \frac{1}{u_1} \left[M_{11} + M_{17} \left(\ddot{x}_d + \beta_x \dot{e}_x + k_{1x} \sqrt{|s_x|} \text{sign}(s_x) + k_{2x} \int \text{sign}(s_x) dt \right) + (C_{17}) \dot{x} \right], \quad (4.3)$$

$$\mu_y = \frac{1}{u_1} \left[M_{22} + M_{28} \left(\ddot{y}_d + \beta_y \dot{e}_y + k_{1y} \sqrt{|s_y|} \text{sign}(s_y) + k_{2y} \int \text{sign}(s_y) dt \right) + (C_{28}) \dot{y} \right], \quad (4.4)$$

y las señales de control para las dinámicas de orientación son:

$$u_2 = \frac{1}{b_4} \left[M_{44} + M_{46} \left(\ddot{\phi}_d + \beta_\phi \dot{e}_\phi + k_{1\phi} \sqrt{|s_\phi|} \text{sign}(s_\phi) + k_{2\phi} \int \text{sign}(s_\phi) dt \right) + (C_{45} + C_{46}) \dot{\phi} \right], \quad (4.5)$$

$$u_3 = \frac{1}{b_5} \left[M_{54} + M_{55} + M_{56} \left(\ddot{\theta}_d + \beta_\theta \dot{e}_\theta + k_{1_\theta} \sqrt{|s_\theta|} \text{sign}(s_\theta) + k_{2_\theta} \int \text{sign}(s_\theta) dt \right) + (C_{54} + C_{55} + C_{56}) \dot{\theta} \right], \quad (4.6)$$

$$u_4 = \frac{1}{b_6} \left[M_{64} + M_{65} + M_{66} \left(\ddot{\psi}_d + \beta_\psi \dot{e}_\psi + k_{1_\psi} \sqrt{|s_\psi|} \text{sign}(s_\psi) + k_{2_\psi} \int \text{sign}(s_\psi) dt \right) + (C_{64} + C_{65} + C_{66}) \dot{\psi} \right], \quad (4.7)$$

donde los valores de b , M y C están definidos en las ecuaciones 1.15, 1.13 y 1.12.

4.3 Identificación de parámetros y optimización del controlador ST-SMC

El procedimiento para la identificación de parámetros y optimización del controlador ST-SMC es la mostrada en el capítulo 2, donde se encontró que el algoritmo HGS alcanza un valor menor al minimizar el error RMSE. Inicialmente el algoritmo HGS es evaluado y entrega un conjunto de ganancias, las cuales se aplican al controlador ST-SMC, quien a su vez, calcula las señales de control para el QUAUV, posteriormente se mide el desempeño dinámico de cada uno de los ejes X, Y y Z y ángulos de orientación, para finalmente calcular el RMSE entre las dinámicas deseadas y las obtenidas. Este ciclo se repite hasta que el número de iteraciones determinado inicial-

mente es alcanzado o hasta que el error o función costo alcanza un valor establecido, como se muestra en la Figura 4.2.

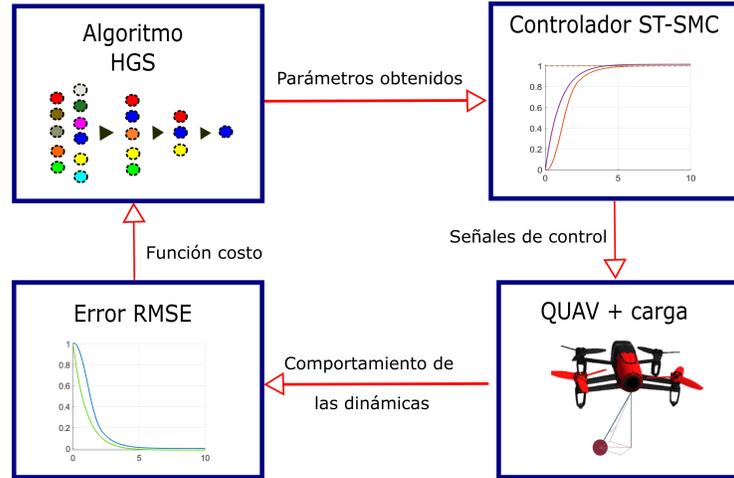


Figura 4.2: Esquema de sintonización de ganancias del controlador ST-SMC implementando el algoritmo HGS.

Para obtener las ganancias β , k_1 y k_2 de cada una de las 6 dinámicas del QUAV, presentes en las ecuaciones 4.2-4.7, la población a optimizar es:

$$P = \{\beta_x, k_{1_x}, k_{2_x}, \beta_y, k_{1_y}, k_{2_y}, \beta_z, k_{1_z}, k_{2_z}, \beta_\phi, k_{1_\phi}, k_{2_\phi}, \beta_\theta, k_{1_\theta}, k_{2_\theta}, \beta_\psi, k_{1_\psi}, k_{2_\psi}\},$$

siendo el espacio de búsqueda para cada uno de los individuos $S = [0, 10]$. La Figura 4.3 muestra la curva de convergencia del error RMSE al optimizar al controlador ST-SMC. La Tabla 4.2 muestra los 18 parámetros durante la optimización para cada una de las dinámicas del QUAV.

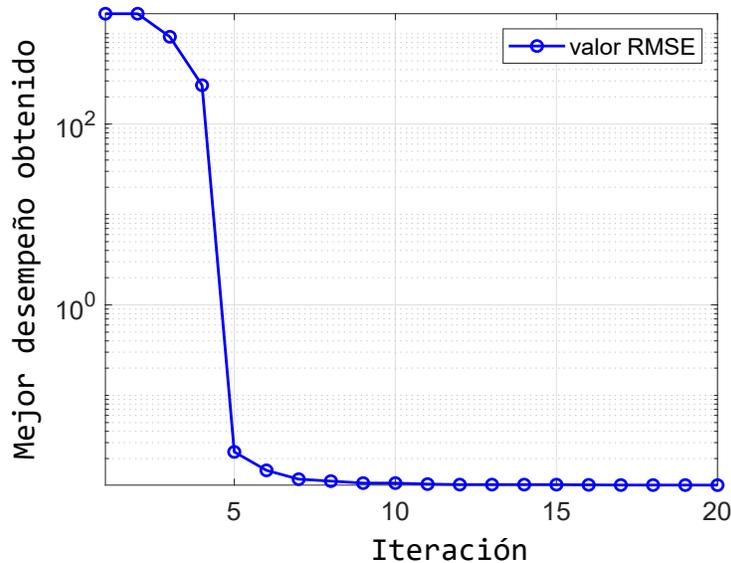


Figura 4.3: Curva de convergencia obtenidas por el HGS, con 20 iteraciones, 50 individuos y 1 repetición.

4.3.1 Resultados

A continuación se muestran los resultados obtenidos del transporte de carga suspendida mediante un QUAUV con control no lineal ST-SMC optimizado, aplicando el algoritmo metaheurístico HGS. El valor de error RMSE alcanzado mediante la optimización es de 0.0109.

En la figura 4.4 se observan el desempeño de las dinámicas traslacionales y de altitud del QUAUV. Las Figuras 4.4a y 4.5b muestran el desempeño del QUAUV en el eje X y Y, donde puede observarse que la trayectoria real sigue a la deseada con un mínimo error. La Figura 4.4c, muestra el desempeño de la dinámica z donde puede observarse una oscilación al inicio de la trayectoria, donde los valores negativos

Tabla 4.2: Parámetros obtenidos mediante el algoritmo HGS.

Dinámica	ST-SMC	
	Parámetro	Valor
x	β	10
	k_1	0.00101
	k_2	0.0012
y	β	10
	k_1	0.0011
	k_2	0.0013
z	β	9.99964
	k_1	0.00216
	k_2	9.84326
ϕ	β	8.27840
	k_1	4.41300
	k_2	0.11514
θ	β	8.96775
	k_1	0.26662
	k_2	0.13964
ψ	β	0.0001
	k_1	2.43695
	k_2	9.01263

mostrados responden a la reacción de la señal de control hasta el momento en que se vence la fuerza de gravedad y logra levantar al QUAUV.

La Figura 4.5, muestra el desempeño de las dinámicas rotacionales y de los ángulos de la carga suspendida. Las Figuras 4.5a y 4.5b muestran a los ángulos ϕ y θ , respectivamente. Puede observarse el efecto de los cambios en las referencias deseadas de x y y , este efecto debido al acoplamiento en las dinámicas traslacionales y de rotación. La dinámica rotacional de ψ se muestra en la Figura 4.5c, puede verse que la dinámica real sigue a la deseada en toda la ejecución de la trayectoria. Los ángulos formados entre la carga suspendida y el QUAUV, se muestran en la Figura 4.5d, se puede destacar que las amplitudes de las oscilaciones de la carga se mantienen menores a 2 grados reduciendo el efecto de la perturbación en el sistema. Por ultimo,

la figura 4.6, muestra la trayectoria seguida por el QUAUV para el transporte de carga suspendida. Puede observarse que el QUAUV sigue a trayectoria deseada en cada uno de los ejes $X - Y - Z$.

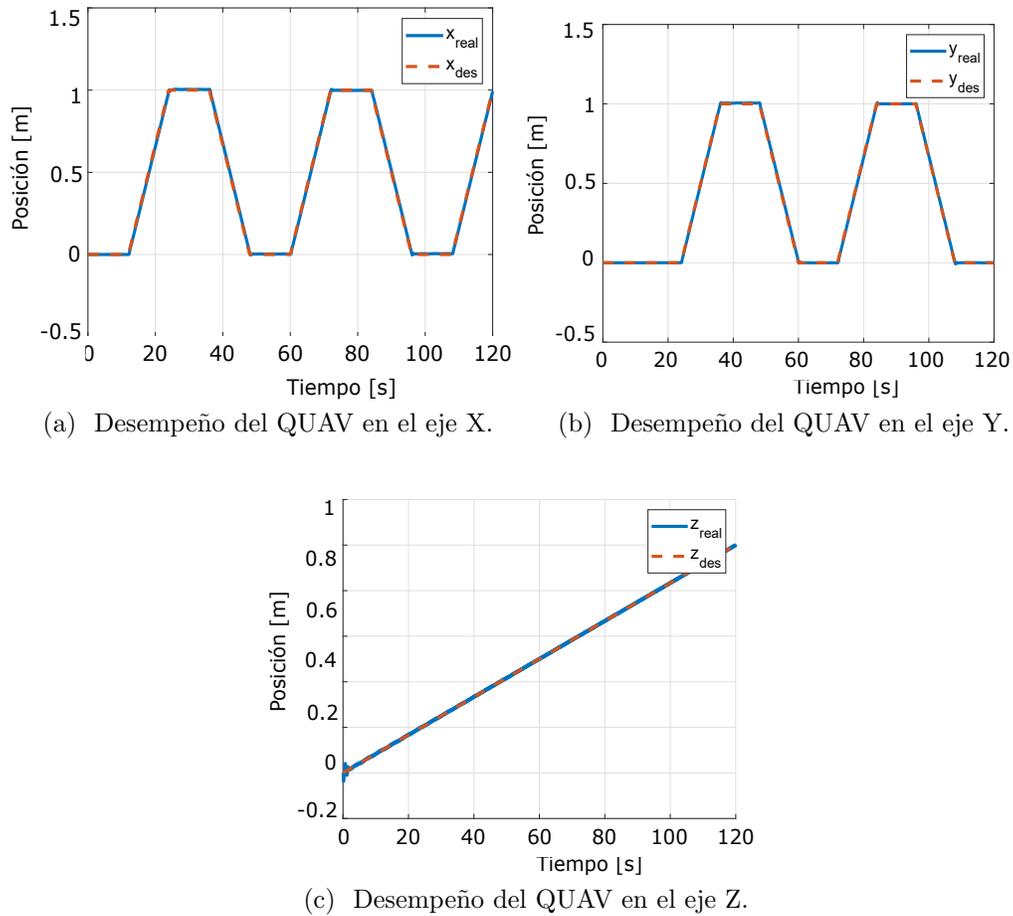
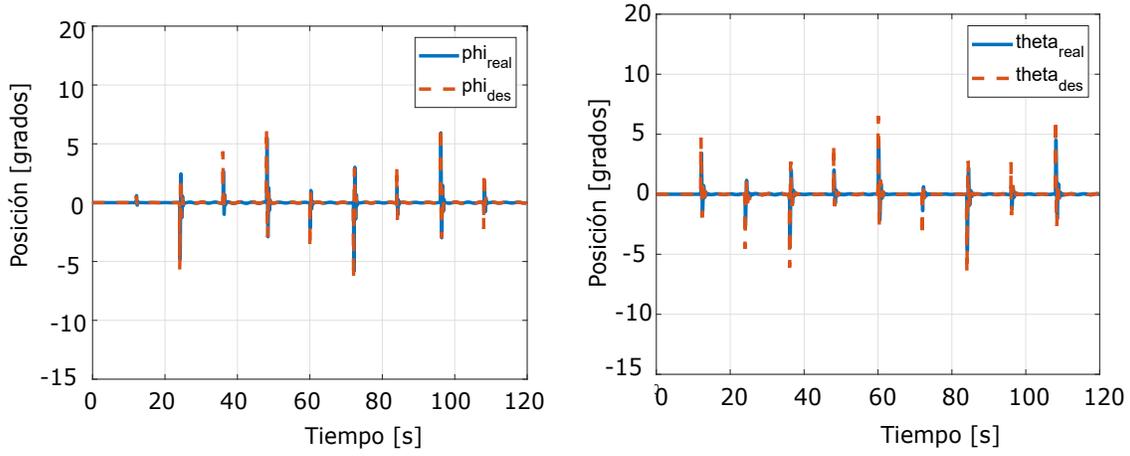
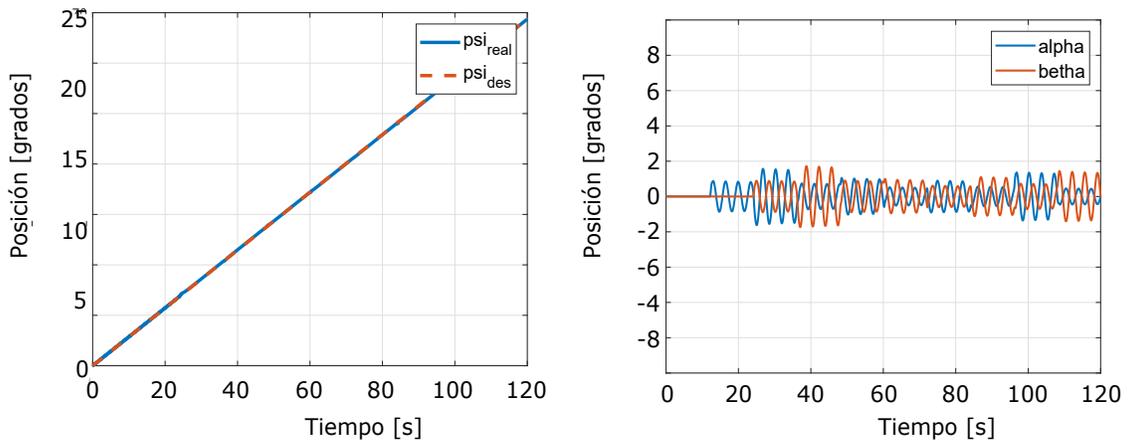


Figura 4.4: Desempeño de las dinámicas traslacionales del QUAUV con carga, controlado con ST-SMC optimizado mediante el algoritmo HGS.



(a) Desempeño del QUAUV en el eje X. (b) Desempeño del QUAUV en el eje Y.



(c) Desempeño del QUAUV en el eje Z. (d) Desempeño del QUAUV en el eje beta alfa.

Figura 4.5: Desempeño de las dinámicas rotaciones del QUAUV con carga, controlado con ST-SMC optimizado mediante el algoritmo HGS.

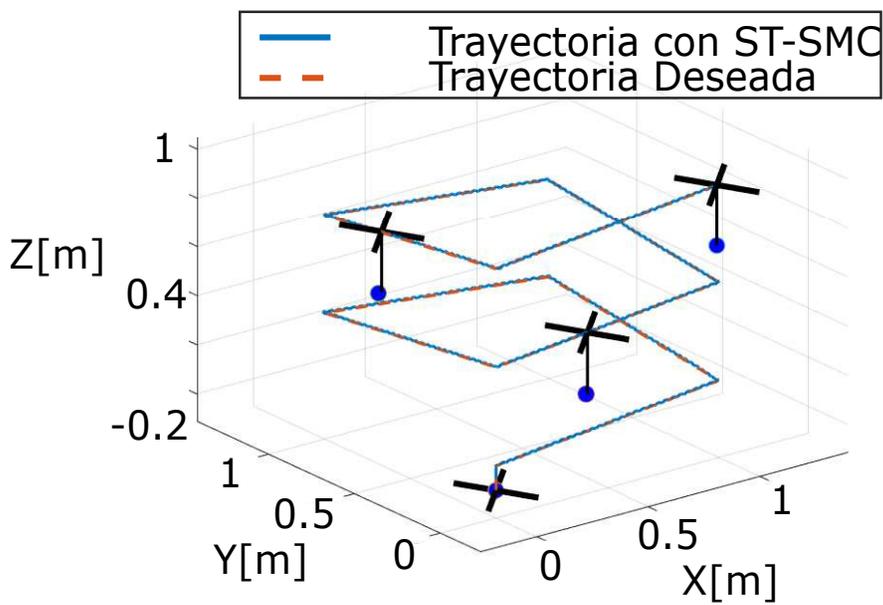


Figura 4.6: Desempeño del QUAUV con carga suspendida en seguimiento de trayectoria en el espacio $X - Y - Z$.

Capítulo 5

Hibridación NN-HGS

En este capítulo se describen a detalle las características de la Red Neuronal Kohonen, para posteriormente mostrar el proceso para su hibridación con el algoritmo metaheurístico HGS. La hibridación se realiza ya que, para la implementación de los diferentes algoritmos metaheurísticos, es necesario determinar el espacio o rango en que se realiza la búsqueda, sin embargo, para sistemas no lineales como el QUAV, no se tiene una certeza del comportamiento y por tanto, no es posible determinar si el espacio de búsqueda es el correcto. La red neuronal ajusta los valores máximo y mínimo del espacio de búsqueda, basada en el desempeño del QUAV.

Algoritmo Hunger Games Search

El algoritmo HGS, se describe a detalle en 2.1.3, en donde se modela el comportamiento de la mayoría de animales para conseguir alimento y sobrevivir.

Red neuronal Kohonen

Este modelo de Red neuronal desarrollado por el Dr. Teuvo Kohonen en 1982, es un modelo de aprendizaje autoorganizable y no supervisado, también conocido como SOM (Self Organizing Map). Existen diferentes aplicaciones para este modelo de Redes Neuronales como: clustering, visualización, clasificación, interpolación polinómica, recuperación de información, reconocimiento de patrones y resolución de problemas de optimización, entre otros [39].

Este algoritmo es un método para la reducción de dimensionalidad, ya que mapea entradas de alta dimensión a una representación discretizada de baja dimensión, usualmente bidimensional, y conserva la estructura de su espacio de entrada.

Todo el aprendizaje ocurre sin supervisión, gracias a que los nodos se autoorganizan. También se denominan mapas de características, ya que esencialmente están volviendo a entrenar las características de los datos de entrada y simplemente se agrupan según la similitud entre ellos. Estas características tienen un alto valor en la práctica, ya que permiten visualizar grandes cantidades de datos de alta dimensión y representar la relación entre ellos en dimensiones interpretables, para su análisis.

Las unidades básicas son las neuronas, y se organizan en dos capas: la capa de entrada y la capa o mapa de salida unidas a través de los pesos. Cada neurona de entrada está conectada a todas las neuronas de salida, y estas conexiones de fuerzas o pesos asociados a ellas inicialmente son aleatorios. Durante el entrenamiento, cada unidad compite con todas las demás para ser activada y sus ponderaciones, junto con las de otras unidades cercanas, denominadas colectivamente vecindario, se ajustan

respecto al valor de la neurona ganadora. Este proceso se repite muchas veces hasta que los cambios se vuelven muy pequeños. A medida que avanza el entrenamiento, los pesos en las unidades de cuadrícula se ajustan para que formen un mapa bidimensional de los grupos, es esta característica por la que se denomina mapa auto-organizado. El mapa de salida es una rejilla bidimensional de neuronas, donde cada una tiene un conjunto de vecindades, sin conexiones entre las unidades ver fig. 5.1.

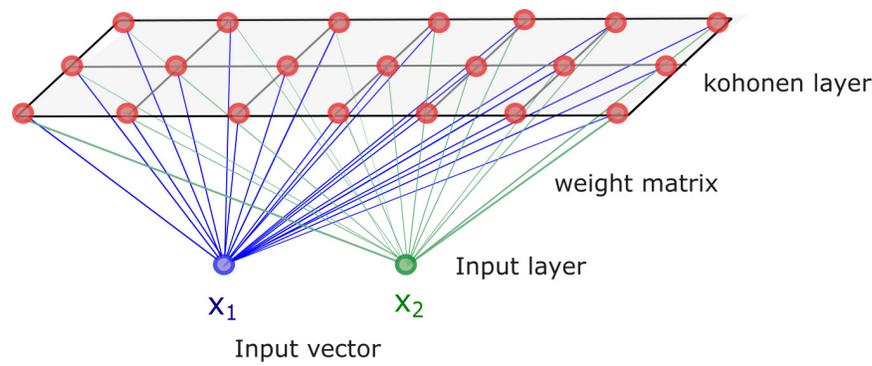


Figura 5.1: Estructura aprendizaje no supervisado de la red neuronal Kohonen.

Matemáticamente las redes neuronales de Kohonen pueden ser expresadas de la siguiente manera, comenzando por el vector de entrada [40]:

$$x = [x_1, x_2, \dots, x_n]^T, \tag{5.1}$$

el vector de peso de la sinapsis de las j-[ésimas] neuronas se inicializa con valores aleatorios pequeños y está dado por:

$$x_j = [w_{j1}, w_{j2}, \dots, w_{jn}]^T, j = 1, 2, \dots, N, \tag{5.2}$$

una vez que existe un vector de entrada se calcula la distancia con cada uno de los pesos mediante la norma euclidiana:

$$d_j = \sum_{t=0}^{N-1} (x_p(t) - w_{jp}(t))^2, \quad (5.3)$$

una vez calculadas las distancias, se encuentra la neurona ganadora del vector de x , que será la que tenga el valor mínimo de d_j . Posteriormente, comienza el proceso de adaptación o de aprendizaje, mediante el cambio de pesos que hacen que los valores se auto-organicen. Este ajuste se logra cambiando los pesos del vecindario de la neurona ganadora mediante la ecuación:

$$\frac{dw_j}{dt} = \eta[x - w_j], \quad (5.4)$$

y en forma discreta:

$$w_j(n+1) = \begin{cases} w_j(n) + \eta[x - w_j(n)], & j \in \Lambda_i(n) \\ w_j(n), & j \notin \Lambda_i(n), \end{cases} \quad (5.5)$$

donde η es la rapidez de aprendizaje y Λ es la vecindad de la neurona ganadora. Aplicando las ecuaciones (5.1)-(5.5), se logra que el vector de pesos obtenga la distribución del vector de entrada, y de esta manera se conserva la información hasta la salida de la red neuronal.

5.1 Hibridación del algoritmo NN-HGS en la optimización de controlador para el QUAV con carga suspendida

La hibridación del algoritmo NN-HGS consiste en reemplazar la fase de verificación del espacio de búsqueda, por la implementación de la Red Neuronal Kohonen para ajustar el espacio de búsqueda en cada iteración del algoritmo, como puede verse en la Figura 5.2. El proceso que sigue el algoritmo NN-HGS es el siguiente: 1) se inicializan los parámetros y posiciones de individuo, 2) se evalúa la función costo para cada uno de los individuos, 3) se hace uso de la propiedad de la red neuronal Kohonen de auto-ajustarse, para encontrar un nuevo espacio de búsqueda, 4) se calcula el hambre y se renueva a los individuos, finalmente, 5) se actualiza a la población y se repite el proceso hasta que se alcanza el número de iteraciones establecidas.

Para la implementación del algoritmo híbrido NN-HGS, se emplea la estructura de la Figura 5.3, donde se muestra el proceso que lleva a cabo el algoritmo HGS para la identificación de parámetros y optimización del controlador.

5.2 Resultados de la implementación

Para la implementación de este algoritmo híbrido NN-HGS, se emplean las mismas condiciones de software y hardware que las reportadas en la secciones 2.3 y 3.4. Se lleva a cabo la identificación de parámetros y optimización de los controladores PID

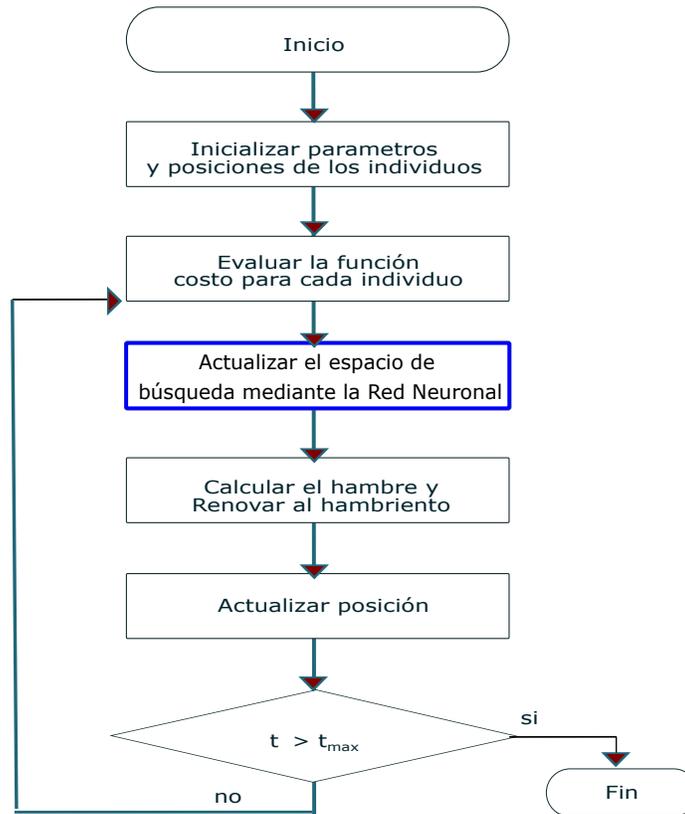


Figura 5.2: Secuencia de acciones basadas en el Hambre con ajuste de espacio de búsqueda mediante Red Neuronal. Algoritmo NN-HGS.

y ST-SMC; primero con el algoritmo HGS y una segunda prueba implementando el algoritmo NN-HGS. Ambos controles se implementan para el seguimiento de trayectoria de un QUAV con carga suspendida, cuya estructura se ha descrito a detalle en el capítulo 4. Las pruebas se desarrollan con 100 individuos, 50 iteraciones y 30 repeticiones.

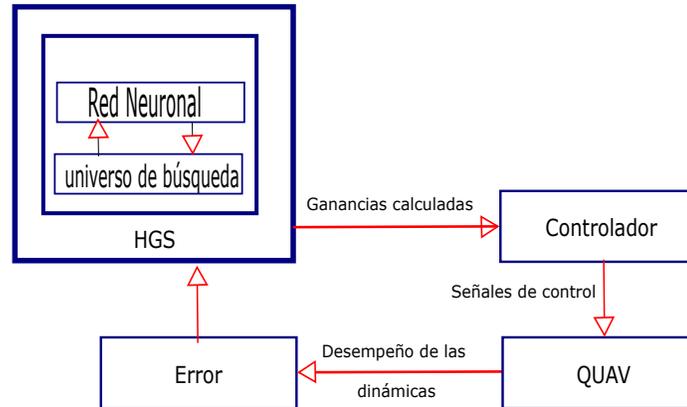


Figura 5.3: Diagrama de proceso para optimización de controlador mediante algoritmo NN-HGS.

5.2.1 Comparación de optimización de control PID con HGS y NN-HGS

El resultado del mejor valor alcanzado para el controlador PID con el algoritmo HGS es de 0.0480 y el obtenido por el NN-HGS es de 0.0204. La Tabla 5.1 muestra los parámetros encontrados por cada uno de los algoritmos. Se puede destacar que los valores de ganancias obtenidas por el HGS se mantienen en el espacio de $[0, 10]$, mientras que con el algoritmo NN-HGS el espacio de abre hasta $[0, 100]$.

En las Figuras 5.4 y 5.5 se muestra la comparación del desempeño dinámico del QUAV con carga, en las dinámicas traslacionales y rotacionales, respectivamente, con el algoritmo PID optimizado mediante HGS y NN-HGS. Los gráficos de la izquierda corresponden a la implementación del algoritmo HGS y los gráficos de la derecha muestran la respuesta con el algoritmo NN-HHGS. Se puede observar que para cada una de las dinámicas del QUAV, la respuesta obtenida con los parámetros optimizados

Tabla 5.1: Parámetros obtenidos mediante el algoritmo HGS y NN-HGS para el controlador PID.

Dinámica	Parámetro	Algoritmo optimizador	
		PID con HGS	PID con NN-HGS
		Valor	Valor
x	k_p	3	18.37026
	k_i	2.7915	3.4227
	k_d	0	0
y	k_p	3	18.3010
	k_i	2.7018	3.6711
	k_d	0	0
z	k_p	3	99.9791
	k_i	3	36.7494
	k_d	0	0.0899
ϕ	k_p	3	0.0012
	k_i	3	19.3180
	k_d	0.2053	1.4847
θ	k_p	3	16.7002
	k_i	3	25.9193
	k_d	0.2879	1.1963
ψ	k_p	3	98.3736
	k_i	3	31.4905
	k_d	0	2.2048

mediante el algoritmo NN-HGS mejora significativamente.

5.2.2 Comparación de optimización de control ST-SMC con HGS y NN-HGS

El resultado del mejor valor alcanzado para el controlador ST-SMC con el algoritmo HGS es de 0.0109 y el obtenido por el NN-HGS es de 0.0081. La Tabla 5.2 muestra los parámetros encontrados por cada uno de los algoritmos. Se puede destacar que los valores de ganancias obtenidas por el HGS se mantienen en el espacio de $[0, 10]$, mientras que con el algoritmo NN-HGS el espacio de ajusta a $[0.001, 10.05]$, lo que

nos muestra que el espacio establecido inicialmente, está muy cerca de ser el adecuado para el controlador ST-SMC.

Tabla 5.2: Parámetros obtenidos mediante el algoritmo HGS y NN-HGS para el controlador ST-SMC.

		Algoritmo optimizador	
		ST-SMC con HGS	ST-SMC con NN-HGS
Dinámica	Parámetro	Valor	Valor
	β	10	10.050
x	k_1	0.00101	0.0012
	k_2	0.0012	0.0015
	β	10	9.2503
y	k_1	0.0011	0.0021
	k_2	0.0013	0.0017
	β	9.99964	0.8851
z	k_1	0.00216	6.1721
	k_2	9.84326	9.7768
	β	8.27840	3.8790
ϕ	k_1	4.41300	0.0015
	k_2	0.11514	8.9040
	β	8.96775	7.3305
θ	k_1	0.26662	0.0011
	k_2	0.13964	9.1827
	β	0.001	10.0105
ψ	k_1	2.43695	0.41196
	k_2	9.01263	9.1316

En la Figuras 5.6 se muestra la comparación del desempeño dinámico del QUAV con carga, en las dinámicas traslacionales en el espacio tridimensional, con el algoritmo ST-SMC optimizado mediante HGS y NN-HGS. En el gráfico no es posible apreciar la diferencia del desempeño, ya que aunque el algoritmo NN-HGS (Figura 5.6a), logra alcanzar un valor de RMSE menor que el HGS (Figura 5.6b), la trayectoria del QUAV es muy cercana a la deseada en ambos casos.

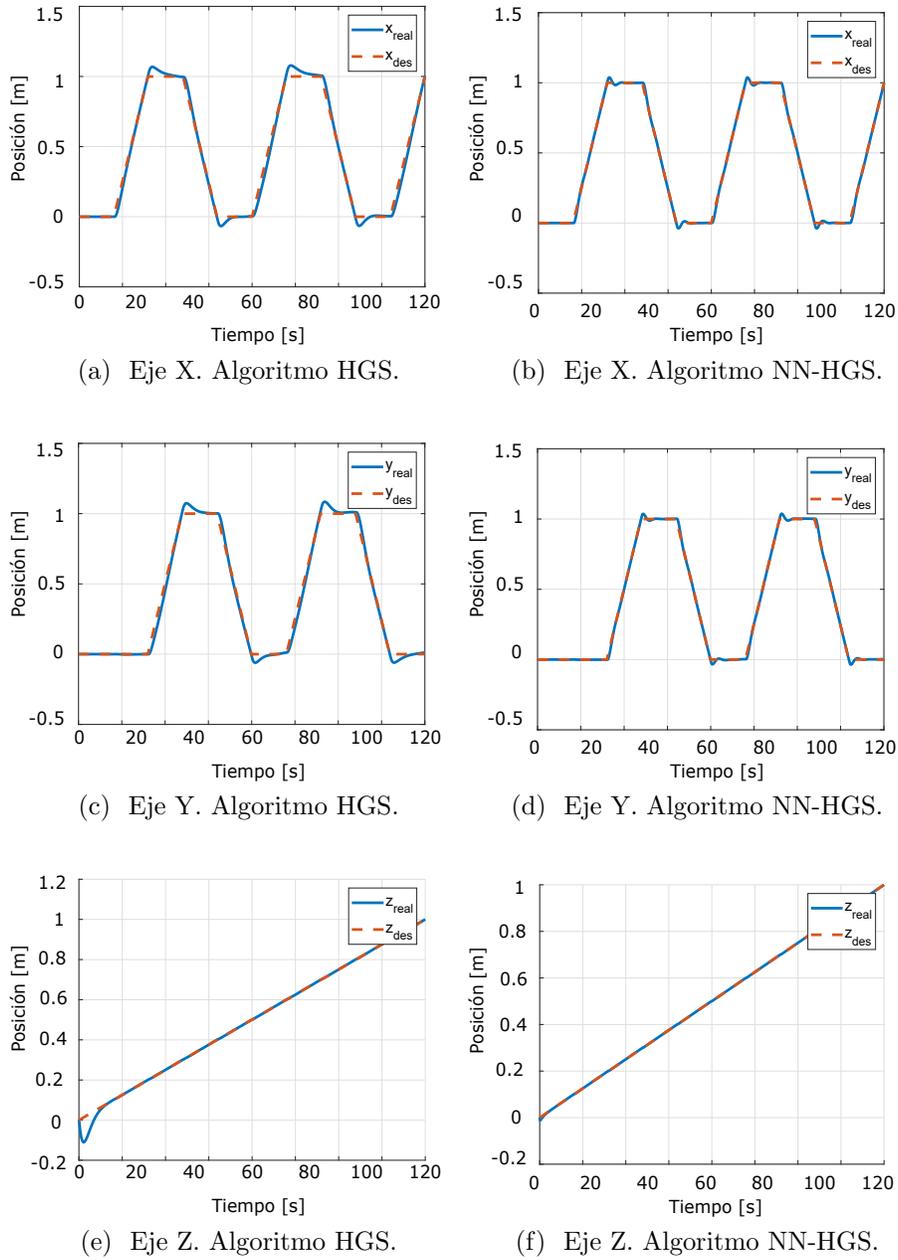
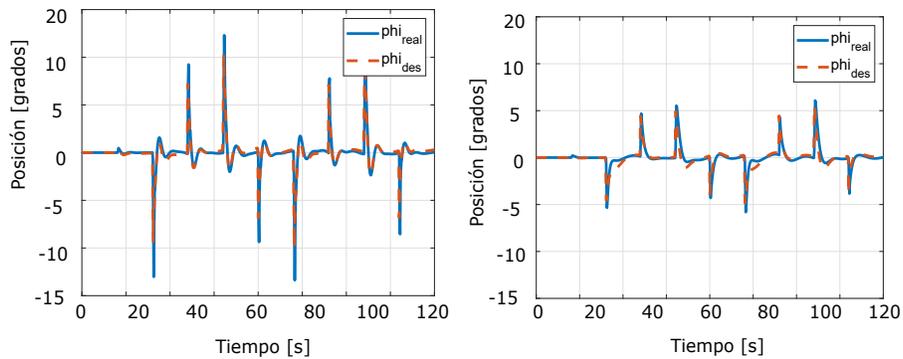
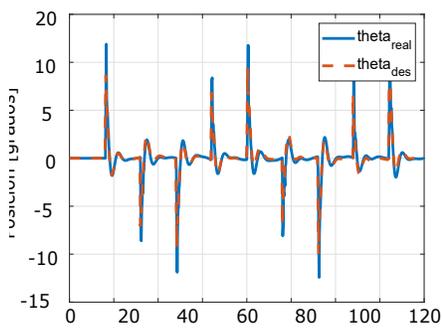


Figura 5.4: Desempeño de las dinámicas de traslación del QUAV con carga suspendida.

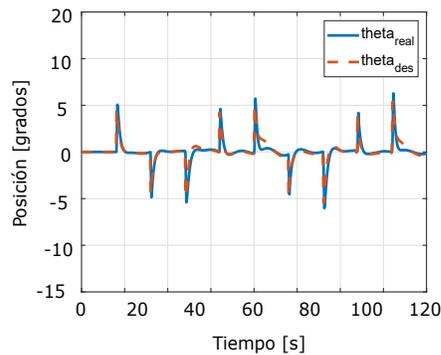


(a) Ángulo ϕ . Algoritmo HGS.

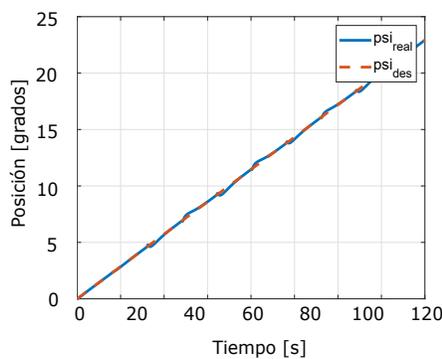
(b) Ángulo ϕ . Algoritmo NN-HGS.



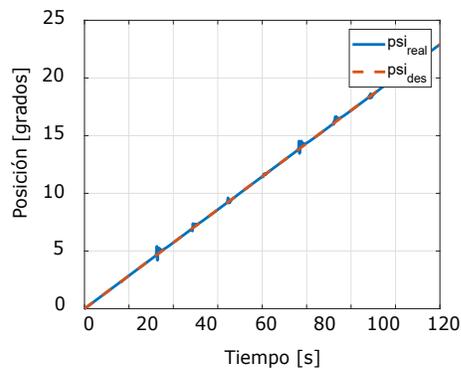
(c) Ángulo θ . Algoritmo HGS.



(d) Ángulo θ . Algoritmo NN-HGS.

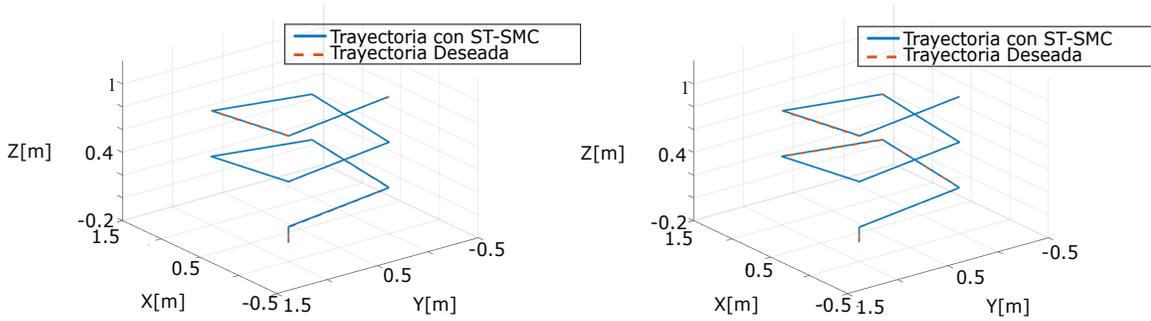


(e) Ángulo ψ . Algoritmo HGS.



(f) Ángulo ψ . Algoritmo NN-HGS.

Figura 5.5: Desempeño de las dinámicas de orientación del QUAUV con carga suspendida.



(a) Desempeño del QUAV con el controlador ST-SMC optimizado mediante HGS. (b) Desempeño del QUAV con el controlador ST-SMC optimizado mediante NN-HGS.

Figura 5.6: Comparación del desempeño del QUAV con el control ST-SMC optimizados mediante HGS y NN-HGS.

Capítulo 6

Conclusiones y trabajos futuros

6.1 Conclusiones

En este trabajo de investigación se ha propuesto una nueva metodología para la identificación, optimización, implementación y evaluación de controladores no lineales para un Vehículo Aéreo no Tripulado de tipo cuadrirrotor con carga suspendida vía cable. El método propuesto consta de 3 etapas principales: 1) elección del algoritmo metaheurístico, 2) elección del controlador y, 3) implementación para transporte de carga mediante QUAV.

En el capítulo 1, se describe a los sistemas QUAV y QUAV con carga suspendida, los cuales permiten tener conocimiento de la estructura del sistema, para proponer la metodología de identificación y optimización, y llevar a cabo la implementación.

En el capítulo 2, se lleva a cabo la comparación de los diferentes algoritmos metaheurísticos: PSO, GWO, HGS, L-SHADE, LSPACMA, MPA, SMA y WOA, para la

identificación de parámetros y optimización de controles, aplicado al vehículo QUAV. La comparación basada en la minimización del error RMSE para el seguimiento de trayectoria del QUAV, permite la elección del algoritmo metaheurístico HGS, el cual obtiene el mejor desempeño en cada uno de los experimentos, variando el número de individuos, evoluciones y repeticiones.

En el capítulo 3, se lleva a cabo la comparación de los controladores PID y ST-SMC optimizados mediante el algoritmo metaheurístico HGS. Para la comparación se llevan a cabo experimentos en simulación y en plataforma física. Adicionalmente, para verificar la robustez de los algoritmos, se agrega una perturbación externa al QUAV. Los resultados obtenidos reflejan un buen desempeño de ambos controladores bajo condiciones sin perturbaciones, sin embargo al agregar un elemento externo de disturbio, el algoritmo ST-SMC es el que logra contrarrestarlo y mantener el error entre la trayectoria deseada y la real en valores mínimos.

En el capítulo 4, se incorporan los resultados de la elección del algoritmo metaheurístico y elección del controlador, para implementarlos al sistema QUAV con carga. Se lleva a cabo la identificación de parámetros y optimización del control ST-SMC para seguimiento de trayectoria, mediante el algoritmo HGS. La estructura de control se propone considerando a la carga suspendida como una perturbación externa que afecta a las dinámicas del QUAV. Los resultados muestran que el controlador ST-SMC optimizado mediante HGS, lleva al vehículo satisfactoriamente a través de la trayectoria deseada con un error mínimo y con las oscilaciones de la carga en valores muy cercanos a cero.

Finalmente, en el capítulo 5, se lleva a cabo una mejora al algoritmo de opti-

mización HGS, ya que para sistemas no lineales como el QUAV con carga suspendida, no se tiene certeza de su comportamiento, por lo que no existe la seguridad de establecer un espacio de búsqueda adecuado para su optimización. La variante del HGS consiste en una hibridación con una red neuronal tipo Kohonen, que permite actualizar el espacio de búsqueda en cada iteración del algoritmo. Los resultados obtenidos muestran que el desempeño del algoritmo metaheurístico NN-HGS es superior que el obtenido por el HGS al minimizar el error RMSE en seguimiento de trayectoria del QUAV con carga suspendida.

6.2 Aportaciones

La principal contribución de este trabajo de investigación, es una nueva metodología para la identificación, optimización, implementación y evaluación de controladores no lineales para un Vehículo Aéreo no Tripulado de tipo cuadrirrotor con carga suspendida vía cable. Adicionalmente, se propone una mejora en el algoritmo metaheurístico HGS para la identificación de parámetros y optimización de controles para seguimiento de trayectoria aplicado al sistema QUAV con carga suspendida.

6.3 Trabajos Futuros

El siguiente paso para este trabajo de investigación es la implementación de los algoritmos optimizados en una plataforma física en exteriores, para validar la robustez en presencia de mas de una perturbación externa. También, se pretende proponer

técnicas de optimización auto-adaptable para el controlador no lineal, que pueda ser implementado en línea para los sistemas QUAV y QUAV con carga suspendida.

Referencias

- [1] S. R. Swathi Gowroju, *Drone Technology: Future Trends and Practical Applications*, vol. 1. Scrivener Publishing LLC, 1st edition ed., 2023.
- [2] “Why amazon, ups and even domino’s is investing in drone delivery services.”
- [3] N. S. Zúñiga, F. Muñoz, M. A. Márquez, E. S. Espinoza, and L. R. G. Carrillo, “Load transportation using single and multiple quadrotor aerial vehicles with swing load attenuation,” in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 269–278, 2018.
- [4] S. Ouahouah, J. Prados-Garzon, T. Taleb, and C. Benzaid, “Energy-aware collision avoidance stochastic optimizer for a uavs set,” in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1636–1641, 2020.
- [5] J. J. Enright, E. Frazzoli, M. Pavone, and K. Savla, *UAV Routing and Coordination in Stochastic, Dynamic Environments*, pp. 2079–2109. Dordrecht: Springer Netherlands, 2015.

-
- [6] L. Mora, R. Lugo, C. Moreno, and J. E. Amaya, "Parameters optimization of pid controllers using metaheuristics with physical implementation," in *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–8, 2016.
- [7] G. García-Gutiérrez, D. Arcos-Aviles, E. V. Carrera, F. Guinjoan, E. Motoasca, P. Ayala, and A. Ibarra, "Fuzzy logic controller parameter optimization using metaheuristic cuckoo search algorithm for a magnetic levitation system," *Applied Sciences*, vol. 9, no. 12, 2019.
- [8] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for pid controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, p. e09399, 2022.
- [9] K. K. E. J. P. D. (Editor), *Optimization Using Evolutionary Algorithms and Metaheuristics-Applications in Engineering*. CRC Press, 1 ed., 2019.
- [10] R. Sarker, J. Kamruzzaman, and C. Newton, "Evolutionary optimization (evopt): A brief review and analysis.," *International Journal of Computational Intelligence and Applications*, vol. 3, pp. 311–330, 12 2003.
- [11] J. C. Seck-Tuoh-Mora, N. Hernandez-Romero, P. Lagos-Eulogio, J. Medina-Marin, and N. S. Zuñiga-Peña, "A continuous-state cellular automata algorithm for global optimization," *Expert Systems with Applications*, vol. 177, p. 114930, 2021.

- [12] H. Moayedi, M. Mehrabi, D. T. Bui, B. Pradhan, and L. K. Foong, “Fuzzy-metaheuristic ensembles for spatial assessment of forest fire susceptibility,” *Journal of Environmental Management*, vol. 260, no. September, p. 109867, 2020.
- [13] M. A. Ebrahim, M. N. Ahmed, H. S. Ramadan, M. Becherif, and J. Zhao, “Optimal metaheuristic-based sliding mode control of VSC-HVDC transmission systems,” *Mathematics and Computers in Simulation*, vol. 179, pp. 178–193, 2021.
- [14] W. Boukadida, A. Benamor, H. Messaoud, and P. Siarry, “Metaheuristics-based Multi-objective Design of Global Robust Optimal Sliding Mode Control of Discrete Uncertain Systems,” *International Journal of Control, Automation and Systems*, vol. 17, no. 6, pp. 1378–1392, 2019.
- [15] M. G. D’Urso, V. Manzari, S. Lucidi, and F. Cuzzocrea, “Rescue management and assessment of structural damage by uav in post-seismic emergency,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-5-2020, pp. 61–70, 2020.
- [16] T. Üstüncök and M. Karakaya, “Effect of pso tuned p, pd, and pid controllers on the stability of a quadrotor,” in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–6, 2019.
- [17] A. Sheta, M. Braik, D. R. Maddi, A. Mahdy, S. Aljahdali, and H. Turabieh, “Optimization of pid controller to stabilize quadcopter movements using metaheuristic search algorithms,” *Applied Sciences*, vol. 11, no. 14, 2021.

- [18] “Optimización de controles mediante algoritmos metaheurísticos aplicado a vehículos aéreos.,” vol. Especial2, p. 114930, 2022.
- [19] O. Kramer, *Genetic Algorithm Essentials*, vol. 679. 2017.
- [20] N. Hernandez-Romero, J. Medina-Marin, and J. C. Seck-Tuoh-Mora, *Introducción a Matlab para Resolver Problemas de Ingeniería Aplicando Algoritmos Genéticos*. 2012.
- [21] S. Subhadip, “Genetic Algorithm : An Approach for Optimization (Using MATLAB),” *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, vol. 3, no. 3, pp. 261–267, 2014.
- [22] R. Tanabe and A. Fukunaga, “Success-history based parameter adaptation for Differential Evolution,” *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, no. 3, pp. 71–78, 2013.
- [23] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, 1995.
- [24] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [25] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, “Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts,” *Expert Systems with Applications*, vol. 177, p. 114864, 2021.

-
- [26] A. P. Piotrowski, “L-shade optimization algorithms with population-wide inertia,” *Information Sciences*, vol. 468, pp. 117–141, 2018.
- [27] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, “Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization,” *Swarm and Evolutionary Computation*, vol. 50, 2019.
- [28] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: A nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, p. 113377, 2020.
- [29] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: A new method for stochastic optimization,” *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [30] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [31] R. Tanabe and A. S. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 1658–1665, 2014.
- [32] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, “Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 145–152, 2017.

-
- [33] N. S. Zuñiga-Peña, N. Hernández-Romero, J. C. Seck-Tuoh-mora, J. Medina-Marin, and I. Barragan-Vite, “Improving 3d path tracking of unmanned aerial vehicles through optimization of compensated pd and pid controllers,” *Applied Sciences (Switzerland)*, vol. 12, no. 1, 2022.
- [34] “Constant Altitude Flight Control for Quadrotor UAVs with Dynamic Feed-forward Compensation,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 14, no. 1, pp. 26–33, 2014.
- [35] H. Khalil, *Nonlinear Control*, vol. 53. England: Pearson Education, 1st editio ed., 2015.
- [36] H. Hamadi, B. Lussier, I. Fantoni, C. Francis, and H. Shraim, “Observer-based super twisting controller robust to wind perturbation for multirotor UAV,” *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, pp. 397–405, 2019.
- [37] S. Sun, R. J. Schilder, and C. C. D. Visser, “Identification of quadrotor aerodynamic model from high speed flight data,” American Institute of Aeronautics and Astronautics Inc, AIAA, 2018.
- [38] F. Muñoz, I. González-Hernández, S. Salazar, E. S. Espinoza, and R. Lozano, “Second order sliding mode controllers for altitude control of a quadrotor UAS: Real-time implementation in outdoor environments,” *Neurocomputing*, vol. 233, pp. 61–71, 2017.

-
- [39] J. Zhang, *Kohonen Self-Organizing Map—An Artificial Neural Network*, pp. 107–125. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [40] N. H. Romero, *Control Neurodifuso de la velocidad de un motor de corriente directa*. Phd thesis, Instituto Tecnológico de la Laguna, Torreón, Coah., Febrero 2000.