

Optimización de control PD compensado mediante algoritmos metaheurísticos aplicado a un QUAV

N. Zúñiga-Peña ^{a,*}, N. Hernández-Romero ^a, J. Medina-Marín ^a, I. Barragán-Vite ^a

^aÁrea Académica de Ingeniería y Arquitectura, Universidad Autónoma del Estado de Hidalgo, 42184, Pachuca, Hidalgo, México.

Resumen

Técnicas basadas en población inspiradas en la naturaleza, han demostrado ser efectivas para resolver problemas complejos de optimización. Estos métodos son capaces de encontrar los parámetros óptimos de controladores. En este trabajo se diseña un controlador Proporcional Derivativo (PD) compensado, el cual permite a un vehículo aéreo no tripulado de cuatro rotores (QUAV) seguir una trayectoria predeterminada. Los parámetros óptimos del controlador son determinados mediante la minimización de la función costo, calculada como el error cuadrático medio (RMSE) entre las trayectorias deseadas y las reales en el espacio tridimensional. Además, se comparan seis diferentes algoritmos meta-heurísticos para su sintonización: Optimización por Nube de Partículas (PSO), la variante del Algoritmo de Evolución Diferencial (L-SHADE), Algoritmo de Optimización de Lobos (GWO), Optimización por Pastoreo de Elefantes (EHO), Algoritmo de Optimización de Ballenas (WOA) y el Algoritmo de Automata Celular de Estado Continuo (CCAA). Los resultados de la implementación muestran que se obtiene un mejor desempeño dinámico del UAV en el seguimiento de la trayectoria proporcionada con el algoritmo PSO.

Palabras Clave: Optimización por Nube de Partículas, Algoritmo de Optimización de Lobos, L-SHADE, Optimización por Pastoreo de Elefantes, Algoritmo de Optimización de Ballenas, CCAA, UAV, PD con compensación.

Abstract

Population-based techniques that are inspired by nature have resulted in effective methods to solve complex optimization problems. These methods are capable of finding optimal parameters for controllers. Here, we designed a compensated Proportional Derivative controller (PD), which enables an unmanned aerial vehicle (UAV) to be positioned in a predetermined path. Optimal parameters for our controller were determined based on minimizing a fitness function computing the error between desired and real dynamics in three-dimensional space. We compared six different metaheuristics methods: Particle Swarm Optimization (PSO), DE variant with linear population size reduction (L-SHADE), Grey Wolf Optimization (GWO), Elephant Herding Optimization (EHO), Whale Optimization Algorithm (WOA) and Continuous-state Cellular Automata Algorithm (CCAA), obtaining the best trajectory when the PSO algorithm was implemented.

Keywords: Particle Swarm Optimization, L-SHADE, Grey Wolf Optimization, Elephant Herding Optimization, Whale Optimization Algorithm, CCAA, QUAV, PD compensated.

1. Introducción

Actualmente existen diferentes problemas complejos de ingeniería, uno de los que se han popularizado en las últimas décadas son los vehículos aéreos no tripulados o UAVs por sus siglas en inglés. De entre los UAVs, el de configuración multirotor ha aventajado a los de ala fija por la capacidad de despegue y aterrizaje de manera vertical. Recientemente, los vehículos aéreos no tripulados de cuatro rotores (QUAV) son un tema de interés abordado por investigadores al rededor del mundo.

Los QUAVs son sistemas subactuados con dinámicas acopladas y comportamiento caótico (Bi et al., 2019), características que los vuelve una plataforma desafiante para el modelado y diseño de control. A pesar de su complejidad, el uso de QUAVs se ha incrementado, debido a su versatilidad de vuelo en entornos poco accesibles y peligrosos que les permite llevar a cabo tareas complejas. Las aplicaciones posibles para los QUAVs son diversas, entre ellas, la inspección y supervisión, monitoreo, recolección de imágenes y transporte de carga (Valavanis

*Nadia Samantha Zúñiga-Peña: zu450520@uaeh.edu.mx

Correo electrónico: zu450520@uaeh.edu.mx (Nadia Samantha Zúñiga-Peña), nhromero@uaeh.edu.mx (Norberto Hernández-Romero), jmedina@uaeh.edu.mx (Joselito Medina Marín), irvingb@uaeh.edu.mx (Irving Barragán Vite).

and Vachtsevanos, 2015), por mencionar algunas.

Una problemática de los QUAVs es la inestabilidad, por ello la importancia de asegurar robustez en el diseño de un buen controlador de orientación y traslación durante su trayectoria. La mayoría de 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 simplicidad de su estructura y su facilidad de implementación, los controladores clásicos de tipo Proporcional (P), Proporcional Derivativo (PD) y Proporcional Integral Derivativo (PID) son de los más recurridos, desafortunadamente, hay pocos métodos sistemáticos para establecer los parámetros como Ziegler Nichols que ha sido profundamente estudiado desde su formulación en 1942 hasta nuestros días (Ziegler and Nichols, 1942), (Åström and Hägglund, 2004) (Allu and Toding, 2020).

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. 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.

Los algoritmos meta-heurísticos 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) (Kramer, 2017), (Hernandez-Romero et al., 2012), (Subhadip, 2014) y el algoritmo de evolución diferencial (Differential Evolution, DE) (Tanabe and Fukunaga, 2013).

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. Podemos mencionar al algoritmo de optimización por enjambre de partículas (Particle Swarm Optimization, PSO) (Talbi, 2009), El Algoritmo de Optimización de Lobos (Grey Wolf Optimization, GWO) (Mirjalili et al., 2014), el Algoritmo de Optimización de Mariposa Monarca (Monarch Butterfly Optimization, MBO) (Alweshah et al., 2020), el algoritmo de Optimización por Pastoreo de Elefantes (Elephant Herding Optimization, EHO) (Wang et al., 2015a), el Algoritmo de Optimización de Ballenas (Whale Optimization Algorithm, WOA) (Mirjalili and Lewis, 2016).

Existen trabajos previos como (Ustunkok and Karakaya, 2019), donde para resolver el problema de los parámetros óptimos, implementan el algoritmo meta-heurístico de nube de partículas (PSO), donde la función de costo es la sumatoria de la media de los errores absolutos; se hace una comparación en-

tre el desempeño de los algoritmos de tipo P, PD y PID. Otro ejemplo es (Erkol, 2018), donde se lleva a cabo la optimización de parámetros de un controlador PID para las dinámicas de orientación y altura empleando los algoritmos de colonia de hormigas (ABC), PSO y genético (GA) y se compara contra el método de sintonización de Ziegler-Nichols. Estos trabajos aplicados únicamente a cuatro de las seis dinámicas de UAV (altura y orientación).

La aportación de este trabajo es la sintonización de un controlador PD con compensación para seguimiento de trayectoria tridimensional de un QUAV, mientras que se evalúa el desempeño de los algoritmos PSO, L-SHADE, GWO, MBO, EHO, WOA y CCAA, con los cuales se busca minimizar el error de seguimiento, así mismo, se muestra una comparativa del desempeño del QUAV con los parámetros de ganancias proporcional y derivativa obtenidos por cada algoritmo.

La estructura del artículo es la siguiente. En la sección 2 se obtiene el modelo matemático del QUAV, el cual se emplea para evaluar el seguimiento de trayectoria en la función costo de los algoritmos meta-heurísticos y se muestra la estrategia de control implementada, en la sección 3 se da una breve explicación de los algoritmos a comparar, en la sección 4 se muestra el proceso de optimización del controlador. En la sección 5 se presentan los resultados obtenidos. Finalmente en la sección 6 se exponen las conclusiones obtenidas.

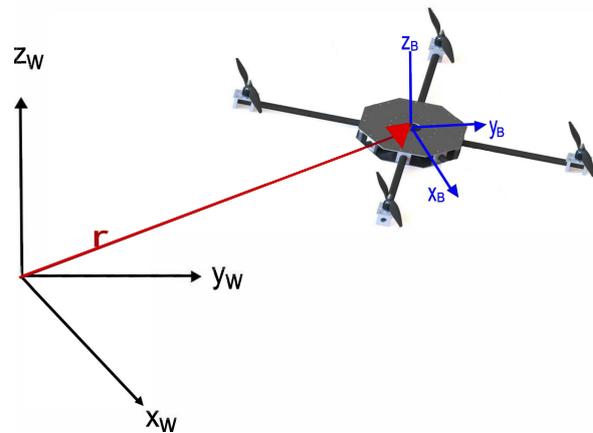


Figura 1: Sistemas de referencia inercial (W) y del cuerpo (B), para obtener el modelo matemático del QUAV.

2. Vehículo Aéreo no tripulado: modelo y estrategia de control

En esta sección se obtiene el modelo matemático que define los movimientos del QUAV, y se describe la estrategia de control implementada para seguimiento de trayectoria en el espacio tridimensional. 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.

2.1. Modelo dinámico de un QUAV

Tabla 1: Tabla 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
M	Momentos aplicados respecto al centro de masa del cuerpo
M_a	Momentos resultantes de las fricciones aerodinámicas
M_b	Momentos generados por los motores.
I_B	Matriz de tensión de inercias
F_B	Vector de fuerza de sustentación o <i>thrust</i>
F_d	Vector de fuerzas de arrastre
F_g	Vector de fuerzas debido a la gravedad
$k_{1,2,\dots,6}$	Coefficientes de arrastre
s, c	$\sin(\cdot), \cos(\cdot)$

$${}^c R_B = \begin{pmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$${}^a R_c = \begin{pmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{pmatrix}$$

$${}^W R_a = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{pmatrix}$$

$${}^W R_B = \begin{pmatrix} c\theta c\psi & c\psi s\theta s\phi - c\phi s\psi & c\psi c\phi s\theta + s\psi s\phi \\ c\theta s\psi & c\psi c\phi + s\theta s\psi s\phi & c\phi s\theta s\psi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix}$$

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 2.

Tabla 2: Relacion 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 $\dot{\psi}$:

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

para $\dot{\theta}$:

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

para $\dot{\phi}$:

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

Entonces :

$$\omega_B = \begin{pmatrix} -s(\theta)\dot{\psi} \\ c(\theta)s(\phi)\dot{\psi} \\ c(\theta)c(\phi)\dot{\psi} \end{pmatrix} + \begin{pmatrix} 0 \\ c(\phi)\dot{\theta} \\ -s(\phi)\dot{\theta} \end{pmatrix} + \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix}$$

$$\omega_B = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} - \dot{\psi}s(\theta) \\ c(\phi)\dot{\theta} + c(\theta)\dot{\psi}s(\phi) \\ c(\theta)c(\phi)\dot{\psi} - \dot{\theta}s(\phi) \end{pmatrix},$$

reescribiendo queda como:

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

El modelo desarrollado en este trabajo se obtiene bajo las siguientes suposiciones:

- La estructura es rígida.
- La estructura es simétrica.
- El centro de gravedad y el origen del marco del cuerpo coinciden con el centro geométrico del vehículo.
- Las propelas son rígidas

La posición absoluta del QUAV, se define como las coordenadas desde el centro de gravedad del QUAV (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 QUAV con respecto al marco inercial, estos ángulos definen a las rotaciones de *yaw*, *roll* y *pitch*, es decir, indican cómo el sistema coordinado, que inicialmente está alineado con el sistema inercial, debe ser rotado, para alinearse con el sistema coordinado 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 2.

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 coordenado del cuerpo, están dadas por las (1) y (2).

$$m\dot{\mathbf{v}} = \mathbf{F}_g + \mathbf{F}_d + \mathbf{F} \quad (1)$$

$$\mathbf{I}_B \dot{\boldsymbol{\omega}}_B + (\boldsymbol{\omega}_B \times \mathbf{I}_B \boldsymbol{\omega}_B) + \mathbf{M}_a = \mathbf{M}_B \quad (2)$$

En la Ecuación (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 (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) y (2) se obtienen las siguientes dinámicas de traslación y rotacionales del QUAUV:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{m} (c\psi s\theta c\phi + s\phi s\psi) u_1 - \frac{k_1}{m} \dot{x} \\ \frac{1}{m} (s\psi s\theta c\phi - s\phi c\psi) u_1 - \frac{k_2}{m} \dot{y} \\ \frac{1}{m} (c\phi c\theta) u_1 - g - \frac{k_3}{m} \dot{z} \\ \frac{(I_y - I_x)}{I_x} \dot{\theta} \dot{\psi} - \frac{k_4}{I_x} \dot{\phi}^2 + \frac{1}{I_x} u_2 \\ \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} - \frac{k_5}{I_y} \dot{\theta}^2 + \frac{1}{I_y} u_3 \\ \frac{(I_x - I_y)}{I_z} \dot{\phi} \dot{\theta} - \frac{k_6}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \end{bmatrix} \quad (3)$$

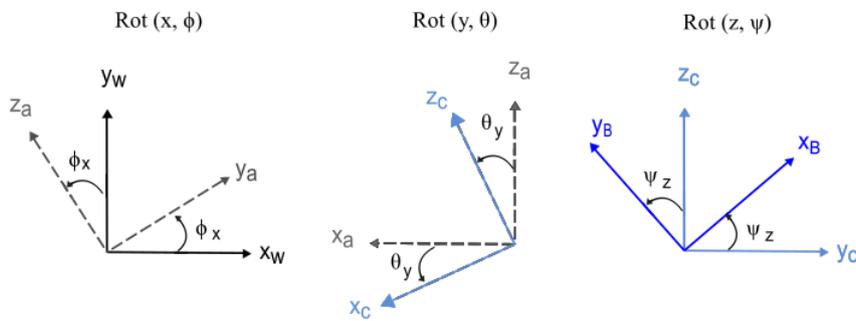


Figura 2: Orden de rotación del marco W a B.

2.2. Estrategia de control para seguimiento de trayectoria

El objetivo del control es llevar al cuadricóptero a la referencia deseada en X- Y - Z. Como se muestra en la Figura 3, 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.

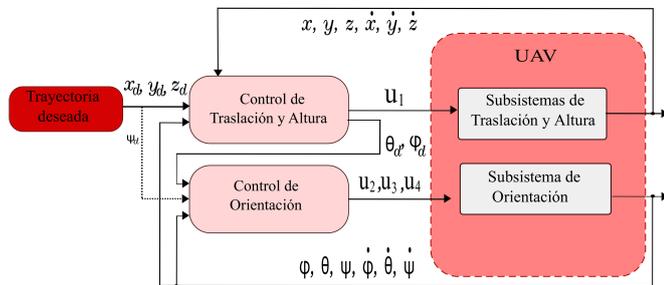


Figura 3: Estructura de control propuesta para seguimiento de trayectoria.

2.3. Diseño de controlador PD

Debido a la naturaleza no lineal del sistema UAV, se propone un controlador PD compensado para seguimiento de trayectoria (Razinkova et al., 2014).

Las dinámicas mostradas en (3) se pueden representar como un sistema de segundo orden, de la forma

$$\dot{\chi} = f(\chi) + g(\chi)u, \quad (4)$$

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

$$u = \frac{1}{g(\chi)} [-f(\chi) + K^T E + \ddot{\chi}_d], \quad (5)$$

donde, $\ddot{\chi}_d$ es la segunda derivada de la trayectoria deseada, $E = [e \ \dot{e}]^T$, con $e = \chi_d - \chi$ sustituyendo (5) in (4), se tiene que la dinámica en lazo cerrado del sistema es

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

2.3.1. Control Traslacional y de Altura

La dinámica de la altitud mostrada en (3), puede representarse como (4),

$$\ddot{z} = \frac{1}{m} (\cos \phi \cos \theta) u_1 - g - \frac{k_3}{m} \dot{z}, \quad (6)$$

de esta manera si se diseña la señal de control como en (5) se obtiene

$$u_1 = \frac{m}{\cos \theta \cos \phi} \left[\frac{k_3}{m} \dot{z} + K_z^T E_z + \ddot{z}_d \right] + g, \quad (7)$$

la ecuación (7) es válida para $-\pi/2 < \phi < \pi/2$ and $-\pi/2 < \theta < \pi/2$.

Para la dinámica traslación en el eje X, se elije a una señal de control virtual como $\mu_x = \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi$, obteniendo la dinámica

$$m\ddot{x} = f(x) + \mu_x u_1, \quad (8)$$

y la señal de control se elige como en (5)

$$\mu_x = \frac{m}{u_1} \left[\frac{k_1}{m} \dot{x} + K_x^T E_x + \ddot{x}_d \right]. \quad (9)$$

De manera similar para la dinámica en el eje Y, se elije a la señal de control virtual como $\mu_y = \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi$, obteniendo la dinámica

$$m\ddot{y} = \frac{k_2}{m} \dot{y} + \mu_y u_1, \quad (10)$$

y la señal de control se elige como en (5)

$$\mu_y = \frac{m}{u_1} \left[\frac{k_2}{m} \dot{y} + K_y^T E_y + \ddot{y}_d \right]. \quad (11)$$

Las ecuaciones (9) y (11) 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 :

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

$$\phi_d = a \sin(\sin \psi \mu_x - \cos \psi \mu_y)$$

$$\theta_d = a \sin \left(\frac{\cos \psi \mu_x + \sin \psi \mu_y}{\cos \phi} \right). \quad (12)$$

2.3.2. Control de Orientación

Para las dinámicas de orientación, se tiene que pueden ser representadas como en (4) y diseñarse la señal de control como en (5).

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

$$u_2 = I_x \left(\frac{k_4}{I_x} \dot{\phi}^2 - \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} + K_\phi^T E_\phi + \ddot{\phi}_d \right), \quad (13)$$

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

$$u_3 = I_y \left(\frac{k_5}{I_y} \dot{\theta}^2 - \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} + K_\theta^T E_\theta + \ddot{\theta}_d \right), \quad (14)$$

Finalmente, para el movimiento de *yaw*, se tiene la señal de

control u_4 como:

$$u_4 = I_z \left(\frac{k_6}{I_z} \dot{\psi}^2 - \frac{(I_x - I_y)}{I_z} \dot{\phi} \dot{\theta} + K_\psi^T E_\psi + \ddot{\psi}_d \right). \quad (15)$$

3. Información general de los algoritmos PSO, L-SHADE, GWO, MBO, EHO, WOA y CCAA

En esta sección se presentan las ideas fundamentales detrás de cada uno de los algoritmos meta-heurísticos a comparar para la sintonización de parámetros de controlador PD compensado.

3.1. 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 (Kennedy and Eberhart, 1995). En el PSO, cada individuo es conocido como partícula que a su vez pertenece al enjambre o nube. La figura 4 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 (16), (17),

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 R_1(P_{ij}(t) - x_{ij}(t)) + c_2 R_2(P_{gj}(t) - x_{ij}(t)), \quad (16)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (17)$$

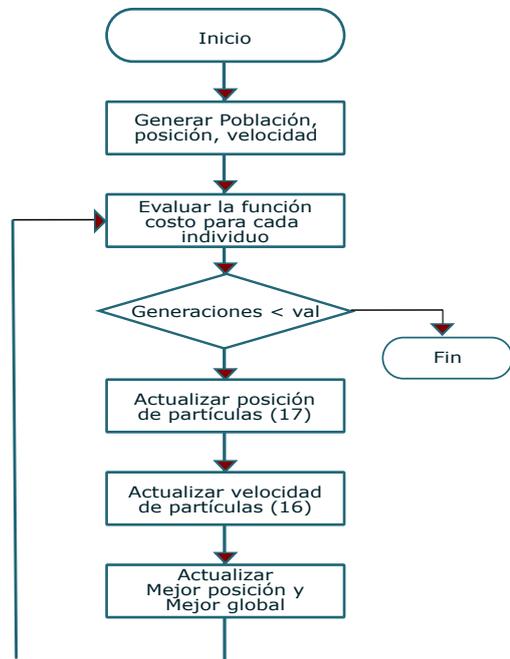


Figura 4: Diagrama de flujo del PSO.

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.

3.2. 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 (Tanabe and Fukunaga, 2013), 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 (Tanabe and Fukunaga, 2014). EL Algoritmo L-SHADE sigue la secuencia mostrada en la Fig. 5 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 (Mohamed et al., 2019).

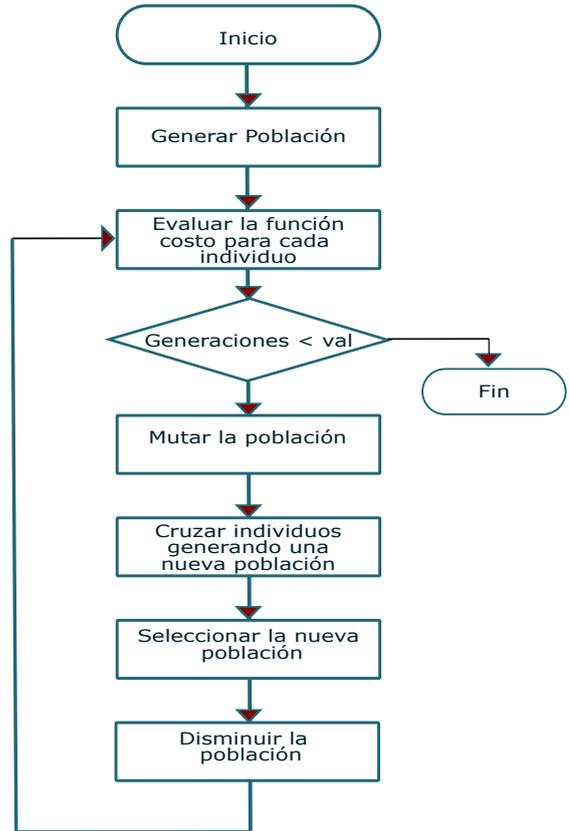


Figura 5: Diagrama de flujo del L-Shade.

3.3. Algoritmo de Optimización de Lobos

El Algoritmo de Optimización de Lobos (GWO), está inspirado en el comportamiento de las manadas de lobos grises y su organización social para la caza de presas. 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, estos son la segunda y tercera mejor solución, respectivamente. y 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 (Mirjalili et al., 2014). El diagrama de flujo de la Fig. 6, describe de manera gráfica el proceso de optimización del GWO.

El proceso de rodear a la presa se modela por las ecuaciones (18)- (19),

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \quad (18)$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \quad \vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \quad \vec{C} = 2\vec{r}_2, \quad (19)$$

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.

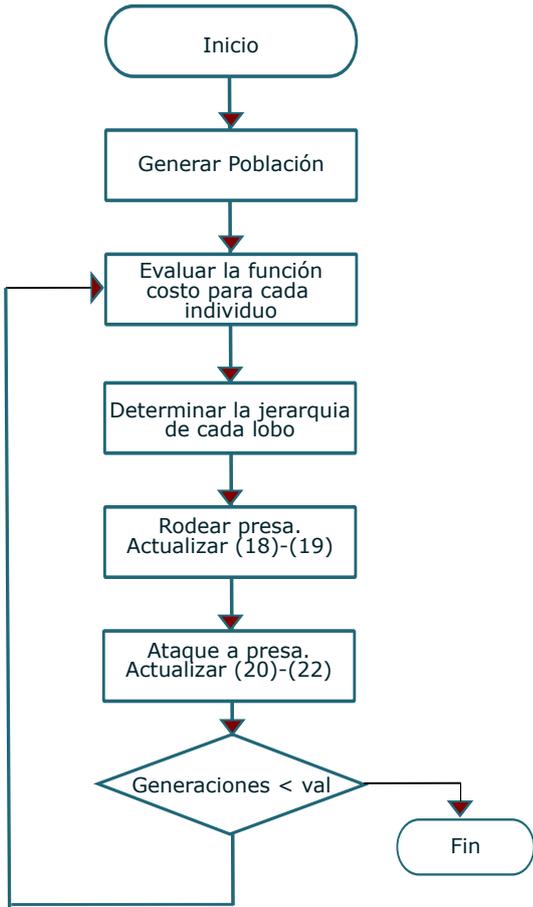


Figura 6: Diagrama de flujo del GWO.

el proceso de ataque a la presa, dirigido por el lobo alfa, se define por (20)-(22)

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (20)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_\beta = \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_\delta = \vec{A}_3 \cdot \vec{D}_\delta, \quad (21)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}. \quad (22)$$

3.4. Algoritmo de Optimización de Mariposa Monarca

El Algoritmo de Optimización de Mariposa Monarca (MBO), está inspirado en el comportamiento migratorio de las mariposas, representado en la Fig. 7. La población después de generarse de manera aleatoria, es evaluada de acuerdo a su posición, posteriormente la población es ordenada por su desempeño y es dividida en dos ubicaciones, 1 y 2, posteriormente, se genera nueva sub-población en cada una de las ubicaciones que sirven para reemplazar a la población antigua, solamente si su desempeño es mejor, de lo contrario los individuos prevalecen a la siguiente generación, con lo que se asegura que la población no se deteriore. El proceso se repite hasta alcanzar el criterio de término (Wang et al., 2015b).

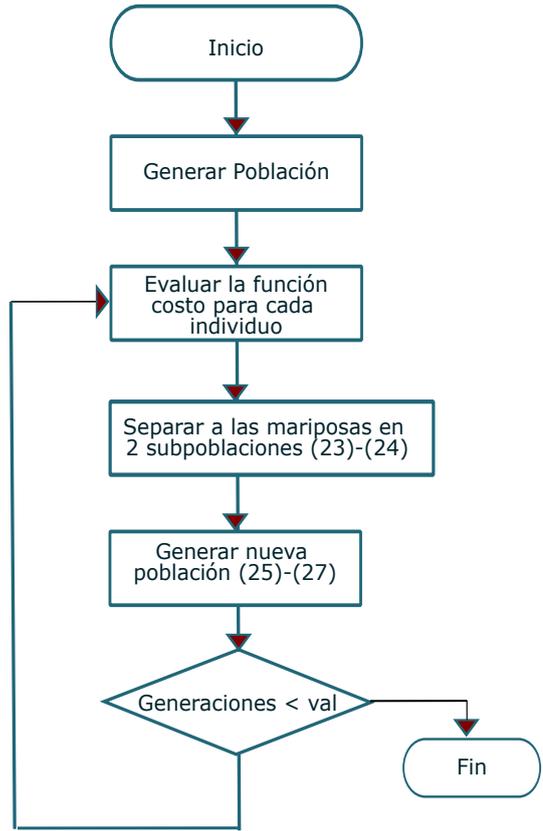


Figura 7: Diagrama de flujo del MBO.

El proceso de migración se expresa por (23), donde mediante una comparación de un número aleatorio (r) con el periodo de migración (p), se ubica a la población en la sub-población 1, mediante (23)-(24) y en la sub-población 2 mediante el operador de ajuste mediante (25)-(27) (Alweshah et al., 2020)

$$x_{i,k}^{t+1} = x_{r_1,k}^t, \quad (23)$$

$$x_{i,k}^{t+1} = x_{r_2,k}^t, \quad r = rand * peri \quad (24)$$

$$x_{j,k}^{t+1} = x_{best,k}^t, \quad (25)$$

$$x_{j,k}^{t+1} = x_{r_3,k}^t = x_{j,k}^{t+1} + \alpha \times (dx_k - 0,5), \quad (26)$$

$$dx = Levy(x_j^t), \quad \alpha = S_{max}/t^2. \quad (27)$$

La expresión $x_{i,k}^{t+1}$, indica el k -ésimo elemento de x_i en la generación $t + 1$ que representa la posición de la mariposa i . De la misma manera, $x_{r_1,k}^t$ indica el k -ésimo elemento de x_{r_1} , que es la nueva posición generada de la mariposa r_1 . La generación actual se representa por t . Si r , el numero aleatorio es menor al periodo de migración (1.2 por los 12 meses del año), se genera una nueva generación de mariposas con distribución uniforme, de lo contrario, una nueva generación de mariposas es calculada mediante las ecuaciones del operador de ajuste, donde $x_{j,k}^{t+1}$, es el k -ésimo elemento de x_j en la generación $t + 1$, que representa la posición de la mariposa j . Así mismo, $x_{best,k}^t$, es la mejor mariposa de la generación actual (25). Lo anterior si el número aleatorio r es igual o menor que p , de lo contrario, $x_{j,k}^{t+1}$ se elige

de manera aleatoria con (26), donde α es un factor de peso, dx es el tamaño de paso calculado por el factor de desempeño de vuelo $Levy$ y S_{max} , es el máximo tamaño de paso.

3.5. Algoritmo de Pastoreo de Elefantes

El algoritmo de Pastoreo de Elefantes (EHO), está inspirado en el comportamiento de grupos de elefantes, quienes formados en clanes con un número fijo de miembros, están liderados por una matriarca. Dos comportamientos principales son modelados en el algoritmo, la actualización del clan con respecto a las posiciones de cada individuo y de la matriarca y la separación de los elefantes macho adultos, lo cual permite diversificar a la población. La matriarca tiene la mejor posición en cada clan e influencia a las posiciones de los demás miembros del clan, mientras que los elefantes macho son los miembros que tienen la peor posición y son separados y reemplazados en el clan (Wang et al., 2015a), (Elhosseini et al., 2019). El proceso que lleva a cabo el algoritmo EHO, puede observarse en la Fig. 8. La posición de cada individuo es actualizada mediante (28),

$$x_{new,c_i,j} = x_{c_i,j} + \alpha \times (x_{best,c_i} - x_{c_i,j}) \times r, \quad (28)$$

donde α , es el factor de influencia de la matriarca, c_i es la posición de cada elefante, $x_{new,c_i,j}$ y $x_{c_i,j}$, son las nuevas posiciones actualizadas así, el mejor elefante en cada clan es actualizado por (29), con β como el factor de influencia del centro del clan

$$x_{new,c_i,j} = \beta \times x_{center,c_i}, \quad x_{center,c_i} = \frac{1}{n_{c_i}} \times \sum_{j=1}^{n_{c_i}} x_{c_i,j,d}, \quad (29)$$

finalmente, el operador de separación, se calcula con (30)

$$x_{worst,c_i} = x_{min} + (x_{mzs} - x_{min} + 1) \times rand. \quad (30)$$

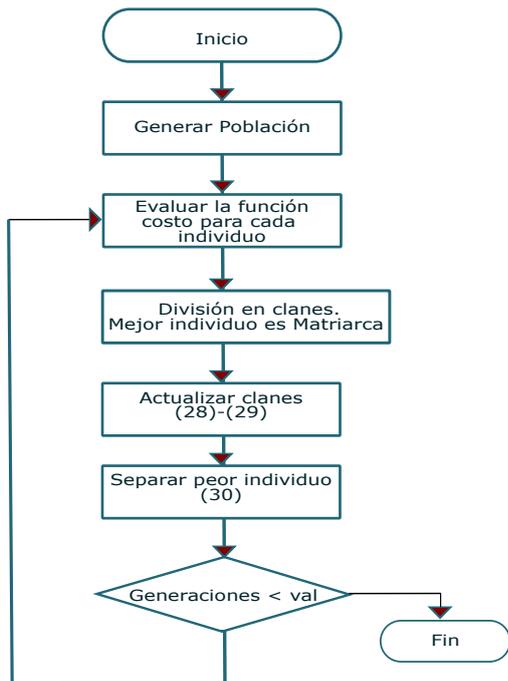


Figura 8: Diagrama de flujo del EHO.

3.6. Algoritmo de Optimización de Ballenas

El Algoritmo de Optimización de Ballenas (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 (Mirjalili and Lewis, 2016). El diagrama de flujo del WOA, se observa en la Fig. 9. Las ecuaciones que describen el comportamiento de rodear a la presa se describen en (31), (32)

$$\vec{D} = |\vec{C} \cdot \vec{X}_{best}(t) - \vec{X}(t)|, \quad \vec{D}' = |\vec{X}_{best}(t) - \vec{X}(t)| \quad (31)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \quad \vec{C} = 2 \cdot \vec{r}, \quad (32)$$

$$\vec{X}(t+1) = \vec{X}_{best}(t) - \vec{A} \cdot \vec{D}, \quad (33)$$

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos 2\pi l + \vec{X}_{best}(t), \quad (34)$$

los movimientos de la estrategia de burbuja puede ser descrito en (33) como mecanismo de cerco, y se obtiene reduciendo el valor de a de 2 a 0, ó mediante (34) que modela un movimiento en espiral, con b constante y l un número aleatorio.

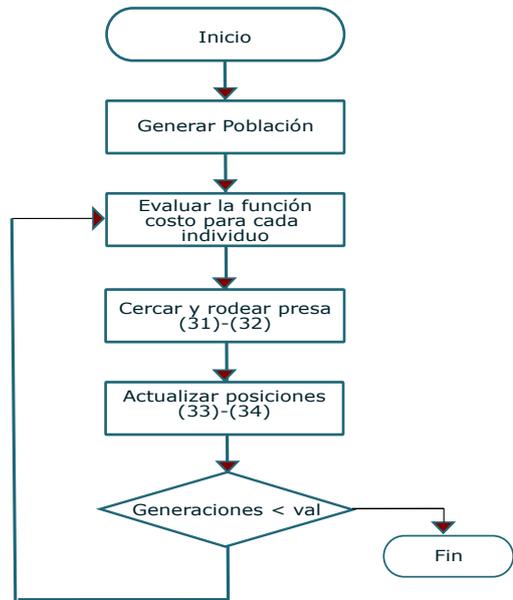


Figura 9: Diagrama de flujo del WOA.

3.7. Algoritmo de Autómata Celular de Estado Continuo

El Algoritmo de Autómata Celular de Estado Continuo (CCAA), es un algoritmo de reciente creación, basado en el concepto de vecindad y las reglas de evolución de los autómatas celulares, denominado de estado continuo por la naturaleza de las variables usadas. Al inicio, el CCAA genera un conjunto de posibles soluciones o células de manera aleatoria, y a partir de ellas genera un vecindario inspirado en las reglas de los autómatas celulares, con el propósito de compartir información entre individuos, tomar información del individuo con el mejor desempeño y generar cambios espontáneos, con lo que se maximiza la exploración y explotación en el espacio de búsqueda (Seck-Tuoh-Mora et al., 2021), (ver Fig. 10). En este algoritmo se usan las siguientes reglas de evolución adaptadas al autómata celular para aplicarse a las células y a su desempeño:

- Regla para aproximarse a las células vecinas. Permite realizar búsqueda en el vecindario cercano con diferentes valores de desempeño, para la explotación .
- Regla para mantenerse alejado de una célula vecina. Permite hacer exploración y salir de mínimos locales.
- Reglas para hacer cambios en una célula.
- Regla para incrementar valores en una célula. Genera pequeños cambios que contribuyen a la explotación en la búsqueda.
- Regla de mayoría de valores en una célula. Toma los elementos más repetidos y forma una nueva solución cercana a dichos valores.
- Regla para rodear valores en una célula. Genera cambios al rededor de valores de las células considerando su desempeño.

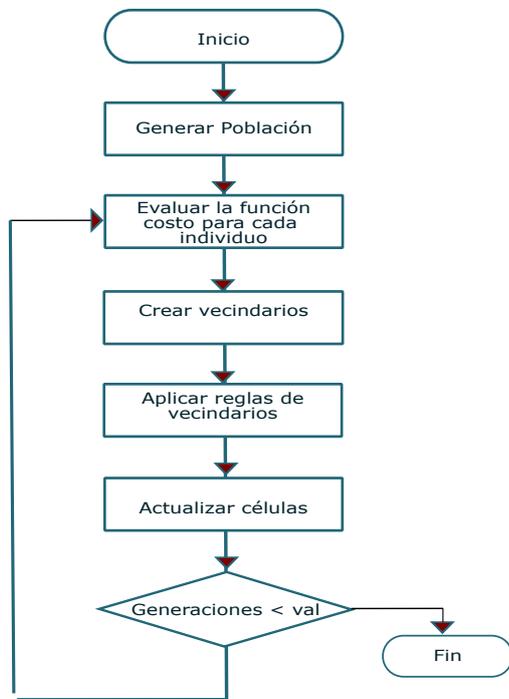


Figura 10: Diagrama de flujo del CCAA.

4. Optimización de PD compensado

La figura 11, muestra el esquema implementado para la sintonización de ganancias del controlador, donde se evalúa cada uno de los algoritmos meta-heurísticos. 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 el error 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.

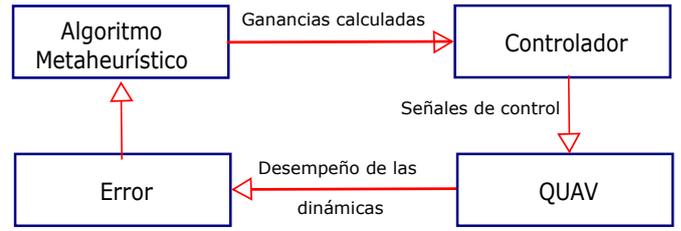


Figura 11: Esquema de sintonización de ganancias del controlador.

Para obtener las ganancias k_p y k_d de cada una de las 6 dinámicas del QUAV, mediante los algoritmos evolutivos descritos en 3, la población a optimizar es:

$$P = \{k_{p_x}, k_{d_x}, k_{p_y}, k_{d_y}, k_{p_z}, k_{d_z}, k_{p_\phi}, k_{d_\phi}, k_{p_\theta}, k_{d_\theta}, k_{p_\psi}, k_{d_\psi}\}, \quad (35)$$

siendo el espacio de búsqueda para cada uno de los individuos $S = [0, 10]$. La función costo calculada mediante el error cuadrático medio:

$$ff = \sqrt{\frac{1}{N} \sum_{i=1}^G \sum_{k=0}^N (\chi_d^i(k) - \chi^i(k))^2}, \quad (36)$$

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.

5. Resultados

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 3, se implementan los algoritmos para la sintonización de ganancias del controlador descrito en 2.2. Los algoritmos PSO, GWO, LSHADE y WOA, no tienen parámetros adicionales que tengan que ser jaustados para la implementación. Los parámetros físicos de QUAV empleados para la evaluación de la función costos se muestran en la Tabla 4. 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 12, se muestra el desempeño de cada uno de los algoritmos al paso de las 50 evoluciones, a excepción del EHO, todos los algoritmos convergen de una manera similar. En las tablas 5 y 6, se muestran los parámetros optimizados por cada algoritmo y la estadística de desempeño de cada uno, respectivamente.

En las figuras 13-16, se muestra el desempeño del controlador con las diferentes ganancias obtenidas con cada uno de los algoritmos, se puede observar que el mejor desempeño lo obtiene el algoritmo PSO. La Fig. 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. 14 muestra el comportamiento de los ángulos de orientación. Puede observarse que las dinámicas responden a los cambios de di-

rección de las trayectorias traslacionales deseadas. Las dinámicas traslacionales pueden observarse en las Figuras 15 y 16, donde las dinámicas obtenidas mediante la implementación de las ganancias de cada uno de los algoritmos meta-heurísticos alcanzan a las trayectorias deseadas.

La figura 17 muestra el desempeño del QUVAV 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 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 QUVAV a la referencia.

Tabla 3: Parámetros utilizados para la evaluación de los algoritmos.

Algoritmo	Parámetro	Valor	Algoritmo	Parámetro	Valor
CCAA	Smart Cells	5	EHO	Tamaño de clan	4
	Neighbors	20		Influencia alfa	0.8
				Influencia beta	0.1

Tabla 4: Parámetros físicos de modelo de QUVAV.

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 5: Ganancias obtenidas con cada uno de los algoritmos.

Parámetro	PSO	CCAA	EHO	GWO	LSHADE	WOA
k_{px}	5.38034917096284	5.47847588433664	4.85420088574228	5.36144297442539	5.48636612067426	5.40176353995843
k_{dx}	10	9.99999544526520	7.94669656129428	10	9.99923829500062	10
k_{py}	5.34363161851617	5.40000000000000	5.02909121828592	5.31460193511851	5.27127654482579	5.21073696588203
k_{dy}	10	9.99999112099921	5.53478957941692	10	9.99929132869554	10
k_{pz}	10	8.31141773499636	7.79363498927639	10	9.91133528486919	10
k_{dz}	10	10	3.45699051087714	9.97609523098564	9.95446256898364	10
$k_{p\phi}$	5.35583778529942	7.60000000000000	2.47826322788889	5.29507961390626	5.27061512810914	4.34069745840423
$k_{d\phi}$	7.76999286985307	9.40000000000000	3.10308826029709	7.58016811309925	6.98220389082724	6.28480002143812
$k_{p\theta}$	4.81932549967459	4.80000000000000	5.71580647629947	4.80003108127045	4.83160399212785	4.82913919976261
$k_{d\theta}$	0	0	0.0881962983926761	0	0.000867947820502628	0
$k_{p\psi}$	1.38053382200646	9.80000000000000	0.263138673522767	4.57682365290264	3.86782052211034	10
$k_{d\psi}$	0.670033962378238	6.04223649221710	6.77365658181977	6.26077192611039	3.70380858735175	5.25928137022298

Tabla 6: Estadística de desempeño de cada algoritmo.

Parámetro	PSO	GWO	WOA	LSHADE	CCAA	EHO
Mejor valor	0.0372456781929264	0.0372473206168949	0.0372507886294010	0.0372676029610235	0.0373377915135839	0.0580722713490713
Peor valor	0.0372459752597788	0.0372859296042066	0.0375577596589272	0.0373402205568464	333.418426913751	0.114337075368488
Media	1.13014047002650e-06	3.84600988055340e-05	0.00113472957318263	3.82667719414363e-05	1825.72579265201	0.0450488764139258
Desviación estándar	0.0372501327845111	0.0374093549686101	0.0432683216413195	0.0374292396126959	10000	0.199483566760266

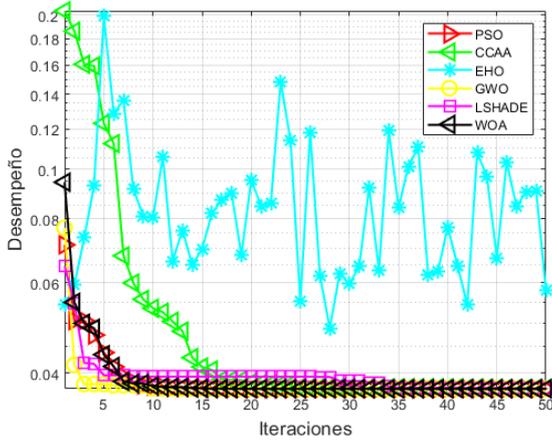


Figura 12: Desempeño de los algoritmos evolutivos en cada evolución.

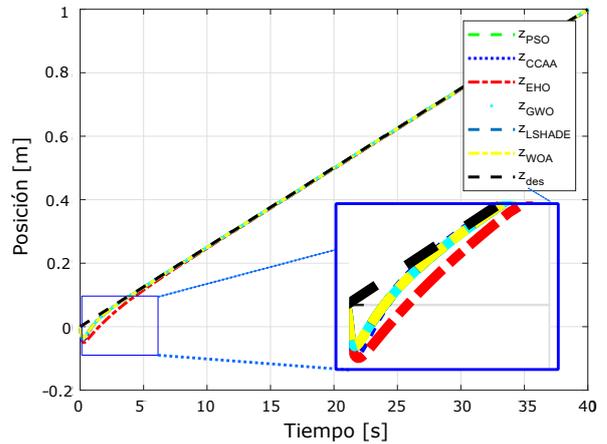


Figura 13: Desempeño de del UAV en el eje Z, comparado contra la trayectoria deseada.

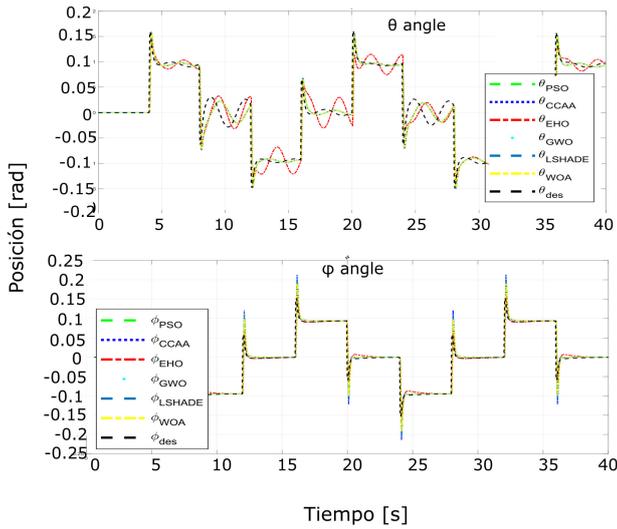


Figura 14: Desempeño de los ángulos ϕ y θ , contra los ángulos deseados calculados mediante el control virtual.

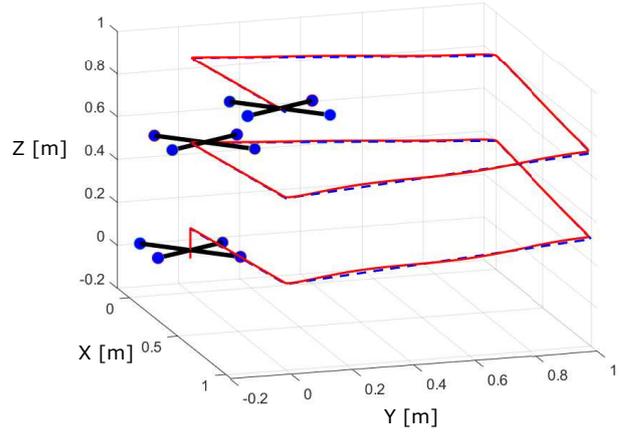


Figura 17: Seguimiento de trayectoria del QUAV en X - Y - Z.

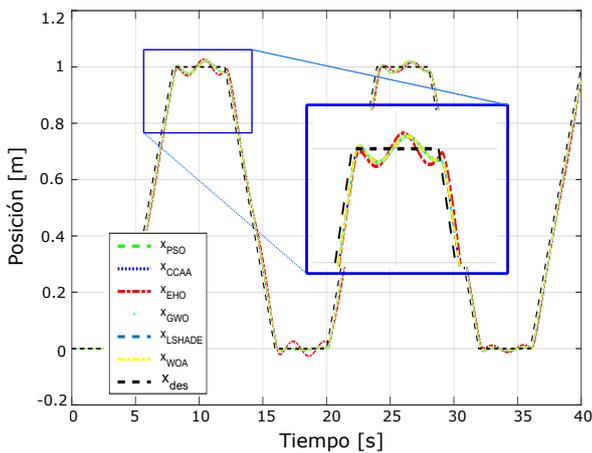


Figura 15: Desempeño de del UAV en el eje X, comparado contra la trayectoria deseada.

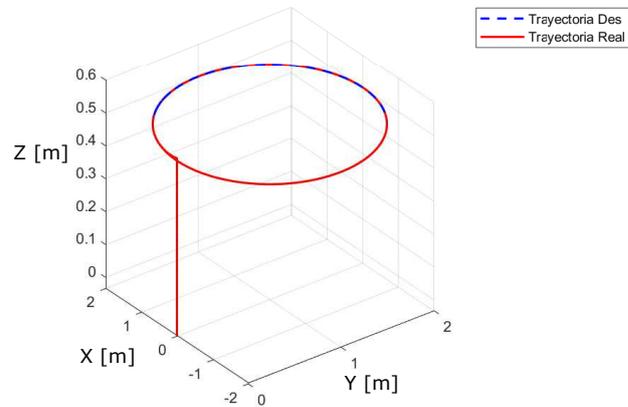


Figura 18: Evaluación de diferente trayectoria usando las ganancias obtenidas mediante optimización.

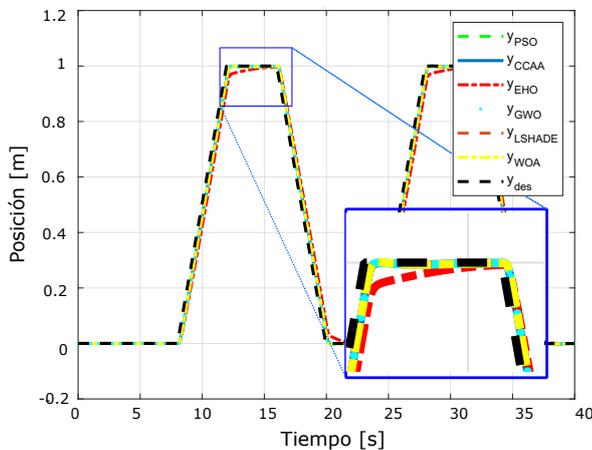


Figura 16: Desempeño de del UAV en el eje Y, comparado contra la trayectoria deseada.

6. Conclusiones

En este trabajo se ha abordado el problema de sintonización del algoritmo PD compensado para el seguimiento de trayectoria de un vehículo QUAV, el cual se modeló incluyendo las seis dinámicas de traslación y orientación. Se compararon los algoritmos meta-heurísticos PSO, GWO, WOA, LSHADE, CCAA y EHO, encontrando parámetros óptimos para seleccionar el algoritmo más adecuado. El algoritmo que tiene un mejor desempeño es el PSO al minimizar el error de seguimiento de mejor manera, con un $RMS E = 0,0372456781929264$. Los resultados de la simulación muestran que los algoritmos propuestos son capaces de optimizar las ganancias de un controlador para seguimiento de trayectoria de un QUAV. Las ganancias obtenidas permiten al QUAV seguir una trayectoria deseada en el espacio tridimensional. Así mismo, las ganancias obtenidas permiten al QUAV seguir una trayectoria deseada diferente a la trayectoria con la cual fueron optimizados los parámetros del controlador.

Referencias

- Allu, N. and Toding, A. (2020). Tuning with Ziegler Nichols Method for Design PID Controller at Rotate Speed DC Motor. *IOP Conference Series: Materials Science and Engineering*, 846(1).
- Alweshah, M., Khalailah, S. A., Gupta, B. B., Almomani, A., Hammouri, A. I., and Al-Betar, M. A. (2020). The monarch butterfly optimization algorithm for solving feature selection problems. *Neural Computing and Applications*, 0.
- Åström, K. J. and Hägglund, T. (2004). Revisiting the Ziegler-Nichols step response method for PID control. *Journal of Process Control*, 14(6):635–650.
- Bi, H., Qi, G., Hu, J., Faradja, P., and Chen, G. (2019). Modeling and Analysis of Chaos and Bifurcations for the Attitude System of a Quadrotor Unmanned Aerial Vehicle. *Chaos, Solitons and Fractals*, 138:109815.
- Elhosseini, M. A., El Sehiemy, R. A., Rashwan, Y. I., and Gao, X. Z. (2019). On the performance improvement of elephant herding optimization algorithm. *Knowledge-Based Systems*, 166:58–70.
- Erkol, H. O. (2018). Attitude controller optimization of four-rotor unmanned air vehicle. *International Journal of Micro Air Vehicles*, 10(1):42–49.
- Hernandez-Romero, N., Medina-Marin, J., and Seck-Tuoh-Mora, J. C. (2012). *Introducción a Matlab para Resolver Problemas de Ingeniería Aplicando Algoritmos Genéticos*. Pachuca, Hgo.
- Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimisation. *Conference on Neural Networks IEEE*, 95:1942–1948.
- Kramer, O. (2017). *Genetic Algorithm Essentials*, volume 679.
- Mirjalili, S. and Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95:51–67.
- Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69:46–61.
- Mohamed, A. W., Hadi, A. A., and Jambi, K. M. (2019). Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. *Swarm and Evolutionary Computation*, 50.
- Razinkova, A., Kang, B.-J., Cho, H.-C., and Jeon, H.-T. (2014). Constant Altitude Flight Control for Quadrotor UAVs with Dynamic Feedforward Compensation. *International Journal of Fuzzy Logic and Intelligent Systems*, 14(1):26–33.
- Seck-Tuoh-Mora, J. C., Hernandez-Romero, N., Lagos-Eulogio, P., Medina-Marin, J., and Zúñiga Peña, N. S. (2021). A continuous-state cellular automata algorithm for global optimization. *Expert Systems with Applications*, page 114930.
- Subhadip, S. (2014). Genetic Algorithm : An Approach for Optimization (Using MATLAB). *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3(3):261–267.
- Talbi, E.-G. (2009). *Metaheuristics : from design to implementation*, volume 1. Wiley & Sons, Canada.
- Tanabe, R. and Fukunaga, A. (2013). Success-history based parameter adaptation for Differential Evolution. *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, (3):71–78.
- Tanabe, R. and Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pages 1658–1665.
- Ustunkok, T. and Karakaya, M. (2019). Effect of PSO Tuned P, PD, and PID Controllers on the Stability of a Quadrotor. *1st International Informatics and Software Engineering Conference: Innovative Technologies for Digital Transformation, IISEC 2019 - Proceedings*, (4):0–5.
- Valavanis, K. and Vachtsevanos, G. (2015). *Handbook of Unmanned Aerial Vehicles*. Springer Netherlands, Dordrecht.
- Wang, G. G., Deb, S., and Coelho, L. D. S. (2015a). Elephant Herding Optimization. *Proceedings - 2015 3rd International Symposium on Computational and Business Intelligence, ISCBI 2015*, pages 1–5.
- Wang, G. G., Deb, S., and Cui, Z. (2015b). Monarch butterfly optimization. *Neural Computing and Applications*, 31(7):1995–2014.
- Ziegler, J. G. and Nichols, N. B. (1942). Optimum settings for automatic controllers. *Trans. ASME*, pages 759–768.