International Journal of Innovative Engineering Applications

Journal homepage: https://dergipark.org.tr/ijiea

ALTITUDE CONTROL OF QUADROTOR BASED ON METAHEURISTIC METHODS

Muhammed Kivanc Kurnaz[1], *Yagmur Olmez*[2], *Gonca Ozmen Koca*[*1]

[1]*Fırat University, Faculty of Technology, Department of Mechatronics Engineering, Elazig, 23200, Turkiye*
[2]*Malatya Turgut Ozal University, Faculty of Engineering and Natural Sciences, Department of Electrical and Electronics Engineering, Malatya, 44210, Turkiye*

Abstract

Original scientific paper

Quadrotor, which is used in many fields and is still a challenge to control, has a complex kinematic and dynamic system, and its flight performance depends on many variables that need to be controlled simultaneously. In this study, the effective determination of PID parameters for altitude control of quadrotors, which presents a complex control problem, has been tested comparatively with innovative metaheuristic approaches. Among the strong metaheuristic algorithms, the Crow Search Algorithm (CSA), Particle Swarm Optimization Algorithm (PSO), Golden Jackal Optimization Algorithm (GJO), and Jellyfish Search Algorithm (JSA) were comparatively analyzed for the determination of PID parameters. The parameters obtained with CSA caused the minimum steady-state error with the value of 6.9580e-04 in the closed-loop control system. A minimum overshoot was also obtained with the parameters optimized with CSA. When these results are evaluated, it can be said that CSA performs better than other altitude control algorithms, considering the quadrotor's stable and accurate positioning performance.

Keywords: *Crow search algorithm; golden jackal optimization algorithm; jellyfish search algorithm; parameter optimization; particle swarm optimization algorithm; quadrotor.*

1 Introduction

An unmanned aerial vehicle (UAV) is a motorized aircraft that can adjust its speed and direction through sensors and software methods without needing a pilot. With the development of technology, UAVs have begun to be used in areas that may pose a threat to human health, especially in military defense and operations, and in search and rescue activities during and after natural disasters.

UAVs can be classified into two main groups. These are fixed-wing and rotatıng wing UAVs. The main advantages of fixed-wing UAVs are that they are more aerodynamically efficient, can stay in the air for a long time, have a longer flight distance, and are easier to control compared to other similar aircraft. But today, the trend towards rotary-wing unmanned aerial vehicles has increased due to changing technological conditions, costs, difficult terrain conditions, and the need for fixed-wing unmanned aerial vehicles to have runways or launch systems for takeoff and landing. Another great advantage of rotary-wing unmanned aerial vehicles is that they provide the opportunity to rise from the ground. In other words, it includes vertical take-off and take-off (VTOL) features [1].

Systems with highly complicated kinematics and dynamics, such as quadrotors, are difficult to control. One of the traditional control methods, PID control, is used very often. There are various ways in which PID control parameters can be tuned. Ziegler-Nichols (ZN) and tuning methods can be easily applied [2]. However, these methods are time-consuming. Because many trials are required to determine the space where the system is stable and therefore it is difficult to provide movements of the quadrotor in the desired direction. In experimental studies, this uncertainty may cause damages and losses. Recently, metaheuristic methods have been progressively applied to find the optimum value of PID parameters [3]. Metaheuristic optimization methods are applied to problems with large solution spaces in engineering fields by studying the animals' behavior that move in flocks to meet their needs such as food and shelter. In this field, many algorithms such as Ant Colony Optimization (ACO based on the movements of ants to find food, and Particle Swarm Optimization (PSO) algorithms inspired by the movements of birds in flocks, can be given as examples [4]. It has been shown in previous studies that these metaheuristic algorithms give more efficient results in controlling the quadrotor than traditional methods.

In this study, the Crow Search Algorithm (CSA), Particle Swarm Optimization Algorithm (PSO), Golden Jackal Optimization Algorithm (GJO), and Jellyfish Search Algorithm (JSA), which are metaheuristic algorithms, are compared. The algorithms used in this study aim to analyze the performance of the proposed control framework by minimizing the error and determining the best parameter values in the solution space without sticking to the local optimum in the altitude control of the quadrotor.

*Corresponding author.

E-mail address: gonca.ozmen@gmail.com (G. Ozmen)

Received 14 October 2024; Received in revised form 06 February 2025; Accepted 24 February 2025

 Doi: https://doi.org/10.46460/ijiea.1564844

The organization of the paper can be introduced as follows: Section 2 represents the dynamic model of the PMD. Section 3 describes the CSA, PSO, GJO, and JSA algorithms for height control of the quadrotor. Experimental analysis is introduced in Section 4. Finally, Section 5 presents the conclusion of the analysis.

1.1 Related Works

Kapnopoulos et al. achieved attitude and position control of a quadrotor based on PID and Model Predictive Control (MPC) methods. Cooperative Particle Swarm Optimization (PSO) was used to determine the parameters of the control methods. In PSO, they used two different swarms for position control and attitude control. Two swarms work cooperatively to explore the search space more efficiently. By performing simulation studies on different trajectories, they compared the cooperative PSO algorithm with GA and standard PSO and showed that their method was better [5]. Hermouche et al. controlled a quadrotor with PID in their study. They used and compared meta-heuristic algorithms as; GOA, GWO, PSO, WOA, ALO, HHO, and SSA to determine PID parameters. Using four different objective functions, they obtained minimum error with SAE and STAE, minimum overshoot with WOA_STA, fastest optimization with SAE and STA, and best altitude with DA_SAE [6]. Sahrir et al. used PSO-based PID control for altitude and attitude stabilization of a quadcopter. They evaluated PSO roll, pitch, and yaw performance utilizing IAE, ISE, ITAE, and ITSE cost functions. Among these cost functions, ITSE gave a superior result in tracking. They also compared ZN-PID and PSO-PID in the presence of wind as a disturbance input and observed that PSO-PID gives better results during roll motion [7]. Belge et al. obtained a hybrid algorithm with Harris Hawk Optimization (HHO)-Grey Wolf Optimization (GWO), which is one of the metaheuristic algorithms for payload hold and release in unmanned aerial vehicles in their study. They compared HHO-GWO with PSO optimization by performing path planning. They examined the effect of changing mass on the system [8]. Gün first obtained the parameters of the PID control method by Ziegler Nicholes and the trying-error method to minimize the energy consumption of the quadrotor in his study, Then, he trained the coefficients using PSO, differential evolutionary algorithm (DE), gravity search algorithm (GSA), charged system search (CSS) algorithm. As a result of the comparison, he observed that DE-PID consumes less energy of the quadrotor compared to other algorithms [9]. Alqudsi et al. used the trajectory generation and optimization algorithm (TGO) to create an unobstructed trajectory over predetermined points. With this algorithm, they aimed to reach the waypoints on the quadrotor's trajectory in minimum time and create new trajectories. The proposed algorithm is compared with constrained quadratic programming (CQP) and unconstrained quadratic programming (UQP) [10]. Meraihi et al. conducted extensive research on the Crow Search algorithm by compiling the work developed on it and combined with different algorithms. They identified several engineering applications for CSA [11]. Sheta et al. used the PID control method to ensure the desired orientation and position of the quadrotor. They used meta-heuristic algorithms PSO, CSA, GA, and the traditional ZN method to determine the PID parameters. They evaluated the results in terms of performance criteria using a multi-objective fitness function. Among the algorithms used, they observed that the parameters obtained with PSO showed the best performance in motion control of the quadrotor [3]. Farzaneh et al. obtained a dynamic model of a quadrotor. They used a neural network as the main control method for the quadrotor system. They conducted simulation and experimental studies to test the performance of the system. The performance of the neural network model is superior to the PID model of the quadrotor [12]
.

Table 1. Summary of recent studies based on control structures developed for quadrotor control.

Authors	Method	Aim	Experimental/ Simulation	Obtained Results	Tuning Method
Sheta et al [3]	PID	Position control	Simulation	PSO-optimized parameters yield impressive control results in quadcopter control, surpassing those of GA and CSA.	PSO, GA, CSA
Kapnopoulos et al. [5]	MPC, PID	Attitude & Position control	Simulation	Trajectory tracking was successfully achieved with MPC and PID.	PSO
Hermouche et al. [6]	PID	Altitude control	Simulation	A PID model with metaheuristics for four objectives was comparatively investigated on the quadrotor.	DA, GOA, WOA, GWO, PSO, ALO, HHO, SSA
Sahrir et al. [7]	PID	Altitude & Position control	Simulation	Altitude and attitude stabilization of a quadrotor is achieved with the PID controller, whose parameters are optimized with PSO.	PSO
Belge et al. [8]	PID	Path planning	Simulation	A fast and safe path planning is provided with HHO and GWO methods.	HHO, GWO
Gün [9]	PID	Attitude control	Simulation	The PID control parameters have the best results optimized with DE	DE, PSO, GSA, CSS
Alqudsi et al. [10]	-	Trajectory tracking	Simulation	A new trajectory producer with an optimizer is created to produce adaptable and collision-free routes.	TGO
Farzaneh et al [12]	NN, PID	Attitude & Position control	Simulation/ Experimental	Optimal Neural Network Controller has an effective performance for the Stabilization of a Quadrotor.	Enumerative Optimization
Proposed study	PID	Altitude control	Simulation	A successful control method has been developed for quadrotors using a CSA-based PID, achieving a minimum steady-state error of 6.9580e-04.	CSA, PSO, GJO, and JSA

1.2 Literature Gaps

Although the quadrotor is a complex kinematic and dynamic system with six degrees of freedom (6-DOF), it is controlled by changing the speed of four rotors, i.e. propellers. The flight performance of the quadrotor depends on many variables, and many variables need to be controlled simultaneously. This makes it difficult to control such a complex system. Generally, classical control methods or modern control algorithms are used to control a quadrotor. Classical control methods are not sufficient for the targeted performance of such a complex system, and since these variables are directly effective in the movement of the aircraft, the parameters must be selected most appropriately. By using an efficient optimization method, suitable parameters can be found much more efficiently and in a short time interval in large search spaces. On the other hand, modern optimization methods may also be insufficient because of model uncertainties. In these cases, more effective results are obtained by using hybrid algorithms for such unstable systems. Thus, the performance of the quadrotor can be improved, energy efficiency can be achieved, and more precise control can be achieved.

1.3 Literature Gaps Motivation and Proposed Method

Our motivation is to realize the altitude control of the quadrotor based on meta-heuristics, which are widely used in the control of unmanned aerial vehicles. Firstly, the dynamic model of the quadrotor, which is a nonlinear system, is obtained. While obtaining the mathematical model, it is taken into account that the PMD has a diagonal structure, and the z-axis is downward based on the right-hand rule starting from the center of gravity of the quadrotor. Secondly, the PID control method was used for the height control of the quadrotor. While calculating the PID control parameters, 4 different metaheuristic algorithms were used to provide more efficient results and more stable movement of the system.

To evaluate the proposed algorithms, the performance criteria of the system such as maximum overshoot, rise time, and settling time are tested and compared. In this paper, 4 different algorithms are used for tuning the parameters of the PID for the height control of a nonlinear quadrotor with a complex structure.

2 Modelling Of Parot Mambo Drone

In this section, the mathematical modeling of the quadrotor, the reference coordinate systems to be used in the model, and the kinematic and dynamic model according to the Newton-Euler motion equations are defined by accepting some assumptions. According to the Newton-Euler motion equations, the kinematic and dynamic model is defined by accepting some assumptions below.

- The propellers and structure of the drone are rigid.
- It has a symmetrical structure.
- The thrust and drag forces continue with the square of the propeller velocity.
- The center of gravity comes across with the beginning of the body frames.
- Four propellers operate under the same conditions. The amount of thrust (b) and the amount of torque (d) are the same for all.

To obtain the mathematical model of the quadrotor, the kinematic and dynamic equations of the vehicle must be found. To obtain equations, coordinate systems must also be determined. These are the body coordinate system $x_b, y_b, z_b,$ and the fixed location coordinate system x_e, y_e, z_e. The specified axis sets are shown on the drone in Fig. 1.

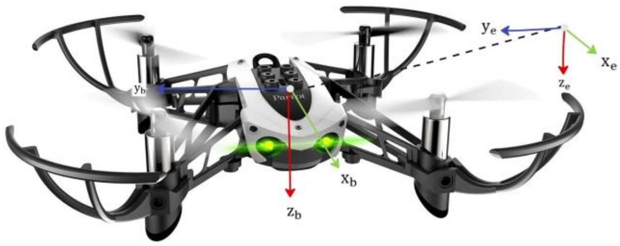

Figure 1. Parrot Mambo Drone and its coordinate system

The quadrotors, schematically shown in Figure 2, are available for two different configurations: x (cross) and + (plus). The vehicle in the plus (+) configuration has a more acrobatic configuration, while the vehicle in the x (cross) configuration is more stable. The plus (+) vehicle uses two rotors to move in the x and y directions, while the cross (x) vehicle uses four rotors. For example, in the 'cross' quadrotor, the speed of rotors 1 and 2 (3 and 4) increases (decreases) at the same time throughout the pitch action. The quadrotor used in this study has a cross (x) configuration. The direction of the rotations is clockwise for the 1st and 3rd rotors and counterclockwise for the 2nd and 4th rotors.

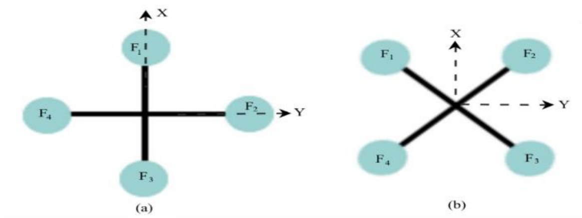

Figure 2 a ‘Plus’ configuration **b** ‘Cross’ configuration

If the relationship between two coordinate systems is expressed as r_e in the E frame, and r_b in the B frame, Equation 1 emerges.

$$r_e = R^{(E,B)} r_b \tag{1}$$

where, $R^{(E, B)}$ is the transformation matrix. Equation 2.1 expresses the conversion from B to E coordinate system. A certain order is followed to perform these transformations. First, the transformation should be made with the help of the z-axis yaw (ψ) angle, then with the help of the y-axis pitch (θ) angle, and then with the help of the x-axis roll (φ) angle. As a result of the transformation, Equation 2 emerges.

$$R=\begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2}$$

Euler angles; roll (ϕ), pitch (θ), and yaw (ψ) compose the orientation of the quadrotor. If a rigid body is not in motion, that is, it is stationary, its inertia is its mass but if it is in motion, its moment of inertia arises against the change in rotation movements around itself. The angular moment acting on the body coordinate system is obtained by Eq. 3.

$$M_B = J\dot{\omega} + \omega x J\omega \tag{3}$$

Where J is the diagonal inertia matrix of the drone, ω is the angular speed vector, and M_B is the moment affecting the body axis assembly. If the moment M_B acting on the body axis set is denoted by L, M, N and the rotation moment on the x_b, y_b, and z_b axes, respectively.

$$M_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} l_{xx} & 0 & 0 \\ 0 & l_{yy} & 0 \\ 0 & 0 & l_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} x \begin{bmatrix} l_{xx} & 0 & 0 \\ 0 & l_{yy} & 0 \\ 0 & 0 & l_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$
(4)

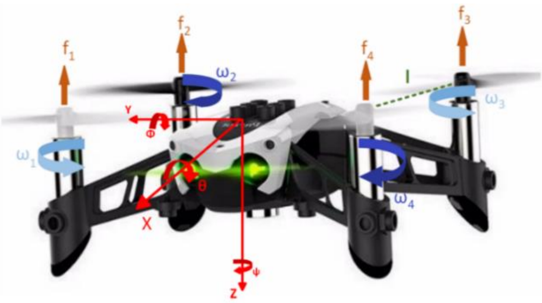

Figure 3. Rotor rotation directions and distances on the quadrotor.

If the moment on the x_b axis is obtained concerning the gravity center of the quadrotor, the moment on the y_b axis is acquired concerning the gravity center of the drone. The moment on the z_b axis is calculated from the rotor torque, the following equations are obtained.

$$\begin{aligned} L_{(roll)} &= l(b\Omega_2^2 - b\Omega_4^2) \\ M_{(pitch)} &= l(b\Omega_3^2 - b\Omega_1^2) \\ N_{(yaw)} &= d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{aligned} \tag{5}$$

Here, Ω is the angular speed of the rotor, b is the aerodynamic force constant, l is the distance between the center of mass and the rotors, and d is the aerodynamic moment constant. Based on Equations 4 and 5, the translation and rotation equations are obtained as follows.

$$\begin{aligned} \ddot{\phi} &= \frac{(l_{yy} - l_{zz})}{l_{xx}} \dot{\psi}\dot{\theta} - \frac{J_r}{l_{xx}}(\Omega_r)\dot{\theta} + \frac{l}{l_{xx}} U_2 \\ \ddot{\theta} &= \frac{(l_{zz} - l_{xx})}{l_{yy}} \dot{\psi}\dot{\phi} + \frac{J_r}{l_{yy}}(\Omega_r)\dot{\phi} + \frac{l}{l_{yy}} U_3 \\ \ddot{\psi} &= \frac{(l_{xx} - l_{yy})}{l_{zz}} \dot{\theta}\dot{\phi} + \frac{l}{l_{zz}} U_4 \end{aligned} \tag{6}$$

Each rotor speed is obtained by Equation 6.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & bl & 0 & -bl \\ -bl & 0 & bl & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{8}$$

3 Control of Parrot Mambo DRONE

PID (Proportional-Integral-Derivate) control is a widely used control method for the control of aircraft. PID provides the appropriate control signal (u(t)) for a closed-loop control system by passing the error through proportional, integral, and derivative components.

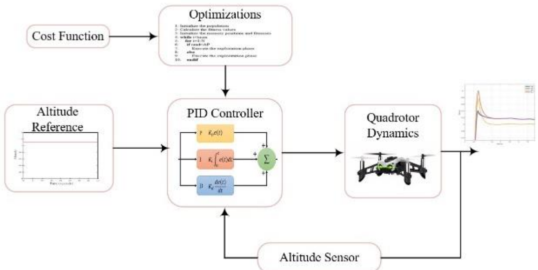

Figure 4. Altitude control of the quadrotor with optimal PID-tuned algorithms.

The control signal of the PID is given as Eq. (9).

$$u(t) = K_p e_z + K_i \int e_z dt + K_d \frac{de_z}{dt} \tag{9}$$

where u(t) is the control output signal. k_p, k_i, and k_d are proportional, integral, and derivative gains, respectively. e(t) is a closed-loop error.

PID control is highly preferred in linear systems as well as in nonlinear systems [13]. This method can be adapted to many systems and is a control method used in industrial applications and automatic control applications since it contains fewer variables compared to other methods. There are many methods to determine PID control parameters. They can be adjusted by using various methods such as trial and error, Ziegler-Nichols, Cohen-Coon, or metaheuristic optimization algorithms.

3.1 Crow Search Algorithm

The crow search algorithm was created by modeling some intelligent behaviors of crows. It was presented by Azkarzadeh in 2016 as a nature-inspired method [14]. Unlike other birds, crows can remember where they hide their food and can obtain more food by following other crows. In this algorithm, crows have two different movement strategies. These are: (1) protecting the location of their food and (2) learning where other crows hide their food. Based on this scenario, the motion equations of crows are created as given in Eqs. (10) - (11). At the t^{th} iteration, j-th crow is required to go to its hiding position and i-th crow agrees to follow j-crow. In the first scenario, if j^{th} crow is not hip to that the i^{th} crow is following him then i-th crow changes its position according to Eq. (10).

$$x_i(t+1) = x_i(t) + rand_i \, x \, fl_i(t) \, x \, (m_j(t) - x_i(t+1)) \tag{10}$$

where x_i denotes the position of i-th crow. fl and m_j represent the flight length and the hiding place of the j-th crow. In the second scenario, j-th crow is aware that i-th crow is following it and updates the next positions by using the following equation:

$$x_i(t+1) = lb + rand * (ub - lb) \quad (11)$$

Algorithm 1. Pseudo code of the Crow Search Algorithm

1. Define fl, AP, N, d, and maxIt.
2. Initialize X and Mem
3. Calculate fitness values of the crows
4. Determine the best and worst ones
5. **while** t < maxIter
    a. **for** i=1:N
        i. Select j-th crow, randomly.
        ii. **if** rand < AP
            1. Generate Xnew with Eq. (10)
        iii. Else
            1. Generate Xnew with Eq. (11)
        iv. Endif
        v. Calculate fitness value of Xnew as f(Xnew)
        vi. if f(Xnew(i)) < f(X(i))
            1. Update X(i) with Xnew(i)
        vii. Endif
    b. Endfor
6. **Endwhile**

** X: Initial population, fl: flight length, AP: awareness probability, N: number of the crows, d: dimension, maxIt: maximum number of the iteration, Mem: crows' memories set, Xnew: new solution.

3.2 Particle Swarm Optimization Algorithm

Particle Swarm Optimization is one of the oldest and most widely used optimization methods. Kennedy developed the algorithm by modeling the animals' behaviors in the flock [15]. In PSO, the positions of the particles are updated based on global and local optimal solutions. The following equations are presented for the updating of the particles' velocities and positions:

$$velocity(t+1) = velocity(t) + r_1\, x\, c_1\, x(p_{best} - x(t)) + r_2\, x\, c_2\, x\,(g_{best} - x(t)) \quad (12)$$

$$x_i(t+1) = x_i(t) + velocity(t+1) \quad (13)$$

where, $velocity$ and x represent velocities and the positions of the particles, respectively. c_1 and c_2 denote the acceleration parameters. P_{best} and g_{best} are the local best and global best solutions in the particles' population. r_1 and r_2 show two random values. The pseudo-code of the PSO is presented in Algorithm 2.

Algorithm 2. Pseudo code of the Particle Swarm Optimization

1. Define v and x
2. Initialize N, d, and maxIt
3. Calculate the fitness values of the particles
4. Assign x as p_{best}
5. Determine g_{best}
6. **while** t < maxIter
    a. Update x and v of the particles with Eq. (12-13)
    b. Calculate fitness values of the particles
    c. **for** i=1:N
        i. if f(Xnew(i)) < f(p_{best}(i))
            1. Update p_{best}(i)
        ii. Endif
    d. Endfor
    e. Determine g_{best}
7. **Endwhile**

** v: velocities of the particles, x: positions of the particles, N: number of the crows, d: dimension, maxIt: maximum number of the iteration, P_{best}: local best solution, and g_{best}: global best solution.

3.3 Golden Jackal Optimization Algorithm

Golden Jackal Optimization is presented by Chopra and Ansari in 2022 [16]. A newly introduced nature-inspired metaheuristic algorithm is developed by modeling the habits of the jackals during hunting. The exploration phase is the searching for the prey stage and it is led by the male jackal. The male jackal finds the prey and the female jackal follows him. In this stage, positions of the male and female jackals are obtained by using Eqs. (14-15).

$$x_m(t+1) = x_m(t) - E.\left|x_m(t) - rl.p(t)\right| \quad (14)$$

$$x_{fm}(t+1) = x_f(t) - E.\left|x_f(t) - rl.p(t)\right| \quad (15)$$

The positions of the preys are updated by using the following function (Eq. (16)):

$$p(t+1) = \frac{x_m(t+1) + x_{fm}(t+1)}{2} \quad (16)$$

In the exploitation phase, the jackals harass the prey and decrease their evading energy of them. The prey with decreased energy is surrounded and is easily hunted by the jackals. The behavior of the jackals is modeled by the following equations:

$$x_m(t+1) = x_m(t) - E.\left|rl.x_m(t) - p(t)\right| \quad (17)$$

$$x_{fm}(t+1) = x_f(t) - E.\left|rl.x_f(t) - p(t)\right| \quad (18)$$

Where rl represents a vector based on the Levy function. The evading energy of the prey is calculated as: $E = E_0 + E_1$. Where, E_0 and E_1 represent the initial and the decreasing energy of the prey, respectively. The initial and the decreasing energy is calculated as: $E_0 = 2\, x\, r - 1$ and $E_1 = c\, x\,(1 - it/maxIt)$. Where, r and c denote the random and a constant number, respectively.

Algorithm 3. Pseudo code of the Golden Jackal Optimization Algorithm

1. Define N, d, maxIt
2. Initialize P
3. **while** t < maxIter
    a. Calculate fitness values of the preys
    b. Assign the best and the second best preys as X_m and X_{fm}
    c. **for** i=1:N
        i. Update E
        ii. Update rl
        iii. **if** E < 1
            1. Update X with Eq. (17)
        iv. **Else**
            1. Update X with Eq. (18)
        v. **endif**
        vi. Calculate fitness values of the X population
    d. **endfor**
4. **endwhile**

** P: Prey population, X_1: Male jackal's position, X_1: Female jackal's position, E: Evading energy, rl: Vector based on Levy movement function,

3.4 Jellyfish Search Algorithm

The Jellyfish Search Algorithm (JSA) is one of the population-based methods which is introduced by Chou and Truong in 2021 [17]. The method is created by modeling the behavior of the jellyfish population. JSA consists of two stages: exploitation and exploration. In the exploration stage, the jellyfish society follows the ocean current to find plenty of nutrients. This movement is realized by using Eq. (19):

$$x_i(t+1) = x_i(t) + r_1 \; x \; ocr \tag{19}$$

$$ocr = x^* - df \tag{20}$$

$$df = \beta \; x \; r_2 \; x \; \mu \tag{21}$$

Where ocr represents the ocean current. r_1 and r_2 are the random values in the range of (0, 1). β and μ denote the distribution coefficient and the mean of the jellyfish population, respectively. In the exploitation stage, the jellyfish search the space by using two different movement strategies, which are called passive (Type A) and active (Type B) movement strategies. In the Type A, the jellyfish move around their positions. The equation of the Type A is realized with the following equation:

$$x_i(t+1) = x_i(t) + \Upsilon \; x \; r_3 \; x \; (ub - lb) \tag{22}$$

Where, ub and lb are upper and lower bounds, respectively. Υ is a motion constant and must be bigger than zero. r_3 is a random value. While forming jellyfish blooms, they aim to move to places where there is plenty of food. This motion is realized in Type B motion strategy and it is modeled mathematically as given in Eq. (23).

$$x_i(t+1) = x_i(t) + Step \tag{23}$$

$$step = r \; x \; D \tag{24}$$

$$D = \begin{cases} x_i - x_j & if \; f(x_i) \geq f(x_j) \\ x_j - x_i & if \; f(x_i) < f(x_j) \end{cases} \tag{25}$$

Where, D and Step represent the direction vector and the length of the motion, respectively. In this algorithm, a time control strategy is used to select the motion type. Eq. (26) introduces the time control strategy:

$$c = 2 \; x \; r \; x \left(1 - \frac{t}{T}\right) - 1 \tag{26}$$

Where t and T are current and the maximum number of the iterations. C is the time control function.

Algorithm 4. Pseudo code of the Jellyfish Search Algorithm

1. Define N, d, and maxIt
2. Initialize X
3. Calculate the fitness values of the
4. **while** t < maxIter
    a. Determine X^*
    b. **for** i=1:N
        i. Calculate c(t) with Eq. (26)
        ii. **if** c(t) ≥ 0.5
            1. Calculate ocr with Eq. (20)
            2. Update X with Eq. (18)
        iii. **Else**
            1. **if** rand < c(t)
                a. Update X with Eq. (22)
            2. **Else**
                a. Update X with Eq. (23)
            3. **Endif**
        iv. **Endif**
        v. Calculate fitness values of the X population
    c. **Endfor**
5. **Endwhile**

X: Jellyfish population, X^*: Best jellyfish, c(t): Time control parameter, ocr: ocean current function

4 Experimental Studies and Results

In this section, the results of parameter tuning with CSA, PSO, GJO, and JSA algorithms for altitude control of the quadrotor using PID control and various analyses calculated as a result of these algorithms are presented. The quadrotor used in this study is the Mambo Mini Drone (PMD) produced by Parrot. There are 4 sensors on the quadrotor. These are the inertial measurement unit (IMU), pressure sensor, ultrasonic sensor, and camera.

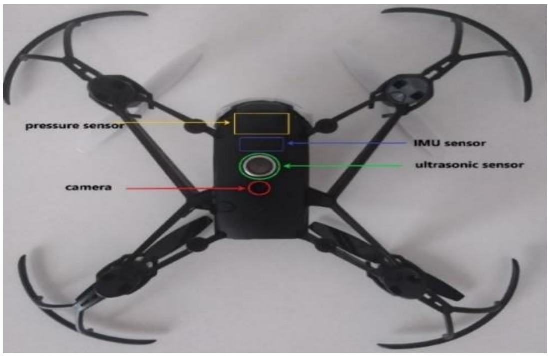

Figure 5. Illustration of sensors on the quadrotor [18].

Table 2 presents the physical parameters of the PMD. These parameters were used to create the mathematical model and control model of the quadrotor.

Table 2. Physical parameters of Parrot Mambo Drone.

Description	Parameter	Unit	Value
Mass	m	kg	0.063
Length of an arm	l	m	0.0624
Drag coefficient	d	Nms^2	78.26e-5
Thrust coefficient	b	Ns^2	0.0107
Inertia Moment along x-axis	l_{xx}	kgm^2	5.82e-5
Inertia Moment along y-axis	l_{yy}	kgm^2	7.16e-5
Inertia Moment along z-axis	l_{zz}	kgm^2	0.0001
Rotor Moment of Inertia	J_r	kgm^2	0.1021e-6

In the experiments conducted to optimize the PID control parameters, by obtaining with CSA, PSO, GJO, and JSA were used as shown in Table 3.

Table 3. Optimization parameters of the CSA, PSO, GJO, and JSA.

Algorithm	Parameter	Symbol	Value
CSA	Number of dimensions	pd	3
	Number of populations	N	50
	Number of iterations	T	250
	Awareness probability	AP	0.1
	Flight length	fl	2
	Lower bound	u	0
	Upper bound	l	100
PSO	Number of dimensions	b	3
	Number of populations	n	50
	Number of iterations	k	250
	Cognitive parameter	c_1	0.12
	Social parameter	c_2	1.2
	Lower bound	down	0
	Upper bound	up	100
GJO	Number of dimensions	dim	3
	Number of populations	n	50
	Number of iterations	iter	250
	Lower bound	lb	0
	Upper bound	ub	100
JSA	Number of dimensions	nd	3
	Number of populations	nPop	50
	Number of iterations	MaxIt	250
	Lower bound	Lb	0
	Upper bound	Ub	100

The PID control parameter values Kp, Ki, and Kd obtained by CSA, PSO, GJO, and JSA for the input shown as thrust (z) to the system as the height control of the quadrotor are presented in Table 4.

Table 4. Optimal PID parameters.

Algorithms	K_p	K_i	K_d
CSA	99.4685	10.0842	16.7308
PSO	89.3500	10.1010	13.7640
GJO	91.2160	0.0676	13.9287
JSA	99.6977	10.1872	16.6593

Figure 5 presents the errors of the quadrotor because of the altitude control of each meta-heuristic algorithm. When we closely examine the peak values of the error values, the PSO algorithm reached the peak error value of 0.1 in 0.0996 seconds. PSO algorithm was followed by GJO which reached 0.074 peak error value in 0.076 seconds, JSA which reached 0.041 peak error value in 0.04096 seconds, and CSA which reached 0.0375 peak error value in 0.03652 seconds. Considering these results, the peak error value of the CSA algorithm was lower than the other algorithms.

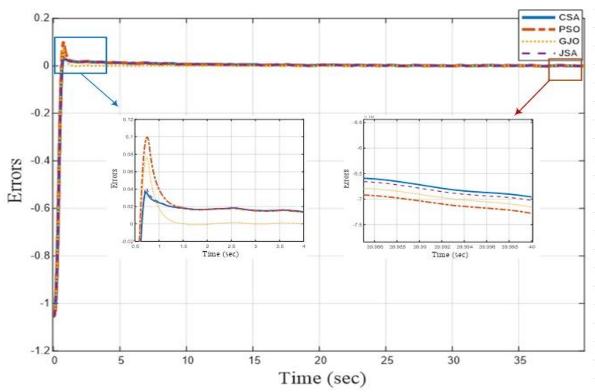

Figure 6. Errors for altitude control of quadrotor.

The overshoot, rise, and settling times of a quadrotor because of PID height control parameters optimized with CSA, PSO, GJO, JSA algorithms-based PIDs, and classic PID are shown in Table 5. Each value here refers to the effectiveness of the quadrotor in controlling the height movement. The overshoot value achieved using the PSO algorithm is notably higher than that of the other algorithms, particularly the CSA and JSA algorithms. The rising time is roughly similar across all algorithms. The GJO algorithm produced the best settling time, while the settling times for the CSA, PSO, and JSA algorithms are nearly identical. Furthermore, the classic PID controller demonstrated poorer performance regarding overshoot, rise time, and settling time.

Table 5. Performance criteria for each algorithm.

Algorithms	Overshoot (%)	Rising Time (s)	Settling Time (s)	Error
CSA	3.1439	0.3976	1.1207	6.9580e-04
PSO	8.8823	0.3824	1.2677	7.2764e-04
GJO	6.7353	0.3875	0.9760	7.1542e-04
JSA	3.5585	0.3963	1.1320	7.0231e-04
Classic PID	12.5471	1.2513	11.3126	7.0162e-04

In this respect, it is shown that the desired altitude control of the quadrotor can be achieved with meta-heuristic algorithms. These results prove the suitability of the optimization approach of a highly nonlinear system with a complex structure such as a quadrotor by using meta-heuristic algorithms in this study.

To further analyze the performances of the algorithms, the convergence curves of CSA, PSO, GJO, and JSA are compared in Figure 7. Since the adjustment of the parameters of the control methods is a minimization problem, the point where the cost function value is minimum is seen as the point where the optimal parameters are. This means the quadrotor can reach the given reference altitude with the highest accuracy. Convergence curves up to 250 iterations were created for the PID altitude control of the quadrotor using CSA, PSO, GJO, and JSA-based approaches. Figure 7 shows that JSA and CSA produce competitive results. In other words, these algorithms converge to the optimum point the fastest among the considering methods. In addition, JSA and CSA methods have competitive results.

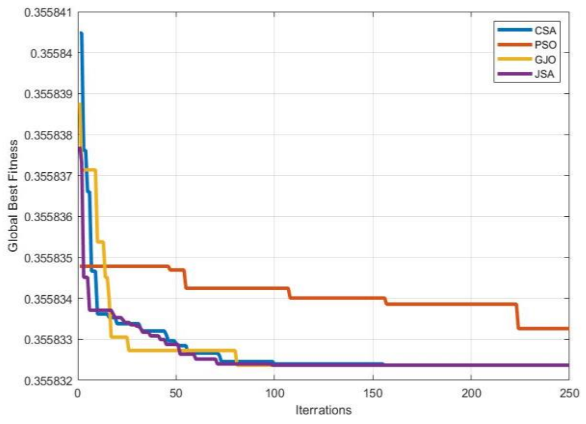

Figure 7. Convergence curves of CSA, PSO, GJO, and JSA.

It is also obviously seen that PSO has a much larger error compared to CSA, GJO, and JSA. The proposed control approach based on CSA, PSO, GJO, and JSA provides adequate performance for the altitude control of the quadrotor for these methods, as shown in the convergence curves in Figure 8.

Examining the convergence behavior of the algorithms to the optimal point is quite effective in the qualitative evaluation of the algorithms. For this purpose, when the convergence curves of the considered methods are examined, it can be concluded that the JSA and CSA methods are qualitatively more adequate than the other algorithms for this problem. The altitude response of the quadrotor is represented in Fig. 8. Previous results in Fig. 6., Fig. 7 and Table 5 prove that the superior control performance is provided by using CSA to optimize the control parameters of PID.

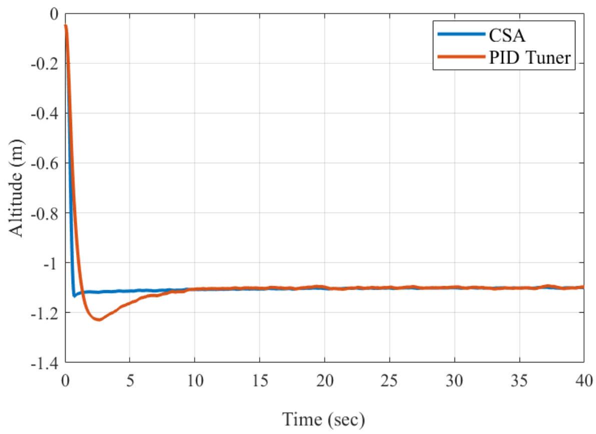

Figure 8. Altitude response of the quadrotor.

The altitude response in Fig. 8. shows that the CSA-optimized PID produces a faster and more stable output than the tuner-optimized PID. The Euler angles of the quadrotor are also given in Fig.9.

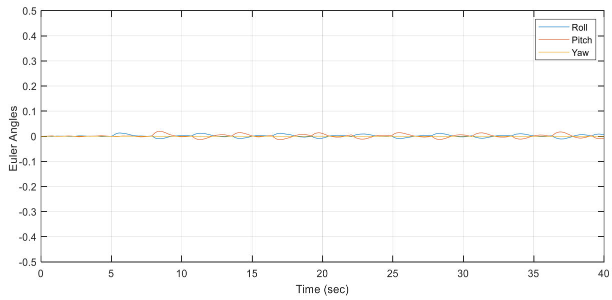

Figure 9. Euler angles of the quadrotor.

The drone will be more stationary during takeoff the closer the roll, pitch, and yaw angle oscillation and error rate are near zero. The references of these three angles roll, pitch, and yaw are therefore regarded as zero.

5 Conclusion

The quadrotor, a type of unmanned aerial vehicle with a complex system, can be effectively controlled using optimization methods, even though it is often challenging to manage with conventional techniques. This characteristic makes it a popular choice for evaluating performance in various engineering problem-solving scenarios.

This study provides an overview of four optimization algorithms: the Crow Search Algorithm (CSA), Particle Swarm Optimization Algorithm (PSO), Golden Jackal Optimization Algorithm (GJO), and Jellyfish Search Algorithm (JSA). These algorithms are utilized to fine-tune the dynamic equations of a quadrotor model and the PID control parameters used for altitude control. A comparison of each metaheuristic algorithm used in controlling the system is presented. In this context, the results obtained from the error performance of each algorithm are analyzed and evaluated. The CSA algorithm achieved the target altitude with the smallest error compared to the others. Following the CSA algorithm in performance are the JSA, GJO, and PSO algorithms, respectively. The convergence curves of these algorithms provide a clearer representation of their control processes. To better analyze the performance of the quadrotor, we examined the maximum overshoot, rise time, and settling time. Among the algorithms tested, the CSA demonstrated the lowest maximum overshoot, measuring at 3.1499, followed by the JSA, GJO, and PSO algorithms in that order. Upon analyzing the rising times, it becomes evident that the results demonstrate a striking similarity. It is seen that the PSO has reached the reference point in a shorter time, followed by GJO, JSA, and finally CSA. Based on the analyses conducted, the PID parameters for altitude control of the quadrotor were optimized using various metaheuristic algorithms, and comparative studies were carried out.

The results obtained using the PID tuner served as the reference point. Based on the overshoot metric, the improvement percentages for each method were as follows: the CSA method showed an improvement of 74.94%, the PSO method improved by 29.21%, the GJO method increased by 46.32%, and the JSA method achieved a 71.64% increase. In terms of rising time, the improvement percentages were as follows: the CSA method showed a 68.23% improvement, the PSO method had a 69.44% increase, the GJO method improved by 69.03%, and the JSA method reported a 68.33% enhancement. Regarding settling time, the improvement percentages were as follows: the CSA method showed a 90.09% improvement, the PSO method increased by 88.79%, the GJO method improved by 91.37%, and the JSA method reported an enhancement of 89.99%.

Future research will focus on position control and trajectory tracking analyses of the quadrotor. Additionally, hybrid metaheuristic algorithms will be employed to achieve more stable control.

Declaration

This study is derived from the thesis titled “Realization of image processing-based trajectory tracking algorithm on Parrot Mambo drone with MATLAB” of the first author (Muhammed Kivanc Kurnaz).
This study was supported by the Scientific Research Projects unit of the Firat University (FUBAP) under the grant number TEKF.21.12. The authors thank the FUBAP for their support.

References

[1] Okasha, M.; Kralev, J.; Islam, M. (2022). Design and Experimental Comparison of PID, LQR, and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone, *Aerospace*, Vol. 9, No. 6, 298.

[2] Khodja, M. A.; Tadjine, M.; Boucherit, M. S.; Benzaoui, M. (2017). Experimental dynamics identification and control of a quadcopter, *2017 6th International Conference on Systems and Control, ICSC 2017*, IEEE, 498–502.

[3] Sheta, A.; Braik, M.; Maddi, D. R.; Mahdy, A.; Aljahdali, S.; Turabieh, H. (2021). Optimization of PID controller to stabilize quadcopter movements using meta-heuristic search algorithms, *Applied Sciences (Switzerland)*, Vol. 11, No. 14.

[4] Derrouaoui, S. H.; Bouzid, Y.; Guiatni, M. (2021). PSO Based Optimal Gain Scheduling Backstepping Flight Controller Design for a Transformable Quadrotor, *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 102, No. 3.

[5] Kapnopoulos, A.; Alexandridis, A. (2022). A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme, *Aerospace Science and Technology*, Vol. 127, 107725.

[6] Hermouche, B.; Zennir, Y.; Kamsu Foguem, B. (2023). Influence of meta-heuristic algorithms on the optimization of quadrotor altitude PID controller, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 45, No. 10, 1–19.

[7] Sahrir, N. H.; Basri, M. A. M. (2023). PSO–PID Controller for Quadcopter UAV: Index Performance Comparison, *Arabian Journal for Science and Engineering*, Vol. 48, No. 11, 15241–15255.

[8] Belge, E.; Altan, A.; Hacıoğlu, R. (2022). Metaheuristic Optimization-Based Path Planning and Tracking of Quadcopter for Payload Hold-Release Mission, *Electronics (Switzerland)*, Vol. 11, No. 8.

[9] Gün, A. (2023). Attitude control of a quadrotor using PID controller based on differential evolution algorithm, *Expert Systems with Applications*, Vol. 229, No. PB, 120518.

[10] Alqudsi, Y.; Makaraci, M.; Kassem, A.; El-Bayoumi, G. (2023). A numerically-stable trajectory generation and optimization algorithm for autonomous quadrotor UAVs, *Robotics and Autonomous Systems*, Vol. 170, No. January 2021, 104532.

[11] Meraihi, Y.; Gabis, A. B.; Ramdane-Cherif, A.; Acheli, D. (2021). A comprehensive survey of Crow Search Algorithm and its applications, *Artificial Intelligence Review*, Vol. 54, No. 4, 2669–2716.

[12]Farzaneh, M. M.; Tavakolpour-Saleh, A. R. (2022). Stabilization of a quadrotor system using an optimal neural network controller, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 44, No. 1, 1–12.

[13]Chang, W.-D.; Shih, S.-P. (2010). PID controller design of nonlinear systems using an improved particle swarm optimization approach, *Communications in Nonlinear Science and Numerical Simulation*, Vol. 15, No. 11, 3632–3639.

[14]Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures*, *169*, 1-12.

[15]Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). ieee.

[16]Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications*, *198*, 116924.

[17]Chou, J. S., & Truong, D. N. (2021). A novel metaheuristic optimizer inspired by the behavior of jellyfish in the ocean. *Applied Mathematics and Computation*, *389*, 125535.

[18]Kurnaz, M. K. (2023). Realization of image processing-based trajectory tracking algorithm on Parrot Mambo drone with MATLAB. Master's Thesis, https://tez.yok.gov.tr/.