



A Comparative Study on the Tuning of the PID Flight Controllers Using Swarm Intelligence

Aziz Kaba^{1*}

¹ Eskisehir Technical University – Faculty of Aeronautics and Astronautics,
azizkaba@eskisehir.edu.tr -  0000-0003-0453-2912



Abstract

QUAVs have some shortcomings in terms of nonlinearities, coupled dynamics, unstable open-loop characteristics, and they are prone to internal and external disturbances. Therefore, control problem of the QUAVs is still an open issue. Designed controllers based on the linear dynamics have limited operating ranges. Therefore, nonlinear dynamics of the QUAVs must be derived and used in the control problem. Although some advanced controllers are presented for QUAV control, PID controllers are the most employed, well-known controllers with the simple structure, ease of implementation, solid functionality and robustness amongst the variations up to a degree. In this paper, PID based controllers are proposed for the nonlinear attitude dynamics to overcome the control problem of the QUAVs. However, since optimality and tuning of the PID controllers are fuzzy because of trial and error approaches, swarm intelligence based meta-heuristic algorithms (ABC, ACO and PSO) are employed to optimize the PID coefficients. Results are compared in terms of transient analysis and MC analysis to cover the rise time, settling time, percentage overshoot, steady-state error for the former and stochastic fitness evaluation for the latter, respectively.

Keywords

UAV
 Quadrotor
 PID Control

Time Scale of Article

Received 21 December 2020
 Revised until 25 December 2020
 Accepted 27 December 2020
 Online date 30 December 2020

1. Introduction

1.1. Background

PID controllers can treat both the transient and steady-state responses of a plant. PID can be considered as the simplest yet the most efficient controller. As the name implies, it has just a three-parameter configuration space to fit the control specifications. First applications of the PID controllers date back to the beginning of the 20th century, and they gain more popularity with the proposition of the ZN tuning method [1]. Although many novel and advanced controller designs have being proposed, PID controllers are still the most widely-used control method in industrial systems [2]. There are many advantages of the PID controllers such as low-implementation efforts, wide operating range, simple

structural design and robustness against disturbance sources [3, 4]. PID and its variants (PI, PD, etc.) are being employed as the standard controller at the lowest level of the process controllers and at the higher level of the engineering areas with over 90% ratio [5, 6].

1.2. Related Works

There are so much research efforts that went into the modern extensions of the PID controllers. Among them, some recent high-level implementations of the PID controllers can be given as follows. PID control design for an inverted pendulum is given in [7]. A fuzzy self-tuning PID algorithm for three-dimensional bio-printing temperature control system is proposed in [8]. An adaptive PID control algorithm for the second order nonlinear systems is derived in [9]. Extremum seeking nonlinear PID based pressure control algorithm is given

*: Corresponding Author Aziz Kaba, azizkaba@eskisehir.edu.tr Tel: +90 222 321 35 50 #6872
 DOI: [10.23890/IJAST.vm01is02.0205](https://doi.org/10.23890/IJAST.vm01is02.0205)

in [10]. Fuzzy PID attitude control of a vehicle is considered in [11]. An adaptive PID controller is presented in [12] for controlling speed of a brushless DC motor. Another speed control application is proposed in [13] using intelligent PID control with applications to an ultrasonic motor. Z-axis position control of a servo system for laser processing is achieved via fuzzy PID control in [14]. PID control of a flexible manipulator is presented in [15] with an opposition based spiral dynamic method. A double fuzzy RBF-NN based PID control for 7-dof manipulator is proposed in [16].

Extensions and improvements of the PID controllers are also applied as the flight controllers because of aforementioned many advantages. A data driven PID controller implemented on a FPGA with applications to UAV is detailed in [17]. According to the work, the presented controllers can control the new generation of intelligent UAVs that can perform their assigned tasks with no human intervention. A fused PID control strategy is presented in [18] for a tilt-rotor VTOL aircraft. There are two different PID controllers that comprise fixed-wing and rotary-wing parts according to the mode of flight. According to the results, the proposed fused PID controllers make a smooth transition between the flight modes. A simple adaptive PID based fault-tolerant flight controller is proposed in [19]. The method is validated with a numerical example and a flight test. According to the results, performance of the system can be improved compared to the classical PID controllers. A PID speed controller is proposed in [20] for a small-scale turbojet engine where a modification is added to the classical controller scheme. A low-pass filter is added to the differential term to reduce the noise in case of high frequencies. Results show that the controller is effective for steady-state loading changes. An enhanced PID controller for fault tolerant control of a quadrotor is proposed in [21]. The controller is tested against the actuator faults of the quadrotor. Enhanced PD structure is based on the saturation of the integral term to overcome the anti wind-up. An attitude controller based on PD and KF is proposed in [22] for a quadrotor. The measurement and modelling errors are eliminated by KF and system states are controlled by the PD controller, respectively. Results show that the proposed method overcome the disturbances with a small residual error rate.

PID and its variant controllers are prone to some shortcoming such as parameter tuning and uncertainty about it [23]. There are some works devoted to the tuning of the PID controllers. A PSO based PD flight control system is proposed in [24] for an aircraft. The results of the method are compared with that of P, PD, PI, and fuzzy controller. Analysis show that with the proposed method much better results can be obtained compared to classical approaches. Another application of PSO optimized PID flight controller is given in [25] for altitude control of a quadrotor. Better altitude control responses are achieved with implementing the PSO based PID controller. An improved BP based NN PID

control is presented in [26] with application to flight tracking control of a UAV. Both the BP based NN and the PID controller are optimized with GA to obtain the ideal parameters. Results show that the proposed method handled the attitude tracking control with robustness. A DE based PID control is presented in [27] for hover position of a quadrotor. A hybrid performance index is proposed in the paper where the proposed method improved the performance index with faster rise time and minimum overshoot, respectively.

Based on the above discussion, it can easily be said that the PID controllers have being employed for over decades in process control and they have being improved with the developing technology. Applications of PID controllers are also widely employed in aircraft and UAV control problems. However, because of complexity of the flight dynamics such as coupled translational and rotational motions on quadrotors, it is getting harder to tune the PID parameters by trial-and-error approaches. There are some works devoted to this area. So, in this work, a comparative study that is based on the swarm intelligence methods, namely particle, bee and ant swarms, is conducted. Optimality, performance and analyses are validated for a nonlinear quadrotor UAV model.

Organization of the paper as follows. The mathematical model of the quadrotor is derived in Section 2. The swarm intelligence and implemented algorithms (ant colony optimization, artificial bee colony and particle swarm optimization) are given in Section 3. PID controller structure is given in Section 4. Results and Discussions are covered in Section 5. Last, conclusions are given in Section 6.

2. Mathematical Model

Quadrotor UAVs have being employed in a wide range of applications from search and rescue, surveillance, aerial photography, to climate forecasting by military, industry, and also hobbyists, respectively. QUAUV comprises four rotors that are placed on the corners of the rigid cross-type frame as seen in Figure 1. where $\{F_i | i = 1,2,3,4\}$ is the set of net forces that are produced by the each rotor, B is the body-fixed frame, E is the earth-fixed (inertial) frame, $\{x_b, y_b, z_b\} \in \mathbb{R}^3$ are the axis elements of the B, $\{x_i, y_i, z_i\} \in \mathbb{R}^3$ are the axis elements of the E, $\{\theta, \phi, \psi\}$ is the set Euler angles defined in E, and lastly mg is the weight of the QUAUV, respectively.

The curved arrows show the directions of the rotors. Two rotors are rotated at clockwise direction while others rotate at counter-clockwise direction, respectively. In a balanced flight, the rotor pairs rotate at the same speed. Since QUAUVs do not have any servo based flight surface controllers, both translational and rotational motions are created with the difference between the rotors. However, it is worth to mention that since QUAUVs are under-actuated vehicles because of inequality between inputs and outputs of the MIMO

system, translational motions of the $\{X, Y\}$ states are achieved through employing the attitude angles. Roll angle ϕ is generated by the difference of rpm between the 2nd and 4th rotors, pitch angle θ is generated by the difference of rpm between the 1st and 3rd rotors, and lastly yaw angle ψ is generated by the difference of rpm between the rotor pairs, respectively. The translational motion set $\{X, Y, Z\}$ is in $[m]$ and attitude angles $\{\theta, \phi, \psi\}$ are in $[rad]$.

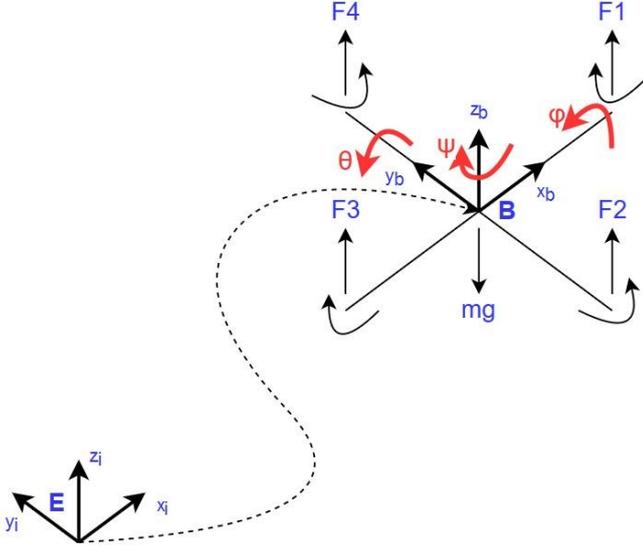


Fig. 1: Coordinates and forces acted on a QUAV

In order to transform a set of vector v defined in one reference frame ($v \in \mathbb{R}^3$) to another reference frame ($\hat{v} \in \mathbb{R}^3$), three sequential rotation is needed. A transformation between \mathbf{B} and \mathbf{E} frames via (ψ, θ, ϕ) rotation sequence can be obtained by Eqs. (1-3).

$$R(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$R(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2)$$

$$R(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2)$$

$$R(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad (3)$$

where R is the rotation. By multiplying the orthogonal rotation matrices given in equations. (1-3), the final rotation matrix is obtained in Eq. (4) as follows.

$$\mathbf{R} = \begin{bmatrix} c\theta c\psi & s\theta s\phi c\psi - s\psi c\phi & s\theta c\phi c\psi + s\psi s\phi \\ c\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (4)$$

where c is the abbreviation of cosine and s is the abbreviation of sine, respectively. The multiplication of

orthogonal matrices is also an orthogonal matrix and reverse transformation can easily be obtained with the following equation.

$$\mathbf{R}_i = \mathbf{R}^{-1} = \mathbf{R}^t \quad (5)$$

where subscript \mathbf{R}_i denotes the inverse rotation, \mathbf{t} is the transpose operator and (-1) is the inverse operation defined in matrices. Since body rates $\{P, Q, R\}$ are measured at \mathbf{B} but Euler rates $\{\dot{\phi}, \dot{\theta}, \dot{\psi}\}$ are defined in \mathbf{E} , a coordinate transformation is needed and can be obtained by using Eq. (6) and reverse transformation can also be achieved with Eq. (7).

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi \tan\theta & c\phi \tan\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi \sec\theta & c\phi \sec\theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (7)$$

The angular acceleration equations can be obtained through the moment equation and given in Eq. (8).

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} \frac{(I_y - I_z)}{I_x} QR \\ \frac{(I_z - I_x)}{I_y} RP \\ \frac{(I_x - I_y)}{I_z} PQ \end{bmatrix} + \begin{bmatrix} \frac{l(F_2 - F_4)}{I_x} \\ \frac{l(F_1 - F_3)}{I_y} \\ \frac{d}{b}(F_1 - F_2 + F_3 - F_4) \end{bmatrix} \quad (8)$$

where I_x, I_y, I_z are the moment of inertias of the QUAV, b is the thrust coefficient of the propellers and $\dot{P}, \dot{Q}, \dot{R}$ are angular accelerations. There are four inputs for the QUAVs comprise of roll, pitch, yaw and altitude control which are defined in Eqs. (9-12), respectively.

$$u_z = \sum_{i=1}^4 F_i \quad (9)$$

$$u_\phi = l(F_4 - F_2) \quad (10)$$

$$u_\theta = l(F_3 - F_1) \quad (11)$$

$$u_\psi = d(F_1 - F_2 + F_3 - F_4) \quad (12)$$

where l is distance between the center of the gravity of quadrotor and center of propeller, d is the ratio between the drag and the thrust coefficients of the propeller, u_z is control input of the altitude u_ϕ is control input of the roll angle, u_θ is control input of the pitch angle, and lastly u_ψ is control input of the yaw angle, respectively.

Nonlinear attitude dynamics of the QUAV regarding angular rates (without disturbances) are given in Eqs. (13-18).

$$\dot{P} = \left[\frac{(I_y - I_z)}{I_x} QR + \frac{1}{I_x} u_\phi \right] \quad (13)$$

$$\dot{Q} = \left[\frac{(I_z - I_x)}{I_y} RP + \frac{1}{I_y} u_\theta \right] \quad (14)$$

$$\dot{R} = \left[\frac{(I_x - I_y)}{I_z} PQ + \frac{1}{I_z} u_\psi \right] \quad (15)$$

$$\dot{\phi} = P + Qs\phi\tan\theta + Rc\phi\tan\theta \quad (16)$$

$$\dot{\theta} = Qc\phi - Rs\phi \quad (17)$$

$$\dot{\psi} = Qs\phi s\theta + Rc\phi s\theta \quad (18)$$

3. Swarm Intelligence

Swarm Intelligence is a concept that deals with the designing of algorithms or distributed problem solvers inspired by the collective behaviour of social insect and animal societies [28]. The swarm term is used not only for social insect species but also used in a common meaning that focus on any restrained population of interacting individuals [29]. Typical examples of swarms can be bee colonies, ant colonies, flock of birds, swarm of cells, and fish swarms. By inspiring from aforementioned colonies, researchers proposed many swarm intelligence methods to deal with the engineering problems. Some well-known swarm intelligence algorithms are PSO [30], ABC [31] and ACO [32].

According to the [29], necessary and sufficient properties for the swarm intelligence methods are self-organization and division of labour. Self-organization property can be defined as the reflections of the interactions from the lowest-level to the global scale. The property must ensure that the interactions must be executed with local information with no governing relation from the global level. Self-organization property relies on the following properties [28].

- Positive feedback,
- Negative feedback,
- Fluctuations (random walks and errors),
- Minimal density of mutually tolerant individuals.

On the other hand, division of labour property is defined as the performing of different tasks simultaneously by the individuals that are experts on the related task. Thus, efficiency of the task performances is increased compared to the sequential tasks performed by the individuals that are not experts on the related task [33].

Ant Colony Optimization

ACO is an optimization method models the swarm of ants that is based on the natural behaviours of the ant colonies and worker ants. When individual ants forage to find a profitable food source, they find an effective route between the nest and food source. These natural behaviours can be modelled to design an optimization algorithm to solve the engineering problems with an optimum solution. The interactions between the ants rely on the pheromone they release. When ants walk

from the nest to a food source, they use different routes. They release pheromone while they walk. The pheromone is a scent that attracts the other ants and decays. The stronger the pheromone, the more ants are attracted. So, if any of the ants returns quicker than the others with a food, the route that the ant used has stronger pheromone trace behind it. Thus, other ants will instinctively follow this shorter route because of stronger pheromone properties. When more ants used the route, the more pheromone is added, which leads attraction of more ants [34]. This effect is called as *stigmergy* that has two main distinct characteristics from other forms of communications as follows.

- Stigmergy is an indirect form of communication that its media is environment,
- Stigmergy is a local form of communication that only the nearby insects can access.

Stigmergic communication in the ant colonies depends on the concentration of the pheromone. If ants perceive higher concentration of pheromone, they follow this path thus ant colony can transport food sources into the nest efficiently [35].

Another important property that ACO simulate from real ant swarms is autocatalysis. Since the pheromone deposited by the ants is decayed over time, if the path between the nest and food source is shorter than more pheromone is deposited, which leads more ants to use the shorter path [36]. This behaviour belongs to the exploitation of positive feedback, in which more ants will produce higher pheromone concentration results with more ants in that shortest or optimal path.

The pheromone updating rules of ACO for the TSP problem can be given in Eqs. (19-21) [37].

$$\tau_{ij}(t+d) = (1-\rho) * \tau_{ij}(t) + \rho * \Delta\tau_{ij}^+ \quad (19)$$

$$\Delta\tau_{ij}^+ = \frac{1}{L^+} \quad (20)$$

$$\tau_{ij}(t+1) = \xi * \tau_{ij}(t) \quad (21)$$

where $\rho \in (0,1)$ is the persistence of pheromone trails, $(1-\rho)$ is the evaporation rate, d is the number of variables, $\Delta\tau_{ij}^+$ is the amount of pheromone increase of elitist ant, L^+ is the length of the solution of the elitist ant, and ξ is an adjustable parameter where $0 < \xi < 1$.

The flowchart of the ACO algorithm is given in Figure 2. Algorithm starts with initialization of the parameters, constraint definition etc. Ants are randomly located in the search space. Then ants look for food sources. Their pheromone is updated and evaporated according to the time and length of the path. If conditions are satisfied (such as minimum objective function value) or exceeded (such as maximum number of iterations), the ACO algorithm is terminated. The results are stored, and the algorithm is finished.

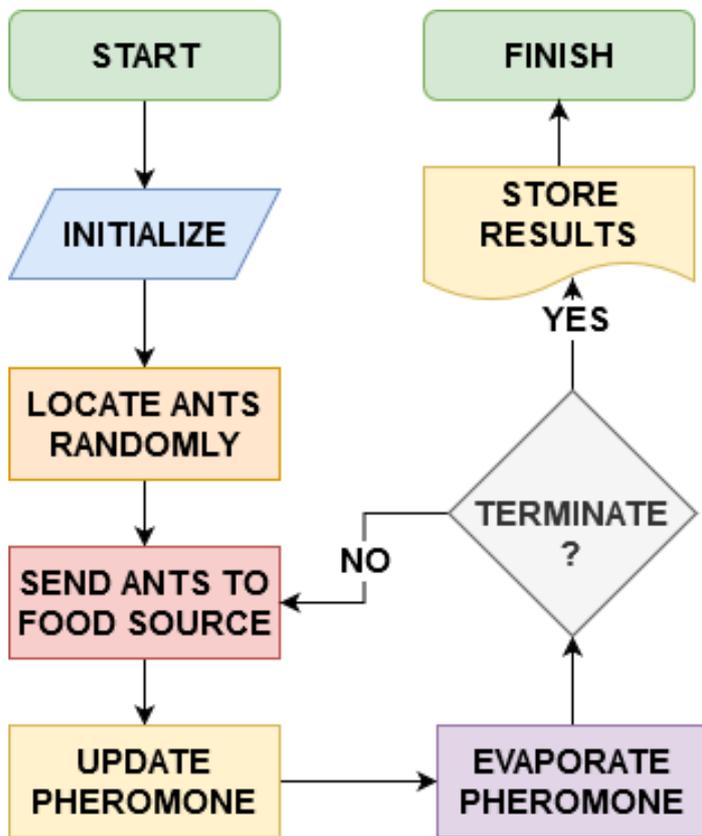


Fig. 2: Flowchart of the ACO algorithm.

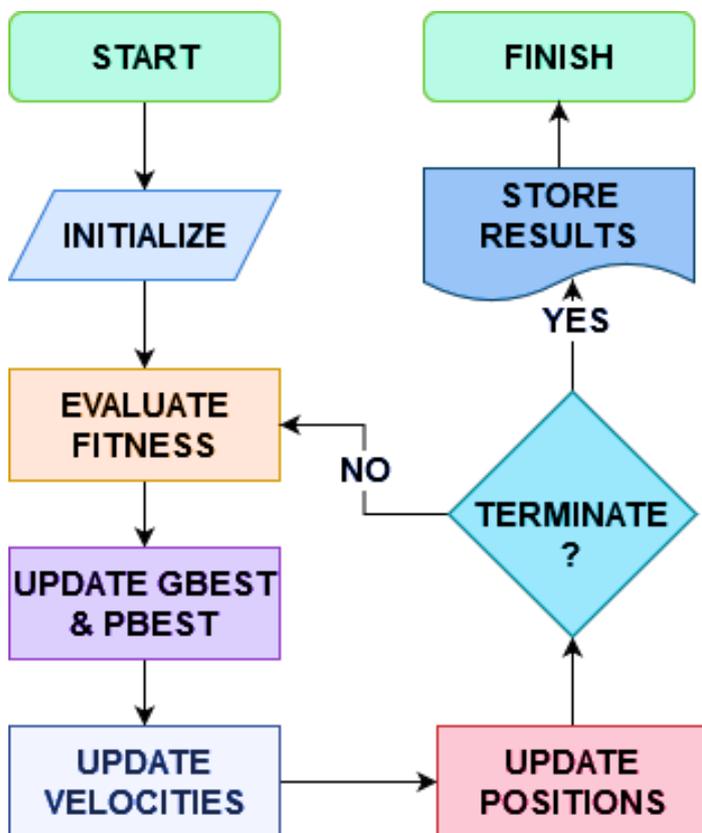


Fig. 3: Flowchart of the PSO algorithm.

Particle Swarm Optimization

PSO is a nature-inspired meta-heuristic optimization algorithm to solve the continuous nonlinear problems related to engineering. PSO is inspired by cognitive and social behaviours of the flock of birds or school of fish that are the swarm of animals hunt for food. PSO is a computationally efficient algorithm since its mathematical operators are primitive [30].

The individuals of the population are called *particle*. Particles are described as collision-proof birds in the initial formulation [34]. PSO algorithm performs the search for the solution through the particles whose trajectories are updated by both stochastic and deterministic rules. Every particle inside the swarm is influenced by the p_{best} and g_{best} in its random walk, where p_{best} is the personal best position of the particle and g_{best} is the global best position of the swarm [38]. Position and velocity update equations of particles are given in Eqs. (22-23).

$$v_{k+1} = w * v_k + c_1 * u_1 * (p_{best} - x_k) + c_2 * u_2 * (g_{best} - x_k) \quad (22)$$

$$x_{k+1} = x_k + v_{k+1} \quad (23)$$

where v is the velocity of the particle, w is the inertia weight, u_1 and u_2 are uniform distributed numbers between [0 1], c_1 is the coefficient of cognitive behaviour, c_2 is the coefficient of social behaviour, x is the position and lastly, subscript k is indexes of generations, respectively.

The flowchart of the PSO algorithm is given in Figure 3. Algorithm starts with initialization of the parameters and continues with fitness function evaluation. According to the fitness values of the particles, p_{best} and g_{best} are updated. Then, velocity and positions are calculated according to Eqs. (22-23). Algorithm continues until the pre-defined conditions are satisfied. Then, solutions are returned and algorithm ends.

Artificial Bee Colony

ABC algorithm can be considered as an extension to the bee swarm algorithm that uses the foraging behaviour. Leading modes of the algorithm are recruitment to a nectar source and abandonment of a source. Bees in the swarm are separated by their expertise areas such as onlooker bees, employed foragers, and scouts. Food source depends on many factors including distance to the nest, richness of source, and ease of extracting this energy [29]. Scout bees exhibit a random exploration search. Onlooker bees wait in the nest to gather information from the employed bees. Each onlooker bee then selects an appropriate food source depending on the calculated probability. The calculated probability mimics the *waggle dance* of the real bees performed by employed foragers. The more amount of nectar leads to

longer dance duration [39]. Search process of the employed bees are same with the onlooker bees. Each employed bee maintains a unique nectar source [40].

The flowchart of the ABC algorithm is given in Figure 4. At first, initial bee swarm is randomly produced in the feasible search space. Bees are sent for food source exploration. After initial searching, employed foragers share the gathered information with the onlooker bees in the nest. Then, onlooker bees choose an appropriate food source depend on the following probability given in Eq. (24).

$$p_i = \frac{f_i}{\sum_{j=1}^{SN} f_j} \quad (24)$$

where p is the calculated probability with subscript i denotes the i th. food source, f is the value of the objective function, SN is the size of the swarm. The modification of the position of the bees to explore better food sources is given in Eq. (25).

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (25)$$

where $-1 \leq \phi_{ij} \leq 1$ is a random number, v_{ij} is the recent generated position, x_{ij} is previous solution, $k \in \{1, 2, \dots, E\}$ where E is the number of employed bees, respectively. After memorizing the best food source algorithm terminates if the predefined conditions are met. Best solution is returned and lastly algorithm ends.

4. PID Controller

PID controllers are one of the most preferred controllers in the areas of process control, motion control, hydraulics, pneumatics thanks to their simplicity, low maintenance costs and ease of implementation [41]. PID controllers deliver a good efficiency in terms of cost per benefit ratio where other controllers may fail [42]. Usage area of the PID controllers are not limited with just low-level control applications but PID controller are also employed in modern applications such as self-driving cars, autonomous robots, and UAVs [43].

PID controllers have three distinct terms as Proportional, Integral and Derivative that each term has one coefficient to adjust the performance and specifications of the controller. In the classical parallel form of the PID structure, time domain formula is given as follows.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (26)$$

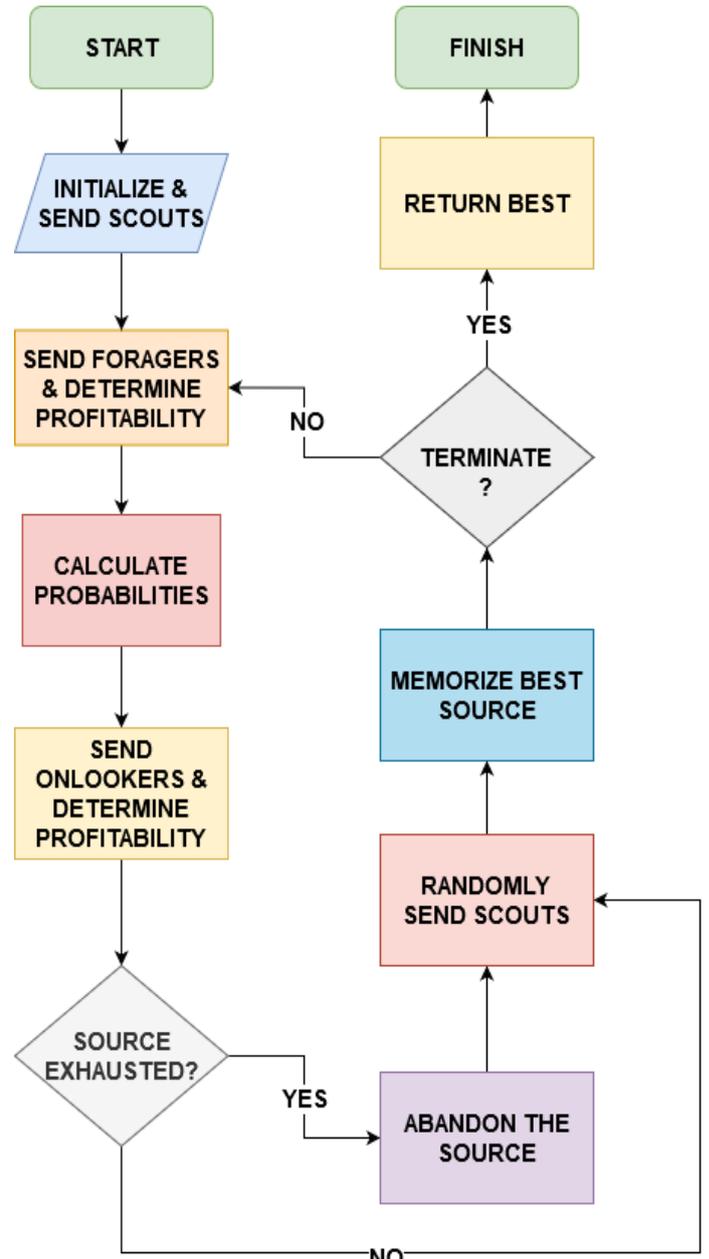


Fig. 4: Flowchart of the ABC algorithm

where $u(t)$ is input of the system, $e(t)$ is the error between the reference and feedback, K_p is the coefficient of proportional term, K_i is the coefficient of integral term and lastly K_d is the coefficient of derivative term, respectively. A block diagram representation of the classical PID controller scheme is given in Figure 5.

In the Figure 5, $x_{ref}(t)$ denotes the reference signal, \hat{x} denotes the measured signal, $e(t)$ is the difference between the observation and reference signals, $K_p e(t)$ is the proportional term, $K_i \int_0^t e(t) dt$ is the integral term, $K_d \frac{de(t)}{dt}$ is the derivative term, $u(t)$ is the input of the system, and $y(t)$ is the output of the plant dynamics, respectively.

The individual effects of PID terms can be given as follows [1].

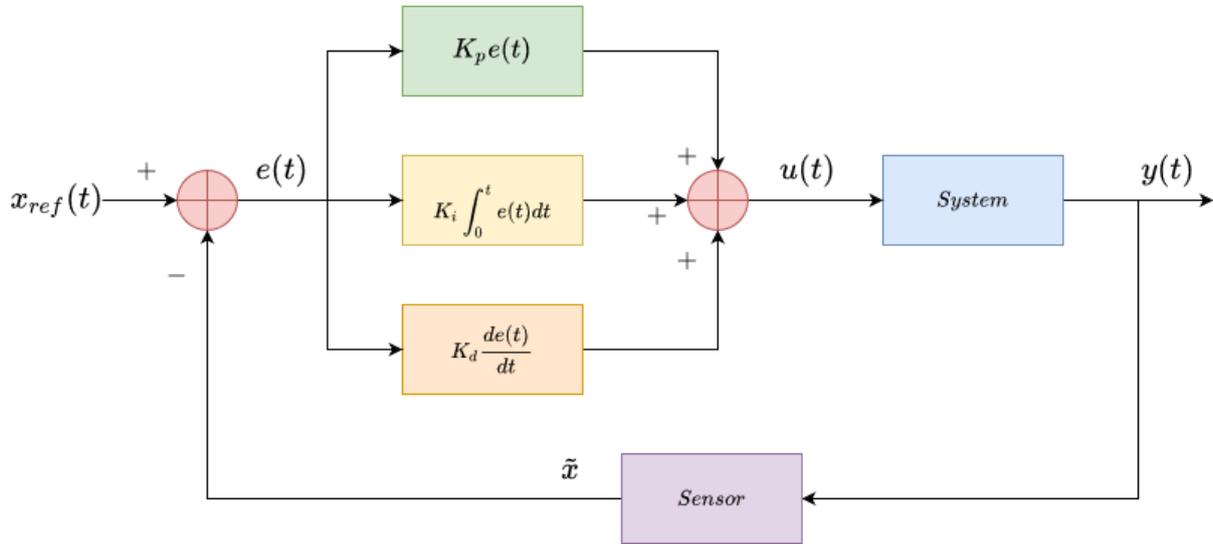


Fig. 5: Block diagram representation of the classical PID control scheme

- P provides an overall control action proportional to the error,
- I reduces the steady-state error by low-frequency compensation,
- D improves the transient response by high-frequency compensation, respectively.

Although K_p, K_i, K_d are mutually dependent in tuning, their individual closed-loop performance effects can be described in Table 1.

Table 1: Independent effects of K_p, K_i, K_d . Source: [1]

Closed-loop response	t_r	%	t_s	e_{ss}	Stability
$K_p \uparrow$ (increasing)	D ↓	I ↑	SI -↑	D ↓	Dg ↓
$K_i \uparrow$ (increasing)	SD -↓	I ↑	I ↑	LD ↓↓	Dg ↓
$K_d \uparrow$ (increasing)	SD -↓	D ↓	D ↓	MC -	Im ↑

where t_r is rise time, t_s is settling time, % is overshoot, e_{ss} is steady-state error, D is decrease, SD is small decrease, I is increase, SI is small increase, LD is large decrease, MC is minor change, Dg is degrade and lastly Im is improve.

5. Results and Discussion

In order to assess the performance and efficiencies of the proposed methods, transient response analysis is conducted. Reference trajectories for the attitude angles are set as $\frac{\pi}{18}$ [rad]. Simulation time is 1.5 [s] with time step of 0.01 [s], respectively. Parameters of transient response analysis are rise time t_r , settling time t_s , maximum percentage overshoot %p and steady-state error e_{ss} . Rise time is the time duration that the response of the controller to rise from 10% to 90%. Settling time is the time duration that the error between the proposed controllers and the final reference value falls within the 2% of the final reference. Maximum percentage

overshoot, as the name suggests, is the ratio of the overshoot with respect to final reference value. Steady-state error is the difference between the final value of the controller and the final value of the reference trajectory. Rise time and settling time are in [s], steady-state error is in [rad] and overshoot is given as [%].

The comparison results of the swarm intelligence based PID attitude controllers are given in Table 2 where boldfaces show the best response. According to the Table 2., it is seen that all the algorithms achieved 0 steady-state error, which shows that all the swarm intelligence based PID controllers can control the attitude states. However, there are some differences among the methods. For ϕ angle, ACO algorithm is the best compared to ABC and PSO with a rise time of 0.15 [s], settling time of 0.32 [s], and maximum percentage of 0.5%. For θ angle, both the ABC and ACO algorithms perform well and better than the PSO algorithm. While rise time of the ABC is the best, settling time and percentage overshoot of the ACO is better than the others. Lastly, for ψ angle, PSO algorithm has the best rise time and settling time responses. However, PSO has percentage overshoot of 0.06% while others have 0% overshoot.

Figures 6–8 show the attitude responses of the proposed controllers for ϕ, θ and ψ angles, respectively. According to the Figure 6 and 7, both the ACO and ABC algorithms perform similar and better than the PSO since PSO algorithm has overshoot and higher rise time characteristics for the given ϕ and θ trajectories. According to the Figure 8, PSO algorithm exhibits better convergence characteristics regarding ABC and ACO with a negligible amount of overshoot.

The algorithms are repeated for 30 MC trials to define the performance of the stochastic behavior of the swarm intelligence based methods. Also, maximum iteration of the meta-heuristics is set as 30, too. The fitness function of the swarm based meta-heuristics is given in Eq. (27).

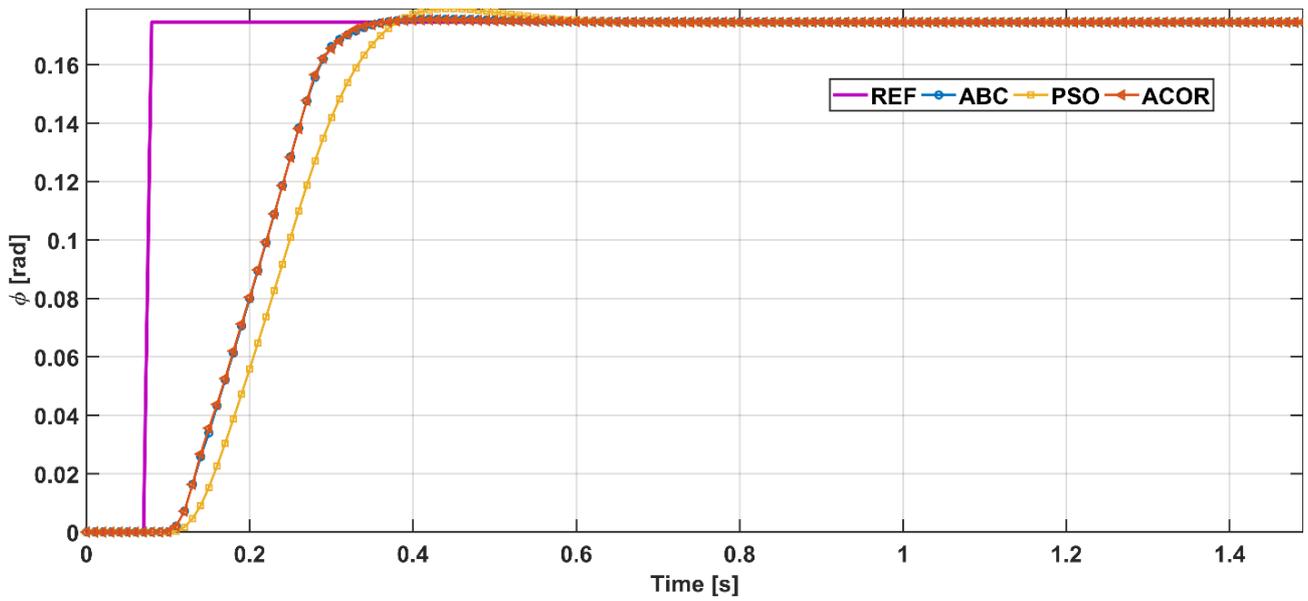


Fig. 6: Phi Response

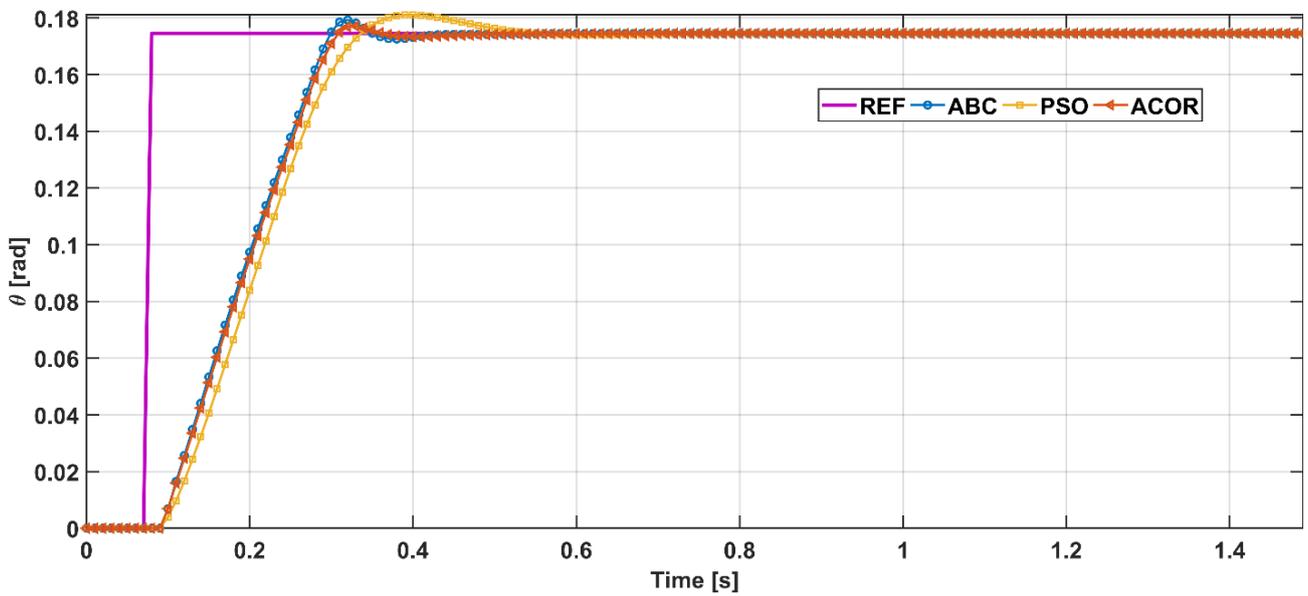


Fig. 7: Theta Response.

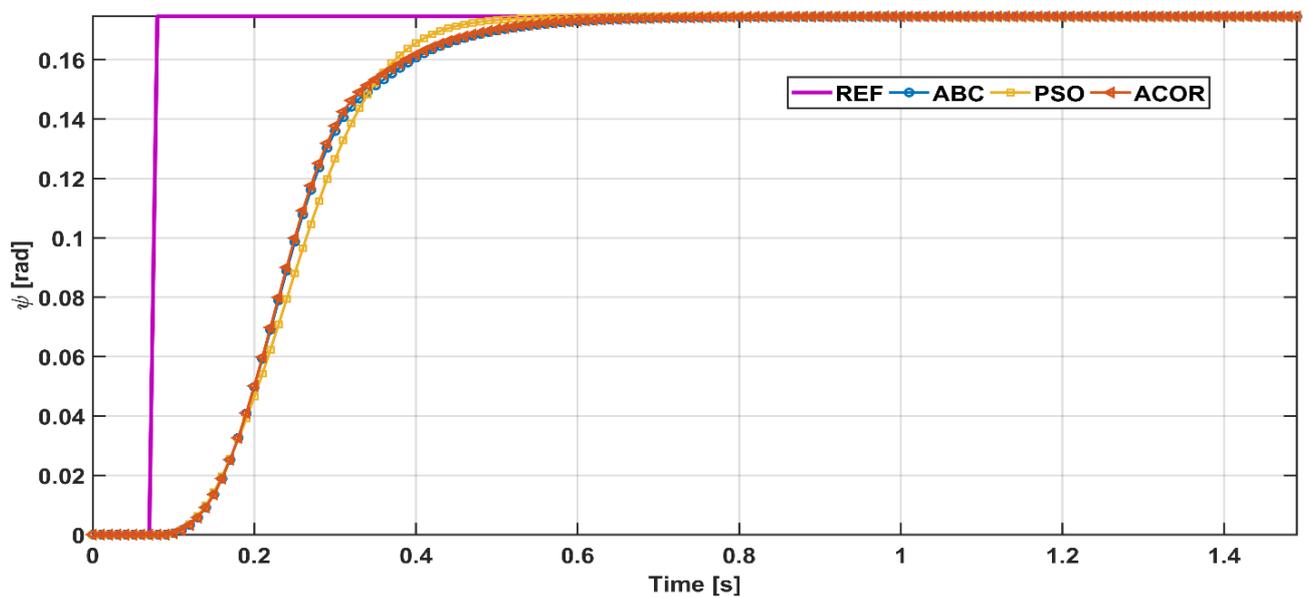


Fig. 8: Psi Response.

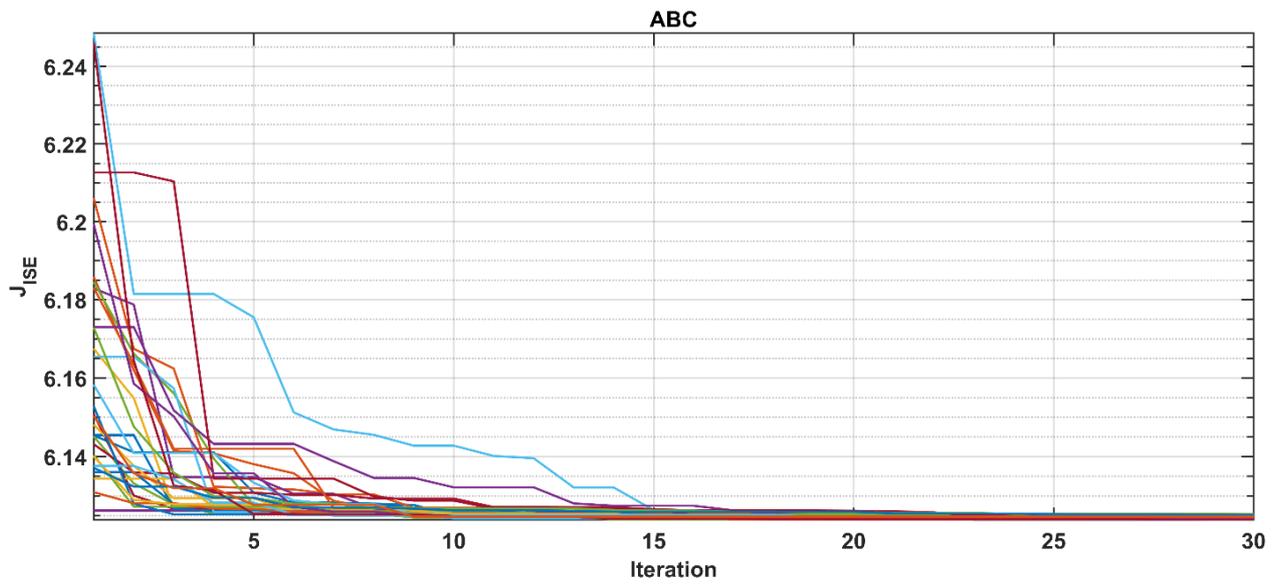


Fig. 9: MC analysis of the ABC algorithm

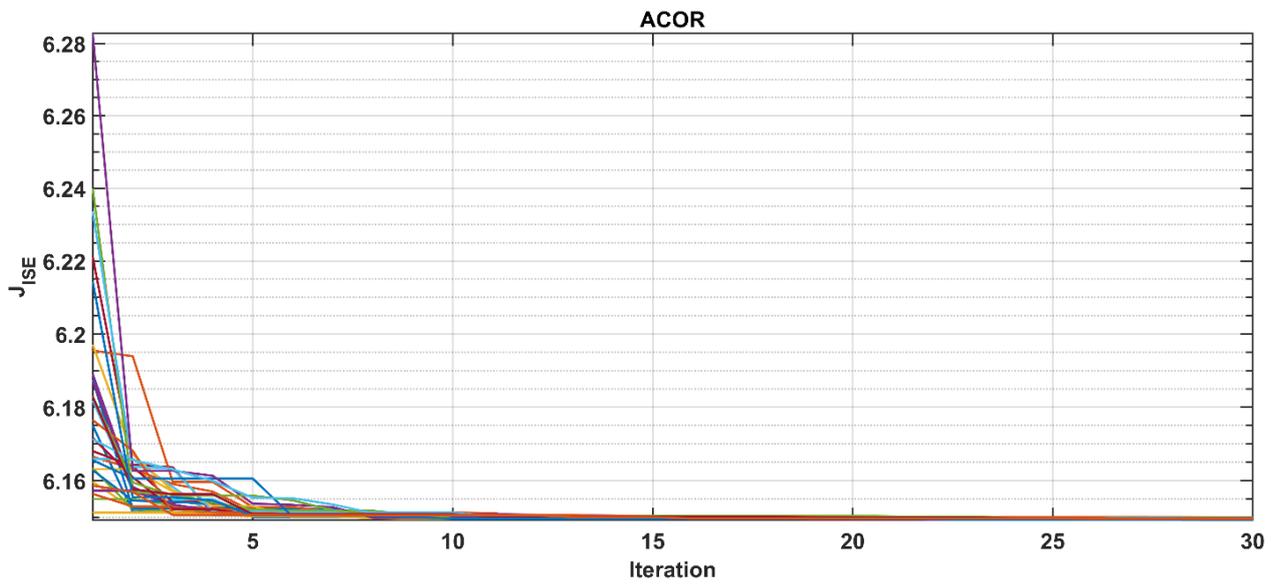


Fig. 10: MC analysis of the ACOR algorithm.

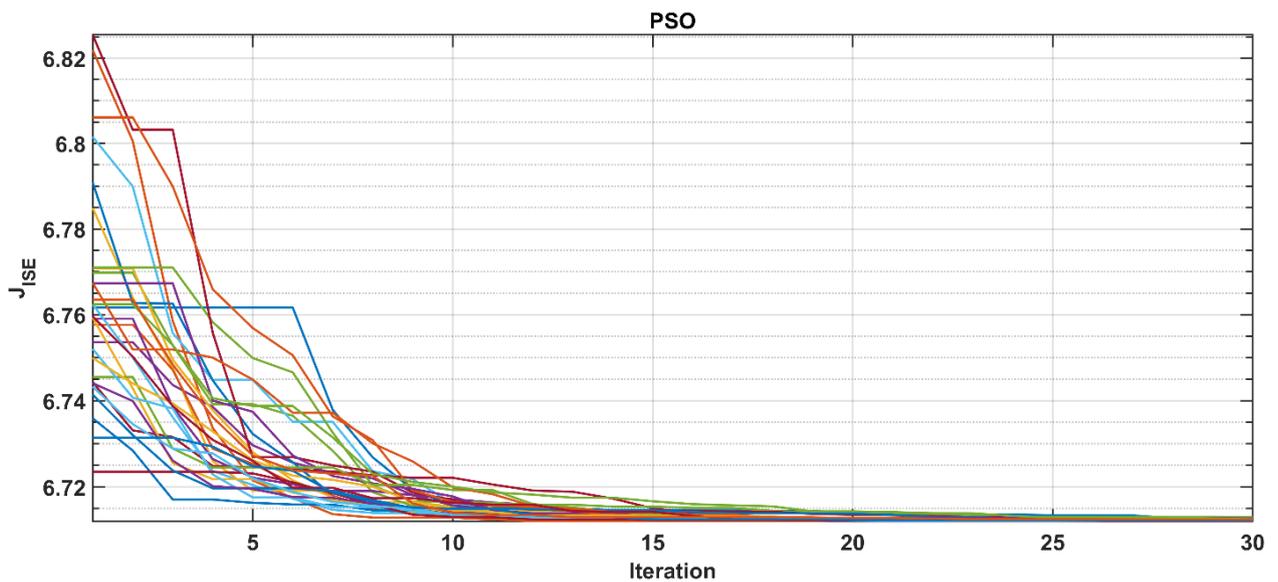


Fig. 11: MC analysis of the PSO algorithm.

Table 2: Transient response analysis of the proposed controllers

		ABC	ACO	PSO
ϕ	t_r	0.1511	0.1498	0.1733
	t_s	0.3260	0.3233	0.4977
	e_{ss}	0.0000	0.0000	0.0000
	%	0.6215	0.4965	2.6892
θ	t_r	0.1632	0.1662	0.1720
	t_s	0.3307	0.3002	0.4671
	e_{ss}	0.0000	0.0000	0.0000
	%	2.7012	1.4619	3.7248
ψ	t_r	0.2222	0.2140	0.2084
	t_s	0.5293	0.5198	0.4467
	e_{ss}	0.0000	0.0000	0.0000
	%	0.0000	0.0000	0.0604

$$J = \int_{t_1}^{t_2} e^2 dt \quad (27)$$

where J is the fitness function, e^2 is the square of the error, t_1 and t_2 are the time limits of sampling frequency. J function is also known as ISE metric: integral squared error. Performances of the swarm intelligence methods for 30 MC run regarding iterations are given in Figures 9 to 11 for ABC, ACO and PSO, respectively. According to the figures, all the 30 MC runs of the ACO algorithm are converged to the sub-optimal region in first 10 iterations of the algorithm. The lowest fitness value of the ACO algorithm is 6.15. Convergence of the ABC algorithm to a sub-optimal point is around first 15 iterations. However, the lowest fitness value of the ABC algorithm is better than the others. MC results of the PSO algorithm are worse than that of both ABC and ACO, respectively.

6 Conclusion

In this work, a comparison of PID controllers based on swarm intelligence methods is conducted for attitude control problem of the nonlinear QUAV dynamics. Three well-known meta-heuristics (ABC, ACO and PSO) are implemented to tune the gains of the attitude controllers. ISE metric is employed as the fitness function. Transient analysis covering rise time, settling time, percentage overshoot and steady-state error is conducted. Also, convergence study of the proposed swarm algorithms based on MC is analysed.

According to the results, all the algorithms have achieved to control and track the given reference attitude trajectories with 0 steady-state error. However, ACO has better characteristics than the others in terms of transient response. MC analysis showed that all the meta-heuristic based methods have good convergence rate into the sub-optimal solution space within 15

iterations for 30 MC runs. So, the proposed controllers can optimize, control and track the pre-determined attitude trajectories within the limited time frame.

The future directions for the manuscript would cover both the employment of novel meta-heuristic algorithms and also the expansion of the full quadrotor states.

Nomenclature

ABC	: Artificial Bee Colony
ACO	: Ant Colony Optimization
B	: Body – fixed frame
BP	: Back Propagation
DC	: Direct Current
DE	: Differential Evolution
E	: Earth – fixed frame
DOF	: Degree – of – freedom
FPGA	: Field Programmable Gate Array
GA	: Genetic Algorithm
ISE	: Integral Squared Error
KF	: Kalman Filter
NN	: Neural Network
MC	: Monte Carlo
MIMO	: Multi Input – Multi Output
P	: Proportional Controller
PI	: Proportional Integral Controller
PD	: Proportional Derivative Controller
PID	: Proportional Integral Derivative Controller
PSO	: Particle Swarm Optimization
RBF	: Radial Basis Function
RPM	: Rotation per minute
TSP	: Traveling Salesman Problem
QUAV	: Quadrotor Unmanned Aerial Vehicle
UAV	: Unmanned Aerial Vehicle
VTOL	: Vertical Take – Off and Landing
ZN	: Ziegler – Nichols

References

- [1] K. H. Ang, G. Chong and Y. Li, "PID Control System Analysis, Design, and Technology," IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, 2005.
- [2] G.-Q. Zeng, X.-Q. Xie, M.-R. Chen and J. Weng, "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems," Swarm and Evolutionary Computation, vol. 44, pp. 320 - 334, 2019.
- [3] E. Çelik and N.Öztürk, "A hybrid symbiotic organisms search and simulated annealing technique applied to efficient design of PID controller for automatic voltage regulator," Soft Computing, vol. 22, pp. 8011 - 8024, 2018.
- [4] S. Panda, B. Sahu and P. Mohanty, "Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization," Journal of Franklin Institute, vol. 349, pp. 2609 - 2625, 2012.

- [5] T. Häggglund, "Signal filtering in PID control," IFAC Proceedings, vol. 45, no. 3, pp. 1 - 10, 2012.
- [6] A. Tepljakov, E. Gonzalez, E. Petlenkov, J. Belikov, C. Monje and I. Petráš, "Incorporation of fractional-order dynamics into an existing PI/PID DC motor control loop," ISA Transactions, vol. 60, pp. 262 - 273, 2016.
- [7] A. Masoumian, P. Kazemi, M. C. Montazer, H. A. Rashwan and D. P. Valls, "Designing and Analyzing the PID and Fuzzy Control System for an Inverted Pendulum," in 6th International Conference on Mechatronics and Robotics Engineering (ICMRE), Barcelona, Spain, 2020.
- [8] X. Wu, X. Wang and G. He, "A Fuzzy Self-tuning Temperature PID Control Algorithms for 3D Bio-printing Temperature Control System," in Chinese Control And Decision Conference (CCDC), Hefei, China, 2020.
- [9] Z. Ge, F. Liu and L. Meng, "Adaptive PID Control for Second Order Nonlinear Systems," in Chinese Control And Decision Conference (CCDC), Hefei, China, 2020.
- [10] S. Zhang, Z. Dan, Q. Qian, Y. Guo and J. Zhang, "Nonlinear PID Pressure Control Based on Extremum Seeking," in 39th Chinese Control Conference (CCC), Shenyang, China, 2020.
- [11] S. Lihan, M. Jie and Y. Baoqing, "Fuzzy PID Design of Vehicle Attitude Control Systems," in Chinese Control And Decision Conference (CCDC), Hefei, China, 2020.
- [12] M. Mahmud, S. M. A. Motakabber, A. H. M. Z. Alam and A. N. Nordin, "Adaptive PID Controller Using for Speed Control of the BLDC Motor," in IEEE International Conference on Semiconductor Electronics (ICSE), Kuala Lumpur, Malaysia, 2020.
- [13] S. Mu, S. Shibata, T. Yamamoto, K. Tanaka, S. Nakashima and T. Liu, "Experimental Study on Speed Control of Ultrasonic Motor using Intelligent IMC-PID Control," in International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Kaohsiung, Taiwan, 2019.
- [14] S. Wei, G. Chen, H. Zhang, H. Lu and S. Wu, "Design of Z-Axis Position Servo Control System for Laser Processing Based on Fuzzy PID Control," in 12th International Conference on Intelligent Computation Technology and Automation (ICICTA), Xiangtan, China, 2019.
- [15] A. A. A. Razak, A. N. K. Nasir, M. F. M. Jusof, S. Mohammad and N. A. M. Rizal, "Opposition based Spiral Dynamic Algorithm with an Application to a PID Control of a Flexible Manipulator," in 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2019.
- [16] H. Zhang and W. Assawinchaichote, "PID Control Based On Double Fuzzy RBF Neural Network For 7-DOF Manipulator," in 8th International Electrical Engineering Congress (IEECON), Chiang Mai, Thailand, 2020.
- [17] E. A. Yfantis and W. Culbreth, "A Data Driven PID Control System," in 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020.
- [18] L. Bauersfeld and G. Ducard, "Fused-PID Control for Tilt-Rotor VTOL Aircraft," in 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphaël, France, 2020.
- [19] H. Matsuki, T. Nishiyama, Y. Omori and S. Suzuki, "Flight test of fault-tolerant flight control system using simple adaptive control with PID controller," Aircraft Engineering and Aerospace Technology, vol. 90, no. 1, pp. 210 - 218, 2018.
- [20] I. Yazar, E. Kiyak, F. Çalışkan and T. H. Karakoç, "Simulation-based dynamic model and speed controller design of a small-scale turbojet engine," Aircraft Engineering and Aerospace Technology, vol. 90, no. 2, pp. 351-358, 2018.
- [21] A. Ermeýdan and E. Kiyak, "Fault tolerant control against actuator faults based on enhanced PID controller for a quadrotor," Aircraft Engineering and Aerospace Technology, vol. 89, no. 3, pp. 468-476, 2017.
- [22] A. Kaba, A. Ermeýdan and E. Kiyak, "Model derivation, attitude control and Kalman filter estimation of a quadcopter," in 4th. International Conference on Electrical and Electronic Engineering, IEEE, Ankara, 2017.
- [23] Z. Long, Z. Jiang, C. Wang, Y. Jin, Z. Cao and Y. Li, "A Novel Approach to Control of Piezo-Transducer in Microelectronics Packaging: PSO-PID and Editing Trajectory Optimization," IEEE Transactions On Components, Packaging And Manufacturing Technology, vol. 10, no. 5, pp. 795 - 805, 2020.
- [24] E. Kiyak, "Tuning of controller for an aircraft flight control system based on particle swarm optimization," Aircraft Engineering and Aerospace Engineering, vol. 88, no. 6, pp. 799-809, 2016.
- [25] J. Connor, M. Seyedmahmoudian and B. Horan, "Using particle swarm optimization for PID optimization for altitude control on a quadrotor," in Australasian Universities Power Engineering Conference, 2017.
- [26] Y. Chen, S. Liu, C. Xiong, Y. Zhu and Jiaheng Wang, "Research on UAV Flight Tracking Control Based on Genetic Algorithm Optimization and Improved bp Neural Network pid Control," in Chinese Automation Congress (CAC), Hangzhou, China, 2019.

- [27] J. O. Pedro, M. Dangor and P. J. Kala, "Differential Evolution-Based PID Control of a Quadrotor System for Hovering Application," in IEEE Congress on Evolutionary Computation, 2016.
- [28] E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford, England: Oxford University Press, 1999.
- [29] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," Technical Report - TR06, Erciyes University, Kayseri, 2005.
- [30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995.
- [31] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459-471, 2007.
- [32] M. Dorigo, V. Maniezzo and A. Coloni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Trans Syst Man Cybernetics - Part B*, vol. 26, pp. 29 - 41, 1996.
- [33] V. Tereshko, "Reaction-diffusion model of a honeybee colony's foraging behaviour," in *Parallel Problem Solving from Nature VI*, Berlin, Springer-Verlag, 2000, pp. 807 - 816.
- [34] J. Christensen and C. Bastien, "Heuristic and Meta-Heuristic Optimization Algorithms," in *Nonlinear Optimization of Vehicle Safety Structures*, Butterworth-Heinemann, 2016, pp. 277 - 314.
- [35] M. Dorigo, M. Birattari and T. Stützle, "Ant Colony Optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28 - 39, 2006.
- [36] M. K. Jha, "Metaheuristic Applications in Highway and Rail Infrastructure Planning and Design: Implications to Energy and Environmental Sustainability," in *Metaheuristics in Water, Geotechnical and Transport Engineering*, Elsevier, 2013, pp. 365 - 384.
- [37] S. Talatahari, V. P. Singh and Y. Hassanzadeh, "Ant Colony Optimization for Estimating Parameters of Flood Frequency Distributions," in *Metaheuristics in Water, Geotechnical and Transport Engineering*, Elsevier, 2013, pp. 121 - 146.
- [38] C. M. Martínez and D. Cao, "Integrated energy management for electrified vehicles," in *Horizon-Enabled Energy Management for Electrified Vehicles*, Butterworth-Heinemann, 2019, pp. 15 - 75.
- [39] A. Kaba and E. Kiyak, "Artificial bee colony-based Kalman filter hybridization for three-dimensional position estimation of a quadrotor," *Aircraft Engineering and Aerospace Technology*, vol. 92, no. 10, pp. 1523-1532, 2020.
- [40] X. Li and G. Yang, "Artificial bee colony algorithm with memory," *Applied Soft Computing*, vol. 41, pp. 362 - 372, 2016.
- [41] Å. KJ and H. T, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163 - 1175, 2001.
- [42] R. P. Borase, D. K. Maghade, S. Y. Sondkar and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, 2020.
- [43] I. D. Díaz-Rodríguez, H. Sangjin and S. P. Bhattacharyya, *Analytical design of PID controllers*, Berlin: Springer, 2019.