O 73. OPTIMAL TUNING OF PID CONTROLLER FOR QUADROTOR SYSTEM USING A NEW ADAPTIVE PARTICLE SWARM OPTIMIZATION

Fahed Sayed[1*], Turker Turker[1]

[1*]*Yildiz Technical University, Istanbul, Turkey*
[1]*Yildiz Technical University, Istanbul, Turkey*

E-mail: sayd.fahed@gmail.com, turker@yildiz.edu.tr

ABSTRACT: In this study, a detailed mathematical nonlinear dynamic model of the quadcopter system which is derived using the Newton-Euler method. PID controllers are used for controlling the roll, pitch, yaw, and altitude movements of the quadcopter. Manual tuning of PID controllers does not always give acceptable results, consume a long time, and difficult. Therefore, the tuning process of PID controllers is done by particle swarm optimization algorithm (PSO). A new adaptive particle swarm optimization (APSO) algorithm that gives better search efficiency and convergence speed than standard particle swarm optimization is suggested. It enables the automatic control of inertia weight which controls the global and local search abilities of the PSO algorithm. Comparing with the trial and error method and standard PSO algorithm, the adaptive PSO algorithm gives better performance in terms of convergence speed and permanent movement toward the optimal solution region.

Keywords: Adaptive Particle Swarm Optimization, Self-Tuning PID Controller, Quadcopter System

1. INTRODUCTION

The quadcopter is the most popular unmanned aerial vehicle (UAV). Recently, it becomes an attractive platform for UAV researches. It is preferred because it has a simple mechanical structure and can perform a different set of tasks. Besides, it has the ability to hover and vertical take-off and landing (VTOL). Quadcopters are used in many civil applications including real-time monitoring, search and rescue operations, disaster management, supplying wireless coverage, remote sensing systems, infrastructure inspection, security and surveillance, delivery of goods, and perform precision agriculture (Shakhatreh et al. 2018).

PID controller is the most widespread type of controller used for ensuring quadcopter stability because it is simple to use and able to offer an effective solution. The tuning process of the PID parameters is too complex and requires enough knowledge about the system being controlled. PID controller performance totally relies on the tuning process of its parameters. A lot of evolutionary algorithms are used for optimal tuning of the PID parameters such as genetic algorithm (Gundogdu, 2005), artificial bees algorithm (Coban et al. 2012), particle swarm optimization algorithm (Berber et al. 2016).

Particle swarm optimization (PSO) algorithm is a population-based search algorithm. It has proved that it can solve many complex optimization problems. In this study, a mathematical modelling of the quadcopter system has been used and performance comparison between various strategies to find the optimal PID parameters has been tested. These strategies are trial and error, standard particle swarm optimization algorithm, and adaptive particle swarm optimization algorithm (APSO). The inertia weight is the most important parameter in particle swarm optimization algorithm. It is the most effective parameter to control global and local search processes. Adaptive particle swarm optimization algorithm (APSO) provides automatic control of inertia weight over time for each particle and iteration. In this study, a new adaptive particle swarm optimization algorithm has been suggested in order to control the local and global search processes and enhance the total performance of the PSO algorithm.

2. MATERIAL AND METHOD

2.1. Quadrotor Model

The basic dynamical model of the quadcopter is the starting point for lots of researches and generally is derived by the Newton-Euler equations or Euler-Lagrange equations (Luukkonen, 2011). Assuming that the quadrotor has symmetrical structure the inertia matrix (I) will be diagonal matrix with (I_{xx}), (I_{yy}), and (I_{zz}) the inertia of the vehicle across each axis.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{1}$$

The thrust force (fi) created by each rotor (i) is vertical to the X-Y plane of the body frame and in the rotor axis direction.

$$\begin{aligned} fi &= k\omega_i^2 \\ \tau_{M_i} &= b\omega_i^2 + I_M \dot{\omega}_i \end{aligned} \tag{2}$$

Where (k) is the lift constant, (ω_i) is the angular velocity for a specific rotor (i), (b) is the drag constant, and (I_M) is the inertia moment of the rotor. The angular velocity and acceleration of the rotor also create torque (τ_{Mi}) that acts to rotate the vehicle about the z-axis. But as the small effect of ($\dot{\omega}_i$) it is usually omitted.

$$T = \sum_{i=1}^{4} f_i = k\sum_{i=1}^{4} \omega_i^2 \;,\; T_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{3}$$

The total lift forces of rotors create thrust (T) in z axis direction of the body frame. And by decreasing (ω_2) and increasing (ω_4) the roll torque is obtained. Likewise by decreasing (ω_1) and increasing (ω_3), pitch torque is acquired. Also Yaw torque is created by increasing of (ω_2, ω_4) and decreasing (ω_1, ω_3). (T_B) is thrust in the body frame.

$$\tau_B = \begin{bmatrix} lk(-\omega_2^2 + \omega_4^2) \\ lk(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix} \tag{4}$$

Where (τ_B) is torque in the body frame and (l) is the arm length of the quadrotor.

$$I\,\dot{v} + v \times (I\,v) + \Gamma = \tau \tag{5}$$

The external torque (τ) in the body frame created by the angular acceleration ($\dot{v}$), gyro scoping forces, and moments (Γ) applied by rotors. Therefor the change in roll (p), pitch (q), and yaw (r) rates can be obtained from this equation:

$$\begin{aligned} \dot{v} &= I^{-1}\left(-\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} - I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_\Gamma + \tau \right) \\ \Leftrightarrow \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} (I_{yy} - I_{zz})qr / I_{xx} \\ (I_{zz} - I_{xx})pr / I_{yy} \\ (I_{xx} - I_{yy})pq / I_{zz} \end{bmatrix} - I_r \begin{bmatrix} q / I_{xx} \\ -p / I_{yy} \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau\phi / I_{xx} \\ \tau\theta / I_{yy} \\ \tau\psi / I_{yy} \end{bmatrix} \end{aligned} \tag{6}$$

The rotation matrix (R) from the inertial frame to the body frame using the aerospace rotation sequence, and this matrix has special importance in resolving the velocity and position state equations.

$$R = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi - C\phi S\psi & S\phi S\psi + C\phi S\theta C\psi \\ C\theta S\psi & C\phi C\psi + S\phi S\theta S\psi & C\phi S\theta S\psi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \tag{7}$$

Where $S_x = \sin(x)$, $C_x = \cos(x)$. The matrix H_Φ represents the transformation matrix for angular velocities from the inertial frame (v) to the body frame ($\dot{\Phi}$).

$$\dot{\Phi} = H_\Phi \, v$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & T_\theta S_\phi & T_\theta C_\phi \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

(8)

In the body frame, the needed force for the acceleration of mass $m \, \dot{V}_B$ and the centrifugal force $v \times (m V_B)$ are equal to the sum of the gravity $R^T G$ and the total thrust of the rotors T_B.

$$m \, \dot{V}_B + v \times (m V_B) = R^T G + T_B \qquad (9)$$

In the inertial frame, the acceleration of the quadrotor is affected just by the magnitude and direction of the thrust and the gravitational force because the centrifugal force is cancelled.

$$m\ddot{X} = G + R T_B$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} c\psi s\theta c\phi + s\psi s\phi \\ s\psi s\theta c\phi - c\psi s\phi \\ c\theta c\phi \end{bmatrix} \qquad (10)$$

2.2. PID Controller Design

PID controller is the most widespread strategy to control the quadcopter system (Bolandi et al. 2013; Mohammed et al. 2014). It is a type of feedback controller which is used to control the quadcopter altitude with the roll, pitch, and yaw angles. The transfer function of the PID controller uses the following form:

$$u = K_p \, e(t) + K_i \int_0^t e(t) dt + K_d \, \frac{de(t)}{dt} \qquad (11)$$

K_p, K_d, and K_i indicate to the proportional, derivative, and integral parameters respectively that need to be tuned. The PID controller goals to minimize the error signal (e) between the reference value and the measured output.

The tuning process of the PID parameters determines the controller performance. In this paper, an adaptive particle swarm optimization algorithm is used to tune the PID parameters in order to find its optimal values. Besides, the tuning process is also done by standard particle swarm optimization and trial and error method.

2.3. Standard Particle Swarm Optimization

Particle swarm optimization algorithm, a population-based evolutionary computational method, was proposed by Eberhart and Kennedy in 1995 (Eberhart and Kennedy, 1995). It searches for the optimal solution of a problem by iteratively improving the available solutions inspired by the flocking of birds and schooling of fish.

PSO algorithm is initialized with a number of random particles and every single particle represents one possible solution to the related problem. These particles are evaluated each iteration according to its fitness function. Particles have velocities to steer them toward the current optimal particle. Figure 1 displays the flow chart of a particle swarm optimization algorithm.

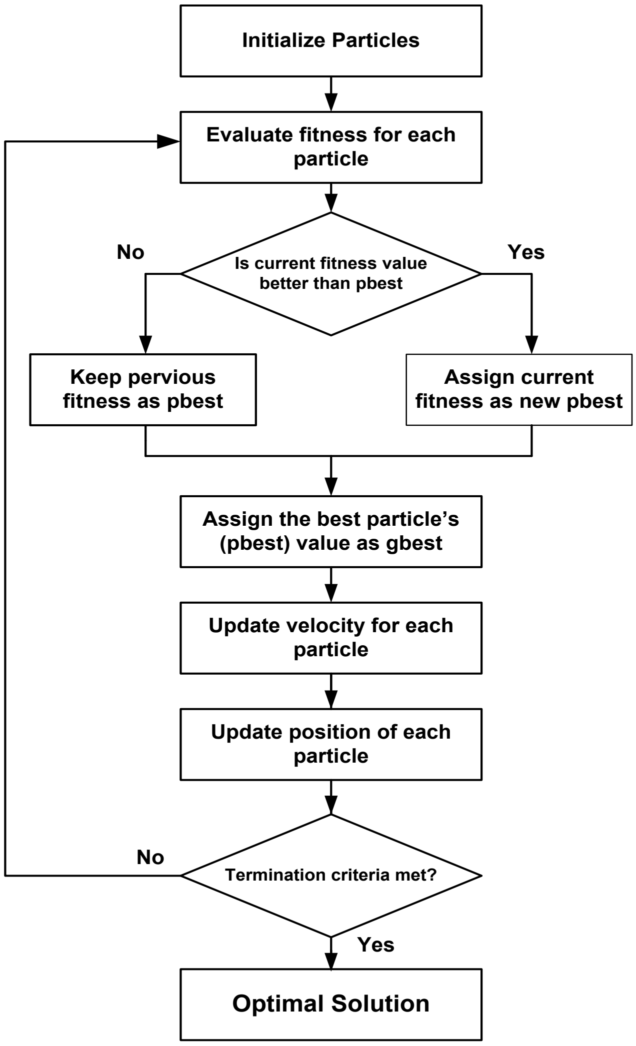

Figure 1. Flow Chart of Particle Swarm Optimization Algorithm

Each particle represents three parameters which are the proportional, derivative and integral gains of PID controller. PSO algorithm starts the first iteration with a number of particles that carry random values. These particles update their values in every iteration by the pbest and gbest particles. pbest particles are the same particle's best-known position during all iterations, while the gbest particle is the swarm's best-known position in the same iteration. PSO algorithm updates the particle's values and velocities depending on the two best values with the following equations:

$$V_i^{(t+1)} = w * V_i^{(t)} + c_1 * r_1 * (x_{i,best}^{(t)} - x_i^{(t)}) + c_2 * r_2 * (x_{gbest}^{(t)} - x_i^{(t)}) \quad (12)$$

$$x_i^{(t+1)} = x_i^{(t)} + V_i^{(t+1)} \quad (13)$$

PSO algorithm uses the inertia weight (w), the acceleration coefficients (c_1) and (c_2), and two random numbers (r_1, r_2) within the range [0, 1]. Standard PSO algorithm supposes that the inertia weight w=1 and the acceleration coefficients $c_1=c_2=2$. The fitness value of particles determines the pbest and gbest particles in each iteration. The gbest particle is the particle that holds the best fitness value among all particles in the same iteration. Also, the pbest particles are the particles that hold the best fitness value among the same particles during all iterations. The stopping condition in PSO algorithm may be a determined number of iterations or a constant value of the fitness value.

2.4. Proposed Adaptive Particle Swarm Optimization

Adaptive particle swarm optimization algorithm (APSO) provides control ability of inertia weight, acceleration coefficients, and other algorithmic parameters in every iteration. Inertia weight parameter noticeably affects the global search ability (exploration) and local search ability (exploitation) in the

PSO process. Since the beginning of Inertia Weight in PSO, a large number of strategies to control inertia weight have been proposed (Bansal et al. 2011; Alhasan and Gunes, 2017).
When inertia weight has a large value, the PSO algorithm facilitates the global search more than the local search. The PSO algorithm also facilitates the local search more than the global search when inertia weight has a low value. This balancing between the global and local search processes improve the performance of PSO algorithm.
In this paper, we have adjusted the value of the inertia weight (w) adaptively by this new equation:

$$\begin{aligned} \alpha_{1,i}^{K} &= \frac{F_{pbest,i}^{K}}{2F_{p,i}^{K}} \\ \alpha_{2,i}^{K} &= \frac{F_{gbest}^{K}}{2F_{pbest,i}^{K}} \\ w_i^K &= w_{\max} - (w_{\max} - w_{\min}) \times \left(\alpha_{1,i}^{K} + \alpha_{2,i}^{K} \right) \end{aligned} \tag{14}$$

The inertia weight (w) values range from the upper limit 'w_{max}=0.9' to the lower limits 'w_{min}=0.3'. Every particle in the swarm has a different inertia weight value in the same iteration. The values of (α_1, α_2) determine the evolutionary states of the particles in the swarm. If a particle has an unsuccessful solution ($F_{p,i} >> F_{pbest,i} >> F_{gbest}$), the values of ($\alpha_1$, α_2) will be too close to zero and the inertia weight will have a large value in order to enhance the global search ability. If a particle has a successful or a good solution ($F_{p,i} \approx F_{pbest,i} \approx F_{gbest}$), the values of ($\alpha_1$, α_2) will be too close to (0.5) and the inertia weight will have a low value in order to enhance the local search ability.
The fitness function used to evaluate particles in PSO algorithm is sum square error (SSE) and as follows:

$$F = 0.25(\phi_r - \phi)^2 + 0.25(\theta_r - \theta)^2 + 0.25(\psi_r - \psi)^2 + 0.25(\mathrm{z}_r - \mathrm{z})^2 \tag{15}$$

The symbols $\phi_r, \theta_r, \psi_r,$ and z_r are the reference values of the roll, pitch, yaw, and altitude respectively.

3. RESEARCH FINDINGS

In this section a comparison and simulation results for three methods to tune the parameters of PID controllers of the quadcopter system. These methods are trial and error, standard particle swarm optimization (PSO) and the proposed adaptive particle swarm optimization (APSO).
PID controllers of quadcopter system are modelled and simulated using the MATLAB/Simulink environment. In this simulation, initially, the roll ϕ, pitch θ, yaw ψ angles are set to (0.2) radians and altitude z initialized to (3) meters. The reference value of ϕ, θ, and ψ angles is set to (0) radians, while the reference value of altitude z is set to (4) meters. Figures (2-4) show the roll (ϕ), pitch (θ), and yaw (ψ) angles response using the PID controller which is tuned by three strategies. Altitude (z) response using the tuned PID controller is shown in Figure 5.
The first strategy to tune the PID controller is a trial and error method. The second strategy to tune the PID controllers is done using the standard particle swarm optimization algorithm. The inertia weight is set to '1' and the acceleration coefficients are set to '2' in the second strategy. The third strategy to tune the PID controllers is done using the adaptive particle swarm optimization algorithm where the inertia weight is adapted according to equation (14) and the acceleration coefficients are set to '2'.

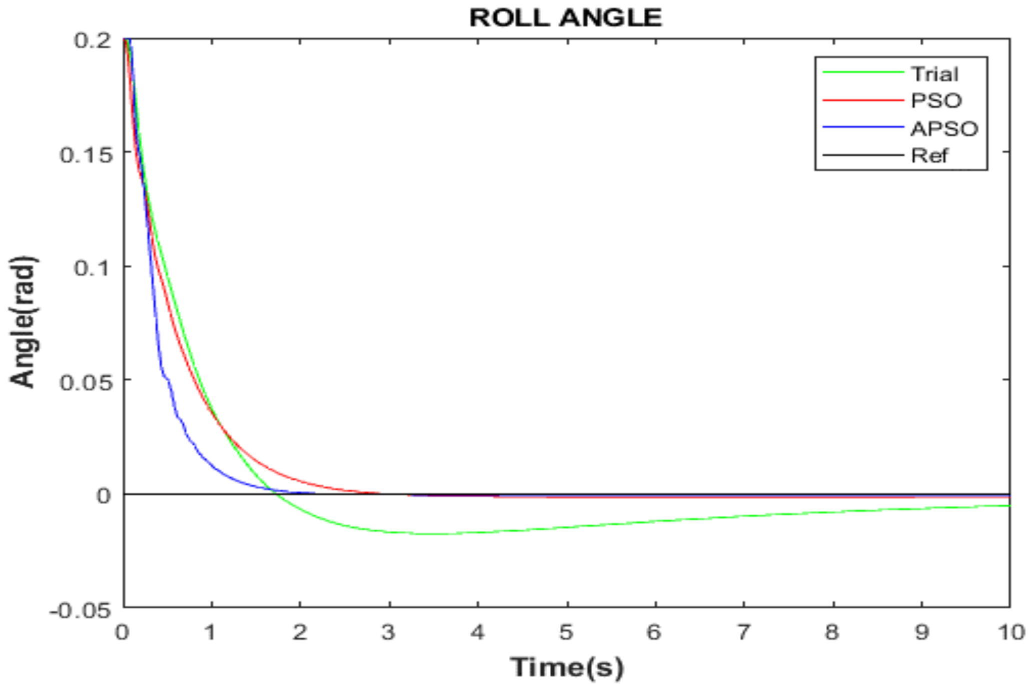

Figure 2. Time Response of the Roll Angle of the Quadcopter System Using PID Controller.

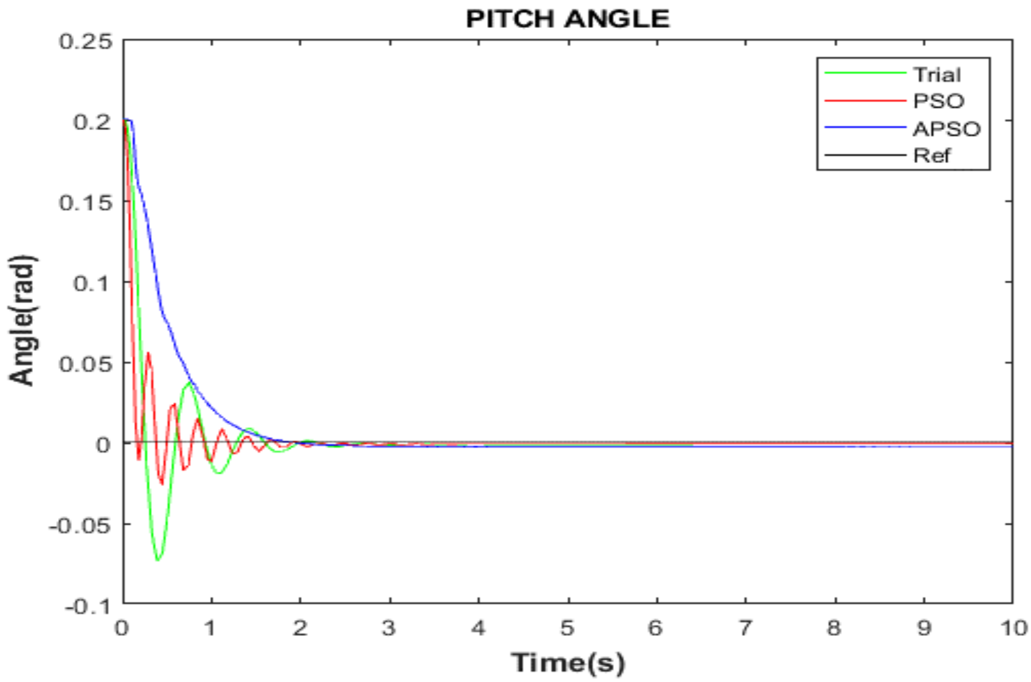

Figure 3. Time Response of the Pitch Angle of the Quadcopter System Using PID Controller.

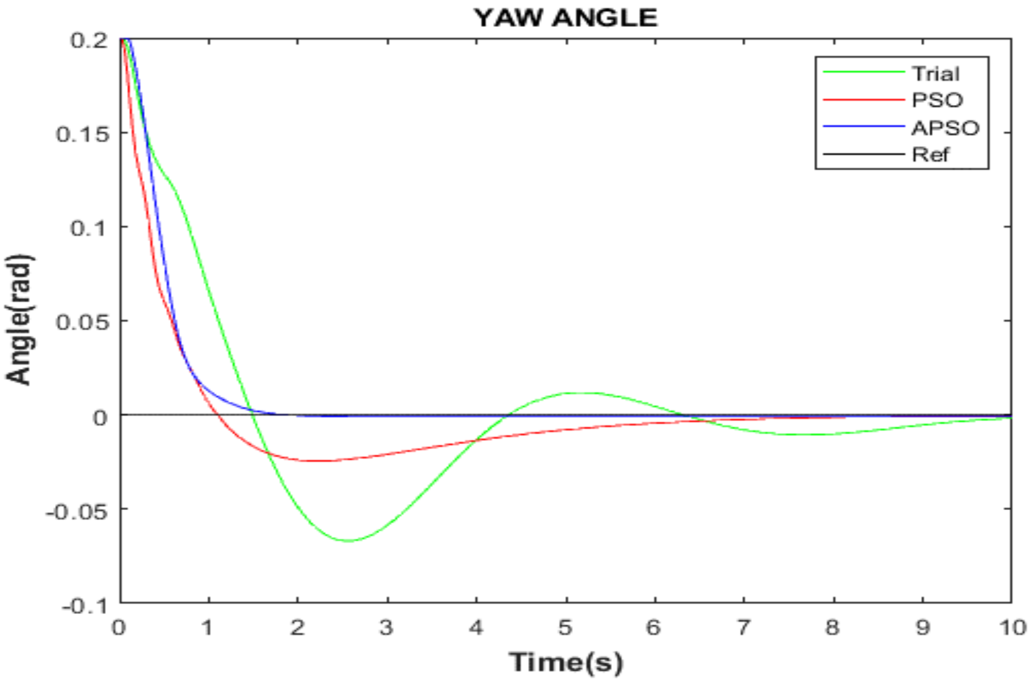

Figure 4. Time Response of the Yaw Angle for the Quadcopter System Using PID Controller.

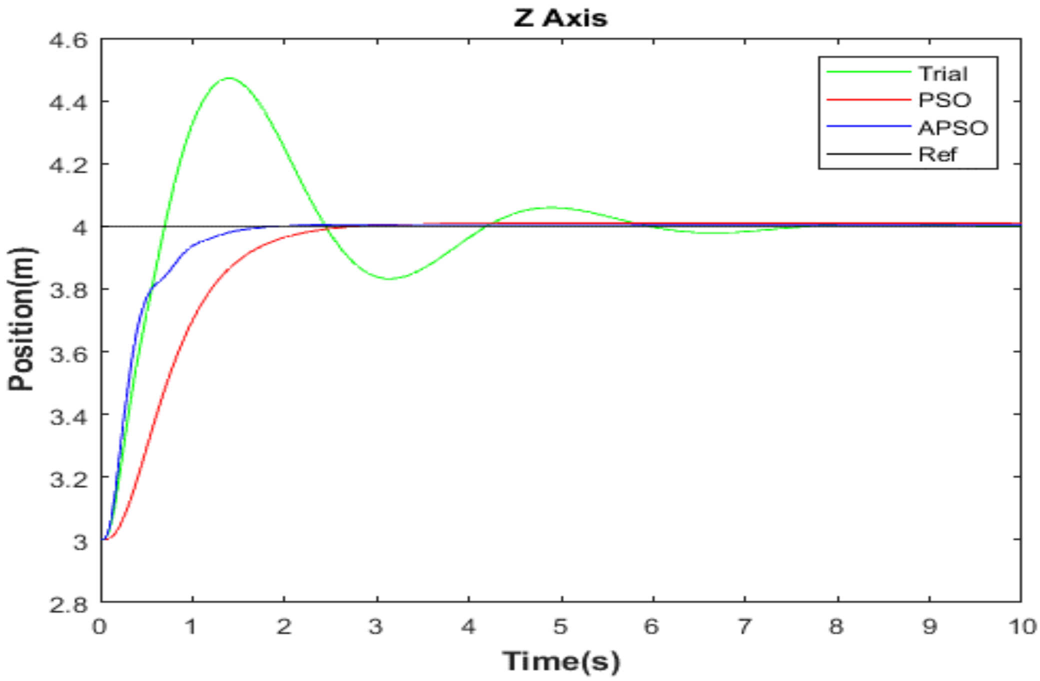

Figure 5. Time Response of the Altitude for the Quadcopter System Using PID Controller.

The time response specification of quadcopter system using PID controllers tuned by trial and error, standard PSO, and the proposed adaptive PSO algorithm are given in Tables 1-4.

Table 1. Time Response Specification for the Roll Angle of the Quadcopter System.

Roll Angle (Φ)	Trial and Error	Standard PSO	Adaptive PSO
Settling time(s)	11.1	2.15	1.42
Rise time(s)	1.14	1.24	0.7
Overshoot (%)	8.68	0.64	0.23

Steady state error	0	0	0

Table 2. Time Response Specification for the Pitch Angle of the Quadcopter System.

Pitch Angle (θ)	Trial and Error	StandardPSO	Adaptive PSO
Settling time(s)	1.86	1.55	1.53
Rise time(s)	0.15	0.09	0.88
Overshoot (%)	36.85	12.87	1.21
Steady state error	0	0	0

Table 3. Time Response Specification for the Yaw Angle of the Quadcopter System.

Yaw Angle (ψ)	Trial and Error	Standard PSO	Adaptive PSO
Settling time(s)	9.27	6.14	1.37
Rise time(s)	1.16	0.75	0.65
Overshoot (%)	33.43	12.19	0.48
Steady state error	0	0	0

Table 4. Time Response Specification for the Altitude Z of the Quadcopter System.

Altitude Position (Z)	Trial and Error	Standard PSO	Adaptive PSO
Settling time(s)	6.82	2.22	1.39
Rise time(s)	0.47	1.24	0.72
Overshoot (%)	11.78	0.23	0.16
Steady state error	0	0	0

4. CONCLUSIONS AND DISCUSSION

In this study, a PID controller is designed and tuned in order to stabilize the quadcopter system. PID controller is tuned by three strategies which are trial-and-error, standard particle swarm optimization (PSO), and the proposed adaptive particle swarm optimization (APSO). The simulation results show that the suggested adaptive PSO algorithm gives the best convergence speed and permanent movement toward the optimal solution region.

The suggested adaptive particle swarm optimization algorithm (APSO) depends on the fitness values of particles, pbest particles, and gbest particle to do an automatic control of inertia weight. The inertia weight in the PSO algorithm does balancing between the global and local search processes. Inertia weight must be large in the global search process and low in the local search process.

From the performance indices, the proposed adaptive PSO strategy gives the best PID parameters to control the roll, pitch, yaw, and altitude that offers a response with the lowest overshoot, settling time and has no steady-state error. The tuning process of the PID controllers using trial and error strategy has the worst response specifications. The standard PSO strategy to tune the PID controllers gives acceptable responses to control of the roll, pitch, yaw, and altitude. Besides, there is no steady-state error for all PID controllers using the three strategies.

REFERENCES

ALHASAN, H.A. and GÜNEŞ, M., 2017. A New Adaptive Particle Swarm Optimization Based on Self-Tuning of PID Controller for DC Motor System. Çukurova University Journal of the Faculty of Engineering and Architecture, 32(3), pp.243-249.

Bansal, J.C., Singh, P.K., Saraswat, M., Verma, A., Jadon, S.S. and Abraham, A., 2011, October. Inertia weight strategies in particle swarm optimization. In 2011 Third world congress on nature and biologically inspired computing (pp. 633-640). IEEE.

Berber, Ö., Muharrem, A.T.E.Ş., ALHASSAN, H.A. and Güneş, M., 2016. Parçacık Sürü Optimizasyonu ve PID ile Mobil Robotun Optimum Yörünge Kontrolü. Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi, 19(3), pp.165-169.

Bolandi, H., Rezaei, M., Mohsenipour, R., Nemati, H. and Smailzadeh, S.M., 2013. Attitude control of a quadrotor with optimized PID controller. Intelligent Control and Automation, 4(03), p.335.

ÇOBAN, R. and ERÇİN, Ö., 2012. Multi-objective Bees Algorithm to Optimal Tuning of PID Controller. Çukurova Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 27(2), pp.13-26.

Eberhart, R. and Kennedy, J., 1995, October. A new optimizer using particle swarm theory. In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science (pp. 39-43). Ieee.

Gündoğdu, Ö., 2005. Optimal-tuning of PID controller gains using genetic algorithms. Journal of Engineering Sciences, 11(1), pp.131-135.

Luukkonen, T., 2011. Modelling and control of quadcopter. Independent research project in applied mathematics, Espoo, 22.

Mohammed, M.J., Rashid, M.T. and Ali, A.A., 2014. Design optimal PID controller for quad rotor system. International Journal of Computer Applications, 106(3), pp.15-20.

Shakhatreh, H., Sawalmeh, A., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N.S., Khreishah, A. and Guizani, M., 2018. Unmanned aerial vehicles: A survey on civil applications and key research challenges. arXiv preprint arXiv:1805.00881.