Journal of Mechanical, Civil and Industrial Engineering
ISSN: 2709-0809
DOI: 10.32996/jmcie
Journal Homepage: www.al-kindipublisher.com/index.php/jmcie

JMCIE
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Particle Swarm Optimized Robust Backstepping Control of a Quadrotor Unmanned Aerial Vehicle under Pink Noise

Mehmet Karahan
Assistant Professor, Electrical and Electronics Engineering, TOBB University of Economics and Technology, Ankara, Turkiye
Corresponding Author: Mehmet Karahan, **E-mail**: mehmetkarahan@etu.edu.tr

| ABSTRACT

Technological developments in sensors, actuators, and energy storage devices have allowed the development of quadrotor unmanned aerial vehicles (UAVs). Quadrotor UAVs are used in sensitive tasks such as surveillance, search and rescue, mapping, mining, cargo carriage, agricultural spraying, firefighting, and photography. Quadrotor UAVs are exposed to effects such as noise and vibration while performing these sensitive tasks. Therefore, robust controller design that is resistant to noise and vibration gains great importance. Noise and vibration can be caused by the sensors, actuator, and propellers of the quadrotor. Background noise in electronic devices is called pink noise. The primary sources of pink noise in electronic devices are generally slow fluctuations of the properties of the condensed matter materials of the devices. These contain fluctuating defect configurations in metals, fluctuating trap occupancy in semiconductors, and fluctuating field structures in magnetic materials. In this study, a particle swarm optimized (PSO) robust backstepping controller is designed for a quadrotor that can follow altitude and attitude references under pink noise. The rise time, overshoot, and settling time of the PSO-optimized proposed backstepping controller and classical PID controller were compared. It has been proven by simulations that the designed PSO-optimized backstepping controller performs more successfully than the classical PID controller.

| KEYWORDS

Quadrotor, Unmanned Aerial Vehicle, PID Control, Backstepping Control, Particle Swarm Optimization, Pink Noise

| ARTICLE INFORMATION

ACCEPTED: 10 June 2025 **PUBLISHED:** 05 July 2025 **DOI:** 10.32996/jmcie.2025.6.3.6

1. Introduction

The latest technological developments in the hardware and software industry have expanded the use of quadrotor UAVs [1], [2]. Today, quadrotor UAVs are used in a wide range of areas such as surveillance, search and rescue, combating natural disasters, mining exploration, agricultural spraying, cargo transportation, organ transportation between hospitals, and combating terrorism [3], [4], [5]. Robust controller design is needed for the quadrotor to successfully perform the given sensitive tasks. While designing the controller, it should be considered that the quadrotor will have to work in non-ideal environments and will be exposed to factors such as noise and vibration [6]. Thus, robust controller design that is resistant to noise and vibration has great importance. Noise and vibration can be caused by the sensors, actuator, and propellers of the quadrotor [7].

Background noise in electronic gadgets is named pink noise [8]. The main sources of the pink noise in electronic gadgets are almost always the slow fluctuations of properties of the condensed matter materials of the gadgets [9], [10]. The pink noise is a signal with a frequency spectrum such that the power spectral density is inversely proportional to the frequency of the signal. In pink noise, each octave interval carries an equal amount of noise energy. The pink noise name arises from the pink appearance of visible light with this power spectrum. This is in contrast with white noise, which has equal intensity per frequency interval. In scientific literature, the term 1/f noise is used to refer to any noise with a power spectral density of the form 1/fa, where f is frequency and a is between 0 and 2 [11]. The canonical case with α = 1 is named pink noise.

This study is the expanded and full-paper version of the Abstract titled "Robust backstepping control of a quadrotor unmanned aerial vehicle under pink noise" presented at the Modern Practice in Stress and Vibration Analysis Conference held at St Anne's College, Oxford University in 2022 [12]. In this study, a PSO-optimized robust backstepping controller design has been developed for the quadrotor. The designed controller can follow the altitude and attitude references given under pink noise. The rise time, overshoot, and settling time data of the proposed backstepping controller were compared with the rise time, overshoot, and settling time data of the PID controller. As a result of the comparisons, it has been proven that the proposed backstepping controller is more successful than the PID controller, which is widely used in industry. Section 2 describes the quadrotor model, Section 3 explains the classical PID controller, Section 4 gives the PSO-optimized backstepping controller, Section 5 explains the simulations, and finally, Section 6 provides the conclusion.

2. The quadrotor model

Quadrotors have four propellers that can take off vertically, hover in the air, and rotate around their axis [13]. The two motors of the quadrotor rotate clockwise, while the two motors rotate counterclockwise [14]. Quadrotor has six degrees of freedom. It means that six variables are required to explain its position and orientation in space. Translational movements consist of x, y, z, and rotational movements consist of roll (φ), pitch (θ), and yaw (ψ) angles. Figure 1 represents the quadrotor mechanism [15].

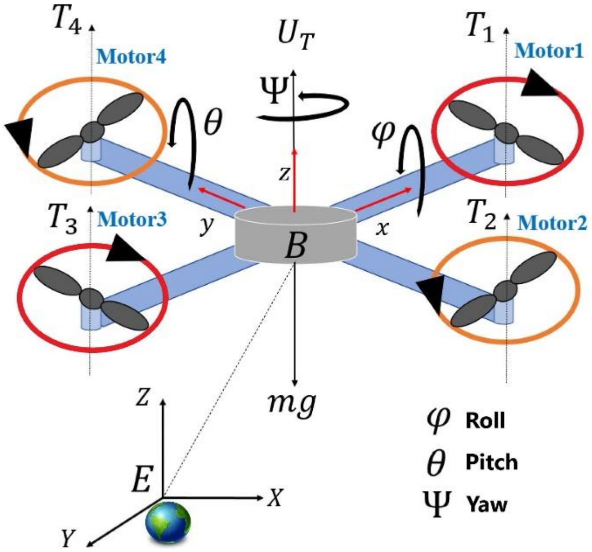

Figure 1. The quadrotor mechanism and coordinate system.

The Earth frame E and body frame B are defined to describe the flight condition of quadrotor. Rotation matrix is used for transformation between Earth frame axis and body frame axis. Equation (1) gives the rotation matrix. In equation (1), c is the abbreviation for cosine, and s is the abbreviation for sine.

$$R = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\theta s\varphi c\psi - s\psi c\varphi & s\psi s\theta s\varphi + c\psi c\varphi & s\varphi c\theta \\ s\theta c\varphi c\psi + s\psi s\varphi & s\psi s\theta c\varphi - c\psi s\varphi & c\varphi c\theta \end{bmatrix} \tag{1}$$

Equation (2) explains the transition from Euler angle rates to body angular rates. Equation (3) gives the transition from body angular rates to Euler angle rates.

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sec\theta\sin\varphi & \sec\theta\cos\varphi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \cos\theta\sin\varphi \\ 0 & -\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \begin{bmatrix} \varphi' \\ \theta' \\ \psi' \end{bmatrix} \tag{3}$$

F_i gives force produced by the ith rotor and T_i represents torque generated by the ith rotor. In below equations, b is thrust and d is drag coefficient.

$$F_i = bw_i^2 \tag{4}$$

$$T_i = dw_i^2 \tag{5}$$

Equation (6) represents the control inputs U_1, U_2, U_3 and U_4. In below equation, l is the arm length of the quadrotor and w_1, w_2, w_3, w_4 represent angular velocities of motors. Equation (7) gives the transformation between the square of the angular velocities and the control inputs.

$$u = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} F \\ T_\varphi \\ T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ lb & 0 & -lb & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b} & 0 & \frac{1}{2bl} & -\frac{1}{4d} \\ \frac{1}{4b} & -\frac{1}{2bl} & 0 & \frac{1}{4d} \\ \frac{1}{4b} & 0 & -\frac{1}{2bl} & -\frac{1}{4d} \\ \frac{1}{4b} & \frac{1}{2bl} & 0 & \frac{1}{4d} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{7}$$

A dynamic model of the quadrotor due to translational and rotational movements is described below.

$$\ddot{X} = (\cos\varphi\cos\psi\sin\theta + \sin\varphi\sin\psi)\frac{U_1}{m} \tag{8}$$

$$\ddot{Y} = (\cos\varphi\sin\psi\sin\theta - \sin\varphi\cos\psi)\frac{U_1}{m} \tag{9}$$

$$\ddot{Z} = -g + (\cos\theta\cos\varphi)\frac{U_1}{m} \tag{10}$$

$$\ddot{\varphi} = \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) + \dot{\theta}w_r\frac{J_R}{I_x} + \frac{l}{I_x}U_2 \tag{11}$$

$$\ddot{\theta} = \dot{\theta}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \dot{\varphi}w_r\frac{J_R}{I_x} + \frac{l}{I_y}U_3 \tag{12}$$

$$\ddot{\psi} = \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \tag{13}$$

Table 1 represents the parameters of the quadrotor UAV.

Table 1. Parameters of the quadrotor UAV.

Constant	Definition	Value
m	Mass of the quadrotor	0.65 kg
g	Gravity	9.81 m/s^2
l	Arm length of the quadrotor	0.23 m
b	Thrust coefficient	3.13 x 10^{-5} Ns^2
d	Drug coefficient	7.5 x 10^{-7} Nms^2
Ix	Inertial moment along x axis	7.5 x 10^{-3} kgm^2
Iy	Inertial moment along y axis	7.5 x 10^{-3} kgm^2
Iz	Inertial moment along z axis	1.3 x 10^{-2} kgm^2
Jr	Moment of inertia of propellers	6.5 x 10^{-5} kgm^2
w_{max}	Maximum motor speed	1000 rad/sec
t_{max}	Maximum torque	0.15 Nm

The general diagram of the quadrotor model created in the MATLAB program is given in Figure 2.

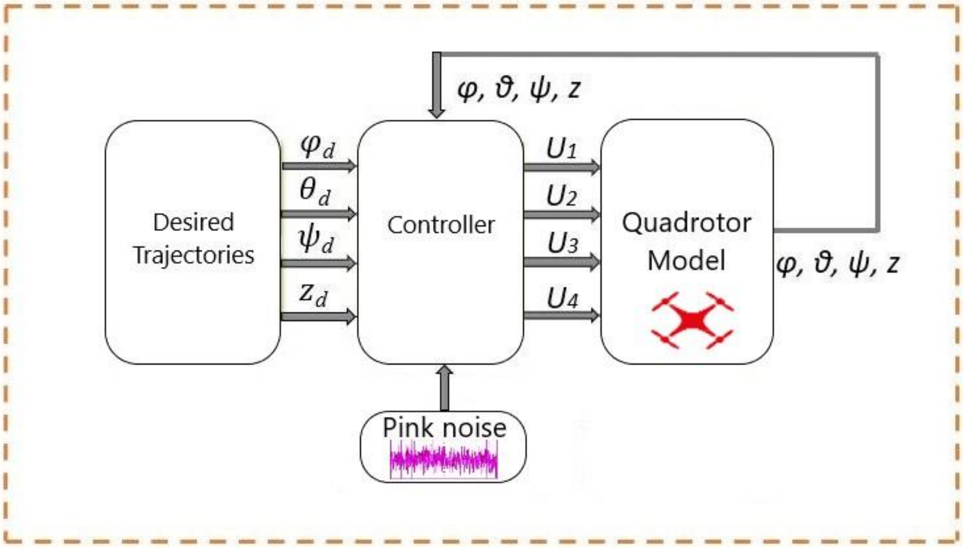

Figure 2. The general structure of the system.

3. Classical PID Controller

The PID controller is widely used in industry due to its simple structure and easy adjustment of parameters. A PID controller has 3 parameters: K_p, K_i, and K_d. These parameters are used in controller inputs. U_1, U_2, U_3, and U_4 controller inputs are given in equations (14) to (17). U_1 controls altitude, U_2 controls roll angle, U_3 controls pitch angle, and U_4 controls yaw angle.

$$U_1 = \frac{m*(g+ \mathrm{e_z}Kp + Ki \int \mathrm{e_z}\mathrm{dt}+ Kd\frac{\mathrm{de_z}}{\mathrm{dt}})}{\cos\Phi\cos\theta} \tag{14}$$

$$U_2 = \mathrm{e_\Phi}Kp + Ki \int \mathrm{e_\Phi}\,\mathrm{dt} + Kd\frac{\mathrm{de_\Phi}}{\mathrm{dt}} \tag{15}$$

$$U_3 = \mathrm{e_\theta}Kp + Ki \int \mathrm{e_\theta}\,\mathrm{dt} + Kd\frac{\mathrm{de_\theta}}{\mathrm{dt}}) \tag{16}$$

$$U_4 = \mathrm{e_\psi}Kp + Ki \int \mathrm{e_\psi}\,\mathrm{dt} + Kd\frac{\mathrm{de_\psi}}{\mathrm{dt}}) \tag{17}$$

Table 2 indicates the PID controller coefficients used to control the altitude and attitude references. PID controller coefficients are calculated with the PID Tuner tool in MATLAB. In the PID Tuner tool, Transfer Function Based tuning is selected. PID Tuner automatically linearizes the plant at the operating point, and it computes an initial compensator design for the linearized plant model.

Table 2. Parameters of the PID controller.

Parameter	Altitude	Roll	Pitch	Yaw
K_p	0.82	0.12	0.14	0.13
K_i	0.1	0.05	0.07	0.05
K_d	1.65	0.06	0.08	0.1

4. Backstepping Controller

In this section, robust backstepping controller design is explained. The backstepping technique is developed for designing stabilizing controls for nonlinear dynamical systems. It depends on a recursive design procedure which links the selection of a Lyapunov function with the feedback controller design and allows a strict feedback system to reach global asymptotic stability. Due to the recursive structure, the designer could start designing at the known-stable system and "back out" new controllers that gradually balance each outer subsystem. This process ends when the last external control is obtained. Therefore, this process is named as backstepping. In our research, we combined Lyapunov's direct method with adaptive control laws. Firstly, we defined the tracking error as in equation (18).

$$z_1 = \Phi_d - \Phi \tag{18}$$

Lyapunov function of this variable is given in equation (19).

$$V(z_1) = \frac{1}{2} z_1^2 \tag{19}$$

The time derivative of the Lyapunov function is given in equation (20).

$$\dot{V}(z_1) = z_1(\dot{\Phi}_d - \dot{\Phi}) \tag{20}$$

Since the Lyapunov function's time derivative should be negative semidefinite, a new virtual control input $\dot{\Phi}$ is defined for the stabilization of z_1 function as below.

$$\dot{\Phi} = \dot{\Phi}_d + a_1 z_1 \tag{21}$$

a_1 should be a positive constant to guarantee negative semi-definiteness. When the virtual control input is substituted in equation (20), we get equation (22).

$$\dot{V}(z_1) = - a_1 z_1^2 \tag{22}$$

The other variable change is done as follows:

$$z_2 = \dot{\varphi} - \dot{\varphi}_d - a_1 z_1 \tag{23}$$

After the variable changes, the Lyapunov function can be rewritten as follows:

$$V(z_1, z_2) = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 \tag{24}$$

Time derivative of the above function can be written in equation (25).

$$\dot{V}(z_1, z_2) = - a_1 z_1^2 - z_1 z_2 + z_2 \ddot{\varphi} - z_2(\ddot{\varphi}_d - a_1(z_2 + a_1 z_1)) \tag{25}$$

According to the equation (11), $\ddot{\Phi}$ variable can be rewritten as follows:

$$\ddot{\Phi} = \dot{\psi}\dot{\theta} a_1 + a_2 \dot{\theta} w_r + \frac{l}{Ix} U_2 \tag{26}$$

The U_2 control input is described as follows under $\ddot{\varphi}$, $\dddot{\varphi}$, $\ddot{\theta}_d = 0$ and $\dot{V}(z_1, z_2) < 0$ conditions.

$$U_2 = \frac{Ix}{l}(z_1 - a_1 \dot{\theta}\dot{\psi} - a_2 \dot{\theta} w_r - a_1(z_2 + a_1 z_1) - a_2 z_2) \tag{27}$$

To stabilize z_1, the $a_2 z_2$ term with $a_2 > 0$ is added. Based on the same steps, U_3 and U_4 control inputs that control pitch and yaw angles are obtained as follows:

$$U_3 = \frac{Iy}{l}(z_3 - a_3 \dot{\Phi}\dot{\psi} - a_4 \dot{\Phi} w_r - a_3(z_4 + a_3 z_3) - a_4 z_4) \tag{28}$$

$$U_4 = \frac{Iz}{1}(z_5 - a_5 \dot{\Phi}\dot{\theta} - a_5(z_6 + a_5 z_5) - a_6 z_6) \tag{29}$$

The variables used in U_3 and U_4 control inputs are defined below in equations.

$$z_3 = \theta_d - \theta \tag{30}$$

$$z_4 = \dot{\theta} - \dot{\theta}_d - a_3 z_3 \tag{31}$$

$$z_5 = \psi_d - \psi \tag{32}$$

$$z_6 = \dot{\psi} - \dot{\psi}_d - a_5 z_5 \tag{33}$$

The tracking error for altitude control is defined as z_7 in equation (34).

$$z_7 = z\text{-}z_d \tag{34}$$

The Lyapunov function of this tracking error is given in equation (35).

$$V(z_7) = \frac{1}{2} z_7^2 \tag{35}$$

Time derivative of this Lyapunov function is represented in equation (36).

$$\dot{V}(z_7) = z_7 (\dot{z}_d - \dot{z}) \tag{36}$$

The virtual control input x_8 is defined as in equation (37) to stabilize the function of z_7.

$$x_8 = \dot{z}_d + a_7 z_7 \tag{37}$$

The second variable change is made as in equation (38).

$$z_8 = x_8 - \dot{z}_d - a_7 z_7 \tag{38}$$

After the variable changes, the new Lyapunov function can be defined as follows:

$$V(z_7, z_8) = \frac{1}{2} z_7^2 + \frac{1}{2} z_8^2 \tag{39}$$

Time derivative of this new Lyapunov function is represented in below equation (40).

$$\dot{V}(z_7, z_8) = - a_7 z_7^2 - z_7 z_8 + z_8 x_8 - z_8(\ddot{z}_d - a_7(z_8 + a_7 z_7)) \tag{40}$$

The derivative of the virtual control input x_8 is given in equation (41).

$$\dot{x}_8 = g\text{-} \cos\theta\cos\Phi \frac{U_1}{m} \tag{41}$$

The U_1 controller input that controls the altitude reference can be defined as in the equation below.

$$U_1 = \frac{m}{\cos\theta\cos\varphi} (z_7 + g - a_7 (z_8 + a_7 z_7) - a_8 z_8) \tag{42}$$

The parameters of the backstepping controller are given in Table 3. The parameters of the backstepping controller should be positive to satisfy the stability criteria. Particle Swarm Optimization (PSO) is used to select optimum parameters. In the PSO system, particles change their positions by flying around in a multidimensional search space. A particle in a swarm adjusts its position in the search space using its present velocity, its own previous experience, and that of neighboring particles [16]. The following intuitive formulation is used [17].

$$v_i^{k+1} = w^k v_i^k + a_1 \gamma_{1,i} (P_i - x_i^k) + a_2 \gamma_{2,i} (G - x_i^k) \tag{43}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{44}$$

The x_i and v_i^k vectors are current position and velocity of the ith particle and kth generation. The swarm includes N particles, i.e. i = {1, … , N}. Moreover, P_i is personal best position of each individual and G is global best position of observed among all particles up to the current generation. The $\gamma_{1,2} \in [0, 1]$ parameters are uniformly distributed random values and $a_{1,2}$ represent acceleration constants. The w function represents particle inertia that gives rise to a certain momentum of particles.

Table 3. Parameters of the backstepping controller.

Roll angle controller (a_1, a_2)	Pitch angle controller (a_3, a_4)	Yaw angle controller (a_5, a_6)	Altitude controller (a_7, a_8)
(8.7, 7)	(8, 4)	(8.5, 4.1)	(1.5, 6)

5. Simulations

In this section, simulations for controlling altitude and attitude angles of quadrotor UAV under pink noise are given. Pink noise is generated with a colored noise block in MATLAB. It has one output channel and its data type is double. Number of samples per output channel is 1 and output sample time is 0.1 second. Initial seed value is 5 for random number generator. Pink noise is added to the end of the controller block before U input. As an example, the block diagram of pink noise added to the yaw angle PID controller is given in Figure 3.

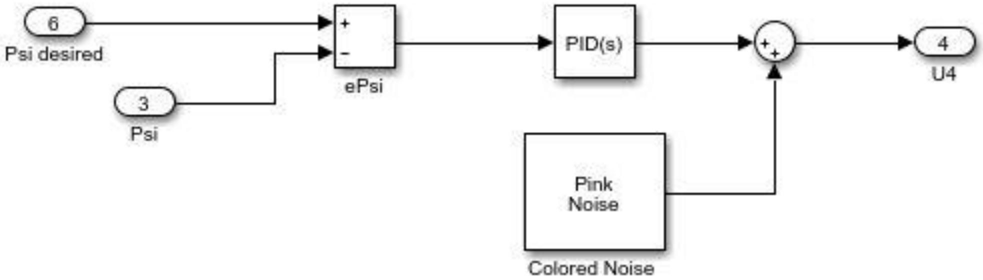

Figure 3. The general structure of the system.

The rise time, overshoot, and settling time values of the proposed backstepping controller and the PID controller, which is widely used in the industry, were compared. In this way, the superiority of the proposed backstepping controller is demonstrated. Figure 4 shows simulations for roll angle controllers, and Figure 5 shows simulations for pitch angle controllers. Figures 6 and 7 show simulations for yaw angle controllers and altitude controllers, respectively. In the graphs, the green line indicates the reference, the blue line indicates the PID controller, and the red line indicates the backstepping controller. In simulations, angle references are in radians, altitude references are in meters, and time is in seconds. Grid is used in graphics.

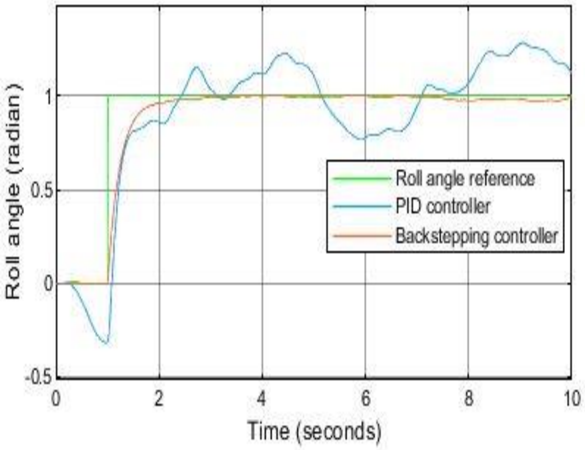

Figure 4. Simulation of the roll angle controllers.

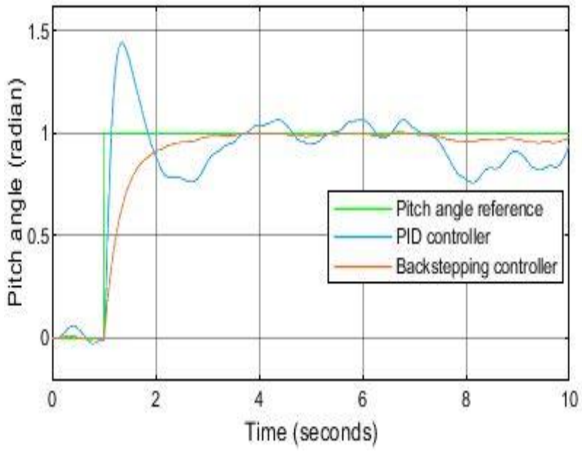

Figure 5. Simulation of the pitch angle controllers.

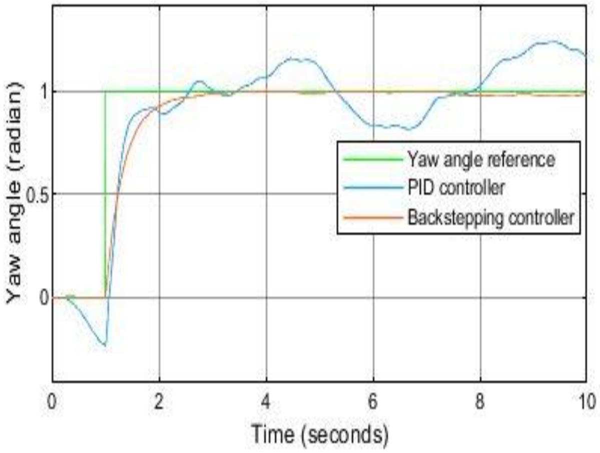

Figure 6. Simulation of the yaw angle controllers.

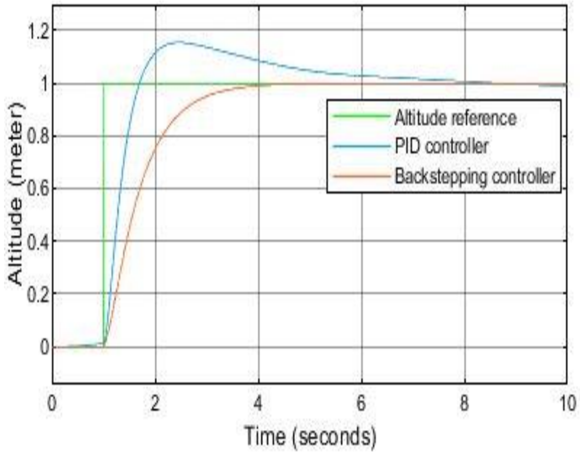

Figure 7. Simulation of the altitude controllers.

Table 4 gives a comparison of the time response of PID and backstepping controllers. Rise time is the time required for the response to rise from 10% to 90% of its final value. Settling time is defined as the time required for the response curve to reach and stay within 5% of the final value. Overshoot is the maximum peak value of the response curve measured from the desired response of the system.

Table 4. Time response comparison of the PID and backstepping controllers.

Controller type	Rise time (s)	Overshoot (%)	Settling time (s)
Roll angle PID controller	1.45	23.28	-----
Roll angle backstepping controller	1.39	0.22	1.83
Pitch angle PID controller	1.12	61.47	-----
Pitch angle backstepping controller	1.5	1.23	2.30
Yaw angle PID controller	1.40	8.30	-----
Yaw angle backstepping controller	1.56	0.34	2.16
Altitude PID controller	1.47	15	4.93
Altitude backstepping controller	2.14	0	2.99

When the roll angle reference tracking is examined, the backstepping controller shows 0.22% overshoot, while the PID controller shows 23.28% overshoot. While PID controller does not have a stable settling time, the backstepping controller catches the given reference in 1.83 seconds. When the pitch angle simulation is examined, the backstepping controller shows 1.23% overshoot, while the PID controller shows 61.47% overshoot. There is no settling time as the PID controller cannot reach the 5% range of the given reference. On the other hand, the backstepping controller has a settling time of 2.3 seconds and can follow the given reference. When the yaw angle data in Table 4 is examined, it is understood that the backstepping controller overshoots 0.34% and the PID controller overshoots 8.30%. Since the PID controller cannot stably follow the given reference, there is no settling time. The backstepping controller has a settling time of 2.16 seconds and successfully follows the reference. When the altitude simulation data is examined, it is seen that the backstepping controller shows no overshoot, and the PID controller shows 15% overshoot. PID controller has a long settling time of 4.93 seconds, while the backstepping controller has a shorter settling time of 2.99 seconds. Although the rise time of PID controller is slightly faster than the backstepping controller in simulations, it cannot follow references because it has a very high overshoot and does not have a settling time. Simulations have proven that the backstepping controller is more robust against noise and successfully follows the references.

6. Conclusions

In this research, a PSO-optimized robust backstepping controller design for altitude and attitude control of a quadrotor under pink noise is explained. The performance of the proposed backstepping controller is compared with that of the PID controller. The time responses of the controllers were compared. Backstepping controller shows less overshoot than the classical PID controller. In addition, the backstepping controller has a short settling time. Although the PID controller has a fast rise time, it shows a very high overshoot and cannot reach a stable settling time value because it is affected by noise too much. As a result of the simulations, the proposed backstepping controller has been proven to be more robust than the classical PID controller.

Funding: This research received no external funding.
Conflicts of Interest: The author declares no conflict of interest.
ORCID iD 0000-0002-5828-497X

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Ahmed, F., Mohanta, J. C., Keshari, A., & Yadav, P. S. (2022). Recent advances in unmanned aerial vehicles: a review. *Arabian Journal for Science and Engineering*, *47*(7), 7963-7984.

[2] Bouabdallah, S., & Siegwart, R. (2005, April). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 2247-2252). IEEE.

[3] Naidoo, Y., Stopforth, R., & Bright, G. (2011, September). Development of an UAV for search & rescue applications. In *IEEE Africon'11* (pp. 1-6). IEEE.

[4] Ma, G., Tong, Z., Tong, M., & Tang, S. (2018, July). Coordinated control of uavs for mine searching. In *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)* (pp. 259-263). IEEE.

[5] Francisco, M. (2016). Organ delivery by 1,000 drones. *Nature Biotechnology*, *34*(7), 684-685.

[6] Karahan, M., & Kasnakoglu, C. (2021). Modeling a Quadrotor Unmanned Aerial Vehicle and robustness analysis of different controller designs under parameter uncertainty and noise disturbance. *Journal of Control Engineering and Applied Informatics*, *23*(4), 13-24.

[7] Guo, J., Qi, J., & Wu, C. (2021). Robust fault diagnosis and fault-tolerant control for nonlinear quadrotor unmanned aerial vehicle system with unknown actuator faults. *International Journal of Advanced Robotic Systems*, *18*(2), 17298814211002734.

[8] Kleinpenning, T. G. M., & De Kuijper, A. H. (1988). Relation between variance and sample duration of 1/f noise signals. *Journal of applied physics*, *63*(1), 43-45.

[9] Kogan, S. (2008). *Electronic noise and fluctuations in solids*. Cambridge University Press.

[10] Weissman, M. B. (1988). 1 f noise and other slow, nonexponential kinetics in condensed matter. *Reviews of modern physics*, *60*(2), 537.

[11] Baxandall, P. J. (1968). Noise in transistor circuits. *Wireless World*, *74*, 388-392.

[12] Karahan, M., Kasnakoglu, C. (2022). "Robust backstepping control of a quadrotor unmanned aerial vehicle under pink noise": In *2022 10th International Conference on Modern Practice in Stress and Vibration Analysis (MPSVA)* [Abstract].

[13] Phang, S. K., Cai, C., Chen, B. M., & Lee, T. H. (2012, July). Design and mathematical modeling of a 4-standard-propeller (4SP) quadrotor. In *Proceedings of the 10th world congress on intelligent control and automation* (pp. 3270-3275). IEEE.

[14] Köse, O., & Oktay, T. (2019). Dynamic Modeling and Simulation of Quadrotor for Different Flight Conditions. *Avrupa Bilim ve Teknoloji Dergisi*.

[15] Ghadiri, H., Khodadadi, H., & Hazareh, G. A. (2024). Finite-time integral fast terminal sliding mode control for uncertain quadrotor UAV based on state-dependent Riccati equation observer subjected to disturbances. *Journal of Vibration and Control*, *30*(11-12), 2528-2548.

[16] Boubertakh, H., Labiod, S., & Tadjine, M. (2012, July). PSO to design decentralized fuzzy PI controllers application for a helicopter. In *2012 20th Mediterranean Conference on Control & Automation (MED)* (pp. 633-637). IEEE.

[17] Ebbesen, S., Kiwitz, P., & Guzzella, L. (2012, June). A generic particle swarm optimization Matlab function. In *2012 American control conference (ACC)* (pp. 1519-1524). IEEE.