Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Reinforcement Learning and PD Control Based Trajectory Tracking for a Quadcopter UAV

Mehmet Karahan
Assistant Professor, Electrical and Electronics Engineering, Atilim University, Ankara, Turkey
Corresponding Author: Mehmet Karahan, **E-mail**: mehmet.karahan@atilim.edu.tr

| ABSTRACT

Nowadays, quadcopter unmanned aerial vehicles (UAV) are used in a wide variety of areas, such as reconnaissance and surveillance, firefighting, search and rescue, agricultural spraying, cargo transportation, photography, and mapping. The use of quadcopters in a very wide area makes their trajectory tracking control important. In order for quadcopters to perform their duties successfully, they must be able to follow the given trajectory with the least error. In this study, the quadcopter's trajectory tracking under random noise is provided by an algorithm based on reinforcement learning and a proportional derivative (PD) controller. Modeling, simulations, and reinforcement learning algorithms were carried out using the MATLAB program. Simulations were made under noise for the x, y, z trajectories and roll, pitch, and yaw angles of the quadcopter. A detailed time response analysis was performed by obtaining rise time, overshoot, and settling time data. It has been observed that the references given were successfully followed thanks to the algorithm based on reinforcement learning.

| KEYWORDS

Quadcopter, Autonomous Aerial Vehicles, Reinforcement Learning, Machine Learning, PD Control, Trajectory, Noise.

| ARTICLE INFORMATION

ACCEPTED: 01 October 2024 **PUBLISHED:** 16 October 2024 **DOI:** 10.32996/jcsts.2024.6.4.15

1. Introduction

Quadcopter UAVs have advantages over winged UAVs because they can take off and land vertically, perform sharp maneuvers, rotate around their own axis, and hover [Xuan, 2022], [Lee, 2022], [Karahan, 2023].

Developments in hardware and software technologies in the last few years have facilitated the manufacturing processes of quadcopters, enhanced their capabilities, and led to the expansion of their usage areas. [Elmeseiry, 2021], [Ahmed, 2022], [Karahan, 2021] Quadcopters are used in different areas such as combating natural disasters, mining, search and rescue operations, cargo transportation, pesticide spraying, combating terrorism, and monitoring [Sabour, 2023], [Chaurasia, 2021], [Karahan, 2022]. Optimum trajectory tracking control is important for quadcopter UAVs to successfully fulfill their missions [Öniz, 2024].

In recent years, many researchers have conducted studies on quadcopter's trajectory tracking [Zare, 2022]. In order to successfully perform trajectory tracking, they have developed controller designs and used various algorithms [Rokonuzzaman, 2021]. Iskander et al. focused on minimum snap trajectory tracking for a quadcopter and used a model predictive controller (MPC) [Iskander, 2020]. Singh et al. (2023) used a proportional derivative (PD) controller with minimum snap trajectory planning for a quadcopter. They used the MATLAB program to generate trajectories in simulations and controller design. Hong et al. (2021) combined an A* search algorithm and minimum snap trajectory generation for the trajectory tracking problem of a quadcopter UAV. Dong et al. (2021) used a heuristic search algorithm to find the optimum trajectory and used minimum instantaneous trajectory optimization to follow the trajectory. Sanwale et al. Realized quaternion-based attitude control of a quadcopter UAV using a 3rd-order sliding mode controller design and disturbance rejection [2020].

However, the above methods require partial or complete knowledge about the environment in which the quadcopter will move. The reinforcement learning method can be applied in dynamic or uncertain environments where no information is known [Tan, 2022]. Yoo et al. (2020) developed a hybrid reinforcement learning-based PD controller and hybrid reinforcement learning-based LQR (linear quadratic regulator) controllers for a micro quadcopter UAV. By performing various simulations, they showed that the hybrid methods they used converged to the given reference orbit faster and exhibited better control performance than standard reinforcement learning methods. Pi et al. (2021) controlled a quadcopter UAV in three axes (x, y, z) using a reinforcement learning algorithm under wind used as an external disturbance. An electric fan was used to generate wind. A 3.6 m/s wind was applied to the center of the x-axis of the quadcopter. The position information of the quadcopter was obtained using GPS. Wen et al. (2021) An optimized tracking control scheme for a quadcopter unmanned aerial vehicle system has been studied by combining both reinforcement learning and backstepping control techniques. Reinforcement learning has been used to overcome the difficulty of solving the Hamilton-Jacobi-Bellman (HJB) equation. In order to achieve optimized position control, reinforcement learning has been established based on the neural network approximation of the solution of the HJB equation by utilizing the extraordinary function approximation ability of the neural network.

In this study, the optimal tracking control problem of a quadcopter with a complex trajectory is solved by off-policy reinforcement learning without any information about the environment. Random noise is added to a complex trajectory, and the quadcopter is made to follow this trajectory through reinforcement learning. Since an irregular trajectory with random noise is used, there is no prior knowledge about the trajectory to be followed. First, the data-driven policy iteration algorithm is explained. Then, this algorithm is applied to the quadcopter. Trajectory tracking simulations are performed with MATLAB for x, y, z positions, and roll, pitch, yaw angles. Simulations show that trajectory tracking is successfully achieved thanks to the developed algorithm.

This study extends the works of the conference paper titled “Optimal Trajectory Tracking Control for a Quadrotor UAV Based on Off-Policy Reinforcement Learning” [Karahan, 2024]. The Introduction section has been expanded by conducting a more comprehensive literature review. The reinforcement learning algorithm for optimal trajectory tracking has been explained in more detail theoretically, and more equations have been included. The control scheme of the quadcopter has been explained in more detail. A large number of new simulations have been performed for attitude and position control.

2. Quadcopter Modeling and Data-Driven Policy Iteration

In this part, the reinforcement learning based data-driven policy iteration algorithm is described. First, reinforcement learning is explained for the optimal trajectory tracking problem. Thereafter, the dynamic model of the quadcopter is given. After, the control diagram of the quadcopter is given. Afterwards, the application of the data-driven policy iteration algorithm for the quadcopter is explained. Lastly, simulations for the attitude and position control are shown.

2.1 Reinforcement Learning for Optimal Trajectory Tracking

Within the framework of the proposed reinforcement learning method, the quadcopter model is transformed into a nonlinear affine form, and the optimal tracking problem is solved. The nonlinear affine system can be expressed as in equation (1) below.

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{1}$$

In the above equation, $x = [x_1 ... x_n]^T \in R^n$, $u = [u_1 ... u_m] \in R^m$, $f(x(t)) \in R^n$, $g(x(t)) \in R^{n\times m}$ is defined. It is assumed that $f(0) = 0$ and $f(x(t)) + g(x(t))u(x(t))$ satisfies the Lipschitz condition for continuity in the field $\Omega \subseteq R^n$.

If $f(x)$ and $g(x)$ are known and $g^{-1}(x)$ is available, the control signal $u_d(t)$ for the desired orbit can be written as in equation (2).

$$u_d(t) = g^{-1}(x_d(t))(\dot{x}_d(t) - f(x_d(t))) \tag{2}$$

Kamalapurkar et al. (2015) used the feed-forward method to solve the optimal trajectory tracking problem. In this method, they optimized the cost function using the feedback signal $u_e(t) = u(t) - u_d(t)$ and obtained equation (3).

$$V_e(e_d(t)) = \int_t^\infty t(e_d(\tau)TQ_e e_d(\tau) + u_e(\tau)TRu_e(\tau))d\tau \tag{3}$$

This cost function can be obtained by solving the HJB equation.

$$u_e^* = -\tfrac{1}{2}[R^{-1}g(x(t))^T \partial V_e(e_d(t))/\partial e_d] \tag{4}$$

The definitions used in the optimal trajectory tracking control problem with reinforcement learning are given in equations (5) and (6). $e_d(t)$ finds the error by giving the difference between the actual trajectory x(t) and the desired trajectory $x_d(t)$.

$$e_d(t) = x(t) - x_d(t) \tag{5}$$

$$V(e_d(t), x_d(t)) = \int_t^\infty e^{-\lambda(\tau-t)}(e_d(\tau)^T Q_e e_d(\tau) + u(\tau)^T R u(\tau))d\tau \tag{6}$$

In equations (5) and (6), $Q_e \in \mathrm{R}^{n\times n} \succ 0$ and $R \in \mathrm{R}^{m\times m} \succ 0$ are defined.

The derivative of the desired orbit is given in equation (7).

$$\dot{x}_d(t) = r_d(x_d(t)),\ r_d(0) = 0 \tag{7}$$

In the equation given above, $x_d(t)$ is bounded, and $r_d(x_d(t)) \in \mathrm{R}^n$ is a continuous Lipschitz function. Data-driven policy iteration algorithm can solve the system with unknown dynamics. Equation (8) is obtained by using $X(t) = [e_d(t)^T\ x_d(t)^T]^T \in X \subset \mathrm{R}^{2n}$ and equation (1).

$$\dot{X}(t) = F(X(t)) + G(X(t))u(t) \tag{8}$$

F(X(t)) and G(X(t)) are defined as in equations (9) and (10) below.

$$F(X(t)) = \begin{bmatrix} f(e_d(t) + x_d(t) - r_d(x(t)) \\ r_d x_d(t) \end{bmatrix} \tag{9}$$

$$G(X(t)) = \begin{bmatrix} g(e_d(t) + x_d(t)) \\ 0 \end{bmatrix} \tag{10}$$

The cost function equation is determined as follows.

$$V(X(t)) = \int_t^\infty e^{-\lambda(\tau-t)}\ (X(\tau)^T Q X(\tau) + u(\tau)^T R u(\tau))d\tau \tag{11}$$

$$Q = \begin{bmatrix} Q_e & 0 \\ 0 & 0 \end{bmatrix} \tag{12}$$

The decay coefficient λ needs to be used in the cost function. If the desired trajectory does not go to zero as time (t) approaches infinity, the cost function with $\lambda = 0$ will be unbounded because the feedforward component $u_d(t)$ depends on the desired state trajectory. To overcome this drawback, the component $e^{-\lambda(\tau-t)}$ is added to handle the above cases; for example, this component is used in cases where the desired trajectory is a finite cyclic signal [Modares, 2014], [Modares, 2014], [Kiumarsi, 2014].

2.2 Dynamic Model of the Quadcopter

In this section, the dynamic model of the quadcopter is presented, and the relevant equations are presented. Figure 1 describes the dynamic model of the quadcopter [Pham, 2022].

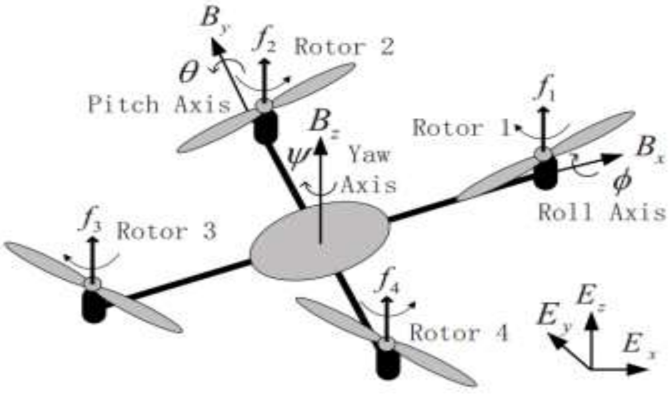

Fig. 1 Dynamic model of the quadcopter

In Figure 1, E_x, E_y, and E_z represent the Earth centered coordinates. B_x, B_y, and B_z represent the body centered coordinates. The roll axis is specified by the roll axis, the pitch axis is specified by the yaw axis, and the yaw axis is specified by the yaw axis [Karahan, 2023]. The position state variables are presented in (13), and the Euler angles state variables are shown in (14).

$$p = [p_x, p_y, p_z]^T \in R^3 \tag{13}$$

$$\Theta = [\phi, \theta, \psi]^T \in R^3 \tag{14}$$

The dynamic model is presented in equations (15) and (16).

$$m\ddot{p} = fRe_3^3 - mge_{3,3} \tag{15}$$

$$J\ddot{\theta} = \tau - C(\Theta, \dot{\theta})\dot{\theta} \tag{16}$$

$T_p \in R$ symbolizes the total lift force of the rotors on the body centered coordinates. Equation (17) shows the total lift force. The torque equation is given in (18). In the following equations, k_w and k_t are the aerodynamic constants, m represents the mass of the quadcopter, and g symbolizes the acceleration due to gravity.

$$T_p = T_1 + T_2 + T_3 + T_4 = k_w u_z \tag{17}$$

$$\tau = [l_\tau k_w u_\phi, l_\tau k_w u_\theta, k_t u_\psi] \tag{18}$$

J_x, J_y, and J_z are the inertial moments about the x, y, and z axes. Equation (19) shows the diagonal matrix of moments of inertia.

$$J = diag(J_x, J_y, J_z) \tag{19}$$

The input signals, depending on the rotation speed of the rotors, are given from (20) to (23), where w_j(j = 1, 2, 3, 4) represents the rotation speed of the corresponding jth rotor blade.

$$u_z = w_1^2 + w_2^2 + w_3^2 + w_4^2 \tag{20}$$

$$u_\phi = w_2^2 - w_4^2 \tag{21}$$

$$u_\theta = w_1^2 - w_3^2 \tag{22}$$

$$u_\psi = w_1^2 - w_2^2 + w_3^2 - w_4^2 \tag{23}$$

2.3 Control Diagram and Data-Driven Policy Iteration Algorithm

Quadcopter is a system with 6 degrees of freedom and four control inputs, namely $[p_x, p_y, p_z, \phi, \theta, \psi]^T$ and $[u_z, u_\phi, u_\theta, u_\psi]^T$. The typical control scheme of the quadcopter is shown in Figure 2.

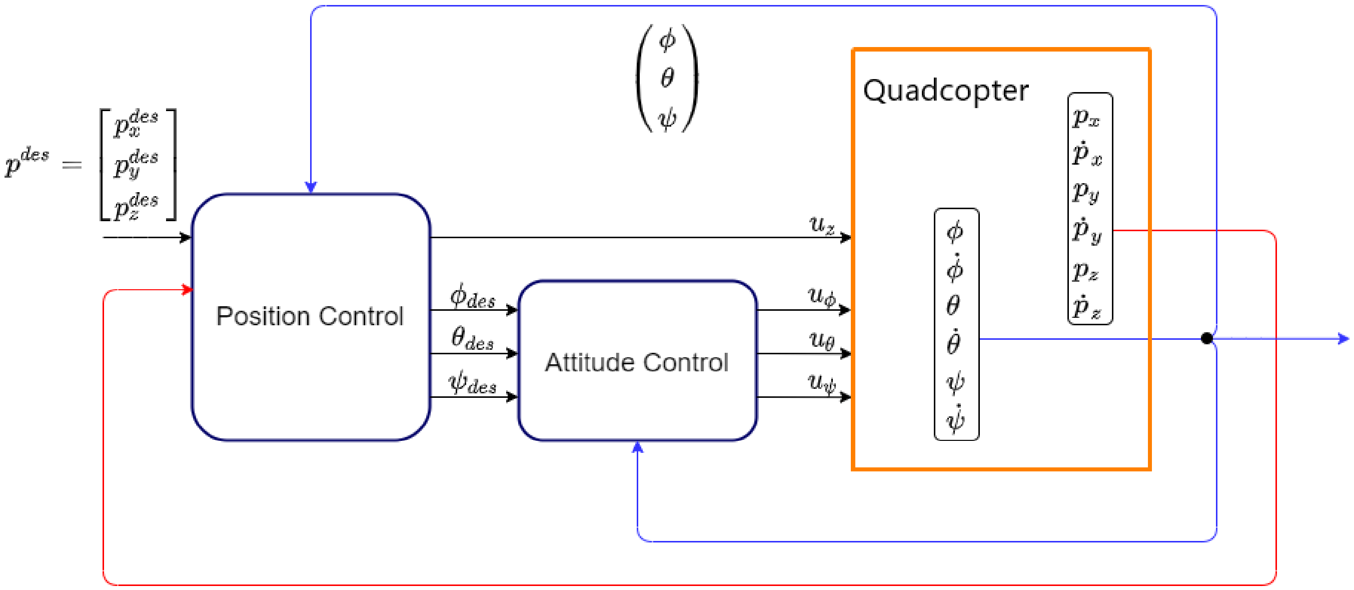
$p^{des} = \begin{bmatrix} p_x^{des} \\ p_y^{des} \\ p_z^{des} \end{bmatrix}$
$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}$
Quadcopter
Position Control
Attitude Control
ϕ_{des}
θ_{des}
ψ_{des}
u_z
u_ϕ
u_θ
u_ψ
ϕ $\dot{\phi}$ θ $\dot{\theta}$ ψ $\dot{\psi}$
p_x $\dot{p}_x$ p_y $\dot{p}_y$ p_z $\dot{p}_z$

Fig. 2 Control Diagram of the Quadcopter

The position control block generates an orbit that sets the desired angles for the attitude control block. The attitude control block tracks the desired angles generated by the position control block. In Figure 2, there are two separate cascade structures, and as a result of the trajectory control, reference attitude angles are generated for the inner control structure. In the following equations, $A_p = \mathrm{diag}(a_p, a_p, a_p) \in R^{6x6}$, $a_p = [0_{2\times1}\ e_{2,1}]$ and $B_p = m^{-1}k_w[e_{6,2}, e_{6,4}, e_{6,6}]$.

Let $x_p = [p_x, \dot{p}_x, p_y, \dot{p}_y, p_z, \dot{p}_z]^T$ and thus transform equations (15) into (24) and (25).

$$\dot{x}_p = A_p x_p + B_p u_p$$ (24)

$$\dot{x}_{pd} = A_{pd} x_{pd}$$ (25)

The variable $\dot{X}_p$ is defined as in (26).

$$\dot{X}_p = \begin{bmatrix} \dot{e}_p \\ \dot{x}_{pd} \end{bmatrix} = \begin{bmatrix} A_p & A_p - A_{pd} \\ 0_{6x6} & A_{pd} \end{bmatrix} X_p + \begin{bmatrix} B_p \\ 0_{6x3} \end{bmatrix} u_p$$ (26

According to the equations above, the cost function is determined as in (27) and (28).

$$V_p(X_p(t)) = \int_t^\infty e^{-\lambda(\tau-t)} (X_p(\tau)^T Q_p X_p(\tau) + u_p(\tau)^T R_p u_p(\tau)) d\tau$$ (27)

$$Q_p = \begin{bmatrix} Q_{ep} & 0_{6x6} \\ 0_{6x6} & 0_{6x6} \end{bmatrix} \in R^{12x12},\ R_p \in R^{3x3}$$ (28)

Then, the optimal control input $u_p = [u_{px}, u_{py}, u_{pz}]^T$ is derived from the off-policy iteration algorithm, the control input u_z is get, and the desired heading angles for the inertial control loop are set according to the following equations. The desired yaw angle ψ_d is chosen as a constant of 0.

$$u_z = \sqrt{u_{px}^2 + u_{py}^2 + (u_{pz} + u_b)^2}$$ (29)

$$\psi_d = 0$$ (30)

$$\phi_d = \arcsin(u_{px}\sin(\psi_d) - u_{py}\cos(\psi_d)/u_z)$$ (31)

$$\theta_d = \arctan(u_{px}\cos(\psi_d) + u_{py}\sin(\psi_d)/(u_{pz} + u_b)$$ (32

Once the desired set of heading angles is obtained, the attitude controller equations are solved in a similar manner. If $x_\Theta = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ is defined, equation (16) can be transformed into equations (33) and (34).

$$\dot{x}_\Theta = F_\Theta x_\Theta + B_\Theta u_\Theta \tag{33}$$

$$\dot{x}_{\Theta d} = F_{\Theta d} x_{\Theta d} \tag{34}$$

Similarly, $\dot{x}_{\Theta d}$ can be written as in (35).

$$\dot{X}_{\Theta d} = \begin{bmatrix} \dot{e}_\Theta \\ \dot{x}_{\Theta d} \end{bmatrix} = \begin{bmatrix} F_\Theta & F_\Theta - F_{\Theta d} \\ 0_{6x6} & F_{\Theta d} \end{bmatrix} X_{\Theta d} + \begin{bmatrix} B_\Theta \\ 0_{6x3} \end{bmatrix} u_\Theta \tag{35}$$

Accordingly, the cost function is defined as follows.

$$V_\Theta(X_\Theta(t)) = \int_t^\infty e^{-\boldsymbol{\lambda(\tau-t)}} (X_\Theta(\tau)^T Q_\Theta X_\Theta(\tau) + u_\Theta(\tau)^T R_\Theta u_\Theta(\tau)) d\tau \tag{36}$$

$$Q_\Theta = \begin{bmatrix} Q_{e\Theta} & 0_{6x6} \\ 0_{6x6} & 0_{6x6} \end{bmatrix} \in R^{12x12},\ R_\Theta \in R^{3x3} \tag{37}$$

The simulation coefficients of the quadcopter are shown in Table 1. Below, m is the mass of the quadcopter k_w and k_t represents the aerodynamic constants, g represents the gravitational acceleration, l represents the arm length of the quadcopter, and J represents the diagonal of the moments of inertia J_x, J_y, and J_z.

Table 1. *Simulation coefficients of the quadcopter*

Coefficient	**Value**
m	2 kg
k_w	1 Ns^2
k_t	1 Ns^2
g	9.8 m/s^2
l	0.2 m
J	diag(5.1, 5.1, 5.2) 10^{-3}kg.m^2

3. Simulations

Simulations were performed using the MATLAB program. The desired orbit is $p_d = [2\sin(at), 2\cos(at), 0.8t]^T$ and a = 0.5. First of all, in order to ensure that the system remains stable during the data collection phase, 2 PD (proportional derivative) controllers were used for position and attitude control loops. K_p and K_d coefficients were selected experimentally with [12, 15] for the outer loop and [5, 100] for the inner loop. Simulations were realized for attitude and position.

3.1 Attitude simulations

In this section, simulations are performed to follow the roll, pitch, and yaw angles of the quadcopter. Figures 3 to 5 show these simulations. During the simulations, noise is added to both PD controllers. $u_{pe} = \sum_{m=1}^{100} 0.01\sin(w_m t)$ represents the noise added to the position control loop, and $u_{\Theta e} = \sum_{m=1}^{500} 0.02\sin(w_m t)$ represents the noise added to the attitude control loop. The w_m values are randomly selected frequencies in the range [−100, 100].

In Figure 3, phi_r shows the roll angle given as a reference. Phi is the roll angle. As seen in Figure 3, distortion is seen in the simulation because of the noise given to the system. As seen in Figure 3, the off-policy reinforcement learning algorithm successfully captured the reference.

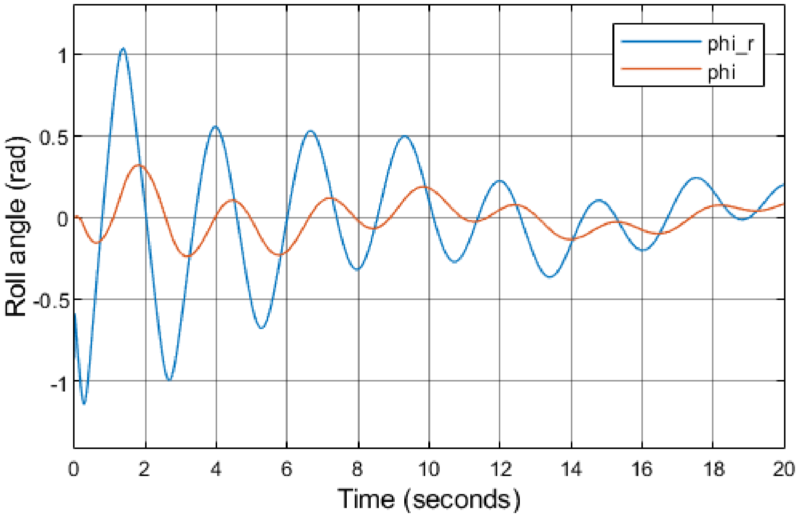

Figure 3. Roll angle reference tracking

In Figure 4, theta_r shows the reference pitch angle, and theta gives the pitch angle. When Figure 4 is examined, it is observed that the reinforcement learning algorithm can successfully track references in a noisy environment.

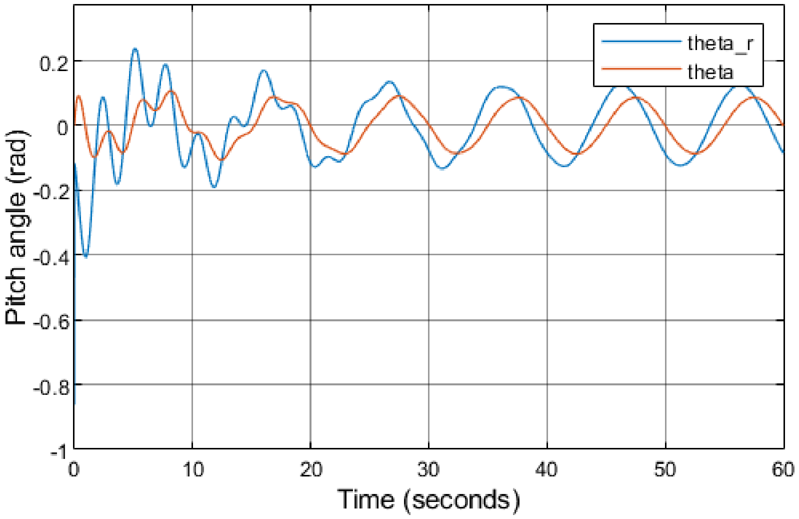

Figure 4. Pitch angle reference tracking

In Figure 5, psi_r symbolizes the reference yaw angle, and psi is the yaw angle. In this work, the reference yaw angle is initially fixed at 0. The proposed reinforcement learning algorithm could exactly track the reference yaw angle fixed at zero.

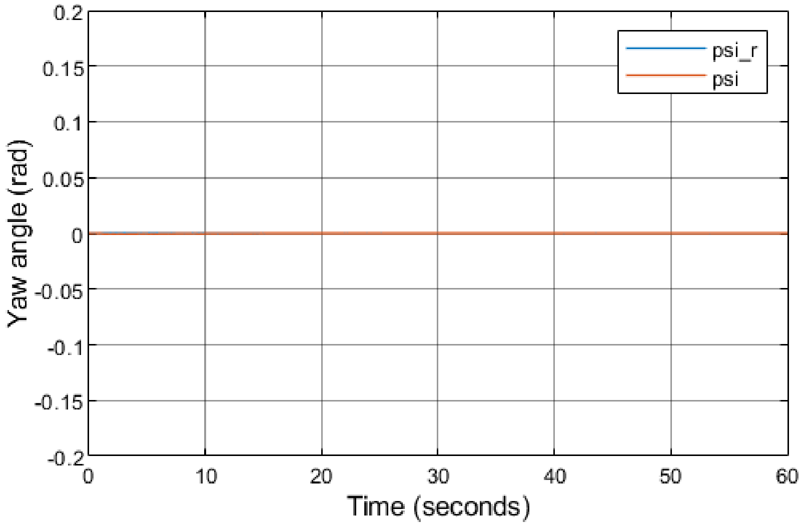

Figure 5. Yaw angle reference tracking

Table 2 shows the time response of roll, pitch, and yaw trajectories. Rise time and settling time are given in seconds, and overshoot is given in percent.

Table 2. *Attitude Controller's Time Response*

Parameter	Rise time	Overshoot	Settling Time
Roll	0.938 s	297.1%	50.12 s
Pitch	9.95 s	0%	37.52 s
Yaw	0 s	0%	0 s

When the values in Table 2 are examined, the rise time of the roll trajectory is 0.938 seconds. This indicates that the quadcopter can quickly arrive 90 percent of the given reference. At the beginning of the simulation, a high overshoot is observed while following the roll trajectory due to intense noise. However, with the help of the reinforcement learning algorithm, the quadcopter learned how to track the desired trajectory over time, and the overshoot decreased. The settling time in the roll angle simulation is 50 seconds and after this time, the quadcopter can follow the given reference in a very close range.

The rise time of the pitch orbit is 9.95 seconds. The quadcopter does not show any overshoot while following the pitch orbit. The quadcopter can follow the pitch orbit more stably than the roll orbit. The settling time of the quadcopter in the pitch orbit is 37 seconds. In this case, the quadcopter has a shorter settling time compared to the roll orbit.

In this study, since the yaw reference is fixed at 0, the rise time and settling time are zero seconds, and the overshoot is 0% for the yaw trajectory.

3.2 Position Simulations

In this part, the x, y, and z axis trajectories of the quadcopter are simulated. As a reference, the sinusoidal function is used for the x and y axes, and the ramp function is used for the z axis. Figures between 6 and 8 present the simulations performed for x, y, z trajectories.

In Figure 6, the blue line shows the reference x-trajectory and the red line shows the trajectory that the quadcopter actually follows. Thanks to the successful operation of the reinforcement learning algorithm, the quadcopter can follow the reference trajectory almost without error. Only at the beginning of the simulation, during the rise of the quadcopter, a small tracking error was seen.

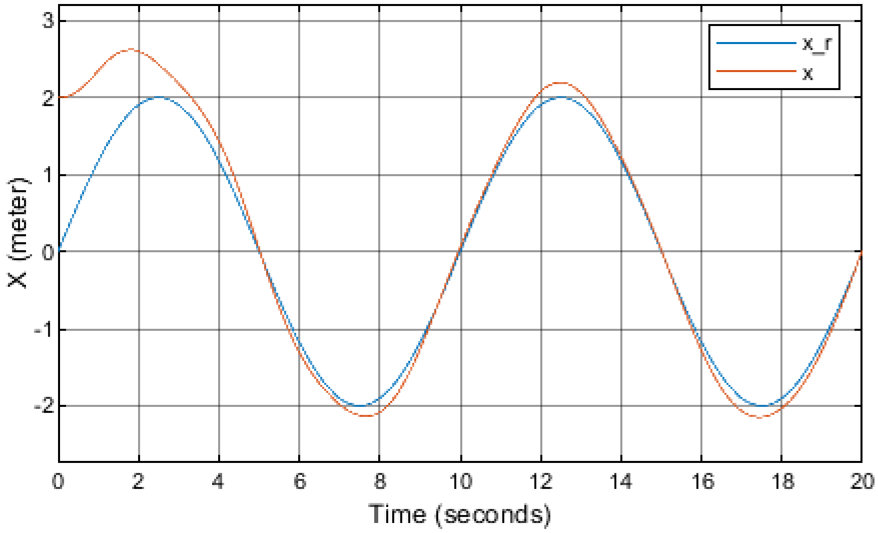

Figure 6. X-axis reference tracking

In Figure 7, the blue line shows the reference y-trajectory and the red line shows the trajectory that the quadcopter actually tracks. Thanks to the reinforcement learning algorithm, the quadcopter could accurately follow the reference orbit. Some very small tracking errors emerge in the first 10 seconds, but later, the reference and the trajectory followed by the quadcopter entirely overlap.

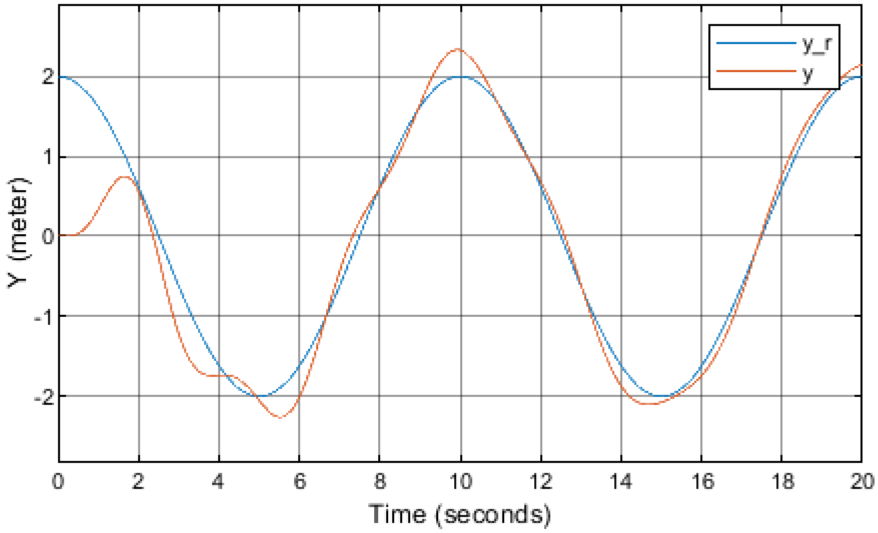

Figure 7. Y-axis reference tracking

In Figure 8, the blue line shows the reference z value, and the red line shows the actual trajectory followed by the quadcopter. With the help of the algorithm, the ramp orbit is tracked nearly without error.

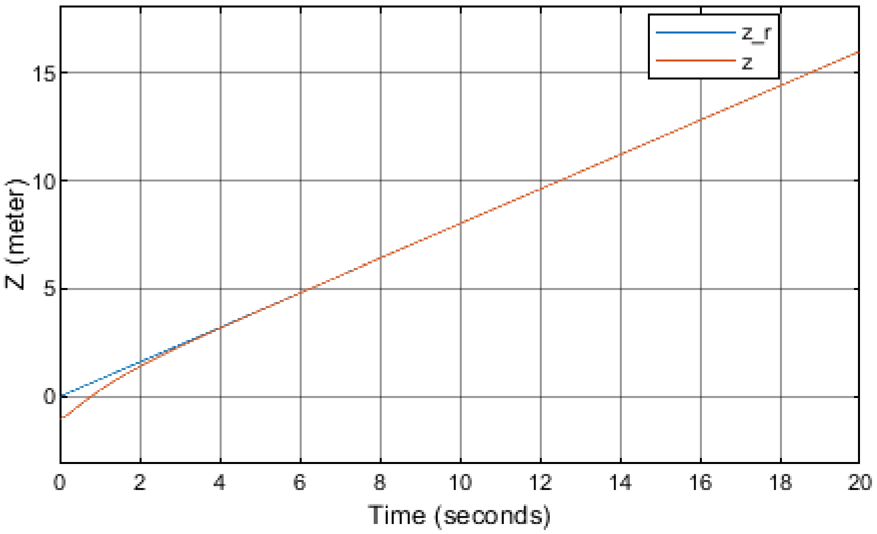

Figure 8. Z-axis reference tracking

Table 3 presents the time response for position simulations.

Table 3. Position Controller's Time Response

Parameter	Rise Time	Overshoot	Settling Time
x	0.735 s	6.97%	3.087 s
y	1.8 s	10.82%	10.72 s
z	1.88 s	0%	3.17 s

The quadcopter reaches the reference value with a very fast rise time while following the x-trajectory. It shows a low overshoot of around 7% and has a very short settling time of 3 seconds.

The quadcopter reaches 90% of the given reference value with a short rise time of 1.8 seconds while following the y-trajectory. It shows a slight overshoot of about 10% and has an acceptable settling time of 10 seconds.

The quadcopter presents a quick rise time of 1.88 seconds while tracking the z-trajectory. It represents 0% overshoot and has a quick settling time of 3 seconds.

According to these data, it is clear that the quadcopter accurately tracks the x, y, z orbits with the help of the reinforcement learning algorithm.

4. Conclusion

In this research, the optimum trajectory tracking of a quadcopter based on reinforcement learning is considered. First, the dynamic equations of the quadcopter are presented. Thereafter, the data-driven policy iteration algorithm is discussed. The cost function used in the algorithm is chosen. Modeling of the quadcopter and its controllers and development of the reinforcement learning algorithm are performed using MATLAB/Simulink program. Thanks to the simulations, the ability of the quadcopter to track the attitude and position orbits under random noise is observed. Time response data are obtained, and a mathematical inspection is realized. Simulations prove that the reinforcement learning-based algorithm could successfully follow the given trajectories under noise. The successful results obtained in this study were limited to simulations. In practical applications, some problems that could not be detected in simulations may be encountered. Therefore, future studies are planned to compare the consistency of simulation results with real-life results by focusing on practical applications.

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

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] Chaurasia, R., & Mohindru, V. (2021). Unmanned aerial vehicle (UAV): A comprehensive survey. *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*, 1-27.

[3] Dong, S. (2021, March). Application research of quadrotor UAV motion planning system based on heuristic search and minimum-snap trajectory optimization. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 2270-2274). IEEE.

[4] Elmeseiry, N., Alshaer, N., & Ismail, T. (2021). A detailed survey and future directions of unmanned aerial vehicles (uavs) with potential applications. *Aerospace*, *8*(12), 363.

[5] Hong, Y., Kim, S., Kim, Y., & Cha, J. (2021). Quadrotor path planning using A* search algorithm and minimum snap trajectory generation. *ETRI Journal*, *43*(6), 1013-1023.

[6] Iskander, A., Elkassed, O., & El-Badawy, A. (2020, October). Minimum snap trajectory tracking for a quadrotor UAV using nonlinear model predictive control. In *2020 2nd Novel intelligent and leading emerging sciences conference (NILES)* (pp. 344-349). IEEE.

[7] Karahan, M., Kasnakoglu, C., & Akay, A. N. (2023). Robust backstepping control of a quadrotor uav under pink noise and sinusoidal disturbance. *Studies in Informatics and Control, 32*(2), 15-24.

[8] 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.

[9] Karahan, M. (2024, May). Optimal Trajectory Tracking Control for a Quadrotor UAV Based on Off-Policy Reinforcement Learning. In *2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-5). IEEE.

[10] Karahan, M., Inal, M., & Kasnakoglu, C. (2023). Fault Tolerant Super Twisting Sliding Mode Control of a Quadrotor UAV Using Control Allocation. *International Journal of Robotics and Control Systems, 3*(2), 270-285.

[11] Kiumarsi, B., & Lewis, F. L. (2014). Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems. *IEEE transactions on neural networks and learning systems*, *26*(1), 140-151.

[12] Karahan, M., & Kasnakoglu, C. (2022, September). Nonlinear Modeling and Robust Backstepping Control of a Quadrotor Unmanned Aerial Vehicle. In *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)* (pp. 94-99). IEEE.

[13] Kamalapurkar, R., Dinh, H., Bhasin, S., & Dixon, W. E. (2015). Approximate optimal trajectory tracking for continuous-time nonlinear systems. *Automatica*, *51*, 40-48.

[14] Lee, C., Kim, S., & Chu, B. (2021). A survey: Flight mechanism and mechanical structure of the UAV. *International Journal of Precision Engineering and Manufacturing*, *22*(4), 719-743.

[15] Modares, H., & Lewis, F. L. (2014). Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica*, *50*(7), 1780-1792.

[16] Modares, H., & Lewis, F. L. (2014). Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning. *IEEE Transactions on Automatic control*, *59*(11), 3051-3056.

[17] Öniz, Y. (2024). Tip-2 Nöro-Bulanık Denetleyiciler ile Döner Kanatlı İnsansız Hava Aracının Yörünge Takibi. *Gazi University Journal of Science Part C: Design and Technology*, *12*(1), 40-56.

[18] Pi, C. H., Ye, W. Y., & Cheng, S. (2021). Robust quadrotor control through reinforcement learning with disturbance compensation. *Applied Sciences*, *11*(7), 3257.

[19] Pham, D. D., Cao, T. T., Nguyen, T. C., & Dao, P. N. (2022). Optimal tracking controllers with Off-policy Reinforcement Learning Algorithm in Quadrotor. *Annals of Computer Science and Information Systems*, *33*.

[20] Rokonuzzaman, M., Mohajer, N., Nahavandi, S., & Mohamed, S. (2021). Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intelligent Transport Systems*, *15*(5), 646-670.

[21] Singh, B. K., & Kumar, A. (2023). Attitude and position control with minimum snap trajectory planning for quadrotor UAV. *International Journal of Dynamics and Control*, *11*(5), 2342-2353.

[22] Sanwale, J., Trivedi, P., Kothari, M., & Malagaudanavar, A. (2020). Quaternion-based position control of a quadrotor unmanned aerial vehicle using robust nonlinear third-order sliding mode control with disturbance cancellation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *234*(4), 997-1013.

[23] Sabour, M. H., Jafary, P., & Nematiyan, S. (2023). Applications and classifications of unmanned aerial vehicles: A literature review with focus on multi-rotors. *The Aeronautical Journal*, *127*(1309), 466-490.

[24] Tan, Z., & Karaköse, M. (2022). Dinamik ortamlarda derin takviyeli öğrenme tabanlı otonom yol planlama yaklaşımları için karşılaştırmalı analiz. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, *9*(16), 248-262.

[25] Wen, G., Hao, W., Feng, W., & Gao, K. (2021). Optimized backstepping tracking control using reinforcement learning for quadrotor unmanned aerial vehicle system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(8), 5004-5015.

[26] Xuan M, N., Nguyen, N. P., Pham, D. B., Dao, N. N., & Hong, S. K. (2022). Synthesized landing strategy for quadcopter to land precisely on a vertically moving apron. *Mathematics*, *10*(8), 1328.

[27] Yoo, J., Jang, D., Kim, H. J., & Johansson, K. H. (2020). Hybrid reinforcement learning control for a micro quadrotor flight. *IEEE Control Systems Letters*, *5*(2), 505-510.

[28] Zare, M., Pazooki, F., & Haghighi, S. E. (2022). Quadrotor UAV position and altitude tracking using an optimized fuzzy-sliding mode control. *IETE Journal of Research*, *68*(6), 4406-4420.