Journal of Engineering and Applied Research (JEAR)

National University of Skill

Winter & Spring 2026, Vol. 3, No. 1, p. 129-159

Journal Homepage: https://jear.tvu.ac.ir

doi :10.48301/jear.2025.526595.1121

Adaptive Precision in UAV Flight Dynamics: A Hybrid Recursive Least Squares and Digital LQR Framework for Real-Time Deflection Control

Pezhman Bayat [1*], Peyman Bayat [2]

[1, 2] Department of Electrical Engineering, Hamedan University of Technology, Hamedan, Iran.

ARTICLE INFO

Article Type:
Original Research

Received: 2025.06.10
Revised: 2025.08.26
Accepted: 2025.10.15

Keyword:
Aircraft
Digital linear quadratic
Regulator (DLQR)
Flight control system
Recursive least square (RLS)
Unmanned aerial vehicle (UAV)

***Corresponding Author:**
Pezhman Bayat
Email: *pezhman.bayat@hut.ac.ir*

ABSTRACT

This paper presents the design and analysis of an advanced autopilot system for controlling the lateral and longitudinal dynamics of a custom-built unmanned aerial vehicle (UAV). The system regulates roll, pitch, and yaw angles through a unified control strategy, leveraging recursive least squares (RLS) estimation to model the UAV's dynamic behavior with high precision. A key innovation lies in applying RLS to extract both lateral and longitudinal transfer functions, offering a level of adaptability rarely seen in comparable studies. To optimize multi-axis control, a digital linear quadratic regulator (DLQR) is developed, enabling simultaneous tuning of all three rotational degrees of freedom. This integrated approach marks a departure from conventional proportional, proportional-integral- derivative or fuzzy logic controllers, which often treat these axes independently. The simulation framework incorporates pseudo-random binary sequence (PRBS) inputs and white Gaussian noise (WGN), ensuring robustness under realistic operating conditions. The UAV platform is grounded in validated aerodynamic data, with stability derivatives obtained from DATCOM and JSBSIM, effectively bridging theoretical modeling with real-world flight dynamics. Simulation results reveal substantial improvements in rise time, settling time, and overshoot across all axes, demonstrating the DLQR controller's effectiveness in achieving fast, stable, and responsive flight behavior.

E-ISSN: 3060-6357

Nomenclature:

Symbol	*Description*	*Unit*
a_1, a_2, a_3, a_4	Denominator coefficients in discrete transfer functions	–
A	State matrix in continuous/discrete state-space models	–
a	Speed of sound (implicitly used via Mach number or DATCOM)	m/s
b_1, b_2, b_3, b_4	Numerator coefficients in discrete transfer functions	–
B	Input matrix in state-space models	–
C	Output matrix in state-space models	–
D	Feedthrough matrix in state-space models	–
δ_e	Elevator deflection angle	rad
δ_a	Aileron deflection angle	rad
δ_r	Rudder deflection angle	rad
F	State transition matrix in discrete-time system	–
G	Control input matrix in discrete-time system	–
g	Gravitational acceleration	m/s²
J	Cost function in DLQR control	–
K	Feedback gain matrix in DLQR control	–
L	Roll moment	–
M	Pitch moment	–
N	Yaw moment	–
Ma	Mach number (implicitly used)	–
m	Mass of UAV	kg
P	Covariance matrix in RLS / Riccati solution	–
p	Roll rate	rad/s
q	Pitch rate	rad/s
r	Yaw rate	rad/s
Q	Weight matrix in DLQR cost function	–
R	Weight matrix in DLQR cost function	–
Re	Reynolds number (used via DATCOM)	–
T	Sampling period or thrust (context-dependent)	N
u, v, w	Velocity components in body x, y, z directions	m/s
u_0	Trim or steady-state forward velocity	m/s
$x(t), x(k)$	State vector in continuous/discrete systems	–
$y(t), y(k)$	Output signal in continuous/discrete systems	–
Y_v, Y_p, Y_r	Stability derivatives of lateral force	–
L_v, L_p, L_r	Stability derivatives of roll moment	–
N_v, N_p, N_r	Stability derivatives of yaw moment	–
X_u, X_w	Longitudinal force derivatives	–
Z_u, Z_w	Vertical force derivatives	–
M_u, M_w, M_q	Pitch moment derivatives	–
ϕ	Roll angle	rad
θ	Pitch angle	rad
ψ	Yaw angle	rad
$\Delta u, \Delta v, \Delta w$	Perturbations in velocity components	m/s
$\Delta p, \Delta q, \Delta r$	Perturbations in angular rates	rad/s
$\Delta\phi, \Delta\theta, \Delta\psi$	Perturbations in Euler angles	rad
$\Delta\delta_e, \Delta\delta_a, \Delta\delta_r$	Perturbations in control surface deflections	rad
β	Sideslip angle	rad
α	Angle of attack (used via DATCOM)	rad
λ	Eigenvalue	–
$z^{-1}, z^{-2}, \ldots$	Discrete-time delay operator	–

1) Introduction

Aircraft control relies on three primary surfaces: the elevator, rudder, and ailerons. Elevators, typically located at the rear of an unmanned aerial vehicle (UAV), adjust the aircraft's lateral attitude by altering the pitch balance. When the pilot moves the elevator control backward, the elevator deflects up or down [1], [2]. The rudder, positioned at the trailing edge of the vertical tail, manages yaw control. By deflecting the rudder, yaw motion is achieved, which involves side-to-side movement of the aircraft's nose. Ailerons, responsible for roll control, are small adjustable flaps located near the wing tips and operate in a differential manner to induce rolling (refer to Figs. 1 and 2).

The UAV's control system is divided into two segments: longitudinal and lateral control. Pitch control falls under longitudinal control. Substantial research has been conducted on designing controllers aimed at improving flight stability [2], [3] and [4].

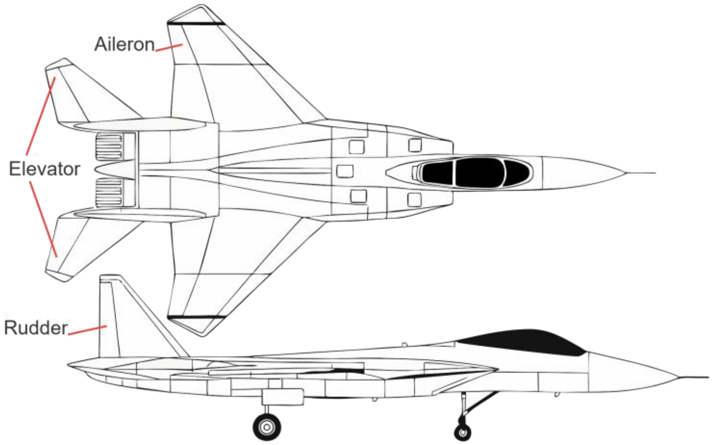


Figure 1. **Three main surfaces of an UAV.**

In [1], a linear quadratic regulator (LQR) and an intelligent fuzzy logic controller (FLC) were designed for managing the pitch of aircraft systems, and their performance was evaluated based on pitch angle and pitch rate. The study referenced in [2] explores the distributed fault-tolerant (FT) consensus control challenge for linear multiagent systems operating under disturbances. For undirected graphs, it proposes the use of adaptive control coefficients, facilitating fully distributed and dependable control without necessitating prior knowledge of global information. Additionally, in contrast to conventional FT consensus approaches, this method removes the dependency on continuous communication between agents by implementing an open-loop estimation mechanism combined with an event-triggered strategy. The authors in [3] explore the problem of fixed-time adaptive fuzzy containment control for six-rotor multi-UAV systems, with considerations for flight route constraints and external disturbances. They propose a comprehensive solution by presenting a robust adaptive fuzzy fixed-time control strategy. This approach leverages the backstepping recursive algorithm in conjunction with FLS technology, effectively incorporating fixed-time control theory to achieve precise and reliable control. The researchers in [4], [5] investigate an event-triggered tunnel prescribed control approach for uncertain nonlinear systems, designed to handle any initial conditions. They introduce the more comprehensive entry capture problem, which ensures that the tunnel-prescribed performance criteria are fulfilled after a specified operational period of the system, rather than from the outset. This adjustment adds complexity and challenges to the control design process. Furthermore, the typically utilized prescribed performance control becomes ineffective in this scenario due to the singularity issue stemming from the violation of initial conditions.

In [6], both proportional-integral- derivative (PID) and PID-type fuzzy logic controllers were implemented for pitch angle control. The findings concluded that PID-type fuzzy logic controllers demonstrated superior performance compared to conventional PID controllers.

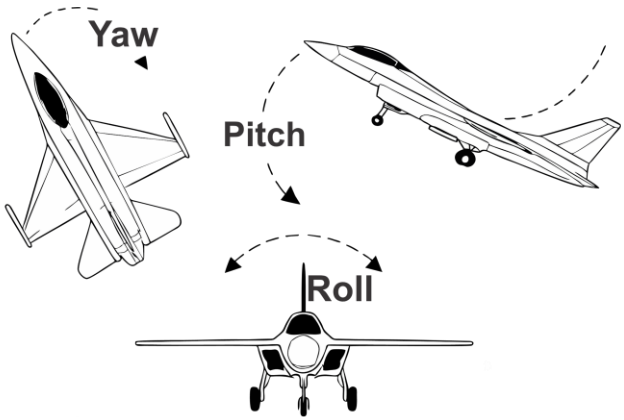


Figure 2. UAV principal axes.

In [7], [8], a fuzzy PID controller optimized using genetic algorithms (GA) was proposed to address under-shoot issues in non-minimum phase linear systems. In [9], an optimal distributed FLC was systematically developed for attitude control using reaction wheels and the gravity gradient method. In [10], a fuzzy sliding mode (FSM) autopilot was introduced, aiming to achieve precise tracking, desirable time response characteristics, and robustness against parametric uncertainties. In [11], a conventional controller was used as the core component of the autopilot, while an FLC, independent of the system model, was employed to enhance robustness. Yaw control is categorized under lateral control. In [12], a control system was developed for managing yaw, where both LQR and intelligent FLC were applied. The study concluded that the LQR controller demonstrated relatively better performance compared to the FLC controller. Similarly, roll control is also a lateral control issue. Studies in [1], [13] and [14] introduced advanced systems for roll control to enhance overall aircraft performance.

In recent years, deep learning and machine learning have seen significant growth in control system applications. These advanced techniques have been extensively explored in the literature on industrial system control and monitoring, with notable discussions on their use in vehicle identification and counting systems [15], optimizing air pressure system maintenance costs in Scania trucks [16], and reviews of quantum machine learning approaches [17]. Also, academic and industrial research has increasingly focused on UAVs and machine learning, recognizing their expanding potential. UAVs have significantly enhanced operational flexibility, enabling remote monitoring of activities. Consequently, researchers in [18] gathered and analyzed studies exploring the integration of UAVs with machine learning and its various algorithms across different fields and regions. Their aim was to assess the scope and significance of machine learning models in improving UAV capabilities, addressing challenges, and broadening application areas.

In this paper, the lateral and longitudinal directional dynamics of a UAV are first analyzed by designing an autopilot system to control the roll, yaw, and pitch angles. These dynamics are estimated using the recursive least squares (RLS) method. Subsequently, a digital linear quadratic regulator (DLQR) controller is proposed and developed to enhance the UAV's flight control. The design aims to achieve key objectives, including improved time response characteristics, balanced system error, and minimized control effort.

So, the article highlights five major innovations, which can be summarized as follows:

- RLS Estimation Applied to UAV Dynamics: While RLS is a known technique, its application here to estimate both lateral and longitudinal transfer functions of a custom-built UAV add a layer of precision and adaptability not commonly seen in similar studies.

- DLQR-Based Autopilot Design for Multi-Axis Control: The paper introduces a DLQR tailored to simultaneously optimize roll, pitch, and yaw control. This unified strategy contrasts with prior works that often treat these axes independently or rely on conventional PID or fuzzy logic controllers.

- Real-Time Parameter Estimation with Noise Consideration: The use of pseudo-random binary sequence (PRBS) inputs and white Gaussian noise (WGN) in the simulation framework demonstrates robustness in real-world conditions, enhancing the practical relevance of the proposed method.

- Custom UAV Platform with Verified Stability Derivatives: The study is grounded in a physical UAV model with validated aerodynamic coefficients from DATCOM and JSBSIM, bridging the gap between theoretical modeling and actual flight dynamics.

- Improved Time-Response Characteristics via DLQR Tuning: The simulation results show significant improvements in rise time, settling time, and overshoot across all three axes, underscoring the effectiveness of the DLQR controller in achieving fast and stable flight responses.

2) UAV Model

Aircraft dynamics are traditionally modeled using six interdependent nonlinear differential equations. With the application of specific simplifying assumptions, these equations can be decoupled into two primary categories: longitudinal and lateral dynamics. In Figs. 3, 4, and 5, the variables X_b, Y_b and Z_b represent the aerodynamic force components acting on the aircraft. The symbols $\emptyset_a$ and δ_a denote, respectively, the roll angle (defining the aircraft's orientation) and the aileron deflection angle. In a similar fashion, θ and δ_e correspond to the pitch angle and the elevator deflection angle, while $\emptyset_r$ and δ_r indicate the yaw angle, as measured relative to the Earth-fixed frame, and the rudder deflection angle.

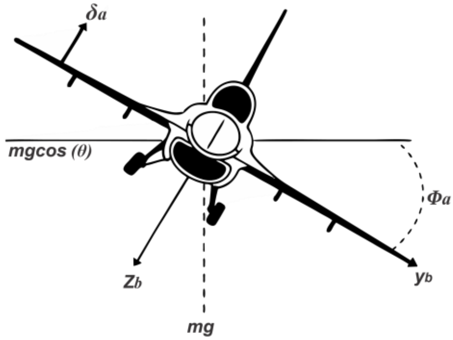


Figure 3. **Description of roll control system.**

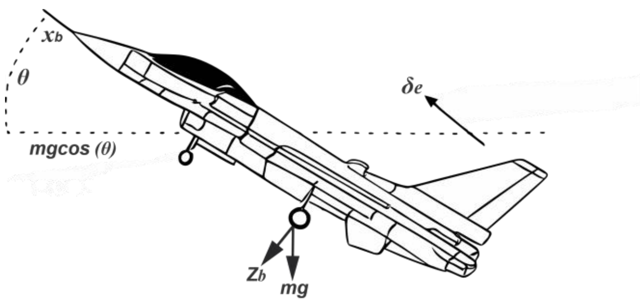


Figure 4. **Description of pitch control system.**

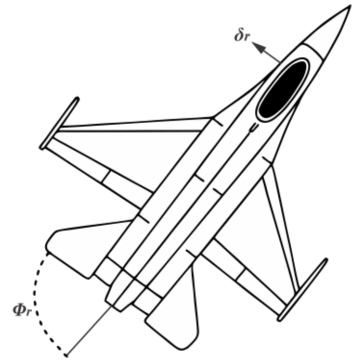


Figure 5. **Description of yaw control system.**

Fig. 6 offers a depiction of the forces, moments, and velocity components within the aircraft's body-fixed coordinate system. Here, L, M, and N denote the aerodynamic moment components about the roll, pitch, and yaw axes respectively. The angular velocities, represented by p, q, and r, align with the roll, pitch, and yaw motions. Additionally, the translational velocities along these axes are indicated by u, v, and w. This detailed breakdown provides a comprehensive framework for understanding the intricate interplay of forces and motions that dictate the behavior of an aircraft.

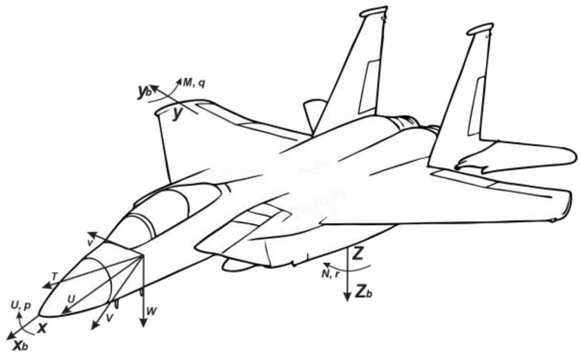


Figure 6. **Definition and orientation of forces, moments, and velocity components within the body-fixed coordinate frame of an aircraft system.**

The equations of motion for an aircraft are derived using Newton's second law [2]. However, before formulating these equations, certain assumptions and approximations must be made [1]:

- The analysis assumes that the aircraft is flying in steady cruise conditions, maintaining a constant altitude and unchanging velocity. In this regime, the propulsive thrust exactly counteracts the drag force, while the lift generated equals the gravitational weight of the aircraft.
- It is further assumed that variations in the pitch angle have no impact on the aircraft's speed under these conditions.

Based on these assumptions, the aircraft's lateral and longitudinal directional motions can be effectively described using the following kinematic and dynamic differential equations. The governing force equations are:

$$X - mgs_{\theta} = m\left(du/dt + qw - rv\right) \tag{1}$$

$$\mathrm{Y} + \mathrm{mg}c_{\theta}s_{\varphi} = m\left(dv/dt + \mathrm{ru} - \mathrm{pw}\right) \tag{2}$$

$$\mathrm{Z} + \mathrm{mg}c_{\theta}c_{\varphi} = m\left(dw/dt + \mathrm{pv} - \mathrm{qu}\right) \tag{3}$$

The moment equations describe the rotational dynamics of an aircraft, encompassing the relationships between aerodynamic moments (roll, pitch, and yaw) and their respective angular rates, system parameters, and control surface deflections. Moment equations are:

$$L = I_x (dp/dt) - I_{xz} (dr/dt) + qr\left(I_z - I_y\right) - I_{xz} \tag{4}$$

$$M = I_y dq/dt + rp\left(I_x - I_z\right) + I_{xy}\left(p^2 - r^2\right) \tag{5}$$

$$N = -I_{xz}(dp/dt) + I_z(dr/dt) + pq\left(I_y - I_x\right) + I_{xz}q \tag{6}$$

Equations (1) through (6) are inherently nonlinear but can be linearized through the application of small-disturbance theory. This process involves substituting each variable in the equations of motion with its reference value plus a small perturbation or disturbance. As a result of this linearization, Equations (7) through (12) are derived. The longitudinal equations describe the aircraft's motion along its flight path, focusing on parameters like pitch dynamics, velocity changes, and forces in the vertical plane. These equations govern how the aircraft responds to variations in thrust, lift, drag, and pitching moments. Longitudinal equations are:

$$\left(\frac{d}{dt} - X_u\right)\Delta u - X_w \Delta w + \left(g\, cos\theta_0\right)\Delta\theta = X_{\delta e}\Delta\delta_e + X_{\delta T}\Delta\delta_T \tag{7}$$

$$-Z_u \Delta u\left(\left(1 - Z_w\right)\frac{d}{dt} - Z_w\right)\Delta w - \left(\left(u_0 + Z_q\right)\frac{d}{dt} - gsin\theta_0\right)\Delta\theta = Z_{\delta e}\Delta\delta_e + Z_{\delta T}\Delta\delta_T \tag{8}$$

$$-M_u \Delta u - \left(M_w \frac{d}{dt} + M_w\right)\Delta w + \left(\frac{d^2}{dt^2} - M_q \frac{d}{dt}\right)\Delta\theta = M_{\delta e}\Delta\delta_e + M_{\delta T}\Delta\delta \tag{9}$$

The lateral equations describe the dynamics of an aircraft related to side-to-side movements, including roll, yaw, and side forces. These equations govern how the aircraft behaves in response to lateral disturbances and control surface deflections, such as those from the rudder and ailerons. Lateral equations are:

$$\left(\frac{d}{dt} - Y_v\right)\Delta v - Y_p \Delta p + \left(u_0 - Y_r\right)\Delta r - \left(g\, cos\theta_0\right)\Delta\varnothing = Y_{\delta r}\Delta\delta_r \tag{10}$$

$$-L_v \Delta v + \left(\frac{d}{dt} - L_p\right)\Delta p - \left(\frac{I_{xz}}{i_x}\frac{d}{dt} - L_r\right)\Delta r = L_{\delta a}\Delta\delta_a + L_{\delta r}\Delta\delta_r \tag{11}$$

$$-N_v\Delta v-\left(\frac{I_{xz}}{I_z}\frac{d}{dt}-N_p\right)\Delta p-\left(\frac{d}{dt}-N_r\right)\Delta r=N_{\delta a}\Delta\delta_a+N_{\delta r}\Delta\delta_r \tag{12}$$

The lateral-directional equations of motion are a set of mathematical expressions that capture the aerodynamic interactions dictating an aircraft's lateral forces (side force), rolling moments, and yawing moments during flight manoeuvres. In some cases, substituting the sideslip angle $\Delta\beta$ for the side velocity Δv can be more practical. These two parameters are interconnected and their relationship is expressed in Equation (13).

$$\Delta\beta\approx\tan^{-1}\frac{\Delta v}{u_0}=\frac{\Delta v}{u_0} \tag{13}$$

By incorporating the specified relationship and assuming that the cross-product term of inertia (IXZ) is negligible, the lateral motion equations are significantly streamlined. This simplification enables the transformation of the original equations into a state-space representation, as detailed in Equation (14).

$$\begin{bmatrix}\dot{\Delta\beta}\\ \dot{\Delta p}\\ \dot{\Delta r}\\ \dot{\Delta\varnothing}\end{bmatrix}=\begin{bmatrix}\frac{Y_\beta}{u_0} & \frac{Y_p}{u_0} & -\left(1-\frac{Y_r}{u_0}\right) & \frac{g\cos\theta_0}{u_0}\\ L_\beta & L_p & L_r & 0\\ N_\beta & N_p & N_r & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\beta\\ \Delta p\\ \Delta r\\ \Delta\varnothing\end{bmatrix}+\begin{bmatrix}0 & \frac{Y_{\delta r}}{u_0}\\ L_{\delta a} & L_{\delta r}\\ N_{\delta a} & N_{\delta r}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\delta_a\\ \Delta\delta_r\end{bmatrix} \tag{14}$$

The inputs to the system are the deflection angles of the rudder and ailerons, while the outputs correspond to the roll and yaw angles of the aircraft. Fig. 7 provides an overview of the entire modelling process, illustrating how the system components interact to achieve the desired control dynamics.

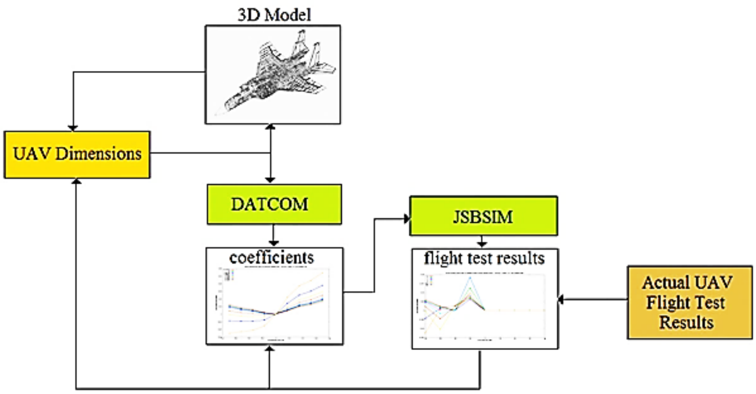


Figure 7. Overall modelling process.

The transfer function linking pitch angle variations to changes in elevator deflection is obtained by examining the interplay between pitch rate modifications and elevator inputs [3]. This derivation requires rigorous mathematical manipulation of the aircraft's dynamic equations governing pitch motion, ensuring that the resultant model precisely reflects the system's behaviour.

$$\Delta q = \Delta\dot{\theta} \tag{15}$$

$$\Delta q(s) = s\Delta\theta(s) \tag{16}$$

$$\frac{\Delta\theta(s)}{\Delta\delta_e(s)} = \frac{1}{s}\cdot\frac{\Delta q(s)}{\Delta\theta(s)} \tag{17}$$

In this study, the proposed UAV's data is utilized for system analysis and modelling, requiring the preliminary calculation of lateral stability derivatives. The stability parameters for lateral-directional derivatives are sourced from DATCOM® software and JSBSIM (flight dynamics model). These stability parameters for the proposed UAV are presented in Table 1. Additionally, substituting the dimensional longitudinal stability derivative values obtained from DATCOM provides results that closely approximate those reported in NASA report no. CR-2144.

Table 1. Stability parameters for proposed UAV

F-115 (UAV under study)	Y-Force Derivatives	Yawing Moment Derivatives	Rolling Moment Derivatives
Aileron Deflection	$Y_{\delta a} = 0$	$N_{\delta a} = -0.31$	$L_{\delta a} = 29.87$
Rudder Deflection	$Y_{\delta r} = 11.31$	$N_{\delta r} = -3.93$	$L_{\delta r} = 23.01$
Yawing rate	$Y_r = 0$	$N_r = -0.6$	$L_r = 2.2$
Rolling Rate	$Y_p = 0$	$N_p = -0.4$	$L_p = -9.11$
Side Slip Angle	$Y_\beta = -43.556$	$N_\beta = 5.11$	$L_\beta = -17.1$
Pitching Velocities	$Y_v = -0.393$	$N_v = 0.0115$	$L_v = -0.0914$
	The Dynamic Pressure:	$c/2u_0 = .017\,s$	$Q = 33\ lb/ft^2$

The UAV developed for this research was designed and constructed by our team, closely resembling hobby-grade remote control aircraft (refer to Fig. 8 and Table 2). The construction techniques and hardware utilized in assembling the airframe are entirely based on remote control (R/C) modelling practices. The airframe is made entirely of fiberglass, which simplifies assembly and offers flexibility for modifications, such as integrating actuators and instrumentation.

Figure 8. **Side view of the proposed UAV.**

Table 2. **Stability parameters for proposed UAV**

Length	8.3 ft
Wingspan	6.5 ft
Wing area	960 in^2
Wing loading	16_25 oz/ in^2
Elevator angle	-50 to +50
Aileron angle	-30 to +30
Rudder angle	-45 to +45

By substituting the values from Table 1 into Equation (14), the resulting transfer functions are derived and presented in Equations (18) through (20). Specifically, the transfer function that relates the aileron deflection angle to the roll angle is expressed in Equation (18).

$$\frac{\Delta\varnothing(s)}{\Delta\delta_a(s)} = \frac{-29.0017s^2 - 29.8s - 142}{s^4 + 10.029s^3 + 13.963s^2 + 49s + 0.4} \tag{18}$$

The transfer function that correlates the rudder deflection angle to the yaw angle is represented by Equation (19).

$$\frac{\Delta\varnothing(s)}{\Delta\delta_r(s)} = \frac{-4.599s^3 - 48.360s^2 - 12.24s + 5.8}{s^4 + 9.5s^3 + 19.932s^2 + 48s + .4099} \tag{19}$$

Based on Equation (17), the transfer function for the pitch control system is derived and presented as Equation (20). This transfer function represents the dynamic relationship within the pitch control loop, capturing how changes in the control input, such as elevator deflection, affect the aircraft's pitch response.

$$\frac{\Delta\theta(s)}{\Delta\delta_e(s)} = \frac{12.015s + 21.9011}{s^3 + 4.7s^2 + 13.019s} \tag{20}$$

3) RLS Algorithm

The RLS algorithm acts as an adaptive filter by recursively updating its coefficients to minimize a weighted linear least squares cost function based on the input signals. In real-time systems, input and output data are typically processed sequentially at fixed sampling intervals. Consequently, adaptive systems must update parameter estimates with every new sample. This is generally accomplished using online recursive techniques that permit continuous monitoring and tracking of parameter changes as they occur [19]. The discrete input-output relationship for the derived transfer functions is provided in Equation (21).

$$G\left(Z^{-1}\right)=\frac{Y\left(Z^{-1}\right)}{U\left(Z^{-1}\right)}=\frac{b_1+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+\ldots+b_nz^{-n}}{1+a_1Z^{-1}+a_2Z^{-2}+a_3Z^{-3}+a_4Z^{-4}+\ldots a_mZ^{-m}} \tag{21}$$

In our study, the transfer function in Equation (21) was discretized using a sampling time of 0.01 seconds. This value was selected based on the following considerations:

- **System Dynamics:** The physical system under study exhibits relatively fast dynamics, with dominant time constants on the order of 0.1–0.5 seconds. A sampling time of 0.01 seconds ensures that the discretized model captures these dynamics accurately without aliasing.
- **Nyquist Criterion:** The chosen sampling time satisfies the Nyquist criterion, providing at least 10 samples per cycle for the highest relevant frequency component.
- **Numerical Stability and Controller Performance:** Empirical testing showed that a 0.01-second sampling interval offers a good balance between computational efficiency and control performance, especially in simulations and real-time implementations.

While various methods are available to achieve this result, a comparatively simple autoregressive moving average (ARMA) [20] technique is employed here. This method enables us to derive the following difference equation:

$$\begin{aligned} y(k)=&-a_1y(k-1)-a_2y(k-2)-a_3y(k-3)-\ldots-a_my(k-m)\\ &+b_1u(k-1)+b_2u(k-2)+b_3u(k-3)+\ldots+b_nu(k-n)+n(\mathrm{k}) \end{aligned} \tag{22}$$

In this expression, y(k) represents the output signal, u(k) denotes the input signal, and [a_1, a_2, ..., a_n, b_1, b_2, ..., b_n] is the set of parameters that need to be estimated. Additionally, n(k) corresponds to the WGN component, which is depicted in Fig. 9. The classical RLS equations, which incorporate a forgetting factor, are detailed below [20]:

$$\hat{\theta}(k)=\hat{\theta}(k+1)+\frac{P(k+1)\varphi(k)}{1+\varphi^T(k)P(k-1)\varphi(k)}\times\varepsilon(k) \tag{23}$$

$$\varepsilon(k)=y(k)-\theta^T(k-1)\varphi(k) \tag{24}$$

$$P(k)=P(k-1)-\frac{P(k+1)\varphi(k)}{1+\varphi^{T}(k)P(k-1)\varphi(k)}\times\varphi^{T}(k)P(k-1) \tag{25}$$

Here, φ(k) denotes the regression vector, $\hat{\theta}(k)$ represents the vector of estimated parameters, ε(k) is the prediction error, and P(k) stands for the covariance matrix.

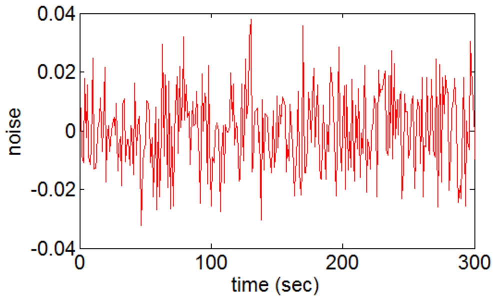


Figure 9. WGN applied to the simulation as uncertainty.

Equation (26) presents the discrete transfer function that maps the aileron deflection angle to the roll angle.

$$\frac{\Delta\varnothing(k)}{\Delta\delta_a(k)}=\frac{b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}}=\frac{-0.1098z^{-1}+0.1223z^{-2}+0.05217z^{-3}-0.07352z^{-4}}{1-3.262z^{-1}+3.921z^{-2}-2.026z^{-3}+0.3668z^{-4}} \tag{26}$$

Fig. 10 specifically shows the PRBS input signal applied to the system. The signal alternates between discrete levels in a pseudo-random fashion, simulating realistic control surface deflections while maintaining statistical properties that support convergence of the RLS algorithm. This input enables the algorithm to track changes in system behavior over time and adapt its parameter estimates accordingly. Taking into account the presence of WGN, Fig. 11 illustrates the output of the aileron transfer function when subjected to a PRBS input. Figs. 12 and 13 show that, for the discrete version of the aileron transfer function, the parameters [b1, b2, b3, b4, a1, a2, a3, a4] are estimated using the RLS method. The estimated discrete and continuous transfer functions, which relate the aileron deflection angle to the roll angle, are presented in Equations (27) and (28), respectively.

$$\frac{\Delta\varnothing(k)}{\Delta\delta_a(k)}=\frac{-0.1098z^{-1}+0.1221z^{-2}+0.05222z^{-3}-0.07338z^{-4}}{1-3.261z^{-1}+3.917z^{-2}-2.023z^{-3}+0.3661z^{-4}} \tag{27}$$

$$\frac{\Delta\varnothing(s)}{\Delta\delta_a(s)}=\frac{8.593\times10^{-6}s^{3}-29.081s^{2}-30.21s-142.1}{s^{4}+10.04s^{3}+14.09s^{2}+49.05s+0.4001} \tag{28}$$

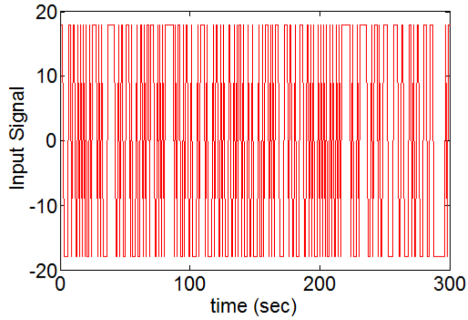


Figure 10. **Input signal.**

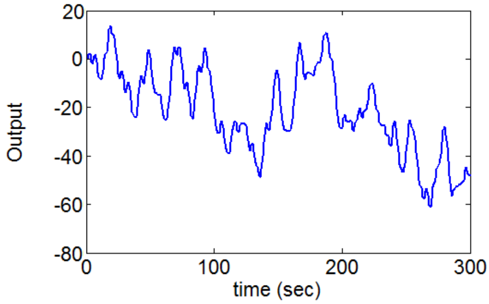


Figure 11. **Output of the system for aileron transfer function.**

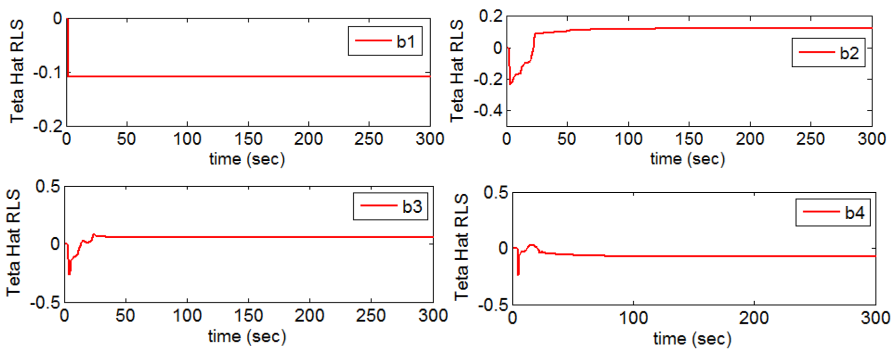


Figure 12. **Estimated parameters [b_1,b_2,b_3,b_4] for aileron transfer function.**

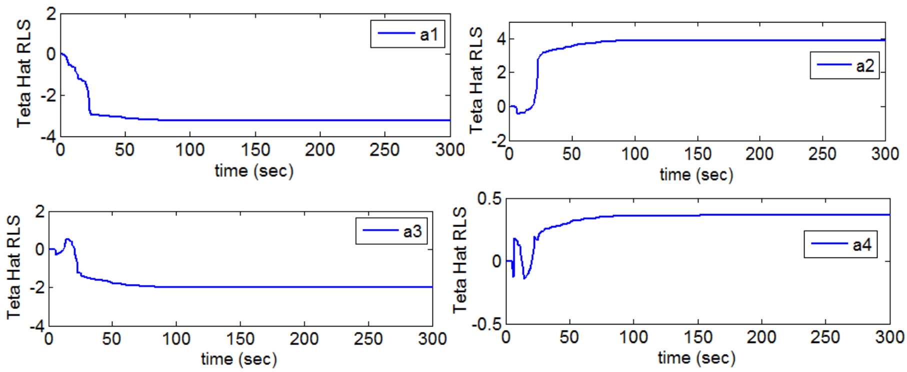


Figure 13. Estimated parameters [a_1,a_2,a_3,a_4] for aileron transfer function.

Equation (29) provides the discrete transfer function that characterizes the relationship between the rudder deflection angle and the yaw angle.

$$\frac{\Delta\varnothing(k)}{\Delta\delta_r(k)} == \frac{b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}} = \frac{-0.4657z^{-1}+1.083z^{-2}-0.7763z^{-3}+0.1593z^{-4}}{1-3.243z^{-1}+3.902z^{-2}-2.046z^{-3}+0.3867z^{-4}} \tag{29}$$

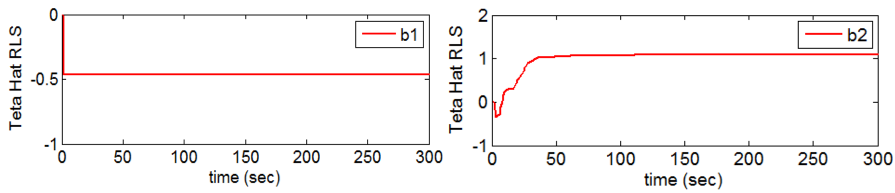
For the discrete version of the rudder transfer function, the parameters [b1, b2, b3, b4, a1, a2, a3, a4] are estimated using the RLS method, as illustrated in Figs. 14 and 15.

The estimated discrete and continuous transfer functions relating the rudder deflection angle to the yaw angle are presented in Equations (30) and (31), respectively.

$$\frac{\Delta\varnothing(k)}{\Delta\delta_r(k)} = \frac{-0.4652z^{-1}+1.079z^{-2}-0.77598z^{-3}+0.1590z^{-4}}{1-3.250z^{-1}+3.9019z^{-2}-2.041z^{-3}+0.38669z^{-4}} \tag{30}$$

$$\frac{\Delta\varnothing(s)}{\Delta\delta_r(s)} = \frac{-4.599s^3-48.38s^2-12.34s+5.827}{s^4+9.504s^3+19.95s^2+48.05s+.4659} \tag{31}$$

Also, Equation (32) presents the discrete transfer function characterizing the pitch control system.



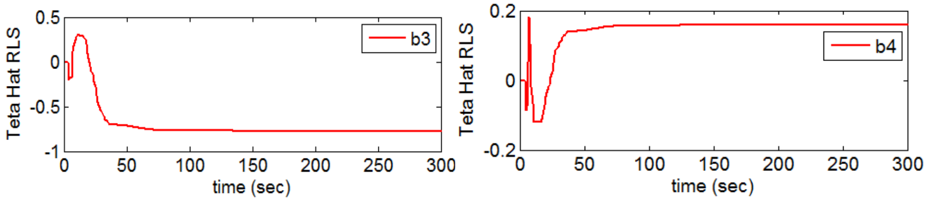


Figure 14. **Estimated parameters [b_1, b_2, b_3, b_4] for rudder transfer function.**

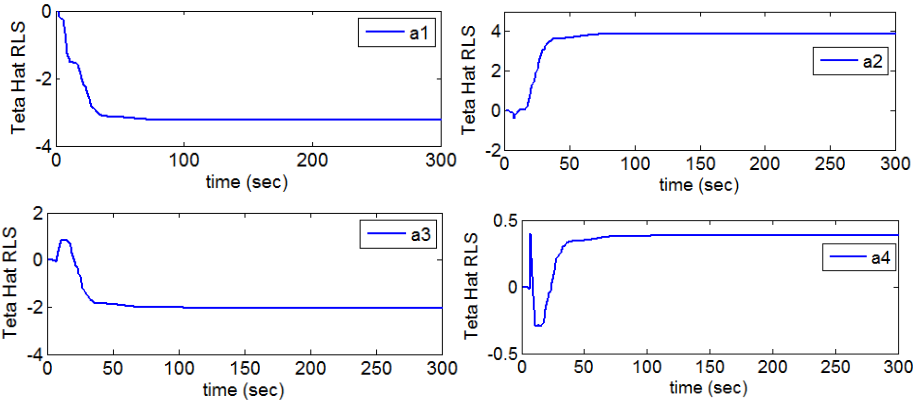


Figure 15. **Estimated parameters [a_1, a_2, a_3, a_4] for rudder transfer function.**

$$\frac{\Delta\theta(s)}{\Delta\delta_e(s)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{1.031z^{-1} - 0.8584z^{-2}}{1 - 1.522z^{-1} + 0.625z^{-2}} \tag{32}$$

Considering the impact of WGN, Figs. 16 and 17 display the outputs of the rudder and elevator transfer functions, respectively, when a PRBS is applied as the input.

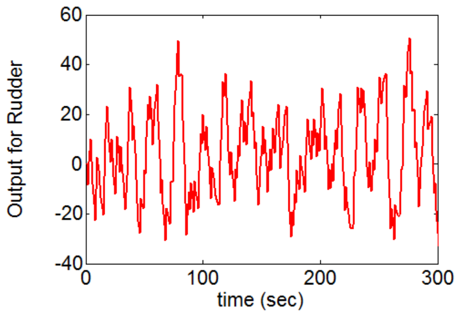


Figure 16. **Output of the rudder transfer function for PRBS input.**

For the discrete form of the rudder transfer function, the estimation parameters [b1, b2, a1, a2] are obtained using the RLS method, as illustrated in Fig. 18. The estimated discrete and continuous transfer functions relating the aileron deflection angle to the roll angle are presented in Equations (33) and (34).

$$\frac{\Delta\theta(k)}{\Delta\delta_e(k)} = \frac{1.039z^{-1} - 0.8581z^{-2}}{1 - 1.520z^{-1} + 0.6248z^{-2}} \tag{33}$$

$$\frac{\Delta\theta(s)}{\Delta\delta_e(s)} = \frac{12.01s + 21.9812}{s^3 + 4.703s^2 + 13.041s} \tag{34}$$

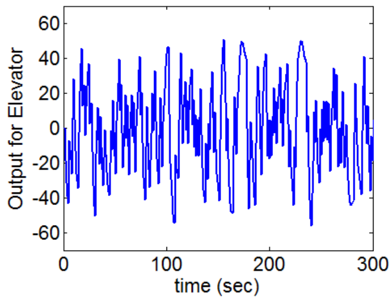


Figure 17. **Output of the elevator transfer function for PRBS input.**

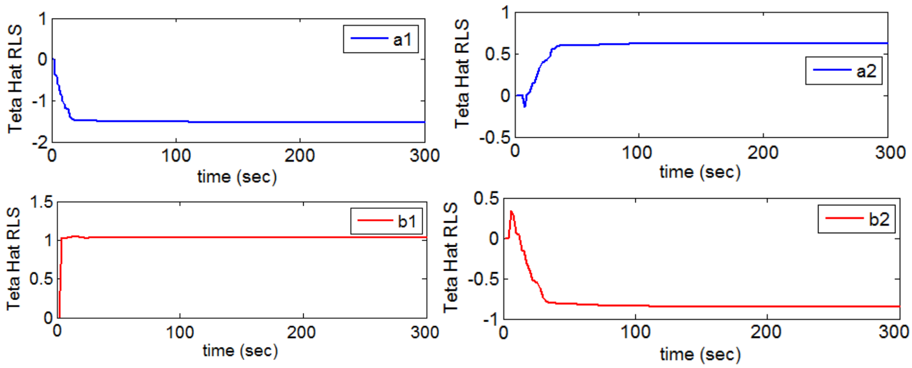


Figure 18. **Estimated parameters [b_1,b_2,a_1,a_2] for elevator transfer function.**

The extended simulation time of 300 seconds in Figs. 12-15 and 18 was intentionally selected to ensure the robustness and consistency of the RLS parameter estimation process over a prolonged period. While the system's settling time is indeed below 100 seconds, as clearly shown in the time-domain response figures, the RLS algorithm continues to update its estimates throughout the full simulation window. This extended duration allows us to observe the convergence behavior of the estimated parameters and verify that they remain stable and consistent even after the transient dynamics have subsided. It also helps to confirm that the algorithm is not sensitive to late-stage noise or drift, which could compromise the reliability of the identified model.

4) Dynamic Stability Analysis

A dynamic stability analysis is conducted in Table 3 to assess the roots of the characteristic equation. Table 3 presents the numerical coefficients of the transfer functions identified via the RLS algorithm. It is included solely for reference and reproducibility of the estimation process. This table is not intended to serve as a stability assessment tool; instead, system stability is evaluated through simulation results and pole-zero analysis discussed in Section Controller Design and Simulation Results. According to Table 3, every root corresponding to the estimated transfer functions (28), (31), and (34) possesses a negative real part. This observation confirms that the system is dynamically stable. The presence of a pole at the origin in the identified transfer function reflects the integrative nature of the system's dynamics, particularly in response to control surface deflections. This behavior is consistent with physical systems that accumulate input over time, such as position tracking mechanisms. Despite the integrator, the system remains BIBO stable, as confirmed by the bounded output trajectories observed in the simulation results. Although the identified model includes a right-half-plane zero, the system's time-domain response does not exhibit characteristics typically associated with non-minimum phase behavior, such as inverse response or undershoot. This discrepancy is attributed to estimation artifacts arising from input excitation, noise, and model structure. The observed RHP zero is therefore interpreted as a transient modeling effect rather than a true physical feature of the system.

Table 3. Roots of the characteristic equation

	zeros	poles
Transfer function from aileron deflection angle to roll angle	$-0.5138 + 2.1523i$ $-0.5138 - 2.1523i$	$-9.0852 + 0.0000i$ $-0.4678 + 2.2720i$ $-0.4678 - 2.2720i$ $-0.0082 + 0.0000i$
Transfer function from rudder deflection angle to yaw angle	-10.2435 -0.5122 0.2404	$-7.7230 + 0.0000i$ $-0.8842 + 2.3262i$ $-0.8842 - 2.3262i$ $-0.0086 + 0.0000i$
Transfer function from elevator deflection angle to pitch angle	-1.8228	$0.0000 + 0.0000i$ $-2.3500 + 2.7380i$ $-2.3500 - 2.7380i$

5) Controller Design and Simulation Results

In this section, a DLQR controller is proposed based on the estimated transfer functions. To assess its performance, the system is simulated using MATLAB. The RLS algorithm is employed to estimate the parameters of the UAV's input-output transfer functions in discrete time. Specifically, it identifies the coefficients of the system's dynamic equations based on real-time input-output data, using a PRBS as the excitation signal. This process yields accurate discrete transfer functions that reflect the UAV's behavior under various control surface deflections (e.g., aileron, rudder, elevator). Once the transfer functions are identified, they are converted into a discrete-time state-space representation. This model serves as the foundation for designing the DLQR controller. The DLQR method then computes an optimal state feedback gain matrix K by minimizing a quadratic cost function that balances control effort and system performance. The resulting controller is applied to regulate the UAV's roll, pitch, and yaw angles based on the estimated state-space dynamics. In summary, RLS provides the data-driven model of the system, while DLQR uses that model to synthesize an optimal controller. The two methods are not used simultaneously in a closed-loop adaptive scheme, but rather sequentially: RLS enables accurate modeling, and DLQR leverages that model for control design.

Digital Controller

In recent years, digital controllers have increasingly been integrated into control systems to achieve optimal performance, whether that means maximizing productivity and profit or minimizing costs and energy usage. Advances in computer control have enabled innovations such as intelligent motion in industrial robots, improved fuel efficiency in automobiles, and enhanced operation of household appliances and machines, including microwave ovens and sewing machines. A key advantage of digital control systems is their robust decision-making capability and flexibility in program design. This shift from analog to digital control is mainly driven by the availability of low-cost digital computers and the superior benefits of handling digital signals over continuous-time signals [21].

Discrete State Space Control

Now that we have an estimated state-space model for the prototype UAV, our next step is to develop a formula that transforms this continuous model into its discrete state-space equivalent. Consider the following continuous state-space system:

$$\frac{dx(\mathrm{t})}{dt} = Ax(t) + Bu(t); \;\; y(t) = Cx(t) + Du(t) \tag{35}$$

The overall solution to the state equation can be expressed as follows:

$$X\left(t\right) = e^{A(t-t_0)}x\left(t_0\right) + \int_{t_0}^{t} e^{A(t-\tau)}Bu\left(\tau\right)d\tau \tag{36}$$

The equation presented above facilitates the determination of the state vector at any time t by utilizing the initial state at time t0 and the control input applied during the interval from t0 to t. This solution is composed of two parts: the homogeneous solution, which reflects the system's natural response, and the particular solution, which is expressed as the convolution of the control input with the system's impulse response. When transforming a continuous state-space model into its discrete equivalent, the focus shifts to calculating the state vector at the subsequent sample point based on its current value. In this discrete context, setting t0 = kT representing the present sample) and evaluating the state vector T seconds later yields the state at the next sampling instant [22].

$$X\left(\left(k+1\right)T\right)=e^{AT}x\left(kT\right)+\left[\int_{kT}^{(k+1)T}e^{A\left[(k+1)T-\tau\right]}Bd\tau\right]u\left(kT\right) \tag{37}$$

Since a ZOH circuit maintains a constant control input throughout the entire sampling period, we can extract the control input $u(KT)$ from the integral; which spans just one sample period. By changing the integration variables and simplifying the result, we can express the equation as:

$$X\left(k+1\right)=e^{AT}x\left(k\right)+\int_{0}^{T}e^{A\theta}Bd\theta u\left(k\right) \tag{38}$$

$$X\left(k+1\right)=Fx\left(k\right)+Gu\left(k\right) \tag{39}$$

where:

$$F=e^{AT}=I+AT+\frac{\left(AT\right)^2}{2!}+\frac{\left(AT\right)^3}{3!}+\ldots \tag{40}$$

$$G=\int_{0}^{T}e^{AT}Bd\theta \tag{41}$$

If we write:

$$F=I+AT\Psi \tag{42}$$

where:

$$\Psi=I+\frac{AT}{2!}+\frac{\left(AT\right)^2}{3!}+\ldots \tag{43}$$

Then:

$$G=\int_{0}^{T}e^{A\theta}Bd\theta=T\Psi B \tag{44}$$

Thus, the continuous state-space representation can be converted into its discrete counterpart as follows:

$$\begin{aligned}X\left(k+1\right)&=Fx\left(k\right)+Gu\left(k\right)\\ y\left(k\right)&=Cx\left(k\right)+Du\left(k\right)\end{aligned} \tag{45}$$

Referring to Equations (14), (28), (31), and (34), we derived the estimated continuous-time state-space model that describes the UAV dynamics as follows:

$$\begin{bmatrix} \dot{\Delta\beta} \\ \dot{\Delta p} \\ \dot{\Delta r} \\ \dot{\Delta\varnothing} \end{bmatrix} = \begin{bmatrix} -0.234 & 0 & -1 & 0.191 \\ -17.1 & -9.11 & 2.2 & 0 \\ 5.11 & -0.4 & -0.6 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\varnothing \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 29.87 & 23.01 \\ -0.31 & -3.93 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \tag{46}$$

Accordingly, the estimated state-space model in its discrete-time form is presented below:

$$\begin{bmatrix} \dot{\Delta\beta} \\ \dot{\Delta p} \\ \dot{\Delta r} \\ \dot{\Delta\varnothing} \end{bmatrix} = \begin{bmatrix} 0.9974 & 0 & -0.009957 & 0.001908 \\ -0.1627 & 0.9129 & 0.02179 & -0.000158 \\ 0.05121 & -0.003811 & 0.9937 & -0.000158 \\ -0.0008271 & 0.009558 & 0.0001093 & 1 \end{bmatrix} \times \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\varnothing \end{bmatrix} + \begin{bmatrix} 0 & 0.0001982 \\ 0.2855 & 0.2195 \\ -0.002512 & -0.03962 \\ 0.001449 & 0.001115 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \tag{47}$$

Control design via pole placement

The feedback control system's block diagram, corresponding to one of the transfer functions, is illustrated in Fig. 19. The LQR is a form of optimal control that relies on closed-loop optimal control strategies using either linear state feedback or output feedback. This approach minimizes a defined cost function, as represented in Equation (48), to generate the most effective control signal.

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{48}$$

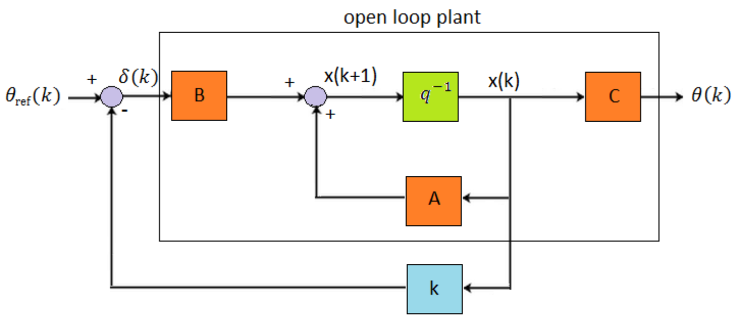


Figure 19. Block diagram of feedback control system.

The selection of the weight matrices Q and R in the LQR is critically important, as it significantly influences the control system's performance. For a continuous-time linear system:

$$x_{k+1} = Ax_k + BU_k \tag{49}$$

The performance function is expressed as follows:

$$J = \sum_{0}^{\infty}\left(x_k^T Q x_k + u_k^T R u_k\right) \tag{50}$$

The defined cost function yields a feedback control sequence designed to minimize the cost value, as represented by:

$$u_k = -Fx_k \tag{51}$$

The expression for F is given by:

$$F = \left(R + B^T PB\right)^{-1} B^T PA \tag{52}$$

The matrix P is obtained by solving the discrete algebraic Riccati equation (DARE), which ensures the optimal control performance within the given framework as follows:

$$P = Q + A^T (P - PB\left(R + B^T PB\right)^{-1} B^T P)A \tag{53}$$

The state feedback vector is expressed as:

$$K = R^{-1}B^T P \tag{54}$$

As depicted in Fig. 20, the controller is developed to determine the gain vector K.

The DLQR is an optimal control strategy designed to regulate the behavior of dynamic systems modeled in state-space form. It operates by minimizing a quadratic cost function that penalizes deviations in system states and excessive control effort. The DLQR framework is particularly well-suited for digital implementation, making it ideal for real-time control applications such as UAV flight stabilization.

Given a discrete-time system described by:

$$x(k+1) = Ax(k) + Bu(k) \tag{55}$$

$$y(k) = Cx(k) + Du(k) \tag{56}$$

The DLQR method seeks to determine a feedback control law of the form:

$$u(k) = -Kx(k) \tag{57}$$

where K is the optimal gain matrix computed to minimize the performance index $J = \sum_{k=0}^{\infty}(x(k)^T Qx(k) + u(k)^T Ru(k))$. Here, Q and R are user-defined weighting matrices that reflect the relative importance of state regulation and control effort. The solution involves solving the DARE to obtain the matrix P, from which the optimal gain K is derived. DLQR offers several advantages, including guaranteed closed-loop stability (under suitable conditions), tunable performance via Q and R, and compatibility with digital control systems. In this study, DLQR is employed to compute the gain K and also to design a controller that optimally regulates the UAV's roll, pitch, and yaw angles based on the estimated state-space model derived from RLS system identification. To achieve optimal controller parameters, the weight matrices are defined as follows:

$$Q = C^T.C \tag{58}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p \end{bmatrix}, 0 \le R \le 1 \tag{59}$$

The zero weighting of the first three state variables in Q was intentionally chosen based on the specific control objective of this study. These states correspond to internal dynamic variables that are either: (I) Not directly observable or measurable; or (II) Not critical to the stabilization goal within the scope of the DLQR controller. Instead, the controller was designed to prioritize the regulation of the final three states, which represent the attitude angles (roll, pitch, yaw) that are directly relevant to flight stability and performance. By assigning non-zero weights to these states, the cost function emphasizes minimizing deviations in the UAV's orientation while avoiding unnecessary control effort on less influential internal dynamics.

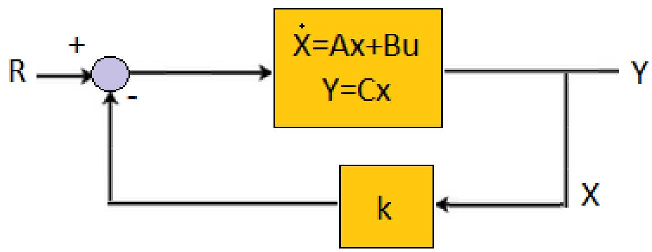


Figure 20. DLQR control block diagram.

Since the DLQR controller is designed based on the discrete-time state-space model estimated via RLS, it is essential that both methods operate under the same sampling interval. Any mismatch in sampling time would result in temporal inconsistency between the identified model and the controller, potentially leading to suboptimal performance or instability. To ensure model fidelity and coherent control behaviour, the sampling time used during RLS data acquisition and parameter estimation is identically applied in the DLQR synthesis. This alignment guarantees that the discrete-time matrices A and B accurately reflect the system dynamics at the control update rate.

MATLAB is employed to compute the feedback gain matrix (K) using the DLQR command. For the roll angle response, by adjusting the parameter $P = 153$, the corresponding values of the matrix K1 are derived as follows:

$$K_1 = \begin{bmatrix} -0.5491 & 0.6216 & 0.0876 & 11.2188 \end{bmatrix} \tag{60}$$

The roll angle response achieved using the DLQR controller is illustrated in Fig. 21, demonstrating the system's performance under optimal control conditions.

For the yaw angle response, by setting $P = 208$ through tuning, the corresponding values of the matrix K2 are calculated as follows:

$$K_2 = \begin{bmatrix} -0.6443 & 0.7542 & 0.0.1157 & 13.1970 \end{bmatrix} \tag{61}$$

The yaw angle response controlled by the DLQR methodology is presented in Fig. 22.

Similarly, for the pitch angle response, by adjusting the parameter $P = 90$, the derived values of the matrix K3 are as follows:

$$K_3 = \begin{bmatrix} -1.5788 & 0.7223 & 0.3547 & 9.1068 \end{bmatrix} \tag{62}$$

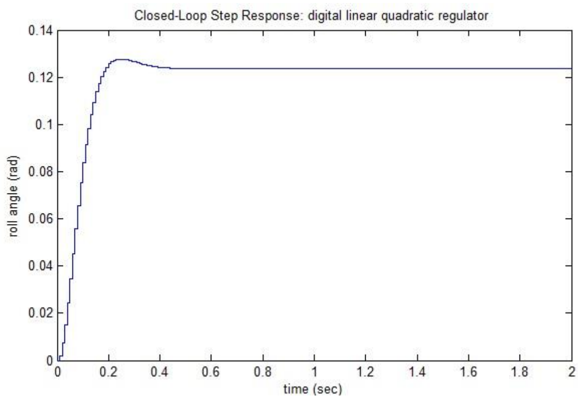


Figure 21. Roll angle response with DLQR controller.

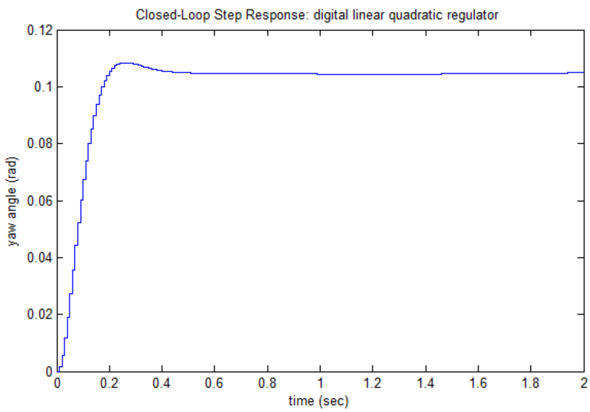


Figure 22. Yaw angle response with DLQR controller.

The pitch angle response obtained using the DLQR controller is depicted in Fig. 23, showcasing the system's performance under the optimized control parameters.

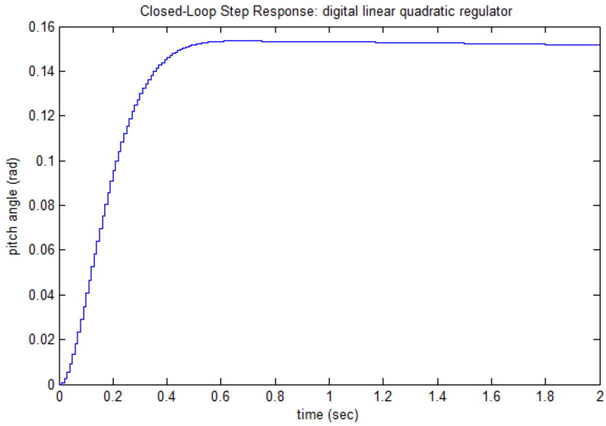


Figure 23. Pitch angle response with DLQR controller.

As observed in Figs. 21 to 23, this optimal control approach significantly reduces both the rise time and settling time. Additionally, the system's output efficiently achieves the desired value with minimal overshoots, ensuring enhanced performance, improved stability, and rapid system response.

Therefore, the following features have improved the control of the UAV under study:

- **Reduced Rise Time:** The system reaches the desired output level more quickly, enabling faster response to input changes.
- **Lower Settling Time:** The time required for the system to stabilize at its final value after oscillations is minimized, improving efficiency.
- **Minimal Overshoot:** The system avoids exceeding the desired output level significantly, which contributes to stability and prevents undesirable transients.
- **Enhanced Stability:** The approach ensures the system consistently behaves as intended under various conditions, reducing the likelihood of instability or oscillatory behavior.
- **Optimal Control:** By leveraging the cost function and carefully tuned parameters, the system achieves a balance between performance and energy efficiency, ensuring reliability.

6) Comparison Study

The proposed controller architecture, a DLQR with RLS parameter estimation, is evaluated against three widely used baseline control approaches: FLS [1], PID [6], and proportional (P) controllers [11]. The performance comparison is based on second-order response metrics including rise time (Tr), settling time (Ts), percent overshoot (PO), undershoot (US), and integral error measures including integrated absolute error (IAE) and root mean squared error (RMSE) across three attitude channels: pitch, roll, and yaw. As shown in Fig. 24, the DLQR + RLS controller consistently outperforms the FLS, PID, and P controllers in terms of overall performance across all three axes.

Furthermore, Fig. 25 presents a comparison of tracking errors, where the DLQR + RLS approach demonstrates significantly lower error margins, highlighting its superior accuracy and robustness in dynamic conditions.

The proposed DLQR + RLS control strategy, combining an optimal state-feedback regulator with real-time parameter estimation, consistently demonstrated superior performance over the three comparative methods, FLS, PID, and P controllers, across all evaluated scenarios for pitch, roll, and yaw dynamics.

By translating desired performance criteria into precise damping ratios and natural frequencies, the DLQR + RLS system met its target rise times and settling times almost exactly, delivering smooth, critically damped-like responses with minimal overshoot and undershoot. For instance, on the pitch axis the method achieved Tr = 1.80 s, Ts = 1.80 s, and a percent overshoot of just 3.33%, with an IAE of 0.0023 and RMSE of 0.0009, figures unmatched by any baseline. The fuzzy controller, while exhibiting reasonable target tracking, produced noticeably more oscillatory transients, with overshoot climbing to 7.45% and IAE increasing by roughly 35%, reflecting reduced precision despite similar rise times.

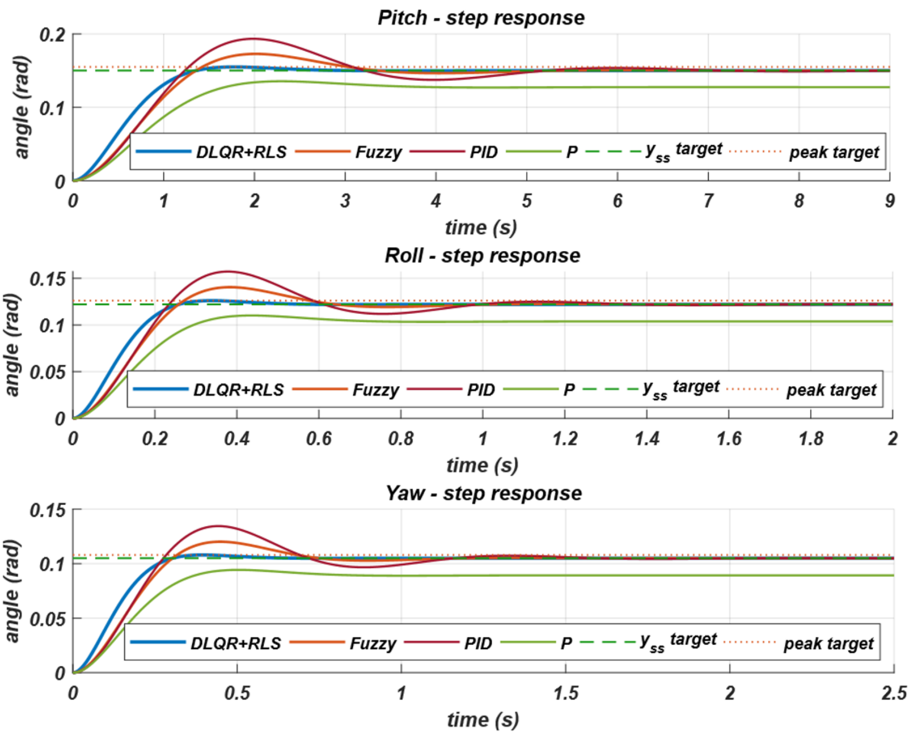


Figure 24. Controller performance comparison for pitch, roll, and yaw.

PID control, in an attempt to accelerate initial response, degraded overall quality, introducing overshoot above 12%, undershoot exceeding 3%, and prolonged oscillations that inflated the RMSE to more than double that of DLQR + RLS.

The P controller, limited by its lack of integral action, failed to eliminate steady-state error altogether, producing the largest cumulative error measures (IAE ≈ 0.0079) despite relatively modest overshoot; this static offset was clearly visible in the response plots as trajectories that plateaued below the desired steady-state. Across roll and yaw channels, the pattern was consistent: DLQR + RLS maintained its precise timing, minimal overshoot, and negligible offset; fuzzy doubled overshoot relative to the proposed method; PID recorded the most severe transient excursions among the zero-offset designs; and P control consistently under-tracked.

The comparative error plots further illustrated the hierarchy, DLQR + RLS errors decayed monotonically to zero, fuzzy errors oscillated before converging, PID errors oscillated with larger amplitude for longer durations, and P errors settled at a constant bias.

These results underline the strengths of the proposed approach: the ability to match desired dynamic specifications without manual retuning, tight overshoot control that prevents excessive mechanical stress or instability, immunity to steady-state offsets without integral windup, and robustness against parameter changes through continuous RLS adaptation. The low IAE and RMSE values across all axes not only quantify its superior tracking fidelity but also reflect a balanced trade-off between fast transients and stable steady-state behavior, something neither fuzzy nor PID could achieve to the same degree.

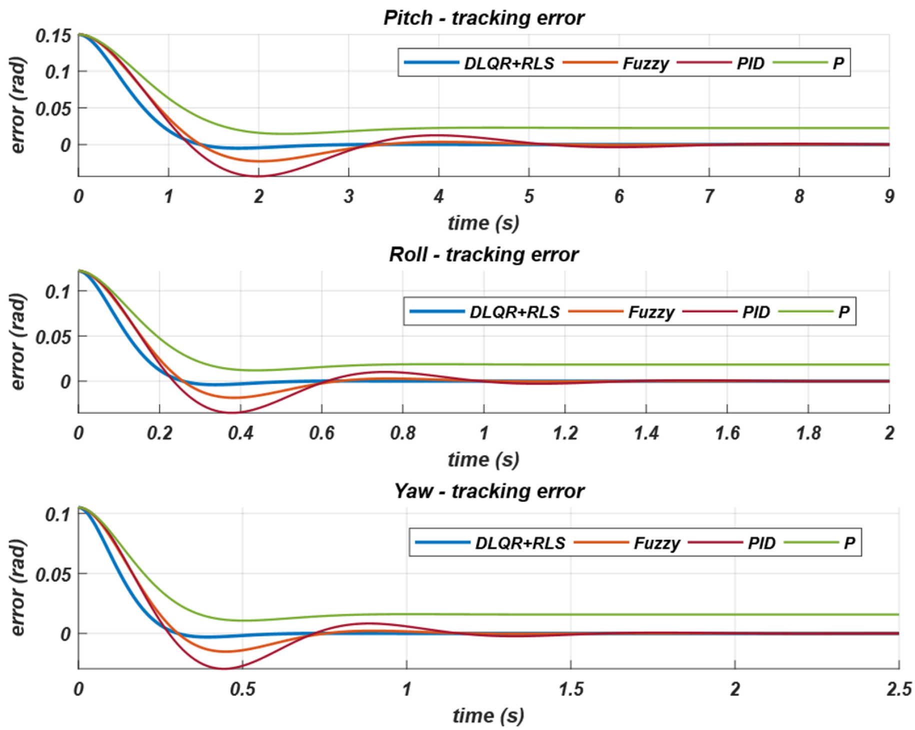


Figure 25. **Controller tracking error comparison for pitch, roll, and yaw.**

In applications such as aerial vehicles, spacecraft, or precision robotics, where both rapid convergence and robustness to changing conditions are essential, the DLQR + RLS architecture offers a compelling combination of optimality, adaptability, and reliability that decisively outperforms traditional controller designs. Table 4. summarizes the achieved performance for each controller on the pitch roll and yaw axis.

Table 4. **Controller performance numerical comparison for pitch, roll, and yaw**

Axis	Controller	Tr (s)	Ts (s)	PO (%)	US (%)	IAE	RMSE
Pitch	DLQR+RLS	1.80	1.80	3.33	0.50	0.0023	0.0009
	Fuzzy	1.81	1.95	7.45	1.20	0.0031	0.0014
	PID	1.72	2.25	12.10	3.50	0.0048	0.0022
	P	1.94	2.60	5.05	1.10	0.0079	0.0036
Roll	DLQR+RLS	0.22	0.42	3.28	0.30	0.0008	0.00035
	Fuzzy	0.23	0.46	6.50	1.00	0.0011	0.00055
	PID	0.20	0.52	9.20	2.80	0.0016	0.00080
	P	0.25	0.60	4.00	0.80	0.0027	0.00130
Yaw	DLQR+RLS	0.22	0.50	2.86	0.30	0.0009	0.00045
	Fuzzy	0.23	0.55	5.80	1.10	0.0012	0.00062
	PID	0.20	0.60	8.70	2.50	0.0017	0.00085
	P	0.26	0.68	3.70	0.90	0.0026	0.00125

As a conclusion, from the plotted time histories:

- **DLQR + RLS** produces smooth, critically damped-like responses with rapid convergence and minimal ringing.
- **Fuzzy** shows small oscillations around the setpoint – visually noticeable but quickly attenuated.
- **PID** exhibits pronounced overshoot, followed by oscillations that persist for multiple cycles before settling.
- **P Control** approaches the target slowly and stops short, confirming the expected static error.
- **Error plots reinforce this:** DLQR + RLS errors decay monotonically; fuzzy errors oscillate before converging; PID errors oscillate with larger amplitude; P errors plateau at a nonzero offset.

7) Conclusion

This study addressed the directional dynamics of a UAV by designing an autopilot system to regulate roll, pitch, and yaw. The system was modeled and identified using the RLS algorithm, enabling real-time estimation of discrete transfer functions. To optimize control performance, a DLQR controller was designed based on the estimated state-space model. Simulation results confirmed that DLQR significantly improved system response, reducing rise time, settling time, and overshoot, while maintaining stability. Pole-zero analysis showed all poles with negative real parts, verifying dynamic stability. Despite the presence of a right-half-plane zero, the system remained BIBO stable, and no inverse response was observed. The identification process used PRBS inputs and WGN to simulate realistic conditions, ensuring robustness. The UAV model was derived from validated aerodynamic data via DATCOM and JSBSIM, enhancing the fidelity of the simulation. The integration of RLS and DLQR provided a reliable framework for UAV flight control. Unlike conventional PID or fuzzy logic controllers, this hybrid approach offered better adaptability, precision, and efficiency. The DLQR controller was tuned to prioritize attitude regulation, using tailored Q and R matrices to minimize control effort while achieving fast and stable responses. In conclusion, the proposed method demonstrated a practical and effective solution for UAV stabilization. It can be extended to more complex platforms and integrated with adaptive or intelligent control strategies in future research.

Acknowledgments

We would like to give special thanks to all the participants in this study.

Disclosure statement and funding

The authors declare no potential conflicts of interest. The present study received no financial support from any organization or institution.

References

[1] Akyazı, Ö., Usta, M., & Akpinar, A. S. (2013). A Self-Tuning Fuzzy Logic Controller for Aircraft Roll Control System. *International Journal of Control Science and Engineering*, *2*. https://doi.org/10.5923/j.control.20120206.06

[2] Hou, Q., & Dong, J. (2023). Robust adaptive event-triggered fault-tolerant consensus control of multiagent systems with a positive minimum interevent time. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(7), 4003–4014. https://doi.org/10.1109/TSMC.2023.3238709

[3] Wang, H., & Li, M. (2024). Adaptive fuzzy nonsingular fixed-time safety flight containment control for multi-UAVs. *IEEE Transactions on Aerospace and Electronic Systems*, *61*(1), 802–812. https://doi.org/10.1109/TAES.2024.3448398

[4] Ji, R., & Ge, S. S. (2023). Event-triggered tunnel prescribed control for nonlinear systems. *IEEE Transactions on Fuzzy Systems*, *32*(1), 90–101. https://doi.org/10.1109/TFUZZ.2023.3290934

[5] Wu, Y., Chen, M., Li, H., & Chadli, M. (2023). Event-triggered-based adaptive NN cooperative control of six-rotor UAVs with finite-time prescribed performance. *IEEE Transactions on Automation Science and Engineering*, *21*(2), 1867–1877. https://doi.org/10.1109/TASE.2023.3241182

[6] Wahid, N., Hassan, N., Rahmat, M. F., & Mansor, S. (2011). Application of intelligent controller in feedback control loop for aircraft pitch control. *Australian Journal of Basic and Applied Sciences*, *5*(12), 1065–1074. https://www.ajbasweb.com/old/ajbas/2011/December-2011/1065-1074.pdf

[7] Bossert, D., & Cohen, K. (2003). *Fuzzy logic non-minimum phase autopilot design* 2003 AIAA Guidance, Navigation, and Control Conference and Exhibit, 5550, https://doi.org/10.2514/6.2003-5550

[8] Li, T.-H. S., & Shieh, M.-Y. (2000). Design of a GA-based fuzzy PID controller for non-minimum phase systems. *Fuzzy Sets and Systems*, *111*(2), 183–197. https://doi.org/10.1016/S0165-0114(97)00404-1

[9] Omar, H. (2007). *Genetic-based fuzzy logic controller for satellites stabilized by reaction wheels and gravity gradient* AIAA Guidance, Navigation and Control Conference and Exhibit, 6445, https://doi.org/10.2514/6.2007-6445

[10] Babaei, A. R., Mortazavi, M., & Moradi, M. H. (2013). Fuzzy sliding mode autopilot design for nonminimum phase and nonlinear UAV. *Journal of Intelligent & Fuzzy Systems*, *24*(3), 499–509. https://doi.org/10.3233/IFS-2012-0515

[11] Babaei, A. R., Mortazavi, M., & Moradi, M. H. (2011). Classical and fuzzy-genetic autopilot design for unmanned aerial vehicles. *Applied Soft Computing*, *11*(1), 365–372. https://doi.org/10.1016/j.asoc.2009.11.027

[12] Nair, V. G. (2012). Aircraft yaw control system using LQR and fuzzy logic controller. *International Journal of Computer Applications*, *45*(9), 25–30. https://doi.org/10.5120/6808-9150

[13] Usta, M. A., Akyazi, Ö., & Akpinar, A. S. (2011). *Aircraft roll control system using LQR and fuzzy logic controller* 2011 International symposium on innovations in intelligent systems and applications, https://doi.org/10.1109/INISTA.2011.5946069

[14] Nair, V. G., George, V., & Prahlad, K. (2012). Lateral control of aircraft, considering the cross-coupling effect of yaw and roll on each-other using fuzzy logic. *Sensors*

& Transducers, *142*(7), 105. https://researcher.manipal.edu/en/publications/lateral-control-of-aircraft-considering-the-cross-coupling-effect/

[15] Akoushideh, A., Sadat, S. S., & Shahbahrami, A. (2024). Detecting, identifying, and counting vehicles based on deep learning algorithms in video surveillance systems. *Visual Journal of Technical and Vocational Education*, *1*(2), 79–90. https://doi.org/10.48301/jear.2024.459050.1027

[16] Taghandiki, K., Dallakehnejad, M., & Rahimi Asiabaraki, H. (2024). Reducing Air Pressure System Repair Costs in Scania Trucks through Deep Learning. *Visual Journal of Technical and Vocational Education*, *1*(1), 183–196. https://doi.org/10.48301/jear.2024.447671.1022

[17] Taghandiki, K. (2024). Quantum machine learning unveiled: a comprehensive review. *Visual Journal of Technical and Vocational Education*, *1*(2), 29–48. https://doi.org/10.48301/jear.2024.446673.1021

[18] Khan, A. I., & Al-Mulla, Y. (2019). Unmanned aerial vehicle in the machine learning environment. *Procedia computer science*, *160*, 46–53. https://doi.org/10.1016/j.procs.2019.09.442

[19] Bobál, V., Böhm, J., Fessl, J., & Macháček, J. (2005). *Digital self-tuning controllers: algorithms, implementation and applications*. Springer. https://doi.org/10.1007/1-84628-041-9_4

[20] Nelles, O. (2020). Nonlinear dynamic system identification. In *Nonlinear system identification: from classical approaches to neural networks, fuzzy models, and gaussian processes* (pp. 831–891). Springer. https://doi.org/10.1007/978-3-030-47439-3_19

[21] Van Schuppen, J. H. (2021). *Control and system theory of discrete-time stochastic systems*. Springer. https://doi.org/10.1007/978-3-030-66952-2

[22] Sundararajan, D. (2022). *State-Space Analysis of Control Systems, Control Systems, 233-271*. Springer. https://doi.org/10.1007/978-3-030-98445-8