DOI 10.52209/2706-977X_2025_3_111 IRSTI 55.47.07 UDC 623.746

Study of FLC, PID, and LQR Control Methods for Precise Drone Landing in Wind Conditions

Ormanov A.N.[1,*], Dyusekeev K.A.[2], Shidlovskiy S.V.[3]
[1]Astana IT University, Astana, Kazakhstan
[2]L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
[3]National Research Tomsk State University, Tomsk, Russia
*corresponding author

Abstract. The article presents a comparative analysis of three controllers — FLC, PID, and LQR — for precise drone landing under external disturbances such as wind. The simulation was performed in MATLAB/SIMULINK with 3D animated models. The controllers were tested under identical initial conditions, evaluating landing accuracy and execution time. The results showed that the FLC demonstrated robustness to uncertainties (85.47% accuracy, 9.2 s), but required more computational resources. The PID controller provided 90.37% accuracy in 4.5 s, remaining simple and effective. The LQR achieved the best results — 98.98% accuracy in 4.5 s, demonstrating optimality and stability. The choice of controller depends on the conditions: FLC for complex uncertain situations, PID for simple tasks, and LQR for maximum accuracy and stability. The obtained data can be used in the development of drone control systems.

Keywords: drone, quadcopter, fuzzy logic controller (FLC), PID Controller, linear-quadratic regulator (LQR), precise drone landing, external disturbances, MATLAB/SIMULINK, drone control.

Introduction

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, have become increasingly significant in various sectors such as agriculture, logistics, infrastructure monitoring, and emergency response [9]. Their ability to operate in environments that are hazardous or inaccessible to humans has made them essential tools for automation and real-time data collection [8]. In particular, the problem of autonomous landing has gained considerable attention, as it is a critical phase for ensuring the safety, endurance, and efficiency of UAV operations [11]. Precise landing is especially important for scenarios such as docking on charging stations, landing on moving platforms, or performing missions in urban and confined environments [13].

The task of autonomous landing poses several challenges. UAVs are often subjected to external disturbances such as wind, turbulence, and sudden payload variations, which may lead to instability or positional inaccuracy [4]. Moreover, conventional control strategies may not be sufficient when dealing with uncertainties, nonlinearities, and rapidly changing operating conditions. As a result, robust and adaptive control algorithms are needed to guarantee stable and accurate landings [10].

Among the most commonly applied methods, the Proportional-Integral-Derivative (PID) controller remains popular due to its simplicity and ease of implementation [10], [16]. However, its performance is limited under nonlinear and highly dynamic conditions. The Fuzzy Logic Controller (FLC) has been introduced to handle uncertainty and approximate reasoning, offering better adaptability compared to classical methods [1], [3]. On the other hand, the Linear Quadratic Regulator (LQR), derived from optimal control theory, provides mathematically grounded solutions that minimize control effort while maintaining system stability [11], [12]. Several studies have demonstrated the successful application of these controllers in UAV path tracking, altitude stabilization, and trajectory control [13], [14], yet their comparative performance in precision landing remains underexplored.

Therefore, this study aims to fill this research gap by conducting a comparative analysis of FLC, PID, and LQR controllers for precise drone landing under external disturbances such as wind. Through simulation in MATLAB/Simulink, the effectiveness of each control method is evaluated in terms of accuracy and stability. The findings of this research are expected to contribute to the development of more reliable UAV landing systems, thereby enhancing their applicability in real-world missions.

1. Methodology

This section discusses the development of a drone control system using Fuzzy Logic Controller (FLC), Proportional-Integral-Derivative (PID) controller, and Linear-Quadratic Regulator (LQR). Each controller has unique features, advantages, and limitations, making them suitable for different scenarios. Mathematical models were developed in MATLAB/SIMULINK, considering the drone's motion dynamics and wind effects, to demonstrate and compare the effectiveness of each controller. The following subsections provide detailed approaches to designing and testing FLC, PID, and LQR under real-world conditions.

For the development of the model in MATLAB/SIMULINK, identical initial conditions were set for all controllers:

- X error — 2 m.
- Y error — 1.5 m.
- Z error — 5 m.
- Rotation angle — 10°.

- Wind speed — 2 m/s.
- Wind direction — 45°.
- Drone weight — 2 kg.

These parameters allow for a proper comparison of the controllers' effectiveness during drone landing, and they can be adjusted to analyze different scenarios.

Fuzzy Logic Controller (FLC)

In this subsection, we will develop fuzzy control algorithms for drone landing.

The advantages of using a fuzzy logic system lie in the ability to apply intuitive data and expert knowledge about the controlled object, as well as in the absence of the need for an exact mathematical model [4].

For the development of the FLC, MATLAB with the FuzzyLogicDesigner library will be used [2], [15]. The Mamdani fuzzy logic system, one of the most popular models, has been chosen. In this system, rules are described in an "if-then" format, which allows for flexible consideration of parameters affecting landing [3]. For example, a rule could be: "If the drone's altitude is high (Z = HIGH), then the descent speed is low (Uz = DOWN_HIGH)."

The system will use 6 input variables: errors along the X (Ex) and Y (Ey) axes, altitude errors (Z), yaw angle (θ), and wind parameters — wind speed (WindForce) and wind direction (WindDir). The output will consist of 4 control variables: along the X (Ux) and Y (Uy) axes, altitude (Uz), and rotation angle (Utheta). The system diagram is presented in Fig. 1.

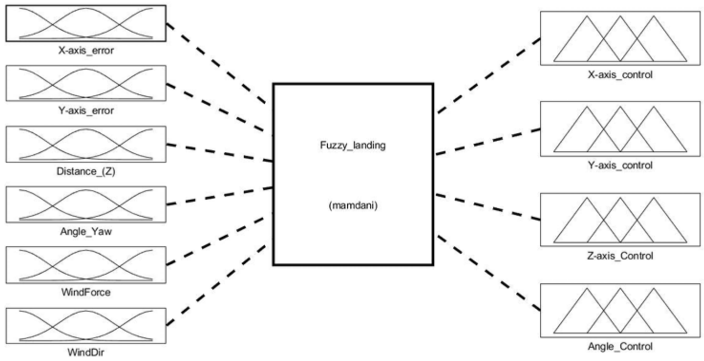

Fig. 1. - Fuzzy logic system diagram for drone landing control

Next, for convenience and to simplify the rule base, we performed fuzzy logic grouping. Dividing the system into logical blocks (such as input and output variables, membership functions, and rules) allows for efficient management of individual components and enables changes to be made without the need to revise the entire control logic [3], [2].

As a result, the system was divided into three groups:

Positioning errors (X, Y).

Errors in altitude and angle.

Compensation for wind influence.

This approach allows for more flexible tuning of the control system and adapting it to different operating conditions. The grouped components of the system are shown in Fig. 2.

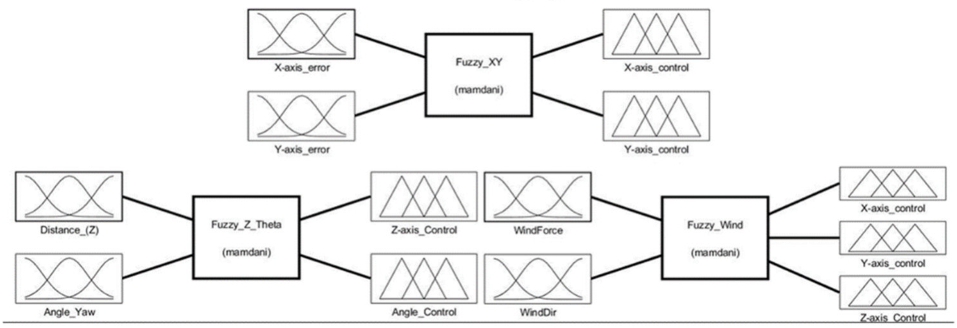

Fig. 2. - Grouping of fuzzy control system components

Positioning Errors (X, Y): The Fuzzy_XY FLC, shown in Figure 1, adjusts the drone's position in the horizontal plane (X and Y axes) [5], minimizing deviations from the target coordinates. During landing, the drone may shift due to navigation errors, wind disturbances, or control characteristics. To ensure precise landing, the system analyzes the current position and generates corrective control actions to eliminate horizontal coordinate errors.

Input Variables:

- Error along the X-axis (Ex): Determines the drone's left or right displacement relative to the landing point (range from -2 to 2 meters) [1].
- Error along the Y-axis (Ey): Similar to Ex but indicates forward or backward deviation.

Output Variables:

- Control signals for the X-axis (Ux) and Y-axis (Uy): Normalized within the range [−1, 1]. The system generates corrective actions to return the drone to the designated point. Control along the X-axis compensates for deviations along X, while control along the Y-axis regulates movement along Y.

Triangular Membership Functions (MFs): Defined as follows, following a similar approach as in [12]:

$$\mu_{BigLeft} = \begin{cases} 1, & x \leq -1.5 \\ \dfrac{-x-1}{0.5}, & -1.5 < x < -1.0 \\ 0, & x \geq -1.0 \end{cases} \tag{1}$$

Similar functions are defined for other linguistic variables: SmallLeft, Zero, SmallRight, and **BigRight**. Detailed membership functions are shown in Fig. 3.

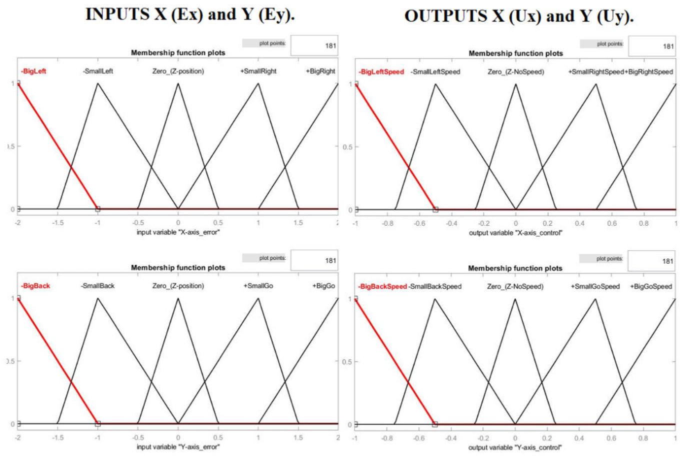

Fig. 3. - Membership Functions for X(Ex), Y(Ey), X(Ux), and Y(Uy)

A key stage is the creation of a rule base for correcting errors along the X and Y axes, establishing the relationship between input and output variables.

For each group of variables, a rule base is developed to define the relationships between input and output variables. The centroid method is used for defuzzification [3].

$$U = \frac{\Sigma_i \mu_{ix} U_i}{\Sigma_i \mu_i} \tag{2}$$

where μi - is the degree of membership of the output variable Ui in the given rule.

Table 1. Rule base for correcting X and Y errors

#	Input (Ex)	Input (Ey)	Output (Ux)	Output (Uy)
1	BigLeft	BigLeft	BigRight	BigRight
2	BigLeft	SmallLeft	BigRight	SmallRight
3	BigLeft	Zero	BigRight	Zero
25	…	…	…	…

Errors in height (Z) and angle (Theta): The Fuzzy_Z_Theta block, shown in Figure 1, controls the drone's height and orientation during landing, ensuring smooth and safe descent. The drone adjusts its horizontal position, regulates the height to avoid hitting the surface, and, if necessary, corrects the yaw angle [6].

Input Variables:

- Distance to the marker (Z): The drone's height (0–10 meters)
- Yaw angle (Theta): The orientation angle relative to the landing pad (-30° to 30°)

Output Variables:

- Height control (Uz): The descent speed (-1 to 1)
- Angle control (Utheta): Correction of the yaw angle for alignment.

Membership functions (MF) for Z:

$$\mu_{FAR} = \begin{cases} 1, & z \geq 5 \\ \dfrac{z-2}{3}, & 2 \leq z \leq 5 \\ 0, & z < 2 \end{cases} \tag{3}$$

Similar membership functions are defined for the linguistic variables **LANDED**, **CLOSE**, and **MID**. For the yaw angle, the membership functions are similar to those used for the positioning errors along the X and Y axes.

Table 2. Rule base for height and yaw angle

#	**Input (Z)**	**Input** (Theta)	**Output (Uz)**	**Output (Utheta)**
1	FAR	-Large	Down_High	+Large
2	MID	Zero	Down_Low	+Large
3	CLOSE	+Small	Zero	-Small
20	…	…	…	…

Wind Effect Compensation: The Fuzzy_Wind block, shown in Figure 1, compensates for the wind's effect on the drone during landing, minimizing its impact on the flight path.

In this block, 28 rules were formulated to ensure effective control and decision-making based on fuzzy logic. These rules allow the system to account for various conditions and adapt to changes in the surrounding environment.

Input Variables:

- Wind Force (WindForce) directly affects the adjustment of yaw angles and control signals [7]. Value range: [0, 10] m/s.
- Wind Direction (WindDir) this parameter describes the wind's direction relative to the drone's movement. It is measured in degrees (0° - 360°). The system accounts for eight primary directions.

Output Variables:

- (Ux, Uy, Uz) remain consistent with the functions from groups 1 and 2.

The membership function equation is defined similarly to equations (1) and (2).

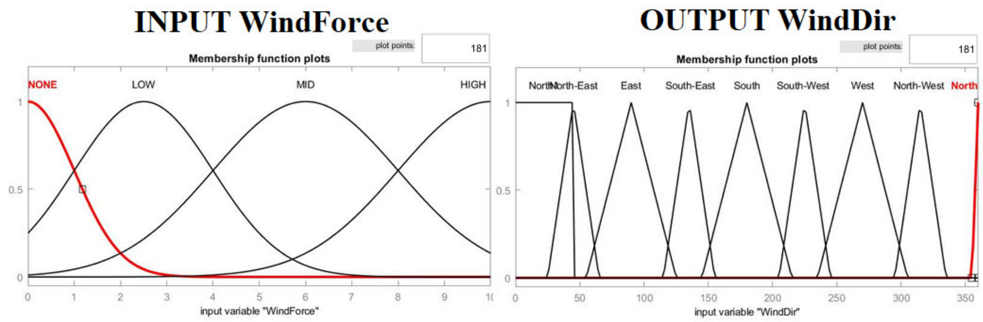

Fig. 4. - Membership Functions of WindForce and WindDir

Fig 6. Surface plot showing the relationship between WindForce, WindDir, and control signals. For example, with a strong wind (around 10 m/s) and a direction of 90 degrees, the control signal can reach a value of -0.5, indicating the need for a leftward adjustment. The graph demonstrates the operation of the fuzzy controller, considering wind strength and direction to generate control signals.

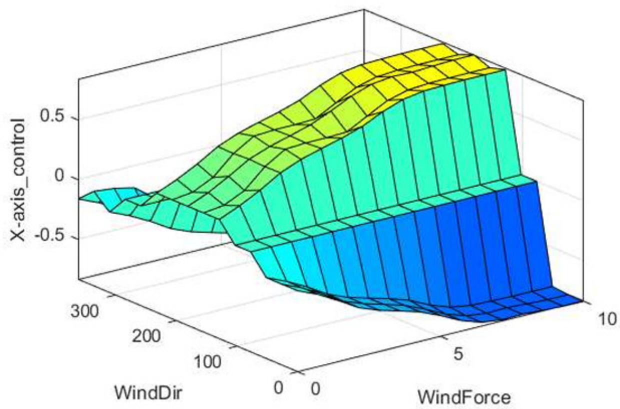

Fig. 5. - Surface plot of wind effect (WindForce and WindDir)

1) ***MATLAB/SIMULINK:*** As noted in [8], FLC is a complex system requiring significant computational resources due to the processing of multiple input parameters and logical rules. The dynamics of the drone's movement are described by mathematical models, including differential equations presented below:

2)

$$\frac{dX}{dt} = \frac{F_x}{M}, \frac{dY}{dt} = \frac{F_y}{M}, \frac{dZ}{dt} = \frac{F_z}{M} \quad (4)$$

The control signals calculated by fuzzy controllers update the drone's coordinates; however, wind creates additional forces that require compensation. To address this, a separate controller adjusts the trajectory by calculating the correction signals U_{x_cor}, U_{z_cor}, U_{z_cor}based on wind speed and direction, improving landing accuracy. The drone's movement is described by equations that account for both primary and correction signals, for example, for the X-axis:

$$X(i) = X(i-1) + \frac{U_x + U_{x\_cor}}{Mass} * dt \quad (5)$$

where $X(i)$ is the current coordinate, U_x is the control signal, U_{x_cor} is the correction signal, $Mass$ is the drone's mass, and dt is the time step. Similar equations are applied for the Y, Z axes, and the yaw angle.

Based on these equations and initial conditions, a model was built, as shown in Fig. 9.

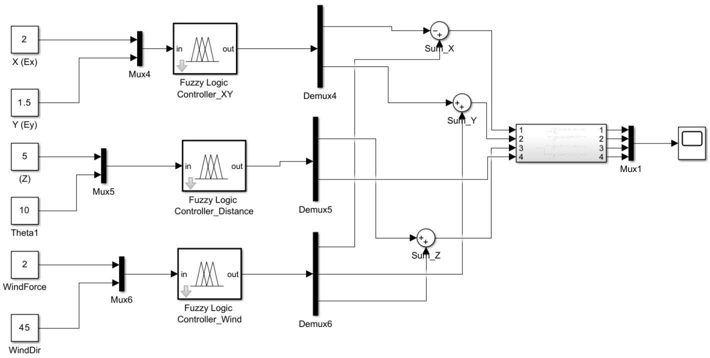

Fig. 6. - Structure of the Fuzzy Controller for Drone Landing

Fig. 6 shows three FLC blocks, each playing a key role in ensuring the drone's stability during landing, effectively compensating for the wind's influence. By considering the wind's strength and direction, as well as adjusting control signals along the X, Y, and Z axes, the drone maintains stability and landing accuracy even in challenging weather conditions. In each block, the rule bases from top to bottom are 25, 20, and 28.

Proportional–Integral–Derivative (PID)

The second stage of the work presents a drone control system model based on a PID controller, ensuring stable and accurate landing. In the presence of wind, the control becomes more complex due to the external forces impacting the trajectory.

The PID controller, widely used in various systems [9], [16], was integrated into the quadcopter system for precise landing. A separate controller is used for each axis (X, Y, Z) and the yaw angle (θ), with carefully selected proportional (Kp), integral (Ki), and derivative (Kd) coefficients, ensuring stability and accuracy of control.

Table 3. PID Controller Coefficients for Drone

Axis	Proportional Coefficient (Kp)	Integral Coefficient (Ki)	Derivative Coefficient (Kd)
X-Axis	1.8	0.01	0.1
Y-Axis	1.8	0.01	0.1
Z-Axis	0.9	0.145	0.1
Angle θ	1.0	0.01	0.1

The PID controller coefficients are selected to minimize the error between the current drone position and the target point, ensuring stability and accuracy. Kp quickly responds to deviations, Ki compensates for errors, and Kd reduces oscillations, improving the smoothness of control. This approach is described in [10], [9].

The drone's motion equations for the X, Y, and Z axes and changes in yaw angle, considering inertia, are similar to equation (5). A larger moment of inertia complicates the change in angular velocity.

The control input is calculated using the following equation:

$$U_x = K_{p\_X} * ex + K_{i\_X} * \int eXdt + K_{d\_X} * \frac{deX}{dt} \quad (6)$$

where $K_{p\_X} * ex$ is the proportional component that responds to the current error, $K_{i\_X} * \int eXdt$ is the integral component that accounts for the accumulated error over time, and $K_{d\_X} * deX/dt$ is the derivative component that responds to the rate of change of the error.

The control inputs are applied similarly for the Y, Z axes, and the yaw angle. Then, taking into account the initial conditions and coefficients, a PID controller is created in SIMULINK. The PID controller model for the X-axis is shown in Fig. 7. The same process is applied for the other axes and the yaw angle.

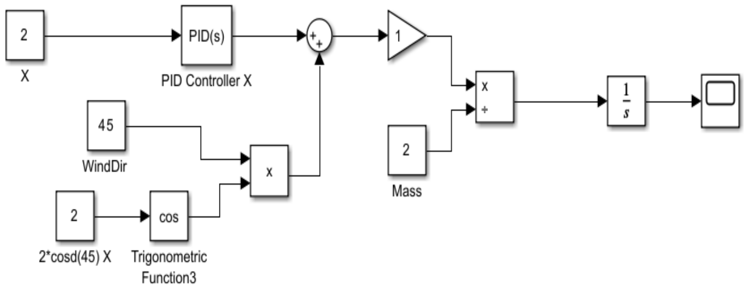

Fig. 7. - PID model in SIMULINK for the X-axis

creates an additional effect on the drone's movement, which is decomposed into components along the X, Y, and Z axes, as well as the yaw angle, depending on the direction.

$$F_{wind\_X} = F_{wind} * \cos(\theta_{wind}) \tag{7}$$

where F_{wind} is the total wind force, and $\cos(\theta_{\text{wind}})$ is the wind direction in degrees.

Linear–Quadratic Regulator (LQR)

This section presents the algorithm for precise drone landing on a specified marker under windy conditions using a Linear-Quadratic Regulator (LQR). The mathematical foundations of the method are discussed, including the system's dynamic equations, the solution to the Riccati equation, and the consideration of external disturbances.

The advantages of using LQR for drones lie in its ability to provide optimal control by minimizing deviations from the desired trajectory and accelerating system stabilization. This allows for high accuracy and efficiency in controlling the drone's movement, particularly in dynamic and changing conditions [4], [14].

The mathematical model equations are essential for analyzing the optimal control system, as these equations will later represent the system's performance [11].

The quadcopter model is described by a system of differential equations, with the drone's movement along the X, Y, and Z axes, as well as the yaw angle, defined by the state equations [11], [12]:

$$x^{\cdot} = A\_x + B\_u \tag{8}$$

where *x* is the state vector of the system, *A* is the system matrix describing the dynamics of the drone, and *B* is the control matrix.

The matrix *A* describes the change in the system's state (position and velocity) without considering control inputs. The motion along each axis is given as follows [13]:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \tag{9}$$

The first row of matrix *A* shows that velocity *v* is the derivative of position *x*. The second row describes acceleration *a* as the derivative of velocity *v*, which is zero in the absence of external forces.

Matrix *B* defines the influence of the control input *u* (applied force) on the state changes of the drone:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u \tag{10}$$

The control input *u* affects only the acceleration, as per Newton's second law: *F=ma,* from which *a = F/m*.

LQR minimizes the quadratic cost function:

$$J = \int_0^\infty (x^T Q_x + u^T R_u) dt \tag{11}$$

where Q is the state weight matrix, determining the priorities for control accuracy, and R is the control effort penalty matrix ($R=1$).

$$Q = \begin{bmatrix} 800 & 0 \\ 0 & 800 \end{bmatrix} \tag{12}$$

In this case, the state weight matrix Q minimizes state deviations (e.g., position and velocity), while R penalizes excessive control effort. A higher value in Q prioritizes minimizing state error, and a value of 1 in R indicates the importance of control effort in the optimization.

The solution to the Riccati equation [13]:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{13}$$

The matrix P obtained from the solution is used to compute the feedback gain matrix:

$$K = R^{-1}B^T P \tag{14}$$

This approach optimizes control by minimizing the cost function J, as confirmed by MATLAB/Simulink simulations. Fig. 8 shows the X-axis model, with similar models for the other axes and yaw angle. Input data includes initial conditions, parameters, and wind effects per equation (6).

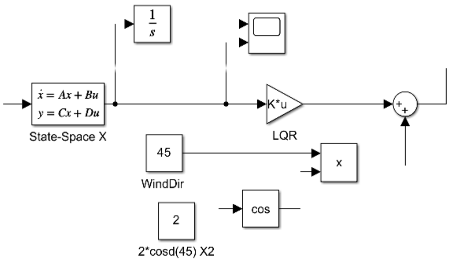

Fig. 8. - LQR Model in SIMULINK for the X-axis

2. Results and discussion

This section of the study presents the results of a drone landing simulation on a target point with coordinates (X=0, Y=0, Z=0) using three different types of controllers: Fuzzy, LQR (Linear-Quadratic Regulator), and PID (Proportional-Integral-Derivative). The results, shown in Fig. 9, demonstrate that all controllers effectively compensate for external factors such as wind, ensuring stable drone control.

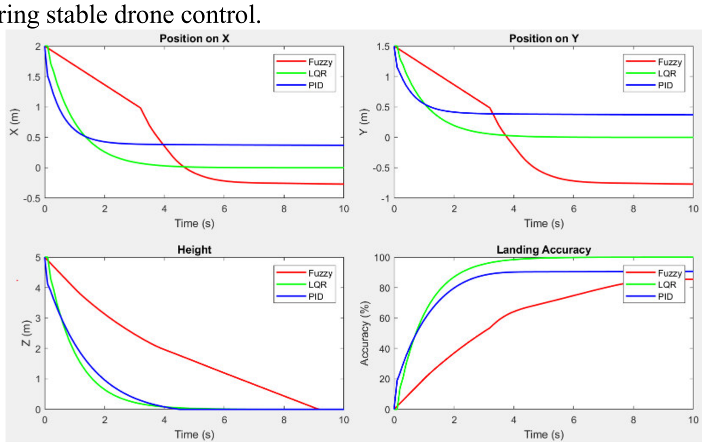

Fig. 9. - Simulation Results of FLC, LQR, and PID Controllers

Table 4. Accuracy and Landing Time Results

#	Accuracy %	Landing *t*
FLC	85.41%	9.20 seconds
LQR	98.86	4.50 seconds
PID	90.37%	4.50 seconds

$$Accuracy(i) = \left(1 - \frac{current\_er}{initial\_er}\right) * 100\% \tag{15}$$

The landing time is defined as the moment when the drone's altitude reaches the target altitude:

$$Landing_{Time} = t \text{ when } Z(t) \leq Z_{target} \tag{16}$$

The FLC controller ensures a smooth position decrease with minor oscillations. The landing accuracy increases to 85.41%, while the landing time of 9.20 seconds is the longest among the three controllers, due to its adaptive nature adjusting control signals based on conditions.

The LQR controller offers the best performance in accuracy and landing time. With a landing accuracy of 98.86% and a landing time of 4.50 seconds, it ensures a fast, precise descent, making it the most efficient for this task.
The PID controller ensures a faster, more stable descent, with landing accuracy reaching 90.37%. The landing time is 4.50 seconds, significantly less than the Fuzzy controllers.

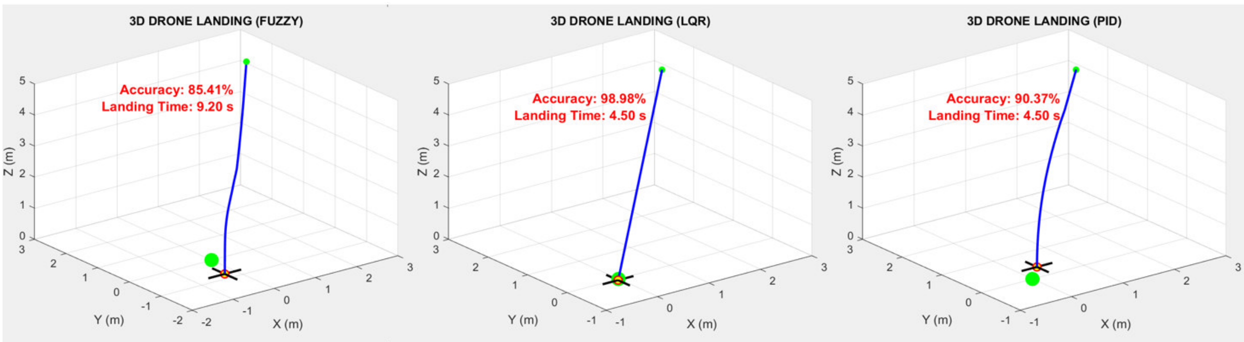

Fig.10. - 3D Animation of the Drone Landing Process Using FLC, LQR, and PID Controllers

Conclusion

The study presents a comparative analysis of three controllers - Fuzzy Logic Controller (FLC), Proportional–Integral–Derivative (PID), and Linear–Quadratic Regulator (LQR) - for precise drone landing in the presence of external disturbances, such as wind. Simulation in MATLAB/SIMULINK with 3D visualization allowed for the evaluation of landing accuracy and time under identical initial conditions.

The FLC demonstrated high adaptability to uncertainties and external disturbances, such as wind, and performs well with nonlinear systems and imprecise data, making it suitable for complex conditions. However, it lags behind other controllers in terms of accuracy and landing time, and its computational complexity may limit its use in real-world applications.

The PID controller provided balanced performance, improving landing time and accuracy compared to FLC. Its simplicity and efficiency make it a reliable choice for tasks where computational efficiency and ease of implementation are important. However, in conditions of high dynamics or uncertainty, the PID may struggle due to its fixed tuning parameters.

The LQR outperformed both FLC and PID in landing accuracy, achieving nearly perfect precision. Its optimal control strategy, with minimal oscillations and fast convergence to the target point, makes it the most effective for tasks requiring high landing accuracy. However, its implementation requires an accurate system model and may be more challenging to tune.

Directions for Future Research:

- The research into combining FLC, PID, and LQR. could lead to the development of more robust and accurate controllers;
- Testing controllers in real-world conditions on physical drones;
- Enhancing disturbance rejection to improve controller performance in the presence of complex external factors.

In conclusion, the choice of controller for drone landing depends on the specific task requirements, including the desired level of accuracy, the complexity of the environment, and available computational resources. The results of this study provide a solid foundation for selecting and optimizing controllers for various drone applications, ensuring safe and accurate landings even in challenging conditions.

References

[1] V. Indrawati, A. Prayitno, and T. A. Kusuma, "Waypoint navigation of AR.Drone quadrotor using fuzzy logic controller," Telkomnika, vol. 13, no. 3, pp. 381–38x, Sep. 2015. DOI: 10.12928/TELKOMNIKA.v13i3.xxxx.

[2] V. S. Fnu and K. Cohen, "Intelligent fuzzy flight control of an autonomous quadrotor UAV," in Proc. 52nd Aerospace Sciences Meeting, National Harbor, MD, USA, Jan. 2014, pp. 1–12. DOI: 10.2514/6.2014-0992.

[3] O. Rodríguez-Abreo, J. Rodríguez-Reséndiz, A. García-Cerezo, and J. R. García-Martínez, "Fuzzy logic controller for UAV with gains optimized via genetic algorithm," Heliyon, vol. 10, p. e26363, Feb. 2024. DOI: 10.1016/j.heliyon.2024.e26363.

[4] D. A. Beloglazov, I. S. Kobersi, E. Y. Kosenko, V. V. Solovyev, and V. V. Shadrina, "Analysis of practical aspects of using controllers in automatic control systems of quadcopters," Engineering Bulletin of the Don, no. 3, 2015. [Online]. Available: http://ivdon.ru/ru/magazine/archive/n3y2015/3078.

[5] V. Indrawati, A. Prayitno, and G. Utomo, "Comparison of two fuzzy logic controller schemes for position control of AR.Drone," in Proc. 2015 7th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE), Chiang Mai, Thailand, Oct. 29–30, 2015. DOI: 10.1109/ICITEED.2015.7408972.
[6] P. Vadakkepat, T. C. Chong, W. A. Arokiasami, and X. Weinan, "Fuzzy logic controllers for navigation and control of AR.Drone using Microsoft Kinect," in Proc. 2016 IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE), Vancouver, BC, Canada, Jul. 24–29, 2016. DOI: 10.1109/FUZZ-IEEE.2016.7737778.
[7] B. Rached, M. Elharoussi, and E. Abdelmounim, "Fuzzy logic control for wind energy conversion system based on DFIG," in Proc. 2019 Int. Conf. Wireless Technol., Embedded Intell. Syst. (WITS), Fez, Morocco, Apr. 3–4, 2019. DOI: 10.1109/WITS.2019.8723722.
[8] M. Rabah, A. Rohan, M. Talha, and others, "Autonomous vision-based target detection and safe landing for UAV," Int. J. Control Autom. Syst., vol. 16, no. 5, pp. 2375–2387, Oct. 2018. DOI: 10.1007/s12555-018-0071-x.
[9] M. Rinaldi, S. Primatesta, and G. Guglieri, "A comparative study for control of quadrotor UAVs," Appl. Sci., vol. 13, no. 6, pp. 3464, Mar. 2023. DOI: 10.3390/app13063464.
[10] O. Polat and A. Sezgin, "Position control of a quadcopter with PID and fuzzy-PID controller," J. Eng. Sci. Design, vol. 12, no. 1, pp. 34–48, 2024. DOI: 10.21923/jesd.1223998.
[11] G. E. Setyawan, W. Kurniawan, and A. C. Lumban Gaol, "Linear quadratic regulator controller (LQR) for AR. Drone's safe landing," in Proc. 2019 Int. Conf. Sustainable Inf. Eng. Technol. (SIET), Lombok, Indonesia, Sep. 2019, pp. 1–6. DOI: 10.1109/SIET48054.2019.8986078.
[12] T. K. Priyambodo, O. A. Dhewa, and T. Susanto, "Model of Linear Quadratic Regulator (LQR) control system in waypoint flight mission of flying wing UAV," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 12, no. 4, pp. 43–49, Oct.-Dec. 2020. [Online]. Available: https://jtec.utem.edu.my/jtec/article/view/5696.
[13] F. Ahmad, P. Kumar, and P. Patil, "Simulation of the quadcopter dynamics with LQR based control," Materials Today Proceedings, vol. 25, pp. 2066–2071, Jan. 2020, doi: 10.1016/j.matpr.2020.04.282.
[14] M. Rinaldi, S. Primatesta, and G. Guglieri, "A comparative study for control of quadrotor UAVs," Appl. Sci., vol. 13, no. 6, pp. 3464, Mar. 2023. DOI: 10.3390/app13063464.
[15] T.-L. Le, N. V. Quynh, N. K. Long, and S. K. Hong, "Multilayer interval type-2 fuzzy controller design for quadcopter unmanned aerial vehicles using Jaya algorithm," IEEE Access, vol. 8, pp. 163539-163548, Oct. 2020, doi: 10.1109/ACCESS.2020.3028617.
[16] A. Eltayeb, M. F. Rahmat, M. A. M. Eltoum, M. H. S. Ibrahim, and M. A. M. Basri, "Trajectory tracking for the quadcopter UAV utilizing fuzzy PID control approach," 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, Feb. 26 - Mar. 1, 2021, doi: 10.1109/ICCCEEE49695.2021.9429636.

Information of the authors

Adilet Ormanov, PhD student, L.N. Gumilyov Eurasian National University, Lecturer, Astana IT University
e-mail: ormanov16@bk.ru

Kanagat Dyusekeev, PhD, L.N. Gumilyov Eurasian National University, Head of the Department of "Computer and Software Engineering"
e-mail: dyussekeyev_ka@enu.kz

Stanislav Shidlovskiy, Doctor of Technical Sciences, Dean of the Faculty of Innovative Technologies
National Research Tomsk State University
e-mail: shidlovskiysv@mail.ru