

# Linear and nonlinear control design for a quadrotor

Samira Hadid<sup>1</sup>, Razika Boushaki Zamoum<sup>2</sup>, Youcef Refis<sup>3</sup>

<sup>1</sup>Laboratoire Ingénierie des Systèmes et Télécommunications, Faculté de Technologie, Université M'hamed Bougara de Boumerdes, Boumerdes, Algeria

<sup>2</sup>Laboratoire d'Automatique Appliquée, Institute of Electrical and Electronic Engineering, Université M'hamed Bougara de Boumerdes, Boumerdes, Algeria

<sup>3</sup>Institute of Electrical and Electronic Engineering, Université M'hamed Bougara de Boumerdes, Boumerdes, Algeria

## Article Info

### Article history:

Received Jan 25, 2024

Revised Oct 17, 2024

Accepted Nov 19, 2024

### Keywords:

Dynamic model

Linear control

Nonlinear control

Quadrotor

Sliding mode

## ABSTRACT

In the current study, the quadrotor's nonlinear dynamic model is developed using the Newton-Euler approach. Following that, several nonlinear and linear control strategies for tracking the quadrotor's trajectory are applied. First, by employing distinct controllers for each output variable, direct application of the linear proportional integral derivative (PID) controller to the nonlinear system is realized. This system may also be linearized about an operational point to generate linear controllers, according to the linear quadratic regulator (LQR) demonstration. Nevertheless, in practice, the system dynamics may not always be accurately reflected by this linear approximation and may even be relatively wasteful. Nonlinear regulators, including the feedback linearization (FBL) controller, sliding mode controller (SMC), and modified sliding mode controller (MSMC), perform better in such situations. The trajectory tracking capabilities, dynamic performance, and potential disruption impact of both methods are evaluated and compared. The FBL with LQR was the best controller among them all. The SMC and the MSMC were also very good in tracking the trajectory.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Samira Hadid

Laboratoire Ingénierie des Systèmes et Télécommunications, Faculté de Technologie

University M'hamed Bougara of Boumerdes

35000 Boumerdes, Algeria

Email: sa.hadid@univ-boumerdes.dz

## 1. INTRODUCTION

During the previous several decades, quadrotors have been widely employed in numerous uses, and a variety of modeling and control strategies have been presented for them. Being a tightly coupled nonlinear multivariable system is one of the difficulties in designing controllers for a quadrotor. Because it employs four actuators to manage six degrees of freedom, it is commonly referred to as an under-actuated system. Considerable progress has been achieved by researchers in the area of quadrotor control by proposing various control techniques [1]–[6]. These techniques aim to improve the stability, maneuverability, and performance of quadrotor systems [7]–[10]. Several linear, nonlinear, and robust controllers have been examined for implementation and simulation [11]–[14].

Proportional integral derivative (PID) and linear quadratic regulator (LQR) controls are used for roll, pitch, and yaw angles as well as height, x, and y locations [13]–[17]. The feedback linearization (FBL) control approach was the focus of several studies and used for a variety of quadrotor control applications. Jiang *et al.* [18] proposed a new controller based on FBL technique. This approach was also employed to design controllers for quadrotors in [19], [20]. Additionally, Reinoso *et al.* [21] and Labbadi *et al.* [22] robust sliding mode control (SMC) was used for tracking trajectories.

In the paper, we explore and compare various linear and nonlinear control techniques for the quadrotor's trajectory tracking. The objective is to analyze the performance and effectiveness of these control techniques in accurately tracking desired trajectories. First, in order to stabilize the system, the controller that we defined was the commonly used linear PID controller. Next, various tests were carried in a closed loop simulation for PID controller starting by step input test then by defining a nonlinear trajectory that the quadcopter was able to track. The LQR was then invented; however, because it only uses an accurate representation of the nonlinear model at the hover point, it may not perform as well or be as robust at other operating points. So, we have to introduce another technique which is an input-output FBL of the system so we get a fully linearized model of the system and we applied on its linear methods like the LQR and pole placement. After that, a simulation was carried out by inserting a step input to the system then performing a trajectory tracking. Finally, the SMC and the modified sliding mode controller (MSMC) are introduced, defined, and explained; they demonstrated a good response to both step input and trajectory tracking. However, the first method had a chattering effect that was rectified by the second one. The proposed method is applied to control a quadrotor to evaluate its performance in terms of trajectory tracking, response time, robustness, and robust stability. Utilizing MATLAB/Simulink software, the simulation is carried out.

The remaining portions of the paper are structured in this way. We present the quadrotor's mathematical model in section 2. In section 3, control techniques are developed and verified by simulations. In section 4, we provide a comparison of the designed controllers. Finally, a brief wrap-up and conclusion are provided.

**2. STATE SPACE MODEL**

**2.1. Mathematical model**

To make the control problem simpler, based on the acquired mathematical models, a state-space model will be developed [20]–[27]. The equations for the dynamic model are as (1):

$$\dot{X} = f(X, U) \tag{1}$$

Where  $U$  is the input vector  $U = [U_1 \ U_2 \ U_3 \ U_4]^T$ ,  $\dot{X}$  is the state vector and  $X$

$$X = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}]^T$$

Following is another way to write (1):

$$\dot{X} = f(X) + \sum_{i=1}^4 g_i(X) u_i \tag{2}$$

With:

$$f(X) = [x_2 \ a_1 x_4 x_6 \ x_4 \ a_2 x_2 x_6 \ x_6 \ a_3 x_2 x_4 \ x_8 \ -g \ x_{10} \ 0 \ x_{12} \ 0]^T \tag{3}$$

$$\text{and } a_1 = \frac{I_y - I_z}{I_x}, a_2 = \frac{I_z - I_x}{I_y}, a_3 = \frac{I_x - I_y}{I_z}, \begin{matrix} g_1 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ G_1 \ 0 \ G_2 \ 0 \ G_3]^T \in \mathbb{R}^{12} \\ g_2 = [0 \ b_1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \in \mathbb{R}^{12} \\ g_3 = [0 \ 0 \ 0 \ b_2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \in \mathbb{R}^{12} \\ g_4 = [0 \ 0 \ 0 \ 0 \ 0 \ b_3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \in \mathbb{R}^{12} \end{matrix}$$

with:

$$\begin{matrix} G_1 = \frac{1}{m} (\cos x_1 \cos x_3) \\ G_2 = \frac{1}{m} (\sin x_1 \sin x_5 + \cos x_1 \cos x_5 \sin x_3) \\ G_3 = \frac{1}{m} (\cos x_1 \sin x_5 \sin x_3 - \cos x_5 \sin x_1) \end{matrix}$$

**2.2. Linearized model**

The linear model is available in the following format:

$$\dot{X} = AX + BU \tag{4}$$

Such that:

$$A = \left. \frac{\partial f(x,U)}{\partial x} \right|_{x=\bar{x}}^{U=\bar{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

$$B = \left. \frac{\partial f(x,U)}{\partial U} \right|_{\substack{x=x_e \\ U=U_e}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_3 \\ 0 & 0 & 0 & 0 \\ \frac{-1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

For (A,B) to be controlled, the following requirements must be met in full [20], [27]:

$$rank(w_c) = n \tag{7}$$

In this case,  $w_c$  also known as Kalman's controllability matrix of dimension  $n \times nm$  equals:

$$w_c = [B \ AB \ A^2B \ A^3B \ A^4B \ A^5B \ A^6B \ A^7B \ A^8B \ A^9B \ A^{10}B \ A^{11}B] \in \mathbb{R}^{12 \times 48}$$

MATLAB was employed to verify the rank of the matrix  $w_c$  and it was determined to have full rank. Thus, (4) describes a controlled system.

### 3. CONTROL STRATEGIES

#### 3.1. Proportional integral derivative controller

The most common controller for quadrotors is the PID, because they are simple to develop and install, and due to how it is possible to optimize the gain parameters to get the required performance [16], [28]–[30].

##### 3.1.1. Altitude control

To control the quadrotor's altitude a PID controller is developed. It provides the control input  $U_1$ , that regulates the quadrotor's altitude applying the equation. The following is the derived control law:

$$U_1 = k_p(z_d - z) + k_d(\dot{z}_d - \dot{z}) + k_i \int (z_d - z)dt \tag{8}$$

The proportional gain, derivative gain, and integral gain are denoted by the terms  $k_p$ ,  $k_d$ , and  $k_i$  respectively. The desired altitude and the desired altitude rate of change are denoted by  $z_d$  and  $\dot{z}_d$  respectively.

##### 3.1.2. Attitude and heading control

– Roll controller

To regulate the roll angle  $\phi$  of the quadrotor, another PID controller is developed. The derived control law generates the input  $U_2$  that controls the roll angle as (9):

$$U_2 = k_p(\phi_d - \phi) + k_d(\dot{\phi}_d - \dot{\phi}) + k_i \int (\phi_d - \phi) dt \quad (9)$$

The proportional, derivative and integral gains are denoted respectively by  $k_p$ ,  $k_d$ , and  $k_i$ .  $\phi_d$  and  $\dot{\phi}_d$  are respectively the desired roll angle and the desired roll angle rate of change.

– Pitch controller

A PID controller is designed for controlling the quadrotor's pitch angle  $\theta$ . The resulting control law generates the input  $U_3$  as (10):

$$U_3 = k_p(\theta_d - \theta) + k_d(\dot{\theta}_d - \dot{\theta}) + k_i \int (\theta_d - \theta) dt \quad (10)$$

The proportional, derivative, and integral gains are denoted respectively by  $k_p$ ,  $k_d$ , and  $k_i$ .  $\theta_d$  and  $\dot{\theta}_d$  are respectively the desired pitch angle and the desired pitch angle rate of change.

– Yaw controller

Similar to the two previous controllers, a yaw controller is designed to generate the control input  $U_4$  based on the following control law:

$$U_4 = k_p(\psi_d - \psi) + k_d(\dot{\psi}_d - \dot{\psi}) + k_i \int (\psi_d - \psi) dt \quad (11)$$

$k_p$ ,  $k_d$ , and  $k_i$  are respectively the proportional gain, the derivative gain and the integral gain.  $\psi_d$  and  $\dot{\psi}_d$  are the desired yaw angle and the desired yaw angle rate of change respectively.

– Position controller

A position controller is developed after achieving reliable controllers for the quadrotor's altitude and attitude. The desired accelerations  $\ddot{x}_d$  and  $\ddot{y}_d$  are then calculated using PID controllers.

$$\begin{aligned} \ddot{x}_d &= k_p(x_d - x) + k_d(\dot{x}_d - \dot{x}) + k_i \int (x_d - x) dt \\ \ddot{y}_d &= k_p(y_d - y) + k_d(\dot{y}_d - \dot{y}) + k_i \int (y_d - y) dt \end{aligned} \quad (12)$$

$k_p$ ,  $k_d$ , and  $k_i$  denote respectively the proportional, derivative and integral gains.  $x_d$ ,  $\dot{x}_d$ ,  $y_d$ , and  $\dot{y}_d$  are respectively the desired x position, the desired x position rate of change, the desired y position and the desired y position rate of change.

### 3.1.3. Proportional integral derivative controller simulation

The following figures display the PID controller simulation results. The PID controller's step input test is shown in Figure 1. Table 1 displays the rising time, the % overshoot, and settling time for positions x, y, and z, and orientations  $\psi(t)$  using the PID controller. These parameters provide crucial information on the PID controller's performance, accuracy, and stability when controlling the system. Figure 2 shows the trajectory tracking for PID control.

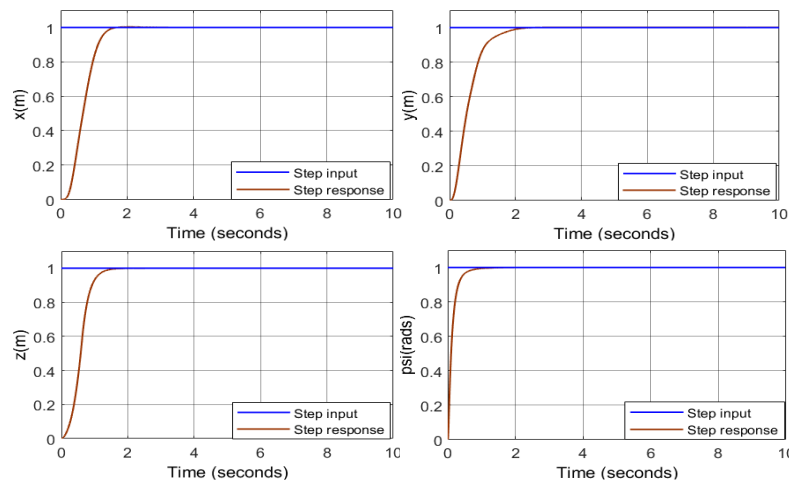


Figure 1. Step input test for PID controller

Table 1. Characteristic performances to a step input when using a PID controller

	x(t)	y(t)	z(t)	$\psi(t)$
Rise time (s)	0.78	0.87	0.64	0.28
Overshoot (%)	0.50	0.50	0.50	0.32
Settling time (s)	1.41	1.8	1.26	0.63

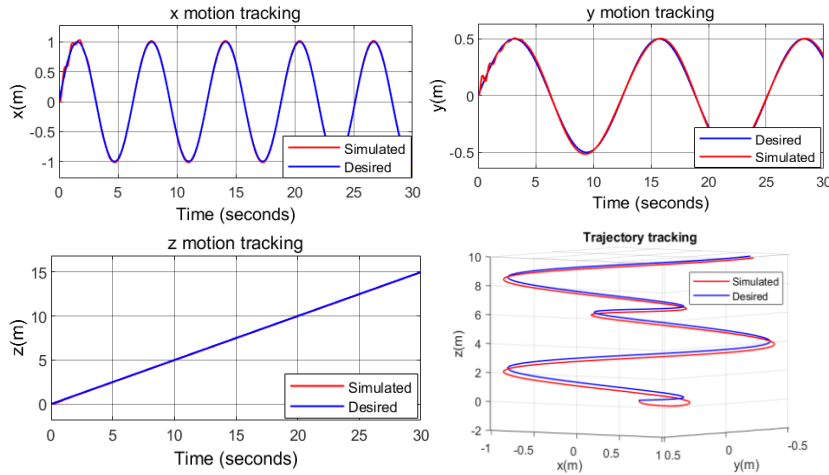


Figure 2. Trajectory tracking for PID control

### 3.2. The linear quadratic regulator

#### 3.2.1. The linear quadratic method

The LQR aims to minimize certain costs while guiding the state  $X$  of the system to follow the desired path  $X_d$  [31], [32]. Take into consideration the dynamic system:

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = C \cdot x \end{cases} \tag{13}$$

The cost function for this optimal problem is given by:

$$J = \int_{t_0}^{\infty} [U(t)^T \cdot R \cdot U(t) + (X(t) - X_d(t))^T \cdot Q \cdot (X(t) - X_d(t))] dt \tag{14}$$

where  $Q$  represents the state's cost and  $R$  represents the actuator cost; the two are positive definite [12]. The control input  $U(t)$  that minimizes the cost function generated as (15):

$$U(t) = -K \cdot [X(t) - X_d(t)] \tag{15}$$

where the process of computing the optimal gains is carried out by:

$$K = R^{-1} \cdot B^T \cdot P \tag{16}$$

The algebraic equation of Riccati can be resolved by the  $P$  matrix:

$$P \cdot A + A^T \cdot P - P \cdot B R^{-1} \cdot B^T \cdot P + C \cdot Q \cdot C = 0 \tag{17}$$

Considering the linear system (3). We select the matrices  $Q$  and  $R$  taking into account  $A$  (4) and  $B$  (5). Using the LQR function from MATLAB/Simulink, we apply the LQR control.

#### 3.2.2. Linear quadratic regulator controller simulation

Results of the LQRs controller simulation are shown in Figures 3 and 4. The LQR position, altitude, and heading response to a step input is displayed in Figure 3. The Table 2 presents the step input's characteristic performances using the LQR controller. Figure 4 displays Trajectory tracking for LQR control.

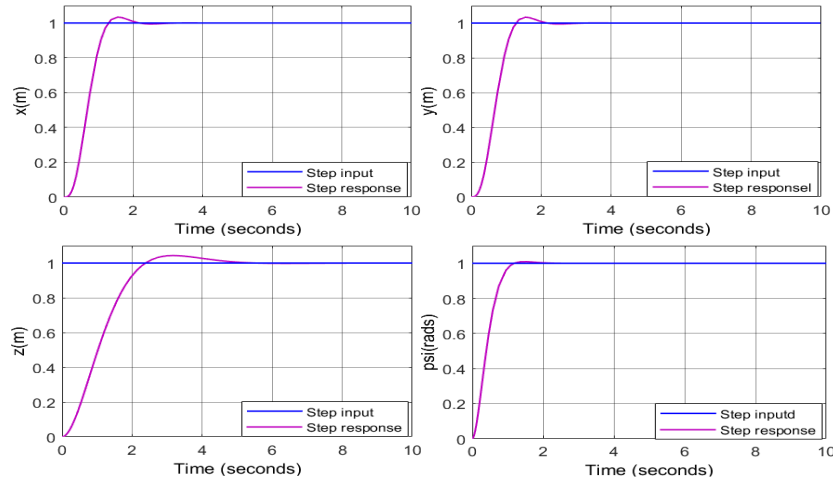


Figure 3. The LQR’s position, altitude and heading response to a step input

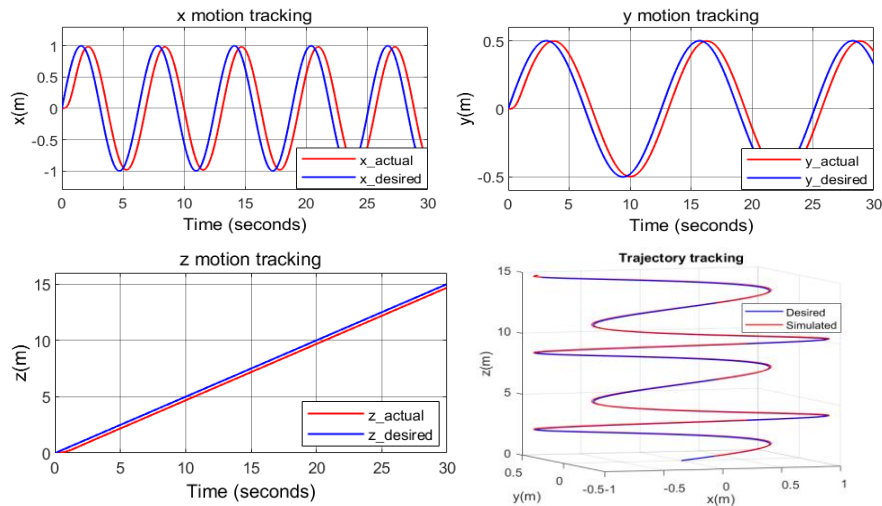


Figure 4. Trajectory tracking for LQR control

Table 2. Performance characteristics of a step input with the LQR controller

	$x(t)$	$y(t)$	$z(t)$	$\psi(t)$
Rise time (s)	0.72	0.72	1.52	0.68
Overshoot (%)	3.64	3.64	4.73	0.50
Settling time (s)	1.22	1.21	2.25	1.04

**3.3. Feedback linearization control**  
**3.3.1. Feedback linearization method**

FBL is a nonlinear control design method whose basic principle is to algebraically transform the dynamics of nonlinear systems into linear ones, allowing linear control design approaches to be employed [33]-[35]. In this study, the input-output linearization strategy was employed to avoid the complexity of the input-state linearization technique [36]-[38]. We choose an output function for the system to determine the control objective. We want to regulate both the quadrotor's absolute position  $[x \ y \ z]^T$  and the yaw angle  $\psi$ . Thus, the selected output function is:

$$y = h(x) = [x \ y \ z \ \psi]^T \tag{18}$$

Assuming that the state  $x$  of the system is completely measurable, we aim to develop a static state feedback control rule of the following form:

$$u = \alpha(x) + \beta(x).v \tag{19}$$

where  $v$  is the external reference input,  $\alpha(x)=[\alpha(x)_1 \alpha(x)_2 \alpha(x)_3 \alpha(x)_4]^T$  and  $\beta(x) \in \mathbb{R}^{4 \times 4}$ . Our system is given as follows in the new coordinates:

$$\begin{cases} \dot{z} = A \cdot z + B \cdot v \\ y = C \cdot z \end{cases} \tag{20}$$

with:

$$z = [z_1 \ z_2 \ z_3 \ z_4 \ z_5 \ z_6 \ z_7 \ z_8 \ z_9 \ z_{10} \ z_{11} \ z_{12} \ z_{13} \ z_{14}]^T \in \mathbb{R}^{14} \tag{21}$$

$$v = [v_1 \ v_2 \ v_3 \ v_4]^T \in \mathbb{R}^4 \tag{22}$$

$$A = \begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 \\ 0 & 0 & A_1 & 0 \\ 0 & 0 & 0 & A_2 \end{bmatrix} \in \mathbb{R}^{14 \times 14}, B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}, C = \begin{bmatrix} c_1^T & 0 & 0 & 0 \\ 0 & c_1^T & 0 & 0 \\ 0 & 0 & c_1^T & 0 \\ 0 & 0 & 0 & c_2^T \end{bmatrix} \in \mathbb{R}^{14 \times 4} \tag{23}$$

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, A_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 2} \tag{24}$$

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \tag{25}$$

$$B_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, B_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 4} \tag{26}$$

$$c_1 = [1 \ 0 \ 0 \ 0]^T \in \mathbb{R}^4, c_2 = [1 \ 0]^T \in \mathbb{R}^2 \tag{27}$$

Since we were able to derive a linear system, it is feasible and simple to use additional feedback control methods, including LQRs and pole placement.

### 3.3.2. Simulation results of feedback linearization with pole placement

Results of the FBL controller simulation are shown on the following figures. Figure 5 displays how the FBL with pole placement responds to a step input in terms of position, altitude, and heading. Using the FBL controller with pole placement, Table 3 displays the rising time, settling time, and overshoot for coordinates  $x$ ,  $y$ , and  $z$ , and orientations  $\psi(t)$ . Figure 6 shows trajectory tracking for FBL control with pole placement.

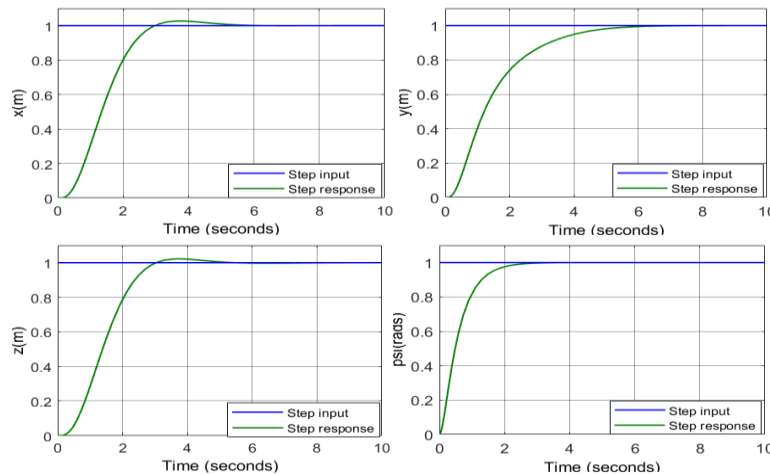


Figure 5. Position, altitude, and heading reaction of the FBL with pole placement to a step input

Table 3. Characteristic performances to a step input when using the FBL controller with pole placement

	$x(t)$	$y(t)$	$z(t)$	$\psi(t)$
Rise time (s)	1.73	2.72	1.73	1.13
Overshoot (%)	2.52	0	2.57	0.50
Settling time (s)	2.78	4.94	2.9	2.1

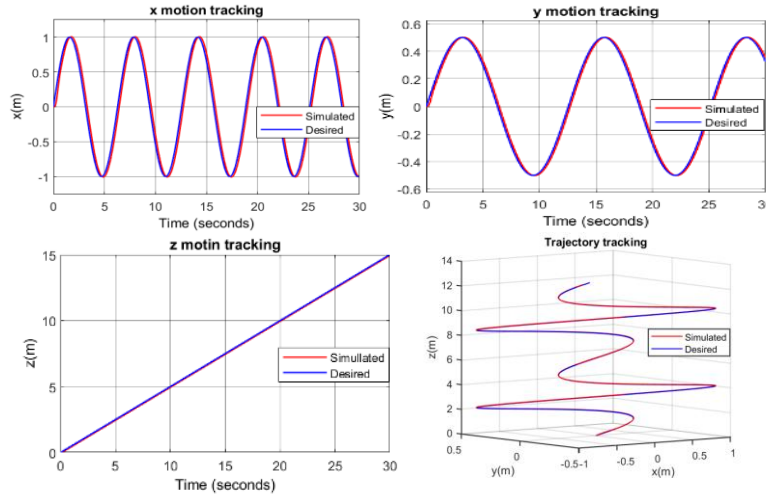


Figure 6. Trajectory tracking for FBL control with pole placement

**3.3.3. Simulation results of feedback linearization with linear quadratic regulator**

Results of the LQR controller simulation are shown on the following figures. Figure 7 illustrates how the FBL using LQR responds to a step input in terms of position, altitude, and heading. A step input's characteristic performances are shown in Table 4 using the FBL controller with LQR. Figure 8 shows trajectory tracking for FBL with LQR control.

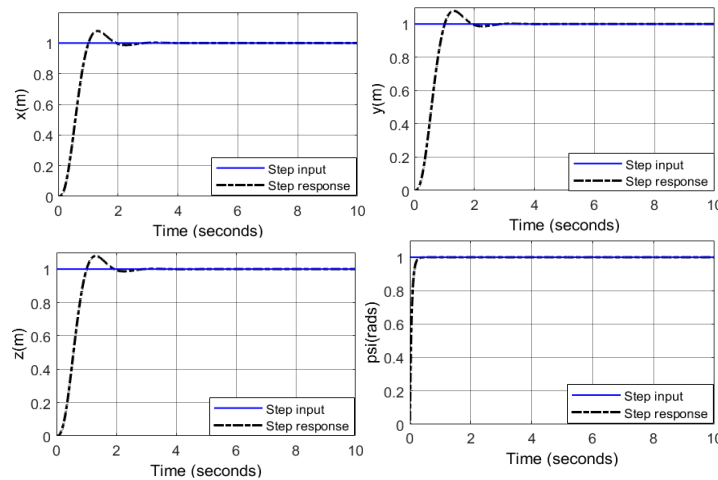


Figure 7. Position, altitude, and heading response to a step input using the FBL controller with LQR

Table 4. Step input's characteristic performances using the FBL with LQR

	$x(t)$	$y(t)$	$z(t)$	$\psi(t)$
Rise time (s)	0.60	0.60	0.60	0.13
Overshoot (%)	8.15	8.15	8.15	0.50
Settling time (s)	1.77	1.77	1.76	0.22



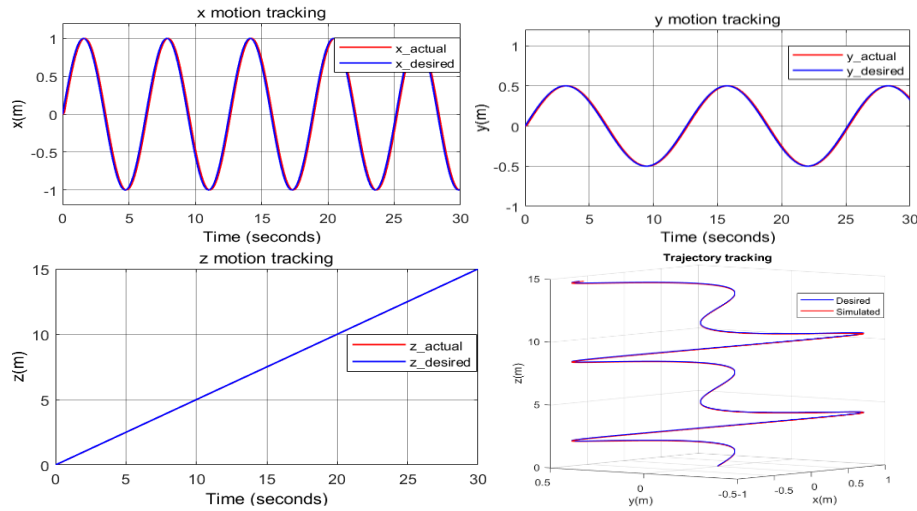


Figure 8. Trajectory tracking for FBL with LQR control

### 3.4. Sliding mode control

#### 3.4.1. Sliding mode control method

A SMC is a robust control method that uses a high-speed switching control rule to retain the state trajectories on a user-defined surface in state space and push them in that trajectory. The core principle of the SMC technique is to guide the system states toward an appropriate sliding surface and then develop a stabilizing control rule to ensure that the system states remain on that surface [39]–[41]. Das *et al.* [8], the general form of (28) was proposed for choosing the shape of the sliding surface:

$$S = \left( \lambda + \frac{d}{dt} \right)^{n-1} e \quad (28)$$

#### – Altitude control

The tracking errors indicate the difference between the state's current and desired values, which are defined as (29):

$$e_z = z - z_d \quad (29)$$

$S_z$  defines the sliding surface:

$$S_z = \lambda_z e_z + \dot{e}_z \quad (30)$$

The following are the exponential reaching laws for altitude sliding surfaces:

$$\dot{S}_z = -\varepsilon_z \operatorname{sgn}(S_z) - k_z S_z \quad (31)$$

Following that, the exponential reaching law is equivalent to the sliding surface's derivative as (32):

$$\dot{S}_z = -\varepsilon_z \operatorname{sgn}(S_z) - k_z S_z = \lambda_z (\dot{z} - \dot{z}_d) + (\ddot{z} - \ddot{z}_d) \quad (32)$$

Control input for altitude  $U_1$  is calculated:

$$U_1 = [-\lambda_z (\dot{z} - \dot{z}_d) + g + \ddot{z}_d - \varepsilon_z \operatorname{sgn}(S_z) - k_z S_z] \frac{m}{\cos \phi \cos \theta} \quad (33)$$

#### – Attitude control

The same procedures used to develop the altitude controller are used to implement an attitude SMC, pitch  $U_2$ , roll  $U_3$ , and yaw  $U_4$  control inputs are calculated as (34):

$$\begin{cases} U_2 = \left[ -\lambda_\phi(\dot{\phi} - \dot{\phi}_d) - \frac{I_r \dot{\theta} \Omega_r}{I_x} - \dot{\theta} \dot{\psi} \left( \frac{I_y - I_z}{I_x} \right) + \ddot{\phi}_d - \varepsilon_\phi \operatorname{sgn}(S_\phi) - k_\phi S_\phi \right] \frac{I_x}{l} \\ U_3 = \left[ -\lambda_\theta(\dot{\theta} - \dot{\theta}_d) + \frac{I_r \dot{\phi} \Omega_r}{I_y} - \dot{\phi} \dot{\psi} \left( \frac{I_z - I_x}{I_y} \right) + \ddot{\theta}_d - \varepsilon_\theta \operatorname{sgn}(S_\theta) - k_\theta S_\theta \right] \frac{I_y}{l} \\ U_4 = \left[ -\lambda_\psi(\dot{\psi} - \dot{\psi}_d) - \dot{\phi} \dot{\theta} \left( \frac{I_x - I_y}{I_z} \right) + \ddot{\psi}_d - \varepsilon_\psi \operatorname{sgn}(S_\psi) - k_\psi S_\psi \right] \frac{I_z}{l} \end{cases} \quad (34)$$

– Position controller

Position tracking of the quadcopter is achieved by calculating the desired rotational angles  $\phi_d$  and  $\theta_d$  around the hover:

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \frac{1}{g} \begin{bmatrix} \sin \psi & -\cos \psi \\ \cos \psi & \sin \psi \end{bmatrix} \begin{bmatrix} -\lambda_x(\dot{x} - \dot{x}_d) + \ddot{x}_d - \varepsilon_x \operatorname{sgn}(S_x) - k_x S_x \\ -\lambda_y(\dot{y} - \dot{y}_d) + \ddot{y}_d - \varepsilon_y \operatorname{sgn}(S_y) - k_y S_y \end{bmatrix} \quad (35)$$

**3.4.2. Sliding mode controller simulation**

Results of the SMC controller simulation are shown on the Figures 9 and 10. Figure 9 shows the SMC's position, altitude, and heading responses. Figure 10 shows trajectory tracking for SMC control. Table 5 presents the characteristic performances to a step input using the SMC.

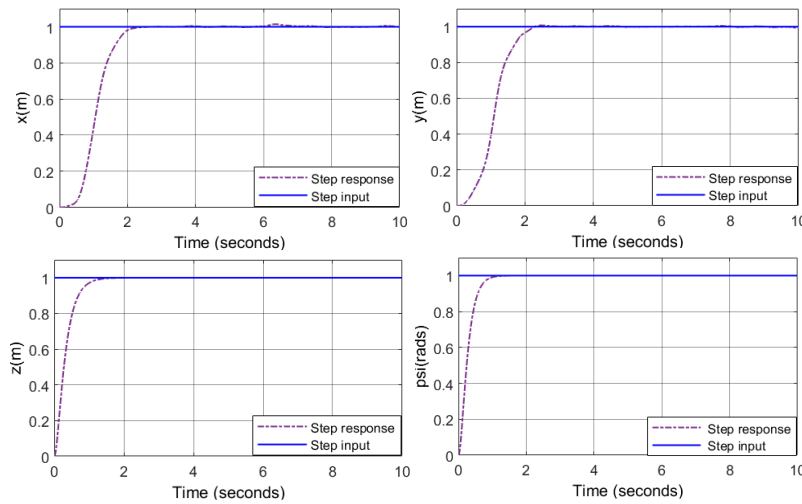


Figure 9. The SMC's position, altitude, and heading responses

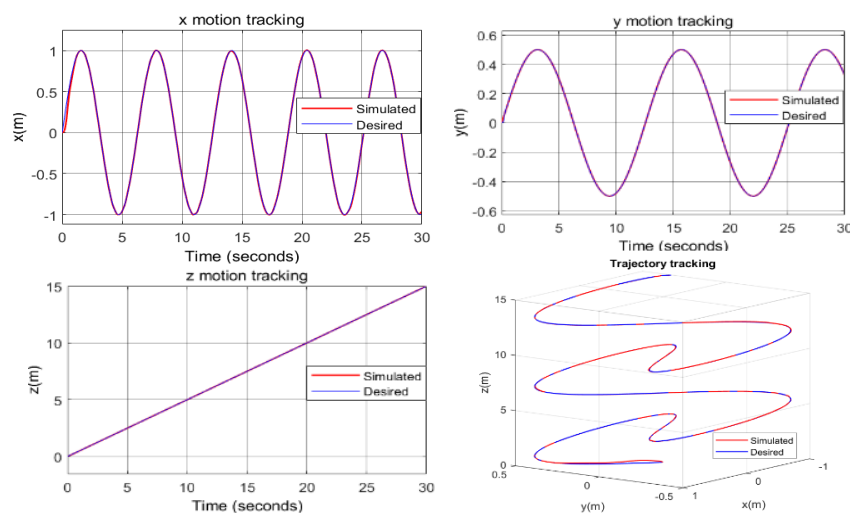


Figure 10. Trajectory tracking for SMC control

Table 5. Step input's characteristic performances using the SMC

	x(t)	y(t)	z(t)	$\psi(t)$
Rise time (s)	0.99	1.15	0.59	0.48
Overshoot (%)	0.03	0.5	0.50	0.50
Settling time (s)	1.99	2.11	1.1	0.85

### 3.5. Modified sliding mode control

#### 3.5.1. Modified sliding mode control method

The chattering phenomenon associated with SMC presents challenges that make it difficult to implement in real-world applications [24], [42]. To mitigate the chattering effect, utilizing a saturation function in place of the sign(s) function. This function is expressed as (36):

$$sat(s) = \begin{cases} s & \text{if } |s| \leq 1 \\ sgn(s) & \text{if } |s| > 1 \end{cases} \tag{36}$$

Therefore, to apply this modification to our system's SMCs, the sign(s) function terms in (33)-(35) should be changed by the sat (s/ε) function. The constant ε represents the line's slope between 1 and -1, this region is the border region or boundary layer.

#### 3.5.2. Modified sliding mode controller simulation

Results of the MSMC controller simulation are shown on the following figures. Figure 11 shows the Modified SMC position, altitude, and heading responses. Table 6 presents the characteristic performances to a step input. Figure 12 shows trajectory tracking for modified SMC control.

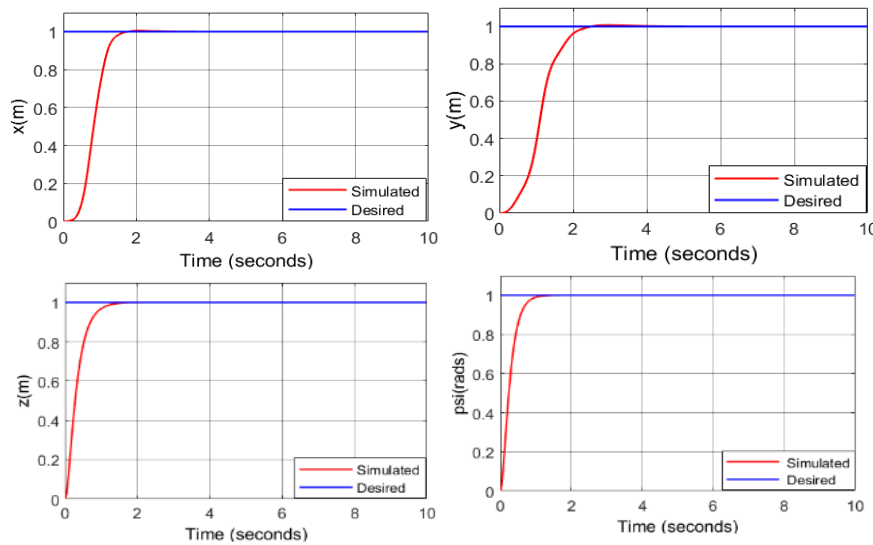


Figure 11. The modified SMC position, altitude, and heading responses

Table 6. Step input's characteristic performances using MSMC

	x(t)	y(t)	z(t)	$\psi(t)$
Rise time (s)	0.70	1.15	0.60	0.48
Overshoot (%)	0.50	0.50	0.50	0.50
Settling time (s)	1.48	2.07	1.11	0.86

The SMC and the MSMC were introduced, defined, and explained; they demonstrated a good response to both step input and trajectory tracking. However, the first method had a chattering effect that was rectified by the second one.

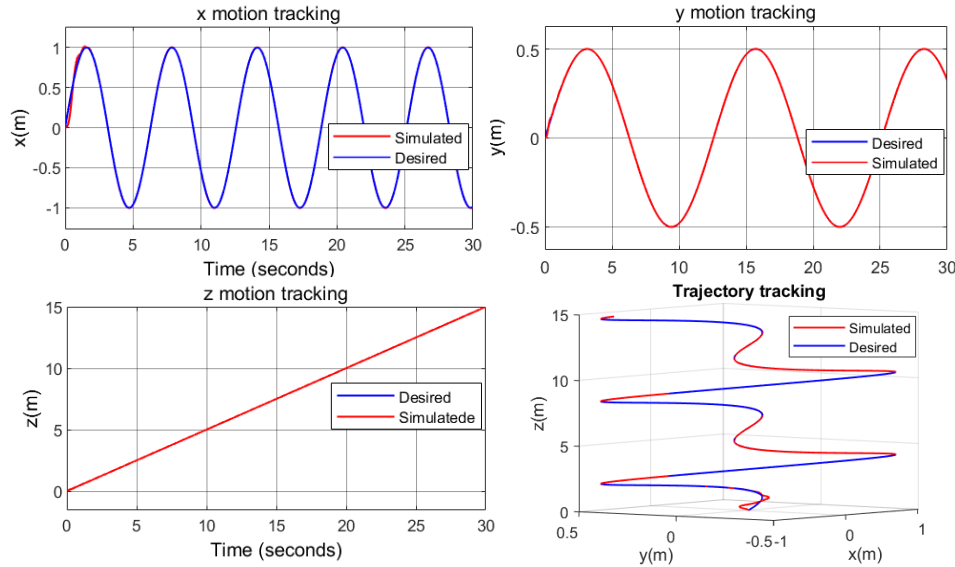


Figure 12. Trajectory tracking for MSMC control

#### 4. DISCUSSION AND RESULTS COMPARISON

##### 4.1. Step response comparison

The response graph for  $(x, y, z, \psi)$  of the system, influenced by each of the six controllers, were plotted on top of one another to facilitate a proper comparison among the six implemented control strategies. Figure 13 displays the output variable  $x$ 's step response using different controllers. Table 7 illustrates step input's characteristic performances for the  $x(t)$  variable using the different controllers. Figure 14 displays the step response of the output variable  $y$  using different controllers. Table 8 illustrates step input's characteristic performances for the  $y(t)$  variable using the different controllers.

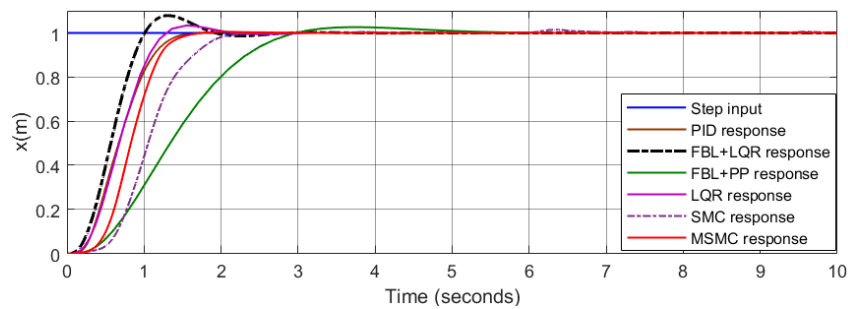


Figure 13.  $x(t)$  response to a step input using the six different controllers

Table 7. Step input's characteristic performances for the  $x(t)$  variable using the different controllers

$x(t)$	PID	LQR	FBL+PP	FBL+LQR	SMC	MSMC
Rise time (s)	0.78	0.72	1.73	0.60	0.99	0.70
Overshoot (%)	0.50	3.64	2.57	8.15	0.03	0.50
Settling time (s)	1.41	1.22	2.78	1.77	1.99	1.48

Figure 15 displays the step response of the output variable  $z$  using different controllers. Table 9 illustrates step input's characteristic performances for the  $z(t)$  variable using the different controllers. Figure 16 shows the step response of the output variable  $\psi$  using the different controllers. Table 10 illustrates step input's characteristic performances for the  $\psi(t)$  variable using different controllers.

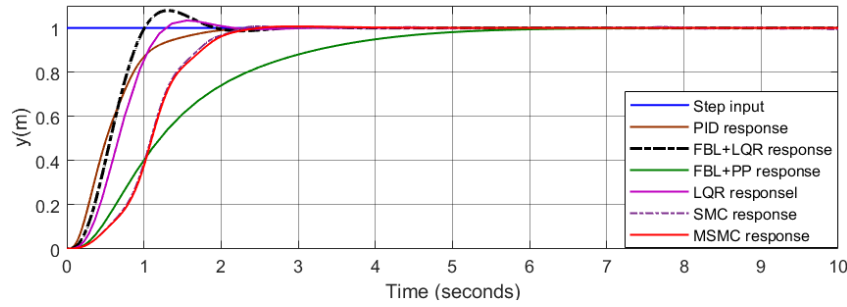


Figure 14.  $y(t)$  response to a step input using the six controllers

Table 8. Step input's characteristic performances for the  $y(t)$  variable using the different controllers

$y(t)$	PID	LQR	FBL+PP	FBL+LQR	SMC	MSMC
Rise time (s)	0.87	0.72	2.72	0.60	1.15	1.15
Overshoot (%)	0.50	3.64	0	8.15	0.5	0.50
Settling time(s)	1.80	1.21	4.94	1.77	2.11	2.07

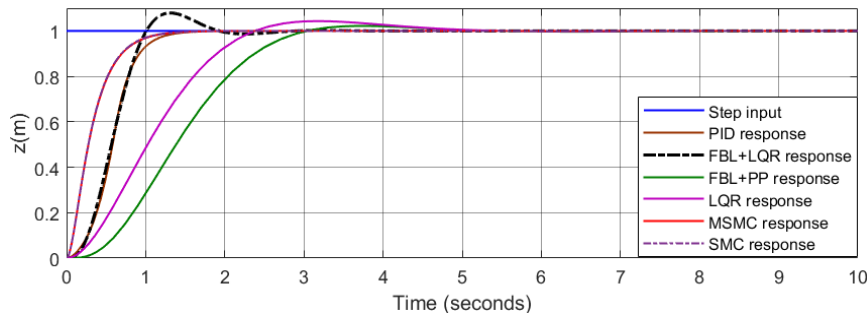


Figure 15.  $z(t)$  response to a step input using the six controllers

Table 9. Step input's characteristic performances for the  $z(t)$  variable with the different controllers

$z(t)$	PID	LQR	FBL+PP	FBL+LQR	SMC	MSMC
Rise time (s)	0.64	1.52	1.73	0.60	0.59	0.60
Overshoot (%)	0.50	4.73	2.57	8.15	0.50	0.50
Settling time(s)	1.26	2.25	2.9	1.76	1.10	1.11

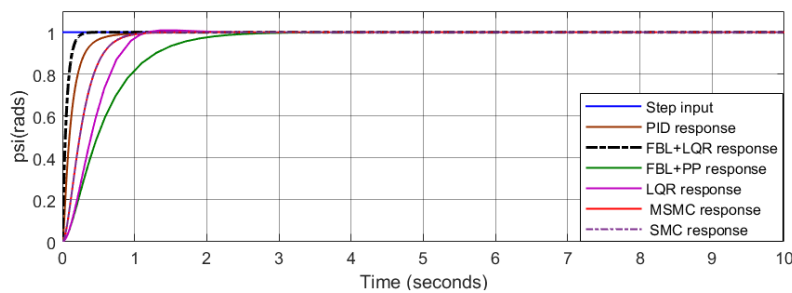


Figure 16.  $\psi(t)$  response to a step input using the six controllers

Table 10. Step input's characteristic performances for the  $\psi(t)$  variable with the different controllers

$\psi(t)$	PID	LQR	FBL+PP	FBL+LQR	SMC	MSMC
Rise time (s)	0.28	0.68	1.13	0.12	0.48	0.48
Overshoot (%)	0.32	0.50	0.50	0.50	0.50	0.50
Settling time(s)	0.63	1.04	2.1	0.22	0.85	0.86

Based on the Tables 7 to 10, the following deductions can be made:

- The combination of FBL and LQR controller has the best overall performance for the four output variables regarding both rise and settling time. However, it has bad overshoot.
- The PID, SMC, and MSMC has an acceptable rise time settling time and almost overshoot is neglected at all in all the cases.
- The LQR controller also showed an acceptable rise and settling time but suffered from a small overshoot.
- The FBL with pole placement had the worst performance concerning both the rise and settling time, yet it had almost no overshoot.

#### 4.2. Trajectory tracking comparison

The PID controller tracks the user-defined trajectory with good performance. However, there is some overshoot in the beginning for about 2 seconds as illustrated in (Figure 2). The performance of the LQR tracking was lower than that of the other controllers. As, we can see that the actual path is slightly above the desired one, and the actual path never reaches the desired one in the zigs and turns as illustrated (Figure 4).

The FBL with pole placement was slightly better than the LQR, unlike the latter which never reached the zigs and turns, The FBL with pole placement reached but passed them as for the path the tracking was good (Figure 6). The FBL with LQR was the best controller among them all, it showed no overshoot at the beginning and the tracking was almost perfect that the desired and the actual trajectory was on top of each other (Figure 8). The SMC and the MSMC were also very good in tracking the trajectory except they both had a small overshoot at the beginning and the SMC suffered from the chattering effect (Figure 10 and 12).

### 5. CONCLUSION

This paper gives a comparison of linear and nonlinear control strategies used to operate quadcopters. To evaluate the performance of the suggested control algorithms, the Newton-Euler technique is used to generate the quadcopter's dynamic model. The step response of all six controllers is examined in terms of rising time, percentage overshoot, and settling time. The best results are obtained when employing the PID, LQR, and FBL control approach using LQR. The SMC achieved an acceptable result, however FBL with pole placement was insufficient. In the trajectory tracking section, when all six controllers are compared to each other, we observed that the best results are obtained when employing the FBL control approach with LQR, and the modified SMC.




Because it is difficult to predict external disturbances such as wind velocities, the mathematical model offered does not take these into account. The controllers should be made robust so that they can effectively deal with external disturbances that were overlooked during modeling. Future research should take the disturbances into account. Another step is the development of a controller capable of dealing with the failure of one or more rotors.

### REFERENCES




- [1] M. A. Basri and A. Noordin, "Optimal backstepping control of quadrotor UAV using gravitational search optimization algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1819–1826, 2020, doi: 10.11591/eei.v9i5.2159.
- [2] R. Benotsmane, A. Reda, and J. Vasarhelyi, "Model Predictive Control for Autonomous Quadrotor Trajectory Tracking," *2022 23rd International Carpathian Control Conference (ICCC)*, IEEE, pp. 215–220, 2022, doi: 10.1109/iccc54292.2022.9805883.
- [3] Y. Bouzid, S. H. Derrouaoui, and M. Guiatni, "PID Gain Scheduling for 3D Trajectory Tracking of a Quadrotor with Rotating and Extendable Arms," *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, IEEE, pp. 1–4, 2021, doi: 10.1109/icrami52622.2021.9585973.
- [4] D. Kucherov, A. Kozub, O. Sushchenko, and R. Skrynkovskyy, "Stabilizing the spatial position of a quadrotor by the backstepping procedure," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, p. 1188, 2021, doi: 10.11591/ijeecs.v23.i2.pp1188-1199.
- [5] H. Samira, B. Razika, B. Kousseila, and H. Gaya, "Gain Scheduling Control for a Quadrotor Based on Proportional Derivative Controller," *2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, IEEE, pp. 84–89, 2023, doi: 10.1109/icvee59738.2023.10348263.
- [6] I. Siti, M. Mjahed, H. Ayad, and A. El Kari, "New Trajectory Tracking Approach for a Quadcopter Using Genetic Algorithm and Reference Model Methods," *Applied Sciences*, vol. 9, no. 9, p. 1780, 2019, doi: 10.3390/app9091780.
- [7] A. Abdulkareem, A. A. Idowu, V. Oguntosin, and O. M. Popoola, "Modeling and Nonlinear Control of a Quadcopter for Stabilization and Trajectory Tracking," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4036208.
- [8] A. Das, K. Subbarao, and F. Lewis, "Dynamic inversion with zero-dynamics stabilisation for quadrotor control," *IET Control Theory & Applications*, vol. 3, no. 3, pp. 303–314, 2009, doi: 10.1049/iet-cta:20080002.
- [9] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012, doi: 10.1002/rob.20414.
- [10] A. Abdalhadi, H. Wahid, and D. H. Burhanuddin, "An optimal proportional integral derivative tuning for a magnetic levitation system using metamodeling approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, p. 1356, 2022, doi: 10.11591/ijeecs.v25.i3.pp1356-1366.
- [11] N. Xuan-Mung and S.-K. Hong, "Improved Altitude Control Algorithm for Quadcopter Unmanned Aerial Vehicles," *Applied Sciences*, vol. 9, no. 10, p. 2122, 2019, doi: 10.3390/app9102122.

- [12] C.-C. Tsui, *Robust Control System Design*. CRC Press, 2022, doi: 10.1201/9781003259572.
- [13] C. T. Chen, *Linear System Theory and Design*. Oxford University Press, Inc, 1984.
- [14] A. Saibi, H. Belaidi, R. Boushaki, R. Z. Eddine, and A. Hafid, "Enhanced backstepping control for disturbances rejection in quadrotors," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 6, pp. 3201–3216, 2022, doi: 10.11591/eei.v11i6.3997.
- [15] A. Noordin, M. A. M. Basri, and Z. Mohamed, "Simulation and experimental study on PID control of a quadrotor MAV with perturbation," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1811–1818, 2020, doi: 10.11591/eei.v9i5.2158.
- [16] T. Bodrumlu, M. T. Soylemez, and I. Mutlu, "Modelling and Control of the Qball X4 Quadrotor System based on Pid and Fuzzy Logic Structure," *Journal of Physics: Conference Series*, vol. 783, p. 12039, 2017, doi: 10.1088/1742-6596/783/1/012039.
- [17] S. Khatoon, D. Gupta, and L. K. Das, "PID & LQR control for a quadrotor: Modeling and simulation," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2014, doi: 10.1109/icacci.2014.6968232.
- [18] H. Jiang, Y. Xia, R. Hu, D. Ma, and C. Hao, "A Feedback Linearization and Saturated Control Structure for Quadrotor UAV," *2019 Chinese Control Conference (CCC)*, IEEE, pp. 8277–8282, 2019, doi: 10.23919/chicc.2019.8866172.
- [19] L. A. Blas, M. Bonilla, S. Salazar, M. Malabre, and V. Azhmyakov, "Synthesis of a robust linear structural feedback linearization scheme for an experimental quadrotor," *2019 18th European Control Conference (ECC)*. IEEE, pp. 1431–1436, 2019, doi: 10.23919/ecc.2019.8796174.
- [20] M. Shaffna and A. Gopinath, "Position control of Quadrotor using Feedback Linearizations," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 7, no. 4, pp. 3942–3950, 2018, doi: 10.15680/IJRSET.2018.0704114.
- [21] M. J. Reinoso, L. I. Minchala, P. Ortiz, D. F. Astudillo, and D. Verdugo, "Trajectory tracking of a quadrotor using sliding mode control," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2157–2166, 2016, doi: 10.1109/tla.2016.7530409.
- [22] M. Labbadi, M. Cherkaoui, Y. El houm, and M. Guisser, "Modeling and Robust Integral Sliding Mode Control for a Quadrotor Unmanned Aerial Vehicle," *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*. IEEE, pp. 1–6, 2018, doi: 10.1109/irsec.2018.8702881.
- [23] C. Massé, "Modélisation et commande d'un quadricoptère en présence de vent," Thesis, Montreal University, 2018.
- [24] A. L'Afflito, R. B. Anderson, and K. Mohammadi, "An Introduction to Nonlinear Robust Control for Unmanned Quadrotor Aircraft: How to Design Control Algorithms for Quadrotors Using Sliding Mode Control and Adaptive Control Techniques [Focus on Education]," *IEEE Control Systems*, vol. 38, no. 3, pp. 102–121, 2018, doi: 10.1109/mcs.2018.2810559.
- [25] J. Kim, M.-S. Kang, and S. Park, "Accurate Modeling and Robust Hovering Control for a Quad-rotor VTOL Aircraft," *Selected papers from the 2nd International Symposium on UAVs*, Reno, Nevada, U.S.A., Springer Netherlands, 2009, pp. 9–26, doi: 10.1007/978-90-481-8764-5\_2.
- [26] I. M. Khairuddin, A. P. P. A. Majeed, A. Lim, J. A. M. Jizat, and A. A. Jaafar, "Modelling and PID Control of a Quadrotor Aerial Robot," *Advanced Materials Research*, vol. 903, pp. 327–331, 2014, doi: 10.4028/www.scientific.net/amr.903.327.
- [27] A. Ataka *et al.*, "Controllability and observability analysis of the gain scheduling based linearization for UAV quadrotor," *2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, IEEE, pp. 212–218, 2013, doi: 10.1109/robionetics.2013.6743606.
- [28] K. S. Gaeid, H. W. Ping, H. A. F. Mohamed, and L. H. Hasam, "NNPID Controller for Induction Motors with Faults," *The Second International Conference on Control, Instrumentation and Mechatronic Engineering (CIM09)*, pp. 548–552, 2009.
- [29] J. Li and Y. Li, "Dynamic analysis and PID control for a quadrotor," *2011 IEEE International Conference on Mechatronics and Automation*. IEEE, 2011, doi: 10.1109/icma.2011.5985724.
- [30] P. Saraf, M. Gupta, and A. M. Parimi, "A Comparative Study Between a Classical and Optimal Controller for a Quadrotor," *2020 IEEE 17th India Council International Conference (INDICON)*, IEEE, pp. 1–6, 2020, doi: 10.1109/indicon49873.2020.9342485.
- [31] F. Ahmad, P. Kumar, A. Bhandari, and P. P. Patil, "Simulation of the Quadcopter Dynamics with LQR based Control," *Materials Today: Proceedings*, vol. 24, pp. 326–332, 2020, doi: 10.1016/j.matpr.2020.04.282.
- [32] L. Martins, C. Cardeira, and P. Oliveira, "Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019, doi: 10.1016/j.ifacol.2019.11.195.
- [33] J.-J. E. Slotine and W. Li, "Applied Nonlinear control (王飞硕士在读)," *Prentice hall*, vol. 8, no. 6, 2011.
- [34] C. Aguilar-Avelar and J. Moreno-Valenzuela, "A feedback linearization controller for trajectory tracking of the Furuta pendulum," *2014 American Control Conference*, IEEE, pp. 4543–4548, 2014, doi: 10.1109/acc.2014.6858724.
- [35] T. Alsuwian, R. Ordonez, and L. Jacobsen, "Comparison of PID and nonlinear feedback linearization controls for longitudinal dynamics of hypersonic vehicle at subsonic speeds," *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*. IEEE, pp. 207–213, 2016, doi: 10.1109/naecon.2016.7856800.
- [36] P. Leng, Y. Li, D. Zhou, J. Li, and S. Zhou, "Decoupling Control of Maglev Train Based on Feedback Linearization," *IEEE Access*, vol. 7, pp. 130352–130362, 2019, doi: 10.1109/access.2019.2940053.
- [37] M. Zolfaghari, M. Abedi, and G. B. Gharehpetian, "Robust Nonlinear State Feedback Control of Bidirectional Interlink Power Converters in Grid-Connected Hybrid Microgrids," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1117–1124, 2020, doi: 10.1109/jsyst.2019.2919551.
- [38] L. Callegaro, M. Ciobotaru, D. J. Pagano, and J. E. Fletcher, "Feedback Linearization Control in Photovoltaic Module Integrated Converters," *IEEE Transactions on Power Electronics*, vol. 34, no. 7, pp. 6876–6889, 2019, doi: 10.1109/tpel.2018.2872677.
- [39] H. Loubar, R. Z. Boushaki, A. Aouati, and M. Bouanzoul, "Sliding Mode Controller for Linear and Nonlinear Trajectory Tracking of a Quadrotor," *International Review of Automatic Control (IREACO)*, vol. 13, no. 3, p. 128, 2020, doi: 10.15866/ireaco.v13i3.18522.
- [40] J. Liu and X. Wang, "Normal Sliding Mode Control," *Advanced Sliding Mode Control for Mechanical Systems*, Springer Berlin Heidelberg, pp. 41–80, 2011, doi: 10.1007/978-3-642-20907-9\_2.
- [41] H. Hou, J. Zhuang, H. Xia, G. Wang, and D. Yu, "A simple controller of minisize quad-rotor vehicle," *2010 IEEE International Conference on Mechatronics and Automation*, IEEE, pp. 1701–1706, 2010, doi: 10.1109/icma.2010.5588802.
- [42] V. Utkin, J. Guldner, and J. Shi, "The Chattering Problem," *Sliding Mode Control in Electro-Mechanical Systems*, CRC Press, pp. 159–204, 2017, doi: 10.1201/9781420065619-8.




**BIOGRAPHIES OF AUTHORS**

**Samira Hadid**    received the engineer degree in Electromecanique from Boumerdes University, Algeria, in 2005. Received the magister degree in Electrical Engineering from Boumerdes University in 2010. She currently pursuing the Ph.D. degree at the same university. She worked as associate Professor since 2019 at Hydrocarbon and Chemistry Faculty in Boumerdes University until this day. Her current research interests include unmanned aerial vehicle systems, advanced control, nonlinear control, and intelligent control. She can be contacted at email: sa.hadid@univ-boumerdes.dz.



**Razika Boushaki Zamoum**    is a Teacher in Electrical Engineering at the university of Boumerdes (Algeria) in the Institute of Electrical and Electronic Engineering since 2003. She obtained Engineer Diploma in 1995, magister Diploma in 2003 at University of Boumerdes and Doctorate degree in June 2013, in Electrical Engineering. She is member in the research laboratory since 2009. She introduced several practical automation systems in industry between 1999 and 2003. Currently, she is Prof. at Institute of Electrical and Electronic Engineering, University M'hamed Bougara of Boumerdes, Algeria. Her research interests include intelligent control, fuzzy logic control, advanced control, nonlinear control, and adaptive control. She can be contacted at email: boushakiraz@yahoo.fr and r.boushaki@univ-boumerdes.dz.



**Youcef Refis**    was a Master student in Automatic "Control" at Institute of Electrical and Electronic Engineering at University M'hamed Bougara of Boumerdes, Algeria in 2022. He is a graduate holding both a Bachelor's degree in Electronics and Electrical Engineering in 2020 from Institute of Electrical & Electronic Engineering – University M'hamed Bougara, and a Master's degree in Control Systems in 2022 from the same university. Passionate about the fields of electronics and control systems, he completed his academic journey with distinction. He can be contacted at email: refisyoucef.05@gmail.com.