



## Advanced Hybrid Nonlinear Control for Morphing Quadrotors

Reyam G. Ghane\*<sup>id</sup>, Mohammed Y. Hassan<sup>id</sup>

Department of Control and Systems Engineering, University of Technology-Iraq, Baghdad 10075, Iraq

Corresponding Author Email: [cse.20.19@grad.uotechnology.edu.iq](mailto:cse.20.19@grad.uotechnology.edu.iq)

<https://doi.org/10.18280/mmep.100414>

### ABSTRACT

**Received:** 13 October 2022

**Revised:** 5 April 2023

**Accepted:** 2 May 2023

**Available online:** 30 August 2023

#### **Keywords:**

*nonlinear PID control, sliding mode control, foldable quadrotor*

This paper presents a novel nonlinear control strategy tailored for foldable quadrotors capable of altering their morphology in-flight. By changing the quadrotor's shape through servo motors, these adaptive systems can effectively overcome various challenges, including diverse environmental conditions, such as obstacles and climate variations. The proposed control approach utilizes a double-loop control method to enhance the robustness and performance of the folding quadrotor during flight. The outer loop consists of a nonlinear PID controller responsible for regulating motion in the X and Y directions, while the inner loop features a sliding mode controller for altitude control in the Z direction and attitude stabilization. This dual-loop structure ensures the quadrotor's stability even during arm folding. The firefly algorithm is employed to optimize the parameters of both controllers, minimizing position errors using the Root Mean Square Error (RMSE) as a performance metric. Our results demonstrate a significant improvement in the quadrotor's performance and adaptability to various proposed morphologies, with error rates reduced to near-negligible levels for all shapes. Specifically, the X position error was reduced by 100% for all morphologies, the Y position error by 95% for the X shape, 97% for the T shape, 98% for the H shape, and 99% for the O shape, and the Z position error by 99% for all morphologies.

## 1. INTRODUCTION

In recent years, quadrotors have garnered significant attention from researchers and scientists due to their potential applications in various fields, both civilian and military. These applications include search and rescue, surveillance, photography, and delivery [1]. Consequently, numerous studies have been conducted to enhance the stability of quadrotors during flight [2, 3] and to improve their utility in delivery applications [4, 5]. However, conventional quadrotor designs are not well-suited for navigating small spaces and irregular surfaces, owing to their inherent stability. This limitation has led to a growing demand for adaptable quadrotors capable of maintaining in-flight stability while encountering obstacles.

To address this need, several researchers have proposed variable-geometry quadrotors capable of altering their shape using techniques such as origami or by simulating bird flight [6-9]. Nonetheless, morphological changes can compromise quadrotor stability, necessitating the development of specialized control systems. For instance, Model Reference Adaptive Control was implemented to control the angular velocity of an X-morphology quadrotor and account for shifts in the center of gravity, proving effective in adapting to folding maneuvers [10]. However, the alignment of the servo motor's axis with the robot's center of rotation weakens the mechanical structure [10]. A PID controller was designed for a quad-morphing robot, but roll control was lost when the robot was folded due to the alignment of the four rotors [11]. The feedback control module, designed using a Lyapunov

stabilization approach, effectively tracked the desired roll angle but became less efficient when the arms were folded, increasing the risk of collision [12]. The Linear Quadratic Regulator controller, in conjunction with the computation of the inertia matrix, ensured stable flight but reduced the aircraft's flight time [13]. The backstepping controller, with parameters optimized using the PSO algorithm, showed acceptable stability, precision, and speed in simulations, but required manual changing of configurations, limiting the quadrotor's autonomy [14].

In this study, the problem of maintaining stability during morphological changes and tracking the desired trajectory of a foldable quadrotor is addressed. A double-loop control strategy is developed, utilizing two types of nonlinear controllers, namely Sliding Mode Control (SMC) and Nonlinear PID. The primary objective is to design the proposed inner and outer loop controllers, optimizing their parameters using the firefly algorithm to minimize position errors across various quadrotor shapes while ensuring flight continuity during shape changes.

The remainder of this paper is organized as follows: The mechanical design of the foldable quadrotor and the effects of its center of gravity are described in Section 2. The proposed controllers for the foldable quadrotor are elucidated in Section 3. The proposed control unit design is presented in detail in Section 4. An overview of the firefly algorithm and its application for parameter optimization are provided in Section 5. The simulation results are discussed in detail in Section 6. Finally, conclusions are offered in Section 7.

## 2. MECHANICAL DESIGN OF A FOLDABLE QUADROTOR

The foldable quadrotor, as shown in Figure 1, consists of a rigid central body  $m_o$  and four arms of masses  $m_{2,i}$  that is foldable around the main body using servo motors and at different angles, see Table 1 [15]. Since these angles represent the extension of the folded arm by calculating the potential control matrix for each shape, the possible shapes of this quad are (X, H, O, Y, YI, T) [14]. Servo motors of masses  $m_{1,i}$  are connected to the central body and the four arms are connected to the servo motors, as well as four rotors of masses  $m_{3,i}$  are located on the arms. Two rotors (2 and 4) are rotated in a clockwise direction and other two rotors (1 and 3) are rotated in a counterclockwise direction. Due to changing the quadrotor's morphology, the servo motors make the system highly coupled. The main body and servo motors are fixed around Z-axis while the arms and rotors rotate on the same axis (Z-axis). So, the most (conventional) setup leads to numerous conceivable arrangements by changing the position of the arms [15].

Some assumptions are presented to make a suitable model for the foldable quadrotor [16]:

- 1- The central body, servo motors, and rotation arms are rigid.
- 2- The propeller is assumed to be rigid, neglecting the effect of changing morphology during flight.

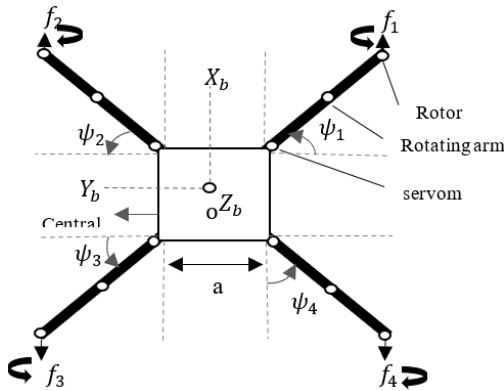


Figure 1. Schematic of the foldable quadrotor [15]

Table 1. Characteristics of the foldable quadrotor configuration [14]

Configuration	Arm Angles			
	$\psi_1(t)$	$\psi_2(t)$	$\psi_3(t)$	$\psi_4(t)$
X	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$
H	$\pi/2$	0	$\pi/2$	0
O	$\pi$	$\pi$	$\pi$	$\pi$
Y	$\pi/4$	$\pi/4$	$\pi/2$	0
YI	$\pi/2$	0	$\pi/4$	$\pi/4$
T	0	$\pi/2$	$\pi/2$	0

### 2.1 Kinematic model of a foldable quadrotor

The mathematical model of the quadrotor is described in two frames see Figure 1. The body frame  $R_b(O_b, X_b, Y_b, Z_b)$ , which is fixed and the mobile frame  $R_m(O_m, X_m, Y_m, Z_m)$ . The rotation from  $R_b$  to  $R_m$  consists of three revolutions around (X, Y, Z) axes, which means that the position is obtained at an angle  $(\varphi, \psi)$ . So, the total rotation matrix  $R$  can be given as [15]:

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

where,

$R \in SO(3) = \{R \in R^{3 \times 3} | R^T R = I_{3 \times 3}, \det(R)=1\}$  s(.) and c(.) are abbreviations for sin(.) and cos(.), respectively. The linear velocity of the system is represented by  $A = [u \ v \ w]^T \in R^3$  and angular velocity is represented by  $\zeta = [p \ q \ r]^T \in R^3$ . Both of the velocities are derived in the mobile frame  $R_m$ . The mass of the foldable quadrotor is  $m$  and the inertia is represented as  $(\psi_i(t)) \in R^{3 \times 3}$ .  $\xi = [X \ Y \ Z]^T \in R^3$  describes the position and  $\eta = [\varphi \ \theta \ \psi]^T \in R^3$  describes the orientation of the quadrotor.

### 2.2 Dynamic model of a foldable quadrotor

The dynamic model of the foldable quadrotor is derived from the Newton-Euler formalism and given as follows [14]:

$$\begin{bmatrix} m J_{3 \times 3}(\psi_i(t)) & O_{3 \times 3} \\ O_{3 \times 3} & J_{3 \times 3}(\psi_i(t)) \end{bmatrix} \begin{bmatrix} \dot{A} \\ \dot{\zeta} \end{bmatrix} + \begin{bmatrix} \zeta \times mA \\ \zeta \times J(\psi_i(t)) \end{bmatrix} = \begin{bmatrix} f \\ \tau \end{bmatrix} \quad (2)$$

where,  $O_{3 \times 3}$  is a zero-matrix dimensional  $3 \times 3$ , and  $\times$  denotes the cross product.

Many forces act on the quadrotor, such as gravity, thrust, hub and drag. So, the total force in the mobile frame is expressed [14]:

$$f = \begin{bmatrix} U_x \\ U_y \\ F_t \end{bmatrix} + R^T \begin{bmatrix} F_{Dx} \\ F_{Dy} \\ F_{Dz} \end{bmatrix} + R^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (3)$$

The relation between the velocities and the external forces is  $f = [f_x \ f_y \ f_z]^T \in R^3$  and  $f_g = [0 \ 0 \ -mg]^T$  is the gravity vector, where  $g$  is the gravity.  $F_i$  is the total thrust force generated by the rotation of each rotor, two components represent the hub forces  $U_x, U_y$ . Also, the quadrotor generates drag force  $F_D \in R^3$  in the body.

The quadrotor can turn around  $x_m, y_m, z_m$  directions by applying the moments  $\tau_\varphi, \tau_\theta, \tau_\psi$ . These moments depend on the folding angles of the arms and are induced by thrust forces  $F_i$ . Thus, the moments of roll, pitch and yaw are in the study [14].

$$\tau = \begin{bmatrix} \tau_\varphi + O_x + L_x + M_{AX} \\ \tau_\theta + O_y + L_y + M_{AY} \\ \tau_\psi + H + M_{AZ} \end{bmatrix} \quad (4)$$

Wind effects can produce an aerodynamic friction torque as [14].

$$M_A = [M_{Ax} \ M_{Ay} \ M_{Az}]^T = \text{diag}(k_{ax} \ k_{ay} \ k_{az}) \zeta^2 \quad (5)$$

where,  $k_{a(x,y,z)}$  is the aerodynamic friction coefficient.

$(O_x, O_y)$  Are the Gyroscopic moments,  $(L_x, L_y)$  are the flapping moment, and  $(H)$  is the hub moment. However, some phenomena have a slight effect, such as hub forces and blade flapping moments. Thus, they are neglected because the quadrotor adopted in this research does not perform any aggressive rotation of the four arms during flight [15].

## 2.3 Center of gravity

The possible shapes of the folding quadrotor are asymmetrical, so the center of gravity changes from one form to another, as each state has its center of gravity. As a result, it is necessary to calculate the center of gravity for each shape. The general formula of the center of gravity is shown below [6]:

$$OG = \frac{\sum_{i=1}^4 \sum_{j=1}^3 m_{j,i} \overline{OG}_{j,i}}{m_o + \sum_{i=1}^4 \sum_{j=1}^3 m_{j,i}} \quad (6)$$

where:

$O$  is the geometric center.

$\overline{OG}$  is the offset between the geometric center of the system and the global center of gravity.

$\overline{OG}_o \in R^{3 \times 1}$  is the vector of the center of gravity of the central body, which is zero.

$\overline{OG}_{1,l} \in R^{3 \times 1}$  is the vector of the center of gravity of the servo motors.

$\overline{OG}_{2,l} \in R^{3 \times 1}$  is the vector of the center of gravity of the rotating arms.

$\overline{OG}_{3,l} \in R^{3 \times 1}$  is the vector of the center of gravity of the rotors.

## 2.4 Control matrix

The control matrix presents the critical values of the foldable quadrotor control architecture, as it converts the square of the rotational speed of the four propellers ( $\Omega_i^2 |_{i=1,2,3,4}$ ) into the total thrust force  $F_i$  and moments  $\tau_\varphi, \tau_\theta, \tau_\psi$  [6]:

$$\Delta(\psi_i(t)) = \begin{bmatrix} b & b(y_{3,1} - y_g) & b(x_g - x_{3,1}) & d \\ b & b(y_{3,2} - y_g) & b(x_g - x_{3,2}) & -d \\ b & b(y_{3,3} - y_g) & b(x_g - x_{3,3}) & d \\ b & b(y_{3,4} - y_g) & b(x_g - x_{3,4}) & -d \end{bmatrix}^T \quad (7)$$

Define the control vector of its altitude and attitude as [6]:  $u = [u_1 \ u_2 \ u_3 \ u_4]^T \in R^{4 \times 4}$ , where,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \Delta(\psi_i(t)) \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (8)$$

$b$  and  $d$  are the aerodynamic coefficients and are taken as [6]:

$$f_i = b\Omega_i^2, \quad Q_i = d\Omega_i^2 \quad (9)$$

where,  $b$  is the Thrust coefficient and  $d$  is the drag coefficient.

The foldable quadrotor has eight control inputs. The primary four inputs deliver the control vector altitude and attitude. The second four control inputs provide the servo motors' control vector. To establish the complete dynamic model, a state space vector is given, which consists of a set of inputs, outputs and states associated with first-order differential equations, which will form a model of the space. A space with state variables as its axes is referred to as a "state space". The state of the system can be represented as a vector.

The general state space model is represented as following:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = cx(t) \end{cases} \quad (10)$$

where,  $x(t)$ ,  $y(t)$ , and  $u(t)$  presented the state, output and control vector, respectively.  $A$ ,  $B$ , and  $C$  shows system matrix, input and output matrix, respectively.

So, based on Eqs. (2), (3) and (4), the state space vector of the nonlinear dynamic model is given as [14]:

$$X = [x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z} \ \varphi \ \dot{\varphi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^T \quad (11)$$

Also, the state space vector is written as [14]:

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \quad (12)$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = U_1 \frac{u_x}{m} + b_1 x_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = U_1 \frac{u_y}{m} + b_2 x_4 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -g + U_1 \frac{\cos \varphi \cos \theta}{m} + b_3 x_6 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_1 \dot{\theta} \dot{\psi} + a_4 \dot{\theta} \Omega_r + \frac{1}{J_{xx}} U_2 + b_4 \dot{\varphi}^2 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_2 \dot{\varphi} \dot{\psi} + a_5 \dot{\varphi} \Omega_r + \frac{1}{J_{yy}} U_3 + b_5 \dot{\theta}^2 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_3 \dot{\theta} \dot{\psi} + \frac{1}{J_{zz}} U_4 + b_6 \dot{\psi}^2 \end{cases} \quad (13)$$

where,  $a_1 = \frac{J_{yy} - J_{zz}}{J_{xx}}$ ,  $a_2 = \frac{J_{zz} - J_{xx}}{J_{yy}}$ ,  $a_3 = \frac{J_{xx} - J_{yy}}{J_{zz}}$ ,  $a_4 = \frac{-J_r}{J_{xx}}$ ,  $a_5 = \frac{J_r}{J_{yy}}$ ,  $b_1 = \frac{-k_{Dx}}{m}$ ,  $b_2 = \frac{-k_{Dy}}{m}$ ,  $b_3 = \frac{-k_{Dz}}{m}$ ,  $b_4 = \frac{-k_{Ax}}{J_{xx}}$ ,  $b_5 = \frac{-k_{Ay}}{J_{yy}}$ ,  $b_6 = \frac{-k_{Az}}{J_{zz}}$ .

$$u_x = \sin \Psi \sin \varphi + \cos \Psi \sin \theta \cos \varphi \quad (14)$$

$$u_y = \sin \Psi \sin \theta \cos \varphi - \cos \Psi \sin \varphi \quad (15)$$

$$\Omega_r = \Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2 \quad (16)$$

where,  $J_{(xx,yy,zz)}$  is the moment of inertia  $J_r$ , is the rotor inertia,  $k_{D(x,y,z)}$ . Are the translation drag coefficients,  $k_{A(x,y,z)}$  are the aerodynamic friction coefficients, and  $\Omega$  is the rotor speed.

## 3. CONTROLLER DESIGN

The foldable quadrotor is a nonlinear, underactuated system, as well as a strong coupling due to servo motors that connect each arm to the main body making the system highly coupled. It has only four control inputs but must control six states as an underactuated system. As a result of these reasons, the double loop control strategy relied upon using two types of controllers, where the control system is divided into two parts. The first part is where the input control is available, called the inner

loop control, and uses the classic sliding mode control. While the second part is where no actual control input is available, it is called outer loop control and uses nonlinear PID.

### 3.1 Classical sliding mode controller

The sliding mode control is one of the methods used to control nonlinear systems, as it provides an efficient design to ensure the stability of the control unit [17]. The sliding mode control consists of two main parts:

- 1- The reaching phase, which urges the control in the sliding mode to go towards the sliding surface.
- 2- The sliding phase forces the reaching phase in the sliding mode to remain on the sliding surface and slide until reaching the original. The sliding mode control law is given as [18]:

$$u = u_{eq} + u_{dis} \quad (17)$$

where,  $u_{eq}$  is the nominal control part,  $u_{dis}$ . It is the discontinuous part. Define the discontinuous part as [18]:

$$u_{dis} = -k \text{sign}(s) \quad (18)$$

where,  $s$  is the sliding surface and it is assumed as:

$$s = \dot{e} + ce \quad (19)$$

### 3.2 Nonlinear PID controller

The traditional PID has been developed to have a greater degree of freedom to be more scalable to design for nonlinear systems. It was reconstructed by replacing the linear PID parts with nonlinear functions as shown in the equations below [19]:

$$U_{NLPID} = f_1(e) + f_2(\dot{e}) + f_3\left(\int edt\right) \quad (20)$$

$$f_i = k_i(\beta)|\beta|^{\alpha_i} \text{sign}(\beta) \quad (21)$$

$$k_i(\beta) = k_{i1} + \frac{k_{i2}}{1 + \exp(\mu_i \beta^2)}, i = 1, 2, 3 \quad (22)$$

## 4. DESIGN OF CONTROLLER FOR FOLDABLE QUADROTOR

The design process for the inner loop ( $u_1, u_2, u_3, u_4$ ) is based on the sliding mode control method, while the design for control signals ( $u_x, u_y$ ) for the motion in the  $(x, y)$  plane are based on the nonlinear PID (NLPID) control, see Figure 2.

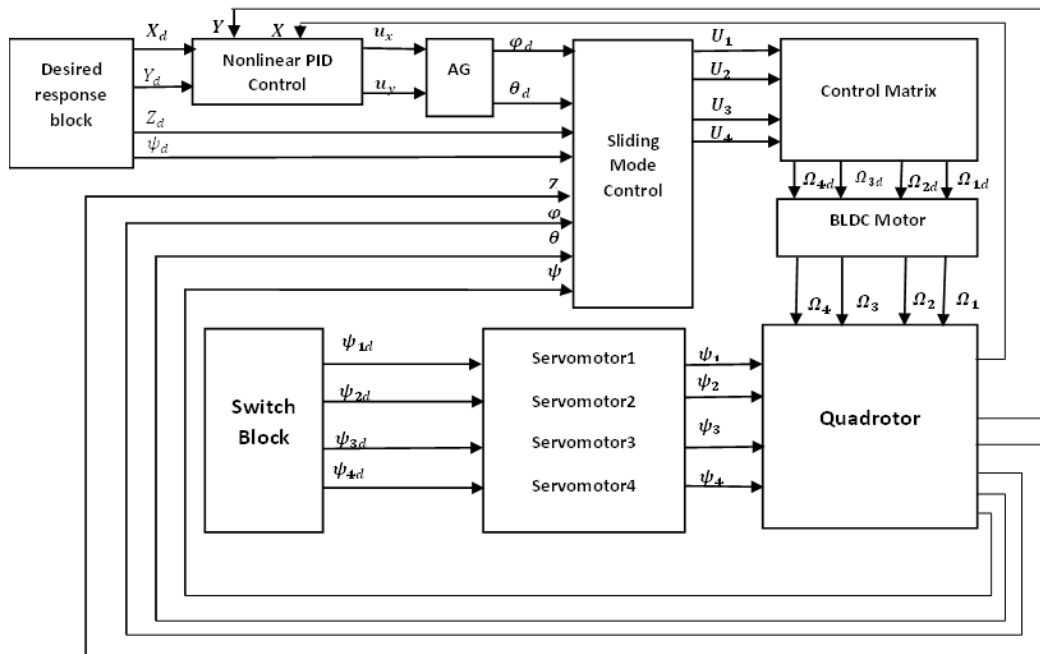


Figure 2. Proposed foldable quadrotor control design

### 4.1 Design of inner loop Sliding Mode Controller (SMC)

For the altitude (Z) [14]:

$$\begin{cases} \dot{x}_5 = x_6 \\ \dot{x}_6 = -g + U_1 \frac{\cos \varphi \cos \theta}{m} + b_3 x_6 \end{cases} \quad (23)$$

The tracking error of the altitude is  $e_5$  is expressed as:

$$e_5 = x_5 - x_{5d} \quad (24)$$

$$\dot{e}_5 = \dot{x}_5 - \dot{x}_{5d} = x_6 - \dot{x}_{5d} \quad (25)$$

$$\ddot{e}_5 = \ddot{x}_5 - \ddot{x}_{5d} = \dot{x}_6 - \ddot{x}_{5d} \quad (26)$$

where,  $x_5$  is the actual altitude and  $x_{5d}$  is the desired altitude.

The sliding surface for the altitude is:

$$s_5 = \dot{e}_5 + c_5 e_5 \quad (27)$$

Taking the derivative of the sliding surface:

$$\dot{s}_5 = \ddot{e}_5 + c_5 \dot{e}_5 \quad (28)$$

To guarantee  $\dot{s}_5 = 0$ ,  $U_1$  must be:

$$U_1 = \frac{m}{\cos x_7 \cos x_9} [\ddot{x}_{5d} + g - b_3 x_6 - c_5 (x_6 - \dot{x}_{5d}) - k_5 \operatorname{sgn}(s_5) - k_6 s_5] \quad (29)$$

Repeat the same steps as above to find the control signal for the attitude ( $\varphi$ ,  $\theta$  and  $\psi$ ), so:

$$U_2 = J_{xx} [\ddot{x}_{7d} - a_1 x_{10} x_{12} - a_4 x_{10} \Omega_r - b_4 x_8^2 - c_7 (x_8 - \dot{x}_{7d}) - k_7 \operatorname{sgn}(s_7) - k_8 s_7] \quad (30)$$

$$U_3 = J_{yy} [\ddot{x}_{9d} - a_2 x_8 x_{12} - a_4 x_8 \Omega_r - b_5 x_9^2 - c_9 (x_{10} - \dot{x}_{9d}) - k_9 \operatorname{sgn}(s_9) - k_{10} s_9] \quad (31)$$

$$U_4 = J_{zz} [\ddot{x}_{11d} - a_3 x_8 x_{10} - b_6 x_{12}^2 - c_{11} (x_{12} - \dot{x}_{11d}) - k_{11} \operatorname{sgn}(s_{11}) - k_{12} s_{11}] \quad (32)$$

## 4.2 Design of outer loop Nonlinear PID Controller (NLPID)

The Quadrotor system has no actual control input for the motion in the (X, Y) plane. The following analysis is proposed to generate suitable control signals ( $u_x$ ,  $u_y$ ) for the motion in the (X, Y) plane:

From Eqs. (14) and (15) [14]:

$$u_x = \sin \psi_d \sin \varphi_d + \cos \psi_d \sin \theta_d \cos \varphi_d$$

$$u_y = \sin \psi_d \sin \theta_d \cos \varphi_d - \cos \psi_d \sin \varphi_d$$

By solving these two equations:

$$\varphi_d = \operatorname{asin}(u_x \sin \psi_d - u_y \cos \psi_d) \quad (33)$$

$$\theta_d = \operatorname{asin}\left(\frac{u_x \cos \psi_d + u_y \sin \psi_d}{\cos \varphi_d}\right) \quad (34)$$

The above two equations represent the desired value for roll and pitch angles. The main objective of the NLPID controller is to approach the errors of (x and y) positions to zero as described below:

$$\ddot{e}_1 + U_{NLPID} = 0 \quad (35)$$

$$\ddot{x}_1 - \ddot{x}_{1d} + f_{x1}(e_x) + f_{x2}(\dot{e}_x) + f_{x3}\left(\int e_x dt\right) = 0 \quad (36)$$

$$\ddot{e}_3 + U_{NLPID} = 0 \quad (37)$$

$$\ddot{x}_3 - \ddot{x}_{3d} + f_{y1}(e_y) + f_{y2}(\dot{e}_y) + f_{y3}\left(\int e_y dt\right) = 0 \quad (38)$$

Letting ( $u_x$ ,  $u_y$ ) be the virtual control signals for (x, y) respectively, thus:

$$u_x = \frac{m}{U_1} \left[ \frac{k_{dx}}{m} x_2 + \ddot{x}_{1d} - f_{x1}(e_x) - f_{x2}(\dot{e}_x) + f_{x3}\left(\int e_x dt\right) \right] \quad (39)$$

$$u_y = \frac{m}{U_1} \left[ \frac{k_{dy}}{m} x_4 + \ddot{x}_{3d} - f_{y1}(e_y) - f_{y2}(\dot{e}_y) + f_{y3}\left(\int e_y dt\right) \right] \quad (40)$$

## 5. FIREFLY ALGORITHM

Firefly algorithm is a bio-inspired metaheuristic algorithm for optimization problems. It is noticeable that the bright fireflies attract all the less bright fireflies and that the fireflies are attracted to each other. Typically since the attraction is straightforwardly relative to the brightness and inversely proportional to the distance between them. As a result, it is conceivable to form an unused arrangement by gravity and irregular strolling of fireflies since fireflies move randomly. This algorithm is based on two key ideas [20]:

1. The light intensity emitted.
2. The degree of Attractiveness that is generated between two fireflies.

For the light intensity of Firefly ( $i$ ,  $I_i$ ) depends on the intensity  $I_o$  of light emitted by firefly  $i$  and the distance  $r$  between firefly  $i$  and  $j$  [20]

$$I_i = I_o e^{-\gamma r_{ij}^2} \quad (41)$$

The Attractiveness  $\beta_{ij}$  of the Firefly,  $i$  depends on the light intensity seen by an adjacent firefly  $j$  and its distance  $r_{ij}$ , then the Attractiveness  $\beta_{ij}$  is [20]:

$$\beta_i = \beta_o e^{-\gamma r_{ij}^2} \quad (42)$$

$\gamma$  is the light absorption coefficient.

$\beta_o$  is the Attractiveness.

$r_{ij}$  is the distance between any two fireflies and can represent by [20]:

$$r_{ij} = \sqrt{\sum (x_i - x_j)^2} \quad (43)$$

The movement of firefly  $i$  towards firefly  $j$  is represented by [20]:

$$x_i(t+1) = x_i(t) + \beta e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \epsilon_i \quad (44)$$

$\epsilon_i$  is a random parameter generated by a uniform distribution.  $\alpha$  is the parameter of scale.

If the light intensity of firefly  $i$  is greater than firefly  $j$ , the firefly moves randomly and is presented by the equation [20]:

$$x_i(t+1) = x_i(t) + \alpha \epsilon_i \quad (45)$$

A summary of the Firefly optimization algorithm is:

- Initialize the parameters.
- Generate a population of (n) fireflies.
- Calculate the fitness value of each firefly.
- Check if (t=1 to maximum iteration).
- Update position and light intensity for each firefly.
- Report the best solution.

In this paper, a Root Mean Square Error (RMSE) is involved in monitoring the performance of the proposed control system (performance index). This is to minimize the occurred errors of the virtual control ( $u_x$ ,  $u_y$ ) parameters and main control ( $u_z$ ,  $u_\psi$ ). The RMSE formula is expressed [21]:

$$RMSE = \sqrt{\frac{\sum_{i=0}^n (x_{dis,i} - x_{act,i})^2}{n}} \quad (46)$$

where,  
 $x_{dis}$  is the desired value.  
 $x_{act}$  is the actual value.

## 6. SIMULATION RESULTS

The nonlinear system of the 6-degrees of freedom and the proposed controller for this quadrotor is simulated using Matlab/Simulink. This foldable quadrotor is simulated under ideal conditions (no external disturbances or uncertainties affected). The parameter values used in the simulation are listed in Table 2 [15]. Since this quadrotor has the ability to change the configuration according to the tasks assigned to it, it will have an important effect on the inertial matrix since each shape has a specific value of inertia, as in the Table 3 [15].

**Table 2.** Foldable quadrotor parameters [15]

Parameters	Description	Value
<b>m</b>	Mass platform	1,133g
<b>g</b>	Gravity	9.81
<b>l</b>	Length of arm	21cm
<b>d</b>	Drag coefficient	$1.61 \times 10^{-4}$
<b>b</b>	Thrust coefficient	$6.317 \times 10^{-4}$

**Table 3.** The possible inertia values for each shape [15]

Configuration	$J_{xx} (kg.m^2)$	$J_{yy} (kg.m^2)$	$J_{zz} (kg.m^2)$
<b>X</b>	$1.211 \times 10^{-2}$	$1.213 \times 10^{-2}$	$2.370 \times 10^{-2}$
<b>H</b>	$2.888 \times 10^{-3}$	$1.978 \times 10^{-2}$	$2.112 \times 10^{-2}$
<b>O</b>	$4.719 \times 10^{-3}$	$7.743 \times 10^{-3}$	$7.491 \times 10^{-3}$
<b>Y</b>	$7 \times 10^{-3}$	$1.597 \times 10^{-2}$	$2.241 \times 10^{-2}$
<b>YI</b>	$7 \times 10^{-2}$	$1.597 \times 10^{-2}$	$2.241 \times 10^{-2}$
<b>T</b>	$1.082 \times 10^{-2}$	$1.084 \times 10^{-2}$	$2.112 \times 10^{-2}$

The Fireflies algorithm is used to find the lowest value of the proposed cost function (Eq. (45)). The parameters used in the firefly algorithm are shown in Table 4. A reference path, shown in Figure 3, is applied. To calculate the cost function to set 32 parameters of the model, 24 parameters are used for the nonlinear PID controller, see Table 5, and 8 parameters are used for the sliding mode control, see Table 6. Satisfactory results were reached by proving that the minimum cost function reached as shown in Figure 4.

Followed the desired trajectory as when  $0 < t < 20$  sec.,  $x_d = 0$ ;  $20 < t < 30$ ,  $x_d = t - 20$ ;  $30 < t < 50$ ,  $x_d = 10$ ;  $50 < t < 60$ ,  $x_d = -1(t - 50)$ ;  $60 < t < 70$ ,  $x_d = 0$ . When  $0 < t < 30$ ,  $y_d = 0$ ;  $30 < t < 40$ ,  $y_d = t - 30$ ;  $40 < t < 60$ ,  $y_d = 10$ ;  $60 < t < 70$ ,  $y_d = -1(t - 60)$ . and when  $0 < t < 70$ ,  $z_d = 10$ . The quadrotor flights upward with a fixed path until it reaches a height of 10 m from the initial positions (X, Y, Z)=(0, 0, 0).

The simulation results were carried out using MATLAB/Simulink with a time set to 70 sec, As shown in the Figures 5 and 6 below.

**Table 4.** Firefly algorithm parameters

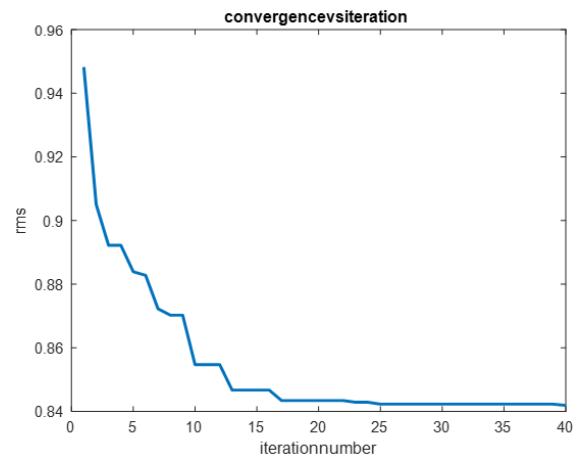
Parameters	Value
Light absorption coefficient	1
Attraction coefficient base value	1
Mutation coefficient	1
Mutation coefficient damping ratio	0.98
Population size	20
Iteration max	40

**Table 5.** Optimal parameters for nonlinear PID

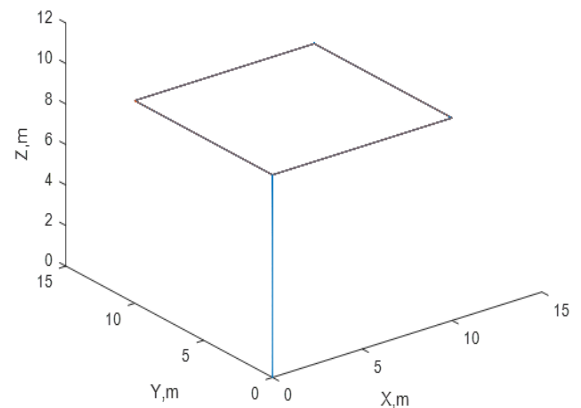
Parameter	Value	Parameter	Value	Parameter	Value
$k_{11x}$	3.0151	$y_{3x}$	0.01	$k_{31y}$	1.0242
$k_{12x}$	1.2504	$a_{1x}$	0.01	$k_{32y}$	2.5960
$k_{21x}$	4	$a_{2x}$	0.3450	$y_{1y}$	1.9688
$k_{22x}$	2.0480	$a_{3x}$	1.3632	$y_{2y}$	0.01
$k_{31x}$	0.1105	$k_{11y}$	4	$y_{3y}$	3.7522
$k_{32x}$	1.9498	$k_{12y}$	2.8314	$a_{1y}$	0.01
$y_{1x}$	2.5793	$k_{21y}$	3.1710	$a_{2y}$	1.9894
$y_{2x}$	0.01	$k_{22y}$	0.01	$a_{3y}$	0.8624

**Table 6.** Optimal parameters for sliding mode controller

Parameters	Value
$k_5$	4
$k_6$	4
$k_7$	3.5165
$k_8$	0.01
$k_9$	0.01
$k_{10}$	0.01
$k_{11}$	0.01
$k_{12}$	0.3410

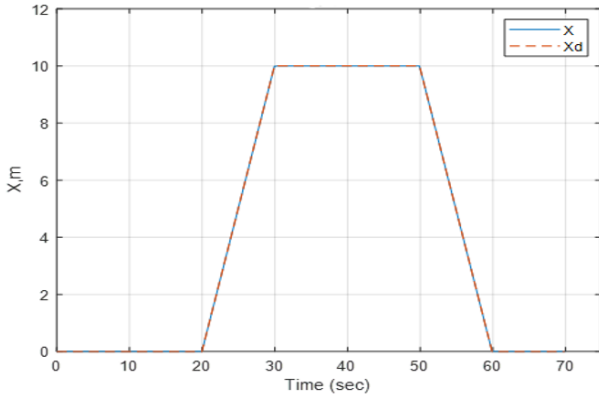


**Figure 3.** Performance of RMSE using Firefly algorithm

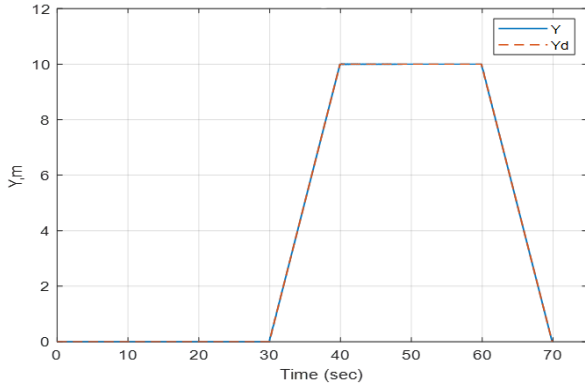


**Figure 4.** Control of a foldable quadrotor using trapezoidal trajectory

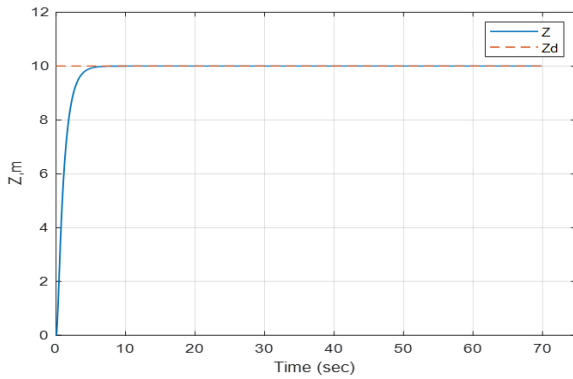
Figures 5 and 6 show the proposed controller's efficiency in ensuring the stability of all possible shapes of the quadrotor while following the proposed path as the error in position approaches zero. Figure 7 shows that the error in the x-axis position varies between 0.02 m to -0.02 m and approaches zero. The error in the y-axis position fluctuates between 0.024 m to -0.024 m and settles at 0.0006686 m.



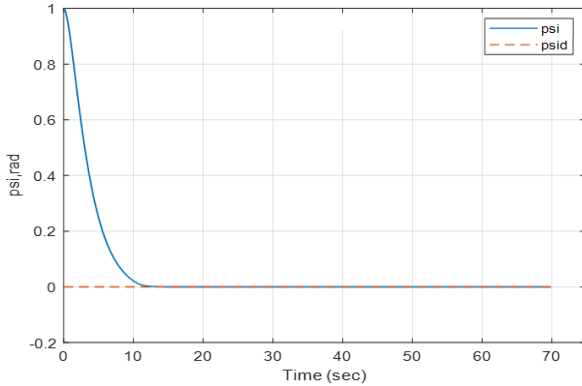
(a) X trajectory



(b) Y trajectory

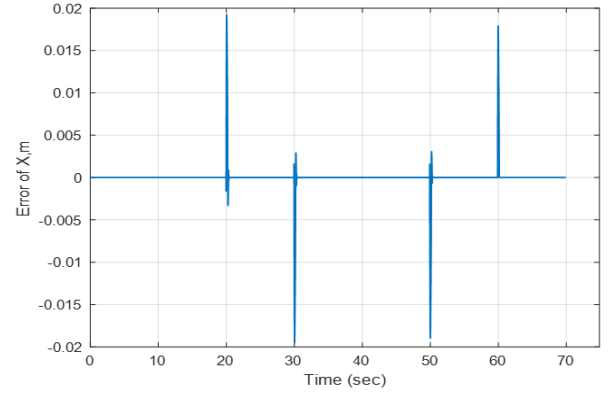


(c) Z trajectory

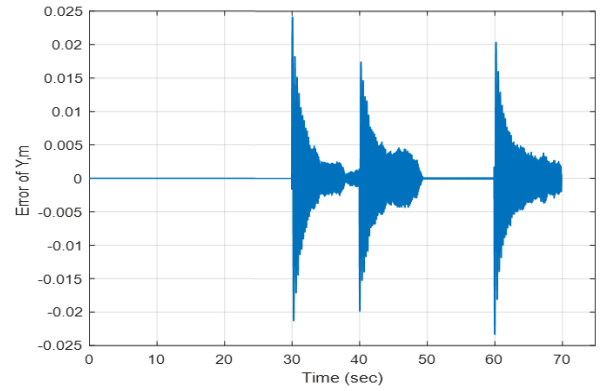


(d) Yaw angle

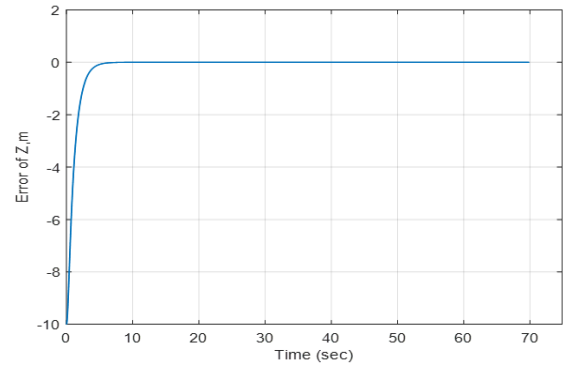
**Figure 5.** Response of the position and yaw angle: (a) X trajectory; (b) Y trajectory; (c) Z trajectory; (d) Yaw angle



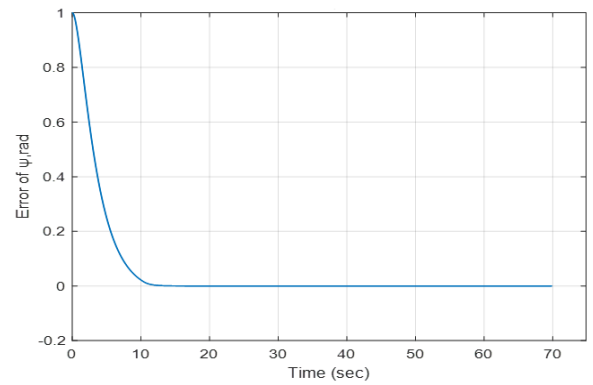
(a) Tracking error in X position



(b) Tracking error in Y position



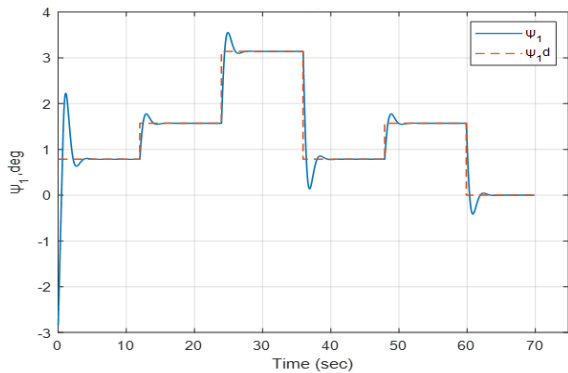
(c) Tracking error in Z position



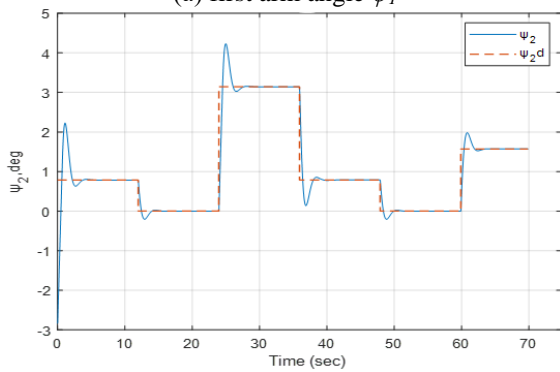
(d) Tracking error in yaw angle

**Figure 6.** Tracking error in position and yaw angle: (a) Tracking error in X position; (b) Tracking error in Y position; (c) Tracking error in Z position; (d) Tracking error in yaw angle

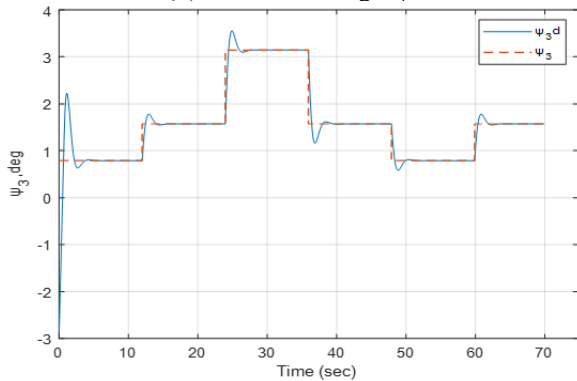
The square shape path, shown in Figure 5, is applied and the quadrotor flights upward with a fixed path until it reaches a height of 5 m from the initial positions (0,0,0). It is noticed from Figure 8 that the quadrotor follows the required path angles with small errors as changed the folded angles every 12 seconds. Using a switch, the foldable angles toggle between these values allowing the quadrotor to change its morphology.



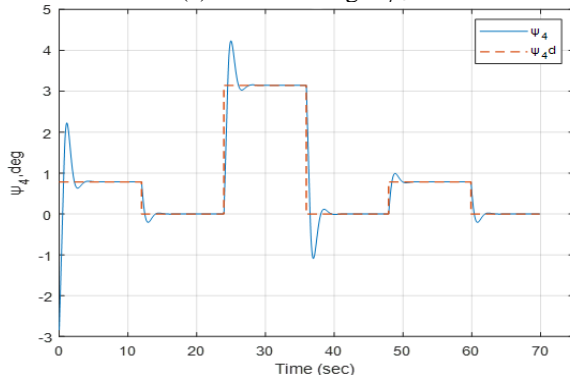
(a) first arm angle  $\psi_1$



(b) second arm angle  $\psi_2$

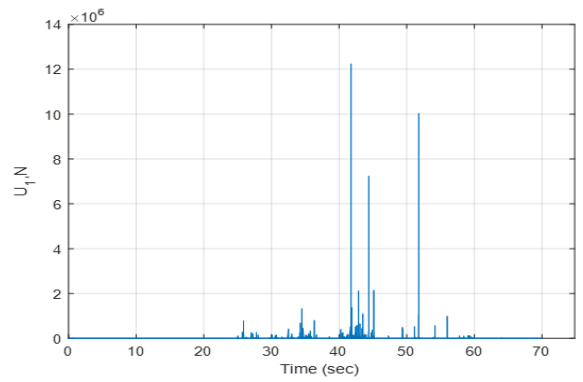


(c) third arm angle  $\psi_3$

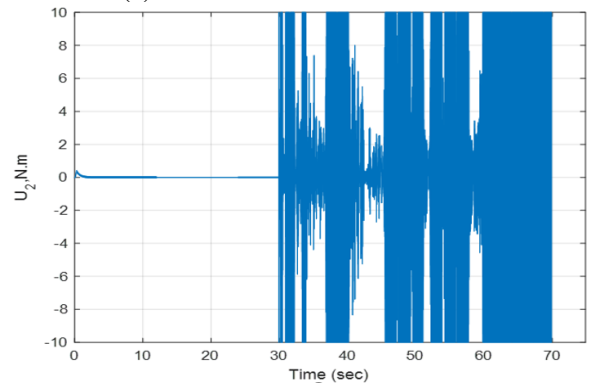


(d) fourth arm angle  $\psi_4$

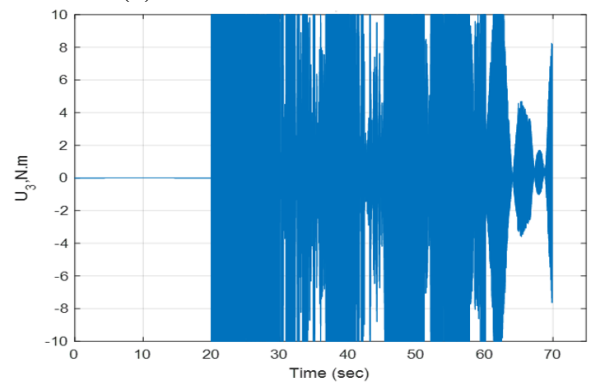
**Figure 7.** Servomotor's angle variations: (a) first arm angle  $\psi_1$ ; (b) second arm angle  $\psi_2$ ; (c) third arm angle  $\psi_3$ ; (d) fourth arm angle  $\psi_4$



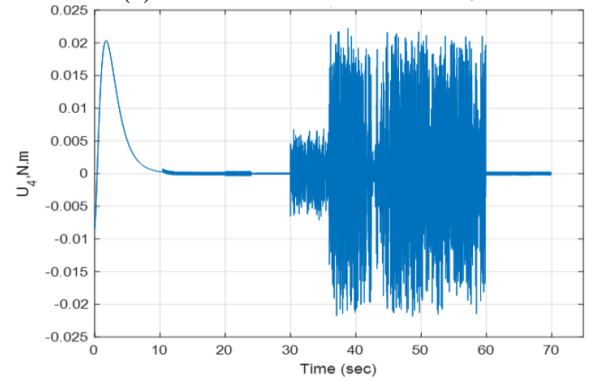
(a) Evolution of the command  $U_1$



(b) Evolution of the command  $U_2$



(c) Evolution of the command  $U_3$



(d) Evolution of the command  $U_4$

**Figure 8.** Control signals. (a) Evolution of the command  $U_1$ ; (b) Evolution of the command  $U_2$ ; (c) Evolution of the command  $U_3$ ; (d) Evolution of the command  $U_4$

The value of the control signal  $U_1$  changes during the trajectory to reach its highest value after 40 seconds and settles at a fixed value of around 17.76 N. As for the second and third control signals  $U_2, U_3$ , differences appear due to the presence of the multirotor. The fourth control signal  $U_4$  takes the highest value at the beginning of the path with the presence of



fluctuations and then stabilizes at zero.

## 7. CONCLUSIONS

This paper presents a proposed architectural control for the foldable quadrotor, known for its complexity and non-linearity, in addition to its high coupling due to the presence of servo motors. Therefore, a control strategy based on the double loop was proposed using two nonlinear controllers (sliding mode control in the inner loop and nonlinear PID in the outer loop). Fireflies algorithm based on the Root Mean Square performance indicator was used to improve parameters in each controller.

The proposed controller proved its efficiency in reducing errors in the position and deflection angles and making them close to zero and the controller is suitable for all possible shapes as compared with reference number 13, while maintaining the transit stability of the quadrotor even with the changes its shape.

## REFERENCES

- [1] Chen, T., Shan, J., Liu, H.H.T. (2022). Transportation of payload using multiple quadrotors via rigid connection. *International Journal of Aerospace Engineering*. <https://doi.org/10.1155/2022/2486561>
- [2] Kadhim, M.Q., Hassan, M.Y. (2020). Design and optimization of backstepping controller applied to autonomous quadrotor. In *IOP Conference Series: Materials Science and Engineering*, 881(1): 012128. <https://doi.org/10.1088/1757-899X/881/1/012128>
- [3] Muhsen, A., Raafat, S. (2021). Optimized PID control of quadrotor system using extremum seeking algorithm. *Engineering and Technology Journal*, 39(6): 996-1010. <https://doi.org/10.30684/etj.v39i6.1850>
- [4] Taha, M.Y., Mohamed, M.J., Ucan, O.N. (2020). Optimal 3D path planning for the delivery quadcopters operating in a city airspace. In *ISMSIT 2020 - Proceedings of the 4th International Symposium on Multidisciplinary Studies and Innovative Technologies*, Istanbul, Turkey, pp. 1-6. <https://doi.org/10.1109/ISMSIT50672.2020.9254715>
- [5] Alawsi, A.A.A., Jasim, B.H., Raafat, S.M. (2020). Improved positioning of delivery quad copter using GPS - accelerometer multimode controller and unscented kalman filter. *IOP Conference Series: Materials Science and Engineering*, 745(1): 012018. <https://doi.org/10.1088/1757-899X/745/1/012018>
- [6] Derrouaoui, S.H., Bouzid, Y., Guiatni, M., Dib, I., Moudjari, N. (2020). Design and modeling of unconventional quadrotors. In *2020 28th Mediterranean Conference on Control and Automation (MED)*, Saint-Rapha, France, pp. 721-726. [https://doi.org/10.0/Linux-x86\\_64](https://doi.org/10.0/Linux-x86_64)
- [7] Zhao, N., Luo, Y., Deng, H., Shen, Y. (2017). The deformable quad-rotor: Design, kinematics and dynamics characterization, and flight performance validation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, pp. 2391-2396. [https://doi.org/10.0/Linux-x86\\_64](https://doi.org/10.0/Linux-x86_64)
- [8] Dilaveroğlu, L., Özcan, O. (2020). Minicore: A miniature, foldable, collision resilient quadcopter. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)* New Haven, CT, USA, pp. 176-181. <https://doi.org/10.1109/RoboSoft48309.2020.9115993>
- [9] Kim, S.J., Lee, D.Y., Jung, G.P., Cho, K.J. (2018). An origami-inspired, self-locking robotic arm that can be folded flat. *Science Robotics*, 3(16): eaar2915. <https://doi.org/10.1126/scirobotics.aar2915>
- [10] Desbiez, A., Expert, F., Boyron, M., Diperi, J., Viollet, S., Ruffier, F. (2017). X-Morf: A crash-separable quadrotor that morfs its X-geometry in flight. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, Sweden, pp. 222-227. <https://doi.org/10.1109/RED-UAS.2017.8101670>
- [11] Riviere, V., Manecy, A., Viollet, S. (2018). Agile robotic fliers: A morphing-based approach. *Soft Robotics*, 5(5): 541-553. <https://doi.org/10.1089/soro.2017.0120>
- [12] Yang, D., Mishra, S., Aukes, D.M., Zhang, W. (2019). Design, planning, and control of an origami-inspired foldable quadrotor. In *2019 American Control Conference (ACC)*, Philadelphia, PA, USA, pp. 2551-2556. <https://doi.org/10.23919/ACC.2019.8814351>
- [13] Falanga, D., Kleber, K., Mintchev, S., Floreano, D., Scaramuzza, D. (2019). The foldable drone: A morphing quadrotor that can squeeze and fly. *IEEE Robotics and Automation Letters*, 4(2): 209-216. <https://doi.org/10.1109/LRA.2018.2885575>
- [14] Derrouaoui, S.H., Bouzid, Y., Guiatni, M. (2021). PSO based optimal gain scheduling backstepping flight controller design for a transformable quadrotor. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 102(3): 67. <https://doi.org/10.1007/s10846-021-01422-1>
- [15] Derrouaoui, S.H., Bouzid, Y., Guiatni, M. (2021). Towards a new design with generic modeling and adaptive control of a transformable quadrotor. *The Aeronautical Journal*, 125(1294): 2169-2199. <https://doi.org/10.1017/aer.2021.54>
- [16] Wissal, T. (2021). Numerical model development of a transformable quadrotor. [http://thesis.essatlemcen.dz/handle/STDB\\_UNAM/249](http://thesis.essatlemcen.dz/handle/STDB_UNAM/249).
- [17] Majeed, H., Kadhim, S., Jaber, A. (2022). Design of a sliding mode controller for a prosthetic human hand's finger. *Engineering and Technology Journal*, 40(1): 257-266. <https://doi.org/10.30684/etj.v40i1.1943>
- [18] Hamoudi, A.K. (2014). Sliding mode controller for nonlinear system based on genetic algorithm. *Engineering and Technology Journal*, 32(11 Part (A) Engineering).
- [19] Najm, A.A., Ibraheem, I.K. (2019). Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Engineering Science and Technology, an International Journal*, 22(4), 1087-1097. <https://doi.org/10.1016/j.jestch.2019.02.005>
- [20] Johari, N.F., Zain, A.M., Mustaffa, N.H., Udin, A. (2013). Firefly algorithm for optimization problem. *Applied Mechanics and Materials*, 421: 512-517. <https://doi.org/10.4028/www.scientific.net/AMM.421.512>
- [21] Wright, H.J., Strydom, R., Srinivasan, M.V. A generalized algorithm for tuning UAS flight controllers. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, Dallas, TX, USA, pp. 1369-1378. <https://doi.org/10.1109/ICUAS.2018.8453451>