Designing a backstepping nonlinear controller for controlling six-blade vertical flight with new structure

Farhad Yousefi[1], Ehsan Nikpour[2]
1- Department of New Science and Technologies, University of Tehran, Tehran, Iran
farhadyousefi@ut.ac.ir
2- Department of Electrical Engineering, University of Shahab Danesh, Qom, Iran
ehsanikpour@gmail.com

Abstract
Quadrators have become one of the most widely used UAV (Unmanned Aerial Vehicle) robots due to their simple mechanical structure and high maneuverability. But due to their low cargo capacity, they are limited in some applications. In the model of this research, to increase the cargo capacity of the drone, two coaxial propellers with constant speed and opposite direction of rotation have been added to the center of the quadrator. The drone dynamic modeling has been done by Newton - Euler method and the obtained equations are non-linear, under-excited, coupled and unstable. To control the vertical flight, the backstepping control method which has a high ability to control nonlinear systems, has been used. The control system and the drone have been simulated in MATLAB software and the simulation results show that the designed controller displays a desirable behavior in vertical flight controlling.
Keywords: *Six-blade vertical flying robot - Unmanned Aerial Vehicle - backstepping control.*

1. Introduction

In recent decades, in robotics fields, many automatic and remote control devices have been developed. Also, there is a growing interest in the development of vertical flying unmanned aerial vehicles with various capabilities. Because of the numerous advantages, the developing of these drones have expanded. Reconnaissance missions, photo shoot, oil and electricity transmission line inspection, fire detection, using in dangerous and inaccessible environments, monitoring of agricultural and forestry borders can be mentioned among the applications [1]. Quadrator is considered as the vertical flying unmanned aerial vehicles. These types of drones can be remotely controlled or be self-driving. Cargo capacity, simplicity of the device structure, high maneuverability, low maintenance cost are among the advantages of using these robots. The low flight persistence of the quadrator is the only aspect that makes that weak compared to other flying robots. Due to the need for electrical energy to run four motors and the limitation of the energy source, these types of robots have a short flight time. Also, the vertical flying system of quadrator is a completely unstable system, that's why its control is very important. Optimal control of the robot makes the system stable, prevents the drone from falling, maintains the balance of the system and also controls the drone position during the flight. In the status control, the speed of the motors is adjusted to reach the desired angles and altitude. The stability of the flying robot is very important to perform a successful maneuver.

Various works have been done in the field of quadrator control. Among other things, in [2], the quadrator motion equations have been extracted and in [3], the effect of aerodynamic forces in quadrature modeling has been analyzed. Also, various non-linear control methods including feedback linearization with structures including cascade and higher order derivatives [4, 5] , sliding mode control method [6], use of adaptive methods [7], in [8, 9], Lyapunov theory is used to ensure the asymptotic stability of the system. In [10], the position tracking and the direction of these systems has been analyzed by using hybrid control. Very low cargo capacity is one of the main disadvantages of the quadrator; to solve this problem, four rotors out of six have been used symmetrically [11] or eight rotors [12] have been used. Increasing weight of drone and complicated control are the disadvantages of these structures. Adding two motors to the center of the quadrator, not only increases the cargo capacity, but also increases its resistance to crosswinds. The running of these central rotors with a specific moment of inertia acts like a gyroscope and resists the horizontal rotations of the robot. In order to keep the structure of the drone simple, the central rotors have no speed feedback and their speed is determined as an open circuit based on the weight of the drone and remains constant during its movement. Figure 1 shows a six-blade vertical flight robot with a new structure including two concentric rotors in the center of the robot.

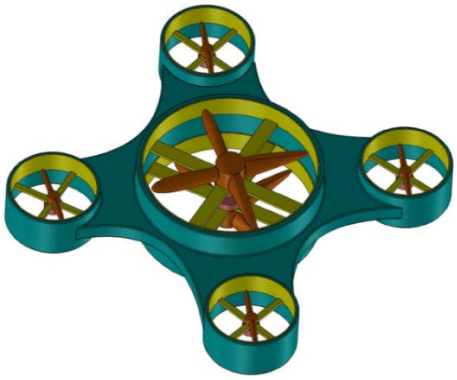

Figure 1. Six-blade vertical flight robot with a new structure

In the second part of this article, the dynamic model of the under research drone has been extracted by using Newton's method. In the third part, the control system is designed using the backstepping method. In the fourth part, the results of the simulation of the control system have been evaluated and analyzed, then the tracking of the system in following different routes has been tested. In the fifth section, the conclusion is given.

2. Dynamic modeling

Basically, Newton-Euler and Euler-Lagrange methods are used to express the dynamic model, in this article, Euler-Newton method is used.

2.1. Dynamic of six-blade drone

The vertical flight drone of this research has six rotors which fours are symmetrically installed around the drone and a pair of coaxial rotors with larger blades are installed in the center of the drone (see Figure 1). Coaxial rotors have a constant and similar speed and basically have no role in controlling the drone, but have been used in order to increase drone cargo capacity. To cancel each other's aerodynamic effect and maintain system balance, the rotating direction of these two rotors is considered to be opposite. Peripheral rotors are responsible for preparing desired movements in the drone, in fact, the speed of these rotors is the input of the system.

The six-blade robot has six degrees of freedom in space, three of them are related to the angles for the rotational position and three of which are related to the translational movement of the robot. In order to determine the rotational situation of the drone, body and ground coordinate systems are employed. The body coordinate system is fixed on the center of mass of the robot and the ground coordinate system is fixed on the ground. In the employed systems, the XY plane is aligned with the horizon and the Z axis is placed perpendicular to it and against the gravity of the earth. Also, the center of the body coordinate system is considered to be the center of the robot mass. The body coordinate system is in such a way that the X axis is aligned with the connecting axis of rotors 1 and 3 and the Y axis is aligned with the connecting axis of rotors 2 and 4. The Z axis is perpendicular to the X axis and the Y axis, and its direction is determined using the right-hand rule. In this drone, each rotor creates a certain torque around its rotation center, the used blades aren't similar and are divided into two groups, right-handed and left-handed, which rotate in opposite directions. The same torques of these rotors will lead to the stability of the flying robot around the center of gravity. To make transitional movement, the drone leans towards the rotor with a lower rotation speed and causes the thrust force to find a component in that direction. There is coupling in the structure of the robot, which means that the robot cannot have a translational movement without having a roll or screw movement, and this factor allows us to control six degrees of freedom degree with four inputs.

2.2. Derivation of equations

In this section, kinematic and dynamic equations are extracted with the following assumptions: [13]

- The center of the mass and the origin of the body coordinate system should be same.
- Its geometric and mass structure is assumed to be symmetrical
- The axes of the body coordinate system connected to the vertical flight are assumed to be parallel to the ground coordinate system axes, in this case the moment of inertia matrix is diagonal and makes the equations simpler.

Due to the rotation of each propeller, the thrust force equals to $F = K_t\Omega^2$ and the drag torque equals to $D = K_d\Omega^2$ be created which K_t is called thrust coefficient and K_d is called drag coefficient. Due to the same rotation speed and opposite direction of rotation, coaxial rotors neutralize each other's drag torque, but their thrust force is upward and they gather together. The resultant of the force and the torque around the center of mass resulting from the rotation of the rotors are according to equations (1) to (4):

$$\begin{aligned} \mathrm{T} &= T_1 + T_2 + T_3 + T_4 \\ &= K_t(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ &\quad + K'_t(\Omega_5^2 + \Omega_6^2) \end{aligned} \tag{1}$$

$$\tau_x = K_t l(\Omega_2^2 - \Omega_4^2) \tag{2}$$

$$\tau_x = K_t l(\Omega_3^2 - \Omega_1^2) \tag{3}$$

$$\tau_z = K_d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \tag{4}$$

In the above equations, T is called the total thrust force, τ_x is called roll torque, τ_y is called pitch torque and τ_z is yaw torque.

2.3. Derivation of dynamic equations

In Figure 2, the fixed coordinate system of the ground is shown with E and the coordinate system connected to the center of the robot mass is shown with B. The rotational transformation matrix of the body coordinate system compared to the inertial coordinate system by considering three consecutive cycles in terms of Euler angles with the rotation order of $\varphi \to \theta \to \psi$ is as follows:

$$\begin{aligned} R(\psi,\theta,\varphi) &= R(x,\varphi).R(y,\theta).R(z,\psi) \\ &= \begin{bmatrix} c\psi c\theta & s\psi c\varphi + c\psi s\theta s\varphi & s\psi s\varphi - c\psi s\theta c\varphi \\ -s\psi c\theta & c\psi c\varphi - s\psi s\theta s\varphi & c\psi s\varphi + s\psi s\theta c\varphi \\ -s\theta & -c\theta s\varphi & c\theta c\varphi \end{bmatrix} \end{aligned} \tag{5}$$

In above matrix, *c* and *s* are respectively abbreviations of *cos* and *sin*. The above rotation matrix is orthogonal. Also, the angular speed of the object in the body coordinate system can be obtained using equation (6) in terms of the rate of Euler angles [14].

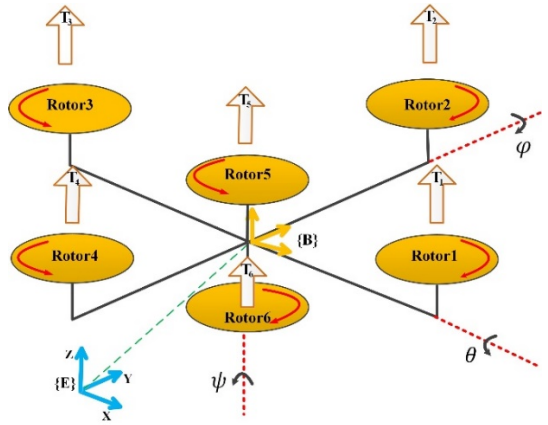

Figure 2. The six-blade robot configuration with roll-pitch-yaw Euler angles

$$\omega^B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\varphi & cos\theta sin\varphi \\ 0 & -sin\varphi & cos\theta cos\varphi \end{bmatrix} \times \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (6)$$

Translational motion equations based on Newton's laws in the ground coordinate system are according to equation (7) that k_{ft} is air coefficient.

$$m_t . a^E = T^E_{ext}$$
$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \frac{1}{m_t} \begin{bmatrix} Tsin\,\theta - k_{ft}\dot{X}^2 \\ -Tsin\,\psi cos\,\theta - k_{ft}\dot{Y}^2 \\ Tcos\,\theta cos\,\varphi - m_t\, g - k_{ft}\dot{Z}^2 \end{bmatrix} \quad (7)$$

The external forces in the above equation include the weight force, the thrust force of the propellers and the aerodynamic forces acting on the drone body.

According to Newton-Euler laws, the equation of angular motion for a system consisting of several connected moving masses is according to equation (8).

$$\Sigma M_G = \sum_i^4 \left(\dot{H}_i + R_{\frac{Oi}{G}} \times m_i a_o \right) = \sum_i^4 \left(\frac{\delta H_i}{\delta t} + \omega_i^B \times H_i + R_{\frac{Oi}{G}} \times m_i a_{oi} \right) \quad (8)$$

In above equation, G point is the center of mass of the system and O_i points are also assumed as the center of mass of each of the moving objects. H is the angular momentum vector of each object around its center of mass, and $R_{\frac{Oi}{G}}$ is the distance from the center of mass of each object to the center of mass of the entire system. a_{oi} is the acceleration of the center of mass of each moving object. The left side of equation (8) is the resultant of external torques around the center of mass of the system. These torques include equations (2) to (4) and are aerodynamic torques acting to the whole system. The right side of that is the angular momentum changes of each component around the center of mass of the system. By calculating each term of equation (8) in the body coordinate system, physical device, the final equations are obtained according to the equation (9).

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} ((I_y - I_z)qr - J_z q\Omega + \tau_x - k_f p)/I_x \\ ((I_z - I_x)pr - J_z p\Omega + \tau_y - k_f q)/I_y \\ ((I_x - I_y)qp - J_z\dot{\Omega} + \tau_z - k_f r)/I_z \end{bmatrix} \quad (9)$$

In the above relationship, k_f is the coefficient of air friction against the rotational movement and $\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ is the algebraic sum of the speed of the rotors and $\dot{\Omega}$ is the algebraic sum of the acceleration of the rotors. $J_z q\Omega$ and $J_z p\Omega$ are the gyroscopic moments of the rotors.

Assuming the Euler angles to be small, it can be seen from equation (6) that the angular speed of the drone in the body coordinate system is equal to the rate of Euler angles [15]. Also, aerodynamic effects have been ignored in the equations. Therefore, according to the proposed assumptions, the dynamic equations of the robot are obtained according to the following equations.

$$\begin{cases} m\ddot{X} = u_1 \sin\theta & (10) \\ m\ddot{Y} = -u_1 \sin\varphi \cos\theta & (11) \\ m\ddot{Z} = u_1 \cos\varphi \cos\theta - mg & (12) \\ I_x\ddot{\Phi} = (I_y - I_z)\dot{\Psi}\dot{\theta} - J_z\dot{\theta}\Omega + u_2 & (13) \\ I_y\ddot{\theta} = (I_z - I_x)B_1\dot{\Psi}\dot{\phi} + J_z\dot{\phi}\Omega + u_3 & (14) \\ I_z\ddot{\Psi} = (I_x - I_y)\dot{\phi}\dot{\theta} - J_z\dot{\Omega} + u_4 & (15) \end{cases}$$

In the above equations, u_1 to u_4 are control inputs and are defined as $[u_1, u_2, u_3, u_4] = [T, \varphi, \theta, \psi]$. Also, in order to avoid the singularity, it is assumed that the yaw and pitch angles are in the $(-\frac{\pi}{2}, \frac{\pi}{2})$ interval.

3. Designing Controller

The backstepping method was first introduced in 1990 as a recursive Lyapunov method. The reason behind the backstepping name for this method comes from the fact that during the design process, the designer goes back one step from the scalar equation that is the farthest from the control input (in terms of the number of integration) to this control input, and repeated as a recursive process. Due to the underexcitation of the system in this strategy, a control loop for translational dynamics and a control loop for rotational dynamics are designed with a sequential structure. Figure 3 shows the two-stage cascade control diagram. The translational control loop will generate the u_1 input and desired angles φ_d and θ_d such that the X_d, Y_d and Z_d route are tracked by the output. The rotational control loop which is responsible for the stabilization of the drone, generates control signals u_2 to u_4 for converging the drone angles to desired values of φ_d, θ_d and ψ_d. The closed loop stability of such structure has been shown in [16].

In this section, the backstepping method to control the position of the six-blade robot is presented. In the backstepping method, by considering the some of the system situations as virtual input a recursive controller is designed and finally a real control input is used for

stabling the whole system. The block diagram of the drone control system has been shown in Figure 3.

By defining state vector $X^T = [\varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}, X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}]^T$, the obtained equations for system is rewritten as equation (16), by transforming equations (10) to (15) in state space.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ (1/I_x)(A_1 x_4 x_6 - J_Z x_4 \Omega + u_2) \\ x_4 \\ (1/I_y)(B_1 x_2 x_6 + J_Z x_2 \Omega + u_3) \\ x_6 \\ (1/I_z)(C_1 x_2 x_4 - J_Z \dot{\Omega} + u_4) \\ x_8 \\ (1/m)(u_1 \sin x_3) \\ x_{10} \\ (1/m)(-u_1 \sin x_1 \cos x_3) \\ x_{12} \\ (1/m)(u_1 \cos x_1 \cos x_3 - mg) \end{bmatrix} \quad (16)$$

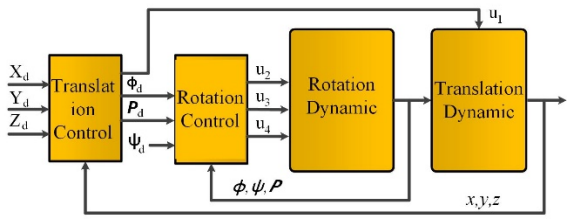

Figure 3. Two-stage cascade control block diagram

Considering the dynamic underactuated of the vertical flying robot, in the equation (16), $sin\, x_3$ and $\sin x_1 \cos x_3$ are defined as new input and they are illustrated by u_x and u_y respectively. By defining new inputs as above, equation (16) will be completely simulated. The controller is designed with this conditions.

In the first step of designing the controller by using the backstepping method, the tracing error is defined as follows:

$$z_1 = x_{1d} - x_1 \quad (17)$$

Using the Lyapunov theory and choosing the Lyapunov function and its time derivative as follows:

$$v(z_1) = \frac{1}{2} z_1^2 \quad (18)$$

$$\dot{v}(z_1) = z_1(\dot{x}_{1d} - x_2) \quad (19)$$

And considering the x_2 state as a virtual input and designing it as follows:

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \quad (20)$$

The time derivative of the Lyapunov function is obtained as follows:

$$\dot{v}(z_1) = -\alpha_1 {z_1}^2 \quad (21)$$

The derivative of the Lyapunov function will be negative and z_1 stabilization will be guaranteed by choosing positive value for α_1.

The x_2 state variable in the equation (20) is actually a desired value (x_2^d) if x_2 state of the system is equal with that, x_1 will be converged to desired value x_{1d}.

In the second step, new error variable is introduced as follows:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \quad (22)$$

By considering the new Lyapunov function as follows:

$$v(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2) \quad (23)$$

Its time derivative will be:

$$\dot{v}(z_1) = -\alpha_1 {z_1}^2 + z_2\left(\frac{A_1}{I_x} x_4 x_6 - \frac{J_Z}{I_x} x_4 \Omega + \frac{u_2}{I_x} - \ddot{x}_{1d} + \alpha_1(z_2 + \alpha_1 z_1) - z_1\right) \quad (24)$$

By designing control input as:

$$u_2 = I_x\left(-\frac{A_1}{I_x} x_4 x_6 + \frac{J_Z}{I_x} x_4 \Omega + \ddot{x}_{1d} + z_1 - \alpha_1^2 z_1 - \alpha_2 z_2\right) \quad (25)$$

Choosing negative ($\alpha_2 > \alpha_1$) the Lyapunov function derivative is obtained as:

$$\dot{v}(z_1, z_2) = -\alpha_1 {z_1}^2 - (\alpha_2 - \alpha_1){z_2}^2 \leq 0 \quad (26)$$

So, tending new states (z_1, z_2) to zero and converging state x_2 to desired value (equation 20) will be guaranteed. Consequently state variable x_1 converges to x_{1d} peripherally.

Following the same process, other control inputs are obtained as follows:

$$u_3 = I_y\left(-\frac{B_1}{I_y} x_2 x_6 - \frac{J_Z}{I_y} x_2 \Omega + \ddot{x}_{3d} + z_3 - \alpha_3^2 z_3 - \alpha_4 z_4\right) \quad (27)$$

$$u_4 = I_z\left(-\frac{C_1}{I_z} x_2 x_4 + \frac{J_Z}{I_z} \dot{\Omega} + \ddot{x}_{5d} + z_5 - \alpha_5^2 z_5 - \alpha_6 z_6\right) \quad (28)$$

$$u_1 = \frac{m}{\cos x_1 \cos x_3}(-g + \ddot{x}_{11d} + z_{11} - \alpha_{11}^2 z_{11} - \alpha_{12} z_{12}) \quad (29)$$

$$u_x = \frac{m}{u_1}(\ddot{x}_{7d} + z_7 - \alpha_7^2 z_7 - \alpha_8 z_8) \quad (30)$$

$$u_y = \frac{m}{u_1}(\ddot{x}_{9d} + z_9 - \alpha_9^2 z_9 - \alpha_{10} z_{10}) \quad (31)$$

In the above equations:

$$\begin{cases} z_3 = x_{3d} - x_3 \\ z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3 \\ z_5 = x_{5d} - x_5 \\ z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5 \\ z_7 = x_{7d} - x_7 \\ z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7 \\ z_9 = x_{9d} - x_9 \\ z_{10} = x_{10} - \dot{x}_{9d} - \alpha_9 z_9 \\ z_{11} = x_{11d} - x_{11} \\ z_{12} = x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11} \end{cases} \quad (32)$$

Two-stage control idea is used to implement the designed control system by backstepping method (Figure 3). So, in the translational control loop u_1 signal is obtained from equation (29) and desired angels x_{1d} and x_{3d} are obtained as follows:

$$x_{3d} = \sin^{-1}(u_x) = \sin^{-1}\left(\frac{m}{u_1}(\ddot{x}_{7d} + z_7 - \alpha_7^2 z_7 - \alpha_8 z_8)\right) \quad (33)$$

$$x_{1d} = \sin^{-1}\left(\frac{u_y}{\cos x_{3d}}\right) = \sin^{-1}\left(-\frac{m}{u_1 \cos x_{3d}}(\ddot{x}_{7d} + z_7 - \alpha_7^2 z_7 - \alpha_8 z_8)\right) \quad (34)$$

In the rotational controller, the control signals u_2 to u_4 are calculated from equations (25, 27 and 28).

4. Simulation Result

In this part, several simulations have been run in the MATLAB software to evaluate the backstepping control system performance for the six-blade drone and the results are discussed and analyzed. In these analysis, in first, system stable capability is investigated. Then, capability of the desired route tracing is analyzed and the controller performance in the interference is investigated too.

4.1. System stability

In order to check the stability of the system, by considering the constant reference input as $X_d = Y_d = Z_d = 1$; $\psi_d = \frac{\pi}{4}$; According to Figure 6 and 7, it can be seen that the system output tends to reference input without the steady-state error and overshoot. By considering that the system is underexcited, the system controller does not affect the components φ and θ directly but because of the asymptotic stability of the whole system, these parameters also tend to zero through time. Figure 8 shows the flight route of the drone from the coordinate origin to the reference point.

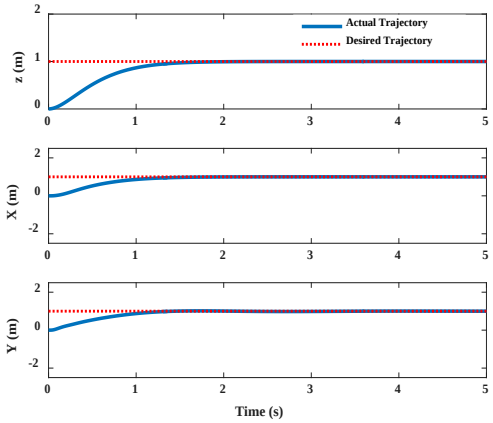

Figure 4. Closed loop system response to unit step input

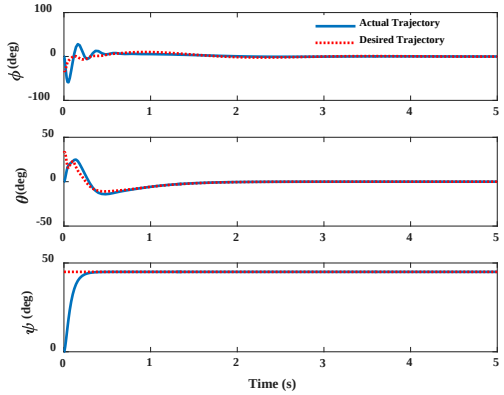

Figure 5. Closed loop system response to unit step input

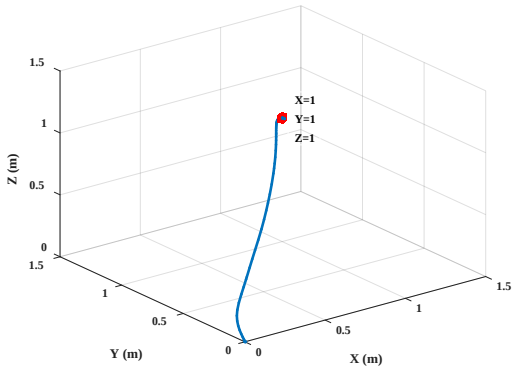

Figure 6. The robot flight path from the origin coordinate to the fixed point (1,1,1)

4.2. Tracing the optimal route of the square route

In this part, tracing of the drone from the desired route is evaluated and analyzed. In the first mission, the vertical flight route includes taking off from the ground, floating in the air, crossing four points in the square corners at a fixed altitude and landing on the primary point. To perform this mission, the drone should go from the origin ($X_d = 0$, $Y_d = 0$, $Z_d = 0$) to the point (0, 0, 2) and then go to the points (0, -2, 2), (-2, -2, 2), (-2, 0, 2) and (0, 0, 2) respectively and finally return to the origin (0, 0, 0). After reaching each point, the drone remains floating there for a while and then flies to the next point. By considering that the controller has high-order derivation, so, a smooth reference route is created as

follows to go to the desired points so that the drone can reach the desired points by tracing these routes [17].

$$\sigma_r(t) = \begin{cases} 0 & t < t_{\sigma 1} \\ \sigma_1 + \sigma_2(1+\sigma_3(t-t_{\sigma 1}))e^{-\sigma_3(t-t_{\sigma 1})} & t_{\sigma 1} \le t \le t_{\sigma 2} \\ \sigma_0 - \sigma_2(1+\sigma_4(t-t_{\sigma 2}))e^{-\sigma_4(t-t_{\sigma 2})} & t > t_{\sigma 2} \end{cases} \quad (35)$$

In the above equation $\sigma_r = x_r, y_r, z_r$ is the reference route parametric equation and other coefficients of this route are mentioned in Table 1.

The optimal value of the yaw angle is considered $\psi_d = \frac{\pi}{3}$ and the initial system conditions are zero. Vertical flight route in this situation is displayed in Figure 7. It should be noted that the square route is not travelled continuously rather drone after achieving the corner points, stops there for a while and then flies to the next point. The simulation results in Figure 8 and 9 imply that system output traces the reference route acceptably. Figure 8 shows that the drone after achieving each point. Rests about 10s and then flies to the next point. Y parameter changes in 21s and 54s only occurred because of angle φ change and X parameter change in 37s and 71s occurred because of θ angle change. The desired route tracing error is displayed in Figure 10.

Table 1. Reference path parameters

value	parameter	value	parameter	value	parameter
-2	x_1	-2	y_1	2	z_1
2	x_2	2	y_2	-2	z_2
0.9	x_3	1	y_3	1.5	z_3
1.4	x_4	1	y_4	1.5	z_4
37	t_{x1}	21	t_{y1}	10	t_{z1}
71	t_{x2}	54	t_{y2}	80	t_{z2}

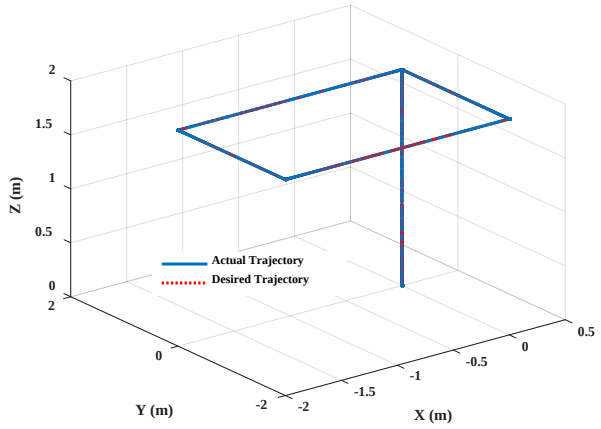

Figure 7. Drone path in tracing the desired path

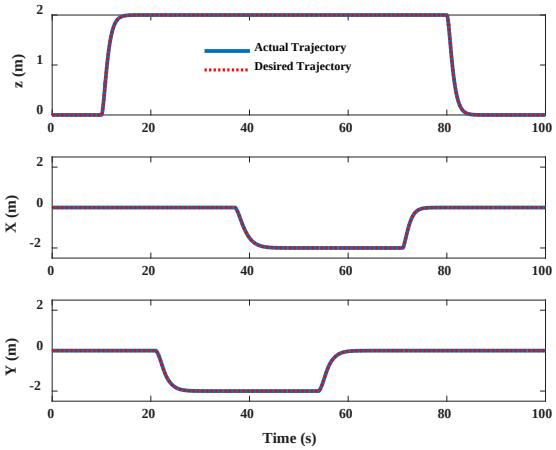

Figure 8. Drone flight route in tracing the position of the desired route

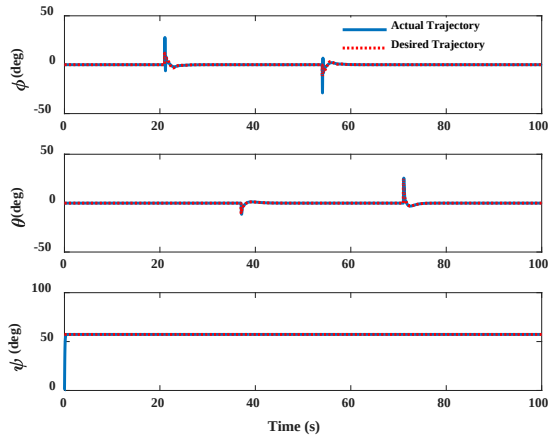

Figure 9. The response of Euler angles of the robot in tracing the desired path

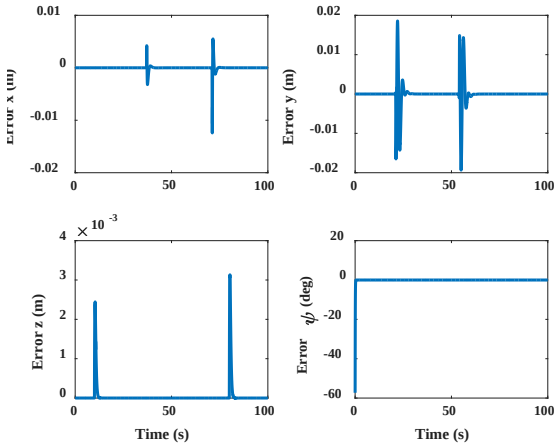

Figure 10. Display of desired route tracing error

In the second mission of the drone, it includes tracing continues route as follows:

$$x_d = 2\cos\left(\frac{t}{2}\right), y_d = 2\sin(t), z_d = 2, \psi_d = \frac{\pi}{4} \quad (36)$$

The initial conditions are considered zero. The robot control system simulation results in tracing the desired route are prepared in Figure 11. This diagram shows that the drone movement route matches the desired route for

a short time and not only the system output tends to the desired values, but also
Euler angles remain low. Figure 12 and 13 show the system output, the drone position desired values and the robot rotation angles. The smallness of twist and roll angles in this diagram confirms the assumption of simplifying the motion equations and the accuracy of the extracted dynamic equations. The two dimensional view of the robot's route in tracing the reference route and the system tracing error diagram are presented in Figure 14 and 15 respectively.

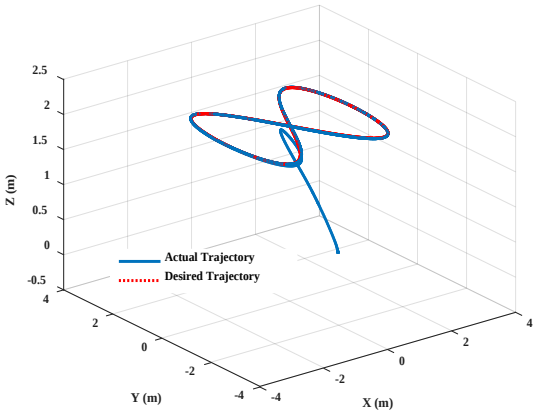

Figure 11. Drone flight route in tracing the position of the desired route

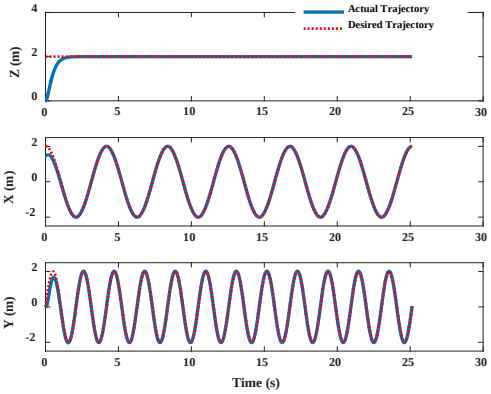

Figure 12. Drone performance in tracing the position of the desired path

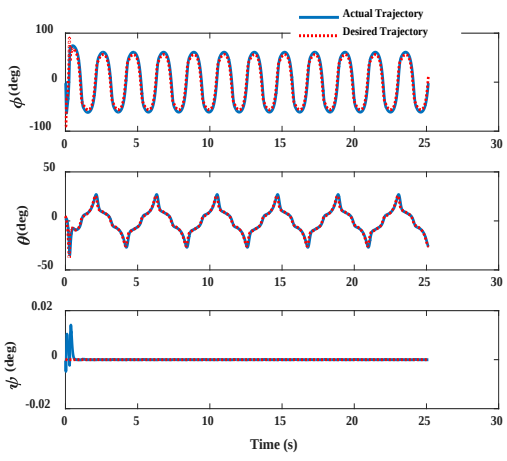

Figure 13. Euler's angles response in tracing desired path

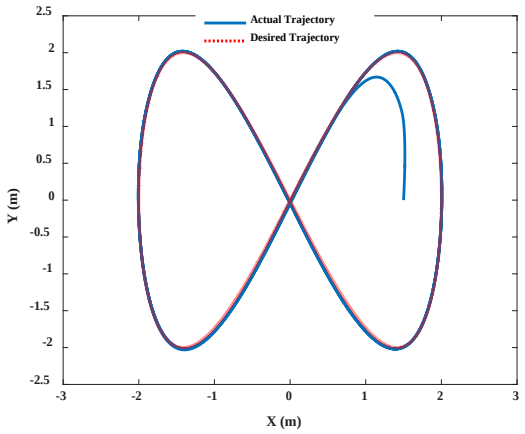

Figure 14. Two-dimensional trajectory of the robot in tracing desired path

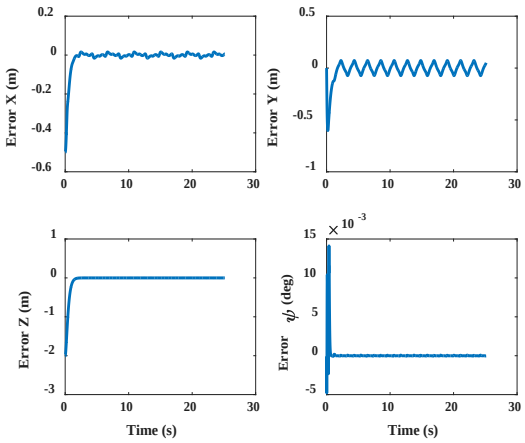

Figure 15. Drone error in tracing desired path

5. Conclusion

In this research, the modeling of a six-blade vertical flying drone with a new structure was implemented by Newton-Euler method. The structure of the drone is so that by adding two coaxial propellers in the center of the drone, the cargo capacity increased and simultaneously its dynamic and control method maintained. So, the six-blade vertical flying drone has a more applications than the normal quadrators. According to the nonlinear dynamics of the robot, nonlinear backstepping controller was designed for tracing the desired route. The control system stability was evaluated by obtaining the unit step response. Then, in order to check the performance of the designed controller in tracing the desired route, two various routes with different frequencies were applied to the drone control system of the drone and the robot behavior in tracing the applied routes was simulated. According to the simulation results, the performance of the presented controller was evaluated favorably.

6. References

[1] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, pp. 315-378, 2012.

[2] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four rotor mini-rotorcraft," in *European Control Conference (ECC), 2003*, 2003, pp. 3123-3128.

[3] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, pp. 1023-1036, 2011.

[4] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, pp. 419-428, 2009.

[5] H. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," in *Mechatronics, 2009. ICM 2009. IEEE International Conference on*, 2009, pp. 1-6.

[6] H. Ramirez-Rodriguez, V. Parra-Vega, A. Sanchez-Orta, and O. Garcia-Salazar, "Robust backstepping control based on integral sliding modes for tracking of quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 73, pp. 51-66, 2014.

[7] C. Nicol, C. Macnab, and A. Ramirez-Serrano, "Robust adaptive control of a quadrotor helicopter," *Mechatronics*, vol. 21, pp. 927-938, 2011.

[8] S. Bouabdallah, R. Siegwart, and G. Caprari, "Design and control of an indoor coaxial helicopter," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2930-2935.

[9] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on control systems technology*, vol. 12, pp. 510-516, 2004.

[10] P. Casau, R. G. Sanfelice, R. Cunha, D. Cabecinhas, and C. Silvestre, "Robust global trajectory tracking for a class of underactuated vehicles," *Automatica*, vol. 58, pp. 90-98, 2015.

[11] C.-S. Yang, Z. Yang, D.-Z. Xu, and L. Ge, "Trajectory tracking control for novel six-rotor aircraft," *Systems Engineering and Electronics*, vol. 34, pp. 2098-2105, 2012.

[12] A. Alaimo, V. Artale, C. L. R. Milazzo, and A. Ricciardello, "PID controller applied to hexacopter flight," *Journal of Intelligent & Robotic Systems*, vol. 73, pp. 261-270, 2014.

[13] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," *MSc Theses*, 2008.

[14] J. H. Ginsberg, *Advanced engineering dynamics*: Cambridge University Press, 1998.

[15] D. Lara, G. Romero, A. Sanchez, R. Lozano, and A. Guerrero, "Robustness margin for attitude control of a four rotor mini-rotorcraft: Case of study," *Mechatronics*, vol. 20, pp. 143-152, 2010.

[16] S. Bertrand, N. Guénard, T. Hamel, H. Piet-Lahanier, and L. Eck, "A hierarchical controller for miniature VTOL UAVs: design and stability analysis using singular perturbation theory," *Control Engineering Practice*, vol. 19, pp. 1099-1108, 2011.

[17] Tofigh M A, Mahjoob M, Ayati M. Modeling and nonlinear tracking control of a novel multi-rotor UAV. Modares Mechanical Engineering 2015; 15 (8) :281-290