Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2017, 4 (2): 135-142

Research Article **ISSN: 2394 - 658X**

Trajectory Tracking and Hurdle Avoidance for an Autonomous Quadrotor UAV

Khayyam Masood[1] and Muhammad Aadil Khan[2]

[1]Northwestern Polytechnical University, Xi'an, China
[2]Department of Aeronautics & Astronautics, Institute of Space Technology, Islamabad, Pakistan
m.aadil.k94@gmail.com

ABSTRACT

The importance of autonomous flight has increased with time and it has become impertinent for Unmanned Aerial Vehicles to be able to track a set of given waypoints to reach its desired destination. Similarly, quadrotors are small UAVs used for various domestic missions; moreover, another important aspect of autonomous quadrotors is their ability to avoid domestic hurdles. Numerous hurdles can appear in quadrotor's navigation path and it is necessary for it to avoid these hurdles to ensure completeness in its mission without failure. The mathematical model of quadrotor is modelled in body-fixed frame and inertial frame. Accurately modeling the dynamics of the quadrotor allows us to check quadrotor in various situations without the need to build a prototype and test in real life. The quadrotor 6 DOF is validated through observations in inputs and outputs of mathematical model whether they conform to realistic values. The control architecture employed is the cascade control in which two loops are used – an inner loop and an outer loop – to control the quadrotor with the aid of Proportional-Derivative (PD) controllers. The aim of this paper is to design an autonomous quadrotor capable of tracking a trajectory and a hurdle avoidance algorithm that could enable the quadrotor to achieve a completely autonomous flight. Ingenuity is captured in hurdle avoidance algorithm by using trigonometric relations for new trajectory generation. The platform used for this work was the MATLAB/Simulink environment that allowed us to develop the algorithm and integrate them with the 6 DOF mathematical model of the quadrotor to achieve a completely autonomous UAV. Simulation results showed that the quadrotor accurately followed its desired trajectory and effectively avoided hurdles in its path.

Key words: Cascade Control, Trajectory tracking, Hurdle Detection, Hurdle Avoidance, Autonomous

---

INTRODUCTION

Starting out as full-scale aerial vehicles, quadrotors were capable of carrying humans when they first came into being. However, it was soon realized that large quadrotors are mechanically inefficient compared to modern day helicopters. Hence, these vehicles were reduced in size and used as Unmanned Aerial Vehicles (UAVs). The biggest attraction of quadrotors is their mechanical simplicity. It possesses a basic cross or plus configuration structure that has four motors at the four ends. These motors are connected to four fixed-pitch propellers that rotate to provide thrust in the vertical direction – which basically generates lift. For movement in lateral direction, the rotational rate of the propellers is varied and motion can be achieved in all directions.

Quadrotor has six degrees of freedom but only four inputs which makes it under actuated. For under actuated vehicles, there is coupling between different states. Controlling such a vehicle requires the use of control systems. Furthermore, unlike an aircraft, quadrotors are inherently unstable so they need a control system to give them stability. Feedback linearization and slide-mode control is used for quadrotor [1] which is a method in which the non-linear system is linearized and simple feedback is used to control the system. Another approach uses robust adaptive control [2] in which the controller adapts to the various changing parameters provided that the disturbances are inside a predefined set. Modified PID control [3] and slide-mode control of under actuated systems [4] are some other methods used for tracking the trajectory of the quadrotor.

H-infinity loop shaping is also employed in the safe landing of a quadrotor [5] with propeller failure. The aim of this method is to achieve stability and good performance despite the fact that there are differences in the designed plant and the actual plant. The response of the plant is checked in the frequency domain and the loop obtained is made more robust using optimization. Research done in [6] uses Piecewise Polynomials to generate a dynamically optimal trajectory whereby the quadrotor follows a series of waypoints that are predefined. This allows the quadrotor to reach the final destination by reaching intermediate destinations, that when combined, will take the quadrotor to its target destination. An improvement on this method uses an unconstrained quadratic program [7] that jointly optimizes the polynomial path segments and is able to handle high order polynomials.

Dynamic programming [8] breaks down the complete problem into various sub problems. Researchers than try to find the solution of the sub problems and once it is obtained, the solution is stored to be used later. When the solution of all the problems is available, it is combined to get the solution of the complete problem. For trajectory generation, this is efficiently applied by breaking the complete trajectory into smaller ones and dealing with them separately. Rodrgiues theorem and modified Rodrigues parameters [9] are used to get global stability conclusions that are used to design a trajectory tracking controller. Work done in [10] gives two non-linear techniques to control the trajectory: back stepping approach and sliding-mode technique. While back stepping gives a stable starting point for controller design “backing out” to non-stable situations, the sliding-mode technique changes the dynamics of the system by giving a discontinuous control signal that slides the system into its normal behavior.

Low-cost ultrasonic sensors [11] is a simple approach used for hurdle avoidance. It has two parts: obstacle detection and obstacle avoidance. Redundant sensors are used to increase the detection angle of the quadrotor. Once the obstacle is detected, the quadrotor measures the distance between itself and the obstacle. Using a predefined range, the quadrotor can categorize this distance in one of the three zones: safe region, close region and danger region represented by green, yellow and red respectively. Depending on the zone the distance comes in, the quadrotor takes appropriate actions to avoid the obstacle.

Vision-based obstacle avoidance [12] is used where optical flow velocities are used to determine the required angles for the quadrotor to avoid the obstacle. Another visual-aid method spatially decomposes instantaneous optical flow to get proximity information [13]. Push and Rapidly Exploring Random Tree (RRT) algorithms [14] are used for obstacle avoidance. A technique measures the distance of the obstacle from the quadrotor and computes the position and orientation by using spatial derivatives of optical flow [15].

MATHEMATICAL MODELING

The complete 6 DOF mathematical model of the quadrotor used for this paper is given in this section.

Assumptions

- The structure of the quadrotor is symmetric and rigid.
- The propellers of the quadrotor are rigid.
- Mass of the quadrotor is constant.
- Blade flapping effects are neglected.
- Gravitational model is taken that of a flat, non-rotating earth.

Reference frames

For this paper, the body frame and the inertial frame were used as shown in Fig.2.

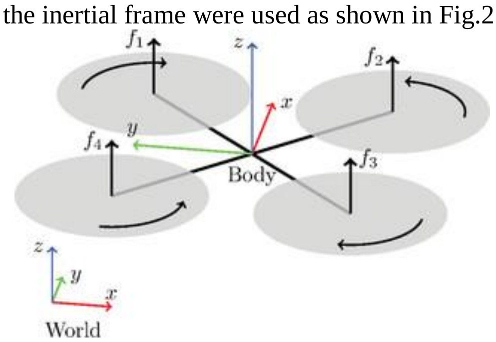


Fig. 1 Body Frame and Inertial Frame1 [1 Courtesy: https://goo.gl/9sLPu5]

Kinematics

The motion of the quadrotor excluding the forces and moments on it are given in this section. For orientation, the Angular rates in terms of the Euler angle rates by equation (1).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{1}$$

The inverse of the above equation is

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

Integrating the Euler angle rates in equation (2) will give the Euler angles in the Inertial frame. Velocity and position of the quadrotor are linked to each other via equation (3).

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} = [R_B^I] \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3}$$

Dynamic Relationships

Forces and moments acting on the quadrotor are governed by the equations below.

$$F = \frac{d}{dt}(mV)|_E \tag{4}$$

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{bmatrix} = \begin{bmatrix} m(\dot{u} + wq - rv) \\ m(\dot{v} + ru - pw) \\ m(\dot{w} + pv - qu) \end{bmatrix} \tag{5}$$

$$M = \frac{d}{dt}(H)|_E \tag{6}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})rp \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq \end{bmatrix} \tag{7}$$

It can be seen below that the inertia matrix of the quadrotor has zero off-diagonal elements which means that the structure of the quadrotor is symmetric.

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{8}$$

Complete Non-Linear Model

A complete 6 DOF quadrotor has twelve states and four inputs. The state vector is shown below.

$$X = [x \quad y \quad z \quad u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi]^T \tag{9}$$

And the input vector is given by

$$U = [u_1\, u_2\, u_3\, u_4]^T \tag{10}$$

The inputs to the quadrotor are defined by

$$u_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{11}$$

$$u_2 = bl(-\Omega_2^2 + \Omega_4^2) \tag{12}$$

$$u_3 = bl(\Omega_1^2 - \Omega_3^2) \tag{13}$$

$$u_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{14}$$

Atmospheric Model

The COESA Atmospheric Model has been used for the quadrotor because the quadrotor is not required to gain a very high altitude. Furthermore, below 32,000 feet, this model is similar to the Standard Atmosphere of ICAO.

CONTROL ARCHITECTURE

Being an under actuated vehicle, it is necessary that controllers are implemented to achieve the control of the quadrotor. For this purpose, Cascade control is used. The control architecture has two loops: the inner loop deals with the attitude control and the outer loop deals with the position and height control of the quadrotor.

Having two loops gives the advantage of isolating the guidance loop from the rotor dynamics. The inner loop is faster and deals with the rotor dynamics while the outer loop sees the inner loop and the quadrotor dynamics as a complete system and adjusts accordingly.

Position and Attitude Control

To ensure that the quadrotor reaches the desired position exactly, it is impertinent that the quadrotor makes the attitude changes accurately so that the position changes are correctly achieved. Based on the control architecture, this is achieved by implementing the position controllers in series with the attitude controllers. With the configuration, the set point of the attitude controller would be the output of the position controller allowing the quadrotor to achieve the exact position precisely.

Proportional-Derivative Controller

The PD controller is a variation of the PID controller with the absence of the Integral controller. One thing to note here is that the Derivative controller is connected directly to the state term instead of the error term. Due to the sudden changes in the set point value, the error term also changes suddenly. This results in spikes appearing in the output. To avoid this, the derivative controller is connected directly to the state term that eliminates the spikes in the output response. This is known as Derivative Kick.

Response with PD Controller

To check the response of the quadrotor, the Unit Step input was used. After tuning the controller intuitively, the values selected are shown in Table 2. We can see that the values used for the Derivative controllers are the same for all three angles with a slight change in the values used for the Proportional controllers.

Simulation Results

Firstly, the Unit-step response of the attitude of the quadrotor is checked and then the positions of the quadrotor are checked. It is evident from the results that the attitude angles have a fast transient response with no overshoot and a quick settling time, as is needed for the inner loop of the control architecture. The unit-step response of X and Y position is slightly slower but equally accurate.

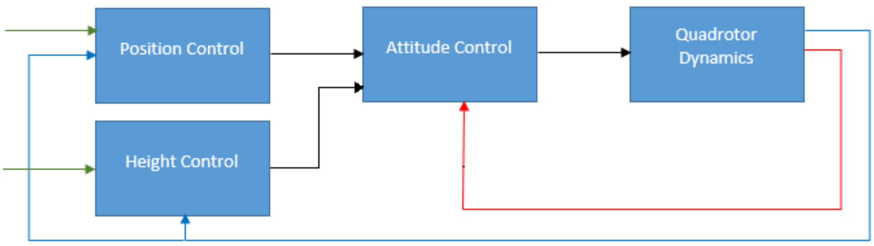


Fig. 2 Control Architecture for the Quadrotor System

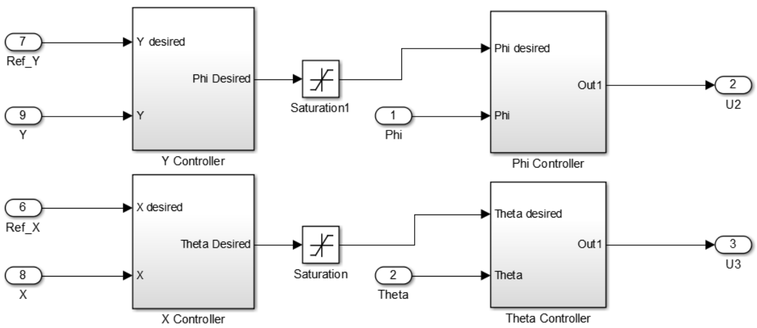


Fig. 3 Position and Attitude Controllers in Series

Table – 1 Gain Values of PD Controllers

	K_P	K_d
Pitch Angle	1.2	0.4
Roll Angle	0.8	0.4
Yaw Angle	1	0.4

TRAJECTORY DESIGN

In autonomous flight, a quadrotor is able to accomplish its task without any human intervention. The path that the quadrotor has to follow can be provided by giving it a set of waypoints defining the path. For this paper, it is assumed that a GPS installed on the quadrotor accurately tells the current position of the quadrotor. Waypoints are given to the quadrotor as inputs which correspond to a square trajectory. The quadrotor is expected to follow the desired trajectory by making autonomously making the necessary adjustments to its attitude throughout the flight.

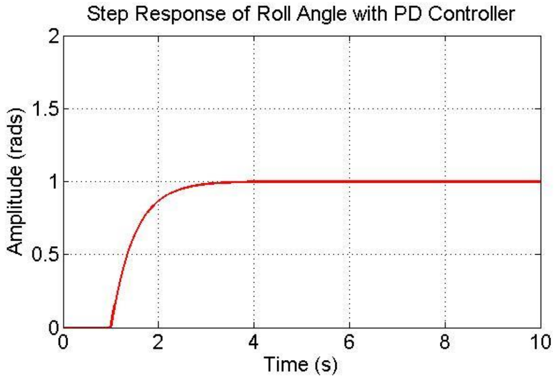


Fig. 4 Step Response of Roll Angle with PD Controller

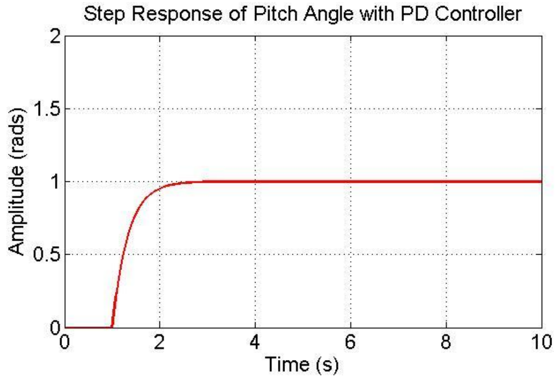


Fig. 5 Step Response of Pitch Angle with PD Controller

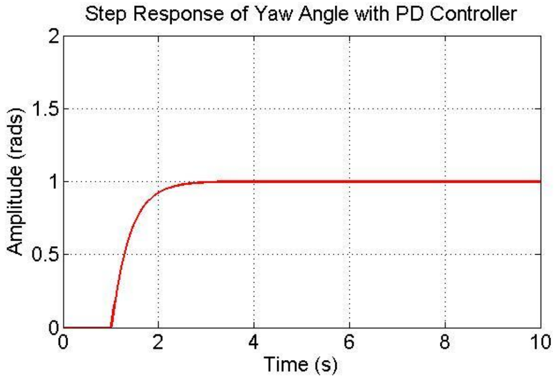


Fig. 6 Step Response of Yaw Angle with PD Controller

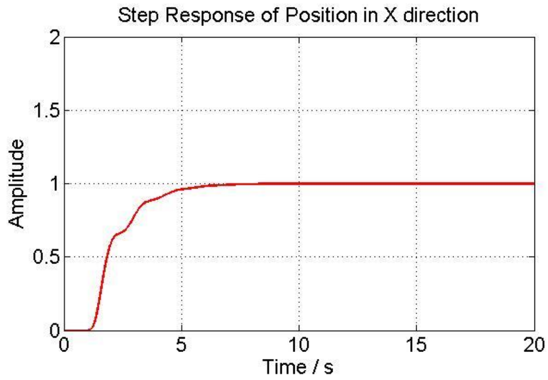


Fig. 7 Step Response of Position in X direction

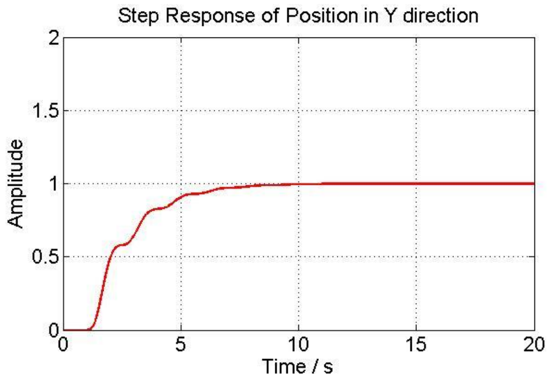
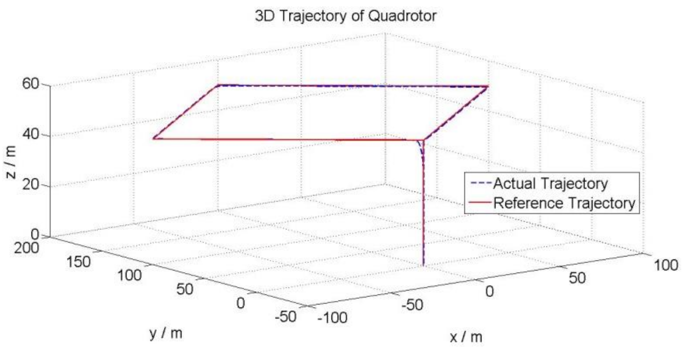


Fig. 8 Step Response of Position in Y direction

Simulation Results



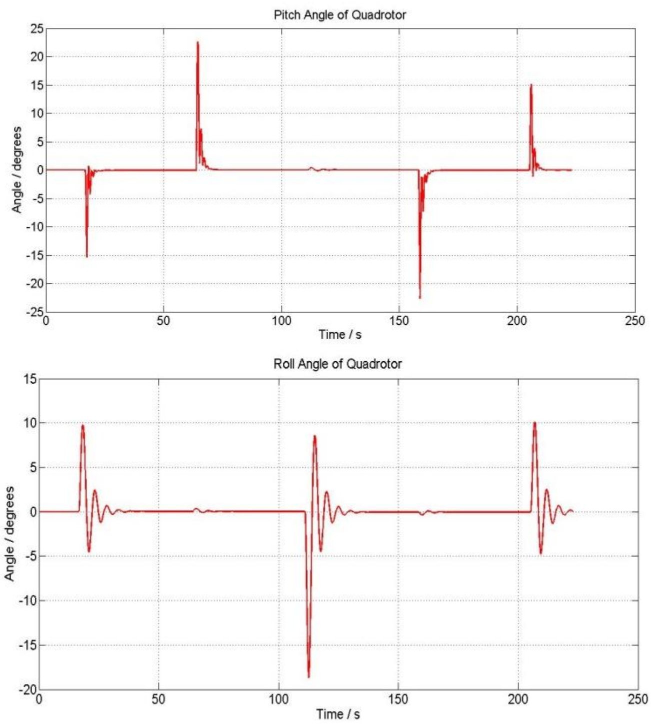


Fig. 9 Square Trajectory Tracking of Quadrotor with Roll and Pitch Angles

HURDLE DETECTION AND AVOIDANCE

Once the quadrotor is completely autonomous, it is necessary to ensure that it is capable of avoiding any kind of hurdles in its path. Otherwise, it would not be useful. Various techniques have been developed and implemented to achieve hurdle avoidance. The technique used in this paper is inspired from work done by Samir Bouabdallah *et al* [16] in which they mention various hurdle avoidance approaches using ultrasonic sensors. Of all the approaches, the most efficient is the triangular approach because there is less chance of losing the hurdle by going out of the sensing range of the ultrasonic sensor. Furthermore, another work done in [17] incorporates a Dynamic Collision Map (DCM) that uses the velocity and the heading of the quadrotor to avoid the hurdle. It does this by using a via point that results in an avoidance manoeuvre similar to the triangular approach mentioned previously.
For hurdle detection and avoidance, a few assumptions are taken.

Assumptions

The assumptions are taken to simplify the hurdle detection and avoidance algorithm.
- Quadrotor is assumed to have a camera and a proximity sensor installed on it.
- The quadrotor knows its exact position using a GPS installed on-board.

Hurdle Detection

The hurdle avoidance algorithm to detect the hurdles in the path of the quadrotor is given as:
- Coordinates of the vertices of the hurdle is used to calculate the gradients of the outer lines of the hurdle
- Y-intercepts of the outer lines of the hurdle are calculated to get the equation of the lines
- The checking distance for the quadrotor is defined. For this paper, it is taken to be 5 meters.
- Gradient of the current path of the quadrotor is calculated using the current and the previous position
- From the current coordinate, new x and y coordinates are calculated that are used to check the presence of the hurdle.

$$xn = xc \pm \frac{d}{\sqrt{m\_path^2 + 1}} \tag{16}$$

$$yn = m\_path(xn - xc) + yc \tag{17}$$

- Coordinates of the point 5 meters away is checked to see if it lies on the outer lines of the hurdle.

$$yn - (mh * xn) - ch < 3 \tag{18}$$

Once all the coordinates are inserted and the result is less than 3, this means that a hurdle is present on the path of the quadrotor. Now, a signal is generated to tell the quadrotor that a hurdle is present.

Hurdle Avoidance

To avoid the hurdle, the quadrotor uses a Three-step Maneuver:

1. Holding the signal.
2. Detecting the vertex of the hurdle that is closest to the new waypoints.
3. Generating a new waypoint to allow the quadrotor to route the hurdle.

Simulation Results

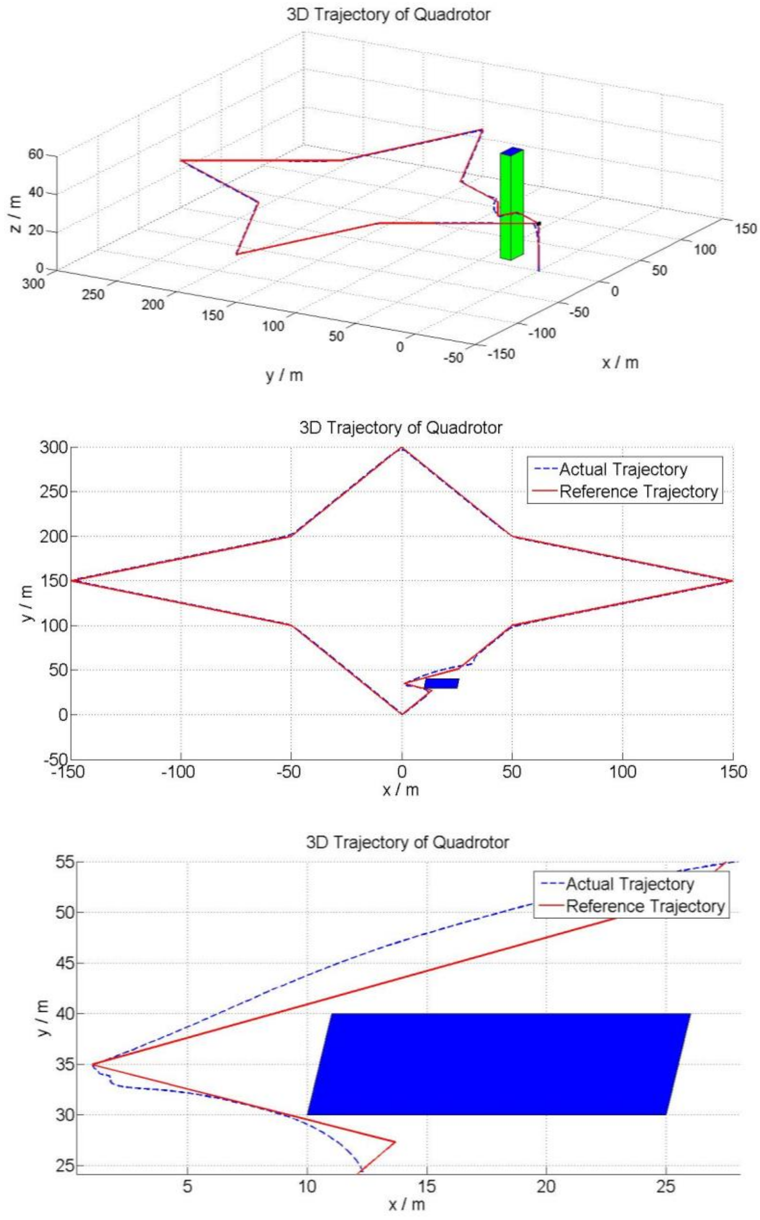


Fig. 10 Hurdle Avoidance plots of Quadrotor

CONCLUSION

In this paper, we used a 6 DOF mathematical model of a quadrotor with cascade control architecture and developed algorithms to make the quadrotor completely autonomous. The quadrotor was designed to follow a desired trajectory by giving it multiple waypoints, which it accurately followed to reach its destination. The results from the hurdle detection and avoidance algorithm also show that the algorithm works well in ideal conditions for an outdoor quadrotor UAV. The use of the triangular hurdle avoidance approach, which minimizes the chances of the quadrotor being in the dead zone, making it incapable of detecting the hurdle anymore, efficiently allows the quadrotor to route the hurdle in its path.

The work done in this paper gives strong basis for developing even more robust techniques for the autonomous flight of a quadrotor UAV. It can also be improved upon by simulating the algorithms in the presence of wind and other disturbances and developing a robust controller to ensure stability of the quadrotor.

REFERENCES

[1] D Lee, HJ Kim and S Sastry, Feedback linearization versus Adaptive Sliding Mode Control for a Quadrotor Helicopter, *International Journal of Control, Automation and Systems*, **2009**, 7(3), 419–428.
[2] C Nicol, CJB Macnab and A Ramirez-Serrano, Robust Adaptive Control of a Quadrotor Helicopter, *Mechatronics*, **2011**, 21(6), 927–938.
[3] MO Efe, Neural Network Assisted Computationally Simple Control of a Quadrotor UAV, *IEEE Transactions on Industrial Informatics*, **2011**, 7(2), 354–361.
[4] X Rong and U Ozguner, Sliding Mode Control of a Class of Underactuated Systems, *Automatica*, **2008**, 44(1), 233–241.
[5] A Lanzon, A Freddi and S Longhi, Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure, *Journal of Guidance, Control, and Dynamics*, **2014**, 37 (2), 580–591.
[6] D Mellinger, *Trajectory Generation and Control for Quadrotors*, Ph.D Dissertation, University of Pennsylvania, Pennsylvania, **2012**.
[7] C Richter, A Bry and N Roy, Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments, *International Symposium of Robotics Research (ISRR)*, Cambridge, India, **2013**, 1-16.
[8] I Palunko, R Fierro and P Cruz, Trajectory Generation for Swing-Free Maneuvers of a Quadrotor with Suspended Payload: A Dynamic Programming Approach, New Mexico, 1st ed., **2012**, 1-7.
[9] L Wang and H Jia, The Trajectory Tracking Problem of Quadrotor UAV: Global Stability Analysis and Control Design based on the Cascade Theory, *Asian Journal of Control*, **2014**, 16(2), 1-15.
[10] S Bouabdallah and R Siegwart, Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, **2015**, 2247–2252.
[11] N Gageik, T Muller and S Montenegro, Obstacle Detection and Collision Avoidance using Ultrasonic Distance Sensors for an Autonomous Quadrocopter, *Aerospace Information Technology*, Wurzburg, 1st ed., **2012**, 1-6.
[12] A Eresen, N İmamoğlu and M O Efe, Autonomous Quadrotor Flight with Vision-Based Obstacle Avoidance in Virtual Environment, *Expert Systems with Applications*, **2012**, 39(1), 894-905.
[13] J Conroy, G Gremillion, B Ranganathan and JS Humbert , Implementation of Wide-Field Integration of Optic Flow for a Autonomous Quadrotor Navigation, *Autonomous Robots*, **2009**, 27(3), 189-198.
[14] J Wagster, M Rose and H Yaralin, *Obstacle Avoidance System for a Quadrotor UAV*, 1st ed., Pomona, **2016**, 1-8.
[15] A Dev, B Krose and F Groen, Navigation of a Mobile Robot on the Temporal Development of the Optic Flow, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, **1997**, 558–563.
[16] Samir Bouabdallah, Marcelo Becker, V Perrot and Roland Siegwart, Toward Obstacle Avoidance on Quadrotors, *Proceedings of the XII International Symposium on Dynamic Problems of Mechanics,* **2007**.
[17] Keum-Bae Cho and Seong-Yun Cho, The Concept of Collision-Free Motion Planning using a Dynamic Collision Map, *International Journal of Advanced Robotic Systems*, **2014,** 11(9).