International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I7P223

Volume 69 Issue 7, 167-173, July, 2021
©2021 Seventh Sense Research Group®

Control of non-linear Quadrotor using PID and Backstepping Techniques

Ritika Thusoo[#1], Sheilza Jain[*2], Sakshi Bangia[#3]

[#1]PhD Scholar, Electronics Dept, JCBUST, YMCA, Faridabad, India
[2]Assistant Professor, Electronics, JCBUST,YMCA,Faridabad,India
[3]Assistant Professor, Electrical, JCBUST, YMCA, Faridabad, India

[1]ritikathusoo@gmail.com, [2]sheilzajain@gmail.com, [3]sakshibangia@gmail.com

Abstract — *Quadrotorcan takeoff and land from any surface vertically. It has four rotors attached to each side of the arm and a processing unit attached to the center. The Quadrotor has rotational and translational movement. Rotational movements are known as roll, pitch, and yaw angles. The translational movements are along the x, y, and z planes. These movements are achieved by varying the voltages of the four rotors, which generate thrust and angular movement of the Quadrotor. In this paper, a mathematical model of Quadrotor is developed in MATLAB Simulink. Angle (Attitude) and Height (Altitude) of the Quadrotor are controlled using a PID controller. For position control, the Backstepping control technique is implemented. But to obtain a stable response for the motion along x and y, PID controller gains are tuned using a Particle Swarm Optimization (PSO) code. Similarly, the PID controller gain used in the subsystem that represents motion along the z-axis is tuned by cascading a PI controller with the subsystem. The paper is concluded with complete control of all six parameters of the Quadrotor system.*

Keywords — *Backstepping, PID, PSO, PI-PID Quadrotor.*

I. INTRODUCTION

UAVs can be categorized based on their wing type as rotor-type or fixed-wing type Unmanned Aerial Vehicles (UAVs)[1]. The quadrotor is a rotor-type UAV with four rotors. Quadrotors have proved to be very useful, over the years, in hazardous areas[2]. The advancement in sensors, their computation capability, and their communication protocols have made them cost-efficient and highly efficient. This has, in turn, increased the applications of UAVs[3]. The advancement in microcontroller technology has further enabled the manufacture of Quadrotors with small size and high efficiency[4]. The Quadrotors can be used in geological mapping, disaster management, crop inspection, fertilizer spraying and manure spreading[5], pipeline inspection, product, and medicine delivery[6]. Researchers and few companies like Amazon have been working on Quadrotors extensively. Amazon has been working on a product delivery-based drone service known as Amazon Prime Air[7] for delivery of products within a certain radius in thirty minutes.

The Quadrotor system is non-linear in nature and is extremely underactuated. They are still widely studied because of their ability to hover in air, land, and take flight vertically from anywhere [8]. Robust position control of a Quadrotor is designed in two parts: First, a Backstepping Sliding Mode controller is designed for the Attitude of the Quadrotor, and second, an Integral Sliding Mode controller is designed for the position control of the Quadrotor. Developed using the backstepping sliding control technique to track trajectories for a Quadrotor system. The robust controllers designed are subjected to disturbances. The efficacy of the controller design is inspected using the Lyapunov Stability test. The system is also studied for different scenarios [9]. A multiple Quadrotor system with multiple actuator faults is controlled using Reinforcement Learning (RL). The RL is applied along with the optimal control theory to develop an optimal control policy that can overcome the actuator faults in multiple Quadrotors. The designed control system is model-free design and does not require system dynamics. The data obtained due to the Quadrotor's motion is used to train the controller. The simulation results showcase the effectiveness of the designed system[10]. The Quadrotor systems are highly non-linear and underactuated. Even then, Quadrotors are studied widely across the world owing to their multitudinous applicability in difficult terrains and their use in applications that can benefit society and save precious lives. Another position and attitude control of such a system is designed and developed using the Adaptive function Approximation technique. The designed controller employs reference signals derived from linear feedback PD control methodology. The Lyapunov function testifies for the stability of this system, and the system is simulated for three different scenarios[11]. Literature survey of some other Quadrotor system designs developed are discussed in Table 1.

In this paper, a mathematical model based on Newton-Euler equations is used to develop a Simulink model. The implementation of the control of Quadrotor is done in two parts: Attitude & Altitude control and Position Control of Quadrotor. The arrangement of this paper is as follows: The mathematical model of Quadrotor is presented in section 2. The implementation of the mathematical model Quadrotor is given in section 3. The results and discussion obtained for Attitude & Altitude control and Position Control of Quadrotor are given in section 4. Section 5 represents the conclusion and future scope.

Table.1. Literature Survey of Quadrotors designed

Control Algorithm Implemented	Comment	Reference
PID controller is used to manipulating the angular velocities to control the Attitude of the Quadrotor.	Quadrotor rotation manipulates the Ultrasonic sensors. This affects the IMU measurement, which causes errors in the angular rate detection.	[12]
Cascade Active Disturbance Rejection Control is used to control the Attitude of the Quadrotor, and a novel Backstepping Sliding-Mode Control is implemented for the Position Control of the Quadrotor.	The designed controller is tested on a Real-Time Quadrotor system that uses a Pixhawk flight controller. The designed system can reduce the tracking error.	[13]
The effect of the external disturbances on the Quadrotor system is overcome by designing a Saturation Integral Backstepping Control (SIBC).	The designed controller technique is better than Classical Backstepping Control and Integral Backstepping Control.	[14]
PID controller gains are tuned using PSO to obtain Attitude control of the Quadrotor.	PSO is used to tune the PID gain parameters to their optimal values.	[15]

II. MATHEMATICAL MODEL OF QUADROTOR

The Quadrotor systems have the following traits: non-linearity,underactuation[16], instability[17], and multi-variables[18]. Quadrotor has six Degree of Freedom (DoF) and four actuators to manipulate the system parameters. These rotors are paired as Pair (1, 3) and Pair (2, 4), as represented in figure 1. The pair (1, 3) rotates in a clockwise direction, and the pair (2, 4) rotates in a counterclockwise direction. The thrust introduced by all the four rotors: F_1, F_2, F_3, and F_4, as shown in figure 1.

In 3D space, an aerial vehicle such as the Quadrotor system can be represented by two frames. The first frame is an inertial frame with respect to Earth and the second frame is a body-fixed frame or B-frame. The B-frame is believed to be attached to the body of the system[19].

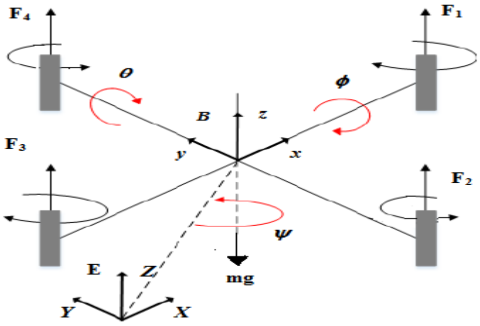

Fig. 1 Diagram of Quadrotor system[20]

The force generated by the four rotors is denoted by $F_{1\text{-}4}$, as depicted in figure 1. The angles are φ (phi) angle for x plane; θ (theta) angle for y plane, and ψ (psi) angle for z plane. The rotational motion is achieved by varying the voltage of the four rotors. The translational motion can also be represented based on these variations in voltages and the variation in tilt angles of the system. The arrangement of Quadrotor is based on two designs: plus structure and cross structure[21]. In figure 1 cross-structure design of a Quadrotor is depicted.

The equations of motion that describe the Quadrotor system are given by:

$$\theta'' I_{xx} = (-F_1 - F_2 + F_3 + F_4)d_{arm} \qquad (1)$$

$$\varphi'' I_{yy} = (-F_1 + F_2 + F_3 - F_4)d_{arm} \qquad (2)$$

$$I_z z = (\tau_{m1} - \tau_{m2} + \tau_{m3} - \tau_{m4})d_{arm} \qquad (3)$$

$$x'' = \frac{1}{mass_{Quad}} \sum_{1}^{4} F_i[\sin\varphi \sin\psi + \cos\varphi \cos\psi \sin\theta] \quad (4)$$

$$y'' = \frac{1}{mass_{Quad}} \sum_{1}^{4} F_i[\sin\psi \sin\theta \cos\varphi - \cos\psi \sin\varphi] \qquad (5)$$

$$z'' = \frac{1}{mass_{Quad}} \sum_{1}^{4} F_i[\cos\varphi \cos\theta] - g \qquad (6)$$

where F_i the thrust is achieved at motors, d_{arm} is the length of the Quadrotor's arm, τ_m is the torque produced by BLDC motor, I_i is the moment of inertia of respective axes, and $mass_{Quad}$ is the mass of the Quadrotor System.

The equations of motion represented by equations (1)-(6) are represented using differential equations.
The values of the thrust are represented by F_i in the equations (1)-(6) to obtain the differential equations for Roll, pitch and yaw angle, and x, y, and z (height) motion.

A. Roll Angle

The roll angle is achieved by changing the motor voltage of the pair (2,4) in a clockwise direction. The differential equation for the roll angle is given as [22]:

$$\varphi" = \frac{1}{I_{yy}}[(2\rho A d_{arm}\left[\frac{\eta f K_t}{K_q}\right]^2 (V_2^2 - V_4^2))] \qquad (7)$$

B. Pitch Angle

The pitch angle is obtained by changing the motor voltage of the pair (1,3) in an anti-clockwise direction. The differential equation for pitch angle is given as [22]:

$$\theta" = \frac{1}{I_{xx}}[(2\rho A d_{arm}\left[\frac{\eta f K_t}{K_q}\right]^2 (V_3^2 - V_1^2))] \qquad (8)$$

C. Yaw Angle

The yaw movement is achieved due to the torque imbalance caused due to the opposing movements of the rotor pairs. The yaw angle is given as [22]:

$$\psi" = \frac{1}{I_{zz}}\left(\Omega_1^{'} + \Omega_3^{'} - \Omega_2^{'} - \Omega_4^{'}\right) + \qquad (9)$$
$$D\left(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2\right)$$

D. Height (Altitude)

The thrust generated by all rotors is responsible for the following differential equation for the altitude in the z direction[22]:

$$z" = \frac{2\rho A}{mass_{Quad}}\left[\left[\frac{\eta f K_t}{K_q}\right]^2 (V_1^2 + V_2^2 + V_3^2 + V_4^2)]\right] \qquad (10)$$
$$(cos\theta cos\varphi) - g$$

E. Motion in the x-direction

The force acting in the x-direction can be denoted by the following equation[22]:

$$x" = \frac{2\rho A}{mass_{Quad}}\left[\left[\frac{\eta f K_t}{K_q}\right]^2 (V_1^2 + V_2^2 + V_3^2 + V_4^2)\right] (11)$$
$$(sin\varphi sin\psi + cos\psi cos\varphi sin\theta)$$

F. Motion in y-direction

The force acting in the y-direction can be denoted by the following equation [22]:

$$y" = \frac{2\rho A}{mass_{Quad}}\left[\left[\frac{\eta f K_t}{K_q}\right]^2 (V_1^2 + V_2^2 + V_3^2 + V_4^2)\right] (12)$$
$$(sin\psi sin\theta cos\varphi - cos\psi sin\varphi)$$

Where ρ denotes the density of air, V represents the voltage generated by BLDC motors, the efficiency of a rotor is represented by η K_q denotes the torque constant of the rotor.
The connection between thrust and the input voltages used is:

$$F = 2\rho A\left[\frac{\eta f K_t}{K_q}\right]^2 V^2 \, (13)$$

This makes use of the fluid dynamic's momentum theory or disk actuator theory where Power is equal to:

$$Power = \sqrt{\frac{Thrust^3}{2\rho A}} \qquad (14)$$

The Quadrotors design parameters used in this have taken inspiration from and referred paper in [23].

III. IMPLEMENTATION OF MATHEMATICAL MODEL OF QUADROTOR

The mathematical model of Quadrotor is implemented using MATLAB and Simulink. The implementation of the control of Quadrotor is divided into two parts. Part-1 includes the Attitude and Altitude Control of the Quadrotor, and part-2 includes the position control of the Quadrotor. Part-1 forms the inner loop, and part-2 from the outer loop in the complete control of the Quadrotor. The Attitude and Altitude control involves the control of roll, pitch, and yaw angles and the height of the Quadrotor. The Position control of the Quadrotor involves the control of the translational motion in the x, y, and z-axis.

A. Height (Altitude) and Angle (Attitude) control of Quadrotor

The Simulink model using Mathematical Equations of Quadrotor is depicted in figure 4. The desired angles, i.e., roll, pitch, yaw angles, are denoted by Phi_d, $Theta_d$, and Psi_d, respectively. The desired height (altitude) is denoted by Z_d. A Proportional-Integral-Derivative (PID) control technique is implemented to control the angles (attitude) and height (altitude) of the Quadrotor.

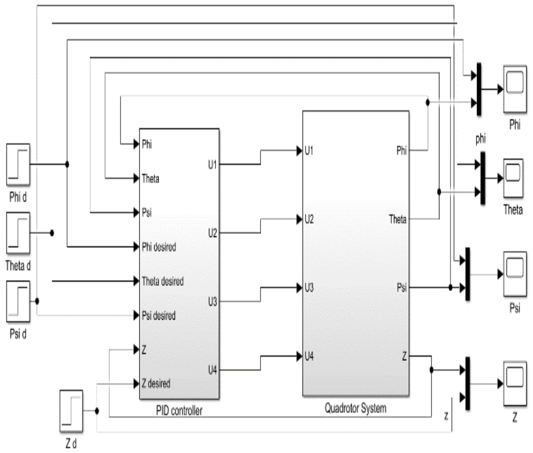

Fig.2 The Simulink model using Mathematical Equations of Quadrotor

Proportional Integral Derivative

In any closed-loop system, the controller guarantees that the Quadrotor model output will track the reference input diligently. As shown in figure 5, the controller sends the control signal to the Quadrotor Model to rectify any disparity in the output from the desired input. A PID (Proportional-Integral-Derivative) control law[24] is employed to maneuver angles (attitude) and height (altitude) of the Quadrotor.

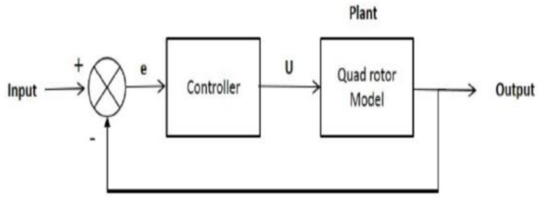

Fig.3 Illustrative representation of Closed-loop Quadrotor system

The PID guiding function is given by equation (15):

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{de(t)}{dt} \quad (15)$$

An Automatic Proportional-Integral-Derivative Tuner is applied to tune the K_p, k_i, and K_d gains for angles (attitude) and height (altitude). This is an inbuilt feature of the Simulink Control Design product in which the system response based on robustness and response time of the system can be manipulated[25]. Each parameter's angles, i.e., roll, pitch and yaw, and height (altitude), are controlled separately using a PID controller of their own.

B. Position control of Quadrotor

The system comprises of two loops, inner and outer loop. The inner loop deals with the height and attitude, whereas the outer loop deals with the position control of the Quadrotor. The mathematical system considered in this paper is non-linear. The mathematical model response for attitude and altitude gives stable results, but when applying position control to the system, responses for x, y, and z become highly interactive and cause instability. To overcome this problem, the backstepping technique has been implemented. The virtual inputs are given as desired inputs to Roll and Theta angles. These virtual inputs are the control output of PID controllers used for translational motion along the x and y-axis, respectively. A small code using Particle Swarm Optimization (PSO) is used to tune the PID gains for translational motion along the x and y-axis. This leads the response for the motion along the z-axis to become unstable. When the same PSO code is used to re-tune the PID gains and achieve stability for the translational motion response along the z-axis, there is a huge offset in the response. The response also does not settle and remains unstable. So, a PI controller was cascaded with the PID controller to obtain a stable response for translational motion along the z-axis. The effect of the interactions from signals of translational motion along the x and y-axis are finally overcome. The Simulink diagram for position control is shown in figure 4.

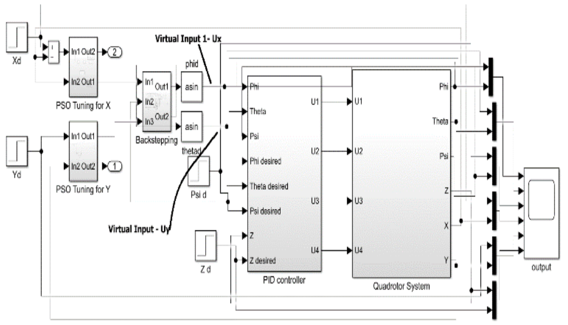

Fig.4 Complete Simulink model of Quadrotor using Backstepping Technique.

Backstepping Control Technique

The system is non-linear in nature, so a feedback linearization approach would lead to the cancellation of useful nonlinearities. But, the backstepping technique shows more flexibility compared to feedback linearization since it does not require a linear system[13]. For this, "Virtual controls" or "pseudo controls" are used[26]. The backstepping technique uses a Lyapunov function derived for the entire system. The virtual inputs for desired roll angle and desired theta angle are labeled by phi_d and $theta_d$, respectively, in figure 7. The equation for the desired roll and pitch angle can be given by the following equations (16) and (17)[27].

$$\varphi^d = arcsin(U_x \sin(\psi^d) - U_y \cos(\psi^d)) \quad (16)$$

$$\theta^d = arcsin((U_x \sin(\psi^d) - U_y \cos(\psi^d))/\cos(\psi^d)) \quad (17)$$

PSO Optimization for PID Gains

The tuning of PID gains for the translational motion along the x and y-axis is done using PSO code[28]. This is represented by 'PSO Subsystem for PID Gains' in figure 4. In PSO, a size of 50 birds in the swarm is used. This is used as PID gains obtained using PID Tuner (an inbuilt feature of the Simulink Control Design product) present in the PID block were unable to obtain a stable response. The nonlinearity arises as there are only four control inputs to control six parameters. The PSO code used was able to obtain PID gains that facilitated a stable response for translational motion along the x and y-axis. The response for translational motion along the z-axis or height (Altitude) still suffers from instability. A PI controller is added in cascade with the already used PID controller to stabilize the Attitude response. Hence, PI-PID in cascade is used for translational motion along the z-axis (Height/ Altitude). The automatic PID Tuner is used to tune the gains of the PI controller. The Tuner is called PID Tuner, and there is an inbuilt option to select different controllers like PID, PI, and PD.

IV. RESULTS AND DISCUSSION

The responses obtained for both the parts, i.e., part-1Attitude and Altitude control and part-2 Position control, are analyzed in this section. The step response is given as input to test the Controllers and control techniques used to control the Quadrotor system. The simulation time observed is 10s for this paper, and the solver used is Ode45.

A. Results for Attitude and Altitude control

The mathematical model of the Quadrotor is designed using equations (7)-(12) in Simulink. A PID control law [29]is utilized to control the attitude and altitude of the Quadrotor. The model response obtained for a step input is depicted in Figures 5, 6, 7, and 8.

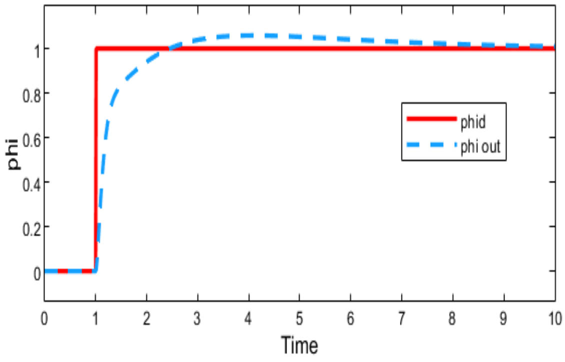

Fig.5. Response of Roll angle (phi)

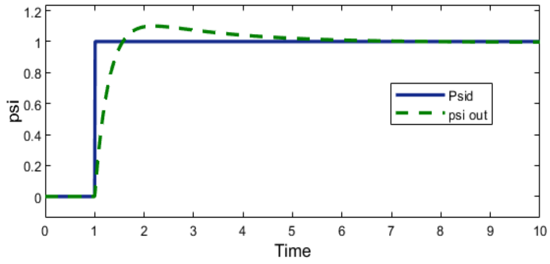

Fig.6. The output was obtained for Yaw (psi) angle.

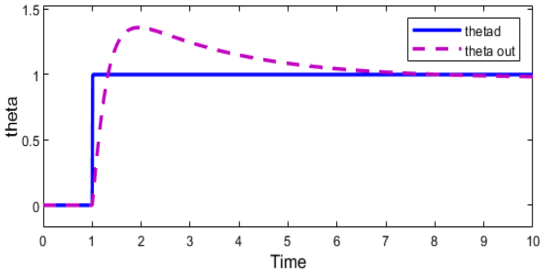

Fig.7. The output was obtained for Theta angle.

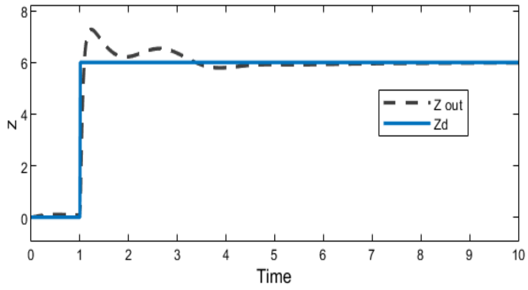

Fig.8. The output obtained for Height (Altitude)

Figure 5 depicts the graphical output response of roll angle utilizing PID control law. The response shows some offset and settles after 5s. Figure 6 shows the graphical output obtained for the yaw angle utilizing PID control law. The output response depicts some overshoot but becomes stable after 5s. The offset obtained is zero. Figure 7 shows the graphical output obtained for pitch (theta) angle. There is an overshoot at 1s, but the response settles to a stable value at 6s with zero offsets. Figure 8 depicts the graphical output for the input Zd, i.e., response for the Altitude. The settling time is around 8s. The PID controller gains for Angles (Attitude) and Height (Altitude) are compiled in table 2, and different values of parameters such as Rise time, settling, and offset are given in table 3.

Table.2. Controller Gains

	Roll	Pitch	Yaw	Height (Altitude)
K_p	0.017733	0.03143	0.013693	5.6074
K_i	0.0012492	0.003984	0.0004754	1.44260
K_d	0.034088	0.05383	0.032586	5.35209

Table.3. Values of different Parameter obtained using PID Control Technique

Phi Angle	**Rise Time (seconds)**	2s
	Settling Time (seconds)	5s
	Offset(units)	Approx. 2 units
Theta Angle	**Rise Time(seconds)**	1.5s
	Settling Time(seconds)	6s
	Offset(units)	0
Psi Angle	**Rise Time(seconds)**	2s
	Settling Time(seconds)	5s
	Offset(units)	0
Altitude (motion along z axis)	**Rise Time(seconds)**	1.5s
	Settling Time(seconds)	8s
	Offset (units)	0

B. Results for Position Control

Figure 9 shows the response of all six parameters, i.e., Roll, Pitch, and Yaw angles, motion along x, y, and z-axis. This response is obtained for the complete control of Quadrotor using the Backstepping Technique. The PSO code is also used here to tune the PID gains for controllers used to manipulate the translational motion along the x and y-axis. The simulation time is set to 10s. The response for roll (phi), pitch (theta), yaw (psi), motion along the x-axis, and motion along the y-axis. But the response for motion along the z-axis or Altitude has become unstable due to the non-linearity.

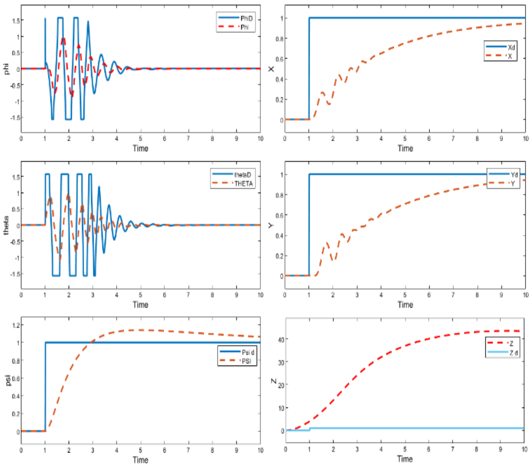

Fig.9. Response of Quadrotor using Backstepping Control Technique

Next, PI-PID in cascade is used to further re-tune the PID gains for z. The stable response for z is obtained and is shown in figure 10.

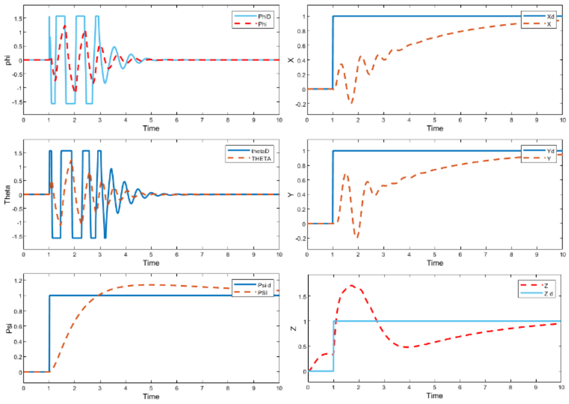

Fig.10 Response of Quadrotor when PI-PID used in cascade for motion along the z-axis

V. CONCLUSIONS

A mathematical model based on voltages applied to the rotors is used to design and develop a Simulink model of the Quadrotor. PID control law is utilized to control the Angle (Attitude) and Height (Altitude) of the Quadrotor. The values were obtained for parameters like Rise time, Settling Time, and Offset as described in Table 6. These results are also analyzed graphically. Next, for position control, the non-linearity of the system is overcome by using the Backstepping control technique. Virtual inputs are given as desired values for phi and theta angles. These are the control outputs of PID controllers used to manipulate motion along the x and y-axis, respectively. A PSO code was used to tune the PID controllers that are used to manipulate motion along the x and y-axis. This was done as the in-built PID Tuner was unable to obtain a stable response. As a result, the response for motion along the z-axis becomes unstable. To overcome this instability, the PI controller is added in cascade with the PID controller for motion along the z-axis. The graphical responses are recorded. The position control of the non-linear Quadrotor system is achieved using the Backstepping Control technique, and the complete control of a Quadrotor system is achieved.

The future scope of this paper would be to implement these techniques on a real-time application-based Quadrotor system.

REFERENCES

[1] M. A. Boon, A. P. Drijfhout, and S. Tesfamichael., Comparison of a fixed-wing and multi-rotor UAV for environmental mapping applications: A case study, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch., 42(2W6) (2017) 47–54.

[2] M. Kamran Joyo, S. F. Ahmed, D. Hazry, M. H. Tanveer, and F. A. Warsi., Position Controller Design for Quad-rotor under Perturbed Condition, Wulfenia, 20(7) (2013) 178–189.

[3] L. Zhou, J. Zhang, H. She, and H. Jin., Quadrotor uav flight control via a novel saturation integral backstepping controller, Automatika, 60(2) (2019) 193–206.

[4] G. Sylvester., E-Agriculture in action: Drones for Agriculture, 53(9) (2018).

[5] S. Kedari, P. Lohagaonkar, M. Nimbokar, G. Palve, and P. P. Yevale., Real-Time Wireless Communication between Quadcopter and Android in Agriculture Field– A Review, Int. J. Adv. Res. Comput. Sci. Softw. Eng. Real-Time Wirel. Commun. Between Quadcopter Android Agric. F. – A Rev., 1 (2015) 706–708.

[6] J. Kim, S. A. Gadsden, and S. A. Wilkerson., A Comprehensive Survey of Control Strategies for Autonomous Quadrotors, arXiv, 43(1) (2020) 3–16

[7] N. H. Motlagh, M. sBagaa, and T. Taleb., UAV-Based IoT Platform: A Crowd Surveillance Use Case, IEEE Commun. Mag., 55(2) (2017) 128–134.

[8] A. Intwala., A Review on Vertical Take-Off and Landing (VTOL) Vehicles, Innov. Res. Adv. Eng., 2(2) (2015) 186–191.

[9] D. J. Almakhles., Robust Backstepping Sliding Mode Control for a Quadrotor Trajectory Tracking Application, IEEE Access, 8 (2020) 5515–5525.

[10] W. Zhao, H. Liu, and F. L. Lewis., Data-Driven Fault-Tolerant Control for Attitude Synchronization of Nonlinear Quadrotors, IEEE Trans. Automat. Contr., 9286 (2021) 1–8.

[11] A. Shamshirgaran, D. Ebeigbe, and D. Simon., Position and Attitude Control of Underactuated Drones Using the Adaptive Function Approximation Technique, (2020).

[12] P. Salaskar, S. Paranjpe, J. Reddy, and A. Shah, Quadcopter – Obstacle Detection and Collision Avoidance., Int. J. Eng. Trends Technol., 17(2) (2014) 84–87.

[13] L. X. Xu, H. J. Ma, D. Guo, A. H. Xie, and D. L. Song., Backstepping Sliding-Mode and Cascade Active Disturbance Rejection Control for a Quadrotor UAV, IEEE/ASME Trans. Mechatronics, 25(6) (2020) 2743–2753.

[14] L. Zhou, J. Zhang, H. She, and H. Jin., Quadrotor uav flight control via a novel saturation integral backstepping controller, Automatika, 60(2) (2019) 193–206.

[15] A. Noordin, M. A. M. Basri, Z. Mohamed, and A. F. Z. Abidin., Modelling and PSO fine-tuned PID control of quadrotor UAV, Int. J. Adv. Sci. Eng. Inf. Technol., 7(4) (2017) 1367–1373.

[16] A. A. Mian and W. Daobo., Nonlinear flight control strategy for an underactuated quadrotor aerial robot, in Proceedings of IEEE International Conference on Networking, Sensing and Control, ICNSC, (2008) 938–942.

[17] R. Babaei and A. Farhad Ehyaei., Robust Backstepping Control of a Quadrotor UAV Using Extended Kalman Bucy Filter, 5(16) (2015) 2276–2291. Accessed: Apr. 30, 2019.

[18] Y. Yali, SunFeng, and W. Yuanxi., Controller Design of Quadrotor Aerial Robot, Phys. Procedia, 33 (2012)1254–1260.

[19] Tewari, Ashish., Advanced control of aircraft, spacecraft and rockets. John Wiley & Sons, (2011).

[20] K. A. Ghamry and Y. Zhang., Formation control of multiple quadrotors based on leader-follower method, in International Conference on Unmanned Aircraft Systems, ICUAS , (2015) 1037–1042.

[21] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao., Dynamics modelling and linear control of quadcopter, Int. Conf. Adv. Mechatron. Syst. ICAMechS, 0 (2016) 498–503.

[22] M. Hong, A. Bousbaine, M. H. Wu, and G. T. Poyi., Modelling

and simulation of a quad-rotor helicopter, (2016).
[23] S. Musa., Techniques for Quadcopter Modelling & Design :A review,(2018).
[24] Q. Jiao, J. Liu, Y. Zhang, and W. Lian., Analysis and design the controller for quadrotors based on PID control method, in Proceedings - 33rd Youth Academic Annual Conference of Chinese Association of Automation, YAC, (2018) 88–92.
[25] Open PID Tuner for PID tuning - MATLAB. https://in.mathworks.com/help/slcontrol/ug/designing-controllers-with-the-pid-tuner.html (accessed Oct. 07, 2019).
[26] I. a Raptis, K. P. Valavanis, and SpringerLink (Online service), Linear and Nonlinear Control of Small-Scale Unmanned Helicopters, (2011).
[27] D. Matouk, O. Gherouat, F. Abdessemed, and A. Hassam., Quadrotor position and attitude control via backstepping approach, Proc. 8th Int. Conf. Model. Identif. Control, ICMIC 2016, (2017)73–79.
[28] Z. Qi, Q. Shi, and H. Zhang., Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-Based DC motor subject to stochastic delays, IEEE Trans. Ind. Electron., 67(7) (2020) 5637–5646.
[29] Kiam Heong Ang, G. Chong, and Yun Li., PID control system analysis, design, and technology, IEEE Trans. Control Syst. Technol.,13(4) (2005) 559–576.