Design and Development of a Non-Linear Controller for Quadrotor type Unmanned Aerial Vehicle

Saptadeep Debnath and Mary Lourde R
Department of Electrical and Electronics Engineering,
BITS Pilani Dubai Campus,
Dubai, UAE
Email: f20140061d@alumni.bits-pilani.ac.in, marylr@dubai.bits-pilani.ac.in

***Abstract* — This paper presents non-linear modelling and simulation of a quadrotor type Unmanned Aerial Vehicle (UAV), and development of a controller for the same. The nonlinearities in the system along with the equations defining the same are studied extensively. The quadrotor is modelled using the mathematical equations in the MATLAB-Simulink environment, with each dynamic block governed by the corresponding equations. Experiments are carried out to further evaluate the coefficients of the proposed nonlinear controller for the quadrotor system and its performance evaluated.**

Keywords— quadrotor; UAV; non-linear control

I. INTRODUCTION

An autonomous vehicle is capable of making its own decisions on every aspect, which in turn is based on a set of protocols. For a system to be autonomous, it has to be robust in an unknown environment. There is extensive research being done on linear control of the Unmanned Aerial Vehicles (UAVs) in an unknown environment, using different control techniques.

A generic quadcopter has an 'X' (the letter X) or a '+' (plus) shape. This type of shape helps it in maintaining symmetry in a plane. Four motors along with electronics speed controllers (ESCs), at the end of each of the arms, are powered by a portable battery, which provides the required thrust to make the quadcopter fly. As there are six ranges of motion (i.e. forward, backward, left, right, up and down), but is controlled by only four motors, this system is generally referred to as an under actuated system.

Generally for the ease of computation the system is considered as linear [1]. But by the use of modern non-linear control theory, the performance of the system can be enhanced. Basic model of the quadcopter can be defined in the 6 ranges of motion as discussed earlier, where (x, y, z) are calculated on the center of mass of the vehicle. The Euler angles, (ψ, θ, φ) defines the orientation of the system. Using these, the system can be defined using the Euler-Lagrange equation. The mathematical model gives a backbone to the further research on this topic. Though widely assumed linear, the nonlinearities can surely be added to the developed mathematical model, as explained in [2]. Essentially, three types of motion need to be controlled, namely attitude (roll, pitch and yaw), altitude (z) and position (x and y).

Proportional-integral-derivative (PID) is the most commonly used controller, for this type of system. A benefit of using this type of controller is that, it does not rely on the accurate model of the quadrotor [3]. In a PID loop, the errors in the loop are compensated by a three-stage function, directly dealing with the error (proportional), dealing with the error accumulated over time (integral) and compensating for the future errors in the system (derivative). Some other controllers include sliding mode controller [4], backstepping method [5], nonlinear H∞ controller [6], and model predictive controller [7].

The MATLAB-Simulink environment is used to model the quadrotor using mathematical equations taking in factor the nonlinearity of the system, and simulate different experiments to calculate the coefficients of the PD controller.

II. QUADCOPTER DYNAMICS

A. Nonlinearities in the System

The paper presents a non-linear control system for a quadrotor with the following nonlinearities. The system is not considered as a point object, thus the object is presumed to have resistance in a fluid environment and have a drag coefficient. The vehicle is assumed to have a moment of inertia along the three axes, symmetrical along the x and y axes, and have the center of mass at the geometric center of the vehicle. Additionally, the propellers are assumed rigid so as to provide equal thrust to all the motors, and the thrust is assumed to be proportional to the square of the propellers' speed.

B. Dynamic Modelling

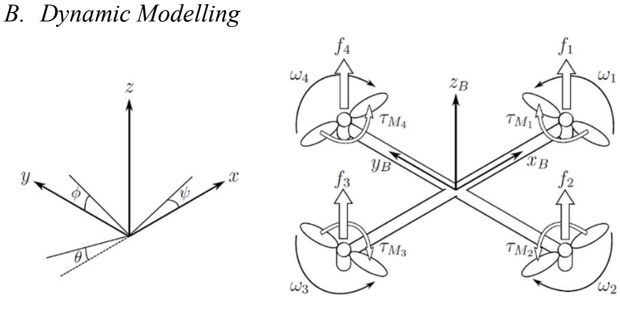

Fig. 1. Inertial and Body frame of a plus '+' type quadrotor

The quadcopter design as represented in Fig .1, demonstrates the torques, angular velocities, forces and the motion in the different directions (x, y, z) & along the different angles (φ, θ, ψ). [9]

A rotation matrix (R) is used which relates the body frame to the inertial frame of the system. A vector $\vec{v}$ in the body frame is represented as R$\vec{v}$ in the inertial frame. Rotation matrix is a powerful tool as it can negate the effect of roll, pitch and yaw in the body frame, and represent the orientation in the inertial frame. (C_x = Cos x, S_y = Sin y)

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \quad (1)$$

As discussed previously, the quadcopter is assumed symmetrical in shape. Two of the arms align with the x-axis and the other two with the y-axis. Therefore, the moment of inertia across x-axis is same as of y-axis ($I_{xx} = I_{yy}$). The inertial matrix can be denoted as,

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2)$$

The angular velocities of the individual motors produce a torque denoted as τ_{M_i}, where *'i'* specifies the motor numbers. The torques produced by the motors depend on two major non-linear components, the drag factor *'d'* and the inertia moment of the rotors I_M.

$$\tau_{M_i} = d\,\omega_i^2 + I_M\,\dot{\omega}_i \quad (3)$$

Apart from the torques produced by the individual motors, three additional torques are also produced with reference to the body frame. The torques corresponds to the body frame angles, namely roll, pitch and yaw.

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l\,b\,(-\omega_2^2 + \omega_4^2) \\ l\,b\,(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix} \quad (4)$$

The four motors provide a thrust T, which allows the quadcopter to achieve a certain change in z-direction in the body frame.

$$T = b\,(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (5)$$

The dynamics of the quadcopter discussed above are further used to derive the acceleration of the system along the x, y and z-axes using the Newton-Euler equations.

$$m\ddot{a} = G + RT_B \quad (6)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{T}{m} \begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

In this representation, $\ddot{a}$ is a column matrix with the acceleration values across the x, y and z-axes. The gravitational constant G when added with the amount of thrust produced in the inertial frame, i.e. taking a product of the rotation matrix R with the thrust in the body frame T_B gives the total forces in the x, y and z-axes. Similar to the acceleration calculation in the x, y and z-axes, the acceleration for roll, pitch and yaw is denoted as follows.

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\dot{\theta}\dot{\psi}/I_{xx} \\ (I_{zz} - I_{xx})\dot{\phi}\dot{\psi}/I_{yy} \\ (I_{xx} - I_{yy})\dot{\theta}\dot{\phi}/I_{zz} \end{bmatrix} + I_M \begin{bmatrix} \dot{\theta}/I_{xx} \\ \dot{\phi}/I_{yy} \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix} \quad (8)$$

Where,

$$\omega_\Gamma = \omega_2 + \omega_4 - \omega_1 - \omega_3 \quad (9)$$

III. Controller Design

A. Attitude Control

A PD controller is used for this research as the system is assumed to have less erratic disturbances [9].

$$\begin{aligned} \tau_\phi &= [K_{P,\phi}(\phi_d - \phi) + K_{D,\phi}(\dot{\phi}_d - \dot{\phi})]I_{xx}, \\ \tau_\theta &= [K_{P,\theta}(\theta_d - \theta) + K_{D,\theta}(\dot{\theta}_d - \dot{\theta})]I_{yy}, \quad (10) \\ \tau_\psi &= [K_{P,\psi}(\psi_d - \psi) + K_{D,\psi}(\dot{\psi}_d - \dot{\psi})]I_{zz}, \end{aligned}$$

The corrected angular velocities can now be calculated from (4) and (5) with the result from (10) and is given by (11).

$$\omega_i^2 = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \frac{T}{4b} + \frac{1}{2dl} \begin{bmatrix} -\tau_\theta \\ -\tau_\phi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} + \frac{1}{4d} \begin{bmatrix} -\tau_\psi \\ \tau_\psi \\ -\tau_\psi \\ \tau_\psi \end{bmatrix} \quad (11)$$

B. Position Control

Position control of the quadrotor is achieved by changing the four control inputs to the quadcopter, i.e. the angular velocities of the four motors. This control loop is generally referred to as the outer control loop, and implemented on an onboard/offboard computer.

$$\begin{aligned} X_d &= K_{P,x}(x_d - x) + K_{D,x}(\dot{x}_d - \dot{x}), \\ Y_d &= K_{P,y}(y_d - y) + K_{D,y}(\dot{y}_d - \dot{y}), \quad (12) \\ Z_d &= K_{P,z}(z_d - z) + K_{D,z}(\dot{z}_d - \dot{z}), \end{aligned}$$

The approach for trajectory control is to calculate the required angular velocities to move the quadcopter from the current position to a desired position. The desired angular velocities can be calculated by (11), using the thrust and the torque values. The torque value are in turn calculated by (10). To correlate both the equations we require a relation between the desired values of roll, pitch and yaw with the desired values of x, y and z, which is given in (13)

$$\phi_d = \arcsin\left(\frac{X_d S_\psi - Y_d C_\psi}{X_d^2 + Y_d^2 + (Z_d+g)^2}\right),$$

$$\theta_d = \arcsin\left(\frac{X_d C_\psi + Y_d S_\psi}{Z_d + g}\right), \qquad (13)$$

$$T = m\,[X_d(C_\psi S_\theta C_\phi + S_\psi S_\phi) + Y_d(S_\psi S_\theta C_\phi - C_\psi S_\phi) + (Z_d + g)(C_\theta C_\phi)],$$

IV. Experimental Setup

Initially, the desired position coordinates are fed to a PD controller. The controlled input is then converted to the desired roll, pitch, yaw and the thrust value. The desired roll, pitch and yaw values then go through a PD controller, which is the inner control loop, which was described in the previous section. The corresponding torque values in addition with the thrust is used to calculate the angular velocities of the individual motors. The angular velocities with the help of the calculated roll, pitch, and yaw values are used to then find the current x, y and z coordinates. The simulations for verifying the model and calculating the PD coefficient values are carried on a MATLAB environment. Fig. 3 shows the overall Simulink block which consists of two main subsystems, with one output terminal.

The system is fed with three sets of input variables: initial attitude values, system values and the final position values. The initial values and the rate of change of x, y, z, roll, pitch and yaw are fed in the initial attitude values; the default values are chosen as shown in Table 1. The default values shown in Table 2 are given as input for the system values, which comprise of quadrotor mass, thrust factor, drag factor, rotor inertia, length of each chord and the moment of inertia along each axes. The system values are chosen so as to replicate a real life quadrotor system. The initial attitude values and system values remain constant throughout the simulation process. The final position values are varied in the different simulations, so as to tune a specific set of values.

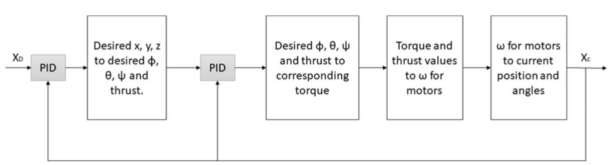

Fig. 2. Interactions between different physical quantities

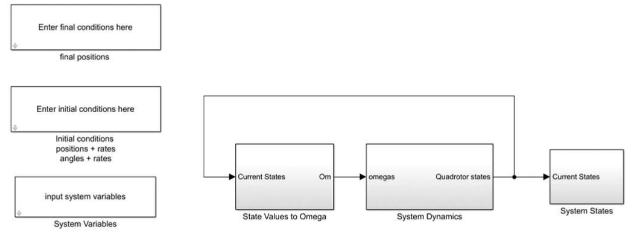

Fig. 3. Complete Simulink Block Diagram

The first dynamic block is responsible for converting the desired position, with the help of the current state values and the initial system variables, to the required omega (RPM) values for each of the four motors. As described in section 3, there are two levels of control mechanism in this system; the inner control loop and the outer control loop. The attitude control block is modelled based on (10). This block converts the desired roll, pitch and yaw values to their corresponding torque values. The torque and the thrust obtained is then parsed into the next block, which then calculates the equivalent omega values, according to (11). The outer loop controls the position values (i.e. x, y and z). The position control block is defined by (12). Using the output obtained from this block, the required thrust, roll and pitch values are calculated, which are modelled by (13).

TABLE I. INITIAL ATTITUDE VALUES

Parameter	***Value***
Initial X position (m)	1
Initial Y position (m)	1
Initial Z position (m)	0
Initial Velocity in X direction (m/sec)	0
Initial Velocity in Y direction (m/sec)	0
Initial Velocity in Z direction (m/sec)	0
Initial Roll Angle (deg)	0
Initial Pitch Angle (deg)	0
Initial Yaw Angle (deg)	0
Initial Roll Rate (deg/sec)	0
Initial Pitch Rate (deg/sec)	0
Initial Yaw Rate (deg/sec)	0

The individual omega values obtained from the previous block is further used to calculate the current states. Equations (7) and (8) are used to calculate the acceleration values across the six coordinates; i.e. x, y, z, roll, pitch and yaw. The double derivatives are then integrated to get the velocity values, and integrated further to obtain the attitude values. Finally, the current state values are parsed into the output block to visualize the values in a systematic manner.

TABLE II. SYSTEM SPECIFICATIONS

Parameter	Value
Quadrotor Mass (kg)	0.468
Quadrotor Moment of Inertia along x axis (kg m^2)	0.004856
Quadrotor Moment of Inertia along y axis (kg m^2)	0.004856
Quadrotor Moment of Inertia along z axis (kg m^2)	0.008801
Thrust Factor	0.00000298
Drag Factor	0.000000114
Rotor Inertia (kg m^2)	0.00003357

V. RESULTS AND DISCUSSION

The system contains six individual PD controllers for x, y, z, roll, pitch and yaw. The PD coefficients are tuned using three sets of experiments.

First the z-control, or the altitude control is tuned. In this simulation the initial coordinates are set as [1, 1, 1] and the desired coordinates as [1, 1, 8]. As seen in Fig. 4(a), there is no change observed in the x and y axis, as was desired. Consequently, there is no change in either roll, pitch or yaw, expect for the desired increase in the coordinate value of the z-axis, from 1 to 8. $K_{P,z}$ and $K_{D,z}$ are tuned in this simulation.

In the second simulation, the x axis is tuned. As the x-axis motion is solely depended on the pitch motion, so along with x the pitch controller is tuned. The initial coordinates are set as [1, 1, 1] and the desired coordinates as [8, 1, 1] which results in the forward motion of the quadrotor. As expected there is a desirable change in the x-axis, as shown in the Fig. 4(c). In addition to that there are minute disturbances observed in the y and z-axis, which attain the steady state value at the end of the simulation. $K_{P,x}$, $K_{D,x}$, $K_{P,\theta}$ and $K_{D,\theta}$ are tuned during this simulation.

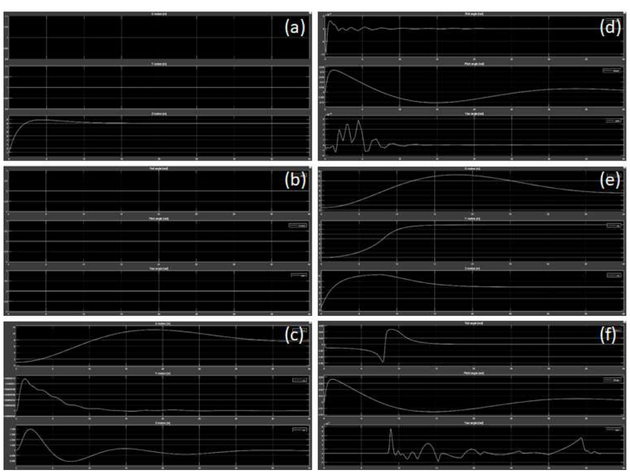

Fig. 4. (a) Position and (b) Attitude graph for simulation 1, (c) Position and (d) Attitude graph for simulation 2, (e) Position and (f) Attitude graph for simulation 3; x, y, and z values are plotted independently in the position graph, and roll, pitch and yaw values are plotted in the attitude graph. [The x-axis denotes the simulation time (0 - 40 sec) and y-axis denotes the coordinate points in-case of position graph and radians in-case of attitude graphs]

TABLE III. TUNED CONTROLLER PARAMETERS

Parameter	Value	Parameter	Value
$K_{P,x}$	0.0058	$K_{P,\theta}$	0.000256
$K_{D,x}$	0.1866	$K_{D,\theta}$	0.0134
$K_{P,y}$	12.828	$K_{P,\phi}$	2.477
$K_{D,y}$	27.984	$K_{D,\phi}$	0.94
$K_{P,z}$	0.21	$K_{P,\psi}$	234.031
$K_{D,z}$	0.976	$K_{D,\psi}$	68.748

The final simulation is carried out for tuning the y-axis motion. As y-axis motion is defined by the roll motion, the controller for roll angle is also controlled in the process. The initial coordinates for the simulation as taken as [1, 1, 1] and the desired coordinates as [8, 8, 8]. From the Fig. 4(e), it can be observed that the desired coordinate values for x, y and z are achieved by the system. In addition to that the changes in the roll, pitch and yaw angles also negate down to zero when the simulation stops. During this simulation the yaw angle is also tuned, which gives $K_{P,y}$, $K_{D,y}$, $K_{P,\phi}$, $K_{D,\phi}$, $K_{P,\psi}$ and $K_{D,\psi}$.

VI. CONCLUSION

This research is presented in two folds. The first part deals with the basics of the quadcopter dynamics, the study of different types of nonlinearities in the system and the equations defining the same. Further, the system dynamics and the control architecture is modelled in the Simulink platform as separate subsystems. A PD controller is used to control the six coordinates as defined in a quadrotor system. This includes a two level control system, an inner control loop and an outer control loop.

The latter half deals with the simulation results from the Simulink block, by feeding in the desired values and observing the anticipated motion. A three step simulation is carried out to tune the six control blocks. The three motions being the upward motion, forward motion and a diagonal motion (i.e. a change in all the three coordinates). The system thus designed responds properly for a localized coordinate system, as the parameters are tuned for a localized coordinate system. This design serves as an ideal non-linear simulation model for a plus '+' configured quadcopter.

ACKNOWLEDGMENT

We will like to express our deep gratitude to all colleagues who contributed to our current knowledge of nonlinear control system and its theoretical implications in the robotics field. The authors are grateful to authorities of BITS Pilani, Dubai Campus for their support and encouragement to carry out this research work.

REFERENCES

[1] Castillo, Pedro, Rogelio Lozano, and Alejandro Dzul. "Stabilization of a mini-rotorcraft having four rotors." Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 3. IEEE, 2004.

[2] Choi, Young-Cheol, and Hyo-Sung Ahn. "Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests." IEEE/ASME transactions on mechatronics 20.3 (2015): 1179-1192.

[3] Salih, Atheer L., et al. "Modelling and PID controller design for a quadrotor unmanned air vehicle." Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on. Vol. 1. IEEE, 2010.

[4] Xu, Rong, and Umit Ozguner. "Sliding mode control of a quadrotor helicopter." Decision and Control, 2006 45th IEEE Conference on. IEEE, 2006.

[5] Madani, Tarek, and Abdelaziz Benallegue. "Backstepping control for a quadrotor helicopter." Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006.

[6] Raffo, Guilherme V., Manuel G. Ortega, and Francisco R. Rubio. "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter." Automatica 46.1 (2010): 29-39.

[7] Bangura, Moses, and Robert Mahony. "Real-time model predictive control for quadrotors." (2014).

[8] Hu, Kaijian, Yuhu Wu, and Xi-Ming Sun. "Attitude controller design for quadrotors based on the controlled Hamiltonian system." Control And Decision Conference (CCDC), 2017 29th Chinese. IEEE, 2017.

[9] Dikmen, İ. Can, Aydemir Arisoy, and Hakan Temeltas. "Attitude control of a quadrotor." Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on. IEEE, 2009.

[10] Zuo, Zongyu. "Trajectory tracking control design with command-filtered compensation for a quadrotor." IET control theory & applications 4.11 (2010): 2343-2355.

[11] Luukkonen, Teppo. "Modelling and control of quadcopter." Independent research project in applied mathematics, Espoo 22 (2011).

[12] Hehn, Markus, and Raffaello D'Andrea. "Quadrocopter trajectory generation and control." IFAC world congress. Vol. 18. No. 1. 2011.

[13] Gibiansky, Andrew. "Quadcopter dynamics, simulation, and control." Andrew. gibiansky. com (2012).

[14] Carrillo, Luis Rodolfo García, et al. "Modeling the quad-rotor mini-rotorcraft." Quad Rotorcraft Control. Springer, London, 2013. 23-34.

[15] Mellinger, Daniel, Nathan Michael, and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors." The International Journal of Robotics Research 31.5 (2012): 664-674.

[16] Bhatkhande, Pranav, and Timothy C. Havens. "Real time fuzzy controller for quadrotor stability control." Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on. IEEE, 2014.