Hovering control of a quadcopter using linear and nonlinear techniques

Harikrishnan Suresh*, Abid Sulficar and Vijay Desai

Department of Mechanical Engineering,
National Institute of Technology Karnataka,
Surathkal, P.O. Srinivasnagar,
Mangalore, 575 025, India
Email: harikrishnan.suresh12@gmail.com
Email: abithsulficar@gmail.com
Email: desai@nitk.ac.in
*Corresponding author

Abstract: This paper presents a comparative study on linear and nonlinear control techniques for the near-hover attitude stabilisation of a quadcopter. A dynamic model of the quadcopter is developed using Newton-Euler equations, which is inherently nonlinear. Firstly, the classical PID controller is implemented directly on the nonlinear system by decoupling the attitude dynamics and using separate controllers for each attitude variable. Linear controllers can also be implemented on this system by linearising it about an operating point, which is shown for the linear quadratic regulator (LQR). Such a linear approximation may not always retain the actual system dynamics and are not very efficient in the real world scenario. Model based nonlinear controllers prove to be superior in such instances, and one such popular technique – Feedback Linearisation using dynamic inversion is discussed in this paper. The proposed control algorithms are tested on the quadcopter model using numerical simulations in MATLAB/Simulink and analysed in terms of fall time, percentage undershoot and computation time.

Keywords: quadcopter; nonlinear control; linear quadratic regulator; model based control; feedback linearisation; dynamic inversion.

Reference to this paper should be made as follows: Suresh, H., Sulficar, A. and Desai, V. (2018) 'Hovering control of a quadcopter using linear and nonlinear techniques', *Int. J. Mechatronics and Automation*, Vol. 6, Nos. 2/3, pp.120–129.

Biographical notes: Harikrishnan Suresh received his Bachelor's degree in Mechanical Engineering from National Institute of Technology Karnataka. He is currently a graduate student in Robotic Systems Development at Carnegie Mellon University. His primary interests include control, motion planning, and perception algorithms applied to mobile robots and unmanned aerial vehicles.

Abid Sulficar received his Bachelor's degree in Mechanical Engineering from National Institute of Technology Karnataka. He currently works as a Senior product development associate at Think & Learn Pvt. Ltd, an ed-tech company based in India. His areas of interest are robotics, control theory, and product development.

Vijay Desai received his PhD from National Institute of Technology Karnataka and MTech from Indian Institute of Science, Bangalore. He is currently a Professor in the Department of Mechanical Engineering at National Institute of Technology Karnataka, Surathkal. He has 30 years of academic experience and has about 40 peer reviewed journal papers and 30 conference papers to his credit. His research areas include manufacturing engineering and mechatronics.

1 Introduction

Unmanned aerial vehicles (UAVs) are now becoming increasingly popular in diverse fields such as reconnaissance, aerial surveillance, search and rescue missions as first responder and industrial automation. Quadcopter UAVs are superior to their counterparts due to the small size and high manoeuvrability which enable them to traverse complex trajectories efficiently. A quadcopter is an under-actuated system with six degrees of freedom but only four control inputs which are the rotor speeds. The thrust and torques required for driving the quadcopter are generated by adjusting the individual rotor speeds. To achieve propulsion in a particular direction, the axis of quadcopter must be tilted with respect to the vertical. The translational motion of the quadcopter is hence coupled with its angular orientation, making quadcopter dynamics and control complex.

Mathematical modelling is the first and most critical step towards understanding the system dynamics, which precedes controller design. The differential equations governing the quadcopter dynamics are mainly derived using the Newton-Euler approach (Luukkonen, 2011) and Euler-Lagrange approach (Bouabdallah et al., 2004; Das et al., 2009). Complex interactions like blade flapping and rotor stiffness effects are neglected in most cases due to challenges in modelling. Quadcopter control focuses mainly on two class of problems: attitude stabilisation and trajectory following. The controllers used for this purpose belong to three categories: linear controllers, model-based nonlinear controllers and learning based controllers as seen in Li et al. (2015). The implementation of the controllers belonging to the first two categories have been discussed in multiple literature whereas the latter approach is still under development.

Attitude control of a quadcopter using proportional-integral-derivative (PID) controllers is discussed in Luukkonen (2011). The concept of independent control action for the attitude variables is shown here. The same decoupled control law is followed in Bouabdallah et al. (2004), where the PID controllers are designed neglecting the gyroscopic effects but including the rotor dynamics. A comparison with linear quadratic regulator (LQR) is shown, for which the system is linearised around each state to accommodate a larger extent of flight motion. The same linearisation scheme has been adopted in this paper. Sabatino (2015) attempts total control of quadcopter using the LQR technique, after performing model linearisation using small angle approximation and applying the equilibrium conditions.

Among the nonlinear controllers, feedback linearisation control (FBL) is one of the most popular approaches seen in literature. Two methods of feedback linearisation control are discussed in Sabatino (2015):

1 Exact linearisation and non-interacting control via dynamic feedback.

2 Dynamic inversion with zero-dynamics stabilisation.

For (1), the position variables and yaw are chosen as the output function and the thrust input is delayed till its second derivative to ensure non-singularity of the feedback law. The system is then extended to include the thrust input and its first derivative as the states, which then fulfils the condition for feedback linearisation. The same extended system approach is presented in Lee et al. (2009) where the small angle approximation is used to simplify the dynamics before performing repeated differentiation. For the inputs terms to appear, the fourth derivative of the position variables is calculated before obtaining the inverse feedback law.

Trajectory control of quadcopter using feedback linearisation control by dynamic inversion is given in Bonna and Camino, 2015. Separate feedback linearisation laws are used for rotational dynamics and translational dynamics after repeated differentiation following which a linear auxiliary control input is used to stabilise the fourth order error dynamics. An important point to be noted is that the approach mentioned in our paper does not require the small angle approximation, unlike other relevant works in literature. The same dynamic inversion technique is used for the attitude control in Das et al. (2009), with the attitude variables chosen as the outputs of interest. However through small angle approximation for the Euler angles, the matrix to be inverted is directly obtained from the dynamics without performing any repeated differentiation. The resulting linearised dynamics can be controlled using any of the standard techniques like PID or back-stepping control. In our paper, an attempt is made to achieve attitude control using a combination of the same dynamic inversion technique and PD control without the small angle approximation.

A comparative study between PID controller, inverse control, sliding mode control and back-stepping control for attitude stabilisation is carried out in Dikmen et al. (2009). Using total error as the criterion to evaluate the performance, the sliding mode control proves to be the best controller. A more detailed analysis between feedback linearisation control and sliding mode control is performed in Lee et al. (2009). For feedback linearisation control, uncertainty in the dynamic model can severely affect performance, and even cause instability. In addition, the dependence on higher derivative terms of the states makes it sensitive to external disturbances. While sliding mode control is a more robust approach which compensates for model uncertainties and external disturbances, handling these uncertainties causes very high input gains and is a serious problem in power-limited systems like mini quadcopters. The feedback linearisation control is simpler to implement and also use more efficient inputs without chattering, compared to sliding mode control.

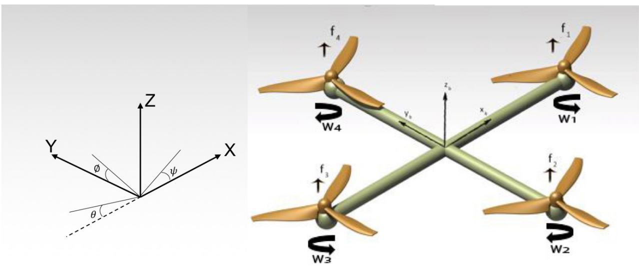
In this paper, the hovering control problem for a quadcopter is addressed. An attempt is made to achieve near hover attitude stabilisation using both linear and nonlinear control techniques. Firstly, PID control is applied individually to each attitude variable to obtain the expression for thrust and corresponding torque components based on the error values. While PID control is implemented without any model linearisation, the LQR control is applied only after linearising the attitude sub model around each state. Finally, the feedback linearisation control based on dynamic inversion is proposed. PD control is used in combination to generate the auxiliary control input to stabilise the second order error dynamics. The dynamic model takes into consideration all the contributing forces including aerodynamic drag so that the controller is well suited to handle the external disturbances.

The paper is organised as follows: Section 2 deals with the dynamic modelling of the quadcopter using Newton-Euler formulation. In Section 3, attitude controllers based on PID, LQR and Feedback Linearisation are proposed. Section 4 contains the simulation results for the proposed controllers along with a comparison study. Section 5 contains the concluding remarks of this paper and future course of action.

2 Mathematical model of quadcopter

The basic model of a quadcopter is given in Figure 1, based on which the operating principle is explained. Rotors 1 and 3 rotate in the counter-clockwise direction with angular velocities ω_1 and ω_3, whereas rotors 2 and 4 rotate in the clockwise direction with angular velocities ω_2 and ω_4. Two frames of reference – an inertial frame {O} and a body-fixed frame {B} are used to describe the quadcopter dynamics. The inertial frame has been chosen with gravity in the negative z direction. The rotations about the x, y and z axes are respectively called roll, pitch and yaw. The position of the centre of mass of the quadcopter is expressed in the inertial frame as $\epsilon = \begin{bmatrix} X\, Y\, Z \end{bmatrix}^T$. The angular position is defined in the inertial frame with the Euler angles $\eta = \begin{bmatrix} \phi\, \theta\, \psi \end{bmatrix}^T$ whereas angular velocities $\nu = \begin{bmatrix} P\, Q\, R \end{bmatrix}^T$ are defined in the body frame.

Figure 1 Quadcopter model with frames of reference (see online version for colours)



The relationship between these two frames is expressed using the rotation matrix $\mathbf{R}$.

$$\mathbf{R} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \quad (2.1)$$

where $C_\theta = \cos\theta$ and $S_\theta = \sin\theta$.

The steady state thrust and reaction torque (due to rotor drag) for a spinning rotor in free air is modelled using momentum theory. Using a lumped parameter approximation, the expressions for thrust and reaction torque are obtained.

$$T_i = K\omega_i^2, \qquad \tau_i = B\omega_i^2$$

in which K is the lift constant and B is the drag constant.

The total thrust developed by the four rotors is given by

$$T = K\sum_{i=1}^{4}\omega_i^2 \quad (2.2)$$

The rolling and pitching moments occur due to the difference in thrust produced by the opposing rotors.

$$M_\phi = L(T_4 - T_2) \quad (2.3)$$

$$M_\theta = L(T_3 - T_1) \quad (2.4)$$

where L is the distance between the rotor and the centre of mass of the quadcopter.

Yawing moment is caused by the drag force acting on all the propellers and opposing their rotation.

$$M_\psi = B(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \quad (2.5)$$

The following are the assumptions underlying the dynamic model of the quadcopter.

- The quadcopter structure is rigid.
- The structure of the quadcopter is symmetrical. Hence, the inertia matrix is diagonal and time-invariant.

$$\mathbf{I} = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix}$$

- The centre of mass of the quadcopter coincides with the origin of the body frame {B}.
- Blade flapping effect (deformation of blades at high velocities) has been neglected.

Newton-Euler formulation is used to derive the dynamic equations of motion for the quadcopter. In the case of a quadcopter, it is convenient to express the velocity dynamics with respect to a mixed frame {M} composing of the linear dynamics with respect to the inertial frame {O} and the angular dynamics with respect to the body frame {B}. The velocity vector in the mixed frame is given by $\begin{bmatrix} \dot{X}\, \dot{Y}\, \dot{Z}\, P\, Q\, R \end{bmatrix}$.

In the inertial frame, the external forces acting on the quadcopter are gravitational force, thrust and aerodynamic drag. While the gravitational force acts in the inertial frame negative z direction, the thrust acts in the body frame z direction. The magnitude of drag force along the coordinate axes are directly proportional to the components of velocities in the corresponding directions $F_D = \begin{bmatrix} A_x\dot{X} & A_y\dot{Y} & A_z\dot{Z} \end{bmatrix}^T$.

The velocities in the inertial frame are represented as

$$\dot{X} = U \quad (2.6a)$$

$$\dot{Y} = V \quad (2.6b)$$

$$\dot{Z} = W \quad (2.6c)$$

Newton's law for translational dynamics can be expressed in the inertial frame in vector form as

$$\ddot{\epsilon} = \frac{1}{m}(\mathbf{G} + \mathbf{R}T - F_D) \quad (2.7)$$

or in the component form as:

$$\ddot{X} = \dot{U} = (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)\frac{T}{m} - \frac{A_x}{m}U \quad (2.8a)$$

$$\ddot{Y} = \dot{V} = (-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi)\frac{T}{m} - \frac{A_y}{m}V \quad (2.8b)$$

$$\ddot{Z} = \dot{W} = -g + (\cos\phi\cos\theta)\frac{T}{m} - \frac{A_z}{m}W \quad (2.8c)$$

The equations of rotational dynamics are expressed in the body frame. The transformation of angular velocities from body frame to inertial frame is given by.

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi tan\theta & \cos\phi tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta}\end{bmatrix}\begin{bmatrix}P\\ Q\\ R\end{bmatrix} \tag{2.9}$$

The angular acceleration due to the quadcopter's inertia $\mathbf{I}\dot{\nu}$, the moment due to centripetal force $\nu \times \mathbf{I}\nu$ and gyroscopic couple Γ are balanced by the external torque and the resistive torque generated due to aerodynamic drag. The torque vector is given by $\mathbf{M_B} = \begin{bmatrix}M_\phi & M_\theta & M_\psi\end{bmatrix}^T$, whereas the moment produced by drag force is $\mathbf{M_D} = \begin{bmatrix}A_r P & A_r Q & A_r R\end{bmatrix}^T$.

The rotational dynamics based on Euler equations is expressed in vector form as

$$\mathbf{I}\dot{\nu} + \nu \times \mathbf{I}\nu + \Gamma = \mathbf{M^B} - \mathbf{M^D} \tag{2.10}$$

or in component form,

$$\dot{P} = \left(\frac{I_{YY} - I_{ZZ}}{I_{XX}}\right) QR - \frac{I_R}{I_{XX}} Q\Omega + \frac{M_\phi}{I_{XX}} - \frac{A_r}{I_{XX}} P \tag{2.11a}$$

$$\dot{Q} = \left(\frac{I_{ZZ} - I_{XX}}{I_{YY}}\right) PR - \frac{I_R}{I_{YY}} P\Omega + \frac{M_\theta}{I_{YY}} - \frac{A_r}{I_{YY}} Q \tag{2.11b}$$

$$\dot{R} = \left(\frac{I_{XX} - I_{YY}}{I_{ZZ}}\right) PQ + \frac{M_\psi}{I_{ZZ}} - \frac{A_r}{I_{ZZ}} R \tag{2.11c}$$

The state vector of the quadcopter written in the frame {M} is given by

$$\mathbf{X} = \begin{bmatrix}X & Y & Z & \phi & \theta & \psi & U & V & W & P & Q & R\end{bmatrix}^T \tag{2.12}$$

Using the expressions for the states given by equations (2.6), (2.8), (2.9), and (2.11), a quadcopter plant is created in Simulink using the Level-2 S-Function block.

3 Attitude control of quadcopter

Of the 12 states, only six states $\begin{bmatrix}\phi, \theta, \psi, X, Y, Z\end{bmatrix}$ are of primary interest for controller design. The under-actuation problem is solved by using two distinct control loops, inner loop dealing with the attitude variables $\begin{bmatrix}\phi, \theta, \psi, Z\end{bmatrix}$ and the outer loop dealing with the position variables $\begin{bmatrix}X, Y\end{bmatrix}$. So, for the purpose of attitude control, we consider a sub model of the quadcopter plant with the state vector:

$$\mathbf{X_{sub}} = \begin{bmatrix}Z & \phi & \theta & \psi & W & P & Q & R\end{bmatrix}^T \tag{3.1}$$

As the translational motion of the quadcopter depends on the angular orientation, it is ideal to control the rotational behaviour first and then control the translation behaviour. For instance, setting ϕ to a particular value helps in controlling the motion of the quadcopter in the Y direction.

3.1 PID based attitude control

Among all the controllers, the classical PID controller has the simplest structure and is the easiest to implement. The linear PID controller can be applied on the nonlinear system directly by decoupling the attitude dynamics and providing independent control action for each variable. This does not require the model to be linearised about the hover condition, and can thus stabilise the quadcopter even in case of strong perturbations.

The thrust and torque components are obtained from the error in attitude variables using individual PID blocks as shown in the following equations.

$$M_\phi = I_{XX}\left(K_{\phi,P}\, e_\phi + K_{\phi,D}\, \dot{e_\phi} + K_{\phi,I}\int_0^t e_\phi \mathrm{d}t\right) \tag{3.2a}$$

$$M_\theta = I_{YY}\left(K_{\theta,P}\, e_\theta + K_{\theta,D}\, \dot{e_\theta} + K_{\theta,I}\int_0^t e_\theta \mathrm{d}t\right) \tag{3.2b}$$

$$M_\psi = I_{ZZ}\left(K_{\psi,P}\, e_\psi + K_{\psi,D}\, \dot{e_\psi} + K_{\psi,I}\int_0^t e_\psi \mathrm{d}t\right) \tag{3.2c}$$

$$T = m\cos\phi\cos\theta\,\Big(g + K_{Z,P} e_Z + K_{Z,D}\dot{e_Z} + K_{Z,I}\int_0^t e_Z \mathrm{d}t\Big) \tag{3.2d}$$

where $e_\phi = \phi_d - \phi$ and so on.

3.2 LQR based attitude control

LQR is an optimal control technique which drives the system states to the desired value by minimising a cost function. Consider a dynamic system of the form:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned} \tag{3.3}$$

The cost function and the static feedback control law are as follows:

$$J = \int_{t_0}^{\infty} \left\{u^T R u + [x - x_d]^T Q [x - x_d]\right\} dt \tag{3.4a}$$

$$u = -K[x - x_d] \tag{3.4b}$$

where $K = R^{-1}BS$

S is a positive definite matrix obtained by solving the Riccati's algebraic equation.

$$SA + A^T S - SBR^{-1}B^T S + C^T QC = 0 \tag{3.5}$$

To implement the LQR based control, the sub model mentioned in equation (3.1) is considered. The expressions for the state variables are generally of the form

$$\dot{\mathbf{X_{sub}}} = F(\mathbf{X_{sub}}) + G(\mathbf{X_{sub}})\, u \tag{3.6}$$

A linear approximation is obtained by linearising around each state of the subsystem and applying the equilibrium conditions (hovering state) represented by $\mathbf{X_{sub}}$. Linearising around each state causes the controller to be effective even during aggressive flight motions that involve large Euler angles. The coefficients in the linearised plant dynamics are obtained as shown in equations (3.7) and (3.8).

$$\bar{\mathbf{X_{sub}}} = \left[\bar{Z}\,0\,0\,0\,0\,0\,0\,0\right]$$

$$A = \frac{\partial F}{\partial \mathbf{X_{sub}}}|_{\mathbf{X_{sub}}=\bar{\mathbf{X_{sub}}}} \tag{3.7}$$

$$B = G\left(\bar{\mathbf{X_{sub}}}\right) \tag{3.8}$$

Applying the above equations to the quadcopter sub model, we get

$$A = \begin{bmatrix} 0\,0\,0\,0 & 1 & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 1 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 0 & 1 & 0 \\ 0\,0\,0\,0 & 0 & 0 & 0 & 1 \\ 0\,0\,0\,0 & -\frac{A_z}{m} & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & -\frac{A_r}{I_{XX}} & 0 & 0 \\ 0\,0\,0\,0 & 0 & 0 & -\frac{A_r}{I_{YY}} & 0 \\ 0\,0\,0\,0 & 0 & 0 & 0 & -\frac{A_r}{I_{ZZ}} \end{bmatrix} \tag{3.9a}$$

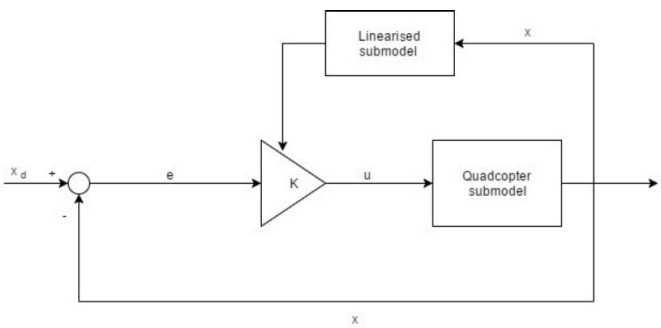
$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{XX}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{YY}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{ZZ}} \end{bmatrix} \tag{3.9b}$$

The matrix C is obtained directly from the dynamic equations as seen in equation (3.3).The value of K is determined by solving the Riccati's algebraic equation, performed by MATLAB using the LQR function:

$$K = LQR\,(A, B, Q, R) \tag{3.10}$$

where $Q = C^T.C$ and R is an identity matrix. The above equation is solved repeatedly varying the weighting matrix Q to find the control gains that give the desired response. The control architecture for LQR is given in Figure 2.

Figure 2 LQR control architecture



3.3 Feedback linearisation Control

Feedback linearisation control (FBL) is a popular nonlinear control approach, where nonlinear systems are algebraically transformed into (fully or partly) linear ones by cancelling the nonlinearities. Most feedback linearisation techniques are based either on input-output linearisation or input-state linearisation. The input-output linearisation scheme is adopted in this work. Input-Output linearisation involves repeated differentiation of the output variables till the input terms appear, the last derivative being the r^{th} one. This helps in obtaining a mapping between the transformed input and the output variables. The concept of dynamic inversion is used to perform feedback linearisation. The output variables not considered above are called residual or internal dynamics. Dynamic inversion need not necessarily yield the internal dynamics stable, which will then require another outer stabilising loop.

Consider a single input single output (SISO) system with state x, input u and output y whose dynamics are given by

$$\dot{x} = f\left(x\right) + g\left(x\right)u \tag{3.11a}$$

$$y = h\left(x\right) \tag{3.11b}$$

The derivative of the output y can be expressed in terms of the lie derivative as shown below:

$$\dot{y} = \frac{\partial h}{\partial x}\left[f\left(x\right) + g\left(x\right)u\right] = L_f h\left(x\right) + L_g h\left(x\right)u \tag{3.12}$$

Following the input-output linearisation scheme, the output y is differentiated continuously until input terms appear in the differential equation. Generally, the ith derivative of y is expressed in terms of lie derivative as:

$$y^{(i)} = L_f^{(i)} h\left(x\right) + L_g L_f^{(i-1)} h\left(x\right)u \tag{3.13}$$

The above equation can be linearised through dynamic inversion choosing the control law as

$$u = \frac{1}{L_g L_f^{(i-1)}}\left[-L_f^{(i)} h\left(x\right) + v\right] \tag{3.14}$$

where v is a linear control input.

This yields a simple linear differential equation

$$y^{(i)} = v \tag{3.15}$$

By choosing appropriate linear control inputs v, the inversion-based control law u can be used to shape the response of the system.

The concepts used for SISO systems can be extended to MIMO systems. In particular, square systems which have the same number of inputs and outputs are considered. For an under-actuated MIMO system like the quadcopter, full feedback linearisation cannot be applied. So, the output function is chosen such that part of the dynamics is linearised. To implement this controller, we consider the sub model M_1 and choose the output function as:

$$y = [Z, \phi, \theta, \psi]^T \tag{3.16}$$

The first derivative with respect to time does not contain input terms, as evident from the following equations. The expressions are obtained from equations (2.8c) and (2.10).

$$\begin{bmatrix} \dot{Z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1\,0 & 0 & 0 \\ 0\,1 & \sin\phi tan\theta & \cos\phi tan\theta \\ 0\,0 & \cos\phi & -\sin\phi \\ 0\,0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} * \begin{bmatrix} W \\ P \\ Q \\ R \end{bmatrix} \tag{3.17}$$

or $\dot{y} = H\gamma$.

The second derivative of all the output variables includes the corresponding input variable, after substituting for $\dot{W}, \dot{P}, \dot{Q}, \dot{R}$ from equations (2.9c) and (2.16).

$$\ddot{y} = \dot{H}\gamma + H\dot{\gamma} \tag{3.18}$$

where $\dot{\gamma} = S + Tu$,

$$S = \begin{bmatrix} -g \\ \left(\frac{I_{YY} - I_{ZZ}}{I_{XX}}\right) QR - \frac{I_R}{I_{XX}} Q\Omega - \frac{A_r}{I_{XX}} P \\ \left(\frac{I_{ZZ} - I_{XX}}{I_{YY}}\right) PR - \frac{I_R}{I_{YY}} P\Omega - \frac{A_r}{I_{YY}} Q \\ \left(\frac{I_{XX} - I_{YY}}{I_{ZZ}}\right) PQ - \frac{A_r}{I_{ZZ}} R \end{bmatrix}$$

and

$$T = \begin{bmatrix} \frac{1}{m}\cos\theta\cos\phi & 0 & 0 & 0 \\ 0 & \frac{1}{I_{XX}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{YY}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{ZZ}} \end{bmatrix}$$

Following the above substitution and rearranging the terms, we get

$$\ddot{y} = \alpha + \beta u \tag{3.19}$$

The total relative degree is eight, which is equal to the number of states of the sub model. This guarantees the stability of the subsystem which is sufficient for attitude control. The control law is written based on the general form

$$u = \beta^{-1}(-\alpha + v) \tag{3.20}$$

with $v = [v_1 \ \ v_2 \ \ v_3 \ \ v_4]^T$ being the auxiliary control input. The resulting second order linear system is

$$\ddot{y} = v \tag{3.21}$$

By using the standard PD controller with a feed forward acceleration term, the error in attitude $e_y = y_d - y$ is forced to converge to zero for a desired attitude y_d.

$$v = \ddot{y_d} + K_p e_y + K_d \dot{e_y} \tag{3.22}$$

The control gains K_p and K_d (positive-definite matrices) are obtained by tuning the PD controller such that the poles of the closed loop error dynamics of the subsystem (3.16,3.17) are in the left half of s-plane.

$$\ddot{e_y} + K_p e_y + K_d \dot{e_y} = 0 \tag{3.23}$$

The control architecture for Feedback Linearisation is given in Figure 3.

4 Simulation results

All the three proposed controllers are tested on the dynamic model developed in the Simulink environment. The values for the parameters in the quadcopter model are taken from Luukkonen (2011) and listed in Table 1.

Table 1 Parameter values for the quadcopter model

Parameter	*Value*	*Unit*
g	9.8	m/s^2
L	.225	m
m	.468	kg
K	2.980 $\cdot 10^{-6}$	
B	0.114 $\cdot 10^{-6}$	
I_{XX}	4.856 $\cdot 10^{-3}$	kg s^2
I_{YY}	4.856 $\cdot 10^{-3}$	kg s^2
I_{ZZ}	8.801 $\cdot 10^{-3}$	kg s^2
I_R	3.357 $\cdot 10^{-5}$	kg s^2

The desired attitude commands are provided from a block, which contains step functions for each of the variables. The step function starts with an initial condition of 10° for the three angles and 5m for height and falls to 0° for the angles and 3m

Figure 3 Feedback linearisation control architecture

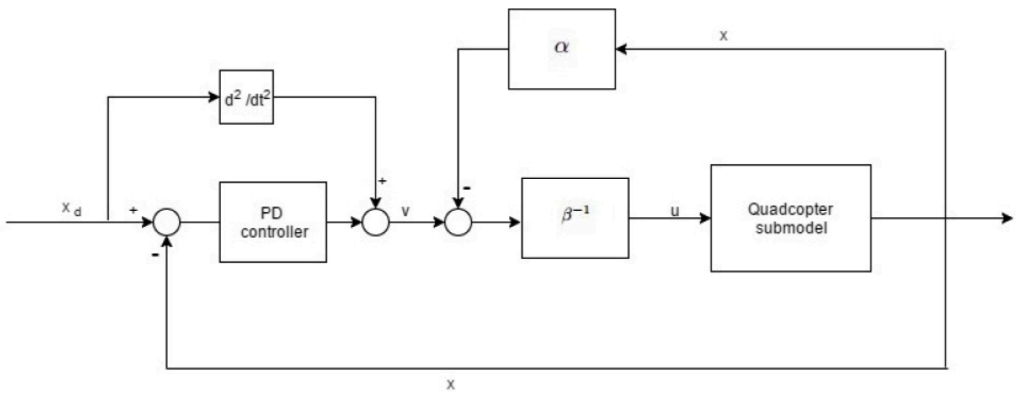


Figure 4 PID controller step response (see online version for colours)

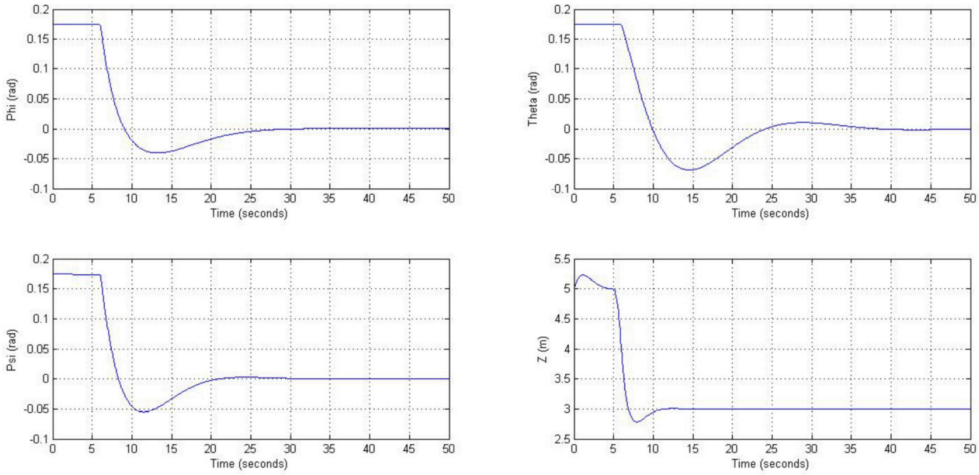


Figure 5 LQR controller step response (see online version for colours)

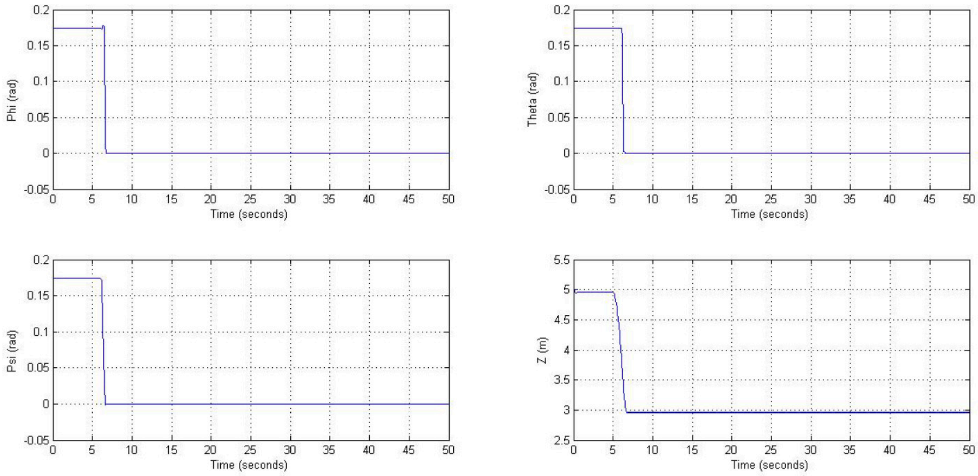
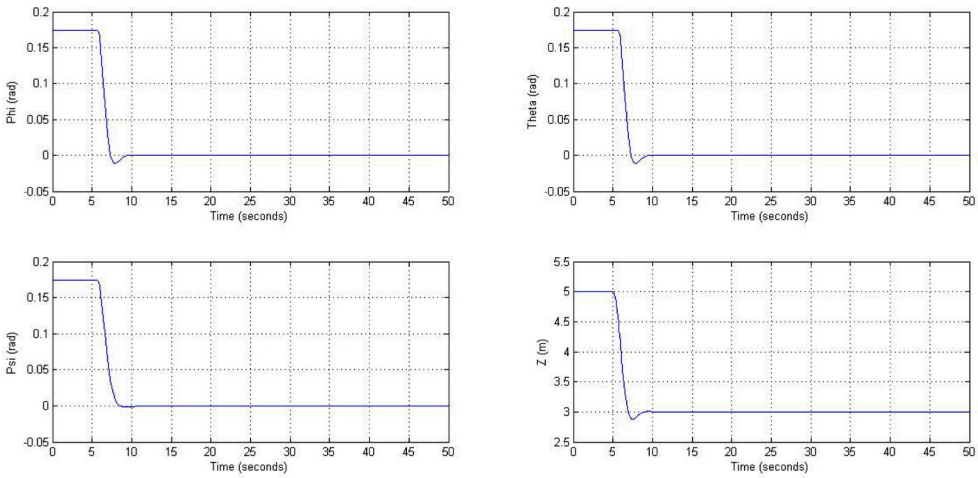


Figure 6 FBL controller step response (see online version for colours)



for the height with a step time of 5 s. The values of all the gains for the three controllers are listed in Tables 2–4. Figures 4–6 show the step response of the three controllers.

Figure 7 clearly shows that the combination of Feedback Linearisation and PD controllers has the best performance in stabilising the height. It shows minimum undershoot and remains steady before the step input is given. The PID controller has the least fall time but high undershoot caused by the step input. LQR shows a very poor performance in this case, as it saturates below the desired height. Table 5 gives a quantitative comparison between the controllers based on the parameters mentioned above.

Looking at the step response for ϕ as seen in Figure 8, it is evident that the combination of Feedback Linearisation and

Figure 7 Comparison between PID, LQR and FBL based on Z step response (see online version for colours)

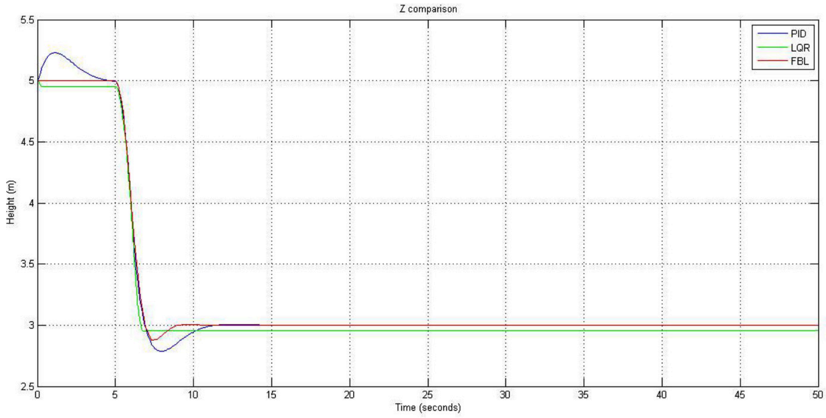
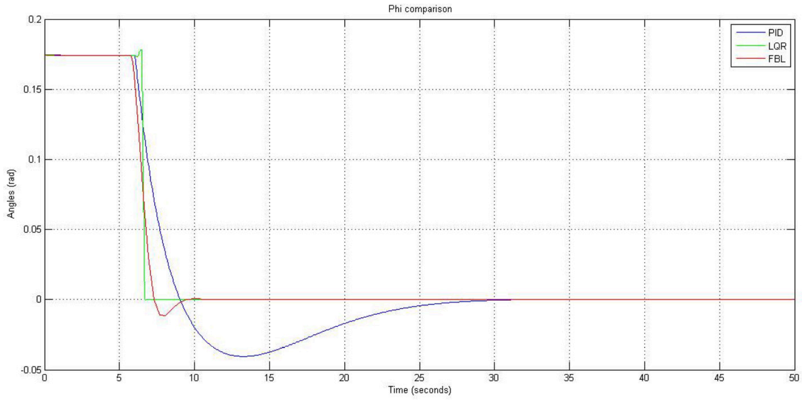


Figure 8 Comparison between PID, LQR and FBL based on ϕ step response (see online version for colours)



PD controllers shows the fastest response to the step input with minimum fall time and undershoot. Though LQR shows the least fall time and undershoot as per Table 6, there is a delay in response to the step input. PID controller shows comparatively poor performance as the fall is very gradual and it takes a long time to settle. The same performance order is seen when the step response for θ is considered, as seen in Figure 9 and Table 7.

Table 2 Gain values for PID controller

Controller	K_P	K_D	K_I
Z	15	10	10
ϕ	6	1.75	1.5
θ	5	3	3
ψ	6	1.75	1.5

Table 3 Gain values for LQR controller

Controller	K_1	K_2
Z	100	9.427
ϕ	10	0.1703
θ	10	0.1703
ψ	10	0.0267

Analysing the step response of ψ based on Figure 10 and Table 8, LQR has a fall time significantly lower than the other two controllers. The combination of FBL and PD controllers has a higher fall time, but same percentage of undershoot. The PID controller lags behind in both parameters.

Table 4 Gain values for Feedback Linearisation controller

Controller	K_P	K_D
Z	3	2
ϕ	3	2
θ	3	2
ψ	2	2

Table 5 Characteristic parameters to a step input for Z

Z	*PID*	*LQR*	*FBL*
Fall time (seconds)	0.68	1.020	1.194
Percentage Undershoot (m)	20.732	0.515	5.851

Table 6 Characteristic parameters to a step input for ϕ

ϕ	*PID*	*LQR*	*FBL*
Fall time (seconds)	2.276	0.131	1.060
Percentage Undershoot (m)	22.84	0.515	6.989

Finally, a comparison can also be made based on the computation time in the simulation environment. All the three controllers are simulated for 50s in Simulink. The computation

Figure 9 Comparison between PID, LQR and FBL based on θ step response (see online version for colours)

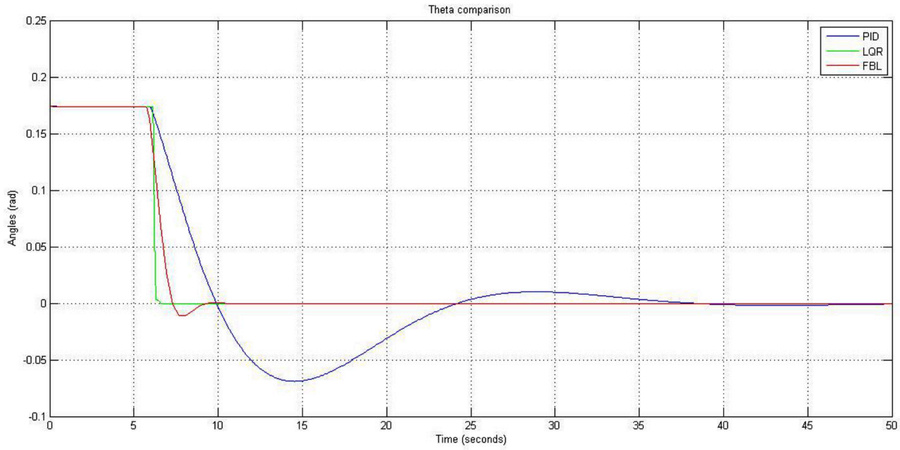
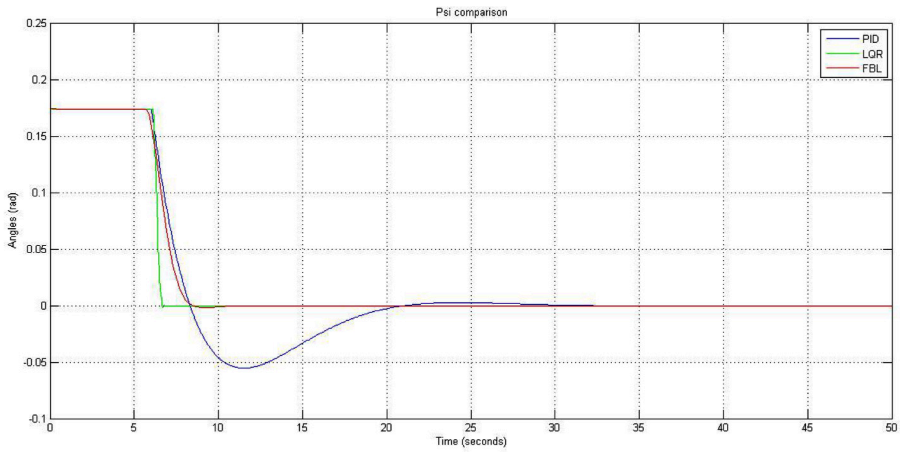


Figure 10 Comparison between PID, LQR and FBL based on ψ step response (see online version for colours)



time is least for the combination of Feedback linearisation and PD controller, followed by the PID controller. LQR has very high computation time owing to the dynamic calculation of control gains.

Table 7 Characteristic parameters to a step input for θ

θ	*PID*	*LQR*	*FBL*
Fall time (seconds)	3.050	.134	1.074
Percentage Undershoot (m)	38.194	0.505	5.851

Table 8 Characteristic parameters to a step input for ψ

ψ	*PID*	*LQR*	*FBL*
Fall time (seconds)	1.748	0.355	1.574
Percentage Undershoot (m)	32.667	1.531	1.531

Based on the above analysis, the following conclusions can be made:

- The combination of Feedback Linearisation and PD controller has the best overall performance for the four attitude variables, with fast response and minimum undershoot and settling time.
- Linear Quadratic Regulator has the least fall time and undershoot in all the cases, but there exists a small delay in control action after the step input is given.
- The PID controller lags behind the other two controllers and shows a very slow response for the three angles. This is due to the direct application of a linear controller on the nonlinear system.

5 Conclusion

This paper presents a comparative study on linear and non-linear control techniques used for the attitude control of quadcopters. The dynamic model of the quadcopter is derived using the Newton-Euler approach to test the performance of the proposed control algorithms. The step response of all three controllers are analysed in terms of fall time, percentage undershoot and computation time. The best results are obtained while using the Feedback Linearisation control technique.

In our future work, we aim to address the challenge of vertical take-off and landing which can lead to increased manoeuvrability and flight capability. In this work, all the simulations were carried out assuming that the quadcopter's entire motion occurs at a sufficient height above the ground, and take-off and landing are not performed by the quadcopter. Another challenge is that the presented mathematical model does not consider external disturbances like wind velocities and ground effects due to the difficulty in modelling these effects. The controllers should be made robust in that they deal effectively with the external disturbances neglected during modelling. Designing a controller that can combat the failure of one or more rotors is a further step in this direction.

References

Bonna, R. and Camino, J.F. (2015) 'Trajectory tracking control of a quadrotor using feedback linearisation', *Proc. of the XVII International Symposium on Dynamic Problems of Mechanics DINAME-2015.-2015*, February.

Bouabdallah, S., Noth, A. and Siegwart, R. (2004) 'PID vs LQ control techniques applied to an indoor micro quadrotor', *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004)*, IEEE, Vol. 3, pp.2451–2456.

Das, A., Subbarao, K. and Lewis, F. (2009) 'Dynamic inversion with zero-dynamics stabilisation for quadrotor control', *IET Control Theory and Applications*, Vol. 3, No. 3, pp.303–314.

Dikmen, Ä.C., Arisoy, A. and Temeltas, H. (2009) 'Attitude control of a quadrotor', *4th International Conference on Recent Advances in Space Technologies, 2009. RAST'09*, June, IEEE, pp.722–727.

Lee, D., Jin Kim, H. and Sastry, S. (2009) 'Feedback linearisation vs. adaptive sliding mode control for a quadrotor helicopter', *International Journal of Control, Automation and Systems*, Vol. 7, No. 3, pp.419–428.

Li, L., Sun, L. and Jin, J. (2015) 'Survey of advances in control algorithms of quadrotor unmanned aerial vehicle', *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, October, IEEE, pp.107–111.

Luukkonen, T. (2011) *Modelling and Control of Quadcopter*, Independent Research Project in Applied Mathematics, Espoo.

Sabatino, F. (2015) *Quadrotor Control: Modeling, Nonlinear Control Design, and Simulation*.

Slotine, J.J.E. and Li, W. (1991) *Applied Nonlinear Control (Vol. 199, No. 1)*, Prentice-Hall, Englewood Cliffs, NJ.