# Chapter 3

# Nonlinear Controller Design

In this chapter, mathematical background of nonlinear control design is presented. Initially, the technique of nonlinear dynamic inversion is briefly discussed. Using this technique, we synthesis controller for a quadrotor. The controller is able to stabilize a quadrotor if it is perturbed from equilibrium. The controller is also modified to track a desired trajectory. For designing controller, two architectures are proposed using attitude and angular rate control concepts. The performance of the proposed design is compared with linear design mainly with proportional-integral-derivative (PID) controller.

## 3.1 Dynamic Inversion

Dynamic inversion is a well established nonlinear control technique to design controllers for nonlinear systems. The nonlinear control design is popular as it get rid of tedious gain scheduling procedure in the linear design and it is applicable for the full range of flight envelop. Consider a trajectory of a nonlinear system as shown in

Fig. 3.1. This system looks nonlinear with respect to the coordinate frame XY. Let us consider that we transform this co-ordinate XY to the new frame $\hat{X}\hat{Y}$, which is behaving in the same nonlinear way as the system behaves. Now, in this new frame $\hat{X}\hat{Y}$ the system always looks linear. Hence, in this new frame $\hat{X}\hat{Y}$, linear control strategies could be used for controlling the behavior of the system.

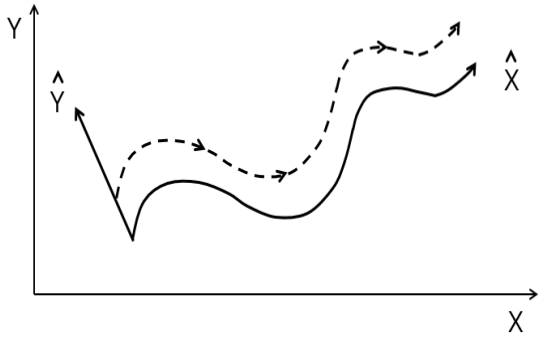


Figure 3.1: DI frame transformation

One relative easy way to design a controller by choosing stable linear error dynamics in the frame $\hat{X}\hat{Y}$. This is the basic idea behind dynamic inversion control strategy. In order to achieve this, the dynamics of the plant is used to determine control.

### 3.1.1 Mathematical Details

Consider a nonlinear system as follows,

$$\begin{aligned} \dot{X} &= f(X, U) \\ Y &= h(X) \end{aligned} \tag{3.1}$$

where X is the state $\in \mathbb{R}^n$, U is the control $\in \mathbb{R}^m$, Y is the output $\in \mathbb{R}^p$, m,n and p are positive scalars.

In this work, we consider the following class of nonlinear system.

$$\begin{aligned} \dot{X} &= f(X) + [g(X)]\, U \\ Y &= h(X) \end{aligned} \tag{3.2}$$

This form is known as a control affine form as control appears linearly in the state equation. Our interest lies in developing a controller for a quadrotor. It can be observed that the quadrotor dynamics derived in Chapter 2 can be written in the form of Eq.(3.2).

The control objective is to track a smooth desired output $Y^*$ as time tends to infinity. Mathematically it can be stated as ,

$$lim_{t\to\infty}\, Y \to Y^*(t) \tag{3.3}$$

Suppose if the desired output $Y^* = 0$, then the problem will be known as stabilization problem.

### 3.1.2 Control Synthesis

As the objective is to tack the output, we write the output dynamics as,

$$\dot{Y} = \left[\frac{\partial h}{\partial X}\right] \dot{X} \tag{3.4}$$

Substituting $\dot{X}$ from Eq.(3.2) we get,

$$\dot{Y} = \left[\frac{\partial h}{\partial X}\right] [f(X) + [g(X)]\, U] \tag{3.5}$$

Let us define $f_Y(X) \triangleq \left[\frac{\partial h}{\partial X} f(X)\right]$ and $g_Y(X) \triangleq \left[\frac{\partial h}{\partial X}\right] [g(X)]$, the above equation can be re-written as,

$$\dot{Y} = f_Y(X) + g_Y(X)\; U \tag{3.6}$$

The control is synthesized by choosing the first order stable error dynamics as follows,

$$\dot{E} + KE = 0 \tag{3.7}$$

where,$E(t) \triangleq [Y(t) - Y^*(t)]$ and K is a fixed gain matrix. One of relative easy way to choose K as $\tau_i$ (i.e) K = diag$\left[\frac{1}{\tau_1}, \frac{1}{\tau_2}....\frac{1}{\tau_i}....\frac{1}{\tau_p}\right]$, here $\tau_i$ is the time constant [38]. Substituting Eq.(3.6) in Eq.(3.7) and carrying out the necessary algebra, we obtain

$$U = [g_Y(X)]^{-1} (\dot{Y}^* - K(Y - Y^*) - f_Y(X)) \tag{3.8}$$

Next, we discuss the important steps involve in designing the controller.

- The matrix $g_Y(X)$ has to be invertible, i.e, it has to be a non singular matrix.
- Number of control inputs has to be either equal or more than the number of output.
- The relative degree of the system should be well defined i.e on successive differentiation of output dynamics (Y), the control variable (U) should appear in the equation.
- A stable error dynamics should be chosen such that the internal dynamics of the system is always stable. Asymptotic stability of zero dynamics is sufficient for local input-to-state stability of internal dynamics.
- The plant parameters used in the control design should the same as the actual plant i.e the plant parameters should be precisely known.

## 3.2 Control Design for Quadrotor

For a quadrotor, we have four control inputs (four rotors) and six states needs to be controlled. Thus, the quadrotor is an under actuated system. In the previous section, it is clearly specified that a system with nonlinear dynamic inversion must have the number of output variable either be equal or more than the number of control inputs. Keeping this in mind, we choose four desired states as output that include position and heading. Thus, the objective of the controller is to track a desired trajectory in three dimensional space with a specific heading angle. The remaining state variables form the internal dynamics of a quadrotor. In order to achieve stanilization and/or tracking, the control design considers two loops: outer loop and inner loop. In the outer loop, we consider translational dynamics as it is slower than the rotational dynamics. In the inner loop we consider rotational dynamics. Next, we present the standard control architecture.

### 3.2.1 Control Architecture

The standard control architecture is given in Fig. 3.2. First, the error in the position of the quadrotor is calculated. Then, the position controller computes the required thrust and angle set points for the attitude controller. The position controller is designed using dynamic inversion approach. Next, the attitude controller determines the required moments to achieve the angle set points provided by the position controller. The thrust and moment data are processed in the control allocation block where these values are mapped as pwm values as described in Chapter 2. These pwm

values are directly fed into the brushless DC motor to achieve the desired position and heading.

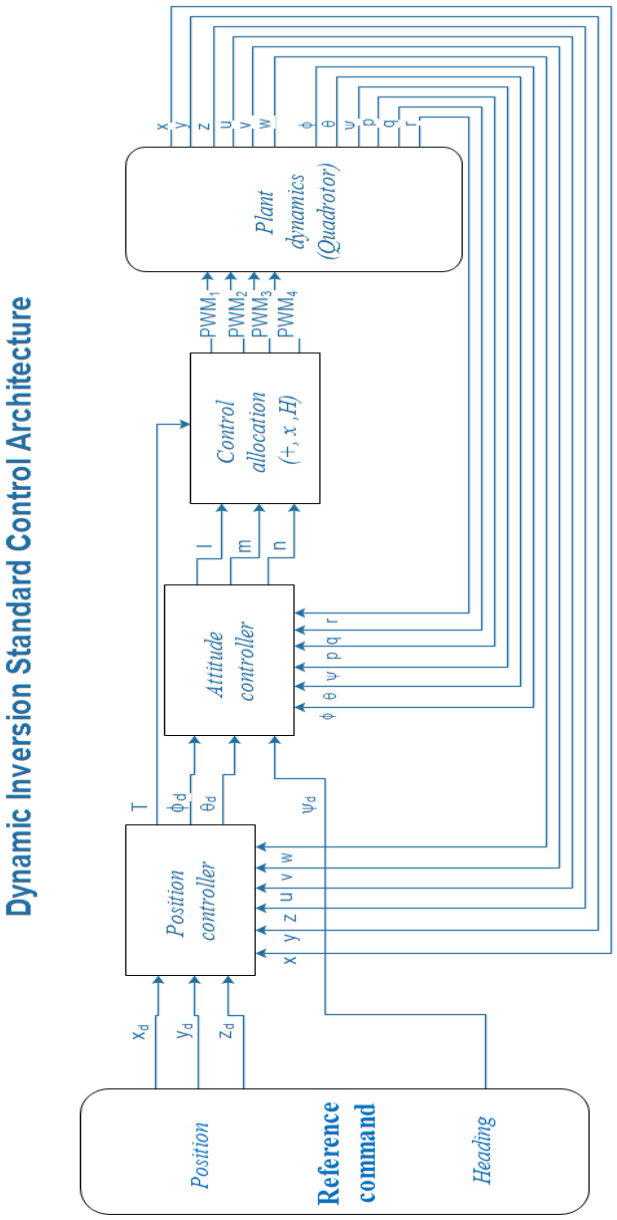


Figure 3.2: Dynamic inversion standard control architecture

## 3.3 Control Synthesis

The tracking error in position can be represented as,

$$E = P - P_d \tag{3.9}$$

where P and $P_d$ are the current and desired position vector of a quadrotor in the global inertial frame. Using the position error, we design the outer loop details of which are given next.

### 3.3.1 Outer Loop

As the relative degree is two, we choose second order stable error dynamics to synthesis control as follows.

$$\ddot{E} + 2\zeta\omega_n\dot{E} + \omega_n^2\ E = 0 \tag{3.10}$$

Substituting Eq.(3.9) in Eq.(3.10), we get

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \\ \ddot{z}_d \end{bmatrix} + 2\zeta\omega_n \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{y}_d - \dot{y} \\ \dot{z}_d - \dot{z} \end{bmatrix} + \omega_n^2 \begin{bmatrix} x_d - x \\ y_d - y \\ z_d - z \end{bmatrix} \tag{3.11}$$

Here $\zeta$ and $\omega_n$ are the damping ratio and natural frequency. $K_1 = 2\zeta\omega_n$ and $K_2 = \omega_n^2$. Here $K_1 = diag\left[\frac{2\zeta}{\tau_1}, \frac{2\zeta}{\tau_2} .... \frac{2\zeta}{\tau_i} .... \frac{2\zeta}{\tau_p}\right]$ and $K_2 = diag\left[\frac{1}{\tau_1^2}, \frac{1}{\tau_2^2} .... \frac{1}{\tau_i^2} .... \frac{1}{\tau_p^2}\right]$, where $\tau_i$ is time constant.

After carrying out the necessary algebra, the required thrust and desired roll and

pitch angles are given as follows.

$$
\begin{aligned}
T &= M\sqrt{\ddot{x}^2 + \ddot{y}^2 + (g - \ddot{z})^2} \\
\phi_d &= sin^{-1}(u_x sin\psi_d - u_y cos\psi_d) \\
\theta_d &= sin^{-1}\frac{u_x cos\psi_d + u_y sin\psi_d}{cos\phi_d} \\
u_x &= -M\ddot{x}/T \\
u_y &= -M\ddot{y}/T \\
\psi_d &\quad \text{is provided by the user.}
\end{aligned}
\tag{3.12}
$$

where $T, (\phi_d, \theta_d, \psi_d)$ are the required thrust and Euler angle set-points. Next, we discuss inner loop design.

### 3.3.2 Inner Loop

For designing inner loop controller, we choose the second order stable error dynamics as follows.

$$
\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \ddot{\phi}_d \\ \ddot{\theta}_d \\ \ddot{\psi}_d \end{bmatrix} + 2\zeta\omega_n \begin{bmatrix} \dot{\phi}_d - \dot{\phi} \\ \dot{\theta}_d - \dot{\theta} \\ \dot{\psi}_d - \dot{\psi} \end{bmatrix} + \omega_n^2 \begin{bmatrix} \phi_d - \phi \\ \theta_d - \theta \\ \psi_d - \psi \end{bmatrix} \tag{3.13}
$$

The Eulerian angular rates in the Eq.(3.13) is obtained by transforming the body rates to Eulerian rates using the rotation matrix as shown in Eq.(3.14).

The onboard sensors measure the rate of rotation of a quadrotor in the body frame.

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.14}
$$

Thus, the Eulerian angular acceleration computed using the error dynamics is transformed to obtain body angular acceleration as follows,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & cos\theta\dot{\theta} \\ 0 & sin\phi\dot{\phi} & -sin\phi sin\theta\dot{\theta} - cos\phi cos\theta\dot{\phi} \\ 0 & cos\phi\dot{\phi} & -sin\theta cos\phi\dot{\theta} + sin\phi cos\theta\dot{\phi} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.15}$$

This body angular acceleration relation in Eq.(3.15) is obtained by first inverting the Eq.(2.18) and differentiating with respect to time.

Next, using this body angular acceleration the required moments are determined as shown in the Eq.(3.19). This equation is obtained by inverting Eq.(2.31).

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq \end{bmatrix} \tag{3.16}$$

Here l, m and n are the required rolling, pitching and yawing moments for tracking/stabilization.

In experiments, the required thrust and moments generated from the outer and inner loop are converted into pwm signals using the relation in Eq.(2.37) and fed into the BLDC motors to achieve the desired position and heading specified by an user. Next, the control architecture of dynamic inversion attitude angle based design is provided in Fig. 3.3.

### 3.3.3 Control Architecture

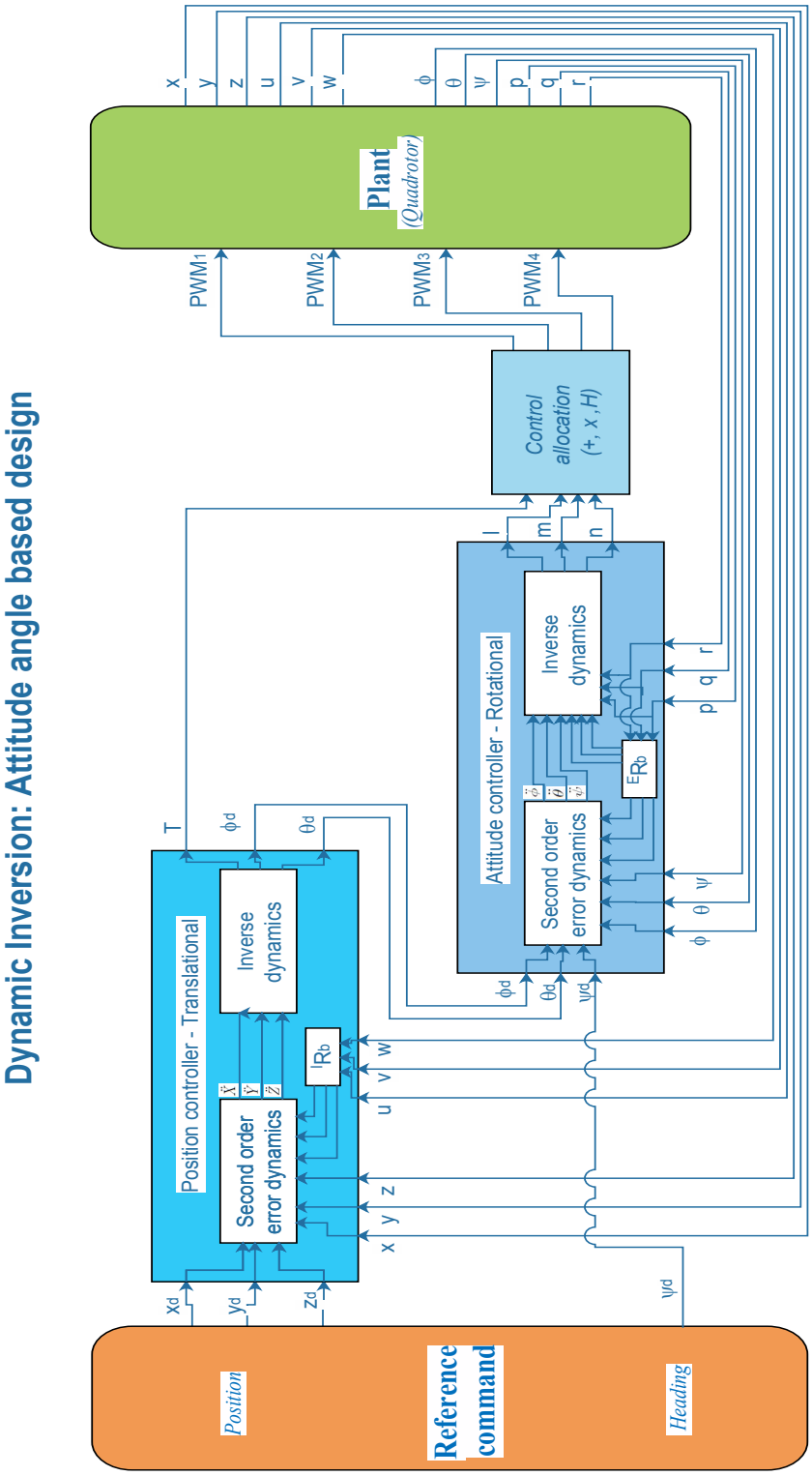


Figure 3.3: Dynamic inversion - attitude angle based design

### 3.3.4 Numerical Results

In this section, we demonstrate the performance of the flight control law through numerical simulations. The algorithm is implemented in MATLAB R2013a and initial conditions are tabulated in Table. 3.1.

| State | x | y | z | $\phi$ | $\theta$ | $\psi$ | u | v | w | p | q | r |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | m | m | m | deg | deg | deg | m/s | m/s | m/s | deg/s | deg/s | deg/s |
| Value | 1 | 1 | 1 | 50 | 50 | 10 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 3.1: Initial conditions

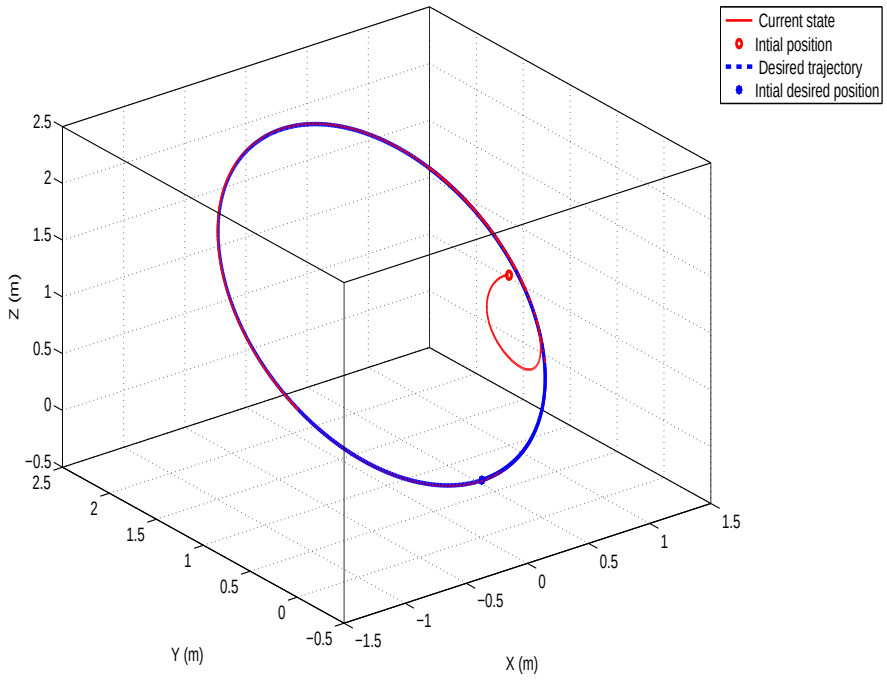


Figure 3.4: Position - 3D circular trajectory tracking

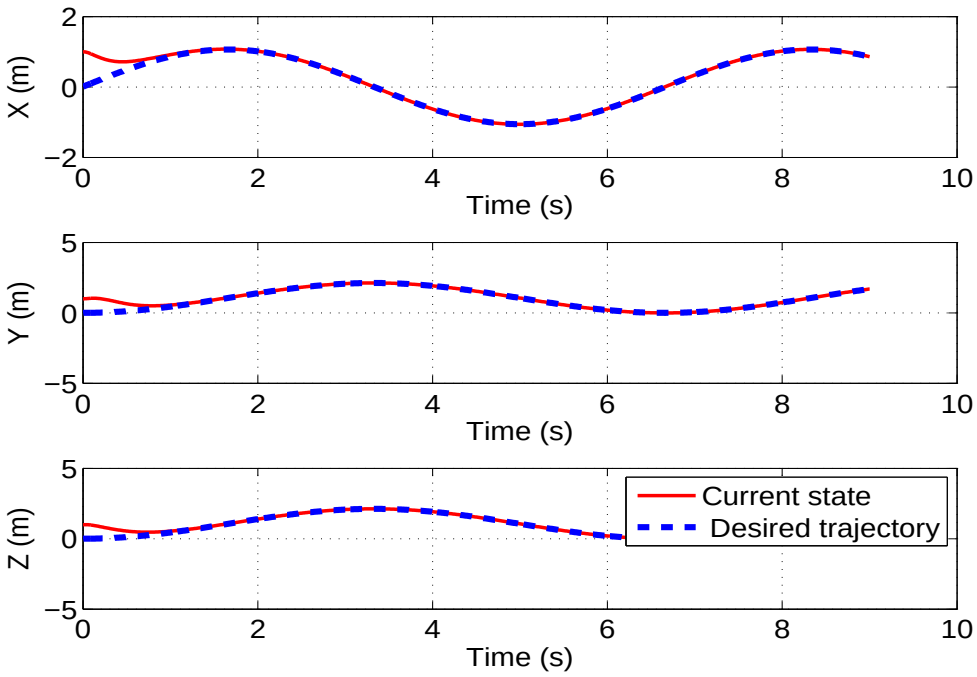


Figure 3.5: Position - circular trajectory tracking

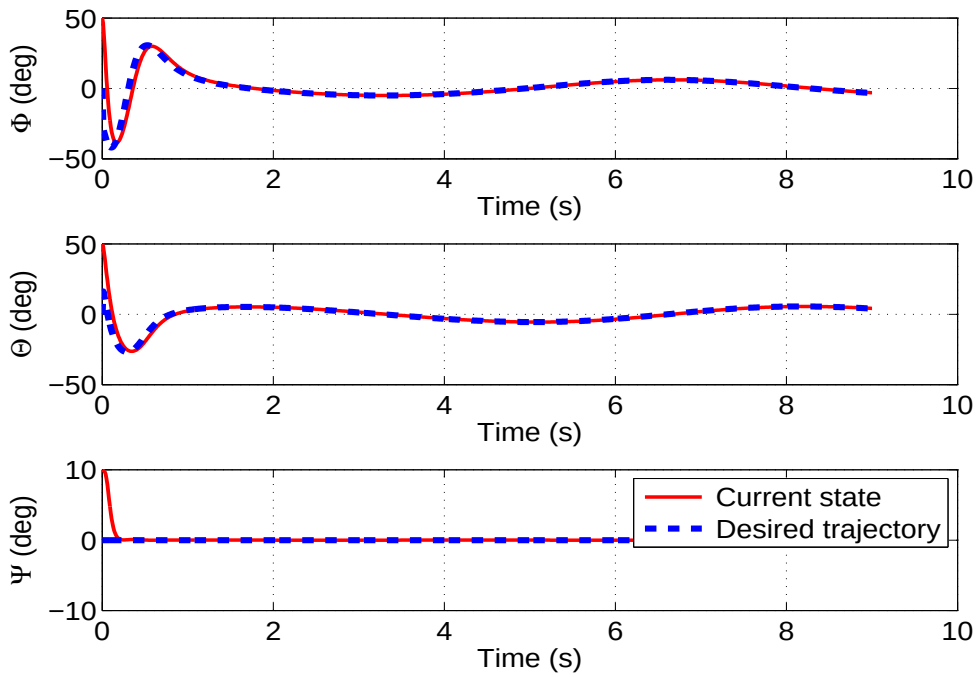


Figure 3.6: Attitude - circular trajectory tracking

It can be seen from Figs. 3.4-3.6 that the desired trajectory is tracked well, also

the position controller generated a smooth attitude set-points and these are tracked well by attitude controller.

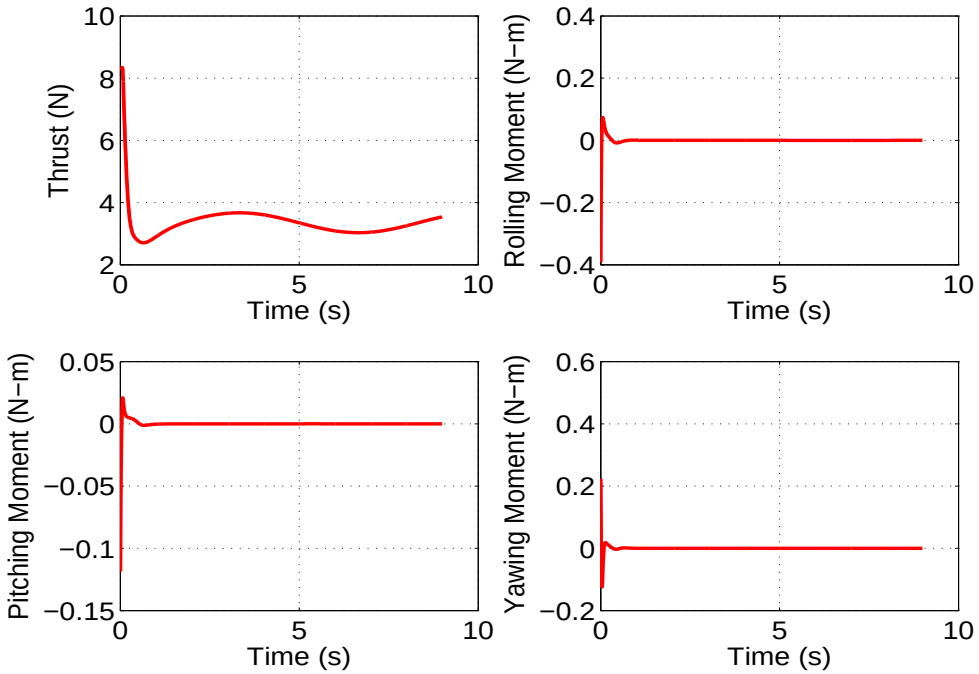


Figure 3.7: Commanded control inputs - circular trajectory tracking

From Fig. 3.7, it can be seen that the required control input generated using the above strategy is smooth.

An extensive simulation was carried out for various reference trajectory frequencies ranges from 4.5 (rad/s) to 11.5 (rad/s). It is observed that the performance of the proposed control law degrades when a reference trajectory of high frequency ($\omega = 9.85(rad/s)$) is demanded. This is clear from the simulation result presented in Figs. 3.8-3.10.

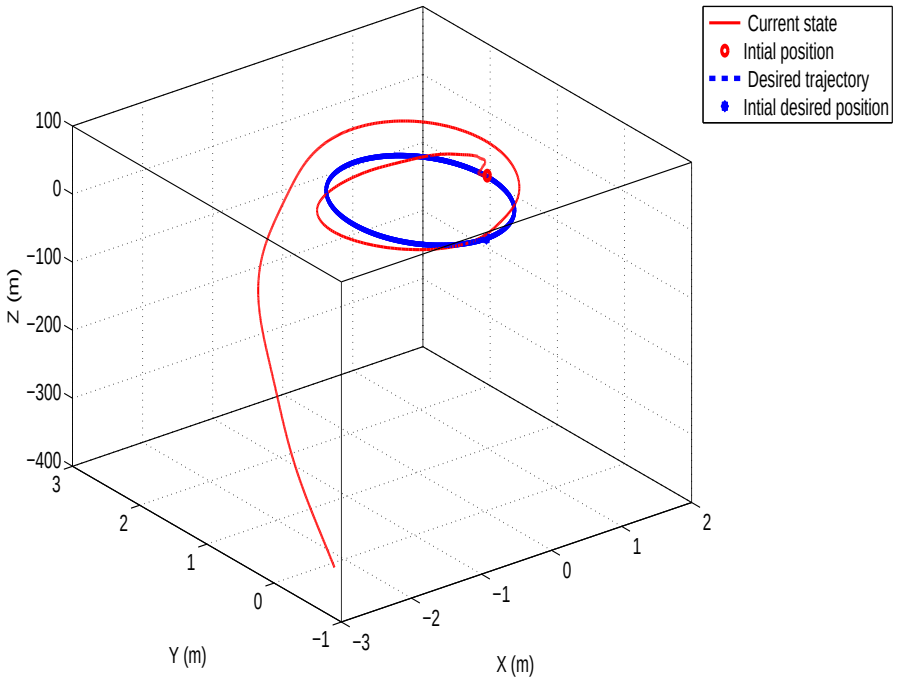


Figure 3.8: Position - high frequency trajectory tracking for control design based on position and attitude tracking

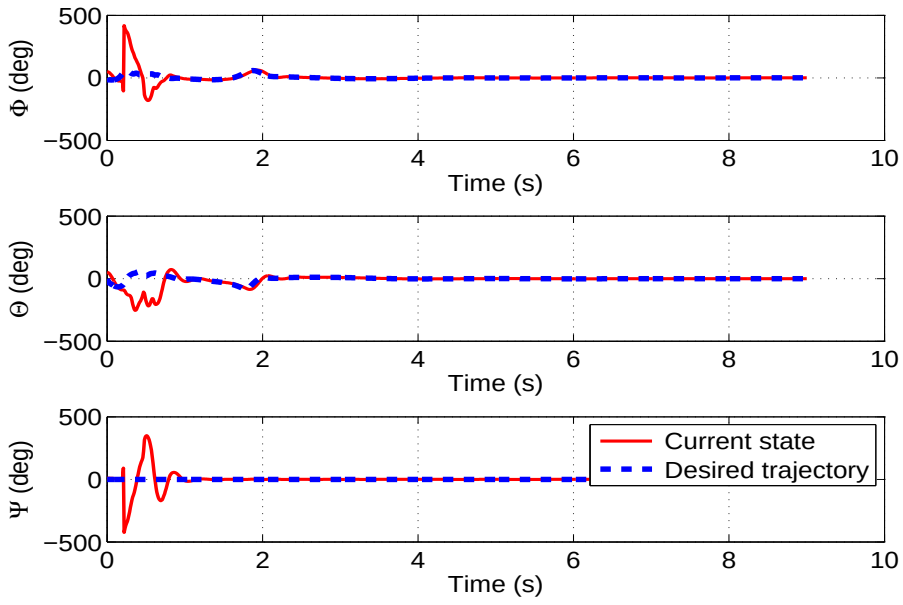


Figure 3.9: Attitude - high frequency trajectory tracking for control design based on position and attitude tracking

From the Fig. 3.8, it can be clearly seen that the controller performance has degraded leading to large error in trajectory tracking. Also, Fig. 3.10 shows that the control has reached saturation limit. Thus the commanded trajectory is no longer tracked properly.

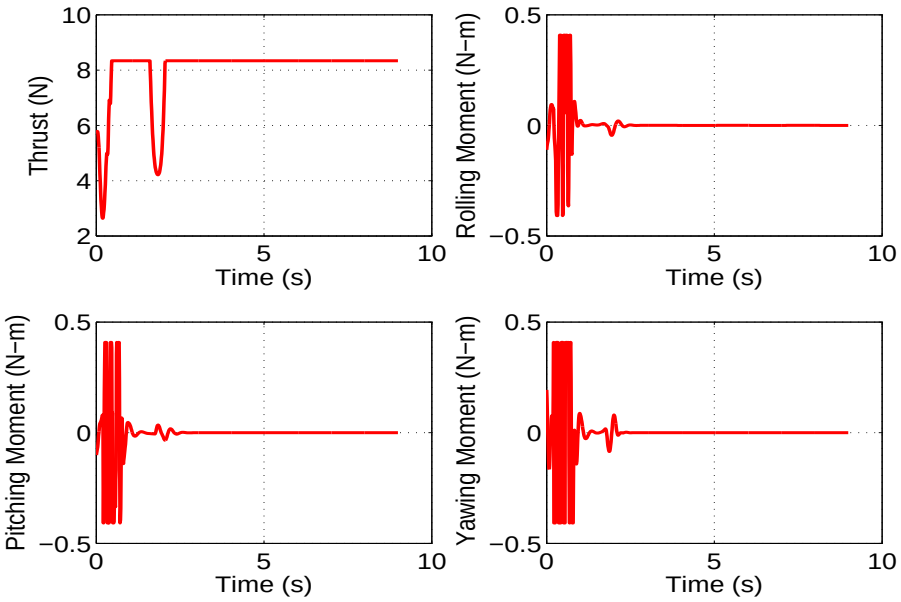


Figure 3.10: Commanded control inputs - high frequency trajectory tracking for control design based on position and attitude tracking

The performance degradation is because the quadrotor fails to maintain the desired attitude at the faster rate as can be seen in Fig. 3.9. To overcome this problem, we propose a new design in the next section.

Comment: Moreover, it can be seen from the Eq.(3.14), Eq.(3.15) that there are many trigonometric terms in the inner loop. The onboard controller (Navstik, Arduini, Odroid, Pixhawk etc.) usually computes the trigonometric terms by approximating them as a finite series, therefore it cause some error that may further degrade the performance. This problem will also be addressed in the next section.

## 3.4 Control Design based on Position and Angular Rate Tracking

In this section, we develop alternative strategy to stabilize a quadrotor. In the proposed strategy, the design of outer loop remains the same whereas the design of inner loop is altered. In the previous design, we stabilized attitude angles by choosing second order stable error dynamics. In the new design, we stabilize body angular rates by choosing the first order stable error dynamics. As the aim is to track the desired body rates, a better tracking performance is expected.

### 3.4.1 Control Synthesis

For designing the controller, we use the dynamic inversion technique, however, instead of tracking attitude angles we track body angular rate in the inner loop. Since the outer loop design remains same, here it is omitted for brevity.

The control is synthesized by choosing the first order stable error dynamics as follows,

$$\begin{bmatrix} \dot{p} - \dot{p}_d \\ \dot{q} - \dot{q}_d \\ \dot{r} - \dot{r}_d \end{bmatrix} + \begin{bmatrix} K_p & 0 & 0 \\ 0 & K_q & 0 \\ 0 & 0 & K_r \end{bmatrix} \begin{bmatrix} p - p_d \\ q - q_d \\ r - r_d \end{bmatrix} = 0 \tag{3.17}$$

Here, $K_p = \frac{1}{\tau_1}, K_q = \frac{1}{\tau_2}, K_r = \frac{1}{\tau_3}$ where, $\tau_1, \tau_2$and $\tau_3$ are time constants. Assuming desired body angular accelerations to be zero (i.e $\dot{p}_d = \dot{q}_d = \dot{r}_d = 0$) and rearranging the Eq.(3.17), we get

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -K_p & 0 & 0 \\ 0 & -K_q & 0 \\ 0 & 0 & -K_r \end{bmatrix} \begin{bmatrix} p - p_d \\ q - q_d \\ r - r_d \end{bmatrix} \tag{3.18}$$

Substituting Eq.(3.18) in Eq.(3.19) (from the equations of motion, Eq.(2.35)), we get the required control moments.

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq \end{bmatrix} \tag{3.19}$$

In experiments, the required force and moments are converted to pwm signals using Eq.(2.37). Next, the control architecture for this alternative design is shown in Fig. 3.11

### 3.4.2 Control Architecture

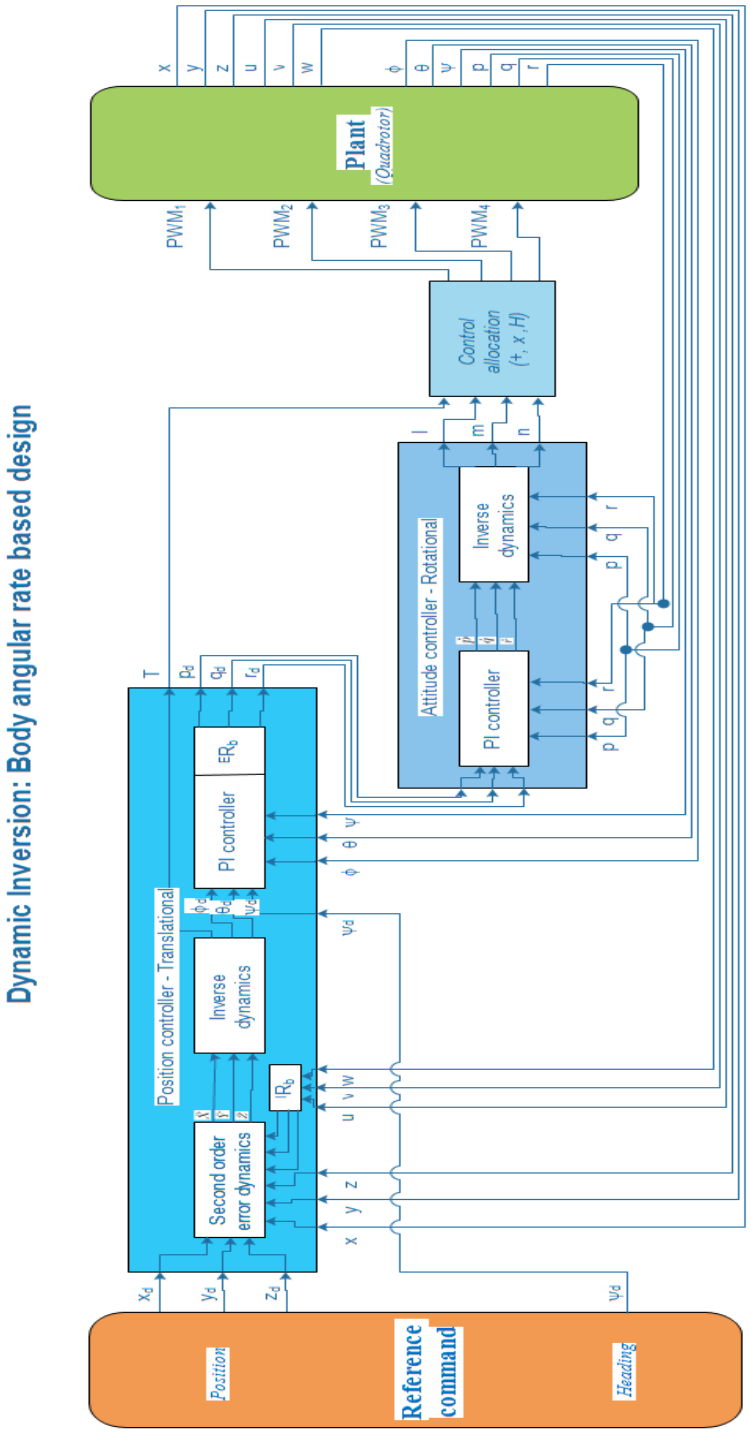


Figure 3.11: Control architecture for control design based on position and angular rate tracking

### 3.4.3 Numerical Results

In this subsection, we evaluate the performance of our algorithm through 3D circular tracking. We also compare the performance with the previous design for tracking trajectories with high frequency ($\omega = 9.85(rad/s)$). The initial conditions remain unaltered as present in Table. 3.1.

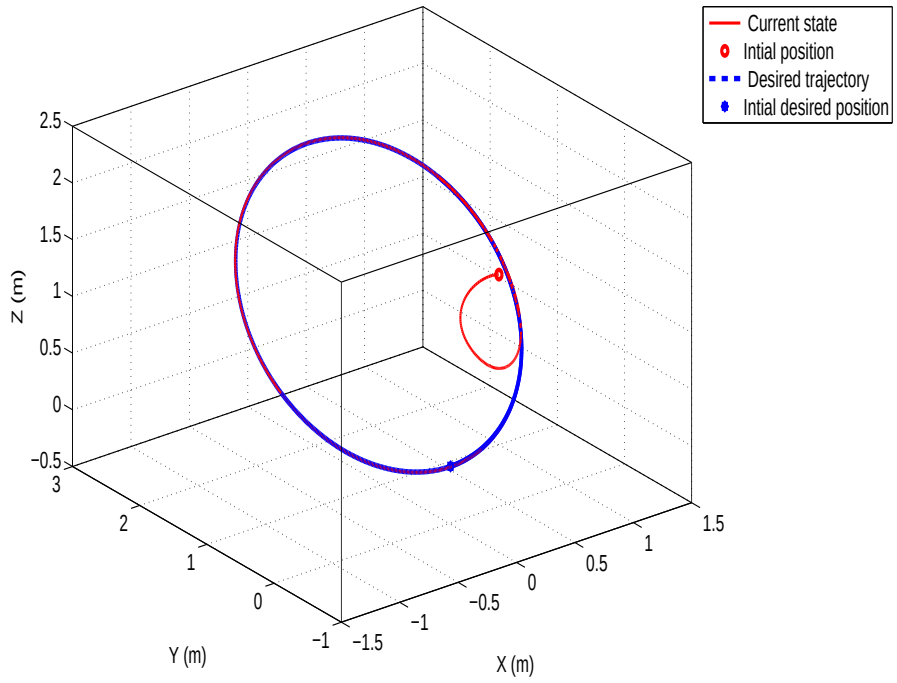


Figure 3.12: 3D Position - trajectory tracking for control design based body angular rates

It can be seen from Fig. 3.12-3.14 that the commanded trajectory is tracked well. The performance of alternative design is tested extensively in simulation for various trajectory frequencies ranges from 4.5 (rad/s) to 11.5 (rad/s).

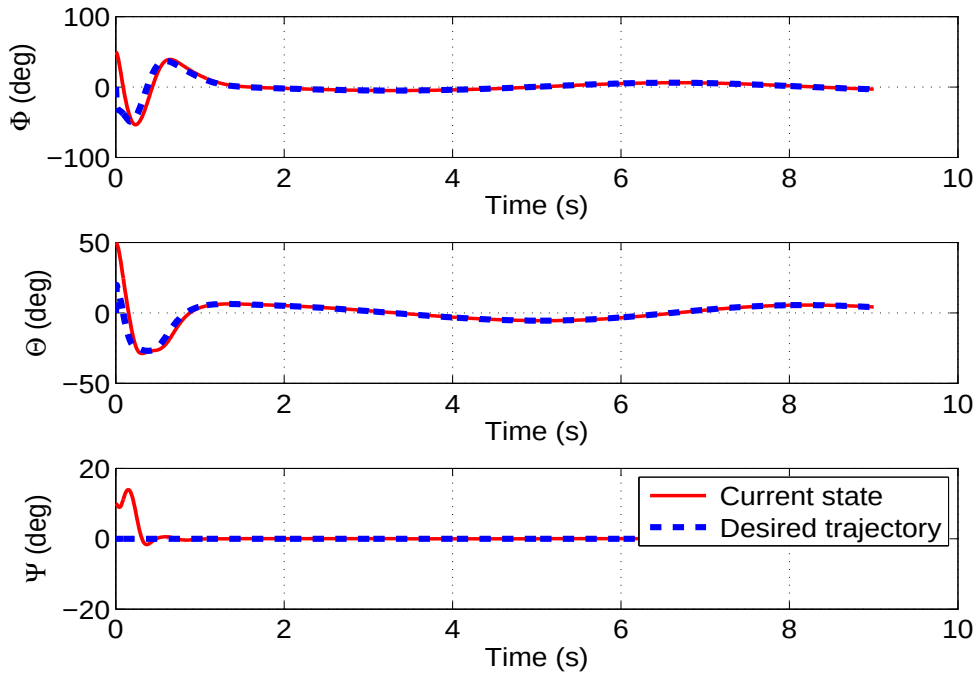


Figure 3.13: Attitude - trajectory tracking for control design based body angular rates

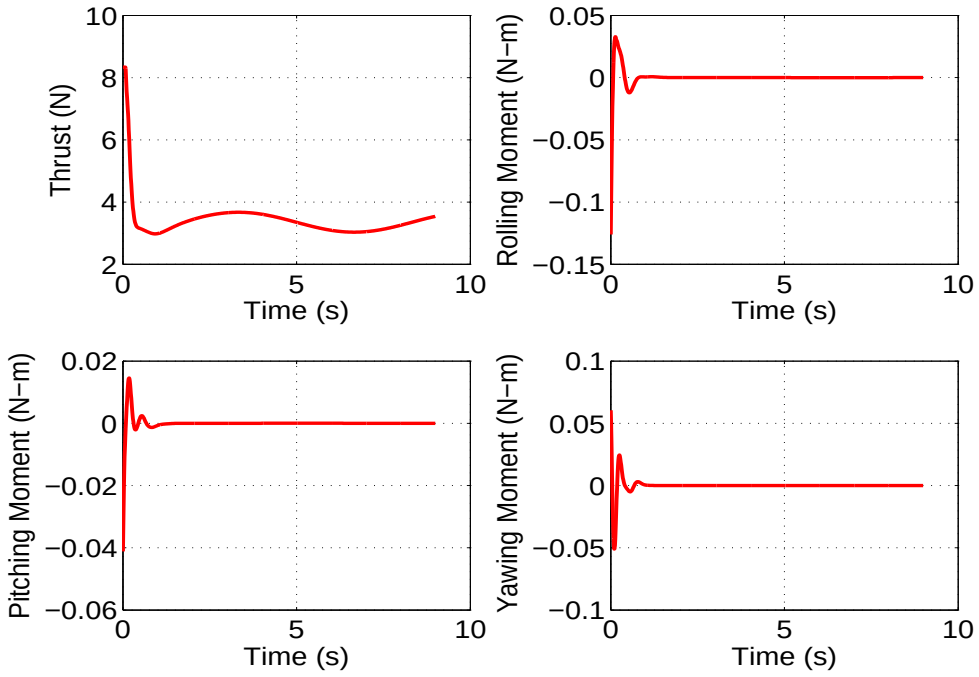


Figure 3.14: Commanded control inputs - trajectory tracking for control design based body angular rates

It is observed that unlike the the previous design (Fig. 3.8- 3.10), the tracking performance of this alternative design is good at high frequency trajectories (Fig. 3.15-3.17). The simulation results for high frequency trajectory ($\omega = 9.85\ (rad/s)$) is presented as follows.

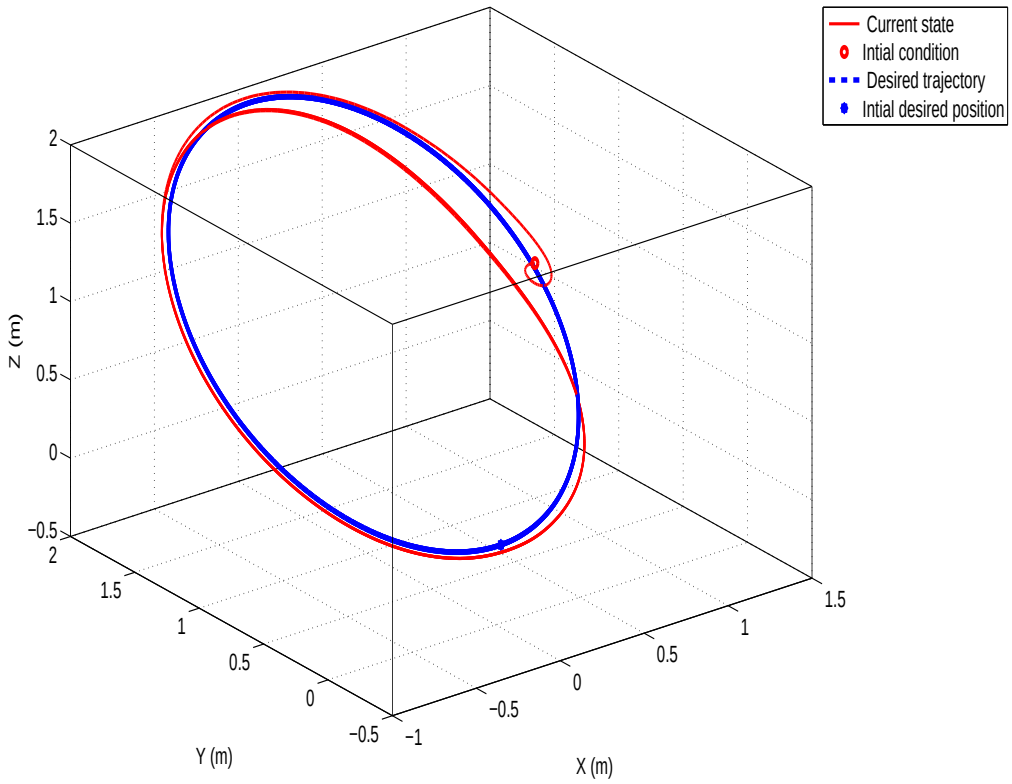


Figure 3.15: Position - high frequency trajectory tracking for control design based on angular rate tracking

From Fig. 3.15 , it can be clearly seen that the new controller is able to track the commanded trajectory far better.

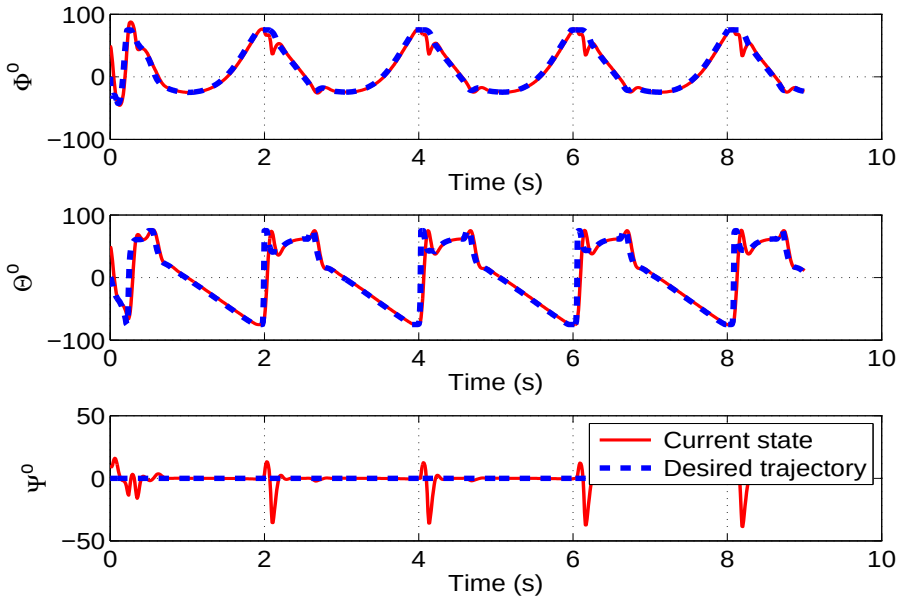


Figure 3.16: Attitude - high frequency trajectory tracking for control design based on angular rate tracking

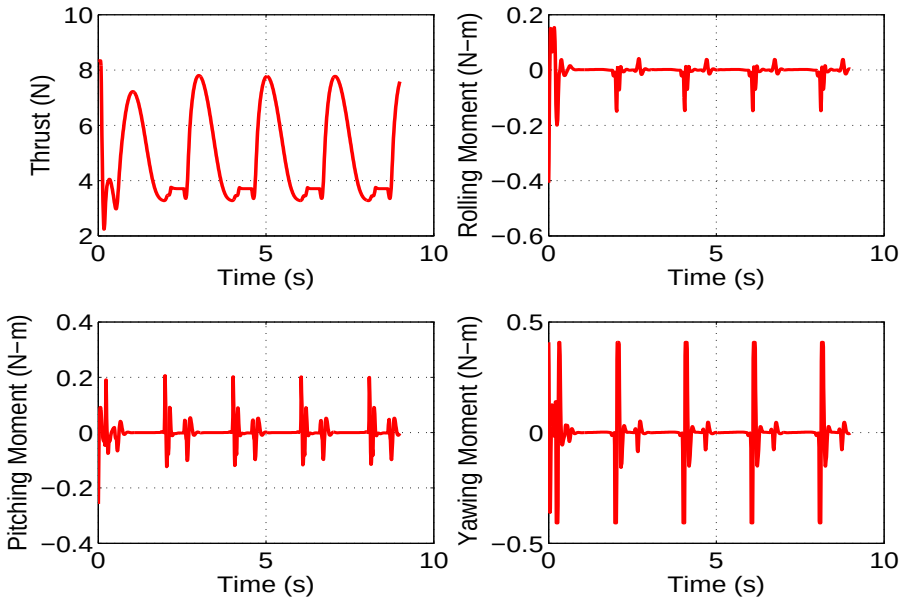


Figure 3.17: Commanded control inputs - high frequency trajectory tracking for control design based on angular rate tracking

From Fig. 3.17, it can be seen that the commanded control is smooth and has not saturated unlike in the previous design.

Comment: Moreover, it can be seen from the Eq.(3.18) that there is no transformation matrix in the inner loop. Thus, the new design has no trigonometric terms in the inner loop. This will lead to better performance of the controller.

## 3.5 Comparison with PID Controller

In this section, the performance of our design based on position and body rate is compared with PID design. For the performance comparative study, we consider 'to and fro' motion of a quadrotor in three dimensional plane. This maneuver is challenging as it involves a rapid acceleration and deceleration motion of the quadrotor. We consider five cases in which the frequency of the demanded trajectory is varied as provided in Table. 3.3. The parameters of trajectories used in comparison are given in Table. 3.2.

| Name | Value | Units |
|---|---|---|
| Acceleration in x ($a_x$) | -$\omega V \sin(\omega t)$ | $m/s^2$ |
| Acceleration in y ($a_y$) | -$\omega V \sin(\omega t)$ | $m/s^2$ |
| Acceleration in z ($a_z$) | -$\omega V \sin(\omega t)$ | $m/s^2$ |
| Velocity in x ($v_x$) | $V \cos(\omega t)$ | $m/s$ |
| Velocity in y ($v_y$) | $V \cos(\omega t)$ | $m/s$ |
| Velocity in z ($v_z$) | $V \cos(\omega t)$ | $m/s$ |
| Position (x) | $x_i + v_x dt$ | $m$ |
| Position (y) | $y_i + v_y dt$ | $m$ |
| Position (z) | $z_i + v_z dt$ | $m$ |

Table 3.2: Instantaneous desired trajectory information

Here, V = 3 m/s is the amplitude of the velocity command and $\omega$ is the angular frequency and suffix i denotes the data at the $i^{th}$ time instant.

| Cases | Trajectory frequency ($\omega$)(rad/s) |
|---|---|
| Case 1 | 4.5 (rad/s) |
| Case 2 | 6.5 (rad/s) |
| Case 3 | 7.325 (rad/s) |
| Case 4 | 9 (rad/s) |
| Case 5 | 11.5 (rad/s) |

Table 3.3: Reference trajectory frequencies for different cases

### 3.5.1 Case 1

The tracking performance of the PID controller design starts to deteriorate as the frequency of the trajectory is increased. It is clear from Fig. 3.18 that the PID controller starts fluctuation in position tracking while our proposed design tracks smoothly.

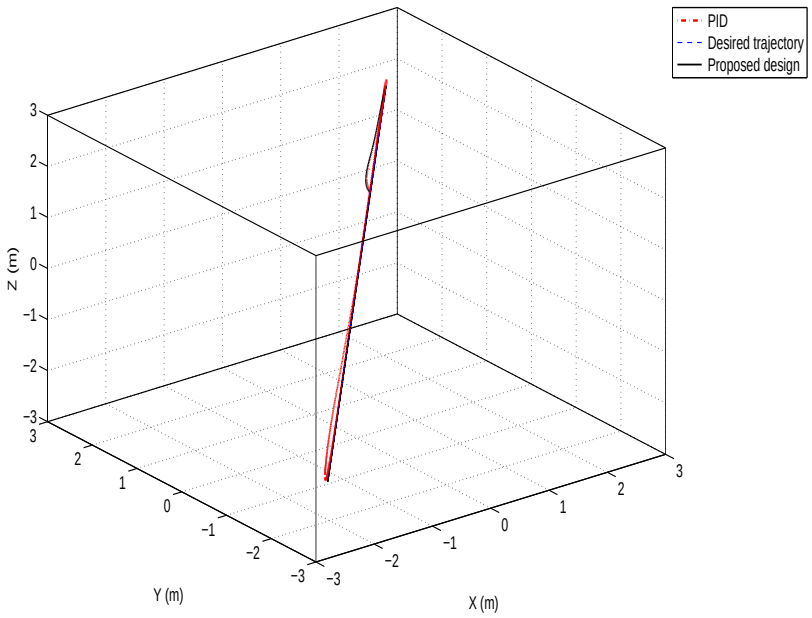


Figure 3.18: Case 1 - Performance comparison of tracking position

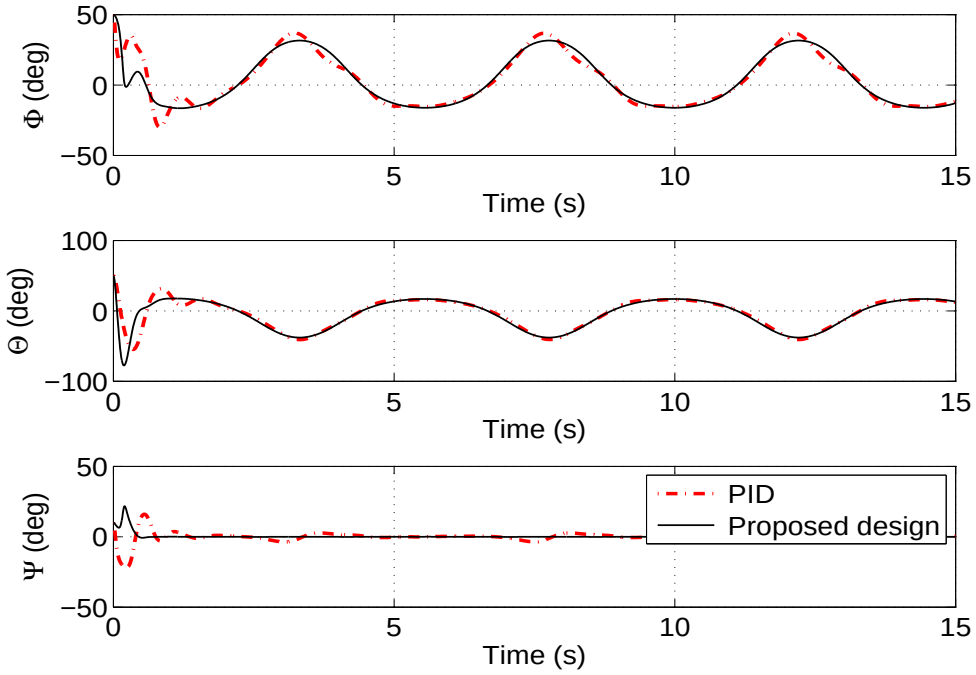


Figure 3.19: Case 1 - Performance comparison of tracking attitude

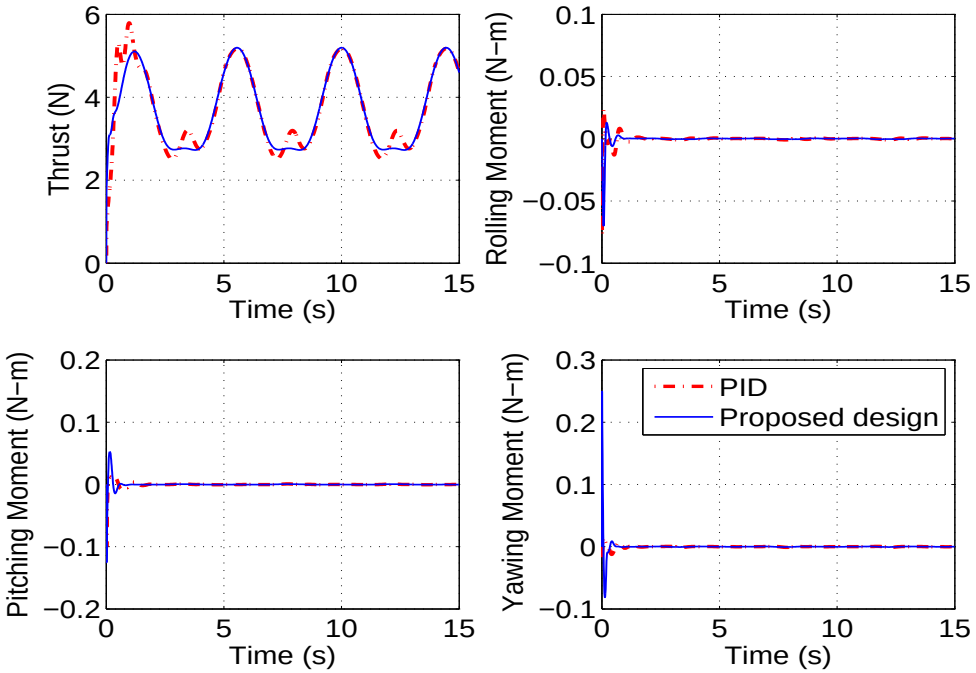


Figure 3.20: Case 1 - Commanded control inputs generated by PID and proposed design

From Fig. 3.19, it can be seen that the attitude tracking by PID controller starts loosing smoothness. Also, it can be observed from Fig. 3.20 that the commanded control inputs generated by the PID controller starts to have control jerks.

### 3.5.2 Case 2

As the frequency of the desired trajectory is further increased, it is observed that the PID controller generates a strong control jerks which cannot be fed into the real quadrotor. However, the proposed design continues to perform well by generating a smooth control inputs.

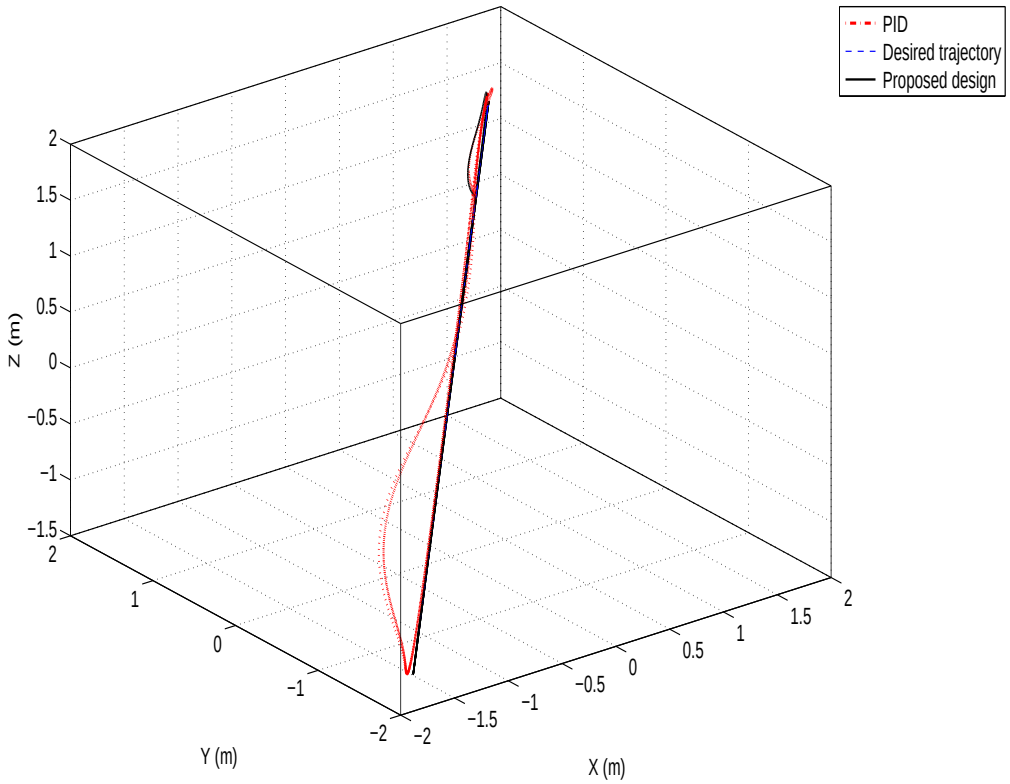


Figure 3.21: Case 2 - Position tracking performance of PID and proposed design

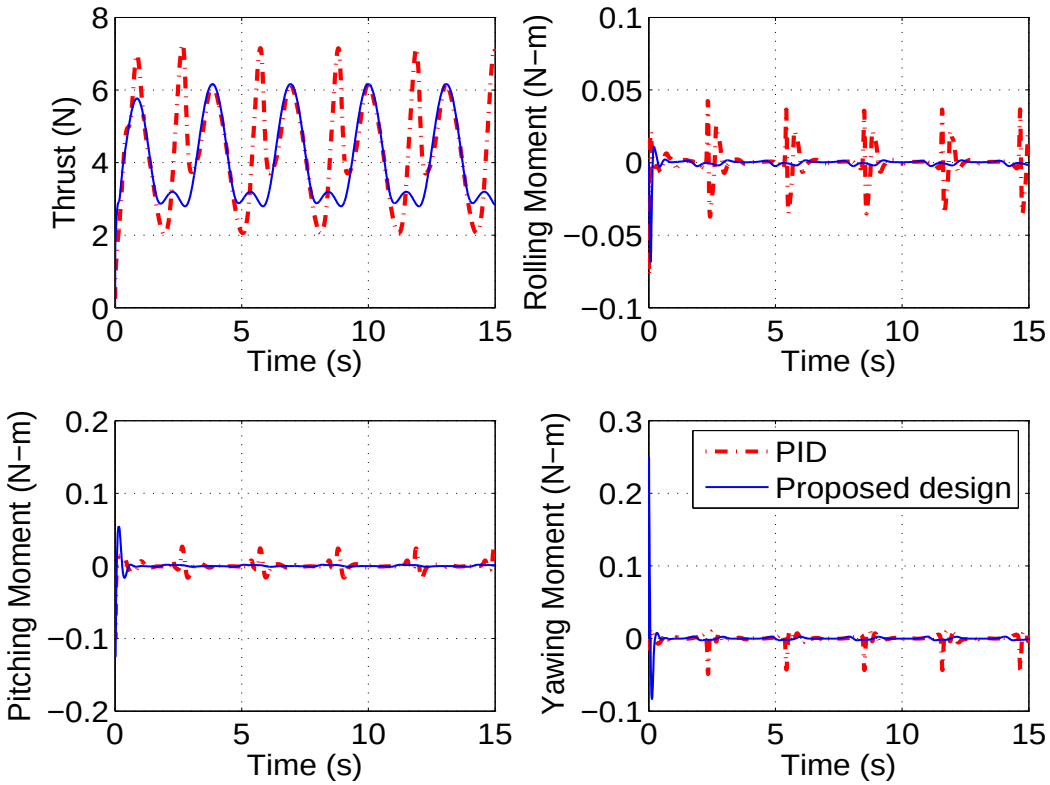


Figure 3.22: Case 2 - Commanded control inputs by PID and proposed design

The PID controller is no longer capable of tracking the desired trajectory close enough. This is evident from Fig. 3.21.The attitude tracking also has jerks and the plot is neglected here for brevity. A strong control jerks can be observed in Fig. 3.22, at the same time there is no control jerk found in the proposed design.

### 3.5.3 Case 3

As the frequency of demanded trajectory is further increased, the PID controller is no longer capable of tracking the desired trajectory and fails completely.

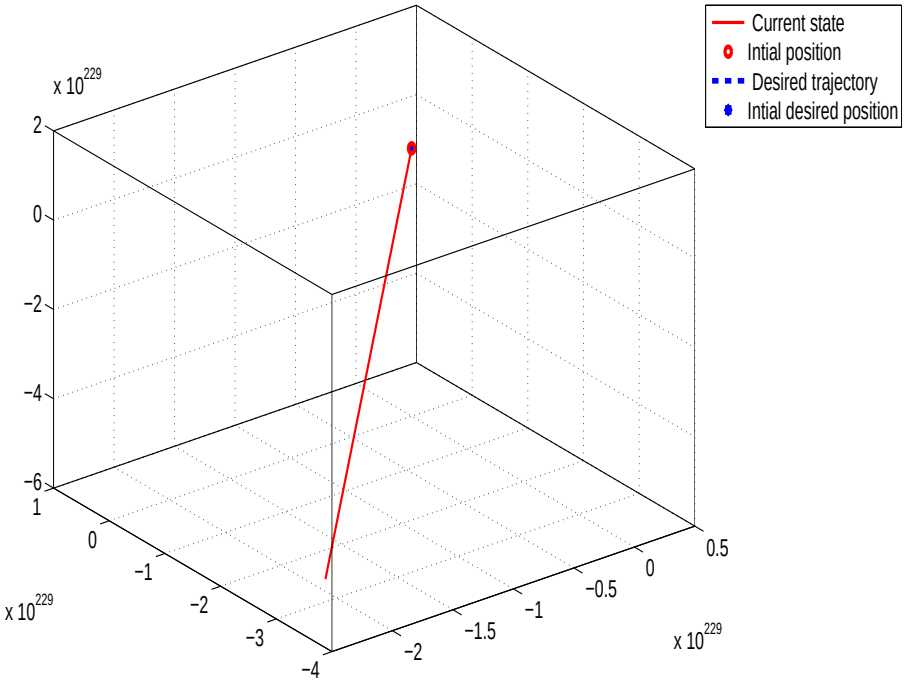


Figure 3.23: Case 3 - Position tracking performance of PID

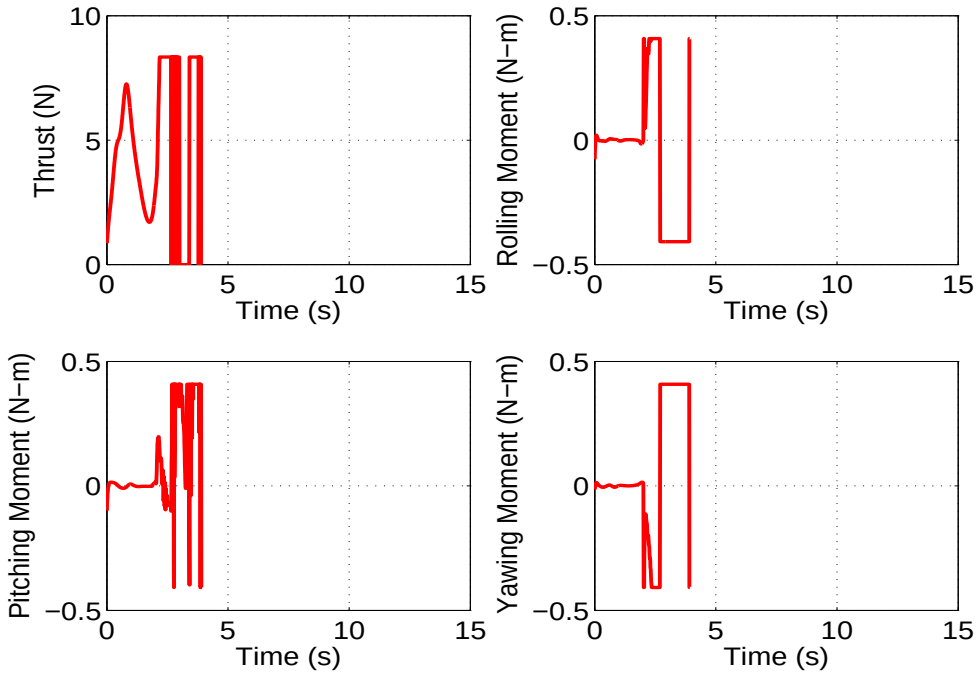


Figure 3.24: Case 3 - Control inputs generated by PID

This simulation result shown in Fig. 3.23 and Fig. 3.24 is very important. The PID design fails to track stringent trajectory that is obvious but the interesting point is the nature of the failure of control strategy. The performance is so bad that if this trajectory is demanded to a quadrotor in the experimental cases then there will be a complete destruction of the vehicle. This is not the case in our proposed design, that can be observed in Fig. 3.28.

### 3.5.4 Case 4

The performance fluctuation is observed in the proposed controller design at high frequency. The associated simulation results are presented as follows.

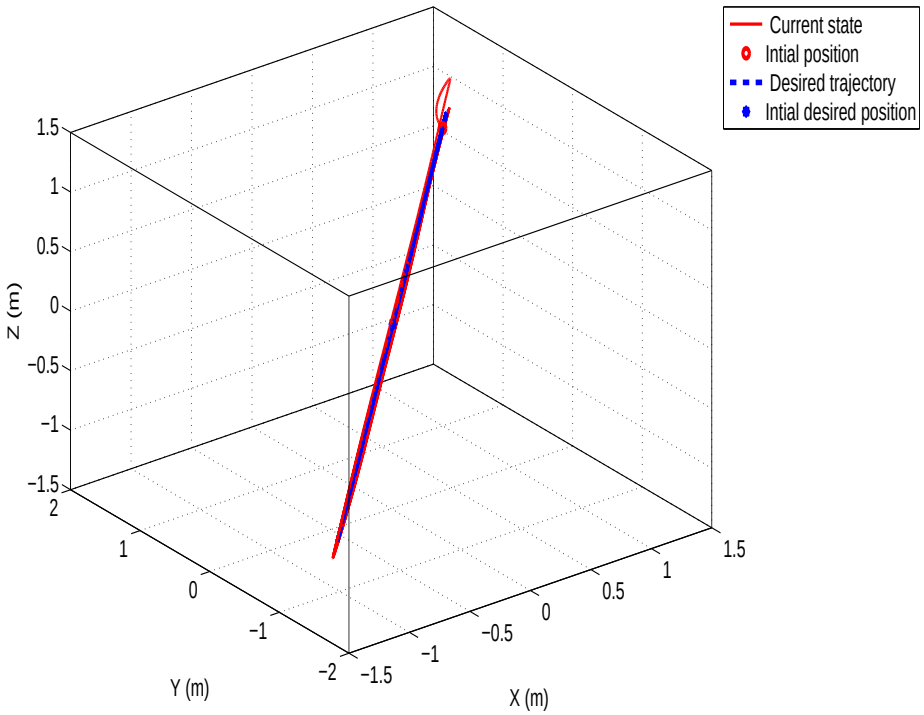


Figure 3.25: Case 4 - Position tracking performance of proposed design

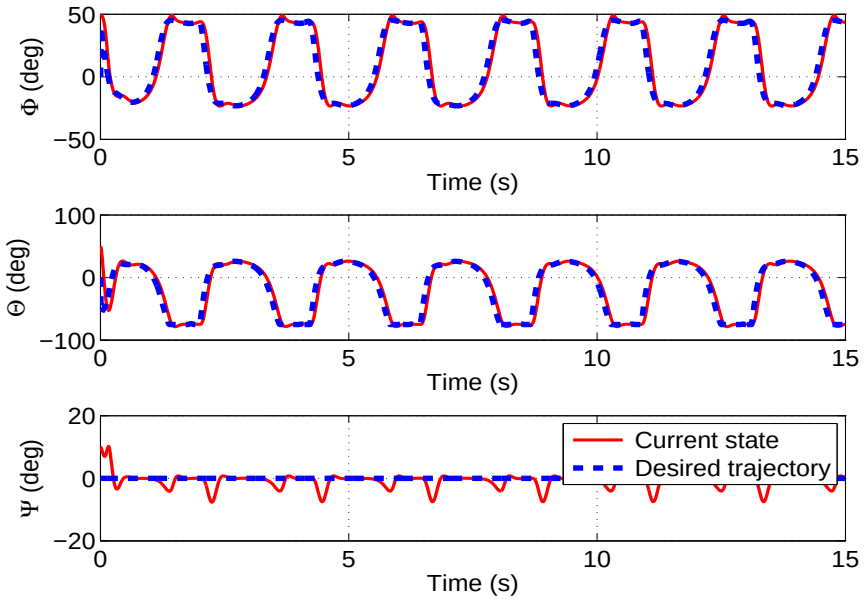


Figure 3.26: Case 4 - Attitude tracking performance of proposed design

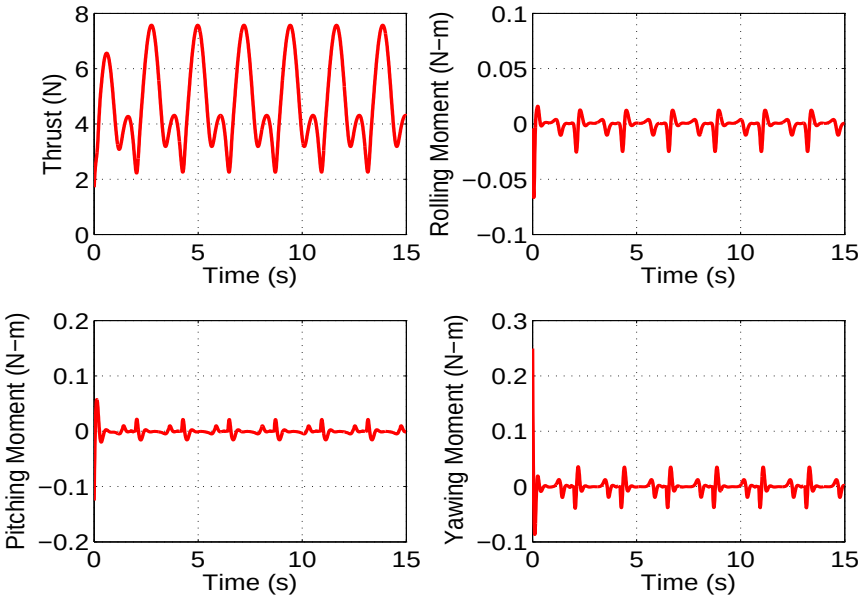


Figure 3.27: Case 4 - Commanded control inputs of proposed design

From Fig. 3.25, it can be observed that the tracking performance of the proposed design starts to deteriorate. It can still be seen from Fig. 3.26 and Fig. 3.27 that the attitude tracking and the commanded control inputs remains smooth.

### 3.5.5 Case 5

This is an important case in the comparison of the proposed design and PID controller. As the frequency of the desired trajectory is increased, we can observe that the proposed design starts generating control jerks. But an important point is that, the controller has started producing jerks only at an extreme frequency.

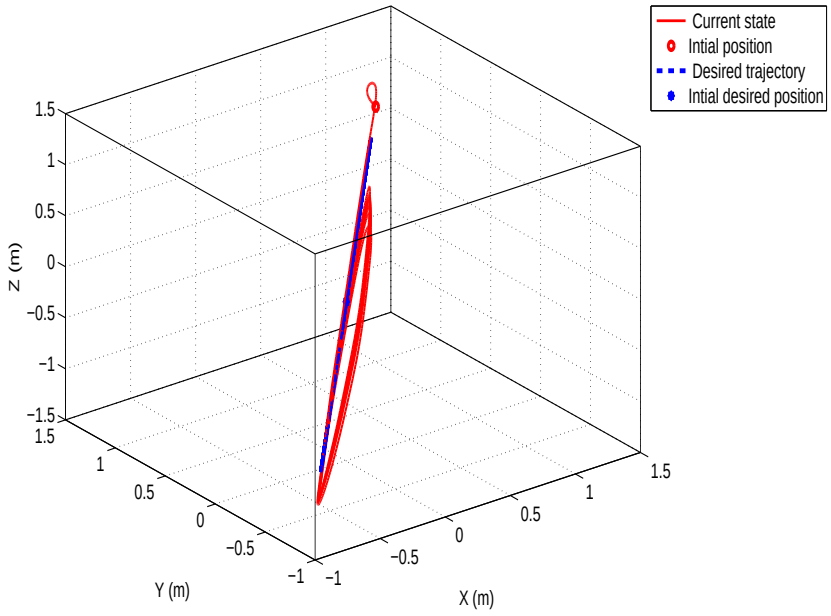


Figure 3.28: Case 5 - Position tracking performance of proposed design

The objective of this Case 5 is to give emphasis to the nature of failure of the controller. An extensive simulation was done for both the PID and proposed designs. It is observed that the PID controller fails more like a brittle material (i.e) the controller fails abruptly and completely which in real world will lead to the severe destruction scenario. But this is not the case in proposed design. The failure is more like a ductile in nature. The performance of the controller deteriorates slowly, this guarantees that

if proper measure is taken by a human pilot, then there is a possibility of recovery.

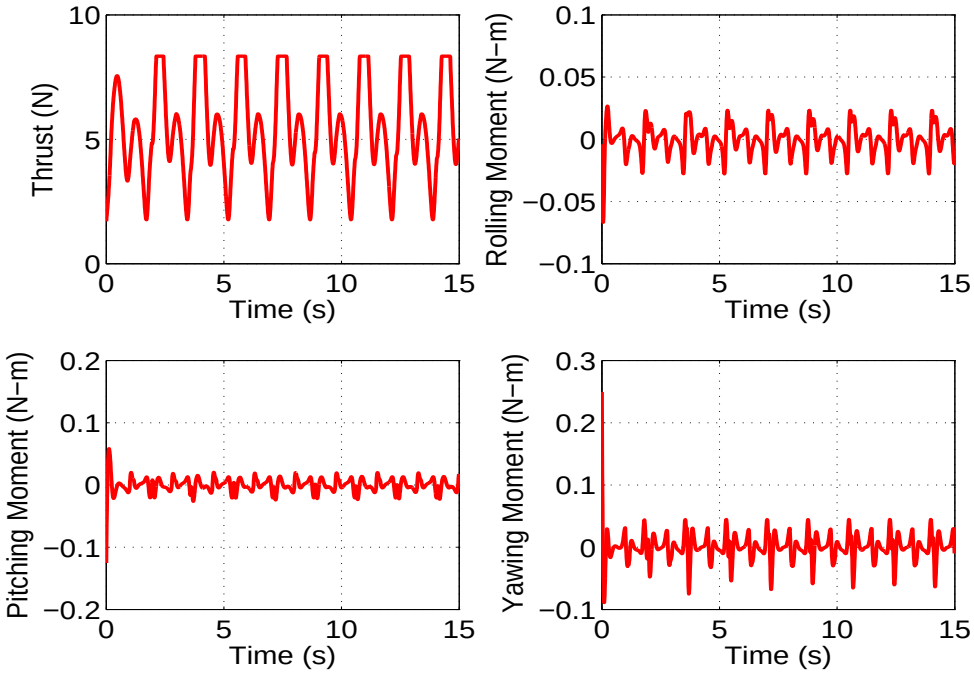


Figure 3.29: Case 5 - Commanded control inputs of proposed design

Thus, it can be clearly seen that PID controller needs tedious gain scheduling process if it has to be used for high performance maneuver. At the same time, nonlinear control design is capable of achieving a good performance throughout the flight regime without the need for gain scheduling. Also the proposed design tracking performance degrades gently. Thus the proposed design could be operated in real world system without the fear of break down.