University of Alberta, Department of Electrical and Computer Engineering

Project Report

ECE 664 Course Project

IOSFBL and Back-stepping control design for pose control of a Quadrotor UAV and implementation on PX4 autopilot firmware

Muhammad Awais Rafique
Amir Moeini
Linjian Xiang

January 26, 2018

Abstract

This document describes the application of IOSFBL for pose control of a Quadrotor UAV using dynamic extension and outlines the process of implementation on a PX4 autopilot firmware. First the dynamics of Quad are presented and IOSFBL methodology is applied which fails to design IOSFBL control for given Quadrotor dynamics. The system model is then dynamically extended by introducing the derivative of thrust input to obtain a new system model which result in a successful attempt of control design using IOSFBL. The controller is simulated in Matlab Simulink for a set point and 8-shaped trajectory.

The second achievement is the implementation of IOSFBL controller in PX4 autopilot firmware which can be used on Pixhawk hardware for flying the original Quad. The process involves learning different aspects to platform programming and advanced use of C/C++ language. The implementation of the IOSFBL on PX4 is tested using hand held tests and the results are compared with the built-in PID inner-outer loop approach on PX4. The results show better performance and quick reaction to target pose.

Furthermore, a back-stepping based controller design approach for quadcopter is analyzed and verified using simulation and a comparison is made with the PID control based on an approximate model.

Contents

1 Input-Output State feedback Linearization

1.1 Quadcopter Dynamics

The dynamics of a Quad rotor UAV are given by

$$\dot{p}^n = v^n \tag{1.1a}$$
$$m\dot{v}^n = R_b^n F^b + mg^n \tag{1.1b}$$
$$\dot{R}_b^n = \mathrm{sk}(\omega^b) R_b^n \tag{1.1c}$$
$$J\dot{w}^b = -\omega^b \times J\omega^b + \tau^b \tag{1.1d}$$

where $p^n \in R^3$ is the position of the vehicle in Navigation frame N, $v^n \in R^3$ is the linear velocity of the vehicle in N, $\omega^b \in R^3$ is the angular velocity of the vehicle in Body frame B, $m \in R$ is the mass of the vehicle, $g^n \in R^3$ is the acceleration due to gravity, $J \in R^{3\times 3}$ is the inertia matrix of the vehicle. The external force and torque acting on the vehicle are denoted $F^b \in R^3$, $\tau^b \in R^3$, respectively, and the function sk converts a vector to a skew symmetric matrix such that

$$\mathrm{sk}(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

In the discussion that follows the variables are considered to be in their corresponding frames as defined above, however, superscripts are not explicitly shown to make the equations look simple. Using Euler angles for rotational dynamics, the above equations can further be written in the following form

$$\dot{x} = f(x) + \sum_{i=1}^{4} g_i(x) u_i = f(x) + G(x)u$$
$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

where

$$x = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ v_1 \\ v_2 \\ v_3 \\ \phi \\ \theta \\ \psi \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \quad f(x) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \\ 0 \\ mg \\ \omega_1 + \omega_2 sin\phi tan\theta + \omega_3 cos\phi tan\theta \\ \omega_2 \cos\phi - \omega_3 \sin\phi \\ \omega_2 \frac{sin\phi}{\cos\theta} + \omega_3 \frac{cos\phi}{\cos\theta} \\ \left(\frac{J_2}{J_1} - \frac{J_3}{J_1}\right) \omega_2\omega_3 \\ \left(\frac{J_3}{J_2} - \frac{J_1}{J_2}\right) \omega_3\omega_1 \\ \left(\frac{J_1}{J_3} - \frac{J_2}{J_3}\right) \omega_1\omega_2 \end{bmatrix},$$

$$g_1(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{-\cos\psi \sin\theta \cos\phi - \sin\psi \sin\phi}{m} \\ \frac{-\cos\phi \sin\theta \sin\psi + \cos\psi \sin\phi}{m} \\ -\frac{\cos\theta \cos\phi}{m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad g_2(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \end{bmatrix}, \quad g_3(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_2} \\ 0 \end{bmatrix} \quad g_4(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_3} \end{bmatrix},$$

and

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T_M \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

whereas T_M is the upward thrust, τ_1, τ_2 and τ_3 are body frame torques around x, y and z axes respectively.

1.2 Linearizability check

Given system is a 12th order system and we are interested in the pose of the system as output which constitutes on quad position and yaw angel. The output vector is given as

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \psi \end{bmatrix}$$

Considering these outputs Quadcopter is a Multi-Input Multi-Output System (MIMO) with four inputs and four outputs. In Input-Output State Feedback Linearizion (IOSFBL) for a single output case, we take the derivatives of the output till the input shows up and if the input has a non-zero coefficient at least locally in the neighborhood of origin, the number of derivative is called relative degree and it is said to be well defined and the system is Input-Output state feedback linearizable. However, in case of multi-input multi-output case, relative degree is a set of numbers, accounting for number of derivatives corresponding to each output when the input shows up and the relative degree is said to be well-defined if a decoupling matrix which forms the matrix coefficient of input vector is non-singular.
In case of Quadcopter, taking derivatives of the individual outputs, the inputs show up after second derivative of each of the output, so the relative degree candidate for given system is the set $\{2, 2, 2, 2\}$ which results in following decoupling matrix

$$
\begin{aligned}
D(x) &= \begin{bmatrix} L_{g_1}L_f h_1 & L_{g_2}L_f h_1 & L_{g_3}L_f h_1 & L_{g_4}L_f h_1 \\ L_{g_1}L_f h_2 & L_{g_2}L_f h_2 & L_{g_3}L_f h_2 & L_{g_4}L_f h_2 \\ L_{g_1}L_f h_3 & L_{g_2}L_f h_3 & L_{g_3}L_f h_3 & L_{g_4}L_f h_3 \\ L_{g_1}L_f h_4 & L_{g_2}L_f h_4 & L_{g_3}L_f h_4 & L_{g_4}L_f h_4 \end{bmatrix} \\
&= \begin{bmatrix} \frac{-\cos\psi\sin\theta\cos\phi-\sin\psi\sin\phi}{m} & 0 & 0 & 0 \\ \frac{-\cos\phi\sin\theta\sin\psi+\cos\psi\sin\phi}{m} & 0 & 0 & 0 \\ -\frac{\cos\theta\cos\phi}{m} & 0 & 0 & 0 \\ 0 & 0 & \frac{\sin\phi}{J_2\cos\theta} & \frac{\cos\phi}{J_3\cos\theta} \end{bmatrix}
\end{aligned}
$$

Clearly the decoupling matrix, consisting of a column of zeros is singular and not invertible in any neighborhood of the origin resulting in an undefined relative degree. So the given system with these dynamics is not Input-Output State Feedback Linearizable (IOSFBL), however, dynamic extension makes the system IOSFBL, which is further discussed in the following section.

1.3 Dynamic Extension and Controller Design

If the quadcopter dynamics are extended by augmenting system states by introduction thrust and its first derivative as states and defining second derivative of thrust as input, making a 14^{th} order system which is IOSFBL. The quadcopter model with dynamic extension is given by:

$$x = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ v_1 \\ v_2 \\ v_3 \\ \phi \\ \theta \\ \psi \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ T_1 \\ T_2 \end{bmatrix}, \quad f(x) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \frac{-\cos\psi\sin\theta\cos\phi - \sin\psi\sin\phi}{m} T_1 \\ \frac{-\cos\phi\sin\theta\sin\psi + \cos\psi\sin\phi}{m} T_1 \\ \frac{mg - \cos\theta\cos\phi}{m} T_1 \\ \omega_1 + \omega_2 sin\phi tan\theta + \omega_3 cos\phi tan\theta \\ \omega_2\cos\phi - \omega_3\sin\phi \\ \omega_2 \frac{sin\phi}{\cos\theta} + \omega_3 \frac{cos\phi}{\cos\theta} \\ \left(\frac{J_2}{J_1} - \frac{J_3}{J_1}\right)\omega_2\omega_3 \\ \left(\frac{J_3}{J_2} - \frac{J_1}{J_2}\right)\omega_3\omega_1 \\ \left(\frac{J_1}{J_3} - \frac{J_2}{J_3}\right)\omega_1\omega_2 \\ T_2 \\ 0 \end{bmatrix},$$

$$g_1(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad g_2(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad g_3(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_2} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad g_4(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_3} \\ 0 \\ 0 \end{bmatrix},$$

where

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \ddot{T}_M \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix},$$

$$T_1 = T_M, \quad T_2 = \dot{T}_M, \quad T_M(t) = \int_0^t \int_0^{t_2} u_1(t_1) dt_1 dt_2.$$

Taking the derivatives of the outputs until the input shows up, we have relative degree for the dynamically extended system as $\rho = \{4, 4, 4, 2\}$ which is well defined since the coupling matrix

$$D(x)=\begin{bmatrix} L_{g_1}L_f^3h_1 & L_{g_2}L_f^3h_1 & L_{g_3}L_f^3h_1 & L_{g_4}L_f^3h_1 \\ L_{g_1}L_f^3h_2 & L_{g_2}L_f^3h_2 & L_{g_3}L_f^3h_2 & L_{g_4}L_f^3h_2 \\ L_{g_1}L_f^3h_3 & L_{g_2}L_f^3h_3 & L_{g_3}L_f^3h_3 & L_{g_4}L_f^3h_3 \\ L_{g_1}L_fh_4 & L_{g_2}L_fh_4 & L_{g_3}L_fh_4 & L_{g_4}L_fh_4 \end{bmatrix}$$

$$Dx := \begin{bmatrix} \frac{T1\,(\cos(\psi)\sin(\theta)\sin(\phi)-\sin(\psi)\cos(\phi))}{J1\,m} & -\frac{T1\cos(\psi)\cos(\theta)}{J2\,m} & 0 & -\frac{\cos(\psi)\sin(\theta)\cos(\phi)+\sin(\psi)\sin(\phi)}{m} \\ \frac{T1\,(\sin(\phi)\sin(\theta)\sin(\psi)+\cos(\phi)\cos(\psi))}{J1\,m} & -\frac{T1\sin(\psi)\cos(\theta)}{J2\,m} & 0 & \frac{-\cos(\phi)\sin(\theta)\sin(\psi)+\sin(\phi)\cos(\psi)}{m} \\ \frac{\cos(\theta)\sin(\phi)\,T1}{J1\,m} & \frac{T1\sin(\theta)}{J2\,m} & 0 & -\frac{\cos(\theta)\cos(\phi)}{m} \\ 0 & \frac{\sin(\phi)}{J2\cos(\theta)} & \frac{\cos(\phi)}{J3\cos(\theta)} & 0 \end{bmatrix}$$

is non-singular and invertible in neighborhood of the origin provided that $T_1 \neq 0$. Where the local region is marked by roll and pitch angles $abs(\theta, \phi) < pi/2$. So the relative degree is well defined and the derivatives of outputs are give as

$$\begin{bmatrix} {p_1}^{(4)} \\ {p_2}^{(4)} \\ {p_3}^{(4)} \\ \psi^{(2)} \end{bmatrix} = \alpha(x) + D(x)u = \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix}$$

whereas $\alpha(x)$ is given by

```
R14 > alpha:=Matrix(4, 1, {(1, 1) = Lf4h1,
                           (2, 1) = Lf4h2,
                           (3, 1) = Lf4h3,
                           (4, 1) = Lf2h4});
```

$$\begin{aligned}\alpha := \Bigg[\Bigg[& \frac{1}{m\,J1\,J2}\big(J1\,J2\,\omega 1^2\,T1\cos(\psi)\sin(\theta)\cos(\phi)+2\,J1\,J2\,\omega 1\,T2\cos(\psi)\sin(\theta)\sin(\phi)+\cos(\psi)\,J1\,J2\cos(\phi)\,\omega 2^2\,T1\sin(\theta) \\ & +J1\,J2\cos(\phi)\,\omega 3\,T1\,\omega 2\sin(\psi)+\omega 3\,\omega 2\,T1\,J2^2\cos(\psi)\sin(\theta)\sin(\phi)+\omega 3\,\omega 2\,T1\,J2\,J3\sin(\psi)\cos(\phi)-\cos(\psi)\cos(\theta)\,J1\,J2\,\omega 3\,T1\,\omega 1 \\ & +J1\,J2\sin(\phi)\,\omega 2^2\,T1\sin(\psi)-\omega 3\,\omega 2\,T1\,J2^2\sin(\psi)\cos(\phi)-\omega 3\,\omega 1\,T1\cos(\psi)\cos(\theta)\,J1\,J3+J1\,J2\,\omega 1^2\,T1\sin(\psi)\sin(\phi) \\ & -2\,J1\,J2\,\omega 1\,T2\sin(\psi)\cos(\phi)+\omega 3\,\omega 1\,T1\cos(\psi)\cos(\theta)\,J1^2-2\,T2\cos(\theta)\,J1\,J2\,\omega 2\cos(\psi)-\cos(\psi)\,J1\,J2\sin(\phi)\,\omega 3\,T1\,\omega 2\sin(\theta) \\ & -\omega 3\,\omega 2\,T1\,J2\,J3\cos(\psi)\sin(\theta)\sin(\phi)\big)\Bigg], \\ \Bigg[& \frac{1}{m\,J1\,J2}\big(-\sin(\psi)\cos(\theta)\,J1\,J2\,\omega 3\,T1\,\omega 1-J1\,J2\cos(\phi)\,\omega 3\,T1\,\omega 2\cos(\psi)+\omega 3\,\omega 2\,T1\,J2^2\sin(\phi)\sin(\theta)\sin(\psi) \\ & -\omega 3\,\omega 2\,T1\,J2\,J3\cos(\phi)\cos(\psi)+J1\,J2\,\omega 1^2\,T1\cos(\phi)\sin(\theta)\sin(\psi)+2\,J1\,J2\,\omega 1\,T2\sin(\phi)\sin(\theta)\sin(\psi)+\sin(\psi)\,J1\,J2\cos(\phi)\,\omega 2^2\,T1\sin(\theta) \\ & -J1\,J2\,\omega 1^2\,T1\sin(\phi)\cos(\psi)+2\,J1\,J2\,\omega 1\,T2\cos(\phi)\cos(\psi)-J1\,J2\sin(\phi)\,\omega 2^2\,T1\cos(\psi)+\omega 3\,\omega 2\,T1\,J2^2\cos(\phi)\cos(\psi) \\ & -\omega 3\,\omega 1\,T1\sin(\psi)\cos(\theta)\,J1\,J3+\omega 3\,\omega 1\,T1\sin(\psi)\cos(\theta)\,J1^2-2\,T2\cos(\theta)\,J1\,J2\,\omega 2\sin(\psi)-\sin(\psi)\,J1\,J2\sin(\phi)\,\omega 3\,T1\,\omega 2\sin(\theta) \\ & -\omega 3\,\omega 2\,T1\,J2\,J3\sin(\phi)\sin(\theta)\sin(\psi)\big)\Bigg], \\ \Bigg[& -\frac{1}{J1\,m\,J2}\big(-J1\,J2\,\omega 1^2\cos(\theta)\,T1\cos(\phi)-2\,J1\,J2\,\omega 1\cos(\theta)\,T2\sin(\phi)-2\,T2\,J1\,J2\,\omega 2\sin(\theta)-J1\,J2\cos(\phi)\,\omega 2^2\,T1\cos(\theta) \\ & -J1\,J2\sin(\theta)\,\omega 3\,T1\,\omega 1+J1\,J2\sin(\phi)\,\omega 3\,T1\,\omega 2\cos(\theta)-\omega 3\,\omega 2\cos(\theta)\sin(\phi)\,T1\,J2^2+\omega 3\,\omega 2\cos(\theta)\sin(\phi)\,T1\,J2\,J3+\omega 3\,\omega 1\,T1\sin(\theta)\,J1^2 \\ & -\omega 3\,\omega 1\,T1\sin(\theta)\,J1\,J3\big)\Bigg], \\ \Bigg[& \frac{(\omega 1\cos(\theta)+\sin(\phi)\sin(\theta)\,\omega 2+\cos(\phi)\sin(\theta)\,\omega 3)\,(\cos(\phi)\,\omega 2-\sin(\phi)\,\omega 3)}{\cos(\theta)^2}+\frac{(\cos(\phi)\,\omega 2-\sin(\phi)\,\omega 3)\,(\omega 2\sin(\phi)+\omega 3\cos(\phi))\sin(\theta)}{\cos(\theta)^2} \\ & -\frac{\omega 3\,\omega 1\,(J1-J3)\sin(\phi)}{J2\cos(\theta)}+\frac{\omega 2\,\omega 1\,(J1-J2)\cos(\phi)}{J3\cos(\theta)}\Bigg]\Bigg]\end{aligned}$$

Now u can be designed in terms of μ so that there is a linear relationship between outputs and μ as follows

$$u = [D(x)]^{-1}(\mu - \alpha(x))$$

After hand simplification, we get the following equations for the IOSFBL control law

$$u := \begin{bmatrix} (J1 - J2 + J3)\,\omega 2\,\omega 3 + \dfrac{J1\,(-2\,\omega 1\,T2 + m\,(\sin(\phi)\,(\mu 1 \sin(\theta)\cos(\psi) + \mu 2 \sin(\theta)\sin(\psi) + \mu 3 \cos(\theta)) - \cos(\phi)\,(\mu 1 \sin(\psi) - \mu 2 \cos(\psi))))}{T1} \\[2ex] (J1 - J2 - J3)\,\omega 1\,\omega 3 + \dfrac{J2\,(-2\,T2\,\omega 2 + m\,(\mu 3 \sin(\theta) - \cos(\theta)\,(\mu 1 \cos(\psi) + \mu 2 \sin(\psi))))}{T1} \\[2ex] \begin{aligned}(-J1 + J2 - J3)\,\omega 1\,\omega 2 + J3 \Bigg(\tan(\phi)\left(\dfrac{2\,T2\,\omega 2 + m\,(\cos(\theta)\,(\mu 1 \cos(\psi) + \mu 2 \sin(\psi)) - \mu 3 \sin(\theta))}{T1} + 2\,\omega 3\,\omega 1\right) \\ + \dfrac{\mu 4 \cos(\theta)}{\cos(\phi)} + 2\tan(\theta)\left(\sin(\phi)\,(\omega 3^2 - \omega 2^2) + \omega 2\,\omega 3\left(\dfrac{1}{\cos(\phi)} - 2\cos(\phi)\right)\right)\Bigg)\end{aligned} \\[2ex] T1\,(\omega 1^2 + \omega 2^2) + m\,(-\mu 1\,(\cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi)) + \mu 2\,(-\cos(\phi)\sin(\theta)\sin(\psi) + \sin(\phi)\cos(\psi)) - \mu 3 \cos(\theta)\cos(\phi)) \end{bmatrix}$$

Using above controller we get a linear relation between outputs and μ as follows:

$$\begin{bmatrix} {p_1}^{(4)} \\ {p_2}^{(4)} \\ {p_3}^{(4)} \\ \psi^{(2)} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix}$$

Now we design μ using linear control techniques, let us take p_1, for example, the linear controller mu_1 is given by

$${p_1}^{(4)} = \mu_1 = {p_{1d}}^{(4)} + K_4({p_1}^{(3)} - {p_{1d}}^{(3)}) + K_3({p_1}^{(2)} - {p_{1d}}^{(2)}) + K_2(\dot{p}_1 - \dot{p}_{1d}) + K_1(p_1 - p_{1d})$$

Defining error $e_1 = p_1 - p_{1d}$, the error dynamics from above equation become

$$\begin{aligned} \dot{e}_1 &= & \dot{p}_1 - \dot{p}_{1d} & &= e_2 \\ \dot{e}_2 &= & \ddot{p}_1 - \ddot{p}_{1d} & &= e_3 \\ \dot{e}_3 &= & {p_1}^{(3)} - {p_{1d}}^{(3)} & &= e_4 \\ \dot{e}_4 &= & {p_1}^{(4)} - {p_{1d}}^{(4)} & &= K_4 e_4 + K_3 e_3 + K_2 e_2 + K_1 e_1 \end{aligned}$$

We select the controller gains K_1 to K_4 such that the poles of the linear error dynamic system above are in open left half plane. Similar controller equation and error dynamics can be written for remaining two components of position. For yaw, we have the following linear control law,

$$\psi^{(2)} = \mu_4 = {\psi_d}^{(2)} + K_2(\dot{\psi} - \dot{\psi}_d) + K_1(\psi - \psi_d)$$

and the second order error dynamics for determining control gains by pole placement method are given by

$$\begin{aligned} \dot{e}_{1\psi} &= & \dot{\psi} - \dot{\psi}_d & &= e_{2\psi} \\ \dot{e}_{2\psi} &= & \psi^{(2)} - {\psi_d}^{(2)} & &= K_{2\psi} e_{2\psi} + K_{1\psi} e_{1\psi}. \end{aligned}$$

Following table shows the pole locations and resulting control gains for linear controllers. For simplicity same pole locations and control gains are used for all three components of position controllers mu_1 to mu_3 while two poles of mu_4 corresponding to yaw are designed separately. Same values are used in simulation and actual controller design.

Parameters	Values
Pole locations (Position)	[-2, -1.5, -1.6, -1.7]
Pole locations (Yaw)	[-4 -5]
$[K_1, K_2, K_3, K_4]$	[-8.16, -19.42, -17.27, -6.80]
$[K_{1\psi}, K_{2\psi}]$	[-20.0, -9.0]

1.4 IOSFBL Simulation

For the purpose of simulation, the model and controller is implemented in MATLAB Simulink and the designed controller is run to track a ∞ shaped trajectory. The model parameters of Quadcopter are given by:

Parameters	Values
m	$1.6kg$
J_1	$0.03kg.m^2$
J_2	$0.03kg.m^2$
J_3	$0.05kg.m^2$

The desired to be tracked trajectory is given by following equations.

$$p_d(t) = r_0[\sin(2\pi t/T_0),\, \sin(4\pi t/T_0)/4,\, 0]$$

$$\psi_d(t) = \text{atan}(\frac{0.5\cos(4\pi t/T_0)}{\cos(2\pi t/T_0)})$$

where $r_0 = 10m$ and $T_0 = 12s$.

The simulation results are given as trajectories platted in following figures. It shows that in simulation the actual trajectory meets the desired trajectory and keep tracking it as shown in fig. 1.1. Similarly the output trajectories and controller inputs are plotted in fig. 1.2 and fig. 1.3 respectively.

Fig. 1.4 shows the error trajectories which go to zero within a finite time. Some discontinuities in the yaw error can be observed these are due to periodic discontinuities in the desired yaw, however, it is evident of the efficiency of controller that it again starts converging to zero as soon as the discontinuity occurs.

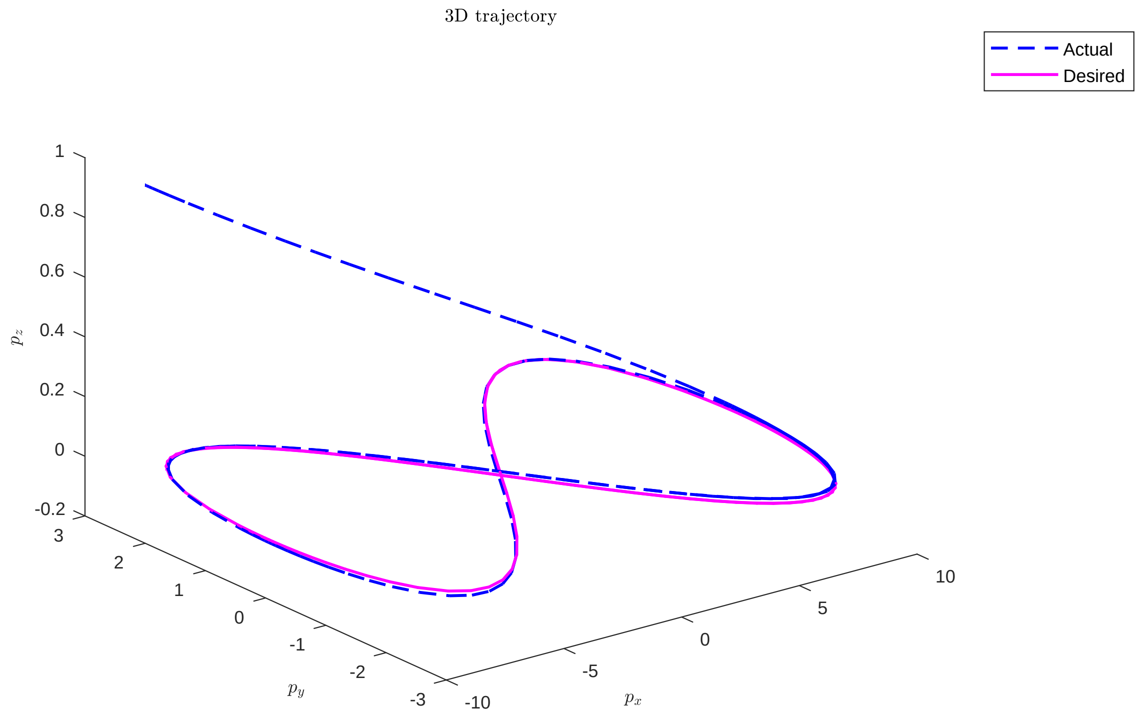

Figure 1.1: 3D position trajectory of the Quadcopter

2 IOSFBL Implementation on PX4 and Experiments

The IOSFBL approach is implemented on PX4 autopilot using two different approaches. Firstly, IOSFBL is implemented using classical approach by employing uORB function for topic subscription and publication without using the modern blocks approach. Based on the more basic library function use, this approach is more close to hardware and uses less memory and processing. This approach has provided authors with in depth understanding of what happens in the background when a topic is published or subscribed and how it can be used to control a UAV. Secondly, block based approach is used using recently the developed block libraries, using this approach a user don't have to worry about things going on on the background and a template can be edited to achieve the desired task. However, an effort has been made to understand the background processing and how the block structure works. This approach has provided authors with understanding of modern programming tools of C++ and how it can be used make complex task user friendly.

2.1 Experiments with Classical approach

After implementation on the Quadcopter, hand held tests are performed to make a comparison built-in PID control. To this point, only comparisons with the manual mode of

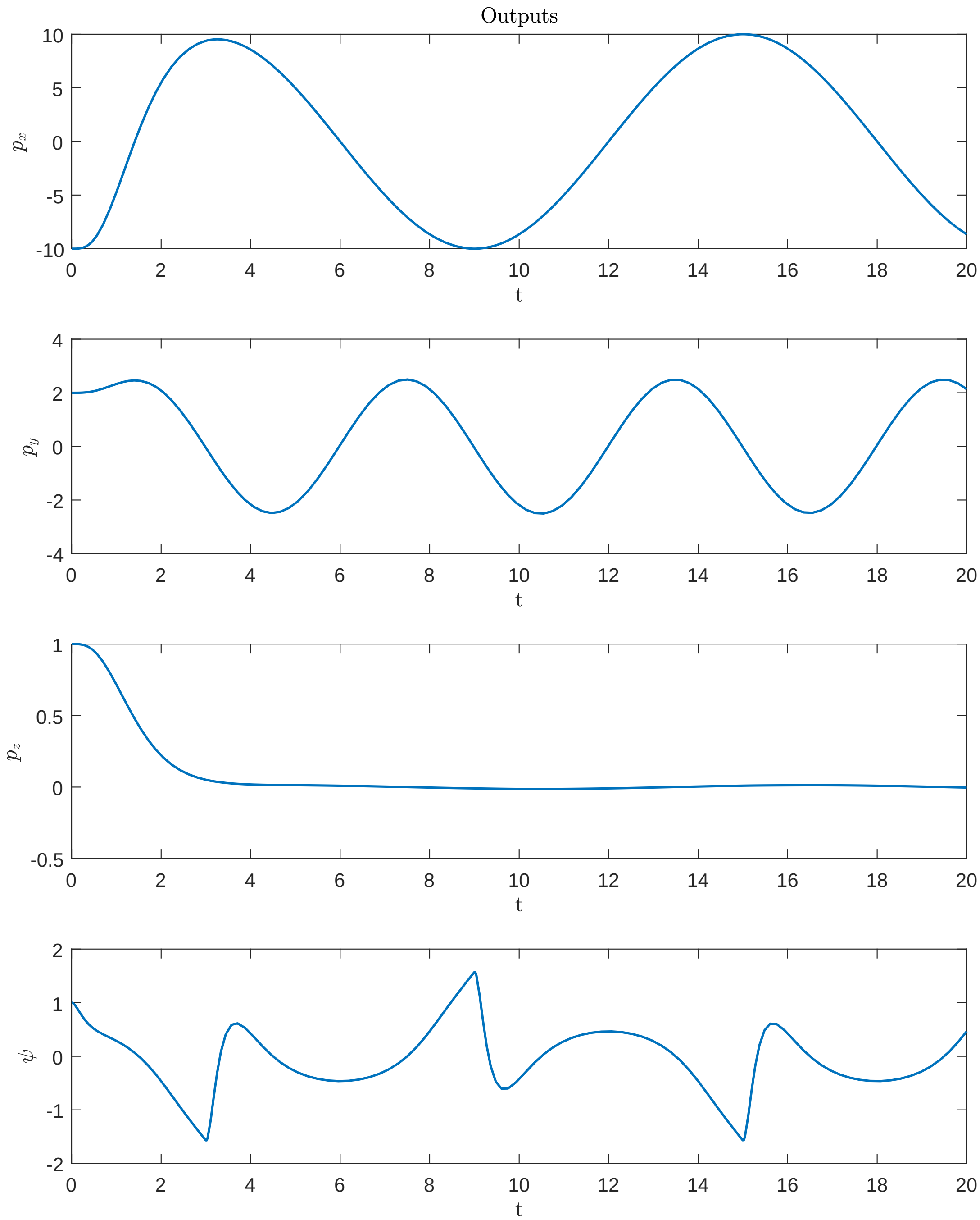

Figure 1.2: Trajectories of the outputs

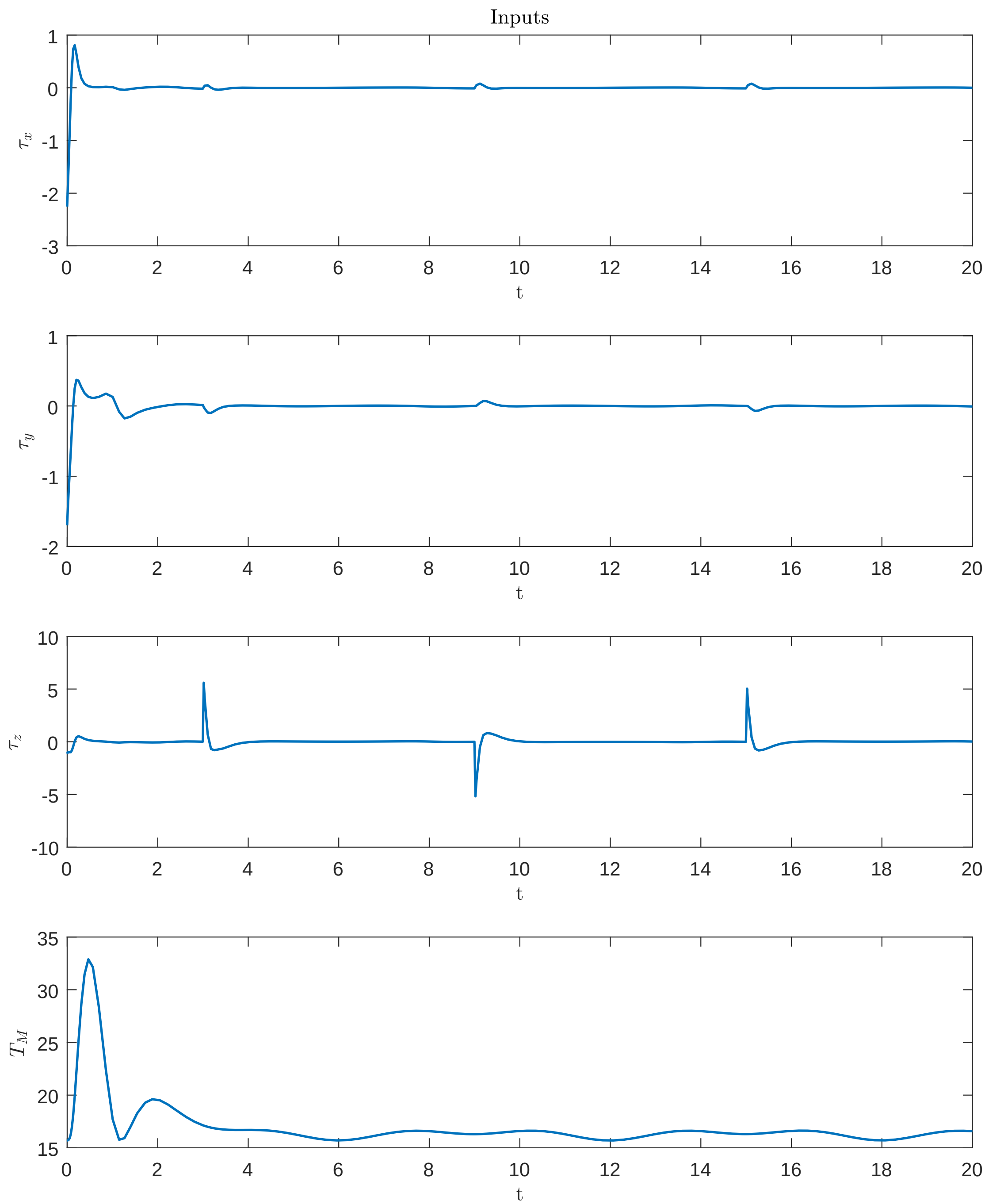

Figure 1.3: Trajectories of the Inputs

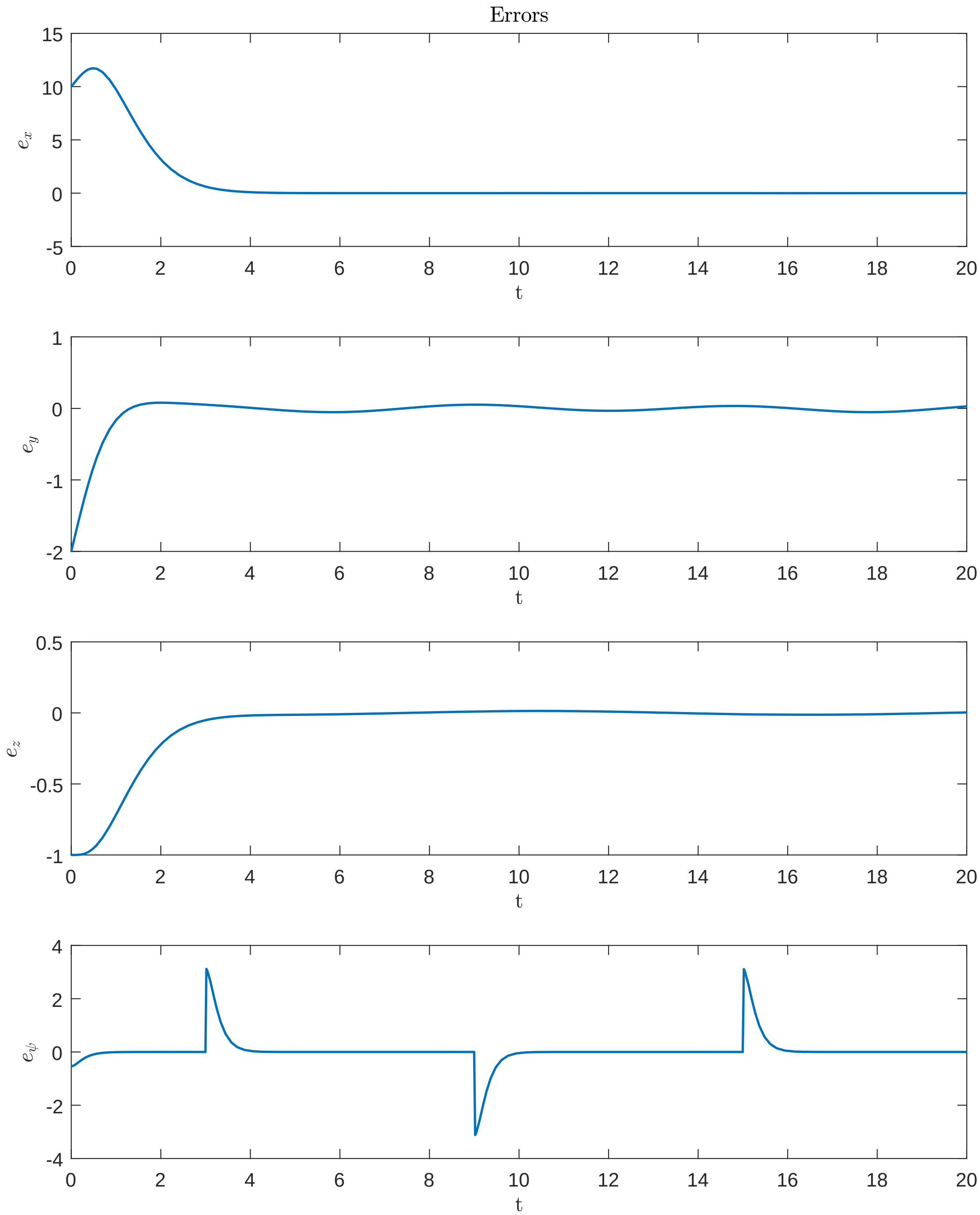

Figure 1.4: Error Trajectories

PID controller are performed where the responses of PID controller can only be validated for attitude.
Fig. 2.1 and fig. 2.2 show that the IOSFBL control is very close to the PID control in behavior and more quick in response for roll and pitch changes. Fig 2.3 shows similar behavior in case of yaw, however, it is has a constant offset due to magnetometer calibration issues on the hardware.
Fig. 2.4 shows the thrust response to various vertical movements and verifies the correct response of the thrust control input to changes in vertical movements.

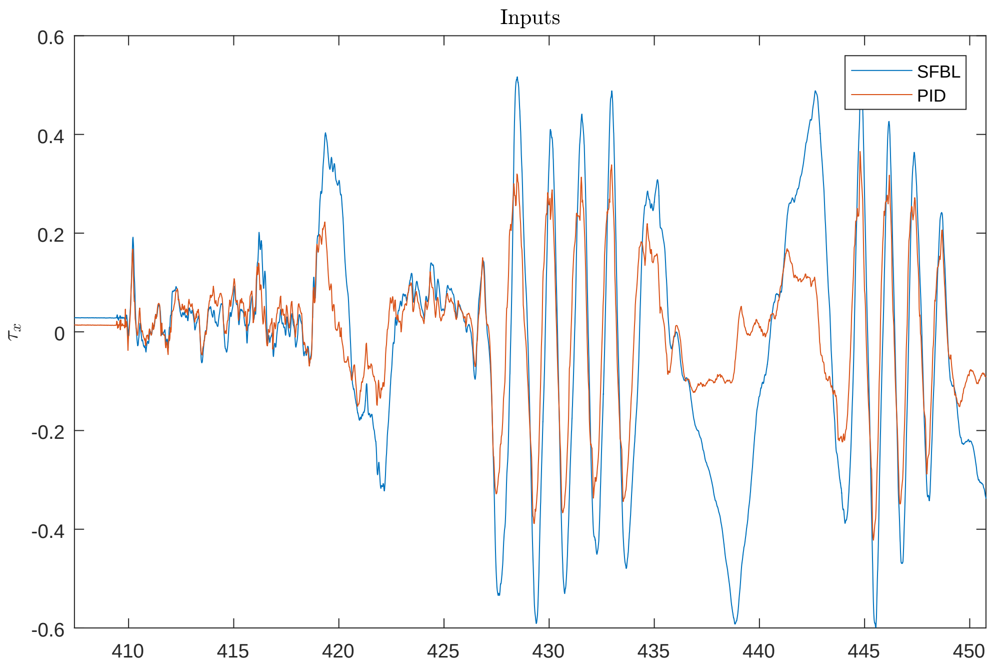

Figure 2.1: τ_x plot responding to changes in roll

2.2 Experiments with Blocks approach

In this section we try to investigate the performance of the controller by doing a hand-held test and compare the control signals of the SFBL controller with the control signals of the default px4 controller. We considered the set point as $[0, 0, -1]$ for position and 0 for the yaw. For doing the hand held test, we just connected the px4v2 to the old quad and placed it on the middle of the lab. We were receiving the real vicon data on the px4 as as well. for logging the data we used a MicroSD card. We started logging the data after starting the mc-TASK module and by using the command logger on. We then just moved the quad around the set point and also did some rotational movement about the axis. You can see the results for the controller signal of both mc-pos-controller and the SFBL controller in Figures 2.5, 2.6, 2.7 and 2.8. It can be seen the response of the SFBL controller is very similar to the mc-pos-controller as the reference controller, so we can say that the SFBl

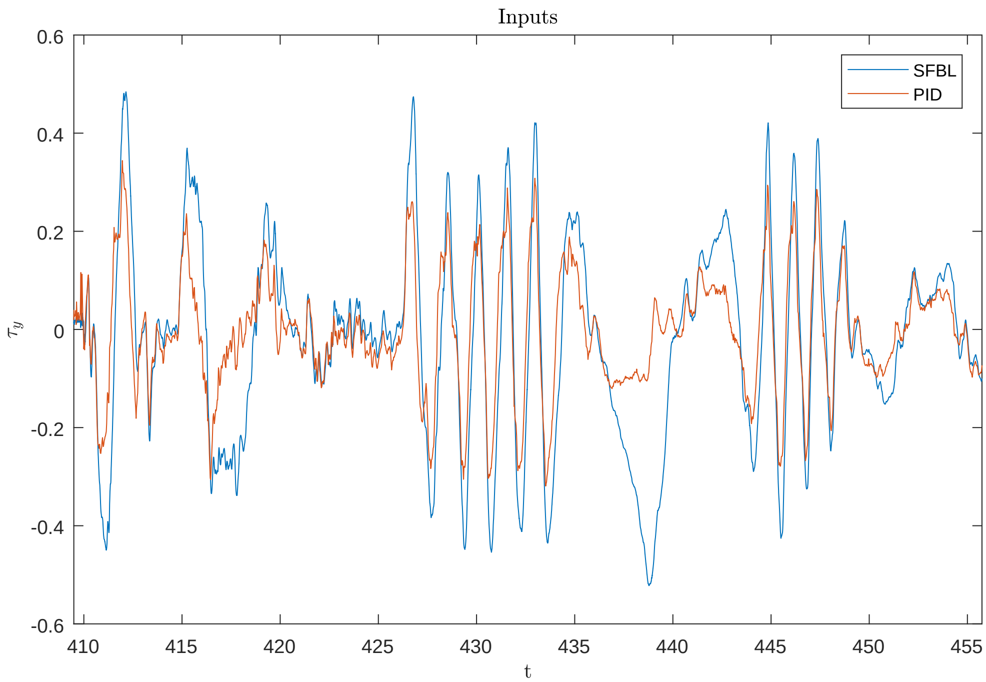

Figure 2.2: τ_y plot responding to changes in pitch

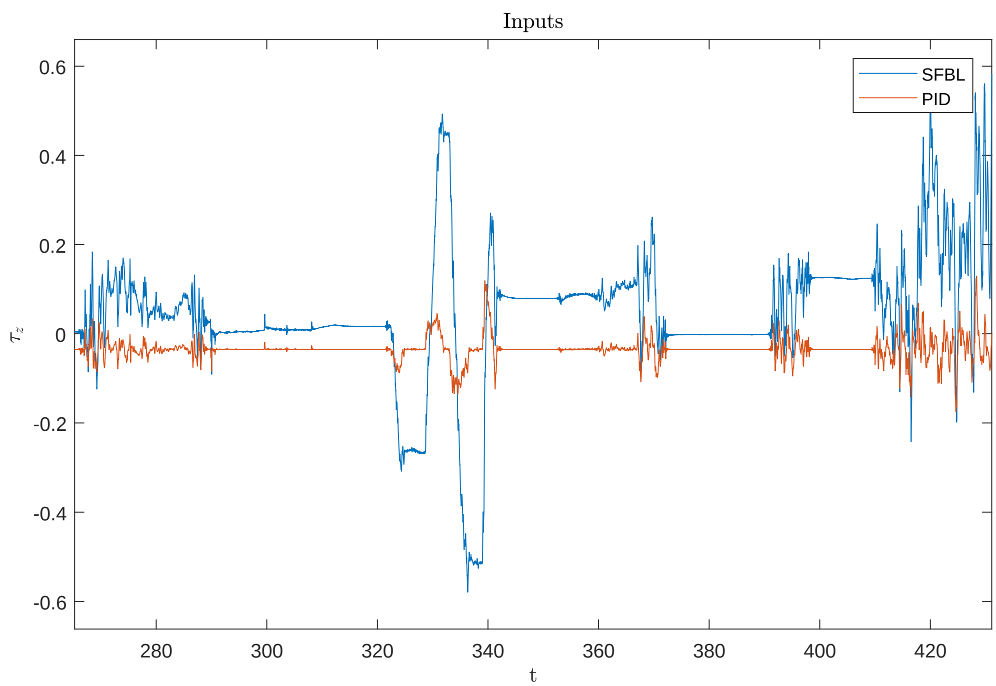

Figure 2.3: τ_z plot responding to changes in yaw

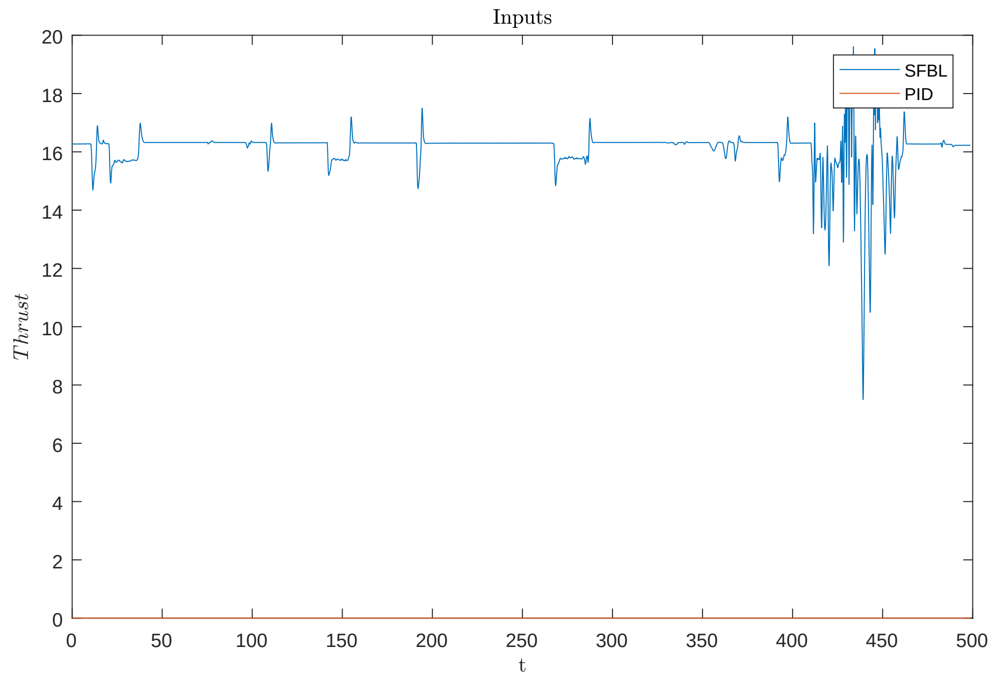

Figure 2.4: T_M plot responding to height changes

controller is working.

3 Backstepping Controller

For the backstepping algorithm, we try to follow the procedure presented in []. However, based on the analysis performed, there are some errors in the algorithm which we try to modify them. The idea behind the backstepping method is to stabilize a system (consisted of subsystems) in a recursive form. From the modeling section, the dynamical equations for the quadrotor is as follows

$$\dot{p} = v \tag{3.1}$$

$$\dot{v} = mge_3 + Re_3(-T) \tag{3.2}$$

$$\dot{\eta} = W(\eta)\omega \tag{3.3}$$

$$J\dot{\omega} = -\omega \times J\omega + \tau \tag{3.4}$$

As we see, the system is in a chained form and we can consider the thrust and the Euler angles as the control input to the translational subsystem and consider the torque as the input to the rotational subsystem. From the equations of the system, we can see that the translational dynamic and the yaw angle are independent, so for these four variables any trajectory could be followed. We assume the desired trajectories for the position and the yaw of the quadrotor are given by $p_d(t)$ and $\psi_d(t)$ (the trajectories must be smooth). As

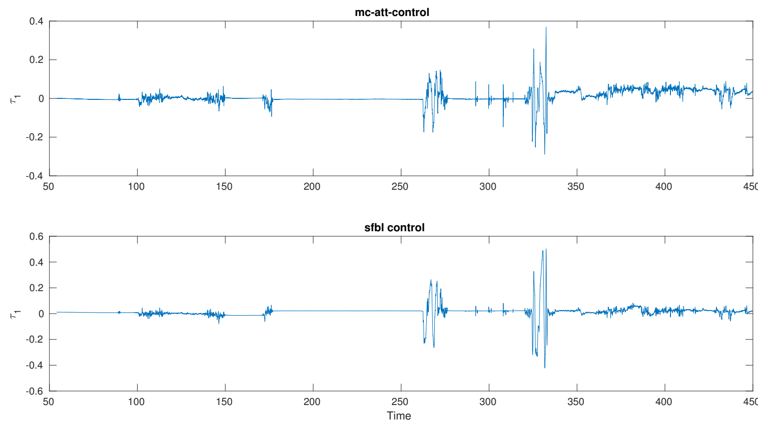

Figure 2.5: τ_1 for mc-pos-control and the SFBL control

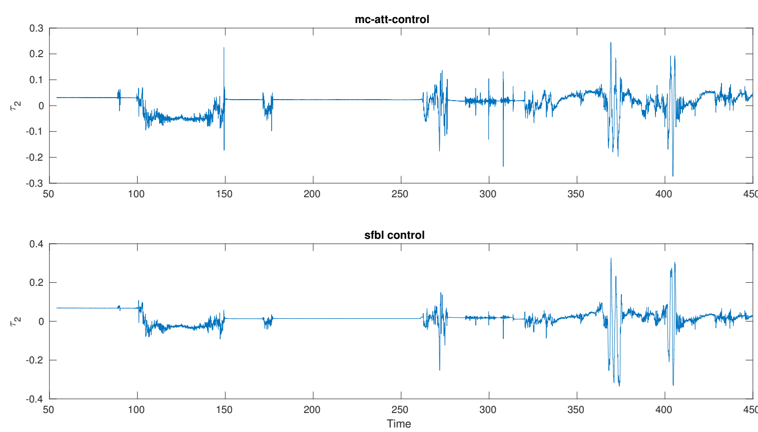

Figure 2.6: τ_2 for mc-pos-control and the SFBL control

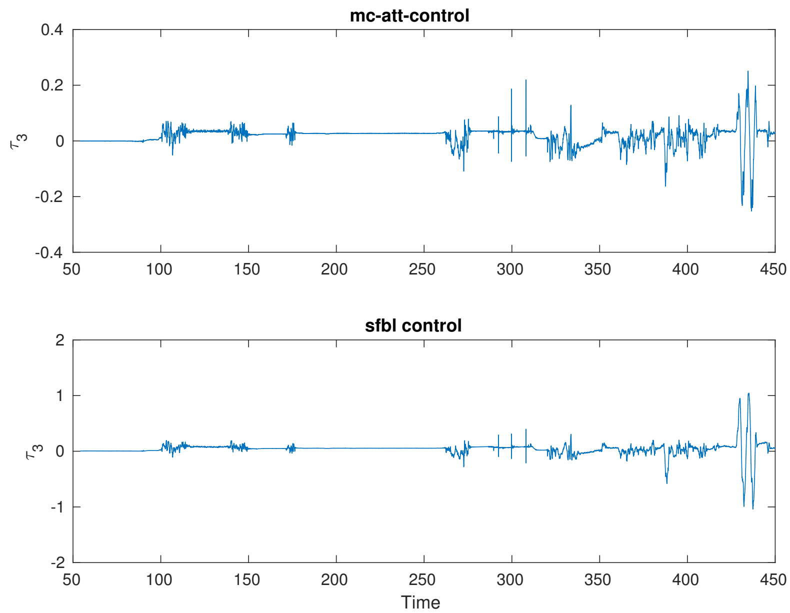

Figure 2.7: τ_3 for mc-pos-control and the SFBL control

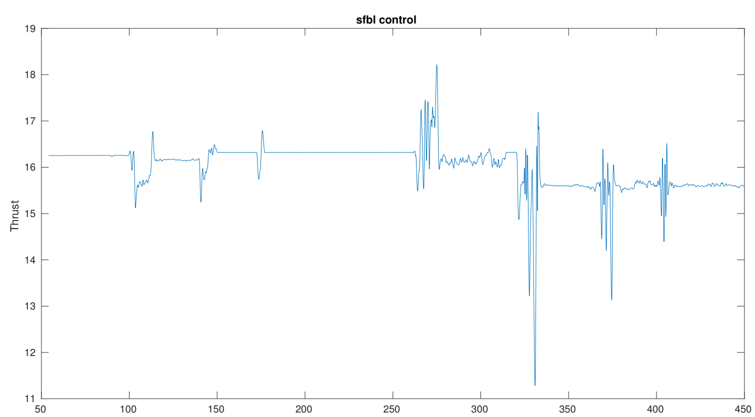

Figure 2.8: Thrust for mc-pos-control and the SFBL control

we know, for tracking a trajectory for position, we need to manipulate the direction of the quadrotor, which this determine trajectories of the roll and pitch. So the problem we are trying to solve is to find the control inputs T, τ_1, τ_2, τ_3, by only measuring the states of the system (p, v, η, Ω), such that the tracking error

$$e = [(p - p_d)^\top, \psi - psi_d]^\top \tag{3.5}$$

goes to zero. If we define the position error as $\delta_1 = p - p_d$, and consider the Lyapunov function candiate as $V_1 = \frac{1}{2}|\delta_1|^2$, we have

$$\dot{V}_1 = \delta_1^\top \dot{\delta}_1 = \delta_1^\top (v - v_d) \tag{3.6}$$

that v_d is the velocity of the desired trajectory. Now, if we consider the translational velocity v as the virtual control v_v to (3.6), and choose it as $v_v = v_d - k_1\delta_1$, where k_1 is a positive constant, then we have for the derivative of the Lyapunov function V_1

$$\dot{V}_1 = -k_1|\delta_1|^2 + \delta_1^\top (v - v_v) \tag{3.7}$$

Now, by defining $\delta_2 = mv - mv_v$ and considering the second Lyapunov function as $V_2 = V_1 + \frac{1}{2}|\delta_2|^2$, we have

$$\dot{V}_2 = -k_1|\delta_1|^2 + \frac{1}{m}\delta_2^\top \delta_1 + \delta_2^\top (-uRe_3 + mge_3 - m\dot{v}_v) \tag{3.8}$$

Now if we consider $(uRe_3)_v$ as the virtual control and consider it as $(uRe_3)_v = X_v = mge_3 - m\dot{v}_v + \frac{1}{m}\delta_1 + k_2\delta_2$ we would get for the derivative

$$\dot{V}_2 = -k_1|\delta_1|^2 - k_2|\delta_2|^2 + \delta_2^\top \delta_3 \tag{3.9}$$

with $\delta_3 = X_v - uRe3$. For the next step we define $\epsilon_3 = \psi - \psi_d$ and consider the third Lyapunov function candiate as

$$V_3 = V_2 + \frac{1}{2}|\delta_3|^2 + \frac{1}{2}|\epsilon_3|^2$$

the reason for introduction of yaw error in this stage is that the relative degree of the yaw and δ_3 with repsect to the inputs are 2. Now by taking derivative of V_3 with respect to time, we get

$$\begin{aligned}
\dot{V}_3 &= -k_1|\delta_1|^2 - k_2|\delta_2|^2 + \delta_3^\top \delta_2 + \delta_3^\top (\dot{X}_v - \dot{u}Re_3 - uRS(\omega)e_3) + \epsilon_3(\dot{\psi} - \dot{\psi}_d) \\
&= -k_1|\delta_1|^2 - k_2|\delta_2|^2 + \delta_3^\top \delta_2 + \delta_3^\top \left(\dot{X}_v - R\begin{bmatrix} u\omega_2 \\ -u\omega_1 \\ \dot{u} \end{bmatrix}\right) + \epsilon_3(\dot{\psi} - \dot{\psi}_d)
\end{aligned}$$

We can assign the value of $\dot{u}$ directly, without the need of using a virtual input because thrust is the actual input control of the system.

$$\dot{u} = e_3^\top R^\top (\dot{X}_v + \delta_2 + k_3\delta_3)$$

If we consider $uRS(\omega)e_3$ and $\dot{\psi}$ as the virtual controls and choose them as

$$\begin{pmatrix} u\omega_2 \\ -u\omega_1 \\ 0 \end{pmatrix}_v = [I - e_3 e_3^\top] R^\top (\dot{X}_v + \delta_2 + k_3\delta_3)$$

$$\dot{\psi}_v = \dot{\psi}_d - k_{31}\epsilon_3$$

Now by considering $Y_v = R[I - e_3e_3^\top]R^\top(\dot{X}_v + \delta_2 + k_3\delta_3)$ and defining $\delta 4 = Y_v - uRS(\omega)e_3$, we have

$$\dot{V}_3 = -k_1|\delta_1|^2 - k_2|\delta_2|^2 - k_3|\delta_3|^2 + \delta_3^\top\delta_4 - k_{31}\epsilon_3^2 + \epsilon_3\epsilon_4$$

with $\epsilon_4 = \dot{\psi} - \dot{\psi}_v$. Consider the new Lyapunov function candidate

$$V_4 = V_3 + \frac{1}{2}|\delta_4|^2 + \frac{1}{2}|\epsilon_4|^2$$

Taking the time derivative results in

$$\dot{V}_4 = -k_1|\delta_1|^2 - k_2|\delta_2|^2 - k_3|\delta_3|^2 + \delta_3^\top\delta_4 - k_{31}\epsilon_3^2 + \epsilon_3\epsilon_4 \tag{3.10}$$
$$+ \delta_4^\top(\dot{Y}_v - (\dot{u}RS(\omega)e_3 - uRS(e_3)\gamma + uRS(\omega)^2e_3)) + \epsilon_4(\ddot{\psi} - \ddot{\psi}_v) \tag{3.11}$$

with $\gamma = \dot{\omega}$ as input transformation. Now if we select γ such that the folowing equations satisfied

$$\dot{u}RS(\omega)e_3 - uRS(e_3)\gamma + uRS(\omega)^2e_3 = \dot{Y}_v + \delta_3 + k_4\delta_4 \tag{3.12}$$

$$\ddot{\psi} = \ddot{\psi}_v - \epsilon_3 - k_{41}\epsilon_4 \tag{3.13}$$

we get for the time derivative of the lyapunov function

$$\dot{V}_4 = -k_1|\delta_1|^2 - k_2|\delta_2|^2 - k_3|\delta_3|^2 - k_4|\delta_4|^2 - k_{31}|\epsilon_3|^2 - k_{41}|\epsilon_4|^2 \tag{3.14}$$

which is negative definite, therefore based on Lyapunov theorem, the errors δ_1, δ_2, δ_3, δ_4, ϵ_1 and ϵ_2 go to zero. So the tracking error converges to zero.
Now if we solve equation (3.12) for γ, we get

$$\gamma_1 = -\frac{e_2^\top R^\top}{u}(\dot{Y}_v - \tilde{u}R\omega_1 Re_3 - uRS(\omega)^2e_3 + \delta_3 + k_5\delta_4) \tag{3.15}$$
$$\gamma_2 = \frac{e_1^\top R^\top}{u}(\dot{Y}_v - \tilde{u}R\omega_1 Re_3 - uRS(\omega)^2e_3 + \delta_3 + k_4\delta_4) \tag{3.16}$$

Now we try to solve the equation (3.13). From $\dot{\eta} = W(\eta)\omega$, we can find that

$$\ddot{\psi} = e_3^\top\dot{W}(\eta)\omega + \frac{\sin(\phi)}{cos(\theta)}\gamma_2 + \frac{\cos(\phi)}{\cos(\theta)}\gamma_3 \tag{3.17}$$

so by substituating (3.17) in (3.13) and solve for γ_3 we get,

$$\gamma_3 = \frac{\cos(\theta)}{\cos(\phi)}(\ddot{\psi}_v - \epsilon_3 - k_{41}\epsilon_4 - e_1^\top \dot{W}(\eta)\omega - \frac{\sin(\phi)}{\cos(\theta)\gamma_2}) \tag{3.18}$$

from the rotational dynamic (3.4), we can find that for having $\dot{\omega} = \gamma$ as (3.15), (3.16) and (3.18), we have torgue as

$$\tau = J\gamma + \omega \times J\omega$$

So the control equations for torgue and thrust are as follows

$$\gamma_1 = -\frac{e_2^\top R^\top}{u}(\dot{Y}_v - \tilde{u}R\omega_1 Re_3 - uRS(\omega)^2 e_3 + \delta_3 + k_5\delta_4) \tag{3.19}$$

$$\gamma_2 = \frac{e_1^\top R^\top}{u}(\dot{Y}_v - \tilde{u}R\Omega_1 Re_3 - uRS(\omega)^2 e_3 + \delta_3 + k_4\delta_4) \tag{3.20}$$

$$\gamma_3 = \frac{\cos(\theta)}{\cos(\phi)}(\ddot{\psi}_v - \epsilon_3 - k_{41}\epsilon_4 - e_1^\top \dot{W}(\eta)\omega - \frac{\sin(\phi)}{\cos(\theta)\gamma_2}) \tag{3.21}$$

$$\dot{u} = e_3^\top R^\top(\dot{X}_v + \delta_2 + k_3\delta_3) \tag{3.22}$$

3.1 Simulation Results

For investigating the performance of the desinged Backstepping controller we have performed a simulation in Matlab/Simulink. The block diagram of the simulation in the Simulink environment is shown in the following figure. The trajectory to be follwoed is $p_d(t) = [10\sin(2\pi t/T_0),\ 10\sin(4\pi t/T_0)/4,\ -1]$ and $\psi_d(t) = \mathrm{atan}((\frac{0.5\cos(4\pi t/T_0)}{\cos(2\pi t/T_0)})$. The initial value for the position and Euler angles of the quadrotor are considered as $p(0) = [3; 3; 0]^\top$ and $\eta = [0; 0; 0]^\top$, respectively. The Inertia matrix and the weight of the system are also considered as $J = \mathrm{diag}(0.03, 0.03, 0.03)$ and $m = 1.6$ .The controller gains also are considered as $k_1 = 1.2$, $k_2 = 0.9$, $k_3 = 0.9$, $k_{31} = 0.9$, $k_4 = 0.8$ and $k_{41} = 0.8$. The tracking error is shown in Fig.3.2. It can be seen that all errors converge to zero asymptotically. The actual and desired trajectories in x-y plane are shown in Fig.3.3.

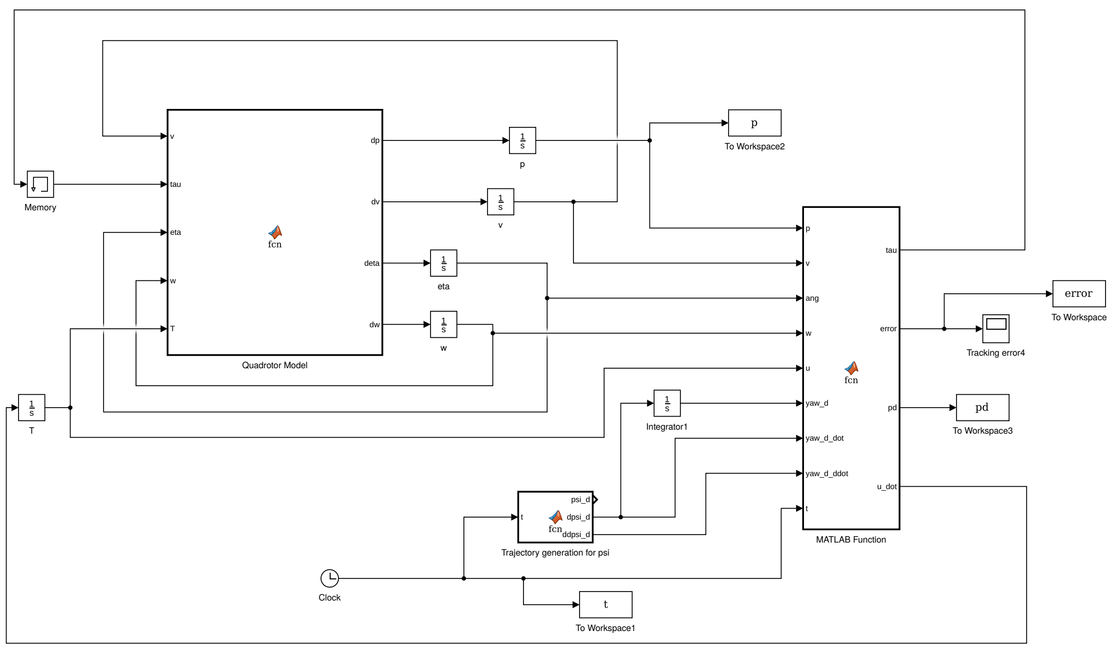

Figure 3.1: Simulink Model

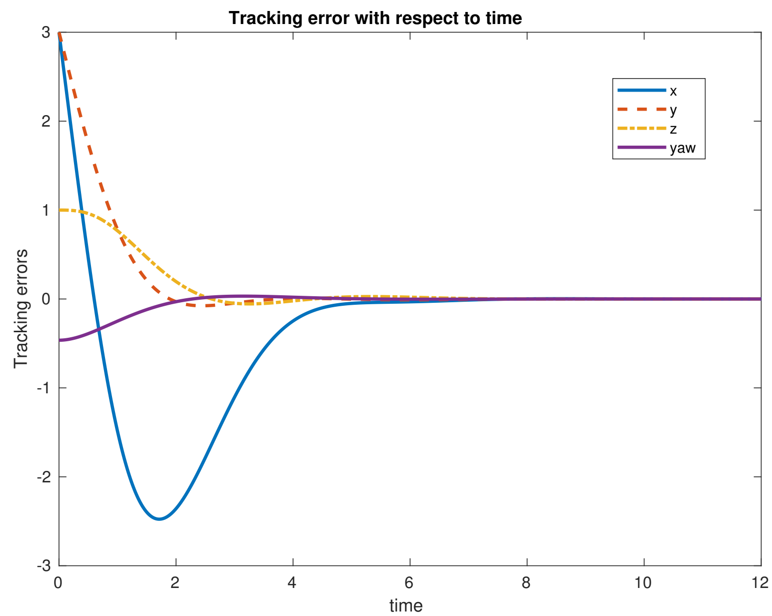

Figure 3.2: Tracking Error

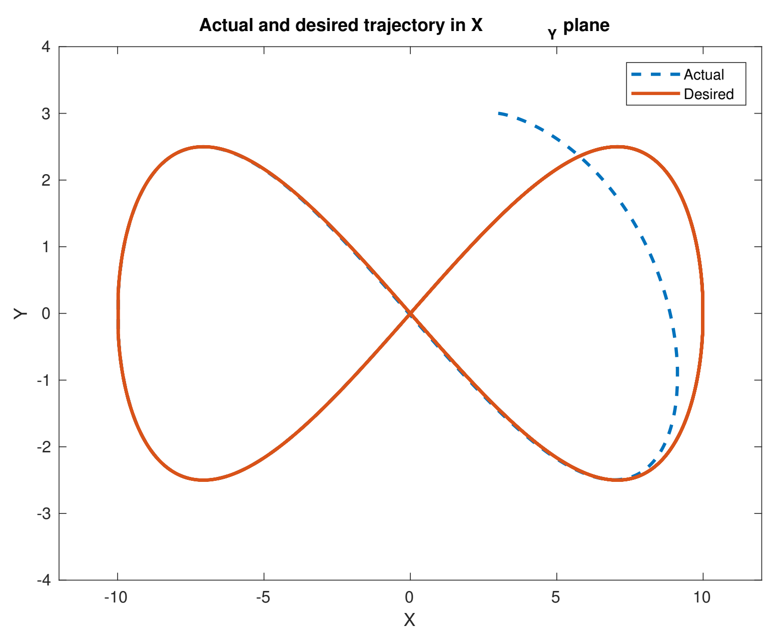

Figure 3.3: Actual and desired trajectory

4 Inner-Outer Loop Controller

In this section we describe the derivation of the Inner Outer Loop Controller and investigate its performance by simulation. Inner-Outer loop control structure is one of the common controller used for controlling the quadrotor. In this case, the controller is divided to two loop. The Inner loop controls the orientation of the vehicle and the outer loop calculates the thrust and also provides the reference trajectories for the roll and pitch of the vehicle. This structure can be seen in Fig. 4.1 As we know, at hover, the set point of the quadrotor

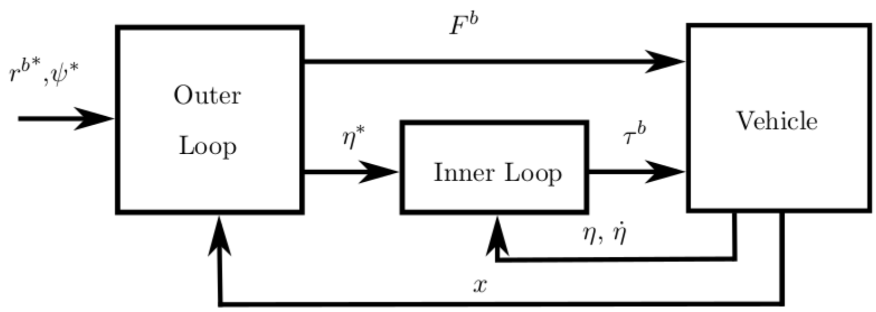

Figure 4.1: Inner Outer Loop Structure

is $p = p_s$, $v = 0$, $\eta = [0, 0, \psi_s]^\top$ and $\omega = 0$. Also we have $T = mg$ So approximating the dynamics of the system in the setpoint results in the following equations

$$\dot{p} = v \tag{4.1}$$
$$\dot{v} = mge_3 + R_\psi e_3(-T) \tag{4.2}$$
$$\dot{\eta} = W(\eta)\omega \tag{4.3}$$
$$J\dot{\omega} = -\omega \times J\omega + \tau \tag{4.4}$$

$$m\dot{v} = R_\psi \begin{pmatrix} -\cos(\phi)\sin(\theta) \\ sin(\phi) \\ -\cos(\phi)cos(\theta) \end{pmatrix} T + mge_3 \approx R_\psi \begin{pmatrix} -mg\theta \\ mg\phi \\ -T + mg \end{pmatrix}$$

and

$$\dot{\eta} = W(\eta)\omega = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{cos(\theta)} \end{bmatrix} \omega \approx \omega$$

and

$$J\dot{\omega} = -\omega \times J\omega + \tau \approx \tau$$

because the rotational dynamics is much faster than the translational dynamic, we can consider the roll and pitch anlges as the input to the translational dynamics.

So from the approximated equation we try to design a PID controller and calculate the ϕ, θ and T for controlling the translational subsystem.

$$m\dot{v} = m\ddot{p} = R_\psi \begin{pmatrix} -mg\theta \\ mg\phi \\ -T + mg \end{pmatrix} = m\ddot{p}_d - k_d\dot{\tilde{p}}_d - k_p\tilde{p} - k_i \int_0^t \tilde{p}(\alpha)d\alpha$$

which results in

$$\begin{pmatrix} mg\theta^* \\ mg\phi^* \\ -T^* + mg \end{pmatrix} = R_\psi^\top (m\ddot{p}_d - k_d\dot{\tilde{p}}_d - k_p\tilde{p} - k_i \int_0^t \tilde{p}(\alpha)d\alpha)$$

from the approximate equations the dynamic of the rotational subsystem is $J\ddot{\eta} = \tau$. Now we try to design a PID controller for controlling the Euler angles to their desired values (ϕ^* and θ^* from the outer loop and ψ^* is from the set point.)

$$\ddot{\eta} = J^{-1}\tau = \begin{pmatrix} -k_{d\phi}\dot{\tilde{\phi}} - k_{p\phi}\tilde{\phi} - k_{i\phi}\int_0^t \tilde{\phi}(\alpha)d\alpha \\ -k_{d\theta}\dot{\tilde{\theta}} - k_{p\theta}\tilde{\theta} - k_{i\theta}\int_0^t \tilde{\theta}(\alpha)d\alpha \\ -k_{d\psi}\dot{\tilde{\psi}} - k_{p\psi}\tilde{\psi} - k_{i\psi}\int_0^t \tilde{\psi}(\alpha)d\alpha \end{pmatrix}$$

with $\tilde{\phi} = \phi - \phi^*$, $\tilde{\theta} = \theta - \theta^*$ and $\tilde{\psi} = \psi - \psi^*$.
For investigating the performance of the derived controller, we have performed a simulation In MATLAB/Simulink. The given trajectory is $p_d(t) = [10\sin(2\pi t/T_0),\ 10\sin(4\pi t/T_0)/4,\ -1]$ and $\psi_d(t) = \text{atan}((\frac{0.5\cos(4\pi t/T_0)}{\cos(2\pi t/T_0)})$. The controller gains for the outer loop controller are $k_p = 1$, $k_d = 1$, $k_i = 0.2$ and for the inner loop controller $k_p = 2$, $k_d = 2$, $k_i = 2$. The tracking error and the actual and desired trajectories are shown in Fig. 4.2 and Fig. 4.3, respectively. As it can be observed the performance of the controller is not acceptable. The reason is that we used an approximate model in designing the controller which as the results show has negative effect on tracking task. Actually the approximate model used was for hovering, but what we are doing here is trajectory tracking which doubles the negative effect of approximation.
However, by decreasing the frequncy of the trajectory by the factor of 2 ($T_0 = 24$), we can see that the performance of the controller is increased (Fig. 4.4 and 4.5)

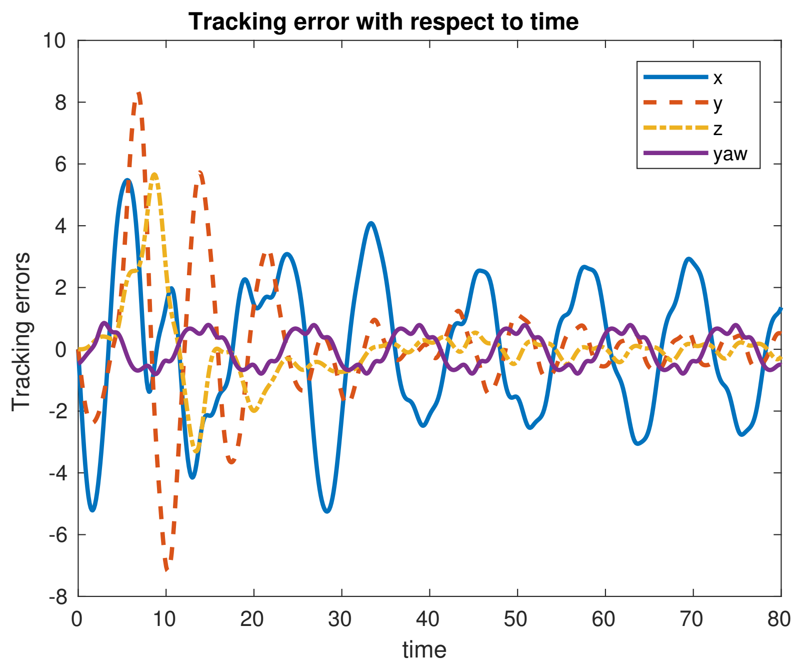

Figure 4.2: Tracking error

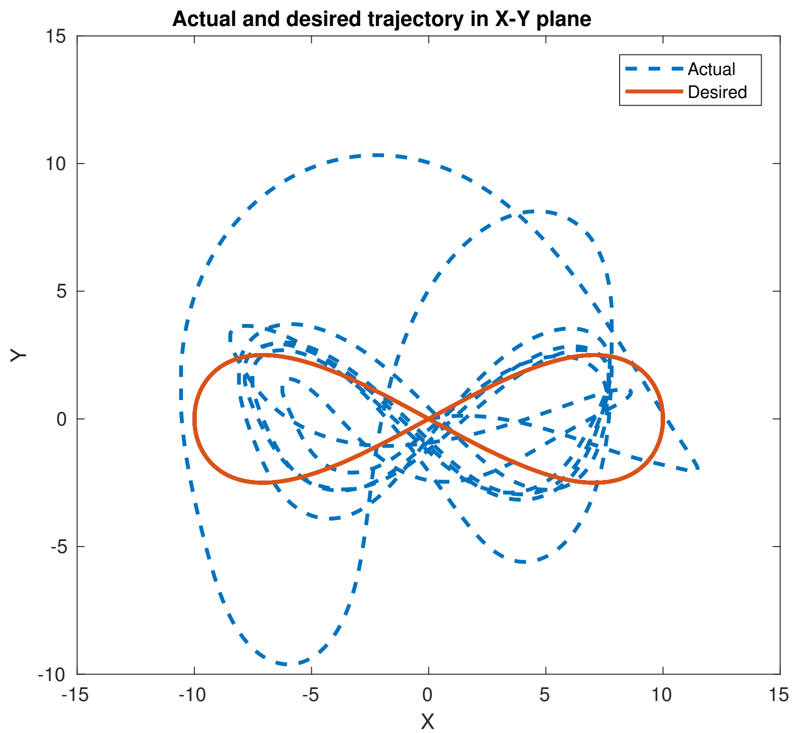

Figure 4.3: Actual and Desired trajectories in x-y plane

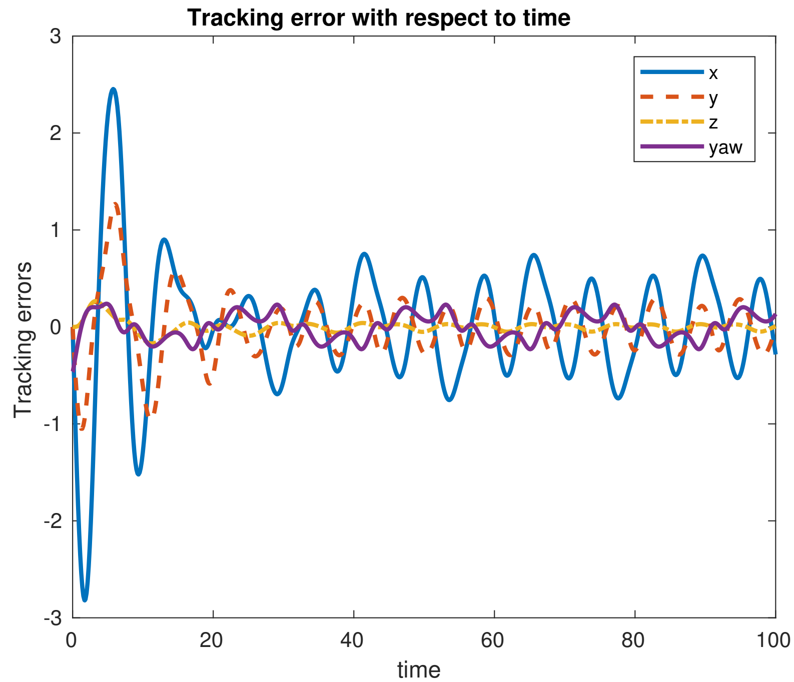

Figure 4.4: Tracking error, $T_0 = 24$

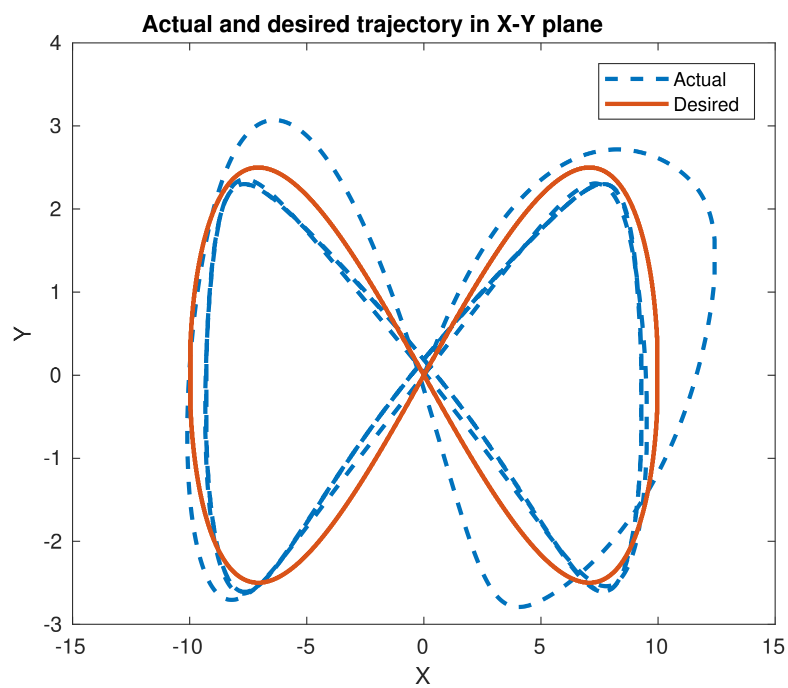

Figure 4.5: Actual and Desired trajectories in x-y plane, $T_0 = 24$