Sabancı Üniversitesi

ME425 Homework 2

MOTION CONTROL OF A FLYING ROBOT (QUADROTOR)

EDIN GUSO 23435

CONTENTS

1) INTRODUCTION

In this project I have worked on the motion control of a flying robot. The flying robot is a quadrotor and as it can be understood from the name, it has 4 motors with propellers attached to them. I had two different control tasks in this project. One of them was to make the quadrotor fly upwards and hover, while the second one was controlling the quadrotor in any direction given. In order to do those tasks, I modelled the quadrotor's dynamics in Simulink and designed two different control algorithms for each task. These algorithms are obviously very different from each other. Therefore, most of the sections in this report will be divided into two parts. Implementation, difficulties, assumptions and results of both of the control algorithms will be discussed in detail in the following parts of the report.

2) PROCEDURE

a) SYSTEM DYNAMICS

The first step before controlling the system is of course, modelling the system. In order to model the system correctly I had to know the physical parameters such as mass and inertia of the quadrotor. I have used the values which were used in Control of a Hovering Quadrotor UAV Subject to Periodic Disturbances (Zaki, H., Unel, M., 2018)[1]. The values I used were as follows:

$$m = 1\ kg, \quad g = 9.8\frac{m}{s^2}, \quad I_{xx} = 0.1kg\ m^2, \quad I_{yy} = 0.1kg\ m^2, \quad I_{zz} = 0.15kg\ m^2$$

I used the system dynamics equations which were derived during our lectures and were provided in the lecture slides. These dynamics model the behavior of the physical system with high precision. Dynamic model equations can be seen below:

$$\ddot{X} = (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)\frac{U_1}{m}$$

$$\ddot{Y} = (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi)\frac{U_1}{m}$$

$$\ddot{Z} = -g + (\cos\theta \cos\phi)\frac{U_1}{m}$$

$$\dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}}\ q\ r + \frac{U_2}{I_{XX}}$$

$$\dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}}\ p\ r + \frac{U_3}{I_{YY}}$$

$$\dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}}\ p\ q + \frac{U_4}{I_Z}$$

Above mentioned equations' outputs are the states of the system. Three of these states, the linear acceleration components are already given in the earth frame. However, the other three states, the angular accelerations are given in the body frame and therefore must be transformed into the earth frame. The transformation matrix can be seen below:

$$T = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}$$

But since the transformation matrix maps angular velocities into Euler rates, we first need to take the integral of the angular accelerations. When the angular velocities are acquired the transformation matrix can be used to transform them into Euler rates as shown below:

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

After this conversion, it is possible to take the integral once more and obtain the Euler angles. Also, it is trivial to get linear velocities and positions by integrating the linear accelerations since they are in the earth frame (The more detailed view at the modelled system dynamics in Simulink can be seen in Appendix). All these obtained states are used in the control algorithms which will be explained later.

b) CONTROL IDEA

Since it would be very difficult trying to control the rotors individually in order to achieve desired motions, the control inputs are abstracted. Instead of having 4 individual rotors as inputs, the inputs are defined as total thrust, rolling moment, pitching moment and yawing moment. These control inputs are defined with U and can be calculated in terms of rotor speeds Ω.

$$U_1 = b\,(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2)$$

$$U_2 = l\,b\,(-\Omega_2{}^2 + \Omega_4{}^2)$$

$$U_3 = l\,b\,(-\Omega_1{}^2 + \Omega_3{}^2)$$

$$U_4 = b\,(-\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 + \Omega_4{}^2)$$

Since these are 4 equations with 4 unknowns it is possible to solve these equations for rotor speeds with given control inputs. Solved equations can be seen below:

$$\Omega_1{}^2 = \frac{1}{4\,b}\,U_1 - \frac{1}{2\,b\,l}\,U_3 - \frac{1}{4\,d}\,U_4$$
$$\Omega_2{}^2 = \frac{1}{4\,b}\,U_1 - \frac{1}{2\,b\,l}\,U_2 + \frac{1}{4\,d}\,U_4$$
$$\Omega_3{}^2 = \frac{1}{4\,b}\,U_1 + \frac{1}{2\,b\,l}\,U_3 - \frac{1}{4\,d}\,U_4$$
$$\Omega_4{}^2 = \frac{1}{4\,b}\,U_1 + \frac{1}{2\,b\,l}\,U_2 + \frac{1}{4\,d}\,U_4$$

As it is possible to easily move from control inputs to rotor speeds, from now on, all the control algorithm will be discussed based on these control inputs.

c) HOVER CONTROL

The hover control task is the easier task of the two control tasks. It is easier since 4 degrees of freedom are being controlled by 4 inputs. Heaving equal number of DOF and inputs makes the control problem very straightforward.

In hover control, system dynamics which are desired to be controlled are altitude and attitude. Altitude is the position of the quadrotor on the Z-axis. Attitude is the quadrotor's orientation, which is composed of roll, pitch and yaw angles.

At this point, it would be difficult to try to control these states if they were to be controlled directly by individual rotors, but since the U control inputs have been defined, the control problem is an easy task. Total thrust will control the quadrotor's altitude. Rolling moment will control the roll angle, pitching moment will control the pitch angle and yawing moment will control the yaw angle.

For every control input a simple PD control was used. In order to improve the control quality and reduce its dependency on system parameters, gravity, mass and inertias were included as feedforward terms. Also, the desired acceleration in Z-axis was included as a feedforward term to improve the system performance. And obviously errors in position and velocity were used as feedback terms in the control algorithm. The control law for each of the control inputs can be seen in the equations below:

$$U_1 = m\,(g + \ddot{Z}_d + K_{p_Z} e_Z + K_{d_Z} \dot{e}_Z)$$

$$U_2 = I_{XX}\,(K_{p_\phi} e_\phi + K_{d_\phi} \dot{e}_\phi)$$

$$U_3 = I_{YY}\,(K_{p_\theta} e_\theta + K_{d_\theta} \dot{e}_\theta)$$

$$U_4 = I_{ZZ}\,(K_{p_\psi} e_\psi + K_{d_\psi} \dot{e}_\psi)$$

The implementation of these control laws can be seen in the Appendix A: Hover Control in HoverControlModel/Controller and the Matlab function blocks inside it. The results of these control laws will be discussed in detail in the following parts.

d) TRAJECTORY TRACKING CONTROL

The problem of trajectory tracking is not as straightforward as hovering. The reason behind that is there are 6 degrees of freedom that need to be controlled in trajectory tracking control while there are still 4 inputs. These degrees of freedom are movements in X-axis, Y-axis, Z-axis and roll, pitch yaw angles. Having more DOF than control inputs requires a more complicated approach than just assigning control inputs to each DOF.

The idea of using total thrust, rolling moment, pitching moment and yawing moment as control inputs instead of individual rotors will be used in this control, too. But the idea of virtual control will be used together with it. Virtual control is basically imagining the quadrotor as a single point and defining three force vectors, one in each axis, which will define the desired movement of the robot. For example, if we want the quadrotor to move in the X direction, desired force vectors in Y-axis and Z-axis will have a magnitude of zero while the desired force vector in X-axis will have a non-zero magnitude.

These virtual force vectors are being calculated by simple PD control in all three directions and feedforwarding the desired accelerations for better performance. The control law for the virtual controls can be seen below:

$$\mu_X = \ddot{X}_d + K_{p_X} e_X + K_{d_X} \dot{e}_X$$

$$\mu_Y = \ddot{Y}_d + K_{p_Y} e_Y + K_{d_Y} \dot{e}_Y$$

$$\mu_Z = \ddot{Z}_d + K_{p_Z} e_Z + K_{d_Z} \dot{e}_Z$$

Having these three virtual forces being applied on the quadrotor, they are used to calculate the desired total thrust, desired rolling angle and desired pitching angle of the quadrotor. This idea is very useful because by implementing this the desired roll and pitch angles depend on virtual controls. Since these two angles are bounded by other parameters, the DOF to be controlled is more like 4 DOF again. The following equations are used for converting these virtual control values into total thrust and desired angles:

$$U_1 = m\sqrt{{\mu_X}^2 + {\mu_Y}^2 + (\mu_Z + g)^2}$$

$$\phi_d = sin^{-1}\left(\frac{sin\,\psi_d\,\mu_X - cos\,\psi_d\,\mu_Y}{\sqrt{{\mu_X}^2 + {\mu_Y}^2 + (\mu_Z + g)^2}}\right)$$

$$\phi_d = sin^{-1}\left(\frac{cos\,\psi_d\,\mu_X + sin\,\psi_d\,\mu_Y}{cos\,\phi_d\,\sqrt{{\mu_X}^2 + {\mu_Y}^2 + (\mu_Z + g)^2}}\right)$$

After this conversion, since desired roll and pitch angles are obtained and the desired yaw angle is set to a desired value by the user, the rest of the control is very simple. The same idea for the control of angles as in hover control can be applied here. A simple PD control shown below can control the system effectively.

$$U_2 = I_{XX}\,(K_{p_\phi} e_\phi + K_{d_\phi} \dot{e}_\phi)$$

$$U_3 = I_{YY}\,(K_{p_\theta} e_\theta + K_{d_\theta} \dot{e}_\theta)$$

$$U_4 = I_{ZZ}\,(K_{p_\psi} e_\psi + K_{d_\psi} \dot{e}_\psi)$$

Implementation of all these steps for trajectory tracking control can be seen in the Appendix B: Trajectory Tracking Control in TrajectoryTrackingControlModel/Controller and the Matlab function blocks inside it. The results will be discussed in detail in the following parts.

e) DISTURBANCE MODELLING

In order to observe the effects of disturbances on my system, I have tried to model a disturbance which would behave like wind. Even though my disturbance model is far from perfect, it allows me to see how my control can compensate outside interferences. I decided to apply disturbance only on Euler angles. I used the rand function which returns a value between 0 and 1 in Matlab to model my disturbance. Also, I defined a coefficient k_η which would allow me to change the disturbance applied easily. My final model was as follows:

$$\eta_\phi = (rand - 0.5)\, k_\eta$$

$$\eta_\theta = (rand - 0.5)\, k_\eta$$

$$\eta_\psi = (rand - 0.5)\, k_\eta$$

This model caused $(-0.5,\ 0.5)\, k_\eta$ rad disturbance to be applied to every Euler angle every step time. The results of these disturbances will be discussed in the following parts.

f) SMOOTH TRAJECTORY PLANNING

In order to avoid the quadrotor performing high acceleration movements, there is a need for smooth trajectory planning. That is done by using initial and final position, velocity and acceleration values of a desired trajectory and creating a 5th order polynomial using those. In order to find the polynomial, it is required to find its coefficients. For finding the coefficients, the following formula is used:

$$T = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \mathrm{inv}(T) \begin{bmatrix} q_0 \\ v_0 \\ \alpha_0 \\ q_f \\ v_f \\ \alpha_f \end{bmatrix}$$

After finding the 5th order desired position trajectory polynomial, its derivative can be taken once and twice in order to get the 4th order velocity trajectory polynomial and the 3rd order acceleration trajectory polynomial, respectively. After finding these polynomials, they are used to give smooth reference signals for the controllers. This causes quadrotor's movements to be smooth and natural.

It is important to note that this method has to be used three times for the trajectory tracking control as there are three axis in which the reference signal has to be computed.

3) RESULTS

a) HOVER CONTROL

For each case in hover control, 4 different data sets have been plotted in order to examine the performance and stability of the system. First plot contains position, velocity and acceleration in Z-axis simulation results vs reference signal. Second plot is the position in X-axis vs position in Y-axis. Third plot shows the attitude (roll, pitch yaw angles) vs time and finally the fourth plot contains the control inputs vs time graphs.

i) Case 1: Flat start

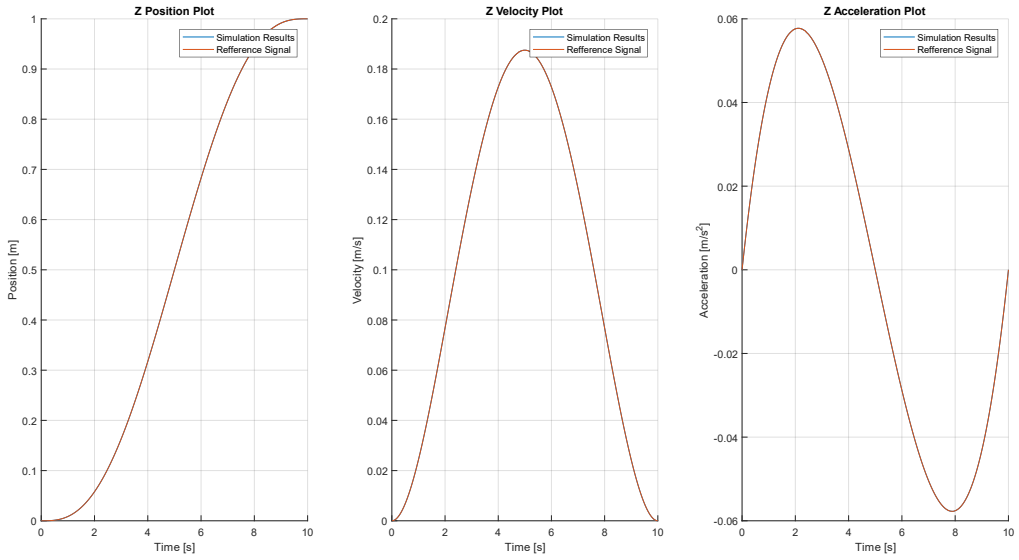
Conditions:

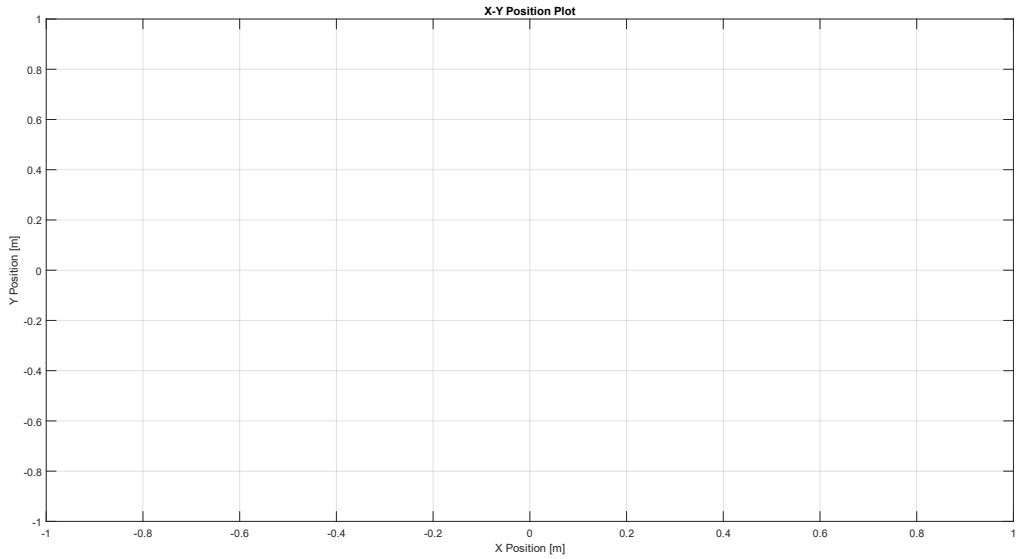
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 8 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 6 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 5 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 2.5 \quad K_{d_\psi} = 4$$

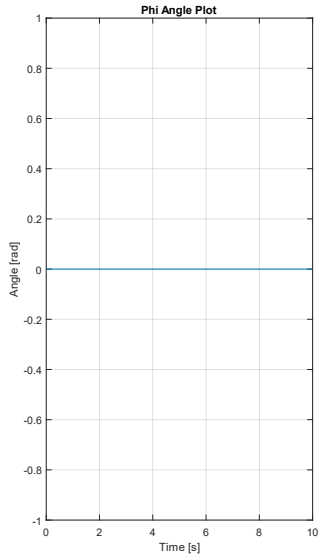
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$

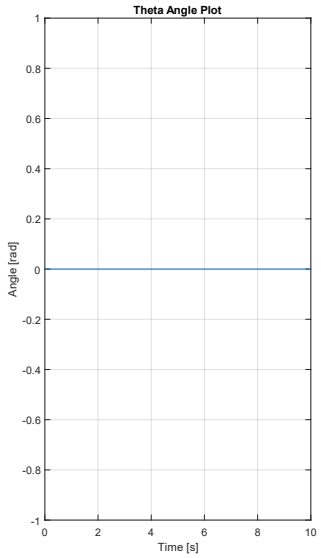
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0 \quad Z_f = 1$$
$$k_\eta = 0$$

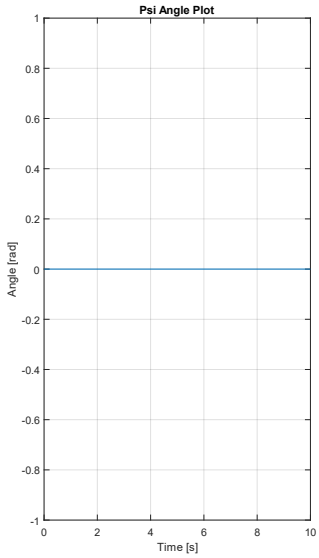
Graphs:

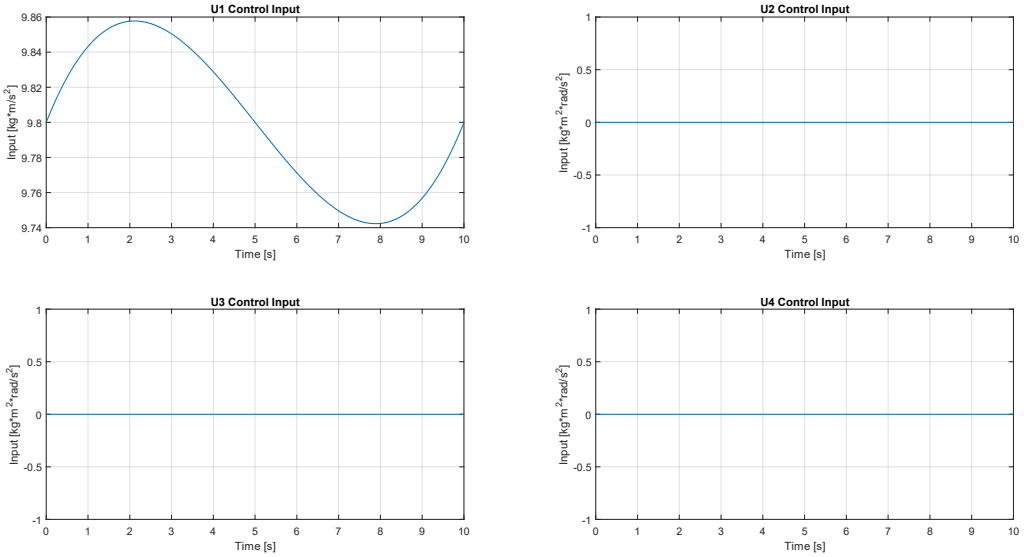



X-Y Position Plot
Y Position [m]
X Position [m]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Phi Angle Plot
Angle [rad]
Time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10


Theta Angle Plot
Angle [rad]
Time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10


Psi Angle Plot
Angle [rad]
Time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10



In the first case of hover control, since there is no disturbance acting on the system and the quadrotor is initially flat, the quadrotor has no change in its attitude and no movement in X-axis and Y-axis. Also, the only control input different from 0 is the total thrust input. This is to be expected since error in attitude starts from 0 and does not change. The quadrotor can very efficiently track the reference signal in Z-axis and is reaching the goal position with no noticeable error.

ii) Case 2: Flat start with disturbance

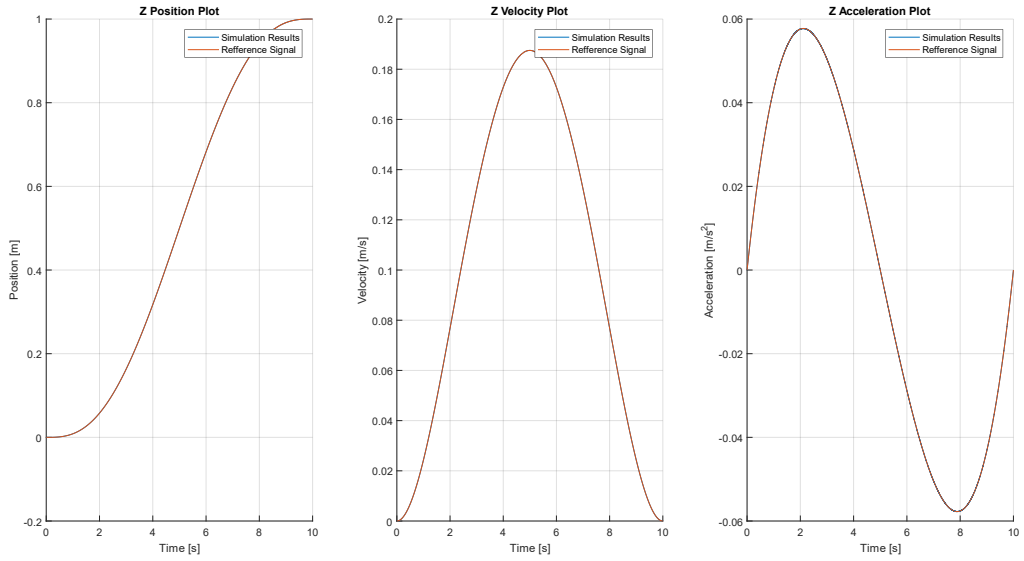
Conditions:

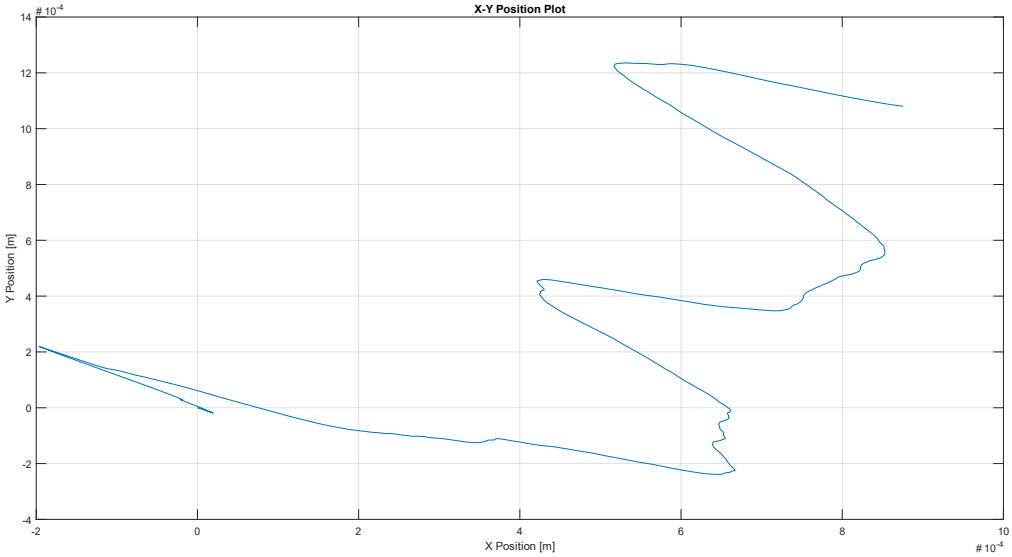
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 8 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 6 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 5 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 2.5 \quad K_{d_\psi} = 4$$

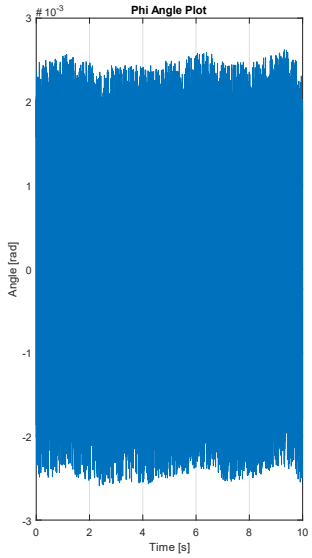
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$

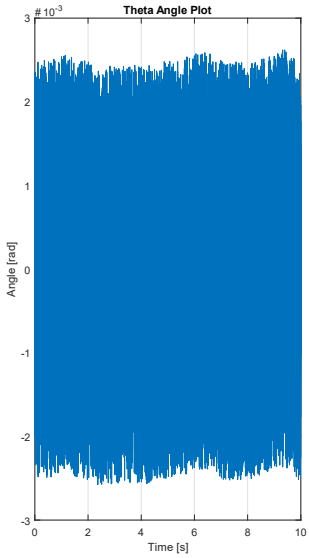
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0 \quad Z_f = 1$$
$$k_\eta = 0.005$$

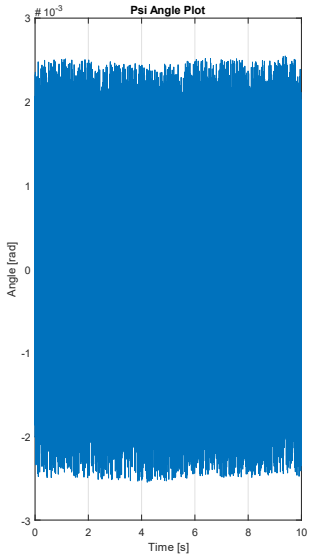
Graphs:

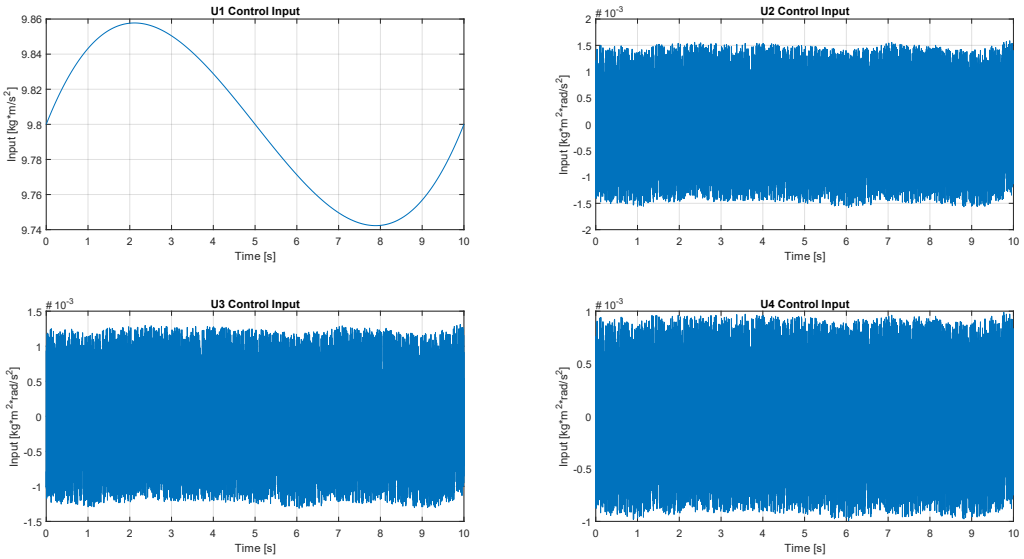



X-Y Position Plot
10^-4
Y Position [m]
X Position [m]
10^-4


Phi Angle Plot
10^-3
Angle [rad]
Time [s]


Theta Angle Plot
10^-3
Angle [rad]
Time [s]


Psi Angle Plot
10^-3
Angle [rad]
Time [s]



In the second case of hover control, there is a small disturbance applied to the system. The amount of disturbance applied is between -0.18 degrees and 0.18 degrees per millisecond. The quadrotor's attitude is therefore changing very quickly but in small amounts. This change in attitude angles affects the positions in X and Y-axis and the quadrotor is slightly drifting. Final drift in both directions is about 1mm at the end of the simulation. Also, the control inputs are changing with a high frequency but with very low magnitudes in order to compensate for this disturbance. The quadrotor is very efficient correcting the attitudes, but since there is no control applied in X and Y-axis, the slight drifting cannot be compensated. The control in altitude is still very efficient and can reach the goal with no noticeable error.

iii) Case 3: Flat start with high disturbance

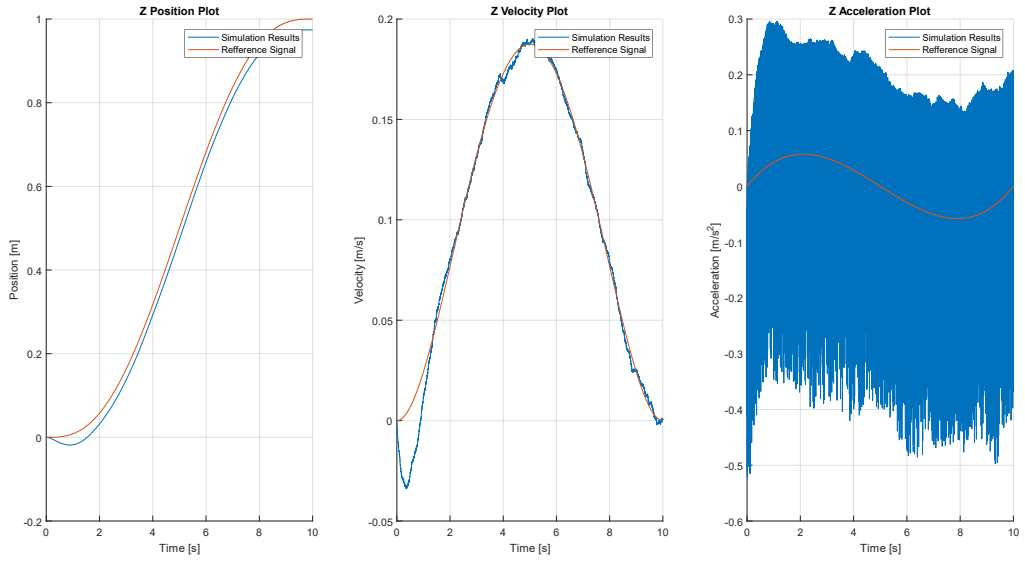
Conditions:

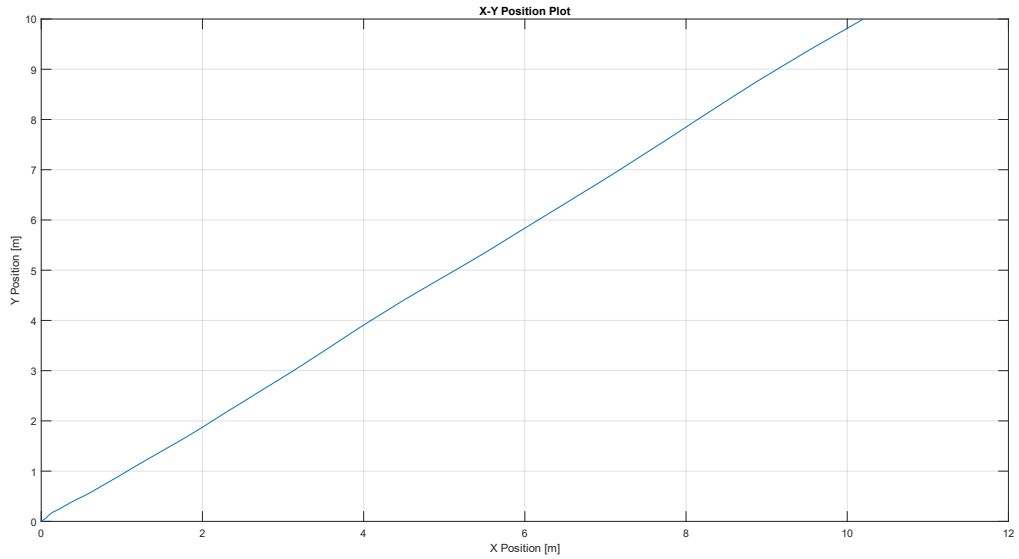
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 8 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 6 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 5 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 2.5 \quad K_{d_\psi} = 4$$

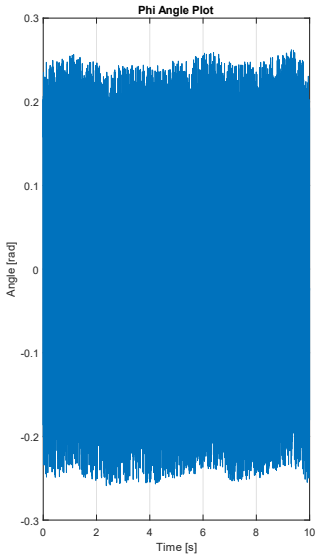
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$

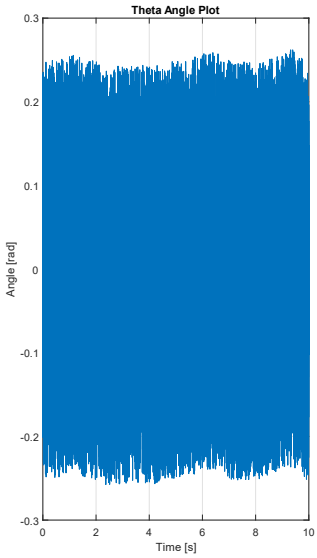
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0 \quad Z_f = 1$$
$$k_\eta = 0.5$$

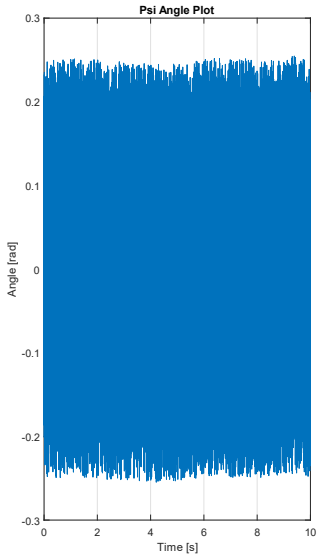
Graphs:

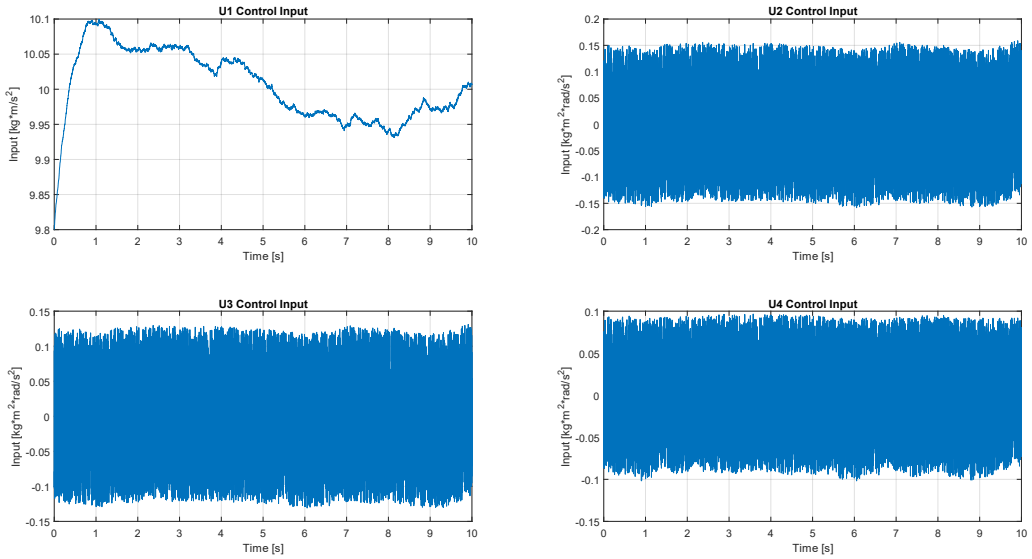



X-Y Position Plot
Y Position [m]
X Position [m]
0
1
2
3
4
5
6
7
8
9
10
0
2
4
6
8
10
12


Phi Angle Plot
Angle [rad]
Time [s]
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
2
4
6
8
10


Theta Angle Plot
Angle [rad]
Time [s]
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
2
4
6
8
10


Psi Angle Plot
Angle [rad]
Time [s]
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
2
4
6
8
10



In the third case of hover control, a very large disturbance is applied to the system. The amount of disturbance applied is between -18 degrees and 18 degrees per millisecond. I applied this extreme amount of disturbance in order to observe the robustness of my system. The quadrotor's attitude was changing very quickly and by small amounts. This change in attitude angles affects the positions in X and Y-axis dramatically and the quadrotor is heavily drifting. Final drift in both directions is around 10m at the end of the simulation. Also, the control inputs are changing with a high frequency and with very high magnitudes in order to compensate for this disturbance. The quadrotor is surprisingly still efficient at correcting the attitudes, but since there is no control applied in X and Y-axis, the drifting cannot be compensated. The control in altitude is not as efficient anymore and has noticeable errors in position and extreme errors in acceleration.

iv) Case 4: Flat start with disturbance and final velocity

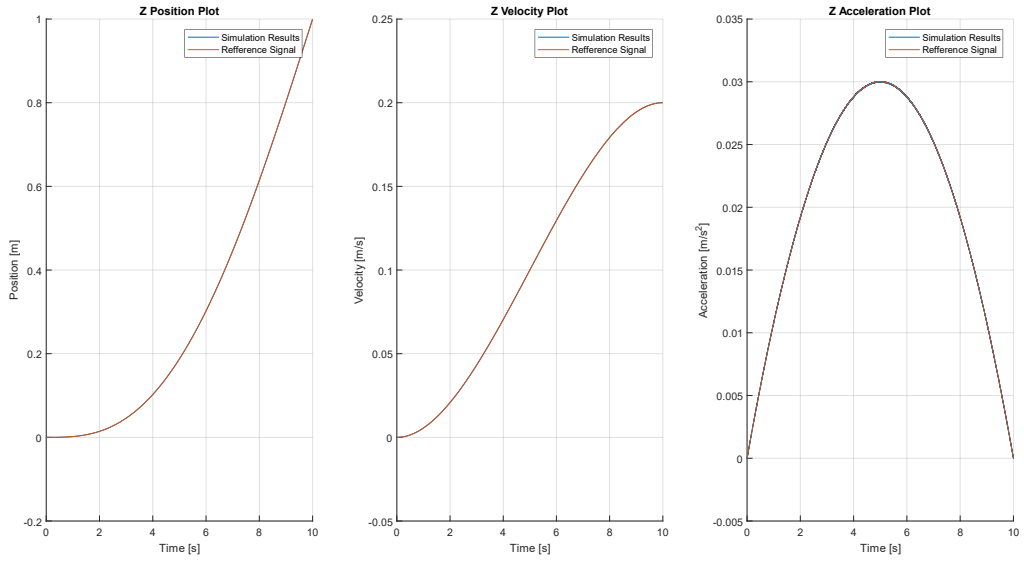
Conditions:

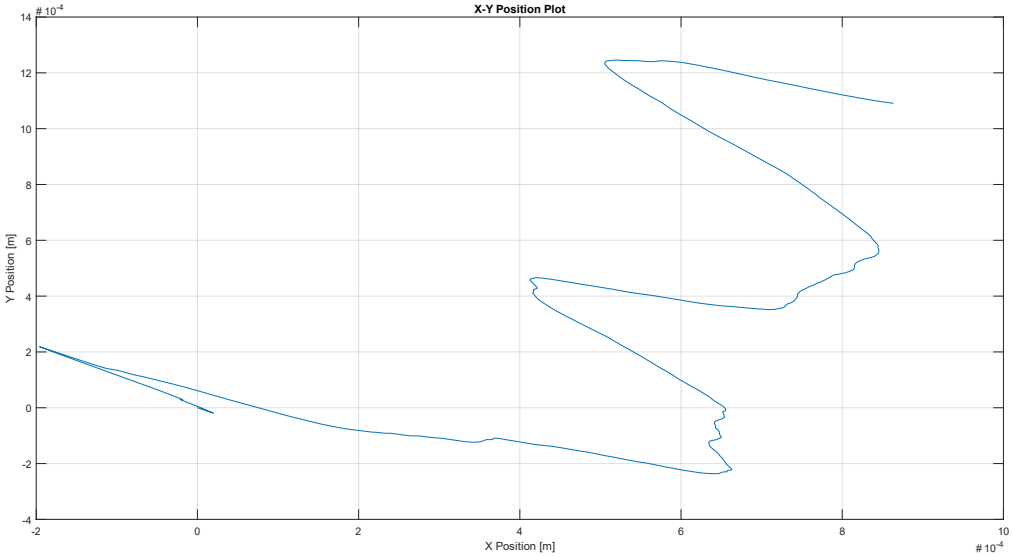
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 8 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 6 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 5 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 2.5 \quad K_{d_\psi} = 4$$

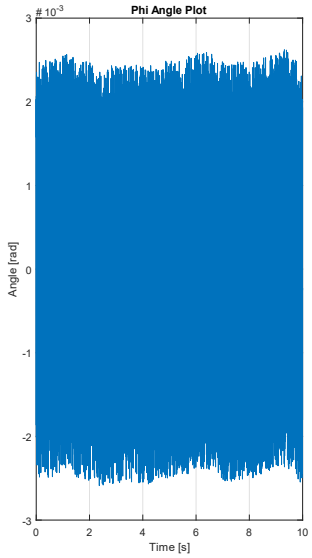
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$

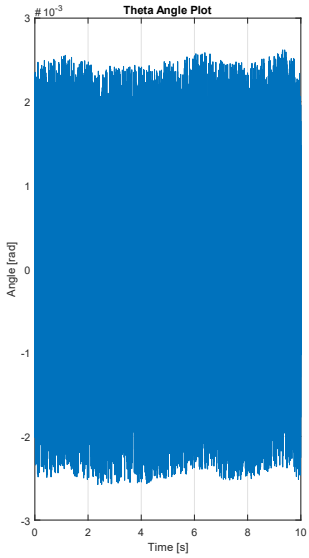
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0.2 \quad Z_f = 1$$
$$k_\eta = 0.005$$

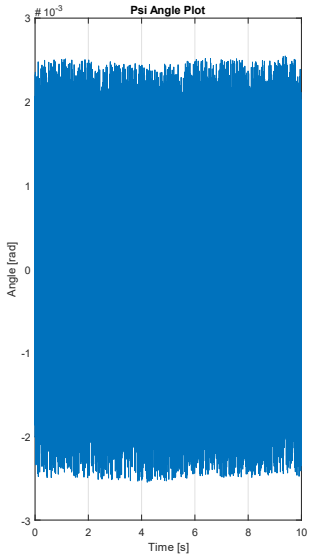
Graphs:

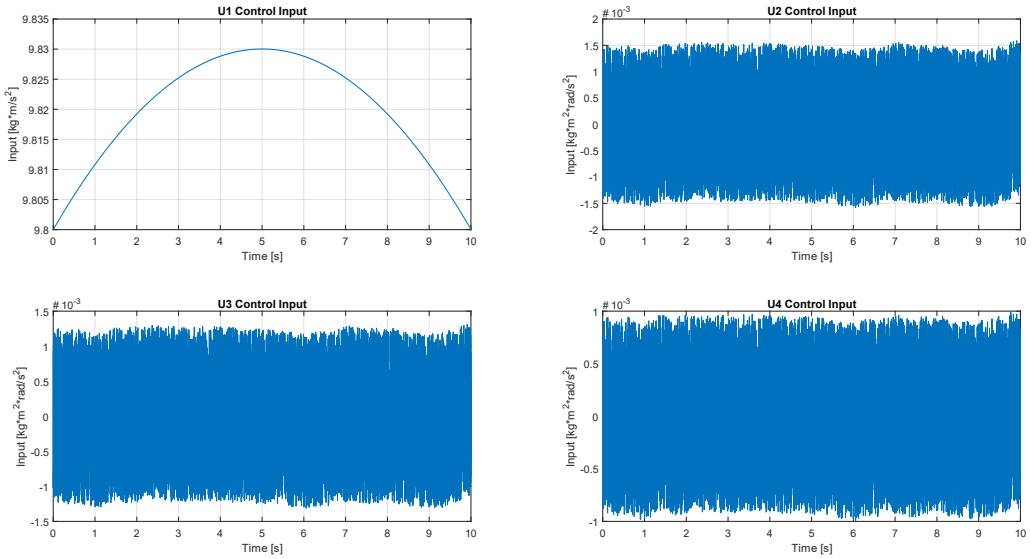



X-Y Position Plot
10^-4
Y Position [m]
X Position [m]
10^-4


Phi Angle Plot
10^-3
Angle [rad]
Time [s]


Theta Angle Plot
10^-3
Angle [rad]
Time [s]


Psi Angle Plot
10^-3
Angle [rad]
Time [s]



In the fourth case of hover control, the goal is changed in order to examine if the quadrotor could reach both a final position and final velocity. Small disturbance is applied again since the quadrotor can control them easily. The amount of disturbance applied is between -0.18 degrees and 0.18 degrees per millisecond. The reference signal for the quadrotor's altitude has changed because of the changed goal and the quadrotor is able to follow the reference signal with no noticeable error. Quadrotor's positions in X and Y-axis are again slightly changing due to disturbance and the quadrotor is slightly drifting. However, the final drift in both directions is very similar to the second case and is about 1mm at the end of the simulation. This shows that the changed reference signal does not affect neither the altitude control efficiency nor the drift in X and Y-axis.

v) Case 5: Tilted start

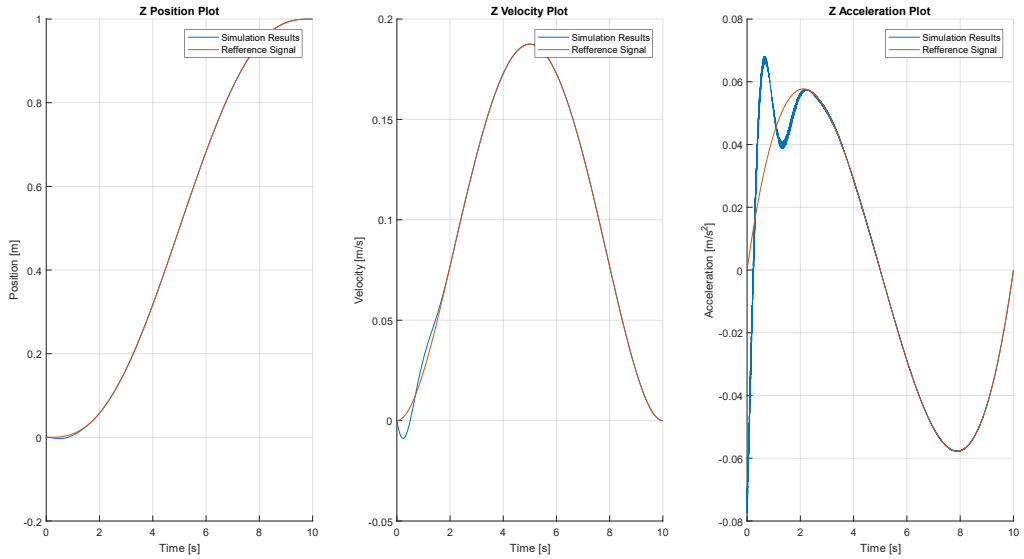
Conditions:

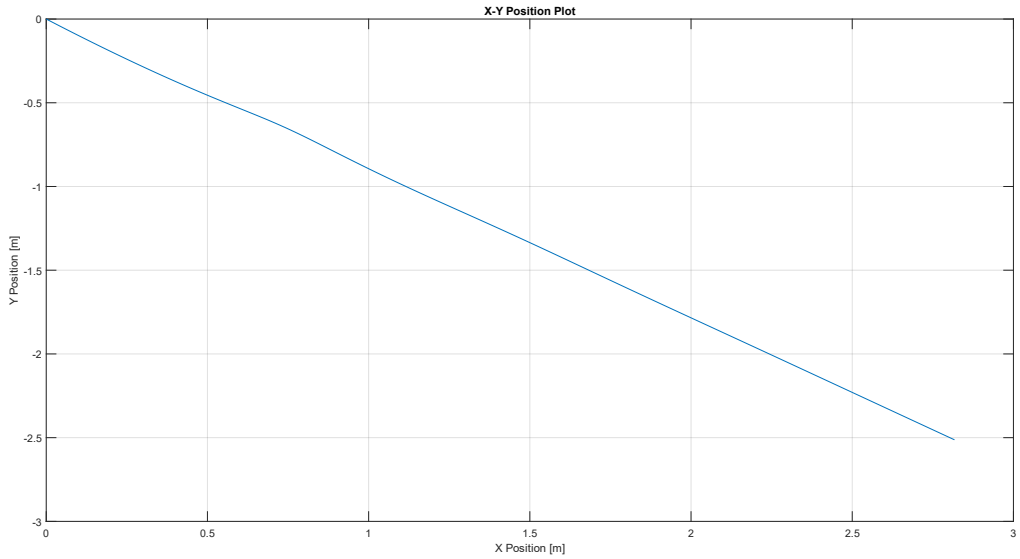
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 8 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 6 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 5 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 2.5 \quad K_{d_\psi} = 4$$

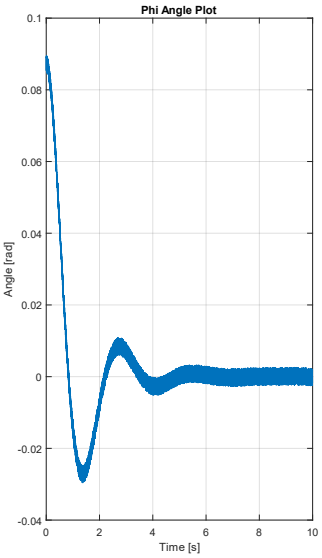
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{36} \quad \theta_0 = \frac{\pi}{36} \quad \psi_0 = 0$$

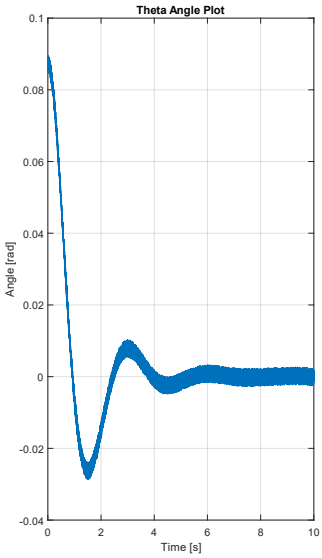
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0 \quad Z_f = 1$$
$$k_\eta = 0.005$$

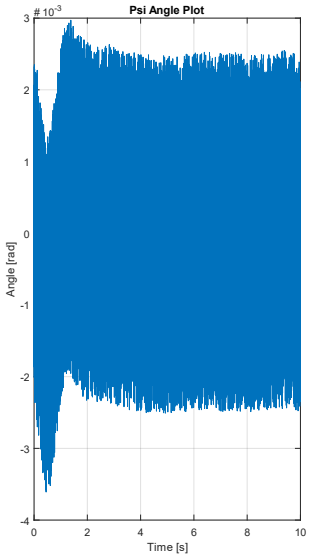
Graphs:

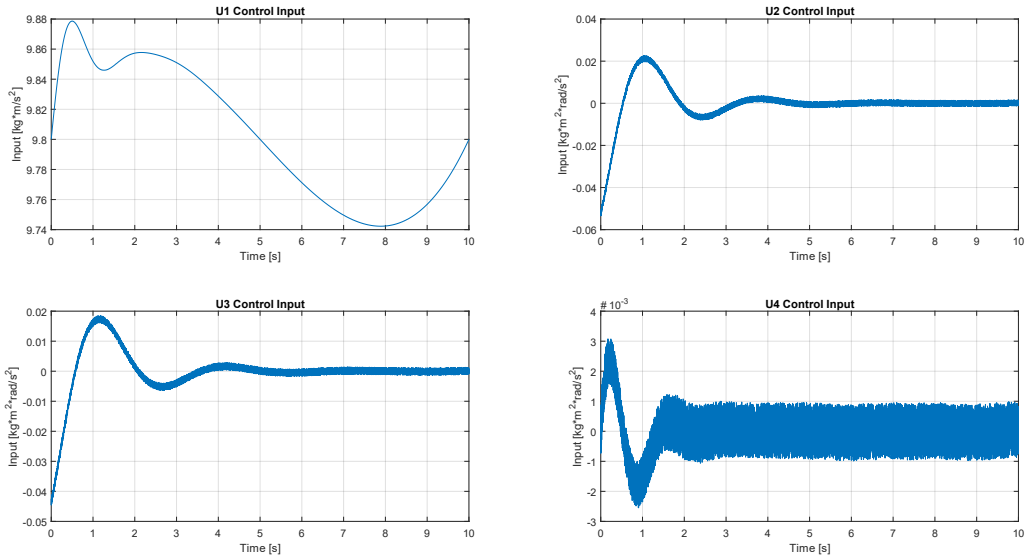



X-Y Position Plot
Y Position [m]
X Position [m]
0
-0.5
-1
-1.5
-2
-2.5
-3
0
0.5
1
1.5
2
2.5
3


Phi Angle Plot
Angle [rad]
Time [s]
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
2
4
6
8
10


Theta Angle Plot
Angle [rad]
Time [s]
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
2
4
6
8
10


Psi Angle Plot
10^-3
Angle [rad]
Time [s]
3
2
1
0
-1
-2
-3
-4
0
2
4
6
8
10



In the fifth case of hover control, there is small disturbance acting on the system and the quadrotor is initially tilted by 5 degrees in ϕ and θ. Therefore, the quadrotor is drifting across the XY plane since there is no control implemented in X and Y-axis. Total drift in the end is close to 3m in both directions. Which is a very noticeable error considering a small initial tilt was given. However, the quadrotor stabilizes its attitude very quickly. Also, there is no final error in attitudes other than slight errors due to disturbance. The quadrotor slightly struggles to control its attitude only in the beginning and starts tracking the reference signal quickly afterwards.

vi) Case 6: Tilted start with higher control gains

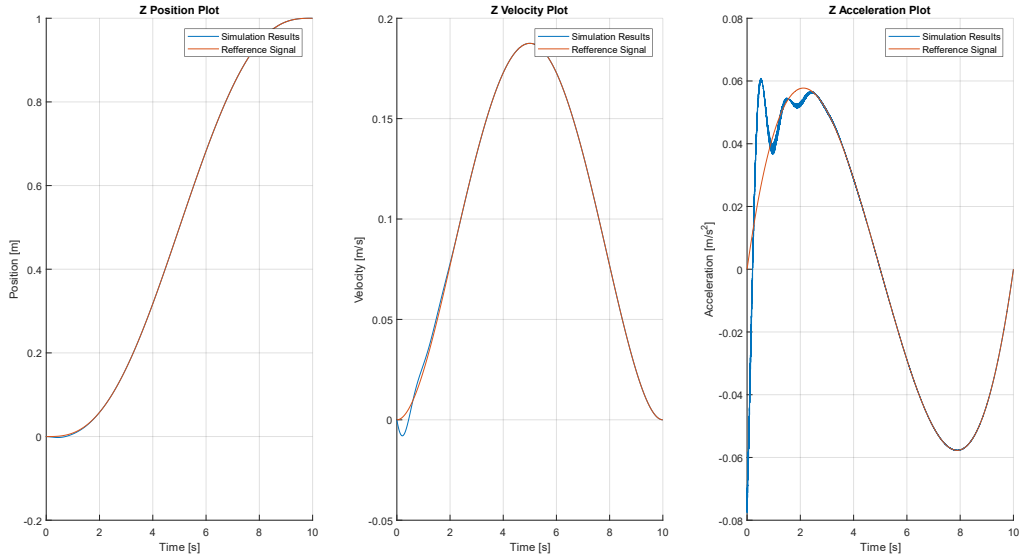
Conditions:

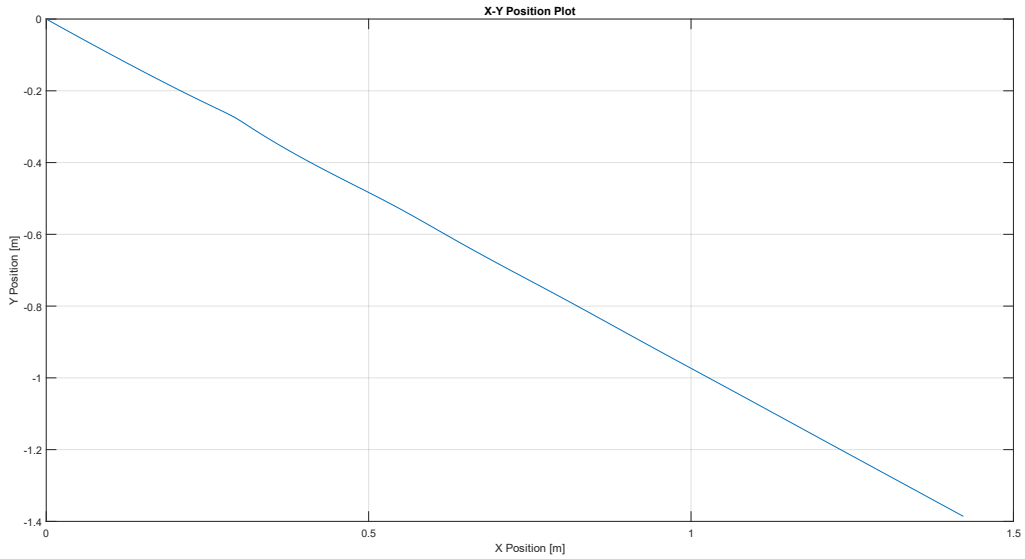
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 13 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 11 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 10 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 7.5 \quad K_{d_\psi} = 4$$

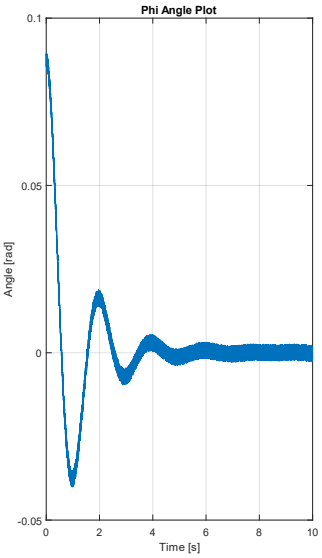
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{36} \quad \theta_0 = \frac{\pi}{36} \quad \psi_0 = 0$$

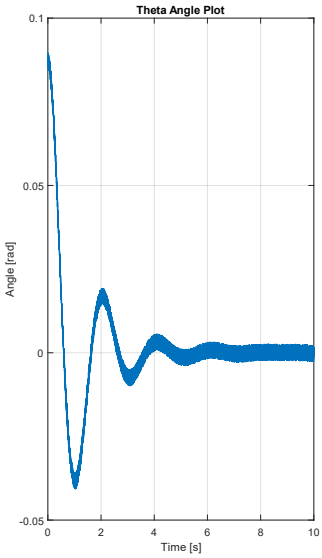
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0 \quad Z_f = 1$$
$$k_\eta = 0.005$$

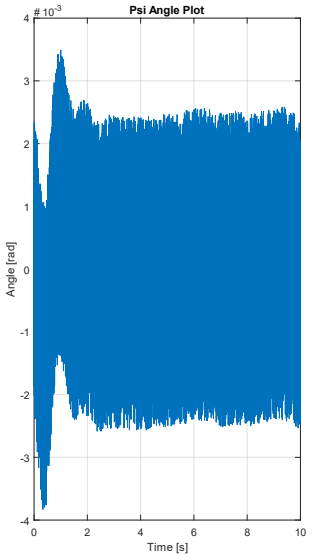
Graphs:

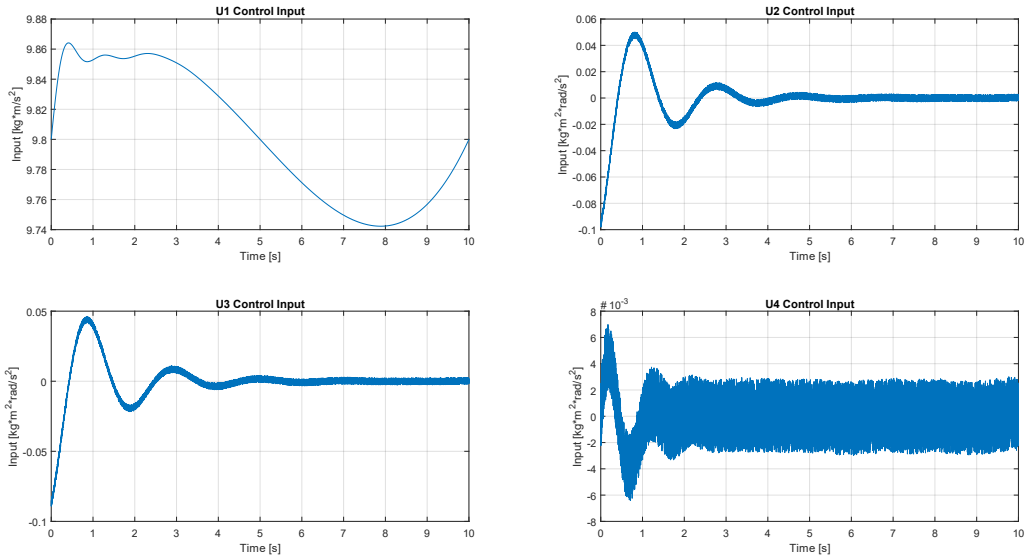



X-Y Position Plot
Y Position [m]
X Position [m]
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
0
0.5
1
1.5


Phi Angle Plot
Angle [rad]
Time [s]
0.1
0.05
0
-0.05
0
2
4
6
8
10


Theta Angle Plot
Angle [rad]
Time [s]
0.1
0.05
0
-0.05
0
2
4
6
8
10


Psi Angle Plot
10^-3
Angle [rad]
Time [s]
4
3
2
1
0
-1
-2
-3
-4
0
2
4
6
8
10



In the sixth case, the same experiment has been done as in the fifth case. The only thing changed was that proportional control gains have been increased in order to observe the effects of control gain changes on the system. With higher proportional control gains, the quadrotor has managed to have half the drift in the XY plane (around 1.5m) and has started following the altitude reference faster and with less error.

vii) Case 7: Very Tilted Start

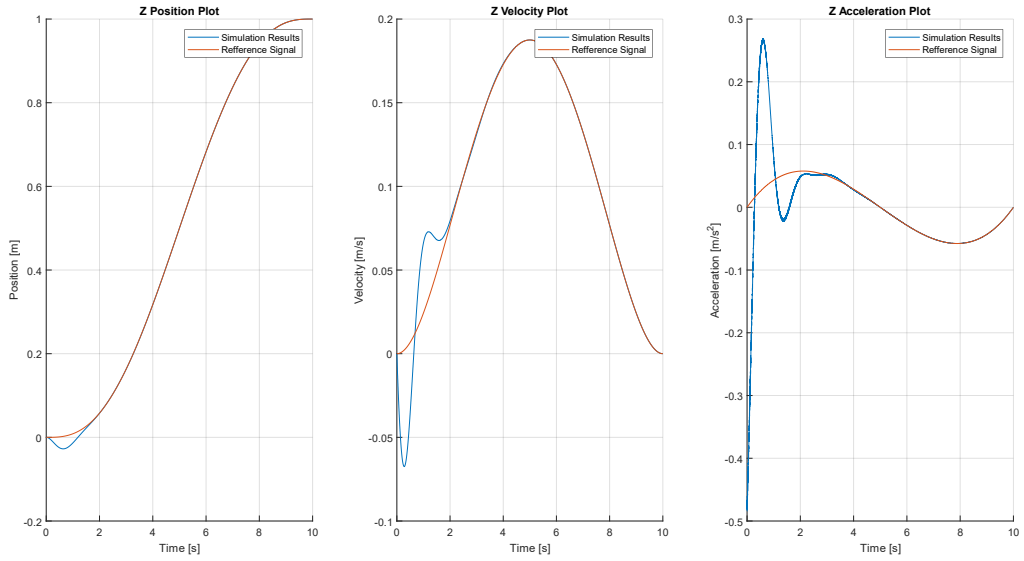
Conditions:

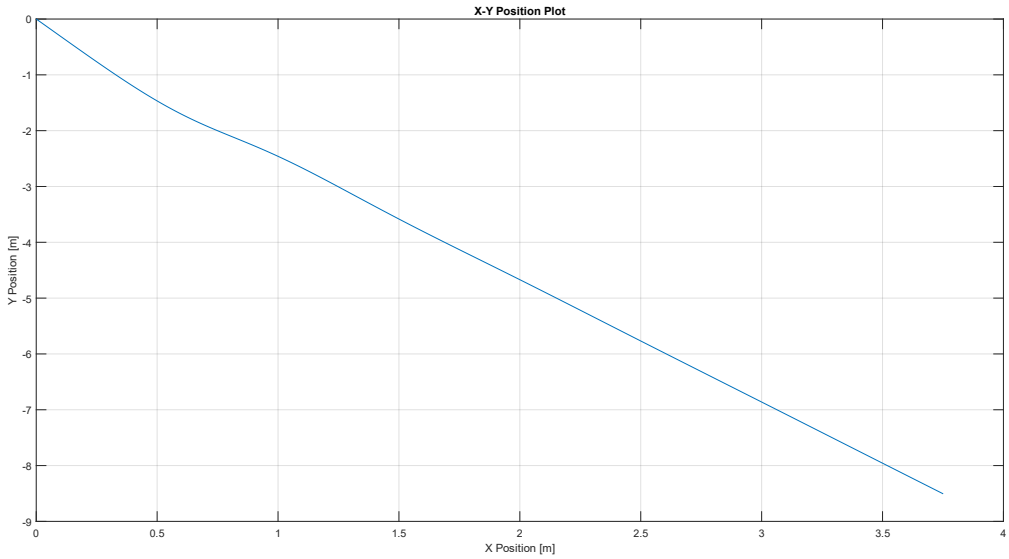
$$K_{p_X} = 0 \quad K_{d_X} = 0$$
$$K_{p_Y} = 0 \quad K_{d_Y} = 0$$
$$K_{p_Z} = 8 \quad K_{d_Z} = 3$$
$$K_{p_\phi} = 6 \quad K_{d_\phi} = 1.7$$
$$K_{p_\theta} = 5 \quad K_{d_\theta} = 1.6$$
$$K_{p_\psi} = 2.5 \quad K_{d_\psi} = 4$$

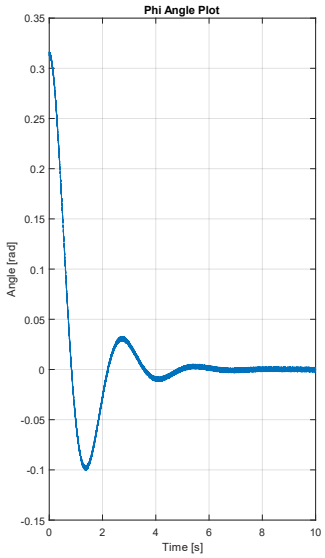
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{10} \quad \theta_0 = 0 \quad \psi_0 = \frac{\pi}{10}$$

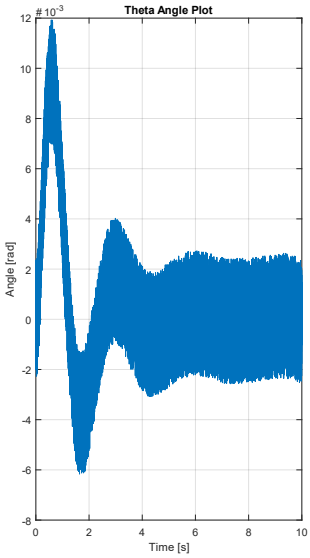
$$\ddot{X}_f = 0 \quad \dot{X}_f = 0 \quad X_f = 0$$
$$\ddot{Y}_f = 0 \quad \dot{Y}_f = 0 \quad Y_f = 0$$
$$Z_f = 0 \quad Z_f = 0 \quad Z_f = 1$$
$$k_\eta = 0.005$$

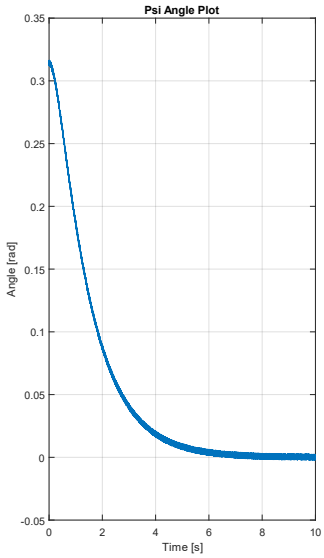
Graphs:

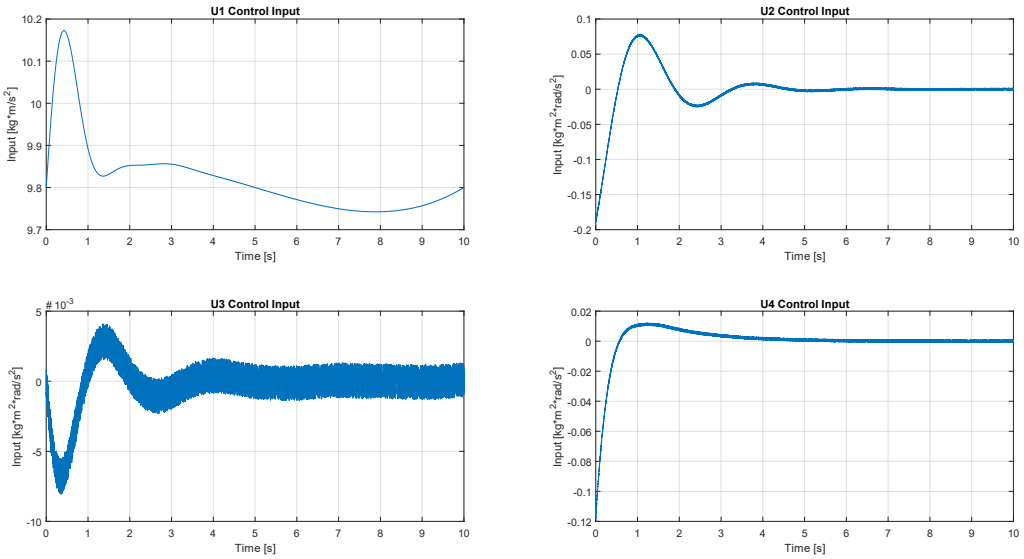



X-Y Position Plot
Y Position [m]
X Position [m]


Phi Angle Plot
Angle [rad]
Time [s]


Theta Angle Plot
#10^-3
Angle [rad]
Time [s]


Psi Angle Plot
Angle [rad]
Time [s]



In the sixth case, a very similar experiment to the one in the fifth case has been done. The quadrotor is initially tilted again but with more tilt and instead of tilt in ϕ and θ the initial tilt is in in ϕ and ψ. The results are as expected, there is more drifting in the XY plane, more error in altitude tracking and it takes longer to take the angles to their reference values.

b) TRAJECTORY TRACKING CONTROL

For each case in trajectory tracking control, 7 different data sets have been plotted in order to examine the performance and stability of the system. First three plots contain the position, velocity and acceleration in X, Y and Z-axis simulation results vs reference signal. Fourth plot is the position in X-axis vs position in Y-axis. Fifth plot shows the attitude (roll, pitch yaw angles) vs time and the sixth plot contains the control inputs vs time graphs. Seventh plot shows the trajectory the quadrotor has taken in a 3D plot.

Also, three goal points have been used in trajectory tracking control instead of in order to be able to create more complex trajectories such as circles.

i) Case 1: Hover, flat start

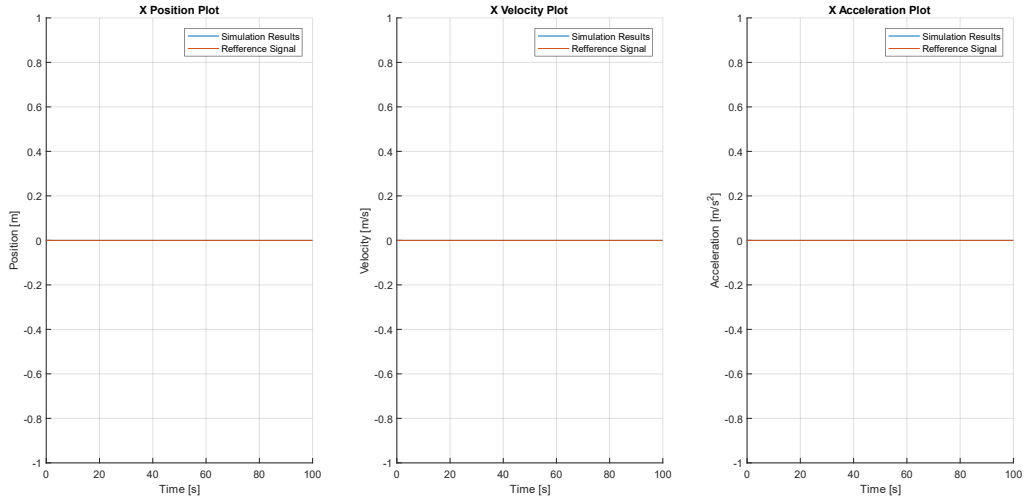
Conditions:

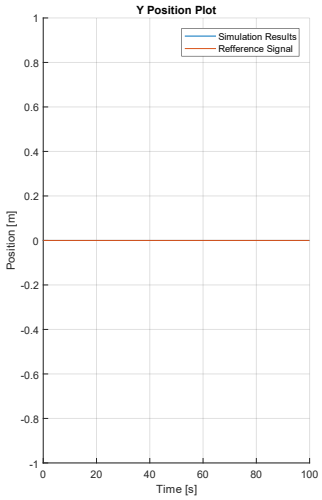
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

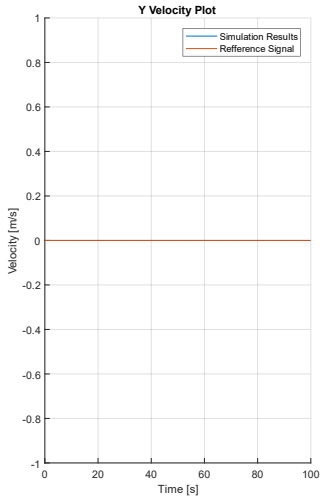
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$
$$k_\eta = 0$$

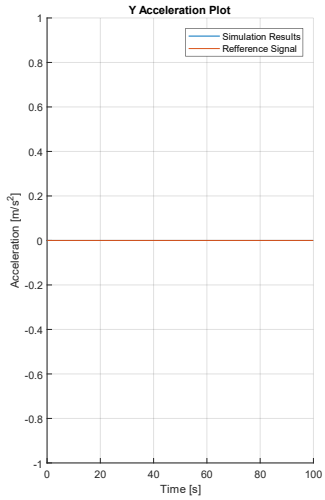
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = 0 \quad X_{f(1)} = 0$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0 \quad Y_{f(1)} = 0$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0 \quad Z_{f(1)} = 5$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0 \quad X_{f(2)} = 0$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = 0 \quad Y_{f(2)} = 0$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 15$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = 0 \quad X_{f(3)} = 0$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0 \quad Y_{f(3)} = 0$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 0$$

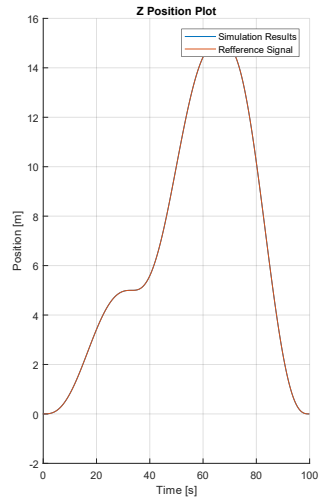
Graphs:

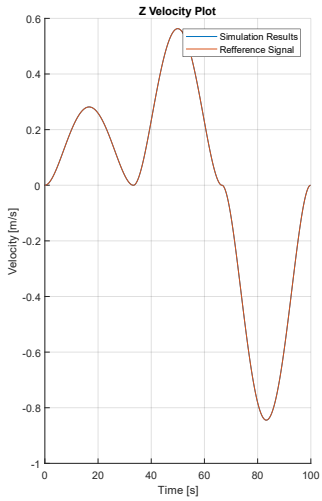


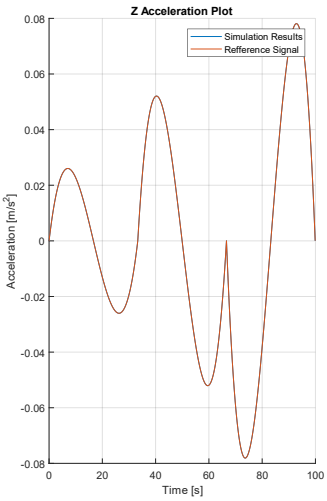

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

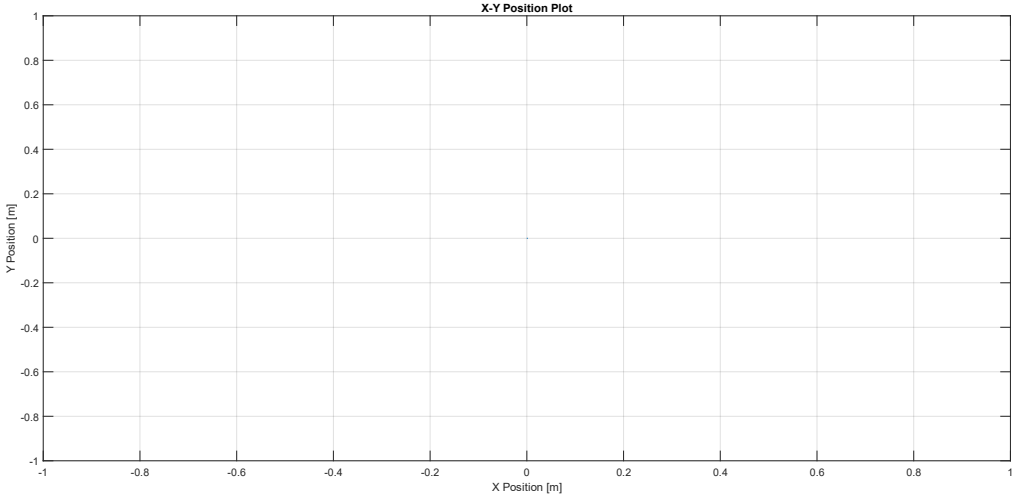

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

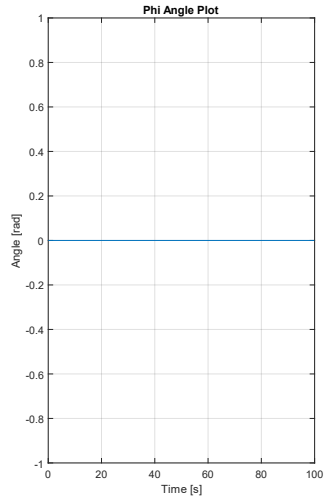

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s²]
Time [s]

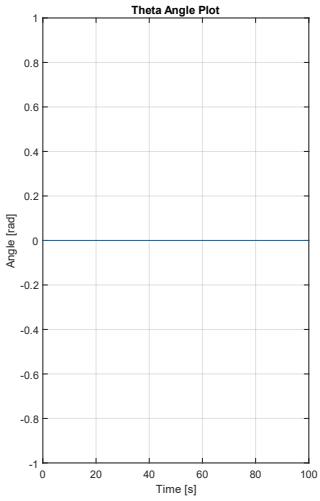

Z Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

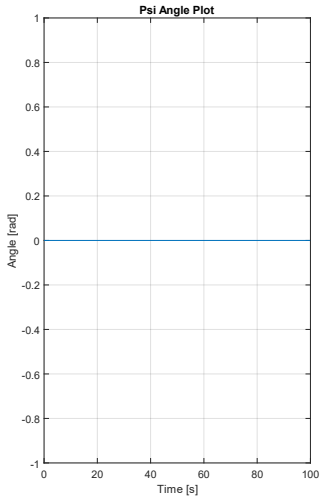

Z Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

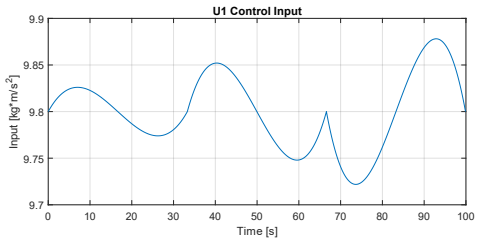

Z Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s²]
Time [s]

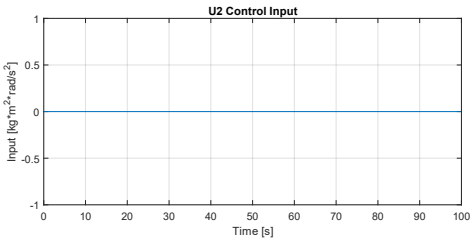

X-Y Position Plot
Y Position [m]
X Position [m]

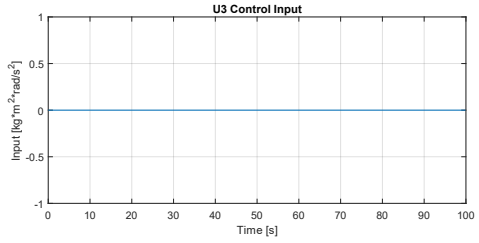

Phi Angle Plot
Angle [rad]
Time [s]

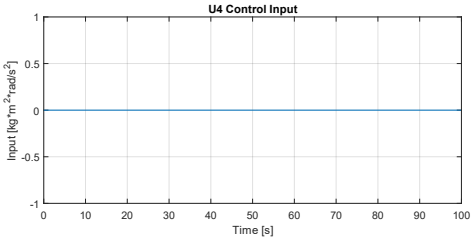

Theta Angle Plot
Angle [rad]
Time [s]

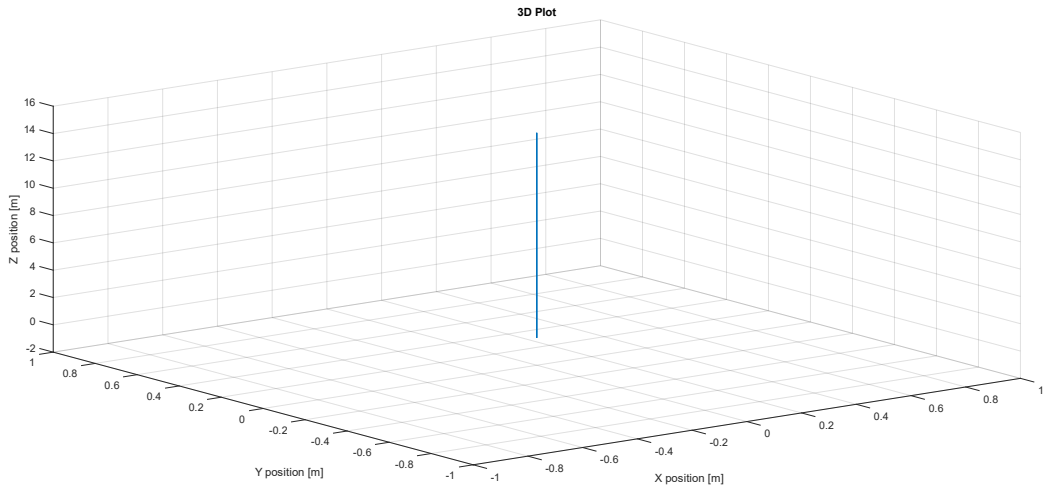

Psi Angle Plot
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


3D Plot
Z position [m]
Y position [m]
X position [m]

In the first case of trajectory tracking control, I wanted to see whether this new controller would be able to perform just hovering. In this case the quadrotor is initially flay and no disturbance is applied. And the quadrotor is set 3 different altitudes to reach in order, 5m, 15m and 0m. The quadrotor was able to follow the reference signal in Z-axis with no noticeable error. Also, it had no changes in XY plane or attitude which was expected as there is no disturbance acting on the system. The quadrotor seemed robust and performed hovering very efficiently.

ii) Case 2: Hover, flat start with disturbance

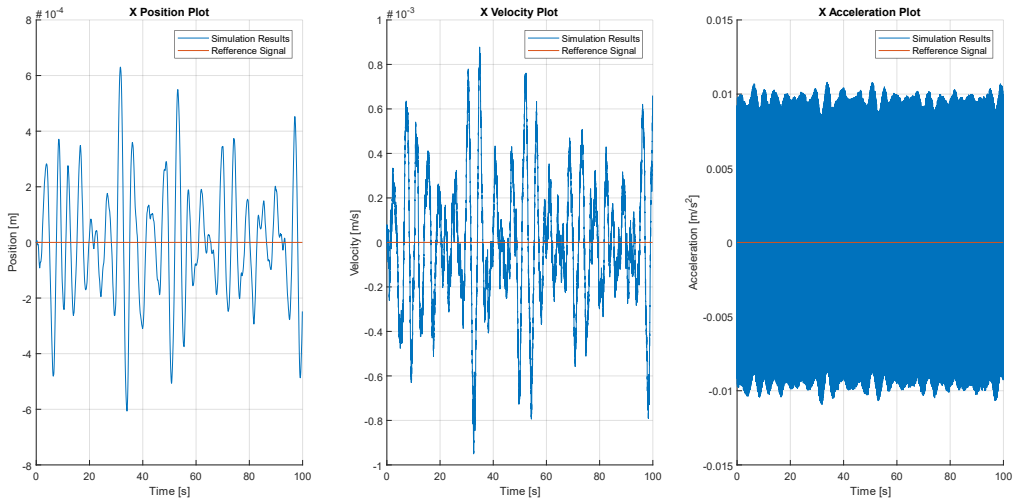
Conditions:

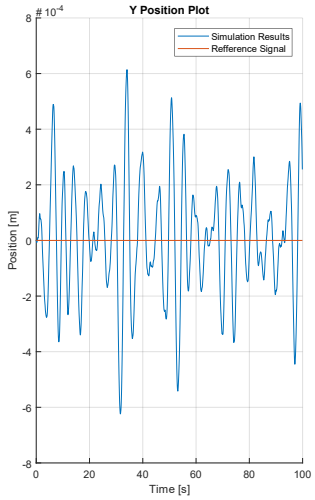
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

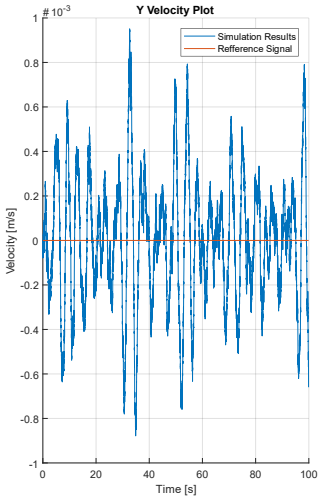
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$
$$k_\eta = 0.002$$

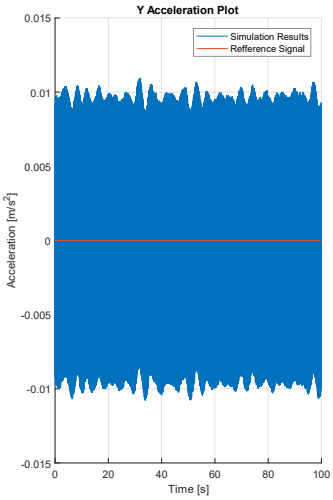
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = 0 \quad X_{f(1)} = 0$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0 \quad Y_{f(1)} = 0$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0 \quad Z_{f(1)} = 5$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0 \quad X_{f(2)} = 0$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = 0 \quad Y_{f(2)} = 0$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 15$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = 0 \quad X_{f(3)} = 0$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0 \quad Y_{f(3)} = 0$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 0$$

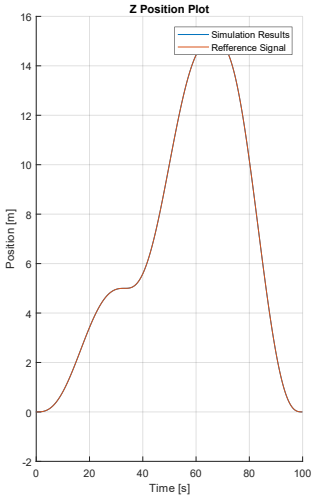
Graphs:

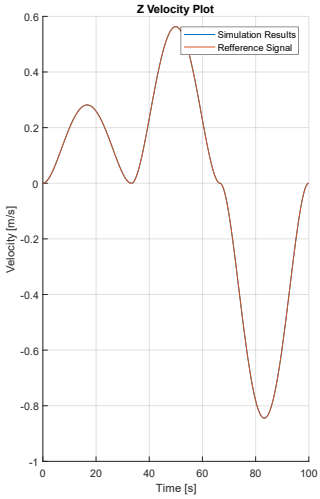


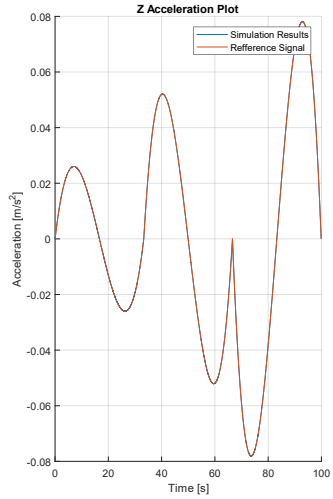

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

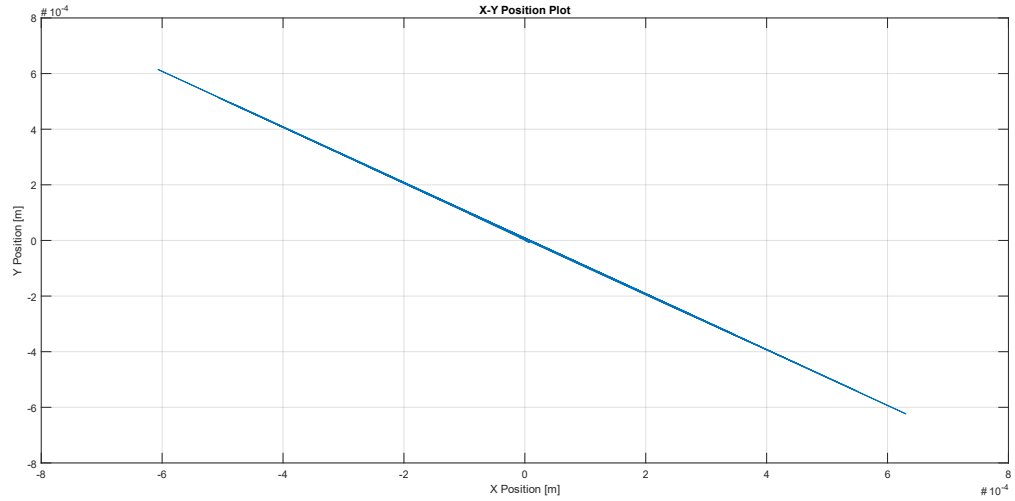

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

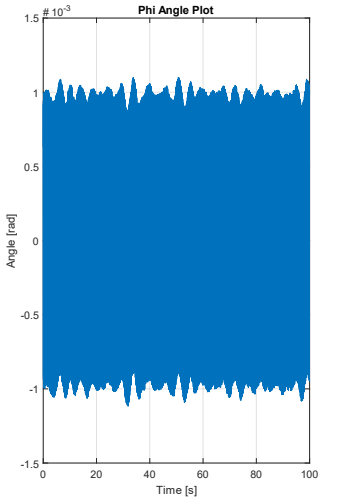

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

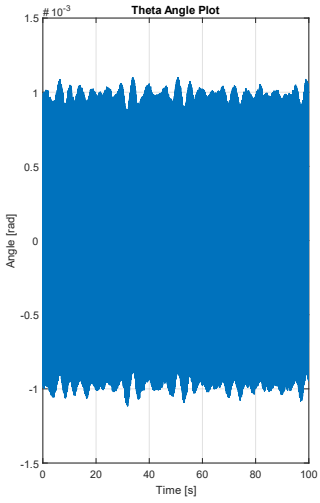

Z Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

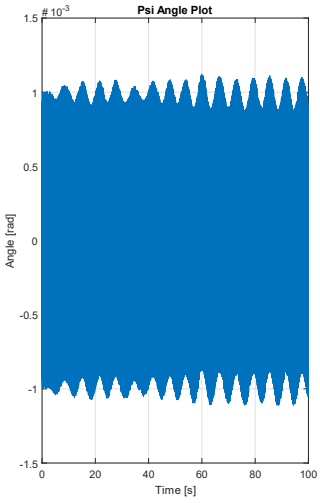

Z Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

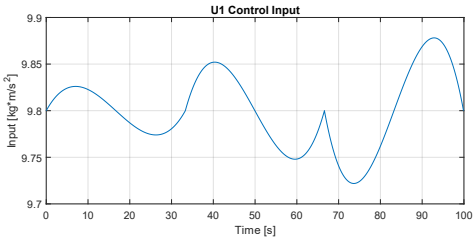

Z Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

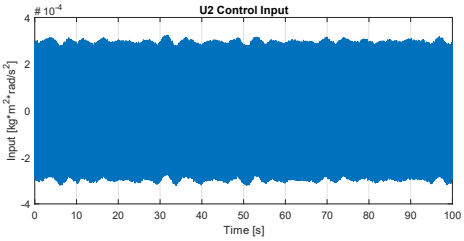

X-Y Position Plot
Y Position [m]
X Position [m]
10^-4

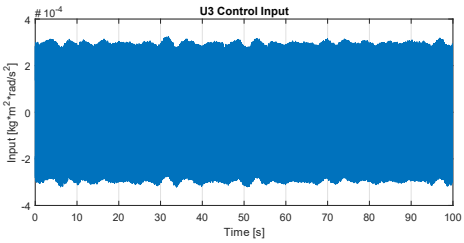

Phi Angle Plot
Angle [rad]
Time [s]
10^-3

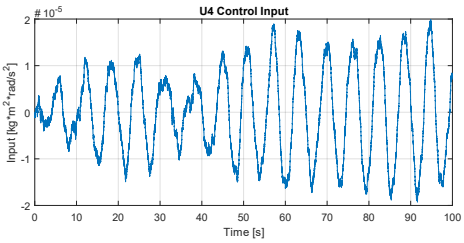

Theta Angle Plot
Angle [rad]
Time [s]
10^-3

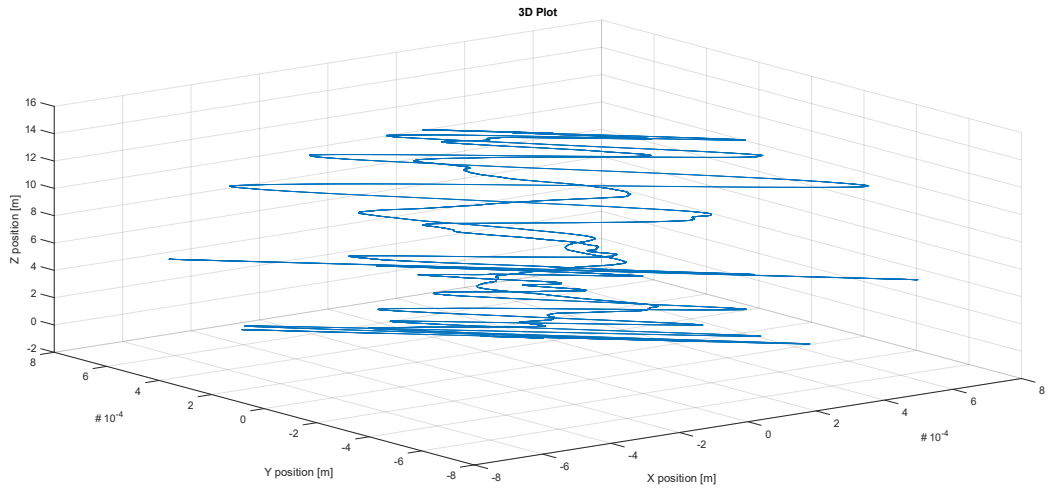

Psi Angle Plot
Angle [rad]
Time [s]
10^-3


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]
10^-4


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]
10^-4


U4 Control Input
Input [kg*m^2*rad/s^2]
Time [s]
10^-5



In the second case, the same test as in the first one was repeated with added disturbance. The system was able to follow the altitude reference with no problems, and the attitude was corrected as the disturbance was applied. The quadrotor managed to keep the drift in XY plane close to 0 and it reached at most 0.6mm. Even though the 3D plot looks like the quadrotor was drifting a lot, it should be seen that the values in X and Y-axis are in the order of $10^{-4}m$. Therefore, it can be understood that the quadrotor was very robust against disturbance, too.

iii) Case 3: Hover, tilted start

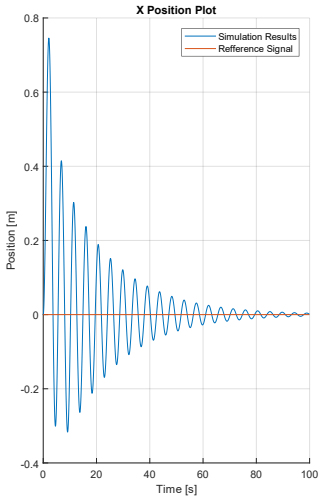
Conditions:

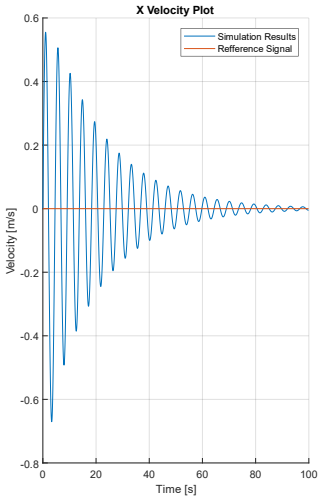
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

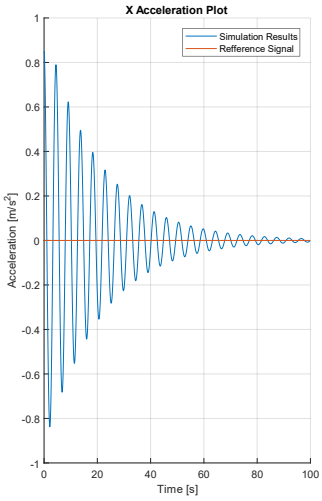
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{36} \quad \theta_0 = \frac{\pi}{36} \quad \psi_0 = 0$$
$$k_\eta = 0$$

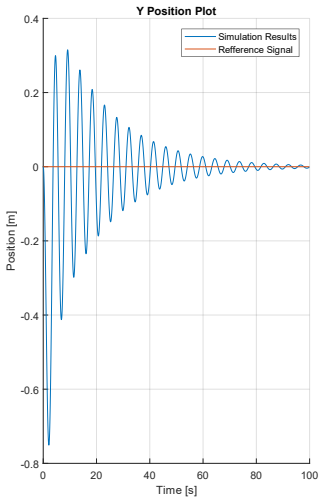
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = 0 \quad X_{f(1)} = 0$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0 \quad Y_{f(1)} = 0$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0 \quad Z_{f(1)} = 5$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0 \quad X_{f(2)} = 0$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = 0 \quad Y_{f(2)} = 0$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 15$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = 0 \quad X_{f(3)} = 0$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0 \quad Y_{f(3)} = 0$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 0$$

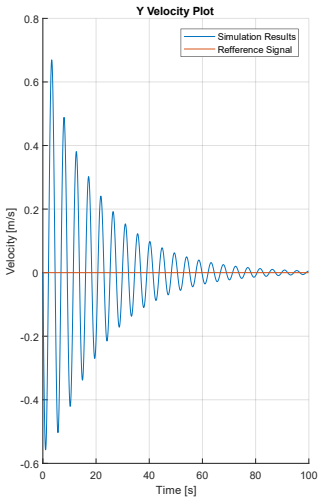
Graphs:

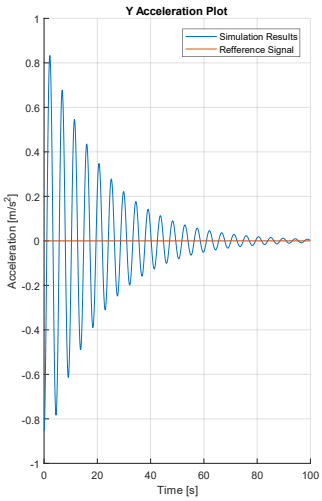

X Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

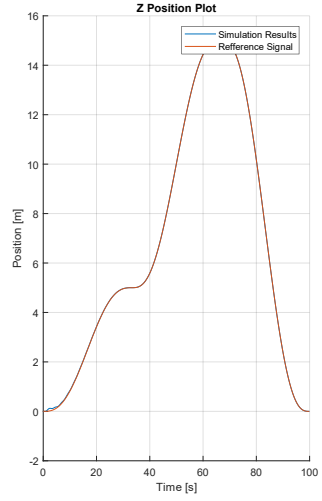

X Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

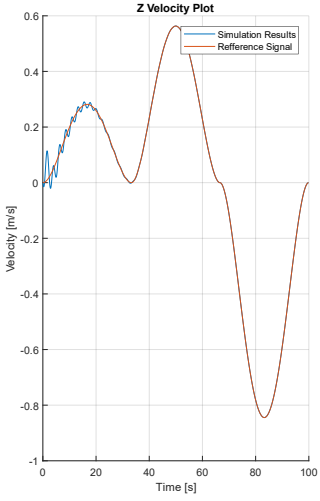

X Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

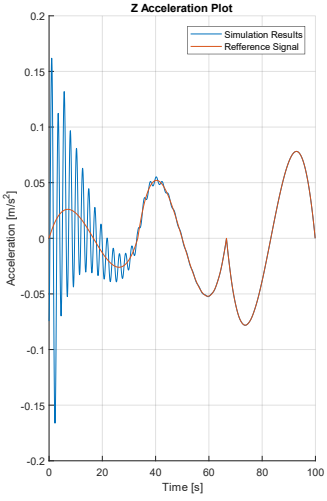

Y Position Plot
Simulation Results
Refference Signal
Postion [m]
Time [s]

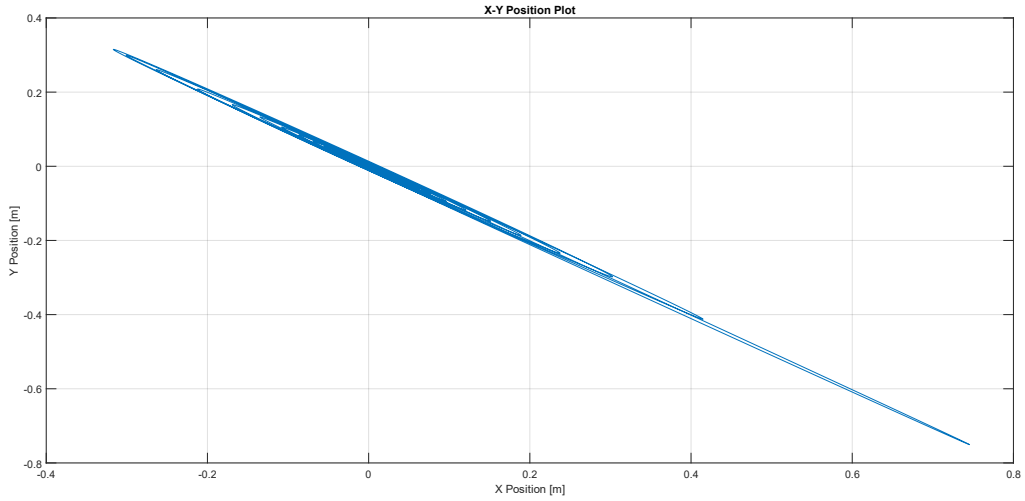

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

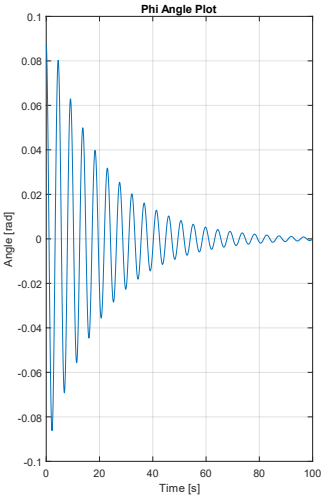

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

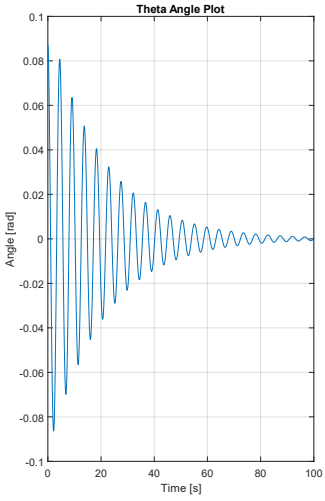

Z Position Plot
Simulation Results
Reference Signal
Position [m]
Time [s]

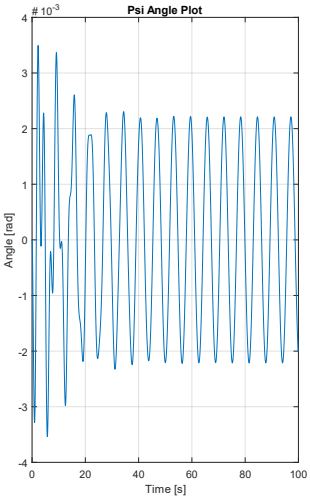

Z Velocity Plot
Simulation Results
Reference Signal
Velocity [m/s]
Time [s]

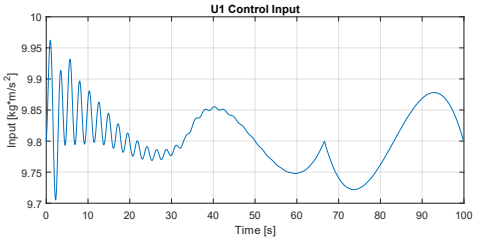

Z Acceleration Plot
Simulation Results
Reference Signal
Acceleration [m/s^2]
Time [s]

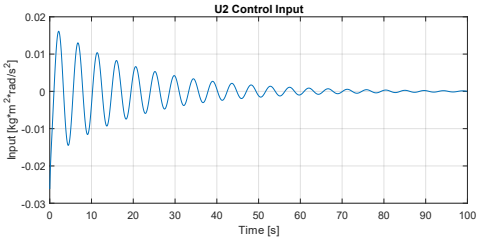

X-Y Position Plot
Y Position [m]
X Position [m]

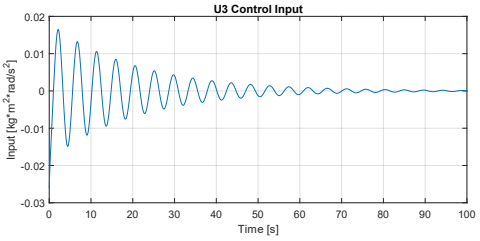

Phi Angle Plot
Angle [rad]
Time [s]

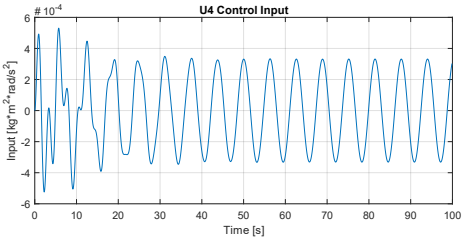

Theta Angle Plot
Angle [rad]
Time [s]

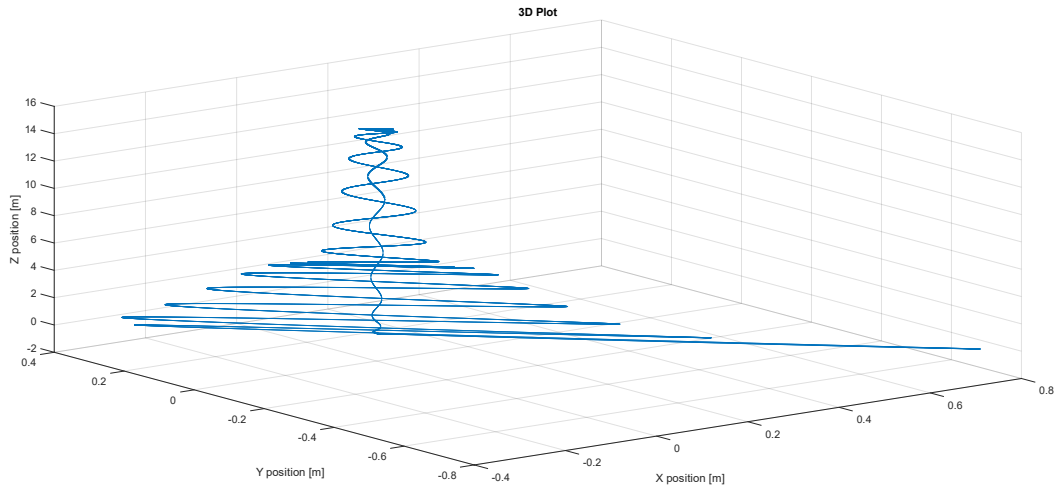

Psi Angle Plot
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
Input [kg*m^2*rad/s^2]
Time [s]



In the third case, the experiment in case 1 was repeated with given initial tilt. The quadrotor was tilted in ϕ and θ by 5 degrees each. The quadrotor was very successful in following the reference signal for altitude other than the beginning. However, there were a lot of oscillations attitude and the X and Y-axis positions. In the end the quadrotor managed to get all these values close to 0 but it took a long time. This showed that the quadrotor is not very robust against initially tilted starts.

iv) Case 4: Trajectory, flat start

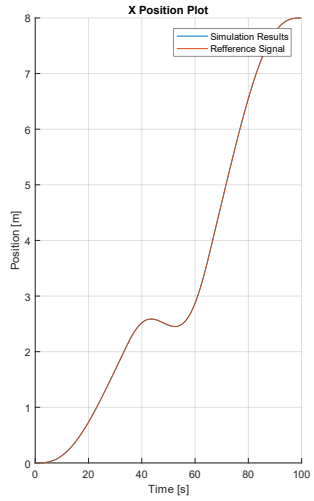
Conditions:

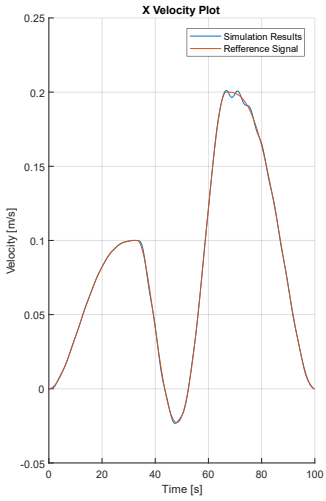
$$\begin{matrix} K_{p_X} = 0.9 & K_{d_X} = 2 \\ K_{p_Y} = 0.9 & K_{d_Y} = 2 \\ K_{p_Z} = 1 & K_{d_Z} = 1 \\ K_{p_\phi} = 3 & K_{d_\phi} = 5 \\ K_{p_\theta} = 3 & K_{d_\theta} = 5 \\ K_{p_\psi} = 1 & K_{d_\psi} = 1 \end{matrix}$$

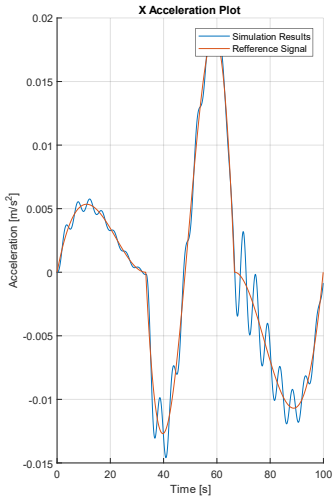
$$\begin{matrix} \ddot{X}_0 = 0 & \dot{X}_0 = 0 & X_0 = 0 \\ \ddot{Y}_0 = 0 & \dot{Y}_0 = 0 & Y_0 = 0 \\ Z_0 = 0 & Z_0 = 0 & Z_0 = 0 \\ p_0 = 0 & q_0 = 0 & r_0 = 0 \\ \phi_0 = 0 & \theta_0 = 0 & \psi_0 = 0 \\ & k_\eta = 0 & \end{matrix}$$

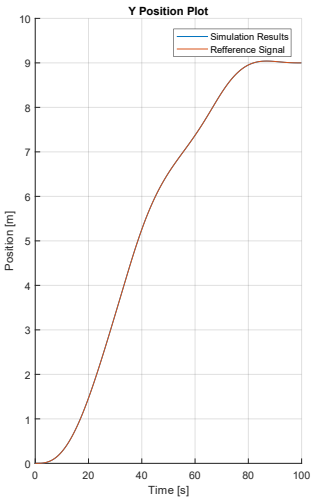
$$\begin{matrix} \ddot{X}_{f(1)} = 0 & \dot{X}_{f(1)} = 0.1 & X_{f(1)} = 2 \\ \ddot{Y}_{f(1)} = 0 & \dot{Y}_{f(1)} = 0.2 & Y_{f(1)} = 4 \\ Z_{f(1)} = 0 & Z_{f(1)} = 0.1 & Z_{f(1)} = 4 \\ \ddot{X}_{f(2)} = 0 & \dot{X}_{f(2)} = 0.2 & X_{f(2)} = 4 \\ \ddot{Y}_{f(2)} = 0 & \dot{Y}_{f(2)} = 0.1 & Y_{f(2)} = 8 \\ Z_{f(2)} = 0 & Z_{f(2)} = 0 & Z_{f(2)} = 5 \\ \ddot{X}_{f(3)} = 0 & \dot{X}_{f(3)} = 0 & X_{f(3)} = 8 \\ \ddot{Y}_{f(3)} = 0 & \dot{Y}_{f(3)} = 0 & Y_{f(3)} = 9 \\ Z_{f(3)} = 0 & Z_{f(3)} = 0 & Z_{f(3)} = 0 \end{matrix}$$

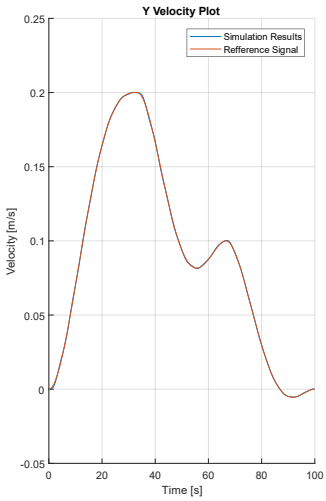
Graphs:

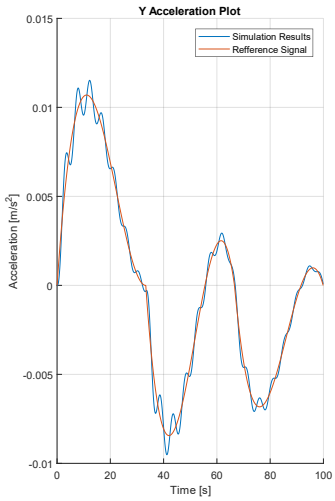

X Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

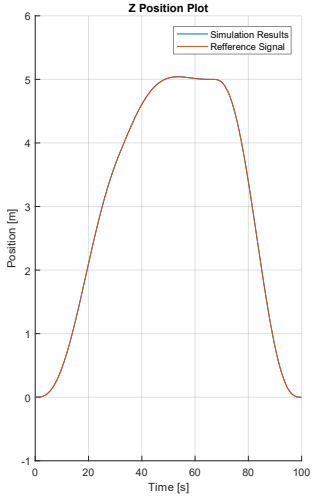

X Velocity Plot
Simulation Results
Reference Signal
Velocity [m/s]
Time [s]

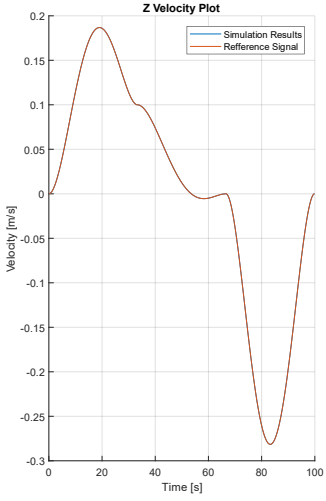

X Acceleration Plot
Simulation Results
Reference Signal
Acceleration [m/s^2]
Time [s]

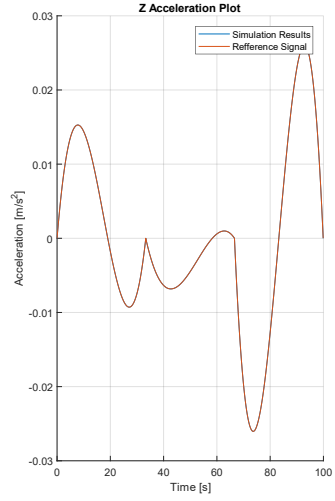

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

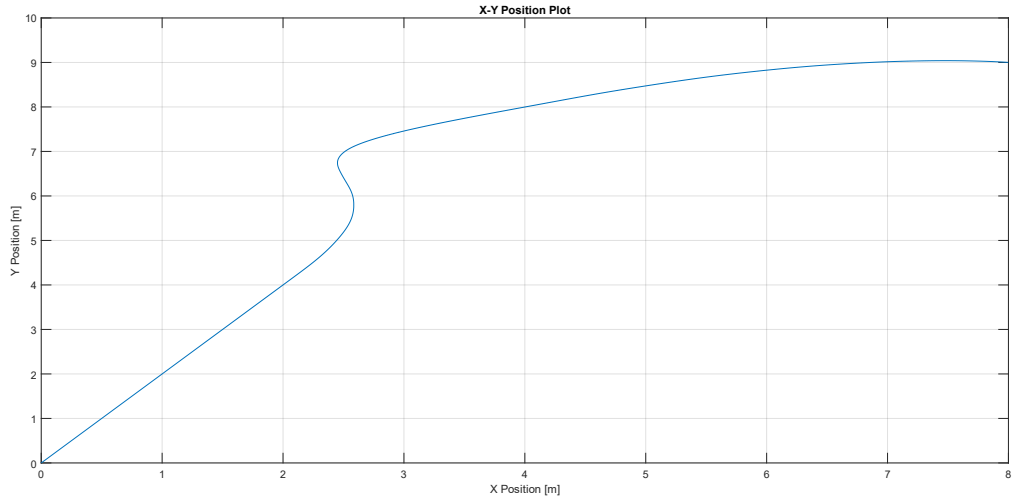

Y Velocity Plot
Simulation Results
Reference Signal
Velocity [m/s]
Time [s]

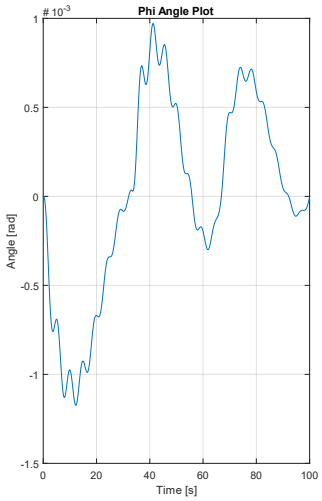

Y Acceleration Plot
Simulation Results
Reference Signal
Acceleration [m/s^2]
Time [s]

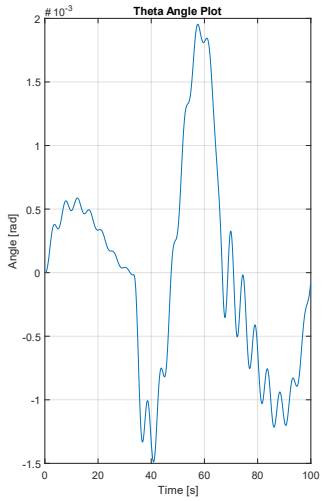

Z Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

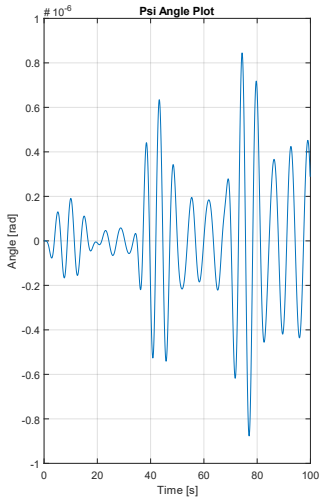

Z Velocity Plot
Simulation Results
Reference Signal
Velocity [m/s]
Time [s]

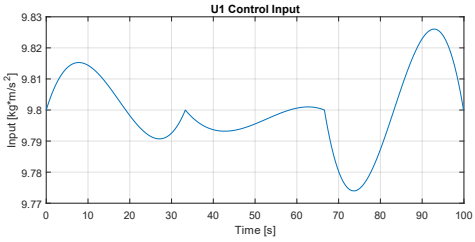

Z Acceleration Plot
Simulation Results
Reference Signal
Acceleration [m/s^2]
Time [s]

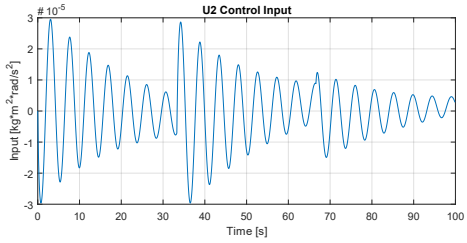

X-Y Position Plot
Y Position [m]
X Position [m]

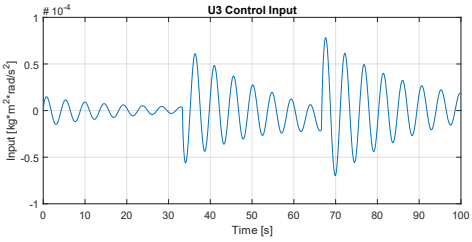

Phi Angle Plot
10^-3
Angle [rad]
Time [s]

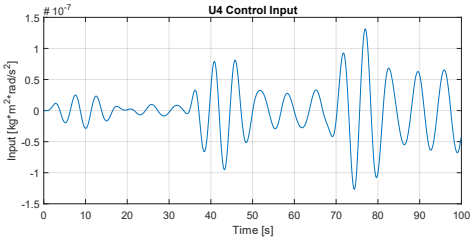

Theta Angle Plot
10^-3
Angle [rad]
Time [s]

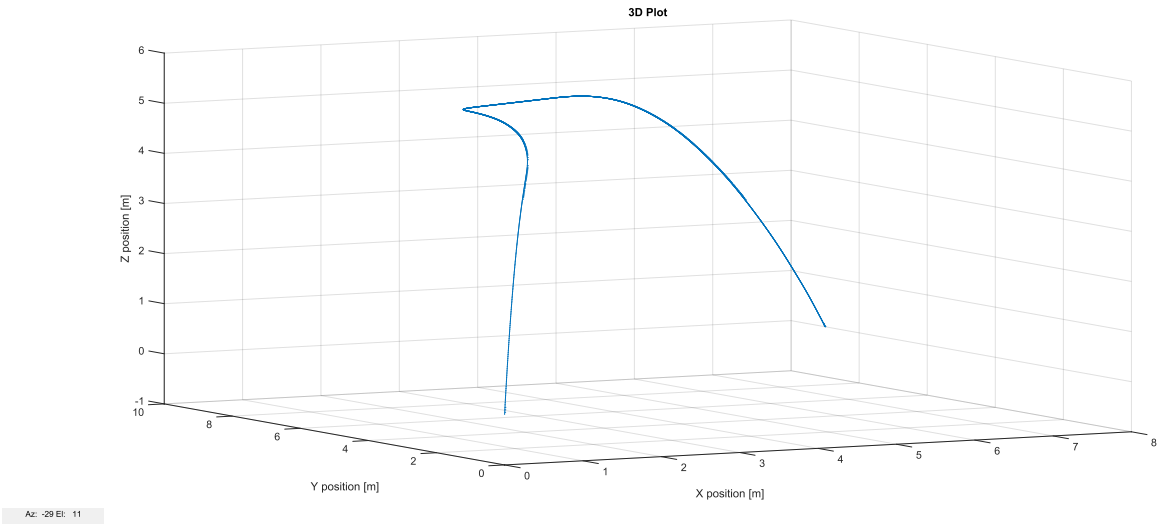

Psi Angle Plot
10^-6
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
10^-5
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
10^-4
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
10^-7
Input [kg*m^2*rad/s^2]
Time [s]



In the fourth case, I have decided to try to give a smooth trajectory which has components in all three directions to the quadrotor. I decided to observe the results under no initial tilt and no disturbance. The quadrotor was able to follow the reference signal in all three directions with no noticeable error. The positions and attitude were changing smoothly given the smooth trajectory. The quadrotor was very successful at following a 3D trajectory.

v) Case 5: Trajectory, tilted start

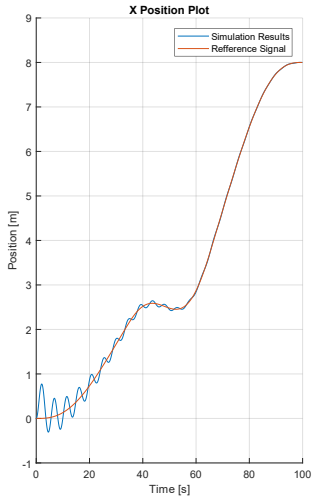
Conditions:

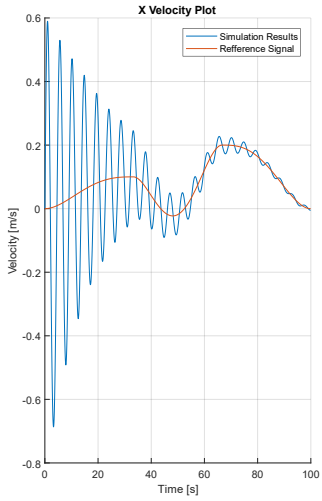
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

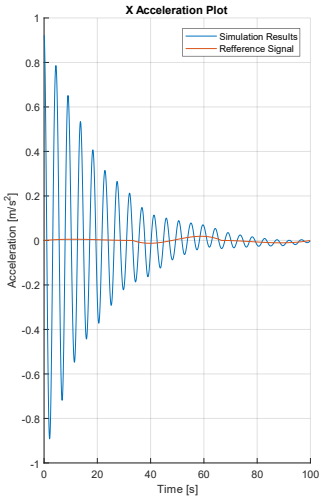
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{36} \quad \theta_0 = \frac{\pi}{36} \quad \psi_0 = \frac{\pi}{36}$$
$$k_\eta = 0$$

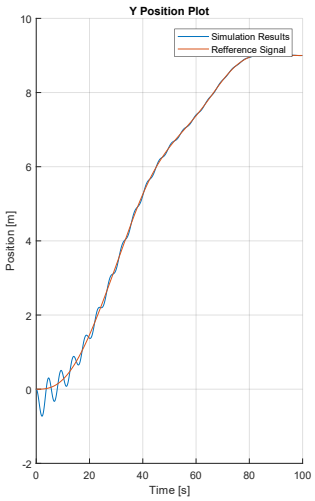
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = 0.1 \quad X_{f(1)} = 2$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0.2 \quad Y_{f(1)} = 4$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0.1 \quad Z_{f(1)} = 4$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0.2 \quad X_{f(2)} = 4$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = 0.1 \quad Y_{f(2)} = 8$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 5$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = 0 \quad X_{f(3)} = 8$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0 \quad Y_{f(3)} = 9$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 0$$

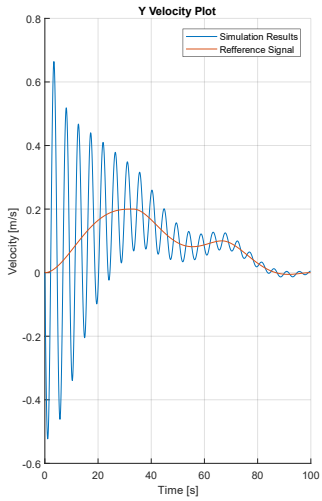
Graphs:

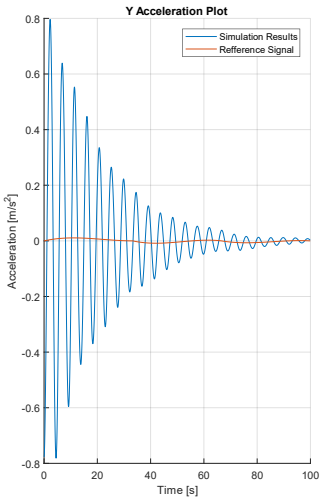

X Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

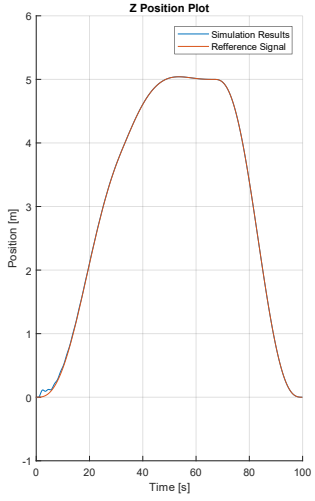

X Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

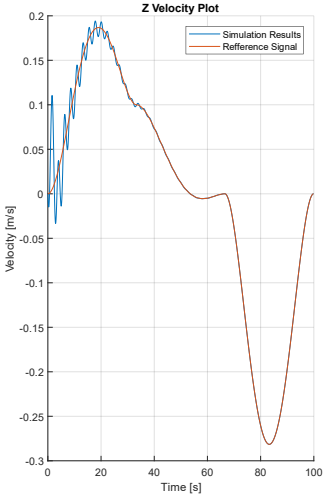

X Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

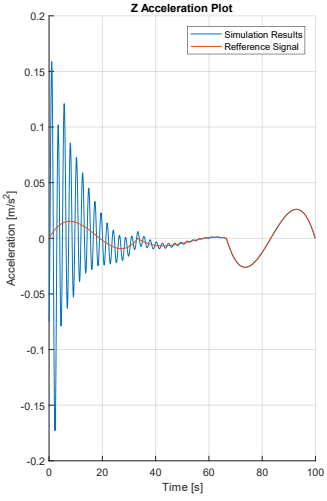

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

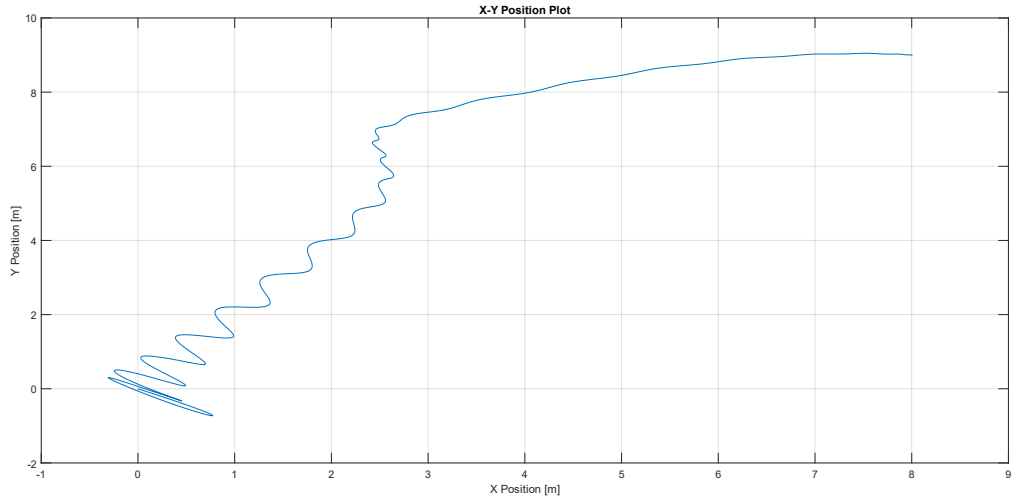

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

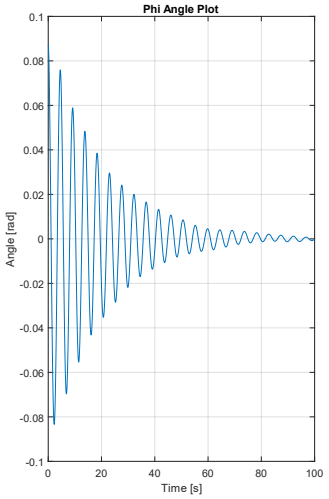

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

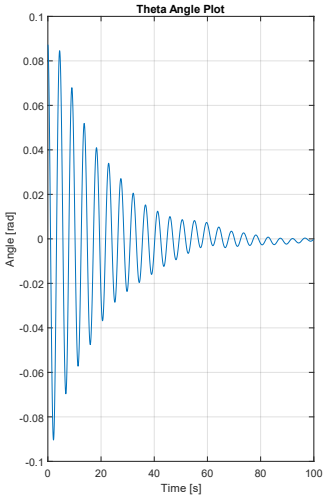

Z Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

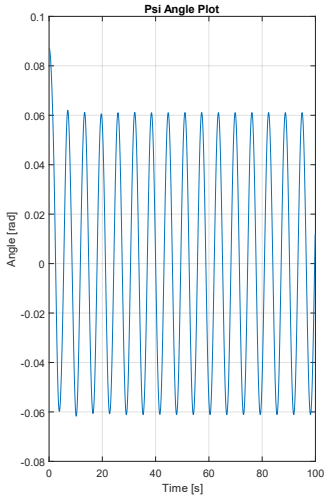

Z Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

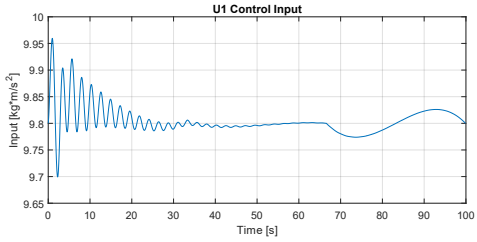

Z Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

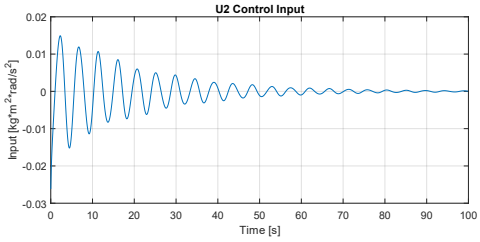

X-Y Position Plot
Y Position [m]
X Position [m]

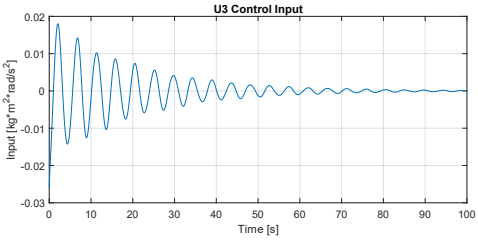

Phi Angle Plot
Angle [rad]
Time [s]

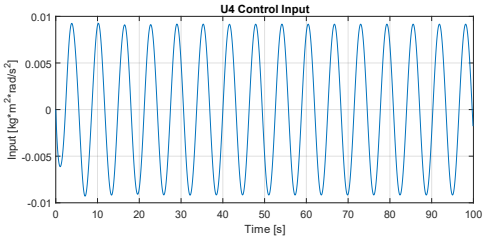

Theta Angle Plot
Angle [rad]
Time [s]

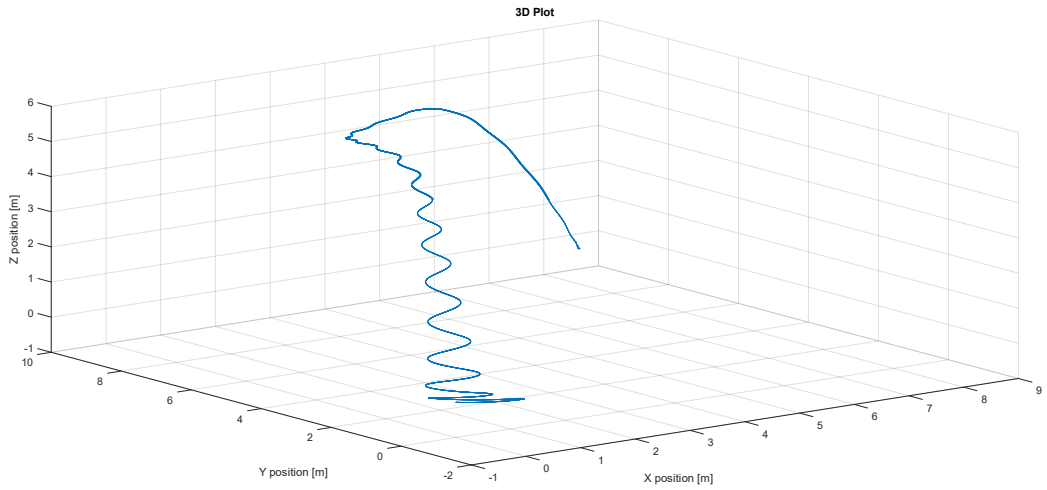

Psi Angle Plot
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
Input [kg*m^2*rad/s^2]
Time [s]



In the fifth case, the same experiment as in the fourth case was done with addition of additional initial tilt. All the Euler angles were tilted by 5 degrees at the start of the simulation. Again, the quadrotor had a similar response to initial tilt angles. It took a long time to suppress the oscillations. Nevertheless, interestingly the oscillations died out faster compared to the hovering with initial tilt and even started following the position reference signal accurately in all three directions.

vi) Case 6: Circular trajectory, flat start

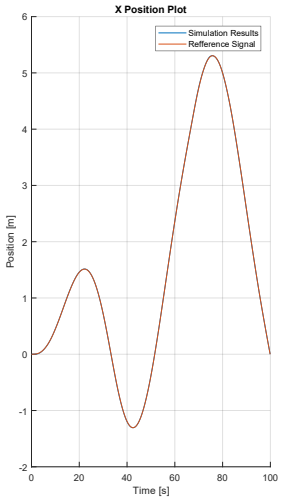
Conditions:

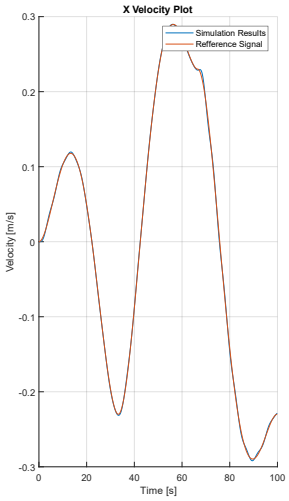
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

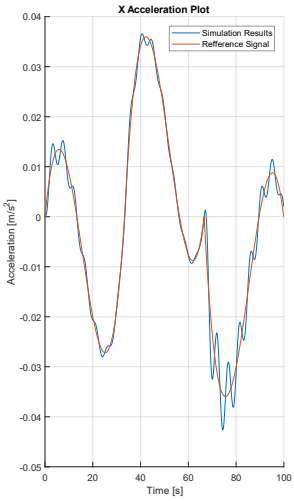
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = 0 \quad \theta_0 = 0 \quad \psi_0 = 0$$
$$k_\eta = 0$$

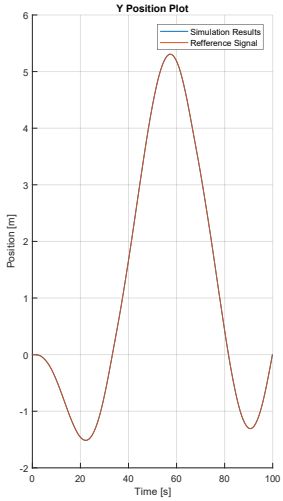
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = -0.23 \quad X_{f(1)} = 0$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0.23 \quad Y_{f(1)} = 0$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0 \quad Z_{f(1)} = 5$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0.23 \quad X_{f(2)} = 4$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = -0.23 \quad Y_{f(2)} = 4$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 5$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = -0.23 \quad X_{f(3)} = 0$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0.23 \quad Y_{f(3)} = 0$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 5$$

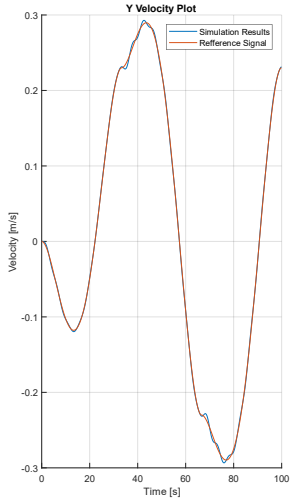
Graphs:

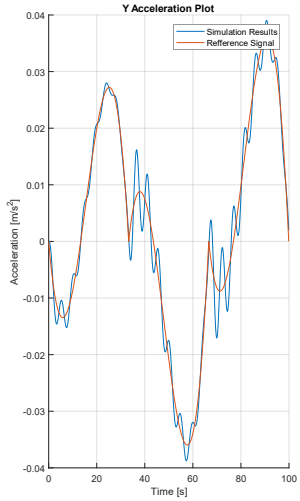

X Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

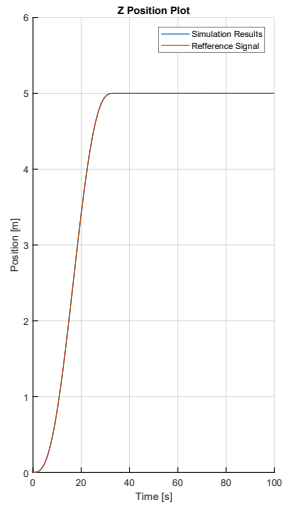

X Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

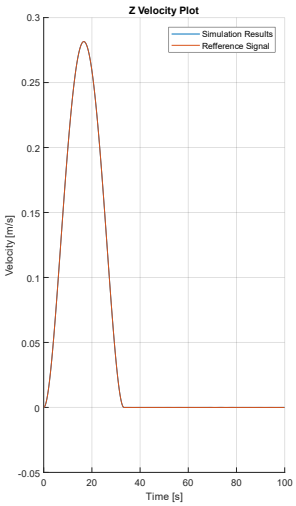

X Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s²]
Time [s]

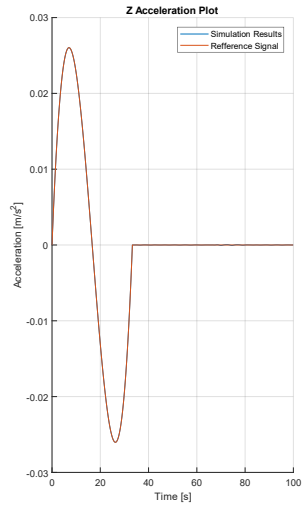

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

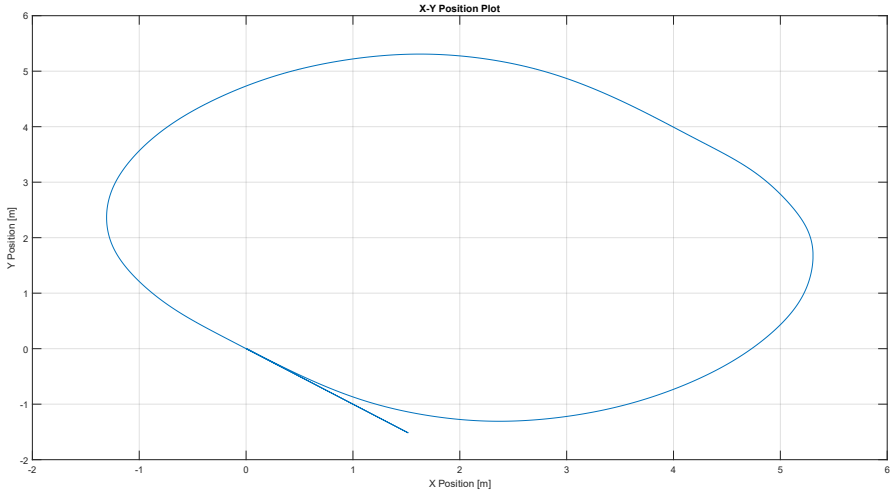

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

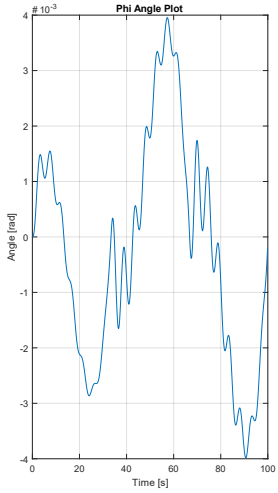

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s²]
Time [s]

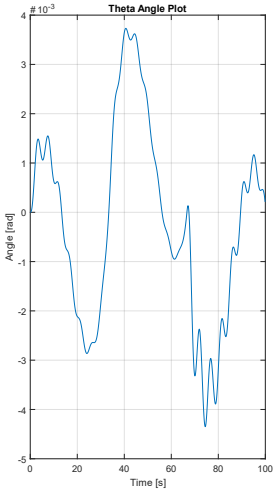

Z Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

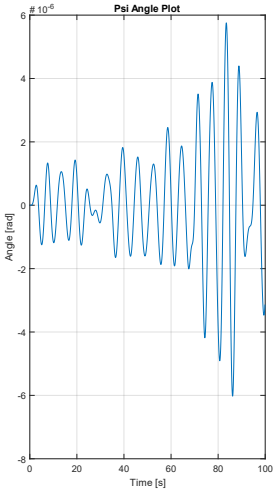

Z Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

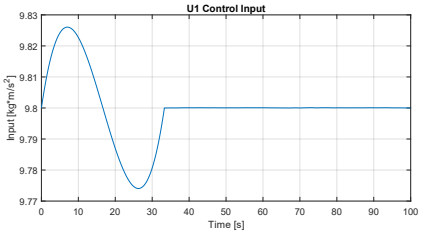

Z Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s²]
Time [s]

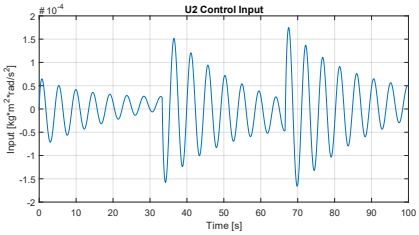

X-Y Position Plot
Y Position [m]
X Position [m]

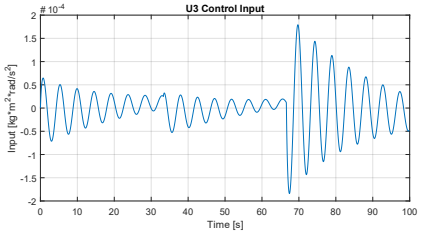

Phi Angle Plot
Angle [rad]
Time [s]

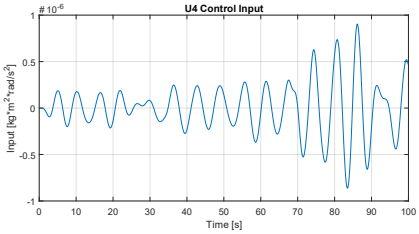

Theta Angle Plot
Angle [rad]
Time [s]

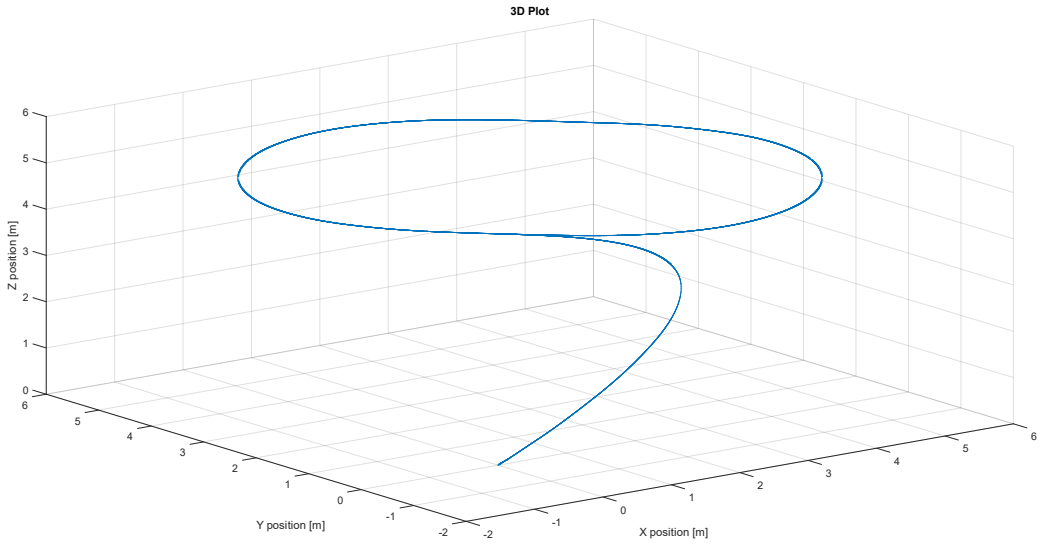

Psi Angle Plot
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
Input [kg*m^2*rad/s^2]
Time [s]



The sixth case has been designed in order to track a circular trajectory. Since I had 3 via points I it was almost impossible to draw a perfectly circular trajectory but tweaking the via point desired positions and desired velocities, I was able to define a desired trajectory close to a circle. The quadrotor was able to follow the desired trajectories in all directions with no noticeable errors. The angles were smoothly changing, and accelerations were very reasonable. The quadrotor turned out to be very robust in following circular trajectories with large enough radius.

vii) Case 7: Circular trajectory, tilted start

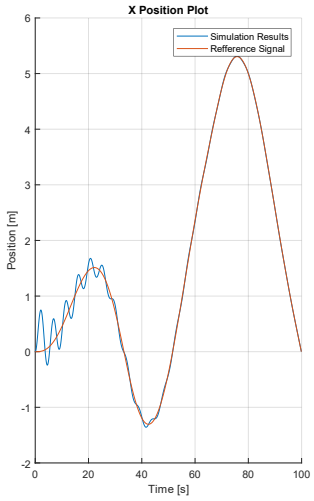
Conditions:

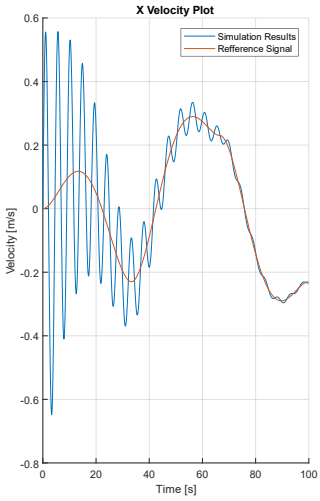
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

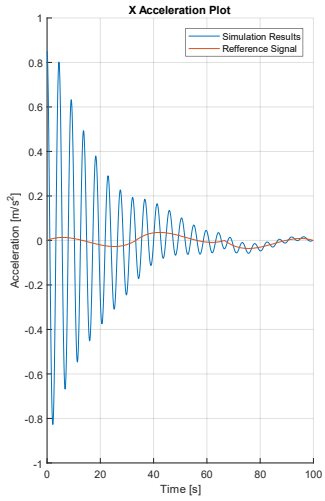
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{36} \quad \theta_0 = \frac{\pi}{36} \quad \psi_0 = 0$$
$$k_\eta = 0$$

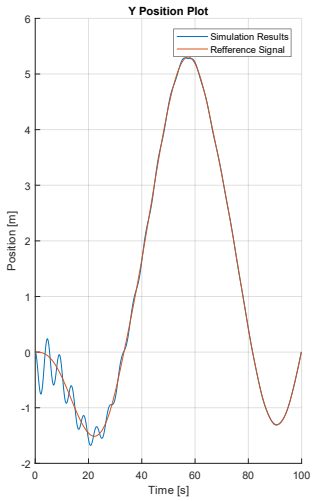
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = -0.23 \quad X_{f(1)} = 0$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0.23 \quad Y_{f(1)} = 0$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0 \quad Z_{f(1)} = 5$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0.23 \quad X_{f(2)} = 4$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = -0.23 \quad Y_{f(2)} = 4$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 5$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = -0.23 \quad X_{f(3)} = 0$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0.23 \quad Y_{f(3)} = 0$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 5$$

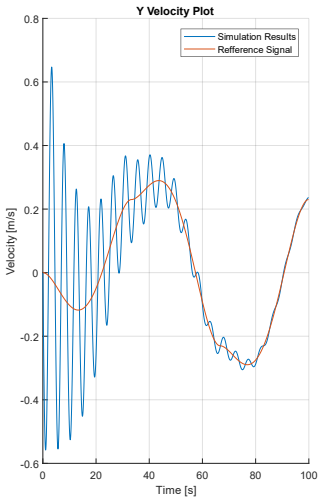
Graphs:

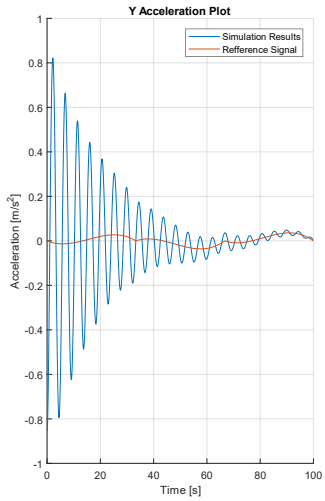

X Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

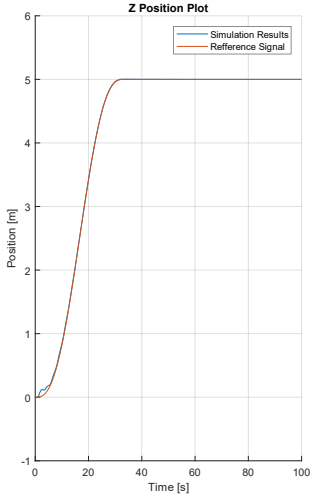

X Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

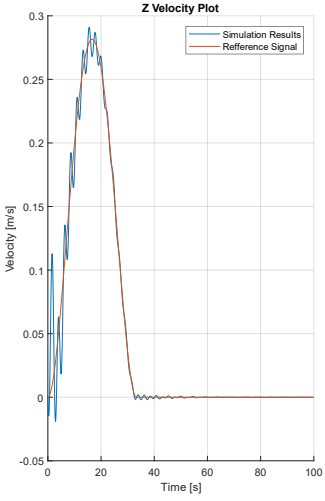

X Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

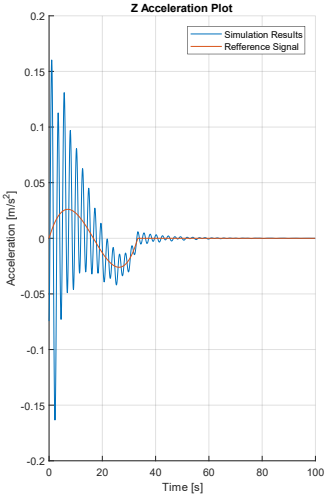

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

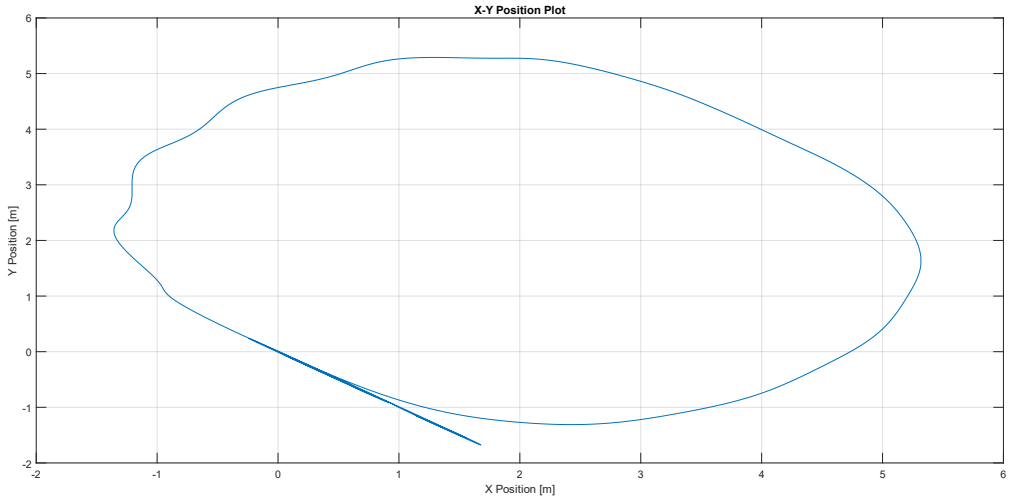

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

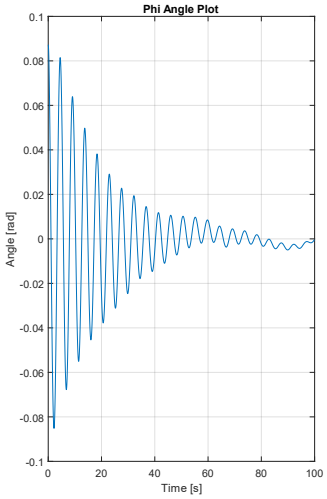

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

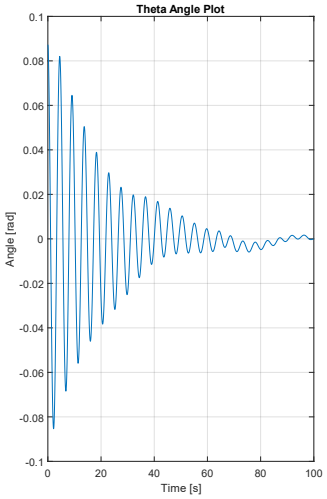

Z Position Plot
Simulation Results
Reference Signal
Position [m]
Time [s]

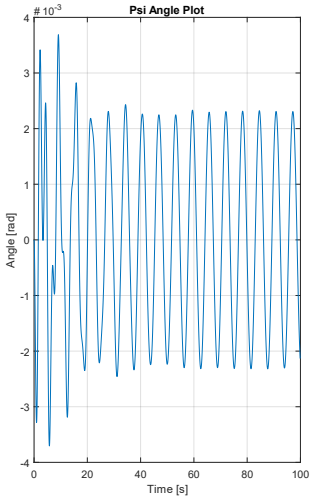

Z Velocity Plot
Simulation Results
Reference Signal
Velocity [m/s]
Time [s]

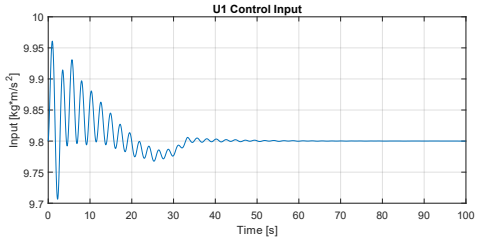

Z Acceleration Plot
Simulation Results
Reference Signal
Acceleration [m/s^2]
Time [s]

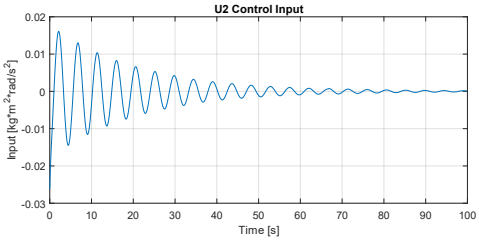

X-Y Position Plot
Y Position [m]
X Position [m]

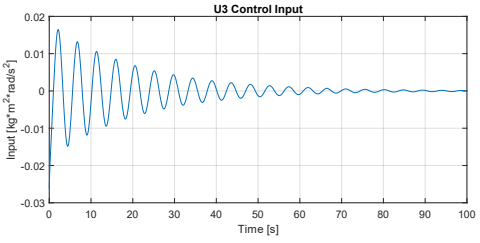

Phi Angle Plot
Angle [rad]
Time [s]

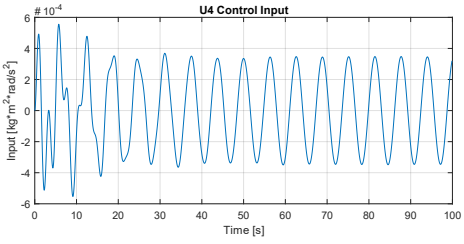

Theta Angle Plot
Angle [rad]
Time [s]

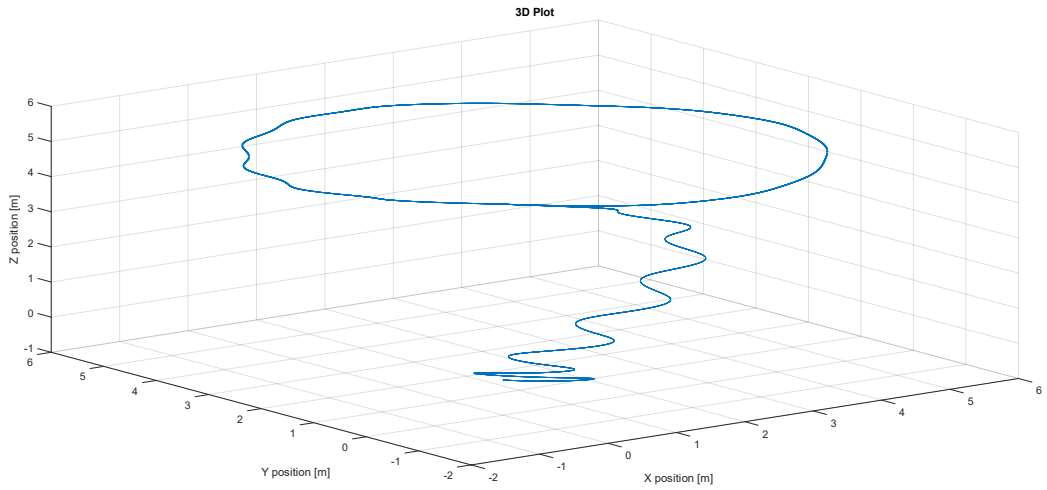

Psi Angle Plot
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
Input [kg*m^2*rad/s^2]
Time [s]



Seventh test case was again the same circular trajectory but with an initial tilted start. The quadrotor had problems following the desired trajectory and had oscillations until reaching the desired altitude. But after that point, the oscillations in X and Y directions also stopped. This was an interesting find because this meant that the quadrotor is able to correct its attitude if it is not moving in the Z-axis.

viii) Case 8: Circular trajectory, tilted start with high disturbance

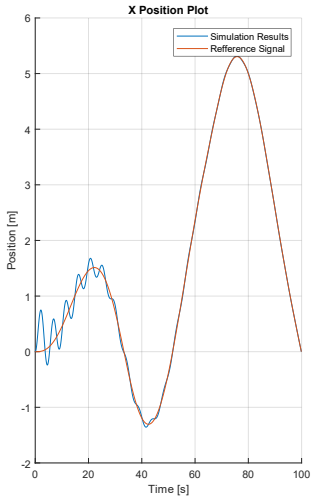
Conditions:

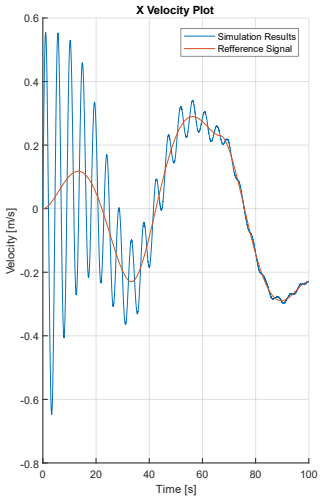
$$K_{p_X} = 0.9 \quad K_{d_X} = 2$$
$$K_{p_Y} = 0.9 \quad K_{d_Y} = 2$$
$$K_{p_Z} = 1 \quad K_{d_Z} = 1$$
$$K_{p_\phi} = 3 \quad K_{d_\phi} = 5$$
$$K_{p_\theta} = 3 \quad K_{d_\theta} = 5$$
$$K_{p_\psi} = 1 \quad K_{d_\psi} = 1$$

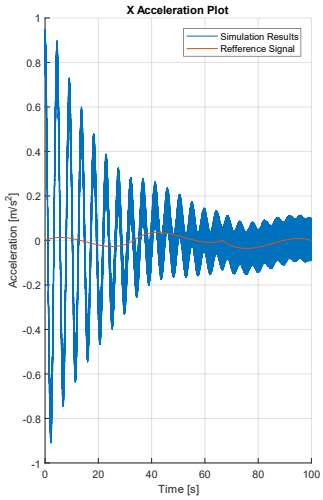
$$\ddot{X}_0 = 0 \quad \dot{X}_0 = 0 \quad X_0 = 0$$
$$\ddot{Y}_0 = 0 \quad \dot{Y}_0 = 0 \quad Y_0 = 0$$
$$Z_0 = 0 \quad Z_0 = 0 \quad Z_0 = 0$$
$$p_0 = 0 \quad q_0 = 0 \quad r_0 = 0$$
$$\phi_0 = \frac{\pi}{36} \quad \theta_0 = \frac{\pi}{36} \quad \psi_0 = 0$$
$$k_\eta = 0.02$$

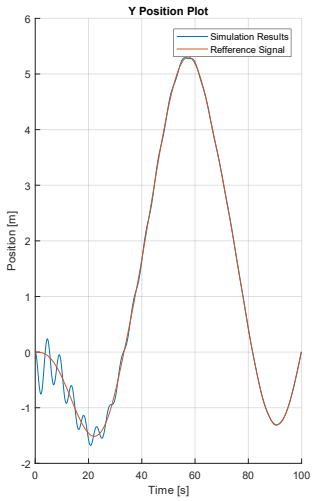
$$\ddot{X}_{f(1)} = 0 \quad \dot{X}_{f(1)} = -0.23 \quad X_{f(1)} = 0$$
$$\ddot{Y}_{f(1)} = 0 \quad \dot{Y}_{f(1)} = 0.23 \quad Y_{f(1)} = 0$$
$$Z_{f(1)} = 0 \quad Z_{f(1)} = 0 \quad Z_{f(1)} = 5$$
$$\ddot{X}_{f(2)} = 0 \quad \dot{X}_{f(2)} = 0.23 \quad X_{f(2)} = 4$$
$$\ddot{Y}_{f(2)} = 0 \quad \dot{Y}_{f(2)} = -0.23 \quad Y_{f(2)} = 4$$
$$Z_{f(2)} = 0 \quad Z_{f(2)} = 0 \quad Z_{f(2)} = 5$$
$$\ddot{X}_{f(3)} = 0 \quad \dot{X}_{f(3)} = -0.23 \quad X_{f(3)} = 0$$
$$\ddot{Y}_{f(3)} = 0 \quad \dot{Y}_{f(3)} = 0.23 \quad Y_{f(3)} = 0$$
$$Z_{f(3)} = 0 \quad Z_{f(3)} = 0 \quad Z_{f(3)} = 5$$

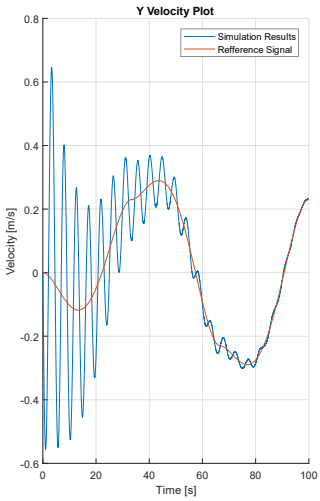
Graphs:

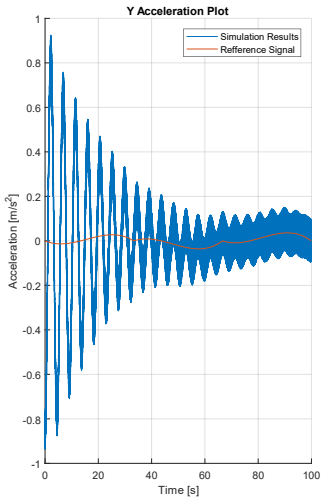

X Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

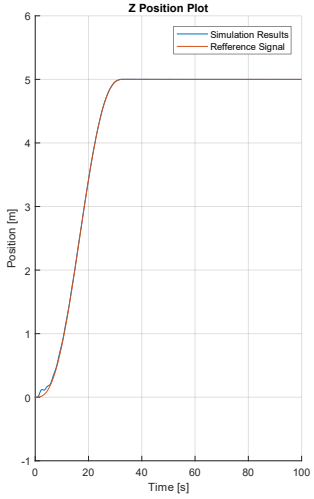

X Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

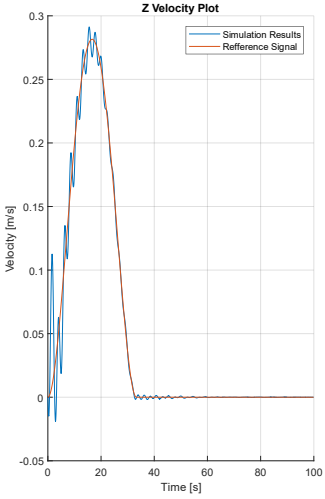

X Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

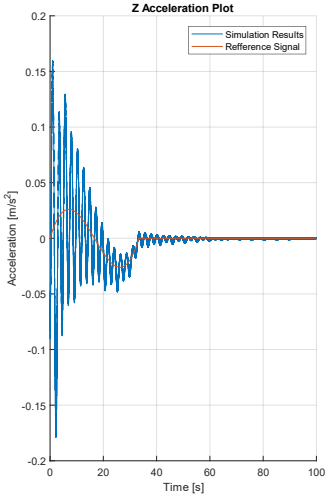

Y Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

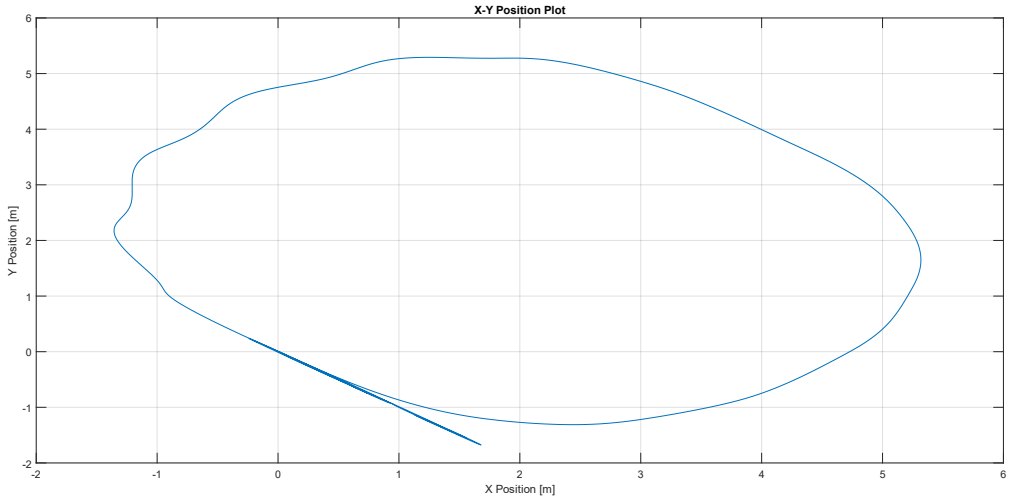

Y Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

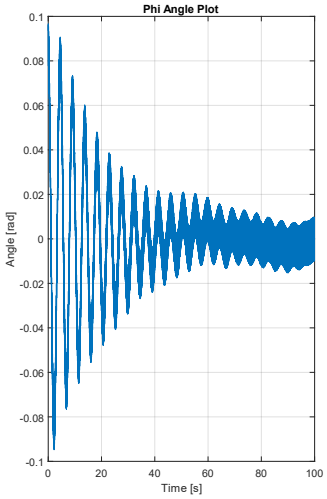

Y Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

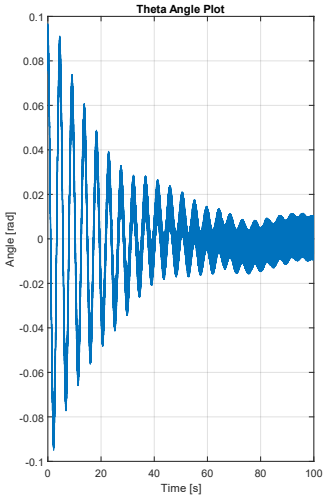

Z Position Plot
Simulation Results
Refference Signal
Position [m]
Time [s]

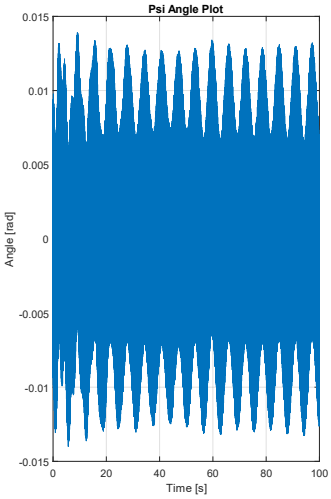

Z Velocity Plot
Simulation Results
Refference Signal
Velocity [m/s]
Time [s]

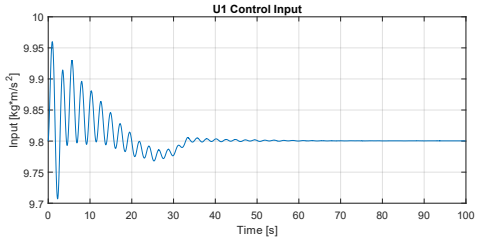

Z Acceleration Plot
Simulation Results
Refference Signal
Acceleration [m/s^2]
Time [s]

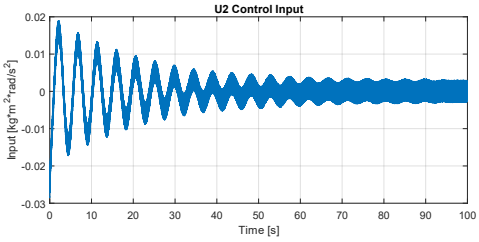

X-Y Position Plot
Y Position [m]
X Position [m]

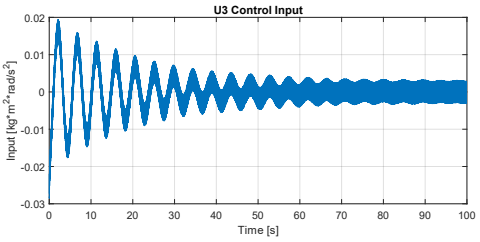

Phi Angle Plot
Angle [rad]
Time [s]

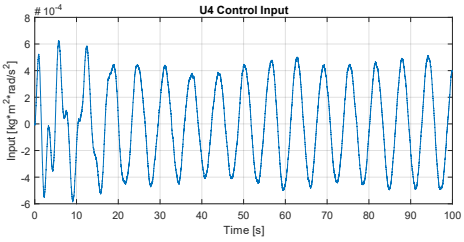

Theta Angle Plot
Angle [rad]
Time [s]

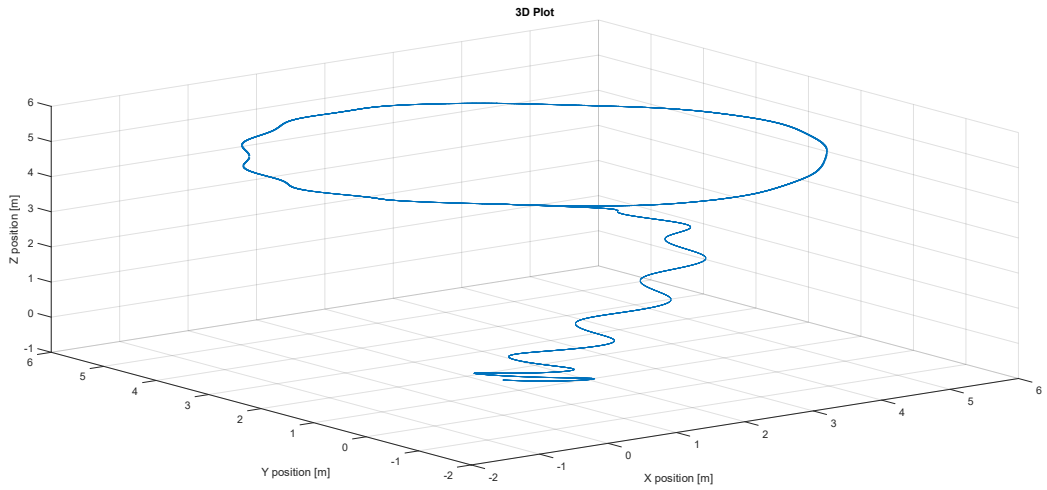

Psi Angle Plot
Angle [rad]
Time [s]


U1 Control Input
Input [kg*m/s^2]
Time [s]


U2 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U3 Control Input
Input [kg*m^2*rad/s^2]
Time [s]


U4 Control Input
10^-4
Input [kg*m^2*rad/s^2]
Time [s]



Eight case was the same experiment as in the seventh case with added disturbance. The added disturbance was a high value, between -1.8 and 1.8 degrees every millisecond. I wanted to push the limits of the quadrotor and see if it would be able to execute such a difficult task. The quadrotor was, again, after some amount time able to follow the desired trajectory with no noticeable errors. This result is very important since it shows that the quadrotor is extremely robust.

4. CONCLUSION

Hover control model can control the altitude and attitude very efficiently and with very little error in all cases other than a case with extreme disturbance. However, it cannot control the drifts that happen in the XY plane caused by initial tilts and disturbances. The system is not very sensitive to control parameters. Higher proportional control gains slightly improve the performance.

In trajectory tracking model, the efficiency of altitude and attitude control is just like in the hover control model, very robust. But the control in X and Y-axis is heavily affected by the initial tilts and can show a lot of oscillations. Nevertheless, if enough time is given, the oscillations in the system disappear and the trajectories can be tracked accurately. The oscillations in X and Y-axis positions die out much quicker if there is no movement in the Z-axis. It is inversely proportional with the movement in the Z-axis. The system is extremely sensitive to control parameters, especially the X and Y gains. With tiny changes the system can become unstable. For example, if K_{p_X} is changed from 0.9 to 1,1, the system becomes unstable.

Both models have shown weaknesses and strengths. Their common attributes are that both models are very robust in altitude and attitude control and are not sensitive against the disturbance model which was used in this project.

It can be concluded that both models are very usable depending on the applications. If the quadrotor will be used as a surveillance tool in a low wind area, the hover control model can be preferred. However, if the quadrotor is expected to move from point A to point B then the trajectory tracking control model is a better choice.

5. REFERENCES

1) Zaki, H., & Unel, M. (2018). Control of a Hovering Quadrotor UAV Subject to Periodic Disturbances. *In: 6th International Conference on Control Engineering & Information Technology, Yildiz Technical University (YTU), in the Davutpasa Convention Centre, Istanbul, Turkey (Accepted/In Press)*

APPENDIX A: Hover Control

ME425_HW2_HoverControl.m

```
clear all; close all; clc;

%simulation parameters
start_time = 0;
step_time = 0.001;
stop_time = 10;
t_sim = start_time:step_time:stop_time;
disturbance_coef = 0.005;

%robot parameters
m = 1;      %kg
g = 9.8;    %m/s^2
Ixx = 0.1;  %kg*m^2
Iyy = 0.1;  %kg*m^2
Izz = 0.15; %kg*m^2

%controller parameters
Kp_x = 0; Kd_x = 0; %No control in X for hovering
Kp_y = 0; Kd_y = 0; %No control in Y for hovering
Kp_z = 8; Kd_z = 3;
Kp_phi = 6; Kd_phi = 1.7;
Kp_theta = 5; Kd_theta = 1.6;
Kp_psi = 2.5; Kd_psi = 4;

%initial values
X_dotdot_init = 0; X_dot_init = 0; X_init = 0;
Y_dotdot_init = 0; Y_dot_init = 0; Y_init = 0;
Z_dotdot_init = 0; Z_dot_init = 0; Z_init = 0;
p_init = 0; q_init = 0; r_init = 0;
Phi_init = 0; Theta_init = 0; Psi_init = 0;

%goal values
X_dotdot_goal = 0; X_dot_goal = 0; X_goal = 0;
Y_dotdot_goal = 0; Y_dot_goal = 0; Y_goal = 0;
Z_dotdot_goal = 0; Z_dot_goal = 0; Z_goal = 1;

[Z_coef, Z_equation, Z_dot_equation, Z_dotdot_equation] =...
    trajectoryPlanner(start_time, stop_time, t_sim,...
    Z_init, Z_dot_init, Z_dotdot_init,...
    Z_goal, Z_dot_goal, Z_dotdot_goal);

sim("HoverControlModel");
```

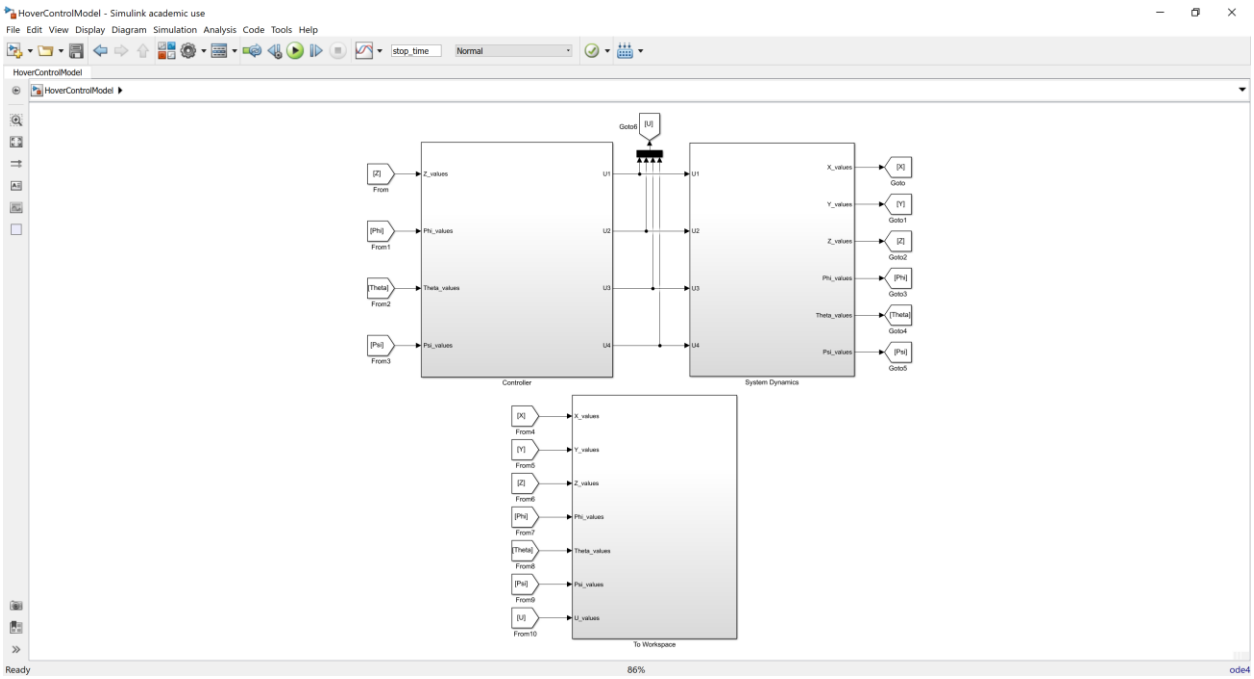
ME425_HW2_HoverControl.m (contd.)

```
fig1 = figure('Name','Z Plot');
subplot(1,3,1); hold on;
plotSimVsRef(Output_Z, Z_equation, t_sim);
title('Z Position Plot'); xlabel('Time [s]'); ylabel('Position
[m]'); grid on; hold off;

subplot(1,3,2); hold on;
plotSimVsRef(Output_Z_dot, Z_dot_equation, t_sim);
title('Z Velocity Plot'); xlabel('Time [s]'); ylabel('Velocity
[m/s]'); grid on; hold off;

subplot(1,3,3); hold on;
plotSimVsRef(Output_Z_dotdot, Z_dotdot_equation, t_sim);
title('Z Acceleration Plot'); xlabel('Time [s]');
ylabel('Acceleration [m/s^2]'); grid on; hold off;

fig2 = figure('Name','X-Y Plot');
plot(Output_X.data, Output_Y.data);
title('X-Y Position Plot'); xlabel('X Position [m]'); ylabel('Y
Position [m]'); grid on;

fig3 = figure('Name','Angle Plot');
subplot(1,3,1); plot(Output_Phi);
title('Phi Angle Plot'); xlabel('Time [s]'); ylabel('Angle [rad]');
grid on;
subplot(1,3,2); plot(Output_Theta);
title('Theta Angle Plot'); xlabel('Time [s]'); ylabel('Angle
[rad]'); grid on;
subplot(1,3,3); plot(Output_Psi);
title('Psi Angle Plot'); xlabel('Time [s]'); ylabel('Angle [rad]');
grid on;

fig4 = figure('Name','U Plot');
subplot(2,2,1); plot(Output_U1);
title('U1 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m/s^2]'); grid on;
subplot(2,2,2); plot(Output_U2);
title('U2 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m^2*rad/s^2]'); grid on;
subplot(2,2,3); plot(Output_U3);
title('U3 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m^2*rad/s^2]'); grid on;
subplot(2,2,4); plot(Output_U4);
title('U2 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m^2*rad/s^2]'); grid on;
```

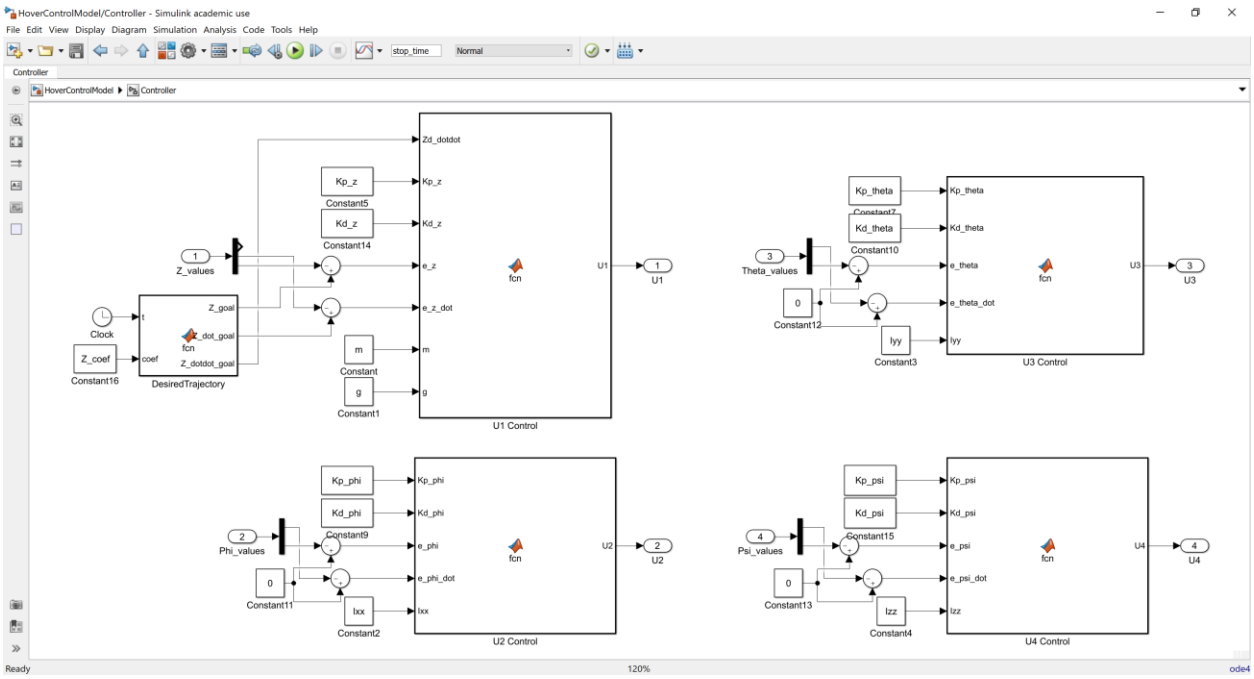
ME425_HW2_HoverControl.m (contd.)

```
saveas(fig1, 'HoverControl_Flat_Normal_Z.svg');
saveas(fig2, 'HoverControl_Flat_Normal_XY.svg');
saveas(fig3, 'HoverControl_Flat_Normal_Angle.svg');
saveas(fig4, 'HoverControl_Flat_Normal_ControlInput.svg');

%gets desired initial values and goal positions for a single
dimension
%and returns the desired trajectory for pos, vel and acc of that
dimension
%as well as the coefficients of the pos trajectory to be used in
simulink
function [coef, pos_equation, vel_equation, acc_equation] =...
    trajectoryPlanner(ti, tf, t, qi, vi, ai, qf, vf, af)

M = [1, ti, ti^2, ti^3, ti^4, ti^5;...
    0, 1, 2*ti, 3*ti^2, 4*ti^3, 5*ti^4;...
    0, 0, 2, 6*ti, 12*ti^2, 20*ti^3;...
    1, tf, tf^2, tf^3, tf^4, tf^5;...
    0, 1, 2*tf, 3*tf^2, 4*tf^3, 5*tf^4;...
    0, 0, 2, 6*tf, 12*tf^2, 20*tf^3];

coef = M \ [qi; vi; ai; qf; vf; af]; % "M\" means "inv(M)*"
pos_equation = coef(1) + coef(2)*t + coef(3)*t.^2 + coef(4)*t.^3 +
coef(5)*t.^4 + coef(6)*t.^5;
vel_equation = coef(2) + 2*coef(3)*t + 3*coef(4)*t.^2 +
4*coef(5)*t.^3 + 5*coef(6)*t.^4;
acc_equation = 2*coef(3) + 6*coef(4)*t + 12*coef(5)*t.^2 +
20*coef(6)*t.^3;
end

function plotSimVsRef(output, equation, t)
plot(output);
plot(t, equation);
legend("Simulation Results", "Refference Signal");
end
```

HoverControlModel



HoverControlModel/Controller



HoverControlModel/Controller/DesiredTrajectory

```
function [Z_goal, Z_dot_goal, Z_dotdot_goal] = fcn(t, coef)

Z_goal = coef(1) + coef(2)*t + coef(3)*t^2 + coef(4)*t^3 +
coef(5)*t^4 + coef(6)*t^5;
Z_dot_goal = coef(2) + 2*coef(3)*t + 3*coef(4)*t^2 + 4*coef(5)*t^3 +
5*coef(6)*t^4;
Z_dotdot_goal = 2*coef(3) + 6*coef(4)*t + 12*coef(5)*t^2 +
20*coef(6)*t^3;
```

HoverControlModel/Controller/U1 Control

```
function U1 = fcn(Zd_dotdot, Kp_z, Kd_z, e_z, e_z_dot, m, g)

U1 = m*(g + Zd_dotdot + Kp_z * e_z + Kd_z * e_z_dot);
```

HoverControlModel/Controller/ U2 Control

```
function U2 = fcn(Kp_phi, Kd_phi, e_phi, e_phi_dot, Ixx)

U2 = Ixx*(Kp_phi * e_phi + Kd_phi * e_phi_dot);
```

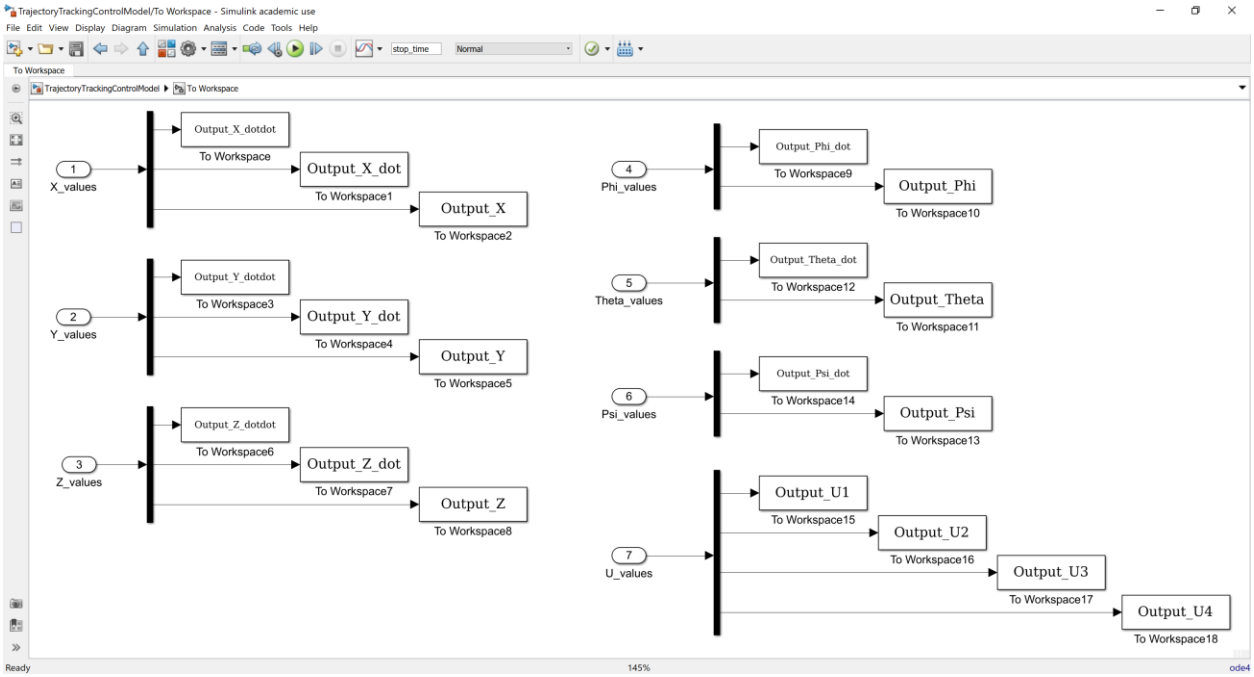
HoverControlModel/Controller/U3 Control

```
function U3 = fcn(Kp_theta, Kd_theta, e_theta, e_theta_dot, Iyy)

U3 = Iyy*(Kp_theta * e_theta + Kd_theta * e_theta_dot);
```

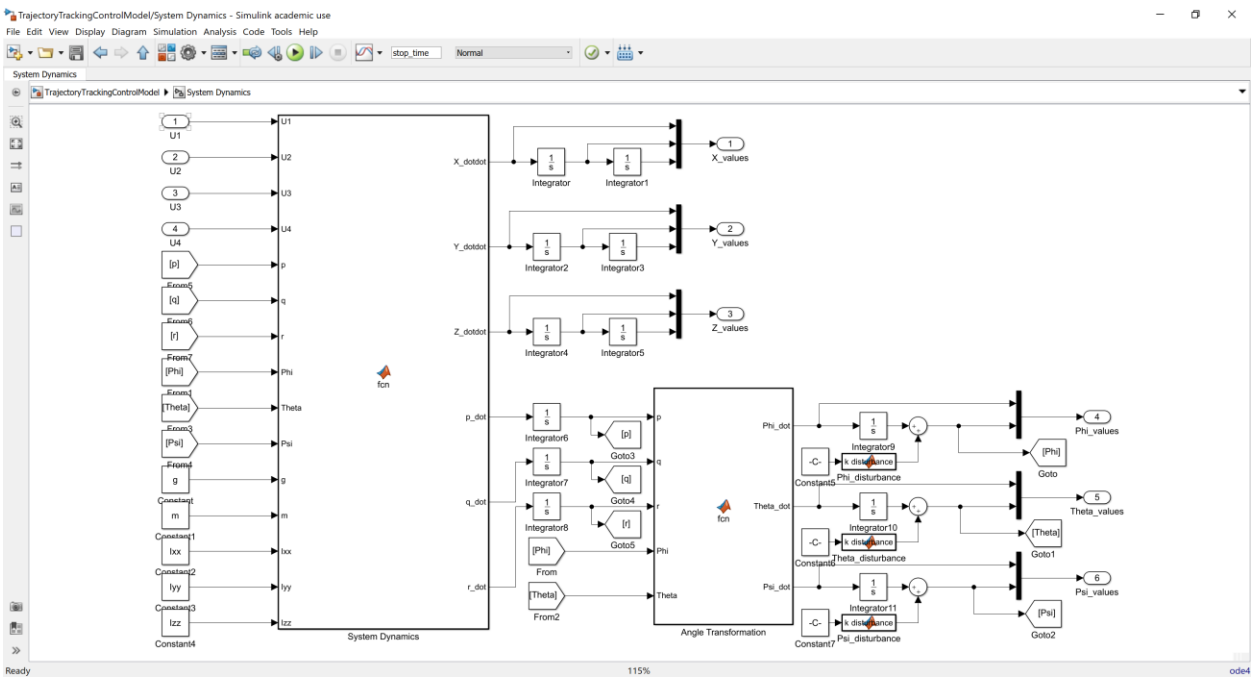
HoverControlModel/Controller/U4 Control

```
function U4 = fcn(Kp_psi, Kd_psi, e_psi, e_psi_dot, Izz)

U4 = Izz*(Kp_psi * e_psi + Kd_psi * e_psi_dot);
```

HoverControlModel/To Workspace



HoverControlModel/System Dynamics



HoverControlModel/System Dynamics/System Dynamics

```
function [X_dotdot, Y_dotdot, Z_dotdot, p_dot, q_dot, r_dot] =
fcn(U1, U2, U3, U4, p, q, r, Phi, Theta, Psi, g, m, Ixx, Iyy, Izz)

X_dotdot = (sin(Psi)*sin(Phi) + cos(Psi)*sin(Theta)*cos(Phi)) *
(U1/m);
Y_dotdot = (-cos(Psi)*sin(Phi) + sin(Psi)*sin(Theta)*cos(Phi)) *
(U1/m);
Z_dotdot = -g + (cos(Theta)*cos(Phi))*(U1/m);
p_dot = ((Iyy-Izz)/Ixx)*q*r + U2/Ixx;
q_dot = ((Izz-Ixx)/Iyy)*p*r + U3/Iyy;
r_dot = ((Ixx-Iyy)/Izz)*p*q + U4/Izz;
```

HoverControlModel/System Dynamics/Angle Transformation

```
function [Phi_dot, Theta_dot, Psi_dot] = fcn(p, q, r, Phi, Theta)

T = [1, sin(Phi)*tan(Theta), cos(Phi)*tan(Theta);...
    0, cos(Phi), -sin(Phi);...
    0, sin(Phi)/cos(Theta), cos(Phi)/cos(Theta)];
y = T*[p; q; r];
Phi_dot = y(1);
Theta_dot = y(2);
Psi_dot = y(3);
```

HoverControlModel/ System Dynamics/Phi_disturbance

```
function disturbance = fcn(k)

disturbance = (rand - 0.5) * k; %disturbance between -0.5*k and
+0.5*k rad
```

HoverControlModel/ System Dynamics/Theta_disturbance

```
function disturbance = fcn(k)

disturbance = (rand - 0.5) * k; %disturbance between -0.5*k and
+0.5*k rad
```

HoverControlModel/ System Dynamics/Psi_disturbance

```
function disturbance = fcn(k)

disturbance = (rand - 0.5) * k; %disturbance between -0.5*k and
+0.5*k rad
```

APPENDIX B: Trajectory Tracking Control

ME425_HW2_TrajectoryTrackingControl.m

```
clear all; close all; clc;

%simulation parameters
start_time = 0;
step_time = 0.001;
stop_time = 99.9;
t_sim = start_time:step_time:stop_time;
t_change(1) = 33.3;
t_change(2) = 66.6;
t_sim_1 = start_time:step_time:t_change(1);
t_sim_2 = t_change(1):step_time:t_change(2);
t_sim_3 = t_change(2):step_time:stop_time;

%disturbance
disturbance_coef = 0;

%robot parameters
m = 1;      %kg
g = 9.8;    %m/s^2
Ixx = 0.1;  %kg*m^2
Iyy = 0.1;  %kg*m^2
Izz = 0.15; %kg*m^2

%controller parameters
Kp_x = 0.9; Kd_x = 2;
Kp_y = 0.9; Kd_y = 2;
Kp_z = 1; Kd_z = 1;
Kp_phi = 3; Kd_phi = 5;
Kp_theta = 3; Kd_theta = 5;
Kp_psi = 1; Kd_psi = 1;

%initial values
X_dotdot_init = 0; X_dot_init = 0; X_init = 0;
Y_dotdot_init = 0; Y_dot_init = 0; Y_init = 0;
Z_dotdot_init = 0; Z_dot_init = 0; Z_init = 0;
p_init = 0; q_init = 0; r_init = 0;
Phi_init = 0; Theta_init = 0; Psi_init = 0;

%via point values
X_dotdot_goal(1) = 0; X_dot_goal(1) = -0.23; X_goal(1) = 0;
Y_dotdot_goal(1) = 0; Y_dot_goal(1) = 0.23; Y_goal(1) = 0;
Z_dotdot_goal(1) = 0; Z_dot_goal(1) = 0; Z_goal(1) = 5;
```

ME425_HW2_TrajectoryTrackingControl.m (contd.)

```
X_dotdot_goal(2) = 0; X_dot_goal(2) = 0.23; X_goal(2) = 4;
Y_dotdot_goal(2) = 0; Y_dot_goal(2) = -0.23; Y_goal(2) = 4;
Z_dotdot_goal(2) = 0; Z_dot_goal(2) = 0; Z_goal(2) = 5;

X_dotdot_goal(3) = 0; X_dot_goal(3) = -0.23; X_goal(3) = 0;
Y_dotdot_goal(3) = 0; Y_dot_goal(3) = 0.23; Y_goal(3) = 0;
Z_dotdot_goal(3) = 0; Z_dot_goal(3) = 0; Z_goal(3) = 5;

%X direction trajectory calculation
[X_coef_1, X_equation(:,1), X_dot_equation(:,1),
X_dotdot_equation(:,1)] =...
    trajectoryPlanner(start_time, t_change(1), t_sim_1,...
    X_init, X_dot_init, X_dotdot_init,...
    X_goal(1), X_dot_goal(1), X_dotdot_goal(1));

[X_coef_2, X_equation(:,2), X_dot_equation(:,2),
X_dotdot_equation(:,2)] =...
    trajectoryPlanner(t_change(1), t_change(2), t_sim_2,...
    X_goal(1), X_dot_goal(1), X_dotdot_goal(1),...
    X_goal(2), X_dot_goal(2), X_dotdot_goal(2));

[X_coef_3, X_equation(:,3), X_dot_equation(:,3),
X_dotdot_equation(:,3)] =...
    trajectoryPlanner(t_change(2), stop_time, t_sim_3,...
    X_goal(2), X_dot_goal(2), X_dotdot_goal(2),...
    X_goal(3), X_dot_goal(3), X_dotdot_goal(3));

%Y direction trajectory calculation
[Y_coef_1, Y_equation(:,1), Y_dot_equation(:,1),
Y_dotdot_equation(:,1)] =...
    trajectoryPlanner(start_time, t_change(1), t_sim_1,...
    Y_init, Y_dot_init, Y_dotdot_init,...
    Y_goal(1), Y_dot_goal(1), Y_dotdot_goal(1));

[Y_coef_2, Y_equation(:,2), Y_dot_equation(:,2),
Y_dotdot_equation(:,2)] =...
    trajectoryPlanner(t_change(1), t_change(2), t_sim_2,...
    Y_goal(1), Y_dot_goal(1), Y_dotdot_goal(1),...
    Y_goal(2), Y_dot_goal(2), Y_dotdot_goal(2));

[Y_coef_3, Y_equation(:,3), Y_dot_equation(:,3),
Y_dotdot_equation(:,3)] =...
    trajectoryPlanner(t_change(2), stop_time, t_sim_3,...
    Y_goal(2), Y_dot_goal(2), Y_dotdot_goal(2),...
    Y_goal(3), Y_dot_goal(3), Y_dotdot_goal(3));
```

ME425_HW2_TrajectoryTrackingControl.m (contd.)

```
%Z direction trajectory calculation
[Z_coef_1, Z_equation(:,1), Z_dot_equation(:,1),
Z_dotdot_equation(:,1)] =...
    trajectoryPlanner(start_time, t_change(1), t_sim_1,...
    Z_init, Z_dot_init, Z_dotdot_init,...
    Z_goal(1), Z_dot_goal(1), Z_dotdot_goal(1));

[Z_coef_2, Z_equation(:,2), Z_dot_equation(:,2),
Z_dotdot_equation(:,2)] =...
    trajectoryPlanner(t_change(1), t_change(2), t_sim_2,...
    Z_goal(1), Z_dot_goal(1), Z_dotdot_goal(1),...
    Z_goal(2), Z_dot_goal(2), Z_dotdot_goal(2));

[Z_coef_3, Z_equation(:,3), Z_dot_equation(:,3),
Z_dotdot_equation(:,3)] =...
    trajectoryPlanner(t_change(2), stop_time, t_sim_3,...
    Z_goal(2), Z_dot_goal(2), Z_dotdot_goal(2),...
    Z_goal(3), Z_dot_goal(3), Z_dotdot_goal(3));

%simulating the simulink model
sim("TrajectoryTrackingControlModel");

%plot X data
X_equation_total = [X_equation(:,1); X_equation(:,2);
X_equation(:,3)];
X_dot_equation_total = [X_dot_equation(:,1); X_dot_equation(:,2);
X_dot_equation(:,3)];
X_dotdot_equation_total = [X_dotdot_equation(:,1);
X_dotdot_equation(:,2); X_dotdot_equation(:,3)];
X_equation_total = X_equation_total(1:end-2,:);
X_dot_equation_total = X_dot_equation_total(1:end-2,:);
X_dotdot_equation_total = X_dotdot_equation_total(1:end-2,:);

fig1 = figure('Name','X Plot');
subplot(1,3,1); hold on;
plotSimVsRef(Output_X, X_equation_total, t_sim);
title('X Position Plot'); xlabel('Time [s]'); ylabel('Position
[m]'); grid on; hold off;

subplot(1,3,2); hold on;
plotSimVsRef(Output_X_dot, X_dot_equation_total, t_sim);
title('X Velocity Plot'); xlabel('Time [s]'); ylabel('Velocity
[m/s]'); grid on; hold off;
```

ME425_HW2_TrajectoryTrackingControl.m (contd.)

```
subplot(1,3,3); hold on;
plotSimVsRef(Output_X_dotdot, X_dotdot_equation_total, t_sim);
title('X Acceleration Plot'); xlabel('Time [s]');
ylabel('Acceleration [m/s^2]'); grid on; hold off;

%Plot Y data
Y_equation_total = [Y_equation(:,1); Y_equation(:,2);
Y_equation(:,3)];
Y_dot_equation_total = [Y_dot_equation(:,1); Y_dot_equation(:,2);
Y_dot_equation(:,3)];
Y_dotdot_equation_total = [Y_dotdot_equation(:,1);
Y_dotdot_equation(:,2); Y_dotdot_equation(:,3)];
Y_equation_total = Y_equation_total(1:end-2,:);
Y_dot_equation_total = Y_dot_equation_total(1:end-2,:);
Y_dotdot_equation_total = Y_dotdot_equation_total(1:end-2,:);

fig2 = figure('Name','Y Plot');
subplot(1,3,1); hold on;
plotSimVsRef(Output_Y, Y_equation_total, t_sim);
title('Y Position Plot'); xlabel('Time [s]'); ylabel('Position
[m]'); grid on; hold off;

subplot(1,3,2); hold on;
plotSimVsRef(Output_Y_dot, Y_dot_equation_total, t_sim);
title('Y Velocity Plot'); xlabel('Time [s]'); ylabel('Velocity
[m/s]'); grid on; hold off;

subplot(1,3,3); hold on;
plotSimVsRef(Output_Y_dotdot, Y_dotdot_equation_total, t_sim);
title('Y Acceleration Plot'); xlabel('Time [s]');
ylabel('Acceleration [m/s^2]'); grid on; hold off;

%Plot Z data
Z_equation_total = [Z_equation(:,1); Z_equation(:,2);
Z_equation(:,3)];
Z_dot_equation_total = [Z_dot_equation(:,1); Z_dot_equation(:,2);
Z_dot_equation(:,3)];
Z_dotdot_equation_total = [Z_dotdot_equation(:,1);
Z_dotdot_equation(:,2); Z_dotdot_equation(:,3)];
Z_equation_total = Z_equation_total(1:end-2,:);
Z_dot_equation_total = Z_dot_equation_total(1:end-2,:);
Z_dotdot_equation_total = Z_dotdot_equation_total(1:end-2,:);
```

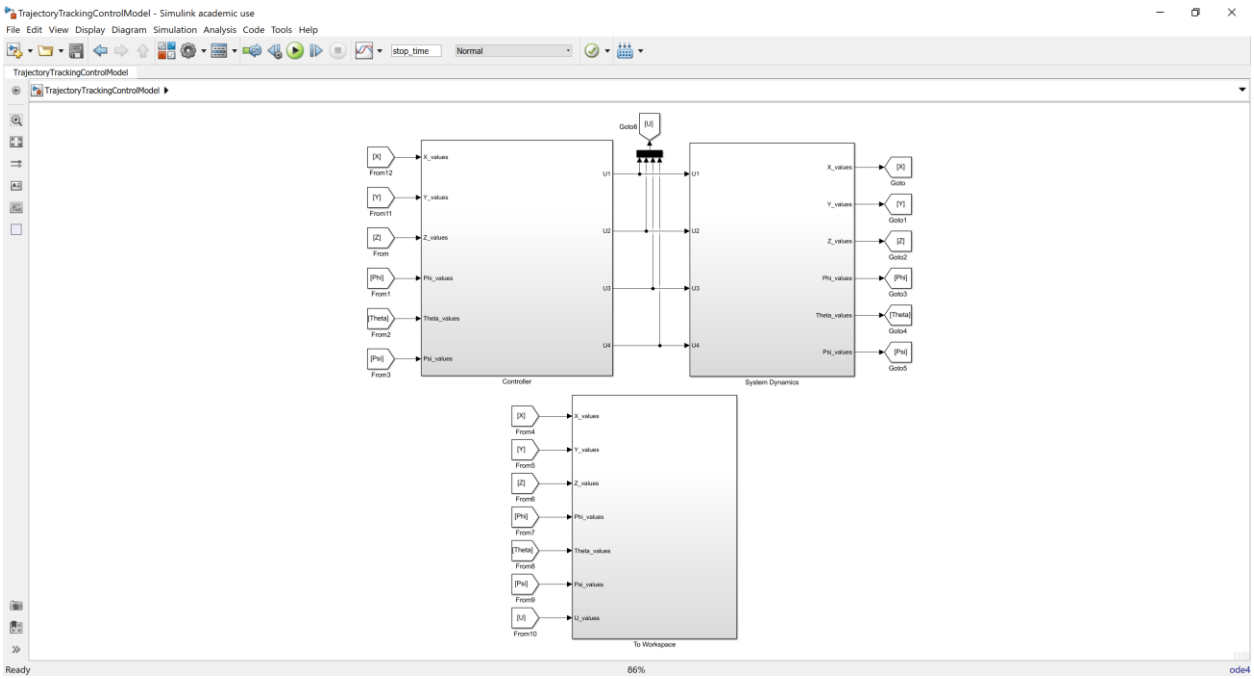
ME425_HW2_TrajectoryTrackingControl.m (contd.)

```
fig3 = figure('Name','Z Plot');
subplot(1,3,1); hold on;
plotSimVsRef(Output_Z, Z_equation_total, t_sim);
title('Z Position Plot'); xlabel('Time [s]'); ylabel('Position
[m]'); grid on; hold off;

subplot(1,3,2); hold on;
plotSimVsRef(Output_Z_dot, Z_dot_equation_total, t_sim);
title('Z Velocity Plot'); xlabel('Time [s]'); ylabel('Velocity
[m/s]'); grid on; hold off;

subplot(1,3,3); hold on;
plotSimVsRef(Output_Z_dotdot, Z_dotdot_equation_total, t_sim);
title('Z Acceleration Plot'); xlabel('Time [s]');
ylabel('Acceleration [m/s^2]'); grid on; hold off;

%plot X vs Y data
fig4 = figure('Name','X-Y Plot');
plot(Output_X.data, Output_Y.data);
title('X-Y Position Plot'); xlabel('X Position [m]'); ylabel('Y
Position [m]'); grid on;

%plot Angles data
fig5 = figure('Name','Angle Plot');
subplot(1,3,1); plot(Output_Phi);
title('Phi Angle Plot'); xlabel('Time [s]'); ylabel('Angle [rad]');
grid on;
subplot(1,3,2); plot(Output_Theta);
title('Theta Angle Plot'); xlabel('Time [s]'); ylabel('Angle
[rad]'); grid on;
subplot(1,3,3); plot(Output_Psi);
title('Psi Angle Plot'); xlabel('Time [s]'); ylabel('Angle [rad]');
grid on;

%plot control inputs
fig6 = figure('Name','U Plot');
subplot(2,2,1); plot(Output_U1);
title('U1 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m/s^2]'); grid on;
subplot(2,2,2); plot(Output_U2);
title('U2 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m^2*rad/s^2]'); grid on;
```

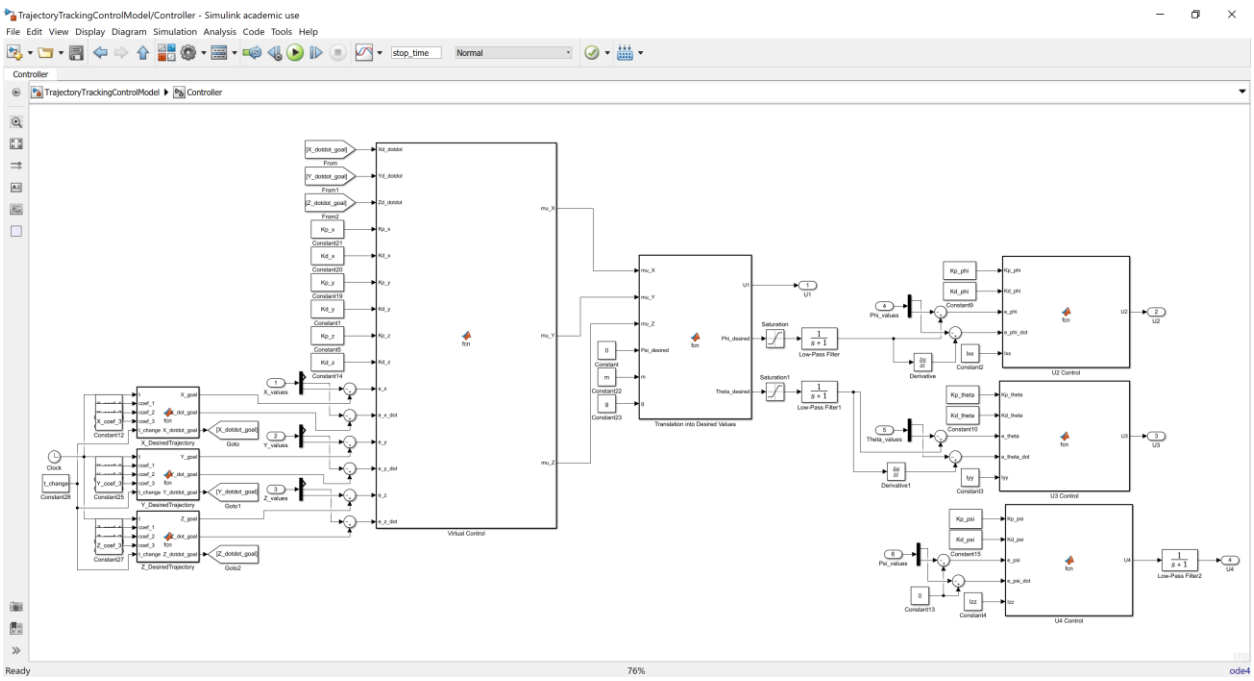
ME425_HW2_TrajectoryTrackingControl.m (contd.)

```
subplot(2,2,3); plot(Output_U3);
title('U3 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m^2*rad/s^2]'); grid on;
subplot(2,2,4); plot(Output_U4);
title('U4 Control Input'); xlabel('Time [s]'); ylabel('Input
[kg*m^2*rad/s^2]'); grid on;

%plot movement in 3D
fig7 = figure('Name','3D Plot');
plot3(Output_X.data, Output_Y.data, Output_Z.data);
title('3D Plot'); xlabel('X position [m]'); ylabel('Y position
[m]'); zlabel('Z position [m]'); grid on;

%gets desired initial values and goal positions for a single
dimension
%and returns the desired trajectory for pos, vel and acc of that
dimension
%as well as the coefficients of the pos trajectory to be used in
simulink
function [coef, pos_equation, vel_equation, acc_equation] =...
    trajectoryPlanner(ti, tf, t, qi, vi, ai, qf, vf, af)

M = [1, ti, ti^2, ti^3, ti^4, ti^5;...
    0, 1, 2*ti, 3*ti^2, 4*ti^3, 5*ti^4;...
    0, 0, 2, 6*ti, 12*ti^2, 20*ti^3;...
    1, tf, tf^2, tf^3, tf^4, tf^5;...
    0, 1, 2*tf, 3*tf^2, 4*tf^3, 5*tf^4;...
    0, 0, 2, 6*tf, 12*tf^2, 20*tf^3];

coef = M \ [qi; vi; ai; qf; vf; af]; % "M\" means "inv(M)*"
pos_equation = coef(1) + coef(2)*t + coef(3)*t.^2 + coef(4)*t.^3 +
coef(5)*t.^4 + coef(6)*t.^5;
vel_equation = coef(2) + 2*coef(3)*t + 3*coef(4)*t.^2 +
4*coef(5)*t.^3 + 5*coef(6)*t.^4;
acc_equation = 2*coef(3) + 6*coef(4)*t + 12*coef(5)*t.^2 +
20*coef(6)*t.^3;
end

function plotSimVsRef(output, equation, t)
plot(output);
plot(t, equation);
legend("Simulation Results", "Refference Signal");
end
```

TrajectoryTrackingControlModel



TrajectoryTrackingControlModel/Controller



TrajectoryTrackingControlModel/Controller/X_DesiredTrajectory

```
function [X_goal, X_dot_goal, X_dotdot_goal] = fcn(t, coef_1,
coef_2, coef_3, t_change)

if t<=t_change(1)
    X_goal = coef_1(1) + coef_1(2)*t + coef_1(3)*t^2 + coef_1(4)*t^3
+ coef_1(5)*t^4 + coef_1(6)*t^5;
    X_dot_goal = coef_1(2) + 2*coef_1(3)*t + 3*coef_1(4)*t^2 +
4*coef_1(5)*t^3 + 5*coef_1(6)*t^4;
    X_dotdot_goal = 2*coef_1(3) + 6*coef_1(4)*t + 12*coef_1(5)*t^2 +
20*coef_1(6)*t^3;
elseif t<=t_change(2)
    X_goal = coef_2(1) + coef_2(2)*t + coef_2(3)*t^2 + coef_2(4)*t^3
+ coef_2(5)*t^4 + coef_2(6)*t^5;
    X_dot_goal = coef_2(2) + 2*coef_2(3)*t + 3*coef_2(4)*t^2 +
4*coef_2(5)*t^3 + 5*coef_2(6)*t^4;
    X_dotdot_goal = 2*coef_2(3) + 6*coef_2(4)*t + 12*coef_2(5)*t^2 +
20*coef_2(6)*t^3;
else
    X_goal = coef_3(1) + coef_3(2)*t + coef_3(3)*t^2 + coef_3(4)*t^3
+ coef_3(5)*t^4 + coef_3(6)*t^5;
    X_dot_goal = coef_3(2) + 2*coef_3(3)*t + 3*coef_3(4)*t^2 +
4*coef_3(5)*t^3 + 5*coef_3(6)*t^4;
    X_dotdot_goal = 2*coef_3(3) + 6*coef_3(4)*t + 12*coef_3(5)*t^2 +
20*coef_3(6)*t^3;
end
```

TrajectoryTrackingControlModel/Controller/Y_DesiredTrajectory

```
function [Y_goal, Y_dot_goal, Y_dotdot_goal] = fcn(t, coef_1,
coef_2, coef_3, t_change)

if t<=t_change(1)
    Y_goal = coef_1(1) + coef_1(2)*t + coef_1(3)*t^2 + coef_1(4)*t^3
+ coef_1(5)*t^4 + coef_1(6)*t^5;
    Y_dot_goal = coef_1(2) + 2*coef_1(3)*t + 3*coef_1(4)*t^2 +
4*coef_1(5)*t^3 + 5*coef_1(6)*t^4;
    Y_dotdot_goal = 2*coef_1(3) + 6*coef_1(4)*t + 12*coef_1(5)*t^2 +
20*coef_1(6)*t^3;
elseif t<=t_change(2)
    Y_goal = coef_2(1) + coef_2(2)*t + coef_2(3)*t^2 + coef_2(4)*t^3
+ coef_2(5)*t^4 + coef_2(6)*t^5;
    Y_dot_goal = coef_2(2) + 2*coef_2(3)*t + 3*coef_2(4)*t^2 +
4*coef_2(5)*t^3 + 5*coef_2(6)*t^4;
    Y_dotdot_goal = 2*coef_2(3) + 6*coef_2(4)*t + 12*coef_2(5)*t^2 +
20*coef_2(6)*t^3;
```

TrajectoryTrackingControlModel/Controller/Y_DesiredTrajectory (contd.)

```
else
    Y_goal = coef_3(1) + coef_3(2)*t + coef_3(3)*t^2 + coef_3(4)*t^3
+ coef_3(5)*t^4 + coef_3(6)*t^5;
    Y_dot_goal = coef_3(2) + 2*coef_3(3)*t + 3*coef_3(4)*t^2 +
4*coef_3(5)*t^3 + 5*coef_3(6)*t^4;
    Y_dotdot_goal = 2*coef_3(3) + 6*coef_3(4)*t + 12*coef_3(5)*t^2 +
20*coef_3(6)*t^3;
end
```

TrajectoryTrackingControlModel/Controller/Z_DesiredTrajectory

```
function [Z_goal, Z_dot_goal, Z_dotdot_goal] = fcn(t, coef_1,
coef_2, coef_3, t_change)

if t<=t_change(1)
    Z_goal = coef_1(1) + coef_1(2)*t + coef_1(3)*t^2 + coef_1(4)*t^3
+ coef_1(5)*t^4 + coef_1(6)*t^5;
    Z_dot_goal = coef_1(2) + 2*coef_1(3)*t + 3*coef_1(4)*t^2 +
4*coef_1(5)*t^3 + 5*coef_1(6)*t^4;
    Z_dotdot_goal = 2*coef_1(3) + 6*coef_1(4)*t + 12*coef_1(5)*t^2 +
20*coef_1(6)*t^3;
elseif t<=t_change(2)
    Z_goal = coef_2(1) + coef_2(2)*t + coef_2(3)*t^2 + coef_2(4)*t^3
+ coef_2(5)*t^4 + coef_2(6)*t^5;
    Z_dot_goal = coef_2(2) + 2*coef_2(3)*t + 3*coef_2(4)*t^2 +
4*coef_2(5)*t^3 + 5*coef_2(6)*t^4;
    Z_dotdot_goal = 2*coef_2(3) + 6*coef_2(4)*t + 12*coef_2(5)*t^2 +
20*coef_2(6)*t^3;
else
    Z_goal = coef_3(1) + coef_3(2)*t + coef_3(3)*t^2 + coef_3(4)*t^3
+ coef_3(5)*t^4 + coef_3(6)*t^5;
    Z_dot_goal = coef_3(2) + 2*coef_3(3)*t + 3*coef_3(4)*t^2 +
4*coef_3(5)*t^3 + 5*coef_3(6)*t^4;
    Z_dotdot_goal = 2*coef_3(3) + 6*coef_3(4)*t + 12*coef_3(5)*t^2 +
20*coef_3(6)*t^3;
end
```

TrajectoryTrackingControlModel/Controller/Virtual Control

```
function [mu_X, mu_Y, mu_Z] = fcn(Xd_dotdot, Yd_dotdot,
Zd_dotdot,...
    Kp_x, Kd_x, Kp_y, Kd_y, Kp_z, Kd_z,...
    e_x, e_x_dot, e_y, e_y_dot, e_z, e_z_dot)

mu_X = Xd_dotdot + Kp_x * e_x + Kd_x * e_x_dot;
mu_Y = Yd_dotdot + Kp_y * e_y + Kd_y * e_y_dot;
mu_Z = Zd_dotdot + Kp_z * e_z + Kd_z * e_z_dot;
```

TrajectoryTrackingControlModel/Controller/Translation into Desired Values

```
function [U1, Phi_desired, Theta_desired] = fcn(mu_X, mu_Y, mu_Z,
Psi_desired, m, g)

U1 = m*(mu_X^2 + mu_Y^2 + (mu_Z + g)^2)^(1/2);
Phi_desired = asin((sin(Psi_desired) * mu_X - cos(Psi_desired) *
mu_Y) / (U1 / m));
Theta_desired = asin((cos(Psi_desired) * mu_X + sin(Psi_desired) *
mu_Y) / (cos(Phi_desired) * (U1 / m)));
```

TrajectoryTrackingControlModel/Controller/U2 Control

```
function U2 = fcn(Kp_phi, Kd_phi, e_phi, e_phi_dot, Ixx)

U2 = Ixx*(Kp_phi * e_phi + Kd_phi * e_phi_dot);
```

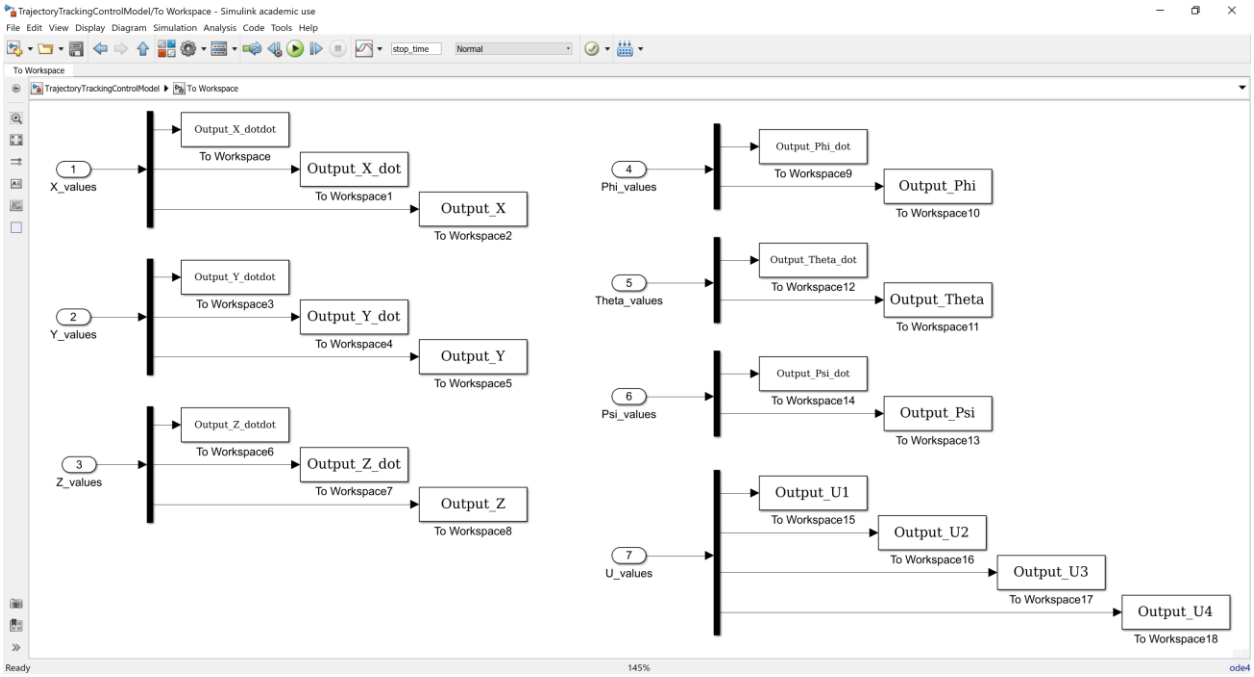
TrajectoryTrackingControlModel/Controller/U3 Control

```
function U3 = fcn(Kp_theta, Kd_theta, e_theta, e_theta_dot, Iyy)

U3 = Iyy*(Kp_theta * e_theta + Kd_theta * e_theta_dot);
```

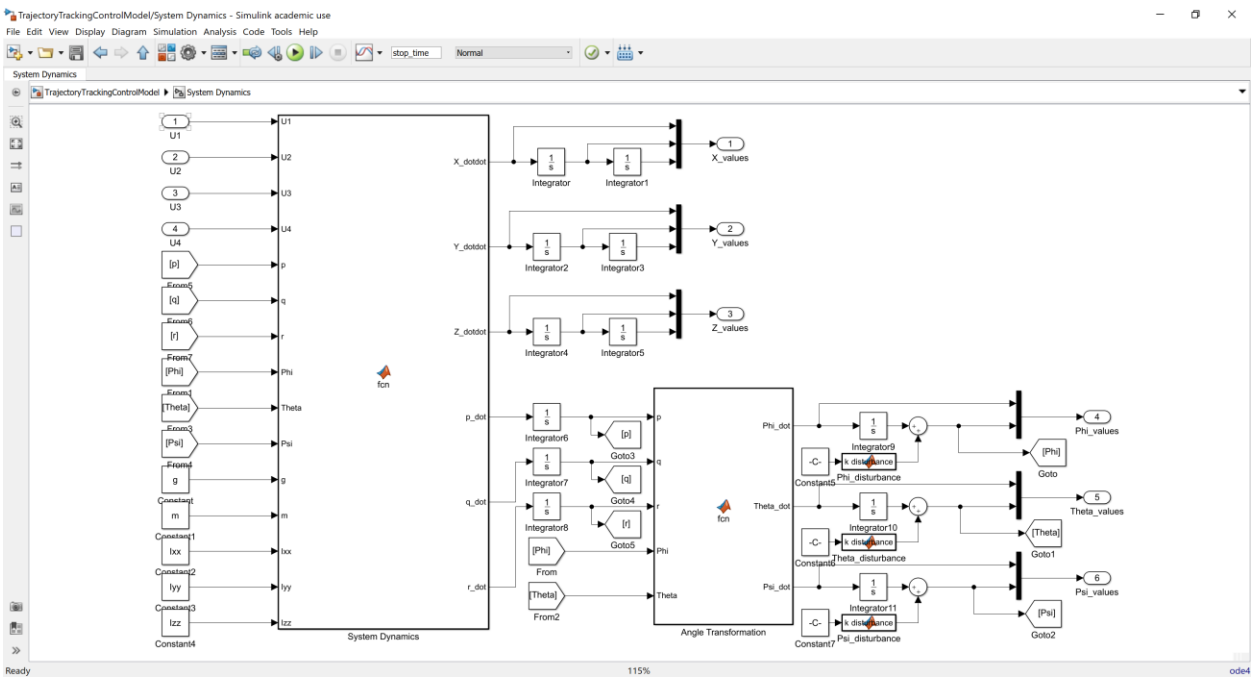
TrajectoryTrackingControlModel/Controller/U4 Control

```
function U4 = fcn(Kp_psi, Kd_psi, e_psi, e_psi_dot, Izz)

U4 = Izz*(Kp_psi * e_psi + Kd_psi * e_psi_dot);
```

TrajectoryTrackingControlModel/To Workspace



TrajectoryTrackingControlModel/System Dynamics



TrajectoryTrackingControlModel/System Dynamics/System Dynamics

```
function [X_dotdot, Y_dotdot, Z_dotdot, p_dot, q_dot, r_dot] =
fcn(U1, U2, U3, U4, p, q, r, Phi, Theta, Psi, g, m, Ixx, Iyy, Izz)

X_dotdot = (sin(Psi)*sin(Phi) + cos(Psi)*sin(Theta)*cos(Phi)) *
(U1/m);
Y_dotdot = (-cos(Psi)*sin(Phi) + sin(Psi)*sin(Theta)*cos(Phi)) *
(U1/m);
Z_dotdot = -g + (cos(Theta)*cos(Phi))*(U1/m);
p_dot = ((Iyy-Izz)/Ixx)*q*r + U2/Ixx;
q_dot = ((Izz-Ixx)/Iyy)*p*r + U3/Iyy;
r_dot = ((Ixx-Iyy)/Izz)*p*q + U4/Izz;
```

TrajectoryTrackingControlModel/System Dynamics/Angle Transformation

```
function [Phi_dot, Theta_dot, Psi_dot] = fcn(p, q, r, Phi, Theta)

T = [1, sin(Phi)*tan(Theta), cos(Phi)*tan(Theta);...
    0, cos(Phi), -sin(Phi);...
    0, sin(Phi)/cos(Theta), cos(Phi)/cos(Theta)];
y = T*[p; q; r];
Phi_dot = y(1);
Theta_dot = y(2);
Psi_dot = y(3);
```

TrajectoryTrackingControlModel/ System Dynamics/Phi_disturbance

```
function disturbance = fcn(k)

disturbance = (rand - 0.5) * k; %disturbance between -0.5*k and
+0.5*k rad
```

TrajectoryTrackingControlModel/ System Dynamics/Theta_disturbance

```
function disturbance = fcn(k)

disturbance = (rand - 0.5) * k; %disturbance between -0.5*k and
+0.5*k rad
```

TrajectoryTrackingControlModel/ System Dynamics/Psi_disturbance

```
function disturbance = fcn(k)

disturbance = (rand - 0.5) * k; %disturbance between -0.5*k and
+0.5*k rad
```