5-2017

Dynamic Model and Control of Quadrotor in the Presence of Uncertainties

Courage Agho
University of South Carolina

DYNAMIC MODEL AND CONTROL OF QUADROTOR IN THE PRESENCE OF UNCERTAINTIES

by

Courage Agho

Bachelor of Science
University of South Carolina, 2015

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2017

Accepted by:

Xiaofeng Wang, Director of Thesis

Bin Zhang, Reader

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

ACKNOWLEDGMENTS

I would like to thank Professor Xiaofeng Wang who is my adviser for giving me this opportunity to work on this thesis. I thank him for his support and time invested in my education. I also want to thank the department, my friends and family for their constant support.

ABSTRACT

This thesis considers the control of quadrotor using a linear PID control and $\mathcal{L}_1$ adaptive control. In a justifiable concept, PID controller can be used to control a quadrotor, but in the presence of uncertainties or disturbance, the quadrotor can't be automatically adjusted to control the changing dynamics of the quadrotor. To solve the problem associated with uncertainties, various control methodology can be used for controlling the changing dynamic of quadrotor, but in this thesis, $\mathcal{L}_1$ adaptive control is used because it allows for fast and robust adaptation for desired transient performance in the presence of matched and unmatched uncertainties.

In this thesis, we would derive the quadrotor model which gives us an access on how we can track positions, then design a controller to track these desired positions using PID control. Same concept used for PID control would be used for $\mathcal{L}_1$ adaptive control in chapter 5 except this control methodology is used for the cancellation of uncertainty at a faster and robust manner. $\mathcal{L}_1$ Adaptive control would use state feedback for its position control and this position control is then used to design the final controller to cancel uncertainties.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent year, research on quadrotors has become popular, but the first quadrotor was built in 1907 by Louis Charles Bréguet. Helicopter development was desired over quadrotor, but with recent progress in technological development, more funds and time has been dedicated to the research of quadrotors. Quadrotor became popular because helicopter uses tail rotors to counterbalance the torque or rotating forces generated by the single main rotor. Because of the counterbalancing tail rotor, it inefficient in term of control, power consumption and cost production. Due to the work of Charles Richet and Dr George de Bothezat in 1956, propellers where used to control the quadcopter roll, pitch and yaw angle. In addition to those improvements, technological advancement in batteries weight, life and density has greatly helped the research of quadrotors. Recent development in processors and low cost efficient sensors has greatly impacted quadrotor development as well.

Common application of quadrotors includes surveillance, inspections, military operation and transportation. It is very applicable because of its size; it is easier for it take off, to land and occupies less space. Because of its advantages, the demand for accurate control is needed for stabilized flight when hovering. The demand for accurate stabilized flight has led for the use of sensors and camera to achieve accurate result. Although PID controller is commonly used for control of quadrotors, however due to uncertainties and disturbance that occur during flight, PID controller is not a good mechanism for control.

This thesis focuses on creating an accurate control mechanism to achieve stabilized flight control using $\mathcal{L}_1$ adaptive control because it mitigates the issues of uncertainties and disturbance.

A quadrotor is a nonlinear system device and control for nonlinear system is complex because we desire a system that has fast adaptation and response in real time and robust enough to mitigate the issue of disturbance and uncertainties. Adaptive control was developed as a technique for automatic adjustment in real time. To achieve and maintain desired system performance, the aerospace industries and institution started research on adaptive controller and today it is used widely by different industries and for different purposed. Model reference adaptive control (MRAC) is a type of adaptive control and $\mathcal{L}_1$ adaptive control is a further use of model reference adaptive control (MRAC). Apart from canceling of uncertainties and disturbance, $\mathcal{L}_1$ adaptive control theory has architecture in place for faster adaptation that is decoupled from robustness. $\mathcal{L}_1$ Adaptive control doesn't losses robustness, because it is resolved by conventional classical control and $\mathcal{L}_1$ adaptive control system guarantees robustness in the presence of fast adaption with high gain feedback. This thesis includes the derivation of quadrotor model, and the model is then controlled by a PID controller and an $\mathcal{L}_1$ adaptive controller in the presence of uncertainty.

CHAPTER 2

QUADROTOR STRUCTURE

2.1 QUADCOPTER COORDINATE SYSTEM

A quad employs different control mechanism such as roll, pitch, and yaw which in most cases are represented by angle of rotation around the center of the quad craft. These angles make up for the control of the altitude of the quadcopter and to track the altitude of the quadcopter, a two-coordinate system is required. There is the body frame system which is attached to the quad at its center of gravity and the earth frame system which is fixed to the earth and it is sometimes refer to as an inertial coordinate system. The angular different between the two coordinate helps define the behavior of the quad altitude in space. The attitude system can be derived by rotating the body frame around the z axis of the earth frame by the yaw angle φ, which is tend followed by rotating around the y-axis by the pitch angle ϕ and finally by rotating around the x-axis by the roll angle θ. This is shown in figure 2.1 as well as its rotation matrix that has the body and earth frame parallel to each other and their sequence of rotation is known as the Z-Y-X rotation and its rotation matrix is shown in equation 2.1.

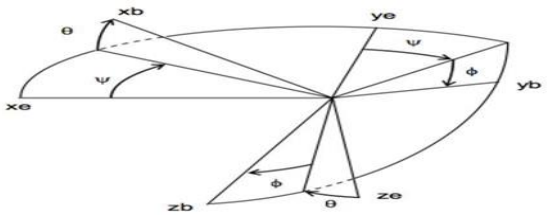


Figure 2.1: Body and Earth Frame with their Corresponding Angle

$$R = \begin{bmatrix} \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\theta)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\theta)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (2.1)$$

2.2 STATES OF THE QUADCOPTER

From the section of coordinate system, the angle of roll, pitch, and yaw are represented as ϕ, θ, ψ in addition to their angle, angular velocity is also required and can be represented as $\dot{\phi}, \dot{\theta}, \dot{\psi}$ these are the first six state of the quadcopter that shows a relationship between the quadcopter and the earth coordinate system. The next six states show a physical relationship of the physical location within the earth fixed system and it is denoted as X, Y, Z. In addition to their physical position is their quad velocity along these axes and it is denoted as $\dot{X}, \dot{Y}, \dot{Z}$. Together they make up the 12 states of the quadcopter and as shown below.

$$\mathrm{X} = [\boldsymbol{\phi} \quad \boldsymbol{\theta} \quad \boldsymbol{\psi} \quad \dot{\boldsymbol{\phi}} \quad \dot{\boldsymbol{\theta}} \quad \dot{\boldsymbol{\psi}} \quad \boldsymbol{X} \quad \boldsymbol{Y} \quad \boldsymbol{Z} \quad \dot{\boldsymbol{X}} \quad \dot{\boldsymbol{Y}} \quad \dot{\boldsymbol{Z}}]$$

2.3 HOW QUADROTOR WORKS

A quad as we know it has four motors and it is important to know that the thrust of each motor force a change around the pitch, roll, and yaw angle. To understand that, it is very important to know that a quadcopter is under actuated which means that the six degree of freedom $(\phi, \theta, \psi, X, Y, Z)$ is only controlled by the four inputs. Two of the six DOF are couple and they are the x-axis and y-axis on the translational size of the quadcopter. The translational part of the quadcopter is dependent on the attitude of the craft with respect to the other four degrees of freedom. More would be discussed on how to account and control the under actuated part of the quad when designing a control method for the quad. It is also important to know that a quadcopter has four motor where two of them spins in the clockwise direction while the order two spins in the counter clockwise direction and if the thrust generated by the motors is equal, the ability for the quadcopter to roll, hover pitch,

and yaw is possible. The diagram in figure 2.2 shows how the force generated by each motor affects a change the direction of the quadcopter along the angles on the coordinate system

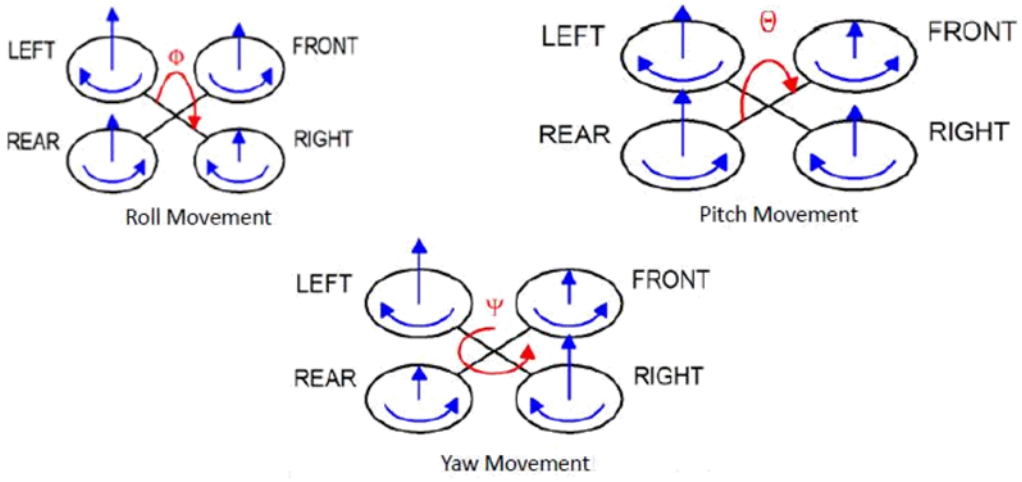


Figure 2.2: Movement around Angles on the Coordinates System because of Force

CHAPTER 3

DYNAMIC MODEL

The dynamic model of a quadcopter is a subsystem that is divided into two subsystems known as the rotational subsystem that represent (roll angle, pitch angle and yaw angle) and translational subsystem that represent (Z position, X position, Y position). The rotational side of the Quadcopter is completely actuated while the translational side of the subsystem is under actuated

3.1 MOMENT ACTION ON THE QUADROTOR

An effect of rotation is the force generated called aerodynamic force and moment generated called aerodynamic moment. The aerodynamic moment is the combination of aerodynamic force multiplied by its distance. It is dependent on the geometry of the propeller and by identifying the moment and force generated by the propeller, we can understand the moment acting on the quadcopter. From figure 3.1, when F2 is multiplied by the moment arm, a negative moment is generated about the y-axis. Using the same concept, F4 generates a positive moment and the total moment about the x-axis can be expresses as

$$M_X = (F4 - F2)l \tag{3.1}$$

Same concept is applied for My Where

$$M_y = (F1 - F3)l \tag{3.2}$$

For the moment about the Z axis, the thrust of the rotor does not cause a moment, but rather the rotor rotation in relation with the rotor speed causes a moment and is represented as

$$M_z = (F1 + F3 - F2 - F4)lc \tag{3.3}$$

Where c gives the relationship between the rotor speed and its effect on the quadrotor rotation about the body frame and the combination of each body frame axis gives us a matrix as shown below.

$$M_b = \begin{bmatrix} (F4 - F2)l \\ (F1 - F3)l \\ (F1 + F3 - F2 - F4)lc \end{bmatrix} \tag{3.4}$$

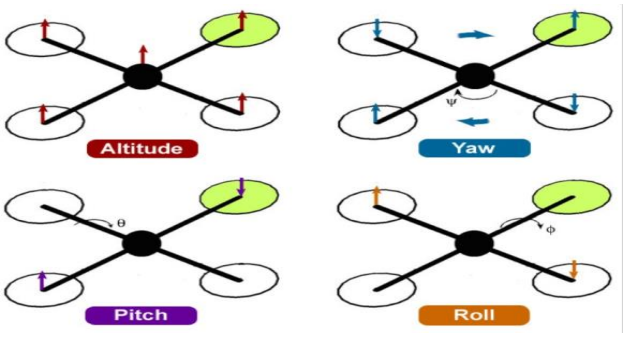


Figure 3.1: Moment acting on the Quadcopter

3.2 INERTIAL MATRIX

The inertial matrix for a quadcopter is a diagonal matrix as shown below. The structure of the matrix is because quadcopters are built symmetrically with respect to the coordinate systems that were explained in section 2.1.

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{3.5}$$

From the matrix above J_x, J_y and J_z are the area moments of inertia about the principle axes on the body frame.

3.3 CONTROL INPUT VECTOR U

The control input is from the controller and the input to a quadcopter is from the force generation by the motor, and for simplicity, Control Input Vector (U) can represent Force (F) where

$$U_{tz} = (F1 + F2 + F3 + F4) \tag{3.6}$$

$$U_{tx} = (F4 - F2) \tag{3.7}$$

$$U_{ty} = (F1 - F3) \tag{3.8}$$

$$U_{rz} = (F1 + F3 - F2 - F4) \tag{3.9}$$

From equation 3.4, the moment acting on the quadcopter in its body frame can be represented as

$$\begin{bmatrix} U_{tx}l \\ U_{ty}l \\ U_{rz}lc \end{bmatrix} \tag{3.10}$$

3.4 ROTATIONAL SUBSYSTEM

The rotational part of the quadcopter is derived from the concept of rotational equation of motion and by using Newton-Euler method derived from the body frame of the quadcopter with a generalized formula as shown below.

$$M_b = \left(J\dot{w} + w \times jw + M_g\right) \tag{3.11}$$

Where: J represents quadrotor inertia Matrix, w represents angular velocity, $M_g$represents the gyroscopic moment generated due to its rotor inertial and M_b represents moments acting on the quadcopter in its body frame. For simplicity, the gyroscopic moment would not be considered because the inertial generated by the quadcopter is much larger that the inertial generated by the rotor. So, our final rotor equation would be

$$M_b = (J\dot{w} + w \times jw) \tag{3.12}$$

From equation 3.12, we would have the expression below

$$\begin{bmatrix} U_{tx}l \\ U_{ty}l \\ U_{rz}lc \end{bmatrix} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

When the matrix is rewritten to have its angular acceleration, we would have

$$\ddot{\phi} = \frac{l}{J_x} U_{tx} + \frac{J_y}{J_x} \dot{\psi}\dot{\theta} - \frac{J_z}{J_x} \dot{\theta}\dot{\psi} \tag{3.13}$$

$$\ddot{\theta} = \frac{l}{J_y} U_{ty} + \frac{J_z}{J_y} \dot{\psi}\dot{\phi} - \frac{J_x}{J_y} \dot{\phi}\dot{\psi} \tag{3.14}$$

$$\ddot{\psi} = \frac{lc}{J_z} U_{rz} + \frac{J_x}{J_z} \dot{\theta}\dot{\phi} - \frac{J_y}{J_z} \dot{\phi}\dot{\theta} \tag{3.15}$$

3.5 TRANSLATION SUBSYSTEM

The translation subsystem is based on translational equation of motion and it is based on newton second law which is derived from the earth inertial frame and it is presented in the format below.

$$m\ddot{d} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R \times F_{ng} \tag{3.16}$$

Where: m is the mass, g is gravitational acceleration, F_{ng} is non-gravitational force which is the physical location within the earth fixed system and R is the rotational matrix. F_{ng} is shown in equation 3.17. It is the addition of all the thrust force produced by the four propellers and the negative sign is because the thrust force generated is acting upward while the z-axis of the body frame is point down. R is for the rotational matrix that is generated to transform the forces generated from the body frame to the earth frame. By substitution and simplification, we would derive the acceleration along the X, Y, and Z axis.

$$F_{ng} = \begin{bmatrix} 0 \\ 0 \\ -U_{tz} \end{bmatrix} \tag{3.17}$$

By substitution of these equations into equation 3.16, we would get

$$m\begin{bmatrix}\ddot{X}\\ \ddot{Y}\\ \ddot{Z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ mg\end{bmatrix} + \begin{bmatrix}\cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\cos\psi & -\cos(\phi)\sin(\theta)\cos(\psi)+\sin(\phi)\sin(\psi)\\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\varphi)+\cos(\theta)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi)-\sin(\theta)\cos(\psi)\\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta)\end{bmatrix}\times\begin{bmatrix}0\\ 0\\ -U_{tz}\end{bmatrix}$$

When the matrix is rewritten to have its acceleration, we would have

$$\ddot{X} = \frac{-U_{tz}}{m}(-\cos(\phi)\sin(\theta)\cos(\psi)+\sin(\phi)\sin(\psi)) \tag{3.18}$$

$$\ddot{Y} = \frac{-U_{tz}}{m}(\cos(\phi)\sin(\theta)\sin(\psi)+\sin(\theta)\cos(\psi)) \tag{3.19}$$

$$\ddot{Z} = \frac{-U_{tz}}{m}(\cos(\phi)\cos(\theta))-g \tag{3.20}$$

The quadrotor parameter are defined below and these parameters would be use for simulation

$m = 0.8kg; l = 0.25m; g = 9.81\frac{N}{kg}; c = 0.02; J_x = J_y = 0.015kgm^2\ ; J_z = 0.02kgm^2$

CHAPTER 4

CONTROL OF THE QUADROTOR

Building a control system to control how the quadcopter operate is very important. The dynamic model of the quadcopter is an open loop and closed loop control is a preferred method for any design because the system records the output instead of the input and modifies its output per a preferred condition. For successful maneuvering of the quadcopter, we need 4 controllers that would be designed to act as in input to the model of the quadcopter. These 4 controllers represent the 4-input coming from the transmitter assuming this was a physical quadcopter. They are throttle, roll, pitch, and yaw input. The throttle coming from the transmitter can be represented and called the altitude controller or translational controller in the z axis. The roll, pitch, and yaw input from the transmitter can be called rotational controller that is dependent on the angle ϕ, θ, φ. The translational subsystem of the model is partially dependent on the rotational subsystem because it is under-actuated.

The dependent axes on the translational controller are the X, Y axis. So, a controller must be built that takes in desired X, Y values and produce an angular desired output that would be sent to rotational controller. For the control of the quadrotor using PID controller, the dynamic model of the system would be linearized using small angle approximation. The equation below represents the linear dynamic model of the quadrotor and an in-depth concept on how the model is linearized is explained in chapter 5.

$$\ddot{Z} = \frac{1}{m}(-U_{tz}) \tag{4.1}$$

$$\ddot{X} = -\frac{1}{m}\theta(U_{tz}) \tag{4.2}$$

$$\ddot{Y} = \frac{1}{m}\phi(U_{tz}) \tag{4.3}$$

$$\ddot{\psi} = \frac{l}{J_z}U_{rz} \tag{4.4}$$

$$\ddot{\theta} = \frac{l}{J_y}U_{tx} \tag{4.5}$$

$$\ddot{\phi} = \frac{l}{J_X}U_{ty} \tag{4.6}$$

4.1 TRANSLATION CONTROL ALONG THE Z AXIS

The translational controller takes an error signal as an input which is the difference between the actual altitude and desired altitude and produces a control signal U_{tz}. The control signal U_{tz} is responsible for the altitude of the quadcopter and mathematical equation is shown below. The equation below uses the concept of a PID controller and for simulation of result, we would multiply the input data by (-1) to compensate for the negative sign in equation 4.1 and the controller response to the closed loop simulation is shown in figure 4.1.

$$U_{tz} = Kp(z - zd) + Ki\int(z - zd) + Kd\frac{dz}{dt}(z - zd) \tag{4.7}$$

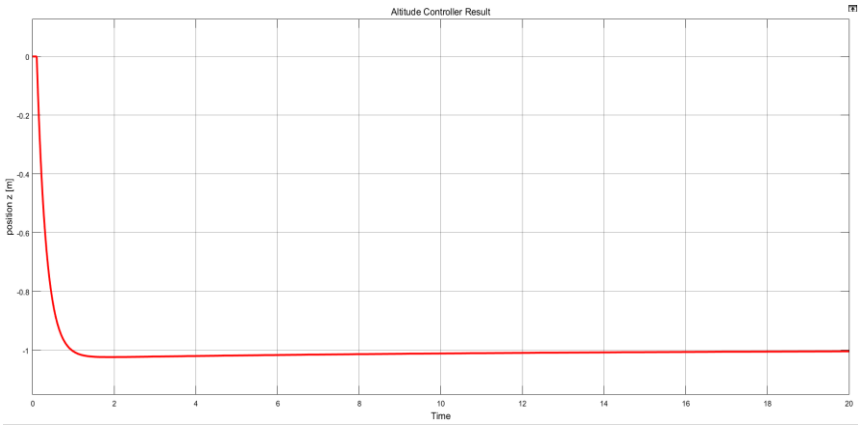


Figure 4.1: Altitude Position Tracking Using PID Controller [Z-Axis]

4.2 TRANSLATION CONTROL ALONG THE X AXIS

The translational controller takes an error signal as an input which is the difference between the actual roll angle and the desired roll angle to produce a control signal U_{tx}. The control signal U_{tx} is responsible for moving the quadcopter left and right about the X axis. The mathematical equation is shown below. The equation below uses the concept of a PID controller and because translation along the X axis is dependent on θ, control of its desired position is not under a direct control.

$$U_{tx} = Kp(x - xd) + Ki\int(x - xd) + Kd\frac{dz}{dt}(x - xd) \tag{4.8}$$

This is because the X positions are under the translational subsystem which is under-actuated. To compensate for the under actuation of the system, the pitch angle can be used to control desired X position.

We would assume $U_{tz} = mg$ which would cause loss in precision, but it also simplifies the model for easier control. Equation 4.2 is then simplified as

$$\ddot{X} = -\theta g \tag{4.9}$$

The rotational control provides us θ which can then be used for translational control. For simulation of result, we would multiply the input data by (-1) to compensate for the

negative sign in equation 4.2 and the controller response to the closed loop simulation is shown in figure 4.5.

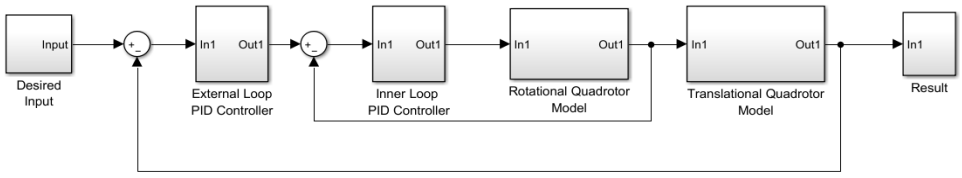


Figure 4.2: Block Diagram for Translational Control

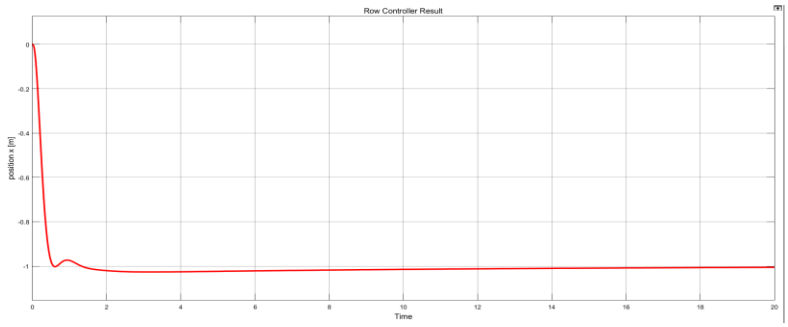


Figure 4.3: Roll Position Tracking Using PID Controller [X-Axis]

4.3 TRANSLATION CONTROL ALONG THE Y AXIS

$$\ddot{Y} = \phi g \tag{4.10}$$

The design for controller movement along the Y axis is design in similar method for translation along the X axis. The only difference is the Y axis is dependent on the roll angle.

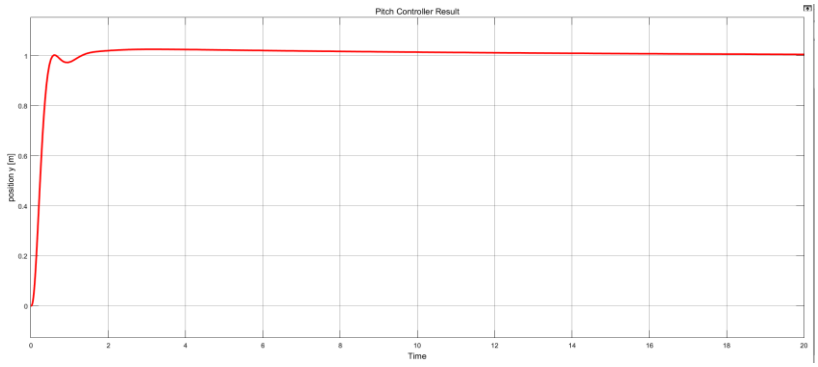


Figure 4.4: Pitch Position Tracking Using PID Controller [Y-Axis]

4.4 ROTATION CONTROL ABOUT Z-AXIS

The yaw controller takes an error signal as an input which is the difference between the actual yaw angle and the desired yaw angle to produce a control signal U_{ry} and the mathematical equation is shown below. The equation below uses the concept of a PID controller and the block diagram represent how the close loop method of the yaw controller would look like.

$$U_{ry} = Kp(\psi - \psi d) + Ki \int (\psi - \psi d) + Kd \frac{dz}{dt}(\psi - \psi d) \tag{4.11}$$

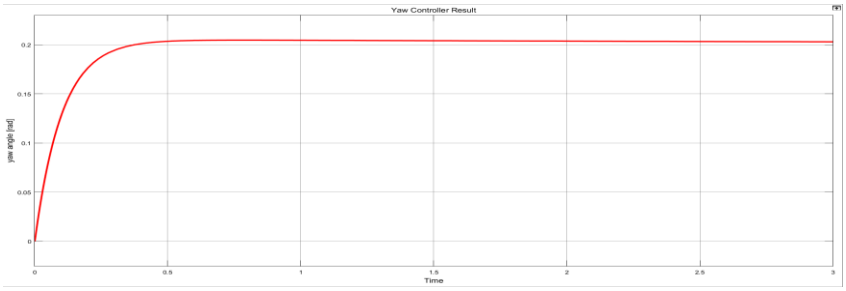


Figure 4.5: Yaw Angle Tracking Using PID Controller [Z-Axis]

From the design concept above, 6 PID controllers were designed with Kp, Ki, and Kd gain derived. Where the controller design for translational control on the X and Y axis would have two PID controller and their gain constants are shown in figure 4.6.

	Kp (gain)	Ki (gain)	Kd (gain)
Translation Control along the Z Axis	0.4001	0.01019	3.491
Translation Control along the Y Axis	0.0366	0.0008775	0.3392
Translation Control along the X Axis	0.0366	0.0008775	0.3392
Rotation Control along the Z Axis	0.21	0.01225	0.7997
Rotation Control along the Y Axis	1.553	0.8375	0.5677
Rotation Control along the X Axis	1.553	0.8375	0.5677

Figure 4.6: Controller Gain Constant

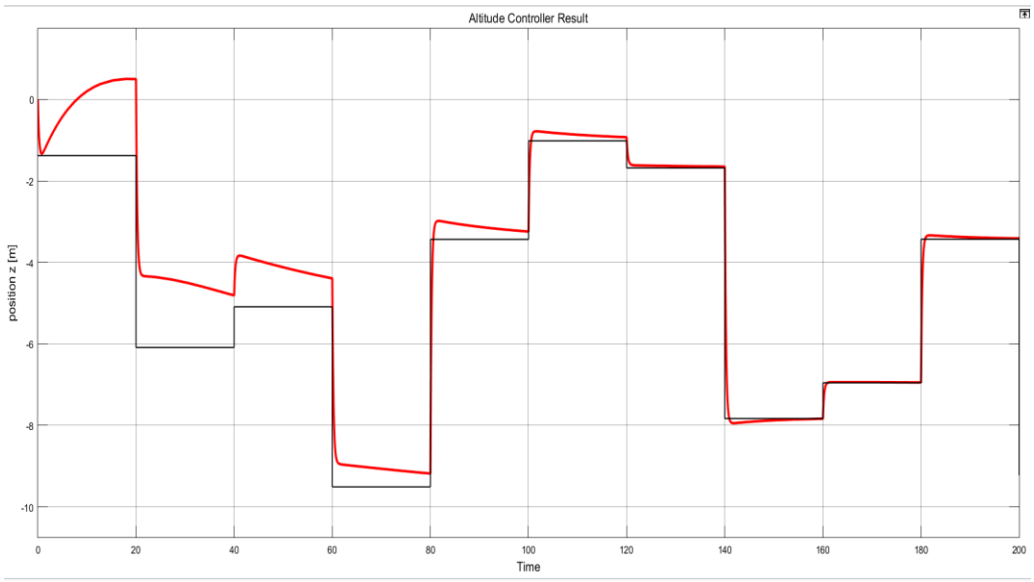


Figure 4.7: Altitude Response to Step Input Disturbance

CHAPTER 5

$\mathcal{L}_1$ ADAPTIVE CONTROL

As you can see from section 4, using a conventional linear controller work fine when there is no disturbance or uncertainty, but when disturbance is added to the system as shown in figure 4.10, the controller is not able to adapt fast enough to compensate for the additional error. The process of creating an adaptation mechanism leads to further insight to the concept of adaptive controller.

Adaptive controller is a control mechanisms used by a controller to control a system with varying parameters or uncertainties. It is a technique used for the automatic adjustment of the controller in real time. Various controllers can be used to cancel uncertainties; the most common one is the robust controller. The concept of adaptive controller to robust controller is the control law for adaptive controller change while the control law for robust controller doesn't change. Adaptive controllers are the combination of online parameter estimators and automatic control design which has two architecture designs. There is direct method which only estimates the controller parameters and there is indirect method which only estimates the state parameter of system. The structure of this architecture is shown below. From the structures, the estimate is calculated using adaptive law where Γ is the adaptive gain.

5.1 DIRECT & INDIRECT ADAPTIVE CONTROLLER

The architecture is shown in figure 5.1 and the differential equation of the real plant is

$$\dot{x}(t) = A_m x(t) + b(u(t) + k_x^T x(t), \quad x(0) = x_0 \tag{5.1}$$

$$y(t) = c^T x(t) \tag{5.2}$$

Where A_m defines matrix of the closed loop system, x(t) is the state of the system, b and c are known constant; k_x is the vector of the unknown constant.

$$u_{norm}(t) = -k_x^T x(t) + k_g r(t) \tag{5.3}$$

$$k_g \triangleq \frac{1}{c^T A_m^{-1} b} \tag{5.4}$$

Ideal system

$$\dot{x}_m(t) = A_m x(t) + b k_g r(t),\ x_m(0) = x_0 \tag{5.5}$$

$$y_m(t) = c^T x(t) \tag{5.6}$$

$$e(t) \triangleq x_m(t) - x \tag{5.7}$$

Adaptation Law becomes

$$\dot{\hat{k}}_x = -\Gamma x(t) \tilde{e}^T(t) P b, \qquad \hat{k}_x(0) = k_{x0} \tag{5.8}$$

And P solves Lyapunov equation

$$A_m^T P + P A_m = -Q \tag{5.9}$$

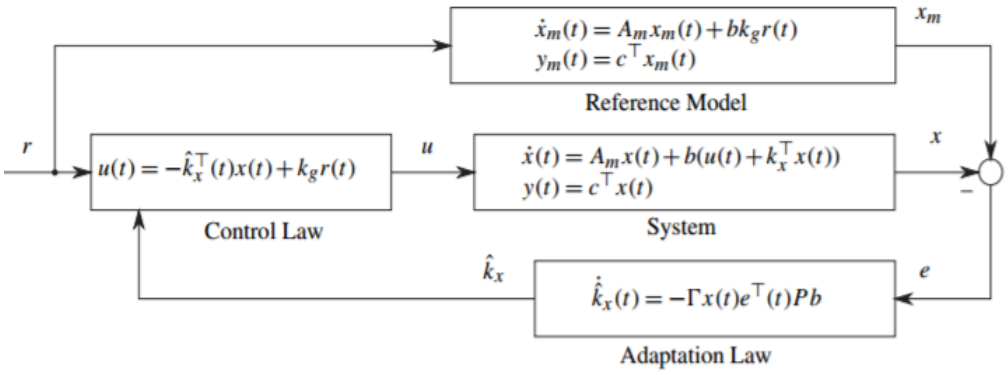


Figure 5.1: Closed Loop Architecture for Direct MRAC

For stability test and analysis, Lyapunov function is used to test Lyapunov stability. By taking the first derivative of Lyapunov function, the signal stays bounded and the second derivative proves that the error converges when $e(t) \longrightarrow 0\ as\ t \longrightarrow \infty$ which in turns proves the application of Barbalat's lemma equation on stability of time varying system.

The theoretical approach used for direct MRAC is used as well for Indirect MRAC. The main difference is that the indirect method estimates the system parameters and the derivation of the adaptive law is independent of the control signal with its system architecture as shown in figure 5.2. The asymptotical convergence of its tracking error is also concluded using Barbalat's lemma equation from Lyapunov stability and its adaptive law becomes

$$\hat{k}_x = \Gamma x(t)\tilde{x}^T(t)Pb, \qquad \hat{k}_x(0) = k_{x0} \tag{5.10}$$

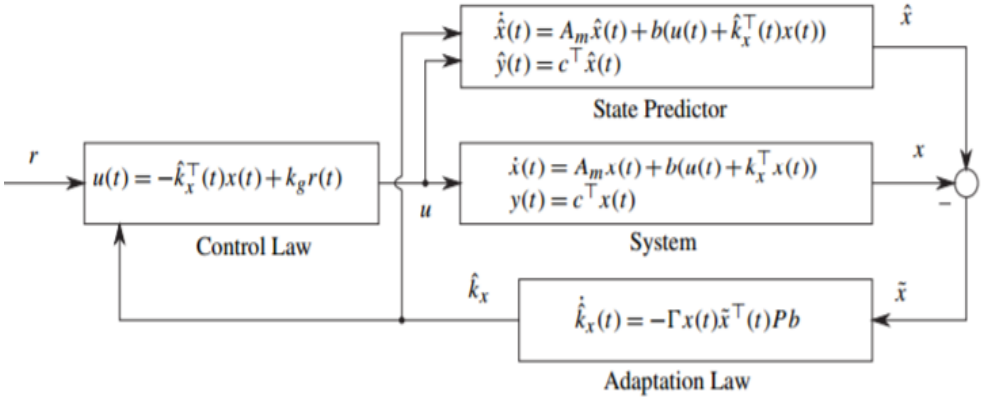


Figure 5.2: Closed Loop Architecture for Indirect MRAC

5.2 $\mathcal{L}_1$ ADAPTIVE CONTROLLER ARCHITECTURE

The architecture of both control method works, but with an increase in Γ, estimate is reduced as t ≥ 0, but high adaptation gain leads to a high frequency oscillation. So, increasing Γ, does allow a faster adaptation, but hurt robustness and stability because of the frequency oscillation. To have a faster adaptation and maintain robustness, a controller

known as the $\mathcal{L}_1$ adaptive controller is used. With $\mathcal{L}_1$ adaptive controller, adaptation can be separated from robustness. The architecture in figure 5.3 shows how $\mathcal{L}1$ adaptive controller is designed. The design approach is a combination of the MRAC state predictor and a low pass filter from the control input to the estimated model and the real plant. The controller compared to indirect MRAC is giving as

$$u(s) = C(s)n(s) \tag{5.11}$$

C(s) transfer function is bounded input-bounded output stable which is subjected to C(s)=1 with zero initialization. This is achievable with a first order low pass filter where

$$C(s) = \frac{\omega c}{s+\omega c} \tag{5.12}$$

and ωc is the filter bandwidth, the filter should be chosen to maintain equation 5.13.

$$\|G(s)\|_{\mathcal{L}1} < \frac{1}{L} \tag{5.13}$$

Where

$$G(s) = H(s)[1 - C(s)], H(s) = \frac{b}{s + a_m}[1 - C(s)] \tag{5.14}$$

$$L \triangleq max\|\theta\|_1, \qquad \theta \in \Theta \subset \mathbb{R}^n \tag{5.15}$$

With low pass filter added to the system, robustness can be maintained with increased adaptation gain. The adaptation law for $\mathcal{L}_1$ adaptive controller is giving by $\Gamma\text{Proj}(\hat{\theta}(t), -\tilde{x}^T(t)PBx(t))$ where the Proj operator ensures that the unknown parameter $\hat{\theta}$ stays bounded. θ would be replaced by σ for the quadrotor controller and the adaptive for $\mathcal{L}1$ adaptive controller is giving by $\Gamma\text{Proj}(\hat{\sigma}(t), -\tilde{x}^T(t)PBx(t))$.

For projection operator with a given compact set$\{\sigma \in \mathbb{R} \mid \|\sigma\| \leq c\}$,

$$\text{Proj}(\hat{\sigma}, \hat{\sigma}') = \begin{cases} \hat{\sigma}'\big(1 - h(\hat{\sigma})\big) & if\ (h(\hat{\sigma}) > 0 \quad \& \quad \hat{\sigma}'h(\hat{\sigma}) > 0 \\ \hat{\sigma}' & otherwise \end{cases} \tag{5.16}$$

Where

$$h(\hat{\sigma}) = \frac{\hat{\sigma}^2 - \hat{\sigma}_{max}^2}{\varepsilon_a \hat{\sigma}_{max}^2} \tag{5.17}$$

$$\hat{\sigma}_{max} = \frac{c}{\sqrt{1+\varepsilon_a}}, \qquad \varepsilon_a \in (0,1), \qquad 0 \leq c \leq 1 \tag{5.18}$$

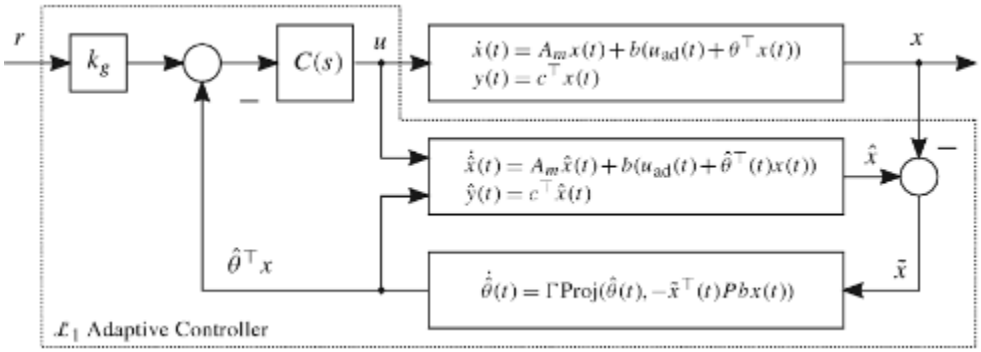


Figure 5.3: Closed Loop Architecture for $\mathcal{L}_1$ Adaptive controller

5.3 CONTROLLER DESIGN WITH $\mathcal{L}_1$ ADAPTIVE CONTROLLER

Just like is session 4, we would design four controllers to control the movement of the quadrotor using a state feedback approach. The information derived from the feedback approach would then be used to design an $\mathcal{L}_1$ adaptive controller to cancel out uncertainties.

5.4 TRANSLATION CONTROL ALONG THE Z-AXIS

From translation about the z axis,

$$\ddot{z} = \frac{1}{m}(\cos(\phi)\cos(\theta))(-U_{tz}) + g \tag{5.19}$$

$$\ddot{z} - g = \frac{1}{m}(\cos(\phi)\cos(\theta))(-U_{tz}) \tag{5.20}$$

U_{tz} is the addition of the position control output and the control output for the cancelation of gravity?

$$U_{tz} = \ U_{zp} + U_{zg} \tag{5.21}$$

Since the quadrotor is moving about the z axis, θ and ϕ would be equal to zero. So $\cos(\phi)\cos(\theta)$ would be equal to 1. Therefore

$$\ddot{z} - g = \frac{1}{m}(1)\left(-\left(U_{zp} + U_{zg}\right)\right) \tag{5.22}$$

From equation 5.13, gravity would be ignored and later compensated where U_{zg} is equal to mg. when it is ignored,

$$\ddot{z} = \frac{1}{m}\left(U_{zp}\right) \tag{5.23}$$

State space equation

$$\dot{x} = A_{zp}x + B_{zp}U_{zp}$$

$$y = C_{zp}^{T}x$$

$$\begin{bmatrix}\dot{z}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}z\\ \dot{z}\end{bmatrix} + \begin{bmatrix}0\\ 1/m\end{bmatrix}u; \qquad y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}z\\ \dot{z}\end{bmatrix} \tag{5.24}$$

With state feedback controller

$$\dot{x} = A_{zp}x + B_{zp}U_{zp}$$

$$\dot{x} = A_{zp}x + B_{zp}\left(kg_r - km_{zp}^{T}x\right) \tag{5.25}$$

where $\mathrm{kg_r}$ is the reference gain and $\mathrm{km_z^T} = [\mathrm{k1_{tz}} \quad \mathrm{k2_{tz}}]^{\mathrm{T}}$ which is the feedback gain.

Therefore

$$\dot{x} = (A_{zp} - B_{zp}km_{zp}^{T})x + B_{p}kg_r \tag{5.26}$$

Where $A_{m_{zp}} = (A_P - B_pkm_{zp}^{T})$ So, the state spaces become

$$\dot{x} = A_{m,_{zp}}x + B_pkg_r$$

This changes the state space model to

$$\begin{bmatrix}\dot{z}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -k1_z(\frac{1}{m}) & -k2_z(\frac{1}{m})\end{bmatrix}\begin{bmatrix}z\\ \dot{z}\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{m}\end{bmatrix}kg_r \tag{5.27}$$

Transfer function of the state space model becomes

$$G_p(s) = \frac{\frac{1}{m}}{s^2 + k2_z\left(\frac{1}{m}\right)s + k1_z\left(\frac{1}{m}\right)} \tag{5.28}$$

Compare to a reference transfer function

$$G_{ref}(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} \tag{5.29}$$

From equation 5.23, ${\omega_n}^2 = 1/m$ and for critical damping, $\zeta = 1$. $k1_{tz} = 1$ and $k2_{tz} = 2\zeta\omega_n$

$$k2_{tz} = 2\sqrt{m} \tag{5.30}$$

Therefore $k2_{tz} = 1.79$, where m=0.8. U_{tz} the controller output would be determined as

$$U_{tz} = (kg_r + k1_z z + k2_z \dot{z} + mg) \tag{5.31}$$

Figure 5.5 is the simulation result tracking desired reference position and figure 5.6 is the Simulink design for control using state feedback.

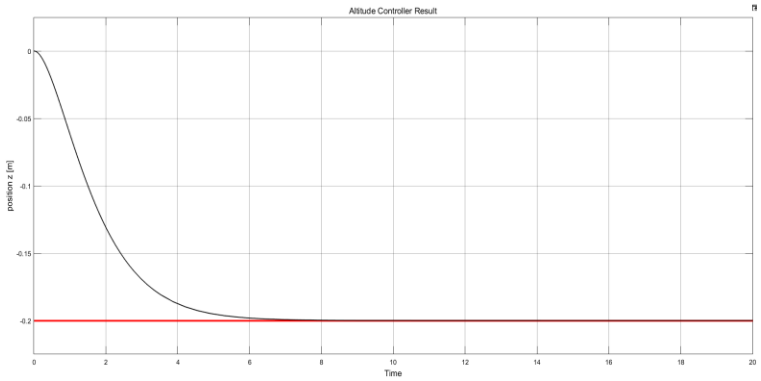


Figure 5.5: Altitude Position Using State Feedback Control [Z-Axis]

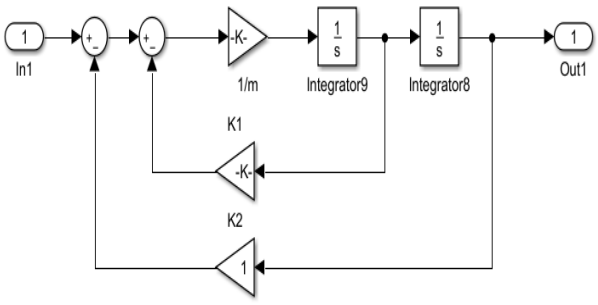


Figure 5.6: State feedback for Altitude Control

With the feedback controller designed to ensure stable flight, we would then design a controller to cancel out uncertainties which is dependent on the adaptation law for uncertainty prediction. From equation 5.27, the state space model of the closed loop system is defined below.

$$A_{m,zp} = \begin{bmatrix} 0 & 1 \\ -k1_z(\frac{1}{m}) & -k2_z(\frac{1}{m}) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1.25 & -2.24 \end{bmatrix} \tag{5.32}$$

$$B_p = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \tag{5.33}$$

For the adaptive law, we first need to determine the P matrix that will satisfy Lyapunov equation $A_m^T P + P A_m = -Q$. In order to achieve that, we would set Q as an identity matrix as shown below.

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.34}$$

$$P = \begin{bmatrix} 1.30 & 0.50 \\ 0.50 & 0.50 \end{bmatrix} \tag{5.35}$$

Adaptation control law

$$\dot{\hat{\theta}} = \Gamma \text{Proj}\left(\hat{\sigma}(t), -\tilde{x}^T(t) P B x(t)\right) \tag{5.36}$$

$$\dot{\hat{\theta}} = \Gamma \text{Proj}\left(\hat{\sigma}(t), -\begin{bmatrix} \tilde{z} \\ \dot{\tilde{z}} \end{bmatrix} \begin{bmatrix} 1.30 & 0.50 \\ 0.50 & 0.50 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}\right) \tag{5.37}$$

$$\dot{\hat{\theta}} = \Gamma \text{Proj}\left(\hat{\sigma}(t), -\left(0.63\tilde{z} + 0.63\dot{\tilde{z}}\right)\right) \tag{5.38}$$

$$G_{\mathcal{L}1,tz} = \left[\left[sI - A_{m,p}\right]^{-1} B_p\right]\left[1 - C_{t,z}(s)\right] \tag{5.39}$$

Using matlab, the transfer function is where ωc=100

$$G_{\mathcal{L}1,tz} = \begin{bmatrix} \dfrac{1.25s}{s^3 + 102.24s^2 + 225.25s + 125} \\ \dfrac{1.25s^2}{s^3 + 102.24s^2 + 225.25s + 125} \end{bmatrix} \tag{5.40}$$

$\left\|G_{\mathcal{L}1,tz}\right\|_{\mathcal{L}1} = 0.003$ which give us an uncertainty limit of 333.33N. with that, the adaptation gain can be big enough for faster adaptation. Also from equation 5.38, $\tilde{z} = \hat{z} - z$ and $\dot{\tilde{z}} = \dot{\hat{z}} - \dot{z}$. For simulation, a varying uncertainty of -10N to 10N would be used with an adaptation gain of 10000. Figure 5.7 is the simulation result tracking desired reference position using $\mathcal{L}_1$ adaptive controller. Figure 5.8 shows the structural design of the altitude controller in Simulink and figure 5.9 shows the simulation of the estimated uncertainty vs the real uncertainty.

Another external force that could affect quadrotor movement in the Z axis is the input of extra mass. Theoretically, there is no limit on the amount of extra mass that a quadrotor can lift, but with a physical system, the amount of thrust and rotor speeds are limited. Most physical quadrotor are designed so that the sum of all four motors can lift at least twice its original mass. Assuming our simulation is designed for that purpose, $m_{extra} \leq 0.8$. Therefore equation (5.23) can be rewritten as $\ddot{z} = \frac{1}{m+m_{extra}}(U_{zp})$. Same position controller input can be used to control the new rewritten model because m_{extra} is constant and the controller output is well bounded. Extra mass can also affect movement in the X and Y axis and same concept would be applied to compensate with the input of extra mass.

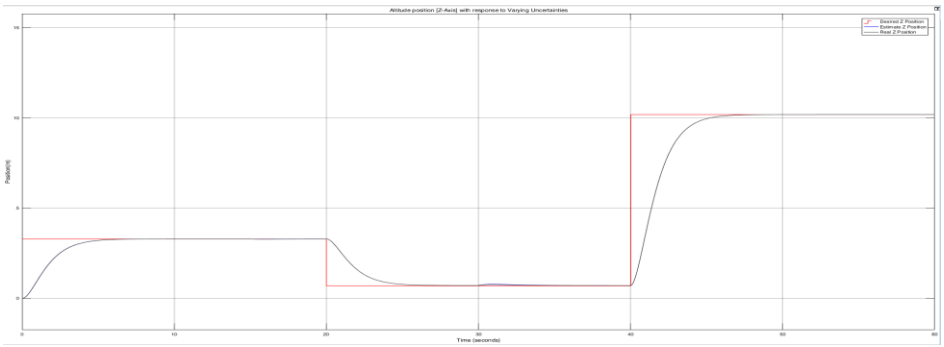
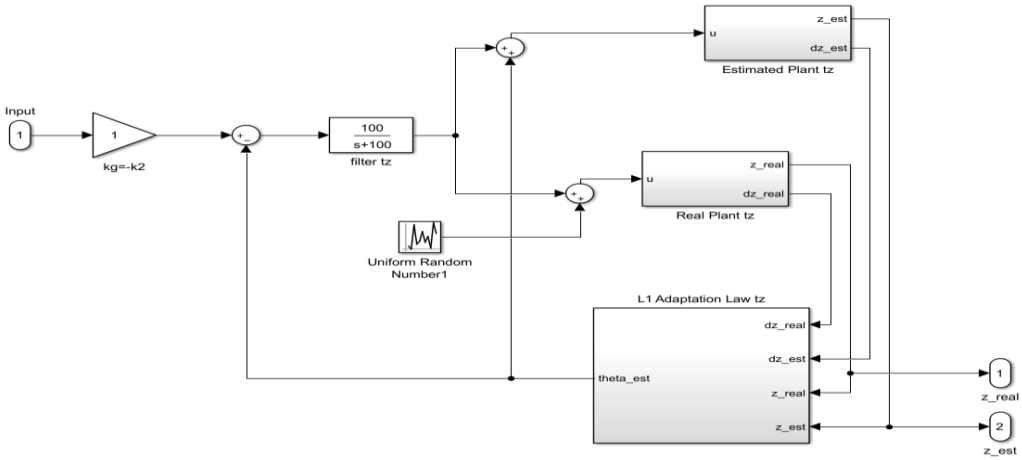
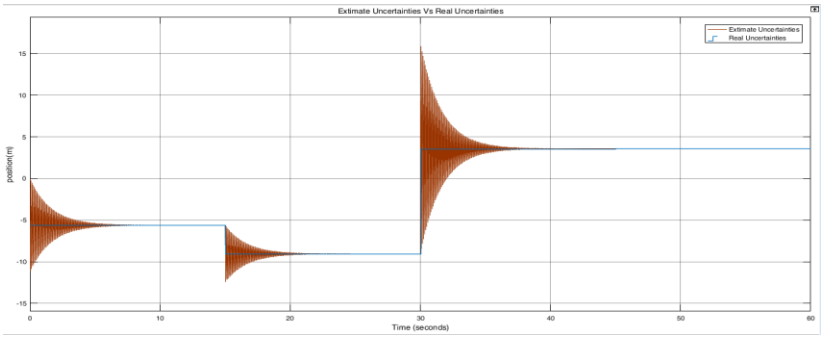


Figure 5.7: Altitude position [Z-Axis] with response to Varying Uncertainties and Extra Mass



5.8: Structural Design of the Altitude Controller



5.9: Estimate Uncertainty vs Real Uncertainty for Altitude Control

5.5 ROTATION CONTROL ABOUT THE Z-AXIS

This is the yaw controller. The aim of the controller is to design a stable yaw rotation controller to cancel uncertainties. From equation 5.4, using small angle approximation

$$\ddot{\psi} = \frac{l}{I_{zz}} U_{rz} \tag{5.41}$$

State space feedback is

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k1_{rz}\left(\frac{lc}{J_z}\right) & -k2_{rz}\left(\frac{lc}{J_z}\right) \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ lc/J_z \end{bmatrix} r_{rz} \tag{5.42}$$

$$y = [1 \;\; 0] \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} \tag{5.43}$$

Transfer function is

$$G_{r,z}(s) = \frac{\frac{lc}{J_z}}{s^2 + k2_{r,z}\left(\frac{lc}{J_z}\right)s + k1_{r,z}\left(\frac{lc}{J_z}\right)} \tag{5.44}$$

From equation 5.23, ${\omega_n}^2 = \frac{lc}{J_z}$ and for critical damping, $\zeta = 1$. $k1_{r,z} = 1$ and $k2_{r,z} = 2\zeta\omega_n$

$$k2_{t,x} = 2\sqrt{\frac{lc}{J_z}} \tag{5.45}$$

$k2_{r,z} = 4 \; where \; l = 0.25m, \; c = 0.02, \; J_z = 0.02kgm^2$

$$U_{rz} = r_{rz} - k1_{r,z}\psi - k2_{r,z}\dot{\psi} \tag{5.46}$$

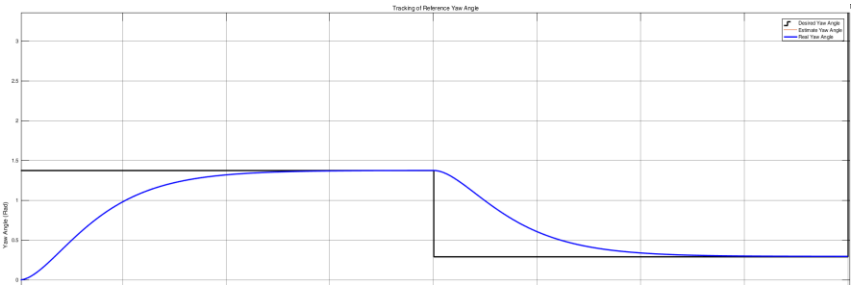


Figure 5.10: Yaw Angle Tracking Using State Feedback Control

Where

$$A_m = \begin{bmatrix} 0 & 1 \\ -k1_z\left(\frac{1}{m}\right) & -k2_z\left(\frac{1}{m}\right) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.25 & -1 \end{bmatrix} \tag{5.47}$$

$$B_m = \begin{bmatrix} 0 \\ \frac{lc}{J_z} \end{bmatrix} \tag{5.48}$$

For the adaptive law, we first need to determine the P matrix that will satisfy Lyapunov equation $A_m^T P + P A_m = -Q$.

The P matrix to satisfy Lyapunov equation is

$$P = \begin{bmatrix} 4.50 & 0.50 \\ 0.50 & 0.625 \end{bmatrix} \tag{5.49}$$

Adaptation control law

$$\dot{\hat{\theta}} = \Gamma\text{Proj}\left(\hat{\sigma}(t), -\tilde{x}^T(t) P B x(t)\right) \tag{5.50}$$

$$\dot{\hat{\theta}} = \Gamma\text{Proj}\left(\hat{\sigma}(t), -\begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix} \begin{bmatrix} 4.50 & 0.50 \\ 0.50 & 0.625 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{lc}{J_z} \end{bmatrix}\right) \tag{5.51}$$

$$\dot{\hat{\theta}} = \Gamma\text{Proj}\left(\hat{\sigma}(t), -\left(0.125\tilde{\psi} + 0.156\dot{\tilde{\psi}}\right)\right) \tag{5.52}$$

$$G_{\mathcal{L}1,rz} = [[sI - A_m]^{-1} B_m]\left[1 - C_{r,z}(s)\right] \tag{5.53}$$

Using matlab, the transfer function is where wc =100 for the low pass filter

$$G_{\mathcal{L}1,rz} = \begin{bmatrix} \frac{0.25s}{s^3 + 101s^2 + 100.25s + 25} \\ \frac{0.25s^2}{s^3 + 101s^2 + 100.25s + 25} \end{bmatrix} \tag{5.54}$$

$\left\|G_{\mathcal{L}1,tz}\right\|_{\mathcal{L}1} = 0.001$ which give us an uncertainty limit of 1000N. With that, the adaptation gain can be big enough for faster adaptation. Also from equation 5.52, $\tilde{\psi} = \hat{\psi} - \psi$ and $\dot{\tilde{\psi}} = \dot{\hat{\psi}} - \dot{\psi}$. For simulation, a varying uncertainty of -10N to 10N would be used with an adaptation gain of 10000. Figure 5.11 provides result tracking desired reference angle using $\mathcal{L}_1$ adaptive controller with varying uncertainties while figure 5.12 provides simulation of the estimated uncertainty vs the real uncertainty.

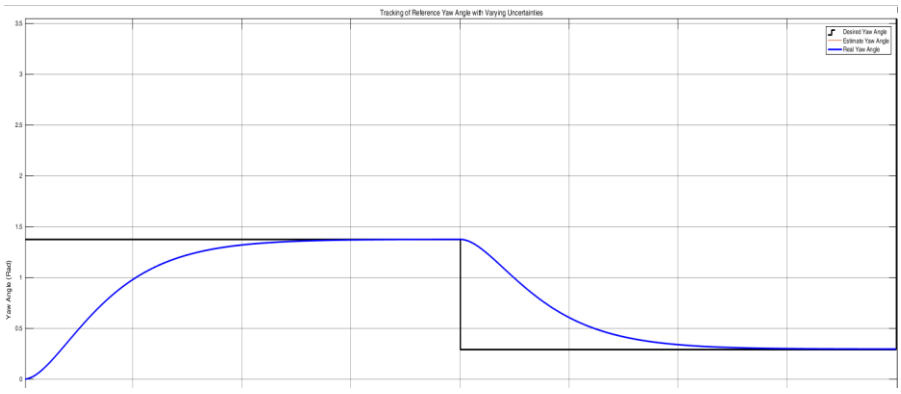


Figure 5.11: Tracking of Desired Reference Yaw Angle with Varying Input Uncertainties

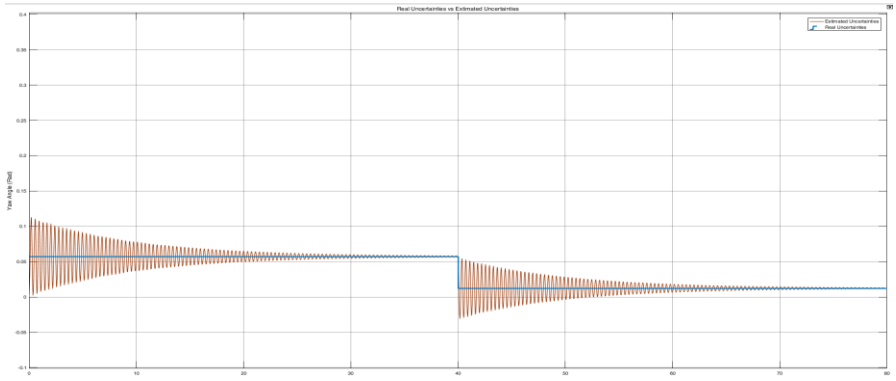


Figure 5.12: Estimated Uncertainty vs Real Uncertainty for Yaw Control

5.6 TRANSLATION CONTROL ALONG THE X-AXIS

For movement along the X or Y axis, the quadrotor needs to rotate because it is dependent on the rotation angle. From equation 5.56, you can notice that the translation movement is dependent on ϕ, θ, φ and by using small angle approximation, we can move in the X direction without the quadrotor getting unstable. For stability to be maintained, the pitch and roll angle would be limited to maintain small angle approximation. For movement along the X axis, the quadrotor rotates about the Y axis which provides us the pitch angle. The yaw angle can then be set equal to zero because the yaw angle is relatively

small and does not have a huge effect on movement in the X and Y direction. Also, equation 5.55 can be linearized around hovering mode were $\dot{\phi}, \dot{\theta}, \dot{\varphi}$ is equal to zero.

$$\ddot{\theta} = \frac{l}{J_y} U_{tx} \tag{5.55}$$

$$\ddot{X} = \frac{1}{m}(\cos(\phi)\sin(\theta)\sin(\varphi) + \sin(\theta)\,sin(\varphi)(-U_{tz}) \tag{5.56}$$

Like the translational controller using small angle approximation,

$$\ddot{X} = -\frac{1}{m}\theta(U_{tz}) \tag{5.57}$$

It is also important to note that when small angle approximation is used,

$$\sin\theta \approx \theta$$

$$\cos\theta \approx 1$$

$$\tan\theta \approx \theta$$

We would assume that the control input is positive and we would compensate for it on the controller output for movement along the x axis. We would also set $U_{tz} = mg$ so that our model can be simplified for easy calculation of the gain constant.

For our state feedback, we would four states and our state space feedback would be modelled as

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ -\left(\frac{l}{Jy}\right)k1_{tx} & -\left(\frac{l}{Jy}\right)k2_{tx} & -\left(\frac{l}{Jy}\right)k3_{tx} & -\left(\frac{l}{Jy}\right)k4_{tx} \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x_1} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{l}{J_y} \end{bmatrix} U_{tx} \tag{5.58}$$

The Simulink model is shown below and the transfer function becomes

$$G_{tx}(s) = \frac{\frac{l}{J_y} g}{s^4 + k4_{r,y}\left(\frac{l}{J_y}\right)gs^3 + k3_{r,y}\left(\frac{l}{J_y}\right)gs^2 + k2_{r,y}\left(\frac{l}{J_y}\right)gs + k1_{r,y}\left(\frac{l}{J_y}\right)g} \tag{5.59}$$

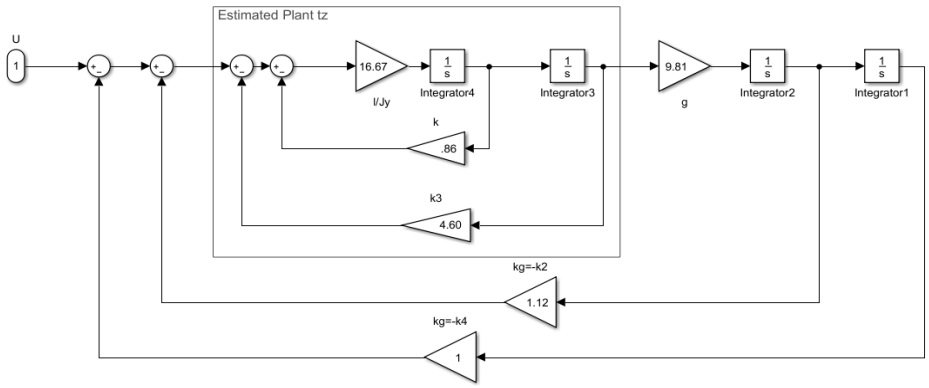


Figure 5.13: State Feedback Design in Simulink

$$s^4 + k4_{r,y}\left(\frac{l}{J_y}\right)gs^3 + k3_{r,y}\left(\frac{l}{J_y}\right)gs^2 + k2_{r,y}\left(\frac{l}{J_y}\right)gs + k1_{r,y}\left(\frac{l}{J_y}\right)g = \left(s + \sqrt[4]{\frac{l}{J_y}g}\right)^4$$

The result using pole placement for the feedback gain constants becomes

$$(k1_{tx} = 1;) \quad \left(k2_{tx} = 4\sqrt[4]{\frac{J_y^3}{l^3}g}\right); \quad \left(k3_{tx} = 6\sqrt[2]{\frac{J_y}{l}g}\right); \quad \left(k4_{tx} = 4\sqrt[4]{\frac{J_y}{lg}}\right)$$

$$\mathrm{k1_{tx}} = 1;\ \mathrm{k2_{tx}} = 1.12;\ \mathrm{k3_{tx}} = \ 4.60;\ \mathrm{k4_{tx}} = .86;$$

Where $J_y = \ 0.015kgm^2; l = 0.25m; and\ g = 9.81N/kg$

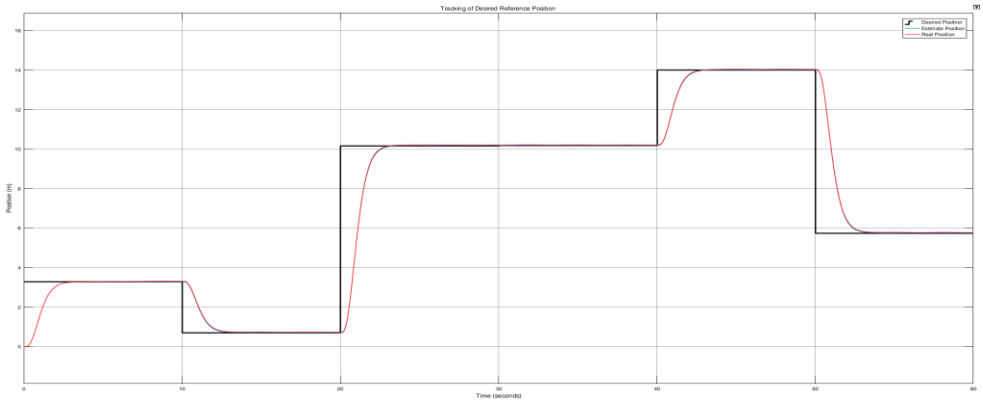


Figure 5.14: Roll Position Tracking without Varying Input Uncertainties

Because we are designing a controller for movement around the x axis, we would focus on the pitch angle θ. It is also important to know that figure 5.14 did not consider the limitation

of the pitch angle, but to maintain small angle approximation, θ should not exceed 14^o or 0.244rad. However, when position is greater than 1.4m as shown in figure 5.16, the pitch angle is greater than 0.244rad and if our desired position is as high as 25m, θ would be higher than 360^o or 3.14rad which makes no sense for the quadrotor to completely rotate and possibly leads to more difficulty for the control of the quad.

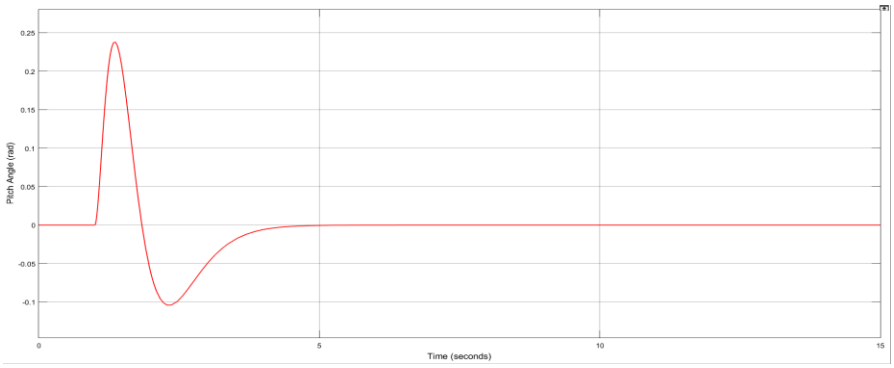


Figure 5.15: Pitch Angle at a Desired Position of 1.4m

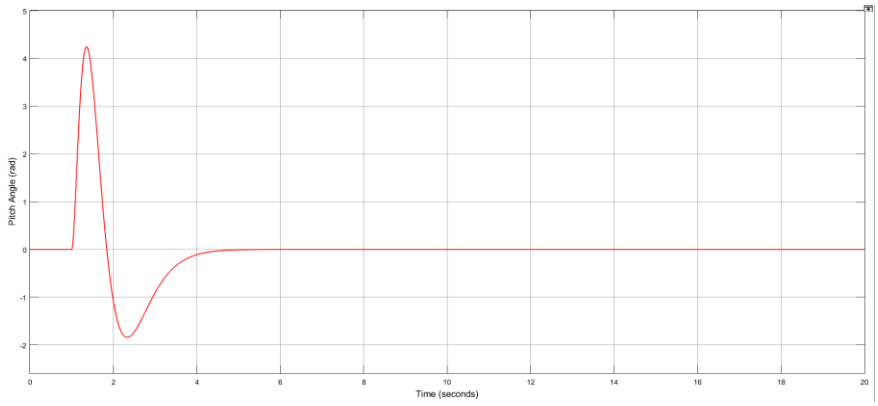


Figure 5:16 Pitch angle at a desired position of 25m

The problem with limitation is that it could affect state feedback control law and to avoid that, we would have to first limit tracking error. Where

$$e_{tx} = x_{tx} + e_{tx_{limit}}$$

$$e_{tx_{limit}} = -3m < e_{tx} < 3m$$

$$e_{tx} = r_{tx} - x_{tx} \tag{5.60}$$

From equation 5.60, if the error between the reference input and the actual position is greater than 3m, then $e_{tx_{limit}}$ is equal to 3m and if less than -3m then $e_{tx_{limit}}$ is equal to -3m. By using this technique, the tracking error cannot exceed 3m even if the actual reference is greater than 3m. With the limitation in place, it takes a longer time for the quadrotor to reach desired position. Also 3m was the best limitation that would result in a faster response and still maintain small angle approximation at a degree higher than 14^o. Figure 5.17 and figure 5.18 shows the pitch angle remains same even if the distance is greater than 1.4m. Figure 5.19 and 5.20 shows the response of the quadrotor at 1.4m and at 25m.

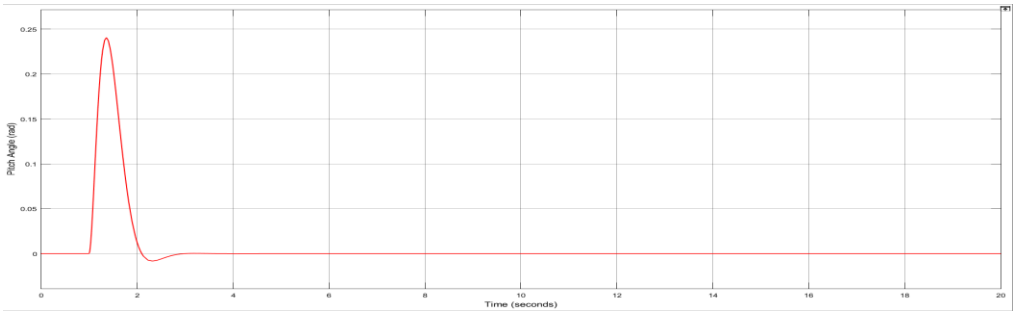


Figure 5.17: Pitch Angle with Desired Position of 25m

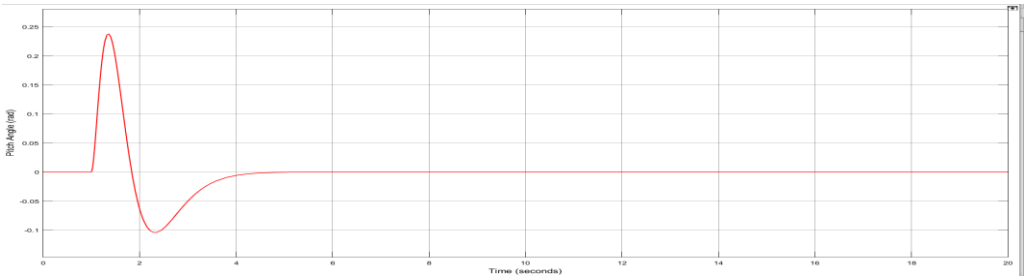


Figure 5.18: Pitch Angle with Desired Position of 1.4m

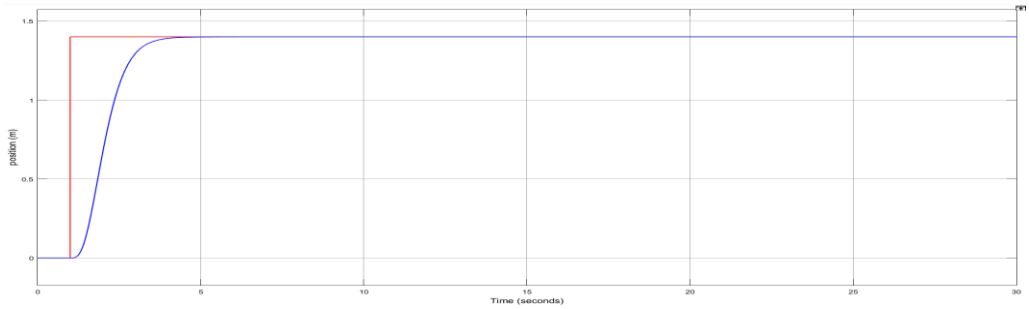


Figure 5.19: Tracking of Desired Position of 1.4m with Pitch Angle Limitation

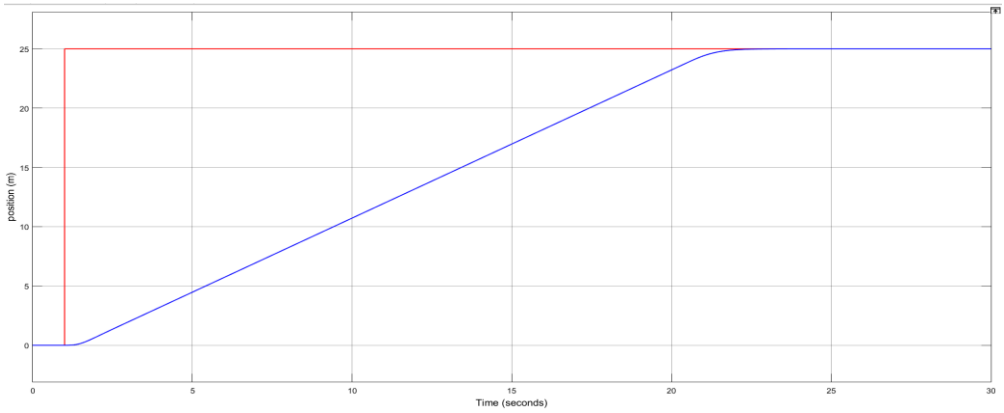


Figure 5.20: Tracking of Desired Position of 25m with Pitch Angle Limitation

Unlike the altitude controller we cannot directly affect translation without affecting rotation. Therefore, we have matched and unmatched uncertainty. Figure 5.21 provides a simulation where unmatched uncertainties were not compensated for and in other to compensate for these uncertainties, the model of the system is redefined as

$$\dot{x}(t) = A_m x(t) + B_m((u(t) + \sigma_m(t)) + B_{um}\sigma_{um} \tag{5.61}$$

Where

$B_{um} \in \mathbb{R}^{n\times n(-m)}$ Is a matrix such that $B_{um}^T = 0$ and rank$([B_m B_{um}]) = n$

There for

$$B_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{l}{J_y} \end{bmatrix} U_{tx} \quad and\ B_{um} = \begin{bmatrix} 0 \\ g \\ 0 \\ 0 \end{bmatrix} \theta \tag{5.62}$$

The control law for the controller to compensate for unmatched uncertainty would be defined as

$$U = -C_1(S)\hat{\sigma}_m - C_2 H_m^{-1} H_{um}\hat{\sigma}_{um} + K_{gr} \tag{5.63}$$

Where

$C_1 \& C_2$ are the low pass filter; $H_m = C[sI - A_m]^{-1}B_m$ and $H_{um} = C[sI - A_m]^{-1}B_{um}$. $C_2 H_m^{-1} H_{um}\hat{\sigma}_{um}$ is needed to cancel the effect of unmatched uncertainty. In this paper, C_2 filter would be a third order low pass filter. It is needed because it makes the transfer function of $C_2 H_m^{-1} H_{um}\hat{\sigma}_{um}$ a proper transfer function. At a filter bandwidth of 100,

$$C_2 H_m^{-1}\mathrm{H_{um}} = \frac{60\mathrm{s}^2 + 860\mathrm{s} + 4600}{\mathrm{s}^3 + 24.14\mathrm{s}^2 + 241.4\mathrm{s} + 1000} \tag{5.64}$$

P matrix to satisfy Lyapunov equation

$$\mathrm{P} = \begin{bmatrix} 1.36 & 0.50 & 0.21 & 0.05 \\ 0.50 & 2.08 & 0.05 & 2.20 \\ 0.21 & 0.05 & 2.22 & 0.50 \\ 0.05 & 2.20 & 0.50 & 5.51 \end{bmatrix} \tag{5.65}$$

Adaptation control law

$$\dot{\hat{\theta}}_{\mathrm{m}} = \Gamma\mathrm{Proj}\left(\hat{\sigma}(\mathrm{t}), -\left(0.834\tilde{\mathrm{x}} + 36.674\dot{\tilde{\mathrm{x}}} + 8.335\tilde{\theta} + 91.852\dot{\tilde{\theta}}\right)\right) \tag{5.66}$$

$$\dot{\hat{\theta}}_{\mathrm{um}} = \Gamma\mathrm{Proj}\left(\hat{\sigma}(\mathrm{t}), -\left(4.905\tilde{\mathrm{x}} + 20.405\dot{\tilde{\mathrm{x}}} + 0.491\tilde{\theta} + 21.582\dot{\tilde{\theta}}\right)\right) \tag{5.67}$$

The adaptation law for matched uncertainties does not include does not include movement on the translational axis and therefore adaptation law for unmatched uncertainties does not include movement on the rotational axis and can be shown in the equation below.

$$\dot{\hat{\theta}}_m = \Gamma\mathrm{Proj}\left(\hat{\sigma}(t), -\left(0\tilde{x} + 0\dot{\tilde{x}} + 8.335\tilde{\theta} + 91.852\dot{\tilde{\theta}}\right)\right) \tag{5.68}$$

$$\dot{\hat{\theta}}_{um} = \Gamma\mathrm{Proj}\left(\hat{\sigma}(t), -\left(4.905\tilde{x} + 20.405\dot{\tilde{x}} + 0\tilde{\theta} + 0\dot{\tilde{\theta}}\right)\right) \tag{5.69}$$

$$G_{\mathcal{L}1,tx(matched)} = [[sI - A_m]^{-1}B_m]\left[1 - C_{r,z}(s)\right] \tag{5.70}$$

$$= \begin{bmatrix} \frac{1.635e06s^3 + 3.925e08s^2 + 3.925e10s}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \\ \frac{1.635e06s^4 + 3.925e08s^3 + 3.925e10s^2}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \\ \frac{1.635e06s^5 + 3.925e08s^4 + 3.925e10s^3}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \\ \frac{1.635e06s^6 + 3.925e08s^5 + 3.925e10s^4}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \end{bmatrix} \quad (5.71)$$

$$G_{\mathcal{L}1,tx(unmatched)} = [[sI - A_m]^{-1}B_{um}][1 - C_{r,z}(s)] \quad (5.72)$$

$$= \begin{bmatrix} \frac{1.635e06\ s\text{^}5 + 3.939e08\ s\text{^}4 + 3.959e10\ s\text{^}3 + 3.554e10\ s\text{^}2 + 1.805e11\ s}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \\ \frac{1.635e06\ s\text{^}6 + 3.939e08\ s\text{^}5 + 3.959e10\ s\text{^}4 + 3.554e10\ s\text{^}3 + 1.805e11\ s\text{^}2}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \\ \frac{-1.832e06\ s\text{^}4 - 4.412e08\ s\text{^}3 - 4.435e10\ s\text{^}2 - 3.925e10\ s}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \\ \frac{-1.832e06\ s\text{^}5 - 4.412e08\ s\text{^}4 - 4.435e10\ s\text{^}3 - 3.925e10\ s\text{^}2}{10000s^7 + 2.543e06s^6 + 2.752e08s^5 + 1.363e10s^4 + 1.621e11s^3 + 8.11e11s^2 + 1.871e12s + 1.635e12} \end{bmatrix} \quad (5.73)$$

$\left\|G_{\mathcal{L}1,tx(matched)}\right\|_{\mathcal{L}1} = 0.008$ which give us an uncertainty limit of 125N. With that, the adaptation gain can be big enough for faster adaptation. From equation 5.68, $\tilde{\theta} = \hat{\theta} - \theta$ and $\dot{\tilde{\theta}} = \dot{\hat{\theta}} - \dot{\theta}$. For simulation, a varying uncertainty of -10N to 10N would be used with an adaptation gain of 10000.

$\left\|G_{\mathcal{L}1,tx(unmatched)}\right\|_{\mathcal{L}1} = 0.056$ which give us an uncertainty limit of 17.54N. From equation 5.69, $\tilde{x} = \hat{x} - x$ and $\dot{\tilde{x}} = \dot{x} - x$. For simulation, a varying uncertainty of -10N to 10N would be used with an adaptation gain of 10000.

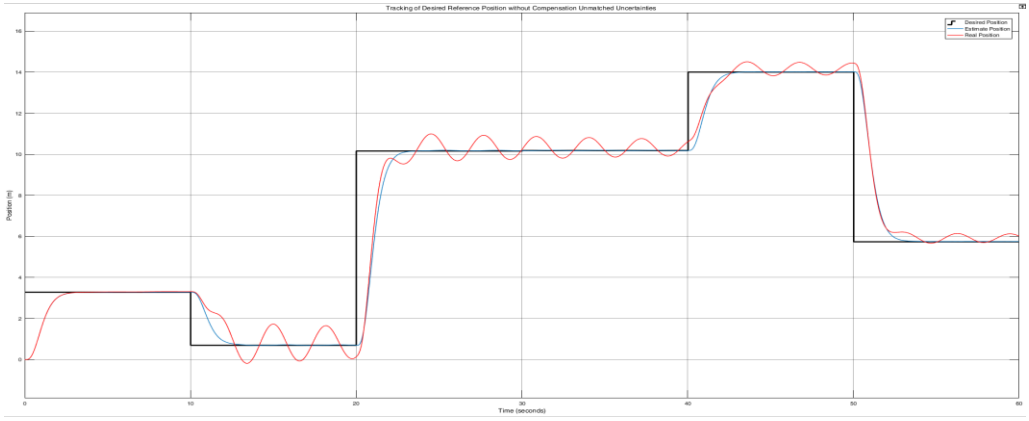


Figure 5.21: Tracking of Desired Reference Position with Varying Matched and Unmatched Uncertainties without compensation for the Unmatched Uncertainties

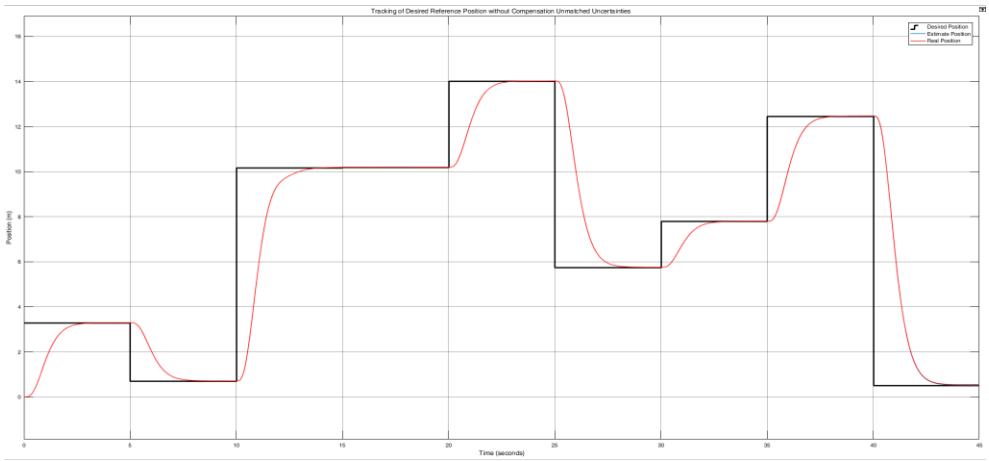


Figure 5.22: Roll Position Tracking with Varying Matched Uncertainties

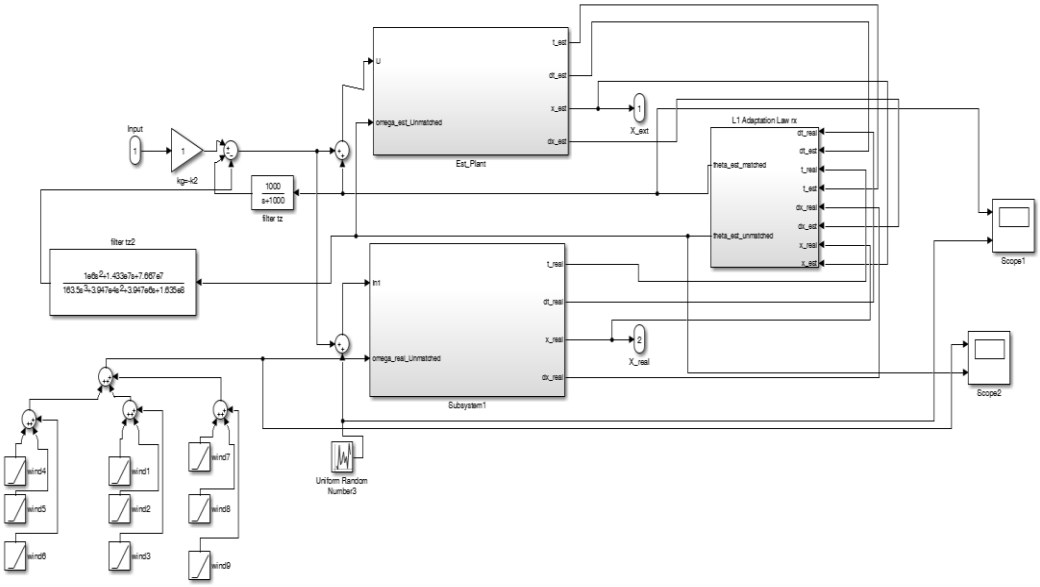


Figure 5.23: Structural Design for Roll Controller

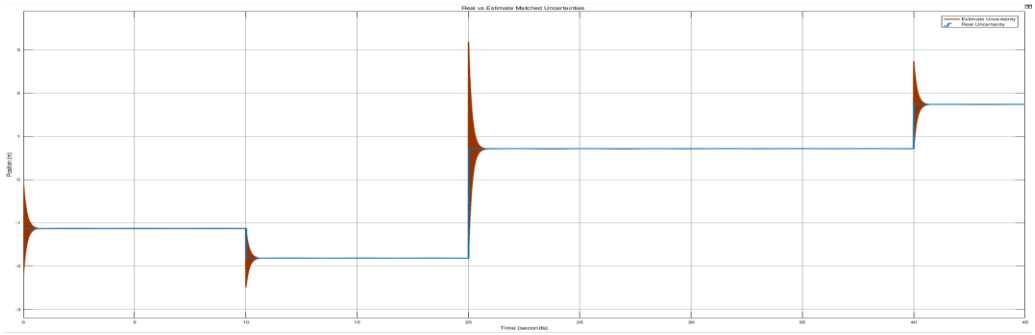


Figure 5.24: Estimate Uncertainty vs Real Uncertainty for Roll Controller

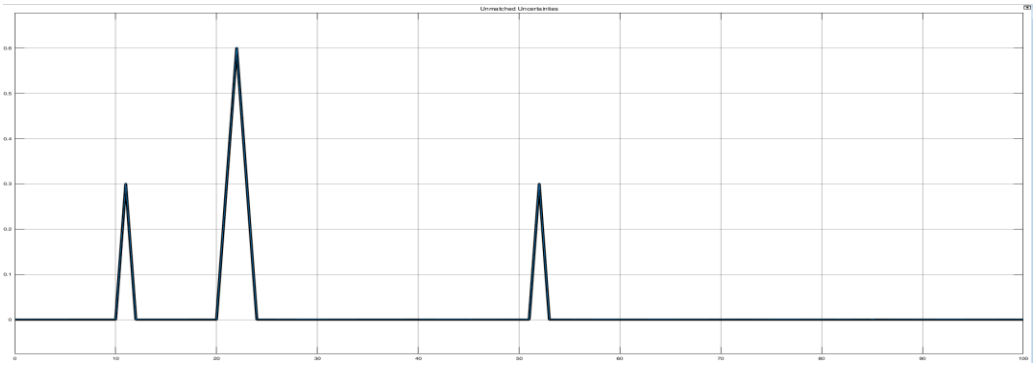


Figure 5.25: Estimate Uncertainty vs Real Uncertainty for Unmatched Uncertainties Roll Control

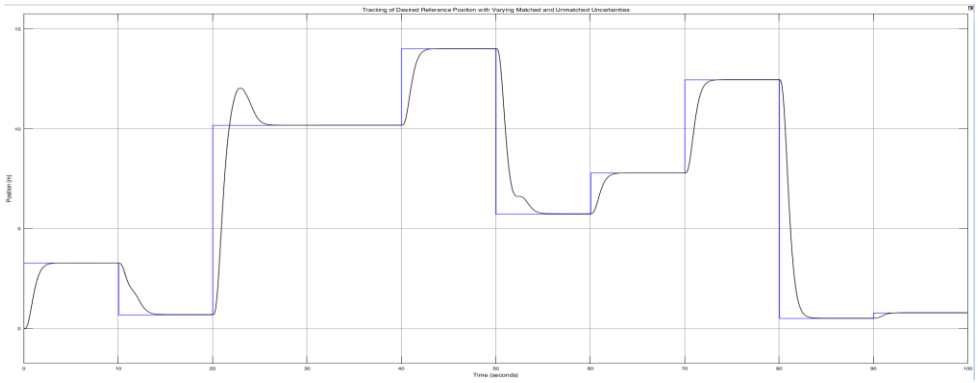


Figure 5.26: Roll Position Tracking with Varying Matched and Unmatched Uncertainties

5.7 TRANSLATION CONTROL ALONG THE Y AXIS

This controller design is similar for the controller design for translation about the x axis, so controller output for movement is

$$u_{ty} = (e_{ty} - k1_{tx}\mathrm{x} - k2_{tx}\dot{x} - k3_{tx}(\phi) - k4_{tx}(\dot{\phi}))$$

You notice there is no multiplication of (-1) at the dynamic model which is the only difference between both design. Therefore

$$\mathrm{k1_{ty}} = 1;\ \mathrm{k2_{ty}} = 1.12;\ \mathrm{k3_{ty}} = 4.60;\ \mathrm{k4_{ty}} = .86;$$

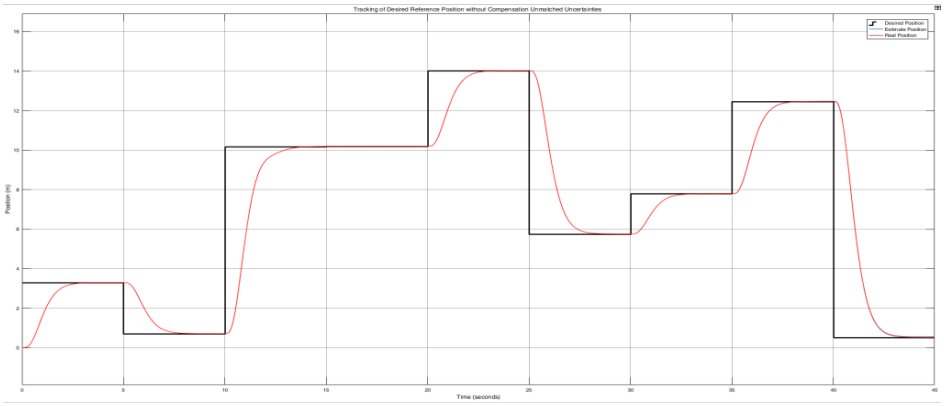


Figure 5.27: Feedback Response without Disturbance

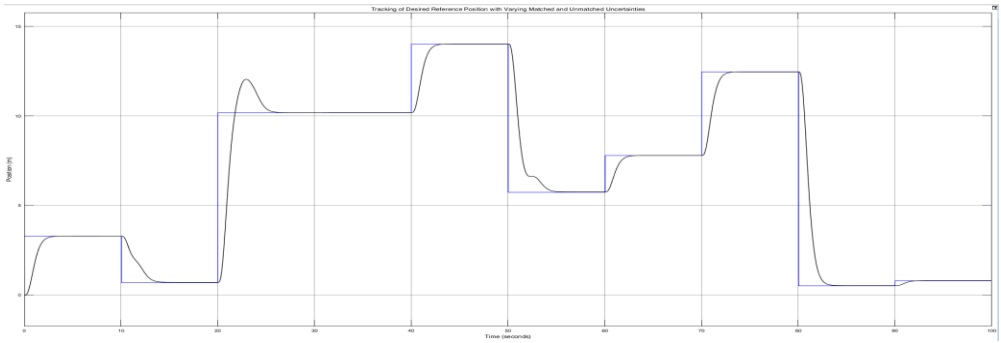


Figure 5.28: Pitch Position Tracking with Varying Matched

and Unmatched [Y-Axis]

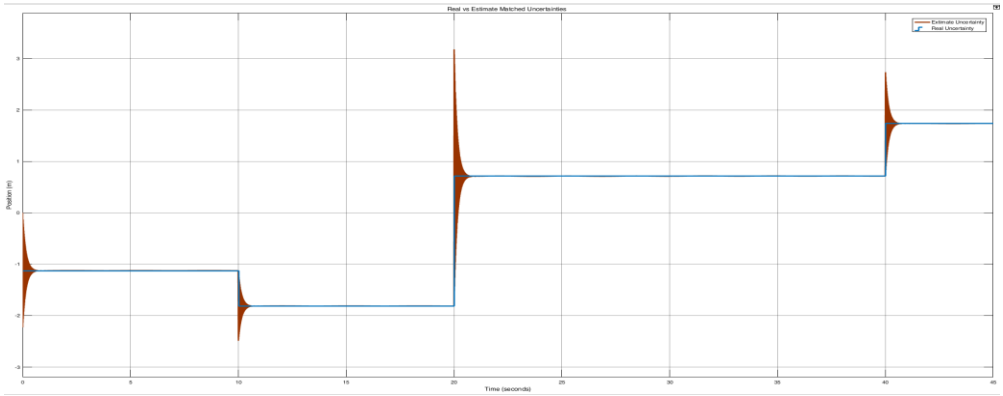


Figure 5.29: Estimate Uncertainty vs Real Uncertainty for Matched Uncertainties Pitch

Control

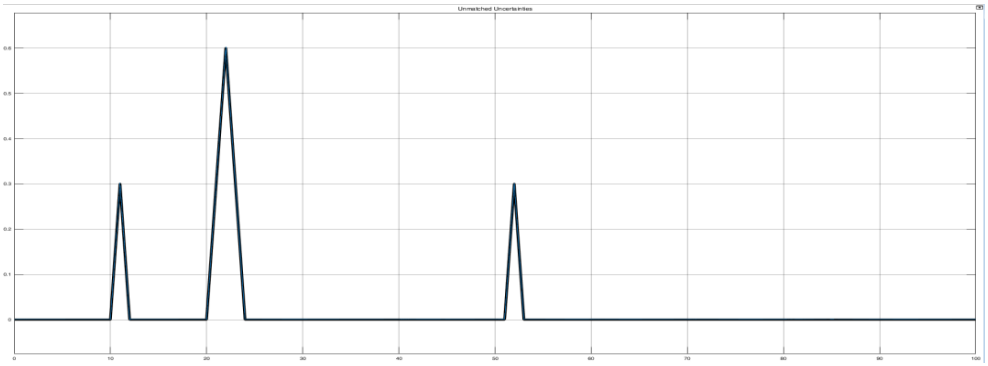


Figure 5.30: Estimate Uncertainty vs Real Uncertainty for Unmatched Uncertainties Pitch Control

CHAPTER 6

CONTROLLER SIMULATION WITH NON-LINEAR MODEL

In Chapter 5, controllers were designed based on a linearized model of the quadrotor, but in this chapter 6, the nonlinear model is used. The non-linear model is based on what was discussed in chapter 3 and in addition to the model we would calculate the thrust generated by each motor as shown below. These thrusts are mathematical relations that were discussed in chapter 3 from equation 3.6-3.9.

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} U_{tz} \\ U_{tx} \\ U_{ty} \\ U_{rz} \end{bmatrix}$$

For the simulation, the table below shows the disturbances that were used on the quadrotor model.

Tz Disturbance	-10< Tz < 10
Ty Disturbance	-0.4< Ty < 0.4
Tx Disturbance	-0.4< Tx < 0.4
Rz disturbance	-10< Tz < 10
Ry disturbance	-12< Ry < 12
Rx disturbance	-12< Rx < 12
Tz Extra mass	-5< Tz < 5

Figure 6.1: Disturbance Limit

Figure 6.1-6.4 shows the quadrotor response to desired position or location. It is exactly same as most of the result shown in chapter 5. To avoid repetition of results such as tracking of uncertainties, pitch and roll limitation would be avoided. Figure 6.5 and 6.6 provides the overall Simulink design and each detail in each subsystem have been discussed from chapter 3 to chapter 5.

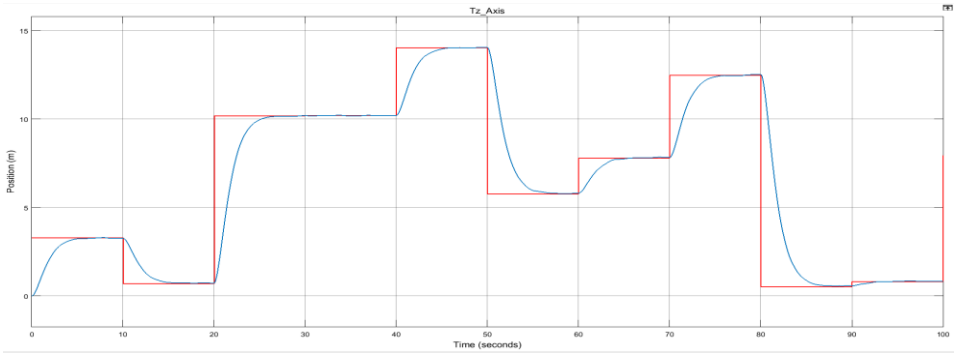


Figure 6.2: Altitude Position Tracking Using $\mathcal{L}_1$ Adaptive Controller [Z-Axis]

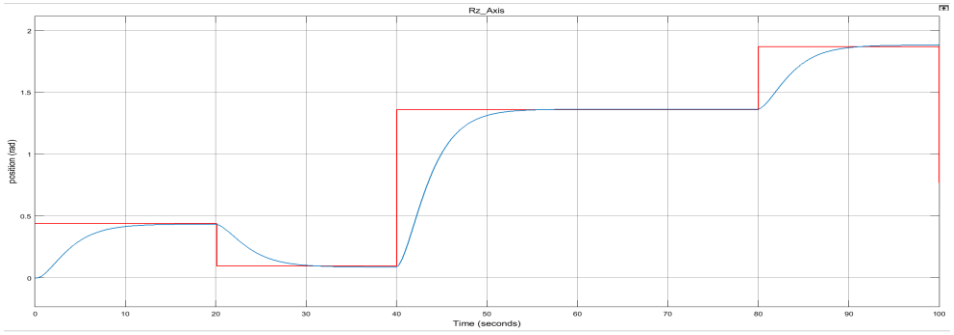


Figure 6.3: Yaw Position Yaw Position Tracking Using $\mathcal{L}_1$ Adaptive Controller [Z-Axis]

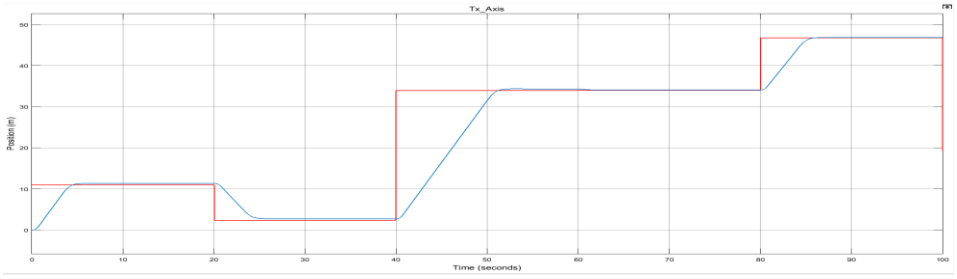


Figure 6.4: Roll Position Tracking Using $\mathcal{L}_1$ Adaptive Controller [X-Axis]

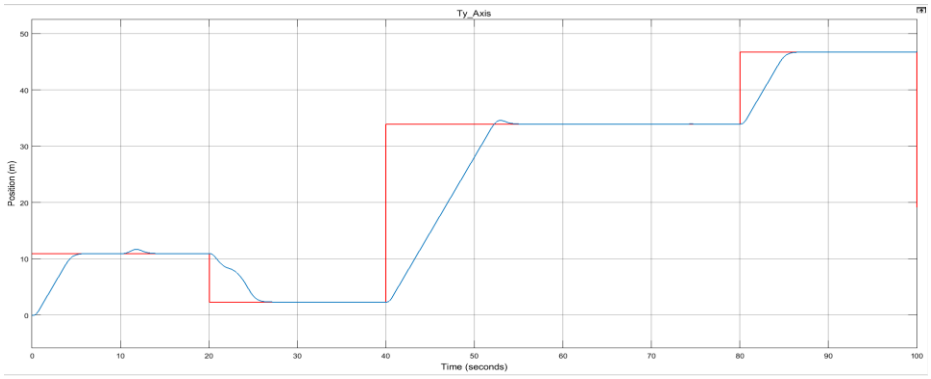


Figure 6.5: Pitch Position Tracking Using $\mathcal{L}_1$ Adaptive Controller [Y-Axis]

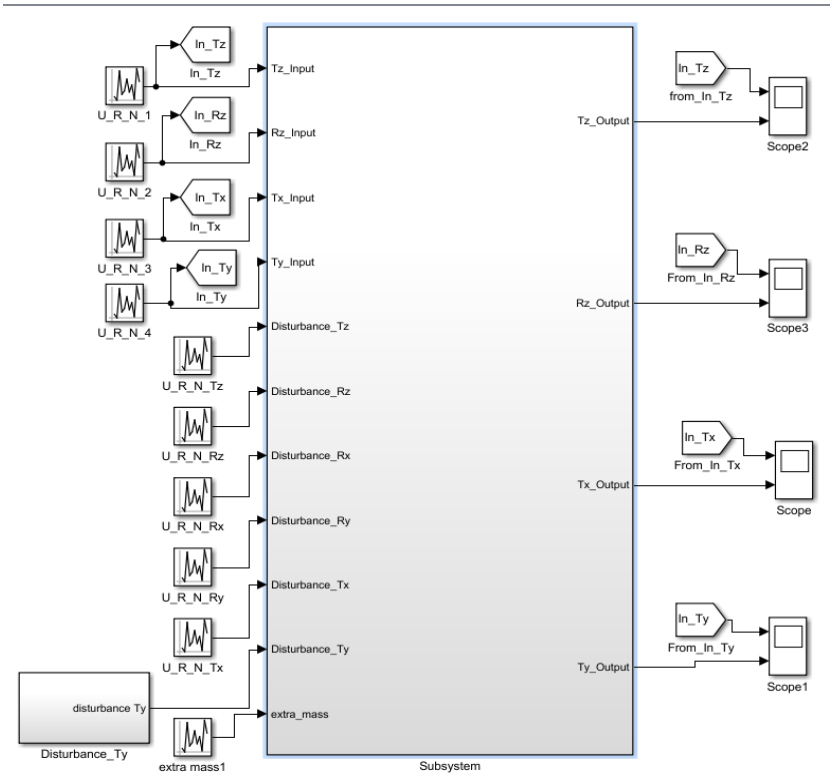


Figure 6.6: Simulink Design with Uncertainties Input

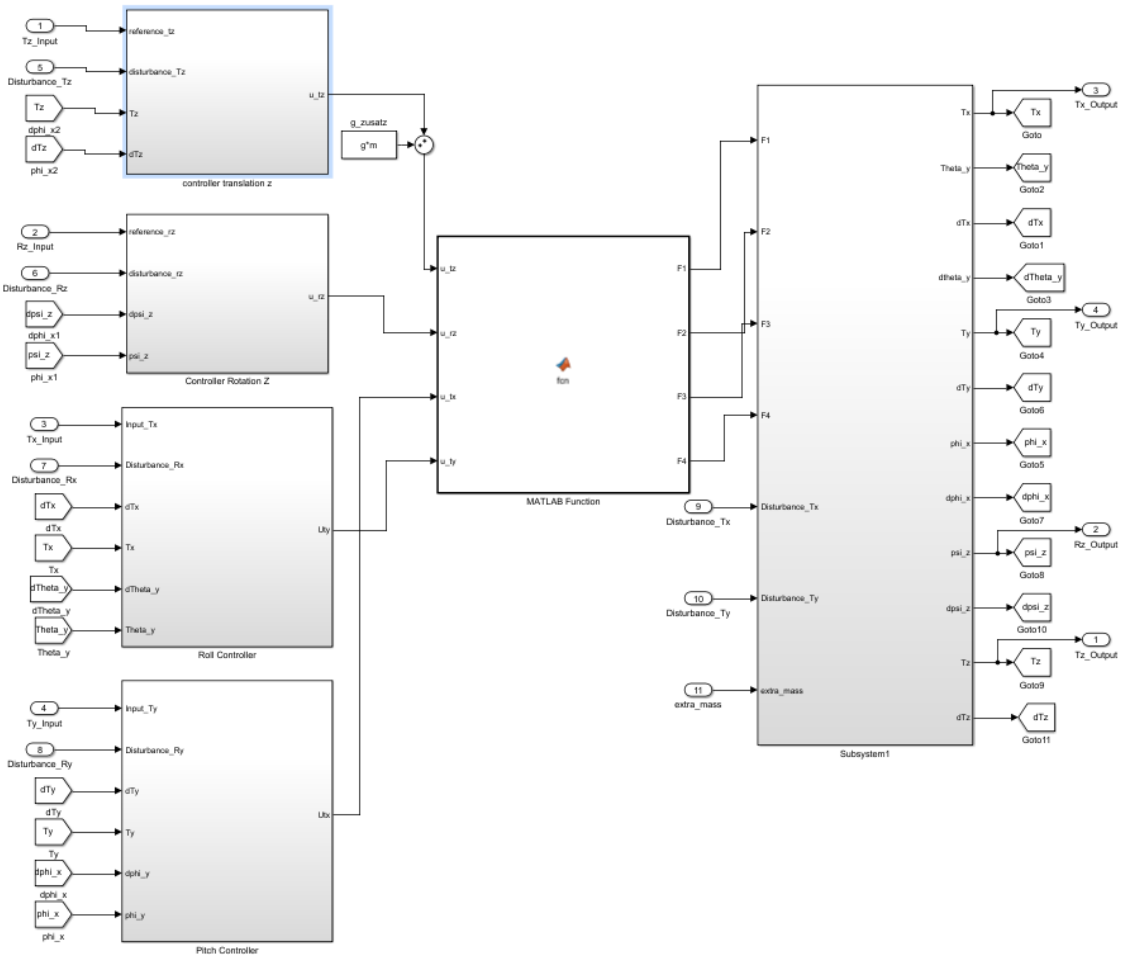


Figure 6.7: Expanded Simulink Design

CHAPTER 7

SUMMARY AND CONCLUSION

The main goal of this thesis is to design a controller that can control a quadrotor in the presence of disturbance or uncertainties. Although they are different control method used such as PID controller, this paper focused on the application of $\mathcal{L}_1$ adaptive controller in simulation. This paper includes the linear and non-linear dynamic model of the quadrotor, application using PID controller, $\mathcal{L}_1$ adaptive controller, and resistance to uncertainties. As figure 4.10 shows, when a step disturbance is added to the input of the altitude controller, it took about 160 seconds for the quadrotor to adapt so the main advantage of $\mathcal{L}_1$ adaptive controller over PID controller is its fast adaptation to uncertainties or disturbance because of its high adaptation gain.

In the process of designing $\mathcal{L}_1$ adaptive controller for the quadrotor, various limitations and problems are taking into consideration such as the under actuation of the quadrotor, presentation of matched and unmatched uncertainties, addition of extra mass and failure of rotating motor. All these limitations were taking care of in this paper, except for failure of the rotating motor. Theoretically when there is a motor failure, it is difficult to have a stable flight. To avoid these problems, a different technique would be used for future work. Future work would also focus on the implementation of $\mathcal{L}_1$ adaptive controller on a physical quadrotor. Various limitation from the micro controller, body

frame battery, IMU and cost of design are expected to affect performance, but with ongoing research in the field of quadrotors, most of these limitations can be subdued.

REFERENCES

Bouabdallah, Samir and Siegwart, Roland. 2007. *Full Control of Quadrotor*. Autonomous Systems Lab: Swiss Federal Institute of Technology, ETHZ Zurich, Switzerland.

Brudigam, Tim. 2014. *Fault-Tolerant Control of Quadrotor*. Thesis, Technical University of Munich.

ElKholy, Heba talla Mohamed Nabil. 2014. *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches.* Thesis, America University in Cairo.

Hovakimyan, Naira. 2013. $\mathcal{L}_1$ *Adaptive Control and Its Transition to Practice.* Department of Mechanical Science and Engineering University of Illinois.

Hovakimyan, Naira. 2013. $\mathcal{L}_1$ *Adaptive Control.* Department of Mechanical Science and Engineering University of Illinois.

Hovakimyan, Naira and Cao, Chengyu. 2010. $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation.* Philadelphia: Society for Industrial & Applied Mathematics, U.S.

Murray, M. Richard and Aström, J. Karl. 2009. *Feedback Systems: An introduction for Scientist and Engineering*. Princeton: Princeton University Press.

Premerlani, William and Bizard, Paul. 2009. *Direction Cosine Matrix IMU: Theory.* Retrieved from Web.

Raza, A. Syed and Gueaieb, Wail. 2010. *Intelligent Flight Control of an Autonomous Quadrotor.* Science, Technology and Medicine. Doi: 10.5772/6968.

Schmidt, M. David. 2011. *Simulation and Control of a Quadrotor Unmanned Aerial Vehicle.* University of Kentucky Library.

Seiler, Peter, Dorobantu, Andrei and Balas, Gary. 2010. *Robustness Analysis of an $\mathcal{L}_1$ Adaptive Controller.* AIAA Guidance, Navigation, and Control Conference.

Wang, Xiaofeng and Hovakimyan, Naira. 2012. *$\mathcal{L}_1$ Adaptive Controller for Nonlinear Time-Varying Reference Systems.* Systems & Control Letters, 61(4): 455-463.