Fabrication, Modeling and Control of a Spherical Tail-Sitter UAV

by

Sri Ram Prasath Ramasubramaniyan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2018 by the
Graduate Supervisory Committee:

Spring Berman, Chair
Marc Mignolet
Konstantinos Tsakalis

ARIZONA STATE UNIVERSITY

August 2018

ABSTRACT

In the past decade, real-world applications of Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicles (UAV) have increased significantly. There has been growing interest in one of these types of UAVs, called a tail-sitter UAV, due to its VTOL and cruise capabilities. This thesis presents the fabrication of a spherical tail-sitter UAV, based on the vehicle design in Loh and Jacob (2013), and derives a nonlinear mathematical model of its dynamics. The singularity in the attitude kinematics of the vehicle is avoided using Modified Rodrigues Parameters (MRP). The model parameters of the fabricated vehicle are calculated using the bifilar pendulum method, a motor stand, and ANSYS simulation software. Then the trim conditions at hover are calculated for the nonlinear model, and the rotational dynamics of the model are linearized around the equilibrium state with the calculated trim conditions. Robust controllers are designed to stabilize the UAV in hover using the H_2 control and H_∞ control methodologies. For H_2 control design, Linear Quadratic Gaussian (LQG) control is used. For the H_∞ control design, Linear Matrix Inequalities (LMI) with frequency-dependent weights are derived and solved using the MATLAB toolbox YALMIP. In addition, a nonlinear controller is designed using the Sum-of-Squares (SOS) method to implement large-angle maneuvers for transitions between horizontal flight and vertical flight. Finally, the linear controllers are implemented in the fabricated spherical tail-sitter UAV for experimental validation. The performance trade-offs and the response of the UAV with the linear and nonlinear controllers are discussed in detail.

To Amma, Appa and Anna

ACKNOWLEDGMENTS

I am grateful to my advisor, Dr. Spring Berman, for believing in me, and letting me work in this exciting research area. She consistently allowed this thesis to be my own work, but steered me in the right direction whenever she thought I needed it.

This thesis would not have been possible without Dr. Mignolet, who has become a mentor and significantly influenced my academic experience at ASU. I am forever indebted, for he patiently guided me through my mistakes.

I would also like to thank my committee member Dr. Tsakalis. He calmly listened to me and made me realize some of the advanced concepts in control. His ideas about the problem helped me to improve it efficiently.

I would also like to thank my colleagues in ACS Lab, my friends and my brother who were curious and helped in the different concepts for this research project: Aniket, Shiba, Ragesh, Karthik, Ruben, Zahi, Venkat, and Rakshit. Without their passionate participation and input, the thesis could not have been successfully.

Finally, I must express my very profound gratitude to my parents and to my brother for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This journey would not have been possible without them. Thank you.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

An Unmanned Aerial Vehicle (UAV) is a flying aircraft piloted by remote control or software, and which is capable of reuse Parker (2018). In this past decade, UAVs have been widely used for personal, military, industry and civil applications. Currently, many people have a personal drone to fly, take pictures or video, and race Coach (2018). In military applications, UAVs are widely used for surveillance, communication, and payload delivery Carapau et al., 2017. In industry, UAVs are mainly used to inspect high voltage power lines and industrial equipment Uranchimeg et al. (2017). There are more than a hundred UAV companies that are developing UAVs for mapping and reconstruction, package delivery, air transport, photography, videos and education Coach (2018). Recently, Uber has been working on the air taxi, which they are planning to demonstrate by 2020 and begin commercial operations by 2023 Elevate (2016).

In general, aircraft are classified as fixed-wing aircraft and rotorcraft based on the design. Fixed-wing aircraft have wings attached to the fuselage, which are used to produce lift to fly, whereas rotorcraft use propellers to generate lift. The rotorcraft are basically a VTOL (Vertical Take Off and Landing) system and fixed wings are a CTOL (Conventional TakeOff and Landing) system. Both types of aircraft have their advantages and disadvantages, summarized as follows:

	Advantages	Disadvantages
Fixed-Wing	Endurance or Long Cruise Time	Need runway for take off and landing, No hover ability
Rotorcraft	Hover, Easy Take off	Short flight duration

Table 1.1: Aircraft Types and Differences

There are many research attempts to develop systems which have the capabilities of both VTOL and fixed-wing aircraft. Different types of vehicle designs that have advantages of both types of aircraft are listed below Gordon (2016):

- **Convertiplane**: It has a fixed rotor like a helicopter for take off, then the lift is produced by fixed wings in forward flight
- **Tiltrotor**: It has rotors mounted perpendicular to the wings for VTOL and then the rotors are tilted for the transition to forward flight
- **Tiltjet**: It is similar to the tiltrotor; instead of propellers, turbojets are used
- **Tiltwing**: For this design, both the wing and propeller are vertical during take off, then both of them rotate for forward flight
- **Tail-sitter**: A tail-sitter sits vertically on its tail for takeoff and landing, then the UAV is rotated for forward flight
- **Vectored thrust**: In this design, the direction of the engine exhaust is changed for transition between horizontal and forward flight
- **Lift jets or fans**: This design includes an auxiliary jet engine or fans for VTOL capability

- **Lift via Coanda effect**: This design uses the Coanda effect for vertical takeoff

In this thesis, we discuss a Spherical VTOL UAV Tail-sitter. The objective is to fabricate, model and control the spherical UAV.

1.1 Literature Review

In 1928, Nicholas Tesla patented the concept of a Tail-Sitter. In the Second World War, the German tail-sitter Focke-Wulf Triebflgel was introduced. After that World War, the U.S. started experimenting with the tail-sitter. The Convair XFY Pogo was the first tail-sitter project of the U.S. in 1954. This tail-sitter was driven by contra-rotating propellers and it had two delta wings perpendicular to each other Chana et al. (1996). This type of tail-sitter is very difficult to operate because of the transition from vertical to horizontal flight of the complete aircraft. Due to many failed attempts, the U.S. stopped their experiments for pilot safety concerns. In the past decade, due to technical advances, UAVs have started exploring the sky. Hence, it has become possible to implement autonomous controllers for these tail-sitters.

In Stone (2004), the guidance and control architecture of a tail-sitter is explained in detail. There are several papers from Brigham Young University on the design and development of tail-sitters Argyle (2016), Argyle et al. (2013) Hogge (2008). The design, analysis, fabrication, and testing of a tail-sitter are explained in Hogge (2008). Several companies, including Martin UAV and Aerovel, are developing tail-sitter UAVs for military and civil applications. Martin UAV developed a ducted fan tail-sitter called the Vertical Bat (V-Bat). A mathematical model of the V-Bat and its control architecture are given in Argyle et al. (2013). In addition, Argyle (2016) explains the quaternion-based attitude error kinematics for the V-Bat and presents a hybrid backstepping controller. In Krogh (2009), the linear and nonlinear control laws for hovering and horizontal-to-vertical transitions on agile aircraft such as the

YAK-54 aircraft and Aerovel's Flexrotor are discussed in detail.

The motivation behind this thesis is to fabricate, model, and control a spherical tail-sitter developed by the Korean Aerospace University Young Bae Lee (2012). This spherical tail-sitter has one propeller and four control surfaces. The propeller generates lift force during vertical take-off and hovering. The control surfaces are used to maneuver the UAV in the plane parallel to the ground and to generate lift during forward flight. In this thesis, an attitude controller for a similar spherical tail-sitter is designed and implemented on a UAV.

The UAV design used in this thesis is based on the spherical tail-sitter designs in Loh and Jacob (2013) and Malandrakis et al. (2016). In Loh and Jacob (2013), a mathematical model for the vehicle is derived and a commercial off-the-shelf (COTS) PID controller is implemented. For a tail-sitter vehicle with a similar design, Sun et al. (2013) designed and implemented a backstepping control method to stabilize the vehicle.

In this thesis, the first principle method in Greenwood (1988) is used to derive the mathematical model for the designed UAV. The different types of attitude kinematics used in this project are discussed in Shuster (1993), and the equations for several attitude kinematics are derived in Wie (1998). In Markley and Crassidis (1996), the attitude estimation for the Modified Rodrigues Parameters (MRP) are discussed. To estimate the moment of inertia in the model, a step-by-step procedure for a bifilar pendulum experiment is given in Habeck and Seiler (2016). The mathematical background behind the vector and matrix (system) norms was introduced in Mackenroth (2013), Rodriguez (2004). In Skogestad and Postlethwaite (2007) the analysis and design procedure for the classical robust control design for SISO and MIMO systems is explained in detail. The weight selection for the mixed sensitivity H_∞ control design is discussed in Skogestad and Postlethwaite (2007), Lundström et al. (1991).

The Linear Matrix Inequalities (LMI) for formulating the H_∞ and H_2 control design problems are given in Duan and Yu (2013), Boyd et al. (1997). In Lofberg (2004), the YALMIP toolbox for solving LMIs was introduced for MATLAB. Several nonlinear control techniques for tail-sitters are discussed in Dixon et al. (2013), Sun et al. (2013), and Argyle (2016). In this thesis, the Sum-of-Squares (SOS) toolbox developed at Caltech Prajna et al. (2002) is used to solve the polynomial optimization problem for nonlinear control design techniques. The control gains for the nonlinear control laws introduced in Tsiotras (1995) are optimized through the SOS method.

1.2 Outline of the Thesis

This thesis describes the fabrication, modeling, and control of a small spherical tail-sitter UAV. First, the vehicle was designed in SolidWorks software, and based on that design, the parts were fabricated using a laser cutter and 3D printer. The components used for developing the UAV and the experimental setup used are described in Chapter 2. In Chapter 3, the complete nonlinear dynamical model of the UAV is derived under the assumption that the motor and control surfaces are rigid bodies attached to the UAV. In several papers like Loh and Jacob (2013), this assumption was not taken into consideration for modeling, and hence the effects of those components on the UAV were neglected. By incorporating these analyses here, the filter design for the actuator and bandwidth are designed to obtain good performance in hovering. One of the main contributions of this thesis is the derivation of the attitude kinematics using Modified Rodrigues Parameters to avoid the singularity in the kinematics equation, as explained in Chapter 4.

In Chapter 5, the coefficients of the vehicle model such as the moment of inertia, aerodynamic coefficients, and motor constants are estimated to a high degree of accuracy. Since the moment of inertia is estimated using the bifilar pendulum ex-

periment, the values are more accurate than an estimation from SolidWorks. The motor constants are found using the experimental setup developed by RCbenchmark rcb (2016). Finally, the aerodynamic coefficients are calculated in ANSYS software. Using these parameters in the nonlinear model, the trim condition and linearized model are derived in Chapter 6.

To establish the mathematical foundations of robust control, several types of norms are briefly discussed in Chapter 7. In Loh and Jacob (2013) and Sun et al. (2013), the controller design for the tail-sitter used nominal PID control and back-stepping, respectively. There is almost no literature on robust control design for a spherical tail-sitter. The LMI techniques in Duan and Yu (2013) and Boyd et al. (1997) are for optimal H_∞ and H_2 control design and not for the mixed sensitivity problem. This control problem leads to a controller of higher order, which makes it difficult to implement. To overcome these limitations, the mixed sensitivity full-state feedback control problem is derived using LMI techniques and the robust control design techniques are explained in Chapter 8. Generally a gain scheduling method is used to control a system at different equilibrium state. In Chapter 9, a nonlinear control design is used to control the attitude for the transition between horizontal and vertical flight. Given the aforementioned limitations, several main contributions of this thesis are:

- robust control design for input and output disturbances within the controller bandwidth for a spherical tail-sitter UAV;
- formulation of a mixed sensitivity problem for a full state feedback controller using LMIs;
- nonlinear controller design for horizontal-to-vertical flight transition of a spherical tail-sitter UAV, developed using the Sum-of-Squares method.

Experiments were conducted in which the robust controller was implemented on the vehicle using the Robot Operating System (ROS) environment in the Beagle-Bone single-board computer. The flight control stack was developed from scratch for flexible controller implementation. A Kalman filter was programmed to run in a parallel thread to improve the state estimation. The simulation results of the linear and nonlinear control methods and the experimental result of the linear control are illustrated in Chapter 10. Conclusions and future possible directions of the work in this thesis are briefly explained in Chapter 11.

Chapter 2

EXPERIMENTAL SETUP

In this chapter, we discuss the design, fabrication, and experimental setup for the tail-sitter UAV. The basic design of the UAV used in this thesis is taken from Loh and Jacob (2013). The design was recreated in SolidWorks and fabricated based on that model. This chapter also explains the selection of the electronic and mechanical components of the UAV.

2.1 Design

The spherical tail-sitter UAV has a single propeller and four control surfaces. During VTOL, the propeller generates the lift force and the control surfaces are used to maneuver in the plane parallel to the ground. After vertical to horizontal transition, the propeller is used for forward motion and lift is produced by two control surfaces. The figure below shows the completed UAV.

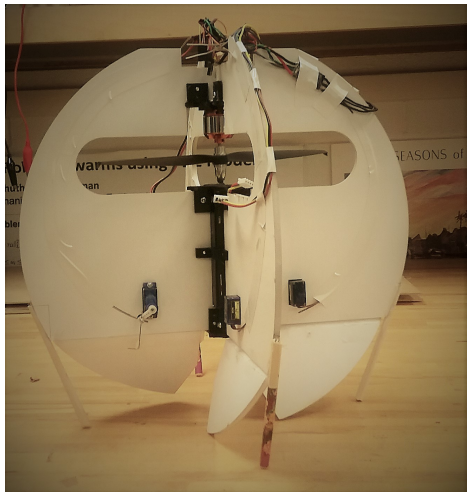
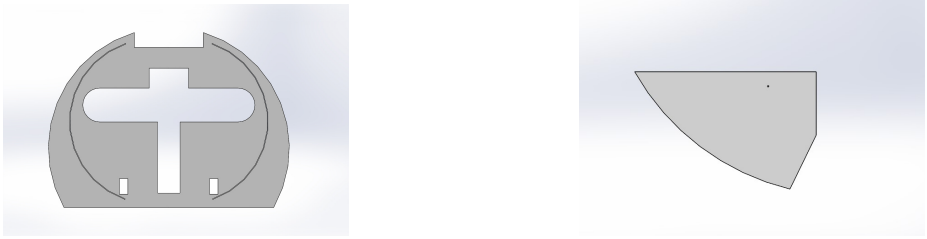


Figure 2.1: Fabricated spherical tail-sitter UAV

First, all the parts of the vehicle were designed and assembled in SolidWorks. The motor and propeller were approximately modeled based on the commercially available components. The spherical structure was formed by two disks that were placed perpendicular to each other. The disks and control surfaces were designed in SolidWorks for fabrication. Additionally, to fasten the battery and motor to the disks, individual lock were designed for 3D printing. These components are shown below.



(i) UAV body (ii) Control surface

Figure 2.2: UAV components: UAV body (i) and Control Surface (ii).

(i) Motor lock (ii) Battery lock

Figure 2.3: Motor lock (i) and Battery lock (ii).

Finally, the components were assembled in SolidWorks as per the design in Loh and Jacob (2013). The final assembled model in SolidWorks is shown in Fig. 2.4.

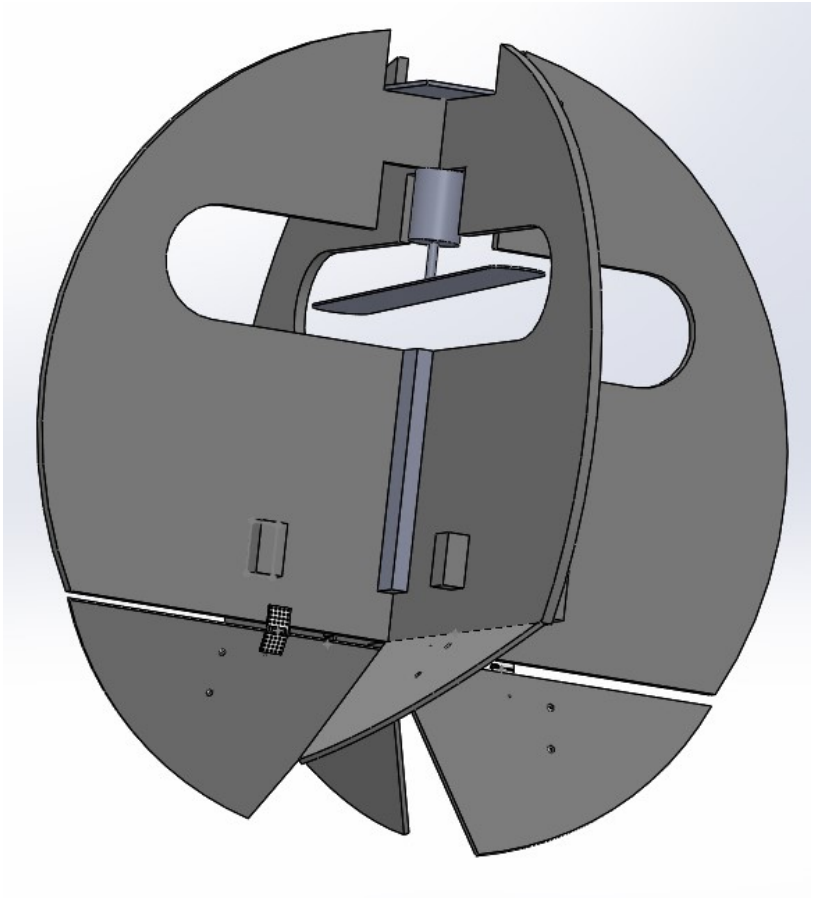


Figure 2.4: SolidWorks design of UAV

2.2 Experimental Setup

2.2.1 Components

These are classified into two groups: body components and electronics. The body components consist of foam board, motor lock, battery lock and control horns. The electronics components are a brushless motor with $9 * 4.5$ inch propeller, Electronic Speed Controller (ESC), mini servos, battery, and microcontroller board.

First, the approximate weight of the body was chosen to be less than $500g$. In order to have VTOL ability, the propeller has to generate a thrust at least 1.2 times

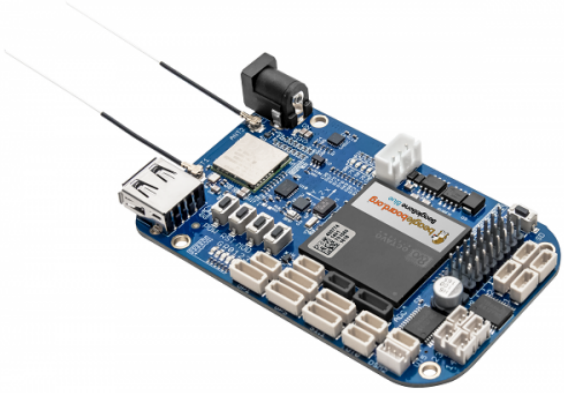


Figure 2.5: BeagleBone Blue Erik Welsh (2016)

the body weight for good maneuvering capability during take-off and landing. The motor and the propeller characteristics for different motors, propellers, ESCs, and batteries can be found in Mueller (2018). Based on Mueller (2018), we found that the $1800kV$ brushless motor with $9 * 4.5$ inch propeller is capable of producing a thrust of approximately 1.2 times the weight. A low-weight ESC with $30A$ was used to control the brushless motor. For actuating control surfaces, an $FS - 90$ Mini servo was used because it weighs only $9g$ and produces enough torque to rotate. The control horns and rods were used to connect the each servos and to its control surface. All the wires from the servos, motor, and battery were connected to the microcontroller board.

Over the past decade, the Beaglebone microcontroller board has become widely popular for low-cost robotics applications Cleva et al. (2012). Recently, BeagleBoard.org released BeagleBone Blue with a Linux-based real-time operating system (RTOS). It consists of an Octavo Systems OSD3358 microprocessor with WiFi, an IMU/barometer, power regulation, H-bridges, and discrete connectors for 4 DC motors and encoders, 8 servos/ESCs, and other ports for additional peripherals in embedded applications. It is fast, streamlined, affordable, and open-source. It is compatible with ROS, Ardupilot, MATLAB, and OpenCV Erik Welsh (2016).

2.2.2 Fabrication

The SolidWorks models in Figure 2.2i and 2.2ii were cut using a laser cutting machine, and the other models in Figures 2.3i and 2.3ii were fabricated using a 3D printer. The battery was placed at the center UAV body, and the disks were tightened using a battery lock and some clamps. The motor with the propeller was fixed to the motor base and tightened to the support structure.

The servos were placed inside the corresponding holes of the support stucture and glued to the support structure. The servo wires were threaded through the support structure to the microcontroller board, and then the control horns from the servos were extended using small rods that were glued to the control surfaces. The microcontroller was taped to the top of the UAV body with sponge as dampeners at the bottom. Finally, small landing sticks were glued to the bottom of the disks for support when the UAV is landed.

A test stand was built for initial flight testing of the spherical tail-sitter UAV. The figure below shows the test stand with the UAV.

Figure 2.6: Test stand for UAV

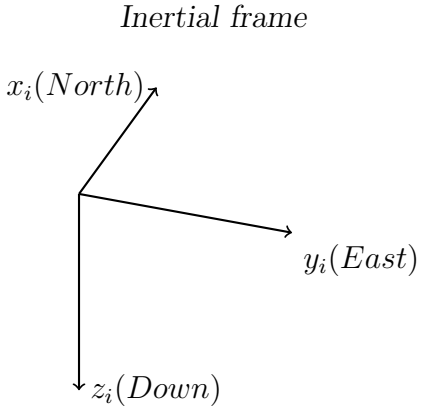
Chapter 3

MATHEMATICAL MODELING OF UAV DYNAMICS

3.1 Coordinate Frames

For this UAV, we need several frames to define the mathematical model. The frames used for the derivation are listed below:

- Inertial frame
- Body frame
- Hinge frame
- Motor Hinge frame
- Stability frame
- Wind frame



The earth-fixed frame or inertial frame is defined with the takeoff location as the origin. The frame is illustrated in the figure above and it is denoted by (x_i, y_i, z_i) .

Since it is a fixed frame, the north direction is defined as x_i, y_i is defined as the east direction, and based on the right-hand rule, z_i is the down direction.

Body Frame

The body frame is chosen based on the Inertial-Measurement Unit (IMU) alignment on the body. The direction from the center of gravity (CG) to the tail is defined as the z_b axis, and the right side direction of the IMU is taken as the positive y_b axis. The final direction x_b is derived from the right-hand rule. The figure below shows the body frame.

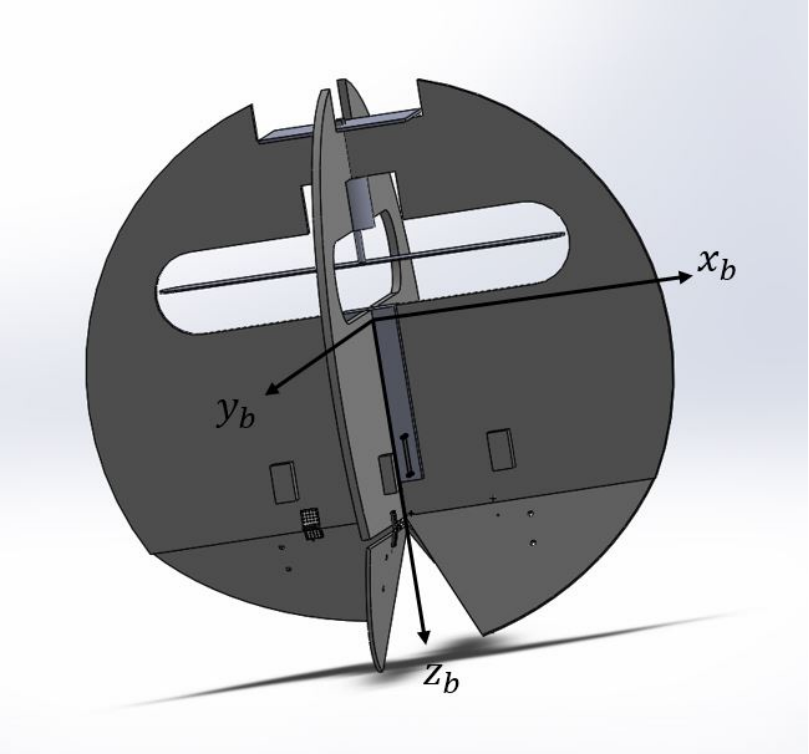


Figure 3.1: Body frame

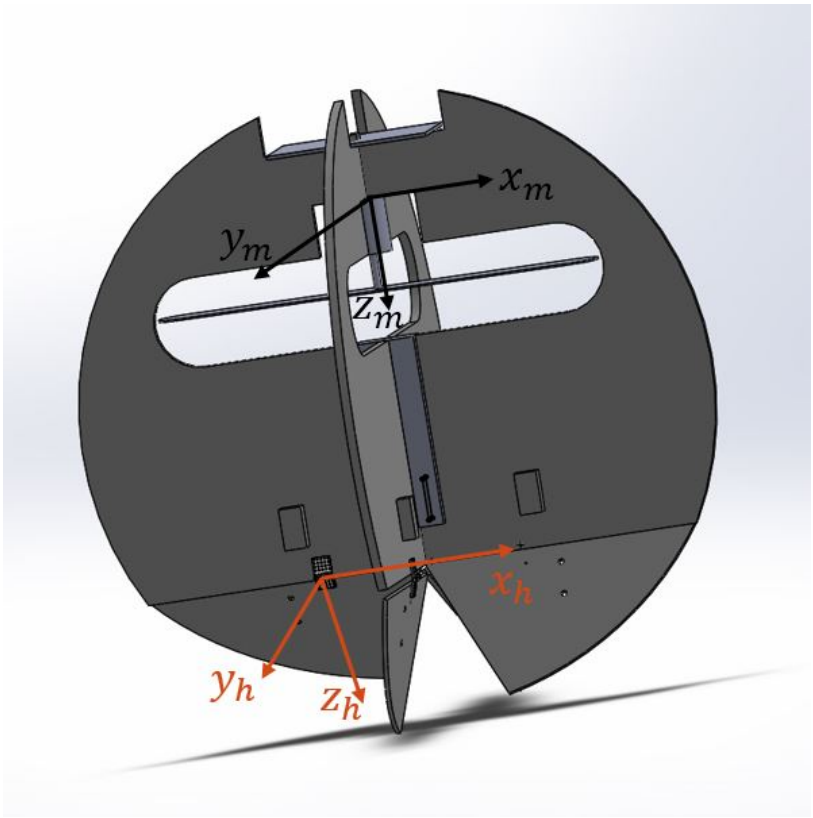


Figure 3.2: Hinge and Motor Hinge Frames

Hinge frame

Each control surface is rotated about its axis; hence, we need a frame to define the forces and moments that it applies on the UAV. The SolidWorks model in the figure 3.2 shows the hinge point for the control surface, and the axes are defined as (x_h, y_h, z_h). The hinge frame is defined parallel to the body frame when the control surface is not rotated, with the hinge point as the origin.

Motor Hinge frame

Similar to the Hinge frame, the motor is attached to motor hinge point and the hinge frame axes are defined as (x_m, y_m, z_m) with motor hinge point as the origin.

This frame is also shown in Figure 3.2.

Stability frame

The stability frame has the same origin as the body frame. The stability frame y_s axis is fixed to the body frame y_b axis and the z_s axis is aligned with the relative wind velocity. Basically, the stability frame is a rotation of the body frame at an angle α about the y_b axis. The angle-of-attack α is defined as the angle between the relative wind velocity and the z_b axis.

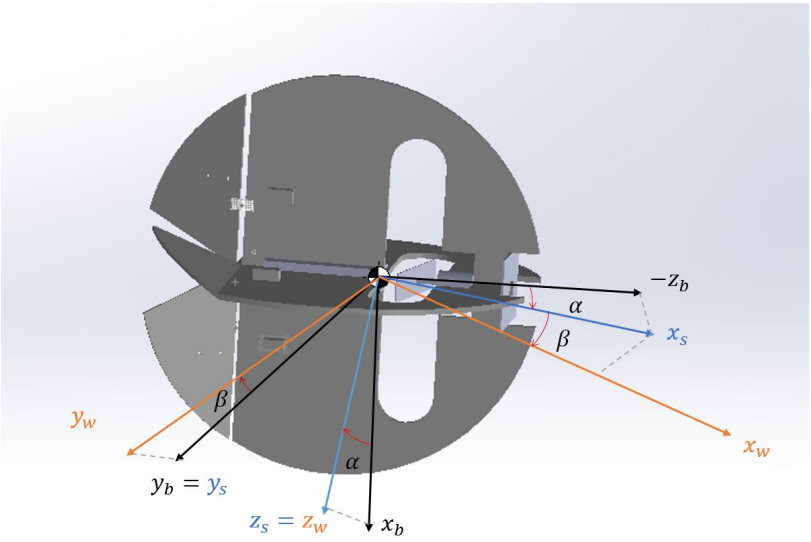


Figure 3.3: Stability and Wind Frames

Wind Frame

The wind frame is the rotation of the stability frame by an angle β about x_s. We define the side slip angle β as the angle between the relative wind velocity and the z_s axis. The stability frame and the wind frame for an aircraft are shown in Fig.3.3.

3.2 Rotation Matrices

The rotation matrix $\tilde{R}$ plays an important role in mathematical modeling of the UAV dynamics. It is used to transform between the different frames of reference. This matrix $\tilde{R}$ has special properties; it is:

- a square matrix with determinant equal to one, i.e., $|\tilde{R}| = 1$
- an orthogonal matrix, i.e., $\tilde{R}^T = \tilde{R}^{-1}$

Note: We use tilde $\sim$ on R to denote the rotation matrix $\tilde{R}$.
$\tilde{R}_A^B$ is the rotation matrix from frame A to B

The rotation matrices are in the **Special Orthogonal** group $SO(n)$, because they have the above properties.

The effect of a rotation matrix for a 2D rotation of a coordinate frame about an angle θ is illustrated below. This rotation can also be considered as a 3D rotation of the frame about the axis perpendicular to the page.

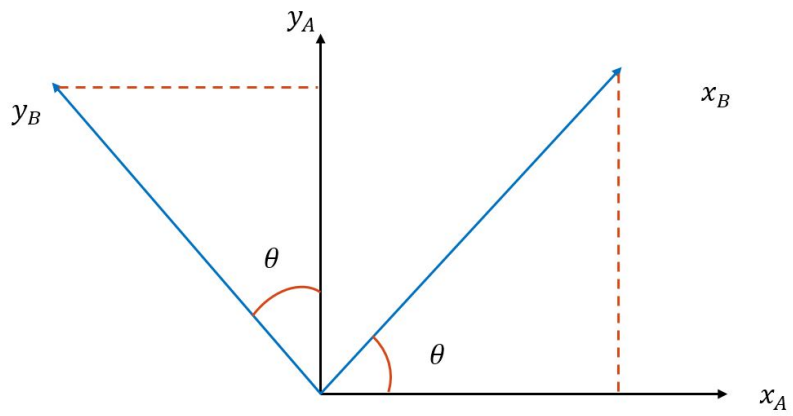


Figure 3.4: Rotation of a coordinate frame in 2D

The rotation matrix from frame A to frame B is given by:

$$\tilde{R}_A^B = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

Euler Angles

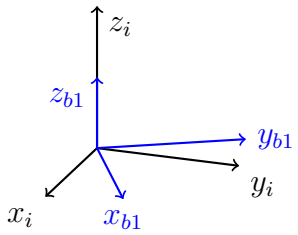
For 3D rotations, there are several techniques used to represent the rotation matrix. The Euler angle method is one of the most widely used methods to convert from one frame to another because of its simplicity. In this method, the 3D rotations are defined by three successive sets of rotations about the three axes.

This method leads to numerous possible representations. Most widely used is $z - y - x$ or $3 - 2 - 1$ representation, i.e., the first axis of rotation is the z axis, followed by the y axis, and finally the x axis.

Inertial Frame to Body Frame

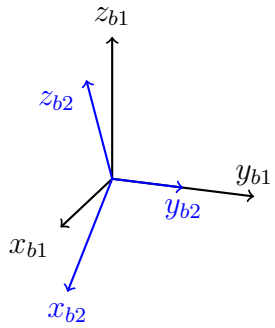
We use Euler's method to move from the inertial frame to the body frame. The first step is to rotate the frame i about z_i at an angle ψ. The resultant frame is $b1$ and the rotation matrix $\tilde{R}_i^{b1}(\psi)$ is defined as:

$$\tilde{R}_i^{b1}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$



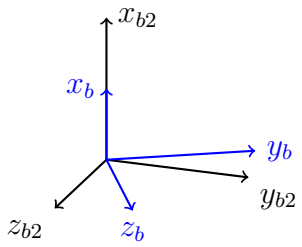
Secondly, we rotate the frame $b1$ about y_{b1} at an angle θ. $\tilde{R}^{b2}_{b1}(\theta)$ is the rotation matrix from the $b1$ frame to the resultant $b2$ frame, defined as:

$$\tilde{R}^{b2}_{b1}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.3}$$



Finally, to get the body frame, we rotate the $b2$ frame by an angle ϕ about the axis x_{b2}. $\tilde{R}^{b}_{b2}(\phi)$ is the rotation matrix from the $b2$ frame to the b frame, defined as:

$$\tilde{R}^{b}_{b2}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3.4}$$



Based on the property of the rotation matrix, the transformation from inertial frame to body frame is given by:

$$\begin{aligned}\tilde{R}_i^b(\phi,\theta,\psi) &= \tilde{R}_{b2}^b(\phi)\tilde{R}_{b1}^{b2}(\theta)\tilde{R}_i^{b1}(\psi) \qquad (3.5)\\ &= \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & \sin\phi\\ 0 & -\sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}\cos\theta & 0 & -\sin\theta\\ 0 & 1 & 0\\ \sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\\ \tilde{R}_i^b &= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta\\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta\\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta\end{bmatrix} \qquad (3.6)\end{aligned}$$

where, $c\theta \equiv \cos\theta$ and $s\psi \equiv \sin\psi$. Moreover, the rotation matrix from body frame to inertial frame is just the transpose of 3.6,

$$\tilde{R}_b^i = \left(\tilde{R}_i^b\right)^T \qquad (3.7)$$

Body Frame to Hinge Frame

The control surface along the positive y_b axis is rotated by an angle δ_1 about the y_h axis. The rotation matrix $\tilde{R}_b^h(0,\delta_1,0)$ is defined as:

$$\tilde{R}_b^h(0,\delta_1,0) = \begin{bmatrix}\cos\delta_1 & 0 & -\sin\delta_1\\ 0 & 1 & 0\\ \sin\delta_1 & 0 & \cos\delta_1\end{bmatrix} \qquad (3.8)$$

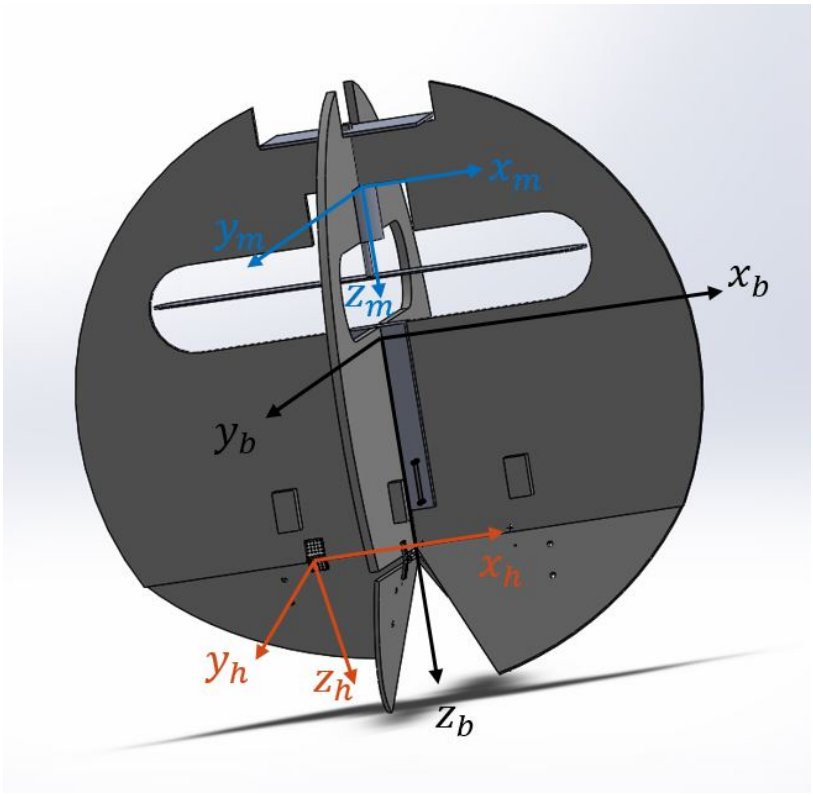


Figure 3.5: Reference frames

Body Frame to Motor Hinge Frame

The motor is rotated by an angle Θ_p about the z_m axis and the rotation matrix from the body frame to the motor frame is defined by $\tilde{R}_b^m(0, 0, \Theta_p)$:

$$\tilde{R}_b^m(0, 0, \Omega_p) = \begin{bmatrix} \cos\Theta_p & \sin\Theta_p & 0 \\ -\sin\Theta_p & \cos\Theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The Fig. 3.5 shows the body frame, Hinge frame and Motor Hinge frame attached to the body.

Body Frame to Stability Frame

The rotation matrix from body frame to stability frame is $\tilde{R}_b^s(0,0,\alpha)$:

$$\tilde{R}_b^s(0,0,\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.10}$$

Body Frame to Wind Frame

The rotation matrix from body frame to wind frame Fig. 3.3 is $\tilde{R}_b^s(0,\beta,\alpha)$:

$$\tilde{R}_b^w(0,\beta,\alpha) = \tilde{R}_s^w(0,\beta,0)\tilde{R}_b^s(0,0,\alpha) \tag{3.11}$$

3.3 Translational Dynamics

First, we define the vectors that are needed for the UAV dynamical model. The figure 3.6 shows the vector from body CG to the hinge point and the vector from hinge point to the control surface.

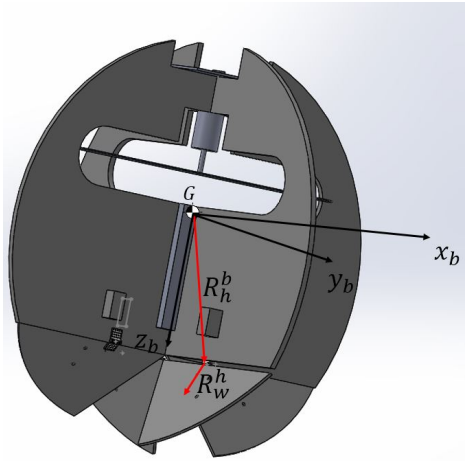


Figure 3.6: Vector from body CG to control surface CG

$R_w^h = \begin{bmatrix} 0 & 0 & d_w \end{bmatrix}^T$ is the vector from hinge to control surface CG w.r.t hinge frame

R_w^b is the vector from hinge to control surface CG w.r.t body frame

$R_h^b = \begin{bmatrix} 0 & d_y & d_z \end{bmatrix}^T$ is the vector from body CG to hinge w.r.t body frame

R_c^b is the vector from body CG to control surface CG w.r.t body frame

Note that:

$$R_w^b = \tilde{R_c^h}\ R_w^h$$
$$R_c^b = R_h + R_w^b \tag{3.12}$$

Differentiating 3.12, we get

$$V_c^b = V_h^b + V_w^b \tag{3.13}$$

where,

V_c^b is the absolute velocity of the control surface CG in the body frame

V_h^b is the absolute velocity of the hinge point in the body frame

V_w^b is the absolute velocity of the control surface in the body frame

The angular velocities are defined as,

$$\omega_c^h = \tilde{R}_b^h\ \omega_b^b + \omega_w^h \tag{3.14}$$

where,

$\omega_b^b = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is the angular velocity of the body in the body frame

$\omega_w^h = \begin{bmatrix} 0 & \Omega_w & 0 \end{bmatrix}^T$ is the angular velocity of the control surface in the hinge frame

ω_c^h is the total angular velocity of the control surface in the hinge frame

Expanding the velocities in 3.13 separately, we obtain:

$$V_h^b = V_b^b + \frac{\partial R_h^b}{\partial t} + (\omega_b^b \times R_h^b)$$
$$V_h^b = V_b^b + (\omega_b^b \times R_h^b) \tag{3.15}$$
$$V_w^h = \frac{\partial R_w^h}{\partial t} + (\omega_c^h \times R_w^h)$$
$$V_w^b = (\omega_b^b \times R_w^b) + \tilde{R}_h^b(\omega_w^h \times R_w^h) \tag{3.16}$$

where, $V_b^b = \begin{bmatrix} u & v & w \end{bmatrix}^T$, is the absolute velocity of the body CG in the body frame

Differentiating 3.15 and 3.16 to get acceleration in the body frame:

$$\frac{dV_h^b}{dt} = \frac{dV_b^b}{dt} + \dot{\omega}_b^b \times R_h^b + \omega_b^b \times (\omega_b^b \times R_h^b) \tag{3.17}$$

$$\frac{dV_w^b}{dt} = \tilde{R}_h^b(\dot{\omega}_c^h \times R_w^h + \omega_c^h \times (\omega_c^h \times R_w^h)) \tag{3.18}$$

$$\frac{dV_b^b}{dt} = a_b^b + (\omega_b^b \times V_b^b) \tag{3.19}$$

where, $\dot{\omega}_b^b = \begin{bmatrix} \dot{p} & \dot{q} & \dot{r} \end{bmatrix}^T$ is the angular acceleration of the body in the body frame, $\dot{\omega}_c^h$ is the angular acceleration of the control surface in the hinge frame, and $a_b^b = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} \end{bmatrix}^T$ is the acceleration of the body CG in the body frame.

Similarly, the dynamics of the motor $\frac{dV_m^b}{dt}$ attached to the body can be derived. The translational dynamics of the system are written as,

$$F^b = M_s \frac{dV_b^b}{dt} + M_c \left(\frac{dV_h^b}{dt} + \frac{dV_w^b}{dt} \right) + M_m \left(\frac{dV_{h_m}^b}{dt} + \frac{dV_m^b}{dt} \right) \tag{3.20}$$

where,

F^b is the total force that acts on the body in the body frame,

M_s is the mass of the body without control surfaces and motor,

M_c is the mass of the control surface,

M_m is the mass of the motor with propeller,

$V_{h_m}^b$ is the absolute velocity of the motor hinge point in the body frame, and

V_m^b is the absolute velocity of the motor in the body frame

Substituting the equations 3.17, 3.18 and 3.19 3.20, the complete translational

dynamics can be derived:

$$F = M_s(a_b^b + \omega_b^b \times V_b^b) + M_c[a_b^b + (\omega_b^b \times V_b^b) + \dot{\omega}_b^b \times R_h^b + \omega_b^b \times (\omega_b^b \times R_h^b) + \tilde{R}_h^b(\dot{\omega}_c^h \times R_w^h + \omega_c^h \times (\omega_c^h \times R_w^h))] + M_m[a_b^b + (\omega_b^b \times V_b^b) + \dot{\omega}_b^b \times R_{h_m}^b + \omega_b^b \times (\omega_b^b \times R_{h_m}^b) + \tilde{R}_m^b(\dot{\omega}_m^m \times R_m^m + \omega_m^m \times (\omega_m^m \times R_m^m))] \tag{3.21}$$

$$F = M_b((a_b) + (\omega_b \times V_b)) + M_c[\dot{\omega}_b \times R_h + \omega_b \times (\omega_b \times R_h) + R_w^b(\dot{\omega}_c \times R_w + \omega_c \times (\omega_c \times R_w))] + M_m[\dot{\omega}_b \times R_{h_m} + \omega_b \times (\omega_b \times R_{h_m}) + R_m^b(\dot{\omega}_m \times R_m + \omega_m \times (\omega_m \times R_m))] \tag{3.22}$$

where, M_b is the total mass of the body ($M_b = M_s + M_m + M_c$),
$R_{h_m}^b = \begin{bmatrix} 0 & 0 & -dm_z \end{bmatrix}^T$ is the vector from body CG to the motor hinge point in body frame,
$R_m^m = \begin{bmatrix} 0 & 0 & m_z \end{bmatrix}^T$ is the vector from the motor hinge point to the motor CG in the motor frame,

3.4 Rotational Dynamics

For rotational dynamics, it is easier to derive the equations in the body frame. The angular momentum of a rigid body with respect to a point is derived in Greenwood (1988). Consider a control surface as a rigid body connected to the body. The total angular momentum is defined as the vectorial addition of each angular momentum. The angular momentum of the control surface is derived as in Greenwood (1988), and the total angular momentum is just the vectorial addition of the body and control surface angular momenta, given by

$$H_c^b = M_c R_h^b \times \bar{V}_w^b + M_c(\tilde{R}_h^b R_w^h) \times \bar{V}_h^b + (\tilde{R}_h^b)(H_w^h) \tag{3.23}$$

where,

H_c^b is the angular momentum of the control surface w.r.t to body CG,

H_w^h is the angular momentum of the control surface w.r.t to its CG,

$\bar{V}_h^b$ is the velocity of hinge point with body CG as origin and written in body frame,

$\bar{V}_w^h$ is the velocity of control surface with body CG as origin and written in hinge frame.

Differentiating 3.23 gives,

$$\frac{dH_c^b}{dt} = \tilde{R}_h^b \left(\frac{dH_w^h}{dt}\right) + M_c R_h \times \tilde{R}_h^b \left(\frac{d\bar{V}_w^h}{dt}\right) + M_c(\tilde{R}_h^b R_w^h) \times \left(\frac{d\bar{V}_h^b}{dt}\right) \tag{3.24}$$

The angular momentum of the control surface about its CG is given by H_w,

$$H_w^h = I_w \omega_w^h \tag{3.25}$$

$$\frac{dH_w^h}{dt} = I_w \frac{\partial \omega_w^h}{\partial t} + \omega_w^h \times I_w \omega_w^h \tag{3.26}$$

where, I_w is the moment of inertia of each control surfaces.

The body angular momentum with respect to its CG is given by H_b,

$$H_b^b = I_b \omega_b^b \tag{3.27}$$

$$\frac{dH_b^b}{dt} = I_b \frac{\partial \omega_b^b}{\partial t} + \omega_b^b \times I_b \omega_b^b \tag{3.28}$$

where, I_b is the moment of inertia of the body.

The total angular momentum is the summation of 3.24 and 3.28,

$$\frac{dH_T^b}{dt} = \frac{dH_b^b}{dt} + \tilde{R}_h^b \left(\frac{dH_w^h}{dt}\right) + M_c R_h \times \tilde{R}_h^b \left(\frac{d\bar{V}_w^h}{dt}\right) + M_c(\tilde{R}_h^b R_w^h) \times \left(\frac{d\bar{V}_h^b}{dt}\right) \tag{3.29}$$

where H_T^b is the total angular momentum of the body.

In the above equation, the time derivatives of $\bar{V}_w^b$ and $\bar{V}_h^b$ are defined as follows:

$$\bar{V}_h^b = 0 \qquad \text{(hinge point is fixed in the body)} \tag{3.30}$$

$$\frac{d\bar{V}_h^b}{dt} = 0 \tag{3.31}$$

$$\bar{V}_w^h = \omega_w^h \times R_w^h \tag{3.32}$$

$$\frac{d\bar{V}_w^h}{dt} = \dot{\omega}_w^h \times R_w^h + \omega_w^h \times (\omega_w^h \times R_w^h) \tag{3.33}$$

3.5 Assumptions

Substituting 3.31 and 3.33 in 3.29, the force acts on the body due to the each control surfaces can be found:

$$M_c \frac{dV_c}{dt} = M_c d_w \begin{bmatrix} \dot{\Omega}_w \cos\delta - \Omega_w^2 \sin\delta \\ 0 \\ \dot{\Omega}_w \sin\delta - \Omega_w^2 \cos\delta \end{bmatrix} \tag{3.34}$$

The properties of the control surface are given below:

$$M_c = 0.001kg$$

$$d_w = 0.01m$$

where M_c is the mass of each control surface, and d_w is the vector from the hinge point to the control surface CG.

The operating conditions for each control surface are given below. The angular velocity and acceleration of the actuator conditions are chosen to minimize the dynamics of the control surface that act on the body in 3.34.

$$\Omega_w = 1 \ rad/s$$

$$\dot{\Omega}_w \approx 0$$

Based on the aerodynamics of the control surface which is explained in Chapter 5, the bound on the rotation of the each control surface is given by:

$$-20^o < \delta < 20^o \Rightarrow -0.349\ rad < \delta < 0.349\ rad$$

The above conditions help to constraint the bandwidth for control design.

3.6 Forces and Moments

In this section, we discuss the forces and moments that act on the UAV. Here, aerodynamic forces, propeller thrust force, and gravity force are the main external forces on the system. The moment due to rotation of the motor was described in the Rotational Dynamics section. First, we will discuss the propeller force.

Propeller Force

The propeller is mounted on the motor, which lies along the z_b axis of the body and is fixed to the body. Hence, the force created by the propeller is in the negative z_b direction. It is defined by,

$$F_p^b = \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \tag{3.35}$$

where F_p^b is the force generated by the propeller defined in the body frame, and T is the thrust developed by the propeller.

Gravity Force

Gravity is an inertial force that acts on the UAV center of gravity. Its direction is along the positive z_i axis. The translational dynamics are defined in the body frame; hence, the gravity force acting on the UAV in the body frame is:

$$F_g^i = \begin{bmatrix} 0 \\ 0 \\ M_T g \end{bmatrix} \tag{3.36}$$

$$F_g^b = \tilde{R}_i^b F_g^i = \begin{bmatrix} -M_T g \sin\theta \\ M_T g \cos\theta \sin\phi \\ M_T g \cos\theta \cos\phi \end{bmatrix} \tag{3.37}$$

Aerodynamic Forces and Moments

The UAV's maneuvering capability along the x_b and y_b directions depend only on the aerodynamic forces and moments. Generally, the aerodynamic force on each control surface is calculated using the formula

$$\begin{aligned} F_A &= (\text{dynamic pressure}) \times (\text{surface area of the control surface}) \times (\text{aerodynamic coefficient}) \\ &= \frac{1}{2}\rho V_a^2 S C_a \end{aligned} \tag{3.38}$$

where ρ is the density of the air ($\rho = 1.225\ kg/m^3$)

V_a is the speed of air (m/s)

S is the control surface area (m^2)

C_a is the aerodynamic coefficient, which depends on the body geometry (dimensionless)

Aerodynamic forces are classified into two types:

- Lift force L, which acts perpendicular to the relative wind direction
- Drag force D, which acts in the direction of the relative wind direction

These forces are defined as:

$$L = \frac{1}{2}\rho V_a^2 S C_L \tag{3.39}$$

$$D = \frac{1}{2}\rho V_a^2 S C_D \tag{3.40}$$

where C_L is the coefficient of lift and C_D is the coefficient of drag.

Similarly, the aerodynamic moment is calculated by the formula,

$$m = \frac{1}{2}\rho V_a^2 S \bar{c} C_m \tag{3.41}$$

where m is the momentum generated due to aerodynamic forces, $\bar{c}$ is the length of the chord of the airfoil, and C_m is the coefficient of momentum.

Consider the control surface along the positive y_b direction. The aerodynamic forces and moment generated on this surface are given by:

$$F_1 = \frac{1}{2}\rho V_a^2 S C_L \tag{3.42}$$

$$D_1 = \frac{1}{2}\rho V_a^2 S C_D \tag{3.43}$$

$$m_1 = \frac{1}{2}\rho V_a^2 S \bar{c} C_m \tag{3.44}$$

Here, the air flow is generated by the propeller. The air speed V_a produced by the propeller is calculated as:

$$T = \frac{1}{2}\rho V_a^2 S_p \tag{3.45}$$

$$V_a = \sqrt{\frac{2T}{\rho S_p}} \tag{3.46}$$

where T is the thrust developed by the propeller (N) and S_p is the area under the propeller (m^2).

Chapter 4

ATTITUDE KINEMATICS

In the previous chapter, we derived the Equation of Motion (EOM) for the spherical VTOL UAV. The translational kinematics is in absolute (inertial) frame and the dynamics is derived with respect to body frame. Hence we need a rotational kinematics of the model to complete the mathematical model. This chapter discusses the three common representations of rotational kinematics in aerospace. The Euler angles, Quaternions and Modified Rodrigues Parameters are explained in detail.

4.1 Introduction

The temporal derivative of a particle in the inertial frame will have only translational kinematics. For a rigid body in 3D space, the temporal derivative depends on the rotating frame. Hence, it is very important to consider the attitude kinematics of the body to write the equation of motion. Basically, the attitude kinematics is the set of equation which maps the angular velocity of the vector in rotating frame to inertial frame Shuster (1993). There are different type of attitude representations to define a attitude of the rigid body in inertial frame. Hence, there are different functions that map the attitude representation in the inertial frame. The basic idea is to find the angular velocity of a vector in the absolute frame of reference. A quick derivation of that idea is given below Zhao (2016),

Differential Kinematics

Let $\tilde{R}$ be the rotation matrix in $SO(3)$ that rotates vector $\overrightarrow{r}$ to $\overrightarrow{r}^{rot}$. Then:

$$\begin{aligned}
\dot{\tilde{R}} &= \frac{d\tilde{R}}{dt} \\
\overrightarrow{r}^{rot} &= \tilde{R}\ \overrightarrow{r} \\
\text{Differentiating: } \frac{d}{dt}\left(\overrightarrow{r}^{rot}\right) &= \frac{d}{dt}\left(\tilde{R}\ \overrightarrow{r}\right) \\
&= \frac{d\tilde{R}}{dt}\overrightarrow{r} + \tilde{R}\frac{d}{dt}\left(\overrightarrow{r}\right) \\
\frac{\partial}{\partial t}\left(\overrightarrow{r}^{rot}\right) + \omega \times \overrightarrow{r}^{rot} &= \dot{\tilde{R}}\ \overrightarrow{r}^{rot} \\
-\left[\tilde{\omega}\right]\overrightarrow{r}^{rot} &= \dot{\tilde{R}}\ \overrightarrow{r}^{rot} \\
-\left[\tilde{\omega}\right]\tilde{R}\ \overrightarrow{r} &= \dot{\tilde{R}}\ \overrightarrow{r} \\
\left(\dot{\tilde{R}} + \left[\tilde{\omega}\right]\tilde{R}\right)\ \overrightarrow{r} &= 0
\end{aligned}$$

$$\implies \dot{\tilde{R}} = -\left[\tilde{\omega}\right]\tilde{R} \tag{4.1}$$

where, $[\tilde{\omega}]$ is a skew-symmetric matrix

$$\tilde{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

4.2 Euler Angles

In the last chapter, we discussed about the rotation matrix and the Euler angle representation of the absolute angles. The Euler angles in the model are taken as the unknown variables in the derivation of the dynamics. However, we just need to calculate the temporal derivatives of the Euler angles $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ from the body angular

rates $\begin{bmatrix} p & q & r \end{bmatrix}^T$. These equations are commonly represented as the rotational kinematics of the model. By following the convention used in Section 3.2, the rotational kinematics is derived Beard and McLain (2012). Consider the successive rotations defined as:

$A \xrightarrow{R_A^{I_1}(\psi)} I_1 \xrightarrow{R_{I_1}^{I_2}(\theta)} I_2 \xrightarrow{R_{I_2}^{B}(\phi)} B,$

where A is the Absolute frame, I_1 is the Intermediate frame 1, I_2 is the Intermediate frame 2, and B is the Body frame.

The angular velocity in each frame is defined as,

$A \to I_1 : \omega_A^{I_1} = \dot{\psi}\ z_{I_1}$

$I_1 \to I_2 : \omega_{I_1}^{I_2} = \dot{\theta}\ y_{I_2}$

$I_2 \to B : \omega_{I_2}^{B} = \dot{\phi}\ x_b$

Angular velocities in each frame can be added up to obtain the final angular velocity, which is given by the result below Beard and McLain (2012),

$$\omega_A^B = \omega_A^{I_1} + \omega_{I_1}^{I_2} + \omega_{I_2}^B \tag{4.2}$$

$$\implies p\ x_b + q\ y_b + r\ z_b = \dot{\psi}\ z_{I_1} + \dot{\theta}\ y_{I_2} + \dot{\phi}\ x_b \tag{4.3}$$

$$\implies \begin{bmatrix} p & q & r \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dot{\psi} \end{bmatrix} \begin{bmatrix} x_{I_1} \\ y_{I_1} \\ z_{I_1} \end{bmatrix} + \begin{bmatrix} 0 & \dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} x_{I_2} \\ y_{I_2} \\ z_{I_2} \end{bmatrix} + \begin{bmatrix} \dot{\phi} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \tag{4.4}$$

$$\implies \begin{bmatrix} p & q & r \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dot{\psi} \end{bmatrix} R_{I_2}^B(\phi) R_{I_1}^{I_2}(\theta) \begin{bmatrix} x_{I_1} \\ y_{I_1} \\ z_{I_1} \end{bmatrix} + \begin{bmatrix} 0 & \dot{\theta} & 0 \end{bmatrix} R_{I_2}^B(\phi) \begin{bmatrix} x_{I_2} \\ y_{I_2} \\ z_{I_2} \end{bmatrix} + \begin{bmatrix} \dot{\phi} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \tag{4.5}$$

Substituting the rotation matrices from Section 3.2, we get:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.6}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.7}$$

This formula is used to calculate the temporal derivative of the Euler angles ($3 \rightarrow 2 \rightarrow 1$) from the body angular rates. The temporal derivatives for Euler angles with a different set of rotations can be similarly derived.

The Euler angles give a unique orientation of the body, but a specific body orientation does not correspond to a unique Euler angle. This method can be easily interpretable by the user. It has three parameters to define the orientation in three dimensional space. Hence, it does not have any constraints for representing the orientation. It is computationally efficient and it is widely used in controller design for aircraft

Gimbal Lock: In general, we represent the rotational kinematics in Eucledian space $\mathbb{R}^n$, when they actually lie in $SO(n)$.Hence the attitude kinematics will always have at least one singularity in their representation. In Euler angles representation, this singularity occurs when the pitch angle in $(3-2-1)$ representation is at $(2n+1)\frac{\pi}{2}\ rad$ for an integer $n = 0, 1, 2....$ When we pitch the body up or down by $\pi/2\ rad$, then the two axes will be parallel, and hence we will lose one degree of freedom. This locking property is called "Gimbal Lock" Beard and McLain (2012).

4.3 Euler's Eigenaxis Rotation

Euler's eigenaxis rotation theorem states that Wie (1998):

Definition 1 *By rotating a rigid body about an axis that is fixed to the body and stationary in an inertial reference frame, the rigid-body attitude can be changed from any given orientation to any other orientation. Such an axis of rotation, whose orientation relative to both an inertial reference frame and the body remains unchanged throughout the motion, is called the Euler axis or eigenaxis.*

Each orientation of the rigid body can be represented as the body rotates by an angles equal to the magnitude of a vector about the direction of that vector.

$$\overrightarrow{n} = \theta\hat{n} \tag{4.8}$$

where $\overrightarrow{n}$ is the eigen vector, θ is the magnitude of $\overrightarrow{n}$, and $\hat{n}$ is the direction of $\overrightarrow{n}$

Derivation of Euler rotation formula

Let $\overrightarrow{r}$ be the vector to rotate about $\hat{n} = n_x\overrightarrow{x} + n_y\overrightarrow{y} + n_z\overrightarrow{z}$ by θ. Then:

$$\begin{aligned}
\overrightarrow{r} &= \overrightarrow{r_\parallel} + \overrightarrow{r_\perp} \\
\overrightarrow{r_\parallel} &= (\overrightarrow{r}\cdot\hat{n})\hat{n} \\
\overrightarrow{r_\perp} &= \overrightarrow{r} - \overrightarrow{r_\parallel} \\
&= \overrightarrow{r} - (\hat{n}\cdot\overrightarrow{r})\hat{n} = -\hat{n}\times(\hat{n}\times\overrightarrow{r}) \\
\text{since,}\quad \overrightarrow{r_\parallel}^{rot} &= \overrightarrow{r_\parallel} \\
|\overrightarrow{r_\perp}^{rot}| &= |\overrightarrow{r^{rot}}| \\
\overrightarrow{r_\perp}^{rot} &= (\cos\theta)\overrightarrow{r_\perp} + \sin\theta(\hat{n}\times\overrightarrow{r_\perp}) \\
\hat{n}\times\overrightarrow{r_\perp} &= \hat{n}\times(\overrightarrow{r} - \overrightarrow{r_\parallel}) \\
&= \hat{n}\times\overrightarrow{r}, \qquad \text{since} \qquad \hat{n}\times\overrightarrow{r_\parallel} = 0
\end{aligned}$$

$$
\begin{aligned}
\implies \overrightarrow{r}^{rot} &= \overrightarrow{r_{\parallel}}^{rot} + \overrightarrow{r_{\perp}}^{rot} \\
&= \overrightarrow{r_{\parallel}} + \cos\theta(\overrightarrow{r} - \overrightarrow{r_{\parallel}}) + \sin\theta(\hat{n} \times \overrightarrow{r}) \\
&= \overrightarrow{r}\cos\theta + (1 - \cos\theta)\overrightarrow{r_{\parallel}} + \sin\theta(\hat{n} \times \overrightarrow{r}) \\
&= \overrightarrow{r}\cos\theta + (1 - \cos\theta)(\overrightarrow{r} \cdot \hat{n})\hat{n} + \sin\theta(\hat{n} \times \overrightarrow{r}) \\
\overrightarrow{r}^{rot} &= [\cos\theta + (1 - \cos\theta)(\hat{n} \cdot \hat{n}) + (\sin\theta)\hat{n}\times]\overrightarrow{r}
\end{aligned}
$$

From the above equation the rotation matrix for $\overrightarrow{r}$ can be written as:

$$\tilde{R} = \cos\theta I + (1 - \cos\theta)nn^T + \sin\theta N \tag{4.9}$$

where I is the identity matrix,

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \qquad N = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}$$

Euler's eigenaxis rotation method is a three-parameter representation for three dimensional rotation and hence is not subject to constraints. However, it is not uniquely defined at $\theta = 0$.

4.4 Quaternions

Hamilton invented quaternions while searching for hyper-complex numbers representation in three-dimensional space. Wie (1998). The quaternions are defined below:

$$
\begin{aligned}
q_1 &= n_x \sin(\theta/2) \\
q_2 &= n_y \sin(\theta/2) \\
q_3 &= n_z \sin(\theta/2) \\
q_4 &= \cos(\theta/2)
\end{aligned}
$$

Defining $q = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$, we get:

$$q = \hat{n}(\sin\theta/2)$$
$$q_4 = \cos(\theta/2) \tag{4.10}$$

Quaternions have a constraint that: $q^T q + q_4^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$

The rotation matrix for the quaternions can be derived from the Euler rotation formula:

$$\begin{aligned}
\tilde{R} &= \cos\theta I + (1 - \cos\theta) nn^T + \sin\theta N \\
&= \left(2\cos^2\frac{\theta}{2} - 1\right) I + \left(2\sin^2\frac{\theta}{2}\right) nn^T - \left(2\sin\frac{\theta}{2}\sin\frac{\theta}{2}\right) N \\
\tilde{R} &= (q_4^2 - q^T q) I + 2qq^T - 2q_4 Q
\end{aligned} \tag{4.11}$$

where, $$Q = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

The Euler parameters are equivalent to the quaternions.

The differential kinematics for quaternions can be found by substituting the quaternion parameters into Eq. (4.1), yielding:

$$\begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \\ \dot{q_4} \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \tag{4.12}$$

Runge-Kutta methods are generally used for integrating the equation above to get the angles in quaternions. Finally, these values can be converted to Euler angles for easier interpretation.

4.5 Modified Rodrigues Parameters

Modified Rodrigues Parameters (MRPs) are triplets in $\mathbb{R}^3$ that are bijectively and rationally mapped to quaternions through stereographic projection Terzakis et al. (2018). The stereographic projection is a function that projects a sphere onto a plane from a projection point. Figure 4.1 shows a stereographic projection of a 3D sphere on a 2D plane Apostol (1974).

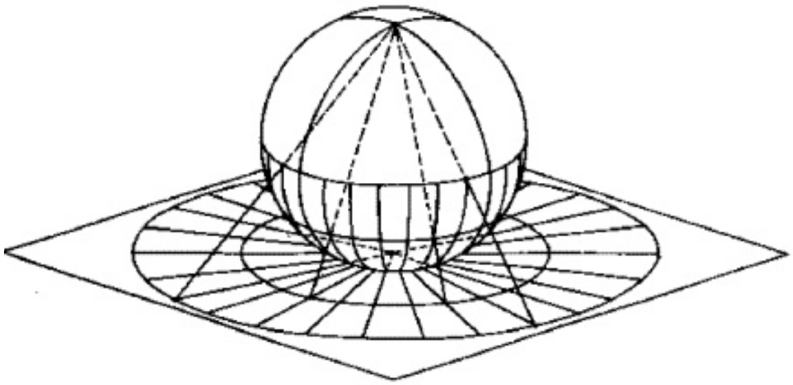


Figure 4.1: Stereographic Projection in 3D Apostol (1974)

Defining the quaternions as (q_1, q_2, q_3, q_4), the Modified Rodrigues Parameters are defined as,

$$\sigma_i = \frac{q_i}{1 + q_4} \qquad i = 1, 2, 3 \tag{4.13}$$

A derivation for the rotation matrix represented by MRP values is given below:

$$\sigma = \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \end{bmatrix}^T$$

$$\sigma^T\sigma = \sigma_1^2 + \sigma_2^2 + \sigma_3^2 = \frac{q_1^2 + q_2^2 + q_3^2}{(1+q_4)^2}$$

$$1 + \sigma^T\sigma = \frac{q_1^2 + q_2^2 + q_3^2 + q_4^2 + 1 + 2q_4}{(1+q_4)^2} = \frac{2}{1+q_4}$$

$$1 + \sigma^T\sigma = \frac{+1 + 2q_4 + q_4^2 - q_1^2 - q_2^2 - q_3^2}{(1+q_4)^2} = \frac{2q_4}{1+q_4}$$

From Eq. 4.11:

$$\tilde{R} = (q_4^2 + q^T q)I + 2qq^T + 2q_4 Q$$

Rearranging, we get

$$\tilde{R} = \frac{4}{(1+\sigma^T\sigma)^2}\left[\frac{(1+\sigma^T\sigma)}{4}I + 2(\sigma\sigma^T - \sigma^T\sigma I) + (1-\sigma^T\sigma)\Sigma\right] \tag{4.14}$$

where, $$\Sigma = \begin{bmatrix} 0 & -\sigma_3 & \sigma_2 \\ \sigma_3 & 0 & -\sigma_1 \\ -\sigma_2 & \sigma_1 & 0 \end{bmatrix}$$

The differential kinematics for Modified Rodrigues Parameters can be found by substituting Eq. 4.13 in Eq. (4.1), yielding: Markley and Crassidis (1996)

$$\begin{bmatrix} \dot{\sigma_1} \\ \dot{\sigma_1} \\ \dot{\sigma_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1+\sigma_1^2-\sigma_2^2-\sigma_3^2) & \sigma_1\sigma_2 - \sigma_3 & \sigma_1\sigma_3 + \sigma_2 \\ \sigma_1\sigma_2 + \sigma_3 & \frac{1}{2}(1+\sigma_2^2-\sigma_1^2-\sigma_3^2) & \sigma_2\sigma_3 - \sigma_1 \\ \sigma_3\sigma_2 - \sigma_2 & \sigma_3\sigma_2 + \sigma_1 & \frac{1}{2}(1+\sigma_3^2-\sigma_1^2-\sigma_2^2) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{4.15}$$

The equation above can be integrated using a Runge-Kutta method.

Chapter 5

ESTIMATION OF MODEL PARAMETERS

The nonlinear model of the UAV was derived in Chapters 3 and 4. For control design, we need values of the model parameters, which will define the flight characteristics of the UAV. In order to develop a good control design, we need good estimates of these model parameters. The PID control method does not require accurate values for the parameters. But with such design techniques, it would be difficult to tune the control gains to achieve desired performance characteristics; i.e., the controller would not be flexible. In most cases, manual tuning is required. Even though it is not possible to find the exact model parameters, it is desired to estimate close approximations to be control the UAV's flight characteristics. For the developed model, we need inertia parameters, motor constants, and aerodynamic coefficients. In this chapter, we will discuss the methods used to find the model parameters.

5.1 Inertia Parameters

For the attitude control problem, we need the inertia value of the UAV with respect to its center of gravity (CG). A bifilar pendulum experiment is used to estimate the moment of inertia of the UAV. In this method, the UAV is suspended with two strings, and the vehicle is allowed to rotate about its each axis of the body frame Habeck and Seiler (2016). The natural frequency is measured and the inertia about the each axis is calculated using the formula,

$$I = \frac{mgd^2}{4l\omega_n^2} \tag{5.1}$$

where m is the mass of the UAV (kg)

g is the acceleration due to gravity $(g = 9.81\ m/s^2)$

d is the distance between the strings (m)

l is the length of the strings (m)

ω_n is the natural frequency (rad/s)

5.1.1 Derivation for Bifilar Pendulum

The sketch in Fig. 5.1 shows part of the bifilar pendulum, used to derive the UAV's moment of inertia. The length of each string is defined as l and the distance between the strings is d. A torque $-\tau_i$ is applied about the z-axis to perturb the system. Here, θ is the perturbed angle and the restoring force acting on the string is F. Each string exerts a force of $\frac{1}{2}m_b g$ on the z-axis, and hence the restoring force is derived as Moloney (2006)

$$F = \frac{1}{2}m_b g \sin\theta \tag{5.2}$$

where, m_b is the mass of the body and g is the acceleration due to gravity

The torque is defined by,

$$\tau = F\frac{d}{2} \tag{5.3}$$

The total torque on both strings is,

$$\tau = Fd \tag{5.4}$$

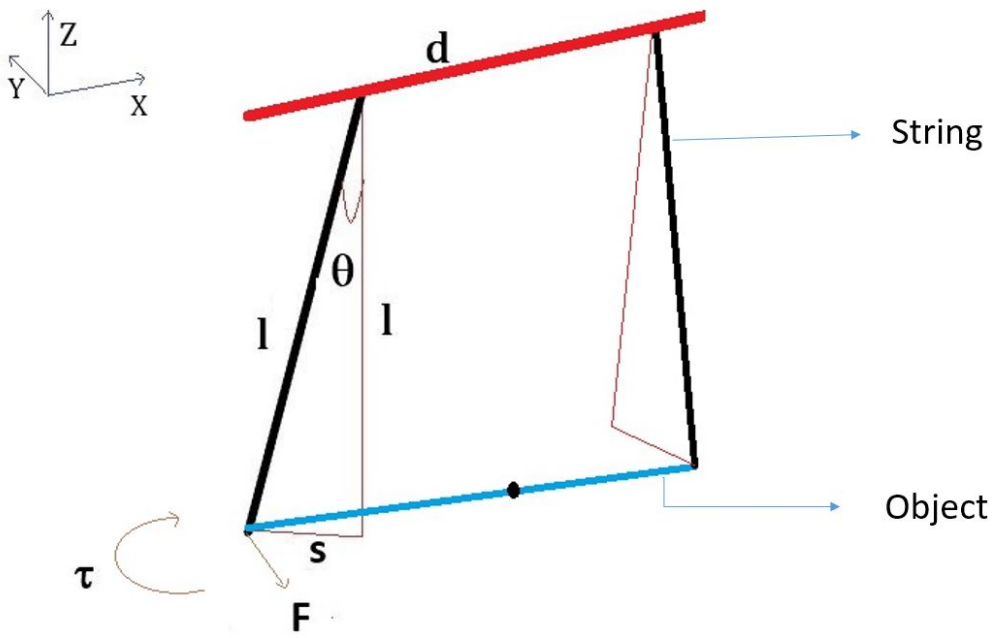


Figure 5.1: Sketch of Bifilar Experiment

From the sketch 5.1, we can write

$$\sin\theta = \frac{s}{l}$$

$$\implies \tau = \frac{Mgd}{2l}s \tag{5.5}$$

For small angles, $s = (d/2)\theta$,

$$\text{Eq. } 5.5 \implies \tau = \frac{Mgd^2}{4l}\theta \tag{5.6}$$

By substituting 5.6 into the rotational dynamics equation $I\alpha = -\tau$, where $\alpha = \frac{d^2\theta}{dt^2}$, we obtain Moloney (2006):

$$I\frac{d^2\theta}{dt^2} = -\frac{Mgd^2}{4l}\theta \tag{5.7}$$

$$\frac{d^2\theta}{dt^2} + \frac{Mgd^2}{4Il}\theta = 0 \tag{5.8}$$

The natural frequency ω_n is,

$$\omega_n = \sqrt{\frac{Mgd^2}{4Il}} \tag{5.9}$$

Solving the equation above for I, we get:

$$I = \frac{Mgd^2}{4l\omega_n^2} \tag{5.10}$$

The Fig. 5.2 shows the experimental setup for the calculating the moment of inertia. A rectangular plate is tied to the strings and the UAV is attached to the plate in the middle. The rectangular plate is used to mount the UAV; the plate moment of inertia is known. By using the parallel axis theorem, we can find the inertia of the UAV from the 5.10 formula.

Figure 5.2: Bifilar Experimental Setup

The IMU on the UAV is used to calculate the natural frequency as follows. The angular velocity of the body is measured using the IMU, and the frequency is calculated as

$$\omega_n = \frac{2\pi}{T},$$

where T is the time difference between two peaks in the angular velocity Figure 5.3.

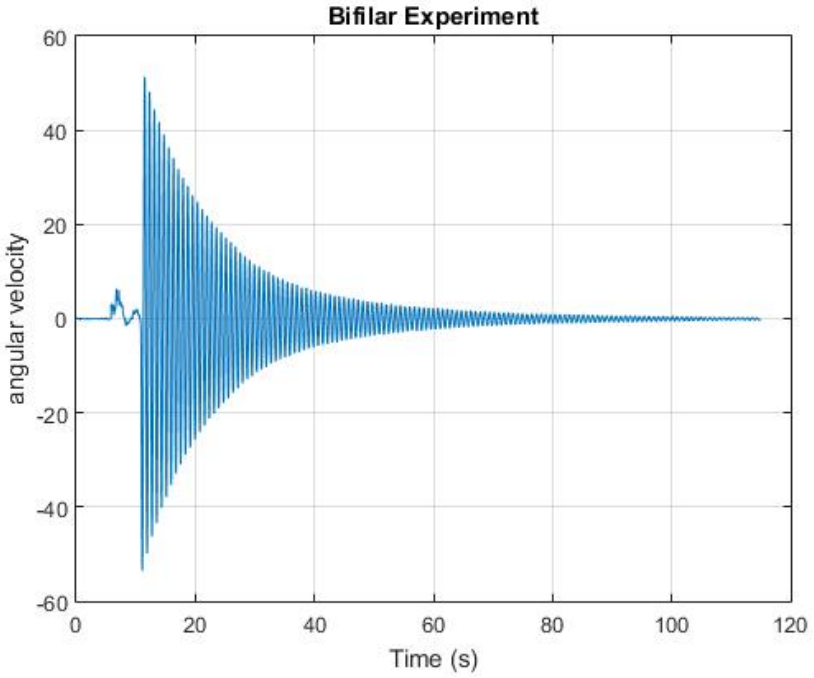


Figure 5.3: Bifilar Experiment Result: UAV Angular Velocity vs. Time

5.2 Motor Constants

In addition to inertia parameters, parameters for the motor are needed to map the control inputs to force and torque values. Here, the motor constants are found using a commercially available experimental setup, the Series 1580 Thrust Stand and Dynamometer rcb (2016) sold by RCbenchmark. It measures thrust, torque, voltage and current, and the data is logged via USB interface. The RCBenchmark

free software is used to send commands to the setup. The setup can withstand up to $\pm$ 5 kg thrust and $\pm$ 1.5 Nm torque. The maximum voltage and current for operation are 35 V and 40 A, respectively. The output is available as .csv files via USB. The setup is shown in Figure 5.4 rcb (2016).

Figure 5.4: RCbenchmark Series 1580 Thrust Stand

The step-by-step procedure to measure the thrust and torque values of the motor is described on the RCbenchmark website RCbenchmark.com (2015). The measured thrust data is shown in Figure 5.5i. From this data, the relationship between the electronic speed control (ESC) signal and the motor thrust is approximated as linear, and the thrust constant is found using the least-squares method:

$$T = k_T V, \tag{5.11}$$

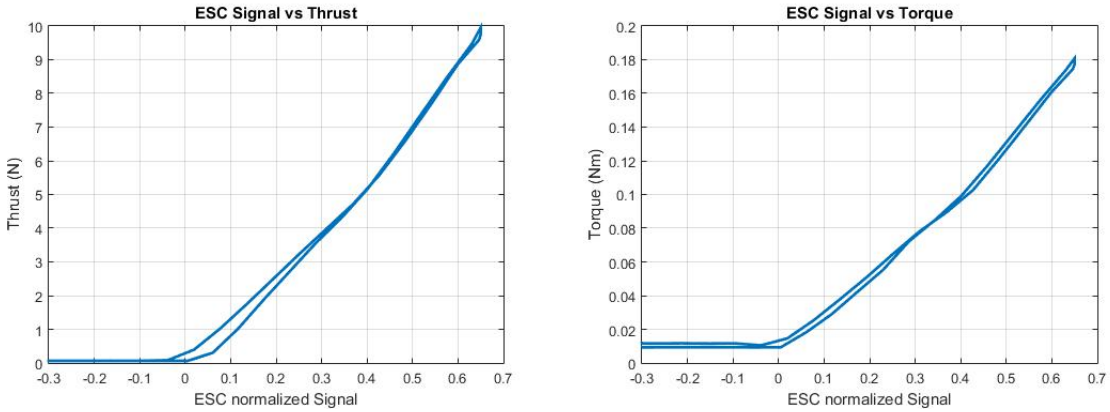
where T is the thrust generated by the propeller (N), k_T is the thrust constant (Nm), and V denotes the normalized ESC signal.

Similarly, the torque data is shown in Figure 5.5ii, and the relationship between

the ESC signal and torque is approximated as linear:

$$\tau = k_\tau V, \tag{5.12}$$

where τ is the torque generated by the propeller, k_τ is the thrust constant, and V denotes the applied ESC signal.



(i) ESC Signal vs Thrust data

(ii) ESC Signal vs Torque data

Figure 5.5: Thrust and Torque data measured using thrust stand

The estimated motor constants for the Turnigy 1800 kV motor with 9×4.5 $inch$ propeller are given in the table below:

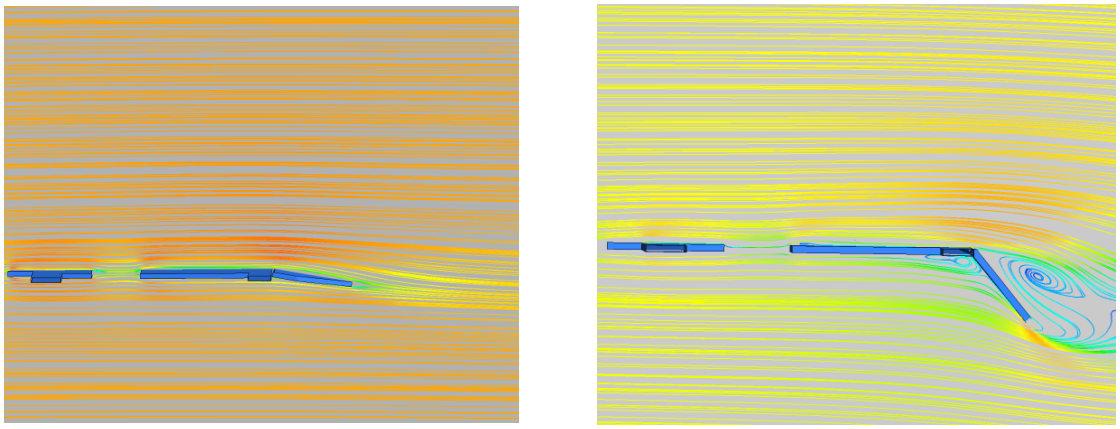
k_T	0.895
k_τ	0.003

Table 5.1: Motor Constants

5.3 Aerodynamic Coefficients

For the model, the aerodynamic coefficients of lift, drag and moment are calculated using ANSYS ans (2017). Initially, a domain is created for the fluid flow, and a control

surface is placed inside the domain for analysis. The velocity streamline around the right half of the body at 10 and 50 deg are shown in Figure 5.6 respectively.



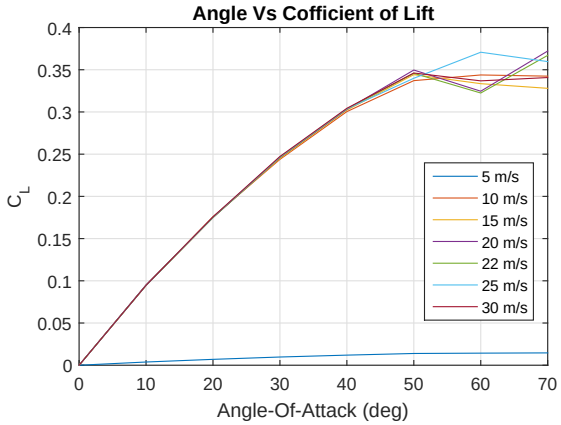
(i) 10 deg (ii) 50 deg

Figure 5.6: Velocity Streamline around the body in ANSYS

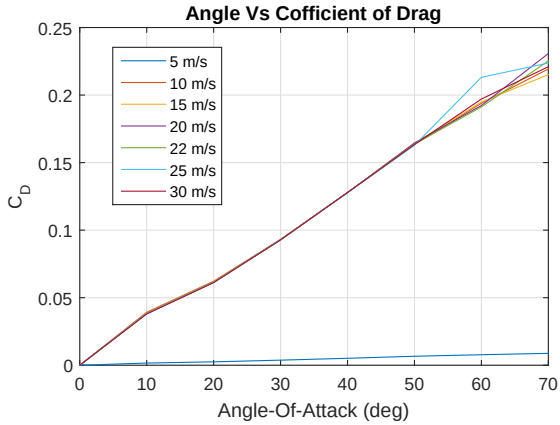
In this analysis, standard mesh triads and a k epsilon realizable solver with standard wall function are used. The velocity of fluid flow is varied over the values $5m/s$, $10m/s$, $20m/s$, $30m/s$, $40m/s$, and $50m/s$, and each solution is calculated for 750 iterations. The solution consists of lift, drag, moment, and their coefficients. The graphs in Fig. 5.7 show the relationship between angle-of-attack (AOA) and the lift, drag and moment coefficients. From the data in the graphs, these relationships can be approximated as linear over a range of 0 to 50^o:

$$C_i = \frac{\partial C_i}{\partial \alpha}\alpha, \qquad i \in \{L, D, m\} \tag{5.13}$$

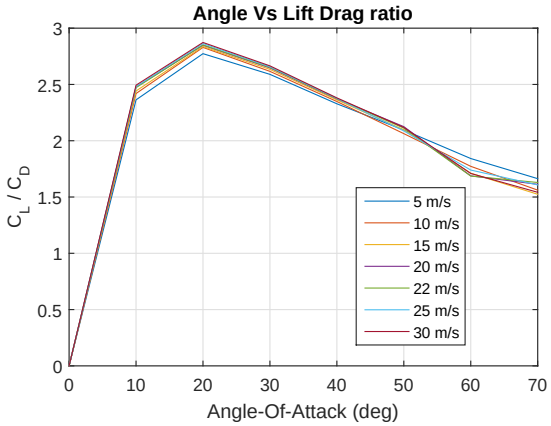
The relationship between the AOA and these coefficients are found using the least-squares method and given in the table below.



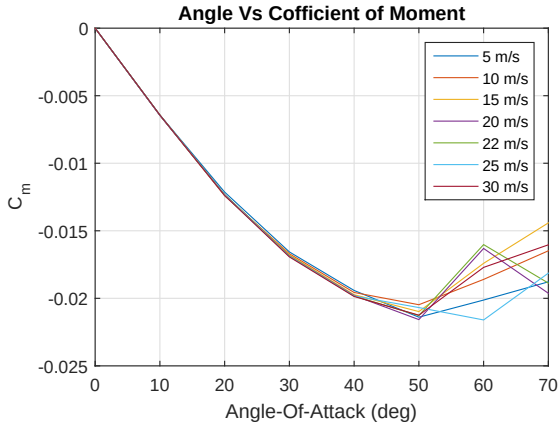
(i) AOA vs C_L



(ii) AOA vs C_D



(iii) AOA vs C_L/C_D



(iv) AOA vs C_m

Figure 5.7: Aerodynamic Coefficients calculated using ANSYS

C_L	0.008017 α
C_D	0.003194 α
C_m	0.0005321 α

Table 5.2: Aerodynamic Coefficients

Chapter 6

LINEAR MODEL

The control objective is to hover the UAV in a plane parallel to the ground. To design a hover controller, first we need to linearize the nonlinear dynamics of the system. For linearization, the equilibrium point and trim conditions at that point are necessary. We specify desired a equilibrium state at which the UAV hovers 5 m above the ground at the origin of an inertial reference frame:

$$\begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{6.1}$$

6.1 Trim Condition

In Chapter 3-5, the nonlinear dynamics are derived and the model parameters are found for the fabricated model. As mentioned earlier, we need the trim condition for linearization. The model is at equilibrium when the system is at equilibrium point with their corresponding trim values. The complete nonlinear system dynamics in Eq. 3.22 and 3.29 are written as,

$$\frac{dX}{dt} = f(X, U) \tag{6.2}$$

The equilibrium condition for the system in 6.2 is,

$$\frac{dX}{dt} = 0 \tag{6.3}$$

$$f(X_e, U_e) = 0 \tag{6.4}$$

where, X_e - Equilibrium point,

U_e is the trim values of the control inputs

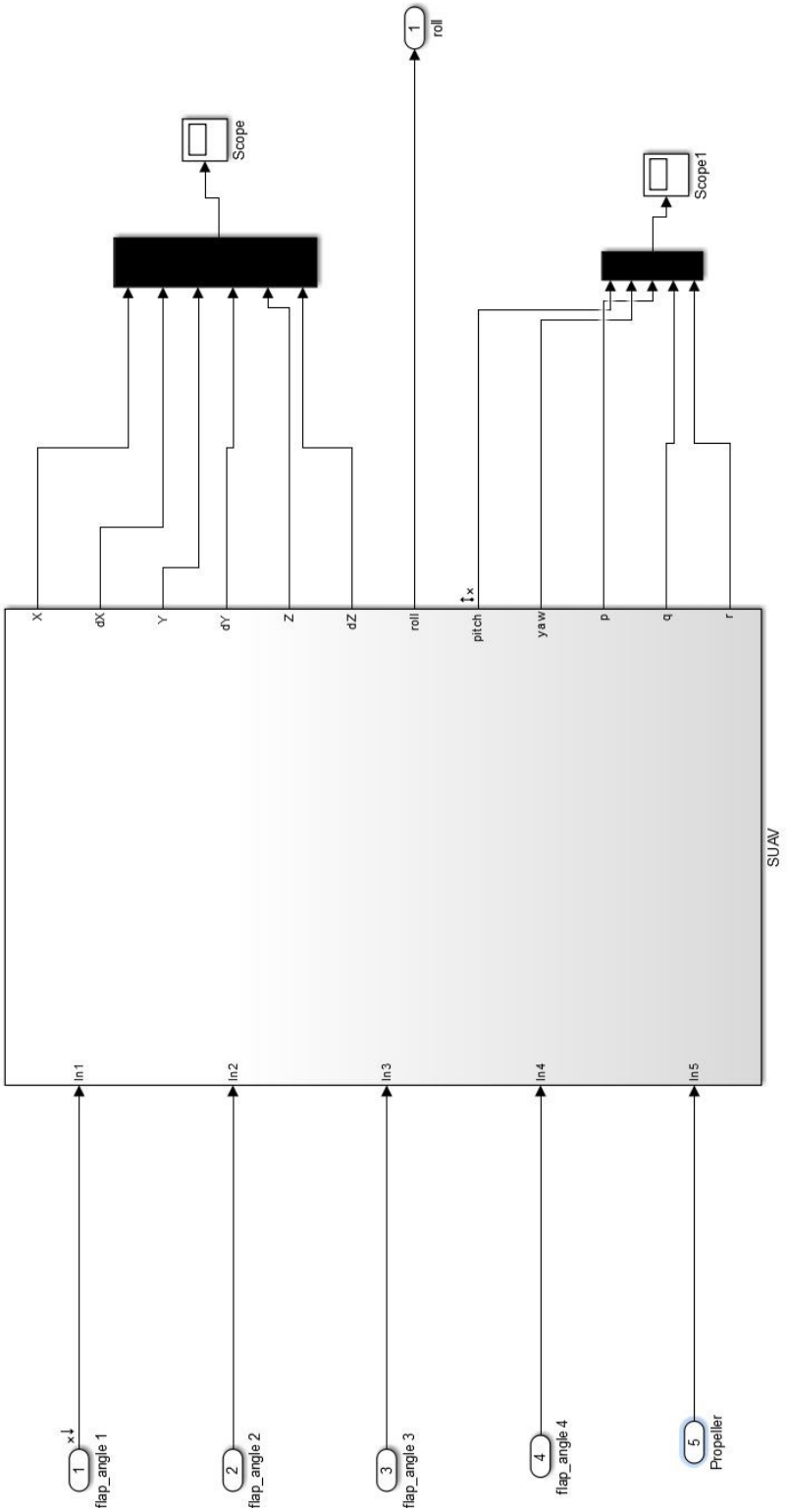


Figure 6.1: Nonlinear Model in Simulink

The trim values can be found by substituting 6.1 in 6.4 and solving for the values of the control inputs. For simplicity, Simulink is used to calculate the trim values. The nonlinear model is drawn in Simulink (Figure 6.1), and the trim condition is calculated using the command below,

$$[X_e, U_e] = trim('SUAV.slsx', X_i) \tag{6.5}$$

where X_i is the initial condition and 'SUAV.slsx' is the name of the Simulink model. Since the model is nonlinear, it has more than one equilibrium point. The initial condition X_i is chosen close to the desired equilibrium point. The trim values for the system are calculated to be:

$$U_e = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \delta_4 & T \end{bmatrix}^T = \begin{bmatrix} 13.50 & 13.50 & 13.50 & 13.50 & 4.694 \end{bmatrix}^T \tag{6.6}$$

where δ_i is the angle of deflection of the i^{th} control surface (deg), and
T is the thrust developed by the propeller (N).

The Simulink model for the rotational dynamics of the system is shown in Figure 6.2.

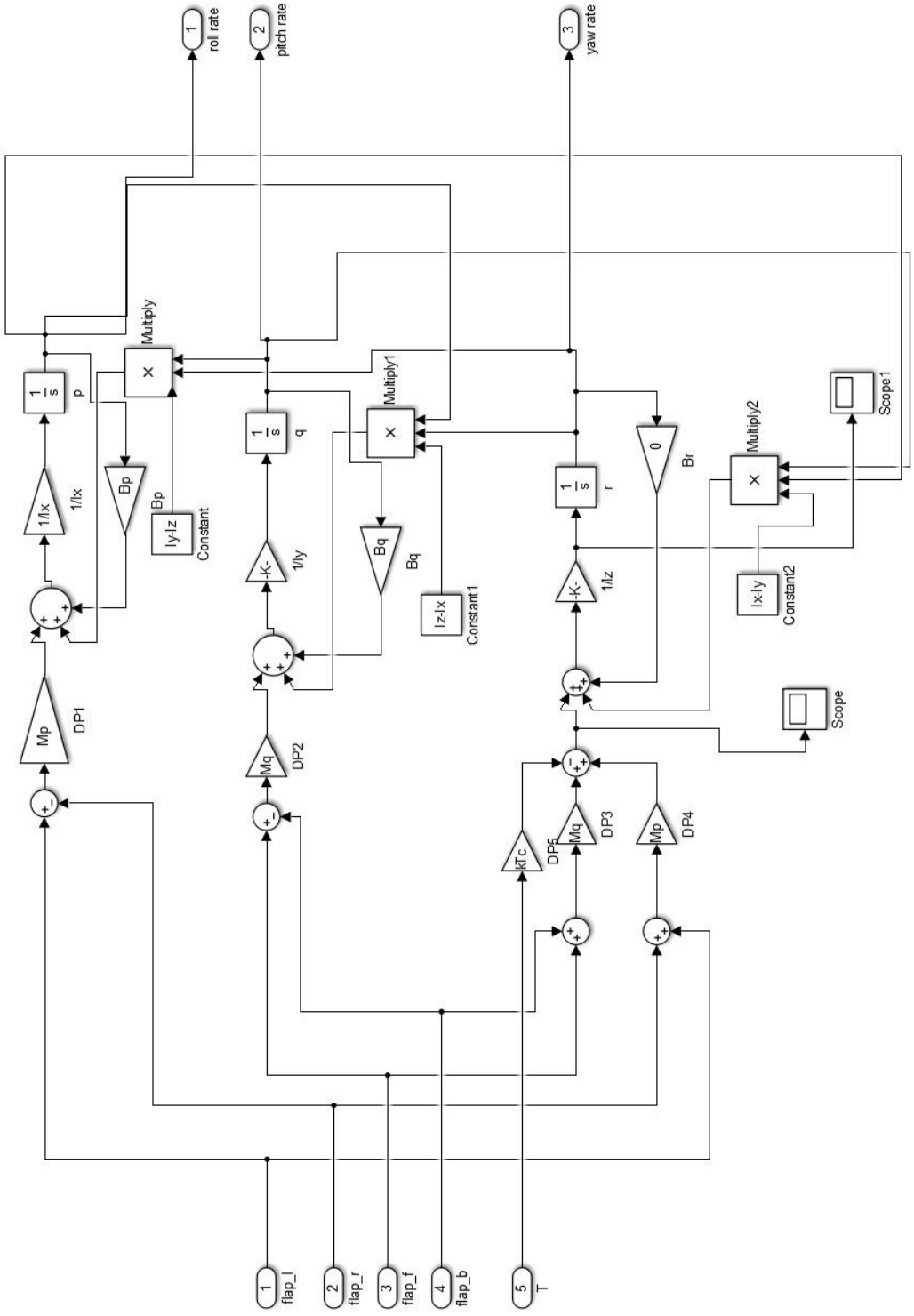


Figure 6.2: Rotational Dynamics in Simulink

6.2 Linearization

The linearized model at hovering can be derived using the small perturbation method or first-order Taylor series expansion at the equilibrium condition. The linearized model is derived manually using the small perturbation method and validated using Simulink.

Small Perturbation Method

We know that the equilibrium point is X_e and the trim value is U_e. We now perturb the system states by δX and the control inputs by δU from the equilibrium points:

$$\delta \dot{X} = f(X_e + \delta X, U_e + \delta U) \tag{6.7}$$

Substitute the values of X_e and U_e in the above equation. Since we consider small perturbations, we neglect the second-order and higher-order terms from the equation. The final model we get is in the form,

$$\delta \dot{X} = A\delta X + B\delta U \tag{6.8}$$

where, δX is the state vector $(n \times 1)$,

δU is the control input vector $(m \times 1)$

A is the system matrix for the linearized model $(n \times n)$, and

B is the control matrix for the linearized system $(n \times m)$

First-order Taylor Series Expansion

This is another method used for linearizing the system dynamics. The Taylor series expansion of a function $f(X)$ near $X = X_e$ is given by:

$$f(X) = f(X_e) + \frac{f'(X_e)}{1!}\delta X + \frac{f''(X_e)}{2!}\delta X^2 + \dots \tag{6.9}$$

Now, the first-order expansion of $f(X)$ near $X = X_e$ is given by:

$$f(X) = f(X_e) + \frac{\partial f}{\partial X}\delta X \tag{6.10}$$

Using a first-order expansion, the linearization of the nonlinear dynamics is,

$$\delta\dot{X} = f(X, U) = \frac{\partial f}{\partial X}\delta X + \frac{\partial f}{\partial U}\delta U \tag{6.11}$$

Comparing 6.8 and 6.11, we get

$$A = \frac{\partial f}{\partial X}, \qquad B = \frac{\partial f}{\partial U} \tag{6.12}$$

Linearization using Simulink

The Fig. 6.1 shows the complete Simulink model. The syntax for linearization is given by:

$$Sys = linmod('SUAV.slsx', X_e, U_e); \tag{6.13}$$

The objective of this thesis is attitude control of the UAV. Hence, the state vector of the UAV is reduced to the three angles and three angular rates:

$$X = \begin{bmatrix} \phi & \theta & \psi & p & q & r \end{bmatrix}^T \tag{6.14}$$

The output is defined as $Y = X$. The linearized system dynamics for the attitude control problem are given by $\dot{X} = AX + BU$, where:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.0335 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0226 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.6196 & -0.61960 & 0 & 0 & 0 \\ 0 & 0 & 0.7353 & -0.7353 & 0 \\ 0.8207 & 0.8207 & 1.003 & 1.003 & 6.209 \end{bmatrix}$$

$$C = I$$

$$D = 0$$

Chapter 7

NORMS

In this chapter, we give a brief introduction to normed spaces and norm computation using Linear Matrix Inqualities (LMIs). This introduction is motivated by the fact that eigenvalues are a poor measure of gain. Eigenvalues only provide the gain of the system for special cases where the inputs and the outputs are in the same direction. For example, consider the input vector $u = v_i$, where v_i is an eigenvector of the matrix G. The output vector is defined as $y = Gv_i$. Then:

$$y = Gv_i = \lambda_i v_i$$
$$\frac{||y||}{||u||} = \frac{||\lambda_i v_i||}{||v_i||} = |\lambda_i|$$

The concept of eigenvalues is useful for stability analysis but not for performance Skogestad and Postlethwaite (2007). Moreover, the eigenvalues do not have a meaning for MIMO systems. Hence, we need a generalized performance measure for the system, and so matrix norms are introduced. The matrix norms give a lot of information regarding the gain of the system. The input and output directions from the SVD are orthogonal. This performance measure can also be extended to MIMO systems.

7.1 Mathematical Introduction

We will define several fundamental mathematical concepts such as vector space, normed spaces, and inner product space. These definitions will provide a foundation for the development of matrix norms Rodriguez (2004), Dullerud and Paganini (2013), Duan and Yu (2013).

Definition 2 *A vector space or linear space is an ordered pair* (X, F) *consisting of a set* X *and a field* F *(*$\mathbb{R}$ *or* $\mathbb{C}$*) such that*

$$a_1 x_1 + a_2 x_2 \in X \tag{7.1}$$

Definition 3 *A norm on a vector space* X *over a field* F *(*$\mathbb{R}$ *or* $\mathbb{C}$*) is a function* $|| \cdot || : X \to R$ *that satisfies the following properties for any* $x, y \in X$ *and* $a \in F$ *:*

- $||x|| \geq 0$
- $||x|| = 0$ *if and only if* $x = 0$
- $||ax|| = |a|||x||$
- $||x + y|| \leq ||x|| + ||y||$ *(Triangle Inequality)*

The triple $(X, F, || \cdot ||)$ *is referred to as a normed linear space*

Definition 4 *An inner product on a vector space* X *over a field* F *is a function* $< \cdot, \cdot >: X \times X \to F$ *that satisfies the following properties for any* $x, y, z \in X$ *and* $a \in F$ *:*

- $< x, x > \geq 0$
- $< x, x >= 0$ *if and only if* $x = 0$
- $< x, y >= \overline{< y, x >}$
- $< ax, y >= a < x, y >$
- $< x + y, z >=< x, z > + < y, z >$

The triple $(X, F, < \cdot, \cdot >)$ *is referred to as a normed linear space.*

Definition 5 *Consider a normed linear space $(X, F, ||\cdot||)$. A sequence $\{x_n\}_{n=1}^{\infty} \subset X$ is said to be a Cauchy sequence if given $\epsilon > 0$, there exists N_ϵ such that*

$$||x_m - x_n|| < \epsilon \quad \textit{for all} \quad m, n > N_\epsilon \tag{7.2}$$

Definition 6 *A normed linear space $(X, F, ||\cdot||)$ is said to be complete (with respect to the topology induced by $||\cdot||$) if every Cauchy sequence of elements converges to an element within the space. Such a space is called a Banach space; i.e., a Banach space is a complete normed linear space.*

Definition 7 *An inner product space $(X, F, <\cdot, \cdot>)$ with norm $||\cdot|| = \sqrt{<\cdot, \cdot>}$ is said to be complete (with respect to the topology induced by $||\cdot||$) if every Cauchy sequence of elements converges to an element within the space. Such a space is called a Hilbert space; i.e. a Hilbert space is a complete normed linear space.*

7.2 Vector Norms

In general, vector norms are used to quantify the "size" (length) of a vector Dullerud and Paganini (2013). In engineering, these vector norms are the signal norms, which help us to quantify the signals such as average gain, root-mean-square (rms) value, power of the signal and maximum bound on the signal.

7.2.1 L^p Vector Norms

Let $f : \mathbb{R}^n \to F$ denote function from a vector to a scalar value. Moreover, this function has to satisfy the properties of a normed space. We can define several functions which have the above property, i.e., there are norms such as $L^1, L^2, \ldots L^\infty$, where L denotes *Lebesque integral*. The Lebesgue integral extends the integral to a larger class of functions. It also extends the domains on which these functions can be defined Pugh and Pugh (2002). The definition of this integral is outside of the scope

of this thesis; further details are given in Pugh and Pugh (2002). If the L^p norm of a vector is a finite value ($||g||_{L^p} < \infty$), then g is in the Lebesgue space L^p.

Some of the vector norms on an infinite sequence are listed below Rodriguez (2004):

$$||g||_{L_p} = \sqrt[p]{\sum_{i=1}^{\infty} g_i^p}, \qquad p = 1, 2, \ldots \tag{7.3}$$

$$||g||_{L_1} = \sum_{i=1}^{\infty} |g_i| \tag{7.4}$$

$$||g||_{L_2} = \sqrt{\sum_{i=1}^{\infty} g_i^2} \tag{7.5}$$

$$||g||_{L_\infty} = \max_{i=1,2,\ldots,\infty} |g_i| \tag{7.6}$$

Similarly, norms on functions are defined as Rodriguez (2004):

$$||g||_{L_p} = \sqrt[p]{\int_0^{\infty} g(t)^p dt}, \qquad p = 1, 2, \ldots \tag{7.7}$$

$$||g||_{L_1} = \int_0^{\infty} |g(t)| dt \tag{7.8}$$

$$||g||_{L_2} = \sqrt{\int_0^{\infty} g(t)^2 dt} \tag{7.9}$$

$$||g||_{L_\infty} = \sup_{t \in [0,\infty)} |g(t)| \tag{7.10}$$

7.2.2 Matrix Norms

A matrix norm cannot be calculated directly; it is derived from a vector norm. Generally, the matrix norms are linear bounded operators Dullerud and Paganini (2013).

Definition 8 *The normed space of bounded linear operator from X to Y is denoted $L(X, Y)$ with a norm*

$$||G||_{L(X,Y)} := \sup_{u \in X} \frac{||Gu||_Y}{||u||_X}, \quad u \neq 0. \tag{7.11}$$

This satisfies the properties of a norm. It is also known as an **induced norm**.

There is one more type of space that is widely used in complex analysis, called *Hardy spaces* (H^p) Dullerud and Paganini (2013). H^p are certain spaces of functions in which one or more complex variables are complex differentiable in a neighborhood of every point on the unit disk or upper half plane Dullerud and Paganini (2013).

The Laplace transform used in control systems is a bounded linear operator from L_2 to H_2 Peet (2017a).

Definition 9 *Given $u \in L_2[0,\infty)$, the Laplace transform of $u(t)$ is $U(s) = \wedge u(t)$, where*

$$U(s) = \wedge u(t) = \lim_{T\to\infty} \int_0^T u(t)e^{-st}dt \tag{7.12}$$
$$\wedge : L_2 \to H_2$$

if this limit exists.

The equivalence between L_2 and H_2 is given by the *Paley-Wiener* Theorem Peet (2017a).

Definition 10 *Paley-Weiner Theorem*

- *If $u \in L_2[0,\infty)$, then $\wedge u \in H_2$*
- *If $U(s) \in H_2$, then there exists a $u \in L_2[0,\infty)$ such that $U(s) = \wedge u$* ***(Onto)***

H_2 Norm

The H_2 norm of a system is the L_2 norm of the impulse response of the system Rodriguez (2004):

$$||G||_{H_2} = \sqrt{\int_0^\infty g(t)^2 dt} \tag{7.13}$$

This norm is equivalent to the L_2 norm of the system in the frequency domain:

$$||G||_{H_2} = \sqrt{\int_{-\infty}^{\infty} G(j\omega)^T G(j\omega) d\omega} \tag{7.14}$$

The impulse input has all the frequencies in it; it is not a periodic signal. For all the frequencies, the magnitude of the impulse input is one. So conceptually, the H_2 norm of the system denotes the **average gain of the system over all frequencies** Toivonen (1995). The H_2 norm of a system exists only for a strictly proper transfer function.

The motivation behind H_2 control design is to minimize the average gain of the system over all frequencies. Generally, the sensor noise that affects the system will be a white noise (which has all frequencies in it). So, the H_2 control design will minimize the average gain of the system as well as reduce the effect of white noise on the system.

H_2 control is equivalent to Linear Quadratic Gaussian (LQG) control design. The LQG controller can be calculated by solving the below optimization problem:

$$\begin{aligned} \min_{U} \quad & ||X||_{L_2} + ||U||_{L_2} + \sqrt{E[\tilde{X}^T \tilde{X}]} \\ \text{subject to} \quad & \dot{X} = AX + BU \end{aligned} \tag{7.15}$$

where X is the state vector of the system,

U is the input vector for the system,

$E[\tilde{X}]$ is the expectation of the estimation error $\tilde{X} = X - \hat{X}$, and

$\hat{X}$ is the estimated state vector of the system.

H_∞ Norm

The H_∞ norm is the induced norm from L_2 to L_2 Peet (2017a). It is the maximum value of the ratio between the L_2 norm of the output and the L_2 norm of the input:

$$||G||_{H_\infty} = \sup_{u \neq 0} \frac{||Gu||_{L_2}}{||u||_{L_2}} \tag{7.16}$$

It is equivalent to the maximum singular value of the system over all frequencies Dullerud and Paganini (2013). It is the peak value in the singular values plot of the sensitivity transfer function:

$$||G||_{H_\infty} = \sup_{\forall \omega} \sigma(G(j\omega)) \tag{7.17}$$

where, $\sigma(G(j\omega))$ denotes the singular values of the transfer function $G(j\omega)$ This norm is defined for all inputs expect the zero input. The H_∞ norm of a system represents the maximum possible amplification that the system can provide Toivonen (1995). The idea behind minimizing the H_∞ norm is to minimizes the maximum amplification by the system when disturbances enter the system. H_∞ control design is defined as the solution u to the following optimization problem:

$$\min_{u} \ \sup_{u \neq 0} \frac{||Gu||_{L_2}}{||u||_{L_2}} \tag{7.18}$$

7.3 Computing H_2 and H_∞ Norms

In classical control theory the graphical plots were used as a tool for computing system norms. After the introduction of Linear Matrix Inequalities (LMIs) with proper solvers to solve the LMIs, it became easier to calculate the system norms Peet (2017a). From Duan and Yu (2013), it is easy to show that LMIs are convex. Hence, there exists a global minimum for the LMI optimization problem, and it is easy to solve the optimization problem using algorithms such as interior point methods.

To compute the system norms, we will use the theorems given below. The proofs of these theorems are given in Peet (2017a), Duan and Yu (2013), Dullerud and Paganini (2013). First, we define the state space formulation of the system as:

$$y = Gu, \quad G : L_2 \to L_2$$
$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y = Cx(t) + Du(t)$$

For the state space system (A, B, C, D),

$$G(s) = C(sI - A)^{-1}B + D$$
$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

To compute the H_2 norm of the transfer function $G(s)$, we use the following definition from Linear Matrix Inequalities (LMI) Peet (2017a):

Definition 11 *Suppose $G(s) = C(sI-A)^{-1}B+D$. Then the following are equivalent:*

- *A is Hurwitz and $||G||_{H_2} < \gamma$, (where γ is a scalar, $\gamma > 0$)*
- *There exists some $P > 0$ such that*

$$trace(CPC^T) < \gamma$$
$$AP + PA^T + BB^T < 0$$

Note: If the eigenvalues of A are negative, then A is Hurwitz.

Similarly, to compute the H_∞ norm of the transfer function $G(s)$, we use the following lemma Duan and Yu (2013):

Definition 12 *The KYP Lemma (also known as The Bounded Real Lemma)*

$$Let \quad G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

Then the following are equivalent:

- $||G||_{H_\infty} < \gamma$, *(where* γ *is a scalar,* $\gamma > 0$*)*
- *There exists some* $P > 0$ *such that*

$$\begin{bmatrix} A^TP + PA & PB \\ B^TP & -\gamma I \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} C^T \\ D^T \end{bmatrix}\begin{bmatrix} C & D \end{bmatrix} < 0$$

- *There exists some* $P > 0$ *such that*

$$\begin{bmatrix} PA^T + AP & B & PC^T \\ B^T & -\gamma I & D^T \\ CP & D & -\gamma I \end{bmatrix} < 0$$

The above LMI can be solved using the YALMIP toolbox in MATLAB. There are several efficient solvers that are interfaced in YALMIP. A brief introduction to YALMIP with some basic examples can be found in Lofberg (2004), Lofberg (2017).

Chapter 8

ROBUST CONTROL DESIGN

In the last chapter, we explained the vector and matrix norms. One objective of this thesis is to design a robust controller for the developed UAV. In this chapter, we will discuss the controller architecture, weight selections, and LMI for the control design.

8.1 Control Architecture

There are different types of architecture for a control system. The nominal architecture has a feedback loop with the controller in the forward path Fig. 8.1. The output of the system y is fed back to the controller K, and the controller output u is defined as the input to the plant P Rodriguez (2004).

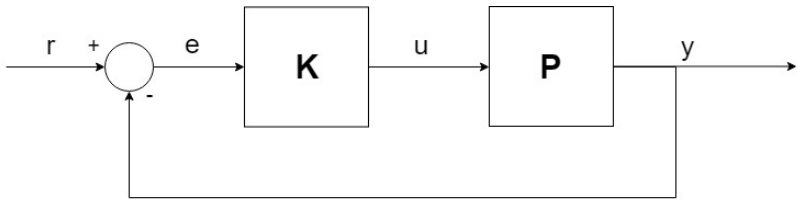


Figure 8.1: Block Diagram for Nominal Closed-Loop Control

There are other controller structures such as cascaded loop and two-level control Skogestad and Postlethwaite (2007). The cascaded system consists of two loops, inner and outer loop. The plant is reduced to two models; the plant model for the inner loop operates faster than the outer loop plant. There are two different controllers for this architecture: the output y_2 of the outer loop is fed back to the controller K_2, and the output from K_2 acts as a set-point for the inner loop controller K_1. The error e_1 is defined as the difference between the K_2 controller output u_2 and the inner loop

plant output y_1. The controller for plant P_2 is designed based on the assumption that $I(s) = 1$, where $I(s)$ is the transfer function of the closed inner loop of the system. Generally, to validate this assumption, the inner loop has to run faster than the outer loop. This architecture is widely used in aircraft since the model can be reduced to two models and also due to its simplicity. In this thesis, we use this architecture for controller implementation. The figure below shows the cascaded control architecture.

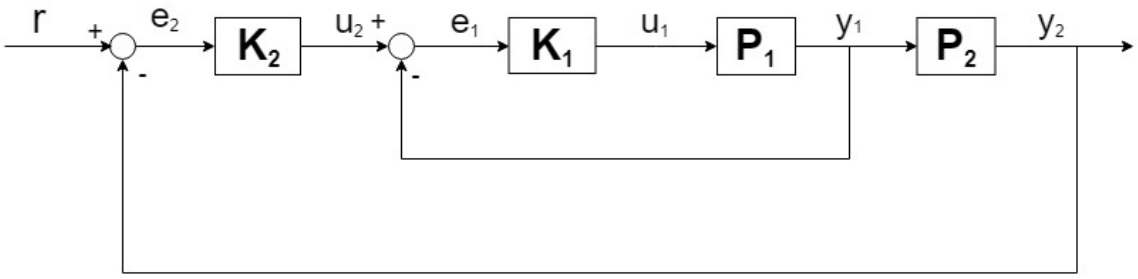


Figure 8.2: Block Diagram for Cascaded-Loop Control

In a two-level control structure Skogestad and Postlethwaite (2007), the error is not fed back to the controller. Here, the controller inputs are the set-point and plant output. The controller is a system with two inputs and an output (for a SISO plant model). This architecture is used in order to achieve more demanding desired specifications. This architecture might yield a high-order controller, which is challenging for controller implementation. The control architecture for the two-level system is shown below.

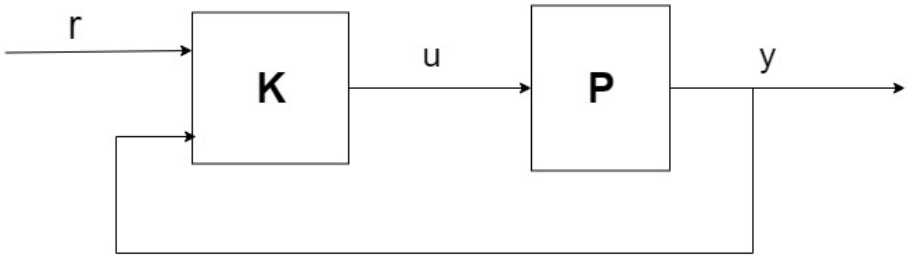


Figure 8.3: Block Diagram for Two-level Control

8.2 Weight Selection

In real-world applications, the plant model in Figure 8.1 is affected by disturbances and noise. Such a feedback loop is shown in Figure 8.4 Here, the plant P has input and output disturbances because of the environmental characteristics and plant high-frequency nonlinear dynamics, and feedback to the controller has high-frequency sensor noise Rodriguez (2004).

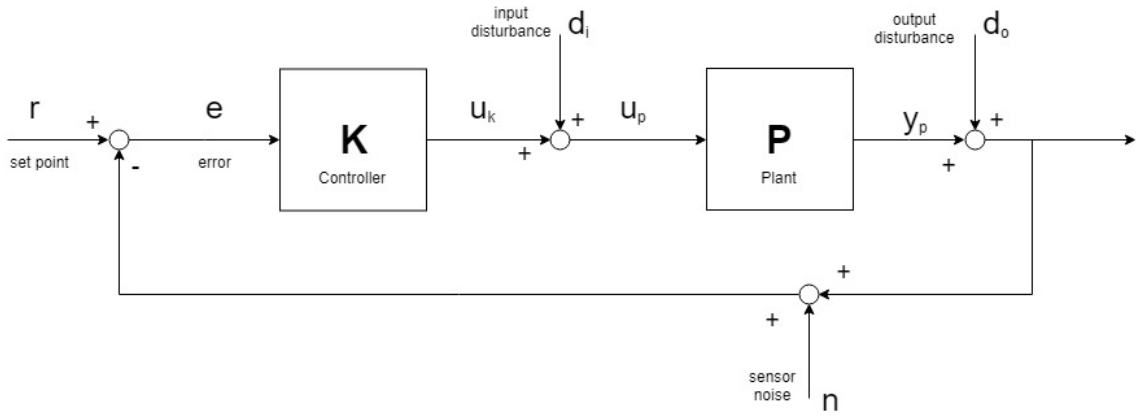


Figure 8.4: Block Diagram for Feedback System with Disturbances and Noise

In order to find the effect of noise on the controller, we need the closed-loop equations for the above system. Let P and K be LTI transfer functions. The plant input is u_p and its output is y_p. The exogenous signals that affect the system are the set-point, input disturbances, output disturbances and sensor noise (r, d_i, d_o, n) Rodriguez (2004). The error is defined by the difference between the set-point and the output with sensor noise. The error e is input to the controller and the output is the control signal u_k.

$$
\begin{aligned}
y &= y_p + d_o \\
&= Pu_p + d_o \\
&= P(u_k + d_i) + d_o \\
&= PKe + Pd_i + d_o \\
&= PK(r - y - n) + Pd_i + d_o \\
y &= PKr - PKy - PKn + Pd_i + d_o
\end{aligned} \tag{8.1}
$$

$$
(I + PK)y = PKr - PKn + Pd_i + d_o
$$

$$
y = (I + PK)^{-1}PKr - (I + PK)^{-1}PKn + (I + PK)^{-1}Pd_i + (I + PK)^{-1}d_o \tag{8.2}
$$

For simplification, let us consider the SISO case Skogestad and Postlethwaite (2007) Rodriguez (2004),

$$
y = \frac{PK}{1 + PK}r + \frac{P}{1 + PK}d_i + \frac{1}{1 + PK}d_o - \frac{PK}{1 + PK}n \tag{8.3}
$$

The above equation shows the importance of trade-offs in the controller design. The following conditions are very important for controller design Skogestad and Postlethwaite (2007), Rodriguez (2004):

- **Tracking** : For good tracking performance, the control gain has to be large, i.e) if $PK >> 1$, then $\frac{PK}{1 + PK} = 1$

  K should be **large**

- **Input and output disturbances** : For good disturbance attenuation, we need large control gains. i.e.) if $1 + PK$ is very high, then $\frac{1}{1 + PK} << 1$

K should be **large**

- **Sensor Noise** : For good sensor noise attenuation, we need to keep the gain as low as possible. i.e.) if $PK < 1$, then $\frac{1}{1+PK} << 1$

K should be **small**

- **High-Frequency RHP zero**: Since we are linearizing the nonlinear system dynamics about the equilibrium point, there might be one or more zeros in the right half plane. Hence, if the control gain is very large, then it will make the system unstable.

K should be **small**

Based on the above derivation, two important terms were introduced. They are the sensitivity and complementary sensitivity functions. The sensitivity function is defined as the transfer function from output disturbance to output and the complementary sensitivity function is defined as the transfer function from sensor noise to output Rodriguez (2004). (We can neglect the negative sign in T_{ny} below.)

$$S = T_{d_{oy}} = \frac{1}{1+PK} \tag{8.4}$$

$$T = T_{ny} = \frac{PK}{1+PK} \tag{8.5}$$

The trade-off characteristics are defined by the relationship between the sensitivity and complementary sensitivity functions. Based on this relationship, we can design the controller for desired specifications Rodriguez (2004). The sensitivity function denotes the input and output disturbance attenuation, whereas the complementary sensitivity function denotes the sensor noise attenuation. Note that:

$$S + T = I \tag{8.6}$$

Usually, the input and output disturbances are low-frequency disturbances and sensors have high-frequency noise. Based on this, we have to select the weights for our controller design. The block diagram for the control system with these weights is shown in Figure 8.5.

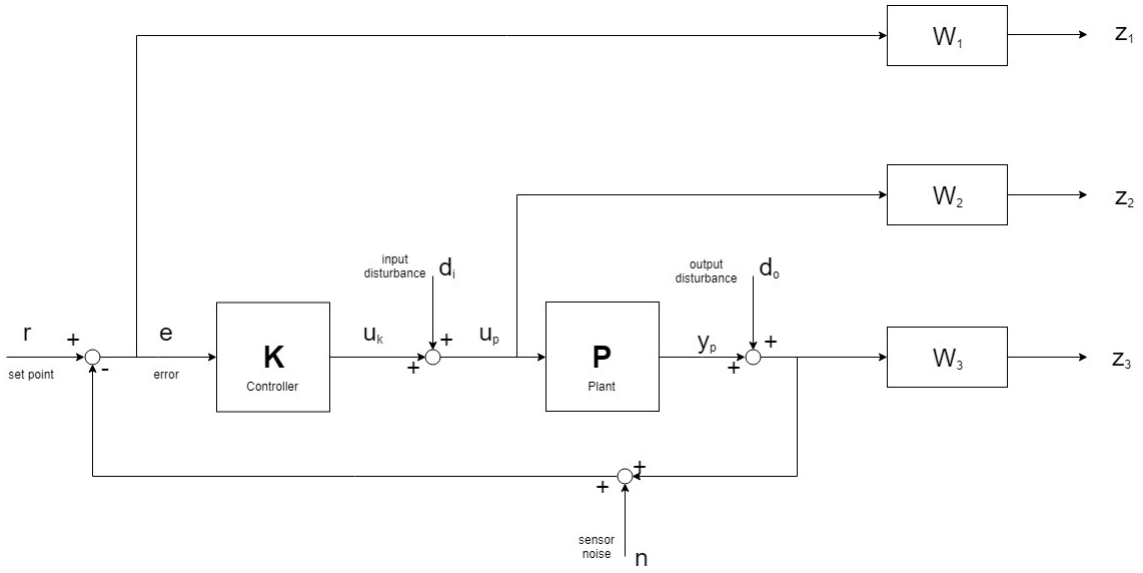


Figure 8.5: Block Diagram for Feedback System with Weights

The H_∞ control design problem is to minimize the H_∞ norm of the system. For real world applications, we have to consider several questions before designing the controller Rodriguez (2004):

- What is the maximum frequency of input and output disturbances?
- What is the actuator bandwidth?
- What is the minimum high-frequency sensor noise?

The above are the tuning parameters for the H_∞ control design. Based on the system requirements, the frequency-dependent weights should be selected for the optimization problem. Now, the problem becomes a weighted mixed sensitivity H_∞

control design problem. It is defined by the equation below Skogestad and Postlethwaite (2007):

$$||T(P,K)||_{H_\infty} = \left\| \begin{bmatrix} W_1 S \\ W_2 K S \\ W_3 T \end{bmatrix} \right\|_{H_\infty} \tag{8.7}$$

This is a suboptimal H_∞ problem. The weights (W_1, W_2, W_3) must be stable transfer functions that are diagonally structured.

Sensitivity Weighting : This is the constraint on the sensitivity function Rodriguez (2004),

$$\sigma_{max}[W_1(s)S(s)] \leq \|W_1 S\|_{H_\infty} < \gamma \qquad (\gamma > 0) \tag{8.8}$$

for all ω in $s = j\omega$. From this, the H_∞ norm is defined as,

$$\begin{aligned} \sigma_{max}[S] &\leq \sigma_{max}[W_1^{-1} W_1 S] \\ &\leq \sigma_{max}[W_1^{-1}]\sigma_{max}[W_1 S] \\ &< \sigma_{max}[W_1^{-1}]\gamma = \frac{1}{\sigma_{min}[W_1]}\gamma \end{aligned} \tag{8.9}$$

Consider the weights to be in the form

$$W_1(j\omega) = \frac{k_1(s + z_1)}{s + p_1} \tag{8.10}$$

where (k_1, z_1, p_1) are the tuning parameters. Since the disturbances are low-frequency signals, the weights are selected such that the sensitivity function S is small at low frequencies. From the above equation, it is easy to see that the frequency plot pf the $S(j\omega)$ lies below the frequency plot of $W_1^{-1}(j\omega)$.

Control Weighting : This is a constraint for the actuator performance. The actuator bandwidth is given as the constraint for this weight selection. The weights

can either depend on frequency or be frequency-independent. Here the weights can be similar to eq.8.10, or they can be defined as Rodriguez (2004)

$$W_2 = k_2 \tag{8.11}$$

where k_2 is a scalar value.

Complementary Sensitivity Weighting : This is a constraint for high-frequency noise, i.e., a constraint on the complementary sensitivity function. It is similar to sensitivity weighting, but it imposes a constraint on the magnitude of the complementary sensitivity function at high frequencies Rodriguez (2004).

8.3 Controller Design

8.3.1 H_2 Control design

From the previous chapter, we know that Linear Quadratic Gaussian (LQG) control is a special case of H_2 control design Peet (2017a). Hence, here we develop an LQG controller for the model in Eq. 7.1. An LQG controller id defined as the optimal output feedback with a quadratic cost function. It is a combination of a Linear Quadratic Regulator (LQR) and a Kalman filter Rodriguez (2004). The outputs are given to a Kalman filter, and the filtered data are used as feedback to the LQR controller. First, we will talk about LQR control design and then move on to Kalman filter design.

For LQR control design, the basic assumptions are as follows Rodriguez (2004):

- The system is LTI
- The system is stabilizable
- The system is detectable

The objective is to minimize the weighted norms of the states and control inputs. The LQR problem statement is given as Rodriguez (2004)

Definition 13

$$\min_U J(U) = \frac{1}{2}\int_0^\infty (X^T Q X + U^T R U)dt$$

subject to

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

$$U = -KX$$

where, Q denotes state weighting matrix, and R denotes control weighting matrix.

The solution to this problem is given by solving the Control Algebraic Riccati Equation (CARE) Rodriguez (2004):

$$PA + A^T P + C^T C - PBR^{-1}B^T P = 0 \tag{8.12}$$

where P is some matrix for which $P > 0$. The control gain is given by

$$K = R^{-1}B^T P \tag{8.13}$$

The CARE can be easily solved using MATLAB. The syntax for LQR control design is given below:

$$[K, P, poles] = lqr(A, B, Q, R) \tag{8.14}$$

Kalman Filter

The Kalman filter design is dual to the LQR problem. For Kalman filter design, the state estimation error dynamics are used to formulate the optimization problem. The basic assumptions for the Kalman filter are as follows Rodriguez (2004):

- The system is LTI
- The system is detectable
- Sensor noise and process noise are assumed to be white noise

We define the system as below,

$$\dot{X} = AX + BU + \zeta$$
$$Y = CX + DU + \theta \tag{8.15}$$

where ζ is the process noise and θ is the sensor noise, both modeled as white noise:

$$\zeta \approxeq N(0, \sigma_\zeta) \tag{8.16}$$
$$\theta \approxeq N(0, \sigma_\theta) \tag{8.17}$$

where σ_ζ and σ_θ are the standard deviation of the process noise and sensor noise, respectively.

The Kalman filter problem statement is given below Triantafyllou and Hover (2003):

Definition 14

$$\min_{\hat{X}} J(\hat{X}) = \sqrt{E(\tilde{X}^T \tilde{X})}$$

subject to

$$\dot{X} = AX + BU + \zeta$$
$$Y = CX + DU + \theta$$

where $\hat{X}$ *is the estimated state vector and,*

$\tilde{X}$ *is the state estimation error vector,* $\tilde{X} = X - \hat{X}$

Similar to LQR, the solution to the Kalman filter problem is obtained by solving the Filter Algebraic Riccati Equation (FARE), which is dual to CARE.

For practical implementation, first we have to design the observer and then update the gains. This is separated into two steps: *Predictor* and *Corrector* Thacker and Lacey (1998).

Consider the discrete-time model for the implementation, where X_k is defined as the k^{th} sample of state X. The initial conditions are given below:
$X_0^+ = X_0$ is the initial value of the state;
U_k is the control input at k^{th} sample;
P_0^+ is the initial error covariance matrix (if $P_0^+ = 0$, then the initial state value is accurate and P_0^+ increases as the accuracy decreases) Thacker and Lacey (1998)

Predictor	
Priori State Estimate	$X_k^- = AX_{k-1}^+ + BU_{k-1}$
Priori Error Covariance	$P_k^- = AP_{k-1}^+A + \sigma_\zeta$
Corrector	
Control Gain Update	$K_k = P_{k-1}^+C^T(CP_k^-C^T + \sigma_\theta)^{-1}$
Posteriori State Estimate	$X_k^+ = X_k^- + K_k(y_k - CX_k^-)$
Posteriori Error Covariance	$P_k^+ = (I - K_kC)P_k^-$

For the attitude control problem, we have six states. The matrix multiplications for this problem are computationally costly in the embedded system, hence the multiplications are processed in a parallel thread. Finally, the filter is validated using Hardware-in-the-loop (HIL) testing.

8.3.2 H_∞ Control Design

The Linear Fractional Transformation (LFT) framework is used here to formulate the H_∞ controller design problem. The LFT framework includes the following

variables Peet (2017a) Dullerud and Paganini (2013):

- $z = [z_1 \ z_2]^T$ is the exogenous output (additional plant outputs)
- y is the measured output (measured outputs with sensor noise)
- u is the plant input (actuator inputs)
- $w = [w_1 \ w_2]^T$ is the exogenous input (e.g., input disturbances)

The LFT can be written as,

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{8.18}$$

P_{ij} are subsystems, each of which is defined by $P_{ij} = C_{ij}(sI - A_{ij})^{-1}B_{ij} + D_{ij}$.

where, P_{11} denotes the system with input w and output z,

P_{12} denotes the system with input u and output z,

P_{21} denotes the system with input w and output y, and

P_{22} denotes the system with input u and output y.

The regulator plant with disturbance and noise is shown below in Figure 8.6. Here $P_0 = (A, B, C, D)$ is the nominal plant; the complete model P is given by

$$\left[\begin{array}{c} \dot{X} \\ \hline z_1 \\ z_2 \\ y \end{array}\right] = \left[\begin{array}{c|ccc} A & B & 0 & B \\ \hline C & D & 0 & D \\ 0 & 0 & 0 & I \\ C & D & I & D \end{array}\right] \left[\begin{array}{c} x \\ \hline w_1 \\ w_2 \\ u \end{array}\right]$$

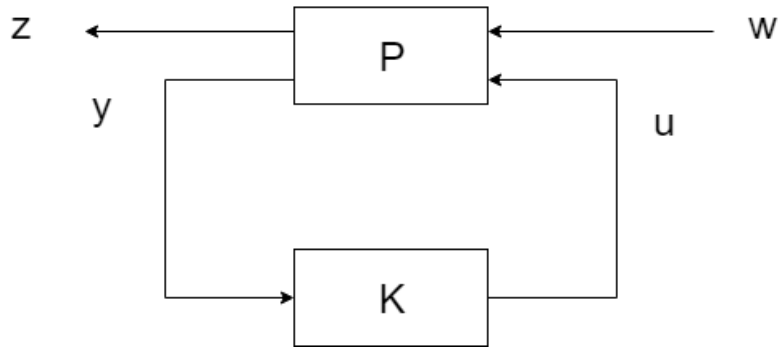


Figure 8.6: Regulator plant with disturbance and noise

The model P is defined as $P = (A_p, B_p, C_p, D_p)$, where

$$A_p = A \tag{8.19}$$

$$B_p = \begin{bmatrix} B & 0 & B \end{bmatrix} \tag{8.20}$$

$$C_p = \begin{bmatrix} C \\ 0 \\ C \end{bmatrix} \tag{8.21}$$

$$D_p = \begin{bmatrix} D & 0 & D \\ 0 & 0 & I \\ D & I & D \end{bmatrix} \tag{8.22}$$

We can further subdivide the above matrices into:

$$B_1 = \begin{bmatrix} B & 0 \end{bmatrix} \qquad B_2 = B$$

$$C_1 = \begin{bmatrix} C \\ 0 \end{bmatrix} \qquad C_2 = C$$

$$D_{11} = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \qquad D_{12} = \begin{bmatrix} D \\ I \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} D & I \end{bmatrix} \qquad D_{22} = D$$

For the controller design, full-state feedback is used because we are able to measure all the outputs, and it is easy to implement full-state feedback in the embedded system. The sensor outputs are directly fed to the Kalman filter, and the filtered states are then used as the feedback.

Now the system dynamics become Duan and Yu (2013),

$$\dot{X} = (A + BK)X \tag{8.23}$$

In order to find the optimal H_∞ controller design, replace A by $A + BK$ in Definition 12. We get Duan and Yu (2013):

Definition 15 *Optimal H_∞ full-state feedback*

Suppose

$$\textit{System } G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right], \qquad \textit{Controller } K(s) = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & F \end{array}\right]$$

Then the following are equivalent:

- *Closed-loop system* $\left\| \dfrac{GK}{I + GK} \right\|_{H_\infty} < \gamma$
- *There exists some matrix* $P > 0$ *such that*

$$\begin{bmatrix} (A + B_2F)^T P + P(A + B_2F) & PB_1 \\ B_1^T P & -\gamma I \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} (C_1 + D_{12}F)^T \\ D_{11}^T \end{bmatrix} \begin{bmatrix} (C_1 + D_{12}F) & D_{11} \end{bmatrix} < 0$$

- *There exists some matrix* $P > 0$ *such that*

$$\begin{bmatrix} P(A + B_2F)^T + (A + B_2F)P & B_1 & P(C_1 + D_{12}F)^T \\ B_1^T & -\gamma I & D_{11}^T \\ (C_1 + D_{12}F)P & D_{11} & -\gamma I \end{bmatrix} < 0$$

The inequality above is bilinear in P and F. By using variable substitution, we can convert it to an LMI and it is given in the following definition Duan and Yu (2013):

Definition 16 *Optimal H_∞ full-state feedback*

Suppose

$$\textit{System } G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right], \qquad \textit{Controller } K(s) = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & F \end{array}\right]$$

Then the following are equivalent:

- *There exists some K such that closed-loop system* $\left\| \dfrac{GK}{I+GK} \right\|_{H_\infty} < \gamma$
- *There exists some matrix $P > 0$ such that*

$$\begin{bmatrix} PA^T + Z^T B_2^T + AP + B_2 Z & B_1 & PC_1^T + Z^T D_{12}^T \\ B_1^T & -\gamma I & D_{11}^T \\ C_1 P + D_{12} Z & D_{11} & -\gamma I \end{bmatrix} < 0$$

Then $F = ZP^{-1}$.

The above definition is for optimal H_∞ control design. This may not give realistic gains for implementation, i.e. in this thesis, gains for the optimal H_∞ controller are of order 10^7. We need a controller with realistic gains and good disturbance and noise attenuation for the for physical implementation. Hence, we need to design a suboptimal mixed H_∞ controller. To do this, we need to add weights to the plant model and then design the controller. Here, we are adding frequency-dependent weights that are each given by the transfer function $W_i(s) = C_{w_i}(sI - A_{w_i})^{-1}B_{w_i} + D_{w_i}$. The block diagram for the model is given below:

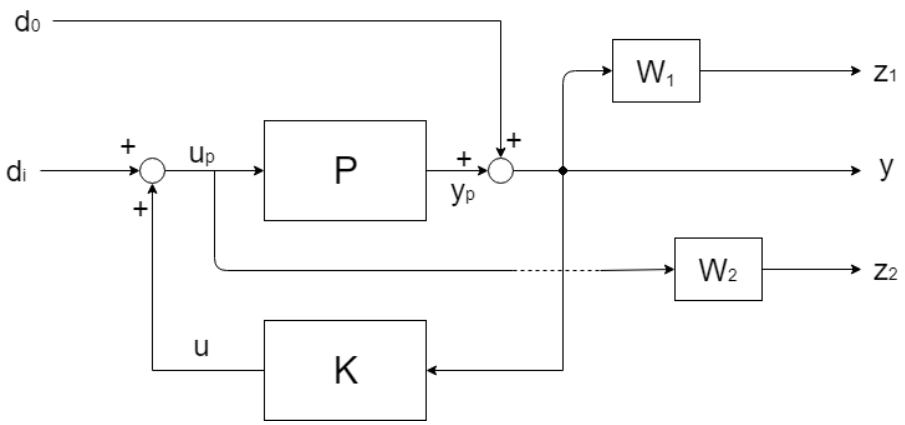


Figure 8.7: Architecture used for Controller Design using LMI

Consider the plant model P, weight $W_1 - (A_{w_1}, B_{w_1}, C_{w_1}, D_{w_1})$ and weight $W_2 - (A_{w_2}, B_{w_2}, C_{w_2}, D_{w_2})$. The portions of the block diagram above with the weights W_1 and W_2, and the corresponding dynamics are given below:

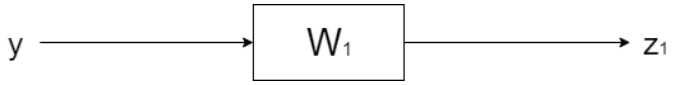


Figure 8.8: Sensitivity Weight

$$\dot{X}_1 = A_{w_1} X_1 + B_{w_1} y_p$$

$$z_1 = C_{w_1} X_1 + D_{w_1} y_p$$

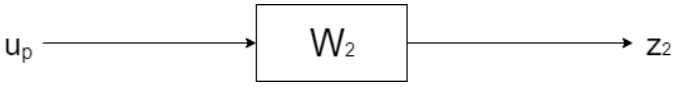


Figure 8.9: Control Weight

$$\dot{X}_2 = A_{w_2} X_2 + B_{w_2} u$$

$$z_2 = C_{w_2} X_2 + D_{w_2} u$$

The plant for mixed sensitivity control design is derived below:

$$y_p = CX + Du + Dd_i$$

$$y = y_p + d_o$$

$$z_1 = C_{w_1} X_1 + D_{w_1} y_p$$

$$z_2 = C_{w_2} X_2 + D_{w_2} u$$

$$\dot{X} = AX + Bu + Bd_i$$

$$\dot{X}_2 = A_{w_2} X_2 + B_{w_2} u$$

$$\dot{X}_1 = A_{w_1} X_1 + B_{w_1} y_p$$

Rearranging the equations above, the plant matrix can be formed with states $X_w = \begin{bmatrix} X_1 & X_2 & X \end{bmatrix}^T$, exogenous inputs $\begin{bmatrix} d_i & d_o \end{bmatrix}^T$ and output y as shown below:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X} \\ z_1 \\ z_2 \\ y \end{bmatrix} = \begin{bmatrix} A_{w_1} & 0 & B_{w_2}C & 0 & B_{w_1} & B_{w_1}D \\ 0 & A_2 & 0 & B_{w_2} & 0 & B_{w_2} \\ 0 & 0 & A & B & 0 & B \\ C_{w_1} & 0 & D_{w_1}C & 0 & D_{w_1} & D_{w_1}D \\ 0 & C_{w_2} & 0 & D_{w_2} & 0 & D_{w_2} \\ 0 & 0 & C & 0 & I & D \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X \\ d_i \\ d_o \\ y \end{bmatrix} \tag{8.24}$$

Substituting the plant matrix in Eq. (8.24) into the Definition 16 plant matrix, we obtain the final state feedback controller K by solving the associated LMI. According to Eq. (8.24), the states are $\begin{bmatrix} X_1 & X_2 & X \end{bmatrix}^T$. Here X_1 and X_2 are the states associated with the weights, and these states are not measured directly. If we include these

weights, the order of the controller transfer function will become high. Thus, in order to design a full-state feedback controller that is practical for implementation, we have to design a decentralized controller for the plant in Eq. (8.24) only with the state X. For decentralized controller design, we define the constraints on P and Z in Definition 16 as,

$$P = \begin{bmatrix} P_1 & 0 & 0 \\ 0 & P_2 & 0 \\ 0 & 0 & P_x \end{bmatrix} \tag{8.25}$$

$$Z = \begin{bmatrix} 0 & 0 & Z_x \end{bmatrix} \tag{8.26}$$

where, P_1, P_2, P_x are positive definite matrices, and
Z_x is a matrix solved for in the optimization.

Solving the LMI in Definition 16 using YALMIP with the above P and Z, we get the final controller matrix F. Since we use cascaded control, first the inner-loop controller is designed with this method. Next, the outer-loop controller is designed using the same method, based on the assumption that the transfer function of the inner loop is one, i.e. $I(s) = 1$ (see Figure 8.2). The control gains that were calculated for the UAV are given in Chapter 10.

Chapter 9

NONLINEAR CONTROL DESIGN

In this chapter, we discuss the nonlinear control design methodology used for horizontal to vertical flight transition. The basic idea behind this methodology is to formulate an optimization problem which has a polynomial objective function and polynomial constraints. This nonlinear problem can be converted to a Linear Matrix Inequality by using Sum-of-Squares (SOS) techniques. An introduction to Sum-of-Squares and controller design is explained in detail.

9.1 Introduction to Sum-of-Squares (SOS)

In general, a convex optimization problem is defined as

$$\begin{aligned} &\max_{x} \quad bx \\ &\text{subject to} \quad Ax \in C \end{aligned} \tag{9.1}$$

The problem is a convex optimization problem if C is a convex cone and (b, A) are affine. The problem is tractable if the variable x lies in a finite-dimensional vector space and set membership tests are verifiable in polynomial time Peet (2017b). Since polynomial functions are defined in an infinite-dimensional space, we need to bound the space by changing it to a finite-dimensional vector space. This finite space can be parameterized using monomials as a basis x and b as their coefficients Peet (2017b).

We define a basis vector Z_d that has monomials of degree d or less. For example,

$$Z_2 = \begin{bmatrix} 1 & x_1 & x_2 & x_1x_2 & x_1^2 & x_2^2 \end{bmatrix}^T \tag{9.2}$$

Any polynomial p can be represented as linear in its monomials; that is,

$$p(x) = c^T Z_d(x) \tag{9.3}$$

for a vector $c \in \mathbb{R}^d$.

We know that optimization of a polynomial function is NP-hard; hence, by expressing the polynomial as a linear function of Z_d with a fixed degree d, the problem can be converted to an LMI Peet (2017b). Now, problem 9.1 becomes:

$$\begin{aligned} &\max_x \quad b^T x \\ &\text{subject to} \quad A_0(y) + \sum_{i=1}^{n} x_i A_i(y) \succeq 0 \quad \forall y \in \mathbb{R}[x] \end{aligned} \tag{9.4}$$

where $A_i(y)$ are matrix functions of polynomial y, and n is the number of constraints.

Definition 17 *A polynomial $p(x) \in \mathbb{R}[x]$ ($\mathbb{R}[x]$ is the polynomial space with real coefficients) is a Sum-of-Squares, denoted by $p \in \sum_s$, if there exist polynomials $g_i(x) \in \mathbb{R}[x]$, $i = 1, ..., k$, such that*

$$p(x) = \sum_{i=1}^{k} g_i(x)^2$$

The polynomial optimization problem 9.4 with finite variables in which $p(x) = b^T x \in \sum_s$ can be easily converted to an LMI optimization problem. The following result and definition are used to formulate the LMI optimization problem Peet (2017b).

Consider a positive matrix $P = Q^T Q > 0$ where Q is any nonzero matrix, then:

$$\begin{aligned} V(x) &= Z_d(x)^T P Z_d(x) \\ P &= Q^T Q, \\ V(x) &= Z_d(x)^T Q^T Q Z_d(x) \\ &= (Q Z_d(x))^T (Q Z_d(x)) \geq 0 \end{aligned}$$

Definition 18 *For a polynomial $V(x)$,*

$$V(x) \geq 0 \qquad \forall x \in \mathbb{R}^n$$

if there exists a positive matrix $P > 0$ such that,

$$V(x) = Z_d(x)^T P Z_d(x)$$

Then the polynomial is a Sum-of-Squares polynomial, i.e., $V \in \sum_s$.

9.2 Controller Design

The Sum-of-Squares (SOS) method can only be used for the polynomial optimization problem 9.4, and this is one of the reasons to represent the attitude kinematics in terms of Modified Rodrigues Parameters (MRP). The nonlinear model of the UAV dynamics in Eq. 3.22 and 3.29, which we use for the control design, is of the form,

$$\dot{x} = f(x) + g(x)u \tag{9.5}$$

For an asymptotically stable system, there exists a Lyapunov function $V(x)$ that satisfies the following conditions Khalil (1996):

$$\begin{aligned} V(x) &> 0 \quad \forall x \neq 0 \\ V(0) &= 0 \\ \dot{V}(x) &< 0 \quad \forall x \neq 0 \end{aligned} \tag{9.6}$$

For controller synthesis, the control inputs are computed based on the constraint below:

$$\dot{V} = \nabla V^T \dot{X} \qquad < 0 \tag{9.7}$$

$$= \nabla V^T (f(x) + g(x)u) \qquad < 0 \tag{9.8}$$

$$= \nabla V^T (f(x)) + \nabla V^T (g(x)u) \qquad < 0 \tag{9.9}$$

Here, $V(x)$ and $\dot{V}(x)$ are polynomials in Sum-of-Squares form, hence this problem can be solved using the SOSMOD toolbox in MATLAB Peet (2017b). For a known Lyapunov function, the controller u can be computed, and conversely, for a known controller u, the Lyapunov function can be computed. In Tsiotras (1995), various Lyapunov functions for different attitude representations and the corresponding control laws are given, but tuning of the control gains was not discussed. Modified Rodrigues Parameters are used for attitude representation in our problem, hence the corresponding Lyapunov function below is chosen from Tsiotras (1995):

$$V(x) = \frac{1}{2}(I_1\, p^2 + I_2\, q^2 + I_3\, r^2) + 2(\sigma_1^2 + \sigma_2^2 + \sigma_3^2) \tag{9.10}$$

where, (I_1, I_2, I_3) denote the moment of inertia of the UAV along $x_b - y_b - z_b$ respectively, and $(\sigma_1, \sigma_2, \sigma_3)$ denote the Modified Rodrigues Parameters.

The controller is defined as:

$$u(x) = QZ_d(x) \tag{9.11}$$

where $Z_d(x)$ is a vector of monomials of order d,
Q is the matrix $(m \times d)$ to be solved for the optimization problem 9.12 below.

The controller $u(x)$ is a polynomial that is not in Sum-of-Squares form. For equation 9.10, the minimum order for the MRP is 3. This can be verified from the controller Eq. (22) in the paper Tsiotras (1995). Hence, the controller design problem is formulated as:

$Solve \quad Q$

$s.t.$,

$$\begin{cases} u(x) = Z_d(x)^T Q Z_d(x) \\ V(x) = \frac{1}{2}(I_1\, p^2 + I_2\, q^2 + I_3\, r^2) + 2(\sigma_1^2 + \sigma_2^2 + \sigma_3^2) \\ \dot{V} = \nabla V^T(f(x)) + \nabla V^T(g(x)u) < 0 \end{cases} \tag{9.12}$$

Initially, the controller in Eq. 9.13 below from Tsiotras (1995) is used in the above optimization problem.

$$
\begin{aligned}
u_1(x) &= -k_p p - \sigma_1^2(1 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2) \\
u_2(x) &= -k_q q - \sigma_2^2(1 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2) \\
u_3(x) &= -k_r r - \sigma_3^2(1 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2)
\end{aligned} \tag{9.13}
$$

where, (u_1, u_2, u_3) are the control inputs to the system, and
(k_p, k_q, k_r) are the positive control gains.

By changing the degree of the monomials of x, different nonlinear controllers can be formed. Note that the SOS optimization problem is solvable in polynomial time, hence the degree of the polynomial determines the time to solve the optimization problem. Moreover for the attitude control problem, it is found that the coefficients of monomials with degree greater than 3 are approximately zero.

The computed controller with monomials of maximum degree 3 is given below:

$$u_1(x) = -k_{3p}p^3 - k_{2p}p^2 - k_{1p}p - \sigma_1^2(1 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2) \tag{9.14}$$

$$u_2(x) = -k_{3q}q^3 - k_{2q}q^2 - k_{1q}q - \sigma_2^2(1 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2) \tag{9.15}$$

$$u_3(x) = -k_{3r}r^3 - k_{2r}r^2 - k_{1r}r - \sigma_3^2(1 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2) \tag{9.16}$$

Chapter 10

SIMULATION AND EXPERIMENTAL RESULTS

In this chapter, the performance trade-offs of the designed robust controller at hover are demonstrated in simulation results. These trade-offs are explained using the sensitivity and complementary sensitivity plots for the linearized system (7.1). Experimental results for the linear controller at hover are described. In addition, the simulated closed-loop response of the system with non-zero initial conditions for the nonlinear controller for horizontal-to-vertical flight transition are shown.

10.1 Robust Controller

In this thesis, the cascaded control methodology is used; hence, the frequency response of each loop is discussed below.

Inner-Loop Frequency Response

The inner loop is a MIMO system; hence, in each plot, the maximum and minimum singular values over all frequencies are shown.

The sensitivity and complementary sensitivity plots of the inner loop both with and without an integrator are shown in Figure 10.1. The plot in Figure 10.1i suggests that the system without an integrator attenuates approximately 28 dB of the output disturbances with a frequency content below 0.1 rad/s. Similarly, for the same system, Figure 10.1ii suggests that sensor noise with frequencies above 100 rad/s is attenuated by approximately 44 dB.

By adding an integrator to the controller, the frequency response at low frequencies can be improved. The plot in Figure 10.1iii shows that it attenuates approximately 40

dB of the output disturbances with a frequency content below 0.1 rad/s, and Figure 10.1iv shows that sensor noise attenuation is the same as in the system without an integrator. However, by adding an integrator, the output disturbance at 0.7428 rad/s is amplified by 3.15 dB (Figure 10.1iii) and the sensor noise at 0.6 rad/s is amplified by approximately 5 dB (Figure 10.1iv).

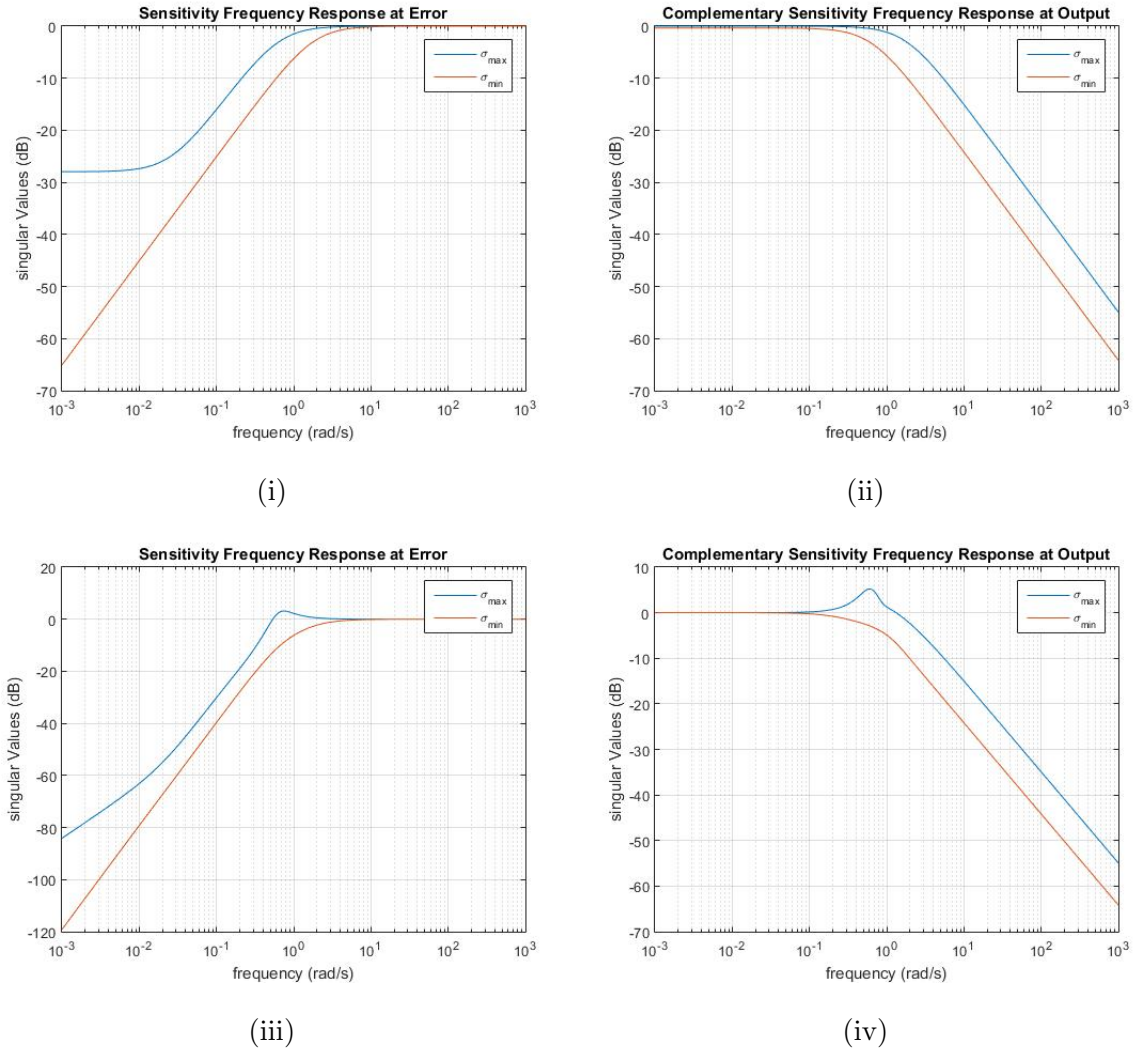
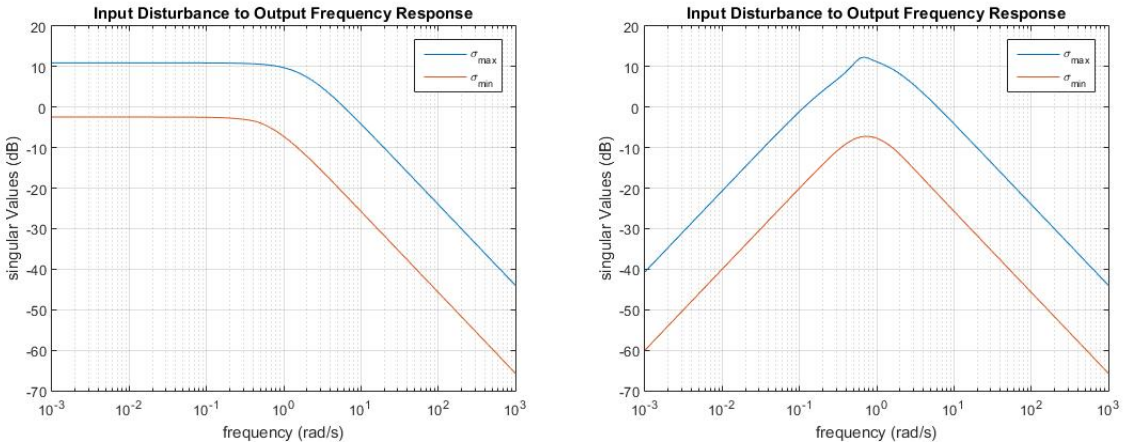


Figure 10.1: Inner-Loop Frequency Response of the Closed-Loop System I

The frequency response plot of the transfer function from input disturbance to output is given in Figure 10.2. The plot in Figure 10.2i suggests that the controller

without an integrator might amplify the input disturbances by 10 dB. If we add an integrator, then it will attenuate input disturbances with frequencies below 0.1 rad/s (see Figure 10.2ii).



(i) (ii)

Figure 10.2: Inner-Loop Frequency Response of the Closed-Loop System II

Outer-Loop Frequency Response

The outer loop is an integrator for all inputs and outputs; hence, the maximum and minimum singular values are the same.

The sensitivity and complementary sensitivity plots of the outer loop both with and without an integrator are shown in Figure 10.3. The plot in Figure 10.3i suggests that the system without an integrator attenuates approximately 20 dB of the output disturbances with a frequency content below 0.1 rad/s. Similarly, for the same system, Figure 10.3ii suggests that sensor noise with frequencies above 100 rad/s is attenuated by approximately 40 dB.

By adding an integrator to the controller, the frequency response at low frequencies can be improved. The plot in Figure 10.3iii shows that it attenuates approximately 40 dB of the output disturbance with a frequency content below 0.1 rad/s, and Figure

10.3iv shows that sensor noise attenuation is the same as in the system without an integrator. However by adding an integrator, the output disturbances at 1.46 rad/s are amplified by 1.12 dB and the sensor noise at 0.85 rad/s is amplified by approximately 3.2 dB (see Figure 10.3iv).

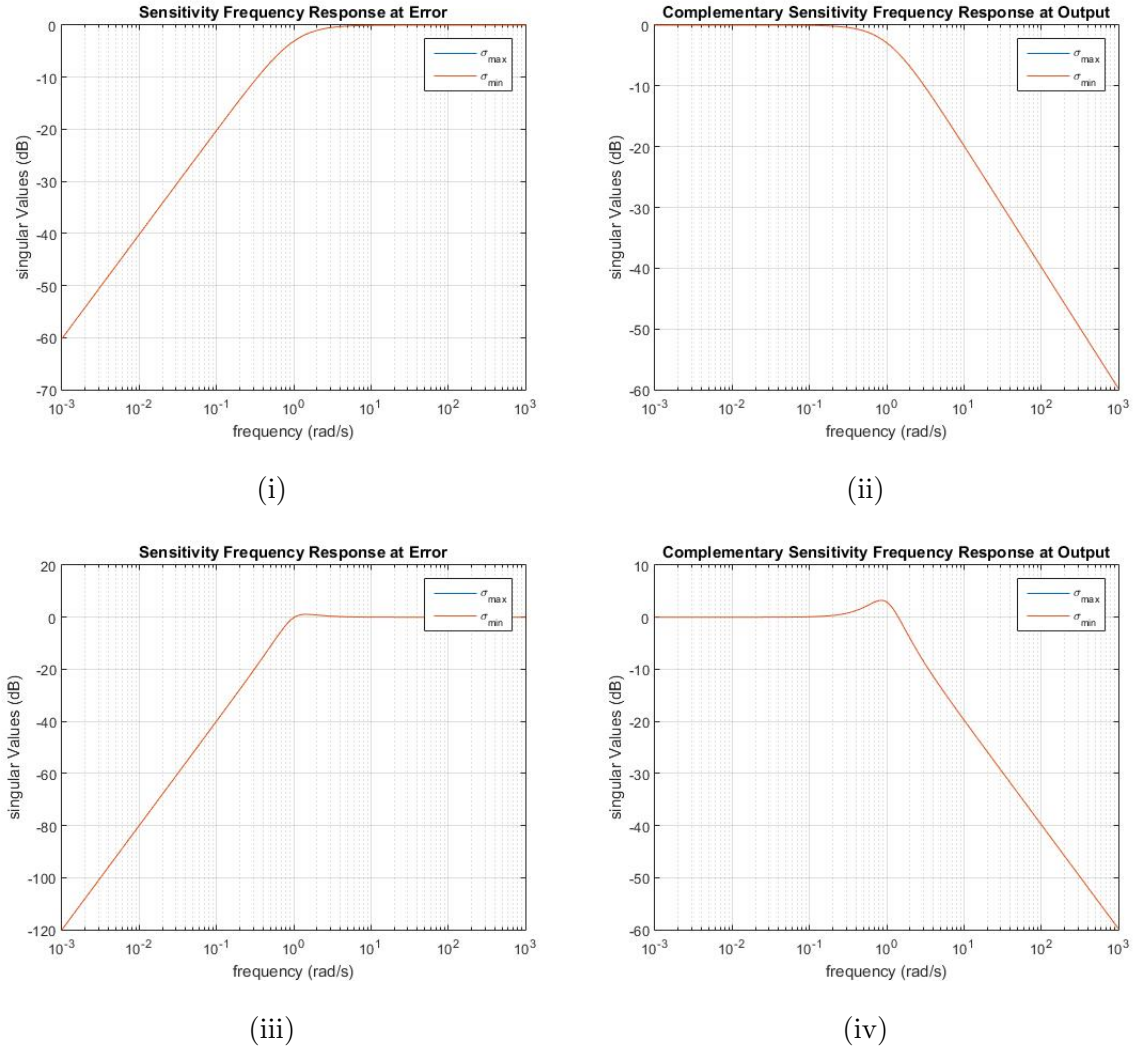
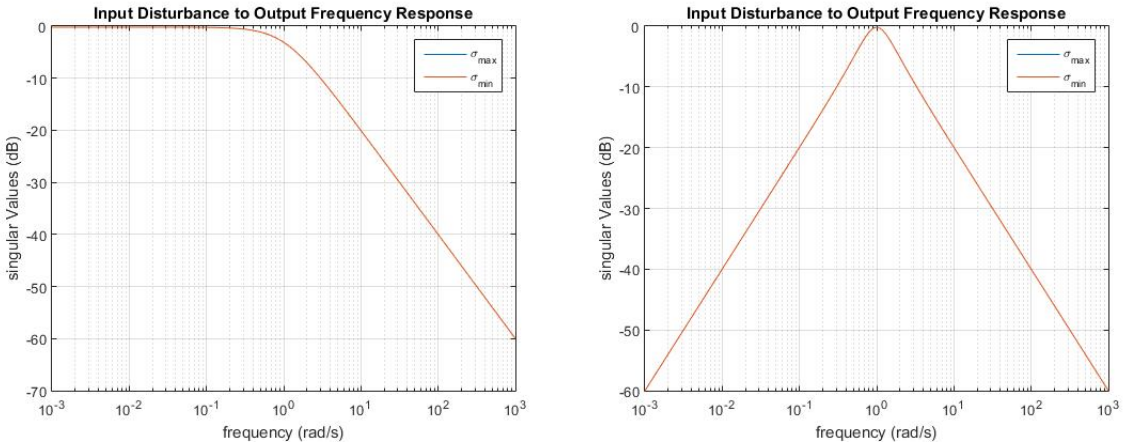


Figure 10.3: Outer-Loop Frequency Response of the Closed-Loop System I

The frequency response plot of the transfer function from the input disturbance to the output is given in Figure 10.4. The plot in Figure 10.4i suggests that the controller without an integrator will allow disturbances with frequencies below 1 rad/s through

the system without attenuation. If we add an integrator, then it will attenuate input disturbances with frequencies below 1 rad/s (see Figure 10.4ii).



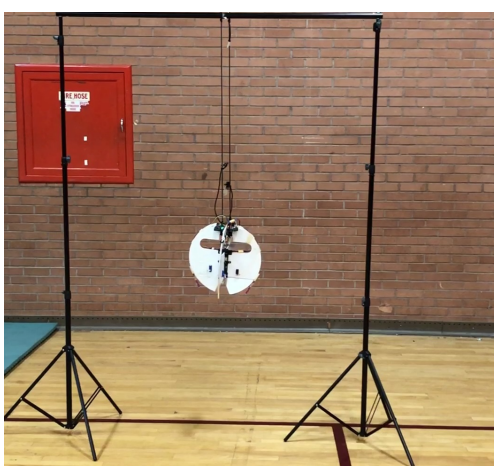
(i) (ii)

Figure 10.4: Outer-Loop Frequency Response of the Closed-Loop System II

Experimental Validation of Robust Controller

The H_∞ controller was implemented on the fabricated spherical UAV for experimental validation. The control objective is for the UAV to hover at a specified height in the plane parallel to the ground.

Since it is challenging to control the UAV to fly from the ground to an equilibrium hovering state, the UAV was tethered to a supporting frame during experiments (see Figure 10.5) to validate the performance of the controller. This setup restricts the UAV to move within the $x_b - y_b$ plane. As shown by the snapshots of the experiment in Figure 10.5, the robust controller successfully drove the UAV to hover at a specified height above the ground.



(i) $t = 0\ s$

(ii) $t = 3\ s$

(iii) $t = 5\ s$

(iv) $t = 7\ s$

Figure 10.5: Snapshots of experimental validation of the H_∞ controller at different times.

10.2 Nonlinear Controller

In this thesis, the nonlinear controller is designed to produce a transition from horizontal to vertical flight by the spherical UAV. The linearized model (7.1) is valid for system states and control inputs near an equilibrium point; hence, we need to use a gain scheduling method to design the controller for flight transitions at each

desired equilibrium point. The controller performance can be improved by designing the controller for the nonlinear mathematical model (3.22) and (3.29).

As described in Chapter 9, the SOS method is used to synthesize the nonlinear controller. Using this method, the controller is computed as:

$$\tau_\phi = -0.6758p - (\sigma_1 - \sigma_1^d)(1 + (\sigma_1 - \sigma_1^d)^2 + (\sigma_2 - \sigma_2^d)^2 + (\sigma_3 - \sigma_3^d)^2)$$

$$\tau_\theta = -0.6758q - (\sigma_2 - \sigma_2^d)(1 + (\sigma_1 - \sigma_1^d)^2 + (\sigma_2 - \sigma_2^d)^2 + (\sigma_3 - \sigma_3^d)^2)$$

$$\tau_\psi = -0.6758r - (\sigma_3 - \sigma_3^d)(1 + (\sigma_1 - \sigma_1^d)^2 + (\sigma_2 - \sigma_2^d)^2 + (\sigma_3 - \sigma_3^d)^2)$$

where $(\tau_\phi, \tau_\theta, \tau_\psi)$ are the torque control inputs, and
$(\sigma_1^d, \sigma_2^d, \sigma_3^d)$ is the desired attitude represented in MRP's.

To validate the nonlinear controller in simulation, the above controller equations are substituted into equations 3.29. The initial and final conditions for the simulated system are defined below:

$$\begin{bmatrix} \phi_i & \theta_i & \psi_i & p_i & q_i & r_i \end{bmatrix} = \begin{bmatrix} 25 & 25 & 10 & 0 & 0 & 0 \end{bmatrix} \tag{10.1}$$

$$\begin{bmatrix} \phi_f & \theta_f & \psi_f & p_f & q_f & r_f \end{bmatrix} = \begin{bmatrix} 90 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10.2}$$

where $[\phi, \theta, \psi]$ are the Euler angles in deg,
and $[p, q, r]$ are the body angular rates in deg/s.

Figure 10.6 shows the control inputs used over time to achieve the desired attitude. The input torques are shown in the subplots. From this figure, it is evident that the nonlinear controller requires high-bandwidth actuators for operation due to the the sharp changes in torque near $t = 0$. The control signals in Figure 10.6 cannot be implemented in the actuators on the fabricated UAV. Hence, the nonlinear controller is not validated experimentally on the fabricated UAV. In Figure 10.7, the body angular rates of the system for the applied control inputs are plotted. All angular rates stabilize to zero, as desired. Finally, Figure 10.8 shows that the Euler angles

of the system successfully converge to the desired attitude from the chosen initial condition.

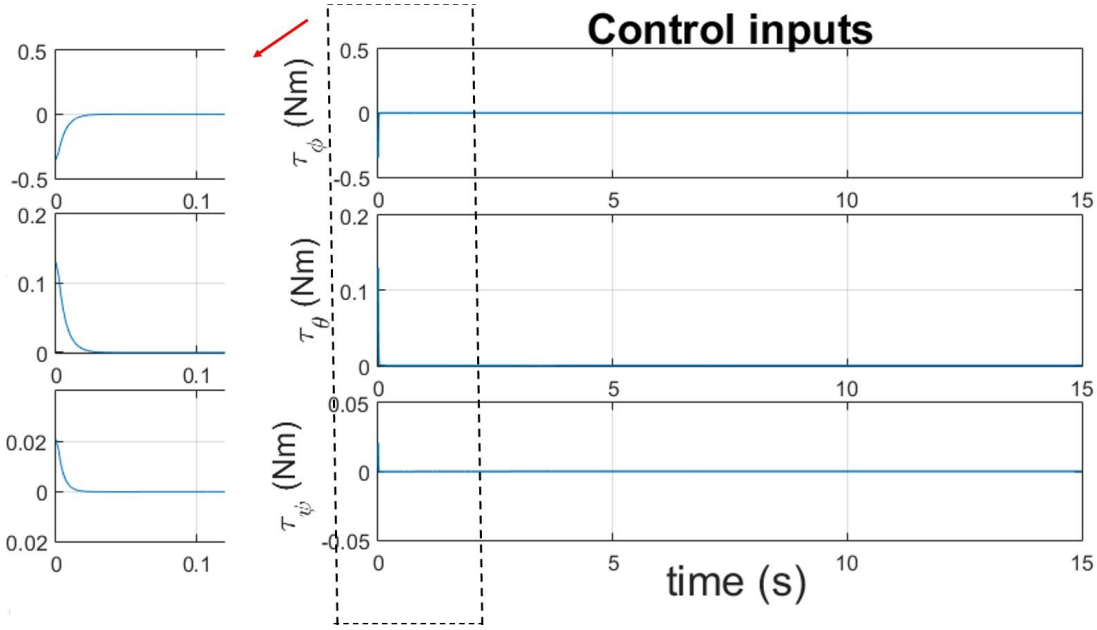


Figure 10.6: Torque control inputs vs. time for nonlinear controller

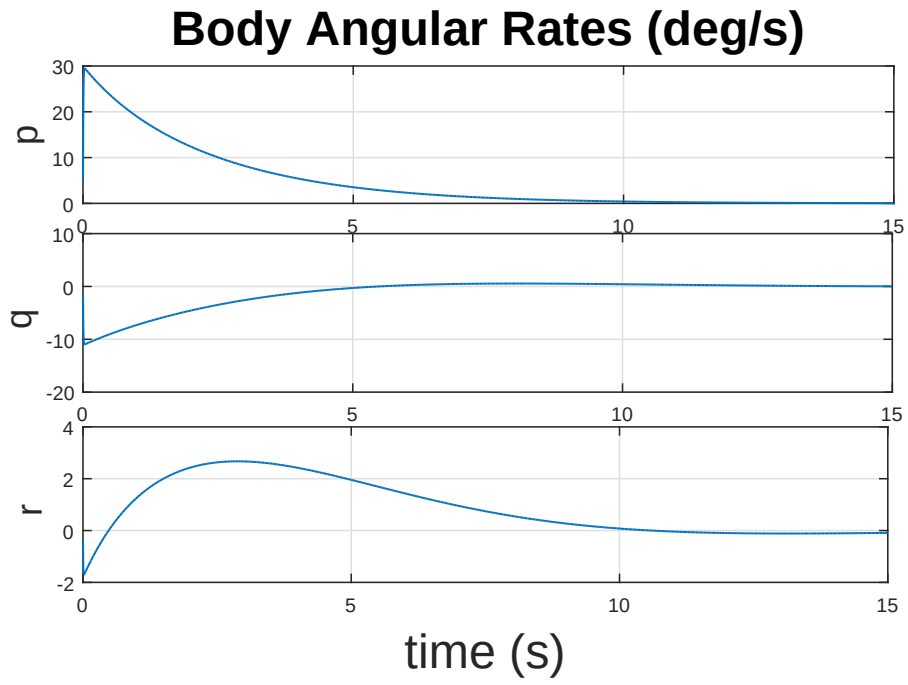


Figure 10.7: Body angular rates vs. time for nonlinear controller

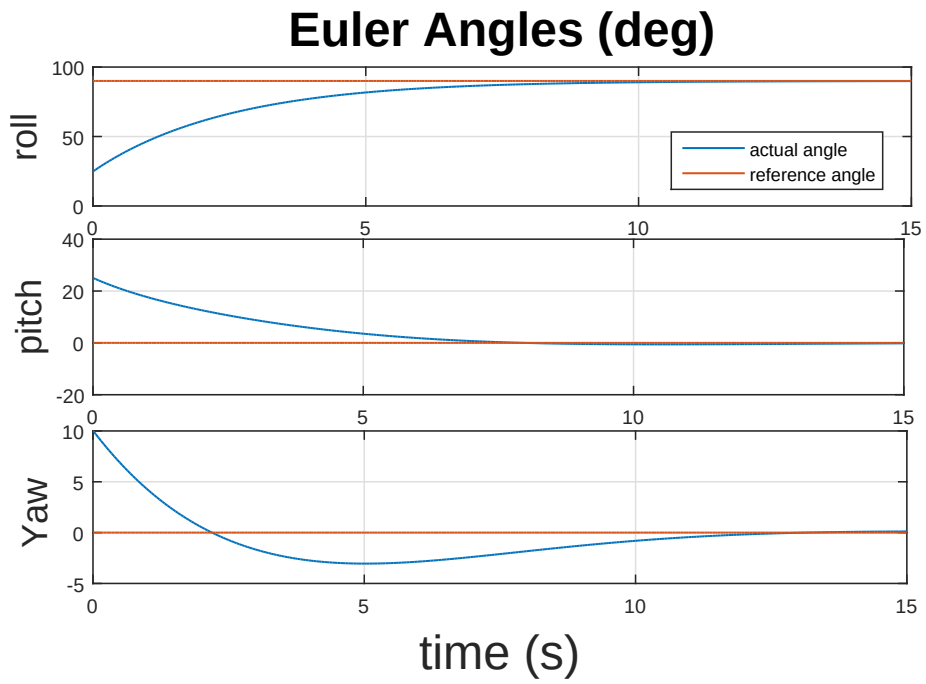


Figure 10.8: Euler angles vs. time for nonlinear controller

Chapter 11

CONCLUSION AND FUTURE WORK

The thesis describes the fabrication, dynamical modeling, and control of a spherical tail-sitter UAV. The main contributions of this thesis are the design of a frequency-dependent robust full-state feedback controller for the UAV and implementation on the fabricated vehicle. In this chapter, conclusions and future directions of this thesis are presented.

11.1 Conclusion

As described in the Introduction, research has been conducted on controller design for tail-sitter UAVs, and several companies such as Martin UAV and Aerovel are developing these types of vehicles for commercial applications. Tail-sitter UAVs are capable of long cruise times like fixed-wing vehicles, but do not require a runway for take-off and landing. Their VTOL and cruise capabilities make them suitable for a wide range of applications.

However, these types of vehicles have degraded stability properties in VTOL and cruise applications. This drawback can be overcome by designing effective controllers for attitude control. The most challenging part of this thesis was the robust controller validation on the developed UAV. Little research has been previously conducted on robust attitude control design for tail-sitters. The controller was designed to produce good robustness properties in the presence of disturbances and noise.

In this thesis, the experimental design of a spherical tail-sitter UAV was described in Chapter 2. The fabrication of each part and the complete system were discussed in detail. The nonlinear model of the UAV dynamics was derived in Chapter 3. The

controller bandwidth for the system was chosen based on effects of the control surfaces on the system. The effects of the aerodynamic and propeller forces were derived. The different types of attitude kinematics were discussed in Chapter 4. The Modified Rodrigues Parameters were used to represent the attitude kinematics in this work. In Chapter 5, the procedures for calculating the model parameters were explained in detail. These parameters were used to determine the characteristics of the system model, such as its pole and zero locations.

The nonlinear model with estimated parameters was coded in Simulink for analysis. The trim conditions for the system at a target equilibrium point were calculated using the Simulink model. The theoretical explanation for this calculation and syntax for implementation in Simulink were explained in Chapter 6. The nonlinear model was linearized about the trim conditions and equilibrium states using MATLAB.

In Chapter 7, a brief mathematical introduction to vector and matrix norms was given. This chapter also outlined a conceptual explanation and method to compute the H_2 and H_∞ norms. The H_2 and H_∞ controller design methodology using Linear Matrix Inequalities (LMI) was shown in Chapter 8. Moreover, the LMI for mixed sensitivity H_∞ control design was derived and then solved using the YALMIP MATLAB Toolbox. A robust controller was designed for the UAV in a hover state. For vertical-to-horizontal flight transitions, a nonlinear controller was designed using a Sum-Of-Squares (SOS) technique, as explained in Chapter 9.

In Chapter 10, the results and performance trade-offs of the robust controllers were shown in simulation. Since the linearized model is a Multi-Input Multi-Output (MIMO) system, the trade-offs were illustrated using singular value plots, which characterized the robustness properties of the controller. The robust controller was implemented on the fabricated UAV in a tethered setup for experimental validation. Finally, the time response of the nonlinear controller for vertical to horizontal transi-

tion of the UAV was shown in simulation.

11.2 Future Work

Several possible ways to improve the work in this thesis are listed below.

Material: Foam board was used to build the body components because it is light and cheap. Alternatively, it can be replaced with carbon fiber to enforce sturdiness. While carbon fiber is costly and slightly heavier than foam board, a carbon fiber body would be strong enough to perform maneuvers like rolling on the ground and hovering in contact with a wall.

Sensors: Since this thesis is focused on attitude control, the sensors used on the UAV are an accelerometer, gyroscope and magnetometer. These sensors do not enable control of the UAV to hover at a particular point. By adding a GPS/positioning system, we can create a outermost position control loop to enforce hovering at a target point.

Robust Control Design: The controller designed in this thesis is robust to input and output disturbances and sensor noise. Since the model parameters of the system are not accurately known, we can design a controller that is also robust to parametric uncertainties.

Nonlinear Control Design: In this thesis, the nonlinear control design ensures smooth transition from vertical to horizontal flight. However, the controller bandwidth is too high for the actuator, and hence we need to modify the corresponding optimization problem to constrain the controller bandwidth.

REFERENCES

Series 1580 dynamometer and thrust stand datasheet, RCbenchmark.com, v1.1 (2016).

"Ansys academic research mechanical", Software Release 18.1 (2017).

Apostol, T. M., Mathematical analysis (Addison Wesley Publishing Company, 1974).

Argyle, M. E., Modeling and Control of a Tailsitter with a Ducted Fan (Brigham Young University, 2016).

Argyle, M. E., R. W. Beard and S. Morris, "The vertical bat tail-sitter: dynamic model and control architecture", in "American Control Conference (ACC), 2013", pp. 806–811 (IEEE, 2013).

Beard, R. W. and T. W. McLain, Small unmanned aircraft: Theory and practice (Princeton university press, 2012).

Boyd, S., L. El-Ghaoui, E. Feron, V. Balakrishnan and E. Yaz, "Linear matrix inequalities in system and control theory", Proceedings of the IEEE **85**, 4, 698–699 (1997).

Carapau, R. S., A. V. Rodrigues, M. M. Marques and V. Lobo, "Unmanned aerial systems in military environments: The benefits of interoperability", Scientific Bulletin" Mircea cel Batran" Naval Academy **20**, 1, 136 (2017).

Chana, W. F. et al., "World's first vtol airplane convair/navy xfy-1 pogo", Tech. rep., SAE Technical Paper (1996).

Cleva, S., A. I. Bogani and L. Pivetta, "A low-cost high-performance embedded platform for accelerator controls", in "Proc. PCaPAC", pp. 68–70 (2012).

Coach, U., "The top 100 drone companies to watch in 2018", `https://uavcoach.com/drone-companies/#guide-0`, online (2018).

Dixon, W. E., A. Behal, D. M. Dawson and S. P. Nagarkatti, Nonlinear control of engineering systems: A Lyapunov-based approach (Springer Science & Business Media, 2013).

Duan, G.-R. and H.-H. Yu, LMIs in control systems: analysis, design and applications (CRC press, 2013).

Dullerud, G. E. and F. Paganini, A course in robust control theory: a convex approach, vol. 36 (Springer Science & Business Media, 2013).

Elevate, U., "Fast-forwarding to a future of on-demand urban air transportation", `Uber.com` (2016).

Erik Welsh, a. J. S., Jason Kridner, "Beaglebone blue", `https://beagleboard.org/blue`, online (2016).

Gordon, Y., "Vertical reality - history of vtol", JETS https://www.scribd.com/document/294024933/The-History-of-VTOL, online (2016).

Greenwood, D. T., Principles of dynamics (Prentice-Hall Englewood Cliffs, NJ, 1988).

Habeck, J. and P. Seiler, "Moment of inertia estimation using a bifilar pendulum", Retrieved from the University of Minnesota Digital Conservancy, http://hdl.handle.net/11299/182514 (2016).

Hogge, J. V., "Development of a miniature vtol tail-sitter unmanned aerial vehicle", All Theses and Dissertations. 1373 https://scholarsarchive.byu.edu/etd/1373, online (2008).

Khalil, H. K., "Nonlinear systems", Prentice-Hall, New Jersey **2**, 5 (1996).

Krogh, K. J., "Developing a framework for control of agile aircraft platforms in autonomous hover", Tech. rep., WASHINGTON UNIV SEATTLE DEPT OF AERONAUTICS AND ASTRONAUTICS (2009).

Lofberg, J., "Yalmip: A toolbox for modeling and optimization in matlab", in "Computer Aided Control Systems Design, 2004 IEEE International Symposium on", pp. 284–289 (IEEE, 2004).

Lofberg, J., "Yalmip", https://yalmip.github.io/, online (2017).

Loh, W. K. and J. Jacob, "Modeling and attitude control analysis of a spherical vtol aerial vehicle", in "51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition", p. 1092 (2013).

Lundström, P., S. Skogestad and Z.-Q. Wang, "Performance weight selection for h-infinity and μ-control methods", Transactions of the Institute of Measurement and Control **13**, 5, 241–252 (1991).

Mackenroth, U., Robust control systems: theory and case studies (Springer Science & Business Media, 2013).

Malandrakis, K., R. Dixon, A. Savvaris and A. Tsourdos, "Design and development of a novel spherical uav", IFAC-PapersOnLine **49**, 17, 320–325 (2016).

Markley, F. L. and J. Crassidis, "Attitude estimation using modified rodrigues parameters", in "Proceedings of the Flight Mechanics/Estimation Theory Symposium. Greenbelt, USA", (1996).

Moloney, M., Lectures: Bifilar, trifilar pendulum theory (https://www.rose-hulman.edu/~moloney/PH235_F06/Bifilar_06_torques.doc, 2006), URL https://www.rose-hulman.edu/~moloney/PH235_F06.

Mueller, M., "ecalc - the most reliable rc calculator on the web", https://www.ecalc.ch/index.htm, online (2018).

Parker, G., "Drone vs. uav - what is the difference?", https://wiki.ezvid.com/m/drone-vs-uav-what-is-the-difference-_2FJYp_SrUkP-, online; accessed 24 January 2018 (2018).

Peet, D. M., Lectures: LMI Methods in Optimal and Robust Control (http://control.asu.edu, 2017a), URL http://control.asu.edu.

Peet, D. M., "Sosmod toolbox", http://control.asu.edu/Software/SOSMOD_vMAE598.zip (2017b).

Prajna, S., A. Papachristodoulou and P. A. Parrilo, "Introducing sostools: A general purpose sum of squares programming solver", in "Decision and Control, 2002, Proceedings of the 41st IEEE Conference on", vol. 1, pp. 741–746 (IEEE, 2002).

Pugh, C. C. and C. Pugh, Real mathematical analysis, vol. 2011 (Springer, 2002).

RCbenchmark.com, "Measure motor and propeller efficiency, for quads, uavs...", https://www.youtube.com/watch?time_continue=4&v=O4ORChwM68g, online (2015).

Rodriguez, A. A., Analysis and Design of Multivariable Feedback Control Systems, vol. 1 (CONTROL3D, L.L.C., Tempe, 2004).

Shuster, M. D., "A survey of attitude representations", Navigation **8**, 9, 439–517 (1993).

Skogestad, S. and I. Postlethwaite, Multivariable feedback control: analysis and design, vol. 2 (Wiley New York, 2007).

Stone, R. H., "Control architecture for a tail-sitter unmanned air vehicle", in "Control Conference, 2004. 5th Asian", vol. 2, pp. 736–744 (IEEE, 2004).

Sun, H., K. Hou and Q. Jia, "Development, analysis and control of a spherical aerial vehicle.", Journal of Vibroengineering **15**, 2 (2013).

Terzakis, G., M. Lourakis and D. Ait-Boudaoud, "Modified rodrigues parameters: an efficient representation of orientation in 3d vision and graphics", Journal of Mathematical Imaging and Vision **60**, 3, 422–442 (2018).

Thacker, N. and A. Lacey, "Tutorial: The kalman filter", Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester. TiNA (1998).

Toivonen, H., "Lecture notes on robust control by state-space methods", Available at: users. abo. fi/htoivone/courses/robust/hsem. pdf (1995).

Triantafyllou, M. S. and F. S. Hover, Maneuvering and control of marine vehicles (Massachusetts of Institute of Technologyq, 2003).

Tsiotras, P., "New control laws for the attitude stabilization of rigid bodies", in "Automatic Control in Aerospace 1994 (Aerospace Control'94)", pp. 321–326 (Elsevier, 1995).

Uranchimeg, T., B. Battseren and W. Hardt, "Unmanned aerial vehicle based automated inspection system for high voltage transmission lines", , 1, 28–32 (2017).

Wie, B., Space vehicle dynamics and control (Aiaa, 1998).

Young Bae Lee, "Spherical Flight Vehicle (Flying ball, Sphere drone, Single rotor, VTOL...)", https://www.youtube.com/watch?v=55d5ppwQBQ4&t=45s, online (2012).

Zhao, S., "Time derivative of rotation matrices: A tutorial", arXiv preprint arXiv:1609.06088 (2016).