DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ROBUST CONTROLLER DESIGN AND IMPLEMENTATION FOR AN AGRICULTURAL UAV

by
Mertcan AKIN

February, 2023
IZMIR

ROBUST CONTROLLER DESIGN AND IMPLEMENTATION FOR AN AGRICULTURAL UAV

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Mechatronics Engineering, Mechatronics Engineering Program

by

Mertcan AKIN

February, 2023
IZMIR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**ROBUST CONTROLLER DESIGN AND IMPLEMENTATION FOR AN AGRICULTURAL UAV**" completed by **MERTCAN AKIN** under supervision of **ASSOC. PROF. DR. AYTAÇ GÖREN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Aytaç GÖREN

Supervisor

Prof. Dr. Levent ÇETİN

(Jury Member)

Assoc. Prof. Dr. Şahin YAVUZ

(Jury Member)

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Assoc. Prof. Dr. Aytaç GÖREN who supported and guided me throughout this study. Also, I would like to thank Prof. Dr. Ahmed RACHID from Université de Picardie Jules Verne for his support and experience in the "Implementation of Sensor Filters and Altitude Estimation of Unmanned Aerial Vehicle using Kalman Filter".

I sincerely thank my colleague Burak ER for his suggestions and technical informations.

I would like to thank my lab colleagues, Furkan YAKIN and Adem SELVI, who helped with the tests. I would also like to thank my friend Umut AKTEPE for his support and motivation.

Finally, I would like to thank my family for supporting and encouraging me in every part of my life.

Mertcan AKIN

ROBUST CONTROLLER DESIGN AND IMPLEMENTATION FOR AN AGRICULTURAL UAV

ABSTRACT

Unmanned aerial vehicles (UAV) are widely used in various fields today. Especially in recent years, UAVs have become increasingly common in civilian areas such as agriculture and photography, as well as military use. UAVs are complex systems with nonlinear structure. Therefore, various studies are carried out on the control of UAVs in industry and academic studies.

In this study, the nonlinear control of a quadrotor's position and attitude is examined. In this regard, the nonlinear mathematical model of the vehicle was created in Matlab/Simulink environment with Newton-Euler method. Various effects such as motor, sensor noise and wind-induced force/torque have been considered to make this model more realistic. In order to make the quadrotor model closer to reality, experiments and measurements were made on the mechanical properties of the vehicle, motor characteristics and sensor noises. For the control of the vehicle, a nonlinear controller, the sliding mode contoller, is examined. Sliding mode contol is a control method that tries to keep the system response on the sliding surface using a switching signal. If the system states go out of the sliding surface, it is tried to be brought back to the sliding surface with a sudden control signal. PID control is also examined to compare the performance of the sliding mode controller against disturbances such as sensors and wind. The performance of the two controllers against different trajectories and disturbances is compared.

Keywords: Unmanned aerial vehicle, modelling, robust control, sliding mode control, PID control

TARIMSAL BİR İHA İÇİN DAYANIKLI KONTROLCÜ TASARIMI VE UYGULAMASI

ÖZ

İnsansız hava araçları günümüzde çeşitli alanlarda yaygın olarak kullanılmaktadır. Özellikle son yıllarda askeri kullanımın yanında tarım ve fotoğrafçılık gibi sivil alanlarda giderek yaygınlaşmışlardır. İnsansız hava araçları doğrusal olmayan bir yapıya sahip karmaşık sistemlerdir. Dolayısıyla endüstride ve akademik çalışmalarda insansız hava araçlarının kontrolü üzerine çeşitli çalışmalar yapılmaktadır.

Bu çalışmada dört rotorlu bir insansız hava aracının pozisyon ve duruşunun doğrusal olmayan kontrolü incelenmiştir. Bu hususta aracın doğrusal olmayan matematiksel modeli Newton-Euler yöntemiyle Matlab/Simulink ortamında oluşturulmuştur. Bu modelin gerçeğe yakın olmasını sağlamak için motor, sensör gürültüsü ve rüzgar kaynaklı kuvvet/moment gibi çeşitli etkiler göz önünde bulundurulmuştur. Quadrotor modelini gerçeğe yaklaştırmak için aracın mekanik özellikleri, kullanılan motorların karakteristiği ve sensör gürültüleri üzerine deneyler ve ölçümler yapılmıştır. Aracın kontrolü için doğrusal olmayan bir kontrolcü olan kayan kipli kontrolcü incelenmiştir. Kayan kipli kontrol, anahtarlama sinyali kullanarak sistem cevabını kayma yüzeyinde tutmaya çalışan bir kontrol yöntemidir. Sistem durumları kayma yüzeyinin dışına çıkarsa ani bir kontrol sinyali ile tekrar kayma yüzeyine getirilmeye çalışılır. Kayan kipli kontrolcünün sensör ve rüzgar gibi bozuculara karşı performansını karşılaştırmak için PID kontrol de incelenmiştir. Farklı yörüngeler ve farklı bozuculara karşı iki kontrolcünün performansı karşılaştırılmıştır.

Anahtar kelimeler: İnsansız hava aracı, modelleme, robust kontrol, kayan kipli kontrol, PID kontrol

CONTENTS

Page

M.Sc THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES ix
LIST OF TABLES xi

CHAPTER 1 - INTRODUCTION 1

1.1 Introduction 1
1.2 Brief History 2
1.3 Motivation 4
1.4 State of the Art 4
1.5 Thesis Structure 8

CHAPTER 2 - SYSTEM MODELING 9

2.1 Introduction 9
2.2 Reference of Frames 9
2.3 Kinematics 10
2.3.1 Euler Angles 11
2.3.2 Translational Kinematics 13
2.3.3 Rotational Kinematics 13
2.4 Control Inputs 14
2.4.1 Thrust 15
2.4.2 Roll Moment 15
2.4.3 Pitch Moment 15
2.4.4 Yaw Moment 16
2.5 Dynamics of Quadrotors 17
2.5.1 Translational Dynamics 18
2.5.2 Rotational Dynamics 19

2.6 Aerodynamics Model 20
2.7 State Space Model 21
2.8 Pesticide Spraying Model 23

CHAPTER 3 - SIMULINK MODEL AND SIMULATION 24

3.1 Model Spesifications 24
3.1.1 Mechanical Structure Spesifications 25
3.1.2 Bifilar Pendulum and Moment of Inertia Measurement 25
3.1.3 Motor Properties 29
3.2. Sensor Model 33
3.3. Wind Model 36
3.3.1 Wind Shear Model 36
3.3.2 Discrete Gust Model 37
3.3.3 Dryden Continous Turbulence Model 38
3.4. Open Loop System Test 39
3.5. Simulation Environment 40

CHAPTER 4 - CONTROL SYSTEM DESIGN 42

4.1 Control Systems Introduction 42
4.2 PID Control 43
4.2.1 Attitude Controller 43
4.2.2 Altitude Controller 45
4.2.3 Position Controller 45
4.2.4. PID Parameter Tuning 46
4.3 Sliding Mode Control 48
4.3.1 SMC Background 49
4.3.2 Altitude Controller 51
4.3.3 Attitude Controller 52
4.3.4 Position Controller 53

CHAPTER 5 - RESULTS 54

5.1 Simulation Environment and Conditions 54

5.2 Attitude Control Results 54
5.3 Position Control Results 58
5.4 Control Inputs Resuts 65
5.5 Parametric Uncertainty 67

CHAPTER 6 - CONCLUSION 69

REFERENCES 70
APPENDICES 75

LIST OF FIGURES

Page

Figure 1.1 Lawrence and Sperry's autonomous UAV (Nonami, et al., 2010) 2
Figure 1.2 Yamaha R50 Prototype (Yamaha Motors, n.d.) 3
Figure 1.3 Agricultural UAV (Croptracker) 3
Figure 1.4 Control Systems Scheme 5
Figure 2.1 Reference of frames 9
Figure 2.2 Quadcopter Schematic 10
Figure 2.3 Euler Angles 12
Figure 2.4 Roll Motion 15
Figure 2.5 Pitch Motion 16
Figure 2.6 Yaw Motion 16
Figure 2.7 Mass Reduction Model 23
Figure 3.1 ACRL Quadcopter Prototype 24
Figure 3.2 Mass Distribution Difference 26
Figure 3.3 Bifilar Pendulum Model 26
Figure 3.4 Bifilar Pendulum and Sample 27
Figure 3.5 Measurements in X, Y, Z axises 28
Figure 3.6 Motor Thrust Platfrom 29
Figure 3.7 Appdesigner User Interface 29
Figure 3.8 UI Measurement Result 30
Figure 3.9 Thrust Generation Diagram 31
Figure 3.10 Thrust Mechanism CAD Model 31
Figure 3.11 Thrust and Drag Coefficient Curve Fit 32
Figure 3.12 IMU Mechanical Model (Analog Devices, 2014) 33
Figure 3.13 Sensor Model 33
Figure 3.14 Meausered and Modeled Results 34
Figure 3.15 Frequency Domain Results 35
Figure 3.17 Filtered Roll and Pitch Angle 36
Figure 3.18 Wind Model for the UAV 36
Figure 3.19 Gust Model Graph 37
Figure 3.20 Dryden Model Outputs 39

Figure 3.21 Test Environment 40
Figure 3.22 External Datalogger 40
Figure 3.23 Open Loop Model 41
Figure 3.24 Open Loop Test and Simulation Comparison 41
Figure 4.1 General Controller Diagram 42
Figure 4.2 PID Angle Controller Model 43
Figure 4.3 PID Angular Velocity Controller Model 44
Figure 4.4 PID Altidude Controller Model 45
Figure 4.5 Position Controller Model 45
Figure 4.6 Genetic Algorithm Flowchart 47
Figure 4.7 SMC Altitude Controller Model 52
Figure 4.8 SMC Attitude Controller Model 53
Figure 5.1 Angle Step Input Results without Disturbance 55
Figure 5.2 Altitude Step Input Results without Disturbance 55
Figure 5.3 Angle Step Input Results with Disturbance 57
Figure 5.4 Altitude Step Input Results with Disturbance 57
Figure 5.5 Trajectory Tracking without Disturbances in Circular Trajectory 59
Figure 5.6 Trajectory Tracking without Disturbances in Square Trajectory 60
Figure 5.7 Trajectory Tracking with Disturbances in Circular Trajectory 62
Figure 5.8 Trajectory Tracking with Disturbances in Square Trajectory 63
Figure 5.9 PID Controller Results at Different Wind Speeds 64
Figure 5.10 SMC Results at Different Wind Speeds 65
Figure 5.11 Control Efforts without Disturbance 66
Figure 5.12 Control Efforts with Disturbance 66
Figure 5.13 PID and SMC Outputs with Parametric Uncertainty 67
Figure 5.14 Control Efforts with Parametric Uncertainty 68

LIST OF TABLES

Page

Table 2.1 Reference Frame Vectors 10
Table 2.2 Kinematics Notations 11
Table 2.3 Effects on Dynamics 17
Table 2.4 State Variables 22
Table 3.1 UAV Components 24
Table 3.2 Test Sample Properties 27
Table 3.3 Sigma Profile Results 27
Table 3.4 MOI Result Table 28
Table 3.5 Mechanical Specification Results 32
Table 5.1 Controller Coefficients 53
Table 5.2 Step Input Results without Disturbance 56
Table 5.3 Step Input Results with Disturbance 58
Table 5.4 Trajectory Tracking Results without Wind Effect 61
Table 5.5 Trajectory Tracking Results with Wind Effect 64

CHAPTER 1
INTRODUCTION

1.1 Introduction

With the development of technology, unmanned aerial vehicles (UAV) are also technically improved and their use has increased. Especially with the camera shots, UAVs provide great advantages in various fields such as archeology. With the increase in reliability and features of UAVs, their use in archaeological areas has increased. Archaeologists are allowed to explore and analyze archaeological sites. By defining a flight route, archaeologists can easily obtain information about the places that can be examined, archaeological remains detection within the route or the route for rough terrain. Satellite-based images are not a very accurate choice for detecting remains (Agudo, et al., 2018). Thanks to the thermal and multispectral cameras added to the UAVs, the quality of the images obtained by archaeologists has increased. NDVI (Normalized Difference Vegetation Index), SAVI (Soil Adjusted Vegetation Index), GNDVI (Green Normalized Difference Vegetation Index) methods provide information about archaeological remains by analyzing the soil and plants on the ground.

Use of UAVs in agriculture has brought conveniences and applications. They are generally used for crop monitoring and spraying but there are new applications like tracking livestocks and sample collection. It is checked if there is disease or not by measuring body temperatures of the animals. If this control is done manually by the workers, animals may get stressed or the possibility of transmission of the disease increases because of the close contact (Merwe, et al., 2020). Another newly developing application in agriculture is sample collection. Instead of walking the whole area, UAVs can collect samples for analyzing the plants. There are companies that develop UAVs for use in agriculture. These UAVs save a great deal of time and resources. In the future, the monitoring and sampling capabilities of UAVs in agriculture will be further increased in order to provide safer food and yield. Spraying will be made more sensitive and thus environmental pollution will be reduced.

UAVs can also be used in agricultural activities on the sea. They are used in border control of farms established for seafood breeding in coastal regions. The vehicle distinguishes foreign creatures entering the cage area and informs the center. With this way, border crises experienced by fishermen are prevented (Balasubramaniam and Janardanan, 2018).

1.2 Brief History

The first UAV in history was produced by Lawrence and Sperry in 1916 as shown in Figure 1.1. One of the most important features of this UAV is the attitude control with the gyroscope on it (Nonami, et al., 2010). Studies on the use of drones for military purposes were carried out in the Second World War, but could not be continued due to the lack of technology. With the Cold War, the development of unmanned aircraft manifested itself in the form of guided missile development studies. Especially in the 90s, vehicles that can stay in the air longer and have longer ranges have been developed. This development of unmanned aerial vehicles in the military field is also reflected in civilian use. For example, UAVs are frequently used in many fields such as photography, firefighting, search and rescue, and agriculture.



Figure 1.1 Lawrence and Sperry's autonomous UAV (Nonami, et al., 2010)

One of the first uses in agriculture dates back to 1906. A farmer from New Zeland named John Chaytor used a hot air baloon, which can be controlled by ropes from the ground, to spread grass seed in a wet valley on his farm. In 1983, the Japanese Agricultural Department requested Yamaha Motor Company to develop a drone for

crop dusting (Krishna, 2018). In 1987 Yamaha released its first R50 model as shown in Figure 1.2. In 1989, the first commercial use began. In 1991 the Swedish University of Agricultural Sciences (SLU) began developing an aircraft with a camera for crop mapping (Frankelius and Johansen, 2019).



Figure 1.2 Yamaha R50 Prototype (Yamaha Motors, n.d.)

Especially in the 2010s, the use of UAVs in agriculture has increased rapidly as parts become more affordable. Besides various drone companies around the world, many people develop and use their own drones. Today, agricultural drones are used in a variety of tasks such as irrigation, spraying, nutrient analysis, insect infestation analysis, moisture analysis and harvest forecasting. With the information collected by drones, farmers can increase the efficiency of their harvests and take precautions against possible dangers. Figure 1.3 shows a drone used for spraying.



Figure 1.3 Agricultural UAV (Croptracker)

1.3 Motivation

In this study, robust control for a quadcopter agricultural UAV will be examined. Due to the camera's features to be added to the vehicle, it can be used for both archaeology and agriculture. It is important to obtain quality images for both agriculture and archaeological work. In this sense, stability and control, as well as features such as sensors and operating capacity added to UAVs, are of great importance. Both the movement and altitude of the vehicle should be constant, as the homogeneous distribution of the pesticide sprayed on plants will ensure the effect of the pesticide in the same way throughout the entire area. Given the uncertainties affecting the vehicle, such as wind effect and mass reduction (water/pesticide that the vehicle will carry), a model closer to reality will be created, and the control of the vehicle will be less affected by external effects. In this way, a more stable flight will occur.

There are several studies related to the control of unmanned aerial vehicles in the literature. PID control has been used in many studies because it is effective and easier to implement. A model is created by making some assumptions when designing a PID control. A completely perfect system cannot be made, as uncertainties such as wind, mass changes, and how efficiently the engines work are not added to the model. In robust control, a more stable system is created because these uncertainties are made by added the model. In the literature, there have been studies showing that the effect of wind on the vehicle is reduced with robust control. The UAV is modeled with unstable outdoor conditions in the simulation of control algorithms via a nonlinear dynamic model. Simulations are performed in order to evaluate the returns of control algorithms in response to the determined irregular wind effects. In addition to the effect of wind during flight, a decrease in the amount of pesticide and possible imbalances caused by spraying will also be taken into account. In addition, it is planned to examine and compare different control methods.

1.4 State of the Art

Various studies have been carried out on the control systems of UAVs, especially in the last 10 years. As can be seen in the Figure 1.4, these methods can be examined under four main headings. In this section, literature studies on some methods are reviewed.

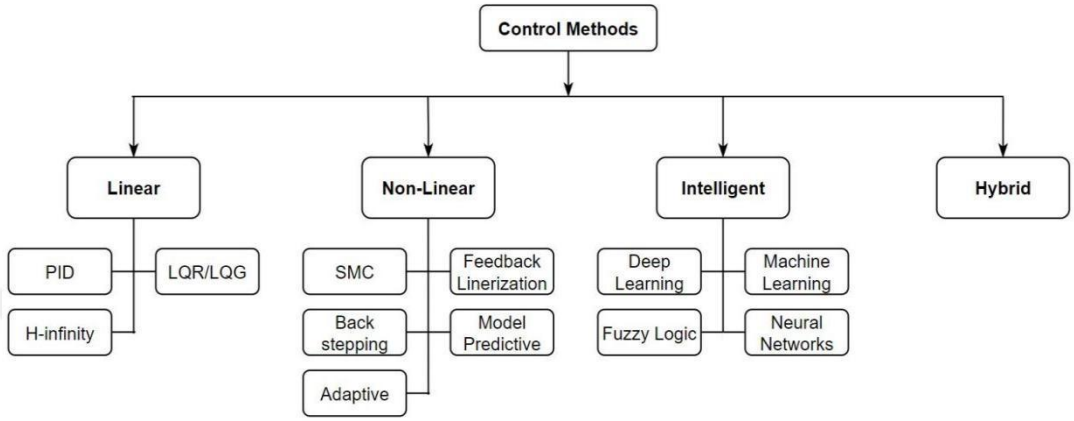


Figure 1.4 Control Systems Scheme

PID control is a frequently preferred controller in terms of its performance and simplicity. Besides being easy to implement, easy adjustment of coefficients and reliable performance make PID control preferable. However, since UAVs are underacted systems with nonlinear structure, it may not always be appropriate to apply PID. In the literature, there are various studies on PID control of quadcopters. Boubdallah compared PID and Linear Quadratic (LQ) controllers on a quadcopter that have modeled with Newton-Lagrance method (Bouabdallah and Siegwart, 2004). Nedim Osmic modeled a octocopter with Newton Euler method and also added parameters that are often neglected, such as the gyroscopic effect, to the model (Osmić and Petrović, 2016). The PD controller was examined and stable results were obtained. There are some methods to calculate PID coefficients. Božek, applied the PID controller calculated by the Ziegler Nichols (ZN) method to a hexacopter and got good results (Božek, et al., 2017). Also Alkamachi and Erçelebi used the genetic algorithm method to calculate PID coefficients (Alkamachi and Erçelebi, 2017).

The basis of the Optimal Control Theory is to ensure the continuity of the system with minumum cost. The problem in which the dynamics of the system is defined by

a set of linear differantials and the cost function is defined by a quadratic function called LQ. Linear Quadratic Regulator (LQR) control is optimized according to a cost function and minumum cost with user provided weighting factors. Linear Quadratic Gaussian (LQG) control is a combination of LQR and Kalman Filter. Rinaldi compared PID and LQR controllers for a coaxial octocopter (Rinaldi and Quagliotti, 2014). According to results both methods can be tuned for better results. While PID controller performed better result in rise time, LQR have better performance in settling time and overshoot. Argentim designed PID, LQR and PID tune LQR controllers and compared these controllers perfomance (Argentim, et al., 2013). Although the LQR controller shows robust results, it is stated that it should not be preferred when system parameters need to change rapidly. It has been concluded that the PID controller responds faster, but is less robust. With both methods more faster and reliable controller was designed.

Sliding Mode Control (SMC) is a nonlinear control method that changes the dynamics of a nonlinear system with an intermittent control signal that forces the system to slide. Its biggest advantage is that it cannot linearize the dynamics through linearization and has good tracking. Xu and Ozguner examined Sliding Mode control on a quadcopter (Xu and Ozguner, 2006). In the study, controllers were created according to full and underactuated designs and it was seen that stable results were obtained. As in this study, it is mentioned that there is a chattering effect in the Cömert's study (Cömert, 2016). This is a frequently encountered problem in Sliding Mode control. Cömert stated that situation was due to quickly changing dynamics and solved this problem by using saturation function. Dixit examined uncertainty and disturbance estimator based sliding mode controller for octocopter (Dixit and Talole, 2020a). Thanks to this method, a stable control is provided against unknown external effects. Khelfi designed robust sliding mode controller against uncertain parameters (Khelfi and Kacimi, 2012). Also Delica, on the other hand, tested the designed controller in a simulation environment and on a real system using a Pixhawk flight controller card (Delica, et al., 2019). According to the results, the SMC controller shows lower RMSE values than the PID controller in roll, pitch and yaw angles.

In backstepping control, the controller is divided into subsystems and each subsystem is checked step by step on a regular basis. The biggest advantage of this controller is that it converges faster with less computational resources and the disturbance rejection well. Madani divided the system into three as under actuated, fully actuated and propeller while doing dynamic modeling and applied a backstepping controller to control the whole system (Madani and Benallegue, 2006). He applied trajectory tracking to examine the controller and got good performance. Gueng Xue applied adaptive backstepping control and the disturbance effect caused by the firing of a weapon placed on the vehicle was investigated (Zhang and Taha, 2017). According to the results obtained, it has been seen that the backstepping controller is resistant to these disturbances.

Feedback linerization transforms a nonlinear system model into an equivalent linear system by changing variables. Disadvantage of this method is precision loss due to linearization. Rahmat linearized attitude and altitude dynamics of a quadcopter with this method and applied PID control (Rahmat, et al., 2020). However, adaptive feedback linearization (AFL) method has been applied to obtain a better performance against disturbance rejection and model parameter uncertainities. Zhang, on the other hand, used the backstepping method for a quadcopter, using the PID controller with feedforward and feedback linearization (Zhang, et al., 2014).

Model Predictive Control (MPC) relies on error tracking for the system to predict future states and provide better performance. In Dixit study, linear model predictive control (LMPC) based collosion avoidance controller was designed (Dixit and Talole, 2020b). It has been stated that LMPC performance depends on cost funtion minimization. Kostos Alexis controlled a quadcopter with MPC and showed stable behavior against atmospheric disturbances (Alexis and Tzes, 2011). Eskandarpour used MPC against trajectory tracking problem (Eskandarpour and Sharf, 2020). Trajectory tracking and disturbance rejection tested with two agressive maneveour in simulation environment and successful results were obtained.

Some controllers can be combined to make them more efficient. Riska Anelia designed a more efficient controller by combining Fuzzy+PID (Analia and Song, 2016). In the study, fuzzy logic was applied to reduce the amount of high overshoot seen as a result of the PID controller.

1.5 Thesis Structure

In the first chapter, in the first chapter, the thesis topic is introduced and a literature survey about the control systems of unmanned aerial vehicles is given. In the second chapter, mathematical model of quadcopters has been examined. In the third chapter, the parameters of the experimental vehicle were extracted and the model was validated. In the fourth chapter, PID control and Sliding Mode Control (SMC) are examined. In the fifth section, the results obtained are analyzed and performance comparisons are made. In the last chapter, there are conclusions and future studies.

CHAPTER 2
SYSTEM MODELING

2.1 Introduction

In this chapter, the mathematical model of the quadcopter is created. Reference of frames, kinematic and dynamic equations of the quadrotor are mentioned for the mathematical model of the vehicle. The Newton-Euler method was used while creating the mathematical model. To make the equations simpler, the following assumptions are made:

- Rigid body and mass distribution is symmetrical.
- Propellers are rigid.
- Center of mass and body fixed frame origin coincidences.
- g, J, b, d are constants.

2.2 Reference of Frames

As shown in Figure 2.1, the first axis is the ground fixed frame, and the second axis is the body fixed frame, fixed to the center of gravity of the vehicle. The axes of the earth fixed frame face north, west, and opposite the center of the Earth. The body frame is the frame fixed to the center of mass of the aircraft. Accelerometers and gyroscopes on the vehicle take measurements on this axis. As shown in the Figure 2.2, the x-axis of the body frame is the front of the UAV, the y-axis is the left of the UAV, and the z-axis is the top of the UAV.

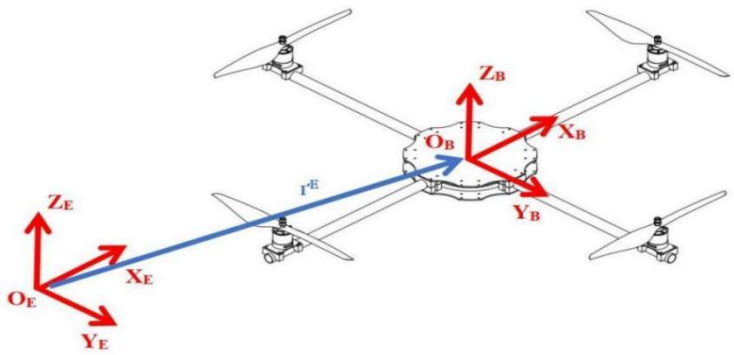


Figure 2.1 Reference of frames

Position of center of mass is defined with ζ in earth frame. Orientation of the quadcopter is defined in earth frame as euler angles η. Linear and angular velocities are defined as V_B and ω in body frame. These frames and representations are shown in Table 2.1.

Table 2.1 Reference Frame Vectors

Earth Frame		**Body Frame**	
$\zeta = [X\ Y\ Z]^T$	Position vector	$V_B = [u\ v\ w]^T$	Linear velocity vector
$\eta = [\varphi\ \theta\ \Psi]^T$	Orientation vector	$\omega = [p\ q\ r]^T$	Rotational velocity vector

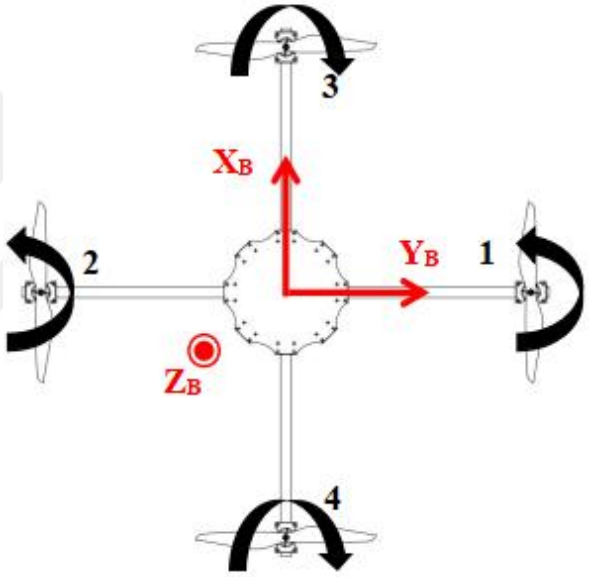


Figure 2.2 Quadcopter Schematic

2.3 Kinematics

In this section, the linear and angular velocities of the UAV will be examined. Kinematics is a subject that studies the motion of objects without taking force and moment into account. In order to obtain the kinematic equations, it is necessary to know the transformations in different coordinate systems. First part includes the Euler angles and rotation matrices and the second part is about linear and angular kinematics and the velocity equations will be provided.

Table 2.2 Kinematics Notations

Symbol	Description	Symbol	Decription
φ	Roll	ζ(X,Y,Z)	Position vector in E.
θ	Pitch	V_B(u,v,w)	Translational velocity in B.
Ψ	Yaw	$V_E(\dot{x},\dot{y},\dot{z})$	Translational velocity in E.($\dot{\zeta}$)
$R_{E\rightarrow B}$	Rotation from E to B	η(φ,θ,Ψ)	Orientation relative to E.
$R_{B\rightarrow E}$	Rotation from B to E	w(p,q,r)	Rotational velocity in B.
T	Transformation matrix		

2.3.1 Euler Angles

Euler angles is the name given to three angles used to describe rotation in 3-dimensional space. These are called yaw, pitch and roll respectively. Roll angle on X_b axis, pitch angle on Y_b axis and yaw angle on Z_b axis are positive angles created by right hand rotation movement on body frame. These three angles are used to express the rotation of an object in 3-dimensional space. The transformation matrix represents the relations of vectors expressed in different frames. It is possible to switch from the earth axis to the body axis or from the body axis to the earth axis with transformation matrices. Another name for the transformation matrix is Direction Cosine Matrix. Euler angles represent a sequence of three fundamental rotations, i.e. rotations around the axes of a coordinate system, since any orientation can be obtained by creating three fundamental rotations.

Euler angles and quaternions are used to obtain the vehicle's attitude and orientation information. Euler angles method will be used in this study. Euler angles work smoothly, except for pitch angles of +/- 90 degrees, although they are easier to apply than quaternions. If the pitch angle is +/- 90 degrees, *the Gimbal lock* event occurs and the system becomes unstable.

Figure 2.3 shows Euler angles representation. These rotations start from a known standard orientation. These rotation matrices are as follows.

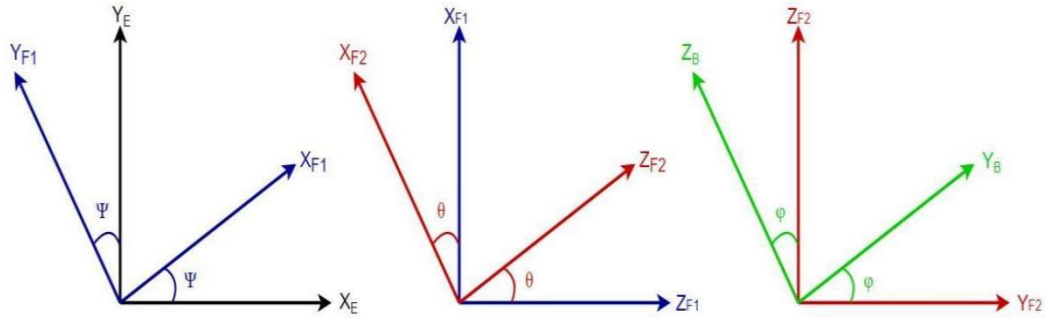


Figure 2.3 Euler Angles

$$R(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{bmatrix} \tag{2.1}$$

$$R(\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \tag{2.2}$$

$$R(\Psi) = \begin{bmatrix} c\Psi & -s\Psi & 0 \\ s\Psi & c\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

By combining these rotations, the rotation matrix between body frame to earth frame is obtained.

$$R_{B\to E} = R(\Psi).R(\theta).R(\varphi) = \begin{bmatrix} c\Psi & s\Psi & 0 \\ -s\Psi & c\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{bmatrix} \tag{2.4}$$

$$R_{B\to E} = \begin{bmatrix} c\theta c\Psi & s\varphi s\theta c\Psi - c\varphi s\Psi & c\varphi s\theta c\Psi + s\varphi s\Psi \\ c\theta s\Psi & s\varphi s\theta s\Psi + c\varphi c\Psi & c\varphi s\theta s\Psi - s\varphi c\Psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \tag{2.5}$$

By taking the transpose of this matrix, the transition from earth frame to body frame is obtained.

$$R_{E\to B} = \begin{bmatrix} c\theta c\Psi & s\Psi c\theta & -s\theta \\ s\varphi s\theta c\Psi - c\varphi s\Psi & s\varphi s\theta s\Psi + c\varphi c\Psi & s\varphi c\theta \\ c\varphi s\theta c\Psi + s\varphi s\Psi & c\varphi s\theta s\Psi - s\varphi c\Psi & c\varphi c\theta \end{bmatrix} \tag{2.6}$$

2.3.2 Translational Kinematics

The vehicle's velocity vector is on the body frame, but position vector is on the earth frame. Therefore, the rotation matrix is used to achieve the transformation between coordinate systems. Using Equation 2.5, earth frame transition is achieved.

$$V_E = R_{B\to E}.V_B \tag{2.7}$$

$$V_E = \dot{\zeta} = [\dot{x} \quad \dot{y} \quad \dot{z}]^T \tag{2.8}$$

Linear velocity equations are provided using matrix form of the equation above.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = R_{B\to E}\begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}c\theta c\Psi & s\varphi s\theta c\Psi - c\varphi s\Psi & c\varphi s\theta c\Psi + s\varphi s\Psi\\ c\theta s\Psi & s\varphi s\theta s\Psi + c\varphi c\Psi & c\varphi s\theta s\Psi - s\varphi c\Psi\\ -s\theta & s\varphi c\theta & c\varphi c\theta\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{2.10}$$

2.3.3 Rotational Kinematics

In order to ensure continuity between the Body frame and the Earth frame, the angular ratios of the vehicle must be associated with the time-dependent derivative of the Euler angles. For this, the transfer matrix is used. Angular velocity equation is as like in Equation 2.11,

$$\begin{bmatrix}p\\ q\\ r\end{bmatrix} = R(\varphi).R(\theta).R(\Psi)\begin{bmatrix}0\\ 0\\ \dot{\Psi}\end{bmatrix} + R(\varphi).R(\theta)\begin{bmatrix}0\\ \dot{\theta}\\ 0\end{bmatrix} + R(\varphi)\begin{bmatrix}\dot{\varphi}\\ 0\\ 0\end{bmatrix} \tag{2.11}$$

$$\begin{bmatrix}p\\ q\\ r\end{bmatrix} = \begin{bmatrix}1 & 0 & -s\theta\\ 0 & c\varphi & s\varphi c\theta\\ 0 & -s\varphi & c\varphi c\theta\end{bmatrix}\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\Psi}\end{bmatrix} \tag{2.12}$$

Transformation matrix (T) is used to obtain the time derivative of the Euler angles.

$$T = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & s\varphi c\theta \\ 0 & -s\varphi & c\varphi c\theta \end{bmatrix}^{-1} = \begin{bmatrix} 1 & s\varphi t\theta & c\varphi t\theta \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi/c\varphi & c\varphi/c\theta \end{bmatrix} \quad (2.13)$$

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & s\varphi t\theta & c\varphi t\theta \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi/c\varphi & c\varphi/c\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.14)$$

When equations 2.10 and 2.14 are removed from the matrix form, the kinematic equations are obtained as follows.

$$\dot{x} = (c\theta c\Psi)u + (s\varphi s\theta c\Psi - c\varphi s\Psi)v + (c\varphi s\theta c\Psi + s\varphi s\Psi)w \quad (2.15)$$

$$\dot{y} = (c\theta s\Psi)u + (s\varphi s\theta s\Psi + c\varphi c\Psi)v + (c\varphi s\theta s\Psi - s\varphi c\Psi)w \quad (2.16)$$

$$\dot{z} = (-s\theta)u + (s\varphi c\theta)v + (c\varphi c\theta)w \quad (2.17)$$

$$\dot{\varphi} = p + (s\varphi t\theta)q + (c\varphi t\theta)r \quad (2.18)$$

$$\dot{\theta} = (c\varphi)q + (-s\varphi)r \quad (2.19)$$

$$\dot{\Psi} = (s\varphi/c\varphi)q + (c\varphi/c\theta)r \quad (2.20)$$

2.4 Control Inputs

The lift force of the vehicle and its motion are provided by the forces and torques created by the motors. Therefore, the control input of the system is the rotation speed of the propellers mounted on output shaft of brushless DC motors. According to the thrust provided by the motors and so the propellers, the vehicle takes off and its orientation is controlled. In multi rotor UAVs, there are four inputs. These are thrust (U_1), roll (U_2), pitch (U_3), and yaw (U_4) moments. In this study, quadcopter modeling has been done, but control inputs for octocopter configuration are given in Appendix 5.

2.4.1 Thrust

Altitude of the UAV is provided using thrust generated. Ideally, propellers rotate with same speed and lifts off the UAV through Z axis which the positive direction is opposite to the direction of gravitational acceleration. Total thrust can ve calculated using the equation follows;

$$U_1 = \sum_{i=1}^{4} F_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (2.21)$$

2.4.2 Roll Moment

Change in roll angle of the UAV is the change in angle wrt x axis which is the result of change of velocities of propellers. As shown in Figure 2.4, if the blue region is higher than green region then the roll angle is positive whereas if it is lower then the roll angle is negative. The roll angle equation is given as follows,

$$U_2 = -\, bl[(\sin 90 \,.\, \Omega_1^2 + \sin 270 \,.\, \Omega_2^2 + \sin 360 \,.\, \Omega_3^2 + \sin 180 \,.\, \Omega_4^2)] \quad (2.22)$$

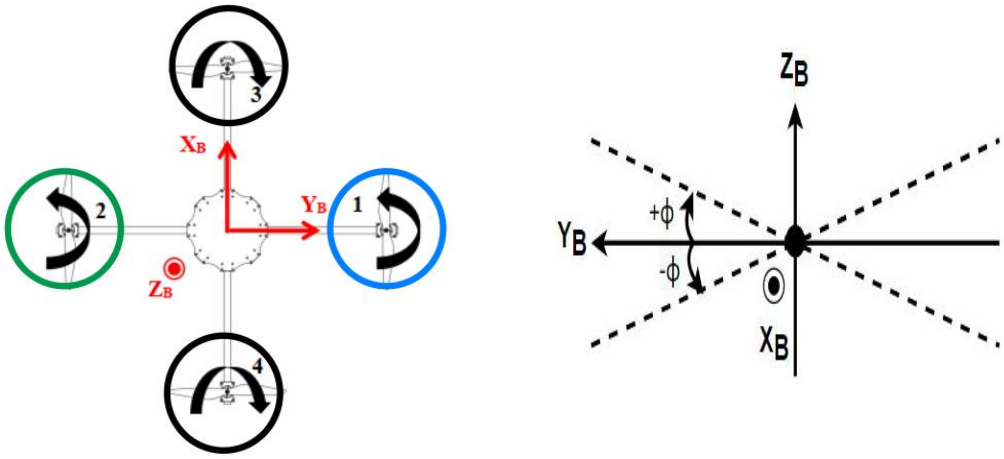


Figure 2.4 Roll Motion

2.4.3 Pitch Moment

The change in the pitch angle occurs due to the differences in the rotation speed of the propellers, like in the roll angle. The roll angle is relative to the x-axis whereas the pitch angle is relative to the vehicle's y-axis. As shown in Figure 2.5, if the part

of the vehicle marked in blue is higher than the green part, the pitch angle is negative. Pitch control input equation is as follows.

$$U_3 = bl[(\cos 90 \,.\, \Omega_1^2 + \cos 270 \,.\, \Omega_2^2 + \cos 360 \,.\, \Omega_3^2 + \cos 180 \,.\, \Omega_4^2)] \qquad (2.23)$$

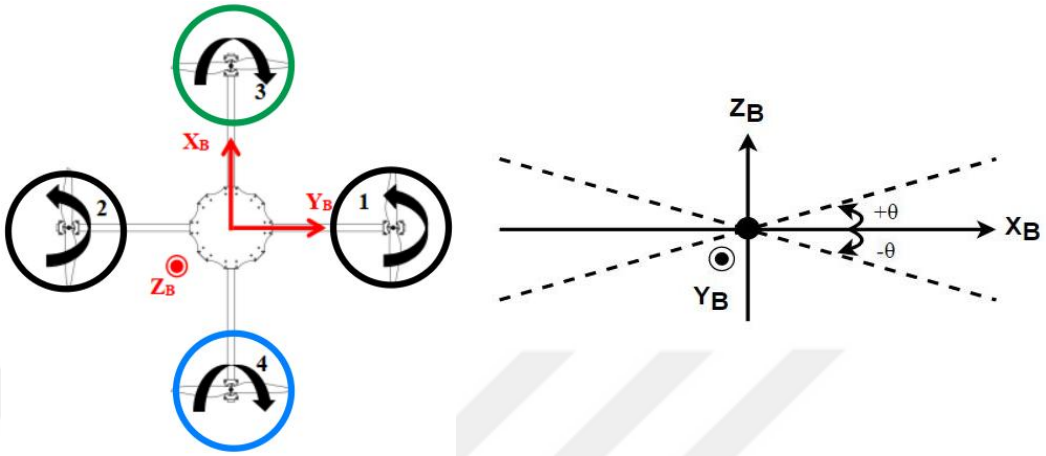


Figure 2.5 Pitch Motion

2.4.4 Yaw Moment

The yaw angle represents the rotation of the tool with respect to the z-axis. As a result of the rotation of the propellers, a moment occurs on the vehicle opposite to the rotation direction of the propellers. Propellers are grouped according to the direction of rotation to balance the moment as can be seen in Figure 2.6. If the vehicle is turning clockwise, the yaw angle is positive, if it is moving counterclockwise, it is negative. Yaw control input equation is as follows.

$$U_4 = d(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \qquad (2.24)$$

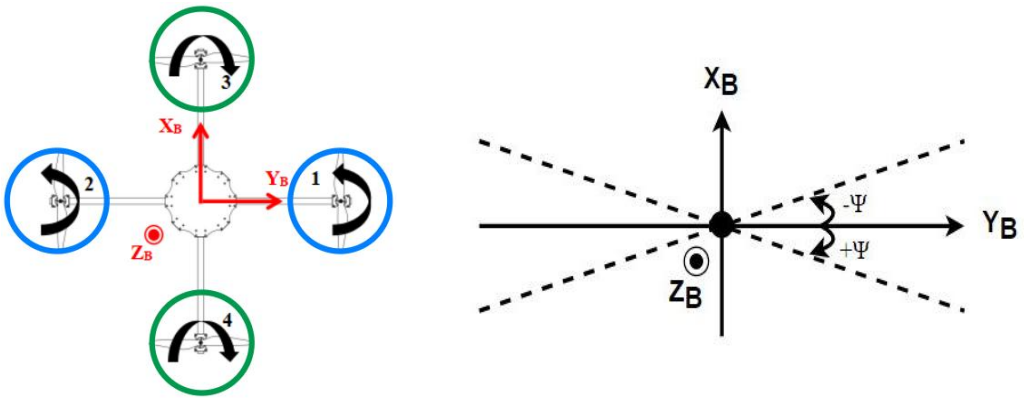


Figure 2.6 Yaw Motion

If the equations can be represented in matrix forms;

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -\mathrm{bl} & bl & 0 & 0 \\ 0 & 0 & bl & -\mathrm{bl} \\ d & d & -d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{2.25}$$

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b} & -\frac{1}{2bl} & 0 & \frac{1}{4d} \\ \frac{1}{4b} & \frac{1}{2bl} & 0 & \frac{1}{4d} \\ \frac{1}{4b} & 0 & \frac{1}{2bl} & -\frac{1}{4d} \\ \frac{1}{4b} & 0 & -\frac{1}{2bl} & -\frac{1}{4d} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{2.26}$$

2.5 Dynamics of Quadrotors

Dynamics is a branch o f mechanics that studies the effects of forces and torques on the movement of an object or system. There are several methods to obtain the equations of a system. In this study Newton-Euler method was examined.

Table 2.3 Effects on Dynamics

Cause	**Effect**
Rotating propellers	Aerodynamic effects
Speed change of propellers	Inertial counter torque
Position of the CoM	Effect of gravity
Change in the direction	Gyroscopic effects

Equations of motion are always written according to the body frame for the following reasons.

- Inertia matrix is invariant.
- To take advantage of body symmetry.
- Measurements are taken more easily.
- Control inputs are taken from the body frame.

The dynamics of the vehicle is obtained according to the following equation.

$$\begin{bmatrix} mI_{3x3} & 0_{3x3} \\ 0_{3x3} & J \end{bmatrix} \begin{bmatrix} \dot{V}_B \\ \dot{w}_B \end{bmatrix} + \begin{bmatrix} w_B \times mV_B \\ w_B \times Iw_B \end{bmatrix} = \begin{bmatrix} F \\ M \end{bmatrix} \tag{2.27}$$

V̇ linear acceleration in body frame, ẇ angular acceleration in body frame, F body frame forces, M body frame torques.

2.5.1 Translational Dynamics

The translational dynamics are derived from Newton's second law. According to this law, the total force acting on an object is equal to the product of the object's mass and its acceleration.

$$\mathrm{F_E} = \frac{d(mV_E)}{dt} \tag{2.28}$$

The following formula is used to define this equation on the body axis.

$$m\dot{V_B} + \omega \times (mV_B) = \sum F \tag{2.29}$$

By taking the derivative of Equation 2.7, the following equation is obtained. To find the $\dot{R_{B\to E}}$ expression, the orthogonal property of the rotation matrix is used.

$$\dot{V_E} = \ddot{\zeta} = \dot{R_{B\to E}}.V_B + R_{B\to E}.\dot{V_B} \tag{2.30}$$

$$\dot{V_E} = \ddot{\zeta} = R_{B\to E}.(\omega \times V_B) + R_{B\to E}.\dot{V_B} \tag{2.31}$$

Then the expression $\dot{V_B}$ is left alone.

$$\dot{V_B} = R_{B\to E}^{-1}\ddot{\zeta} - (\omega \times V_B) \tag{2.32}$$

Equation 2.29 is rearranged.

$$m\dot{V}_B + \omega \times (mV_B) = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + mR_{E\to B} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{2.33}$$

$$\mathrm{m}R_{B\to E}^{-1}\ddot{\zeta} - \omega \times V_B + \omega \times mV_B = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + mR_{E\to B} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{2.34}$$

The translation dynamics are obtained by multiplying both sides by $R_{B\to E}$ and dividing by m.

$$R_{B\to E} \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} + R_{B\to E}.R_{E\to B} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = R_{B\to E}.{R_{B\to E}}^{-1}\ddot{\zeta} \tag{2.35}$$

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = R_{B\to E} \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{2.36}$$

2.5.2 Rotational Dynamics

Rotation dynamics are obtained similarly to translational dynamics. Using equation 2.37, rotational equations of motion are obtained.

$$\mathrm{J}\dot{\omega} + \omega \times \mathrm{J}\omega = \sum \mathrm{M} \tag{2.37}$$

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ J_r\Omega \end{bmatrix} \tag{2.38}$$

As can be seen, the equations in the rotational motion part are more complex than the equations in the translational motion part. Before being included in the control system, a number of simplifications are required. As seen in equation 2.38, transformation between these representations $[\dot{p}\ \dot{q}\ \dot{r}]$ and $[\ddot{\varphi}\ \ddot{\theta}\ \ddot{\Psi}]$ becomes complex because of derivation and trigonometric functions.

Therefore, the dynamic rotation model can be obtained with simplification of body angular velocities, [p q r] are equal to change in euler angles [φ θ Ψ].

Transformation matrix can be assumed as identity matrix using small angle approximation.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \varphi \\ \theta \\ \Psi \end{bmatrix}, \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\Psi} \end{bmatrix} \tag{2.39}$$

As a result, the following motion equations are obtained.

$$\ddot{X} = (c\varphi s\theta c\Psi + s\varphi s\Psi)\frac{\mathrm{U}_1}{\mathrm{m}} \tag{2.40}$$

$$\ddot{Y} = (c\varphi s\theta s\Psi - s\varphi c\Psi)\frac{\mathrm{U}_1}{\mathrm{m}} \tag{2.41}$$

$$\ddot{Z} = (c\varphi c\theta)\frac{\mathrm{U}_1}{\mathrm{m}} - \mathrm{g} \tag{2.42}$$

$$\ddot{\varphi} = \left(\frac{J_{yy}-J_{zz}}{J_{xx}}\right)\dot{\theta}\dot{\Psi} - \frac{J_r\Omega_r\dot{\theta}}{J_{xx}} + \frac{\mathrm{U}_2}{J_{xx}} \tag{2.43}$$

$$\ddot{\theta} = \left(\frac{J_{zz}-J_{xx}}{J_{yy}}\right)\dot{\varphi}\dot{\Psi} + \frac{J_r\Omega_r\dot{\varphi}}{J_{yy}} + \frac{\mathrm{U}_3}{J_{yy}} \tag{2.44}$$

$$\ddot{\Psi} = \left(\frac{J_{xx}-J_{yy}}{J_{zz}}\right)\dot{\varphi}\dot{\theta} + \frac{\mathrm{U}_4}{J_{zz}} \tag{2.45}$$

2.6 Aerodynamics Model

Forces and moments occur on the UAV due to the wind effect. Air resistance is the basis of these forces and moments. Due to air friction, a force occurs that resists the movement of the vehicle. As the speed of the vehicle increases, the effect of this force increases. The formula for drag force is given below.

$$F_{drag} = -\frac{1}{2}\rho C_D A v^2 \tag{2.46}$$

In this equation, ρ is the air density, A is the surface area at which the velocity is perpendicular, and C_D is the drag coefficient. Drag moment also occurs due to air friction. Its formula is similar to drag force. C_D in the equation is the drag coefficient. These forces and moments are added to the 6DOF model.

$$M_{drag} = -\frac{1}{2}\rho C_D A v^2 \tag{2.47}$$

2.7 State Space Model

Mathematical model of systems provide information about how the system will respond to inputs. While transfer functions represent a system in terms of n-order differential equations, state-space modeling represents the state variables associated with the system's inputs and outputs using first-order differential equations. They provide a simpler representation of systems, especially multi-input multi-output (MIMO) systems.

The state space equation of a nth order system with r inputs and m outputs can be written as follows. x is a state vector of dimensions *n x 1* and u is a control input vector of dimensions *r x 1*.

$$\dot{x} = f(x, u) \tag{2.48}$$

The vehicle's state-space variables can be seen in below. X is a state vector and U is a vector of control inputs.

$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}] \tag{2.49}$$

$$X = [\varphi \quad \dot{\varphi} \quad \theta \quad \dot{\theta} \quad \Psi \quad \dot{\Psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}] \tag{2.50}$$

$$U = [U_1 \quad U_2 \quad U_3 \quad U_4] \tag{2.51}$$

In vector X, Euler angles and Euler rates are used for rotational dynamics and translations and linear velocities are used for translational dynamics. The U vector reprents the control inputs.

State variables and their derivatives are shown in Table 2.4. The equations of motion obtained above using the state space method are written in terms of first-order differential equations.

Table 2.4 State Variables

$\boldsymbol{x_1}$	φ	$\dot{\boldsymbol{x_1}}$	$\dot{\varphi} = x_2$
$\boldsymbol{x_2}$	$\dot{x1} = \dot{\varphi}$	$\dot{\boldsymbol{x_2}}$	$\ddot{\varphi} = x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2$
$\boldsymbol{x_3}$	θ	$\dot{\boldsymbol{x_3}}$	$\dot{\theta} = x_4$
$\boldsymbol{x_4}$	$\dot{x_3} = \dot{\theta}$	$\dot{\boldsymbol{x_4}}$	$x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3$
$\boldsymbol{x_5}$	Ψ	$\dot{\boldsymbol{x_5}}$	$\dot{\Psi} = x_6$
$\boldsymbol{x_6}$	$\dot{x_5} = \dot{\Psi}$	$\dot{\boldsymbol{x_6}}$	$\ddot{\Psi} = x_2 x_4 a_5 + b_3 U_4$
$\boldsymbol{x_7}$	x	$\dot{\boldsymbol{x_7}}$	$\dot{x} = x_8$
$\boldsymbol{x_8}$	$\dot{x_7} = \dot{x}$	$\dot{\boldsymbol{x_8}}$	$\ddot{x} = (cosx_1 sinx_3 cosx_5)\frac{U_1}{m}$
$\boldsymbol{x_9}$	y	$\dot{\boldsymbol{x_9}}$	$\dot{y} = x_{10}$
$\boldsymbol{x_{10}}$	$\dot{x_9} = \dot{y}$	$\dot{\boldsymbol{x_{10}}}$	$(sinx_1 cosx_5 - cosx_1 sinx_3 sinx_5)\frac{U_1}{m}$
$\boldsymbol{x_{11}}$	z	$\dot{\boldsymbol{x_{11}}}$	$\dot{z} = x_{12}$
$\boldsymbol{x_{12}}$	$\dot{x_{11}} = \dot{z}$	$\dot{\boldsymbol{x_{12}}}$	$\ddot{z} = (cosx_1 cosx_3)\frac{U_1}{m} - g$
$a_1 = \frac{J_{yy}-J_{zz}}{J_{xx}}$, $a_2 = \frac{J_r}{J_{xx}}$, $a_3 = \frac{J_{zz}-J_{xx}}{J_{yy}}$, $a_4 = \frac{J_r}{J_{yy}}$, $a_5 = \frac{J_{xx}-J_{yy}}{J_{zz}}$, $b_1 = \frac{1}{J_{xx}}$, $b_2 = \frac{1}{J_{yy}}$, $b_3 = \frac{1}{J_{zz}}$			

$$\dot{x} = f(x,u) = \begin{bmatrix} x2 \\ x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 \\ x_4 \\ x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3 \\ x_6 \\ x_2 x_4 a_5 + b_3 U_4 \\ x_8 \\ (cosx_1 sinx_3 cosx_5)\frac{U_1}{m} \\ x_{10} \\ (sinx_1 cosx_5 - cosx_1 sinx_3 sinx_5)\frac{U_1}{m} \\ x_{12} \\ (cosx_1 cosx_3)\frac{U_1}{m} - g \end{bmatrix} \quad (2.49)$$

If the results in the Table 2.4 are written as in equation 2.48, the state-space representation of the system is obtained as Equation 2.49. Mass, arm length and moments of inertia in the table are as seen in Table 3.5.

2.8 Pesticide Spraying Model

Considering that the modeled vehicle will be used for spraying purposes, mass reduction has been added to the model. For this purpose, the integrator block was used as seen in Figure 2.7. It is assumed that 1.8 g per second decreases while spraying. In this case, after 150 seconds, the vehicle drops from 6 kg to its empty weight of 3.1 kg. This decrease in mass also adds uncertainty to the system. In addition, inertia changes due to mass reduction will be added as uncertainty later.

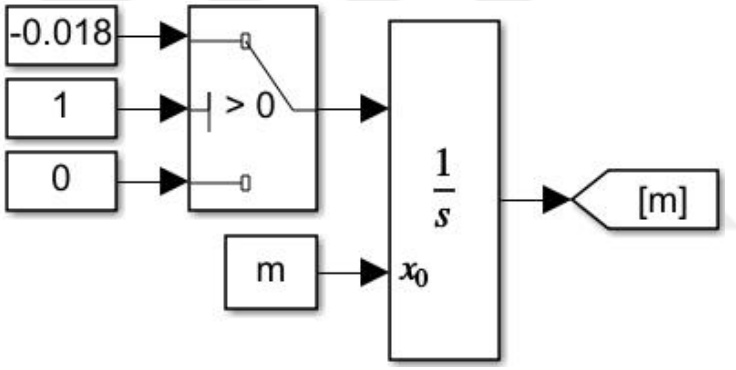


Figure 2.7 Mass Reduction Model

CHAPTER 3
SIMULINK MODEL AND SIMULATION

3.1 Model Spesifications

In this section, experiments and measurements were made on the vehicle's mechanical properties, motor characteristics and sensor noise to create a realistic model of the test vehicle. As a test vehicle, the quadcopter in Automatic Control and Robotics Laboratory (ACRL) in Figure 3.1 was used. Components used in the UAV can be seen Table 3.1.



Figure 3.1 ACRL Quadcopter Prototype

Table 3.1 UAV Components

No	Component	Model	Description
1	Brushless DC Motor	Emax GT-2826	- 1090 KV - 175 gr - Diameter: 35mm - Length: 49.5mm
2	ESC	Emax BLHeli 50A	- 50 A - 41 gr - 2S-6S
3	Battery	Gens Ace Li-Po	- 5800 mAh

			- 4S1P - 14.8V 45C
4	Propeller	10x45	- Plastic
5	Flight Controller	APM 2.6	- Atmel's ATMEGA2560 - 3-Axis Gyro - 3-Axis Accelerometer - High Resolution Altimeter - 5Hz GPS Module - Dimension: 70.5x45x13.5mm - 31 gr
6	Receiver	Flysky R9B	- 8 channels - Receiving Sensitivity: -105dBm - Frequency Range: 2.4-2.48GHz - Frequency Width: 500KHz - Wave Band: 160 - 18 gr
7	GPS	UBlox M8N	- Receiver type: 72-channel u-blox M8

3.1.1 Mechanical Structure Spesifications

While modeling the vehicle, structural parameters such as arm length, moment of inertias and weight are needed. Arm length and weight can be measured directly but to measure moment of inertia there must be CAD file or measurement platform. In this study bifilar pendulum platform was used to measure.

3.1.2 Bifilar Pendulum and Moment of Inertia Measurement

The MOI is a moment of resistance caused by mass elements distribution. In Figure 3.2, the first distribution has more higher Moment of Inertia (MOI) then the second one. That means the first one will be harder to rotate.

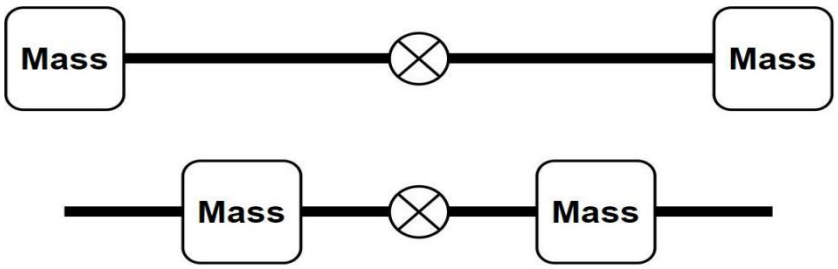


Figure 3.2 Mass Distribution Difference

In bifilar pendulum, gravity provides the torque once the test component is set into a torsional oscillation. The period of the oscillation will be measured and used to calculate the MOI.

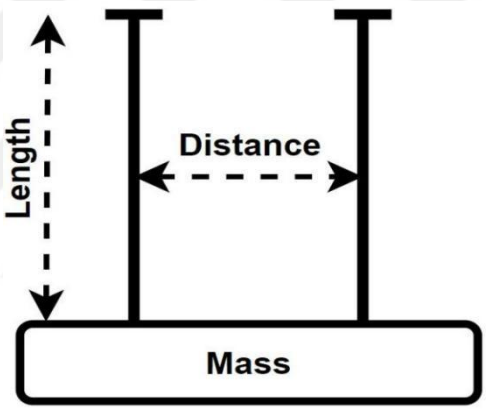


Figure 3.3 Bifilar Pendulum Model

The MOI can be calculated by using equation 3.1.

$$\text{MOI} = \frac{distance\ [m]^2 period\ [sec]^2 weight[kg]}{16\pi^2 length\ [m]} \tag{3.1}$$

When starting the measurement, the test component is held at an angle. Then it is released and timed for 3 oscillations. These steps are repeated and the average period is found by taking the average of the results found.

Tests were carried out on two different samples. The first one is made on aluminum sigma profile, shown in Figure 3.4. The other is made on a hollow tube.

The properties of the test samples are given in Table 3.2. Sigma profile's MOI is calculated by using SolidWorks. Hollow tube's MOI is calculated by using equation 3.2.

$$I_x, I_y = \frac{1}{12} m(3(r_{outer}^2 + r_{inner}^2) + l^2) \quad (3.2)$$

Table 3.2 Test Sample Properties

Property	Sigma Profile	Hollow Tube
Length [m]	0.53	0.534
Distance [m]	0.602	0.606
Mass [kg]	0.544	2.244
MOI [kgm^2]	0.02	0.122

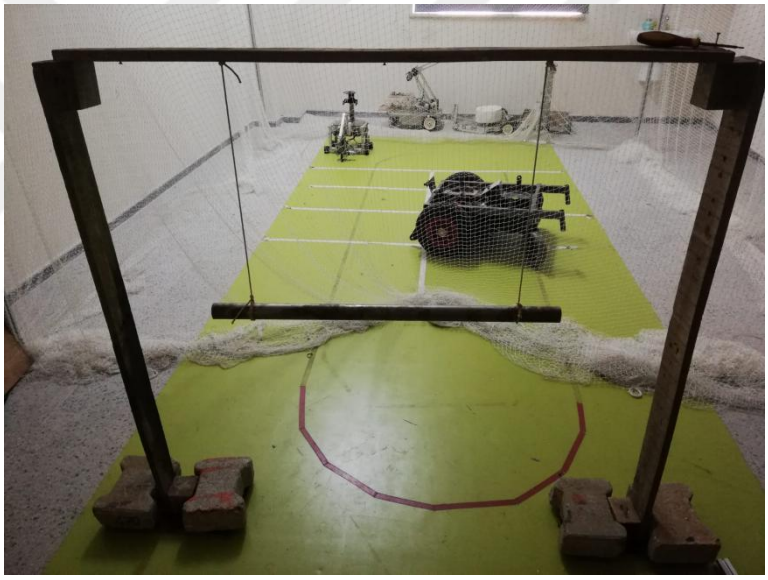


Figure 3.4 Bifilar Pendulum and Sample

Table 3.3 Sigma Profile Results

Trial	Time for 3 Oscillations [sec]	Period [sec]
1	2.69	0.896
2	2.55	0.85
3	2.85	0.95
4	2.85	0.95
5	2.71	0.9
Average Period = 0.91 s		

When the values were put in the MOI equation, the sigma profile was measured as 0.0191 kgm^2 and the hollow tube as 0.126 kgm^2. When we compare the results in Table 3.4, it is seen that the results are similar.

For the UAV , measurements were taken in X, Y and Z axis and the vehicle was placed as shown in Figure 3.5.

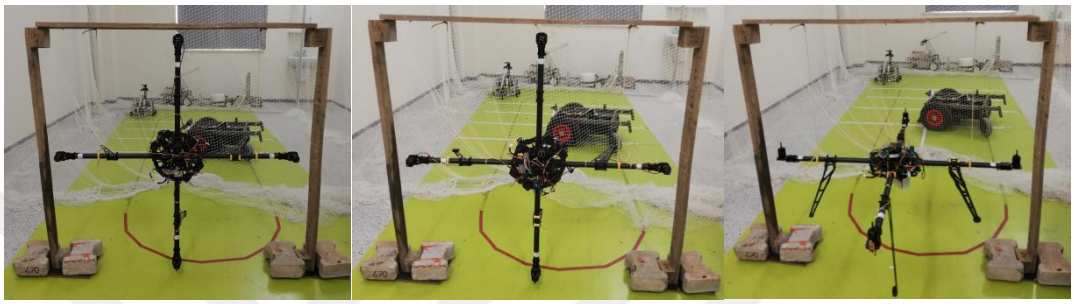


Figure 3.5 Measurements in X, Y, Z axises

Table 3.4 MOI Result Table

X-axis			**Y-axis**		**Z-axis**	
Mass	**Length**	**Distance**	**Length**	**Distance**	**Length**	**Distance**
3,1kg	0,59m	0,725m	0,6m	0,725m	0,65m	0,715m
Sample	**3 Period**	**1 Period**	**3 Period**	**1 Period**	**3 Period**	**1 Period**
1	2,94	0,980	3,23	1,077	4,58	1,527
2	3,1	1,033	3,07	1,023	4,66	1,553
3	3,07	1,023	3,25	1,083	4,51	1,503
4	3,16	1,053	3,13	1,043	4,78	1,593
5	3,16	1,053	3,2	1,067	4,2	1,400
6	3,14	1,047	3,25	1,083	4,37	1,457
7	3,25	1,083	3,11	1,037	4,32	1,440
8	2,94	0,980	3,19	1,063	4,42	1,473
9	3,03	1,010	3,1	1,033	4,37	1,457
10	3,14	1,047	3,12	1,040	4,41	1,470
Average Period		1,031		1,055		1,487

When the values were substituted in the MOI equation, they were found to be 0.182, 0.188, and 0.335 kgm^2 , respectively.

3.1.3 Motor Properties

A test platform was built to obtain data such as thrust, RPM, Amps and Voltage. An ampermeter is connected to battery to measure amper value, a voltmeter measure battery voltage, a loadcell measures thrust value in gram, CNY70 infrared sensor measures RPM value. Microcontroller generates PWM value and with a potentiometer this PWM value can be controlled. This platform can be seen in Figure 3.6.

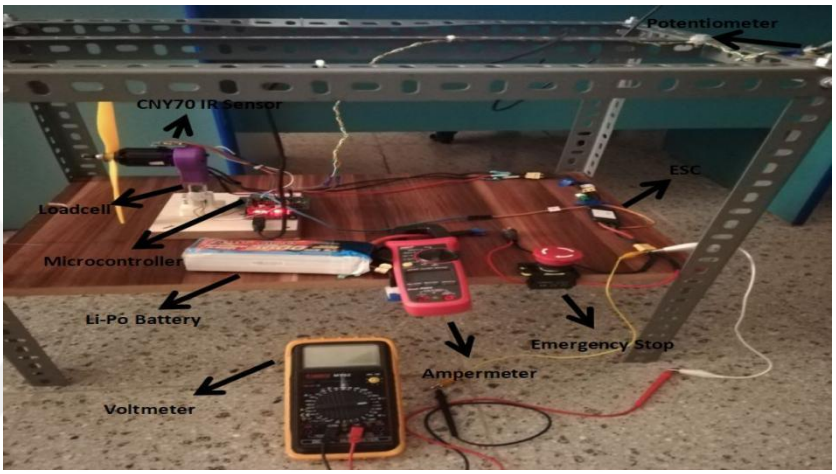


Figure 3.6 Motor Thrust Platfrom

An interface design in Figure 3.7 was made using Matlab Appdesigner to get the data faster and make the testing process more comfortable.

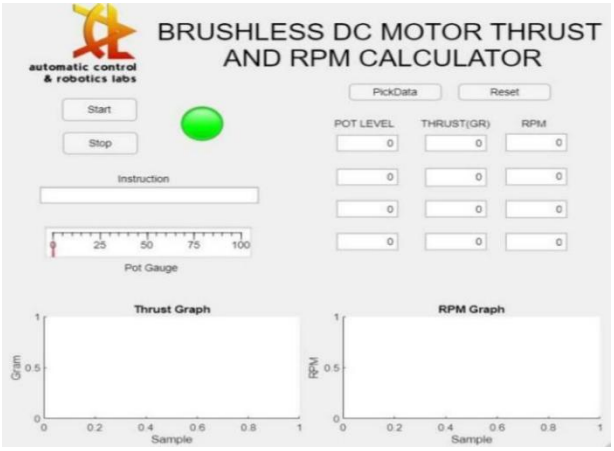


Figure 3.7 Appdesigner User Interface

Firtly an Arduino code has been written to obtain data from sensors and to generate PWM signals. Thrust, RPM and potentiometer values were send to computer via serial port. The User Interface(UI) checks the port is open or not and find the active port. After that the user can press “Start” button and start receiving measurements. Potentiometer value (0-100) represents PWM value in percent. The user can adjust the PWM value and by clicking the “Pick Data” button, the measurements can be printed on the screen. At the end of the fourth click, the measurements taken throughout the whole process are displayed in graphs. In addition, after the measurements, an excel file is created in which the data is written to the folder where the program is located. By clicking the “Reset” button, the measurements taken can be reset or the program can be terminated with the stop button. As shown in Figure 3.8, an example result screen can be seen.

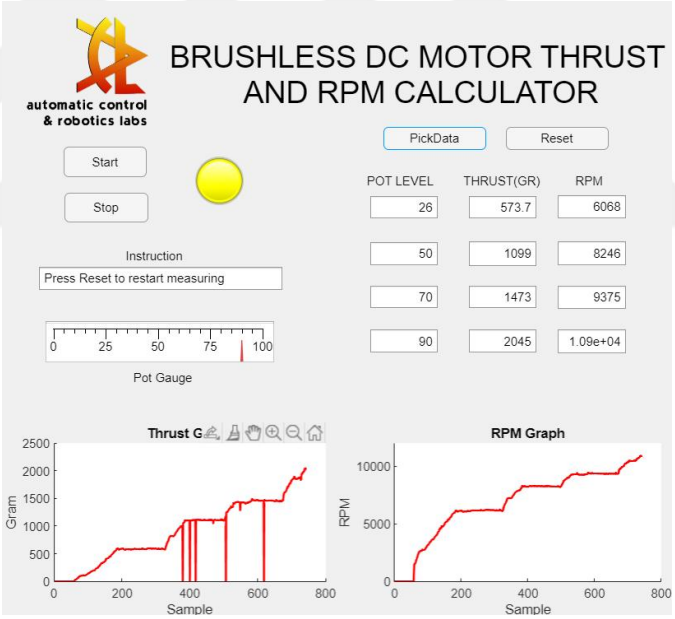


Figure 3.8 UI Measurement Result

Figure 3.9 shows the components required to generate thrust and the connection between them. The flight control board determines a setpoint and the ESCs transmit the required signal to the motors. The motors rotate according to the PWM signal created to reach the determined setpoint and they are driven by the propellers.

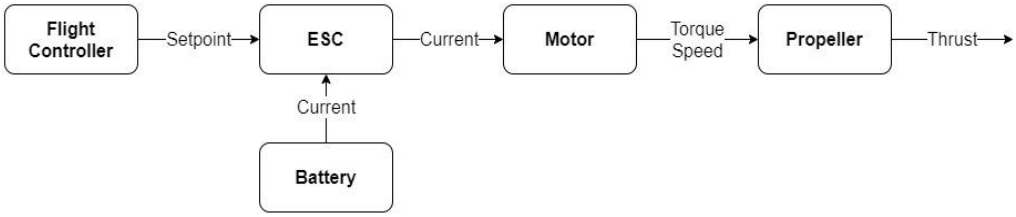


Figure 3.9 Thrust Generation Diagram

The thrust mechanism CAD model can be seen in Figure 3.10. Thrust value is measured by using loadcell and RPM value is measured by using IR sensor. The torque value is found by multiplying the measured thrust value and the distance between the connection point of the loadcell. To calculate trust coefficient K_T and drag coefficient K_Q, Thrust and Torque vs RPM graphs are used as shown in Figure 3.11. The graphs were plotted, then using curve fitting the trendline equation were found.

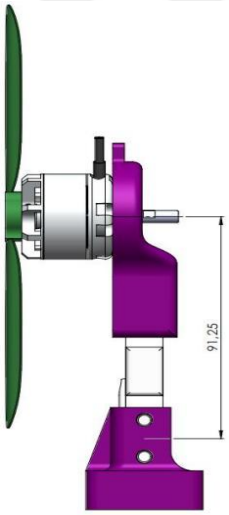
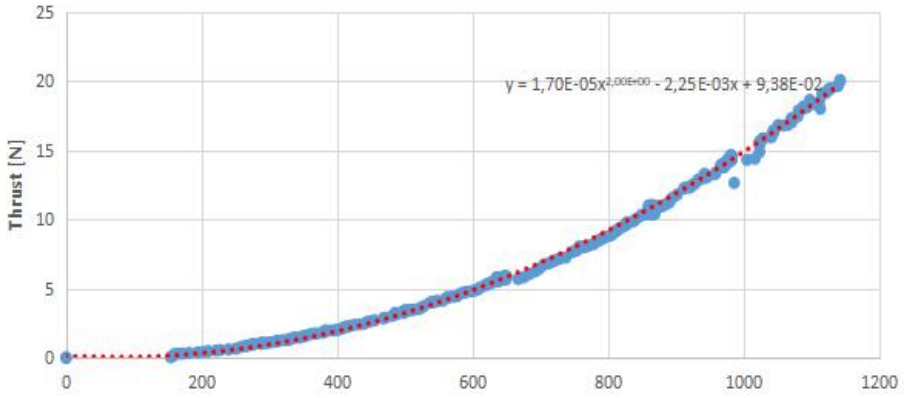
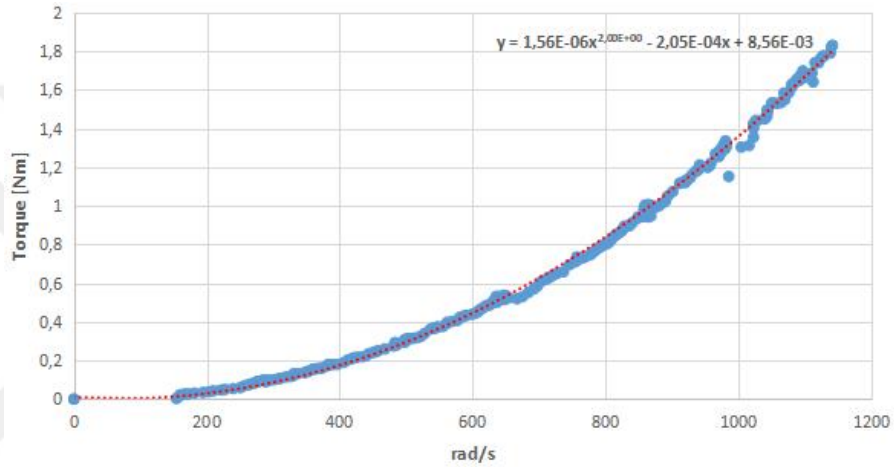


Figure 3.10 Thrust Mechanism CAD Model



(a) Thrust Curve Fit



(b) Torque Curve Fit

Figure 3.11 Thrust and Drag Coefficient Curve Fit

Complete parameter list can be seen in Table 3.5.

Table 3.5 Mechanical Specification Results

Parameter	**Value**	**Unit**
Mass	3.1	kg
Arm Length	0.50	m
Moment of inertia of X axis	0.182	kgm^2
Moment of inertia of Y axis	0.188	kgm^2
Moment of inertia of Z axis	0.355	kgm^2
Thrust Coefficient	1.7*10^(-5)	Ns^2
Torque Coefficient	1.56*10^(-6)	Nms^2

3.2. Sensor Model

One of the most important sensors in aircraft is IMU sensors. The IMU contains two separate sensors, an accelerometer and a gyroscope. Accelerometers have a structure similar to the mass-spring system as seen in the Figure 3.12. When an acceleration is applied in a certain direction, the mass will move and the capacitance value between this mass and the fixed plates will change. Based on this change, acceleration measurement is made.

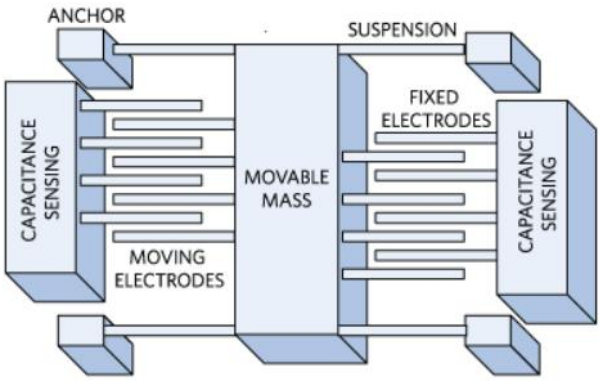


Figure 3.12 IMU Mechanical Model (Analog Devices, 2014)

The working principle of gyroscopes is based on the Coriolis effect. The Coriolis effect is the change in the orbits of objects moving on a system that rotates around its axis. Similar to accelerometers, the measurement will be made according to the change in capacitance from this displacement.

There is an internal MPU6000 IMU sensor on the APM 2.6 Flight Board. 100 Hz interval measurements were taken from this sensor to create the sensor model as shown in Figure 3.13.

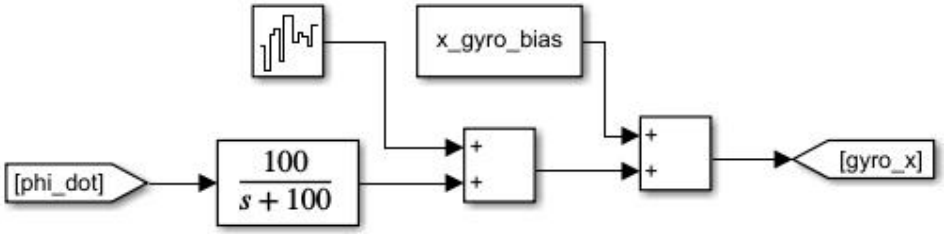
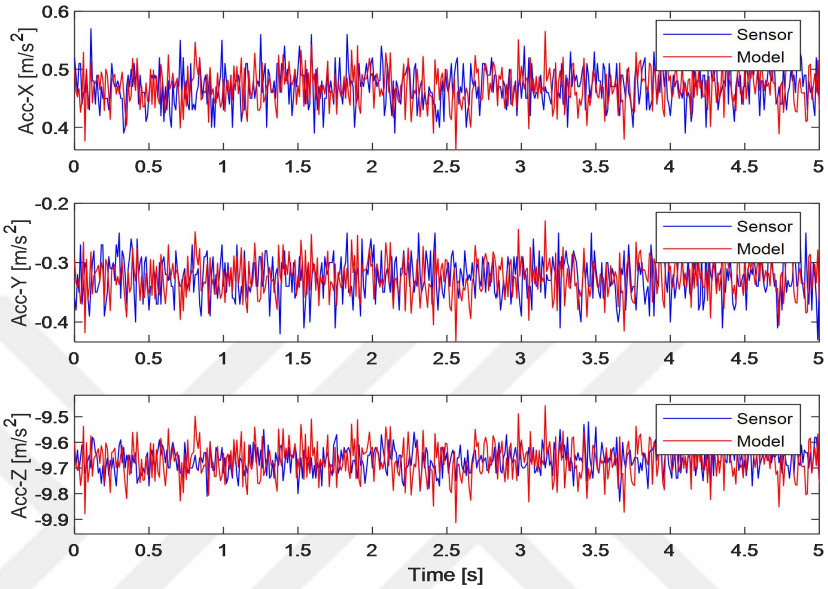
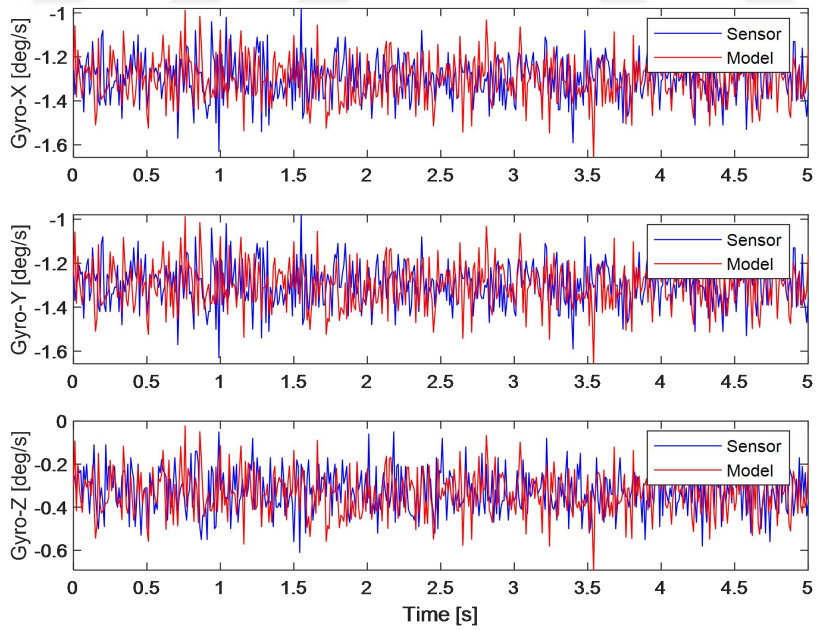


Figure 3.13 Sensor Model

In order to make the simulations of the control system converge to reality, it is important that the models of the components in the system are close to the measurements taken.



(a) Acceleration Result



(b) Gyroscope Result

Figure 3.14 Meausered and Modeled Results

As can be seen in Figure 3.14, the time domain results of the measurements and the model output are close to each other. Figure 3.15 shows the results in the frequency domain. When the graphs were examined, similar results were obtained as in the time domain.

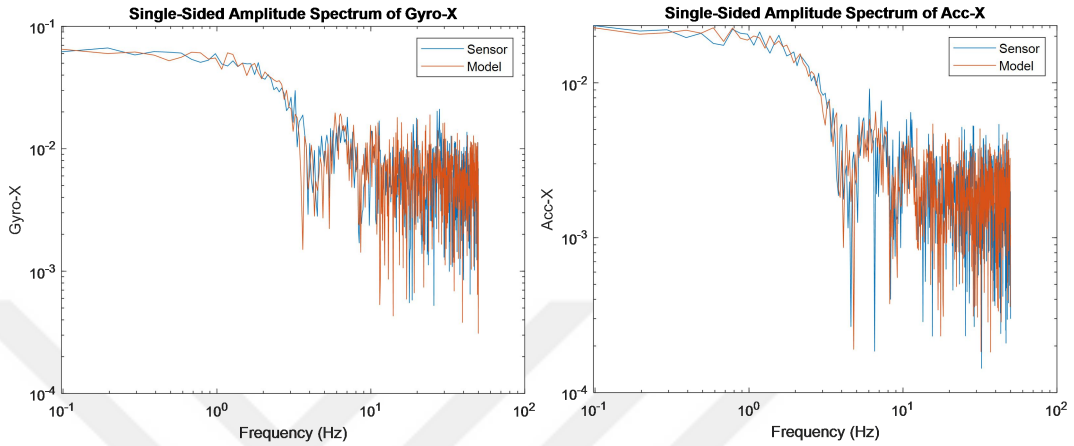


Figure 3.15 Frequency Domain Results

Since raw data from sensors is noisy, it is not healthy to use directly on UAVs (Akın, et al., 2021). Various filters and estimation methods are used to avoid this situation. One of the most effective methods is the Kalman Filter. The Kalman Filter is a filter that can predict values using model equations under the assumptions that the noise has a Gaussian distribution. The algorithm of the Kalman filter is as given in Figure 3.16. In the first two equations, the estimation of the current state and the error covariance matrix are updated. Then the Kalman gain is calculated and the next estimate of the current state is made. Finally, the error covariance matrix is updated.

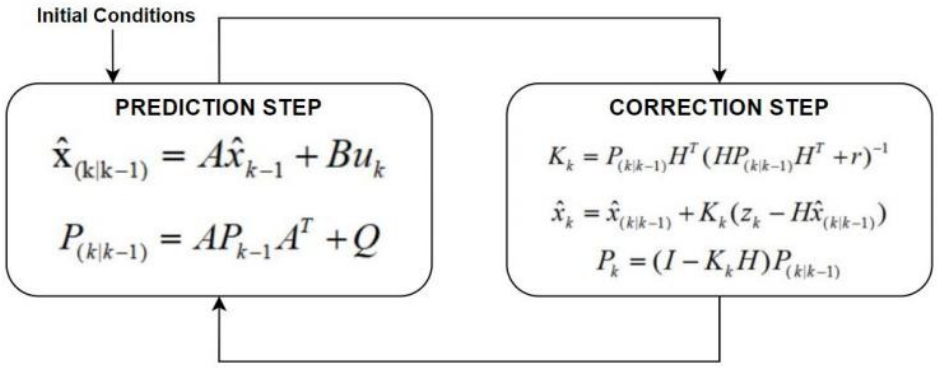


Figure 3.16 Kalman Filter Algorithm

Roll and pitch angles were estimated by combining the accelerometer and gyroscope data with the Kalman filter. As can be seen in Figure 3.17, the noise is greatly reduced at the angles.

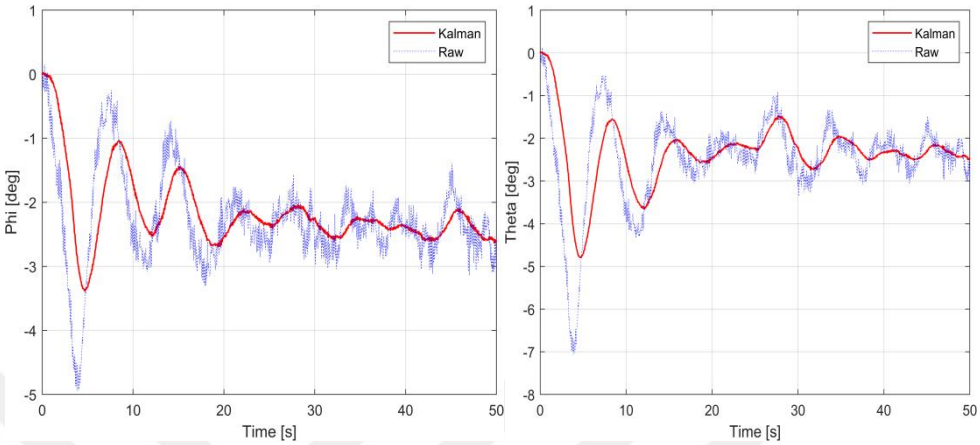


Figure 3.17 Filtered Roll and Pitch Angle

3.3. Wind Model

A wind model was created to increase the fidelity in the model. While creating the wind model, SkyHogg model in MATLAB (Mathworks, 2022) was used. In this model, Dryden win turbulence model, wind shear model and discrete gust model are used together.

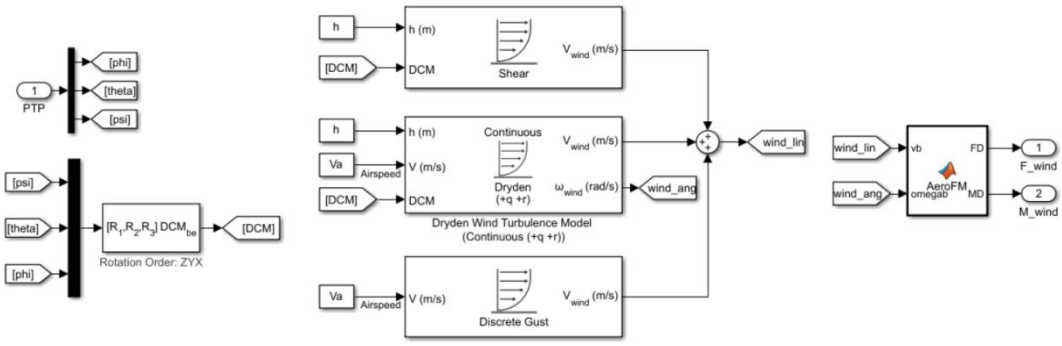


Figure 3.18 Wind Model for the UAV

3.3.1 Wind Shear Model

Wind shear is the change in wind speed over a short distance. The wind shear amplitude can be calculated with the following equation (Mathworks, 2022).

$$U_w = W_{20}.\frac{ln(\frac{h}{z_0})}{ln(\frac{20}{z_0})}, 3ft < h < 1000\ ft \tag{3.3}$$

In this equation, U_w is the mean wind speed, W_{20} is the measured wind speed at 20 ft altitude, h is the altitude, and z_0 is a constant about flight phases. The output of this model is on the body axis.

3.3.2 Discrete Gust Model

In this model, a standard 1-cosine wind is applied. Wind is applied to each axis separately or to all three axes simultaneously (Mathworks, 2022).

$$V_{wind} = \begin{matrix} 0 & x < 0 \\ \frac{V_m}{2}(1 - cos(\frac{\pi x}{d_m})) & 0 \leq x \leq d_m \\ V_m & x > d_m \end{matrix} \tag{3.4}$$

In this equation, V_m is gust amplitude, d_m is gust length, x is the distance traveled, and V_{wind} is the resultant wind speed in body axis. Figure 3.19 shows gust model result.

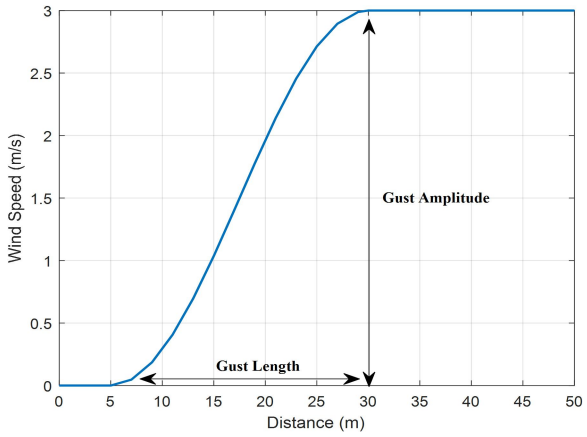


Figure 3.19 Gust Model Graph

3.3.3 Dryden Continous Turbulence Model

Dryden and von Karman wind models are the most common models used to analyze wind effects in aircraft. While the power spectral density (PSD) functions used in the Dryden wind model are rational and can be factored, the PSD functions used in von Karman are irrational (Abichandani, et al., 2020). Dryden turbulence model was used in this study. The dryden turbulence model is characterized according to the PSDs of the linear velocity and rotational velocity components (Hakim and Arifianto, 2018).

$$\Phi_u(\omega) = \frac{2\sigma_u^2 L_u}{\pi V} \frac{1}{1+(L_u \frac{\omega}{V})^2} \tag{3.5}$$

$$\Phi_v(\omega) = \frac{2\sigma_v^2 L_v}{\pi V} \frac{1+12(L_v \frac{\omega}{V})^2}{(1+4(L_v \frac{\omega}{V})^2)^2} \tag{3.6}$$

$$\Phi_w(\omega) = \frac{2\sigma_w^2 L_w}{\pi V} \frac{1+12(L_w \frac{\omega}{V})^2}{(1+4(L_w \frac{\omega}{V})^2)^2} \tag{3.7}$$

$$\Phi_p(\omega) = \frac{\sigma_w^2}{V L_w} \frac{0.8(\frac{2\pi L_w}{4b})^{1/3}}{1+(\frac{4b\omega}{\pi V})^2} \tag{3.8}$$

$$\Phi_q(\omega) = \frac{\pm(\frac{\omega}{V})^2}{1+(\frac{4b\omega}{\pi V})^2} \Phi_w(\omega) \tag{3.9}$$

$$\Phi_r(\omega) = \frac{\pm(\frac{\omega}{V})^2}{1+(\frac{3b\omega}{\pi V})^2} \Phi_v(\omega) \tag{3.10}$$

In these equations, σ turbulence intensities are L scale length and b wing span. To produce a signal with the correct characteristic, filters with band limited white noise are used. These filters are obtained by square roots of Dryden PSD equations. Figure 3.20 shows Dryden model result.

$$\mathrm{H}_u(s) = \sigma_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1+\frac{L_u}{V}s} \tag{3.11}$$

$$\mathrm{H}_v(s) = \sigma_v \sqrt{\frac{2L_v}{\pi V}} \frac{1+\frac{2\sqrt{3}L_v}{V}s}{(1+\frac{2L_v}{V}s)^2} \tag{3.12}$$

$$\mathrm{H}_w(s) = \sigma_w \sqrt{\frac{2L_w}{\pi V}} \frac{1+\frac{2\sqrt{3}L_w}{V}s}{(1+\frac{2L_w}{V}s)^2} \tag{3.13}$$

$$H_p(s) = \sigma_w \sqrt{\frac{0.8}{V}} \frac{(\frac{\pi}{4b})^{1/6}}{(2L_w)^{1/3}(1+\frac{4b}{\pi V}s)} \tag{3.14}$$

$$H_q(s) = \frac{\pm\frac{s}{V}}{(1+\frac{4b}{\pi V}s)} H_w(s) \tag{3.15}$$

$$H_r(s) = \frac{\pm\frac{s}{V}}{(1+\frac{3b}{\pi V}s)} H_v(s) \tag{3.16}$$

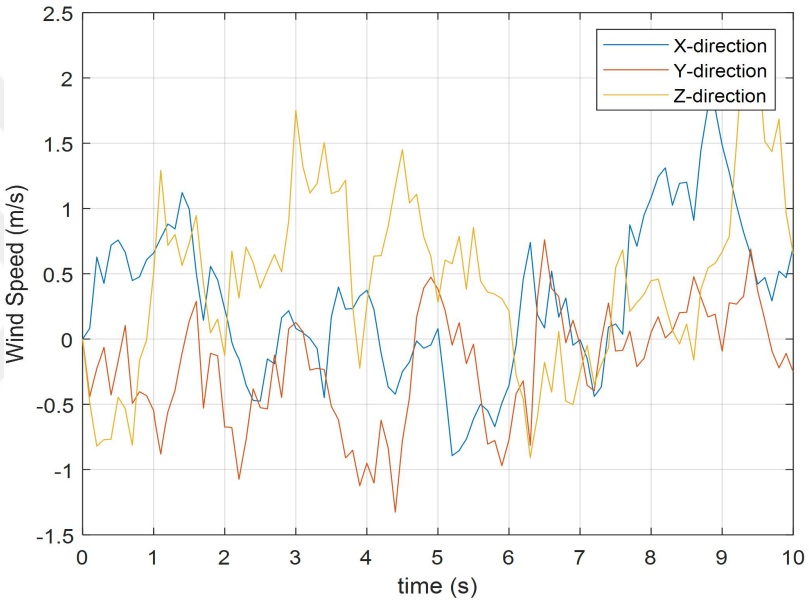


Figure 3.20 Dryden Model Outputs

3.4. Open Loop System Test

The tests were carried out in the Automatic Control and Robotics Laboratory in the Mechanical Engineering Department of Dokuz Eylül University. The vehicle was flown by restricting its height in order to create a safety and simulation environment in a room with a net as shown in Figure 3.21.

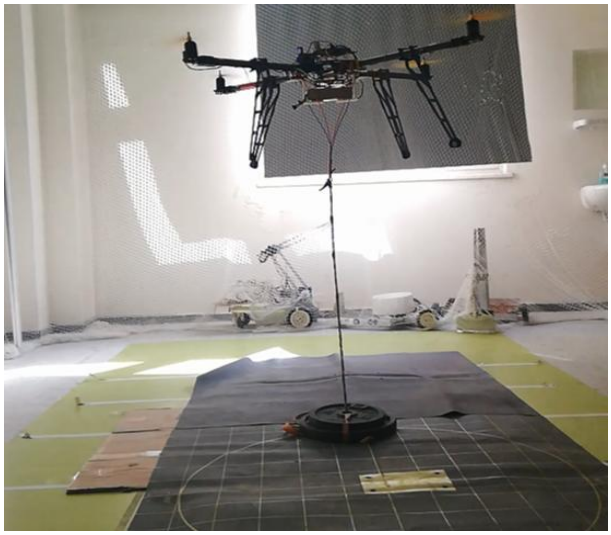


Figure 3.21 Test Environment

During the flight, both the flight control card and an external datalogger seen in Figure 3.22 were used. The datalogger has real-time module, CNY70 IR sensor for measuring rotor RPM, Ultrasonic sensor for measuring altitude data and SD card module for logging. The blue LED indicates that data is being transferred to the SD card.

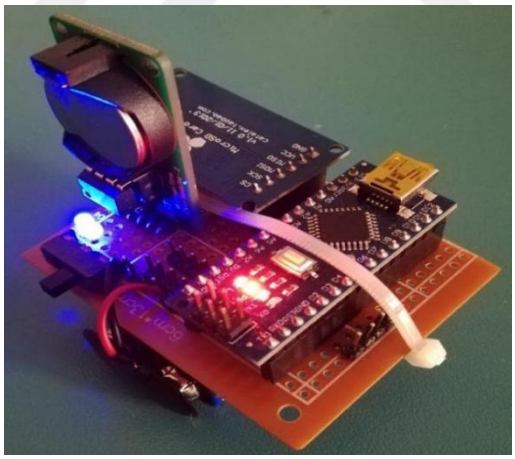


Figure 3.22 External Datalogger

3.5. Simulation Environment

As can be seen in Figure 3.23, simulations were made on the open system. Rotor RPM data is given as input and the altitude result is examined. A more realistic environment has been created by adding acceleration values of the flight control card as well as the parameters of the vehicle.

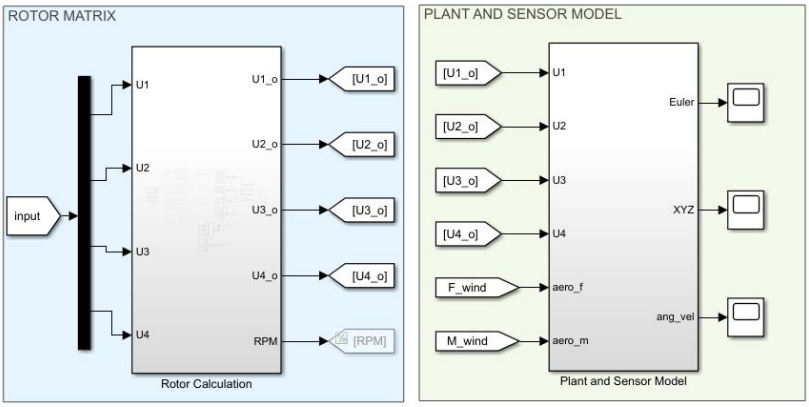


Figure 3.23 Open Loop Model

The model has “Rotor Calculation Block” and “Plant Model”. Rotor Calculation Block includes control input equations. Plant model includes equations of motion and sensor model (acceleration and gyroscope).

As mentioned earlier, the altitude of the vehicle was examined in response to the rotor speed. The take-off and landing times of the vehicle have been taken into account. Figure 3.24 shows the rotor RPM and height graph of the vehicle. As can be seen in the second graph, the behavior of the vehicle shows similar results to the simulation. In addition to altitude data, IMU data will be examined later.

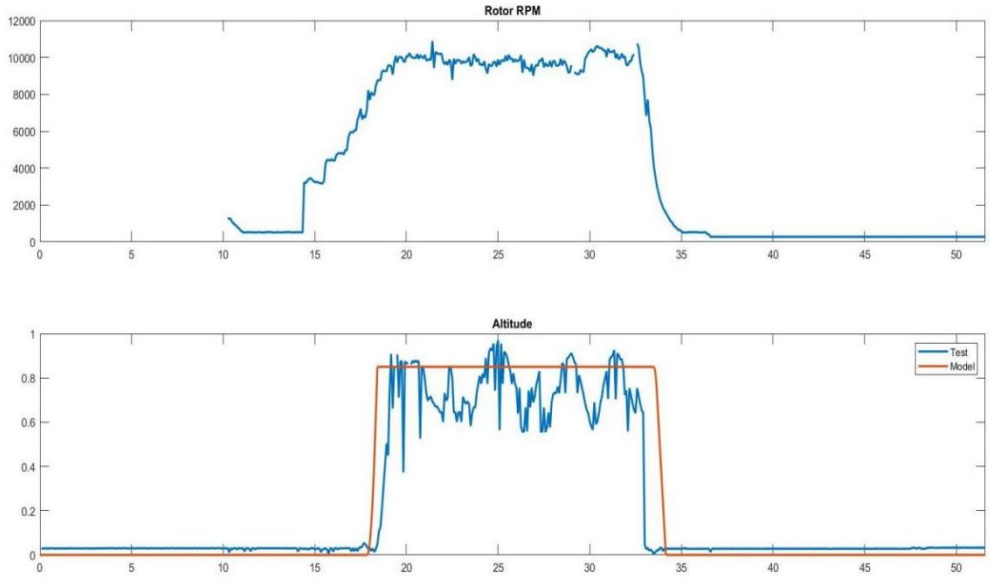


Figure 3.24 Open Loop Test and Simulation Comparison

CHAPTER 4
CONTROL SYSTEM DESIGN

4.1 Control Systems Introduction

In this chapter, different controller designs were examined. A modular environment has been created for simulations. There are controller, rotor and plant model in blocks, respectively. There are position, attitude, altitude and heading controllers in the controller block, rotor speed and control input equations int he rotor block. Plant model consists of linear and angular acceleration equations. The acceleration equations were built in different subsystems to work more comfortably. Wind gust effect and sensor noise have also been added to create a more realistic environment. Figure 4.1 shows the general controller structure. There are two cycles in SMC, position and angle loops. In PID, besides the position and angle loop, there is also an angular speed controller in the inner loop. As seen in the figure, x_d, y_d, z_d and Ψ_d are given as position and angle references for controllers. The position controller uses x_d, y_d and Ψ angles to output ϕ_d and θ_d. Then, these outputs are sent to the angle controllers and the control inputs (U_1,U_2,U_3 and U_4) are created. These values are given as input to the dynamic model, and position and angle results are obtained as outputs.

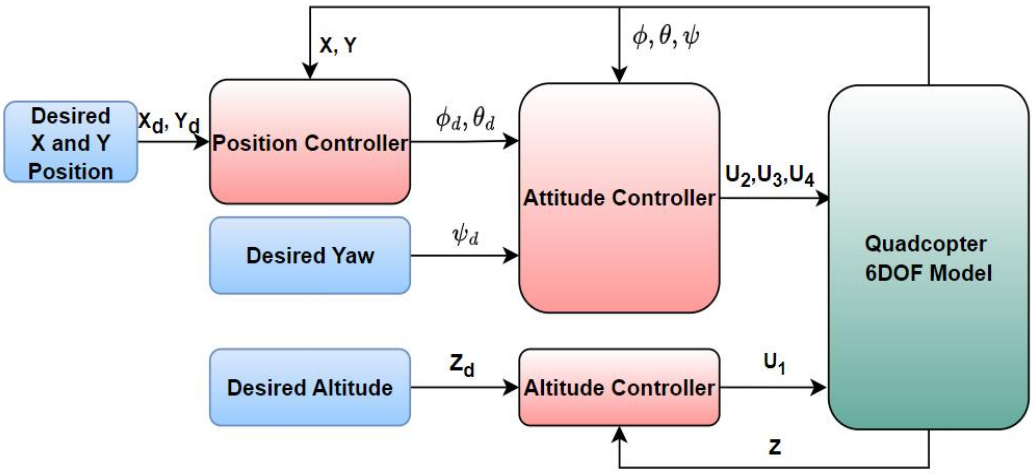


Figure 4.1 General Controller Diagram

4.2 PID Control

PID control is an abbreviation for proportional-integrator-derivative control. PID control is a linear controller which is popular in academic and industrial studies. Also it has the advantage that parameter gains can be easily adjusted, the controller is simple to design and has good robustness (Zhang, et al., 2014). But unmanned aerial vehicles have complex flight dynamics and nonlinearities, so PID control cant perform stability against uncertainties such as wind gusts.

4.2.1 Attitude Controller

Cascade PID controller is used to provide attitude control of the UAV. Since the outer loop calculates the reference for the inner loop in the Cascade control structure, the inner loop is also part of the outer loop. To increase stability, the outer loop shoul be slower than the inner loop. Because the inner loop must work at high frequency due to fast dynamics. Therefore higher gains can be chosen for the inner loop. Simulink models of roll angle and roll angular velocity cycles are shown in Figure 4.2 and Figure 4.3.

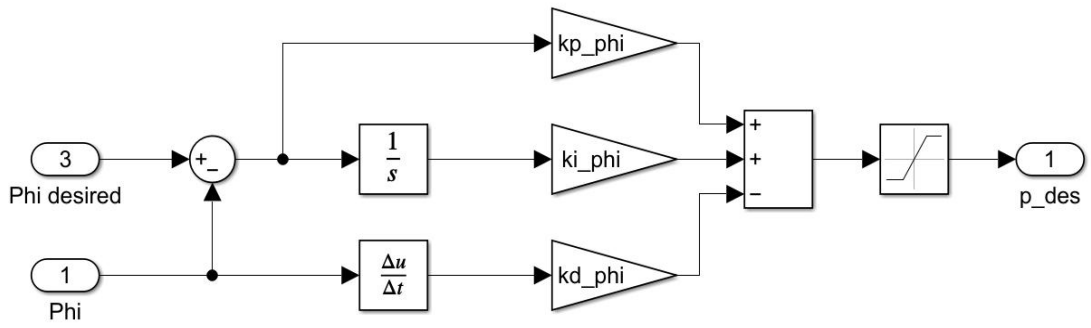


Figure 4.2 PID Angle Controller Model

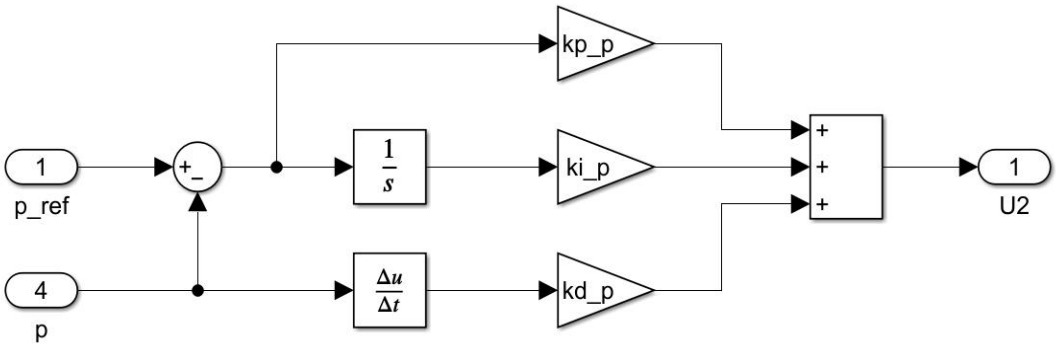


Figure 4.3 PID Angular Velocity Controller Model

As shown in the figures, *derivative kick* used in the controllers. As the error between the reference and the measurement suddenly increases, it causes the derivative of the error to become large instantly and the controller output to become very large.

The following equations are used for angle controllers. The error values here are the difference between the reference value and the measured value. p_{des}, q_{des} and r_{des} values in the equations are the angular velocity commands on the roll, pitch and yaw axes, respectively.

$$p_{des}(t) = K_P e_\varphi(t) + K_I \int_0^t e_\varphi + K_D \frac{de_\varphi(t)}{dt} \tag{4.1}$$

$$q_{des}(t) = K_P e_\theta(t) + K_I \int_0^t e_\theta + K_D \frac{de_\theta(t)}{dt} \tag{4.2}$$

$$r_{des}(t) = K_P e_\Psi(t) + K_I \int_0^t e_\Psi + K_D \frac{de_\Psi(t)}{dt} \tag{4.3}$$

The angular velocity reference values obtained from the angle controllers are sent to the angular speed controllers, and U_2, U_3 and U_4 outputs are obtained.

$$U_2(t) = K_P e_p(t) + K_I \int_0^t e_p + K_D \frac{de_p(t)}{dt} \tag{4.4}$$

$$U_3(t) = K_P e_q(t) + K_I \int_0^t e_q + K_D \frac{de_q(t)}{dt} \tag{4.5}$$

$$U_4(t) = K_P e_r(t) + K_I \int_0^t e_r + K_D \frac{de_r(t)}{dt} \tag{4.6}$$

4.2.2 Altitude Controller

The altitude controller generates the control input U_1 which is used for altitude of the UAV. The PID structure used in the Attitude controller is also used here. In addition, weight has been added to keep the vehicle in balance considering the hover situation. Figure 4.4 shows the PID Altitude controller model in Simulink.

$$U_1(t) = K_P e_z(t) + K_I \int_0^t e_z + K_D \frac{de_z(t)}{dt} + mg \tag{4.7}$$

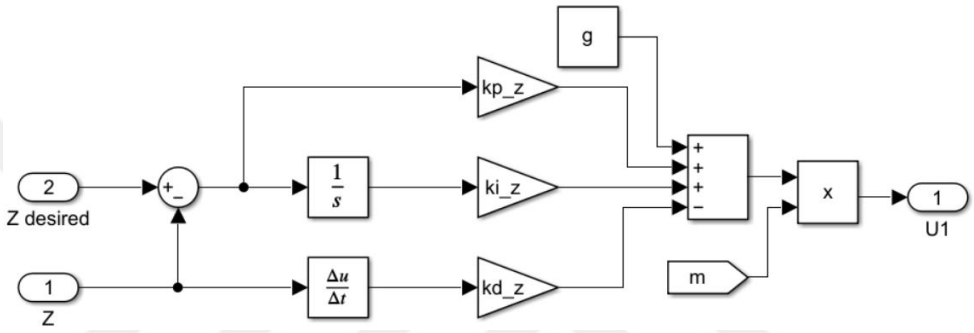


Figure 4.4 PID Altidude Controller Model

4.2.3 Position Controller

In order to control the position of the vehicle, ϕ and θ values are generated to go to the desired location. θ angle is used to move the vehicle in the X direction, and the ϕ angle is used for its movement in the Y axis. The following equations are used to generate these values. PID position controller can be seen in Figure 4.5.

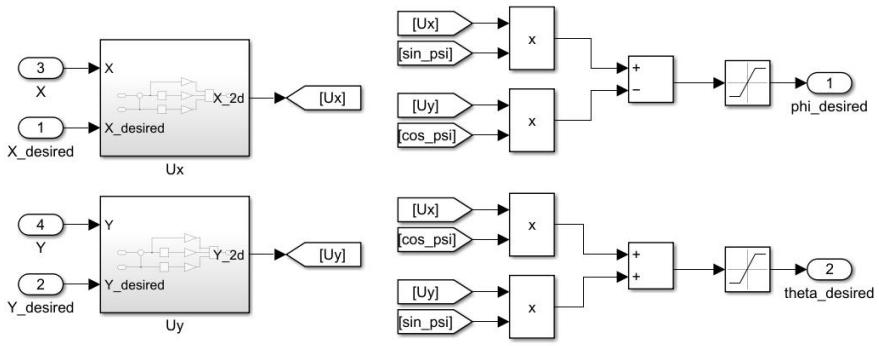


Figure 4.5 Position Controller Model

Let the command to advance the vehicle in the X axis be U_x and the command to advance in the Y axis be U_y. These commands are generated with PID control and the desired angle values are obtained with the following equations.

$$\varphi_d = \sin^{-1}\left(U_x \sin\Psi - U_y \cos\Psi\right) \tag{4.8}$$

$$\theta_d = \sin^{-1}\left(U_x \cos\Psi + U_y \sin\Psi\right) \tag{4.9}$$

4.2.4. PID Parameter Tuning

As mentioned before, PID controllers are frequently used in academic and industrial studies. In addition to being simple in terms of application, it is preferred because it gives effective results with right coefficients. There are various methods for determining these coefficients. Some of these are Ziegler-Nichols method, Cohen-Coon method, Genetic Algorithm and manual tuning. Even if good results are obtained with Ziegler-Nichols, it may take a long time to obtain coefficients due to calculations for complex systems. Another method is changing parameters manually according to the effects of the coefficients. This method is not an healthy method in terms of time. In this study Genetic Algorithm (GA) method was used to obtain PID coefficients for angle controller. A package program that can used in future projects was written using MATLAB.

Genetic algorithms, which are in the class of evolutionary algorithms based on Darwin's biological evolution principles, are population-based optimization algorithms (Messac, 2015). In Figure 4.6, genetic algorithm flowchart can be seen.

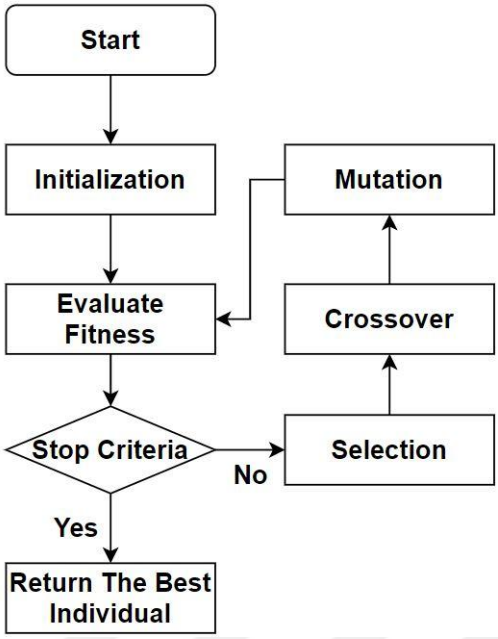


Figure 4.6 Genetic Algorithm Flowchart

The model is linearized by ignoring gyroscopic effects and drag force (Alexis and Tzes, 2011; Eskandarpour and Sharf, 2020). As a result, the following equations were obtained.

$$J_{xx}\ddot{\varphi} = lU_2 \tag{4.10}$$

$$J_{yy}\ddot{\theta} = lU_3 \tag{4.11}$$

$$J_{zz}\ddot{\Psi} = U_4 \tag{4.12}$$

If these equations are rewitten as tranfer funtion, below equations will be obtained,

$$F_{\varphi}(s) = \frac{\varphi(s)}{U_2(s)} = \frac{l}{J_{xx}s^2} \tag{4.13}$$

$$F_{\theta}(s) = \frac{\theta(s)}{U_3(s)} = \frac{l}{J_{yy}s^2} \tag{4.14}$$

$$F_{\Psi}(s) = \frac{\Psi(s)}{U_4(s)} = \frac{1}{J_{zz}s^2} \tag{4.15}$$

Multirotor unmanned aerial vehicles are Multi-Input-Multi-Output systems. Linearized equation set was used to examine the system dynamics more easily and to implement the genetic algorithm. There are various tuning criteria to minimize the amount of error. In this study, ITAE (Integral time absolute error) criterion is used as

an objective function to tune the PID parameters. As the result of the tuning process, PID coefficients for roll, pitch and yaw angles are obtained.

$$\mathrm{ITAE} = \int_0^\infty t|e(t)|dt \tag{4.16}$$

The simulations were carried out in such a way that the population number was 50 and the generation number was 50. Approximately 20 simulations have been made and it has been observed that there are very close values in each simulation.

4.3 Sliding Mode Control

Control systems have two main goals: Firstly, creating the system's response to perform the desired behavior. Secondly, protecting current state against disturbances affecting the system during the process (Analia and Song, 2016).

The biggest advantage of robust control is it can achieve both goals. The theory evaluates feasibility, generates an appropriate control law, and provides assurance on its validity range. Robust control methods are very effective especially in MIMO systems. Unmodeled dynamics, uncertainities in parameters, assumptions, measurement errors and unpredicted external factors may disrupt the system. Therefore modeling the system and prevent against uncertainties is also important. The main purpose of robust control systems is to find the control law that preserves the outputs and errors of the system within the allowable limits, despite the existence of an uncertainty in the controlled system (Madoyan, 2009). In this way, there are some robust control methods such as Sliding Mode Control, Backstepping Control and H-infinity.

SMC was introduced by Emelyanov in the 1950s and became widespread with the work of Utkin in the 1970s (Utkin, 1977). It is the Variable Structure System Control (VSSC) method, on which various improvements have been made until today. VSSC control has two main goals:

- It drives the state trajectory of a nonlinear system to a specific and user-selected surface in the state space, called the sliding or switching surface. The state trajectory of the system has different gains according to its position relative to this surface.

- During the next time interval the system maintains its state trajectory on this surface. During the process, the structure of the control system changes from one to the next, hence the name variable structure control (Liu and Wang, 2011).

Sliding mode control is robust control method used to control systems with measurement errors and uncertainties. The basic idea in this control method is to force an n-order system to perform a first-order behavior. First order dynamics called sliding surface is created in an nth order system and the system is provided to reach this surface and stay there. One of the advantages of SMC is the reduced importance of the underlying dynamics of the system.

4.3.1 SMC Background

The purpose of SMC is to control the system by keeping around the surface defined as the sliding surface. Let's consider the system below. In this equation, x is output and u is control input (Slotine and Li, 1991).

$$x^n = f(x) + b(x).u \tag{4.17}$$

Control law for the SMC can be examined in equation below. This equation has two parts: corrective control and equivalent control.

$$U(t) = U_d(t) + U_{eq}(t) \tag{4.18}$$

Corrective function is necessary for disturbance rejection and the equivalent function ensures that the time derivative of the surface is kept at zero so that the state trajectories remain on the sliding surface.

The error value is calculated to obtain sliding surface.

$$e = x_d - x \tag{4.29}$$

The sliding surface is determined according to this tracking error found.

$$\mathrm{s(x,t)} = (\frac{\mathrm{d}}{\mathrm{dt}} + \lambda)^{\mathrm{n}-1}\mathrm{e} \tag{4.20}$$

For second order equation n = 2 and λ is a positive tuning parameter.

$$\mathrm{s} = \dot{\mathrm{e}} + \lambda e \tag{4.21}$$

By replacing the error function in sliding surface,

$$\mathrm{s} = (\dot{\mathrm{x}}_\mathrm{d} - \dot{\mathrm{x}}) + \lambda(\mathrm{x_d} - \mathrm{x}) \tag{4.22}$$

Sliding condition is $\dot{s} = 0$, so the following equation is obtained.

$$\dot{s} = \ddot{e} + \lambda\dot{e} = (\ddot{x_d} - \ddot{x}) + \lambda(\dot{x_d} - \dot{x}) \tag{4.23}$$

$$\dot{s} = f + u - \ddot{x_d} + \lambda\dot{e} \tag{4.24}$$

If the controller is designed to make the derivative of the sliding surface zero, the estimated equivalent controller is shown below.

$$\hat{u} = -\hat{f} + \ddot{x_d} - \lambda\dot{e} \tag{4.25}$$

A Lyapunov function is defined to find $U_d(t)$. Lyapunov function is a function used to examine the stability of nonlinear systems. Using Lyapunov function limits the selected switching control law gain (Runcharoon and Srichatrapimuk, 2013). This function itself must be positive.

$$\mathrm{V} = \frac{1}{2}\mathrm{s}^2 > 0 \tag{4.26}$$

The derivative of the Lyapanuv function must be negative definite to confirm the motion of the state trajectory.

$$\dot{V} = \dot{s}s < 0 \tag{4.27}$$

The following equation is used to satisfy this condition.

$$U_d = k_d . sgn(s) \tag{4.28}$$

Where

$$\text{sgn(s)} = \begin{matrix} 1 \text{ if s} > 0 \\ -1 \text{ if s} < 0 \end{matrix}$$

SMC is a switched control method and high frequency oscillations occur to stay on the sliding surface. This phenomenon is called the chattering effect. To reduce the chattering effect, this function can be written as below (Herrera, et al., 2015).

$$U_d = k_d . \frac{s}{|s|+\delta} \tag{4.29}$$

4.3.2 Altitude Controller

As explained in the previous section, the sliding surface is first defined according to the error value.

$$\text{e} = \text{z}_\text{d} - \text{z} \tag{4.30}$$

$$\text{s}_\text{z} = \dot{\text{z}}_\text{d} - \dot{\text{z}} + \lambda(\text{z}_\text{d} - \text{z}) \tag{4.31}$$

Then the sliding condition is met.

$$\dot{s_z} = (\ddot{z_d} - \ddot{z}) + \lambda(\dot{z_d} - \dot{z}) \tag{4.32}$$

The sliding condition is rewritten according to the altitude equation. Since the system is in sliding condition, the assumption of U_1=U_{eq} can be made (Herrera, et al., 2015).

$$\dot{s_z} = (\ddot{z_d} + g - \frac{cos\varphi.cos\theta}{m} U_{eq} + \lambda(\dot{z_d} - \dot{z})) \tag{4.33}$$

Since $\dot{s_z} = 0$, the equation can be arranged as follows.

$$U_{eq} = m(\frac{\ddot{z_d}+g+\lambda(\dot{z_d}-\dot{z})}{cos\varphi.cos\theta}) \tag{4.34}$$

To design U_d, Lyapunov function is defined and the following equation is used to make this function negative.

$$U_d = k_z.\frac{s_z}{|s_z|+\delta} \tag{4.35}$$

After finding the U_d, the controller equation is obtained. The Simulink model for altitude controller is as shown in Figure 4.7.

$$U_1 = m(\frac{\ddot{z_d}+g+\lambda(\dot{z_d}-\dot{z})}{cos\varphi.cos\theta}) + k_z.\frac{s_z}{|s_z|+\delta} \tag{4.36}$$

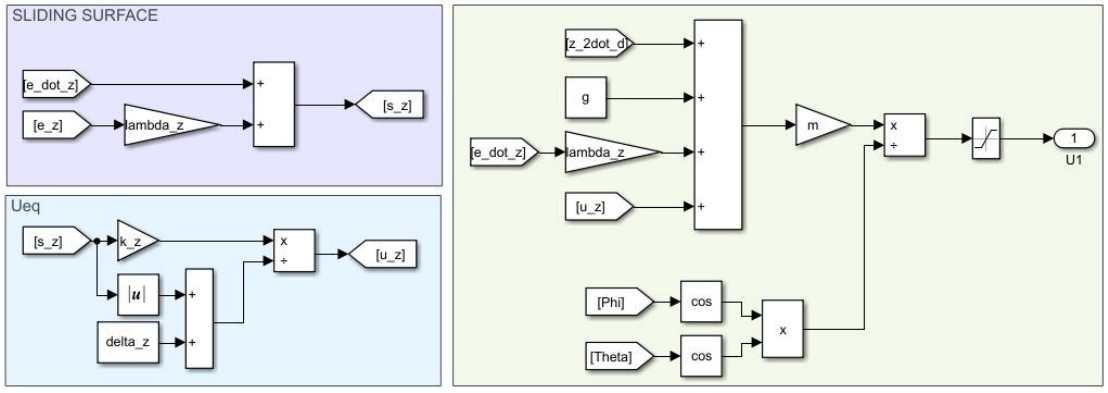


Figure 4.7 SMC Altitude Controller Model

4.3.3 Attitude Controller

Similar to the altitude controller, the attitude controller is also created. Unlike the altitude controller, the U_{eq} function is rewritten according to the angle equations. The Simulink model for attitude controller is as shown in Figure 4.8.

$$U_2 = \frac{J_x}{l}[\lambda(\dot{\varphi}_d - \dot{\varphi}) + \ddot{\varphi}_d - \frac{(J_y - J_z)}{J_x}\dot{\theta}\dot{\Psi} + \frac{J_m w_r}{J_x}\dot{\theta} + k_{\varphi}\cdot\frac{s_\varphi}{|s_\varphi| + \delta}] \quad (4.37)$$

$$U_3 = \frac{J_y}{l}[\lambda(\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d - \frac{(J_z - J_x)}{J_y}\dot{\varphi}\dot{\Psi} - \frac{J_m w_r}{J_y}\dot{\varphi} + k_{\theta}\cdot\frac{s_\theta}{|s_\theta| + \delta}] \quad (4.38)$$

$$U_4 = \frac{J_z}{l}[\lambda(\dot{\Psi}_d - \dot{\Psi}) + \ddot{\Psi}_d - \frac{(J_x - J_y)}{J_z}\dot{\varphi}\dot{\theta} + k_{\Psi}\cdot\frac{s_\Psi}{|s_\Psi| + \delta}] \quad (4.39)$$

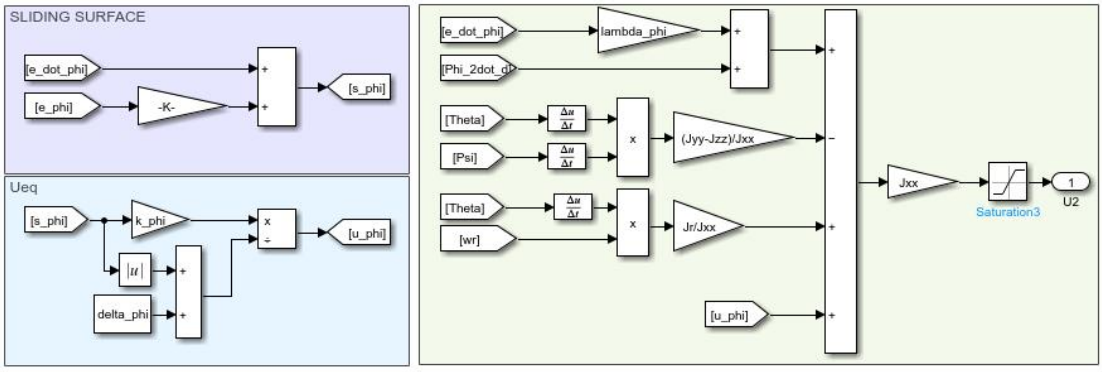


Figure 4.8 SMC Attitude Controller Model

4.3.4 Position Controller

As in PID control, phi and theta angles are produced for position control. First of all, U_x and U_y virtual inputs are defined.

$$U_x = \frac{[\ddot{x}_d + (\dot{x}_d - \dot{x})\lambda_x + k_x\cdot\frac{s_x}{|s_x| + \delta_x}]m}{U_1} \quad (4.40)$$

$$U_y = \frac{[\ddot{y}_d + (\dot{y}_d - \dot{y})\lambda_y + k_y\cdot\frac{s_y}{|s_y| + \delta_y}]m}{U_1} \quad (4.41)$$

Then, according to these inputs, the desired phi and theta angles are obtained according to the equation below.

$$\varphi_d = asin(U_x sin\Psi - U_y cos\Psi) \quad (4.42)$$

$$\theta_d = \frac{asin(U_x cos\Psi + U_y sin\Psi)}{cos\varphi_d} \quad (4.43)$$

CHAPTER 5
RESULTS

5.1 Simulation Environment and Conditions

In this section, different simulations are made to compare controller performance in various conditions and environments. Firstly, the angle command was given as step input and the vehicle's attitude control was examined. In order to examine the vehicle's position control, simulations were made in different trajectories and conditions. As described in the previous sections, nonlinear effects such as sensor model, wind model and mass reduction for agricultural applications have been added to increase the reality of the model. The effect of these disturbances on the controllers was investigated. Table 5.1 includes controller coefficients.

Table 5.1 Controller Coefficients

	PID			**SMC**		
	$\mathbf{K_p}$	$\mathbf{K_i}$	$\mathbf{K_d}$	$\boldsymbol{\lambda}$	**k**	$\boldsymbol{\delta}$
$\boldsymbol{\varphi}$	7.266	0	0.9841	3	5	0.5
$\boldsymbol{\theta}$	8.0253	0	1.118	3	5	0.8
$\boldsymbol{\Psi}$	9.9	0	1.4465	4	4	0.5
$\boldsymbol{x}$	4	0.1	3	4	4	0.5
$\boldsymbol{y}$	4	0.1	3	4	4	0.5
z	10	2	6	4	4	0.1
p	5	0.5	0	-	-	-
q	5	0.5	0	-	-	-
r	5	0.5	0	-	-	-

5.2 Attitude Control Results

In these simulations, the attitude control of PID and SMC controllers in a noiseless and windless environment were examined first. Then, the simulations were made by turning on the sensor noises. A step input of 0.25 radians for angles and 1 m

for height is applied. The results are as seen in Figure 5.1 and 5.2. Settling time and overshoot comparisions can be seen in Table 5.2.

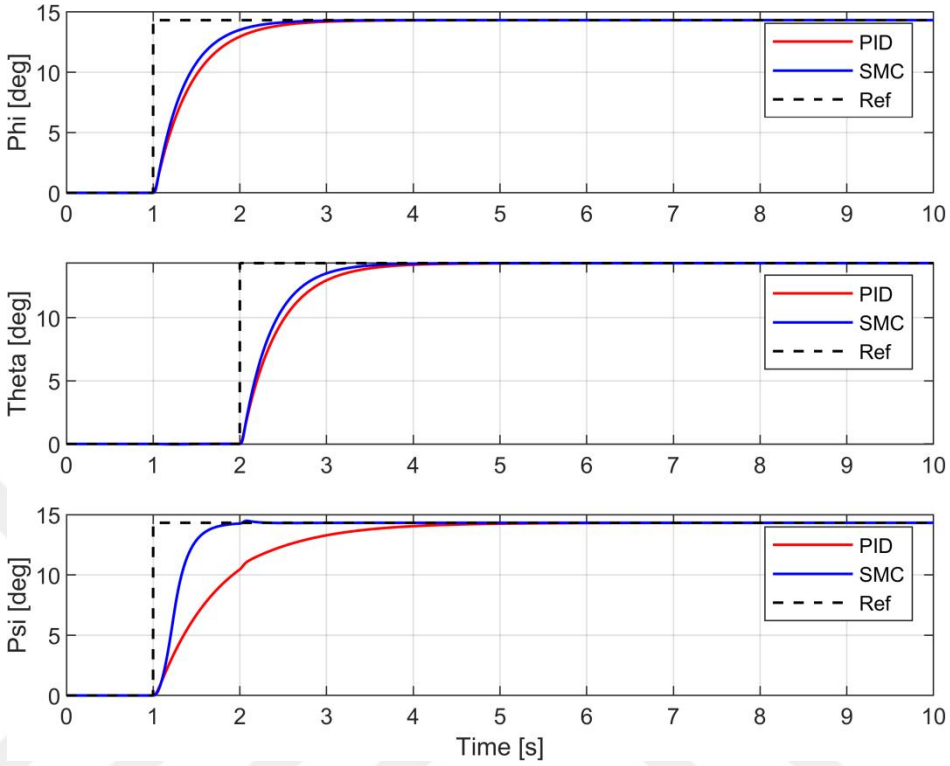


Figure 5.1 Angle Step Input Results without Disturbance

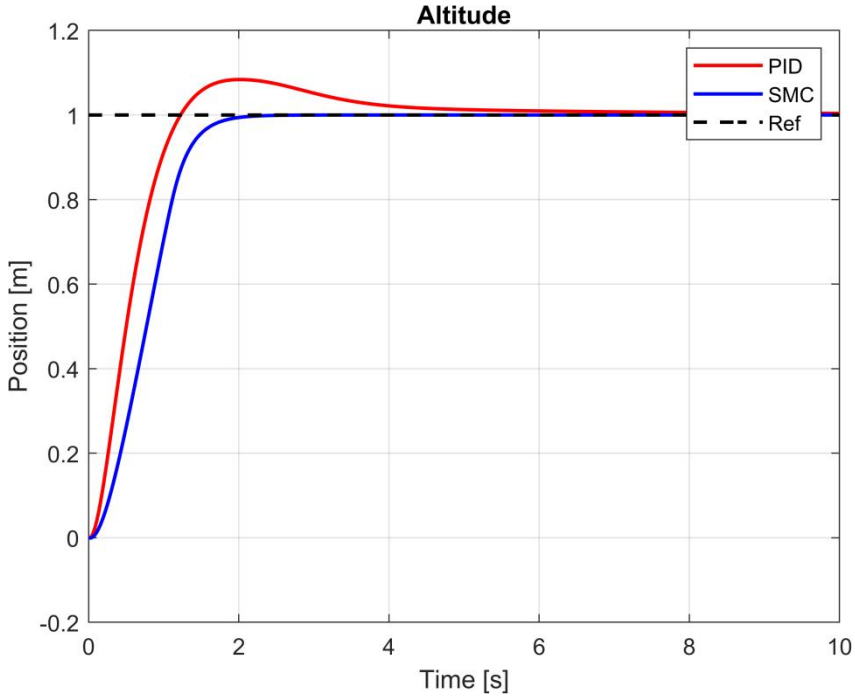


Figure 5.2 Altitude Step Input Results without Disturbance

Table 5.2 Step Input Results without Disturbance

Controller	Disturbance	Input	Settling Time (s)	Overshoot (%)	Steady-State Error
PID	**-**	**Phi**	2.6490	0	7.11e-04 deg
		Theta	3.6513	0	0.0019 deg
		Psi	3.9902	0	0.0017 deg
		Altitude	4.1165	8.3966	-0.0038 m
SMC	**-**	**Phi**	2.3471	0	-4.48e-05 deg
		Theta	3.3490	0	-4.48e-05 deg
		Psi	1.7563	1.1925	-4.48e-05 deg
		Altitude	1.6946	0.0020	-9.88e-06 m

As seen in the results, PID and SMC showed close results when there were no disturbance effects. As mentioned earlier, the PID is a linear controller and performs well in the absence of disturbing inputs. According to the results in the table, SMC performed better in terms of both settling time and overshoot.

By activating the sensor model, nonlinearity was added to the system and the results were examined. The same step inputs as previous simulations were applied. The results are as seen in the Figure 5.3 and 5.4. Settling time and overshoot comparisions can be seen in Table 5.3.

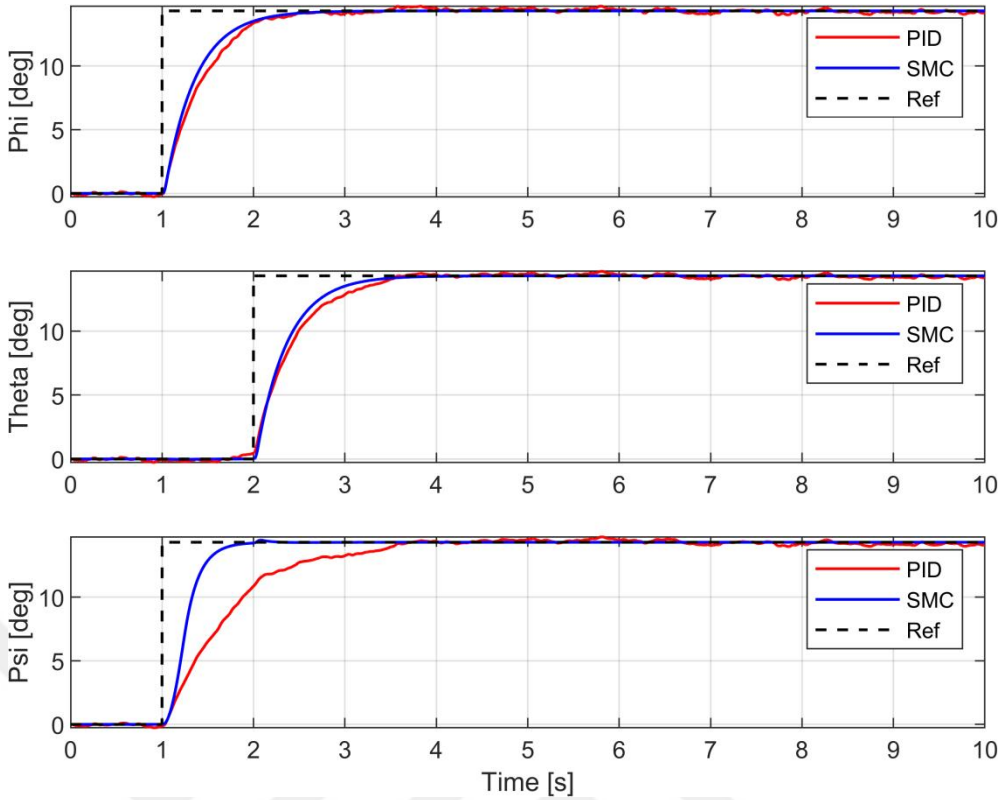


Figure 5.3 Angle Step Input Results with Disturbance

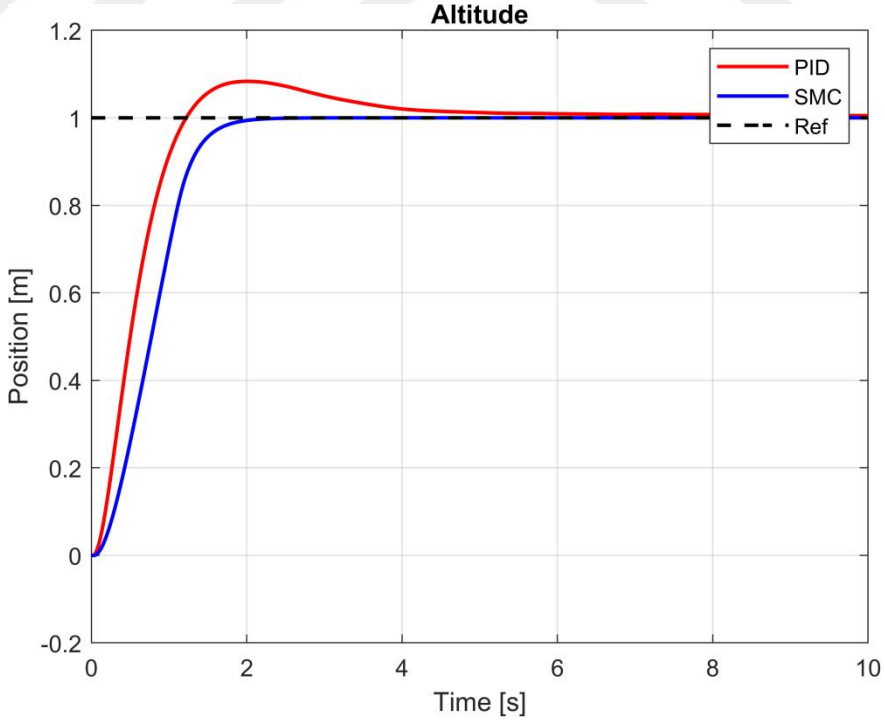


Figure 5.4 Altitude Step Input Results with Disturbance

Table 5.3 Step Input Results with Disturbance

Controller	Disturbance	Input	Settling Time (s)	Overshoot (%)	Steady-State Error
PID	Sensor	Phi	8.7517	2.6646	0.1435 deg
		Theta	8.7507	2.4909	0.1492 deg
		Psi	9.8044	2.9705	0.2247 deg
		Altitude	4.0239	8.3440	-0.0051 m
SMC	Sensor	Phi	2.3471	0	-4.48e-05 deg
		Theta	3.3490	0	-4.48e-05 deg
		Psi	1.7563	1.1925	-4.48e-05 deg
		Altitude	1.6958	0.0082	-3.39e-05 m

As can be seen in the results, after the sensor model was activated, great changes occurred in the PID control compared to the previous simulation results. In addition, oscillating responses are seen in the outputs. In contrast, the SMC control was not much affected by these sensor noises and showed very close results with the previous simulation.

5.3 Position Control Results

Different trajectories were created to examine the performance of the position controllers. These trajectories are circular and square. First of all, the performance in the windless environment was examined together with the sensor model. The results are as seen in Figure 5.5 and Figure 5.6.

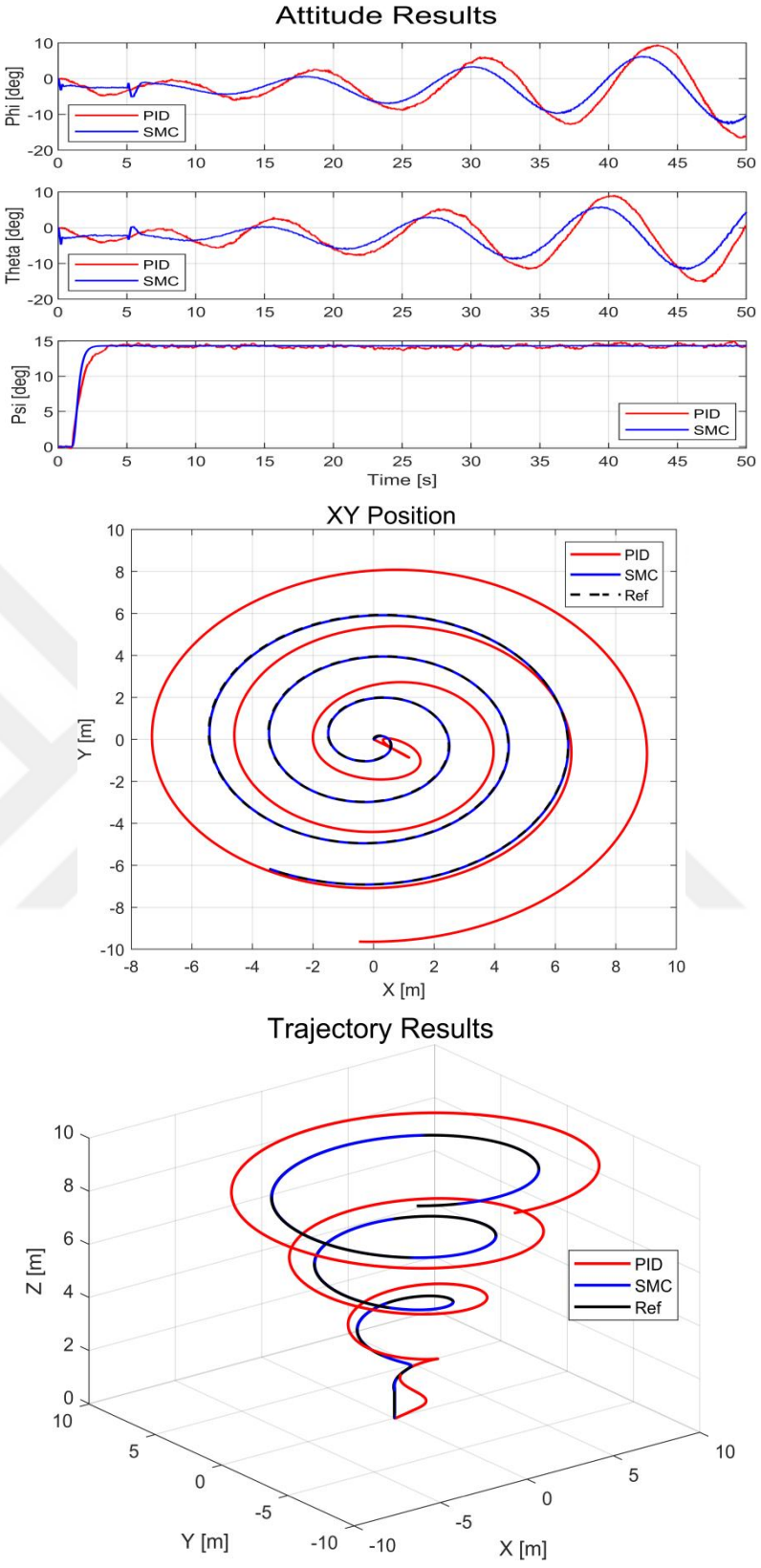


Figure 5.5 Trajectory Tracking without Disturbances in Circular Trajectory

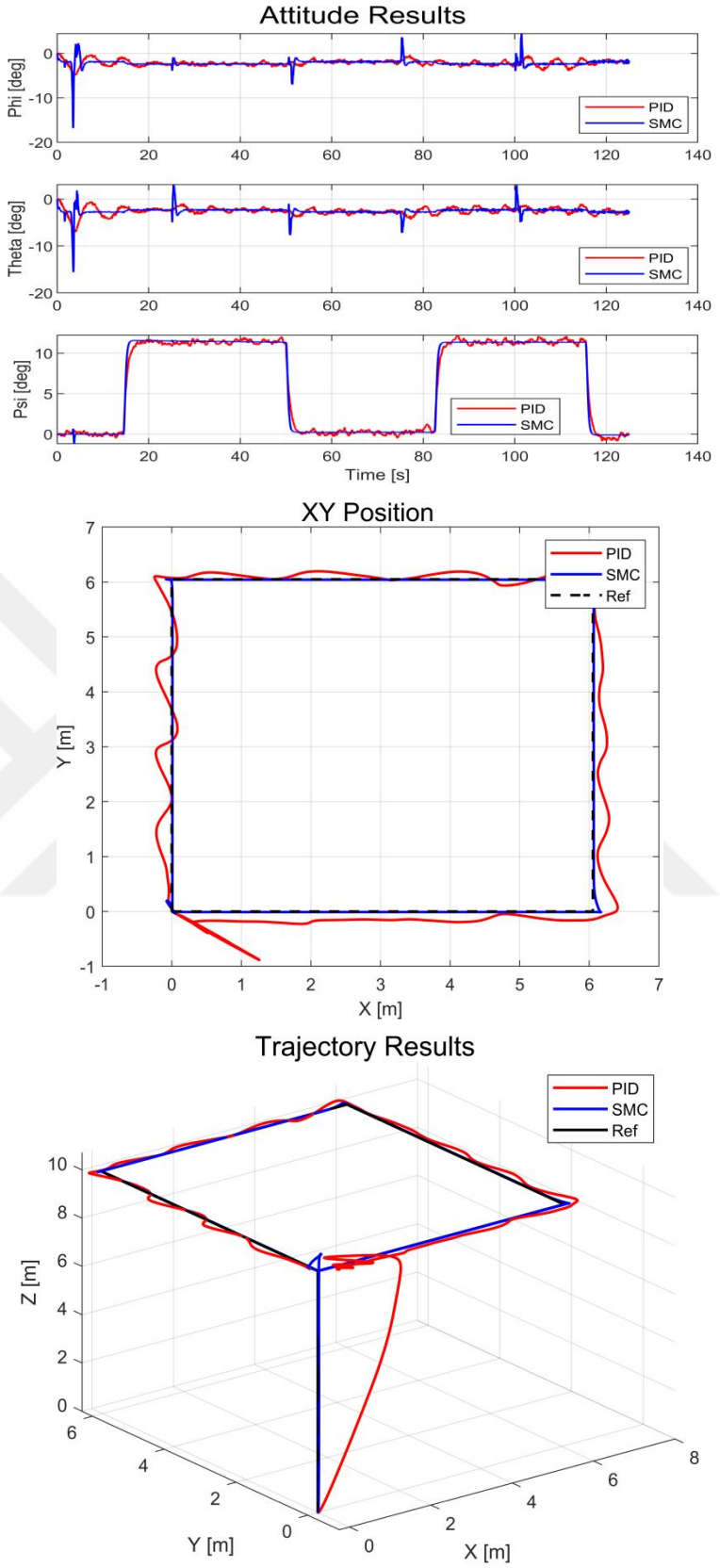


Figure 5.6 Trajectory Tracking without Disturbances in Square Trajectory

As seen in the graphics, the SMC controller provided better position tracking. When the circular motion was examined, the SMC both followed the trajectory properly and less oscillation was observed at the angles. Although the oscillation is low in PID control, the error in X and Y is increasing gradually. Better results can be obtained by adjusting the coefficients. When the square trajectory is examined, high amplitude overshoots occurred at the angles at the first start of the motion in X and Y, and then the overshoots decreased. Due to the initial overshoots, the vehicle deviated by 1 m and then started to follow the trajectory. In PID control, overshoots were observed throughout the flight. When the RMSE results are examined as shown in Table 5.4, it is seen that SMC performs better.

Table 5.4 Trajectory Tracking Results without Wind Effect

Trajectory	**Disturbance**	**Controller**	**Axis**	**RMSE**
Circular	Sensor	PID	X	1.8671
			Y	1.7217
			Z	0.0507
		SMC	X	0.0150
			Y	0.0145
			Z	0.0013
Square	Sensor	PID	X	0.2998
			Y	0.2438
			Z	0.3719
		SMC	X	0.0215
			Y	0.0226
			Z	0.0959

In the next simulation, variable wind input is given. The wind is added to be around 10 m/s starting after 20 seconds. The results can be seen in Figure 5.7, Figure 5.8 and Table 5.5.

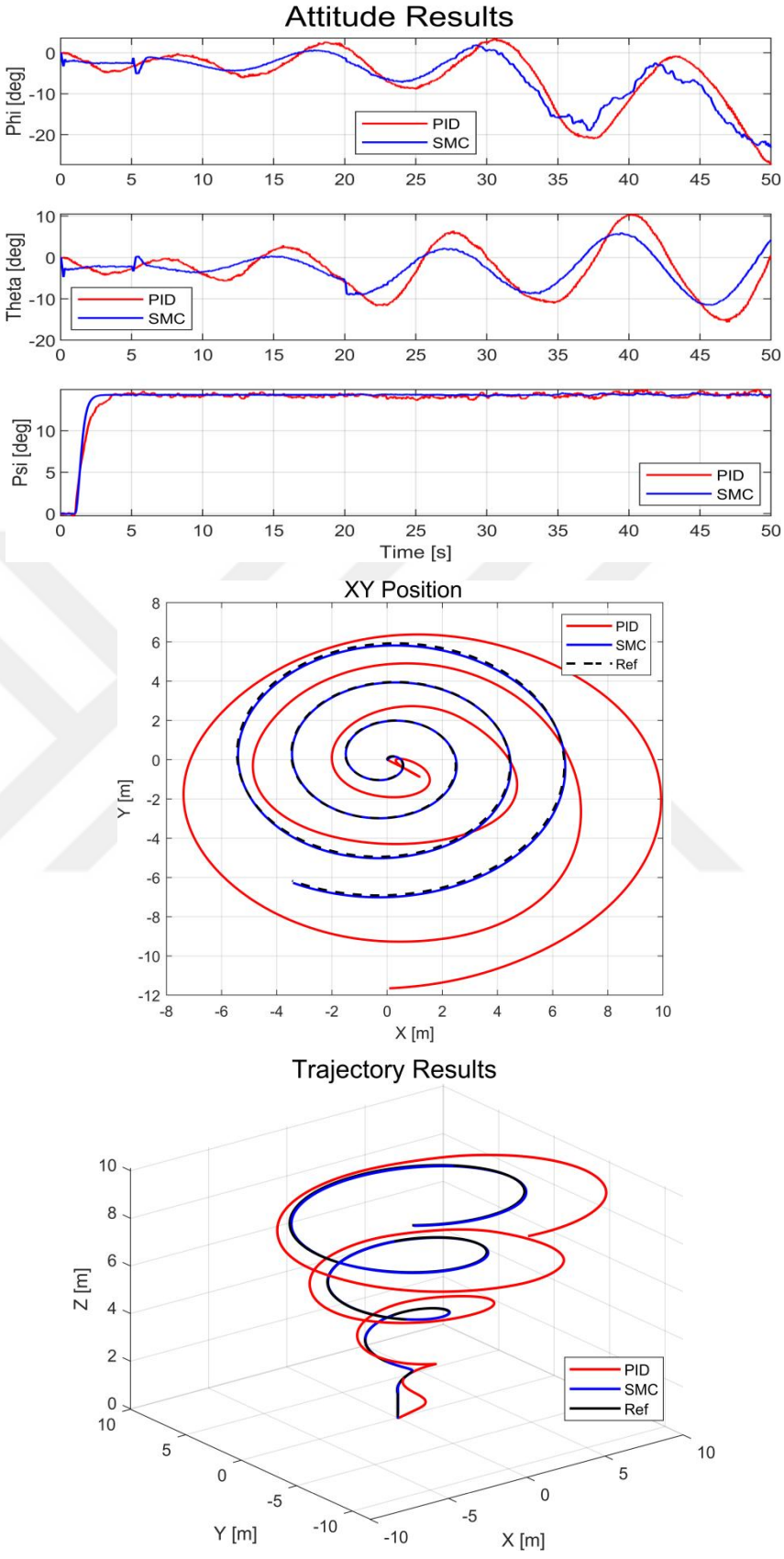


Figure 5.7 Trajectory Tracking with Disturbances in Circular Trajectory

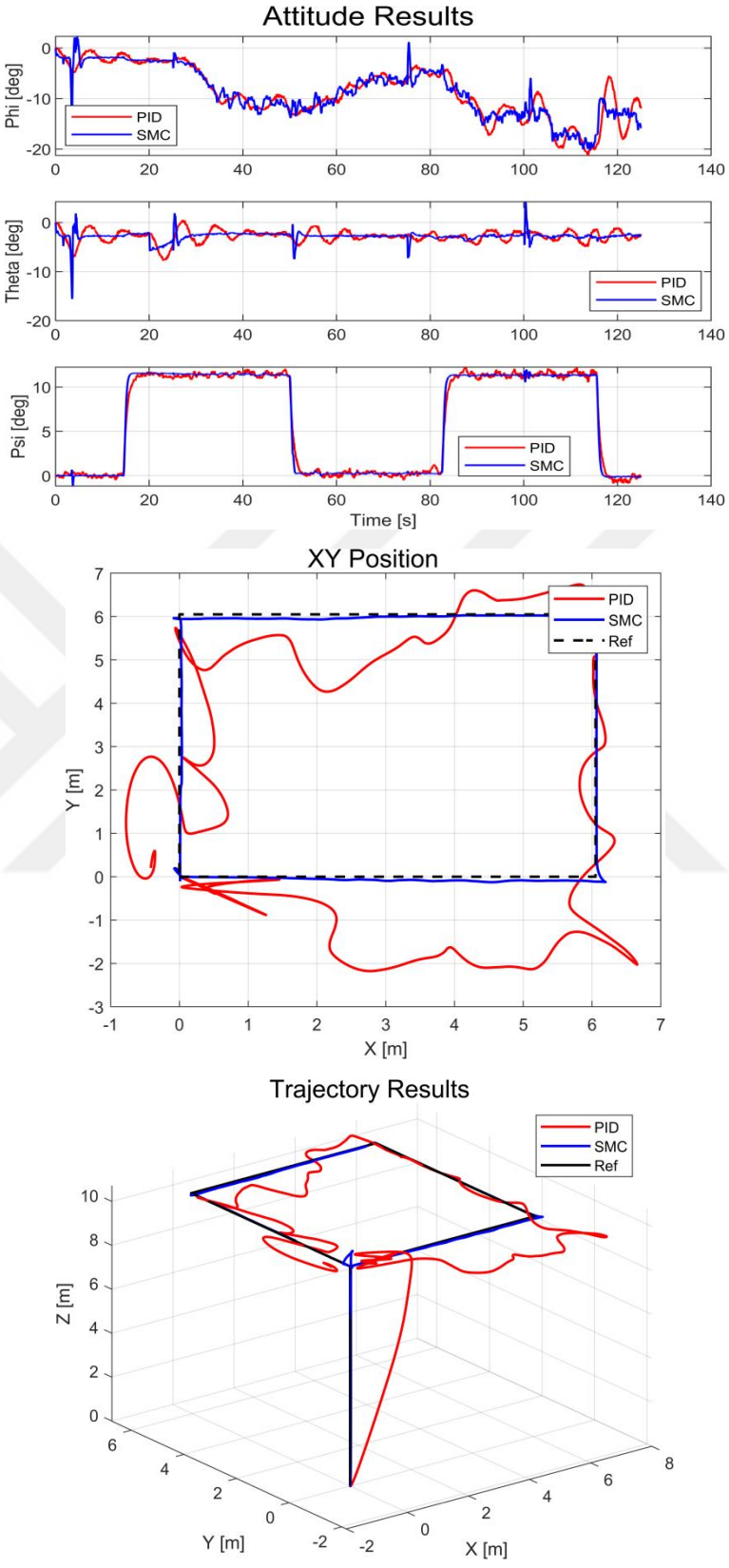


Figure 5.8 Trajectory Tracking with Disturbances in Square Trajectory

When the graphs are examined, the PID controller has drifted due to the wind effect and has an increasing error. In square trajectory, it could not follow the trajectory properly. SMC control, on the other hand, followed the trajectory better than the PID controller, even if there was some deterioration.

Table 5.5 Trajectory Tracking Results with Wind Effec

Trajectory	Disturbance	Controller	Axis	RMSE
Circular	Sensor Wind	PID	X	2.1688
			Y	2.0374
			Z	0.0965
		SMC	X	0.0255
			Y	0.0591
			Z	0.0072
Square	Sensor Wind	PID	X	0.4792
			Y	1.0420
			Z	0.3750
		SMC	X	0.0283
			Y	0.0840
			Z	0.0965

The performance of the PID and SMC controller was compared by examining the behavior of the vehicle at different wind speeds. Figure 5.9 shows the results of the PID controller at different wind speeds. As can be seen in the graph, the PID controller is getting more and more difficult in trajectory tracking at an average wind speed of 5 m/s. At 10 m/s, it gradually drifts away from orbit.

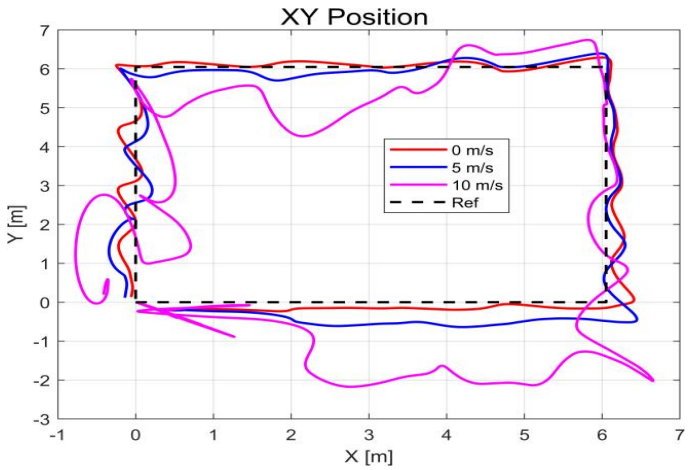


Figure 5.9 PID Controller Results at Different Wind Speeds

Figure 5.10 shows the results of SMC at different wind speeds. Unlike the PID controller, the SMC has successfully completed trajectory tracking. It shows better results against external influences due to high frequency switching of SMC. The large mass of the vehicle and its moments of inertia also increase the resistance of the vehicle against the wind.

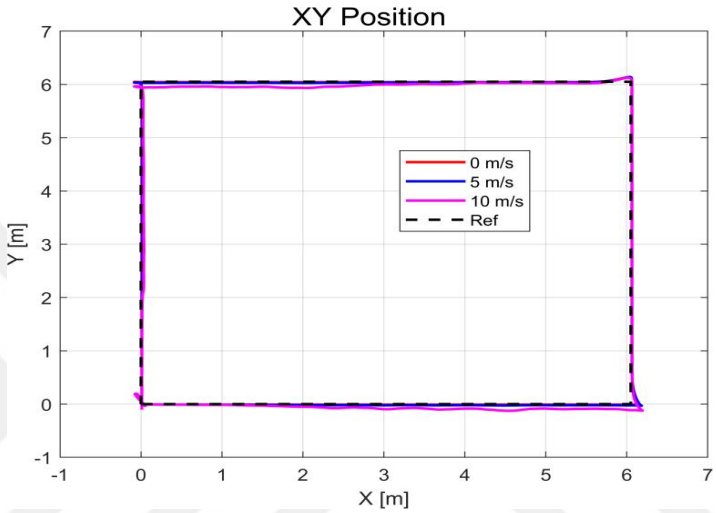


Figure 5.10 SMC Results at Different Wind Speeds

5.4 Control Inputs Resuts

Figure 5.11 shows the control inputs that occur when there is no disturbance. When U_1 (Thrust) is examined, initially SMC increased up to 100 N, while PID increased to 80 N. Afterwards, both controllers gave the same results. The reason why the thrust decreases with time is due to the decrease in mass with time. When U_2 (Roll Moment), U_3 (Pitch Moment) and U_4 (Yaw Moment) are examined, both controllers gave results close to zero when the vehicle was going straight. However, during the turns, the PID controller produced a moment of approximately 0.01 Nm in the roll and pitch axes and 0.5 Nm in the yaw axis. On the other hand, SMC produced a moment of 5 Nm, which was determined as the limit at the turns. In addition, as seen in the graphics, the SMC reaches steady-state much faster than the PID controller.

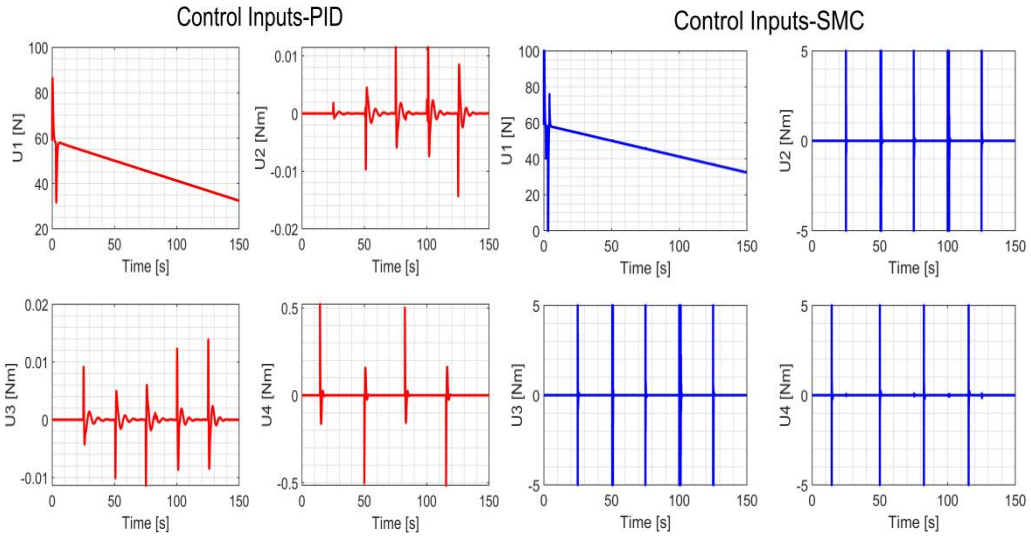


Figure 5.11 Control Efforts without Disturbance

In Figure 5.12, the situation in which sensor noises are given as disturbance is examined. While there is no change in the impulses produced by the controllers, there are serious differences in the moments. Initially, a moment of 0.1 Nm occurred in the roll axis and 0.2 Nm in the pitch axis. Moments of about 0.01 Nm are observed in the turns. However, more oscillating moment is produced due to the disturbance. SMC, on the other hand, generates an oscillating moment with an amplitude of about 1 Nm to counter these disturbances.

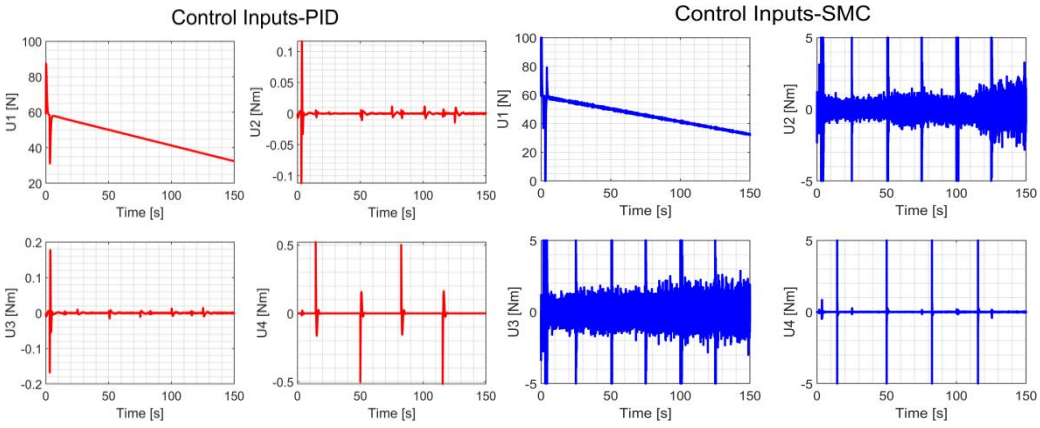


Figure 5.12 Control Efforts with Disturbance

As seen in the graphics, both controllers give close results in the straight movement of the vehicle when there is no disturbance. In the turns, although the

moments produced by the SMC reach their limits, they reach equilibrium in a much shorter time than the PID controller. This is due to the switched structure of the SMC with fast dynamics. In the case of disturbance input, Although SMC performs better against disturbances, it puts a lot more effort into it. Therefore, the energy consumed by the motors will increase.

5.5 Parametric Uncertainty

In order to examine the robustness of the controller against parametric uncertainties, step input responses were examined by adding 40% uncertainty to the mass and moments of inertia. Due to the reduction in the moments of inertia, the vehicle has faster dynamics. For both controllers, no significant change was observed in the angles.

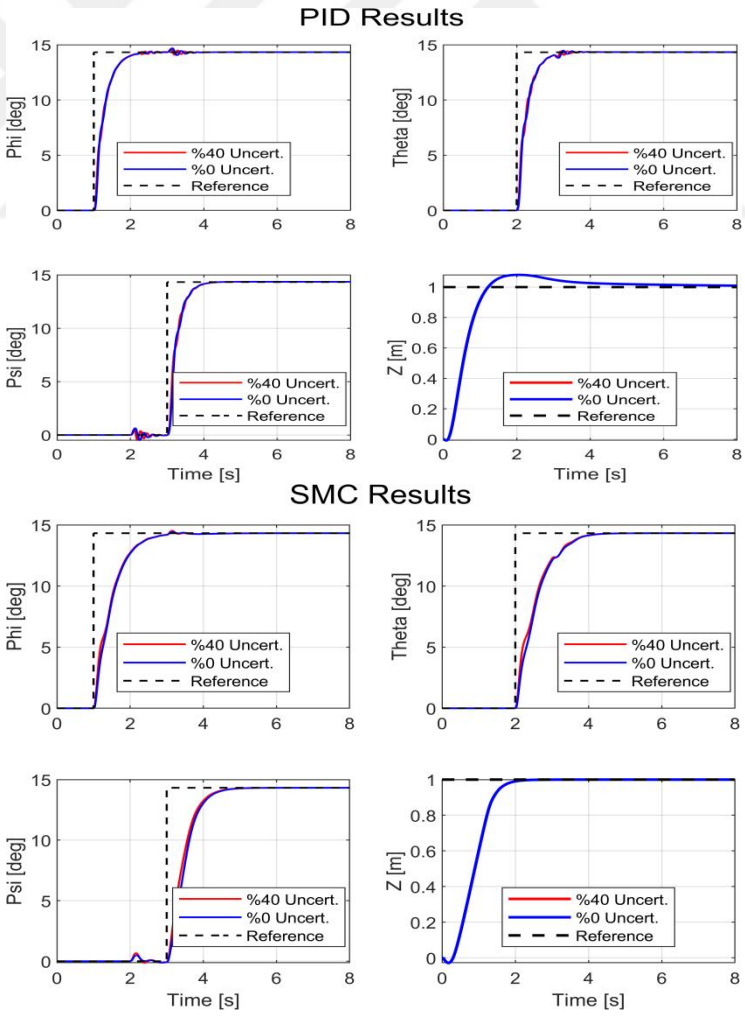


Figure 5.13 PID and SMC Outputs with Parametric Uncertainty

Figure 5.14 shows the control inputs against the uncertainties added to the model. Due to the 40% uncertainty in mass, the amount of thrust (U_1) in both controllers decreased by 40%. On the other hand, if the uncertainty in the PID controller is given in the moments, the amount of oscillation increases as the moments of inertia decrease. Since SMC already has fast dynamics, it gave about the same results against the increased dynamics of the vehicle.

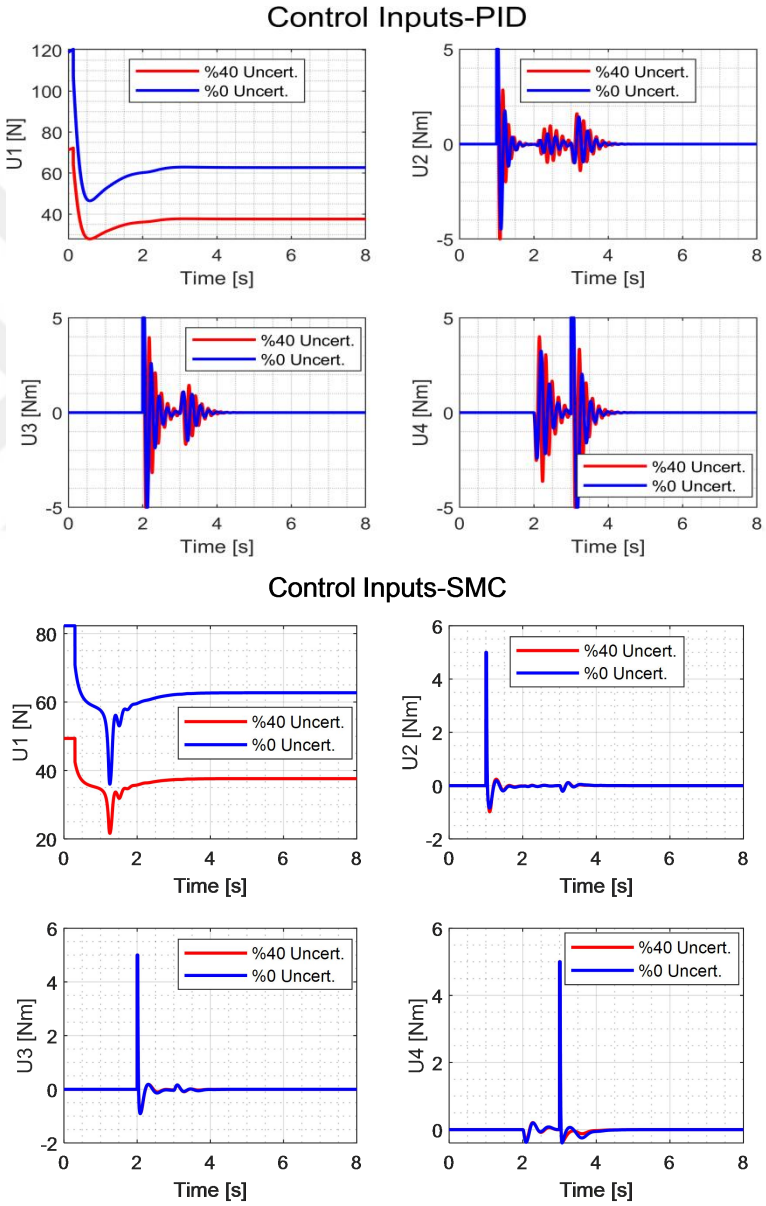


Figure 5.14 Control Efforts with Parametric Uncertainty

CHAPTER 6

CONCLUSION

The aim of this study is to examine the performance of different control algorithms against nonlinear effects. In this sense, PID control, which is a linear control method, and Sliding Mode Control, which is a nonlinear control method, are examined. To make the quadcopter model more realistic, some disturbances added such as motor model, sensor noises, wind effect and mass reduction due to spraying make the system non-linear. Therefore, PID control becomes more vulnerable to disturbances. On the other hand, SMC is more resistant to these disturbances due to its structure.

While choosing the controller, the usage area of the vehicle and the applicability of the controller are important. For flights in outdoor and windy environment, SMC may perform better, but it is more difficult to implement than PID control. On the other hand, PID control may be sufficient for a vehicle to be used indoors. When the graphs without wind effect are examined, both controllers show a good performance. However, considering both wind effect and other disturbances, SMC is more stable than PID control

REFERENCES

Abichandani, P., Lobo, D., Ford, G., Bucci, D., & Kam, M. (2020). Wind measurement and simulation techniques in multi-rotor small unmanned aerial vehicles. *IEEE Access*, 8, 54910-54927.

Agudo, P. U., Pajas, J. A., Pérez-Cabello, F., Redón, J. V., & Lebrón, B. E. (2018). The potential of drones and sensors to enhance detection of archaeological cropmarks: A comparative study between multi-spectral and thermal imagery. *Drones*, 2(3), 29.

Akın, M., Gören, A. & Rachid, A. (2021). Implementation of Sensor Filters and Altitude Estimation of Unmanned Aerial Vehicle using Kalman Filter. *Journal of Mechatronics and Robotics*, 5(1), 8-17. https://doi.org/10.3844/jmrsp.2021.8.17

Alexis, K., Nikolakopoulos, G., & Tzes, A. (2011). Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10), 1195-1207.

Alkamachi, A., & Erçelebi, E. (2017). Modelling and genetic algorithm based-PID control of H-shaped racing quadcopter. *Arabian Journal for Science and Engineering*, 42(7), 2777-2786.

Analia, R., & Song, K. T. (2016, March). Fuzzy+ PID attitude control of a co-axial octocopter. *In 2016 IEEE International Conference on Industrial Technology (ICIT),* 1494-1499.

Analog Devices, (2014). Retrieved June 10, 2022, from https://www.analog.com/en/technical-articles/accelerometer-and-gyroscopes-sensors-operation-sensing-and-applications.html

Argentim, L. M., Rezende, W. C., Santos, P. E., & Aguiar, R. A. (2013, May). PID, LQR and LQR-PID on a quadcopter platform. *In 2013 International Conference on Informatics, Electronics and Vision (ICIEV),* 1-6.

Balasubramaniam, A., Reshma, S., & Janardanan, A. (2018). Enhancing the Proficiency of the Drone with an Application of Ocean Farming. *Procedia computer science*, 133, 725-732.

Bouabdallah, S., Noth, A., & Siegwart, R. (2004, September). PID vs LQ control techniques applied to an indoor micro quadrotor. *In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (3), 2451-2456

Božek, P., Al Akkad M, A., Blištan, P., & Ibrahim N, I. (2017). Navigation control and stability investigation of a mobile robot based on a hexacopter equipped with an integrated manipulator. *International Journal of Advanced Robotic Systems*, 14(6), 1729881417738103.

Croptracker, (n.d). Retrieved November 14, 2022, from https://www.croptracker.com/blog/drone-technology-in-agriculture.html

Cömert, C. (2016). *Dört rotorlu bir insansız hava aracı için PID ve kayan kipli kontrolcü tasarımı ve kıyaslanması.* TOBB ETÜ Fen Bilimleri Enstitüsü, Ankara.

Delica, V., Orquia, A., Piquero, J., Ilao, J. P., Reynaldo, E., Roque, M. A., & Jayakody, H. (2019). A new sliding mode controller implementation on an autonomous quadcopter system. *International Journal of Automation and Smart Technology*, 9(2), 53.

Demiryürek, A. (2018). *Bir Dörtpervanelinin Modellenmesi ve Denetimi*. Hacettepe University, Ankara.

Dixit, A., Misra, A., & Talole, S. E. (2020, February). Model Predictive Control based Collision Avoidance Controller for Octocopter. *In 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN),* 630-635.

Dixit, A., Misra, A., & Talole, S. E. (2020, June). UDE based Sliding Mode Control for Octocopter. *In 2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA)*, 1-6.

Eskandarpour, A., & Sharf, I. (2020). A constrained error-based MPC for path following of quadrotor with stability analysis. *Nonlinear Dynamics,* 99(2), 899-918.

Frankelius, P., Norrman, C., & Johansen, K. (2019). Agricultural innovation and the role of institutions: lessons from the game of drones. *Journal of Agricultural and Environmental Ethics*, 32(5), 681-707.

Hakim, T. M. I., & Arifianto, O. (2018, April). Implementation of Dryden continuous turbulence model into simulink for LSA-02 flight test simulation. *In Journal of Physics: Conference Series,*1005, 012017).

Herrera, M., Chamorro, W., Gómez, A. P., & Camacho, O. (2015, July). Sliding mode control: An approach to control a quadrotor. *In 2015 Asia-Pacific conference on computer aided system engineering,* 314-319.

Mathworks. (2022). *Aerospace Blockset™ User's Guide,* Retrieved May 10, 2022, from https://www.mathworks.com/help/pdf_doc/aeroblks/aeroblks.pdf

Khelfi, M. F., & Kacimi, A. (2012, May). Robust control with sliding mode for a quadrotor unmanned aerial vehicle. *In 2012 IEEE International Symposium on Industrial Electronics,* 886-892.

Krishna, K. R. (2018). *Agricultural drones: a peaceful pursuit* (1st ed.). Apple Academic Press.

Liu, J., & Wang, X. (2012). *Advanced sliding mode control for mechanical systems.* Springer.

Madani, T., & Benallegue, A. (2006, October). Backstepping control for a quadrotor helicopter. *In 2006 IEEE/RSJ International Conference on Intelligent Robots and System*s, 3255-3260.

Madoyan, A. (2009). *Design and Comparison of Mixed Hinf/H2 Controller for AMB System*. Lappeenranta University, Finland.

Messac, A. (2015). *Optimization in practice with MATLAB for engineering students and professionals* (1st ed.). Cambridge University Press.

Nonami, K., Kendoul, F., Suzuki, S., Wang, W., & Nakazawa, D. (2010). *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles* (1st ed.). Springer Science & Business Media.

Osmić, N., Kurić, M., & Petrović, I. (2016, October). Detailed octorotor modeling and PD control. *In 2016 IEEE international conference on systems, man, and cybernetics (smc)*, 002182-002189.

Rahmat, M. F., Eltayeb, A., & Basri, M. A. M. (2020). Adaptive feedback linearization controller for stabilization of quadrotor UAV. *International Journal of Integrated Engineering*, 12(4), 1-17.

Rinaldi, F., Gargioli, A., & Quagliotti, F. (2014). PID and LQ regulation of a multirotor attitude: Mathematical modelling, simulations and experimental results. *Journal of Intelligent & Robotic Systems*, 73(1), 33-50.

Runcharoon, K., & Srichatrapimuk, V. (2013, May). Sliding mode control of quadrotor. *In 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE),* 552-557.

Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control* (1st ed.). Prentice Hall.

van der Merwe, D., Burchfield, D., Witt, T., Price, K., & Sharda, A. (2020). Drones in agriculture. 10.1016/bs.agron.2020.03.001.

Utkin, V. (1977). Variable structure systems with sliding modes. *IEEE Transactions on Automatic control*, 22(2), 212-222

Xu, R., & Ozguner, U. (2006, December). Sliding mode control of a quadrotor helicopter. *45th IEEE Conference on Decision and Control,* 4957-4962.

Yamaha Motors, (n.d). Retrieved November 12, 2022, from https://global.yamaha-motor.com/stories/history/stories/0028.html

Zhang, D., Qi, H., Wu, X., Xie, Y., & Xu, J. (2014). The quadrotor dynamic modeling and indoor target tracking control method. *Mathematical problems in engineering*, 2014.

Zhang, G., & Taha, H. (2017). Adaptive back-stepping control applied on octocopter under recoil disturbance. *Journal of Engineering Science and Military Technologies,* 1(1), 12-2

APPENDICES

APPENDIX 1: MATLAB and Arduino Codes

Model Parameters Code

```
clc; clear; close all

%% Controller Parameters
% PID
kp_phi=7.266; ki_phi=0.0; kd_phi=0.9841;
kp_theta=8.0253; ki_theta=0; kd_theta=1.118;
kp_psi=9.9; ki_psi=0; kd_psi=1.4465;

kp_x=4; ki_x=0.1; kd_x=3;
kp_y=4; ki_y=0.1; kd_y=3;
kp_z=10; ki_z=2; kd_z=6;

kp_p=5; ki_p=0.5; kd_p=0;
kp_q=5; ki_q=0.5; kd_q=0;
kp_r=5; ki_r=0.5; kd_r=0;

% SMC
lambda_phi = 3; k_phi =  5; delta_phi = 0.5;
lambda_theta = 3; k_theta =  5; delta_theta = 0.8;
lambda_psi = 4; k_psi =  4; delta_psi = 0.5;

lambda_x = 4; k_x = 4; delta_x = 0.5;
lambda_y = 4; k_y = 4; delta_y = 0.5;
lambda_z = 4; k_z = 4; delta_z = 0.1;

un1 = 0.5;
un2 = 0.7;
un3 = 0.6;
un4 = 0.8;

%% UAV parameters
g = 9.81;          % Gravitational acceleration
m = 6*1;             % Mass of the UAV in Kg
Jxx = 0.182*1;       % Moment of inertia X
Jyy = 0.188*1;       % Moment of inertia Y
Jzz = 0.355*1;       % Moment of inertia Z
Jr  =  2*10^(-5);       % Rotational moment of inertia around the
propeller axis
l = 0.50;         % Distance to the center of the Quadrotor
kt = 1.7*10^(-5);   % Thrust factor
kq = 1.56*10^(-6);  % Drag factor
windspeed = 10;     % Windspeed for simulations
accel_z_min = -1.5; % Max and min values of Z acceleration(m/s^2)
accel_z_max = 2.5;
h = 10;    % For wind model
Va = 5;    % For wind model
%% Motor parameters
mot_num = 100;
mot_den = 100;
```

```
tau = 0.015;

%% U limits
U1_max = 2*m*g; U1_min = 0;
U2_max = 5; U2_min = -5;
U3_max = 5; U3_min = -5;
U4_max = 5; U4_min = -5;
```

Result Plotting Code

```
clc; clear; close all

run('model_parameters.m');

smc = sim('smc_step_position.slx');
pid = sim('pid_step_position.slx');

% Root Mean Squared Error
RMSE_smc = sqrt(mean((smc.Desired_Pos(:,1)-smc.Position(:,1)).^2));
RMSE_pid = sqrt(mean((pid.Desired_Pos(:,1)-pid.Position(:,1)).^2));

% %% Angle Plot
figure(1)
subplot(3,1,1)
plot(pid.tout, pid.Euler(:,1), 'r', 'LineWidth', 1);
hold on
plot(smc.tout, smc.Euler(:,1), 'b', 'LineWidth', 1);
grid on; grid minor;
ylabel('Phi [deg]');
legend('PID','SMC')

subplot(3,1,2)
plot(pid.tout, pid.Euler(:,2), 'r', 'LineWidth', 1);
hold on
plot(smc.tout, smc.Euler(:,2), 'b', 'LineWidth', 1);
grid on; grid minor;
ylabel('Theta [deg]');
legend('PID','SMC')

subplot(3,1,3)
plot(pid.tout, pid.Euler(:,3), 'r', 'LineWidth', 1);
hold on
plot(smc.tout, smc.Euler(:,3), 'b', 'LineWidth', 1);
grid on; grid minor;
xlabel('Time [s]'); ylabel('Psi [deg]');
legend('PID','SMC')
exportgraphics(gcf,'angle.png','Resolution',300)

%% Position Plot
figure(2)
plot(pid.Position(:,1), pid.Position(:,2), 'r', 'LineWidth', 1.5);
hold on
plot(smc.Position(:,1), smc.Position(:,2), 'b', 'LineWidth', 1.5);
hold on
plot(pid.Desired_Pos(:,1), pid.Desired_Pos(:,2), 'k--', 'LineWidth', 1.5);
grid on; grid minor; title('Z');
```

```
xlabel('X [m]'); ylabel('Y [m]');
legend('PID', 'SMC','Ref')
exportgraphics(gcf,'XY.png','Resolution',300)

figure(3)
plot3(pid.Position(:,1), pid.Position(:,2), pid.Position(:,3), 'r',
'LineWidth', 1.5);
hold on
plot3(smc.Position(:,1), smc.Position(:,2), smc.Position(:,3), 'b',
'LineWidth', 1.5);
hold on
plot3(pid.Desired_Pos(:,1), pid.Desired_Pos(:,2), pid.Desired_Pos(:,3),
'k', 'LineWidth', 1.5);
grid on; grid minor; title('Z');
legend('PID', 'SMC','Ref')
exportgraphics(gcf,'traj.png','Resolution',300)
```

Thrust Mechanism Code

```
#include "HX711.h"
#include <Servo.h>
// ***** RPM *****
const int ir_pin = 2;  // digital pin input for reading the ir
volatile double time = 0;
volatile double d_time = 0;
volatile double rpm;
// ***** Thrust *****
HX711 scale(8, 7);
float calibration_factor = 420; // this calibration factor is adjusted
according to my load cell
double units;
float ounces;
// ***** Motor *****
Servo esc;
int pot_pin = A5;
int pot_value;

void setup() {
  esc.attach(11,1000,2000);
  esc.write(0);
  scale.set_scale();
  scale.tare();  //Reset the scale to 0
  long zero_factor = scale.read_average();
```

```
  pinMode(ir_pin,INPUT);
  pinMode(pot_pin,INPUT);

  attachInterrupt(0, calculate, FALLING);
  Serial.begin(9600);  // Begin serial port
}

void loop()
{
  pot_value = analogRead(pot_pin);
  int ui_value = map(pot_value, 0, 1023, 0, 100);
  pot_value = map(pot_value, 0, 1023, 0, 180);
  esc.write(pot_value);
  scale.set_scale(calibration_factor);//Calibration
  units = scale.get_units(), 10;
  if (units < 0)
  {
    units = 0.00;
  }
  ounces = units * 0.035274;

  int ir_val = digitalRead(ir_pin);

  Serial.print(rpm);
  Serial.print("*");
  Serial.print(units);
  Serial.print("*");
  Serial.println(ui_value);
  delay(100);
}

void calculate(){
  d_time = micros()-time;
  time = time + d_time;
  rpm = 1/d_time*1000000*60;
}
```

APPENDIX 2: Simulink Model

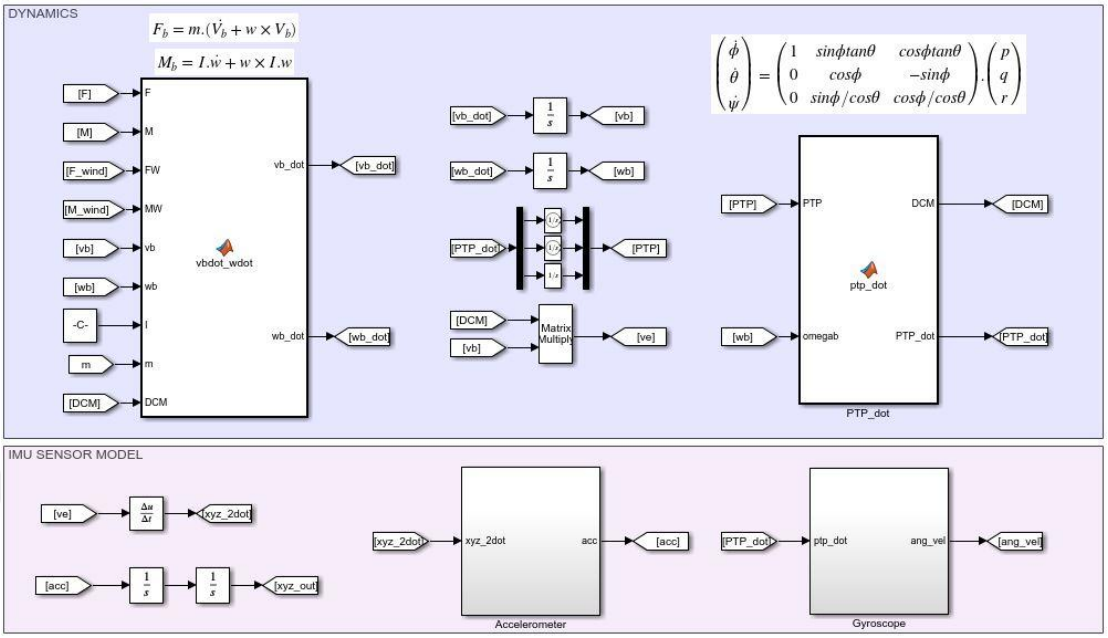


Figure A2.1 Dynamics and Sensor Model

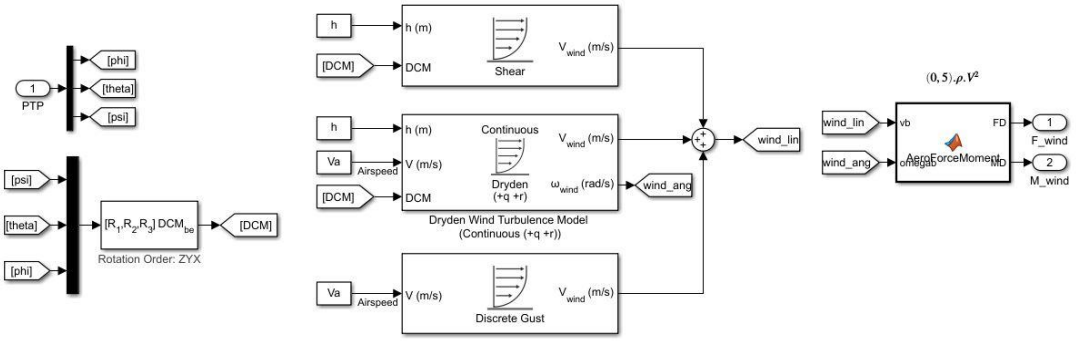


Figure A2.2 Wind Model

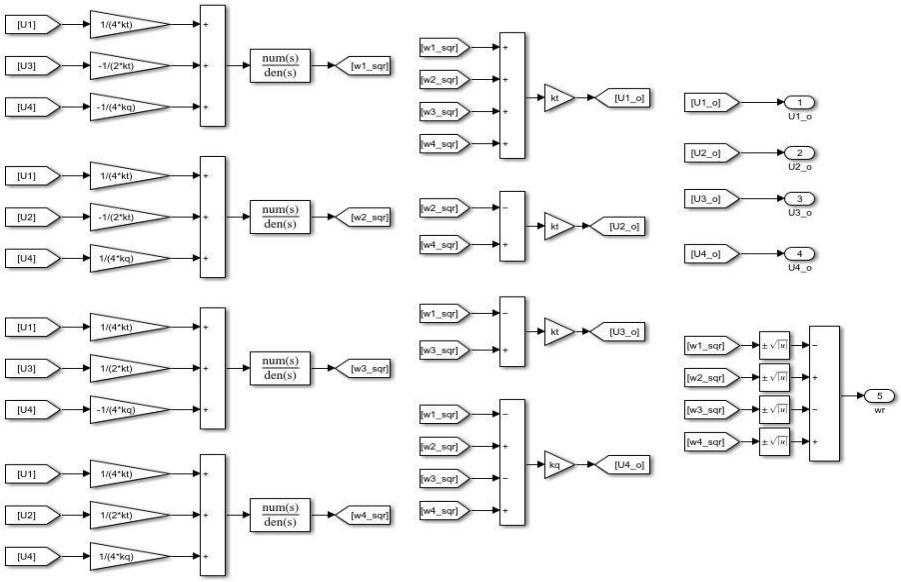


Figure A2.3 Rotor Model

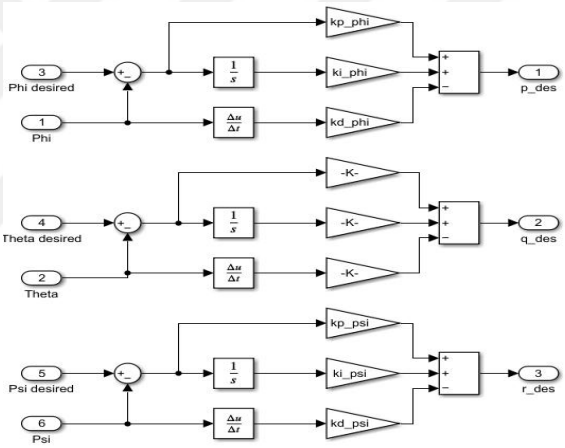


Figure A2.4 Angular Velocity Controller (PID)

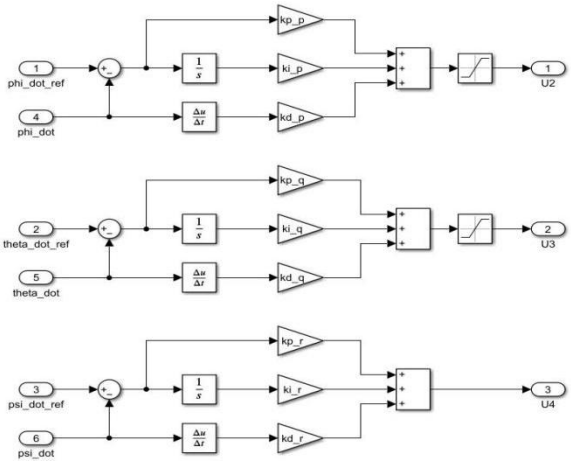


Figure A2.5 Angle Controller (PID)

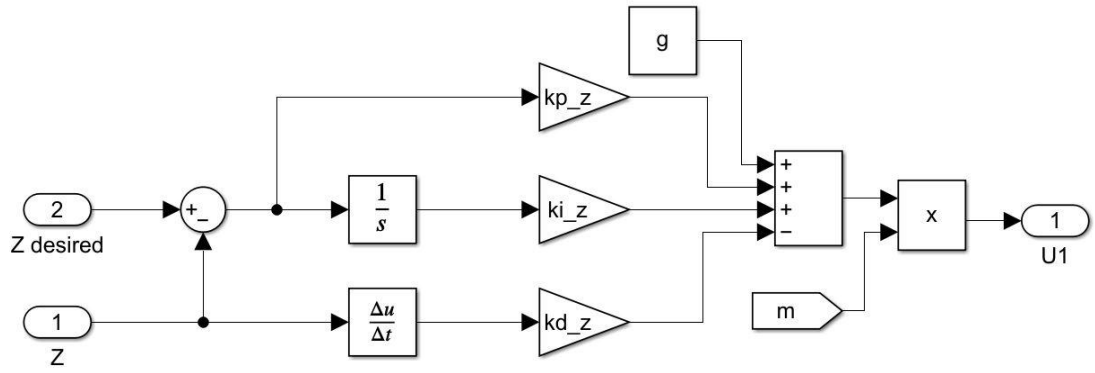


Figure A2.6 Altitude Controller (PID)

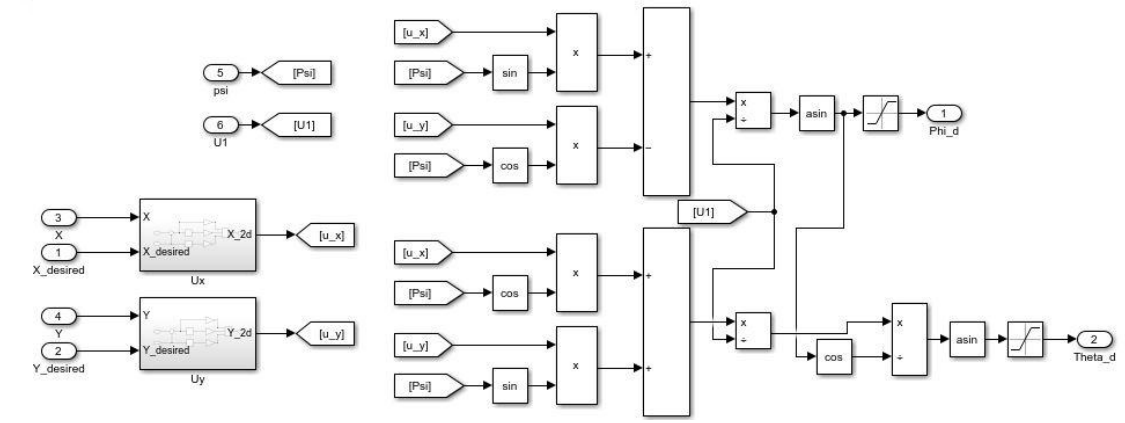


Figure A2.7 Position Controller (PID)

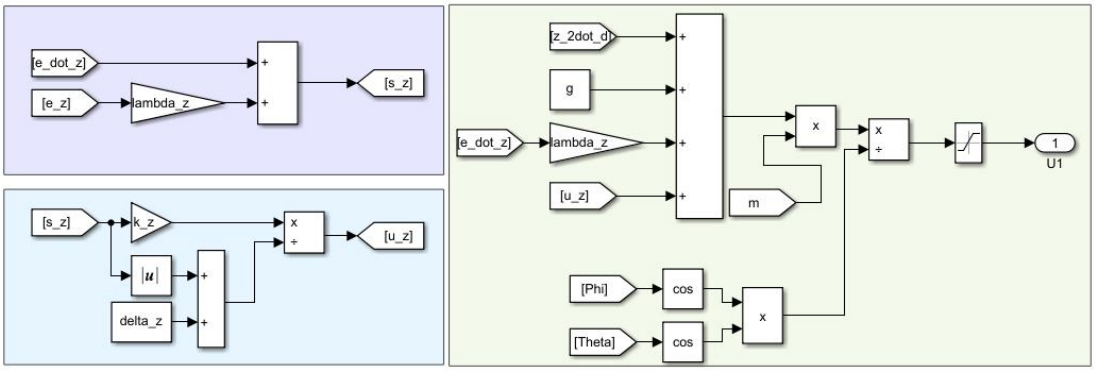


Figure A2.8 Altitude Controller (SMC)

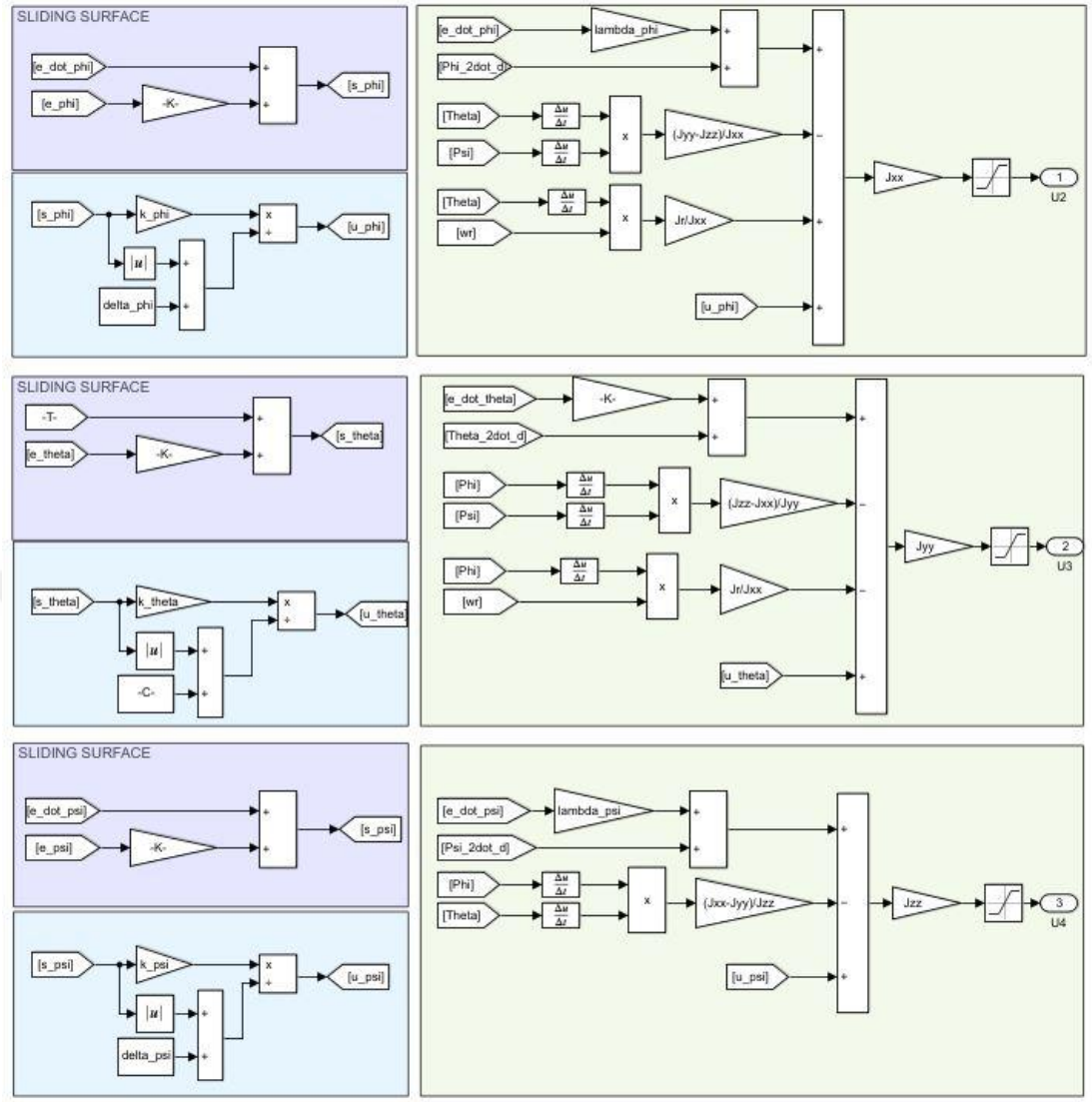


Figure A2.9 Angle Controller (SMC)

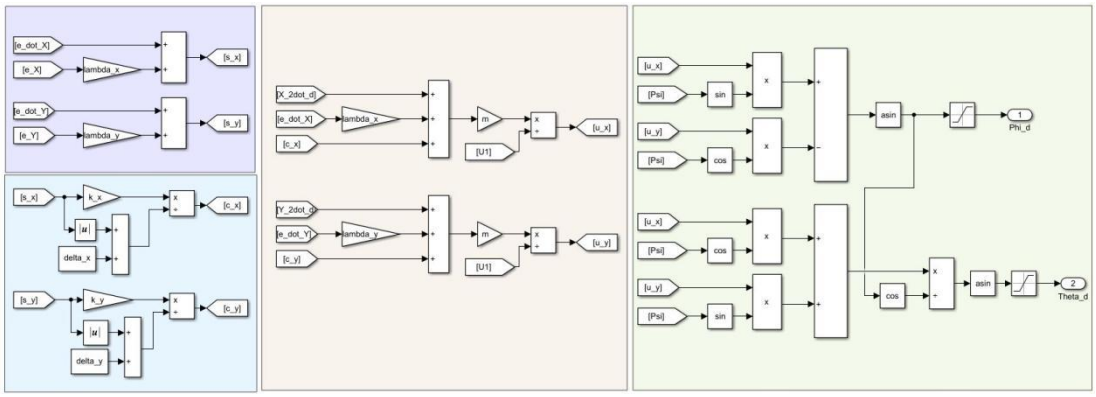


Figure A2.10 Position Controller (SMC)

APPENDIX 3: Component Selection

Propellers : Propellers are characterized by three main parameters: Diameter(D), Pitch(θ) and number of blades. Diameters indicate the size of the propeller. Larger diameter means more thrust but requires more torque from the motor. Pitch is a measurement of how far that a propeller will move through the air for every single rotation. As the number of blades increases, more precise control is provided, but it provides lower performance and less speed. The thrust provided by the propeller is as shown in Equation A3.1.

$$T = C_T \rho n^2 D^4 \tag{A3.1}$$

Motors: The thrust-to-weight ratio is important when choosing the motor. The UAV must be lifted off with half throttle. For example, for a 5 kilos vehicle, 10 kilos of thrust should be provided.

Consider the SunnySky X4108S KV690 Brushless DC Motor as an example. “41” means diameter of the stator in mm and “08” means height of the stator in mm. The KV value is the RPM of the motor that will spin at when 1 Volt is applied. In other words how powerful a motor is. The propeller with high KV turns fast and is generally used in small propellers. Low KV value provides more torque and is generally used in large propellers.

$$w_0 = KV.V \tag{A3.2}$$

RPM value for a 690KV motor according to equation A3.2 :

690 KV. (14,8V)=10212 RPM

As the take-off weight increases, the power required for thrust will increase.

$$P = IV \tag{A3.3}$$

Prop (inch)	Volts (V)	Amps (A)	Thrust(g)	Watts (W)	Efficiency (g/W)
APC1147	14.8	1	170	14.8	11.48648649
		2	280	29.6	9.459459459
		3	390	44.4	8.783783784
		4	490	59.2	8.277027027
		5	580	74	7.837837838
		6	660	88.8	7.432432432
		7	730	103.6	7.046332046
		8	790	118.4	6.672297297
		9	850	133.2	6.381381381
		10	940	148	6.351351351
		12	1050	177.6	5.912162162
		15	1230	222	5.540540541
		18	1380	266.4	5.18018018
		21.9	1620	324.12	4.998148834
APC1238	14.8	1	180	14.8	12.16216216
		2	290	29.6	9.797297297
		3	400	44.4	9.009009009
		4	500	59.2	8.445945946
		5	600	74	8.108108108
		6	690	88.8	7.77027027
		7	750	103.6	7.239382239
		8	820	118.4	6.925675676
		9	900	133.2	6.756756757
		10	970	148	6.554054054
		12	1100	177.6	6.193693694
		15	1270	222	5.720720721
		18	1430	266.4	5.367867868
		21	1560	310.8	5.019305019
		27.4	1870	405.52	4.611363188

Figure A3.1 Motor Specs

Electronic Speed Controller(ESC): When selecting ESC, attention should be paid to the current drawn by the motor. As shown in Figure A1.1, the maximum current will be 27.4 A for 12x38 propellers. ESCs with a value of 30 or 40 amps will be sufficient for the UAV to be produced. Also, the "S" value of ESC must match the battery. 40A 2S-4s ESC will be used in this study. In order to have a higher tolerance and prevent possible accidents, the 40A ESC was preferred.

Battery: Batteries are characterized by their capacity in mAh and C-rating. mAh(milliamp hours) value means the capacity of the battery. This indicates how much the battery can be drained to drain within an hour. S value means cell counts of the battery and C value means how many amps the battery can handle.

Consider 14,8V 5000mAh 45C - 4s Parallel 2 Lipo Batteries as an example.

5000 mAh * 2 = 10000 mAh

10000 mAh / 1000 = 10 Ah

10 Ah * 45 C = 450 Amper

The motors are rated at 27.4 Amps. For 8 motors 18*8=219.2 Amps < 450Amps. So the batteries can safely work.

Table A3.1 Voltage Chart

Voltage	**1S**	**2S**	**3S**	**4S**	**5S**	**6S**
100%	4.2V	8.4V	12.6V	16.8V	21V	25.2V
50%	3.85V	7.67V	11.5V	15.34V	19.18V	23.01V
20%	3.7V	7.45V	11.18V	14.91V	18.63V	22.36V

Thrust and Flight Time Calculation

Calculations were made considering that the planned weight would be approximately 6 kilograms.

In half throttle: 690KV * 16.8V / 2 = 5796 RPM

With this RPM value, the c_T value will be selected from the catalogue provided by the manufacturer. For 12x3.8 propeller c_T=0.1140. Diameter of the propeller is 12 inch(0.3048 meters)

n=690*(16.8/2) / 60 = 96.6 RPS

$\rho = 1.225\ kg/m^3$

From Equation A3.1,

$$T = C_T \rho n^2 D^4 = 0.1140 \times 1.225 \times 96.6^2 \times 0.3048^4 = 11.24\ N$$

$$T_{required} = \frac{mg}{8} = 6 * 9.81/8 = 7.35\ N$$

According to these calculations, the vehicle will start to take off in half throttle. From motor specs 1 Amps produce 180 g.

6000g / 8 = 750 gr per motor

8 motors*750/180 = 33 A

(10000mAh/1000) * (80%) * 60 / 33 = 14.5 minutes

APPENDIX 4: APM 2.6 Spesifications and Setup

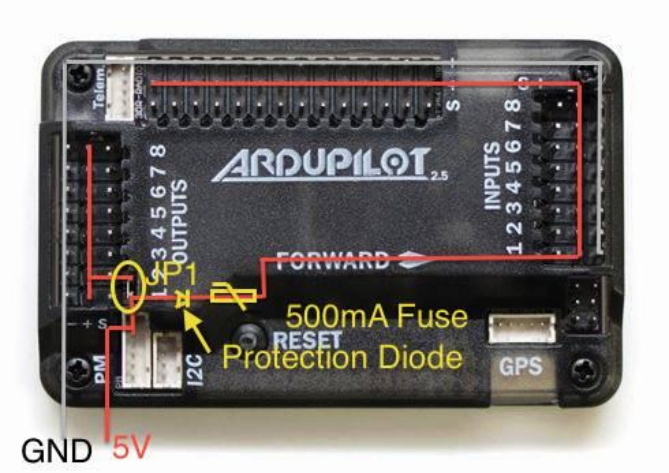


Figure A4.1 APM 2.6 Flight Controller

- Includes 3-axis gyro, accelerometer and magnetometer, along with a high performance barometer
- Onboard 4 MegaByte Dataflash chip for automatic datalogging
- Optional off-board GPS, uBlox LEA-6H module with Compass.
- Invensense's 6 DoF Accelerometer/Gyro MPU-6000.
- Barometric pressure sensor upgraded to MS5611-01BA03.
- Dimension: 70.5x45x13.5mm
- Weight: 31g

Flight Controller Setup:

1. Download Mission Planner

2. Plug the USB to the laptop.

3. Click Install Firmware Legacy and select the vehicle's model and downloading the firmware will be started.

4. Select COM Port and BaudRate (115200) in top right, then click the Connect button.

5. After the connection click Mandatory Hardware in Setup section. Then follow the steps below.

5.1. Frame Type: Choose the frame type of the vehicle.



Figure A4.2 Quadcopter Configurations

5.2. Compass: Check ”Use this compass and Externally Mounted”, then click Live Calibration. After click OK rotate the autopilot around all axises in a circular motion.

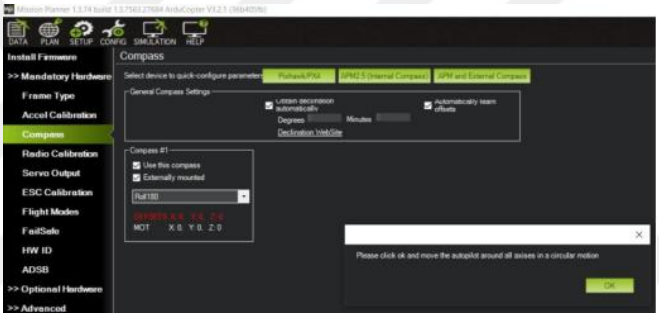


Figure A4.3: Compass

5.3. Accel Calibration: Pose the autopilot according to instructions after click Calibrate Accel.



Figure A4.4: Accel Calibration

5.4. Radio Calibration: Click on Calibrate Radio and check all joysticks and swithes whether they are in working condition or not. Then click on "Click when Done".

5.5. Flight Modes: Flight modes are adjusted according to the flight.

5.6. FailSafe: Select Enabled always RTL in Radio subsection.

6. After these settings, ESCs must be calibrated in order to stabile flight. Be careful while calibrating and check there are no propellers.

All at once calibration:

6.1. Turn on your transmitter and put the throttle stick at maximum.

6.2. Connect the Lipo battery. The autopilot's red, blue and yellow LEDs will light up in a cyclical pattern. This means the it's ready to go into ESC calibration mode the next time you plug it in.

6.3. With the transmitter throttle stick still high, disconnect and reconnect the battery.

6.4. The autopilot is now in ESC calibration mode. Wait for your ESCs to emit the musical tone, the regular number of beeps indicating your battery's cell count (i.e. 3 for 3S, 4 for 4S) and then an additional two beeps to indicate that the maximum throttle has been captured.

6.5. The ESCs should then emit a long tone indicating that the minimum throttle has been captured and the calibration is complete.

6.6. Set the throttle to minimum and disconnect the battery to exit ESC- calibration mode.

APPENDIX 5: Octocopter Control Inputs

Thrust

Altitude of the UAV is provided using thrust generated. Ideally, propellers rotate with same speed and lifts off the UAV through Z axis which the positive direction is opposite to the direction of gravitational acceleration.Total thrust can ve calculated using the equation follows;

$$U_1 = \sum_{i=1}^{8} F_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 + \Omega_5^2 + \Omega_6^2 + \Omega_7^2 + \Omega_8^2) \qquad \text{(A5.1)}$$

Roll Moment

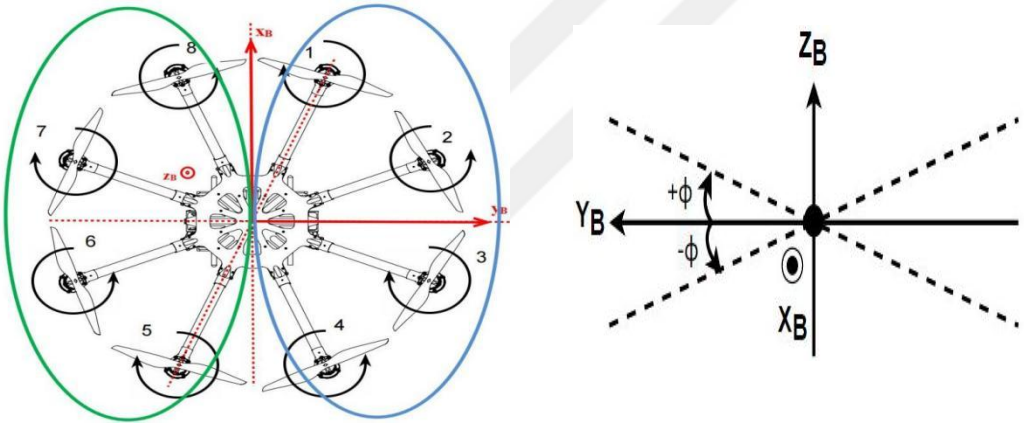


Figure A5.1 Roll Moment Configuration

Change in roll angle of the UAV is the change in angle wrt x axis which is the result of change of velocities of propellers. As in Figure A5.1, roll angle changes according to the motion in green and blue regions. It is assumed that, if the blue region is higher than green region then the roll angle is positive whereas if it is lower then the roll angle is negative. The roll angle equation is given as follows,

$$U_2 = bl[(\sin\tfrac{\pi}{8}.\Omega_5^2 + \sin\tfrac{\pi}{8}.\Omega_8^2 + \cos\tfrac{\pi}{8}.\Omega_6^2 + \cos\tfrac{\pi}{8}.\Omega_7^2) - (\sin\tfrac{\pi}{8}.\Omega_1^2 + \sin\tfrac{\pi}{8}.\Omega_4^2 + \cos\tfrac{\pi}{8}.\Omega_2^2 + \cos\tfrac{\pi}{8}.\Omega_3^2)] \qquad \text{(A5.2)}$$

Pitch Moment

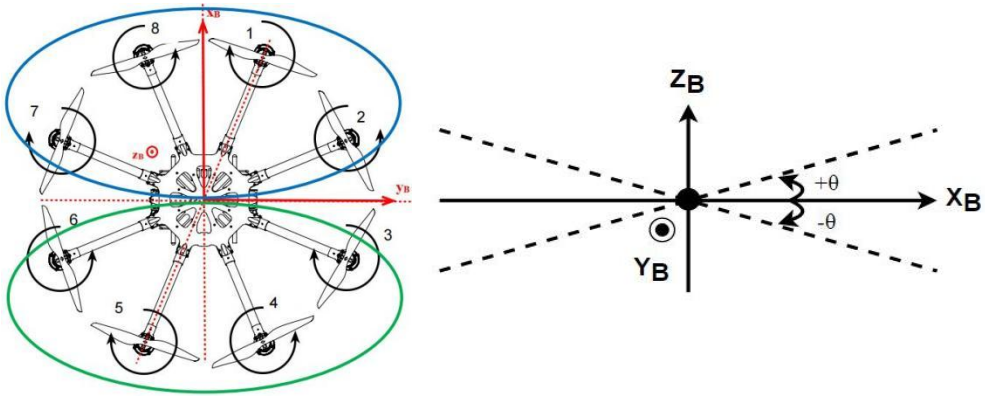


Figure A5.2 Pitch Moment Configuration

The change in the pitch angle occurs due to the differences in the rotation speed of the propellers, like in the roll angle. The roll angle is relative to the x-axis whereas the pitch angle is relative to the vehicle's y-axis. As seen in Figure A5.2, the propellers are grouped as being on front or back region. If the part of the vehicle marked in blue is higher than the green part, the pitch angle is positive and negative in the lower part. Pitch control input equation is as follows.

$$U_3 = bl[(\sin\frac{\pi}{8}.\Omega_2^2 + \sin\frac{\pi}{8}.\Omega_7^2 + \cos\frac{\pi}{8}.\Omega_1^2 + \cos\frac{\pi}{8}.\Omega_8^2) - (\sin\frac{\pi}{8}.\Omega_3^2 + \sin\frac{\pi}{8}.\Omega_6^2 + \cos\frac{\pi}{8}.\Omega_4^2 + \cos\frac{\pi}{8}.\Omega_5^2)] \quad \text{(A5.3)}$$

Yaw Moment

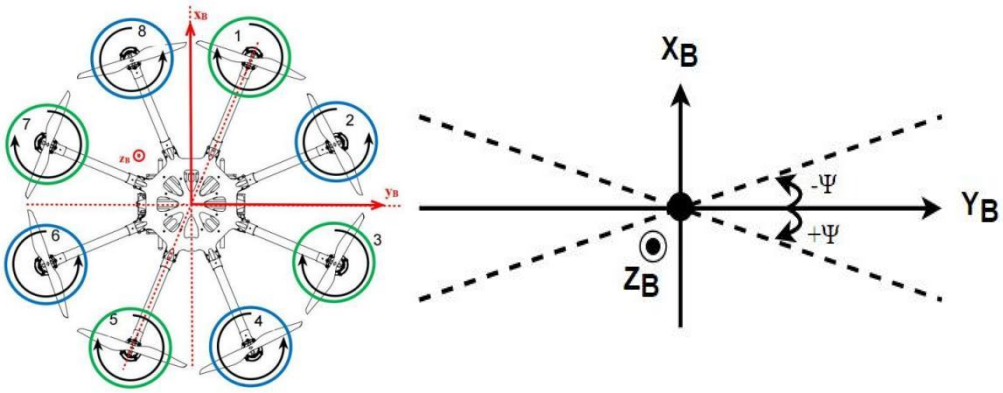


Figure A5.3 Yaw Moment Configuration

The yaw angle represents the rotation of the tool with respect to the z-axis. As a result of the rotation of the propellers, a moment occurs on the vehicle opposite to the rotation direction of the propellers. Propellers are grouped according to the direction of rotation to balance the moment. Figure A5.3 shows the grouping of propellers according to their rotation directions. If the vehicle is turning clockwise, the yaw angle is positive, if it is moving counterclockwise, it is negative. Yaw control input equation is as follows.

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 - \Omega_5^2 + \Omega_6^2 - \Omega_7^2 + \Omega_8^2) \qquad \text{(A5.4)}$$

If the equations can be represented in matrix forms;

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b & b & b & b & b \\ -\sin\frac{\pi}{8}bl & -\cos\frac{\pi}{8}bl & -\cos\frac{\pi}{8}bl & -\sin\frac{\pi}{8}bl & \sin\frac{\pi}{8}bl & \cos\frac{\pi}{8}bl & \cos\frac{\pi}{8}bl & \sin\frac{\pi}{8}bl \\ \cos\frac{\pi}{8}bl & \sin\frac{\pi}{8}bl & -\sin\frac{\pi}{8}bl & -\cos\frac{\pi}{8}bl & -\cos\frac{\pi}{8}bl & -\sin\frac{\pi}{8}bl & \sin\frac{\pi}{8}bl & \cos\frac{\pi}{8}bl \\ -d & d & -d & d & -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \\ \Omega_5^2 \\ \Omega_6^2 \\ \Omega_7^2 \\ \Omega_8^2 \end{bmatrix} \qquad \text{(A5.5)}$$

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \\ \Omega_5^2 \\ \Omega_6^2 \\ \Omega_7^2 \\ \Omega_8^2 \end{bmatrix} = \begin{bmatrix} b & b & b & b & b & b & b & b \\ 0 & -\frac{\sqrt{2}}{2}bl & -bl & -\frac{\sqrt{2}}{2}bl & 0 & \frac{\sqrt{2}}{2}bl & bl & \frac{\sqrt{2}}{2}bl \\ -bl & -\frac{\sqrt{2}}{2}bl & 0 & \frac{\sqrt{2}}{2}bl & bl & \frac{\sqrt{2}}{2}bl & 0 & -\frac{\sqrt{2}}{2}bl \\ -d & d & -d & d & -d & d & -d & d \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \qquad \text{(A5.6)}$$