DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ALTITUDE CONTROL OF AN OCTOCOPTER UAV USING ARTIFICIAL NEURAL NETWORKS

by
Şahin Ekmel KARAKAYA

February, 2022
İZMİR

ALTITUDE CONTROL OF AN OCTOCOPTER UAV USING ARTIFICIAL NEURAL NETWORKS

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

Şahin Ekmel KARAKAYA

February, 2022
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"ALTITUDE CONTROL OF AN OCTOCOPTER UAV USING ARTIFICIAL NEURAL NETWORKS"** completed by **ŞAHİN EKMEL KARAKAYA** 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

Assoc. Prof. Dr. Süleyman Murat BAĞDATLI
(Jury Member)

Prof.Dr. Hasan ÖZTÜRK
(Jury Member)

Prof. Dr. Okan FISTIKOĞLU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First of all, I would like to thank Assoc. Prof. Dr. Aytaç GÖREN whose expertise is invaluable in both technical and theoretical aspects. I sincerely thank him for his motivation, guidance, critical evaluations, and endless support from the beginning to the end of the project. I will be forever grateful to him for giving me a chance to improve myself on my own.

I would like to express my deepest appreciation to my colleagues Onur ÇELİK and Gökmen AŞÇIOĞLU, for their critical suggestions, kindness, and sharing experiences. They helped me a lot during the thesis writing process that I will never forget.

I also would like to express my deepest appreciation to my friends Serdar SAYIN and Rahime ALSANĞUR, for their motivational support in my desperate times.

I would like to thank my old manager Osman KORKUT, for his endless support on fixing electronic speed controllers and the montage process of the octocopter.

Especially I would like to thank my little architect Elif Ecmel KARAKAYA for drawing and reworking every piece of graphic, and figure. If you are reading this, you have to prepare yourself for upcoming projects and improvements. This will never end for you because of the immutable rule of brother and sister.

I would like to thank my family members for their endless support in every part of my life. They always stood beside me from my surgeries to the happiest moments of my life. I don't know how may I return all these favors to every one of you even the words are not enough. I dedicate this work to you.

Şahin Ekmel KARAKAYA

ALTITUDE CONTROL OF AN OCTOCOPTER UAV USING ARTIFICIAL NEURAL NETWORKS

ABSTRACT

The use of unmanned aerial vehicles (UAV) has become widespread in recent years. UAVs pioneers scientific research and variety of use in space technologies, also it has a wide range of use in sectors such as military, commercial, and security. The control of flying vehicles has always been difficult due to disruptive inputs. This study aims to design and control the attitude and altitude of an autonomous octocopter with the NARX NN algorithm to be an alternative to commonly used control algorithms on UAVs with continuous learning capability. Roll, pitch, yaw, and altitude control of eight rotor multi-copter also known as octocopter have been designed. The non-linear mathematical model of the octocopter has been implemented on Matlab Simulink with the help of the Newton-Euler method. A rigid body model has been designed on Solidworks to identify mass properties and inertial moments of the proposed octocopter design. Classical proportional-integrator-derivative (PID) algorithms and dynamic time series non-linear autoregressive exogenous input (NARX) neural networks (NN) have been used for the altitude and attitude control of the feedback loop system. The PID controller gains have been calculated with the help of the Ziegler-Nichols method. The NARX NN controllers have trained with the Levenberg-Marquardt algorithm. The performances of the designed controllers for each maneuver have been compared under Gaussian noise. The PID and NARX NN controllers have performed similar results in the simulation environment.

Keywords: Altitude, attitude, unmanned aerial vehicle, octocopter, proportional-integrator-derivative, dynamic time series non-linear autoregressive with external (Exogenous) input neural network, feedback loop, gaussian noise.

BİR SEKİZ KANATLI İHA'NIN YAPAY SİNİR AĞLARI İLE YÜKSEKLİK KONTROLÜ

ÖZ

İnsansız hava araçlarının (İHA) kullanımı son yıllarda yaygınlaşmıştır. Bilimsel araştırmalara ve uzay teknolojilerinde kullanım çeşitliliğine öncülük eden İHA'lar, askeri, ticari, güvenlik gibi sektörlerde de geniş bir kullanım alanına sahiptir. Uçan araçların kontrolü, bozucu girdiler nedeniyle her zaman zor olmuştur. Bu çalışma, sürekli öğrenme yeteneğine sahip İHA'larda yaygın olarak kullanılan kontrol algoritmalarına alternatif olacak şekilde NARX NN algoritması ile otonom bir oktokopterin tutum ve irtifasını tasarlamayı ve kontrol etmeyi amaçlamaktadır. Oktokopter olarak da bilinen sekiz rotorlu multikopterin yuvarlanma, yunuslama, yalpalama ve irtifa kontrolü tasarlanmıştır. Oktokopterin lineer olmayan matematiksel modeli Newton-Euler yöntemi yardımıyla Matlab Simulink üzerinde gerçekleştirilmiştir. Önerilen oktokopter tasarımının kütle özelliklerini ve atalet momentlerini belirlemek için Solidworks üzerinde bir rijit gövde modeli tasarlanmıştır. Geri besleme döngü sisteminin irtifa ve durum kontrolü için klasik oransal-entegratör-türev (PID) algoritmaları ve dinamik zaman serisi doğrusal olmayan otoregresif dışsal girdi (NARX) sinir ağları (SA) kullanılmıştır. PID kontrolör kazançları Ziegler-Nichols yöntemi yardımıyla hesaplanmıştır. NARX NN kontrolörleri, Levenberg-Marquardt algoritması ile eğitilmiştir. Tasarlanan kontrolörlerin her bir manevra için performansları Gauss gürültüsü altında karşılaştırılmıştır. PID ve NARX NN kontrolörleri simülasyon ortamında benzer sonuçlar vermiştir.

Anahtar kelimeler: Yükseklik, davranış, insansız hava aracı, oktokopter, orantısal-entegratör-türev, dinamik zaman serisi dış (Dıştan) girdi ile doğrusal olmayan otoregresif sinir ağı, geri besleme döngüsü, gauss gürültüsü.

CONTENTS

Page

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

CHAPTER ONE - INTRODUCTION 1

1.1 Motivation Purpose 1
1.2 A Brief History 2
1.3 Literature Review 7
1.4 Thesis Structure 14

CHAPTER TWO - SYSTEM MODELING 15

2.1 Introduction 15
2.2 Coordinate Systems 15
2.3 Octocopter Kinematics 17
2.4 Transformations Between Coordinate Systems 19
2.5 Euler Angles 22
2.6 Time Derivatives of Rotation Matrices 23
2.7 Quaternions 25
2.8 Equations of Motion 27
2.9 Motor Forces and Torques 29
2.10 Control Algorithms 31
2.10.1 PID Control Algorithm 32
2.10.2 NARX Neural Network Control 34
2.10.3 Levenberg-Marquardt Algorithm 35

2.10.4 Gaussian Noise Algorithm 35

CHAPTER THREE – MODEL BASED DESIGN 36

3.1 Introduction 36
3.2 Methodology 36
3.3 Solid Model of the Octocopter 37
3.4 Model-Based Design 38
3.5 Model-Based Design of Attitude and Altitude Controllers 43
3.5.1 PID Control and Tuning 44
3.5.2 NARX NN Control and Training 48

CHAPTER FOUR- RESULTS 58

4.1 Introduction 58
4.2 PID Controller Tuning Results 58
4.3 NARX NN Controller Training Results 61
4.4 Comparison of Performance Results 76

CHAPTER FIVE - CONCLUSION 80

5.1 Conclusions 80
5.2 Future Works 81

REFERENCES 82

APPENDICES 91

LIST OF FIGURES

Page

Figure 1.1 Elmer and Elsie 3
Figure 1.2 The first autonomous UAV 3
Figure 1.3 The fixed-wing UAV 4
Figure 1.4 The rotary-wing UAV 4
Figure 1.5 The blimp 5
Figure 1.6 The flapping-wing UAV 5
Figure 2.1 The earth-fixed coordinate system 16
Figure 2.2 The body-fixed coordinate system 16
Figure 2.3 The linear position vector and illustration of the coordinate systems 17
Figure 2.4 Definition of roll, pitch, yaw maneuvers on an octocopter 18
Figure 2.5 The rotor actions for the four maneuvers of an octocopter 19
Figure 2.6 Rotations and angles between Q and B frames 19
Figure 2.7 The rotation from Q to $F1$ with Ψ angle 20
Figure 2.8 The rotation from $F1$ to $F2$ with θ angle 21
Figure 2.9 The rotation from $F2$ to B with ϕ angle 21
Figure 2.10 Motor rotations, angular velocities, and arm length 30
Figure 2.11 Feedback control 32
Figure 2.12 Flowchart of PID control 32
Figure 2.13 The NARX NN architecture 34
Figure 3.1 Flowchart of the designed system 37
Figure 3.2 Rendered octocopter model 38
Figure 3.3 Model-based design of the Octocopter 39
Figure 3.4 The earth-fixed frame, x, y location, and altitude sensor 40
Figure 3.5 Attitude sensor values and the body-fixed frame connection 40
Figure 3.6 The octocopter components and the body-fixed frame 41
Figure 3.7 Rotor and propeller connection 42
Figure 3.8 Detailed rotor and propeller design 42
Figure 3.9 Attitude and altitude control system 44
Figure 3.10 Altitude control with PID algorithm 45

Figure 3.11 The oscillation under ultimate gain Ku for altitude maneuver 45
Figure 3.12 Roll maneuver control with PID algorithm 46
Figure 3.13 The oscillation under ultimate gain Ku for roll maneuver 46
Figure 3.14 Pitch maneuver control with PID algorithm 47
Figure 3.15 The oscillation under ultimate gain Ku for pitch maneuver 47
Figure 3.16 Yaw maneuver control with proportional gain 48
Figure 3.17 The NARX NN control model 49
Figure 3.18 Altitude reference inputs without Gaussian noise 50
Figure 3.19 Altitude reference inputs with Gaussian noise 50
Figure 3.20 Altitude maneuver outputs of octocopter 51
Figure 3.21 Roll reference inputs without Gaussian noise 52
Figure 3.22 Roll reference inputs with Gaussian noise 52
Figure 3.23 Roll maneuver outputs of octocopter 53
Figure 3.24 Pitch reference inputs without Gaussian noise 54
Figure 3.25 Pitch reference inputs with Gaussian noise 54
Figure 3.26 Pitch maneuver outputs of octocopter 55
Figure 3.27 Yaw reference inputs without Gaussian noise 56
Figure 3.28 Yaw reference inputs with Gaussian noise 56
Figure 3.29 Yaw maneuver outputs of octocopter 57
Figure 4.1 Matlab simulation environment 58
Figure 4.2 The altitude maneuver response for tuned PID coefficients 59
Figure 4.3 The roll maneuver response for tuned PID coefficients 60
Figure 4.4 The pitch maneuver response for tuned PID coefficients 60
Figure 4.5 The yaw maneuver response for the proportional gain 61
Figure 4.6 NARX NN training performance results for the altitude maneuver 62
Figure 4.7 NARX NN controller for altitude maneuver training states 63
Figure 4.8 NARX NN autocorrelation of error results for the altitude maneuver 63
Figure 4.9 NARX NN error histogram results for the altitude maneuver 64
Figure 4.10 NARX NN input-error correlation results for the altitude maneuver 64
Figure 4.11 NARX NN regression results for the altitude maneuver 65
Figure 4.12 NARX NN training performance results for the roll maneuver 66
Figure 4.13 NARX NN controller for roll maneuver training states 66

Figure 4.14 NARX NN autocorrelation of error results for the roll maneuver..... 67
Figure 4.15 NARX NN error histogram results for the roll maneuver 67
Figure 4.16 NARX NN input-error correlation results for the roll maneuver 68
Figure 4.17 NARX NN regression results for the roll maneuver 68
Figure 4.18 NARX NN training performance results for the pitch maneuver..... 69
Figure 4.19 NARX NN controller for pitch maneuver training states..... 70
Figure 4.20 NARX NN autocorrelation of error results for the pitch maneuver 70
Figure 4.21 NARX NN error histogram results for the pitch maneuver..... 71
Figure 4.22 NARX NN input-error correlation results for the pitch maneuver..... 71
Figure 4.23 NARX NN regression results for the pitch maneuver 72
Figure 4.24 NARX NN training performance results for the yaw maneuver 73
Figure 4.25 NARX NN controller for yaw maneuver training states 73
Figure 4.26 NARX NN autocorrelation of error results for the yaw maneuver 74
Figure 4.27 NARX NN error histogram results for the yaw maneuver..... 74
Figure 4.28 NARX NN input-error correlation results for the yaw maneuver 75
Figure 4.29 NARX NN regression results for the yaw maneuver 75
Figure 4.30 Altitude maneuver reference inputs and controller output comparison.. 77
Figure 4.31 Roll maneuver reference inputs and controller output comparison..... 78
Figure 4.32 Pitch maneuver reference inputs and controller output comparison 78
Figure 4.33 Yaw maneuver reference inputs and controller output comparison 79

LIST OF TABLES

Page

Table 2.1 Ziegler-Nichols method coefficients for different approximations 34
Table 3.1 Mass properties of the octocopter system 38
Table 4.1 The MSE and RMSE performance results of attitude and altitude maneuver controllers..... 76

CHAPTER ONE
INTRODUCTION

1.1 Motivation Purpose

This study aims to design and analyze an autonomous octocopter which is a rotary-wing unmanned aerial vehicle (UAV). The octocopter has been designed in SolidWorks to import mechanical properties into the mathematical model. The mathematical model has been designed with coordinate system allocation, transformation matrix, Newton-Euler equations of motion definitions, quaternions, linear control theory, and control algorithms on the octocopter. The virtual design and properties of the octocopter model have been imported with the help of Simmechanics blocks to the Matlab environment. Moreover, gravitational force, body-frame, world-frame, and sensors have been added to the model with the Simscape library. The rotors, propellers, and fixed power supply have been designed with help of the Simelectronics library. The altitude and attitude of the octocopter have been controlled with classic PID and NARX NN algorithms to drive eight rotors separately. The PID controller gains for each maneuver have been found with the help of the Ziegler-Nichols method. On the other hand, the NARX NN algorithms have trained with the subtract diversified reference inputs, and desired outputs with the help of the Levenberg-Marquardt method. The mathematical model has been simulated on Matlab to observe the real-time reactions of octocopter in the virtual world. The performance results of the control algorithms have been compared with mean squared error (MSE) and root mean squared error (RMSE).

The rotary-wing UAVs have been controlled with different types of algorithms such as PID control, sliding mode control (SMC), fuzzy logic control, neural network algorithms, etc. to control the attitude and altitude of vehicles in literature. This study has been focused on the NARX NN algorithm controlling performance on the octocopter attitude and altitude dynamics as an alternative method. In addition, the NARX NN controller performance has been tested on a different platform with the same methodology to see the effect of dynamics and rotor numbers on quadcopters.

Accordingly, the quadcopter study has been derived from the scope of this thesis (Karakaya & Gören, 2022).

1.2 A Brief History

The advance in technology has increased the use of an electro-mechanical system called robots to perform risky tasks or cost-effective solutions. These machines have been controlled as manual, remote, and autonomous (Fahimi, 2009; Nonami, et al., 2010). Manual control is a control methodology that is performed by individuals with direct human interaction on the system. Likewise, the remote control has a direct interaction between individuals and robots where the operator controls the device by wired or wireless connections to send the commands. On the other hand, autonomous control is a control methodology where the system is self-controlled to perform its tasks independently (Fahimi, 2009; Visioli, 2006; Rabunal & Dorrado, 2006).

The first application of autonomous robots have been invented by neurobiologist William Grey Walter in the late 1940s. He has constructed the first two intelligent robots with 2 neuron brains called Elmer and Elsie, also known as tortoises that are shown in Figure 1.1 (Norman, 2018). In the following years, autonomous robot technology has been significantly improved with the help of researchers, scientists, and engineers to improve outcomes in the different sectors of industry such as military, pharmaceutical, space, shipment, aerospace, etc. For instance, multi-disciplinary inventions have carried the sign of tortoises like unmanned aerial vehicles, unmanned ground vehicles, satellites, submarines, factories, etc. (Fahimi, 2009).

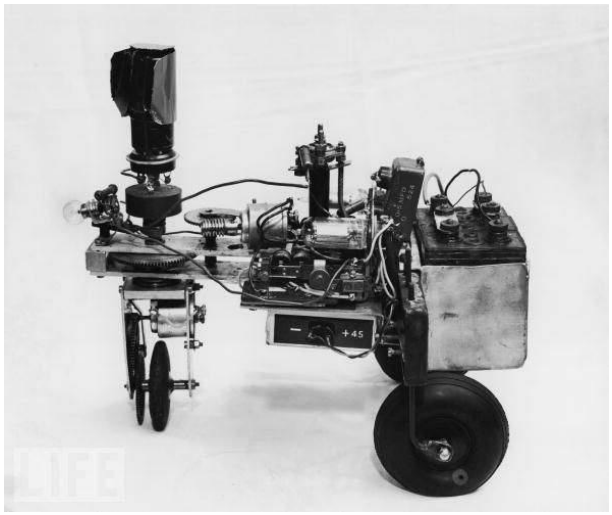

Figure 1.1 Elmer and Elsie (http://www.theoldrobots.com/)

The first foundation of the unmanned aerial vehicle has been built in 1916 by Lawrence Burst Sperry who is a pilot and engineer. Sperry invented the first autopilot. He has developed a working gyroscope and artificial horizon, in order to control the attitude (Nonami, et al., 2010). The first UAV is shown in Figure 1.2.



Figure 1.2 The first autonomous UAV (Nonami, et al., 2010)

The UAVs have been classified in different types of rotor and wing placement with the help of development and researches made in aerospace, space, and computer science technologies to be a solution to the needs of industry. These types have been classified by characteristics of UAVs such as fixed-wing UAVs, rotary-wing UAVs,

blimps, and flapping-wing UAVs. First, the fixed-wing UAVs are forward landing and take-off vehicles that can carry heavier payloads, long-distance flight, and endurance features. It is shown in Figure 1.3. Second, the rotary-wing UAVs also known as vertical take-off and landing (VTOL) UAVs have hovering capability, high maneuverability, high reliability, and low maintenance cost features. It is shown in Figure 1.4. Third, the blimps are balloons and airships that are mostly large-sized, lighter than air, slow speed but high endurance vehicles. It is shown in Figure 1.5. Fourth, the flapping-wing UAVs are using flexible wings that are inspired by the nature such as birds and insects. It is shown in Figure 1.6. (Nonami, et al., 2010; Quan, 2017).



Figure 1.3 The fixed-wing UAV ("General Atomics MQ-9 Reaper", 2021)



Figure 1.4 The rotary-wing UAV (Nonami, et al., 2010)

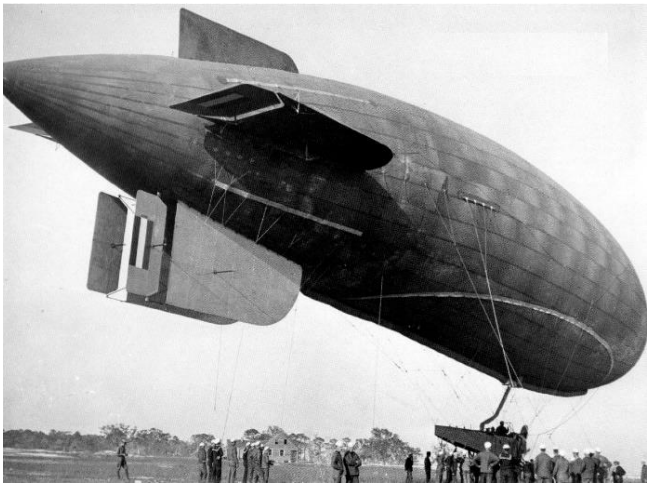

Figure 1.5 The blimp (“Blimp”, 2021)

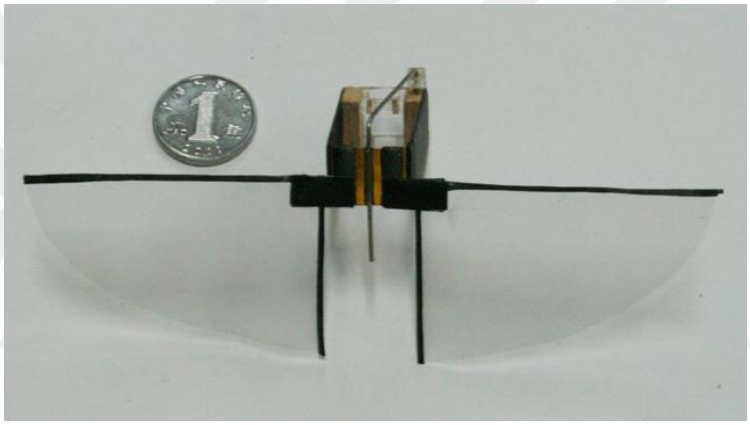

Figure 1.6 The flapping-wing UAV (http://spie.org)

PID control system’s proportional, integrator and derivative coefficients are set to meet defined performance requirements for the PID controllers. There are some advantages and disadvantages to using a PID control system (Muliadi & Kusumoputro, 2018; Visioli, 2006).

Advantages:

- Easy to implement,
- Working robust,
- Easy to achieve desired output with interested overshoot, settling time, response and oscillation,
- Capable to work in black-box models.

Disadvantages:

- Lesser reliability in different inputs (disturbances),
- Slower than Fuzzy logic, model predictive and neural network control methods.

PID control method has a drawback when the disturbance results in a large error in the system (Wang, et al., 2016). Controlling non-linear systems with artificial neural network control method gives better performance than PID control systems in most of the applications (Kusumoputro, et al., 2016).

The artificial neural network is largely used in many fields with the advancement of technology. It is mainly used for the following purposes (Nielsen, 2019; Goodfellow, et al., 2016; Graupe, 2007; Rabunal & Dorrado, 2006):

- Classification: Classification of input data
- Pattern matching: finding the most relevant pattern based on input data
- Pattern completion: Completing missing patterns based on input data
- Optimization: Calculation of optimum targeted values for an optimization problem.
- Control: Determining the most appropriate movement values according to the given input values.
- Function Estimation: Estimating the relationship between the given input and expected output.
- Data Mining: Revealing hidden patterns in the processed data set.

There are some advantages and disadvantages of artificial neural network algorithms (Tu, 1996; Nielsen, 2019; Goodfellow, et al., 2016; Graupe, 2007; Rabunal & Dorrado, 2006):

Advantages:

- ANN algorithms can produce very fast results.
- ANN can respond to unknown situations with training data.

- ANN gives better results when more data is used during its training.
- The learning mechanism can process large amounts of input data according to its memory capacity.

Disadvantages:

- It is difficult to detect mathematical errors because the mathematical algorithm works in itself,
- Training time and response time depend on the speed of the technical equipment of the device being trained.
- Although it differs according to the procedure to be performed, much more data is needed to increase a certain percentage of success.

1.3 Literature Review

The fundamentals of robots such as kinematics, dynamics, control logics, and mathematical solutions have been derived by various types of equations from mechanical, electrical, mathematical, and control aspects of the system. The book named “Autonomous Robots” has been covered these equations by links for manipulators, mobile robots, autonomous surface, and autonomous aerial vehicles (Fahimi, 2009). Xiang et al. has been worked on a mathematical model and parameter identification for brushless dc motor (Xiang, et al., 2015). Yudho Suprapto et al. have studied a mechanical model of a hexacopter, and a mathematical model of motors and propellers (Yudho Suprapto, et al., 2017). To summarise, the mechanical design and the force generation mechanisms like propellers and motors have been covered under multiple research with electrical and mechanical impacts on the robots with the same fundamental equations.

The UAVs have been classified into three main variants that are mentioned in the previous title. The mathematical modeling, controller design, and sensor usages for every UAV have been examined in two different books that are named “Autonomous Flying Robots” and “Introduction to Multicopter Design and Control” (Nonami, et al., 2010; Quan, 2017). Ding et al. have been summarized the design,

modeling, and control methodologies and mechanical structures for the small-scale rotorcrafts (DING, et al., 2019). Mahony et al. have summarized the commonly used mathematical model and control methodologies of a quadcopter (Mahony, et al., 2012). Carlton has studied a system identification with the help of a software package called Comprehensive Identification from FrEquency Responses (CIFER) for a manufactured quadcopter (Carlton, 2017). Chapala et al. have derived the mathematical model with the Newton-Euler method for hexacopter (Chapala, et al., 2016). Given the above, the system identification and mechanical design for UAVs have been derived from fluid dynamics, machine dynamics, and mechanics of materials equations to handle multiple tasks. In addition, the mathematical model of UAVs has been derived from Newton-Euler's equations of motion and quaternions.

The implementation and development of control methodologies into industrial use has always been a challenge. The most common control method called PID has been designed, analyzed, tuned, and simulated in the book named "Practical PID Control". In addition, the mathematical models have been implemented with real-life uses to enrich the quality and performance of different types of PID methodologies (Visioli, 2006). Li et al. has covered the use cases, tuning methods, and mathematical model for PID control systems in their study (Li, et al., 2006). The tilt-rotor tri-copter which is a vertical take-off and land vehicle (VTOL) has been controlled with the PID algorithm (Summers, 2017). Çakıcı and Leblebicioğlu have studied a PID controller design for a fixed-wing (FW) vertical take-off and landing (VTOL) UAV. Consequently, they have compared the real-flight and simulation results (ÇAKICI & LEBLEBİCİOĞLU, 2016). Wang et al. have proposed a mathematical model of a quadcopter to compare the cascade PID algorithm and classic PID control algorithm performance where the cascade PID gives better performance results (Wang, et al., 2016). Mac et al. has been proposed a particle swarm optimization for autonomous navigation in indoor environments that is applied on a quadcopter with a PID controller (Mac, et al., 2018). Mustapa et al. have proposed an altitude PID controller for quadcopter (Mustapa, et al., 2014). Paiva et al. have proposed a mathematical model, and attitude and altitude control with a PID algorithm for a quadcopter (Paiva, et al., 2016). Moreover, Luukkonen has studied a mathematical model, and

attitude control with a PD controller on a quadcopter (Luukkonen, 2011). Praveen and Pillai have studied a mathematical model and PID algorithm to control the attitude of a quadcopter that is implemented into a real-life vehicle (Praveen & Pillai, 2016). Tagay et al. have proposed a determination of mathematical model and PID control algorithm for attitude control on a quadcopter (Tagay, et al., 2021). Cedro and Wieczorkowski have proposed a PID controller and its tuning methods for a quadcopter where the mathematical model has been derived from Newton's equations of motion (Cedro & Wieczorkowski, 2019). As can be seen, the mathematical model for UAVs has been derived from the Newton-Euler method. The PID controller which is the most commonly used controller for most industries has been used to achieve multiple tasks such as controlling the attitude and trajectory for UAVs.

Artificial neural network algorithms have been used to solve problems and make predictions about the outcomes with trained data and self-learning capability from observational data. The book named "Neural Networks and Deep Learning" has been presented the neural networks architecture, backpropagation algorithms, deep learning, function solving capability of ANN, learning and training examples of ANN structure (Nielsen, 2019). In addition, the book named "Deep Learning" has been presented mathematical applications of machine learning, types of deep learning algorithms, and a variety of training methods (Goodfellow, et al., 2016). Moreover, the book named "Principles of Artificial Neural Networks" has been presented the mentality of ANN structures, the single and multi-layer perceptron structure, back-propagation algorithms, recurrent algorithms, adaptive learning algorithms that have been explained within the detailed mathematical formulations (Graupe, 2007). On the other hand, the book named "Artificial Neural Networks in Real-Life Applications" has been presented multiple use cases of ANN algorithms such as finance to stellar technologies (Rabunal & Dorrado, 2006). El Hamidi et al. have compared the control performance of non-linear system control for the nonlinear autoregressive with exogenous input (NARX) neural network, nonlinear autoregressive moving average (NARMA-L2), and a feedforward neural network (FFNN) algorithm. As a result of this study, ANN algorithms have been successfully

utilized for controlling different non-linear dynamic systems (El Hamidi, et al., 2020). Tu has been used an artificial neural network to predict the non-linear relationships between medical outcomes (Tu, 1996). El Dakrory and Tawfik have been used NARX neural networks to identify non-linear dynamic systems (El Dakrory & Tawfik, 2017). Tijani et al. have been used the NARX NN for the non-linear model identification from experimental flight data for a small-scale unmanned helicopter (Tijani, et al., 2014). Alternatively, the back-propagation neural network has been used to compensate for the vibrational noise that occurred on pilot inputs (Kumar, et al., 2016). Padhy et al. have been proposed a convolutional neural network (CNN) trajectory control method for quadcopter navigation in indoor environments (Padhy, et al., 2018). Ultimately, the ANN algorithms have been used as a predictor, dynamic system identifier, non-linear system controller, and compensator.

It is proposed a feed-forward neural network controller to stable flight under disturbances such as wind, payloads, and ground effects for a quadcopter. Also, it has derived a mathematical model with newton's second law and kinematic equations. (Kotzé, et al., 2021). Razmi and Afshinfar have been proposed a position and attitude control design with ANN-based adaptive sliding mode (ASM) control for a quadcopter (Razmi & Afshinfar, 2019). Say et al. have been used the NARX NN algorithm for attitude estimation of a quadcopter. The performance has been compared with the extended Kalman filter (EKF) method. The NARX NN algorithm has given lesser MSE errors than EKF (Say, et al., 2019). In addition, Akın et al. has been worked on the sensor filtering implementation with the Kalman filter method for predicting the altitude (Akın, et al., 2021). Muladi and Kusumoputro have designed a PID and direct inverse control (DIC) ANN algorithm for altitude control of a quadcopter. In addition, the DIC ANN algorithm has performed better than the PID algorithm in test results (Muliadi & Kusumoputro, 2018). Kusumoputro et al. have developed an Elman recurrent neural network for the attitude control system of a hexacopter. The real test flights have been used to train the ANN algorithm that is successfully controlled the attitude maneuvers (Kusumoputro, et al., 2016). Altan et al. have proposed real-time attitude, altitude, and trajectory control on NARX NN of

a hexacopter. The NARX NN algorithm has performed better than the PID control algorithm on path tracking performance (Altan, et al., 2018). It can be concluded that different types of ANN algorithms have been implemented successfully to control the attitude and altitude of UAVs. The ANN algorithms mentioned above have given promising and more successful results than other control algorithms.

In literature, multiple studies have been made to control the altitude and attitude on the dynamic model of a quadcopter which is derived from the Newton-Euler method, with sliding mode control (SMC) algorithm (Elhennawy, 2018). Zulu and John have been reviewed the mathematical model which is derived from the Newton-Eulers method, and the autonomous quadcopter control algorithms such as PID, linear quadratic regulator (LQR), sliding mode, backstepping, feedback linearization, adaptive, robust, optimal, $\mathcal{L}_1$, H_∞, fuzzy logic, and ANN (Zulu & John, 2014). Rice has been proposed PID, fuzzy logic, and non-linear dynamic inversion (NLDI) controllers to test the performance of a quadcopter on the formation flight duty (Rice, 2014). Wei has proposed PID and fuzzy logic controller designs for a quadcopter. Moreover, the performance of these controllers has been compared under simulation and real-life tests (Wei Wei, 2015). Shupeng has been proposed trajectory generation in obstacle-dense environments and robust control for rotorcraft UAVs (Shupeng, 2016). Labbadi et al. have proposed a mathematical model and attitude control with integral sliding mode control for a quadcopter. The integral sliding mode control has acted much more robust and faster response than sliding mode control (Labbadi, et al., 2018). Ahmad et al. have proposed a linear quadratic regulator to control the attitude and altitude of a quadcopter (Ahmad, et al., 2018). Pérez has proposed an $\mathcal{L}_1$ adaptive controller to neglect disturbance effects and also five control methods have been used to minimize tracing errors on trajectory control. Iterative learning control (ILC), multi-robot transfer learning, multi-task transfer learning, adaptive model predictive control (MPC), and model predictive control methods have been used for trajectory control for a quadcopter (Pereida Pérez, 2020). Given the above, different types of control algorithms have been used to control the attitude, altitude, and trajectory control of UAVs. The SMC has been used more commonly than other methods.

Some studies have been combined multiple control algorithms to control the attitude, altitude, and trajectory of UAVs. Analia and Song have been proposed a combined attitude controller with Fuzzy and PID controller for a co-axial copter. They have achieved similar results between simulation and real experimental results (Analia & Song, 2016). Rmilah has been proposed a fuzzy-tuned PID controller to control the altitude, attitude, and path tracking control for a quadcopter dynamic model (Rmilah, 2016). Nguyen et al. have proposed neural network (NN) based adaptive sliding mode control for attitude and altitude maneuvers of a quadcopter. The proposed controller has given a faster reaction than the adaptive SMC (Nguyen, et al., 2021). Ferdaus et al. have proposed a self-organizing controller. The position of the UAV has controlled with a neuro-fuzzy controller. In addition, SMC has been used to overcome tracking errors (Ferdaus, et al., 2018). El Hamidi et al. have proposed auto-tuned PD-PID algorithms with neural networks method and fuzzy-tuned particle swarm optimization method to control attitude and altitude of a quadcopter. The neural network algorithm has given the best results on PD and PID control than fuzzy logic method and fuzzy-tuned particle swarm optimization method (El Hamidi, et al., 2019). In addition, Yoon et al. have proposed a mechanical design and PID-based attitude and altitude control which is updated the PID gains with a neural network, for a quadcopter (Yoon, et al., 2016). The studies mentioned above have been used these combined control algorithms to achieve more robust and stable control for UAVs. The combined control algorithms have given better results than non-combined versions.

In this study, the inputs have diversified with Gaussian noise to add disturbance effects on the octocopter. Also, the NARX NN algorithm is trained with Levenberg-Marquardt algorithms to achieve the best results from the training process. The following researches have been used in this study to aid the previously mentioned methods. Levenberg and Marquardt have proposed two different failure prevention methods for solving non-linear simultaneous equations (Levenberg, 1944; Marquardt, 1963). Hagan and Menhaj have used Levenber-Marquardt algorithm with backpropagation method to train feedforward neural networks (Hagan & Menhaj,

1994). McHutchon and Rasmussen have studied a Gaussian Process regression to predict the noise-free inputs which are corrupted by Gaussian noise (Mchutchon & Rasmussen, 2011). Krivec et al. have proposed Gaussian process approximations to form variational Gaussian process NARX models (Krivec, et al., 2021). Moreover, Hagiwara et al. have proposed a method to minimize errors for a Gaussian noise sequence under non-linear system training with a neural network regression algorithm (Hagiwara, et al., 2001). Silva and Adeodato have studied multilayer perceptron (MLP) neural network (NN) on transformed inputs with principal component analysis (PCA) and additive Gaussian noise (Silva & Adeodato, 2011).

Bergman & Ekström have been proposed the development of a physical octocopter model, parameter identification for the mathematical model, an $\mathcal{L}_1$ adaptive controller, and PID controller design, and performance comparison of the controllers in the simulation environment (Bergman & Ekström, 2014). Mikkelsen has studied on design and mathematical modeling of an octocopter. In addition, state feedback controllers with an integrator, and without an integrator have been used to control the attitude and altitude maneuvers (Mikkelsen, 2015). Oscarson has proposed a design and mathematical model for an octocopter. Moreover, the state feedback control and Lyapunov-based control have been simulated on the mathematical model compared to the simulation results on manufactured octocopter (Oscarson, 2015). Bouzid et al. have proposed an octocopter mathematical model in order to find controller effectiveness which is internal model-feedback linearization control on trajectory tracking (Bouzid, et al., 2016). Donmez et al. have been studied to detect agricultural plants to find the total plant amount by connected components labeling (CBL) algorithm on the octocopter (Donmez, et al., 2021). Zhu et al. have been proposed a computational fluid dynamics (CFD) analysis to compare two different octocopter designs. In addition, they have simulated the flight performance of octocopter designs with traditional PID controllers (Zhu, et al., 2020). Yıldırım et al. have designed the mathematical and physical model for octocopter and dodecacopter. In addition, they also have proposed a trajectory control system with PID algorithms to compare the performance of both vehicles (Yıldırım, et al., 2019). Zhang et al. have proposed a mechanical design and PID-

based attitude and altitude control for a mini-octocopter (Zhang, et al., 2021). Moussid et al. have derived the mathematical model with the help of the Newton-Euler method for quadcopter, hexacopter, and octocopter. They have proposed three control algorithms to control attitude and altitude. These are backstepping, sliding-mode, and a hybrid backstepping Frenet-Serret theory (FST) controllers (Moussid, et al., 2017). Finally, the mathematical model for octocopter has been derived from Newton-Euler's equations of motion like all else. $\mathcal{L}_1$ adaptive, Lyapunov-based, backstepping, sliding-mode, and hybrid backstepping FST controllers have been used to control the attitude and altitude of octocopters in literature.

1.4 Thesis Structure

This study has been divided into 5 main chapters. First, the purpose and motivation of this study, a brief history of robots and UAVs, and a literature review have been covered in Chapter I. Second, the mathematical model has been derived in Chapter II. Third, methodology, model-based design, and control algorithm implementation have been described in Chapter III. Fourth, the PID tuning results, the NARX NN training results, and performance comparison of PID and NARX NN controllers have been covered in Chapter IV. In the end, the thesis has been finalized by findings, conclusions, and future work recommendations in Chapter V.

CHAPTER TWO
SYSTEM MODELING

2.1 Introduction

In this chapter, the mathematical formulations that are used in modeling, have been explained and formulated as coordinate system placement and transformations, octocopter kinematics, Euler angles, rotation matrices, quaternions, equations of motion, motor forces and torques, control algorithms, and tuning and training methods for control algorithms.

2.2 Coordinate Systems

UAV movements and maneuvers have been described in the three-dimensional six degrees of freedom (DOF) space. In the literature, five different coordinate systems have been used to define the maneuvers of UAVs. These coordinate systems have been classified as the earth-fixed frame, the body-fixed frame, the inertial coordinate system, the navigation coordinate system, and the wander azimuth navigation system. In this study, the earth-fixed frame and the body-fixed frame have been used to derive maneuvers of the octocopter. The earth-fixed frame has been used in the three-dimensional cartesian coordinates $[x, y, z]$ where the center point is fixed on the surface of the earth. In the aviation and aerospace industry, the most commonly used coordinate system is north, east, down (NED) placement. In the NED coordinate system, the X^E axis has always been used to point the north direction, the Y^E axis points to the east, and the Z^E axis points center of the world. The earth-fixed coordinate system is shown in Figure 2.1. (Bergman & Ekström, 2014; Mikkelsen, 2015; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

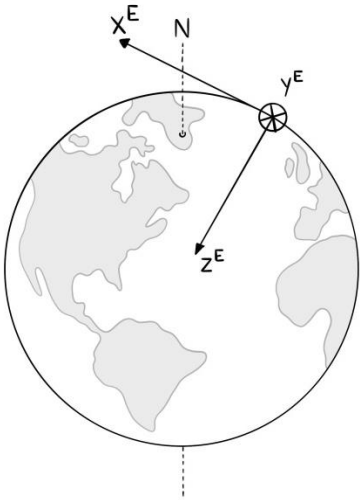

Figure 2.1 The earth-fixed coordinate system

The body-fixed frame has been used to describe angular displacements of the vehicle. This frame has been chosen because of the following advantages. First, like the inertia matrix is time-invariant Second, the body symmetry can be used to simplify the equations. Third, measurements taken onboard can easily convert into the body-fixed frame. Last, the control forces can be placed on the body-fixed frame. In the body-fixed coordinate system center of the octocopter has been placed into the center of the vehicle's coordinate system. In the body-fixed coordinate system, three-dimensional placement is shown in Figure 2.2 where the X^B axis is shown as red arm, Y^B axis is the perpendicular arm of the octocopter and the Z^B axis is towards the bottom. (Bergman & Ekström, 2014; Mikkelsen, 2015; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

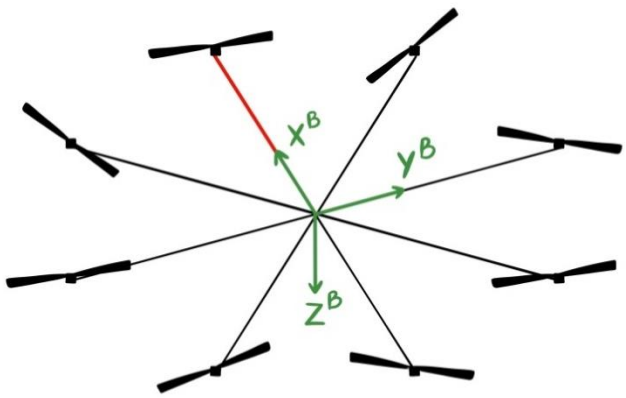

Figure 2.2 The body-fixed coordinate system

The inertial coordinate system has used the earth's center as the origin of the system and its non-rotating axes. On the other hand, the navigation coordinate system has taken the origin as the center of the aerial vehicle where the axes are aligned in the directions of east, north, and vertical up, also known as ENU-frame. Wander azimuth navigation system has been used to avoid singularities that occur at the poles of the navigation system. (Chapala, et al., 2016)

2.3 Octocopter Kinematics

In this study, the body-fixed and the earth-fixed coordinate system have been used to define the kinematic and the motion equations of octocopter in space. In addition, linear and angular positions have been used to link the coordinate systems and movement equations. The linear position has been defined as a position vector Γ between the earth-fixed frame and the center of the octocopter. The linear position is shown in Figure 2.3 and defined in Equation 2.1. In addition, the relation between the earth-fixed and body-fixed coordinate system is shown in Figure 2.3. (Bergman & Ekström, 2014; Oscarson, 2015; Mahony, et al., 2012; Mikkelsen, 2015; Mac, et al., 2018; Chapala, et al., 2016)

$$r^E \triangleq [x^E \; y^E \; z^E]^T \tag{2.1}$$

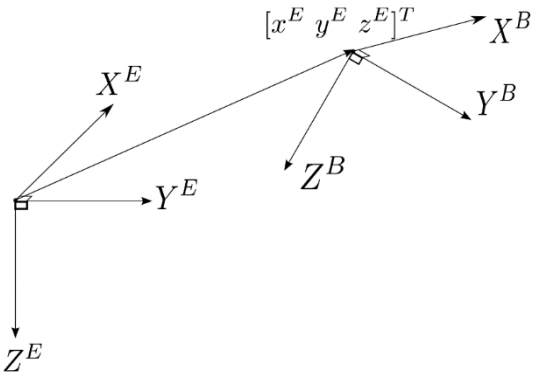

Figure 2.3 The linear position vector and illustration of the coordinate systems

The yaw, pitch, and roll maneuvers, also known as Euler rotation angles, have been derived with the aviation standards and Euler angles by angular position. The

angular position Θ^E has been derived from the rotation of the earth-fixed frame to the body-fixed frame. The angular position vector is shown in Equation 2.2 where the ϕ, θ, ψ are roll, pitch, and yaw angle. The linear and angular velocities have been derived from the first order of linear and angular positions. The linear velocity is shown in Equation 2.3 and the angular velocity is shown in Equation 2.4. (Bergman & Ekström, 2014; Oscarson, 2015; Mahony, et al., 2012; Mikkelsen, 2015; Mac, et al., 2018; Chapala, et al., 2016)

$$\eta^E \triangleq [\phi\ \theta\ \psi]^T \tag{2.2}$$

$$v^B \triangleq [\,u\ v\ w]^T \tag{2.3}$$

$$\omega^B \triangleq [p\ q\ r]^T \tag{2.4}$$

The octocopter maneuvers have been defined as roll, pitch, yaw, and vertical up-down which can be easily shown in body-fixed coordinate systems. The yaw, pitch, and roll maneuvers are shown in Figure 2.4 where the yaw maneuver has been defined as rotation around the Z axis, the roll maneuver has been defined as rotation around X axis and the pitch maneuver has been defined as rotation around Y axis. The vertical up-down has been defined as displacement on Z axis. The rotations of every motor and maneuvers are shown in Figure 2.5 where green color means faster rotation, red means slower rotation, and gray means stable rotation around rotors.

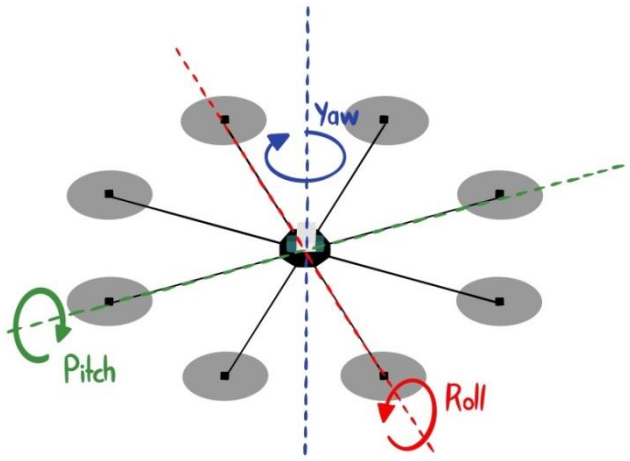

Figure 2.4 Definition of roll, pitch, yaw maneuvers on an octocopter

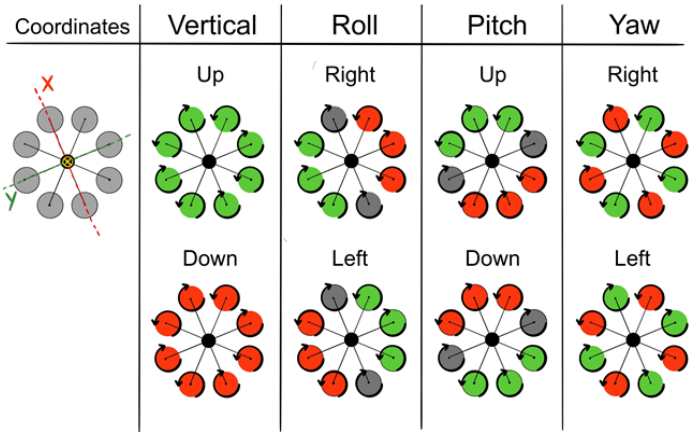

Figure 2.5 The rotor actions for the four maneuvers of an octocopter

2.4 Transformations Between Coordinate Systems

The coordinate systems have been used to define maneuvers and the location of the UAVs. In this study, dynamic equations of motion, forces, and torques have been expressed with the rotation of the body-fixed frame coordinate system around the earth-fixed inertial coordinate system. The equations have been derived in three rotations to align the coordinate systems with help of simple trigonometric transformations. The origin of the earth-fixed coordinate system has been placed on earth as Q where the origin of the UAV has been defined as B. The first rotation of Z^Q around the Z^B has been specified with Ψ angle where the rotation is expressed as $R_{Q\to F_1}$. Likewise, the second rotation of $Y^{Q'}$ has been specified with θ where the rotation is expressed as $R_{F_1\to F_2}$. Similarly, the third rotation of $X^{Q''}$ has been specified with Φ where the rotation is expressed as $R_{F_2\to B}$. The rotations of the coordinate systems are shown in Figure 2.6 (Bergman & Ekström, 2014; Oscarson, 2015; Mahony, et al., 2012; Mikkelsen, 2015; Mac, et al., 2018; Chapala, et al., 2016).

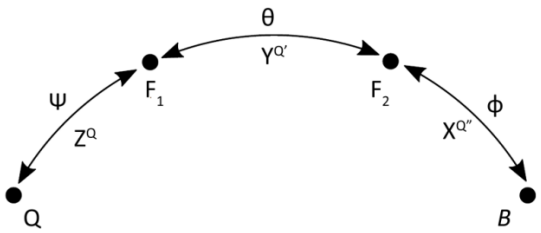

Figure 2.6 Rotations and angles between Q and B frames

The first rotation around z^Q with Ψ angle is shown in Figure 2.7. According to the first rotation of body-fixed frame coordinate system around earth-fixed inertia coordinate system, the coordinates in F_1 is shown in Equation 2.5 with earth-frame coordinates and the rotation matrix $R_{Q\to F_1}$ is shown in Equation 2.6. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

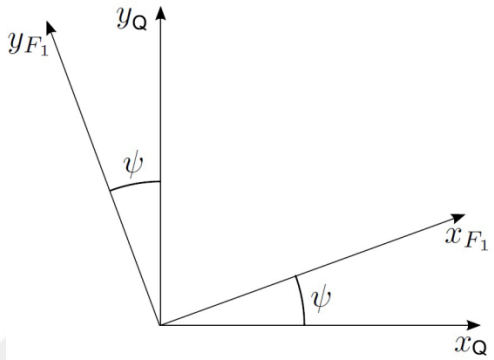

Figure 2.7 The rotation from Q to F_1 with Ψ angle (Oscarson, 2015)

$$\begin{bmatrix} x^{F_1} \\ y^{F_1} \\ z^{F_1} \end{bmatrix} = \begin{bmatrix} x^Q cos\Psi - y^Q sin\Psi \\ x^Q sin\Psi + y^Q cos\Psi \\ z^Q \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} x^{F_1} \\ y^{F_1} \\ z^{F_1} \end{bmatrix} = \begin{bmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^Q \\ y^Q \\ z^Q \end{bmatrix} = R_{Q\to F_1} \tag{2.6}$$

The second rotation around $Y^{Q'}$ with θ angle is shown in Figure 2.8. It follows that the second rotation of body-fixed frame coordinate system around earth-fixed inertia coordinate system, the coordinates in F_2 is shown in Equation 2.7 with F_1 coordinates and the rotation matrix $R_{F_1\to F_2}$ is shown in Equation 2.8. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

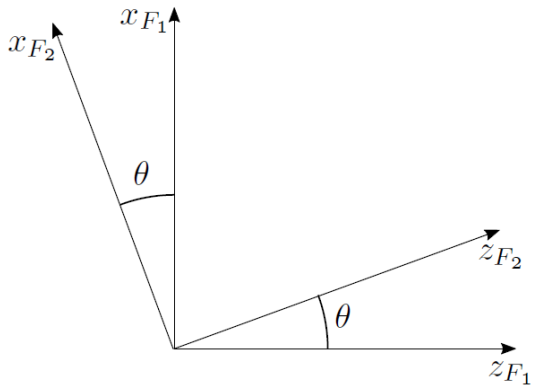

Figure 2.8 The rotation from F_1 to F_2 with θ angle (Oscarson, 2015)

$$\begin{bmatrix} x^{F_2} \\ y^{F_2} \\ z^{F_2} \end{bmatrix} = \begin{bmatrix} x^{F_1} cos\theta - z^{F_1} \sin\theta \\ y^{F_1} \\ x^{F_1} \sin\theta + z^{F_1} \cos\theta \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} x^{F_2} \\ y^{F_2} \\ z^{F_2} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} x^{F_1} \\ y^{F_1} \\ z^{F_1} \end{bmatrix} = R_{F_1 \to F_2} \tag{2.8}$$

The second rotation around $X^{Q''}$ with ϕ angle is shown in Figure 2.9. Therefore the third rotation of body-fixed frame coordinate system around earth-fixed inertia coordinate system, the coordinates in B is shown in Equation 2.9 with F_2 coordinates and the rotation matrix $R_{F_2 \to B}$ is shown in Equation 2.10. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

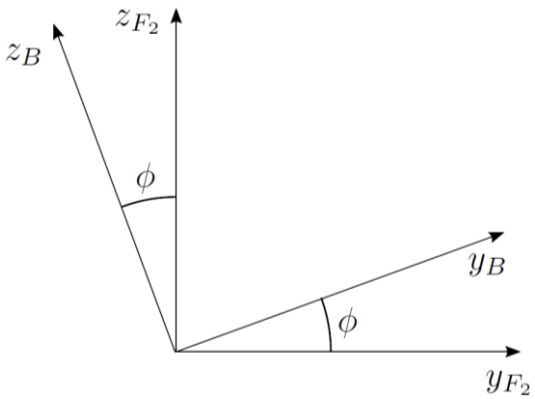

Figure 2.9 The rotation from F_2 to B with ϕ angle (Oscarson, 2015)

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = \begin{bmatrix} x^{F_2} \\ y^{F_2}cos\phi + z^{F_2}sin\phi \\ -y^{F_2}sin\phi + z^{F_2}cos\phi \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x^{F_2} \\ y^{F_2} \\ z^{F_2} \end{bmatrix} = R_{F_2 \to B} \tag{2.10}$$

The rotation matrix $R_{Q\to B}$ has been obtained by $R_{Q\to F_1}$, $R_{F_1\to F_2}$, and $R_{F_2\to B}$. In Equation 2.11 abridgments have been made in trigonometric functions that are $\cos x \equiv c_x$ and $sinx \equiv c_y$. $R_{B\to Q}$ has been obtained by the transpose of $R_{Q\to B}$. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$R_{Q\to B} = R_{F_2\to B}R_{F_1\to F_2}R_{Q\to F_1} = \begin{bmatrix} c_\theta c_\Psi & c_\theta s_\Psi & -s_\theta \\ s_\phi s_\theta c_\Psi - c_\phi s_\Psi & s_\phi s_\theta s_\Psi + c_\phi c_\Psi & s_\phi c_\theta \\ c_\phi s_\theta c_\Psi + s_\phi s_\Psi & c_\phi s_\theta s_\Psi - s_\phi c_\Psi & c_\phi c_\theta \end{bmatrix} \tag{2.11}$$

2.5 Euler Angles

The relation between the body-fixed frame and earth-fixed frame coordinate systems has been derived from the time-derivative of the Euler angles. The Q coordinate system has been related to the earth-fixed coordinate system and the B coordinate system has been related to the body-fixed coordinate system. This rotation is shown in Figure 2.6. The time derivative of Euler angles is shown with $\dot{\eta}$. The angular velocity of the body frame has been defined with ω shown is in Equation 2.12 where ω_x is the angular rate of the body around the x axis, ω_y is the angular rate of the body around the y axis, and ω_z is the angular rate of the body around the z axis. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.12}$$

In the rotation sequence x, y, z shown in Figure 2.6, $\dot{\Psi}$ angular-rate has been rotated two times to complete $F_1 \to B$ rotation, $\dot{\theta}$ angular-rate has been rotated one

time to complete $F_2 \rightarrow B$ rotation. On the other hand, the $\dot{\phi}$ angular rate is not needed rotation. This operation is shown in Equation 2.13 and Equation 2.14 with the corresponding rotation matrices. W is the transformation matrix (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = R_{F_{1\rightarrow B}}\begin{bmatrix}0\\ 0\\ \dot{\psi}\end{bmatrix} + R_{F_{2\rightarrow B}}\begin{bmatrix}0\\ \dot{\theta}\\ 0\end{bmatrix} + \begin{bmatrix}\dot{\phi}\\ 0\\ 0\end{bmatrix} \tag{2.13}$$

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}1 & 0 & -sin\theta\\ 0 & cos\theta & cos\theta sin\phi\\ 0 & -sin\phi & cos\theta cos\phi\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = W_{Q\rightarrow B}\dot{\eta} \tag{2.14}$$

The Inverse of $W_{Q\rightarrow B}\dot{\eta}$ matrix gives the $W_{B\rightarrow Q}\omega$ matrix that is shown in Equation 2.15. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & sin\phi tan\theta & cos\phi tan\theta\\ 0 & cos\phi & -sin\phi\\ 0 & sin\phi sec\theta & cos\phi sec\theta\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = W_{B\rightarrow Q}\omega \tag{2.15}$$

The equation of motions in 6 DoF has been derived from the body-fixed frame and earth-fixed inertial frame. The arbitrary vector has been assigned as a and the time derivative as $\dot{a}$. The body-fixed frame has rotated with the angular velocity ω. The time derivative of a with respect to the earth-fixed inertial frame a_Q, and relation with the time derivative of a respect to the body-fixed frame $\dot{a}_B$ is shown in Equation 2.16. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$(\dot{a})_Q = (\dot{a})_B + \omega \times r \tag{2.16}$$

2.6 Time Derivatives of Rotation Matrices

The time derivative of rotation matrices has been derived in this section. Let's assume that a point in the free space has a position vector r and the velocity vector

v. The earth-fixed frame position and velocity vectors are r^{Q}, v^{Q} and the body-fixed frame position and velocity vectors are r^{B}, v^{B} and the angular velocity vector of the body-fixed frame is ω^{B}. R is the rotation matrix expressed in Equation 2.11. The position vector and velocity vector are shown in Equation 2.17 and Equation 2.18. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$r^{Q} = Rr^{B} \tag{2.17}$$

$$v^{Q} = Rv^{B} \tag{2.18}$$

The time derivative of the position vector of the earth-fixed system r^{Q} is shown in Equation 2.19 also equal to v^{Q}. The velocity of the body-fixed frame v^{B} has been calculated with Equation 2.16 and shown in Equation 2.20. The last equation shown in Equation 2.21 has been obtained with the combination of Equation 2.18, Equation 2.19, and 2.20. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\dot{r}^{E} = v^{Q} = \dot{R}r^{B} + R\left(\frac{dr}{dt}\right)_{B}^{B} \tag{2.19}$$

$$\dot{r}^{B} = v^{B} = \left(\frac{dr}{dt}\right)_{B}^{B} + \omega^{B} \times r^{B} \tag{2.20}$$

$$\dot{r}^{Q} = v^{Q} = Rv^{B} = R\left(\left(\frac{dr}{dt}\right)_{B}^{B} + \omega^{B} \times r^{B}\right) = \dot{R}r^{B} + R\left(\frac{dr}{dt}\right)_{B}^{B} \tag{2.21}$$

The equality in Equation 2.21 the $R\left(\frac{dr}{dt}\right)_{B}^{B} = R\dot{r}^{B}$ has been derived with the help of Equation 2.11 and Equation 2.20. On the other hand, the second part of addition which is shown in Equation 2.22, has been expressed with the help of the skew-symmetric matrix ω_{x}^{B}, and position vector a. To summarize that the time derivative of the rotation matrix which is shown in Equation 2.25, has been derived from Equation 2.23 and Equation 2.24. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\dot{R}r^B = R(\omega^B \times r^B) \tag{2.22}$$

$$\omega_x^B a \triangleq \omega^B \times a \tag{2.23}$$

$$\omega_x^B = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{2.24}$$

$$\dot{R} = R\omega_x^B \tag{2.25}$$

2.7 Quaternions

The Euler angles and rotation matrices are lack clarity to illustrate certain orientations and maneuvers. Moreover, these methods have trouble with pitch angle $\theta = \pm 2\pi$ radians. In addition, the rotations have been derived in two dimensions with three components $(t, cost, sint)$. It follows that there is an alternative presentation called quaternions has been used in this study. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

The quaternion is a vector that consists of a real part and imaginary parts of x, y, z axis. The imaginary parts of x, y, z are presented as i, j, k where the second power of these variables are equal to each other $i^2 = j^2 = k^2 = ijk = -1$. One real and three imaginary quaternion vectors have been used to define rotations in three-dimensional space. The quaternion vector q is shown in Equation 2.26 and the quaternion unit vector is shown in Equation 2.27. The scalar part is presented as S and the imaginary part is presented as V. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$q \equiv (q_0\ q_1\ q_2\ q_3)^T \equiv (S\ V)^T \tag{2.26}$$

$$q = (cos\frac{\theta}{2}\ sin\frac{\theta}{2}V)^T \tag{2.27}$$

The further calculations have been derived from the norm of the quaternion which is shown in Equation 2.28. The quaternion multiplication has been used to multiply vector parts of quaternions that are shown in Equation 2.29. It follows that resultant quaternion $\bar{q}$ is shown in Equation 2.30. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\|q\| = \sqrt{{q_0}^2\ {q_1}^2\ {q_2}^2\ {q_3}^2} = 1 \tag{2.28}$$

$$q_1 \times q_2 = (S_1\ V_1)^T \times (S_2\ V_2)^T = (S_1S_2 - V_1V_2 \quad S_1V_2 + S_2V_1 + V_1 \times V_2)^T \tag{2.29}$$

$$\bar{q} = (S\ -V)^T \tag{2.30}$$

The rotation of body-fixed frame around inertial earth-fixed frame has achieved by a unit quaternion $q_{B\to Q}$ which is equal to $(q_0\ q_1\ q_2\ q_3)$. The pure quaternions have been presented with zero scalar value. As a result, the quaternion relation between frames is shown in Equation 2.31. The relation between rotation matrix $Q_{B\to Q}$, and arbitrary position vectors for both frames are shown in Equation 2.32 and Equation 2.33. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$\left(0\ r_Q\right)^T = q_{B\to Q}\ \times (0\ r_B)^T \times \bar{q}_{B\to Q} \tag{2.31}$$

$$r_Q = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2 & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} r_B \tag{2.32}$$

$$r_Q = Q_{B\to Q} r_B \tag{2.33}$$

The quaternion of $q_{B\to Q}$ has been represented with cartesian unit vectors and Euler rotation angles which is shown in Equation 2.34. In addition, Euler angles $\Phi\ \theta\ \Psi$ have been derived from solving rotation matrices for Equation 2.34 which is shown in Equation 2.35. The time derivative of quaternions has been used to make

continuity in time which is shown in Equation 2.36. (Bergman & Ekström, 2014; Mikkelsen, 2015; Oscarson, 2015)

$$q_{B\to Q} = \begin{pmatrix} cos\frac{\Psi}{2} \\ sin\frac{\Psi}{2}e_z \end{pmatrix} \times \begin{pmatrix} cos\frac{\theta}{2} \\ sin\frac{\theta}{2}e_y \end{pmatrix} \times \begin{pmatrix} cos\frac{\Phi}{2} \\ sin\frac{\Phi}{2}e_x \end{pmatrix} \tag{2.34}$$

$$\begin{bmatrix} \Phi \\ \theta \\ \Psi \end{bmatrix} = \begin{bmatrix} atan2(2(q_0q_1 + q_2q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2) \\ asin2(q_0q_2 - q_1q_3) \\ atan2(2(q_0q_3 + q_1q_2), q_0^2 + q_1^2 - q_2^2 - q_3^2) \end{bmatrix} \tag{2.35}$$

$$\dot{q} = \frac{1}{2} q \times (0\ \omega)^T \tag{2.36}$$

2.8 Equations of Motion

In this study, the equations of motion for octocopter have been derived from the Euler-Lagrange method which uses Newton's second law. In this method, some assumptions have been made to simplify the equations. First, the body-fixed frame that is placed to its geometric center. Second, the vehicle body is accepted as a rigid body where distances between parts are accepted as a constant. The external force F is shown in Equation 2.36, torque M is shown in Equation 2.37, linear momentum p is shown in Equation 2.38, and angular momentum L is shown in Equation 2.39. In these equations, abbreviations stand as m is the mass, I is the moment of inertia, ω is the angular velocity, and v is the position vector. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

$$\dot{p} = F \tag{2.37}$$

$$\dot{L} = M \tag{2.38}$$

$$p = mv \tag{2.39}$$

$$L = I\omega \tag{2.40}$$

The equations of motion in the earth-fixed inertial frame have been derived from the first-order time derivative of Equation 2.39 and Equation 2.40 where $\dot{m} = 0$ is assumed. The equations of motion are shown in Equation 2.41 and Equation 2.42. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

$$\dot{v}_Q = \frac{1}{m} F_I \tag{2.41}$$

$$\dot{\omega}_I = I_I^{-1}(M_I - \dot{I}_I \omega_I) \tag{2.42}$$

The equations of motion in the body-fixed frame have been derived with the help of rotation matrices $R_{Q\to B}$. The vector of body-fixed frame v_B is shown in Equation 2.43 and the time derivative process is shown in Equation 2.44 where the time derivative of a rotation matrix $\dot{R}_{Q\to B}$ is shown in Equation 2.25. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

$$v_B = R_{Q\to B} v_I \tag{2.43}$$

$$\dot{v}_B = \dot{R}_{Q\to B} v_I + R_{Q\to B} \dot{v}_I = -\omega_B \times v_B + \frac{1}{m} F_B \tag{2.44}$$

The angular momentum ω_B has been derived in the same way shown above for body-fixed frames. These equations are shown in Equations 2.45 and 2.46. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

$$L_B = R_{Q\to B} L_Q \tag{2.45}$$

$$\dot{L}_B = \dot{R}_{Q\to B} L_Q + R_{Q\to B} \dot{L}_Q = -\omega_B \times L_B + M_B \tag{2.46}$$

The time derivative of angular momentum in the body-fixed frame has been derived in the same way in Equation 2.42 which is shown in Equation 2.47. The final equations of motion are shown in Equation 2.48 and Equation 2.49. In addition, the time derivative of inertia of the body-fixed frame $\dot{I}_B$ is equal to zero. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

$$\dot{L}_B = \dot{I}_B \omega_B + I_B \dot{\omega}_B = I_B \dot{\omega}_B \tag{2.47}$$

$$\dot{v}_B = \frac{1}{m} F_B - \omega_B \times v_B \tag{2.48}$$

$$\dot{\omega}_B = I_B^{-1}(M_B - \omega_B \times I_B \omega_B) \tag{2.49}$$

2.9 Motor Forces and Torques

The octocopter model has been designed with constant arm length. The odd-numbered rotors turn clockwise and even-numbered rotors are turning counterclockwise. The propellers produce a vertical force on z axis with the help of rotor turns. In addition, the turn type of each rotor produces a moment around the center of the body-fixed coordinate system. The placement of forces, the rotor turns, and angular velocities are shown in Figure 2.10. The position of the geometric center of rotors in body-fixed coordinates is shown in Equations 2.50 to 2.57. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

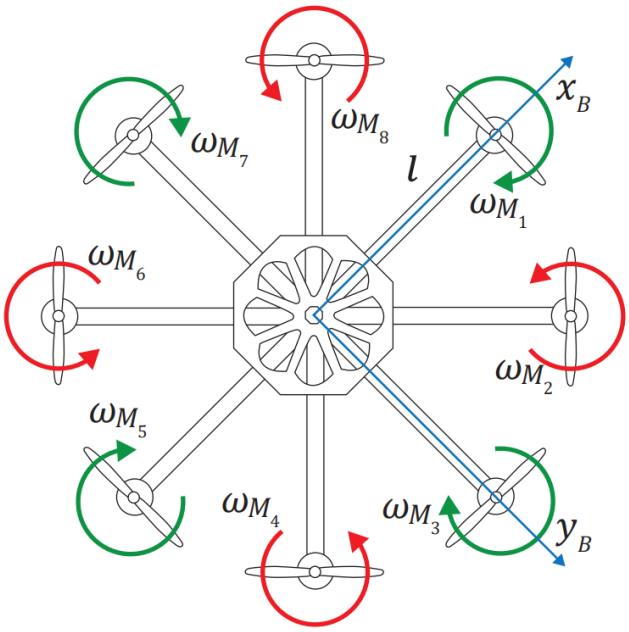

Figure 2.10 Motor rotations, angular velocities, and arm length

$$r_{M_1} = (l\ 0\ 0)^T \tag{2.50}$$

$$r_{M_2} = \frac{1}{\sqrt{2}}(l\ l\ 0)^T \tag{2.51}$$

$$r_{M_3} = (0\ l\ 0)^T \tag{2.52}$$

$$r_{M_4} = \frac{1}{\sqrt{2}}(-l\ l\ 0)^T \tag{2.53}$$

$$r_{M_5} = (-l\ 0\ 0)^T \tag{2.54}$$

$$r_{M_6} = \frac{1}{\sqrt{2}}(-l\ -l\ 0)^T \tag{2.55}$$

$$r_{M_7} = (0\ -l\ 0)^T \tag{2.56}$$

$$r_{M_8} = \frac{1}{\sqrt{2}}(l\ -l\ 0)^T \tag{2.57}$$

The generated force and torques produced from each rotor are shown in Equations 2.58 and 2.59 where c_T, c_Q are thrust constant and torque constant. The generated forces act on octocopter is shown in Equation 2.60. The generated torque act on the octocopter is shown in Equation 2.61. Note that on the torque calculation, the counter-clockwise turns generate counter torques, and clockwise turns to generate positive torques. i is the rotor number. (Fahimi, 2009; Nonami, et al., 2010; Oscarson, 2015; Quan, 2017)

$$T_{M_i} = c_T \omega_{M_i}^2 \tag{2.58}$$

$$Q_{M_i} = c_Q \omega_{M_i}^2 \tag{2.59}$$

$$F_M = \sum_{i=1}^{8} T_{M_i} \tag{2.60}$$

$$M_M = \sum_{i=1}^{8} r_{M_i} \times T_{M_i} + \sum_{i=1}^{8} (-1)^{i+1} Q_{M_i} \tag{2.61}$$

2.10 Control Algorithms

In this study, the feedback control methodology has been used to control the attitude and altitude maneuvers of the octocopter. In this method, subtracting the sensor value from the reference input which is called the error value, has been processed with a control algorithm. In order to drive the octocopter, the control output has been used to feed the system from actuators. The actuator output, also known as the force generated by rotors and propellers, moves the vehicle in space. In this manner, the displacement of the octocopter in every position change has been calculated via sensors that feed the system back. The generalized feedback control is shown in Figure 2.11. (Visioli, 2006)

In this study, the proposed control algorithms have been explained in detail on the following topics which are the PID control algorithm and NARX NN control algorithm. The neural network algorithms consist of two processes. The first one is the training process, and the second one is the collection of training data. In this study, the Levenberg-Marquardt algorithm has been used to train the NARX NN algorithm. In addition Gaussian noise has been used to make a variation in data.

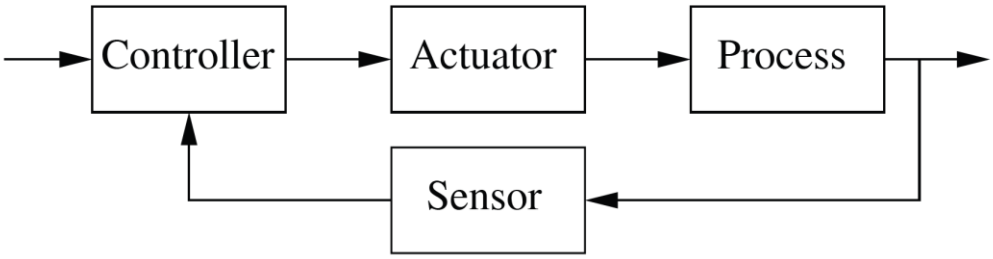

Figure 2.11 Feedback control (Visioli, 2006)

2.10.1PID Control Algorithm

The PID control is an input regulator that is used to set rise time, settling time, overshoot, and undershoot in the feedback control system. In this study, the sum of all three actions of PID has been used to set desired outputs. These actions are proportional action, derivative action, and integral action. The generalized PID algorithm is shown in Figure 2.12. (Visioli, 2006)

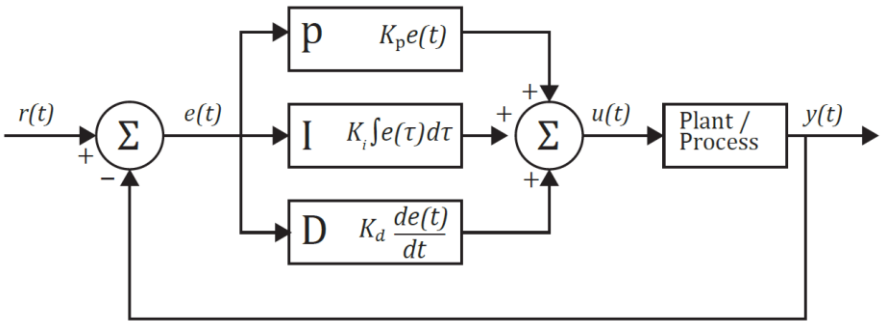

Figure 2.12 Flowchart of PID control

First, the proportional action works as a multiplier of input error. The transfer function for proportional action is shown in Equation 2.61 and the control function is shown in Equation 2.62. Second, the integral action increases the action with the positive error until the error reaches zero. The transfer function for integral action is shown in Equation 2.63 and the control function is shown in Equation 2.64. Third, the derivative action sets the slope of error over time. The transfer function for derivative action is shown in Equation 2.65 and the control function is shown in Equation 2.66. To summarise, the PID algorithm transfer function is shown in Equation 2.67. (Visioli, 2006)

$$C(s) = K_p \tag{2.61}$$

$$u(t) = K_p e(t) \tag{2.62}$$

$$C(s) = \frac{K_i}{s} = K_p \left(1 + \frac{1}{T_i s}\right) \tag{2.63}$$

$$u(t) = K_i \int_0^t e(\tau) d\tau \tag{2.64}$$

$$C(s) = K_d s \tag{2.65}$$

$$u(t) = K_d \frac{de(t)}{dt} = K_p \left(e(t) + T_d \frac{de(t)}{dt}\right) \tag{2.66}$$

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \tag{2.67}$$

The PID coefficients have been tuned with the help of the Ziegler-Nichols method which is a trial-error method. In this method, the proportional gain K_u is increased until constant oscillation than finding oscillation period T_u. Finally, coefficients of PID can be found with the help of Table 1. (Visioli, 2006)

Table 2.1 Ziegler-Nichols method coefficients for different approximations

Control Type	K_p	T_i	T_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$T_u/1.2$	-
PD	$0.8K_u$	-	$T_u/8$
classic PID	$0.6K_u$	$T_u/1.2$	$T_u/8$
Pessen Integral Rule	$0.7K_u$	$T_u/2.5$	$3T_u/20$
some overshoot	$0.33K_u$	$T_u/2$	$T_u/3$
no overshoot	$0.2K_u$	$T_u/2$	$T_u/3$

2.10.2 NARX Neural Network Control

The non-linear autoregressive exogenous (NARX) neural network (NN) is a multi-layer neural-based solver that has self-learning and updating features. This algorithm process past values of time series, and current and past values of exogenous series to predict the outcome. The generalized NARX NN architecture is shown in Figure 2.13 where the TDi is the total number of inputs and TDL is the total number of outputs. The NARX NN algorithm is shown in Equation 2.68 where the $y(t)$ is output, $y(t-d)$ is previous output, $x(t)$ is input, $x(t-d)$ is previous input, and d is the delay number. (Tijani, et al., 2014; El Hamidi, et al., 2020)

$$y(t) = f(x(t-1), \dots, x(t-d), y(t-1), \dots, y(t-d)) \tag{2.68}$$

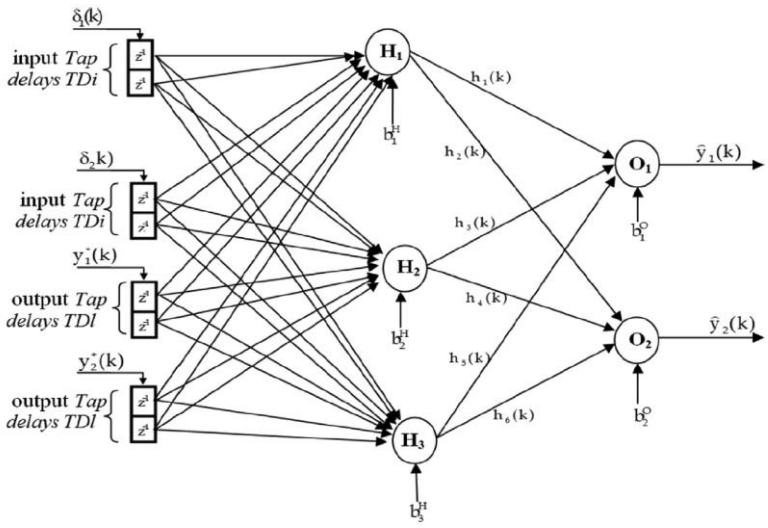

Figure 2.13 The NARX NN architecture (Tijani, et al., 2014)

2.10.3 Levenberg-Marquardt Algorithm

The Levenberg and Marquardt training algorithm is an estimation method for non-linear algebraic equations that is an alternative for least-squares estimation, the Taylor series, and many more. This training method has been derived with the help of the steepest descent and Newton algorithm that is shown in Equation Equation 2.69. Note that, $\Delta\underline{x}$ is the weight vector, J is the Jacobian matrix, μ is an editable parameter, and e is the error vector. (Levenberg, 1944; Marquardt, 1963; Hagan & Menhaj, 1994)

$$\Delta\underline{x} = \left[J^T(\underline{x})J(\underline{x}) + \mu I\right]^{-1} J^T(\underline{x})e(\underline{x}) \tag{2.69}$$

2.10.4 Gaussian Noise Algorithm

The Gaussian noise is a probability density function (PDF) that is widely used to replicate sensor noise. In this study, the Gaussian noise is applied to the original inputs of the system to create a variety of input and output patterns for NARX NN algorithm training. The random Gaussian variable is shown in Equation 2.70 where p stands for density function, z the grey level, μ mean grey value, and σ standard deviation. (Hagiwara, et al., 2001; Krivec, et al., 2021; Mchutchon & Rasmussen, 2011; Silva & Adeodato, 2011)

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{2.70}$$

CHAPTER THREE
MODEL BASED DESIGN

3.1 Introduction

In this chapter, the model-based design methodology has been mentioned to simulate the control algorithms. This chapter has divided into 4 main topics. First, the methodology behind the model-based design is explained with the flowchart of the system. Second, the 3D model design of the octocopter has been mentioned where it is used to gather the mass properties of the simulated system with the help of Solidworks. Third, the Matlab Simulink model has been proposed for a real-time simulation environment. Fourth, the tuning and training methods have been covered for the PID and NARX NN controllers.

3.2 Methodology

In this study, Matlab Simulink, Simscape, SimMechanics, SimElectronics, and Neural Network toolbox have been used to implement the proposed mathematical model, control algorithms, and 3D octocopter model into the real-time simulation environment. The vehicle parameters and visual model have been designed using Solidworks to find mass properties such as density, mass, surface area, the center of mass, principal axes of inertia, moments of inertia, and products of inertia. The flowchart of the system is shown in Figure 3.1. The automated pilot commands have been sent to the system thus altitude in meter, and roll, pitch, yaw commands in radian. First, the error which is described as the difference between input and output has been calculated to find the next system action. Second, the control system has been processed the error input to create motor inputs. The motor inputs have been used to spin rotors and propellers to create a lifting force on the octocopter. Consequently, the sensors have been used to find the angular and linear displacement of the octocopter. The resultant actions of the system have been observed with the help of the Matlab virtual simulation environment.

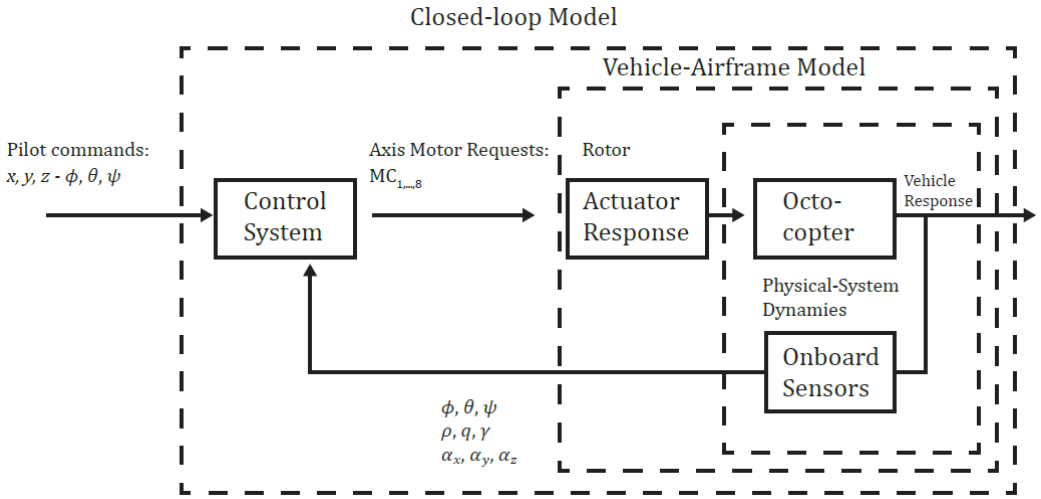

Figure 3.1 Flowchart of the designed system

3.3 Solid Model of the Octocopter

The solid model of the octocopter has been designed to obtain mass properties, joint placements, and visual model of the components and chassis from Solidworks model design in order to implement it on dynamic modeling in Matlab Simulink. The solid model design is based on the dimensional constraints of the Tarot Multirotor Airframe TL1000B01 model. The total mass of the system which includes batteries, the control card, electronic speed controllers, rotors, and chassis has been limited by 8.5 kilograms to decrease the total load needed by each rotor. The endpoint of the wings has been placed 850 mm away from the center of the octocopter. In addition, the center of rotors has been placed 797 mm away from the center. The total height of the vehicle has been designed as 210 mm distance between the peak point and bottom point. In order to avoid the balancing problem, the center of gravity of the system has been designed to intersect with the origin of the system. As a result, the octocopter has been designed counterpoised dimensional axis to suppress affects the balance on the flight by the mass moment of inertia about x, y, z axis "$I_{xx}, I_{xy}, I_{xz}, I_{yx}, I_{yy}, I_{yz}, I_{zx}, I_{zy}, I_{zz}$". The mass properties obtained after the material assignments of the assembled system are shown in Table 2. The visual of the designed system is shown in Figure 3.2.

Table 3.1 Mass properties of the octocopter system

Properties	Symbols	Unit	Value
Total Mass	m	kg	6.215
Gravity	g	$m.s^{-2}$	9.81
Distance between the octocopter center and the motors	l	m	0.797
Mass moment of inertia about xx axis	I_{xx}	$kg.m^2$	21.67
Mass moment of inertia about yy axis	I_{yy}	$kg.m^2$	21.76
Mass moment of inertia about zz axis	I_{zz}	$kg.m^2$	2.59
Mass moment of inertia about xy-yx axis	I_{xy}, I_{yx}	$kg.m^2$	0.06
Mass moment of inertia about xz-zx axis	I_{xz}, I_{zx}	$kg.m^2$	-1.58
Mass moment of inertia about yz-zy axis	I_{yz}, I_{zy}	$kg.m^2$	-0.75



Figure 3.2 Rendered octocopter model

3.4 Model-Based Design

The corresponding mathematical model has been designed under five main systems that are shown in Figure 3.3. First, the maneuver commands have been placed under the reference inputs subsystem. To illustrate, the inputs are provided by signal builder blocks to create Gaussian noise sequence, square wave inputs, sinusoidal wave inputs, and sawtooth wave inputs. Second, the error values for altitude and attitude maneuvers have been placed under the GetErrors subsystem. To clarify, the error values have been calculated by subtracting the sensor outputs from the reference inputs. Third, the process of control algorithms for each maneuver and

command distribution for each rotor has been calculated under this subsystem. Fourth, the connection between body-fixed frame and earth-fixed frame, and sensor values have been covered under the Base and Sensors subsystem. Fifth, the parts of the octocopter such as rotors, propellers, and connection joints, RPM and DAC generator, H-bridge, DC rotor, and propeller thrust generators have been placed under the Octocopter subsystem.

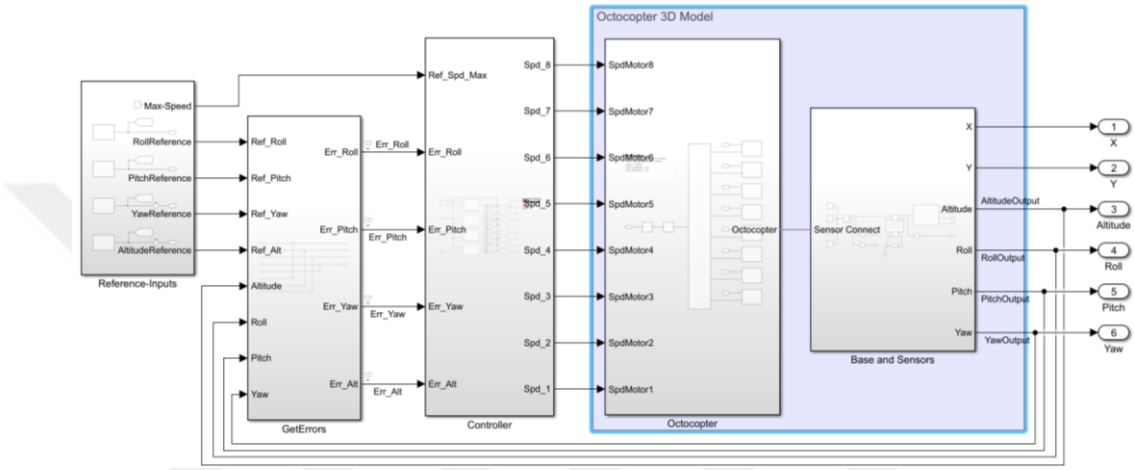

Figure 3.3 Model-based design of the Octocopter

The sensor value calculations for attitude, altitude, and position of the system have been placed between the earth-fixed frame and body-fixed frame. The earth-fixed frame and ground have been connected to each other directly. It follows that the sensor values have been calculated with the help of joints. The x, y position of the vehicle has been achieved by rectangular joint. The displacement on z axis has been calculated by prismatic joint where the hard stop is used to prevent falling lower than zero meters. These are shown in Figure 3.4. Likewise, the attitude calculations have been designed in the same way that is shown in Figure 3.5. Accordingly, the angular displacements have been achieved with the help of revolute joints. The positional change on the revolute joint has been used to calculate the change in angle for the specific attitude. In addition, there are some transformations have been implemented between the joints to divert the coordinates in order to calculate other maneuvers. The earth-fixed frame has been connected to the body-fixed frame with Sensor Connect port after the sensor calculations and reverts the changes on the coordinate system with transformation blocks.

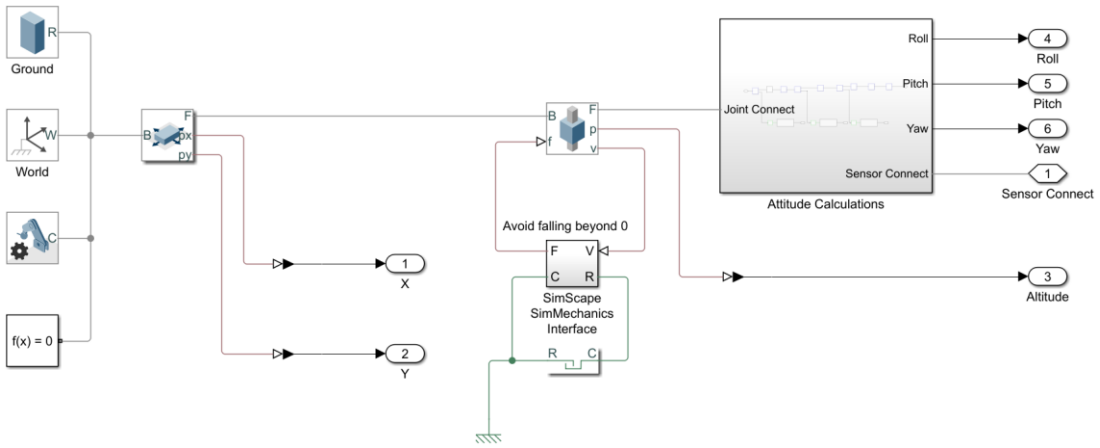

Figure 3.4 The earth-fixed frame, x, y location, and altitude sensor

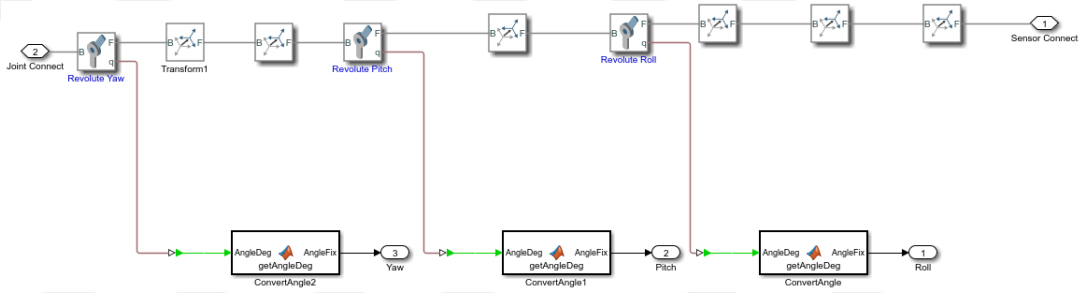

Figure 3.5 Attitude sensor values and the body-fixed frame connection

The octocopter has been placed as a plus placement where the first rotor is located on the positive x axis and the third rotor is located on the positive y axis. The Sensor Connect port which is mentioned before has been connected as an Octocopter port that is directly connected to the body-fixed frame. In the Octocopter_Core subsystem, the Simmechanics blocks and reference frames have been covered where the mass properties, dimensions, virtual models, moments of inertia for each component, placements, and connections between each part are implemented. In addition, the rotors and propellers have been connected to the arms of the octocopter in the Rotor_Propeller subsystems. Moreover, the actuation and force generations have been implemented in this subsystem. The Octocopter subsystem is shown in Figure 3.6.

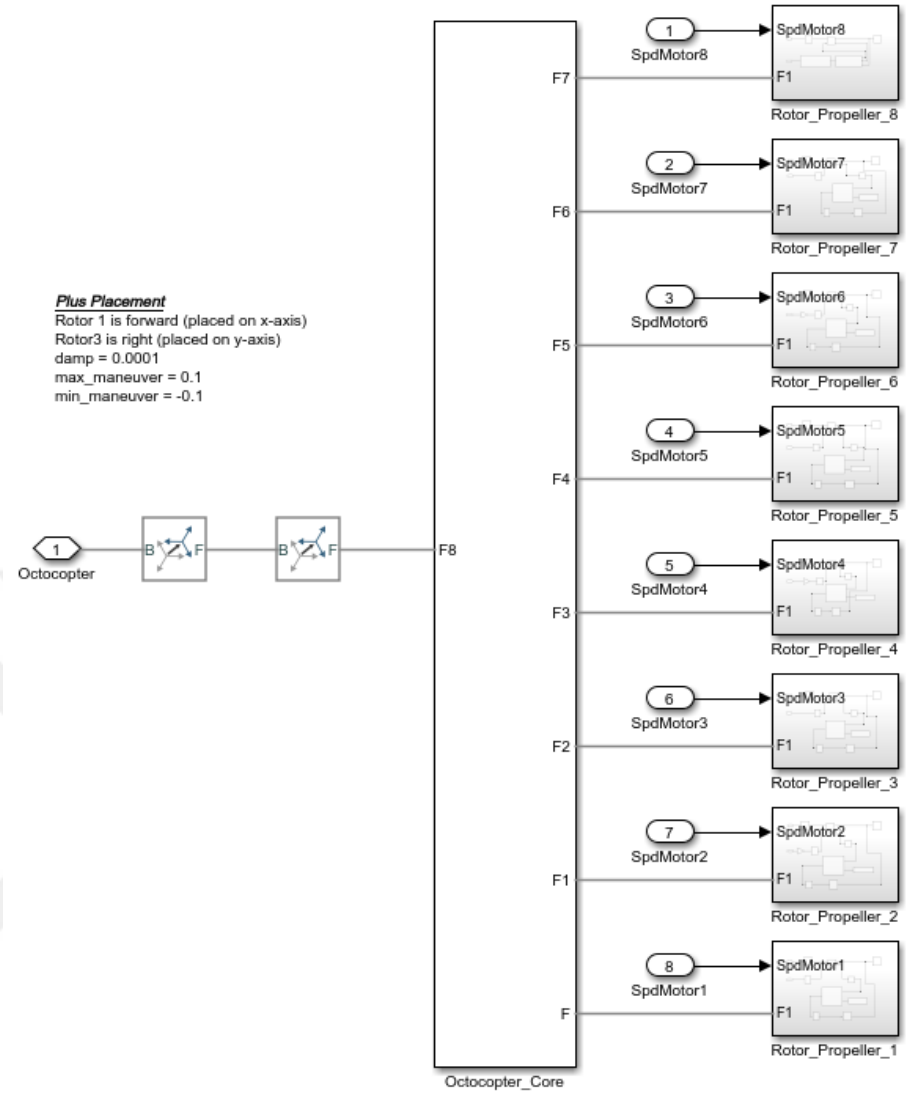

Figure 3.6 The octocopter components and the body-fixed frame

The Rotor_Propeller subsystem contains the connection port that is attached to the rotor and body frame, revolute joint to transfer the rotational movement and moment into the system, propeller model, pulse width modulation (PWM) generator, and Simelectronics rotor model which is shown in Figure 3.7. The RPM and DAC generator has been used to convert the saturated motor commands into the duty and revolution signal which is required to control the DC-rotor. First, generated signals are converted into electrical signals via converters. Second, the duty signal is used to generate controlled PWM voltage. Third, an H-Bridge is used to drive the DC rotor with the help of provided signals. Fourth, the DC-rotor has been used to convert the electrical source into the mechanical angular velocity. Fifth, mechanical rotation has

been realized with the help of Angular Velocity Source to create torque. Sixth, the torque generated by the DC-rotor has been used to send the Revolute Joint in order to generate angular velocity which is used to feed the feedback rotor system and generate force. The angular velocity provided by the rotor has been used to generate force. The rotor model is shown in Figure 3.8.

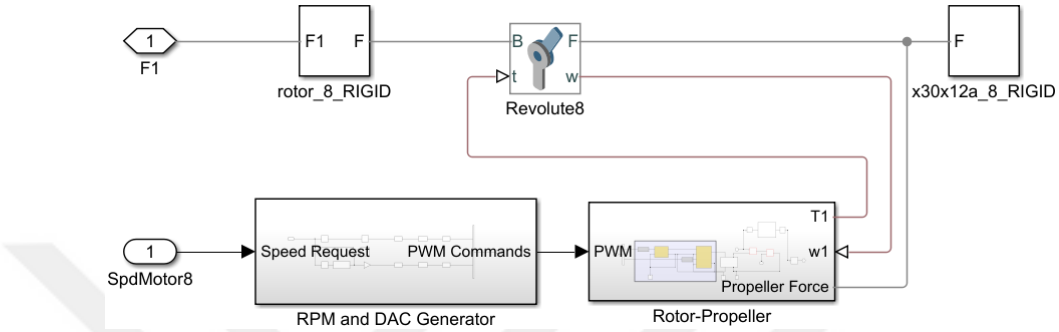

Figure 3.7 Rotor and propeller connection

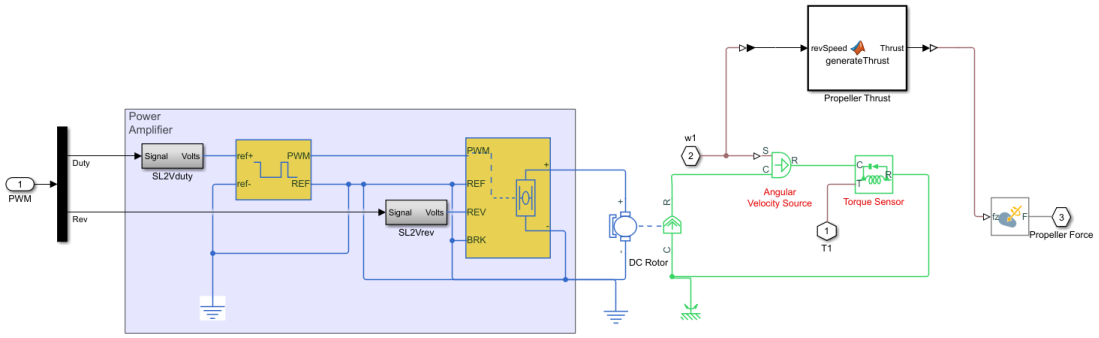

Figure 3.8 Detailed rotor and propeller design

The electronic parts have been designed with some assumptions to increase the quality of outputs. The voltage has been selected as an 11.2 V constant voltage source. The enable, reverse and braking thresholds have been limited with 9 V, and the amplitude of the PWM signal has been provided with 11.2 V on the H-Bridge. In addition, the output voltage has been switched on 10 V with a very low amount of resistances on the H-bridge. The DC rotor properties have been selected as 15000 rpm no-load speed, 20 W mechanical power, 10 V rated DC supply voltage, and neglected the other effects such as armature inductance, rotor damping ratios. To summarize, the force production of the propellers has been provided from the

performance data of the APCPROP catalog. The resultant force has been chosen force values from 4000rpm to 15000 rpm with 20W power.

3.5 Model-Based Design of Attitude and Altitude Controllers

The control algorithms have been implemented separately for each maneuver which is shown in Figure 3.9. The PID and NARX NN algorithms have been used to control each rotor command in the Control subsystems. The motor command has been limited between resting point and maximum rotor speed. The maneuver commands have been simplified in a plus placement where the altitude is achieved by all motors directly. The pitch maneuver has been achieved by $1^{th}, 2^{nd}, 8^{th}$ rotors decreased rotor speeds and $4^{th}, 5^{th}, 6^{th}$ rotors increased rotor speeds. The roll maneuver has been achieved by $2^{nd}, 3^{th}, 4^{th}$ rotors decreased rotor speeds and $6^{th}, 7^{th}, 8^{th}$ rotors increased rotor speeds. The yaw maneuver has been achieved by $2^{nd}, 4^{th}, 6^{th}, 8^{th}$ rotors decreased rotor speeds and $1^{th}, 3^{th}, 5^{th}, 7^{th}$rotors increased rotor speeds where the altitude is maintained. To summarize, the motor command mathematical equations are shown in Section 2.9.

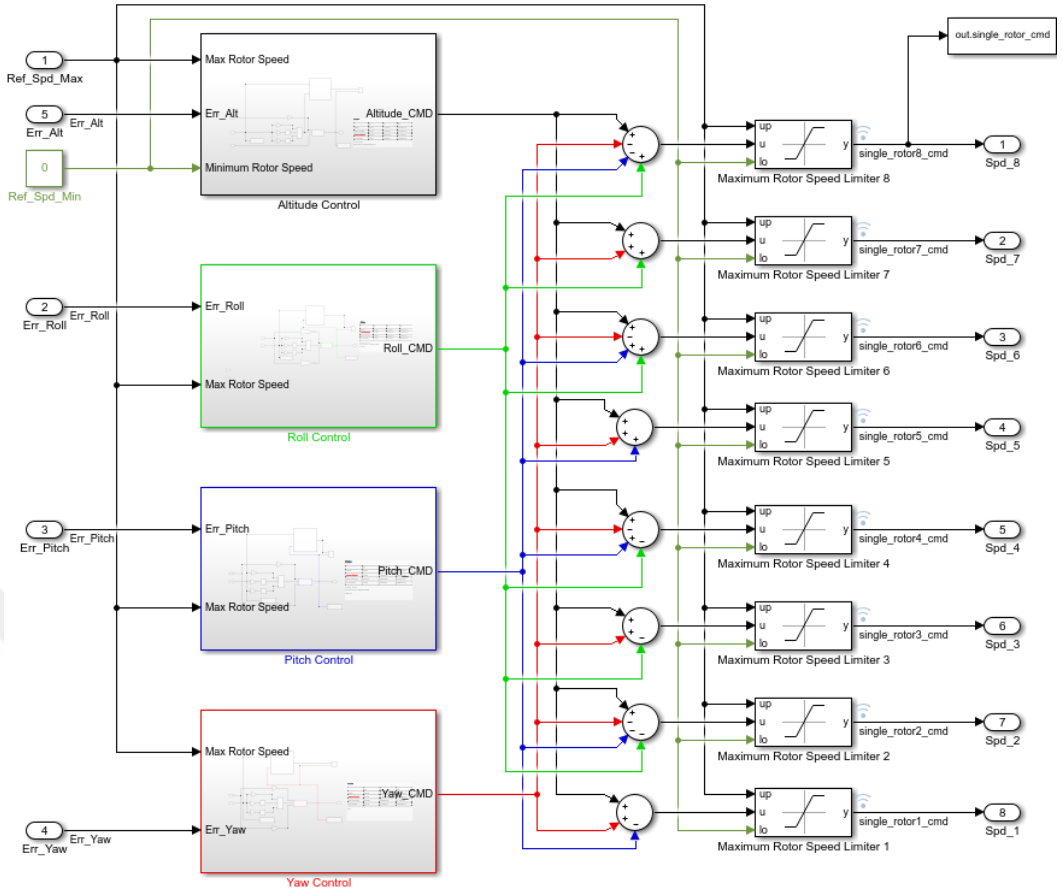

Figure 3.9 Attitude and altitude control system

3.5.1 PID Control and Tuning

The altitude and attitude maneuvers have been controlled with the help of PID algorithms. The dead-time effect has been observed as very small in the simulation environment on tuning the PID controllers. Therefore, the forces generated by propellers have passed over the total weight of the octocopter within 2 to 5 simulation steps on every maneuver where the step size is defined as 1 millisecond. The altitude maneuver control model is shown in Figure 3.10. The PID coefficients for altitude maneuver have been derived with the help of the Ziegler-Nichols method. The ultimate gain and oscillation period have been gathered from Figure 3.11 where $K_u = 75$, and $T_u = 3.9s$.

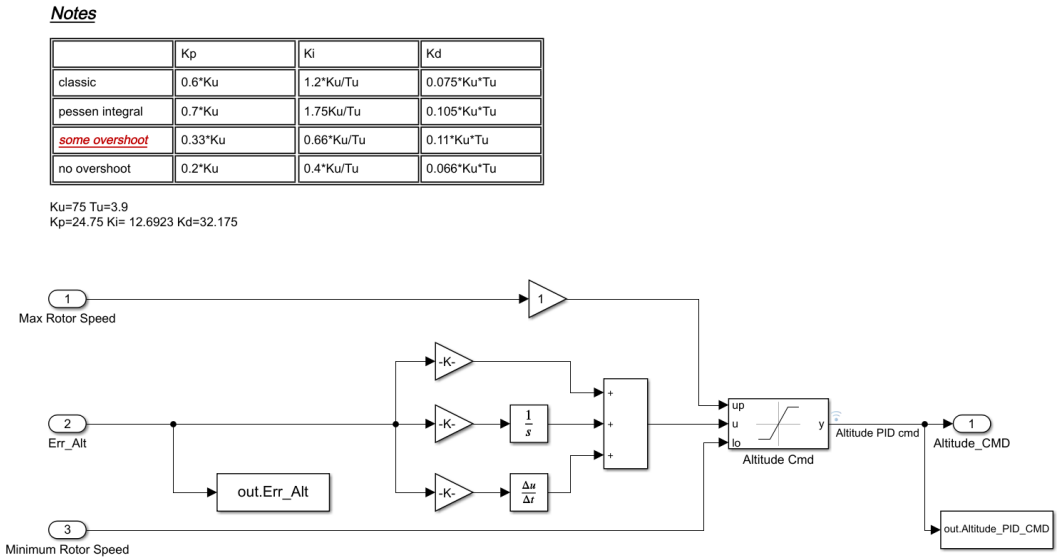

Figure 3.10 Altitude control with PID algorithm

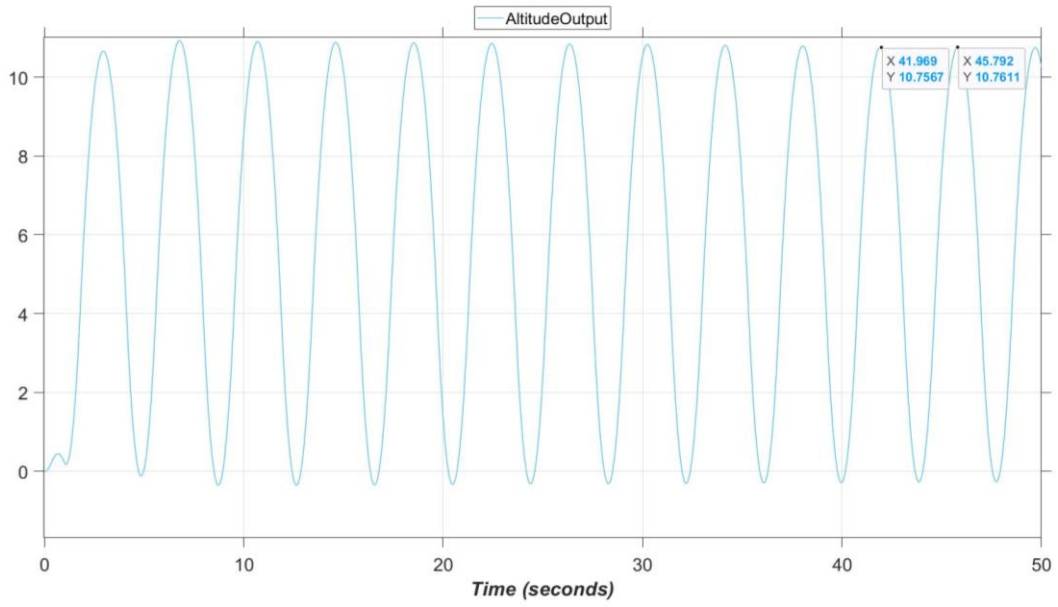

Figure 3.11 The oscillation under ultimate gain K_u for altitude maneuver

The roll maneuver control model is shown in Figure 3.12. The PID coefficients for roll maneuver have been derived with the help of the Ziegler-Nichols method. The ultimate gain and oscillation period have been gathered from Figure 3.13 where $K_u = 12.81$, and $T_u = 1.5s$.

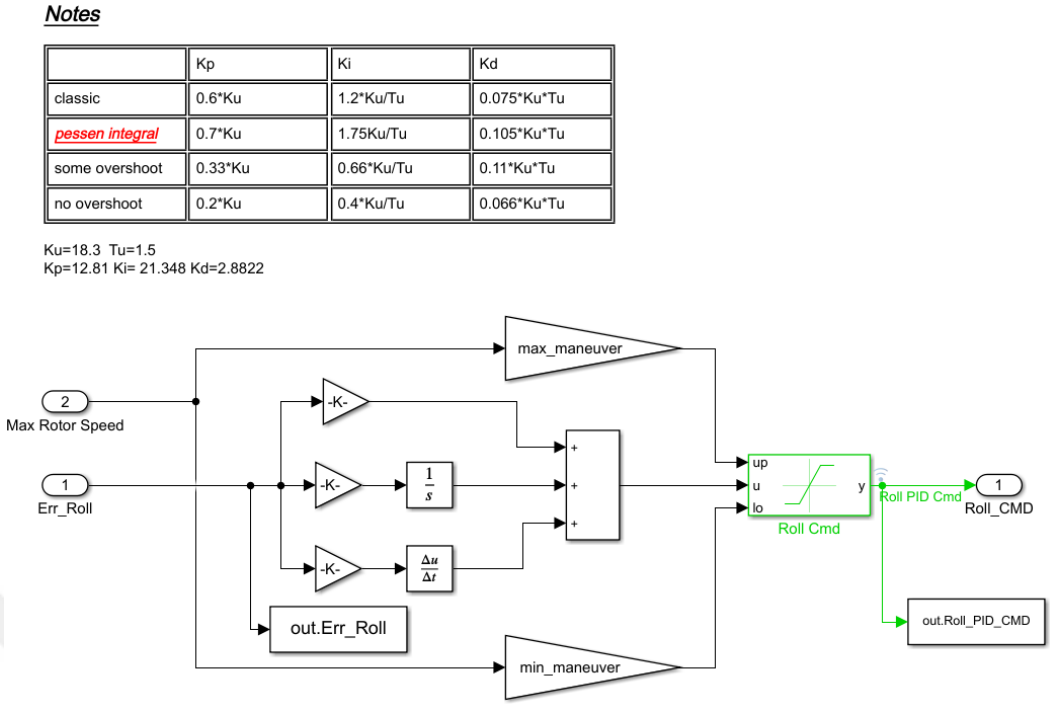

Figure 3.12 Roll maneuver control with PID algorithm

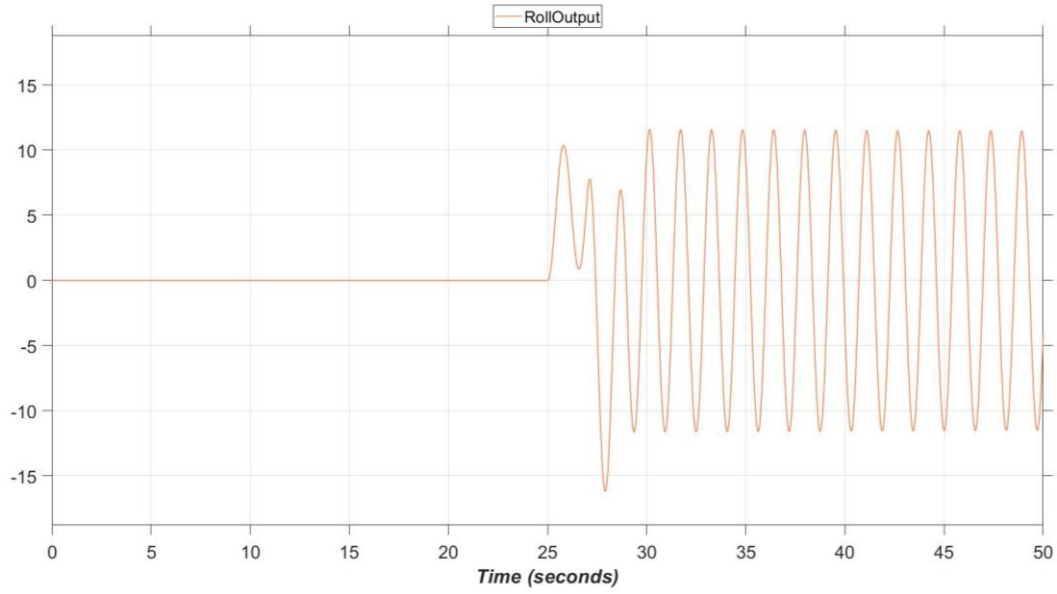

Figure 3.13 The oscillation under ultimate gain K_u for roll maneuver

The pitch maneuver control model is shown in Figure 3.14. The PID coefficients for pitch maneuver have been derived with the help of the Ziegler-Nichols method. The ultimate gain and oscillation period have been gathered from Figure 3.15 where $K_u = 12.81$, and $T_u = 1.5s$.

Notes

	Kp	Ki	Kd
classic	0.6*Ku	1.2*Ku/Tu	0.075*Ku*Tu
pessen integral	0.7*Ku	1.75Ku/Tu	0.105*Ku*Tu
some overshoot	0.33*Ku	0.66*Ku/Tu	0.11*Ku*Tu
no overshoot	0.2*Ku	0.4*Ku/Tu	0.066*Ku*Tu

Ku=18.3 Tu=1.5
Kp=12.81 Ki= 21.348 Kd=2.8822

0.305 1.5

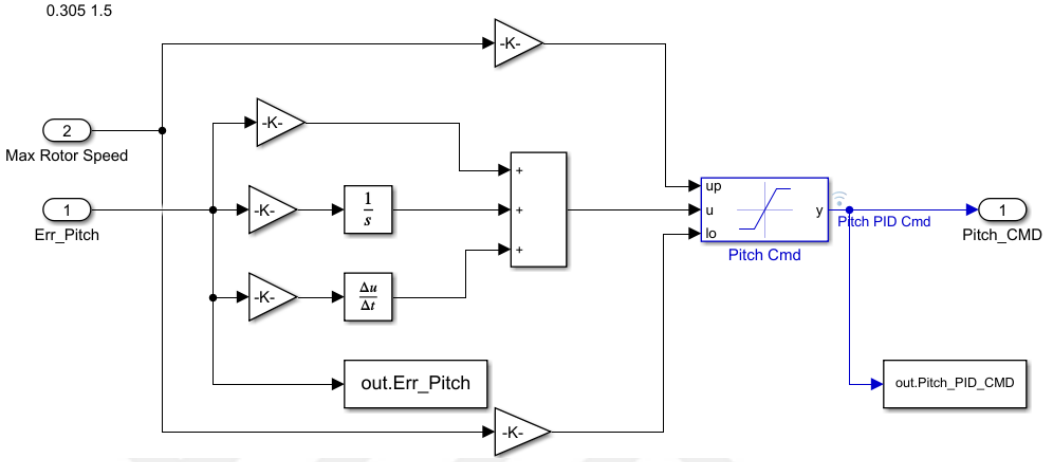

Figure 3.14 Pitch maneuver control with PID algorithm

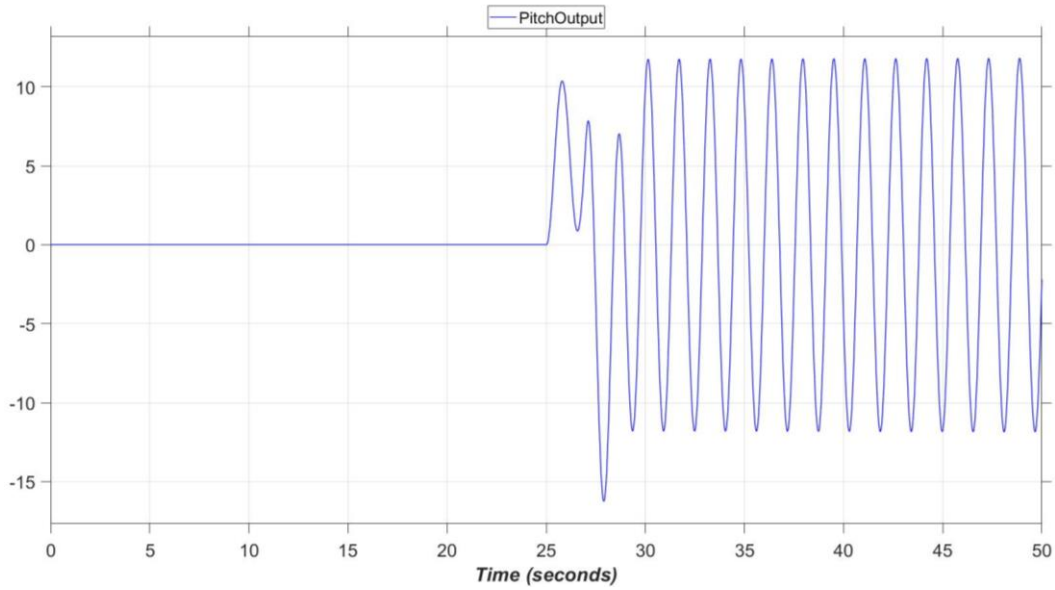

Figure 3.15 The oscillation under ultimate gain K_u for pitch maneuver

The yaw maneuver control model is shown in Figure 3.16 where the maneuver is only controlled by proportional gain.

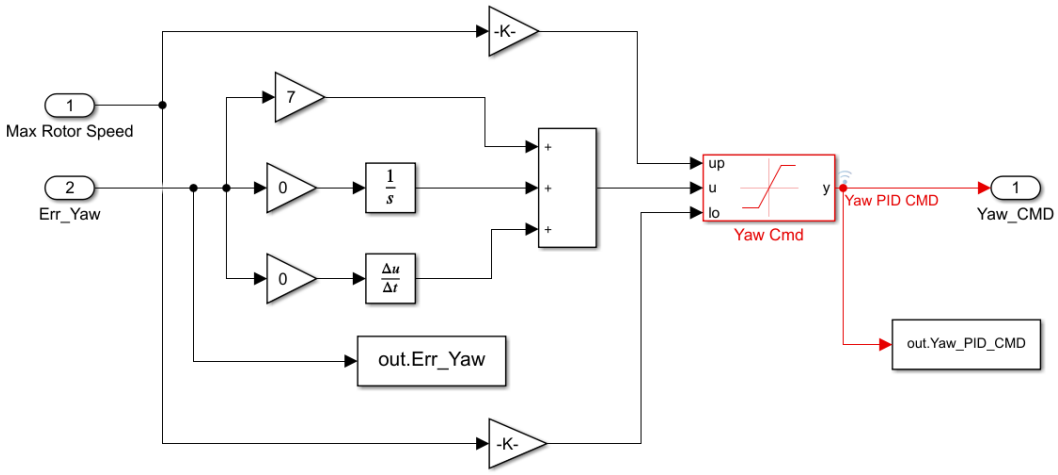

Figure 3.16 Yaw maneuver control with proportional gain

3.5.2 NARX NN Control and Training

The NARX NN algorithm has been used in a feedback architecture which is shown in Figure 3.17. The altitude and attitude maneuvers have been controlled with the same architecture. The NARX NN has been used to predict future outcomes by processing the error values and neural network results where process output has been provided with the help of Equation 2.68. The NARX NN architecture has been designed with 3 layers and 45 neurons for every maneuver control. The Levenberg-Marquardt algorithm has been used to train the neurons. In addition, the training has been limited by 1000 epochs where 70% of the data is used for training, %15 percent of data is used for validation, and the last 15% of data is used for testing. To summarize, randomly selected training data have been presented to the network according to the error values. In addition, randomly selected validation data have been used to measure network generalization where the generalization stops improving 6 times in a row. Moreover, randomly selected testing data has been used to measure training performance independently. The output feedback to the network has been delayed only one step. 300000 total number of inputs and outputs in time-series have been used as input and target data. The NN has been trained again on unsuccessful results where the system validation is not achieved 6 times in a row.

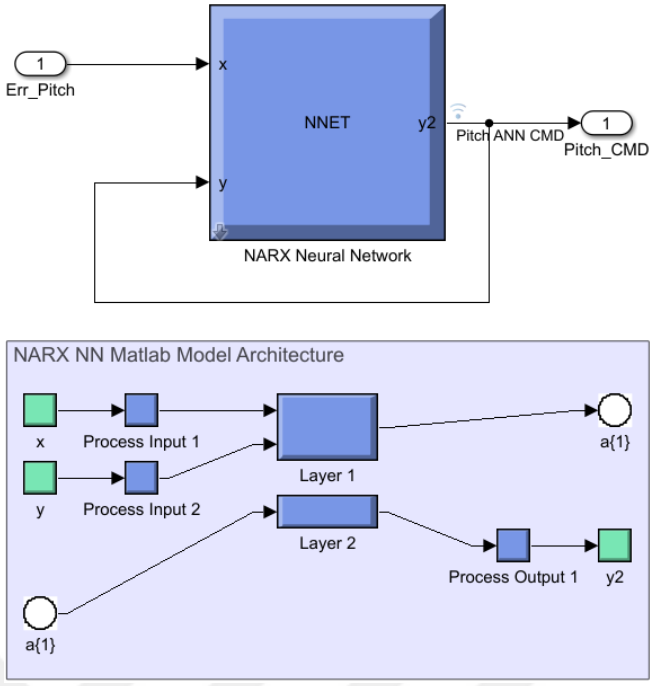

Figure 3.17 The NARX NN control model

The neurons in the layers of the NARX NN algorithm have been trained with the data that are provided from PID control where the NARX NN controller covers PID coefficients and first-level saturation of rotor speeds. The altitude maneuver reference input has been applied with a sine wave, square wave, and randomized step inputs where the Gaussian noise inputs are applied from the beginning to the end. The sine wave has been generated with 0.1 Hz frequency, 2.5 meters offset, and 1-meter amplitude between 0-100 seconds. The square wave has been generated with 0.1 Hz frequency, 2.5 meters offset, and 1-meter amplitude between 100-200 seconds. The random step input has been generated with 0.5 meters to 8 meters amplitude variation as step input for 5 seconds between 200-300 seconds. The Gaussian noise sequence has been generated with 1 Hz frequency and 0.5-meter amplitude. The altitude maneuver reference inputs without Gaussian noise are shown in Figure 3.18, the altitude maneuver reference inputs with Gaussian noise are shown in Figure 3.19, the altitude maneuver outputs of the octocopter are shown in Figure 3.20.

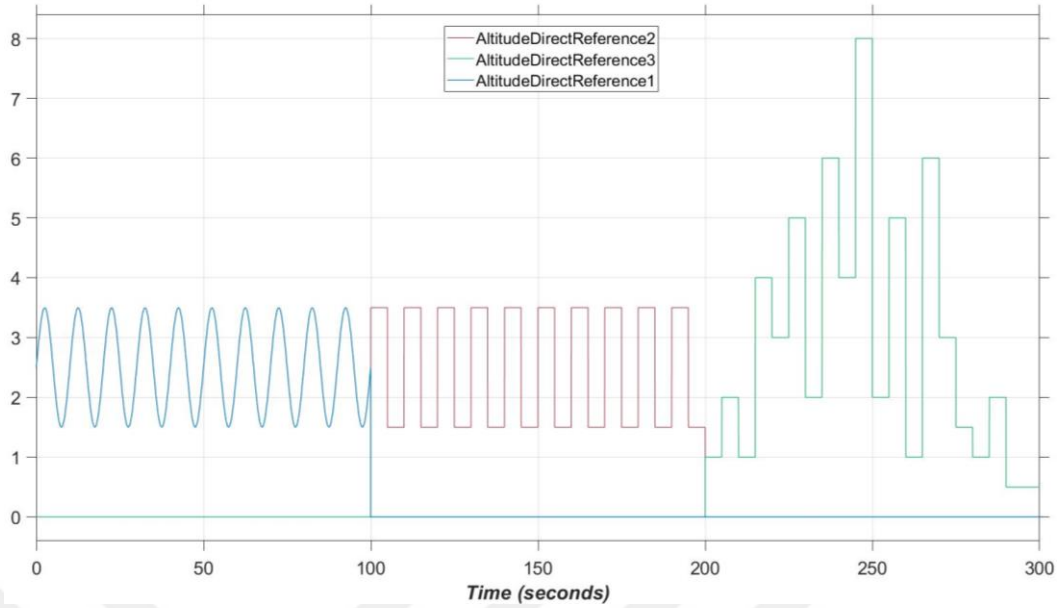

Figure 3.18 Altitude reference inputs without Gaussian noise

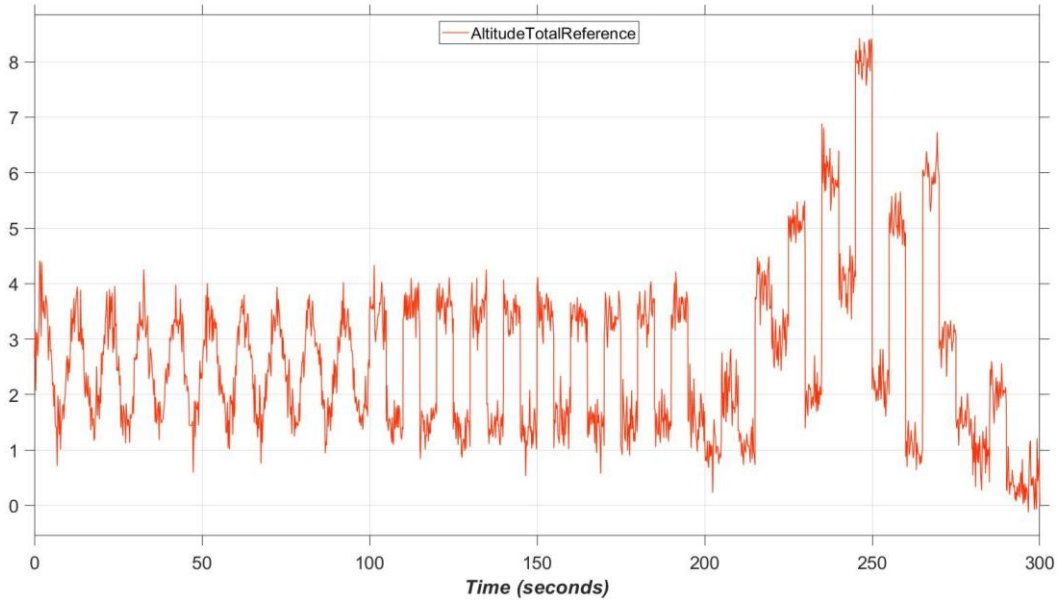

Figure 3.19 Altitude reference inputs with Gaussian noise

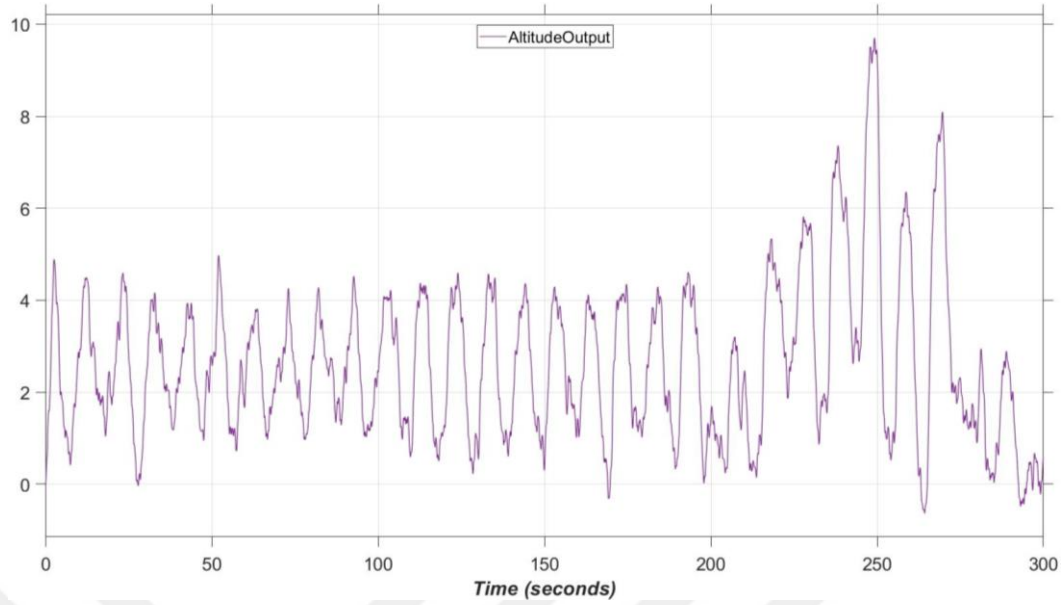

Figure 3.20 Altitude maneuver outputs of octocopter

The roll maneuver reference input has been applied with a sine wave, square wave, and sawtooth step inputs where the Gaussian noise inputs are applied from the beginning to the end. The sine wave has been generated with 0.1 Hz frequency, 5 degrees amplitude between 0-100 seconds. The square wave has been generated with 0.5 Hz frequency, 5 degrees amplitude between 100-200 seconds. The sawtooth wave has been generated with 0.5 Hz frequency, 5 degrees amplitude between 200-300 seconds. The Gaussian noise sequence has been generated with 1 Hz frequency and 2 degrees amplitude. The roll maneuver reference inputs without Gaussian noise are shown in Figure 3.21, the roll maneuver reference inputs with Gaussian noise are shown in Figure 3.22, the roll maneuver outputs of the octocopter are shown in Figure 3.23.

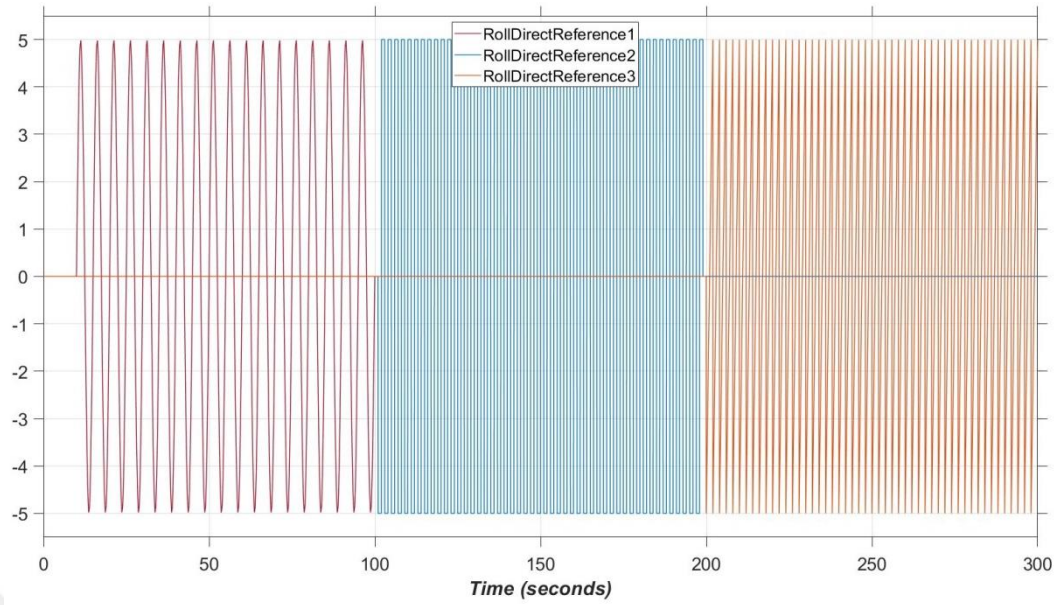

Figure 3.21 Roll reference inputs without Gaussian noise

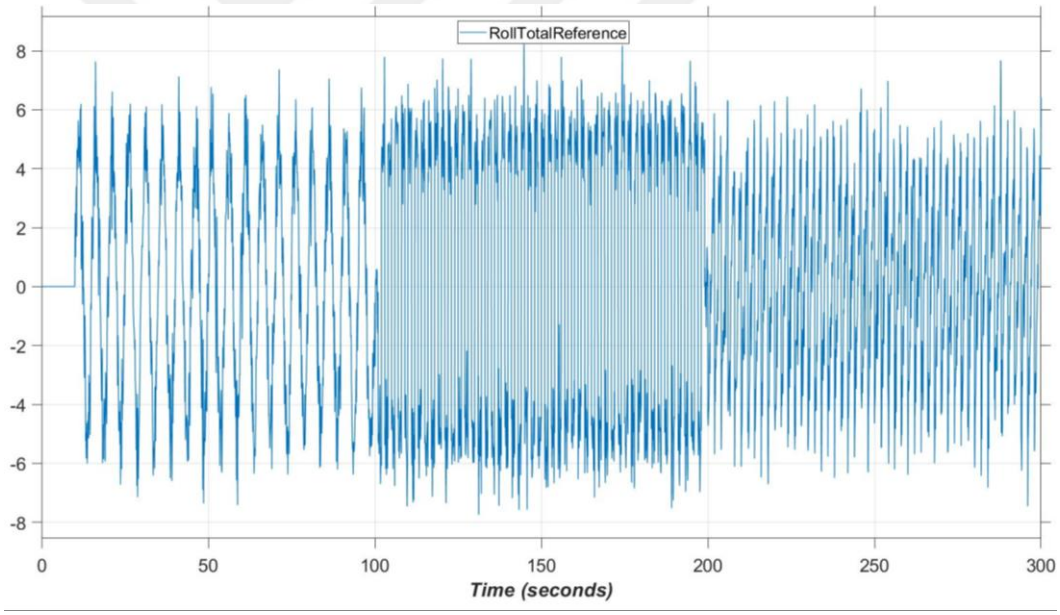

Figure 3.22 Roll reference inputs with Gaussian noise

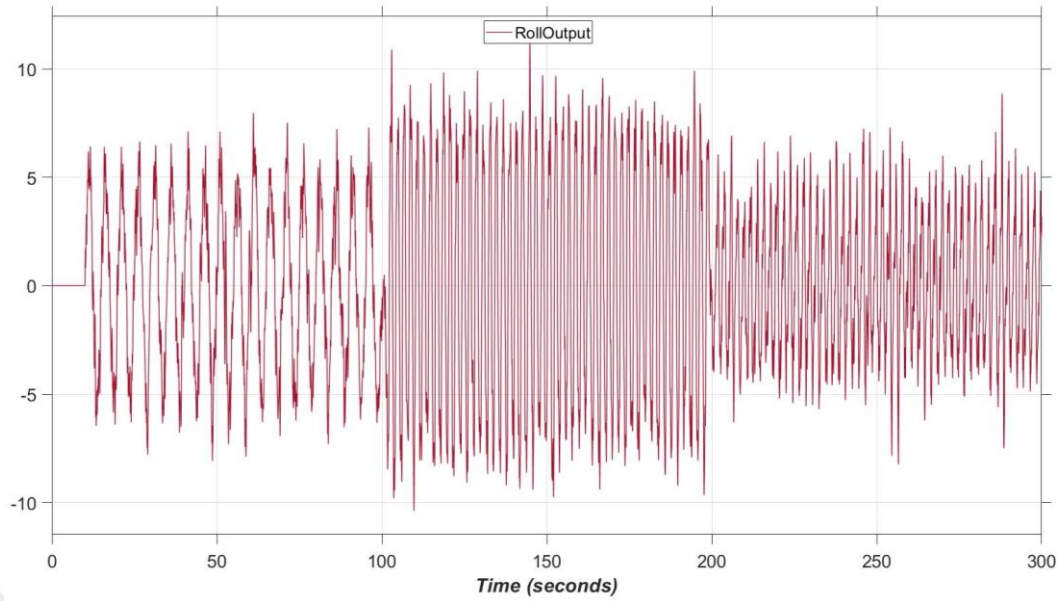

Figure 3.23 Roll maneuver outputs of octocopter

The pitch maneuver reference input has been applied with a sine wave, square wave, and sawtooth step inputs where the Gaussian noise inputs are applied from the beginning to the end. The sine wave has been generated with 0.1 Hz frequency, 5 degrees amplitude between 0-100 seconds. The square wave has been generated with 0.5 Hz frequency, 5 degrees amplitude between 100-200 seconds. The sawtooth wave has been generated with 0.5 Hz frequency, 5 degrees amplitude between 200-300 seconds. The Gaussian noise sequence has been generated with 1 Hz frequency and 2 degrees amplitude. The pitch maneuver reference inputs without Gaussian noise are shown in Figure 3.24, the pitch maneuver reference inputs with Gaussian noise are shown in Figure 3.25, the pitch maneuver outputs of the octocopter are shown in Figure 3.26.

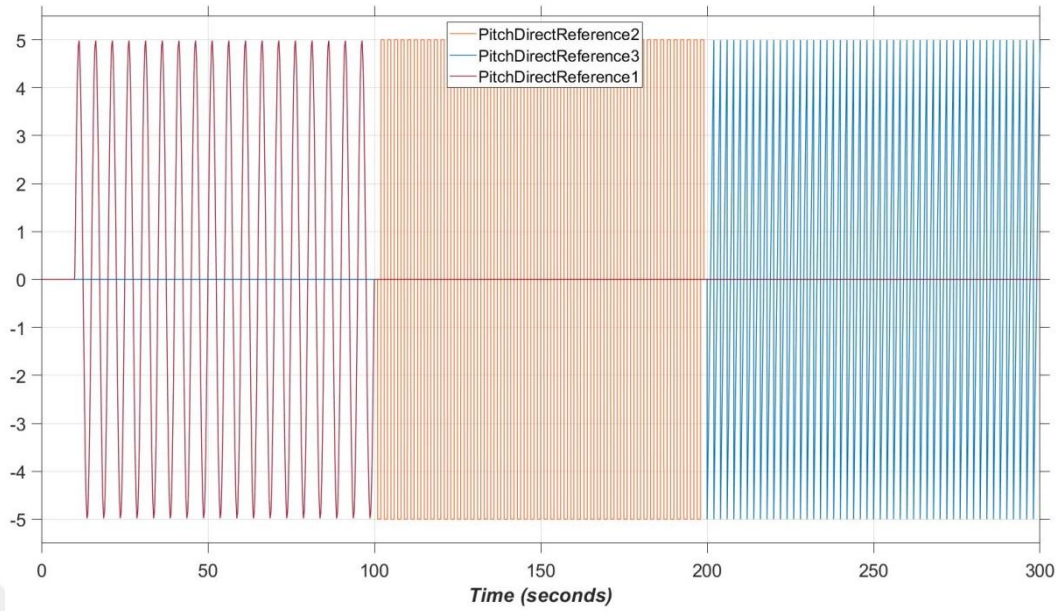

Figure 3.24 Pitch reference inputs without Gaussian noise

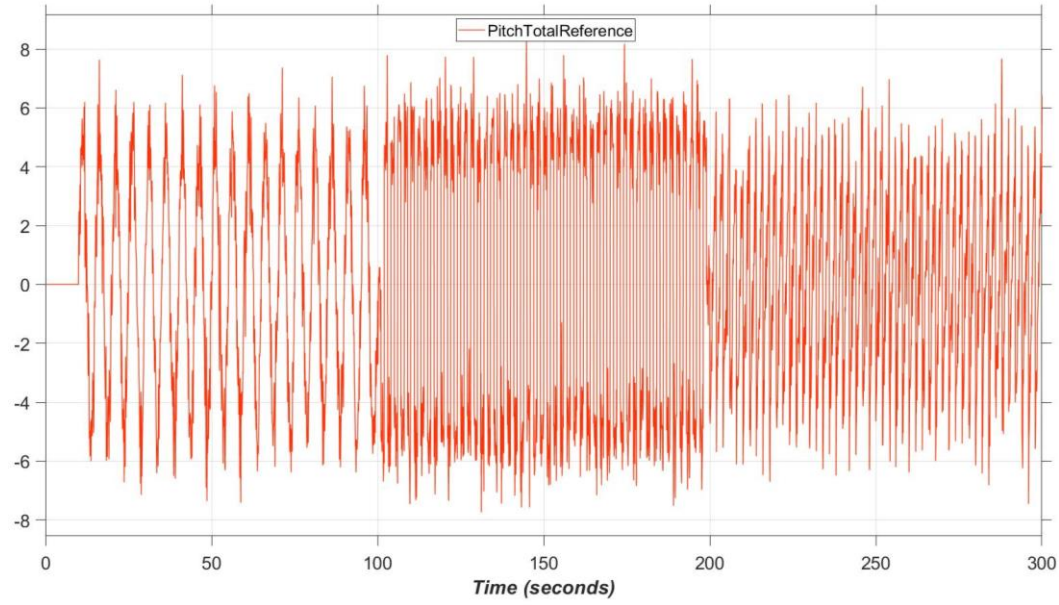

Figure 3.25 Pitch reference inputs with Gaussian noise

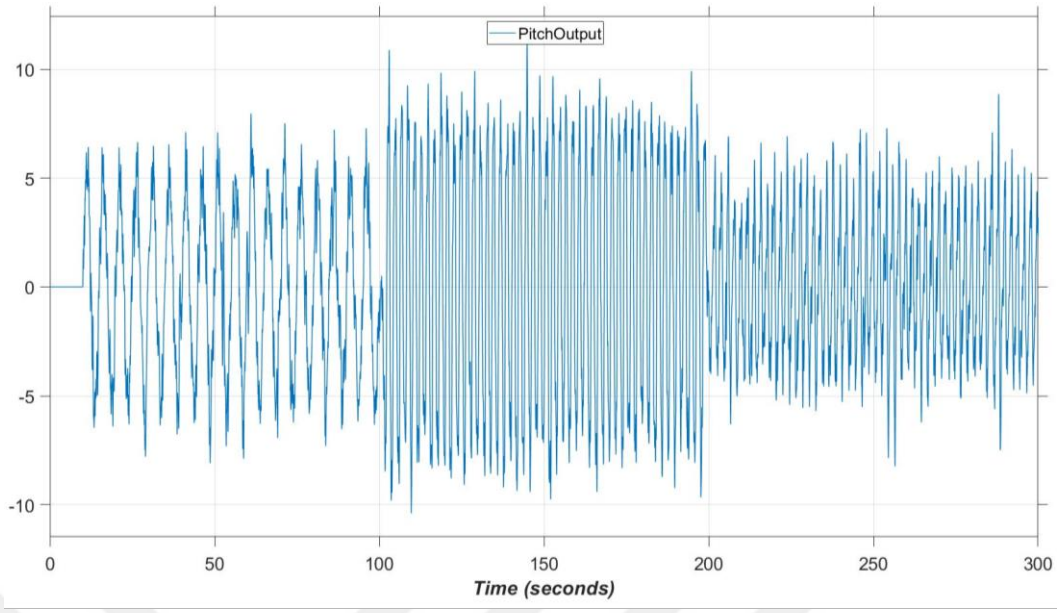

Figure 3.26 Pitch maneuver outputs of octocopter

The yaw maneuver reference input has been applied with a sine wave, square wave, and sawtooth step inputs where the Gaussian noise inputs are applied from the beginning to the end. The sine wave has been generated with 0.1 Hz frequency, 5 degrees amplitude between 0-100 seconds. The square wave has been generated with 0.1 Hz frequency, 5 degrees amplitude between 100-200 seconds. The sawtooth wave has been generated with 0.1 Hz frequency, 5 degrees amplitude between 200-300 seconds. The Gaussian noise sequence has been generated with 1 Hz frequency and 2 degrees amplitude. The yaw maneuver reference inputs without Gaussian noise are shown in Figure 3.27, the yaw maneuver reference inputs with Gaussian noise are shown in Figure 3.28, the yaw maneuver outputs of the octocopter are shown in Figure 3.29.

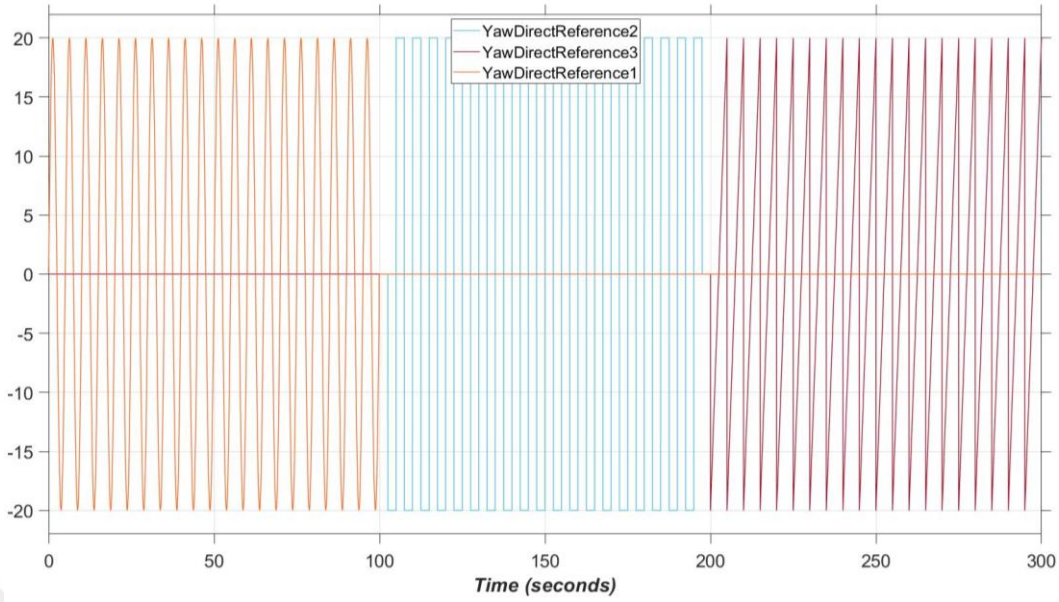

Figure 3.27 Yaw reference inputs without Gaussian noise

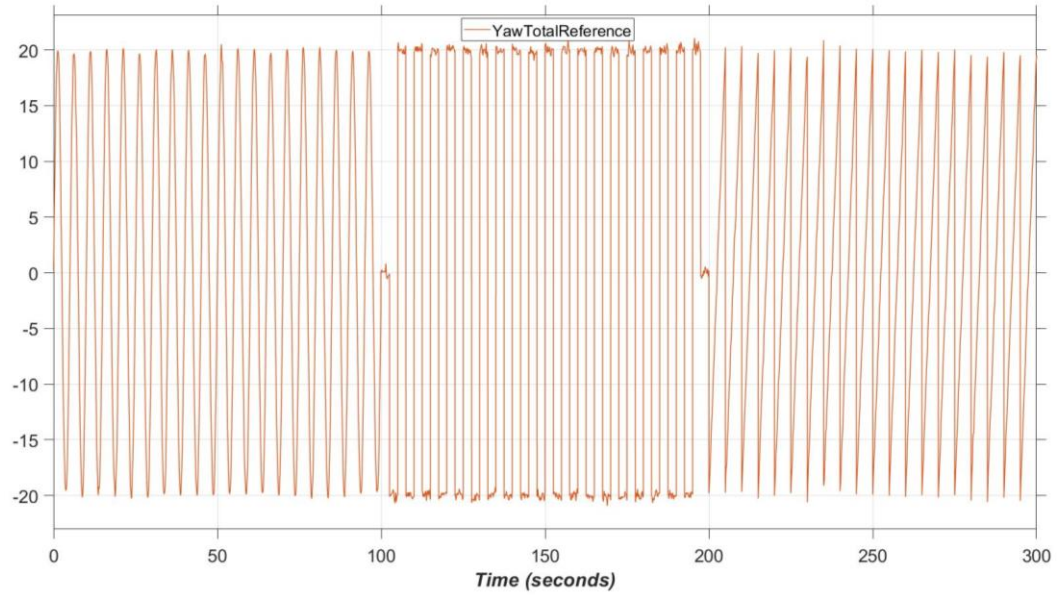

Figure 3.28 Yaw reference inputs with Gaussian noise

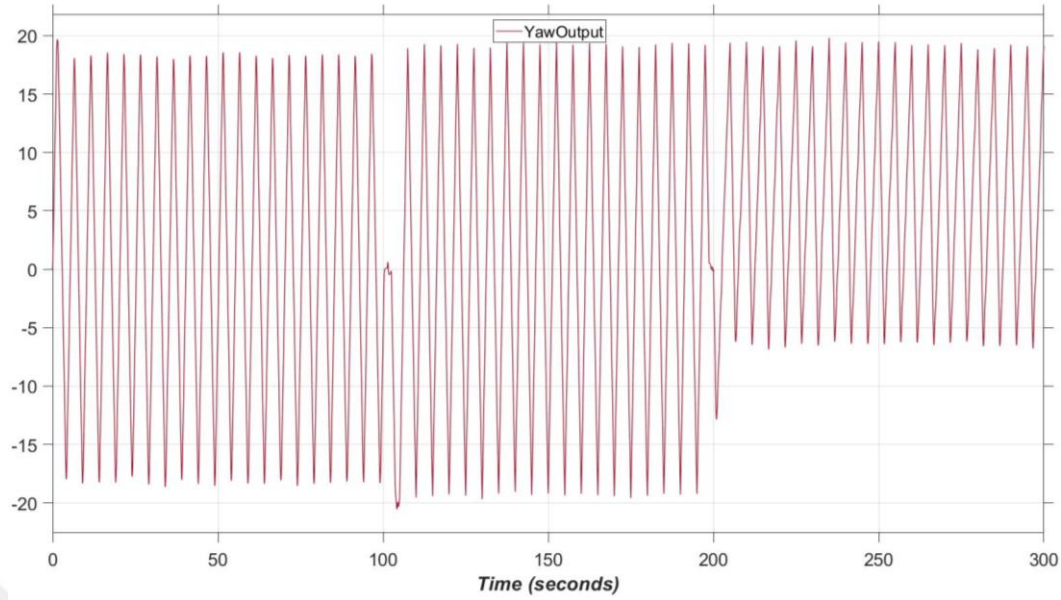

Figure 3.29 Yaw maneuver outputs of octocopter

CHAPTER FOUR
RESULTS

4.1 Introduction

The mathematical model has been simulated on Matlab Simulink. In addition, the altitude and attitude maneuvers have been visualized with the help of the Matlab Mechanics Explorer tool to observe the real-time response of the octocopter which is shown in Figure 4.1. In this chapter, the PID tuning results, the NARX NN training results, and simulation results have been compared for the attitude and altitude maneuver of the octocopter.

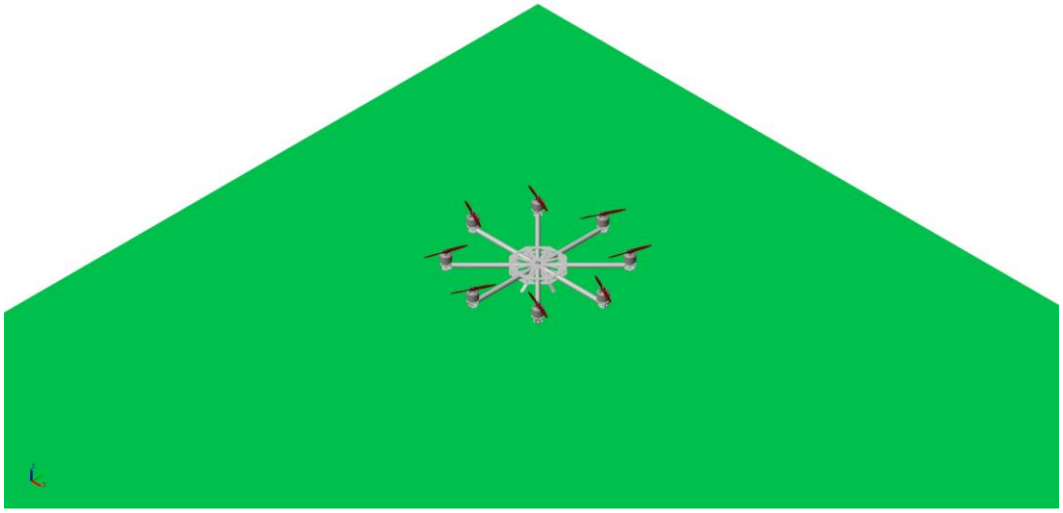

Figure 4.1 Matlab simulation environment

4.2 PID Controller Tuning Results

The PID coefficients have been calculated with the help of the Ziegler-Nichols method where the mathematical model and the tuning parameters are described in Section 2.10.1 and Section 3.5.1. The altitude PID control coefficients have been found with some overshoot as $K_p = 24.5,\ K_i = 12.6923,\ K_d = 32.175$ for ultimate gain is $K_u = 75$ and the oscillation period is $T_u = 3.9s$ where the Pessen integral coefficients are $K_p = 0.33 * K_u,\ K_i = 0.66 * K_u/T_u\,,\ K_d = 0.11\,K_u/T_u.$

The tuned PID altitude maneuver output of the octocopter model is shown in Figure 4.2.

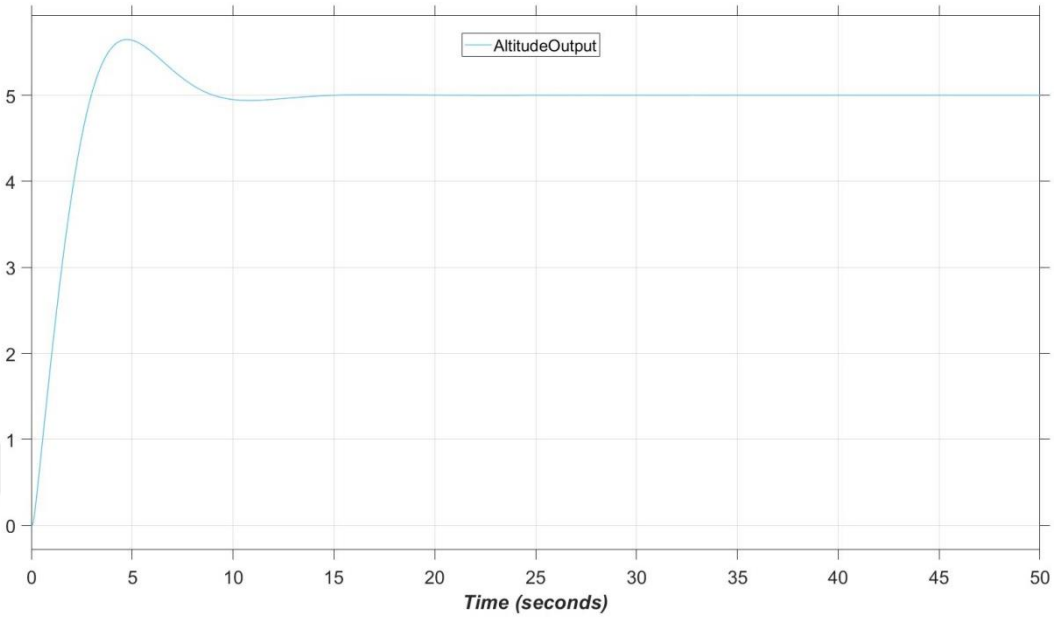

Figure 4.2 The altitude maneuver response for tuned PID coefficients

The roll and pitch PID control coefficients have been found as $K_p = 12.81$, $K_i = 21.348$, $K_d = 2.8822$ for ultimate gain is $K_u = 18$ and the oscillation period is $T_u = 1.5s$ where the Pessen integral coefficients are $K_p = 0.7 * K_u$, $K_i = 1.75 * K_u/T_u$, $K_d = 0.105\, K_u/T_u$. The tuned PID roll and pitch maneuver outputs of the octocopter model are shown in Figure 4.3, Figure 4.4.

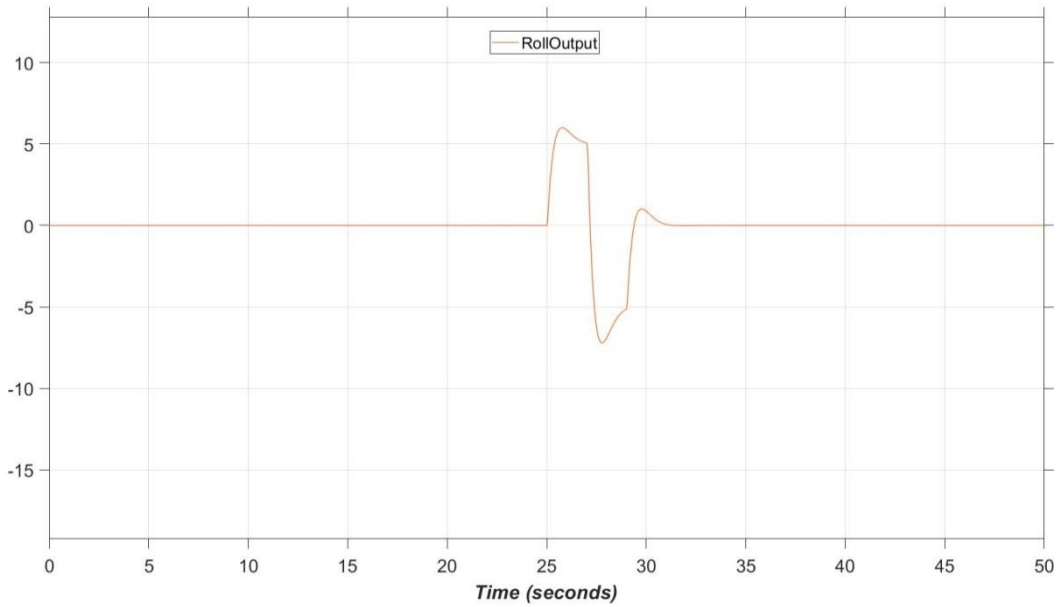

Figure 4.3 The roll maneuver response for tuned PID coefficients

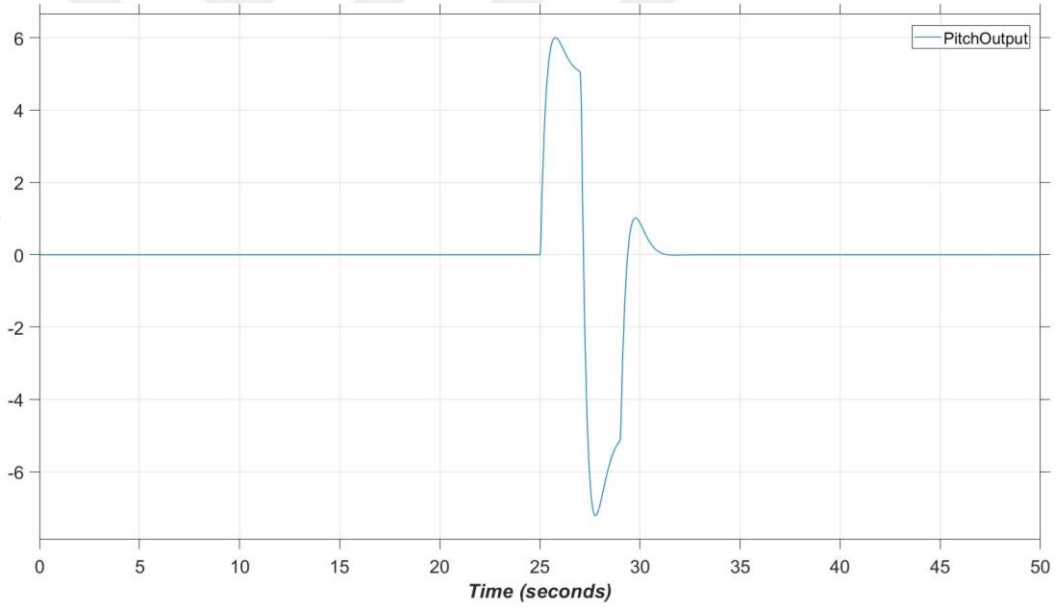

Figure 4.4 The pitch maneuver response for tuned PID coefficients

The yaw maneuver PID coefficients have been implemented with only proportional gain as $K_p = 7$. The yaw maneuver output of the octocopter model is shown in Figure 4.5.

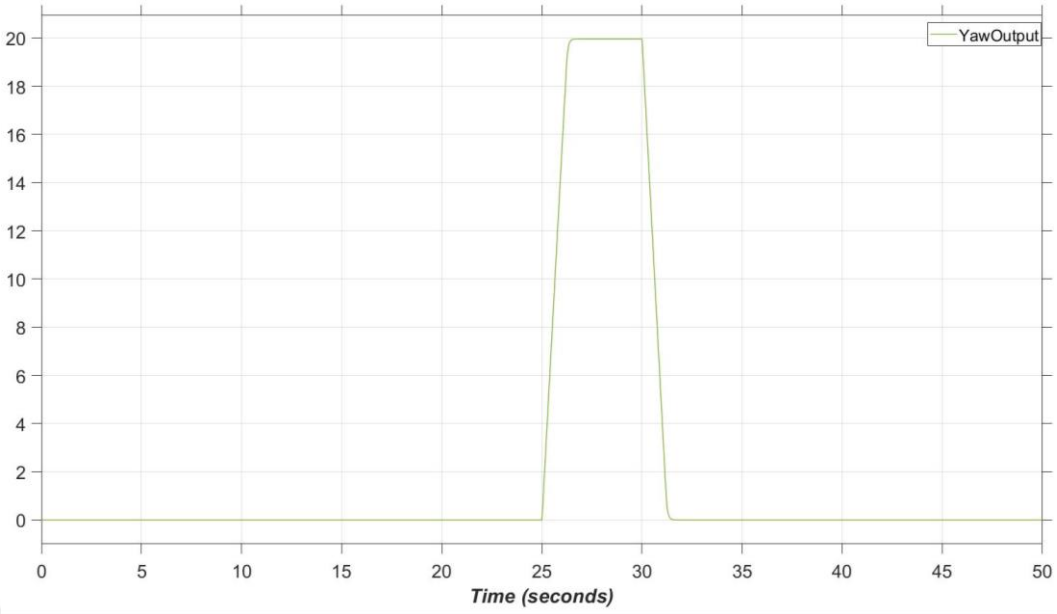

Figure 4.5 The yaw maneuver response for the proportional gain

4.3 NARX NN Controller Training Results

The NARX NN algorithm controllers have been trained with the help of the Matlab Neural Network toolbox. The network architecture, training preferences, training approaches, training methodology, and input-target data for each maneuver have been presented in Section 2.10.2, Section 2.10.3, and Section 3.5.2. The Levenberg-Marquardt algorithm has been optimized to set bias and weight values on each epoch. The training process of NN algorithms with the Levenberg-Marquardt training algorithm according to the inputs and target data has been stopped when the mean squared error is not decreasing on following epochs. The training stops on any of five conditions are met. First, training reaches 1000 epochs. Second, performance reaches minimized MSE goals. Third, the performance gradient falls below $1e^{-7}$. Fourth, the maximum mu passes over $1e^{10}$. Fifth, validation performance reaches 6 validation failures.

The NARX NN training for altitude maneuver has been stabilized at the 123rd epoch with 17.6037 MSE. The data performance on the NARX NN altitude controller for each data set as train, validation, test is shown in Figure 4.6. The training state results for NARX NN altitude controller are shown in Figure 4.7 where

the adaptive mu is μ, the maximum validation number is 6, the gradient is $g = J^T e$. The autocorrelation of errors on the training process has been placed for an error time series between varying lags which is shown in Figure 4.8 for the NARX NN altitude controller on the training process. The error histogram for the NARX NN controller of the altitude maneuver is shown in Figure 4.9 where the total number of instances is 300000 data. To illustrate, the error values are varied between -4.8 and +4.8 for every instance. The input and output correlation is shown in Figure 4.10 where the error between target and output data varies from -0.013 to 0.001. The linear regression results for each NARX NN altitude controller training data are shown in Figure 4.11.

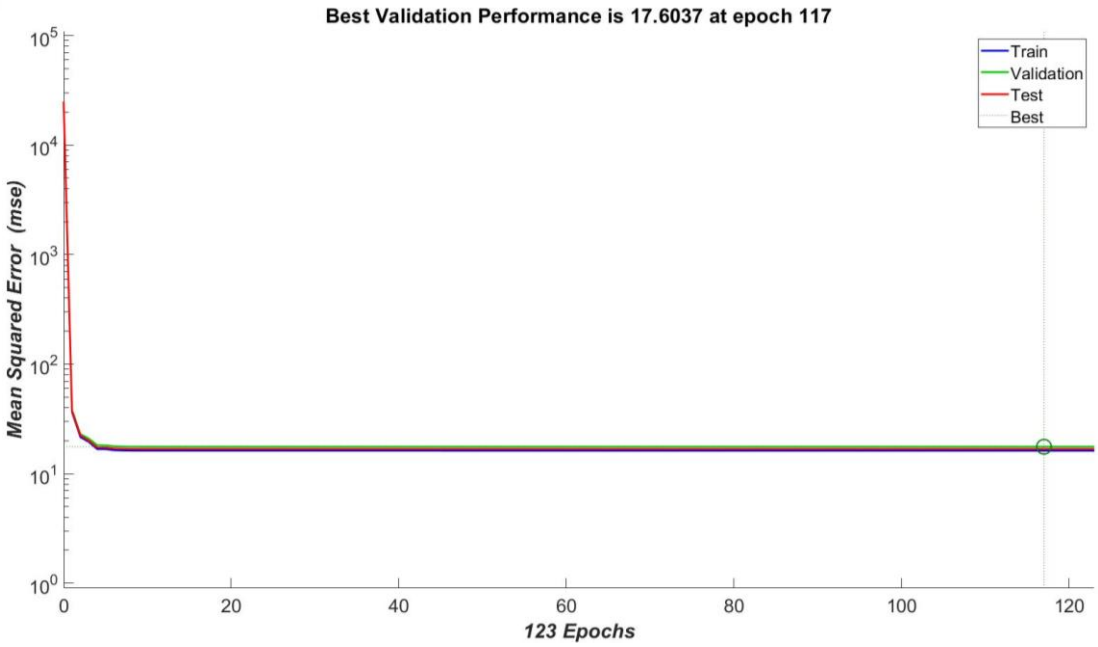

Figure 4.6 NARX NN training performance results for the altitude maneuver

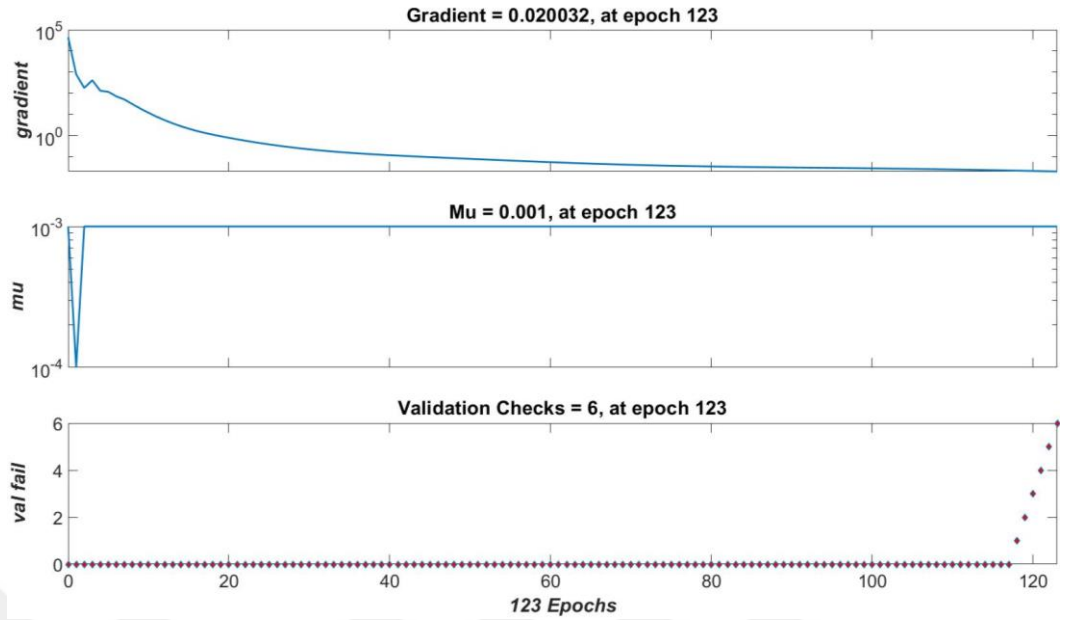

Figure 4.7 NARX NN controller for altitude maneuver training states

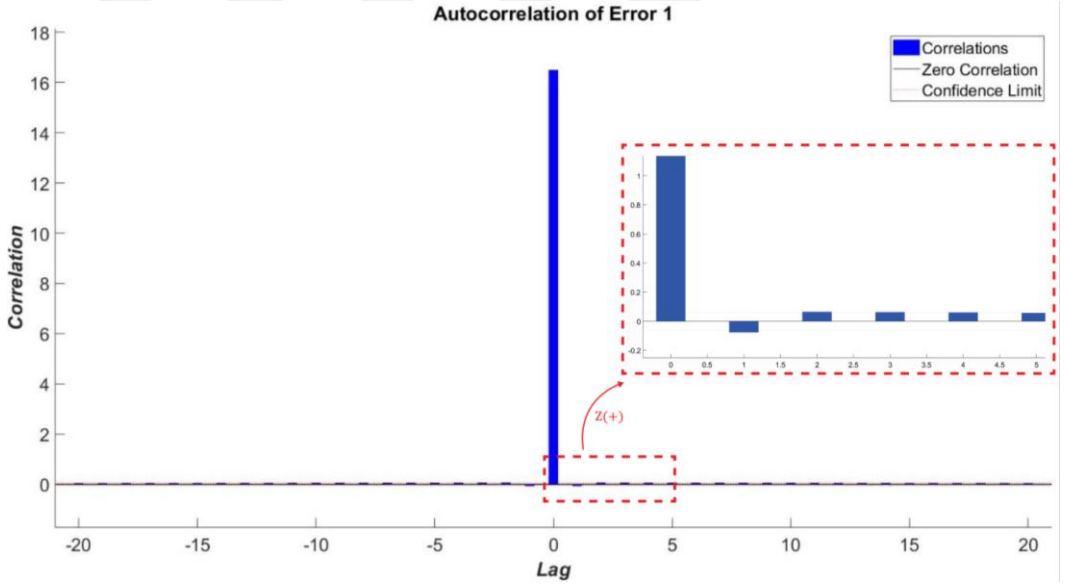

Figure 4.8 NARX NN autocorrelation of error results for the altitude maneuver

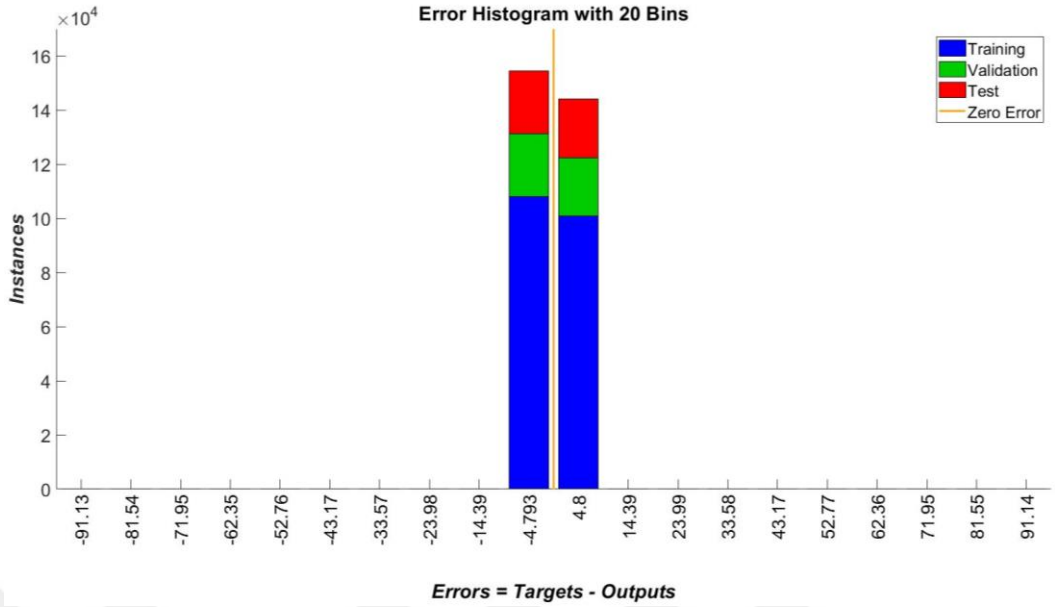

Figure 4.9 NARX NN error histogram results for the altitude maneuver

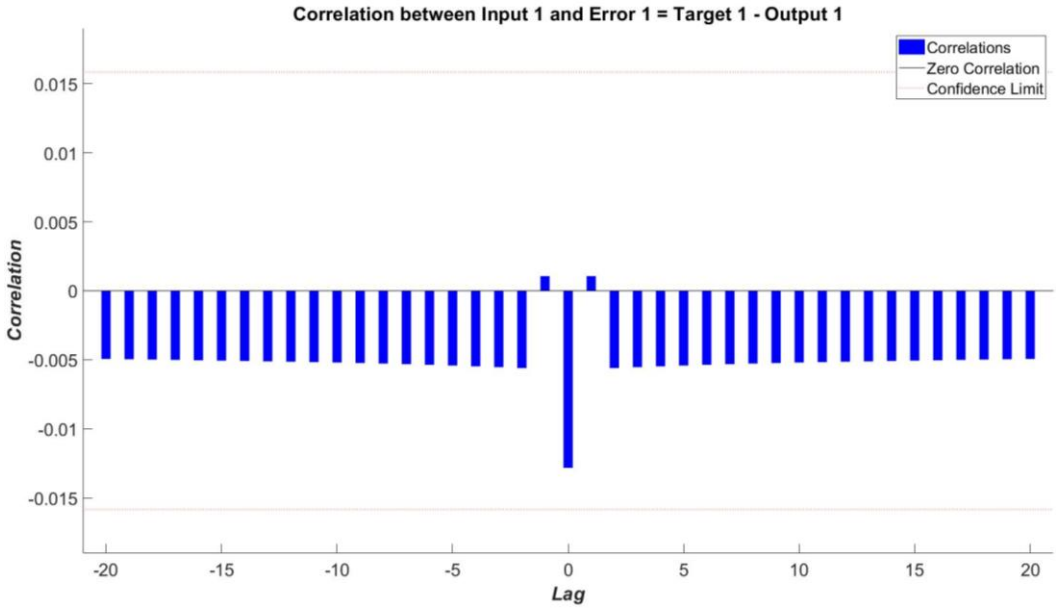

Figure 4.10 NARX NN input-error correlation results for the altitude maneuver

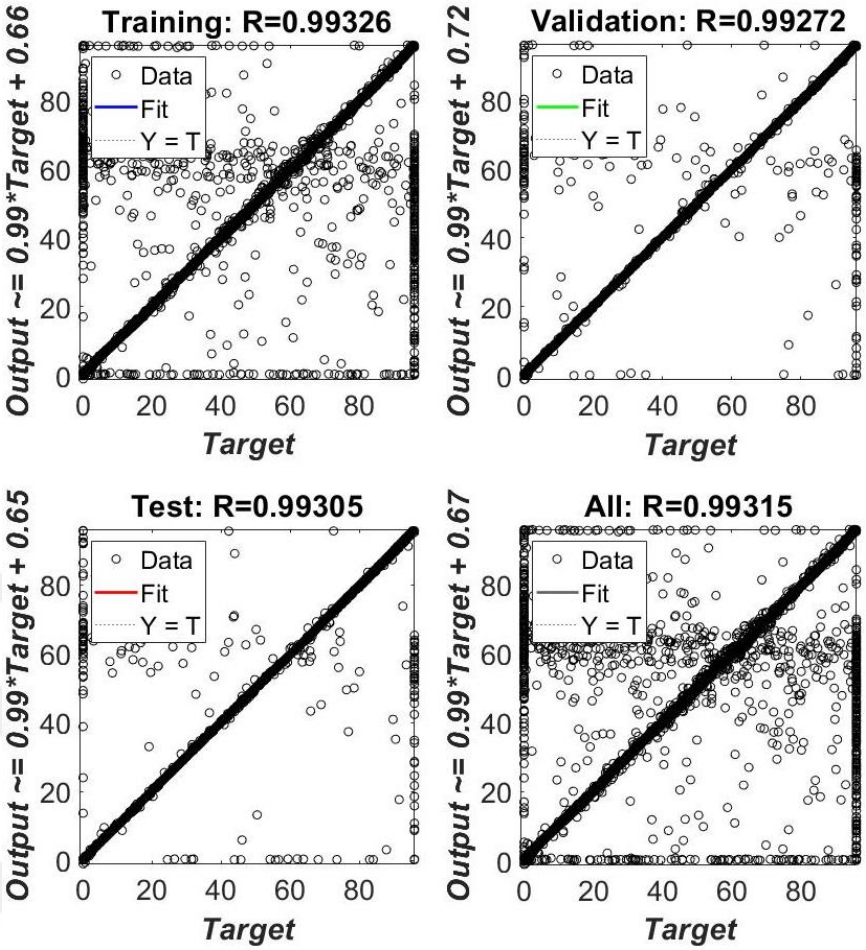

Figure 4.11 NARX NN regression results for the altitude maneuver

The NARX NN training for roll maneuver has been stabilized at the 147th epoch with 1.1041 MSE. The data performance on the NARX NN roll controller for each data set as train, validation, test is shown in Figure 4.12. The training state results for NARX NN roll maneuver controller are shown in Figure 4.13 where the adaptive mu is μ, the maximum validation number is 6, the gradient is $g = J^T e$. The autocorrelation of errors on the training process has been placed for an error time series between varying lags which is shown in Figure 4.14 for the NARX NN roll maneuver controller on the training process. The error histogram for the NARX NN controller of the roll maneuver is shown in Figure 4.15 where the total number of instances is 300000 data. To illustrate, the error values are varied between -0.9673 and +0.9527 for every instance. The input and output correlation is shown in Figure 4.16 where the error between target and output data varies from -0.04 to 0.0072. The

linear regression results for each NARX NN roll controller training data are shown in Figure 4.17.

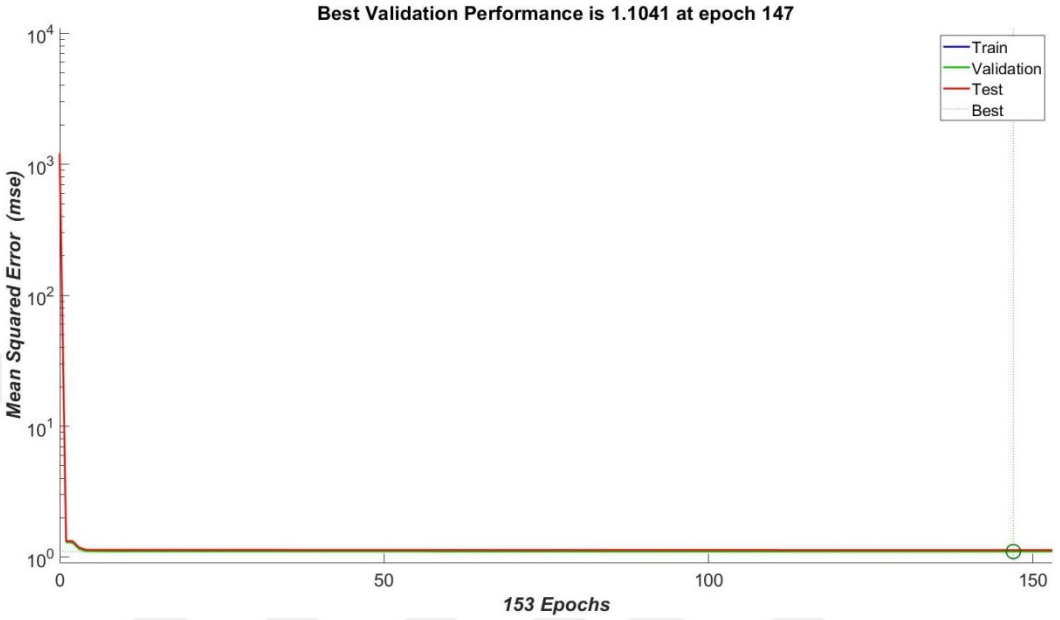

Figure 4.12 NARX NN training performance results for the roll maneuver

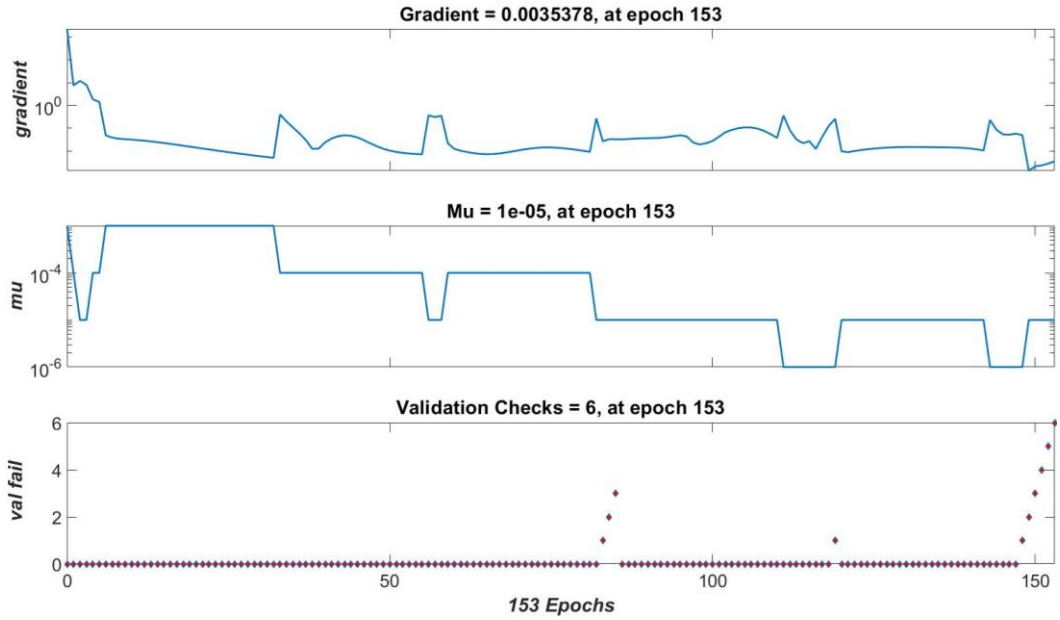

Figure 4.13 NARX NN controller for roll maneuver training states

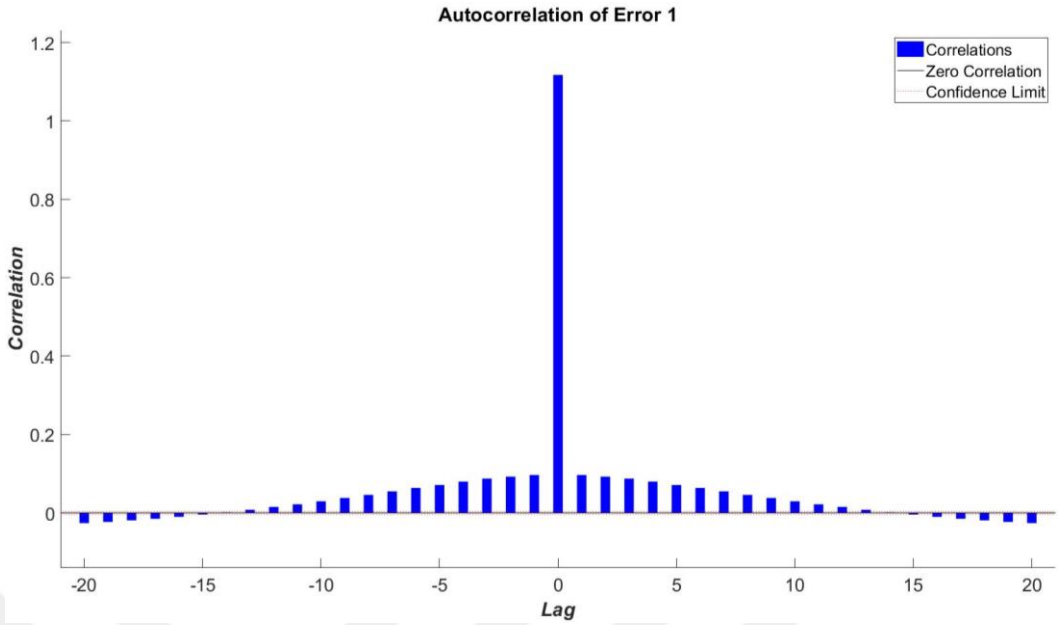

Figure 4.14 NARX NN autocorrelation of error results for the roll maneuver

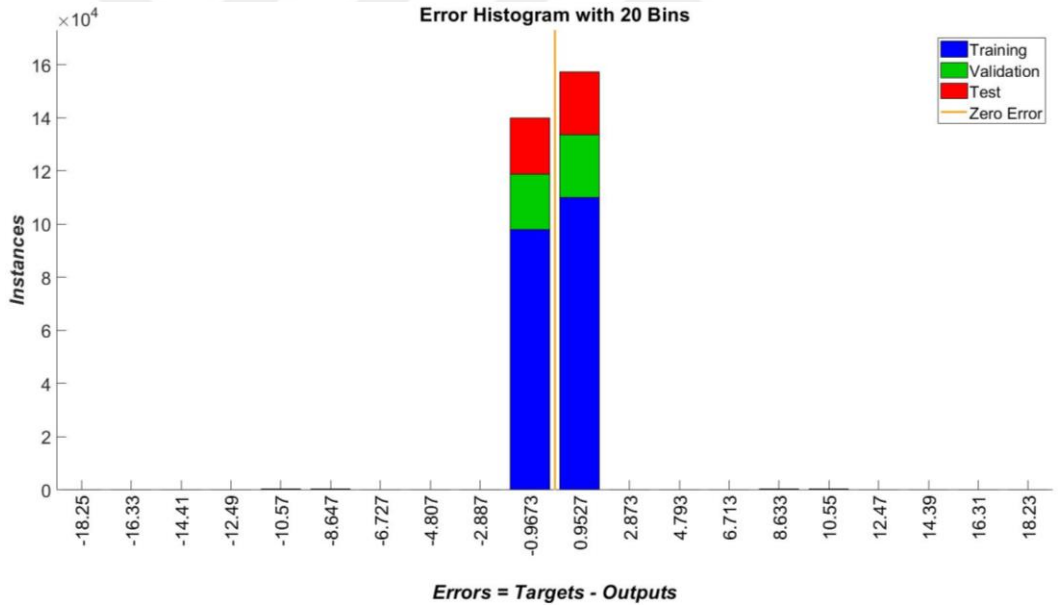

Figure 4.15 NARX NN error histogram results for the roll maneuver

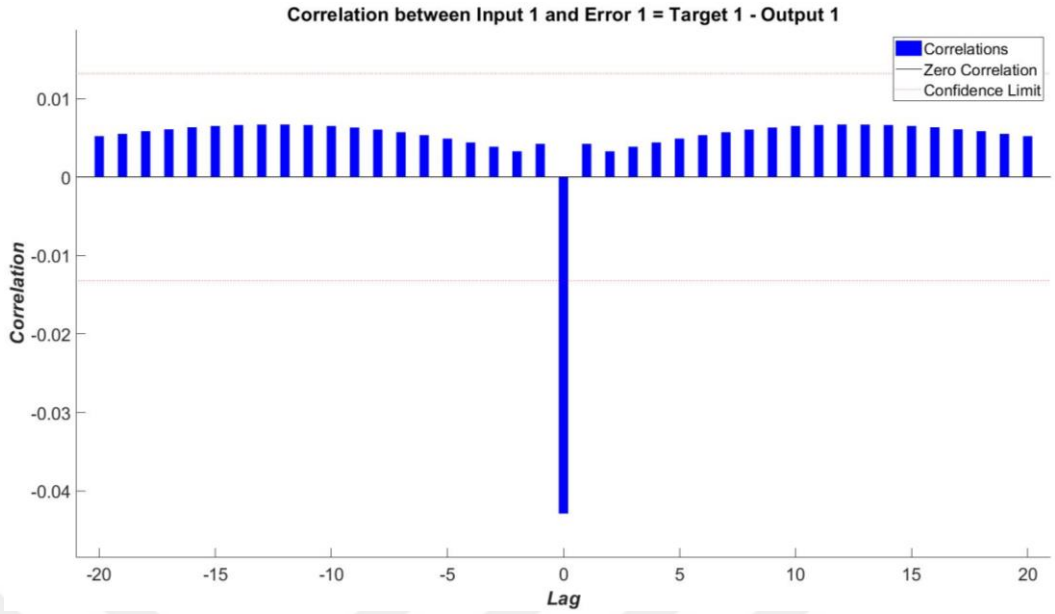

Figure 4.16 NARX NN input-error correlation results for the roll maneuver

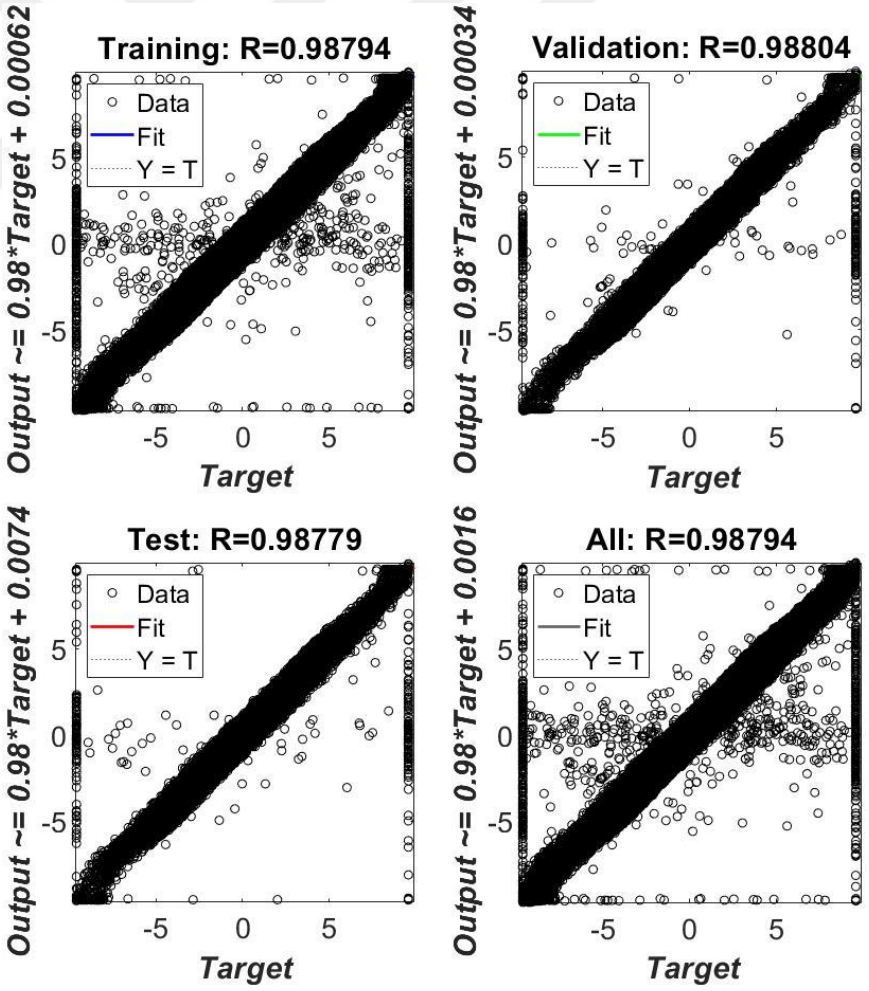

Figure 4.17 NARX NN regression results for the roll maneuver

The NARX NN training for pitch maneuver has been stabilized at the 151st epoch with 1.2013 MSE. The data performance on the NARX NN pitch controller for each data set as train, validation, test is shown in Figure 4.18. The training state results for NARX NN pitch maneuver controller are shown in Figure 4.19 where the adaptive mu is μ, the maximum validation number is 6, the gradient is $g = J^T e$. The autocorrelation of errors on the training process has been placed for an error time series between varying lags which is shown in Figure 4.20 for the NARX NN pitch maneuver controller on the training process. The error histogram for the NARX NN controller of the pitch maneuver is shown in Figure 4.21 where the total number of instances is 300000 data. To illustrate, the error values are varied between -0.9602 and +0.9596 for every instance. The input and output correlation is shown in Figure 4.22 where the error between target and output data varies from -0.049 to 0.0002. The linear regression results for each NARX NN pitch controller training data are shown in Figure 4.23.

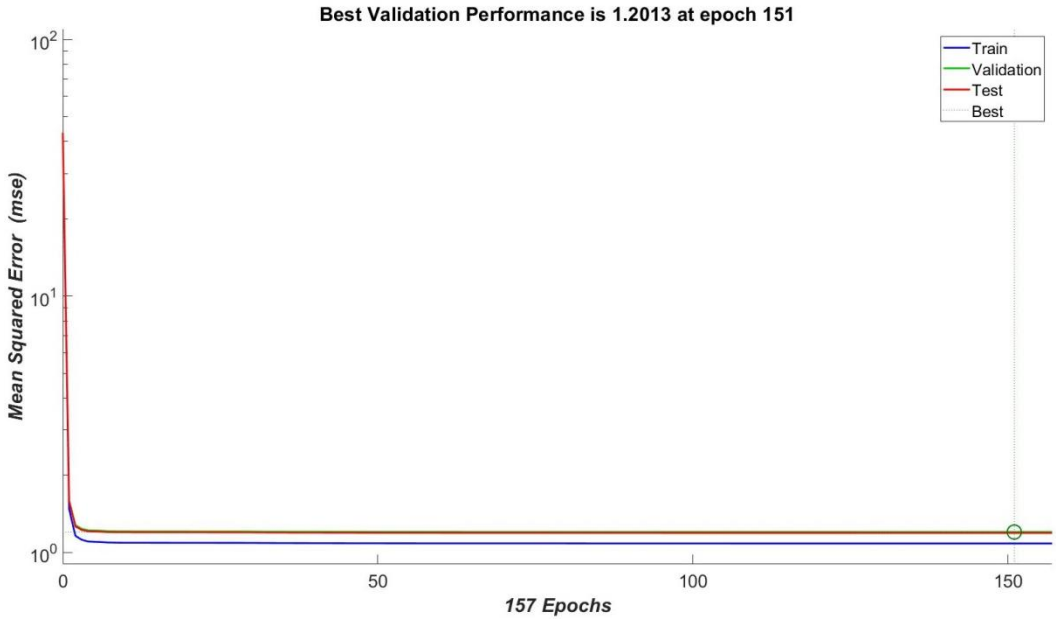

Figure 4.18 NARX NN training performance results for the pitch maneuver

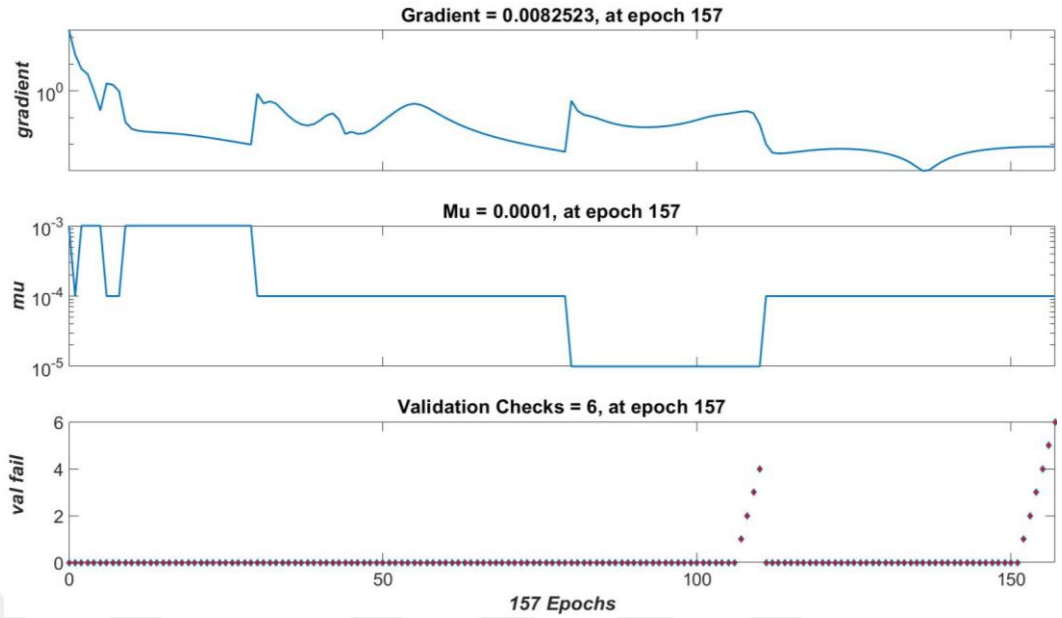

Figure 4.19 NARX NN controller for pitch maneuver training states

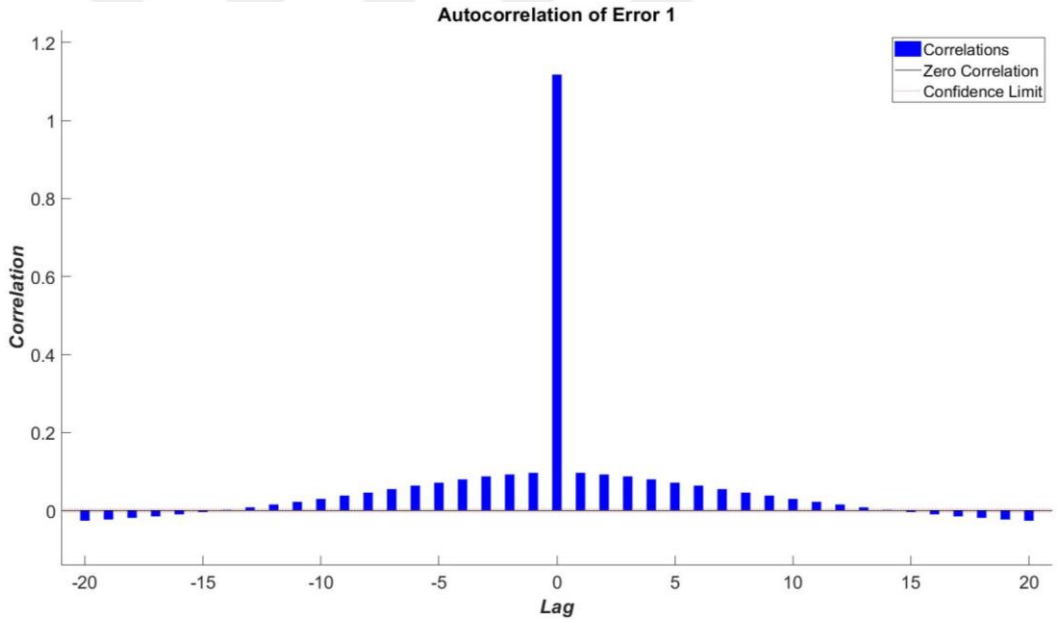

Figure 4.20 NARX NN autocorrelation of error results for the pitch maneuver

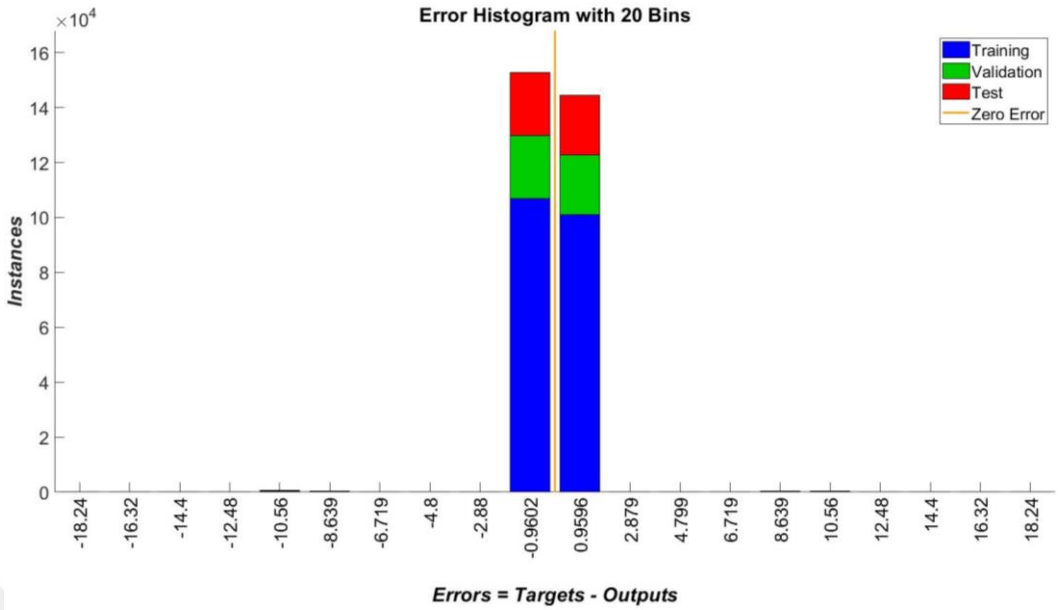

Figure 4.21 NARX NN error histogram results for the pitch maneuver

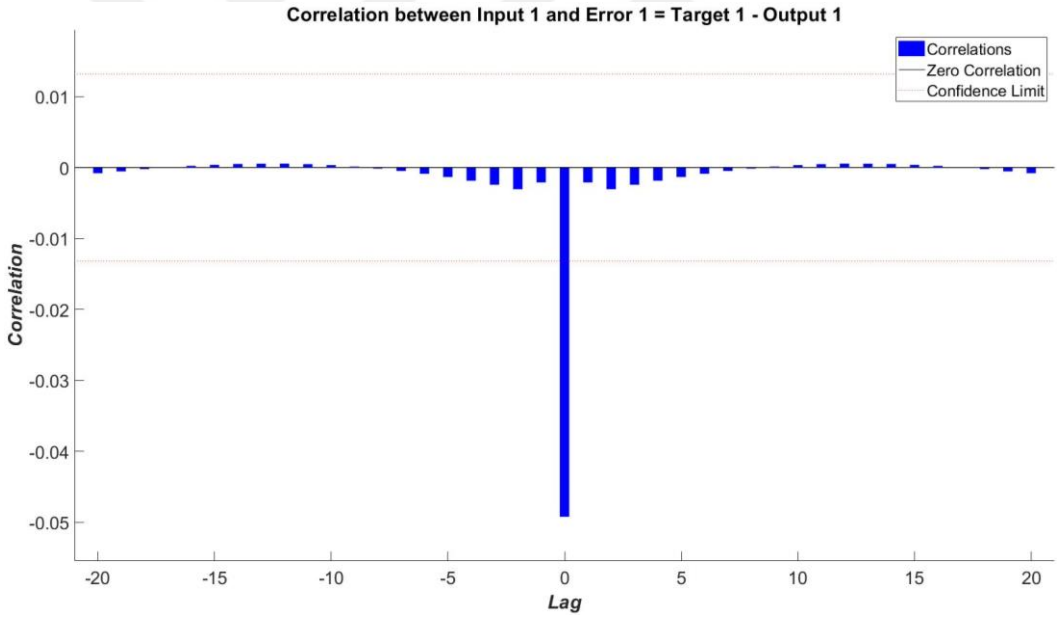

Figure 4.22 NARX NN input-error correlation results for the pitch maneuver

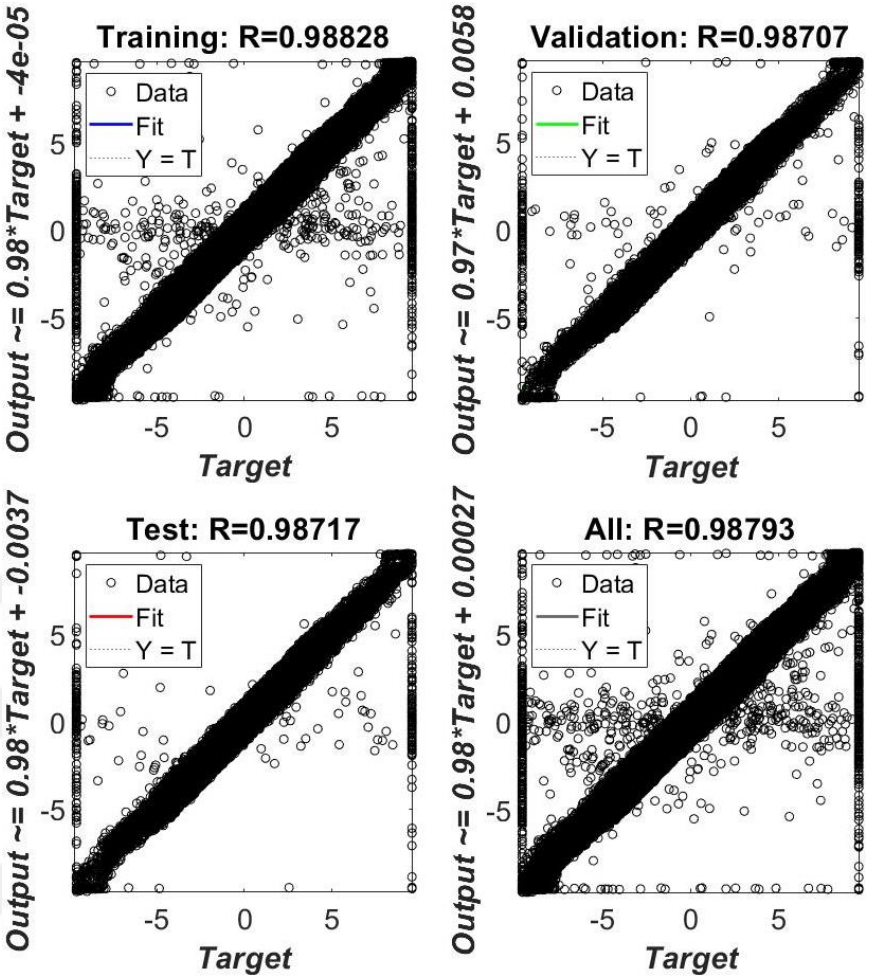

Figure 4.23 NARX NN regression results for the pitch maneuver

The NARX NN training for the yaw maneuver has been stabilized at the 32nd epoch with 0.036395 MSE. The data performance on the NARX NN yaw controller for each data set as train, validation, test is shown in Figure 4.24. The training state results for NARX NN yaw maneuver controller are shown in Figure 4.25 where the adaptive mu is μ, the maximum validation number is 6, the gradient is $g = J^T e$. The autocorrelation of errors on the training process has been placed for an error time series between varying lags which is shown in Figure 4.26 for the NARX NN yaw maneuver controller on the training process. The error histogram for the NARX NN controller of the yaw maneuver is shown in Figure 4.27 where the total number of instances is 300000 data. To illustrate, the error values are varied between -0.039 and +0.3449 for every instance. The input and output correlation is shown in Figure 4.28 where the error between target and output data varies from -0.187 to 0.00013. The

linear regression results for each NARX NN yaw controller training data are shown in Figure 4.29.

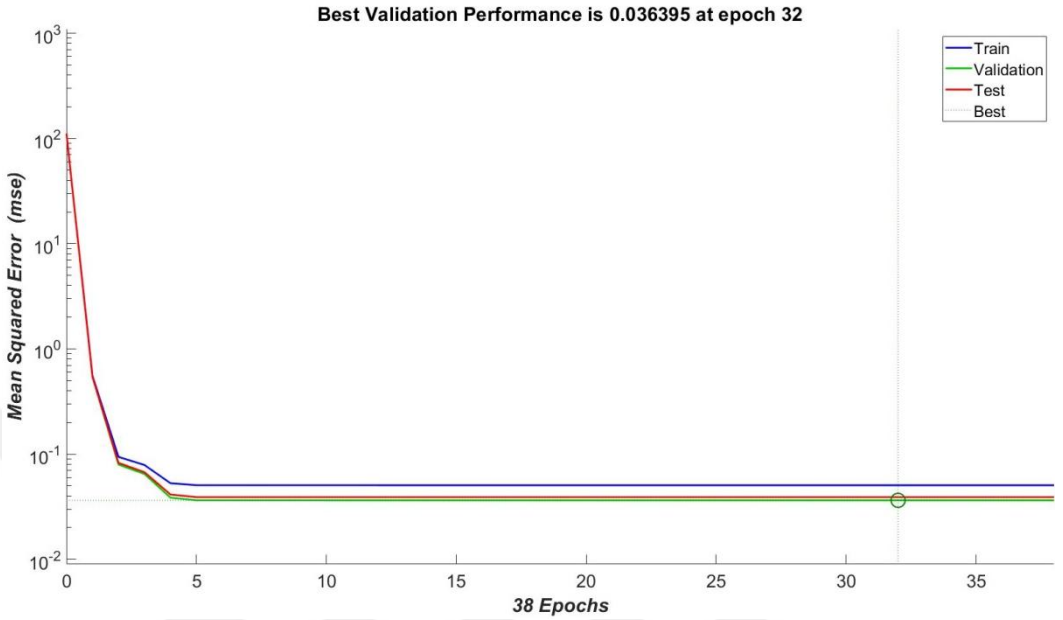

Figure 4.24 NARX NN training performance results for the yaw maneuver

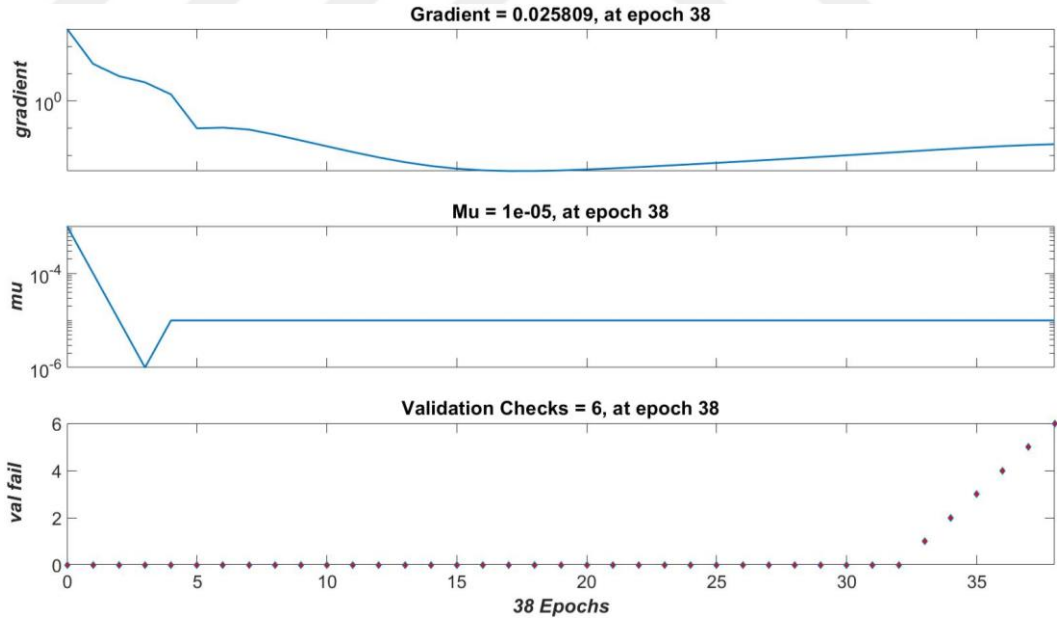

Figure 4.25 NARX NN controller for yaw maneuver training states

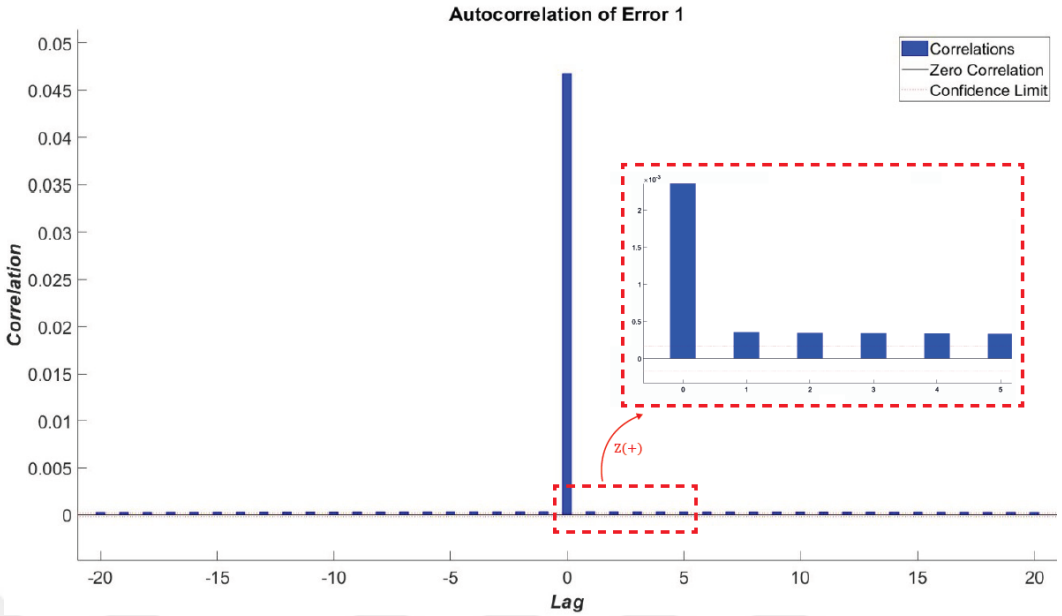

Figure 4.26 NARX NN autocorrelation of error results for the yaw maneuver

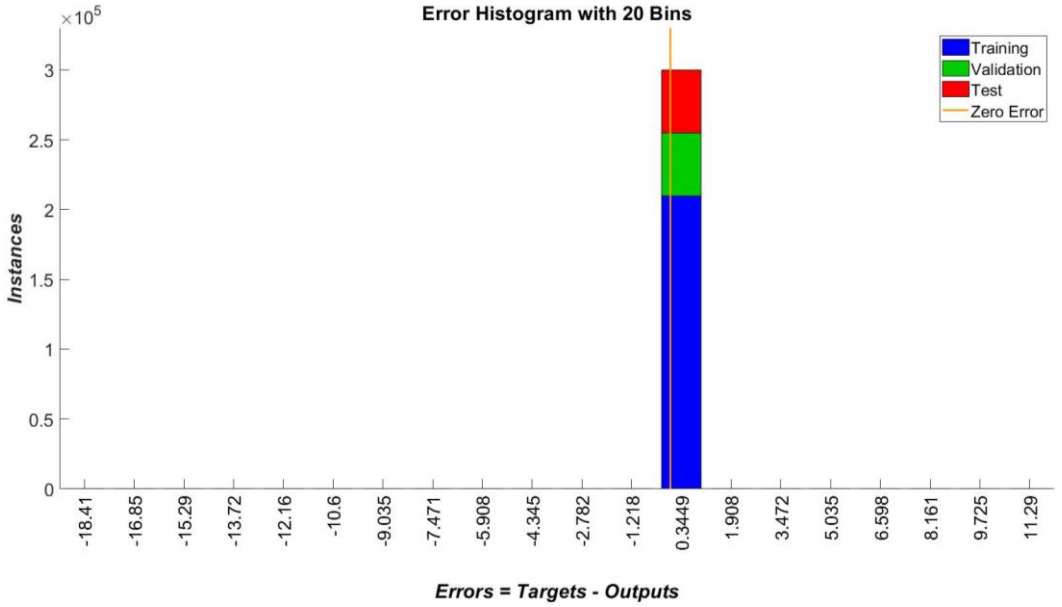

Figure 4.27 NARX NN error histogram results for the yaw maneuver

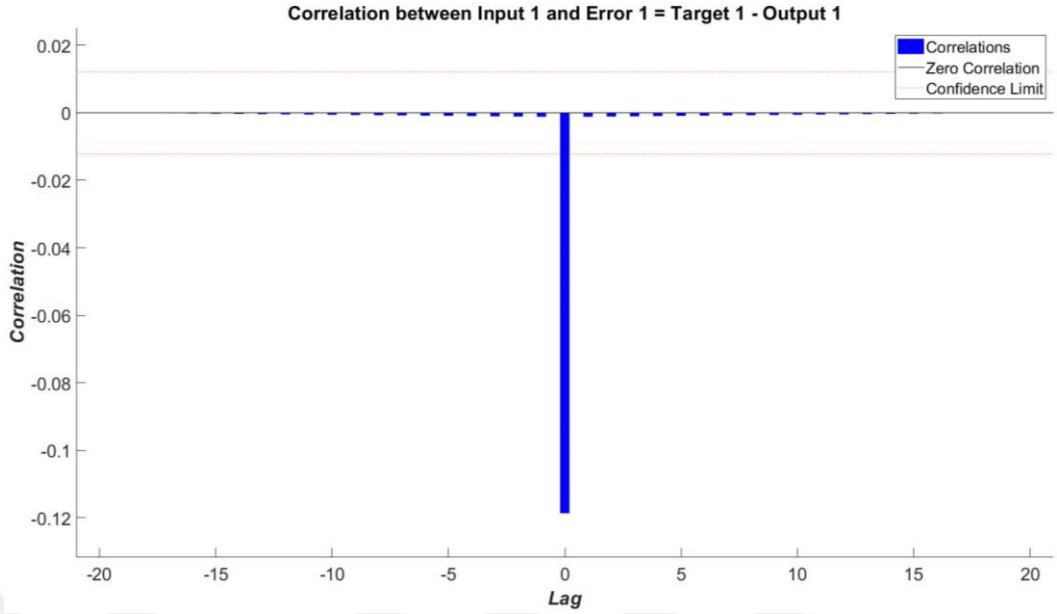

Figure 4.28 NARX NN input-error correlation results for the yaw maneuver

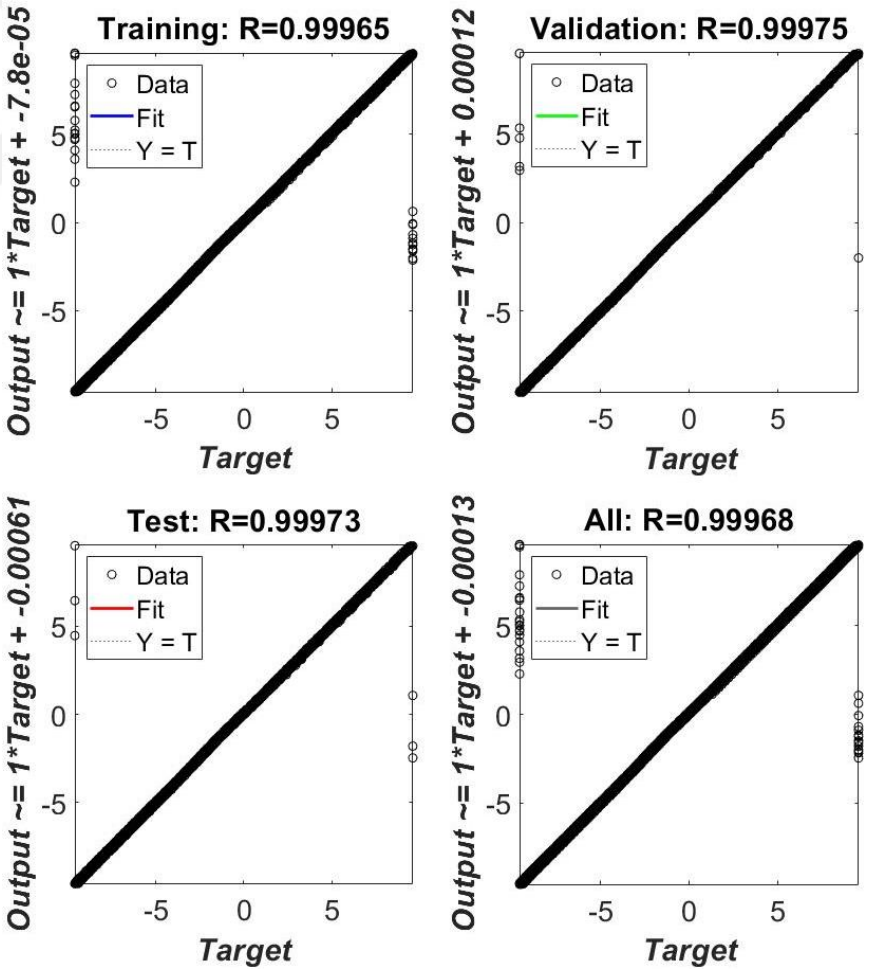

Figure 4.29 NARX NN regression results for the yaw maneuver

4.4 Comparison of Performance Results

The verification of the PID and NARX NN controllers has been compared with the help of mean square error (MSE), and root means square error (RMSE). The NARX NN controllers have been trained with the same error inputs and outputs of the first level saturation block for the PID controller. As a result, there are a very small amount of differences has been observed in the graphical, MSE, and RMSE results. The NARX NN controllers have achieved a successful control on each maneuver like PID controllers. The RMSE and MSE values for both types of attitude and altitude controllers are shown in Table 3.

Table 4.1 The MSE and RMSE performance results of attitude and altitude maneuver controllers

Error Type	Altitude (meter)	Roll (degree)	Pitch (degree)	Yaw (degree)
Mean Square Error (MSE) of the NARX NN	16.20272	1.11671	1.0840	0.003900
Root Mean Square Error (RMSE) of the NARX NN	0.993264	0.987942	0.988281	0.999734
Mean Square Error (MSE) of the PID Results	16.78403	1.13408	1.19132	0.005063
Root Mean Square Error (RMSE) of the PID Results	0.993046	0.98779	0.987117	0.999654

The graphical comparisons have been taken without random Gaussian noises to increase observability. The graphical comparison of the PID altitude controller, the NARX NN altitude controller, and altitude reference inputs are shown in Figure 4.30. In addition, the altitude controllers for both algorithms have given the highest error values where the MSE error is 16.20272 for the NARX NN controller and 16.78403 for the PID controller. The graphical comparison of the PID roll controller, the NARX NN roll controller, and roll reference inputs are shown in Figure 4.31. The graphical comparison of the PID pitch controller, the NARX NN pitch controller, and pitch reference inputs are shown in Figure 4.32. Moreover, the roll and pitch controllers for both algorithms have given the mid-range error values where the MSE error is 1.11671 for the NARX NN roll controller, 1.13408 for the PID roll controller, 1.0840 for the NARX NN pitch controller, and 1.19132 for the PID pitch controller. The graphical comparison of the PID yaw controller, the NARX NN yaw

controller, and yaw reference inputs are shown in Figure 4.33. In addition, the altitude controllers for both algorithms have given the lowest error values where the MSE error is 0.00390073 for the NARX NN controller and 0.00506391 for the PID controller.

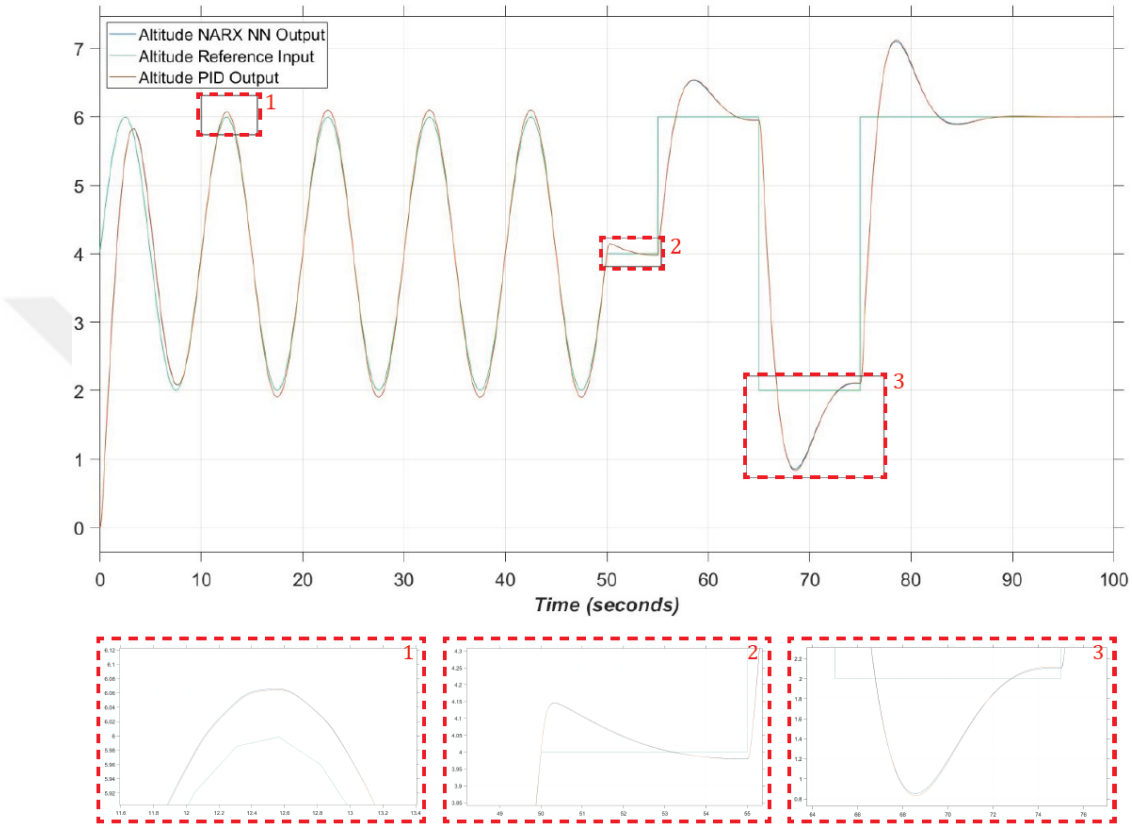

Figure 4.30 Altitude maneuver reference inputs and controller output comparison

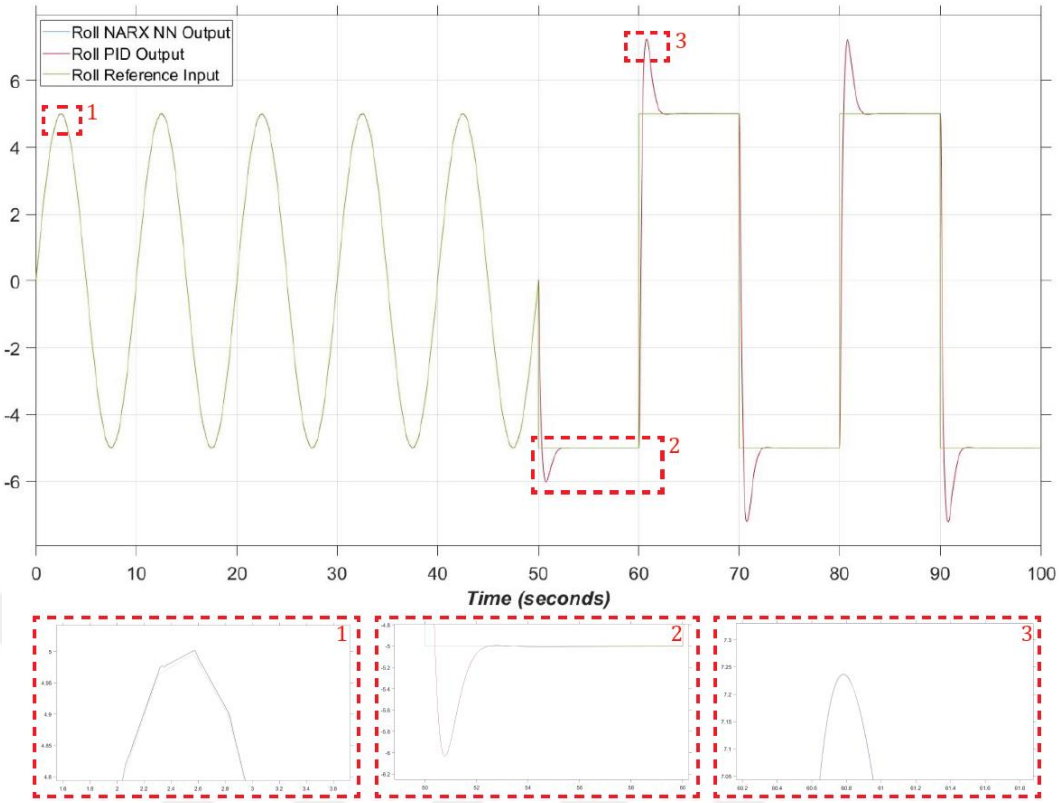

Figure 4.31 Roll maneuver reference inputs and controller output comparison

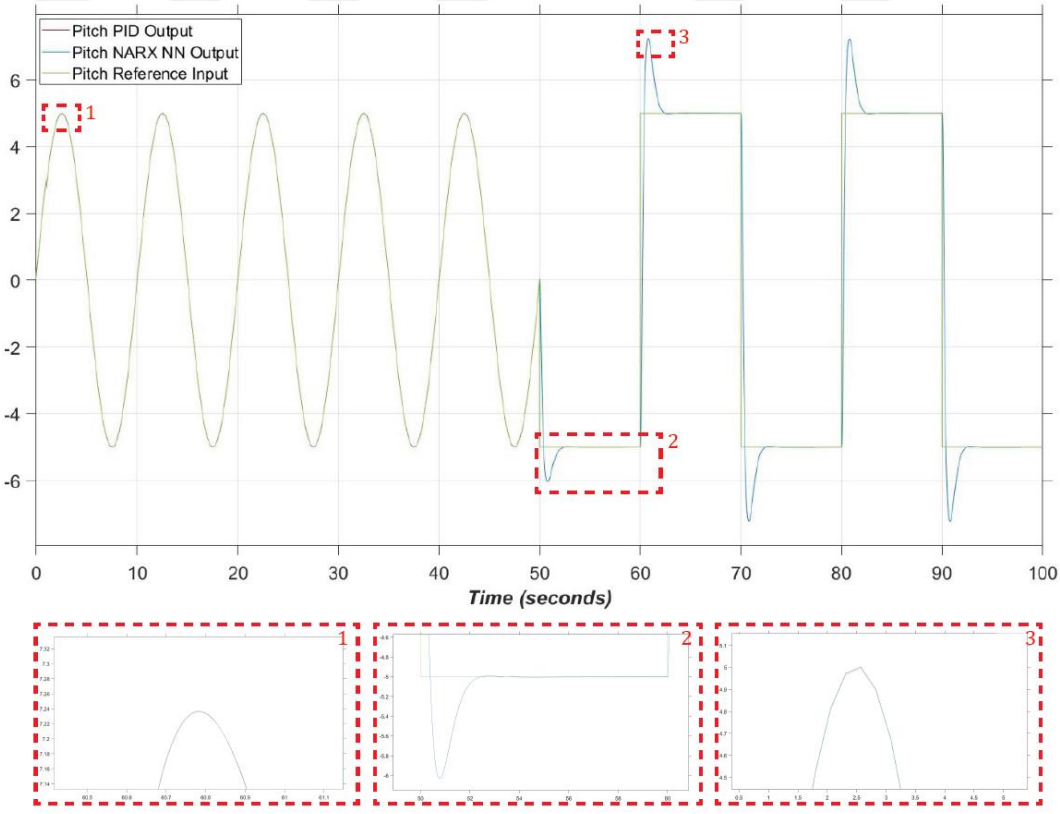

Figure 4.32 Pitch maneuver reference inputs and controller output comparison

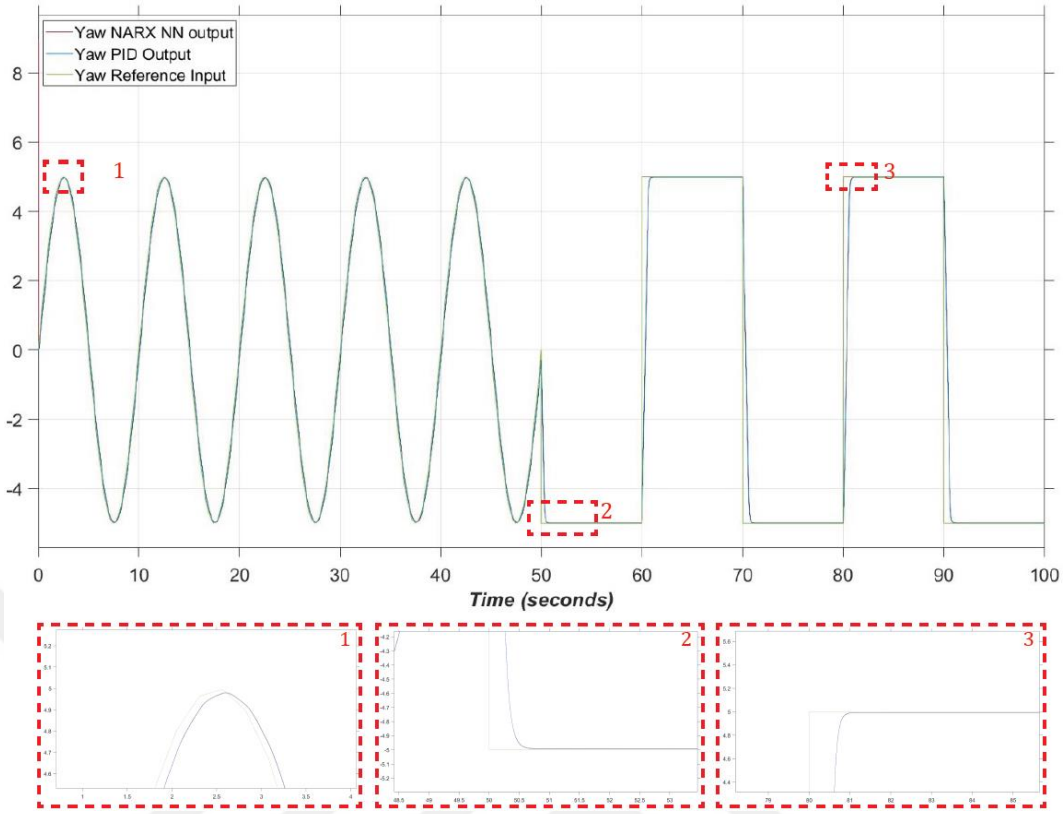

Figure 4.33 Yaw maneuver reference inputs and controller output comparison

CHAPTER FIVE
CONCLUSION

5.1 Conclusions

In this study, a mathematical model of the octocopter and dynamic Matlab Simulink model-based design has been introduced. In addition, the altitude and attitude maneuver controllers have been proposed the PID and the NARX NN algorithms where the PID controller is tuned with the help of the Ziegler-Nichols method and the NARX NN controller is trained with the help of the Levenberg-Marquardt method. First, the requirement sets for model-based design have been set for octocopter. Second, the flowchart of the system has been decided to cover similar process high-level functions (PHLF) under the same subsystem. After the flowchart creation, the mathematical equations have been derived to pair flowchart and PHLFs. Third, the 3D octocopter has been designed to implement environment, visual information, and mass properties with Simmechanics blocks. Fourth, the rotors and propellers have been designed and implemented to the Simulink model with the help of Matlab Simmechanics blocks. In addition, the properties of rotor and propeller models have been selected to lift the UAV. Finally, the maneuver controllers have been implemented to control the octocopter. Alternatively, all of the controllers have to be applied to the system even with non-tuned PID controllers for other maneuvers to come over the balance problem due to differences in moments of inertia.

The NARX NN controllers have been trained with the help of tuned PID controller results. According to the training process with the PID data set, the NARX NN controllers have been accomplished similar outcomes. The performance results of the NARX NN algorithm have been proved to control the attitude and altitude of UAVs as an alternative approach. The NARX NN RMSE results for each maneuver have been calculated as 0.993264 for altitude maneuver controller, 0.987942 for roll maneuver controller, 0.988281 for pitch maneuver controller, and 0.999734 for yaw maneuver controller. Likewise, the PID RMSE results for each maneuver have been calculated as 0.993046 for altitude maneuver controller, 0.98779 for roll maneuver

controller, 0.987117 for pitch maneuver controller, and 0.999654 for yaw maneuver controller. It can be concluded that the NARX NN algorithm has been performed better on every maneuver according to the RMSE comparison. The NARX NN MSE results for each maneuver have been calculated as 16.20272 for altitude maneuver controller, 1.11671 for roll maneuver controller, 1.084 for pitch maneuver controller, and 0.0039 for yaw maneuver controller. Likewise, the PID MSE results for each maneuver have been calculated as 16.78403 for altitude maneuver controller, 1.13408 for roll maneuver controller, 1.19132 for pitch maneuver controller, and 0.005063 for yaw maneuver controller. As can be seen, the NARX NN algorithm has been performed better on every maneuver according to the MSE comparison.

5.2 Future Works

In this study, NARX NN and PID attitude and altitude controllers development have been aimed for an octocopter. The Simulink model can be improved with real-life test flight data for a similitude model to real-life. The controllers can be implemented in the manufactured octocopter to observe maneuver reaction performances. The NARX NN algorithm controller performance can be improved with the help of real-life results where the new data sets will improve the prediction power of the NN algorithm. The trajectory and path tracking controllers can be implemented into the model with the help of navigation systems.

REFERENCES

Ahmad, F., Kumar, P., Bhandari, A. & Patil, P. P., (2018). Simulation of the quadcopter dynamics with LQR based control. *Materials Today: Proceedings*, *24*, 326-332.

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.

Altan, A., Aslan, O. & Hacioglu, R., (2018). Real-time control based on NARX neural network of hexarotor UAV with load transporting system for path tracking. *International Conference on Control Engineering & Information Technology*, *6*, 1-6.

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

Bergman, K. & Ekström, J., (2014). *Modeling, estimation and attitude control of an octorotor using PID and L 1 adaptive control techniques*. Bsc Thesis, Linköping University, Linköping.

Bouzid, Y., Siguerdidjane, H. & Bestaoui, Y., (2016). Improved 3D trajectory tracking by nonlinear internal model-feedback linearization control strategy for autonomous systems. *IFAC PapersOnLine*, *49*(9), 13-18.

Carlton, Z. M., (2017). *System identification and verification of rotorcraft UAVs*. Msc Thesis, University of Cincinnati, Ohio.

Cedro, L. & Wieczorkowski, K., (2019). Optimizing PID controller gains to model the performance of a quadcopter. *Transportation Research Procedia*, *40*, 156-169.

Chapala, S. R., Pirati, G. S. & Nelakuditi, U. R., (2016). Determination of coordinate transformations in UAVS. Second International Conference on Cognitive *Computing and Information Processing*, 1-5.

Çakıcı, F. & Leblebicioğlu, M. K., (2016). Control system design of a vertical take-off and landing fixed-wing UAV. *IFAC-PapersOnLine, 49*(3), 267-272.

Ding, X., Guo, P., Xu, K. & Yu, Y., (2019). A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems. *Chinese Journal of Aeronautics*, *32*(1), 200-214.

Donmez, C., Villi, O., Berberoglu, S. & Cilek, A., (2021). Computer vision-based citrus tree detection in a cultivated environment using UAV imagery. *Computers and Electronics in Agriculture*, *187*, 1-13.

El Dakrory, A. M. & Tawfik, M., (2017). Identifying the attitude of dynamic systems using neuralnetwork. *International Workshop on Recent Advances in Robotics and Sensor Technology for Humanitarian Demining and Counter-IEDs*, 1-4.

El Hamidi, K., Mjahed, M., El Kari, A. & Ayad, H., (2020). Adaptive control using neural networks and approximate models for nonlinear dynamic systems. *Modelling and Simulation in Engineering*, 1-13.

El Hamidi, K., Mjahed, M., El Kari, A. E. & Ayad, H., (2019). Neural network and fuzzy-logic-based self-tuning PID control for quadcopter path tracking. *Studies in Informatics and Control, 28*(4), 401-412.

Elhennawy, A., (2018). *Dynamic modeling and robust nonlinear control of unmanned quadrotor vehicle*. Msc Thesis, American University, Cairo.

Fahimi, F., (2009). *Autonomous robots* (1th Ed.). Springer.

Ferdaus, M. M., Anavatti, S. G., Pratama, M. & Garratt, M. A., (2018). A novel self-organizing neuro-fuzzy based intelligent control system for a AR.Drone quadcopter. *IEEE Symposium Series on Computational Intelligence*, 2026-2032.

Goodfellow, I., Bengio, Y. & Courville, A., (2016). *Deep learning.* Retrieved 5 February, 2019, from https://www.deeplearningbook.org/.

Graupe, D., (2007). *Principles of artificial neural networks* (2nd Ed.)., World Scientific.

Hagan, M. T. & Menhaj, M. B., (1994). Brief papers training feedforward networks with the Marquardt algorithm. *Transactions on Neural Networks*, 5(6), 989-993.

Hagiwara, K., Hayasaka, T., Toda, N., Usui, S. & Kuno, K., (2001). Upper bound of the expected training error of neural network regression for a Gaussian noise sequence. *Neural Networks, 14*, 1419-1429.

Karakaya, Ş. E. & Gören, A., (2022). Performance comparison of PID and NARX neural network for attitude control of a quadcopter UAV. *Journal of Materials and Mechatronics: A, 3(*1*)*, 1-20.

Kotzé, H., Jordaan, H. W. & Kamper, H., (2021). *Neural disturbance rejection for a multirotor*. Msc Thesis, Stellenbosch University, Stellenbosch.

Krivec, T., Papa, G. & Kocijan, J., (2021). Simulation of variational Gaussian process NARX models with GPGPU. *ISA Transactions, 109*, 141-151.

Kumar, A., Ben-Tzvi, P., Saab, W. & Snyder, M. R., (2016). Wireless telemetry system for real-time estimation of ship air wakes with UAVs. *Mechatronics*, *36*, 18-26.

Kusumoputro, B., Suprijono, H., Ary Heryanto, M. & Yudho Suprapto, B., (2016). Development of an attitude control system of a heavy-lift hexacopter using Elman recurrent neural networks. *International Conference on Automation and Computing*, *22*, 1-5.

Labbadi, M., Cherkaoui, M., El Houm, Y. & Guisser, M., (2018). Modeling and robust integral sliding mode controlor a quadrotor unmanned aerial vehicle. *International Renewable and Sustainable Energy Conference*, *6*, 1-6.

Levenberg, K., (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, *2*, 164-168.

Li, Y., Ang, K. H. & Chong, G. C., (2006). PID control system analysis and design: problems, remedies, and future Directions. *IEEE Control Systems Magazine*, *26*, 32-41.

Luukkonen, T., (2011). Modelling and control of quadcopter. *Aalto University Open Access*, 1-26.

Mac, T. T., Copot, C., Keyser, R. D. & Ionescu, C. M., (2018). The development of an autonomous navigation system with optimal control of an UAV in partly unknown indoor environment. *Mechatronics*, *49*, 187-196.

Mahony, R., Kumar, V. & Corke, P., (2012). Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, *19*, 20-32.

Marquardt, D. W., (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*, 431-441.

Mchutchon, A. & Rasmussen, C. E., (2011). Gaussian process training with input noise. *Advances in Neural Information Processing Systems, 24*, 1-9.

Mikkelsen, M., (2015). *Development, modelling and control of a multirotor vehicle*. Msc Thesis, Umeå University, Umeå.

Moussid, M., Bensaid, A., Sayouti, A., Radoui, M. & Medromi, H., (2017). A comparison of closed-Loop performance of multirotor configurations using nonlinear dynamic control. *International Research Journal of Engineering and Technology, 4*, 578-588.

Muliadi, J. & Kusumoputro, B., (2018). Neural network control system of UAV altitude dynamics and its comparison with the PID control system. *Journal of Advanced Transportation, 18*, 1-18.

Mustapa, Z., Saat, S., Husin, S. H. & Abas, N., (2014). Altitude controller design for multi-copter UAV. International Conference on Computer, Communication, and *Control Technology*, 382-387.

Nguyen, N. T., Prodan, I., Stoican, F. & Lefevre, L. (2017). Reliable nonlinear control for quadcopter trajectory tracking through differential flatness. *IFAC Papers Online*, 50(1), 6971-6976.

Nguyen, N. P. Mung, N.X., Thanh, H. L. N. N., Huynh, T. T., Lam, N. T. & Hong, S. K., (2021). Adaptive sliding mode control for attitude and altitude system of a quadcopter UAV via neural network. *IEEE Access, Vol. 9*, 40076-40085.

Nielsen, M., (2019). *Neural networks and deep learning*. Retrieved 13 January, 2021, from http://neuralnetworksanddeeplearning.com.

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

Norman, J., (2018). *The first electronic autonomous robots: the origin of social robotics (1948 – 1949)*. Retrieved 9 December 01, 2021, from https://www.historyofinformation.com/detail.php?entryid=854.

Oscarson, O., (2015). *Design, modeling and control of an octocopter*. Msc Thesis, Royal Institute of Technology, Stockholm.

Padhy, R. P., Verma, S., Ahmad, S., Choudhury, S. K. & Sa, P. K., (2018). Deep neural network for autonomous UAV navigation in indoor corridor environments. *Procedia Computer Science*, *133*, 643-650.

Paiva, E., Soto, J., Salinas, J. & Ipanaque, W., (2016). Modeling, simulation and implementation of a modified PID controller for stabilizing a quadcopter. *International Conference on Automatica*, 1-6.

Pereida Pérez, K., (2020). *Adaptive and learning controllers for high-accuracy trajectory tracking in changing conditions*. Phd Thesis, University of Toronto, Toronto.

Praveen, V. & Pillai, A. S., (2016). Modeling and simulation of quadcopter using PID controller. *International Science Press*, *9*(15), 7151-7158.

Quan, Q., (2017). *Introduction to multicopter design and control introduction to multicopter design and control introduction to multicopter design and control*, (1th Ed.). Springer.

Rabunal, J. R. & Dorrado, J., (2006). *Artificial neural networks in real-life applications*, (1th Ed.). Idea Group Publishing.

Razmi, H. & Afshinfar, S., (2019). Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV. *Aerospace Science and Technology*, *91*, 12-27.

Rice, C. M., (2014). *Control design and performance analysis for autonomous control design and performance analysis for autonomous formation flight experiments formation flight experiments*. Msc Thesis, West Virginia University, West Virginia.

Rmilah, M. H. Y. A., (2016). *An integrated simulation platform for a quadcopter system with fuzz-tuned PID controllers*. Msc Thesis, University of Malaya Kuala Lumpur, Kuala Lumpur.

Say, M. F. Q., Sybingco, E., Bandala, R. R. P. V. & Chua, A. Y., (2019). Unmanned aerial vehicle (UAV) attitude estimation using artificial neural network approach. *International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management*, *11*, 1-4.

Shupeng, L., (2016). *Trajectory generation based guidence and control of rotorcraft unmanned aerial vehicles*. Bsc Thesis, National University of Singapore, Singapore.

Silva, I. B. D. & Adeodato, P. J., (2011). PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets. *International Joint Conference on Neural Networks*, 1-6.

Summers, A., (2017). *Modeling and control of a fixed wing tilt-rotor tri-copter*. Msc Thesis, University of Washington, Washington.

Tagay, A., Omar, A. & Ali, M. H., (2021). Development of control algorithm for a quadcopter. *Procedia Computer Science*, *179*, 242-251.

Tijani, I. B., Akmeliawati, R., Legowo, A. & Budiyono, A., (2014). Nonlinear identification of a small scale unmanned helicopter using optimized NARX network with multiobjective differential evolution. *Engineering Applications of Artificial Intelligence*, *33*, 99-115.

Tu, J. V., (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J Clin Epidemiol*, *49*, 1225-1231.

Visioli, A., (2006). *Practical PID control* (1th ed.). Springer.

Wang, P. et al., (2016). Dynamics modelling and linear control of quadcopter. *International Conference on Advanced Mechatronic Systems*, *16*, 498-503.

Wei Wei, b. B., (2015). *Development of an effective system identification and control capability for quad-copter UAVs.* Phd Thesis, University of Cincinnati, Ohio.

Xiang, C., Wang, X., Ma, Y. & Xu, B., (2015). Practical modeling and comprehensive system identification of a BLDC motor. *Mathematical Problems in Engineering*, 1-12.

Yıldırım, Ş., Çabuk, N. & Bakırcıoğlu, V., (2019). Design and trajectory control of universal drone system. *Measurements*, *147*, 1-9.

Yoon, G. Y., Yamamoto, A. & Lim, H. O., (2016). Mechanism and neural network based on PID control of quadcopter. *International Conference on Control, Automation and Systems*, *16*, 19-24.

Yudho Suprapto, B., Suprijono, H., Muliadi, J. & Kusumoputro, B., (2017). Design and development of heavy-lift hexacopter for heavy payload. *International Seminar on Application for Technology of Information and Communication*, 242-248.

Zhang, B., Tsuchiya, S. & Lim, H., (2021). Development of a lightweight octocopter drone for monitoring complex indoor environment. *Asia-Pacific Conference on Intelligent Robot Systems*, *6*, 57-61.

Zhu, H. Nie, H., Zhang, L., Wei, X. & Zhang, M. (2020). Design and assessment of octocopter drones with improved aerodynamic efficiency and performance. *Aerospace Science and Technology*, *106*, 1-16.

Zulu, A. & John, S., (2014). A review of control algorithms for autonomous quadrotors. *Open Journal of Applied Sciences*, 547-556.

APPENDICES

APPENDIX 1: Nomenclature

x	The position on the x-axis
y	The position on the y-axis
z	The position on the z-axis
v	The velocity vector
Q	Origin of the earth-fixed coordinate system
B	Origin of the body-fixed coordinate system
R	The rotation matrix
a	The arbitrary vector
i	The imaginary value for the x-axis
j	The imaginary value for the y-axis
k	The imaginary value for the z-axis
q	The quaternion vector
W	The transformation matrix
S	The scalar value of the position
V	The imaginary value of the position
F	The force
M	The torque
L	The angular momentum
p	The linear momentum
l	The arm length
m	Total weight of the octocopter
I	The moment of inertia
r_{M_i}	Distance between rotor center and origin
c_T	Thrust constant
c_Q	Torque constant
T	Generated force by each rotor
Q	Generated torque by each rotor

F_M	The total force acts on the octocopter
M_M	The total torque act on the octocopter
$C(s)$	The transfer function
$u(t)$	The control function
K_p	The proportional gain
K_i	The integral gain
K_d	The derivative gain
e	The error value
s	Time derivatice of laplace transform
T_d	The derivative time
T_i	The integral time
T_u	The oscillation period
K_u	The ultimate gain
$y(t)$	The output function for NN
$y(t-d)$	The previous output function for NN
$x(t)$	The input function for NN
$x(t-d)$	The previous input function for NN
d	The delay number
J	The jacobian matrix

Greek symbols

Γ	The position vector
η	The angular position vector
ϕ	The roll angle
θ	The pitch angle
ψ	The yaw angle
π	The Pi constant
ω	The angular velocity
τ	The variable integration of time
μ	Editable parameter

$\Delta\underline{x}$	The weight vector

Abbreviations

TDi	The total number of inputs
TDL	The total number of outputs
W	Watt
V	Voltage
HZ	Hertz
H	Henry
PWM	Pulse width modulation
DC	Direct current
RPM	Round per minute
DAC	Digital to analog converter
PDF	Probability density function
ENU	East, north, and vertical up
NED	North, east, and down
DOF	Degrees of freedom
FW	Fixed-wing
VTOL	Vertical take-off and land vehicle
NARMA-L2	Nonlinear autoregressive moving average
FFNN	Feedforward neural network
CNN	Convolutional neural network
ASM	Adaptive sliding mode
EKF	Extended Kalman filter
DIC	Direct inverse control
LQR	Linear quadratic regulator
ILC	Iterative learning control
MPC	Model predictive control
SMC	Sliding mode control
PCA	Principal component analysis

MLP	Multi-layer perceptron
CBL	Components labeling
CFD	Computational fluid dynamics
FST	Frenet-Serret theory
PID	Proportional-integrator-derivative
NN	Neural networks
NARX	Non-linear autoregressive exogenous input
RMSE	Root mean squared error
MSE	Mean squared error
UAV	Unmanned aerial vehicle

Subscripts

max	Maximum
min	Minimum