TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING

DESIGN OF INTELLIGENT CONTROL SYSTEM FOR UNMANNED AERIAL VEHICLE

MEHITAMATA ÕHUSÕIDUKI INTELLIGENTSE JUHTSÜSTEEMI KAVANDAMINE

MASTER THESIS

Student: AMANUEL SAHLE GEBREYESUS

Student code: 196646MAHM

Supervisors: PROFESSOR OLEG BORISOV
ENGINEER EVEN SEKHRI

AUTHOR’S DECLARATION

Hereby I declare, that I have written this thesis independently.

No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

“.......” .... 2020.

Author: Amanuel Sahle Gebreyesus

/signature /

Thesis is in accordance with terms and requirements “.......” .... 2020.

Supervisor: Professor Oleg Borisov, Engineer Even Sekhri

/signature1/ /signature2/

Accepted for defense “.......” ....2020.

Chairman of theses defense commission: ....

/name and signature/

Non-exclusive License for Publication and Reproduction of GraduationTthesis[1]

I, Amanuel Sahle Gebreyesus, (Date of birth: 17.06.1988)

hereby

1. grant Tallinn University of Technology (TalTech) a non-exclusive license for my thesis

Design of intelligent control system for unmanned aerial vehicle

supervised by

Professor Oleg Borisov, Engineer Even Sekhri,

1.1 reproduced for the purposes of preservation and electronic publication, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright;

1.2 published via the web of TalTech, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright.

1.3 I am aware that the author also retains the rights specified in clause 1 of this license.

2. I confirm that granting the non-exclusive license does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

---

[1] *Non-exclusive License for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university`s right to reproduce the thesis only for preservation purposes.*

______________(*signature*)

______________(*date*)

Mechatronics
THESIS TASK

Student: Amanuel Sahle Gebreyesus, 196646MAHM

Study programme: MAHM02/18
main speciality: Mechatronics
Supervisor(1): Oleg Borisov, Associate Professor

Supervisor(2): Even Sekhri, Engineer

Thesis topic:

(in English) *Design of Intelligent Control System for Unmanned Aerial Vehicle*

(in Estonian) Armastamata Õhusõiduki Intellektse Juhtimissüsteemi Kava

Thesis main objectives:

1. Derivation of kinematic and dynamical equations.
2. Design position and attitude control of a quadcopter.
3. Implement designed control in Simulink/Matlab.

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Literature review	January
2.	Quadcopter dynamics	February
3.	Control design	March
4.	Implementation on Simulink/Matlab	April
5.	Modification and finalizing of thesis	May
6.	Thesis Defense	June

Language: English **Deadline for submission of thesis:** “…….”…………2020

Student: Amanuel Sahle Gebreyesus ……………… “….”…………………….2020
/signature/

Supervisor(1): Oleg Borisov ……………………… “…..”……………………..2020
/signature/

Supervisor(2): Even Sekhri ……………………… “…..”……………………..2020
/signature/

Head of study programme:

……………………………….. ……………………………….. “……” ……………………2020
/signature/

TABLE OF CONTENTS

AUTHOR’S DECLARATION ii

Non-exclusive License for Publication and Reproduction of GraduationTthesis¹ iii

THESIS TASK iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES viii

LIST OF ABBREVIATIONS ix

LIST OF SYMBOLS x

INTRODUCTION 1

CHAPTER 1 4

- 1. LITERATURE REVIEW 4
  - 1.1. Quadcopter history 4
  - 1.2. Quadcopter Control Systems 6
  - 1.3. Conclusion 9

CHAPTER 2 11

- 2. MATHEMATICAL MODELLING 11
  - 2.1. Preliminary notions 11
  - 2.2. Kinematics 13
  - 2.3. Dynamics 15
  - 2.4. Forces and moments 16

CHAPTER 3 22

- 3. FLIGHT CONTROL DESIGN 22
  - 3.1. Control Problem Statement 22
  - 3.2. Controller design 23
    - 3.2.1. Stability 26
  - 3.3. Attitude trajectory generation 26
  - 3.4. Adaptive control design using ANN 27

CHAPTER 4 30

4. SIMULATION RESULTS 30
4.1. Trajectory command 1 32
4.2. Trajectory command 2 39
CHAPTER 5 43
5. CONCLUSIONS AND FUTURE RECOMMENDATIONS 43
5.1. Summary and conclusion 43
5.1.1. Transient and steady-state response 43
5.1.2. Robustness..43
5.1.3. Tracking Performance 44
5.1.4. Adaptivity 44
5.1.5. Attitude command generation 44
5.2. Limitations 44
5.3. Future recommendations..45
REFERENCES 46

LIST OF FIGURES

Figure 1.1 Oehmichen Helicopter No 2, France-1924 [5].. 4
Figure 1.2 De Bothezat's helicopter [6] 5
Figure 1.3 Convortawings model A quadcopter [5] 5
Figure 1.4 VZ-7 quadcopter [5] 6
Figure 1.5 Quadcopter control system and its dynamics. 7
Figure 2.1 Inertial and body frames. 11
Figure 2.2 Euler angles and linear velocities in body frame. 12
Figure 2.3 Thrusts and directions of angular speed of rotors. 17
Figure 3.1 Block diagram of the quadcopter control system. 27
Figure 3.2 Structure of the neural network. 28
Figure 4.1 Simulink block diagram of the whole system. 30
Figure 4.2 X-position response for tanh and sigmoid functions 32
Figure 4.3 Y-position response for tanh and sigmoid functions...............................32
Figure 4.4 X-state response of the SMC and SMC with NN controllers. 33
Figure 4.5 Y-state response of the SMC and SMC with NN controllers. 34
Figure 4.6 Z-state response of the SMC and SMC with NN controllers. 34
Figure 4.7 Plot of Euler angles. 36
Figure 4.8 Linear velocities in the body frame. 37
Figure 4.9 Angular rates in the body frame. 37
Figure 4.10 Output of the four controllers. 38
Figure 4.11 Values of the sliding surface coefficient. 39
Figure 4.12 Trajectory tracking of the X-state. 40
Figure 4.13 Trajectory tracking of the Y-state. 40
Figure 4.14 Trajectory tracking of the Z-state. 41
Figure 4.15 Three-dimensional view of the trajectory followed. 42
Figure 4.16 Lateral view of the trajectory followed. 42

LIST OF TABLES

Table 4.1 Quadcopter model parameter values. 31
Table 4.2 Controller parameters used for simulation. 31
Table 4.3 Steady-state transient response performance indicators. 35

LIST OF ABBREVIATIONS

VTOL	Vertical takeoff and landing
UAV	Unmanned aerial vehicle
PID	Proportional integral derivative
LQR	Linear quadratic regulator
MPC	Model predictive control
SMC	Sliding mode control
GENEFIS	Generic evolving neuro-fuzzy inference system
DTSMC	Dynamic terminal sliding mode control
AFGS-SMC	Adaptive fuzzy gain scheduling sliding node control
ANGS-SMC	Adaptive Neural Gain Scheduling sliding node control
FFT	Fast Fourier transform
DOF	Degrees of freedom
ANN	Artificial neural networks
NN	Neural networks
tanh	Hyperbolic tangent

LIST OF SYMBOLS

$\mathcal{F}_B$	Body frame
$\mathcal{F}_E$	Inertial frame of reference
ξ	Position of the origin of the body frame in the inertial frame
η	Attitude of the quadcopter in inertial frame
ε	Position and attitude of the quadcopter in inertial frame
x, y, z	X, Y and Z values of the quadcopter in the inertial frame
φ, θ, ψ	Roll, pitch and yaw angles
V_B	Linear velocities in the body frame
ω_B	angular rates in body frame
ν	Generalized velocity vector in body frame
u, v, w	Linear velocities in X, Y and Z directions of the body frame
p, q, r	Angular rates about X, Y and Z directions of the body frame
R	Rotation matrix
T	Transformation matrix
m	Quadcopter mass
F_B	Total force acting on the quadcopter body
I	Inertia matrix
M_B	Total torque and moment acting on the body
F_g	Gravitational force in the body frame
T_i	Thrust force produced by rotor i
C_T	Thrust proportionality coefficient
H_i	Torque created by rotor i
C_H	Torque coefficient
Ω	Motor angular speed
J_m	Inertia of each rotor
C_d	Drag force coefficient
C_a	Drag torque coefficient
s_{x_i}	Sliding surface

α_{x_i}	Slope of sliding surface
β_{x_i}	Coefficient of the reaching law
γ	Neural network learning rate

INTRODUCTION

Nowadays, drones especially quadcopters are being more widely applied in vast areas of applications including commercial purposes. The increase in commercial use is due to more efficient, smaller and cheaper electronic components, more powerful processors, more reliable and cheaper sensors, and better battery life. Drones are being applied in a wide area of applications like transportation, first aid, journalism, recreation, and military. And this wide interest motivated the number of researches and investments on drones to increase.

Quadcopters, also called quadrotors, are the most popular type of small-sized drones. This is because they offer some better features than other drones. First, the mechanical design of the quadcopter is simpler than other types of drones like helicopters. The only moving parts of the body are the rotors with the propellers, which have fixed pitch. Second, quadcopters are more efficient than helicopters for not having a tail rotor. They use the counter-rotation method in their rotors for torque balance. In addition to this, because of the small-sized propellers, quadcopters are safer. This makes them preferable in indoor and outdoor applications. Also, the ability of quadcopter to hover and VTOL (vertical takeoff and landing) adds to their popularity.

Generally, a quadcopter is a small unmanned aerial vehicle (UAV), containing four identical rotors. The rotors are placed diagonally across a square configuration in an upward direction. Fixed pitched propellers are mounted on them to produce thrust and torque, which are responsible for the motion of the quadcopter. Opposite rotors are considered as a pair and have the same direction of rotation. If one pair rotates in an anti-clockwise direction, the other rotor pair should rotate in a clockwise direction.

By altering the speed of the rotors different cases of movement can be executed. In the first case, if all the rotors have the same speed, the rotors produce the same thrust and the quadcopter changes altitude or hovers. In the second case, a quadcopter changes its yaw angle by giving more speed to one pair. And in the third case, if one of the rotors in a pair has more speed and the other less speed, the quadcopter can change its roll or pitch angles.

Problem statement

Even though quadcopter possesses the advantages mentioned in the introduction, their flight control is not an easy task. This arises from its non-linearity and uncertainty in its dynamics, under-actuated design, and coupled non-linear dynamics [1]–[3]. Many flight control algorithms for quadcopters have been proposed by researchers, each having their advantages and disadvantages. These control algorithms differ in their robustness, adaptiveness, intelligence, tracking ability, precision, disturbance rejection, etc. Many researchers prefer to use linear controllers for simplicity purpose. And a big amount of the researches is simulation-based in which it is necessary to prove their performance practically.

In this thesis work, based on sliding mode control and neural networks a full flight controller (position and attitude) of an autopilot for a quadcopter will be designed and implemented practically.

Motivation and objectives

Nowadays, we can't think of many areas where quadcopters cannot be applied. With this wide application, quadcopters come with specific designs and challenges. Bearing this in mind, one can guess that the studies needed to master the technology of quadcopters is at its beginning stage. Drone technology is a growing field, in which its impact will change the traditional ways of different sectors like transportation.

Even though, in the future, quadcopters will have wide usage in our lives, the design and control system of quadcopters needs a lot of research and must be improved. Talking about dynamics, not only its non-linearity but having four inputs and six outputs, quadcopter dynamics is underactuated. This makes controlling quadcopters challenging and at the same time interesting. The second point to note is that their propellers being fixed pitch, quadcopter's response to command input to change their motion is slow when compared with helicopters. And finally, the low-quality sensors equipped in nowadays quadcopters are hindering the controllability and stability of quadcopters from reaching its best.

Hence, with the wide area of application, the various purposes and challenges and the positive impact of quadcopters in society, the study of controlling and designing quadcopters makes it an interesting field. And this project aims at improving current challenges in controlling quadcopters partially.

As a result of this thesis project, a sliding mode control based full flight controller of an autopilot for quadcopter will be developed in Matlab. A neural network control strategy will be combined to eliminate the chattering problem and to update control parameters.

Simulations will be conducted, checking the tracking error of the attitude and position of the quadcopter. Necessary improvements will be added to diminish the tracking error and improve the robustness of the system.

Outline

The thesis documentation is organized as shown below.

Chapter 1: Here, an overview of the literature will be analyzed. Different control algorithms will be discussed, and one will be proposed for this thesis work.

Chapter 2: In this chapter, the theoretical background of the quadcopter is discussed, and the mathematical model will be formulated.

Chapter 3: In this chapter, a Sliding Mode Controller (SMC) combined with artificial neural networks will be developed for the flight control of the autopilot.

Chapter 4: Here, the simulation results of the flight control system is shown, and the performance measurements are given.

Chapter 5: In the final chapter, the overall conclusion is discussed and recommendations for future improvement and implementation are listed out.

CHAPTER 1

1. LITERATURE REVIEW

In this chapter, the overview of the literature will be analyzed. Different topics will be raised, starting from the early history of quadcopters to the different control theories used in current literature.

1.1. Quadcopter history

The first attempt of flying rotorcraft was done in 1907 by French brothers Jacques and Louis Breguet [4]. They managed to fly their quadcopter, Gyroplane No 1, few feet from the ground although it was very unstable.

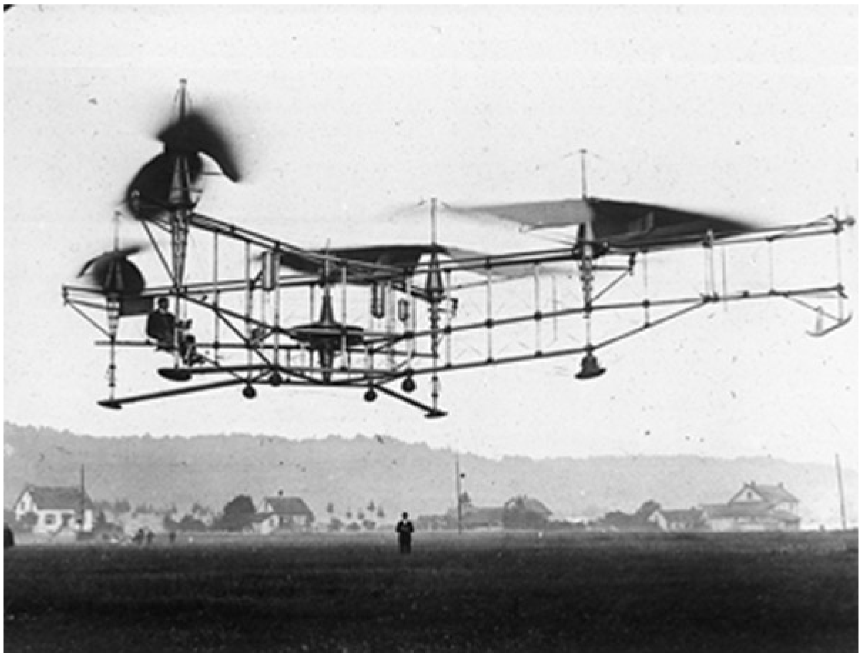

Figure 1.1 Oehmichen Helicopter No 2, France-1924 [5]

In 1924, Étienne Oehmichen was able to fly the Oehmichen No 2, a distance of 360 m and later a distance of 1 kilometer [4]–[6]. Oehmichen No 2 had a single motor that drove four rotors and eight propellers.

At a similar time, George de Bothezat with his assistant Ivan Jerome developed a quadcopter for the US Army and completed several test flights [4], [6]. The quadcopter had six-bladed rotors in an X-configuration. Around 100 test flights were conducted where they attained 5 m as the maximum height. The de Bothezat's helicopter suffered from low power, slow response, mechanical complexity, and high demand for the pilot workload [6].

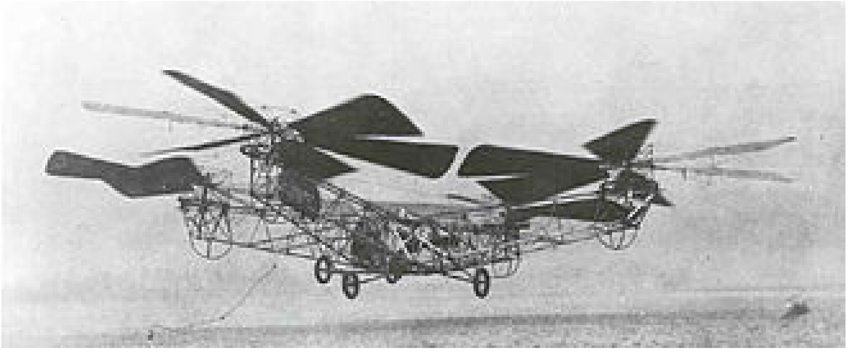

Figure 1.2 De Bothezat's helicopter [6]

In 1956, Convertawings Model A Quadrotor was developed. It had two engines driving four rotors using belts [5], [6]. The control was achieved by varying the thrust without any need of the tail rotor. Many test flights were conducted, and it was the first four-rotor helicopter which resulted in a successful forward flight.

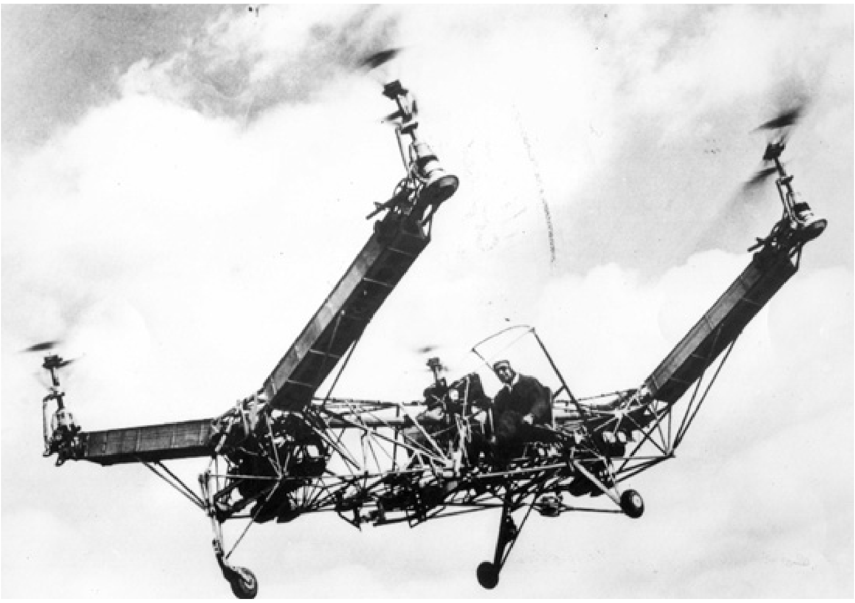

Figure 1.3 Convortawings model A quadcopter [5]

Curtiss-Wright company developed the VZ-7 quadcopter in 1958 for the US army [5], [6]. The VZ-7 quadcopters are the first drones that have a resemblance to nowadays drones. They use a change of thrust of propellers to control motion. Two VZ-7 prototypes were created and performed well in hovering and forward flight. In 1960, the projected was terminated because it failed to meet military standards.

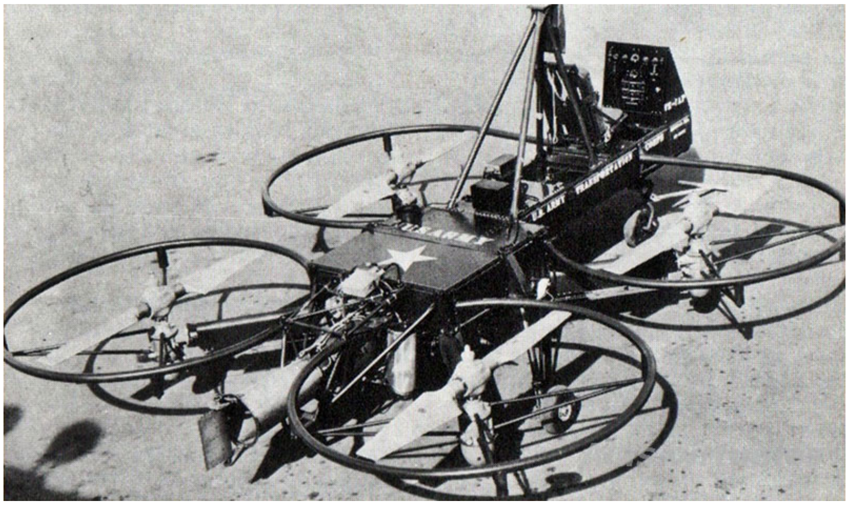

Figure 1.4 VZ-7 quadcopter [5]

1.2. Quadcopter Control Systems

Currently the main problems of quadcopter flight control systems are the non-linear dynamics and under-actuated design. Many researchers proposed linear controllers like PID for the purpose of taking advantage of designing simplicity. But other researches show that the overall performance increases if non-linear controllers are used [7]. The under-actuation problem is solved by coupling different motion dynamics of the quadcopter. Other things that should be addressed are uncertainties in the model, external disturbances, aerodynamic effects, etc. Different algorithms have been proposed specifically to solve some of the challenges mentioned above [1], [3]. A common general block diagram is shown in figure 1.5, which could have different form according to the design method though. As can be noted from the literature below, not all papers address the overall design of the flight controller. Some are focused on designing either attitude controller or altitude controller alone. And another point to be noted is that most of the researches are simulation-based which requires extra practical verifications and modifications. Some of the researches related to Quadcopter Control Systems are discussed in the following paragraphs.

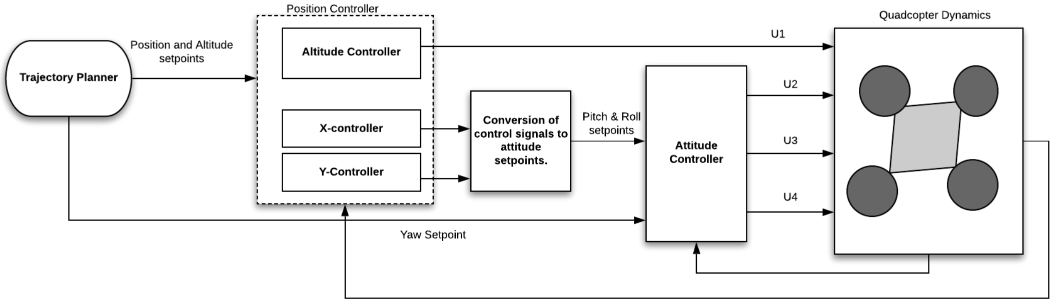

Figure 1.5 Quadcopter control system and its dynamics.

In [8], an improved PID controller is implemented where the altitude controller consists of a multi-loop controller. The outer loop is a multi-mode combination of linear and non-linear controller, the inner loop was a PID controller. The authors proved a smooth transition between the linear and non-linear modes. The performance of the new algorithm is shown in comparison to a conventional PID controller. The authors improved the altitude reference tracking while taking advantage of the computationally simple control system. The controller was employed on the Pixhawk flight controller operating DJI-F450 quadcopter.

Robust output controllers are experimentally verified for roll and pitch angle tracking in [9]. Three robust output controllers, a consecutive compensator and its modifications [10]–[12], were verified. The authors used an integral term in the consecutive compensator to remove steady-state error triggered by external disturbance. And to avoid integrator windup, a back-calculation approach to the controller is used.

In [7], a review of 12 types of controllers for the quadcopter is given. The advantages and disadvantages of several linear and non-linear algorithms namely, PID, Backstepping, Sliding Mode, Linear Quadratic Regulator (LQR), Feedback Linearization, Robust, Adaptive, Optimal, H_∞, L_1, Fuzzy Logic and Artificial Neural Networks are thoroughly analyzed. The authors provided a comparison of the control algorithms in terms of their adaptiveness, robustness, optimality, tracking ability, intelligence, convergence, simplicity, etc. They concluded that even the finest non-linear or linear controllers have limits and proposed hybrid systems offer better performance as they include advantages from more than one control algorithm. Based on the idea of this paper, a hybrid controller is going to be implemented in this thesis work. The preferred control algorithm is Sliding Mode Control combined with some intelligent control algorithms like Artificial Neural Networks or Fuzzy logic.

In [13], the authors implemented a hybrid non-linear position control system composed of MPC and fuzzy feedforward compensator that can eliminate cross-coupling disturbances and external disturbances. The fuzzy feedforward compensator is used to deliver feedforward compensation to the altitude loop. Performance comparisons were made against the conventional PD control systems and they proved good trajectory tracking using simulations. The proposed control system showed robustness to uncertainties.

In [14], a hybrid non-linear position control system composed of traditional PD and PD-type fuzzy was implemented. The control system showed robustness when subjected to external disturbances and uncertainties and good noise rejection capability when the Fuzzy logic compensator was implemented. The results were verified experimentally using the AR.Drone2.

Neuro-fuzzy controller integrated with SMC was implemented in [15] to control the position of a quadcopter. A Generic evolving neuro-fuzzy controller was proposed where the evolving structure was created using an incremental learning strategy called GENEFIS, and the parameters were updated using a learning algorithm based on SMC. The results of the proposed controller were compared to a PID controller and excelled in robustness, tracking error. Although the proposed controller has a slower rise time than the PID but stabilized the quadcopter quicker.

A global fast Dynamic Terminal Sliding Mode Control (DTSMC) is studied in [16] for attitude and position tracking control problem. In this paper, the TSMC is used to achieve finite-time convergence. The terminal switching plane variable has a non-linear term of the velocity error, unlike the conventional linear SMC. The DTSMC proved to eliminate the chattering effect shown in TSMC in addition to finite-time convergence and reduction of error in finite time. This paper focused on finite-time stabilization rather than robust flight control design which is common in the most existing literature.

In [17], a sliding mode control based on least square is proposed for chattering elimination and energy saving. A saturation function is used around the boundary to minimize chattering and sliding surface with integral term is designed to improve tracking capability. The result is compared with PD and PID controllers. And the proposed controller performed well in chattering and tracking error reduction.

In [18], neural-network was combined with SMC for position and attitude control. The designed controller had four sub-controllers, namely the roll angle, pitch angle, yaw angle, and altitude controllers. The coefficients of the sliding surface in all these controllers were adaptively adjusted by the neural network technique. The results showed the cancelation of the chattering effect of SMC. The authors compared their proposed algorithm with those in [19] and [16]. The proposed algorithm outperformed in speed, accuracy, sensitivity to disturbances and parameter variations, and chattering effect. Similarly, the author from [20], used a simple Analog Neural Networks to adaptively adjust the coefficient of the altitude sliding surface. The results were compared with a second-order sliding mode controller in [19] and excelled in performance.

Adaptive Fuzzy Gain Scheduling Sliding Mode Control (AFGS-SMC) is designed in [21] to control attitude. The method proved to eliminate the chattering effect of SMC under external disturbances and parametric uncertainties. The performance of the fuzzy gain scheduling is compared with another technique called the Boundary Layer method and outperformed in chattering elimination. Similarly, adaptive fuzzy gain scheduling is used for similar purposes in [22] for an integral sliding mode controller.

In [23], two techniques to eliminate the chattering effect of SMC for attitude control were used. The authors compared Adaptive Fuzzy Gain Scheduling based SMC (AFGS-SMC) and Adaptive Neural Gain Scheduling based SMC (ANGS-SMC). And FFT of the control torques indicates ANGS-SMC reduced the chattering effect better than AFGS-SMC without compromising the transient characteristics. Hence the authors proved the better performance of simple neural networks over the fuzzy logic system in the elimination of chattering.

1.3. Conclusion

SMC is one of the most popular control algorithms used. Its popularity arises from its quick response, good transient response, good robustness to external disturbances and uncertainties [18]. Besides, it uses fast switching control law to jump the system from any initial state on to the switching surface (reaching or hitting phase) and to keep the states on the surface (sliding phase) for all time [21]. But SMC has some drawbacks. To name some; high-frequency chattering of the control signal, tracking performance in finite time, vulnerability to sensor noise, production of unnecessary large control signal to overcome parametric uncertainties, and dependency on accuracy of plant dynamics. The chattering effect is the most concerning problem of SMC which can be a cause for unnecessary energy loss, hardware damage over time and it may excite unmodelled high-frequency dynamics. Many methods are used to reduce this problem such as boundary layer method, higher-order SMC. However, these methods come with performance and complexity tradeoffs respectively.

Knowledge-based intelligent controllers have become famed for their model-independent methods. They don’t require any model structure. The most common are fuzzy logic and neural network control methods. As seen in the review both methods complemented the SMC in removing the chattering problem and tuned the sliding surface parameters online.

Thus, by combining fuzzy logic and neural networks with SMC a combined advantage can be achieved. The purpose of this thesis is to design a sliding mode controller for a quadcopter where the parameters of the sliding surface are adaptively tuned by neural networks. And the expected results of the hybrid flight controller are; robust against parameter uncertainties and external disturbances, small tracking error, reduced chattering and adaptive behavior.

So, to conclude the chapter, the overview of different articles has been discussed and it would be good to mention the contribution of the thesis. The purpose of the thesis is to design a full flight controller for an autopilot which,

- has fast response and good transient response,
- is robust to uncertainties and external disturbances,
- is adaptive and self-learning to changes and
- stable, accurate, and efficient.

CHAPTER 2

2. MATHEMATICAL MODELLING

In this chapter, the mathematical model of the dynamics of the quadcopter is formulated. Starting from the kinematics of the quadcopter, the dynamic model will be formulated based on Newton-Euler equations. The following formulations are based on [24]–[27].

2.1. Preliminary notions

For formulating the mathematical model, two frames of reference are considered, the body frame ($\mathcal{F}_B$) and the inertial frame of reference ($\mathcal{F}_E$). The origin of the $\mathcal{F}_B$ is presumed to coincide with the center of Gravity of the quadcopter structure. Commonly, two types of configurations are used in quadcopter applications, namely the "+" configuration and the "X" configuration. For this thesis project, the "+" configuration is selected, where the x-axis of the body frame is along the arm of the first rotor, the y-axis is along the second rotor arm and the z-axis completes the right-hand rule. Positive forward motion is assumed in the positive x-axis direction. In $\mathcal{F}_E$, the positive direction of the z-axis points away from Earth's surface.

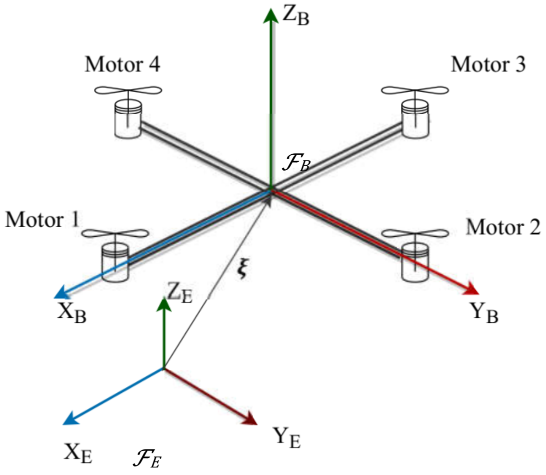

Figure 2.1 Inertial and body frames.

The reason for having two frames of reference is that some quantities of the motion are measured in the body while others are measured in the inertial frame.

The vectors ξ and η define the position and orientation of the body frame in the inertial frame of reference, which are expressed in the following equations [27].

$$\xi = [x\ y\ z]^T \in \mathbb{R}^3 \tag{2.1}$$

$$\eta = [\varphi\ \theta\ \psi]^T \in \mathbb{R}^3 \tag{2.2}$$

$$\varepsilon = [\xi\ \eta] = [x\ y\ z\ \varphi\ \theta\ \psi]^T \in \mathbb{R}^6 \tag{2.3}$$

The position of the quadcopter is defined by three points (x, y, z) in the 3-D inertial frame. Similarly, the attitude is described by three angles (φ, θ, ψ), which are commonly called the Euler angles. These angles (φ, θ, ψ) represent the roll, pitch and yaw angles which are rotations around the x-axis, y-axis and z-axis respectively.

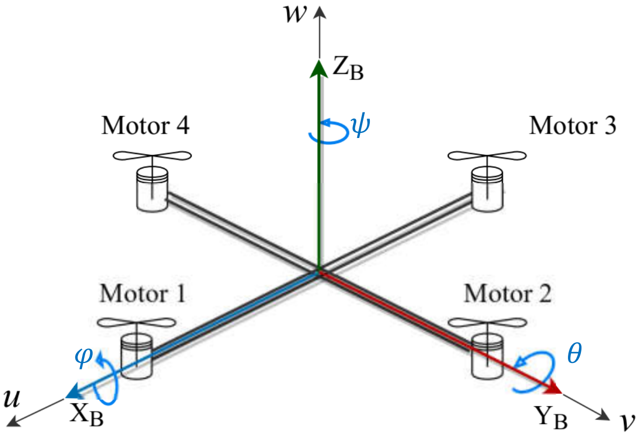

Figure 2.2 Euler angles and linear velocities in body frame.

Linear velocities in the body frame are defined by the vector V_B and similarly, the angular rates are defined by vector ω_B in body frame [27].

$$V_B = [u\ v\ w]^T \in \mathbb{R}^3 \tag{2.4}$$

$$\omega_B = [p\ q\ r]^T \in \mathbb{R}^3 \qquad (2.5)$$

$$\nu = [V_B\ \omega_B\ ] = [u\ v\ w\ p\ q\ r]^T \in \mathbb{R}^6 \qquad (2.6)$$

2.2. Kinematics

In kinematics, the motion of the quadcopter will be described without considering forces and moments acting on the body. The kinematics of the 6 DOF quadcopter is given as follows [27]:

$$\dot{\varepsilon} = J \cdot \nu \qquad (2.7)$$

Where $\dot{\varepsilon}$ – is the generalized velocity vector in the inertial frame,

ν – the is generalized velocity vector in body frame, and

J – is generalized transformation and rotation matrix.

Equation (2.7) can be separately expressed for the linear and angular velocities as follows:

$$\begin{aligned} \dot{\xi} &= R \cdot V_B \\ \dot{\eta} &= T \cdot \omega_B \end{aligned} \qquad (2.8)$$

Where R – is a rotational matrix and

T – is a transformation matrix.

The rotation matrix is used to transform measured linear velocity vectors from the body frame to the inertial frame. The rotation sequence of the rotation matrix is described as a sequence of rotations about the z-axis (yaw), about the y-axis (pitch) and then about the x-axis (roll) consecutively. The rotation matrix is given as follows [24], [25]:

$$R = R_Z(\psi) \cdot R_Y(\theta) \cdot R_X(\varphi)$$

$$R = \begin{bmatrix} c(\theta)c(\psi) & s(\varphi)s(\theta)c(\psi) - c(\varphi)s(\psi) & c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi) \\ c(\theta)s(\psi) & s(\varphi)s(\theta)s(\psi) + c(\varphi)c(\psi) & c(\varphi)s(\theta)s(\psi) - s(\varphi)c(\psi) \\ -s(\theta) & s(\varphi)c(\theta) & c(\varphi)c(\theta) \end{bmatrix} \qquad (2.9)$$

Where s and c represent sine and cosine respectively.

The transformation matrix *T* is used to transform angular rates in the body frame to Euler angular rates in the inertial frame. And it is given as [27]:

$$\mathrm{T} = \begin{bmatrix} 1 & \tan(\theta)\sin(\varphi) & \tan(\theta)\cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \dfrac{\sin(\varphi)}{\cos(\theta)} & \dfrac{\cos(\varphi)}{\cos(\theta)} \end{bmatrix} \qquad (2.10)$$

As can be seen from the transformation matrix, a singularity occurs at $\theta = \pm 90^0$. So, to keep the quadcopter from becoming unstable, the pitch angle should be kept small. Equations (2.8) can be written in matrix form as shown below.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} R & 0_{3X3} \\ 0_{3X3} & T \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \qquad (2.11)$$

Therefore, the kinematic equations are given as:

$$\begin{cases} \dot{x} = u[c(\psi)c(\theta)] - v[c(\varphi)s(\psi) - c(\psi)s(\varphi)s(\theta)] + w[s(\varphi)s(\psi) + c(\varphi)c(\psi)s(\theta)] \\ \dot{y} = u[c(\theta)s(\psi)] + v[c(\varphi)c(\psi) + s(\varphi)s(\psi)s(\theta)] - w[c(\psi)s(\varphi) - c(\varphi)s(\psi)s(\theta)] \\ \dot{z} = -u[s(\theta)] + v[c(\theta)s(\varphi)] + w[c(\varphi)c(\theta)] \\ \dot{\varphi} = p + q[s(\varphi)t(\theta)] + r[c(\varphi)t(\theta)] \\ \dot{\theta} = q[c(\varphi)] - r[s(\varphi)] \\ \dot{\psi} = q\left[\dfrac{s(\varphi)}{c(\theta)}\right] + r\left[\dfrac{c(\varphi)}{c(\theta)}\right] \end{cases} \qquad (2.12)$$

2.3. Dynamics

To formulate the dynamic equations, Euler's equations and Newton's law will be followed. The assumptions shown below are made for the formulation of dynamic equations.

- The quadcopter body is rigid and symmetrical.
- The propellers are also rigid.
- The thrust force and reactive torque are directly proportional to the square of the angular speed of the propellers.
- Ground effect, air drag, air resistance and blade flapping are ignored.

Newton's law for the net forces acting on the quadcopter can be expressed as given below:

$$m\left(\omega_B \times V_B + \dot{V}_B\right) = F_B \tag{2.13}$$

Where m – is mass of the quadcopter and

F_B – is the total force acting on the quadcopter body.

The total moment acting using Euler's equation is expressed as in Equation 2.14.

$$I \cdot \dot{\omega}_B + \omega_B \times (I + \omega_B) = M_B \tag{2.14}$$

Where I – is inertia matrix and

M_B – is the total torque and moment acting on the body.

Assuming the quadcopter body is symmetrical, the off-diagonal elements of the inertia matrix are equal to zero. The inertia matrix is given as follows.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.15}$$

So, the dynamic equations in the body frame can be written as follows:

$$\begin{cases} f_x = m\,(qw - rv + \dot{u}) \\ f_y = m\,(ru - pw + \dot{v}\,) \\ f_z = m\,(pv - qu + \dot{w}) \\ m_x = \dot{p}I_x + qrI_z - qrI_y \\ m_y = \dot{q}I_y + prI_x - prI_z \\ m_z = \dot{r}I_z + pqI_y - pqI_x \end{cases} \tag{2.16}$$

2.4. Forces and moments

Here, the external forces, moments and aerodynamic effects will be discussed.

1. **The gravitational force**: this force is due to the gravity of the Earth and acts on the center of gravity of the quadcopter body. It is directed along the negative z-axis direction of the inertial frame of reference and can be expressed in the body frame as:

$$F_g = -mg\,R^T \tag{2.17}$$

Where F_g – is a gravitational force in the body frame.

2. **Movement thrust and torques**: these forces and torques are influenced by the propeller angular speed, Ω_i. They are controllable and include the thrust force along the positive z-axis direction in the body frame and three torques about each body axis. Each actuator can produce a thrust force, which can be expressed as follows:

$$T_i = C_T\,\Omega_i^2 \tag{2.18}$$

Where T_i – is the thrust force produced by rotor i and

C_T – is the thrust proportionality coefficient

The difference in angular speed between the two pairs of rotors produces a reactive torque (hub torque or yaw) about the z-axis and torque produced by each rotor can be given as:

$$H_i = C_H \, \Omega_i^2 \tag{2.19}$$

Where H_i – is the torque created by rotor i

C_H – is torque coefficient.

The quadcopter has four movement variables, which can be controlled to achieve the desired attitude and position. These are controlled by manipulating the angular speeds of the rotors. A 6 DOF motion is achieved by combining these basic movements. As shown in figure 2.3, opposite propellers should rotate in the same direction. In this thesis the directions are that motors 1 and 3 produce rotations in a counterclockwise direction and motors 2 and 4 in the clockwise direction when seen from the top. The forward movement is in the direction of the positive x-axis.

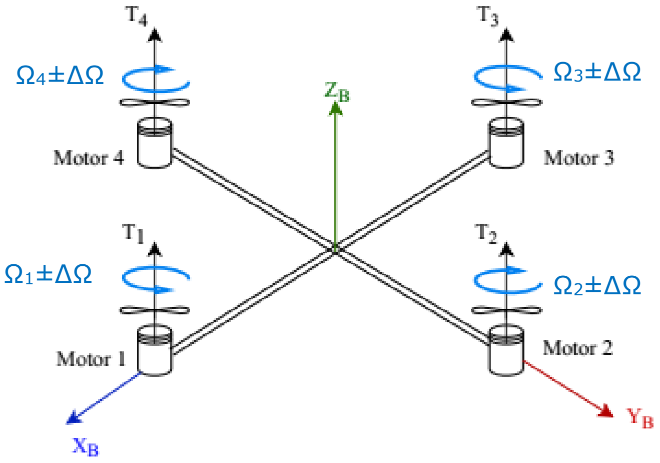
T_4
T_3
$\Omega_4 \pm \Delta\Omega$
Z_B
$\Omega_3 \pm \Delta\Omega$
Motor 4
Motor 3
T_1
T_2
$\Omega_1 \pm \Delta\Omega$
$\Omega_2 \pm \Delta\Omega$
Motor 1
Motor 2
X_B
Y_B

Figure 2.3 Thrusts and directions of angular speed of rotors.

The first controlled movement U_1, which is in the positive direction of Z_B, is realized by producing the same amount of thrust. The total thrust can be increased or decreased by increasing or increasing the amount of angular speed in all rotors equally. The total thrust creates pure vertical motion when the pitch and roll angles are equal to zero. If these angles are non-zero, a combination of horizontal and vertical motions is realized. Controlled movement U_1 is expressed as follows:

$$U_1 = F_T = C_T\,(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{2.20}$$

Where F_T - is the sum of all thrusts, $\Sigma\, T_i$.

The second controlled movement U_2 is the roll motion τ_φ, which is torque about X_B. Positive rotation about the x-axis is realized by increasing the angular speed of propeller 2 and decreasing that of propeller 4 by the same quantity. This positive rotation about the x-axis creates a translational motion in the negative direction of the y-axis. The amount of change in angular speed should be selected in such a way that the roll motion doesn’t affect the vertical motion. Controlled movement U_2 is expressed as follows.

$$U_2 = \tau_\varphi = lC_T\,(\Omega_2^2 - \Omega_4^2) \tag{2.21}$$

Where l – is the length between the axis of rotation of the rotor and center of the quadcopter.

Controlled movement U_3 is the pitch motion τ_θ, which is torque about Y_B. Positive rotation about the y-axis is realized by increasing the angular speed of propeller 3 and decreasing that of propeller 1 by the same quantity. This positive rotation about the y-axis creates a translational motion in the positive direction of the x-axis. Similarly, the amount of change in angular speed should be selected in such a way that the pitch motion doesn’t affect the altitude position. The controlled movement U_3 is expressed as shown below:

$$U_3 = \tau_\theta = lC_T\,(\Omega_3^2 - \Omega_1^2) \tag{2.22}$$

The last controlled movement U_4 is the yaw motion τ_ψ, which is torque about Z_B. This torque is created due to the unbalance of moments of the pair rotors. To get a positive rotation about the z-axis, the angular speed of the counterclockwise rotating propellers (propeller 1 and 3) should be decreased and the angular speed of the other pair should be increased. Controlled movement U_4 is expressed as follows:

$$U_4 = \tau_\psi = C_H\,(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \tag{2.23}$$

The above controlled thrust and torques can be written in matrix form as shown below.

$$\begin{bmatrix} \Sigma\, T_i \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} C_T & C_T & C_T & C_T \\ 0 & lC_T & 0 & -lC_T \\ -lC_T & 0 & lC_T & 0 \\ -C_H & C_H & -C_H & C_H \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{2.24}$$

3. **Gyroscopic torque**: this torque occurs due to the alteration in the axis of rotation of the rotating body. It depends on the angular speed of the propellers, the motor's inertia, and the rolling pitching rates (p and q).

$$\tau_{\theta gyro} = J_m\, p \left(\frac{\pi}{30}\right) (\, \Omega_1 + \Omega_3 - \Omega_2 - \Omega_4 \,)$$

$$\tau_{\varphi gyro} = J_m\, q \left(\frac{\pi}{30}\right) (- \Omega_1 - \Omega_3 + \Omega_2 + \Omega_4 \,) \tag{2.25}$$

Where J_m – is the inertia of each rotor.

4. **Drag force and torque**: these are created due to the aerodynamic drag and air frictions. They can be modeled in terms of velocities and angular rates as follows.

$$F_d = \begin{bmatrix} C_{dx}\dot{x} \\ C_{dy}\dot{y} \\ C_{dz}\dot{z} \end{bmatrix}$$

$$\tau_d = \begin{bmatrix} C_{ax}\dot{\varphi} \\ C_{ay}\dot{\theta} \\ C_{az}\dot{\psi} \end{bmatrix} \tag{2.26}$$

Where F_d and τ_d – are drag force and torque.

C_d and C_a – are drag force and torque coefficients.

There are other aerodynamic effects which can influence the flight dynamics. The ground effect, which occurs when flying near the ground, and blade flapping are going to be neglected in this thesis. Air resistance and air drag will be considered in the acceleration equations.

Therefore, considering the moments and forces discussed above, equations (2.16) can be re-written as:

$$\begin{cases} mg[s(\theta)] \ = \ m\,(\dot{u} \ + qw - rv) \\ -mg[c(\theta)s(\varphi)] \ = \ m\,(\dot{v} \ + ru - pw) \\ -mg[c(\theta)c(\varphi)] + \ F_T \ = \ m\,(\dot{w} \ + pv - qu) \\ \tau_{\varphi} + \tau_{\varphi gyro} \ = \ \dot{p}I_x + qr\big(I_z - I_y\big) \\ \tau_{\theta} + \tau_{\theta gyro} \ = \ \dot{q}I_y + pr\,(I_x - I_z) \\ \tau_{\psi} \ = \ \dot{r}I_z + pq\big(\,I_y - I_x\big) \end{cases} \tag{2.27}$$

Equation (2.27) can be re-written as follows.

$$\begin{cases} \dot{u} = \ (\,rv - qw) + \dfrac{f_x}{m} \\ \dot{v} = \ (\,pw - ru) + \dfrac{f_y}{m} \\ \dot{w} = \ (\,qu - pv) + \dfrac{f_z}{m} \\ \dot{p} = \ \dfrac{m_x}{I_x} + \dfrac{qr\,\big(I_y - \ I_z\big)}{I_x} \\ \dot{q} = \ \dfrac{m_y}{I_y} + \dfrac{pr\,(\,I_z - I_x)}{I_y} \\ \dot{r} = \ \dfrac{m_z}{I_z} + \dfrac{pq\,\big(\,I_x - I_y\big)}{I_z} \end{cases} \tag{2.28}$$

Considering equations (2.12) and (2.28) a vector of 12 states can be formed as follows.

$$S = \ [x\;y\;z\;u\;v\;w\;\varphi\;\theta\;\psi\;p\;q\;r] \ \in \ \mathbb{R}^{12} \tag{2.29}$$

Using Newton's second law and Euler's kinematic equation (equations 2.14), a second variant of dynamic state equations, which is useful for designing a control system, can be formulated. Air drag and air resistance will be considered in the following equations. According to Newton's second law,

$$m\,\dot{V}_B = R\,F_b - F_d = F_T\,R\cdot\hat{z} - mg\begin{bmatrix}0\\0\\1\end{bmatrix} - C_d\begin{bmatrix}\dot{x}\\\dot{y}\\\dot{z}\end{bmatrix} \tag{2.30}$$

Where $\hat{z}$ – unit vector in the z-direction,

F_T – thrust force,

Therefore, the position state equations can be given as:

$$\begin{cases}\ddot{x} = \dfrac{F_T}{m}[c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi)] - \dfrac{C_{dx}}{m}\dot{x}\\ \ddot{y} = \dfrac{F_T}{m}[c(\varphi)s(\theta)s(\psi) - s(\varphi)c(\psi)] - \dfrac{C_{dy}}{m}\dot{y}\\ \ddot{z} = \dfrac{F_T}{m}[c(\varphi)c(\theta)] - \dfrac{C_{dz}}{m}\dot{z} - g\end{cases} \tag{2.31}$$

To formulate attitude state equations, an assumption $[\dot{\varphi}\;\dot{\theta}\;\dot{\psi}]^T = [p\;q\;r]^T$, is made considering small angles of movement [24]. Torque resulting from air drag and resistance is introduced. The last three equations of (2.28) can be re-written as follows:

$$\begin{cases}\ddot{\varphi} = \dfrac{\tau_x}{I_x} + \dfrac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} - \dfrac{\tau_{\varphi gyro}}{I_x} - \dfrac{C_{ax}}{I_x}\dot{\varphi}\\ \ddot{\theta} = \dfrac{\tau_y}{I_y} + \dfrac{(I_z - I_x)}{I_y}\dot{\varphi}\dot{\psi} - \dfrac{\tau_{\theta gyro}}{I_y} - \dfrac{C_{ay}}{I_y}\dot{\theta}\\ \ddot{\psi} = \dfrac{\tau_z}{I_z} + \dfrac{(I_x - I_y)}{I_z}\dot{\varphi}\dot{\theta} - \dfrac{C_{az}}{I_z}\dot{\psi}\end{cases} \tag{2.32}$$

Considering equations (2.12), (2.31) and (2.32) another vector of 12 states can be formed as follows.

$$S = [x\;\dot{x}\;y\;\dot{y}\;z\;\dot{z}\;\varphi\;\dot{\varphi}\;\theta\;\dot{\theta}\;\psi\;\dot{\psi}] \in \mathbb{R}^{12} \tag{2.33}$$

CHAPTER 3

3. FLIGHT CONTROL DESIGN

In this chapter, a flight controller for the attitude and position tracking will be designed. A sliding mode control will be designed. Then with the aid of a simple neural network back propagation technique, a parameter of the sliding surface will be tuned for adaptation purposes. The full control structure includes an outer-loop that controls the position and an inner-loop which controls the attitude of the quadcopter.

3.1. Control Problem Statement

Equations (2.31) and (2.32) can be re-written as compact affine non-linear space form as follows:

$$\begin{cases} \dot{x}_i = x_{i+1} \\ \dot{x}_{i+1} = f(x) + h(x)u_j + d(x) \end{cases} \tag{3.1}$$

Where x_i - represents the 12 states of equation (2.33),

$f(x)$ and $h(x) \neq 0$ - are non-linear, real and known functions of x,

$d(x)$ - represents disturbances and uncertainties, and

u_j - represents the four control inputs.

In designing the controller, $d(x)$ will be considered as unmodelled dynamics and it will be ignored.

The control problem is to track position and attitude asymptotically, $x_i \rightarrow x_{id}$, in the occurrence of parametric uncertainties in $f(x)$ and $h(x)$ and disturbances $d(x)$.

3.2. Controller design

To design the controller, the dynamic equations (2.31) and (2.32) can be re-written in the form of equation (3.1) as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = [0] + \left[\frac{c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi)}{m}\right]U_1 + \left[-\frac{C_{dx}}{m}\dot{x}\right] \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = [0] + \left[\frac{c(\varphi)s(\theta)s(\psi) - s(\varphi)c(\psi)}{m}\right]U_1 + \left[-\frac{C_{dy}}{m}\dot{y}\right] \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = [-g] + \left[\frac{c(\varphi)c(\theta)}{m}\right]U_1 + \left[-\frac{C_{dz}}{m}\dot{z}\right] \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = [0] + \left[\frac{1}{I_x}\right]U_2 + \left[\frac{\left(I_y - I_z\right)}{I_x}\dot{\theta}\dot{\psi} - \frac{\tau_{\varphi gyro}}{I_x} - \frac{C_{ax}}{I_x}\dot{\varphi}\right] \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = [0] + \left[\frac{1}{I_y}\right]U_3 + \left[\frac{(I_z - I_x)}{I_y}\dot{\varphi}\dot{\psi} - \frac{\tau_{\theta gyro}}{I_y} - \frac{C_{ay}}{I_y}\dot{\theta}\right] \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = [0] + \left[\frac{1}{I_z}\right]U_4 + \left[\frac{\left(I_x - I_y\right)}{I_z}\dot{\varphi}\dot{\theta} - \frac{C_{az}}{I_z}\dot{\psi}\right] \end{cases} \tag{3.2}$$

Driving the control law for the above dynamic equation is similar. The altitude controller is designed below.

The sliding surface or manifold where the sliding motion takes place is defined by:

$$s_z = \alpha_z(z_d - z) + (\dot{z}_d - \dot{z}) \tag{3.3}$$

Where $\alpha_z > 0$ – is slope of the sliding line.

z_d – is the desired value of z.

The time derivative of the sliding surface is given in equation (3.4) and equation (3.5) is found by replacing the altitude dynamics in equation (3.4). As mentioned above, $d(x)$ will be ignored which represents the unmodelled dynamics and disturbances.

$$\dot{s}_z = \alpha_z(\dot{z}_d - \dot{z}) + (\ddot{z}_d - \ddot{z}) \tag{3.4}$$

$$\dot{s}_z = \alpha_z(\dot{z}_d - \dot{z}) + \ddot{z}_d - \frac{c(\varphi)c(\theta)}{m}U_1 + g \tag{3.5}$$

Let the reaching law be defined by:

$$\dot{s}_z = -\beta_z \tanh(s_z) \tag{3.6}$$

Where β_z – a positive design parameter

By substituting equation (3.6) into equation (3.5), the control law is designed as:

$$U_1 = m\left[\frac{\alpha_z(\dot{z}_d - \dot{z}) + \ddot{z}_d + g + \beta_z \tanh(s_z)}{c(\varphi)c(\theta)}\right] \tag{3.7}$$

The other controllers are also designed in a similar way. The altitude command derived from the x-dynamics is given by equation (3.8).

$$U_1 = m\left[\frac{\alpha_x(\dot{x}_d - \dot{x}) + \ddot{x}_d + \beta_x \tanh(s_x)}{c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi)}\right] \tag{3.8}$$

The altitude command derived from the y-dynamics is given by equation (3.9).

$$U_1 = m\left[\frac{\alpha_y(\dot{y}_d - \dot{y}) + \ddot{y}_d + \beta_y \tanh(s_y)}{c(\varphi)s(\theta)s(\psi) - s(\varphi)c(\psi)}\right] \tag{3.9}$$

The roll command is given by equation (3.10).

$$U_2 = I_x\left[\alpha_\varphi(\dot{\varphi}_d - \dot{\varphi}) + \ddot{\varphi}_d + \beta_\varphi \tanh{(s_\varphi)}\right] \tag{3.10}$$

The pitch command is given by equation (3.11).

$$U_3 = I_y\left[\alpha_\theta(\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d + \beta_\theta \tanh{(s_\theta)}\right] \tag{3.11}$$

The yaw command is given by equation (3.12).

$$U_4 = I_z\left[\alpha_\psi(\dot{\psi}_d - \dot{\psi}) + \ddot{\psi}_d + \beta_\psi \tanh{(s_\psi)}\right] \tag{3.12}$$

As can be seen from equations (3.7), (3.8) and (3.9) the altitude control law is overdetermined. So, to approximate the altitude control law, the least-squares method [17] is used. Equations (3.7), (3.8) and (3.9) can be rewritten in the following matrix from.

$$U_1 \begin{bmatrix} c(\varphi)s(\theta)c(\psi) + s(\varphi)s(\psi) \\ c(\varphi)s(\theta)s(\psi) - s(\varphi)c(\psi) \\ c(\varphi)c(\theta) \end{bmatrix} = m \begin{bmatrix} O_x \\ O_y \\ O_z + g \end{bmatrix} \tag{3.13}$$

Where $O_x = \alpha_x(\dot{x}_d - \dot{x}) + \ddot{x}_d + \beta_x \tanh{(s_x)}$,

$O_y = \alpha_y(\dot{y}_d - \dot{y}) + \ddot{y}_d + \beta_y \tanh{(s_y)}$ and

$O_z = \alpha_z(\dot{z}_d - \dot{z}) + \ddot{z}_d + g + \beta_z \tanh{(s_z)}$

After rearranging equation (3.13) it can be written as given below.

$$ABU_1 = C \tag{3.14}$$

Where $A = \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ s(\psi) & -c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} c(\varphi)s(\theta) \\ s(\varphi) \\ c(\varphi)c(\theta) \end{bmatrix}$ and $C = m \begin{bmatrix} O_x \\ O_y \\ O_z + g \end{bmatrix}$

Now, applying the least-squares algorithm to equation (3.14), we get

$$
\begin{aligned}
A^T A B U_1 &= A^T C \\
B U_1 &= A^T C
\end{aligned}
\tag{3.15}
$$

Matrix A is orthogonal and its square yields an identity matrix. Squaring equation (3.15) and computing for U_1 the following altitude control law is achieved.

$$
\begin{aligned}
B^T B U_1^2 &= (A^T C)^T A^T C \\
U_1 &= m\sqrt{O_x^2 + O_y^2 + (O_z + g)^2}
\end{aligned}
\tag{3.16}
$$

3.2.1. Stability

To proof the stability of the controllers, the following Lyapunov function candidate is selected. The derivative of the function $\dot{V}$, will be negative if β_{x_i} is selected as real positive, which ensures the Lyapunov stability of the system.

$$
\begin{aligned}
V &= \frac{1}{2} s_{x_i} \\
\dot{V} &= s_{x_i} \dot{s_{x_i}} = -\beta_{x_i}\, s_{x_i} \tanh\left(s_{x_i}\right)
\end{aligned}
\tag{3.17}
$$

3.3. Attitude trajectory generation

In autonomous quadcopters the desired position and yaw is inputted by the operator. This is because of the underactuated behavior of the quadcopter. The desired trajectories in x and y-directions are tracked by pitching and rolling the quadcopter respectively. Therefore, the desired trajectories of roll and pitch angles are determined from the position controller values in the outer loop as in [17]. Expanding the matrix of equation (3.15), we get

$$\begin{aligned} c(\varphi)s(\theta)\,U_1 &= m(c(\psi)O_x + s(\psi)O_y) \\ s(\varphi)\,U_1 &= m(s(\psi)O_x - c(\psi)O_y) \\ c(\varphi)c(\theta)\,U_1 &= m(O_z + g) \end{aligned} \tag{3.18}$$

Solving for roll and pitch angles from equations (3.17) and expressing them as desired values yields equation (3.18).

$$\begin{aligned} \varphi_d &= \arctan\left(\frac{s(\psi_d)O_x - c(\psi_d)O_y}{\sqrt{(c(\psi_d)O_x + s(\psi_d)O_y)^2 + (O_z + g)^2}}\right) \\ \theta_d &= \arctan\left(\frac{c(\psi_d)O_x + s(\psi_d)O_y}{O_z + g}\right) \end{aligned} \tag{3.19}$$

Figure 3.1 shows the full block diagram of the quadcopter control structure and dynamics.

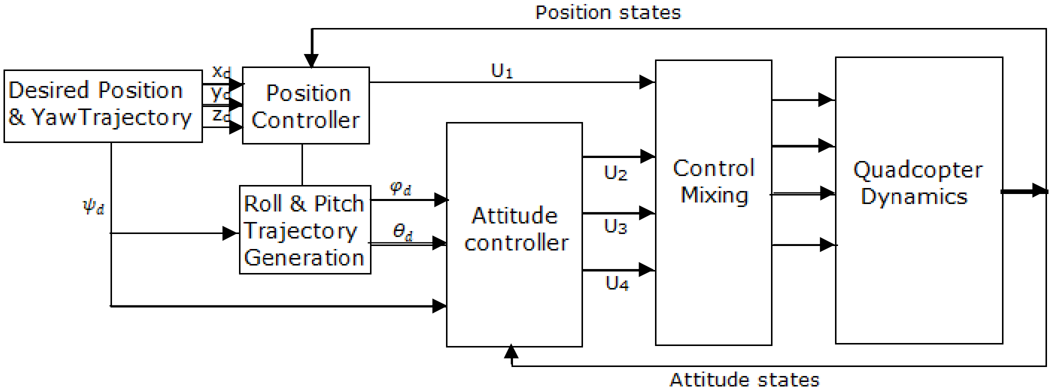

Figure 3.1 Block diagram of the quadcopter control system.

3.4. Adaptive control design using ANN

An analog neural network which updates the parameter of the sliding surface α_{x_i}, is implemented [20]. The structure of the neural network is shown in figure 3.2. The inputs to the neural network are the tracking error and its derivative, while the output is s_{x_i}.

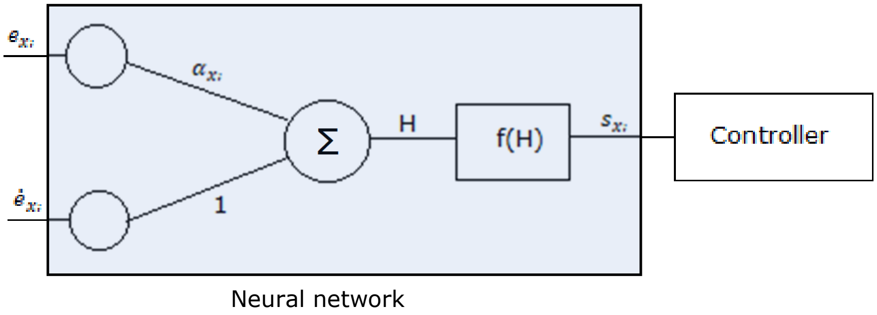

Figure 3.2 Structure of the neural network.

As can be seen from the above figure, s_{x_i} is an output of an activation function. An activation function in neural networks transforms the summed weighted inputs into output and adds non-linearity property. There are different types of activation functions. Some of the common ones are sigmoid and hyperbolic tangent functions. In this thesis, both functions will be experimented, and the more suitable function will be selected. Both functions are non-linear, differentiable and monotonic. The sigmoid and hyperbolic tangent functions are given in equations (3.21) and (3.22) respectively.

$$H = \alpha_{x_i} e_{x_i} + \dot{e}_{x_i} \tag{3.20}$$

$$f(H) = \tanh(H) = \frac{exp^{aH} - exp^{-aH}}{exp^{aH} + exp^{-aH}} \tag{3.21}$$

$$f(H) = \text{sigmoid}(H) = \frac{a(1 - exp^{-aH})}{(1 + exp^{-aH})} \tag{3.22}$$

Where a – parameter that determines the shape of the tanh function.

For updating the sliding surface parameter α_{x_i} in a way that minimizes the tracking error, a back propagation with gradient descent algorithm is implemented, which is given by equation (3.23) [20], [28].

$$\alpha_{x_{i,new}} = \alpha_{x_i} - \gamma \frac{\partial E}{\partial \alpha_{x_i}} \tag{3.23}$$

Where $E = \frac{1}{2} e_{xi}^2$,

$\alpha_{x_{i,new}}$ – updated value and

γ – learning rate.

The gradient descent is calculated using chain rule as follows.

$$\frac{\partial E}{\partial \alpha_{x_i}} = \frac{\partial E}{\partial x_i} \cdot \frac{\partial x_i}{\partial s_{x_i}} \cdot \frac{\partial s_{x_i}}{\partial H} \cdot \frac{\partial H}{\partial \alpha_{x_i}} = e_{x_i} \cdot \frac{\partial E}{\partial \alpha_{x_i}} \cdot \ f'(H) \cdot e_{x_i} \tag{3.24}$$

Assuming the hyperbolic tangent is the activation function, the derivative of $f(H)$ is $f'(H) = a - \mathrm{a} \cdot \tanh(H)$ and from [20], [28], $\frac{\partial E}{\partial \alpha_{x_i}} = 1$. Replacing these values in equation (3.23), we get

$$\alpha_{x_{i,new}} = \alpha_{x_i} - \gamma e_{xi}^2 (a - \mathrm{a} \cdot \tanh(H)) \tag{3.25}$$

So, all the necessary controllers are designed, and the next step is to implement them in Simulink/Matlab. The simulation results of the implemented system will be given in the following chapter.

CHAPTER 4

4. SIMULATION RESULTS

The control system with the quadcopter dynamics is implemented in Simulink/Matlab. The Simulink quadcopter model of [29] is used as a starting point. A new control system is implemented, and the dynamics block is modified so that to fit the dynamic equations of this thesis. Two groups of trajectory commands are used to test the performance of the system. The first group consists of constant value inputs that are used to analyze the settling time, maximum overshoot and reaching time. The second group contains varying linear commands which helps to evaluate the tracking ability of the system.

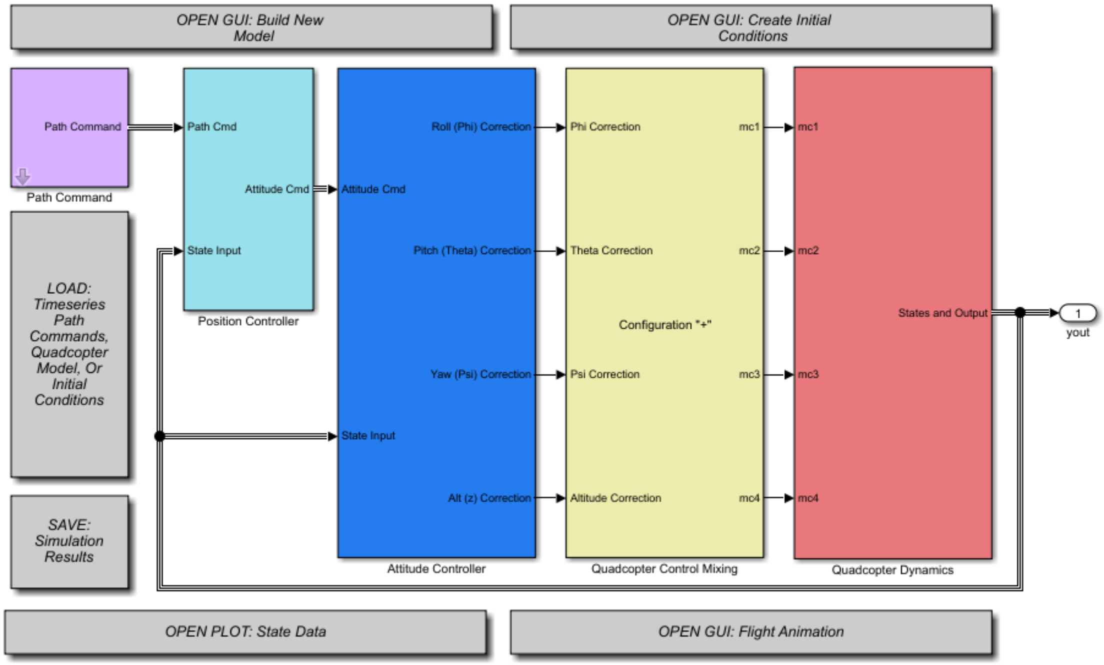

Figure 4.1 Simulink block diagram of the whole system.

The parameter values of the quadcopter model simulated are given in Table 4.1. And in Table 4.2, the controllers’ parameter values are given.

Table 4.1 Quadcopter model parameter values.

Nr	**Parameter**	**Value**	**Unit**
1	Mass (m)	1.023	kg
2	I_{xx}	0.0095	$N \cdot s^2 \cdot rad^{-1}$
3	I_{yy}	0.0095	$N \cdot s^2 \cdot rad^{-1}$
4	I_{zz}	0.0186	$N \cdot s^2 \cdot rad^{-1}$
5	J_m	3.7882e-06	$N \cdot s^2 \cdot rad^{-1}$
6	C_T	1.4865e-07	rad^{-2}
7	C_H	2.9250e-09	$kg \cdot m \cdot rad^{-2}$
8	g	9.81	$m \cdot s^{-2}$
9	l	0.2223	m

Table 4.2 Controller parameters used for simulation.

Nr	**Parameter**	**Value**
1	β_x	2.2
2	β_y	2.2
3	β_z	14.8
4	α_x	0.2
5	α_y	0.22
6	α_z	1.3
7	β_φ	97
8	β_θ	97
9	β_ψ	50
10	α_φ	1.9
11	α_θ	1.8
12	α_ψ	3

4.1. Trajectory command 1

Here the first type of trajectory commands is going to be used. First, the comparison between the two activation functions used in the controllers will be shown. As a sample the X and Y-position controllers are selected.

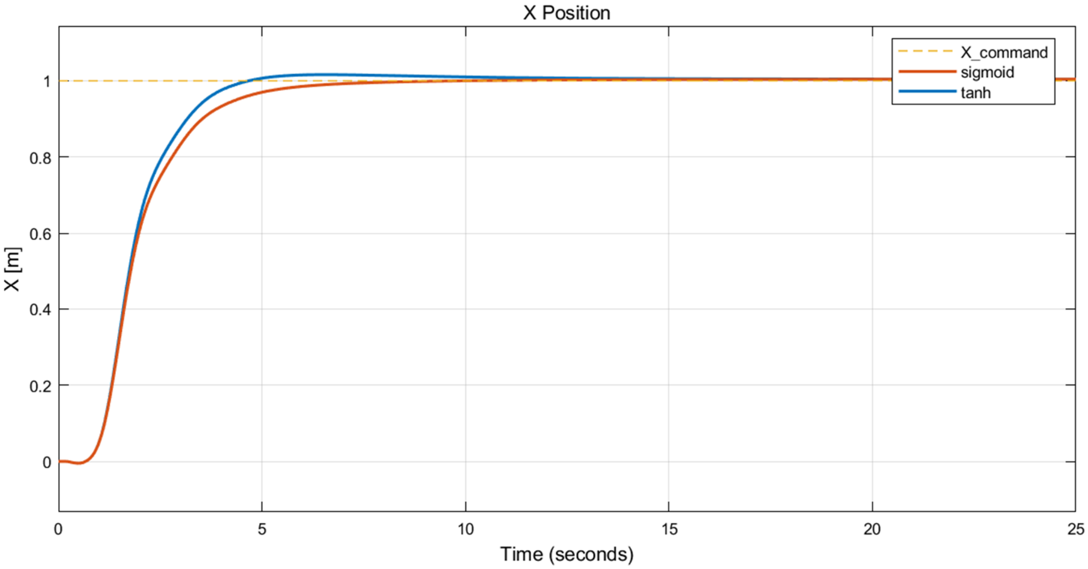

Figure 4.2 X-position response for tanh and sigmoid functions

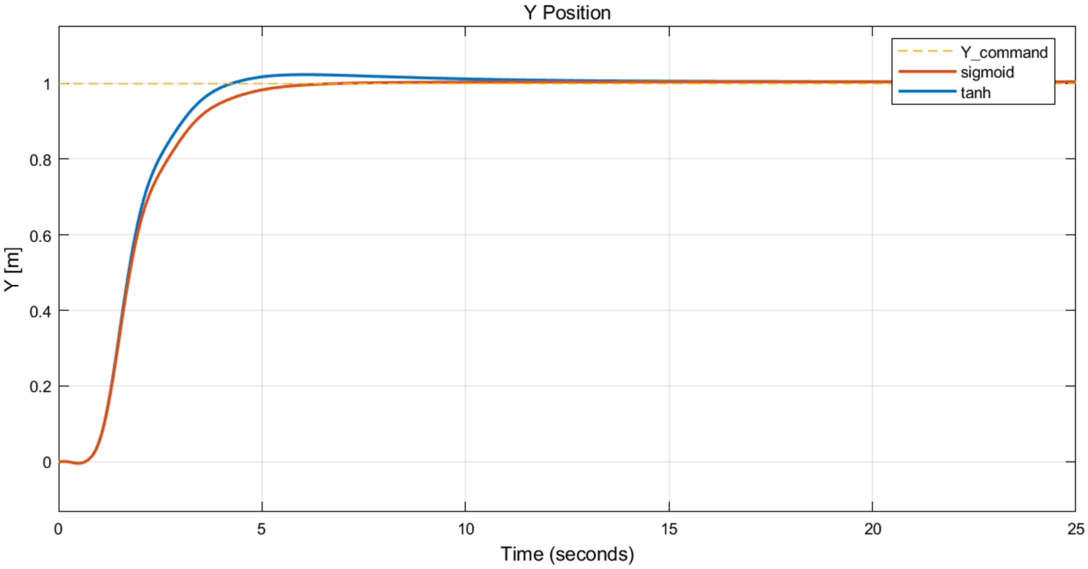

Figure 4.3 Y-position response for tanh and sigmoid functions

As can be seen from figures 4.2 and 4.3, the response of the controller, where tanh is used as an activation function, has faster reach-time but more overshoot. Besides, similar experiment was carried for Z-position controller, but the performance of both activation functions was the similar. In the following simulation results, tanh was used as an activation function.

Next, the position state responses of two controllers, the SMC and SMC with ANN, is shown. Figures (4.4), (4.5) and (4.6) show the responses of both controllers to input values of (1, 1, 3) given as (x, y, z) target point.

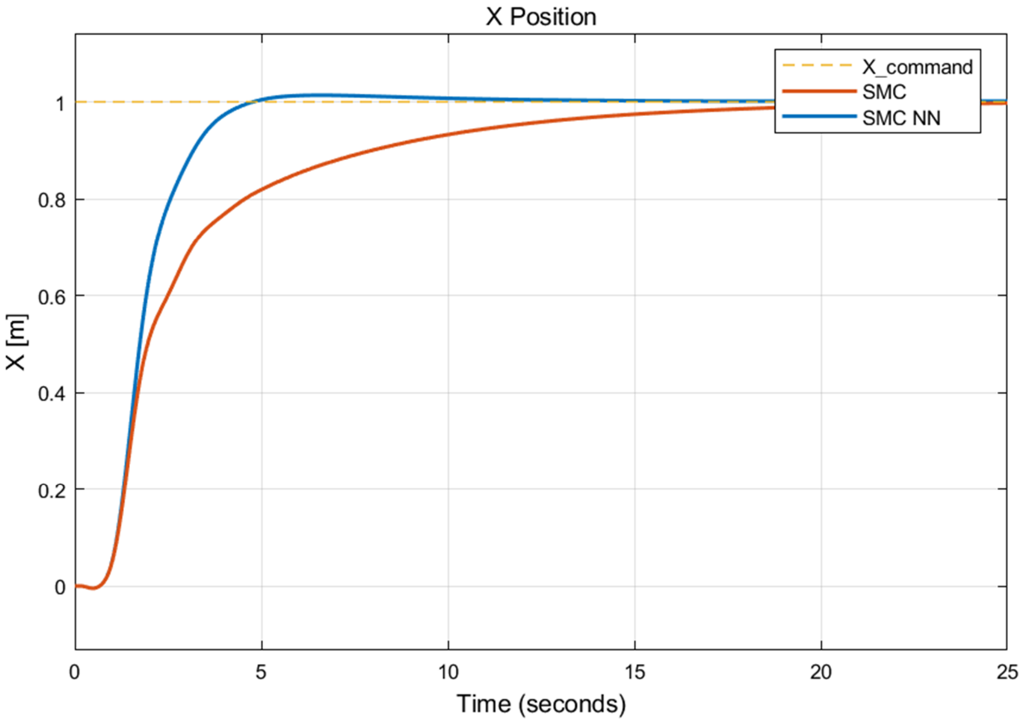

Figure 4.4 X-state response of the SMC and SMC with NN controllers.

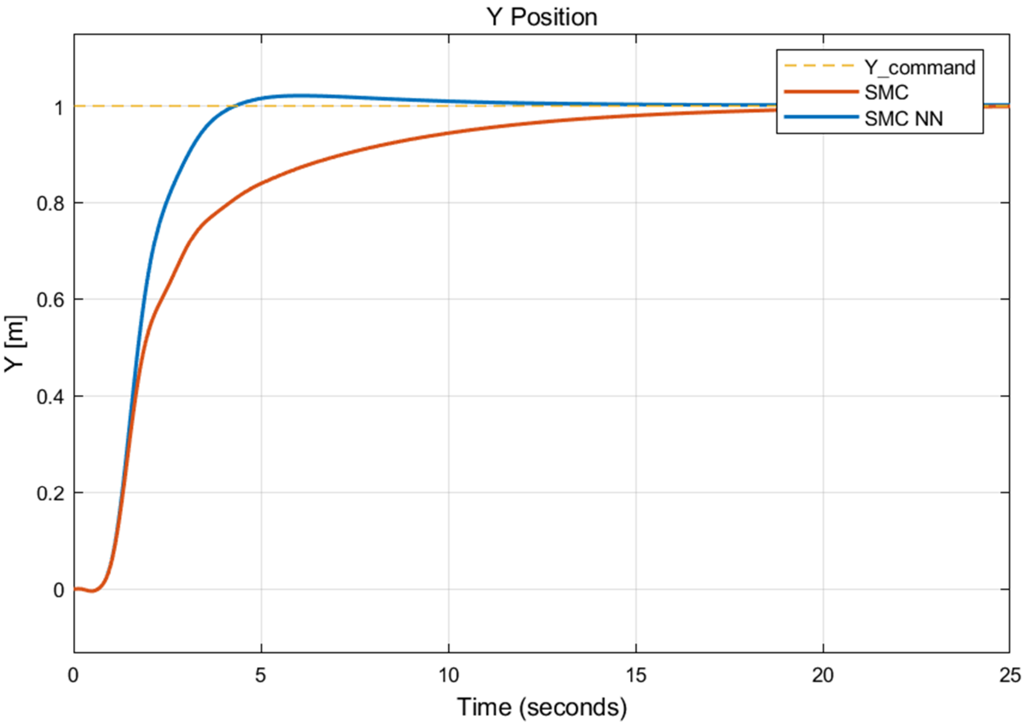

Figure 4.5 Y-state response of the SMC and SMC with NN controllers.

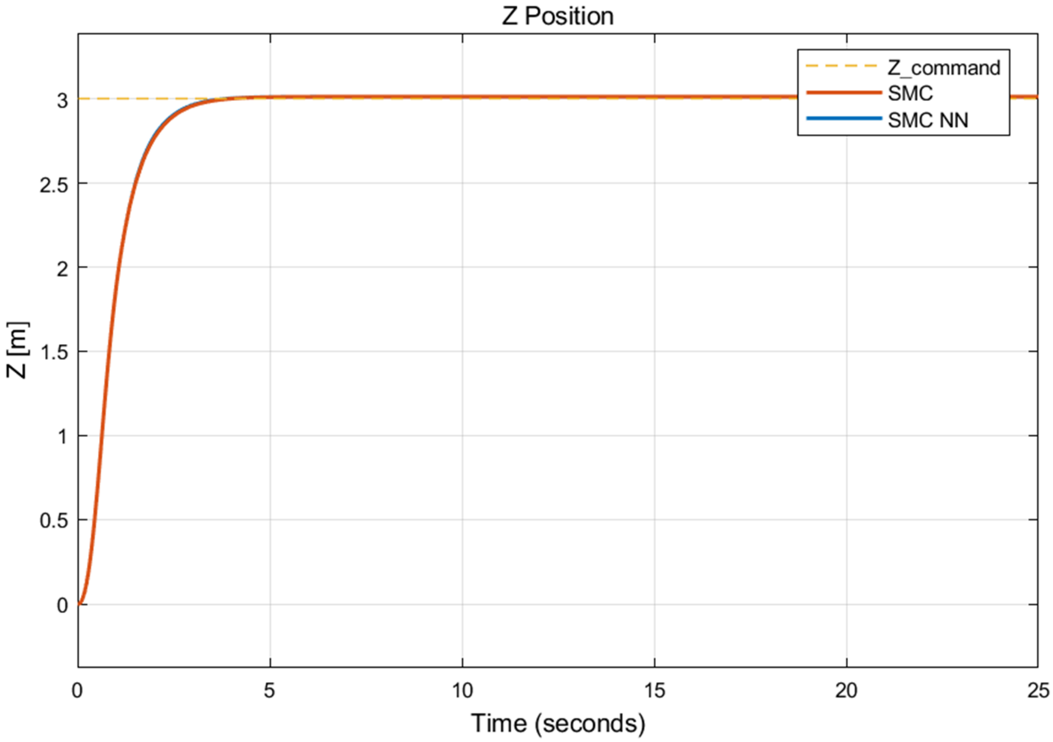

Figure 4.6 Z-state response of the SMC and SMC with NN controllers.

The summary of the performance of the controllers for the given input is shown in Table (4.3) below. Error accuracy of 0.02 is taken.

Table 4.3 Steady-state transient response performance indicators.

Nr		**State**	**SMC NN**	**SMC**
1	Reaching time in seconds	X	4.750	16.070
		Y	4.328	23.367
		Z	3.240	3.310
2	Overshoot by percentage (%)	X	1.4	0
		Y	2.1	0
		Z	0.47	0.47
3	Settling time in seconds	X	4.750	16.070
		Y	4.328	23.367
		Z	3.240	3.310
4	Steady-state error in meters	X	0.002	0.002
		Y	0.002	0.008
		Z	0.014	0.014

Below, the plots of the Euler angles, linear velocity and angular rate in the body frame and the control law for the first input group are shown.

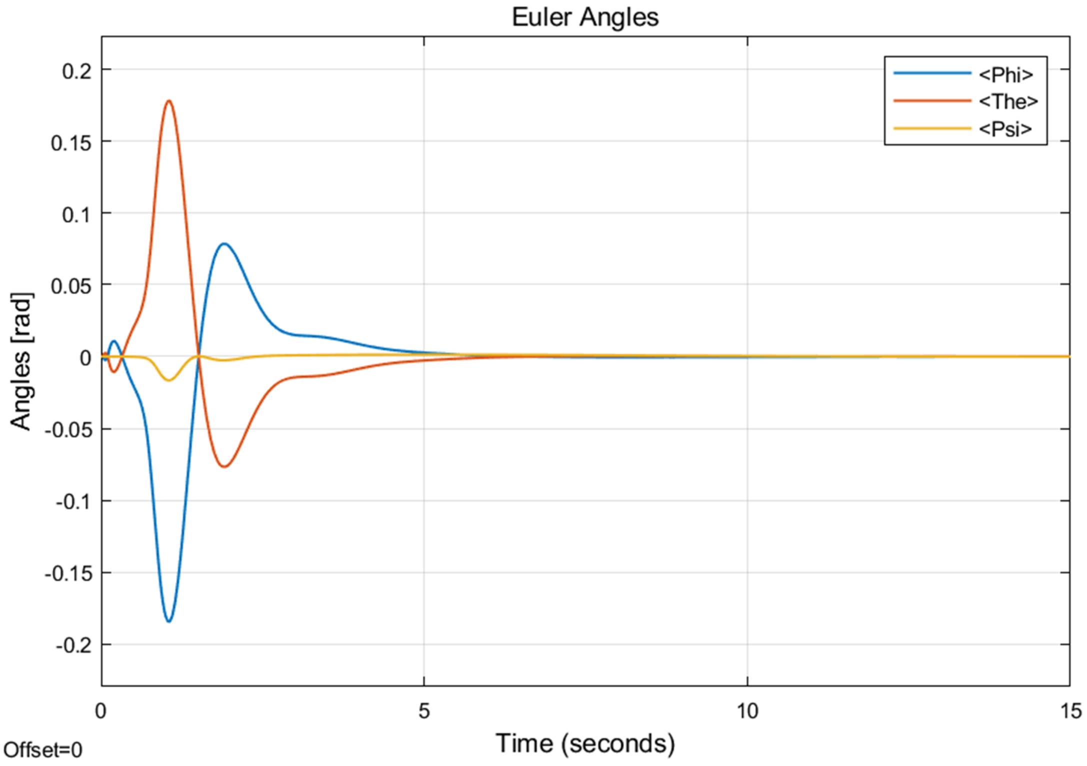

Figure 4.7 Plot of Euler angles.

Observing the plots of Euler angles, body linear velocities and body angular rates in figures 4.7, 4.8 and 4.9 respectively, it can be concluded that all the states are stable and have acceptable range of values.

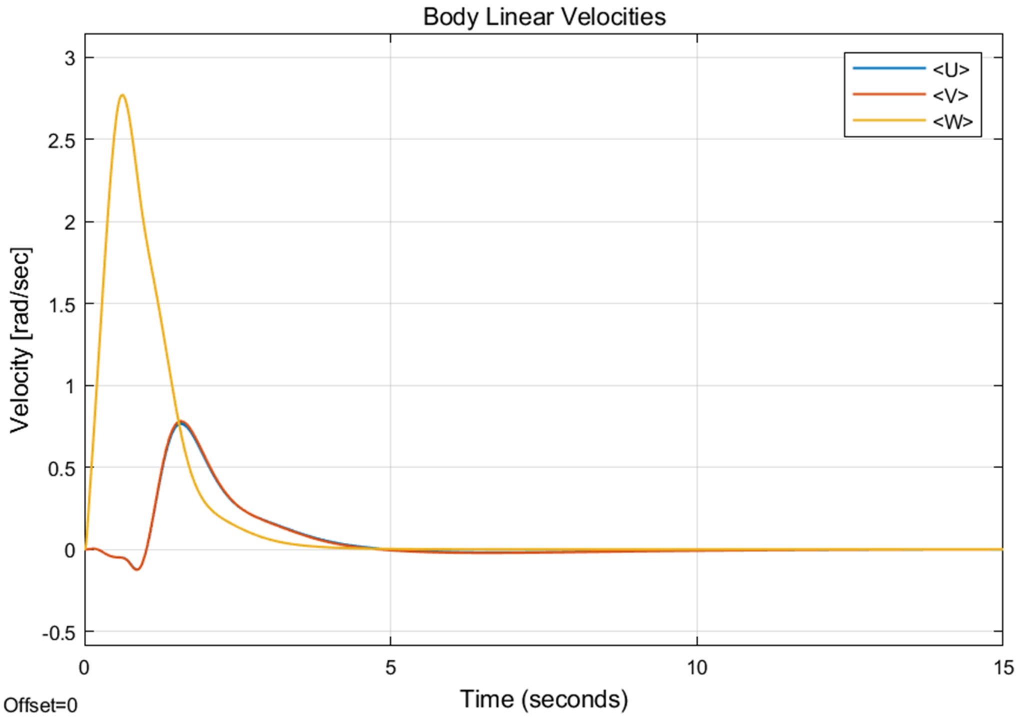

Figure 4.8 Linear velocities in the body frame.

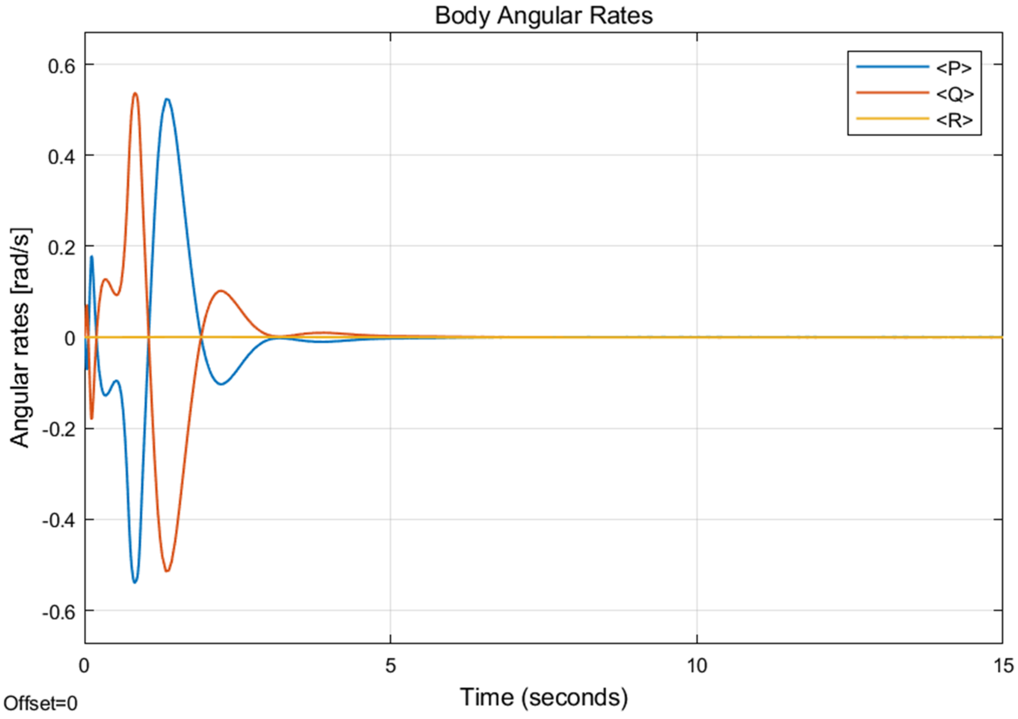

Figure 4.9 Angular rates in the body frame.

Figure 4.10 displays the output of the four controllers (control law). As can be seen from the plots, the chattering effect can be seen in small levels in phi and theta-controller.

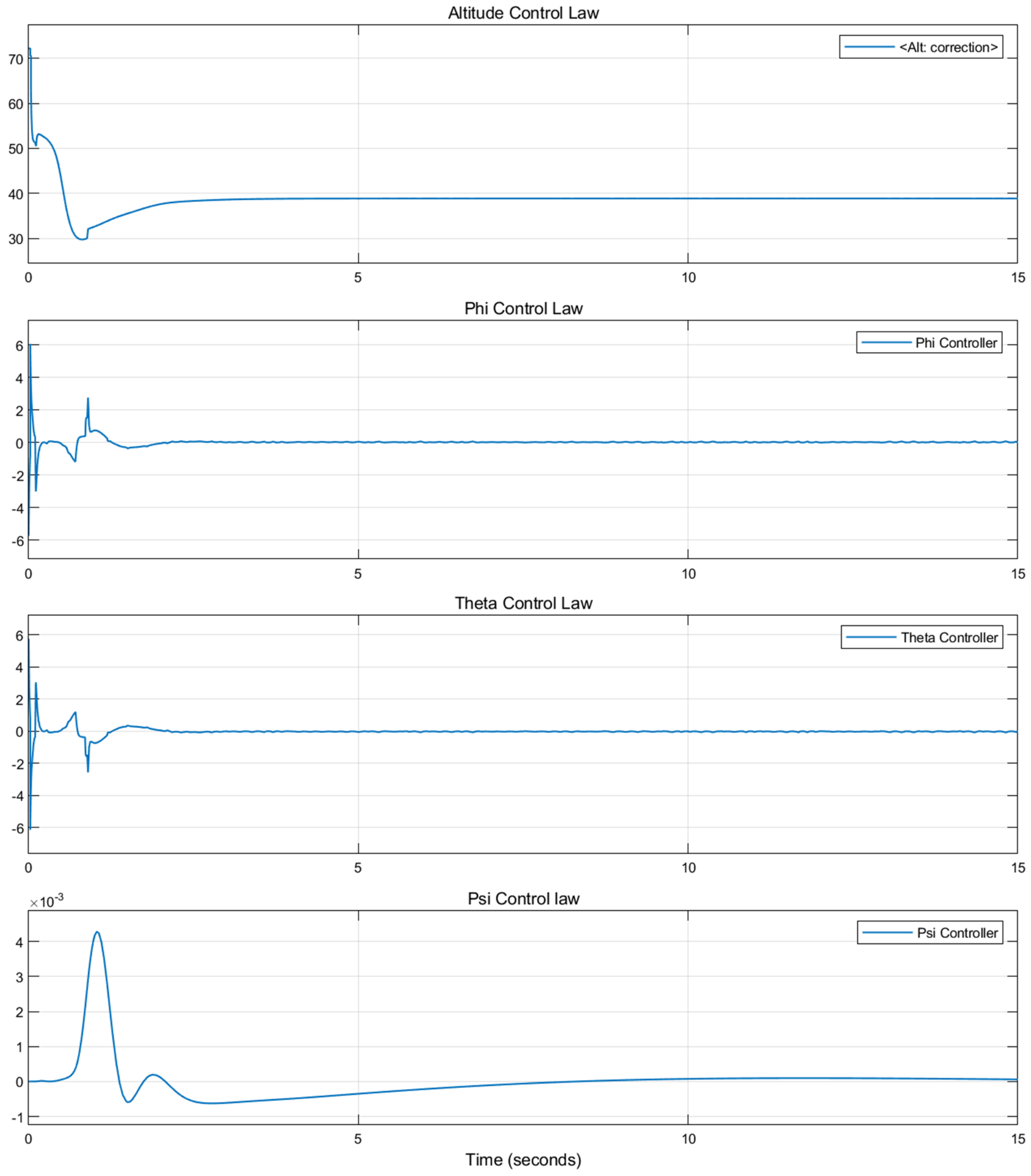

Figure 4.10 Output of the four controllers.

Figure 4.11 shows the online update of the alpha parameter of the sliding surface by the neural network in the position controllers. As we can see from the figure, the neural network updates the coefficient of the sliding surface, in which it decreases as the state error decreases.

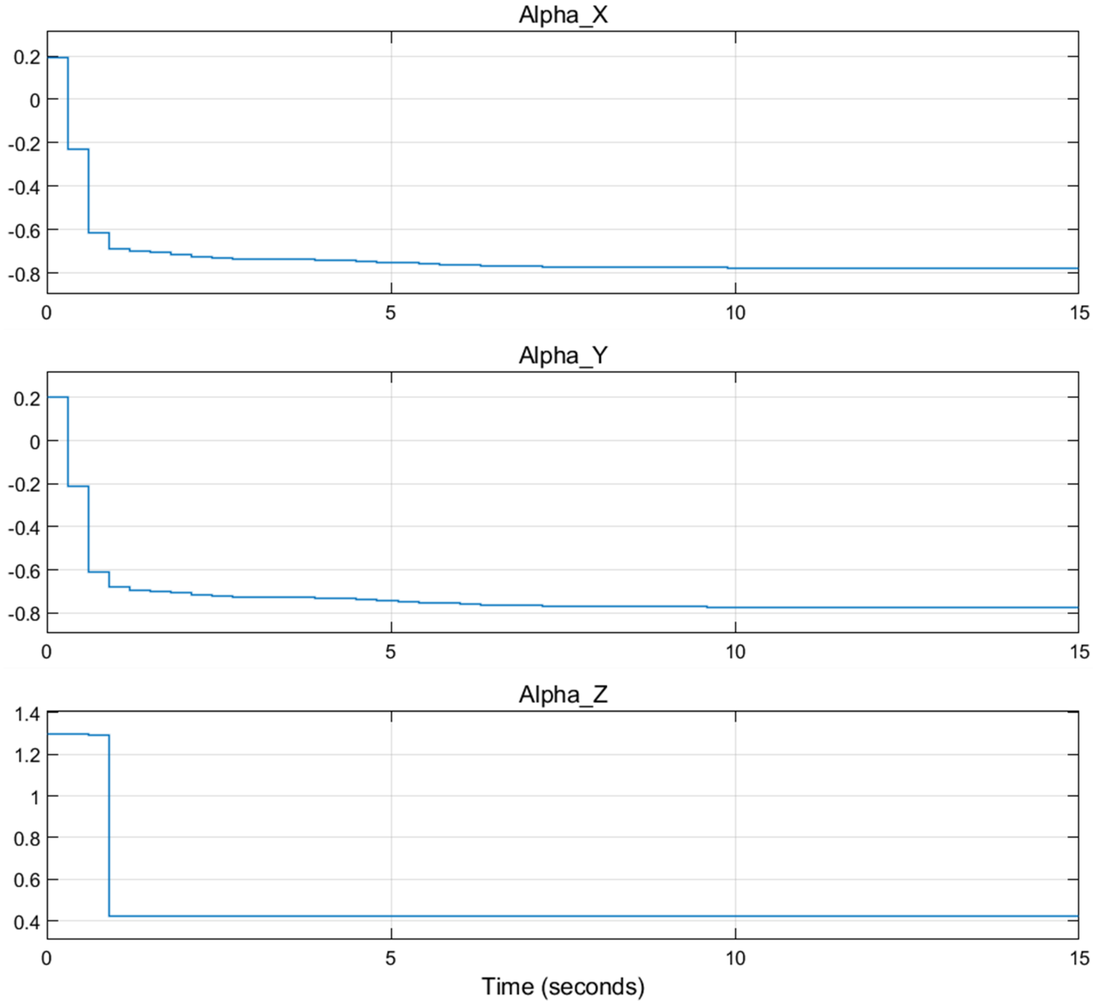

Figure 4.11 Values of the sliding surface coefficient.

4.2. Trajectory command 2

To check the tracking capability of the controllers, a continuously varying trajectory is given as an input. As a comparison, a PID controller from [29] is included. First the plots of X, Y and Z are shown separately and later the 3-D view of the path followed is shown.

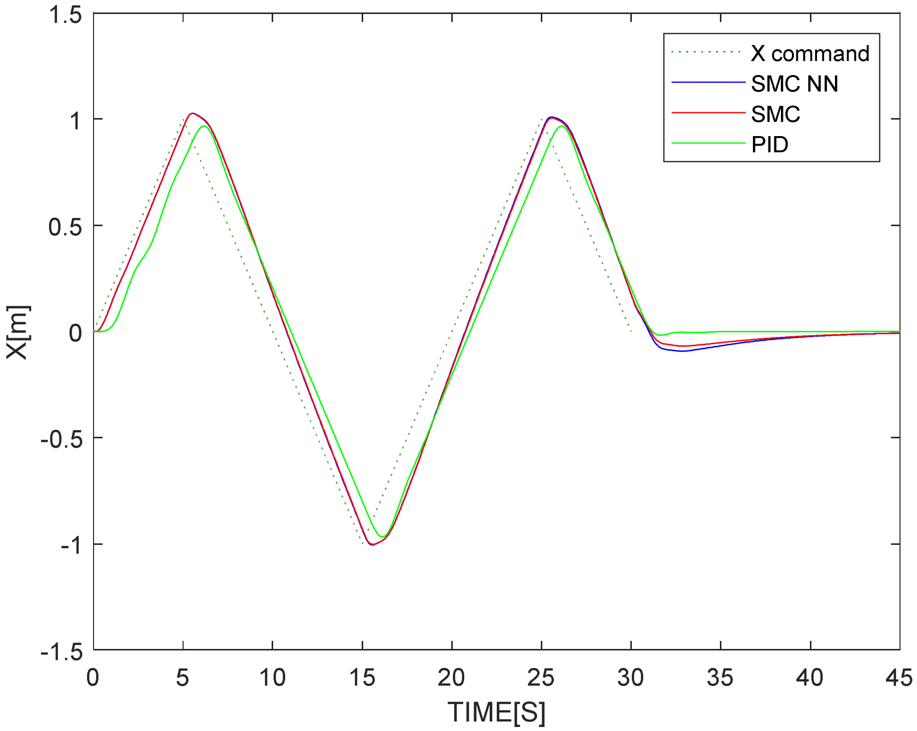

Figure 4.12 Trajectory tracking of the X-state.

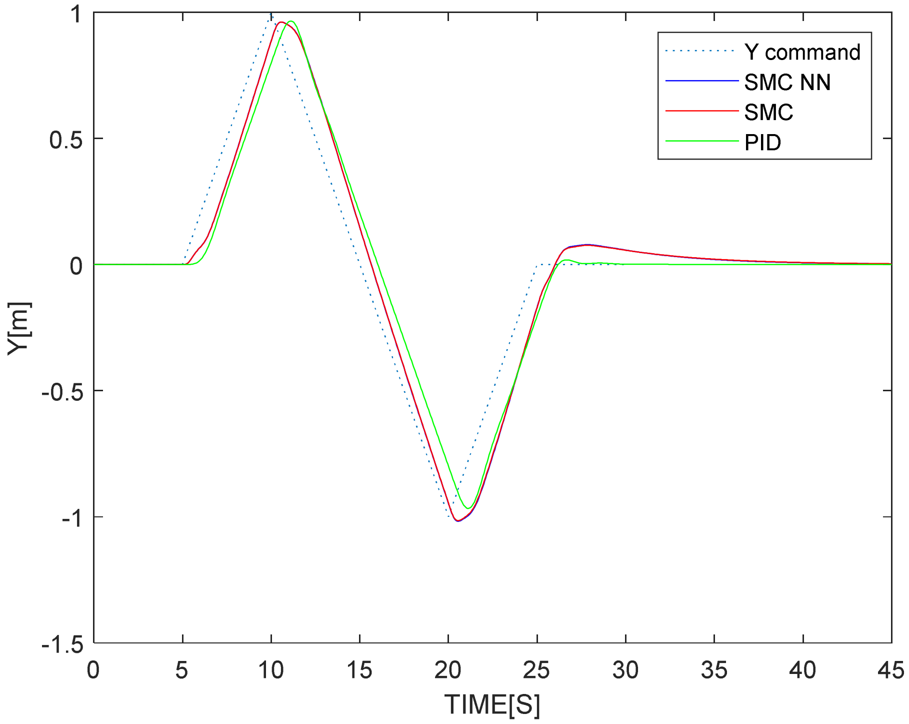

Figure 4.13 Trajectory tracking of the Y-state.

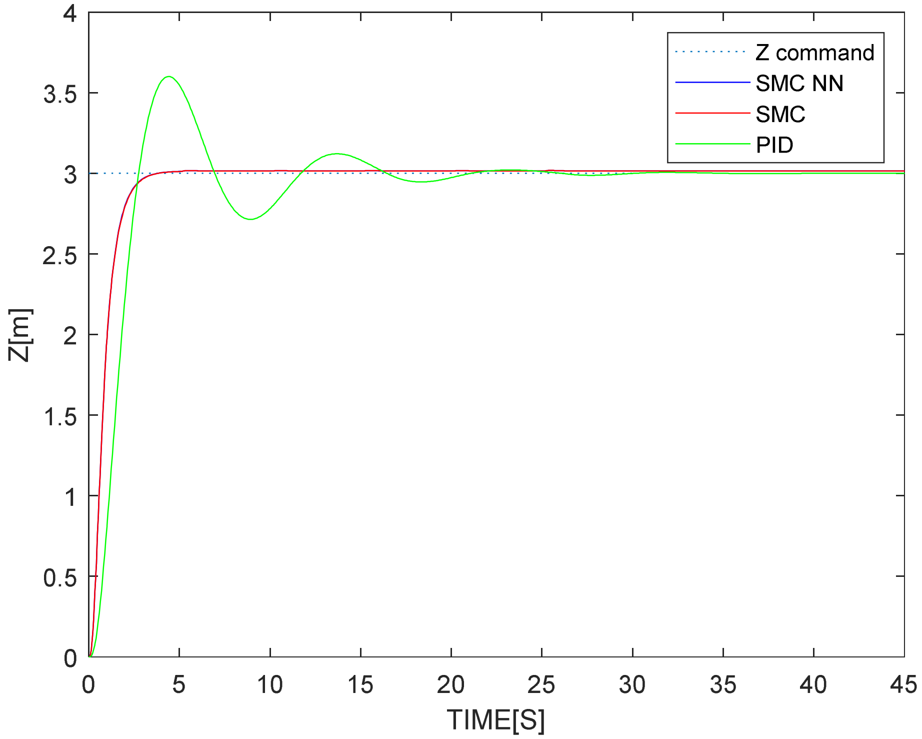

Figure 4.14 Trajectory tracking of the Z-state.

As can be seen from the above figures, both sliding mode controllers showed better tracking performance over the PID controller. It can be noted that when the trajectory path changes abruptly, the sliding mode controllers are seen to have larger tracking error than the PID. But with time, the tracking error is minimized contrary to the constant tracking error in the PID.

For better visualization, the path followed by the quadcopter for the second group of inputs is shown in 3-dimensional plot in the following figures.

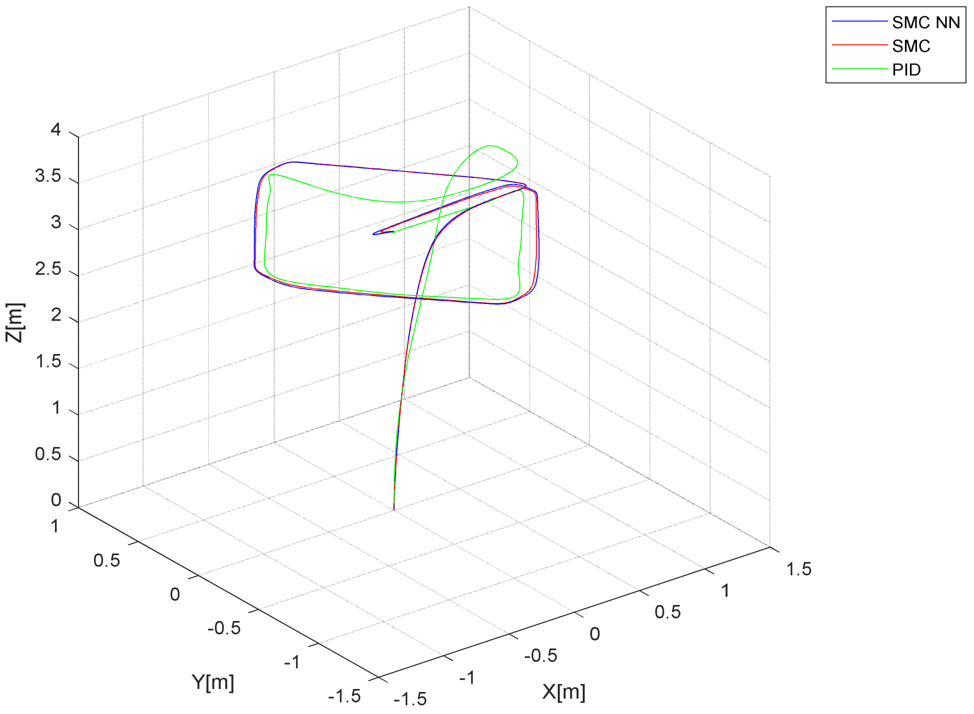

Figure 4.15 Three-dimensional view of the trajectory followed.

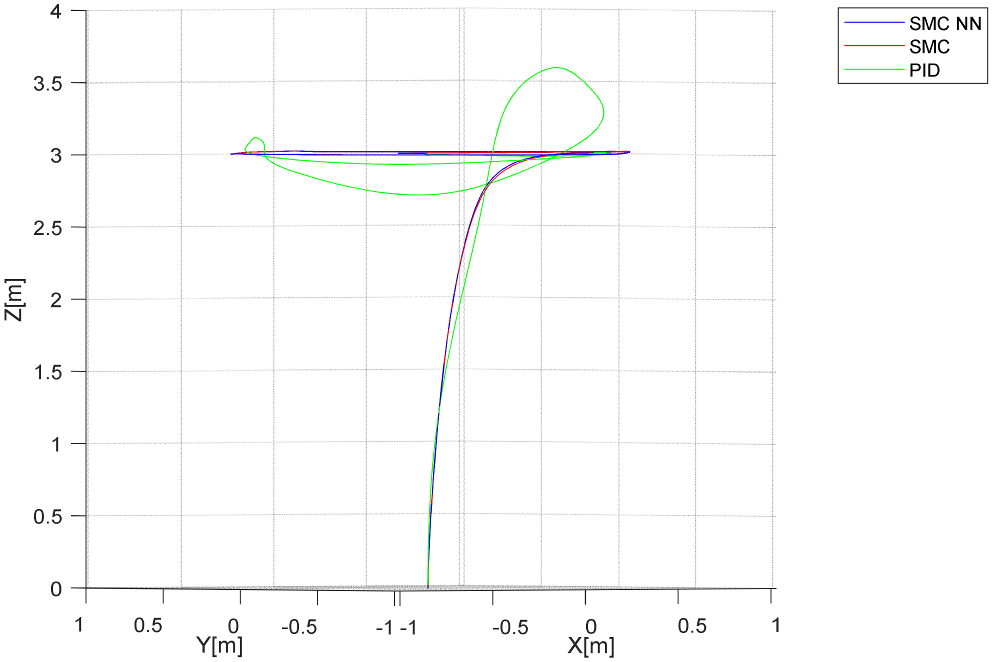

Figure 4.16 Lateral view of the trajectory followed.

CHAPTER 5

5. CONCLUSIONS AND FUTURE RECOMMENDATIONS

In this chapter the overall conclusion of the thesis will be discussed. And finally, recommendations for future improvement and practical implementation are given.

5.1. Summary and conclusion

The purpose of this thesis is to design and evaluate a non-linear position and attitude flight controller for a quadcopter using sliding mode technique aided by artificial neural networks. First the mathematical model is derived based on two frames of references. Then 6 controllers, 3 for the position and 3 for the attitude are derived. But since the quadcopter dynamics is underactuated, the overdetermined controlled variable is solved using least-squares method. And for autonomous control purposes the trajectory commands of the rolling and pitching motions are determined from the X and Y controllers. Then a simple neural network structure is implemented to adaptively tune a coefficient of the sliding surface. Finally, the designed control system and dynamics are implemented and evaluated in Simulink/Matlab. Below are some important points of the designed control system.

5.1.1. Transient and steady-state response

As seen in chapter 4, the sliding mode controller with a neural network shows good transient and steady-state response. It shows good settling time, small overshoot and small steady-state error. And in terms of the chattering effect, there was no major chattering problem.

5.1.2. Robustness

In the design of the controller the term $d(x)$ in the dynamical equations is ignored, to account for modeling inaccuracy, uncertainties and external disturbances. And the control system designed shows good performance, even though this dynamic term is ignored. Additionally, the controller is robust to an increase in mass up to 20%. But, in terms of external disturbance, the system showed a larger steady-state error.

5.1.3. Tracking Performance

In terms of tracking capability, the sliding mode control system showed very good performance. Its tendency to minimize tracking error was better than the PID controller shown in chapter 4. The only problem seen is that when the trajectory changes abruptly, the system takes time to minimize the tracking error. And this arises from the trajectory speed and acceleration limitations of the controller.

5.1.4. Adaptivity

The addition of the artificial neural network structure is to adaptively tune the coefficient of the sliding surface. And with the help of this, in the first group of inputs, the sliding mode controller with neural network shows better performance in terms of settling time. But in the second group of trajectories the performance is similar.

5.1.5. Attitude command generation

In autonomous quadcopters, the commands from the operator are only the position and the yaw commands. And the roll and pitch commands should be determined from the position controllers. In this thesis the roll and pitch commands are determined from the position controllers which makes it suitable for autonomous applications.

5.2. Limitations

The following limitations were observed in the design and performance of the overall system.

- Due to the singularity in the transformation matrix, $\theta = \pm 90^0$, it was made that trajectory command for the pitch and roll angles are in the range of -12 to 12 degrees. This makes the transient response of X and Y slower (the motions in X and Y directions depend on pitching and rolling actions respectively).
- Due to the first and second derivative terms of the trajectory command in the controller equations, a path with a large slope generates large values of these terms. This situation may drive the system to go out of stability. For example, for the given controller parameters, trajectory speeds of more than 5 m/s in the X and Y commands, derives the system out of stability. But of course, it can be

improved by changing the controller parameter and mainly, by changing the value of the coefficient of the filtered derivative block used.

5.3. Future recommendations

Below are listed some recommendations for the future improvement and implementation of the system designed.

- To eliminate steady-state errors and better performance under disturbances a term of integration of the error could be introduced in the sliding surface equation.
- The neural network can be extended so that it can also calculate and update the coefficient of the reaching law, β_{xi}. This may result in a better transient performance.
- To fully evaluate the performance of the designed controllers, it should be practically implemented. In the future, the author plans to implement the designed flight control system in a Pixhawk autopilot system.

REFERENCES

[1] B. J. Emran and H. Najjaran, "A review of quadrotor: An underactuated mechanical system," *Annu. Rev. Control*, vol. 46, pp. 165–180, 2018.

[2] T. S. Venkatasundarakumar, R. Suwathy, T. M. Haripriya, and M. Venkatesan, "Motion control analysis of a quadcopter system Part II – Modelling," pp. 1–4, 2016.

[3] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors," no. December, pp. 547–556, 2014.

[4] "History of Quadcopters and Other Multirotors," 2020. [Online]. Available: https://www.krossblade.com/history-of-quadcopters-and-multirotors.

[5] "A Brief History of Drones," 2020. [Online]. Available: https://buybestquadcopter.com/history-future-drones/.

[6] "Quadcopter," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Quadcopter.

[7] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors," *Open J. Appl. Sci.*, 2014.

[8] N. Xuan-mung, "applied sciences Improved Altitude Control Algorithm for Quadcopter Unmanned Aerial Vehicles," 2019.

[9] S. I. Tomashevich, O. I. Borisov, V. S. Gromov, A. A. Pyrkin, and A. A. Bobtsov, "Experimental study on robust output control for quadcopters," *2017 25th Mediterr. Conf. Control Autom. MED 2017*, pp. 1029–1034, 2017.

[10] A. A. Pyrkin, A. A. Bobtsov, S. A. Kolyubin, O. I. Borisov, and V. S. Gromov, "Output controller for quadcopters based on mathematical model decomposition," in *2014 22nd Mediterranean Conference on Control and Automation, MED 2014*, 2014.

[11] A. A. Pyrkin, A. A. Bobtsov, S. A. Kolyubin, O. I. Borisov, V. S. Gromov, and S. V. Aranovskiy, "Output controller for quadcopters with wind disturbance cancellation," in *2014 IEEE Conference on Control Applications, CCA 2014*, 2014.

[12] O. I. Borisov, V. S. Gromov, A. A. Pyrkin, A. A. Bobtsov, and N. A. Nikolaev, "Output Robust Control with Anti-Windup Compensation for Quadcopters**This article is supported by Russian Science Foundation, project 16-11-00049.," *IFAC-PapersOnLine*, 2016.

[13] F. Santoso, M. A. Garratt, S. G. Anavatti, and I. Petersen, "Robust Hybrid Nonlinear Control Systems for the Dynamics of a Quadcopter Drone," *IEEE Trans. Syst. Man, Cybern. Syst.*, pp. 1–13, 2018.

[14] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Hybrid PD-Fuzzy and PD Controllers for Trajectory Tracking of a Quadrotor Unmanned Aerial Vehicle: Autopilot Designs and Real-Time Flight Tests," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. PP, pp. 1–13, 2019.

[15] M. M. Ferdaus, S. G. Anavatti, M. Pratama, and M. A. Garratt, "A Novel Self-Organizing Neuro-Fuzzy based Intelligent Control System for a AR.Drone Quadcopter," *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 2026–2032, 2019.

[16] J. J. Xiong and G. B. Zhang, "Global fast dynamic terminal sliding mode control

for a quadrotor UAV," *ISA Trans.*, 2017.

[17] B. Sumantri, N. Uchiyama, and S. Sano, "Least square based sliding mode control for a quad-rotor helicopter and energy saving by chattering reduction," *Mech. Syst. Signal Process.*, 2016.

[18] H. Razmi and S. Afshinfar, "Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV," *Aerosp. Sci. Technol.*, vol. 91, pp. 12–27, 2019.

[19] E. H. Zheng, J. J. Xiong, and J. L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Trans.*, 2014.

[20] H. Razmi, "Adaptive neural network based sliding mode altitude control for a quadrotor UAV," pp. 0–3, 2018.

[21] Y. Yang and Y. Yan, "Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control," *Aerosp. Sci. Technol.*, 2016.

[22] N. Ben Ammar, S. Bouallègue, and J. Haggège, "Fuzzy Gains-Scheduling of an Integral Sliding Mode Controller for a quadrotor Unmanned Aerial Vehicle," *Int. J. Adv. Comput. Sci. Appl.*, 2018.

[23] N. Cibiraj and M. Varatharajan, "Chattering reduction in sliding mode control of quadcopters using neural networks," in *Energy Procedia*, 2017.

[24] FRANCESCO SABATINO, "Quadrotor control: modeling, nonlinear control design, and simulation," 2015.

[25] A. Hussein, "Autopilot Design for a Quadcopter," no. October 2017, 2017.

[26] S. Abdelhay and A. Zakriti, "Modeling of a Quadcopter Trajectory Tracking System Using PID Controller," *Procedia Manuf.*, vol. 32, pp. 564–571, 2019.

[27] Z. Benic, P. Piljek, and D. Kotarski, "Mathematical Modelling of Unmanned Aerial Vehicles with Four Rotors," *Interdiscip. Descr. Complex Syst.*, 2016.

[28] J. Ye, "Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot," vol. 71, pp. 1561–1565, 2008.

[29] D. Hartman, K. Landis, M. Mehrer, S. Moreno, J. Kim, "Quadcopter Dynamic Modeling and Simulation (Quad-Sim) v 1.00," 2014. [Online]. Available: https://github.com/dch33/Quad-Sim.