NEURO-FUZZY SYSTEM AND GENETIC ALGORITHM FOR CUBESAT LQR BASED ATTITUDE CONTROL SYSTEM

A MASTER'S THESIS

BY

HAYLEYESUS ANDUALEM ALEMAYEHU

DEPARTMENT OF ELECTROMECHANICAL ENGINEERING

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

May 2020

NEURO-FUZZY SYSTEM AND GENETIC ALGORITHM FOR CUBESAT LQR BASED ATTITUDE CONTROL SYSTEM

By

HAYLEYESUS ANDUALEM ALEMAYEHU

A Thesis Submitted as a Partial Fulfillment for the Degree of Master of Science in Electro-Mechanical Engineering (Mechatronics Engineering)

to

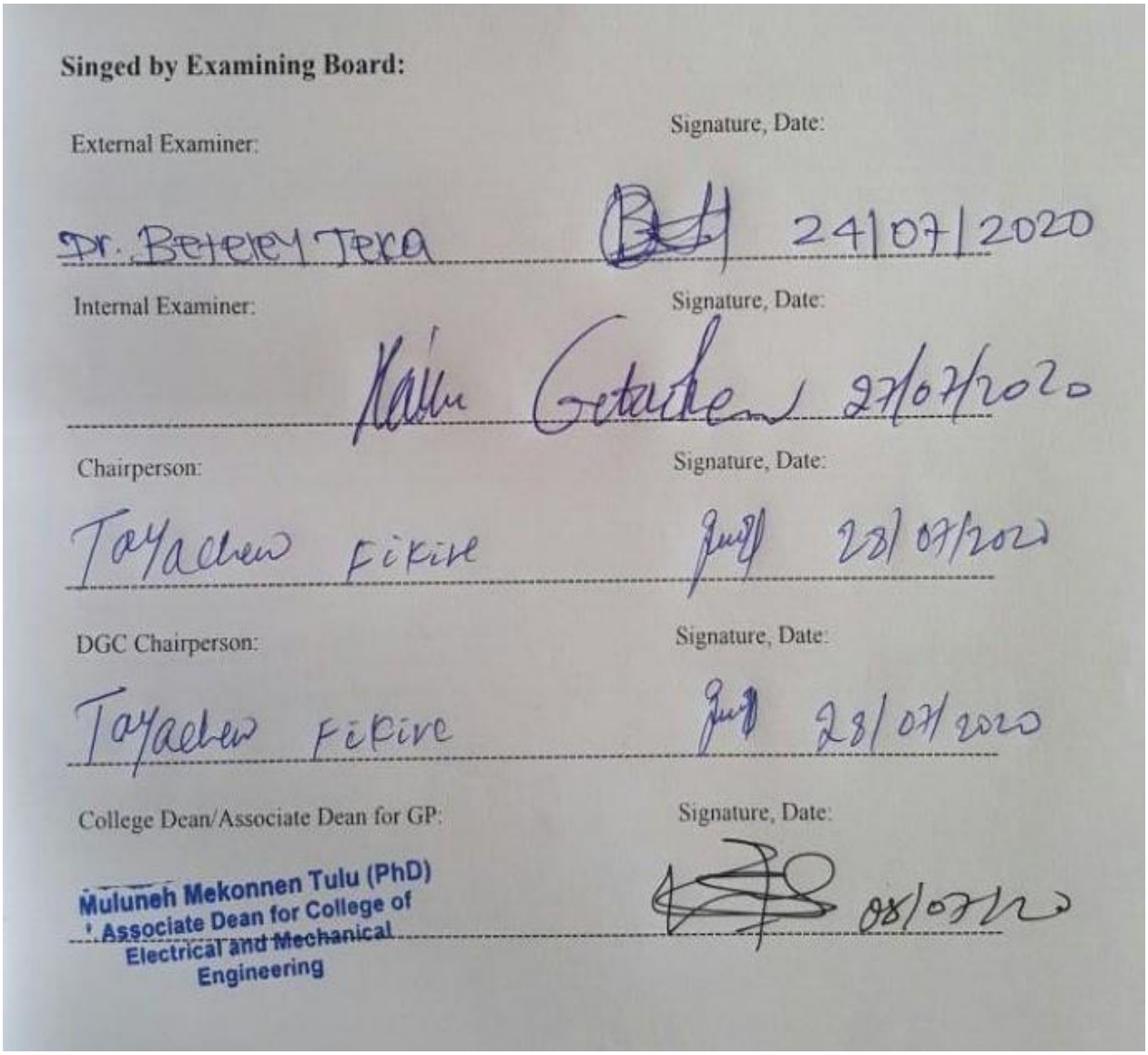
DEPARTMENT OF ELECTROMECHANICAL ENGINEERING

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

May 2020

Certificate

This is to certify that the thesis prepared by **Mr. Hayleyesus Andualem Alemayehu** entitled "**Neuro-Fuzzy System and Genetic Algorithm for CubeSat LQR Based Attitude Control System"** and submitted as a partial fulfillment for the Degree of Master of Science complies with the regulations of the University and meets the accepted standards with respect to originality, content and quality.



Singed by Examining Board:

External Examiner: Signature, Date:

Dr. Beteley Tesfa 24/07/2020

Internal Examiner: Signature, Date:

Kalkidan Getachew 27/07/2020

Chairperson: Signature, Date:

Tayachew Fikire 28/07/2020

DGC Chairperson: Signature, Date:

Tayachew Fikire 28/07/2020

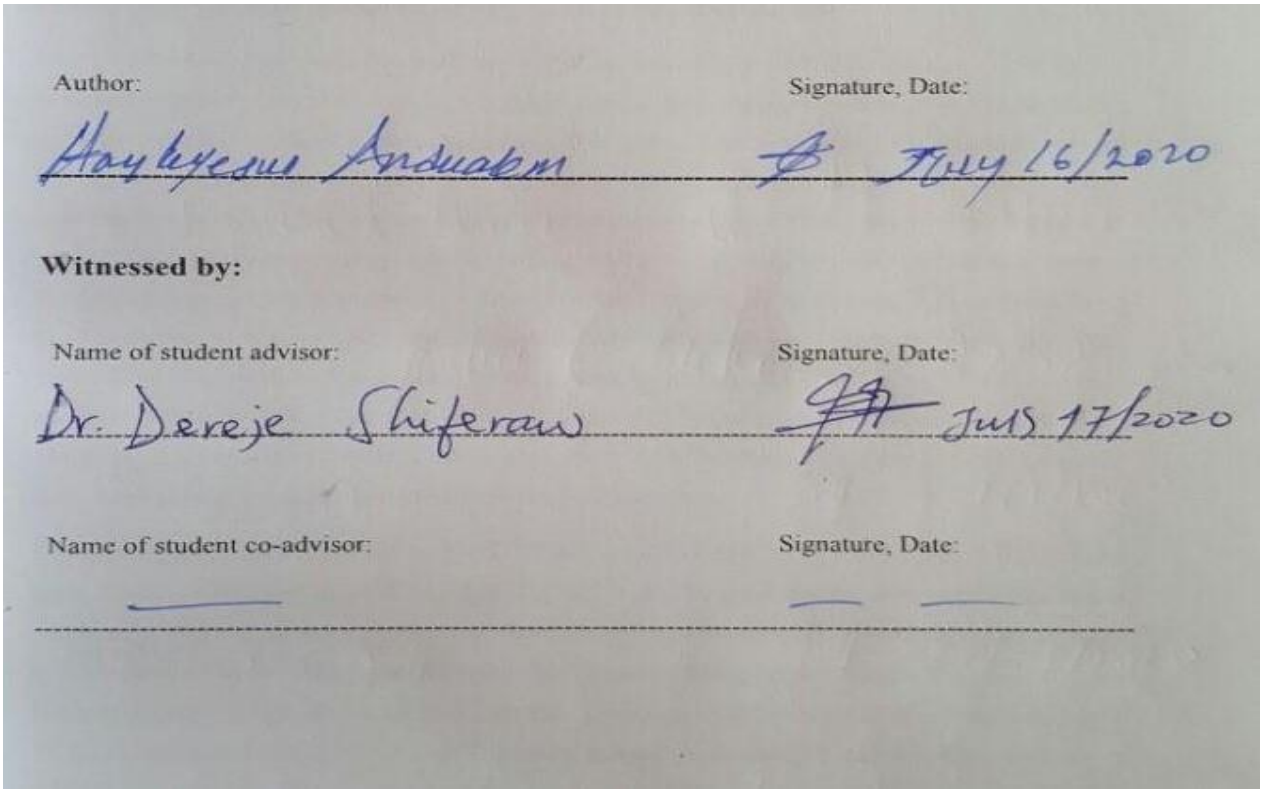
College Dean/Associate Dean for GP: Signature, Date:

Muluneh Mekonnen Tulu (PhD)
Associate Dean for College of
Electrical and Mechanical
Engineering

08/07/20

Declaration

I hereby declare that this thesis entitled "**Neuro-Fuzzy System and Genetic Algorithm for CubeSat LQR Based Attitude Control System"** was prepared by me, with the guidance of my advisor. The work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted, in whole or in part, for any other degree or professional qualification.



`

Abstract

CubeSat is next generation, promising, and small-sized satellites that can be easily assembled using commercially available components with a low investment cost. Attitude control is one of the core subsystems in the CubeSat system that deals with how to orient the CubeSat in any desired direction. Proper attitude controller of satellite design is necessary since devices that need pointing direction like antennae, camera, and some measurement devices are always mounted on the satellite. Uncontrolled CubeSat may even cause the entire mission loss.

Fuzzy controllers and linear quadratic controller are among the commonly employed controllers in CubeSat attitude control. Although manually tuned linear quadratic controller design shows good performance without actuator saturation, it is not optimal in handling the tradeoff between the desired performance and actuator saturation. A genetic algorithm is employed to handle this optimization problem. Besides, developing fuzzy controllers is challenging for multiple inputs and multiple outputs systems using expert knowledge and intuitive rational guess. An adaptive neuro-fuzzy inference system is proposed to develop a fuzzy system using training data sampled from the simulation of a genetically tuned linear quadratic regulator. These fuzzy systems mimic the linear quadratic regulator that is tuned by the genetic algorithm. MATLAB is used for the genetic algorithm optimization of linear quadratic regulator and learning based fuzzy controller design. The attitude kinematics is modeled using quaternion while the dynamics of the CubeSat's attitude considers reaction wheel actuation and gravity gradient torque.

The optimal state penalizing matrix Q= 10^{-3}*diag ([1.053 1.053 1.053 1.053 1.053 1.053]), state input matrix R=diag ([20.2422 20.2422 20.2422]) and K gain matrix that minimizes linear quadratic performance index are obtained from genetic optimization. Furthermore, the three fuzzy systems that mimic the state inputs of the genetically tuned linear quadratic regulator are developed using the adaptive neuro-fuzzy inference system. Each fuzzy system has 729 fuzzy rule bases with 18 generalized bell-shaped input and 729 linear output membership functions. Both controllers achieved zero steady-state error and settling time less than 12.5 seconds. The peak control signals in the linear quadratic regulator and fuzzy systems are below the maximum limit of 0.635 mN.m.

Key Words: *CubeSat, Linear Quadratic Regulator, Attitude Control, Genetic Algorithm, Adaptive Neuro-Fuzzy Inference system, ANFIS, Fuzzy Logic Controller, Quaternions, Reaction Wheel.*

Acknowledgments

I am very thankful for my advisor Dereje Shiferaw (Ph.D.) for his valuable comments and technical supports throughout the thesis. Also, I would like to thank Dr. Samson Mekbib for his consultations and guidance regarding research methodology. Finally, I am grateful for Mr. Biniam Hussien, who allowed me to use internet access in his office, for his kindness.

Table of Contents

Acronyms

ADCS	Attitude Determination and Control System
ANFIS	Adaptive Neuro-Fuzzy Inference System/ Adaptive Network-Based Fuzzy Inference System
BFBF	Body-Fixed Body Frame
BCOF	Body-Centered Orbital Frame
BPA	Backpropagation algorithm
FLC	Fuzzy Logic Controller
GIF	Geocentric Inertial Frame
ISS	International Space Station
J-SSOD	Japanese Small Satellite Orbital Deployer
LMS	Least Mean Squared
LQR	Linear Quadratic Regulator
MF	Membership Function
MSDR	Modified State-Dependent Riccati
NRCSD	Nano Racks CubeSat Deployer
POS	Particle Swarm Optimization
P-POD	Poly Picosat Orbital Deployer
SDRE	State-Dependent Riccati Equation
TSK	Takagi-Sugeno-Kang

List of Tables

List of Figures

1 Introduction

1.1 Background

1.1.1 Introduction about CubeSat

CubeSat is a miniaturized satellite designed for researches and space applications. The unit CubeSat has cube-shaped geometry that is 10 x 10 x10 centimeters cube and weighs 1.33 kilograms. Also, a number of a unit CubeSat can be combined to develop two-unit, three-unit, four-unit, or N-unit CubeSat. Since CubeSat's components are available in the market as commercial-off-the-shelf components, any hobbyist, technologist, or University researchers can develop a satellite with little cost using CubeSats.

In 1999 two Professors, Professor Jordi Puig-Suari and Professor Bob Twiggs brought the idea of standard CubeSat designs. The Professors' idea was to create a chance for the students to study satellites from designing to operation stages [1].

Miniature satellites are usually classified into different groups based on their mass. These are Femto (0.01-0.1 kg), Pico (0.1-1 kg), Nano (1-10 kg), Micro (10-100 kg), and Mini (100-500 kg) satellites. The term CubeSat is to indicate nanosatellites.

CubeSat can be deployed to space using specially designed deploying devices that are mounted on International Space Station or other space vehicles like rockets, large satellites, and spacecraft. The most common deployment devices are Nano Racks CubeSat Deployer (NRCSD), Poly Picosat Orbital Deployer (P-POD), Japanese Small Satellite Orbital Deployer (J-SSOD), and other deploying devices designed by deployment service providers.

CubeSat is applicable in the areas that the normal applications of the large satellites are applicable at which miniaturization of CubeSat has no effect. These applications involve experiments that can be done on a miniatured level, observations, amateur radio service, test and demonstration of new technologies, testing mission's feasibility, and research applications for students and researchers at low cost. Several deep space missions are planned with CubeSat. Besides this, the CubeSat formation flight attracts many researchers.

The development of satellite needs a synergy of different disciplines and so do the CubeSat. The components and systems in the CubeSat are designed based on the mission goals from different

discipline perspectives. The CubeSat as a single system has command and data handling, power, structure, attitude determination and control, thermal control, and communication subsystems. Attitude determination and control system (ADCS) is one of the fundamental subsystems responsible for the determination and control of the satellite orientation.

Generally, the CubeSat can be used for the desired mission that may possess antenna for communication, camera for image data collection for the specific issue, payload, and the supplementary electronics system that is sensitive to sun radiation, solar panels for the power generation, the scientific data collection devices, and similar components. The satellite orientation can determine the performances and lifetime of those devices. Thus, attitude determination and control of the CubeSat is a crucial subsystem to the total functioning of the satellite.

1.1.2 Satellite Attitude Determination and Control

The attitude of the satellite is about its orientation in the three-dimensional reference frame. Generally, it is studied as attitude determination and attitude control. Attitude determination is determining the past and the current attitude with respect to a given inertial reference system while attitude control deals with enabling the satellite to orient itself to the desired stable orientations with the desired performance by applying different control algorithms.

A. Attitude Determination

Attitude is determined by implementing attitude determination algorithms that use the current and the past directional measurement vectors, such as the direction of the sun and star constellations, albedo ray and infrared vectors of Earth and magnetic field vectors of Earth about satellite body-fixed body frame of reference provided that the inertial orientations of those measured vectors are known. Then a determined attitude is given as feedback to an attitude control system. This feedback is usually considered to be fully determined and available in controller design.

B. Attitude Control

The attitude control in the aerospace field of study is commonly named as attitude stabilization. Attitude control of the satellite uses the determined attitude as feedback to the controllers and sends a control signal to the actuator to reorient the satellite in the desired attitude.

The attitude control methods employed in satellite attitude stabilization can be either passive or active control. The passive control method does not need any active power source. It uses external

disturbances and passive devices in the satellite to develop actuating torque. Torque developed due to the interaction of permanent magnet and Earth's magnetic field, gravity gradient, aerodynamics, and solar radiation torques are used for passive stabilization. It is cost-effective but less efficient for pointing control. Moreover, this controller system is predefined, non-robust, and non-adaptive. Many of the early satellites use passive controllers.

On the other hand, the second group of control method is an active control which uses active power source and the control signal is actively induced to the actuators. Active controls are either spin stabilization or three-axis stabilizations [2].

In spin stabilization, the satellite spins about the desired spin axis. The stability of the satellite attitude depends on as gyroscopic resistance of the spinning. It can also be a single-axis or double axis spin-stabilized [3]. But the better attitude stabilization with high inertial and non-inertial pointing accuracy is achieved using three-axis stabilization. The majority of the control design progress in the satellite attitude control is made in the three-axis stabilization [3, 4]. It is actuated using internally developed torques and torque developed from the interaction of external disturbance and internally embedded devices. The common actuation devices are magnetic-torques, momentum exchanging devices (control moment gyros, reaction wheel, and momentum wheels) and thrusters [3].

In addition to separately applying the active and the passive controllers, they can sometimes be combined in attitude control. Aerodynamics control is usually combined with active control methods [5, 6]. Moreover, solar radiation torque can be embedded in attitude control with active magnetically actuated attitude control [7].

1.2 Motivation and Problem Justification

Satellite design and launching is a sophisticated technology that needs a lot of investment and highly qualified human resources. The low-cost satellites can be developed by miniaturizing the satellite. It is important for developing countries, universities, and amateur technologists to develop low scaled cost-effective satellites. Students can develop their satellite using cost-efficient off-shelf components. Also, CubeSat technology can be a startup technology for countries that their space technology is not developed yet.

CubeSat is the future promising satellites that different disciplines come together giving life to this technology. One of the challenges of CubeSat development is its orientation control, scientifically named as attitude control.

Different control techniques are developed and implemented for attitude control problems. There are still researches to be conducted to address the attitude control problem. From the different control methods, reaction wheel actuated linear quadratic regulation (LQR) and fuzzy logic controller (FLC) are the common methods in this area.

LQR for CubeSat attitude control needs a proper tuning with the consideration of the system as an energy-constrained system to avoid actuator saturation. This makes the LQR tuning problem as constraint optimization. This optimization is considered to be addressed using a genetic algorithm. Also, FLC for CubeSat attitude control in the situation of multiple inputs and multiple outputs system becomes a complex problem that involves too many membership functions and fuzzy rules. Thus, the system becomes tedious to develop the fuzzy system using rational guess and expertise experience. Besides, it will not be well designed FLC since it passes in try and error tuning; therefore, learning-based fuzzy tuning using the Adaptive Neuro-Fuzzy Inference System (ANFIS) is proposed for FLC system development.

1.3 The objective of the Study

1.3.1 General Objective

The thesis's general objective is to develop a genetic algorithm and a neuro-fuzzy system for LQR based attitude control.

1.3.2 Specific Objective

The specific objectives are

- Developing kinematics and dynamics model of the attitude of the CubeSat system.
- Developing state-space modeling of the attitude of CubeSat from kinematics and dynamics models.
- Tuning the LQR controller using genetic algorithms.
- Sampling state and control signals from the simulation of LQR controlled attitude dynamics and kinematics

- Apply adaptive neuro-fuzzy inference system (ANFIS) to generate fuzzy rules and tune parameters of selected input membership functions using a sampled state and control signal.
- Validate and test the ANFIS training.
- Designing a fuzzy control using generated fuzzy rules and membership functions.
- Simulate the fuzzy controlled CubeSat attitude.

1.4 Significance of the Study

Different attitude control algorithms have been used and one of those control methods is applying a linear quadratic controller. Tuning this controller has been a trial and error task. This is a tedious and time-consuming process that does not directly consider actuator energy to avoid actuator saturations. In this thesis, optimal and constraint submissive LQR tuning is developed using a genetic algorithm based on actuator constraint objective or cost function to handle optimization between controller performance and actuator saturation.

The other commonly studied controller is a fuzzy controller. The fuzzy membership functions and fuzzy rules are selected by the fuzzy system designers and may not be properly designed. Besides, rules are also developed from the knowledge of the dynamics of the plant and experience of the expertise that will be tuned by trial and error. FLC that passes in this process is not efficient especially for multiple inputs and multiple outputs systems. The second attribute of this thesis is addressing the fuzzy controller design problem using ANFIS training by taking data from the simulation of genetically tuned LQR of the attitude of CubeSat. Best fuzzy rules and proper tuning of membership functions are developed from the training.

1.5 Thesis Outline

This document is organized into seven chapters. Chapter one introduces this thesis. It starts by introducing the general introduction about CubeSat and the highlights of attitude determination and control as a background. Moreover, the statement of the problems that the thesis tries to address, the general and specific objective as the goal of the research, the contribution of the thesis that it brings and its outline are contents of the first chapter.

The second chapter consists of the literature reviews on the area of attitude control of the satellite specifically the CubeSat. The different active control trends are considered. The magnetic and

reaction wheel actuators with currently applied active control algorithms are reviewed. The limitation and gaps of these controllers are stated.

Chapter three is about system dynamics and kinematics modeling. This chapter discussed different reference frames, different attitude parametrization, and quaternion rates as attitude kinematics model, different disturbance torques in the low earth space environment, and attitude dynamics. Furthermore, the kinematics and dynamics model of the attitude is linearized and combined to give the state-space model.

Chapter four discusses the general background of the fuzzy membership functions, the fuzzy rule base, the fuzzy inference system, the neuro-fuzzy system and its learning method, and the genetic algorithm.

Chapter five is on controller design. Linear quadratic regulator design with genetic algorithm optimization is included. Also, the second controller design, fuzzy logic controllers using adaptive neuro-fuzzy inference system hybrid learning rule by combining the least square method and backpropagation are considered.

Chapter six is the controller simulation of the controllers designed in chapter five. Graphical representation of different state responses and input control signals are shown and their interpretations are discussed.

The last chapter is a conclusion and recommendation. The conclusions from simulation results and its interpretations are drawn and future recommendations are stated.

2 Literature Review

The literature on satellite attitude control is vast and extensive. The attitude control includes linear, non-linear, adaptive, robust, optimal, and intelligent controls. The synergies of these controllers have also been used. This section tries to review the active attitude control of satellites and investigates different control goals with a distinct type of controller.

2.1 Commonly Used Actuators in Attitude Control of CubeSat

The control actuators define the model types of attitude dynamics. The magnetic-torquer and reaction wheels are the common actuators used in the CubeSat attitude control. Satellites use these actuators as solely magnetic actuation, solely reaction wheel actuation and combining magnetic actuation and reaction wheels. Each actuation has its own cons and pros.

Different researchers studied the magnetically actuated attitude control. The major challenge of magnetic attitude control is the attitude is not controllable along the Earth's magnetic field direction since the magnetic field strength of Earth's magnetic field and magnetic dipole moment of the actuator are parallel; therefore, conventional magnetic controller losses its controllability. This makes the satellite take a longer period to reach steady-state. It became steady after a few counts of full orbital rotations. Thus, fully magnetically actuated attitude control is periodically controllable and challenging for instantaneous control. One way of solving the problem of uncontrollability of attitude with full magnetic actuators at any instant is combining reaction wheel and magnate-torque [8]. Another new approach is a two-step rotation using magnate-torque to counteract periodic controllability of satellites using magnetic actuation. The two-step rotation proportional-derivative controller can give a better result in terms of controllability [9].

Reaction wheels are the other widely used actuator for CubeSat attitude control. It is more effective compared to magnetic actuators since it allows fully three axes controllability of attitude of the satellite and it can also bring the attitude of CubeSat to equilibrium in a short time. Although it is advantageous to use reaction wheels, it is exposed to momentum accumulation that needs a means of dissipation. Magnetic actuators are usually used to damp the accumulated momentum [2, 4]. Another disadvantage of the reaction wheel is practically they are prone to wear and fatigue failure.

2.2 Attitude Controller

Classical controllers like proportional, proportional-derivative, and proportional-derivative-integral controllers have been a solution design for attitude control problems [4]. The coupled and non-linear attitude models are treated as a linear and decoupled different single input single output equations. The performances of those controllers are not as effective as the mission requirements.

Also, the linearized and non-linear attitude dynamics treated by a linear and non-linear controller. These models with constant infinite quadratic cost [10], gain quadratic cost, and finite quadratic cost [11], sliding mode [12], and energy-based Lyapunov-stable controllers [13] are investigated using full magnetic actuation. Further, magnetic controllers are applied for spacecraft subjected to gravity gradient torque using energy-based Lyapunov-stable controllers [13]. Likewise, the non-linear controllers are applied directly to linear time-variant and non-linear behavior of the system. Some of these are state dependent Riccati equation (SDRE) [14], PD and quadratic regulators for linear time-variant and compared to linear time-invariant control [3], Modified state-dependent Riccati (MSDR) equation using pseudo linearization [15], sliding mode control for fully magnetic actuation to solve under actuation problem of magnate-torque at transient motion [16] and finite sliding mode control to have a continuous control signal and reduce chattering [17]. There are also other classes of controllers developed for attitude control of a satellite. The next section of this chapter discusses these controllers.

2.2.1 Adaptive and Robust Controllers

Besides the controllers developed considering accurate mathematical models, there are robust and adaptive controllers studied based on the parameter and model uncertainty in the attitude dynamics and kinematics due to wear, inaccurate mass inertial prediction, and disturbance variations. Thus, some papers try to investigate the attitude control problem with a robust and adaptive controller design.

The robust controller design ensures controller performance during inertial matrix uncertainty [18] and disturbance torque [19]. A robust fuzzy controller for variable mass inertia can solve model uncertainty [8]. In addition, some satellite attitude problems need adaptive controls. The disturbances within the known bound can be actively rejected using adaptive controls [20]. A control problem of satellite reorientation exposed to parameter uncertainty, actuator fault, external

disturbance, and input saturation is addressed using fault tolerance adaptive control [21]. Non-linear adaptive fault tolerance method of control is developed to achieve a tracking three-axis attitude control when the attitude control design is exposed to those problems [22]. Also, the same control problem with the reaction wheel fault is solved with fault tolerance attitude control based on adaptive extended-state observer [23].

Moreover, adaptive fuzzy controls consider disturbance estimation and improving the membership function that is compliant to lower power cost and mission requirements compared to classical controllers. This adaptive fuzzy controller is actuated using three magnet-torquers and one momentum wheel [24].

2.2.2 Linear Optimal Controllers

Linear optimal controllers are usually used in the attitude control design due to easy of design, effective performance in the given tolerance range, the possibility to consider energy constraints, and the ability to achieve the mission goals without complex controller designs.

Commonly, the linearized model of magnetic actuation is evaluated with linear quadratic controllers [3, 25, 26]. Besides, the optimal control of periodic linear regulators control methods, and optimal periodic disturbance rejection is investigated [27]. Studies of those controllers including the linear quadratic controller and its non-linear extension show that they perform better over PID controllers [28].

The challenges come when the tuning of linear optimal controllers based on some objective performance range is needed. Manual tuning of this controller is so challenging that it needs a lot of iterations till better performance is achieved. Manual tuning is tiresome and computationally intense. Even the result of manual iteration has less possibility in achieving optimal design. This optimization problem is addressed using different optimization methods. A genetic algorithm is one of the optimization techniques [29]. It is an evolutionary algorithm that can tune the linear quadratic regulator (LQR), one of the most common linear optimal controllers. Genetically tuned LQR based magnetic actuation out performances conventional PID, PD, quaternion feedback controller, and LQR methods [30].

The proper tuning of the LQR controller using a reaction wheel actuator is commonly achieved manually which brings doubt about the optimality in terms of performance and actuator saturation.

Intelligent optimization methods are commonly recommended in different research as a means of tuning LQR.

2.2.3 Intelligent Controllers

The need of achieving space missions without failure increases satellite complexity. This leads to a search for better attitude control techniques. Intelligent controllers take the position to address this need. Intelligent controllers are based on fuzzy logic, neural network, neuro-fuzzy, and intelligent optimization techniques.

Fuzzy controllers are appropriate for attitude magnetic control since the attitude kinematics and dynamics model is time-variant and non-linear. Comparatively, fuzzy controllers show better performance compared to classical controllers in terms of convergence and steady-state error [31]. Fuzzy controller outfits linear quadratic controllers. It was investigated for a magnetically actuated fuzzy logic controller based on a gain scheduled proportional-derivative controller that they are lower cost and fast settling solution over linear quadratic controllers [29].

The fuzzy controller usually developed intuitively and using a rational guess. PD based fuzzy classifiers can be tuned using genetic algorithms to avoid the intuitive fuzzy inference controller [32]. Furthermore, Real-time correction and attitude control modification modifies the fuzzy system and gives better performance [33]. The desired torque to actuate the attitude to the desired orientation using magnetic attitude control is determined using fuzzy logic and the desired magnetic dipole moment is selected from the desired torque [34]. Similarly, magnetically actuated fuzzy control combined with gravity gradient stabilization improves the sole gravity gradient control [35]. The fuzzy controller like linear quadratic controller can be tuned using genetic algorithms to develop optimal controllers that are submissive to the desired objective function [29].

Besides, intelligent controls and a combination of intelligent control with conventional control are studied for satellite attitude control. MSDR and neuro-fuzzy controller designed using the solution of the MSDR equation are combined to reduce the computation of burden MDRE [36]. A similar approach to reduce the computational burden of SDRE is developed as a good approximator of SDRE using the adaptive neuro-fuzzy inference system [37]. The other studied combination of intelligent and conventional controls are adaptive fuzzy and robust fuzzy controllers. The adaptive

fuzzy control considers disturbance estimation and membership function improvement [24] while a robust fuzzy controller considers model uncertainty [8].

The common means that the intelligent controller applied is handling attitude problems that suffer from the limit cycle due to noisy and inaccurate measurements. To cope with this problem, a non-linear fuzzy controller tuned by reinforcement learning is developed [38]. Also, some satellites need articulation of components and movement of flexible appendages that increase the non-linearity of the system dynamics. Adaptive fuzzy sliding mode utilized to control the attitude control of satellites exposed to this type of non-linearity [39]. Besides, intelligent adaptability using a neural network helps to achieve better performance during unknown disturbance and actuator faults [40].

2.3 Research Gap

Generally, the attitude determinations and control using non-linear and linear control methods for a CubeSat have been studied. There are still different problems in CubeSat attitude control that needs to be addressed. Some of these are:

- Proper tuning of reaction wheel actuated LQR needs consideration of good performance and avoiding reaction wheel saturation without a need of extra actuator like magnetic torquer to damp the saturation.
- Fuzzy controllers are tuned and designed based on rational guesses and through try and error iteration. FLC for CubeSat attitude control in the situation of multiple inputs and multiple outputs system becomes a complex problem that involves too many membership functions and fuzzy rules. It will be challenging to addressed this type of fuzzy logic control design problem using manual try and error based rational guess.

This thesis tries to investigate the following solutions that are thought to solve the problems mentioned earlier.

- **Constraint Genetic Algorithm Optimization**

  The proper tuning of reaction wheel actuated LQR is taken as an optimization problem that balances actuator energy saturation by considering constrained actuator and good

performance. This will be addressed with Genetic algorithm optimization that uses fitness function developed by taking accounts of performance and actuator saturation.

- **Adaptive Neuro-Fuzzy Inference System/ANFIS**

  FLC of CubeSat attitude control for multiple inputs and multiple outputs system is a complex problem. Thus, the system becomes tedious to develop the fuzzy system using rational guess and expertise experience. Learning-based fuzzy tuning using the Adaptive Neuro-Fuzzy Inference System (ANFIS) is proposed for optimal FLC system development.

  The state and state input sampled data from genetically tuned LQR simulation will be given to ANFIS. A hybrid learning method that combines backpropagation and least square estimator available in the ANFIS system will develop three fuzzy systems using the sampled data. The three distinct fuzzy systems separately control the three reaction wheels.

3 Mathematical Model of Cubesat's Attitude

The knowledge of the orientation of a satellite is the basis for the success of the mission and the payload performance. This orientation is usually called the attitude of the satellite. Thus, any satellite with a deviation from the desired orientation needs an actuation to reorient to the right attitude.

The attitude varies depending on the reference frames referred to. It can also be expressed in a three by three matrix format, or one or more axis of rotation with the corresponding angle which yields different methods of attitude parametrization.

This chapter tries to address the concepts of reference frames, attitude parametrizations, and disturbance torques that can be the cause for attitude deviation. The main mathematical model of attitude is also discussed in terms of attitude kinematics and dynamics.

3.1 Reference Frames

Reference frames determine the attitude of the satellite since the attitude depends on the measurement vectors like sun vector, earth nadir vector, gyroscope measured angular position vector, magnetic field vector, or star pointing vectors.

The reference frames can be either an inertial reference frame or non-inertial frames. Inertial reference frames are fixed or move within a constant speed along a straight line relative to a universally agreed inertial reference frame. Non-inertial reference frames are either accelerating or decelerating relative to universally agreed inertial reference frame.

The thesis applies both inertial and non-inertial reference frames to model and expresses the kinematics and dynamics of both attitudes. Three reference frames that are used in this thesis are discussed.

3.1.1 Body-Fixed Body Frame (BCBF)

It is a non-inertial reference frame attached to the body of the CubeSat. Its origin is usually located at the center of gravity of the satellite. Measurements using different sensors like sun sensor, magnetic field sensor, earth horizon sensor, star trackers, gyroscopes, rate gyros and other similar vector measuring sensors will refer to this reference frame.

The axis and origin of this reference frame are user-defined. This thesis assumes origin at the center of gravity of the CubeSat. The X-axis, Y-axis, and Z-axis are perpendicular to the faces of the CubeSat sides and the three-axis aligned with the three principal moments of inertia of the CubeSat. Z-axis is pointing from origin to earth core center while Y-axis is pointing negative normal to the orbit that the satellite rotates around the earth and X-axis is can be defined based on the right-hand vector rule.

3.1.2 Geocentric Inertial Frame (GIF)

This inertial reference frame has an origin located at the center of mass of the earth, X-axis points in the direction of the Autumn Vernal Equinox (it lies along the line of intersection of the earth's equator plane and ecliptic plane (a plane that holds the earth's orbit path around the sun)) and it is defined by the direction from Earth's center to the sun's center at the time the sun appears to pass through the Earth's equatorial plane moving from south to north at the first day of spring around September 23 southern hemisphere or march 21 in the northern hemisphere. Y-axis points 90 degrees to the east of the X-axis through the Earth's equatorial plane. The Z-axis is normal to the equatorial plane or it can be defined by the right-hand vector rule.

The geocentric inertial reference frame is assumed to be an inertial reference frame for the satellite even though the earth rotating around the sun in an elliptical path with a certain acceleration. In addition, the line of intersection of the earth's equator plane and ecliptic plane experiences nodal precession at a slower rate. The Vernal Equinox direction changes as a consequence. This reference frame is defined at some epoch due to the Vernal Equinox change. This thesis refers to an epoch of J2000 reference.

3.1.3 Body-Centered Orbital Frame (BFOF)

The origin is located at the center of gravity of the CubeSat. The Z-axis pointing from the origin to the Earth center. The Y-axis directed in the opposite direction to the orbit normal. The X-axis can be defined in the right-hand rule.

3.2 Attitude Parametrization

Attitudes parameters are attitude coordinates that describe the orientation of any rigid body about the desired reference frame. There are different attitude coordinates as an analogy to translation coordinates, such as rectangular, cylindrical, and spherical coordinates. In the case of translational

coordinates, the separation distance between two points is expressed at maximum as infinity distance. But for attitude coordinates, the orientation differs by 180^0 at maximum.

The choice of attitude coordinates affects the mathematical model linearity or non-linearity and the mathematical or geometrical singularity. After all, it affects the limit and type of operation of the attitude control system.

Generally, the relative angular position between any two reference systems requires at least three coordinates. These minimum three coordinates will have at least one singular geometrical orientation. The corresponding kinematic differential equation is singular at that geometrical singularity point. This mathematical and geometrical singularity can be regularized by using universally defined redundant coordinates which applies more than three coordinates.

Directional cosine matrix, Euler angle representation, Euler angle /axis representations or principal rotation vectors and principal angle, quaternions, Classical Rodrigues parameters, and modified Rodrigues parameters are commonly used attitude parameterizations. From all aforementioned attitude representation methods, quaternions and directional cosine matrixes are free from the geometrical and mathematical singularity at any point since they use parameters more than the minimum requirement of three parameters. The directional cosine matrix has 9 angles to describe the attitude which needs six more parameters than the minimum requirement of three parameters whilst quaternion uses four parameters to define a given attitude which needs one more parameter than the minimum requirement of three parameters. This infers Quaternions are less redundant and non-singular attitude parametrization. Therefore, the quaternion is considered to be the attitude parametrization for this thesis [41, 42].

Euler parameters or quaternions are a non-singular parametrization method that has four parameters. Thus, quaternions exceed the minimum three parameters to express any orientation and they are always involving a redundant fourth parameter [43]. Any orientation of a rigid body with respect to any reference frame can be defined using four parameters which are defined in terms of a unit Eigenvector and an angle. The Eigenvector is a three-dimensional vector that relates the origin of the two reference frames and the angle is the rotation of one reference frame with respect to the other reference frame about the Eigenvector.

The quaternion can be expressed as follows [41]:

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\ {}^{\theta}/_{2} \\ e_1 sin\ {}^{\theta}/_{2} \\ e_2 sin\ {}^{\theta}/_{2} \\ e_3\ sin\ {}^{\theta}/_{2} \end{bmatrix} \tag{3.1}$$

From the four parameters definition, it is clear that q_0 is a scaler, and q_1, q_2, and q_3 are the vector components of the quaternion.

The four quaternions and Eigenvector magnitudes are constrained.

$$\begin{aligned} 1 &= {q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 \\ 1 &= {e_1}^2 + {e_2}^2 + {e_3}^2 \end{aligned} \tag{3.2}$$

The transformation matrix using quaternions is given in [41, 42] and it is defined as:

$$C(q) = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix} \tag{3.3}$$

For the attitude of the rigid body, there are two sets of quaternions –q and q that describe the same orientations. One describes the shortest single-axis rotation the other describes the longest. The shortest rotation necessarily satisfies $|\theta| \leq 180^0$. This indicates that $cos\ {}^{\theta}/_{2} \geq 0$ and q_0 is always positive for the shortest rotation direction.

The reverse process of finding the quaternion from the transformation matrix is developed by Sheppard [41].

$$\begin{aligned} {q_0}^2 &= \frac{1}{4}(1 + trace[C]) \\ {q_1}^2 &= \frac{1}{4}(1 + 2C_{11} - trace[C]) \\ {q_2}^2 &= \frac{1}{4}(1 + 2C_{22} - trace[C]) \\ {q_3}^2 &= \frac{1}{4}(1 + 2C_{33} - trace[C]) \end{aligned} \tag{3.4}$$

Then Sheppard takes the square root of the largest ${q_i}^2$ from the above equations and the sign of q_i is arbitrarily chosen to be positive. The other quaternion values can be found by dividing three of the appropriate equations shown below with the largest square root of ${q_i}^2$.

$$\begin{aligned} q_0 q_1 &= (C_{23} - C_{32})/4 \\ q_0 q_2 &= (C_{31} - C_{13})/4 \\ q_0 q_3 &= (C_{12} - C_{21})/4 \\ q_2 q_3 &= (C_{23} + C_{32})/4 \\ q_3 q_1 &= (C_{31} + C_{13})/4 \\ q_1 q_2 &= (C_{12} + C_{21})/4 \end{aligned} \tag{3.5}$$

Where C_{ij} is for i row and j column in the transformation matrix of C. An alternative quaternion values are the negatives of the q_i values found. Composite or sequential rotation combines by multiplying the rotations in reverse order.

Quaternions provide a continuous representation of three-dimensional space without singularity or gimbal lock. They are computationally simple to apply. Kinematics of the quaternions are linear differential equations. But they are four element parameterizations developed from one Eigenvector and one rotational angle.

3.3 Attitude Kinematics

The kinematics of the attitude is expressed in terms of the rates of the attitude parameterization. The rate of quaternion will be used to model the kinematics. Assuming any single or multiple sequence rotation can be equivalently expressed using a rotation about unit Eigenvector, e, with rotation angle, θ, as stated in the Euler rotation theorem,

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\ {\theta}/{2} \\ e_1 sin\ {\theta}/{2} \\ e_2 sin\ {\theta}/{2} \\ e_3\ sin\ {\theta}/{2} \end{bmatrix} \tag{3.6}$$

$$\dot{q} = \lim_{t \to 0} \frac{q(t + \Delta t)\ -\ q(t)}{\Delta t} \tag{3.7}$$

$$q(t+\Delta t) = \begin{bmatrix} cos\,(\theta/_2 + \Delta\theta/_2) \\ e_1 \sin\,(\theta/_2 + \Delta\theta/_2) \\ e_2 sin(\theta/_2 + \Delta\theta/_2) \\ e_3\, sin(\theta/_2 + \Delta\theta/_2) \end{bmatrix} = \begin{bmatrix} cos\,(\theta/_2) cos(\Delta\theta/_2) - \sin\,(\theta/_2) sin\,(\Delta\theta/_2) \\ e_1\,(\sin\,(\theta/_2) cos(\Delta\theta/_2) + \sin\,(\Delta\theta/_2) cos\,(\theta/_2)) \\ e_2\,(\sin\,(\theta/_2) cos(\Delta\theta/_2) + \sin\,(\Delta\theta/_2) cos\,(\theta/_2)) \\ e_3\,(\sin\,(\theta/_2) cos(\Delta\theta/_2) + \sin\,(\Delta\theta/_2) cos\,(\theta/_2)) \end{bmatrix} \tag{3.8}$$

Introducing $(e_1{}^2 + e_2{}^2 + e_3{}^2)$ in the first row and $e_2e_3 \sin(\theta/_2) \sin(\Delta\theta/_2) - e_2e_3 \sin(\theta/_2) \sin(\Delta\theta/_2)$, $e_1e_3 \sin(\theta/_2) \sin(\Delta\theta/_2) - e_1e_3 \sin(\theta/_2) \sin(\Delta\theta/_2)$ and $e_1e_2 \sin(\theta/_2) \sin(\Delta\theta/_2) - e_1e_2 \sin(\theta/_2) \sin(\Delta\theta/_2)$ in the second, third and fourth rows respectively,

$$q(t+\Delta t) = \begin{bmatrix} cos\,(\theta/_2) cos(\Delta\theta/_2) - (e_1{}^2 + e_2{}^2 + e_3{}^2) \sin(\theta/_2) \sin(\Delta\theta/_2) \\ e_1 \sin(\theta/_2) cos(\Delta\theta/_2) + e_1 \sin\,(\Delta\theta/_2) cos(\theta/_2) + e_2e_3 \sin(\theta/_2) \sin(\Delta\theta/_2) - e_2e_3 \sin(\theta/_2) \sin(\Delta\theta/_2) \\ e_2 \sin(\theta/_2) cos(\Delta\theta/_2) + e_2 \sin\,(\Delta\theta/_2) cos(\theta/_2) + e_1e_3 \sin(\theta/_2) \sin(\Delta\theta/_2) - e_1e_3 \sin(\theta/_2) \sin(\Delta\theta/_2) \\ e_3 \sin(\theta/_2) cos(\Delta\theta/_2) + e_3 \sin(\Delta\theta/_2)\, cos(\theta/_2) + e_1e_2 \sin(\theta/_2) \sin(\Delta\theta/_2) - e_1e_2 \sin(\theta/_2) \sin(\Delta\theta/_2) \end{bmatrix} \tag{3.9}$$

$$q(t+\Delta t) = \begin{bmatrix} cos(\Delta\theta/_2) & -e_1 \sin(\Delta\theta/_2) & -e_2 \sin(\Delta\theta/_2) & -e_3 \sin(\Delta\theta/_2) \\ e_1 \sin\,(\Delta\theta/_2) & cos(\Delta\theta/_2) & e_3 \sin(\Delta\theta/_2) & -e_2 \sin(\Delta\theta/_2) \\ e_2 \sin(\Delta\theta/_2) & -e_3 \sin(\Delta\theta/_2) & cos(\Delta\theta/_2) & e_1 \sin(\Delta\theta/_2) \\ e_3 \sin(\Delta\theta/_2) & e_2 \sin(\Delta\theta/_2) & -e_1 \sin(\Delta\theta/_2) & cos(\Delta\theta/_2) \end{bmatrix} \begin{bmatrix} cos\,\theta/_2 \\ e_1 sin\,\theta/_2 \\ e_2 sin\,\theta/_2 \\ e_3\, sin\,\theta/_2 \end{bmatrix} \tag{3.10}$$

Using small-angle approximation, $\sin(\Delta\theta/_2) \approx \Delta\theta/_2 \approx {}^1/_2\, w\Delta t$ *and* $cos(\Delta\theta/_2) \approx 1$

$$q(t+\Delta t) = \begin{bmatrix} 1 & -{}^1/_2\, w_x\Delta t & -{}^1/_2\, w_y\Delta t & -{}^1/_2\, w_z\Delta t \\ {}^1/_2\, w_x\Delta t & 1 & {}^1/_2\, w_z\Delta t & -{}^1/_2\, w_y\Delta t \\ {}^1/_2\, w_y\Delta t & -{}^1/_2\, w_z\Delta t & 1 & {}^1/_2\, w_x\Delta t \\ {}^1/_2\, w_z\Delta t & {}^1/_2\, w_y\Delta t & -{}^1/_2\, w_x\Delta t & 1 \end{bmatrix} \begin{bmatrix} cos\,\theta/_2 \\ e_1 sin\,\theta/_2 \\ e_2 sin\,\theta/_2 \\ e_3\, sin\,\theta/_2 \end{bmatrix} \tag{3.11}$$

$$\dot{q} = \lim_{t \to 0} \frac{q(t+\Delta t)\; -\; q(t)}{\Delta t} \tag{3.12}$$

$$\dot{q} = \lim_{t\to 0}\left(\begin{bmatrix} 1 & -1/2\, w_x\Delta t & -1/2\, w_y\Delta t & -1/2\, w_z\Delta t \\ 1/2\, w_x\Delta t & 1 & 1/2\, w_z\Delta t & -1/2\, w_y\Delta t \\ 1/2\, w_y\Delta t & -1/2\, w_z\Delta t & 1 & 1/2\, w_x\Delta t \\ 1/2\, w_z\Delta t & 1/2\, w_y\Delta t & -1/2\, w_x\Delta t & 1 \end{bmatrix}\begin{bmatrix} cos\ \theta/2 \\ e_1 sin\,\theta/2 \\ e_2 sin\,\theta/2 \\ e_3\ sin\,\theta/2 \end{bmatrix} - \begin{bmatrix} cos\ \theta/2 \\ e_1 sin\,\theta/2 \\ e_2 sin\,\theta/2 \\ e_3\ sin\,\theta/2 \end{bmatrix}\right)\frac{1}{\Delta t} \quad (3.13)$$

$$\dot{q} = \lim_{t\to 0}\left(\begin{bmatrix} 1 & -1/2\, w_x\Delta t & -1/2\, w_y\Delta t & -1/2\, w_z\Delta t \\ 1/2\, w_x\Delta t & 1 & 1/2\, w_z\Delta t & -1/2\, w_y\Delta t \\ 1/2\, w_y\Delta t & -1/2\, w_z\Delta t & 1 & 1/2\, w_x\Delta t \\ 1/2\, w_z\Delta t & 1/2\, w_y\Delta t & -1/2\, w_x\Delta t & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)\begin{bmatrix} cos\ \theta/2 \\ e_1 sin\,\theta/2 \\ e_2 sin\,\theta/2 \\ e_3\ sin\,\theta/2 \end{bmatrix}\frac{1}{\Delta t} \quad (3.14)$$

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} q(t) \quad (3.15)$$

The kinematics equation can be further rearranged to a more elegant form that relates angular velocity and quaternions rates.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}\begin{bmatrix} 0 \\ w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.16)$$

The transformation matrix between angular velocity ‘w’ and quaternion rates is a non-singular and orthogonal matrix.

$$\begin{bmatrix} 0 \\ w_x \\ w_y \\ w_z \end{bmatrix} = 2\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (3.17)$$

The quaternion rate vectors $\dot{q} = (\dot{q}_1, \dot{q}_2, \dot{q}_3)$ and the angular velocity can be related using the above kinematics equation as shown below.

$$w = 2(q_0\dot{q} - \dot{q}_0 q) - 2[q\times]\dot{q}_b^0 \quad (3.18)$$

Where: $[q\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$

3.4 Disturbance Torques in Low Earth Orbit Space Environment

The disturbance on the system depends on the system environment. The space environment consists of different disturbances. These disturbances can also be used for the actuator, as an example, magnet-torquer uses the earth's magnetic field. For satellites in the vicinity of the Earth, the major disturbance torques are, magnetic-torque, solar radiation pressure torque, aerodynamic torque, and gravity-gradient torque.

The torque due to the magnetic field is neglected since there is no magnetic coil used as an actuator that will interact with the earth's magnetic field. In addition, the residual magnetic dipole moment produced by the onboard electronics is too low to produce torque by interacting with Earth's magnetic field. The torques due to solar pressure and drag are also neglected since center action of the resultant solar pressure force and the resultant drag force are the geometrical centers of the six faces of the CubeSat. The geometrical center is considered as the center of mass of the CubeSat; therefore, the distance between either solar pressure force or drag force action point normal, and the principal axis of rotation is zero. This makes the torques due to the solar pressure and drag is zero.

Moreover, this thesis considers gravity gradient torque as a disturbance torque that affects the CubeSat. The gravitational force is varying with the distance from the center of the Earth. It decreases as a distance increases from the center of the earth. Torque will be induced due to this gravity-gradient.

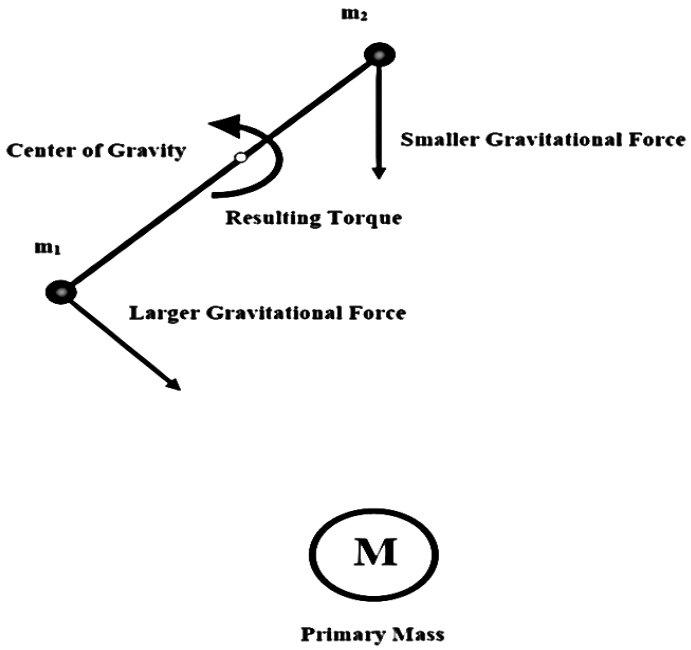


Figure 3.1 The Effect of Gravity Gradient Torque

Considering the above figure, two mass connected by the massless rigid rod as a system, there is unequal gravitational force acted on the m_1 and m_2 by M. The torque will be developed on the system due to this unbalanced force.

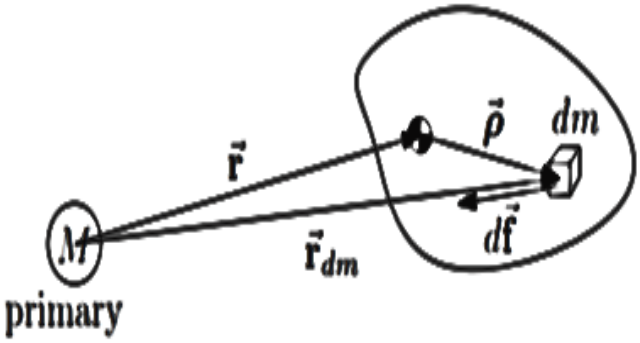


Figure 3.2 Position Vector Representation between Two Bodies

By considering the continuous rigid body, gravity-gradient torque can be modeled. This model is developed using the reference [44]. The gravitational force on the infinitesimal mass element dm positioned at $\vec{\rho}$ from the satellite center of mass is

$$d\vec{F}_g = -\frac{\mu(\vec{r}+\vec{\rho})}{|\vec{r}+\vec{\rho}|^3}dm \tag{3.19}$$

Where, $\vec{r}$ is the orbital position of the satellite center of mass, μ is universal gravitational constant. It is assumed that $\rho \ll r = \vec{r}$. By applying, first-order Taylor series expansion approximation,

$$|\vec{r} + \vec{\rho}|^{-3} \approx \frac{1}{r^3}\left(1 - \frac{3\vec{r}.\vec{\rho}}{r^2}\right) \tag{3.20}$$

Where, $r = |\vec{r}|$. Expanding the gravitation force expression by substituting the above equation, the gravitational force acting on the mass element dm will be

$$d\vec{F}_g = -\frac{\mu(\vec{r} + \vec{\rho})}{r^3}\left(1 - \frac{3\vec{r}.\vec{\rho}}{r^2}\right)dm \tag{3.21}$$

The total torque on the center of mass is

$$\vec{T}_g = \int \vec{\rho} \times d\vec{F}_g = \int \vec{\rho} \times \left(-\frac{\mu(\vec{r} + \vec{\rho})}{r^3}\right)dm + \int \vec{\rho} \times \left(\frac{3\mu(\vec{r}.\vec{\rho})}{r^5}(\vec{r} + \vec{\rho})\right)dm \tag{3.22}$$

The integral over the entire satellite body, $\int \vec{\rho}dm = 0$ since the center of mass is at the origin. By taking $\int \vec{\rho} \times \vec{\rho} = 0$ and $\int \vec{\rho}dm = 0$, the total torque is simplified:

$$\vec{T}_g = \frac{3\mu}{r^5}\int \vec{\rho} \times \vec{r}(\vec{r}.\vec{\rho})dm \tag{3.23}$$

The torque, orbital position vector, and position vector of infinitesimal mass element dm from the satellite center of mass are expressed in the body reference frame (b_1, b_2, b_3).

$$\begin{gathered}\vec{T}_g = T_{gb1}\vec{b}_1 + T_{gb2}\vec{b}_2 + T_{gb3}\vec{b}_3 \\ \vec{r} == r_{b1}\vec{b}_1 + r_{b2}\vec{b}_2 + r_{b3}\vec{b}_3 \\ \vec{\rho} = \rho_{b1}\vec{b}_1 + p_{b2}\vec{b}_2 + \rho_{b3}\vec{b}_3\end{gathered} \tag{3.24}$$

The cross product $\vec{\rho} \times \vec{r}$ can be expanded as

$$\vec{\rho} = [\rho_{b1} \quad p_{b2} \quad \rho_{b3}]\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} and\ \vec{r} = [r_{b1} \quad r_{b2} \quad r_{b3}]\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{3.25}$$

$$\vec{\rho} \times \vec{r} = [\rho_{b1} \quad p_{b2} \quad \rho_{b3}]\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \times [b_1 \quad b_2 \quad b_3]\begin{bmatrix} r_{b1} \\ r_{b2} \\ r_{b3} \end{bmatrix} \tag{3.26}$$

$$\vec{\rho} \times \vec{r} = [\rho_{b1} \quad p_{b2} \quad \rho_{b3}] \begin{bmatrix} b_1 \times b_1 & b_1 \times b_2 & b_1 \times b_3 \\ b_2 \times b_1 & b_2 \times b_2 & b_2 \times b_3 \\ b_3 \times b_1 & b_3 \times b_2 & b_3 \times b_3 \end{bmatrix} \begin{bmatrix} r_{b1} \\ r_{b2} \\ r_{b3} \end{bmatrix} \tag{3.27}$$

$$\vec{\rho} \times \vec{r} = [\rho_{b1} \quad p_{b2} \quad \rho_{b3}] \begin{bmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{bmatrix} \begin{bmatrix} r_{b1} \\ r_{b2} \\ r_{b3} \end{bmatrix} \tag{3.28}$$

$$\vec{\rho} \times \vec{r} = -[b_1 \quad b_2 \quad b_3] \begin{bmatrix} 0 & \rho_{b3} & -p_{b2} \\ -\rho_{b3} & 0 & \rho_{b1} \\ p_{b2} & -\rho_{b1} & 0 \end{bmatrix} \begin{bmatrix} r_{b1} \\ r_{b2} \\ r_{b3} \end{bmatrix}, \rho^{\times} = \begin{bmatrix} 0 & \rho_{b3} & -p_{b2} \\ -\rho_{b3} & 0 & \rho_{b1} \\ p_{b2} & -\rho_{b1} & 0 \end{bmatrix} \tag{3.29}$$

Defining with respect to the body frame,

$$\vec{\rho} \times \vec{r} = \rho^{\times} r = -r^{\times} \rho, \tag{3.30}$$

And the dot product defined as

$$(\vec{r}.\vec{\rho}) = \rho^T r \tag{3.31}$$

Substituting dot and cross products in the gravity gradient torque equations

$$\vec{T}_g = \frac{3\mu}{r^5} \int -r^{\times} \rho \rho^T r dm = -\frac{3\mu}{r^5} r^{\times} \int \rho \rho^T dm r \tag{3.32}$$

From the identity of dot and cross product,

$$\begin{gathered} \rho \times (\rho \times r) = \rho^{\times}(\rho^{\times} r) = (\rho^T r)\rho - (\rho^T \rho) r \\ \rho^{\times}(\rho^{\times} r) = (\rho \rho^T - (\rho^T \rho) I) r \\ \rho^{\times} \rho^{\times} = \rho \rho^T - (\rho^T \rho) I \\ \rho \rho^T = \rho^{\times} \rho^{\times} + (\rho^T \rho) I \end{gathered} \tag{3.33}$$

Where "I" is the identity matrix. Substituting the above relation in the final simplified gravity gradient torque will give

$$\vec{T}_g = -\frac{3\mu}{r^5} r^{\times} \int \rho^{\times} \rho^{\times} r dm - \frac{3\mu}{r^5} r^{\times} \int (\rho^T \rho) I dm r \tag{3.34}$$

Note that $\rho^T \rho$ is a scaler quantity and $r^{\times} \rho^T \rho r = \rho^T \rho r^{\times} r = 0$, the above equation further simplified as

$$\vec{T}_g = \frac{3\mu}{r^5} r^{\times} \int (-\rho^{\times}\rho^{\times}) dm r \tag{3.35}$$

Form the definition the moment of inertia matrix, 'I' is given as

$$I = \int (-\rho^{\times}\rho^{\times}) dm \tag{3.36}$$

Therefore, the gravity-gradient torque is

$$\vec{T}_g = \frac{3\mu}{r^5} r^{\times} I r \tag{3.37}$$

3.5 Attitude Dynamics

The dynamics of the CubeSat is modeled as a rigid body dynamic exposed to external disturbance forces. The attitude dynamics of the CubeSat is modeled using reference [2, 3, 44, 45].

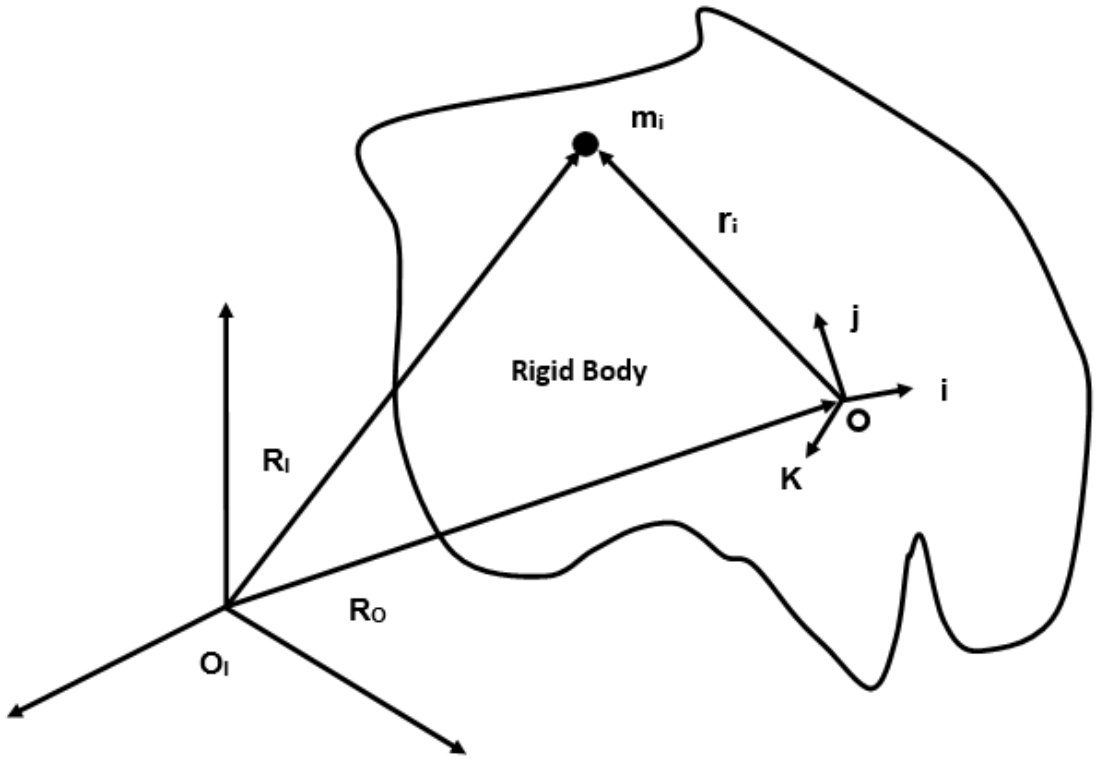


Figure 3.3 Position Vector Representation with respect to Inertial Frame of Reference

$$R_I = R_O + r_i \tag{3.38}$$

$$\dot{R}_I = \dot{R_O} + \dot{r_i} + \omega \times r_i = V_o + V_i + \omega \times r_i \tag{3.39}$$

Since the CubeSat is considered to be a rigid body, the distance between the particles is constant. Angular momentum of a particle in the system will be

$$\dot{r_i} = V_i = 0 \tag{3.40}$$

$$h_i = -V_o \times m_i r_i + r_i \times m_i(\omega \times r_i) \tag{3.41}$$

The total angular momentum is

$$h = \sum_{m_i} -v_o \times m_i r_i + \sum_{m_i} r_i \times (\omega \times r_i)\, m_i \tag{3.42}$$

The sum of all mass moments of the particles of the CubeSat about the mass center of the CubeSat is zero. The total angular momentum will be simplified as follows:

$$h = \sum_{m_i} r_i \times (\omega \times r_i)\, m_i \tag{3.43}$$

$$\omega = [\omega_x \quad \omega_y \quad \omega_z]\,, r_i = [x_i \quad y_i \quad z_i] \tag{3.44}$$

Substituting the r and ω in terms of vector expression,

$$\begin{aligned} h = i\left[\omega_x \sum ({y_i}^2 + {z_i}^2)\, m_i - \omega_y \sum y_i x_i m_i - \omega_z \sum x_i z_i m_i\right] \\ + j\left[\omega_y \sum ({x_i}^2 + {z_i}^2) m_i - \omega_x \sum y_i z_i m_i - \omega_z \sum y_i z_i m_i\right] \\ + k\left[w_z \sum ({x_i}^2 + {y_i}^2) m_i - \omega_x \sum x_i z_i m_i - \omega_y \sum y_i z_i m_i\right] \end{aligned} \tag{3.45}$$

$$I_{xx} = \sum_{m_i} ({y_i}^2 + {z_i}^2) m_i, \;\; I_{yy} = \sum_{m_i} ({x_i}^2 + {z_i}^2) m_i, \;\; I_{zz} = \sum_{m_i} ({x_i}^2 + {y_i}^2) m_i, \tag{3.46}$$

$$I_{yz} = I_{zy} = \sum_{m_i} (y_i z_i) m_i, \quad I_{zx} = I_{xz} = \sum_{m_i} (x_i z_i) m_i, \qquad I_{xy} = I_{yx} = \sum_{m_i} (y_i x_i) m_i, \tag{3.47}$$

$$h = \begin{bmatrix} I_{xx} & -I_{xy} & I_{xz} \\ -I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ w_z \end{bmatrix} = [I]\omega \,, \tag{3.48}$$

The new orientation of a CubeSat is due to the rotation of the CubeSat about the CubeSat reference frame. This reference frame is assumed to be aligned with the principal axis of the CubeSat; so that the CubeSat center of mass is the same as its center gravity. The moment of inertial will be the principal moment of inertia. The product of inertial becomes zero.

$$[I] = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.49}$$

Euler's law states that the total moment acting on a body about its center of mass equals the time rate of change of its total angular momentum.

$$H = h_i + h = h_i + [I]\omega \tag{3.50}$$

$$\dot{H} = T - \omega \times H = T - \omega \times (h_i + h) = T - \omega \times (h_i + [I]\omega) \tag{3.51}$$

$$\dot{H} = \dot{h}_i + [\dot{I}]\omega + \dot{\omega}[I] = T - \omega \times (h_i + [I]\omega) \tag{3.52}$$

$$\dot{\omega}[I] = \; T - \dot{h}_i - [\dot{I}]\omega - \omega \times h_l - w \times [I]\omega \tag{3.53}$$

Where:

T — External torque/Disturbance torques (like torques due to magnetic dipole moment, gravity gradient, aerodynamic drag, solar radiation pressure). These torques can be used for actuating purposes if not they are considered

as a disturbance in control design. Magnetorquers are used as an external actuator in CubeSat using the disturbance from Earth's magnetic field.

$\dot{h_l}$ — Internal torques due to onboard changes by rotating or displacing objects which can alter the spacecraft or satellite rotational velocity. These internal torques can be actuators like control moment gyros, reaction wheels, and momentum wheels or any disturbance torques in the satellite like liquid slosh or moving components. The reaction wheels are used as an internal actuator for CubeSat.

$[\dot{I}]\omega$ — Represents the change in moments of inertia of the satellite and spacecraft due to solar panel articulations, fuel usage, and onboard fluid slosh.

$\omega \times h_i$ — Gyroscopic torques onboard rotating components (how angular momentum of internally moving components change direction)

$w \times [I]\omega$ — Gyroscopic torques on satellite (how angular momentum of the whole satellite body change direction)

Since the structures of the CubeSat is thought to be rigid structure and there is no slosh or no solar panel articulation, the changes of moments of inertia are considered as zero.

$$[\dot{I}]\omega = 0 \tag{3.54}$$

The disturbance torque considered is the gravity gradient which is represented as T_g.

$$T = T_g \tag{3.55}$$

The internal torque considered is the reaction wheel symbolized as h_w.

$$\dot{h_l} = h_w = Ch_{ww} \tag{3.56}$$

The angular momentum of the reaction wheel about the reaction wheel axis is designated as h_{ww} while the angular momentum of the reaction wheel about the CubeSat body-fixed body frame axis is designated as h_w. C is three rows and 'n' number of the columns of the transformation matrix of

reaction wheel's angular momentum vector between CubeSat body-fixed body frame and the reaction wheel axis frame. Where 'n' is the number of reaction wheels in the given satellite under consideration. For this thesis, three reaction wheels that their rotation axis are along the CubeSat body-fixed body frame axis are used. Thus, the transformation matrix C is three by three identity matrix.

$$h_w = C h_{ww}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.57)$$
$$\dot{h_l} = h_w = h_{ww}$$

Angular velocity of the satellite, ω is considered to be the angular velocity of body-fixed body frame with respect to the geocentric inertial frame ω_b^i because the body-fixed body frame is attached to the satellite body.

$$\omega_b^i = \omega$$

Thus, the final dynamics equation will be

$$\omega_b^i = I^{-1}\left[T_g - \dot{h}_w - \omega_b^i \times h_w - \omega_b^i \times [I]\omega_b^i\right] \qquad (3.58)$$

Where,

T_g	Gravity gradient torque
$\dot{h}_w$	Main reaction torque (i.e. reaction wheel angular momentum rate)
$\omega \times h_w$	Gyroscopic torque effect on reaction wheel
$\omega \times [I]\omega$	Gyroscopic torque effect on satellite

3.6 State Space Representation of Attitude Kinematics and Dynamics

The state-space model of the system is based on the linearization of the perturbation of the kinematic and dynamics of attitude at the equilibrium points of the states. The vector parts of the quaternion and their rates are considered as states. These states are $q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3$.

The equilibrium point is taken by considering every axis of body-fixed body reference frame perfectly parallel to the corresponding axis of the body-centered orbital frames; therefore, the third axis of both frames of references are pointing to the earth center. Satellites that can achieve this

pointing usually named as nadir pointing satellites. The rotation matrix of the body-fixed body frame of nadir pointing satellites as seen from the body-centered orbital frame is the identity matrix. This infers the equilibrium point is

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.59}$$

The rotation matrix between the body-centered orbital frame and body-fixed body frame applies the quaternion rotation matrix and it will be linearized with the consideration of neglecting second-order perturbations which are multiplications between quaternion vectors and using equilibrium points [3].

$$C(q) = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix}$$

Thus,

$${q_1}^2 = {q_2}^2 = {q_3}^2 = q_1q_2 = q_1q_3 = q_2q_3 \approx 0,$$

$${q_0}^2 \approx 1,$$

$$q_0q_1 \approx q_1, \qquad q_0q_2 \approx q_2, \quad and\ q_0q_3 \approx q_3$$

The linearized transformation matrix will be

$$\begin{bmatrix} 1 & 2q_3 & -2q_2 \\ -2q_3 & 1 & 2q_1 \\ 2q_2 & -2q_1 & 1 \end{bmatrix} \tag{3.60}$$

Which can be expressed as

$$\hat{O}_1 = \begin{bmatrix} 1 \\ -2q_3 \\ 2q_2 \end{bmatrix}, \qquad \hat{O}_2 = \begin{bmatrix} 2q_3 \\ 1 \\ -2q_1 \end{bmatrix}, \qquad \hat{O}_3 = \begin{bmatrix} -2q_2 \\ 2q_1 \\ 1 \end{bmatrix} \tag{3.61}$$

Considering circular orbit, the orbital velocity seen from the body-centered orbital frame is

$$\begin{bmatrix} 0 \\ -\omega_c \\ 0 \end{bmatrix} \tag{3.62}$$

The angular rotation rate that body-centered orbital frame experience with respect to geocentric inertial frame as seen from body-fixed body frame will be

$$\omega_i^o = \begin{bmatrix} 1 & 2q_3 & -2q_2 \\ -2q_3 & 1 & 2q_1 \\ 2q_2 & -2q_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -\omega_c \\ 0 \end{bmatrix} \tag{3.63}$$

$$\omega_i^o = \begin{bmatrix} -2\omega_c q_3 \\ -\omega_c \\ 2\omega_c q_1 \end{bmatrix} \tag{3.64}$$

From the kinematics of the quaternion that relates body-fixed body frame to body-centered orbital frame is given as [3].

$$\omega_o^b = 2(q_0\dot{q} - \dot{q}_0 q) - 2[q\times]\dot{q} \tag{3.65}$$

Where, $[q\times]$ is a quaternion vector tilde operator.

$$[q\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{3.66}$$

Considering small quaternion approximation, $q_0 \approx 1$ and $\dot{q}_0 \approx 0$. Therefore,

$$\omega_o^b = 2\dot{q} \tag{3.67}$$

The angular velocity and angular acceleration of CubeSat can be expressed using quaternion as follows

$$\omega_i^b = \omega_o^b + \omega_i^o = 2\dot{q} + \begin{bmatrix} -2\omega_c q_3 \\ -\omega_c \\ 2\omega_c q_1 \end{bmatrix} = \begin{bmatrix} 2\dot{q}_1 - 2\omega_c q_3 \\ 2\dot{q}_2 - \omega_c \\ 2\dot{q}_3 + 2\omega_c q_1 \end{bmatrix} \tag{3.68}$$

$$\dot{\omega}_i^b = \begin{bmatrix} 2\ddot{q}_1 - 2\omega_c \dot{q}_3 \\ 2\ddot{q}_2 \\ 2\ddot{q}_3 + 2\omega_c \dot{q}_1 \end{bmatrix} \tag{3.69}$$

Where,

ω_i^b - angular velocity of body-fixed body frame with respect to the geocentric inertial frame

ω_o^b - angular velocity of body-fixed body frame with respect to the body-centered orbital frame

ω_i^o - angular velocity of the body-centered orbital frame with respect to the geocentric inertial frame

Now, attitude dynamics can be linearized by linearizing every mathematical term of the dynamics of the CubeSat attitude from equation 3.58.

$$\omega_b^i = I^{-1}\left[T_g - \dot{h}_w - \omega_b^i \times h_w - \omega_b^i \times [I]\omega_b^i\right]$$

Redefining a control torque as

$$T_c = \dot{h}_w - \omega_b^i \times h_w$$

The attitude dynamics will become

$$\omega_b^i = I^{-1}\left[T_g - T_c - \omega_b^i \times [I]\omega_b^i\right]$$

The linearization will be as follows:

- **Linearizing gravity gradient torque per mass inertia:**

  Gravity gradient torque from equation (3.37)

  $$\vec{T}_g = \frac{3\mu}{r^5} r^{\times} Ir \qquad (3.70)$$

  Vector "r" is located along the z-axis of the body-centered orbital frame as it is seen from the body-fixed body frame. Thus, the vector, r can be expressed using linearized transformation matrix expression between the body-centered orbital frame and the body-fixed body frame as follows

  $$\begin{bmatrix} 1 & 2q_3 & -2q_2 \\ -2q_3 & 1 & 2q_1 \\ 2q_2 & -2q_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} = r \begin{bmatrix} -2q_2 \\ 2q_1 \\ 1 \end{bmatrix} = r\hat{O}_3$$

  Gravity gradient torque is further simplified using equation (3.61) as

$$\vec{T}_g = \frac{3\mu}{r^5} r^2 \hat{O}_3{}^{\times} I \hat{O}_3 = \frac{3\mu}{r^3} \hat{O}_3{}^{\times} I \hat{O}_3$$

Considering circular orbit, the radius vector is constant as seen from the body-centered orbital frame. The orbital velocity is given as

$$\omega_c = \sqrt{\frac{\mu}{r^3}} \tag{3.71}$$

Therefore,

$$\vec{T}_g = {\omega_c}^2 [\hat{O}_3 \times][I]\hat{O}_3 \tag{3.72}$$

$$\hat{O}_3{}^{\times} = [\hat{O}_3 \times] = \begin{bmatrix} 0 & -1 & 2q_1 \\ 1 & 0 & 2q_2 \\ -2q_1 & -2q_2 & 0 \end{bmatrix} \tag{3.73}$$

Using an approximation of quaternions,

$$q_1 q_2 \approx 0$$

gravity gradient torque per mass inertia will be

$$I^{-1}\vec{T}_g = 3{\omega_c}^2 \begin{bmatrix} 2q_1 \dfrac{(I_{zz} - I_{yy})}{I_{xx}} \\ -2q_2 \dfrac{(I_{xx} - I_{zz})}{I_{yy}} \\ 0 \end{bmatrix} \tag{3.74}$$

- **Linearizing the gyroscopic torques on satellite per mass inertia**

$$-I^{-1}\,(\omega_b^i \times [I]\omega_b^i) = -I^{-1} \left(\begin{bmatrix} \omega_{i1}^b \\ \omega_{i2}^b \\ \omega_{i3}^b \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_{i1}^b \\ \omega_{i2}^b \\ \omega_{i3}^b \end{bmatrix} \right) \tag{3.75}$$

Substituting angular velocity of body-fixed body frame with respect to the geocentric inertial frame (equation (3.68))

$$\omega_i^b = \begin{bmatrix} \omega_{i1}^b \\ \omega_{i2}^b \\ \omega_{i3}^b \end{bmatrix} = \begin{bmatrix} 2\dot{q}_1 - 2\omega_c q_3 \\ 2\dot{q}_2 - \omega_c \\ 2\dot{q}_3 + 2\omega_c q_1 \end{bmatrix}$$

in the cross-coupling components of total torque (gyroscopic torques on satellite) and using the approximation of quaternions,

$$\dot{q}_2 q_3 = \dot{q}_2 \dot{q}_1 = \dot{q}_2 \dot{q}_3 = \dot{q}_2 q_1 = \dot{q}_3 \dot{q}_1 = \dot{q}_3 q_3 = q_1 \dot{q}_1 = q_1 q_3 \approx 0$$

$$-I^{-1}\left(\omega_b^i \times [I]\omega_b^i\right) = -\begin{bmatrix} \omega_{i3}^b \omega_{i2}^b \frac{(I_{zz}-I_{yy})}{I_{xx}} \\ \omega_{i3}^b \omega_{i1}^b \frac{(I_{xx}-I_{zz})}{I_{yy}} \\ \omega_{i1}^b \omega_{i2}^b \frac{(I_{yy}-I_{xx})}{I_{zz}} \end{bmatrix} = -\begin{bmatrix} \frac{I_{zz}-I_{yy}}{I_{xx}}(-2\omega_c \dot{q}_3 - 2{\omega_c}^2 q_1) \\ 0 \\ \frac{I_{yy}-I_{xx}}{I_{zz}}(-2\omega_c \dot{q}_1 + 2{\omega_c}^2 q_3) \end{bmatrix} \quad (3.76)$$

- **Linearizing Control Torque**

  Neglecting the gyroscopic effect of reaction wheel in the control torque because it is very small for CubeSat compared to the CubeSat reaction wheel torque, so $\omega \times h_w \approx 0$ [46].

$$\dot{h}_w \gg \omega_b^i \times h_w$$

$$T_c = \dot{h}_w - \omega_b^i \times h_w$$

  The control torque can be approximated as

$$T_c \approx \dot{h}_w$$

- **The Overall Linearization**

$$\omega_b^i = 3{\omega_c}^2 \begin{bmatrix} 2q_1 \frac{(I_{zz}-I_{yy})}{I_{xx}} \\ -2q_2 \frac{(I_{xx}-I_{zz})}{I_{yy}} \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{h_{w1}}{I_{xx}} \\ \frac{h_{w1}}{I_{yy}} \\ \frac{h_{w1}}{I_{zz}} \end{bmatrix} - \begin{bmatrix} \frac{I_{zz}-I_{yy}}{I_{xx}}(-2\omega_c \dot{q}_3 - 2{\omega_c}^2 q_1) \\ 0 \\ \frac{I_{yy}-I_{xx}}{I_{zz}}(-2\omega_c \dot{q}_1 + 2{\omega_c}^2 q_3) \end{bmatrix} \quad (3.77)$$

The final linearization after introducing new mass inertia and control torques representation for simplification and angular acceleration of body-fixed body frame with respect to the geocentric inertial frame from equation (3.69)

$$G_1 = \frac{(I_{zz}-I_{yy})}{I_{xx}}, \qquad G_2 = \frac{(I_{xx}-I_{zz})}{I_{yy}} \; and \; G_3 = \frac{I_{yy}-I_{xx}}{I_{zz}} \quad (3.78)$$

$$\dot{h}_{w1} = T_{c1}, \dot{h}_{w2} = T_{c2} \; and \; \dot{h}_{w3} = T_{c3}$$

$$\dot{\omega}_i^b = \begin{bmatrix} 2\ddot{q}_1 - 2\omega_c\dot{q}_3 \\ 2\ddot{q}_2 \\ 2\ddot{q}_3 + 2\omega_c\dot{q}_1 \end{bmatrix} \tag{3.79}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = 3{\omega_c}^2 \begin{bmatrix} q_1 G_1 \\ -q_2 G_2 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{T_{c1}}{2I_{xx}} \\ \frac{T_{c2}}{2I_{yy}} \\ \frac{T_{c3}}{2I_{zz}} \end{bmatrix} - \begin{bmatrix} G_1(-\omega_c\dot{q}_3 - {\omega_c}^2 q_1) \\ 0 \\ G_3(-\omega_c\dot{q}_1 + {\omega_c}^2 q_3) \end{bmatrix} + \begin{bmatrix} \omega_c\dot{q}_3 \\ 0 \\ -\omega_c\dot{q}_1 \end{bmatrix} \tag{3.80}$$

- **State Space Model**

$$\dot{x} = Ax + Bu$$

$$y = cx + Du$$

$$d/dt \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 4{\omega_c}^2 G_1 & 0 & 0 & 0 & 0 & \omega_c + \omega_c G_1 \\ 0 & -3{\omega_c}^2 G_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -{\omega_c}^2 G_3 & -\omega_c + \omega_c G_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1/2I_{xx} & 0 & 0 \\ 0 & -1/2I_{yy} & 0 \\ 0 & 0 & -1/2I_{zz} \end{bmatrix} \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \end{bmatrix} \tag{3.81}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \end{bmatrix} \tag{3.82}$$

Where,

$$\dot{x} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix}$$

$$x = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 4\omega_c{}^2 G_1 & 0 & 0 & 0 & 0 & \omega_c + \omega_c G_1 \\ 0 & -3\omega_c{}^2 G_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\omega_c{}^2 G_3 & -\omega_c + \omega_c G_3 & 0 & 0 \end{bmatrix}$$

$$\mathrm{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1/2I_{xx} & 0 & 0 \\ 0 & -1/2I_{yy} & 0 \\ 0 & 0 & -1/2I_{zz} \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$u = \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \end{bmatrix}$$

4 Overview of Fuzzy Logic, Neuro-Fuzzy and Genetic Algorithms

This section tries to address the highlight of the fuzzy logic, genetic algorithm, and neuro-fuzzy system which is the synergy of the fuzzy logic and neural network.

4.1 Fuzzy Logic

The common binary logic works for exact and clear expressions and data. Vague expressions, non-exact data classification, and intuitive decisions involved in engineering problems. Fuzzy logic and fuzzy sets can address this problem by applying modeling uncertain human reasoning and decision making [47]. The fuzzy system includes fuzzification, fuzzy rule base, fuzzy inference system, and defuzzification [48].

4.1.1 Fuzzification

Fuzzification is the way to transform crisp value to a fuzzy subset. The fuzzifications are specified by the membership functions of the fuzzy subsets. All crisp values are grouped into a single range and this range is the universe of discourse of the fuzzy set. Again, the fuzzy set is divided into fuzzy subsets. Each crisp value is the member of one or more fuzzy subsets with its corresponding level of membership value is in the interval of zero to one.

The membership functions (MF) defines to what level the elements are members of the fuzzy set. The selection of MF is subjective to the designer's requirement and the problem model. The generalized bell-shaped membership function is considered as a membership function in this thesis and it is parameterized by three parameters (a, b, c) [48]. Since it is sharp corners free and defined in three parameters, it gives more freedom to adjust it and ease differentiation steps in the learning method employed. The parameters a, b and c control width, slope, and the location of the center points of the generalized bell-shaped function, respectively.

$$bell(x; a, b, c) = \frac{1}{1 + \left(\frac{x-c}{a}\right)^{2b}} \tag{4.1}$$

4.1.2 Fuzzy Rule

A rule base is a form of "If premise, then action". The premise can be single or multiple antecedents that can be negated or related by logical operators. These operators are defined as T-norm or S-norm definitions [49]. In this thesis, the membership function (MF) of six states will be combined as the fuzzy rule pre-condition and the action at the consequent will be a linear function that takes a state as a variable.

4.1.3 Fuzzy Inference System

The fuzzy inference system will be used to logically infer the fuzzy rule base. Sugeno, Takagi, and Kang (TSK) fuzzy inference system is used as an inference system in this thesis. TSK has a form of

$$\text{If x is A and y is B then } Z = f(x, y) \tag{4.2}$$

Where, A and B are fuzzy sets in the antecedent while z=f (x, y) is a crisp function. The output of this model is a polynomial function parametrized with the antecedent inputs. The simplest output is a constant or a linear polynomial. The advantage of this model is it avoids the need for defuzzification and the rules can be generated from the data since the output involves parameters that can be modified from the learning experience.

4.1.4 Defuzzification

The outputs of most fuzzy inference systems are fuzzy output. These outputs have to be changed to crisp value to use them as a control signal. The fuzzy inference system used in this thesis is TSK that does not need any means of defuzzification.

4.2 Neuro-Fuzzy

The synergy of the neural network and fuzzy logic draws benefits from each method and tries to fill the gaps of one method by the other method. It combines the advantages of the fuzzy and artificial neural network (ANN) systems. The ANN has good adaptability through learning but with poor decision-making ability and fuzzy systems mimic human decision making but they cannot be able to cope with the changes in the system.

The combination of the ANN and fuzzy systems will advance the problem-solving capabilities of the two systems compare to stand-alone [50]. There are different types of the synergy of the neuro-

fuzzy system. One of the most common neuro-fuzzy systems is the adaptive neuro-fuzzy inference system (ANFIS) which is developed by Jang [51].

4.2.1 Adaptive Neuro-Fuzzy Inference System Architecture

The adaptive neuro-fuzzy inference system (ANFIS) is also known as the adaptive network-based fuzzy inference system (ANFIS). It is a network that functionally resembles the fuzzy inference system. ANFIS considered in this model represents the TSK model [51].

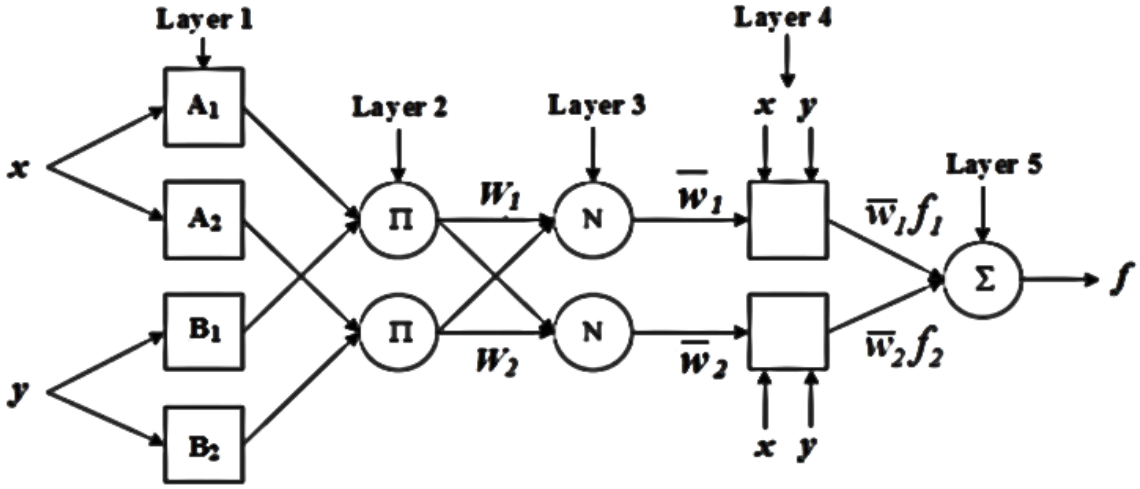


Figure 4.1 ANFIS Architecture

Assuming the TSK fuzzy model with a common rule set of two fuzzy rules,

$$\begin{aligned} &\text{If x is } A_1 \text{ and y is } B_1, \text{ then } f_1 = p_1x + q_1y + r_1. \\ &\text{If x is } A_2 \text{ and y is } B_2, \text{ then } f_2 = p_2x + q_2y + r_2. \end{aligned} \tag{4.3}$$

The ANFIS structure is composed of five layers. Each layer has its functionality. The number of layers can be reduced by performing two or more-layer functions in a single layer. The five-layer ANFIS model is considered in this thesis.

Layer 1: It is a combination of adaptive nodes. Each node is a fuzzy subset of a fuzzy set which is a membership function. From the figure above, A is a fuzzy set while A_1 and A_2 are the fuzzy subsets. Various membership functions can be used. The parameters of these functions are called premise parameters.

Layer 2: This layer is a fixed node labeled as II. The outputs of nodes are T-norm operation that performs fuzzy AND logic or intersection operation to the incoming signals. The outputs of each node represent the firing strength of a rule. The product of the incoming signals is one of the commonly used T-norm operations.

$$w_i = \mu_{A_i}(x)\mu_{B_i}(x), \qquad i = 1, 2 \tag{4.4}$$

Layer 3: The nodes in this layer are a fixed node labeled as N. Each node calculates the ratio of its firing strength to the sum of all rules' firing strengths. Thus, the output is the normalized firing strength.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \tag{4.5}$$

Layer 4: The node of this layer is an adaptive node with a node function. Considering linear TSK output, the node function is given by:

$$\bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{4.6}$$

Where, $\bar{w}_i$ is a normalized firing strength and the parameters (p_i, q_i, r_i) in this layer are consequent parameters**.**

Layer 5: This layer is a single fixed node labeled as Σ. The node executes the summation of all incoming singles.

$$overall\ output = \sum \bar{w}_i f_i = \frac{\sum w_i f_i}{\sum w_i} \tag{4.7}$$

4.2.2 ANFIS Learning Rules

ANFIS uses backpropagation and hybrid learning methods. The hybrid learning is a combination of backpropagation and least square methods. In solely the backpropagation learning method, the premise, and the consequent parameters will be updated till optimal values are gained using the back-propagation learning algorithm. This is inefficient in terms of time. The hybrid learning method uses two learning methods combining the backpropagation learning algorithm and the least square estimator. The least-square method will be used to develop optimal consequent parameters

through sequential forward updates while the backpropagation will be used to develop optimal premise parameters through the sequential back passes. Hybrid learning is employed as the ANFIS learning rule in this thesis.

A. Back Propagation Learning Algorithm

The backpropagation learning algorithm is gradient-descent optimization that minimizes the network prediction or classification ability error using feed-back error propagation through the connected networks. It is a supervised learning algorithm.

The backpropagation algorithm (BPA) is a commonly used learning rule in a multi-layer perceptron artificial neural network. It is based on the cost function called mean square error. This descriptive performance index cost function is

$$J = {}^{1}/_{2} \sum_{j=1}^{n} (r_j - y_j)^2 \tag{4.8}$$

Where, r_j : desired network outputs

y_j : the actual network outputs through activation Functions

Applying the gradient descent, the network's weight increment is proportional to the slope of the cost function which is called delta rule.

$$\Delta w_{ji} = -\varsigma_1 \frac{\partial J}{\partial w_{ji}} \tag{4.9}$$

$$\frac{\partial J}{\partial w_{ji}} = \frac{1}{n} \sum_{j=1}^{n} \frac{\partial}{\partial w_{ji}} (r_j - y_j)^2 \tag{4.10}$$

Where, ς_1 : constant proportional value

w_{ji} : weight values between j-layer and i^{th} neurons in multi-layer perceptron

Expanding the above equation using the chain rule

$$\frac{\partial J}{\partial w_{ji}} = \frac{2}{n}\sum_{j=1}^{n}(r_j - y_j)\frac{\partial y_j}{\partial w_{ji}} \tag{4.11}$$

The right side of the above equation can be expanded again using chain rule as

$$\frac{\partial y_j}{\partial w_{ji}} = \frac{\partial y_i}{\partial s_j}\frac{\partial s_j}{\partial w_{ji}} \tag{4.12}$$

For the matter of simple representation, the partial differentiation of activation function, y with respect to s_j is symbolized as

$$y_i{}' = \frac{\partial y_i}{\partial s_j} \tag{4.13}$$

The partial differentiation of activation function, y with respect to w_{ji}

$$\frac{\partial y_j}{\partial w_{ji}} = y_i{}'\frac{\partial s_j}{\partial w_{ji}} \tag{4.14}$$

The activation functions, y has an input s_j which is the summation of inputs to a given neuron

$$s_j = \sum_{j=1}^{m}(w_{ji}x_i + b_j) \tag{4.15}$$

b_j is a bias that can be considered as a w_{j0} weight with a unit input and s_j redefined as

$$s_j = \sum_{j=0}^{m} w_{ji}x_i \tag{4.16}$$

The partial derivative of the summation of inputs at any neuron with respect to the weight is

$$\frac{\partial s_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}}\sum_{j=0}^{m} w_{ji}x_i = \sum_{j=0}^{m}\frac{\partial w_{ji}}{\partial w_{ji}}x_i = x_i \tag{4.17}$$

Thus, the partial differentiation of activation function, y with respect to w_{ji}

$$\frac{\partial y_j}{\partial w_{ji}} = y_i{}'x_i \tag{4.18}$$

Combining the partial differentiation of activation function, y with respect to w_{ji} and the partial differentiation of cost function, J with respect to w_{ji}

$$\frac{\partial J}{\partial w_{ji}} = \frac{2}{n}\sum_{j=1}^{n}\left(r_j - y_j\right)y_i{}'x_i \tag{4.19}$$

$$\frac{\partial J}{\partial w_{ji}} = \frac{2}{n}\sum_{j=1}^{n}\delta_j x_i \tag{4.20}$$

Where, $\delta_j = \left(r_j - y_j\right)y_i{}'$

Using the above expression, and the weight increment delta rule

$$\Delta w_{ji} = -\varsigma_1\frac{\partial J}{\partial w_{ji}} = -\frac{2\varsigma_1}{n}\sum_{j=1}^{n}\delta_j x_i \tag{4.21}$$

$$\Delta w_{ji} = \varsigma\sum_{j=1}^{n}\delta_j x_i \tag{4.22}$$

Where, $\varsigma = -\frac{2\varsigma_1}{n}$ is learning rate which is between 0 and 1

The weight increment delta rule of each weight will be in discrete form as

$$\Delta w_{ji}(kT) = \varsigma\delta_j x_i \tag{4.23}$$

The new updated weight will be

$$\begin{gathered} w_{ji}(kT) = w_{ji}((k-1)T) + \Delta w_{ji}(kT) \\ w_{ji}(kT) = w_{ji}((k-1)T) + \varsigma\delta_j x_i \end{gathered} \tag{4.24}$$

The weight between the inner layers within the hidden layers will be the same and only the δ_j will be changed

$$\left[\delta_j\right]_l = \left[y_i{}'\right]_l\left[\sum_{1}^{m}\Delta w_{ji}\delta_j\right]_{l+1} \tag{4.25}$$

Thus, the l^{th} layer jth neuron δ_j is the multiplication of the partial derivative of l^{th} layer jth neuron activation function with respect to the summation of the input to jth neuron and multiplication of delta rule Δw_{ji} at $(l+1)^{th}$ layer and δ_j at $(l+1)^{th}$ layer.

B. Least Square Estimator

The least-square method addresses the problem of developing a generalized predictive or classifier expressions by putting its basis on minimizing the mean square error between measured available data results and the results of the developed least square estimator expression.

Assume a suggested model for raw input-output data

$$y = a_1 g_1(x) + a_2 g_2(x) + \cdots + a_n g_n(x) \tag{4.26}$$

Where,

$a_i, i = 1,2,3 \cdots n$: Regression Parameters

$g_i, i = 1,2,3 \cdots n$: Known Functions of x

x : Input Variables

y : Output Variable

Using known m numbers of input-output pairs

$$\begin{cases} y_1 = a_1 g_1(x_1) + a_2 g_2(x_1) + \cdots + a_n g_n(x_1) \\ y_2 = a_1 g_1(x_2) + a_2 g_2(x_2) + \cdots + a_n g_n(x_2) \\ \vdots \\ y_m = a_1 g_1(x_m) + a_2 g_2(x_m) + \cdots + a_n g_n(x_m) \end{cases} \tag{4.27}$$

And,

$$g_i^T = [g_1(x_1) \quad \cdots \quad a_n g_n]$$

Rewriting in a matrix

$$GA = y \tag{4.28}$$

Where,

$$G = \begin{bmatrix} g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & \vdots & \vdots \\ g_1(x_m) & \cdots & g_n(x_m) \end{bmatrix}$$

$$A = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

To find the unknown values of regression parameter values, simple analogical solution of the system of equations $like\ A = G^{-1}y$ could not be taken as a solution since the matrix G is m x n matrix and not square matrix which implies G is not invertible. However, the least-square estimator with the goals of minimizing the error between the estimated value and output raw data will be helpful to find the optimal values of the regression parameter values.

The error vector between the estimated value and output raw data, y

$$y - GA = e \tag{4.29}$$

The sum of squared error

$$\begin{aligned} E(A) = \sum_{i=1}^{m} (y_i - g_i^T A)^2 = e^T e = (y - GA)^T (y - GA) \\ = (y^T - A^T G^T)(y - GA) \\ = y^T y - y^T GA - A^T G^T y + A^T G^T GA \\ = y^T y - 2y^T GA + A^T G^T GA \end{aligned} \tag{4.30}$$

Since $y^T GA$ and $A^T G^T y$ are scaler quantities and they are taken as equivalent

$$-y^T GA - A^T G^T y = -2y^T GA$$

Find the optimal value of regression parameter A, differentiate $E(A)$ with respect to A.

$$E(A) = y^T y - 2y^T GA + A^T G^T GA$$

$$\frac{\partial E(A)}{\partial A} = 2G^T GA - 2G^T y \tag{4.31}$$

Equating $\frac{\partial E(A)}{\partial A}$ as zero

$$A = (G^T G)^{-1} G^T y \tag{4.32}$$

The multiplication $(G^T G)$ is considered to be non-singular. The optimal estimated generalized expression that fits the raw input-output data will be

$$GA = y, \tag{4.33}$$

4.3 Genetic Algorithms

Genetic algorithms are a derivative-free evolutionary optimization process that candidate solution population from a given search space are first randomly selected to be evaluated in the fitness function and they evolve to optimal solution through biologically inspired genetical operators like mutation and cross over [52].

The first initial population as a candidate solution is encoded as a binary, integer, or float gene and forms the chromosome solutions. These chromosomes are evaluated in the fitness function and scaled based on the values they scored in the fitness function. Genetic algorithm selection methods will select the fit candidate solutions using the scaling information. Fittest candidates will have a high scale and thus, it gets a better chance of selection. After the fittest members of the population of the first generation are selected using the fitness function, the next generations are reproduced from the chromosomes of the first-generation fit population's members. This reproduction is a genetic operation called crossover or recombination. Crossover mimics the biological recombination of two single chromosomes or haploids organisms. This process is followed by mutation and in some cases with inversion. A mutation is randomly changing the allele (usually 0 or 1 of binary genes) values in the chromosomes at some locations while inversions are the operation of reversing the order of the chromosomes at certain locations [53].

This process will be executed iteratively to minimize the given objective/fitness functions until the genetic algorithm terminates the optimization process as the pre-settled stopping criteria is fulfilled. Pre-settled stopping criteria are necessary since some optimization problems could be difficult to figure out the exact optimal value or they need a longer computational time to find an optimal solution. Finding the optimal solution is a tradeoff among the level of optimality, computational resources, and execution time; therefore, there should be a means to stop the execution to handle this tradeoff.

Also, the genetic algorithm stops execution depending on the stopping criteria and gives the best fit solution from all solutions within the time of its execution. There are different means of terminating genetic algorithm execution. Some of the common execution termination means are:

- **Maximum Number of Generation**
  The genetic algorithm stops when the maximum number of iterations is reached.
- **Time Limit**
  The maximum time the genetic algorithm runs before its execution stops.
- **Fitness Value Limit**
  The genetic algorithm stops when the best fitness value returned is below the fitness limit.
- **Number of Stall Generation**
  If fitness value average change over specified stall generations is below function tolerance, the genetic algorithm terminates.
- **Stall Time Limit**
  The best fit solution does not show any improvement within stall time, genetic algorithm iteration terminates.
- **Function Tolerance**
  The average change in fit solution within the stall generation is below the threshold level, the execution stops.

One of the drawbacks of genetic algorithm is best individuals in generations have a chance of being lost for the next generations because genetic operations like cross-over and mutation may lead to unfit individuals as compared to the individuals before the operations. Thus, there is a need to keep some of the best individuals for the next generation without letting them pass through genetic operators. This process is commonly known as Elitism. It guarantees the probability of losing the best individuals in the cross over and mutation. In elitism process, some percentages of the best individual in the given generation directly will become a member of the next generation.

5 Attitude Control

In this section, controller designing methods will be discussed. The dynamics of the plants are taken from the state space mathematical model of the attitude dynamics and kinematics. First, the LQR controller is discussed using constraint genetic algorithm optimization by considering actuator saturation. The second Takagi Sugeno Kang (TSK) based fuzzy logic controller is considered using the adaptive neuro-fuzzy inference system (ANFIS). ANFIS will use state errors with their corresponding control signals that are sampled from genetically tuned LQR simulation as training data. The predefined fuzzy input membership function parameters are tuned as well as the fuzzy rules and output functions in fuzzy rules are developed applying hybrid training combining backpropagation and least square methods in the ANFIS neuro-fuzzy system.

5.1 Linear Quadratic Regulator

Linear quadratic regulator (LQR) is a feedback controller that the states are controlled to settle to zero state values. The satellite attitude control using LQR in this thesis enables the identity rotational matrix of between the body-fixed body frame and the body-centered orbit frame of the CubeSat that all corresponding axis of the two reference frames are parallel. This attitude control enables nadir pointing attitude control by pointing the third axis of the satellite body-fixed body frame points to the Earth's center.

For the given state-space model,

$$\dot{x} = Ax + Bu \tag{5.1}$$

the linear quadratic regulator determines the "K" gain matrix and the control vector signal u

$$u = -Kx \tag{5.2}$$

that minimizes the performance index J.

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt \tag{5.3}$$

Where, R- is a positive definite matrix, and Q- is a positive definite matrix or semi-positive definite matrix that penalize the control energy and the states, respectively.

The gain K is given by

$$K = R^{-1}B^{T}P \tag{5.4}$$

Where, P- is a symmetric positive definite matrix and it is a solution of algebraic Riccati equation.

$$PA + A^{T}P + Q - PBR^{-1}B^{T}P = 0 \tag{5.5}$$

The controller designing process is an iterative selection of the Q and R matrixes until the desired performance is obtained. This iterative computation is tiresome and tedious. The genetic algorithm can be used for optimal selection of Q and R matrix based on the given constraint cost or objective function. This thesis applies a genetic algorithm for optimal LQR design.

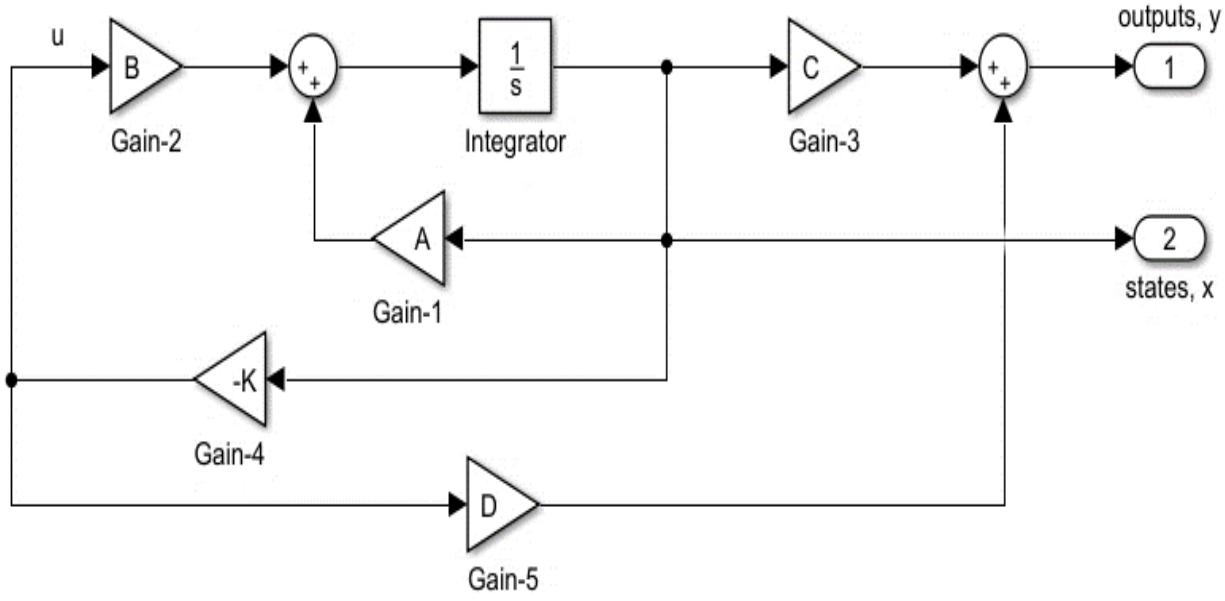


Figure 5.1 LQR Controller for General Linear system

5.1.1 Fitness Function

The fitness function is the core concept of genetic algorithm optimization. The optimization process tries to achieve the goal of the fitness or objective function. The genetic algorithm search for the values of the parameters that satisfy the fitness function within the given search space.

The goal of the LQR based attitude control of CubeSat in this thesis is to achieve a highly stable and fast settling attitude controller with unsaturated control energy. This goal can be used to develop a fitness function that the genetic algorithm will use for the optimization of LQR.

- **Fitness Objective Function-1**

The fitness function that achieves fast settling time and high stability considered. In the S-plane, this indicates the real part of the eigenvalues of the states must be located far from the origin to the left of S-plane. The larger the magnitude of the real part of the eigenvalues located on the left side of the S-plane the faster the system settles and the more stable system becomes. Also, the larger sum of all the magnitude of the real part of the eigenvalues located on the left side the faster the system settles and the more stable system becomes. It follows that the goal of the fitness function is to maximize this sum which is the same as minimizing the reciprocal of the sum. Thus, fitness function, f(x) is developed as follows:

$$f_1(x) = {}^{1}/_{\left(\sum_{i=1}^{n} E(i)\right)}$$
$$where: i = 1,2,3 \ldots 6 \ \ is\ idex\ to\ refere\ six\ eigen\ value \tag{5.6}$$
$$E = abs(real(eig(A - BK)))$$

Where, A is state matrix, B input matrix, and K gain matrix. The function ‘abs’ returns a positive value or magnitude of a given vector or scaler. The function ‘real’ returns the real part of the complex number. The function ‘eig’ returns the eigenvalues of a given matrix.

Although this fitness function favors fast settling and stable system, this goal needs larger actuator energy that causes actuator saturation and this will be addressed using the second objective function and constraint function.

- **Fitness Objective Function-2**

The torque in the LQR is input signal for states $[x_1, x_2, x_3, x_4, x_5, x_6] = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]$

$$torque = u = -Kx \tag{5.7}$$

Thus, the time-domain solution of the state is

$$x = e^{(A-BK)t}x_0 \tag{5.8}$$

The input is

$$u = -Ke^{(A-BK)t}x_0 \tag{5.9}$$

The input is the torque

$$torque = -Ke^{(A-BK)t}x_0 \tag{5.10}$$

It is assumed that the LQR will stabilize the system so that real part of the eigenvalue of $(A - BK)$ is located in the left side of S-plane, therefore; the t$orque = Ke^{(A-BK)t}x_0$ is maximum at zero seconds or the initial state.

$$torque_{max} = torque(0) = -Ke^{(A-BK)(0)}x_0 = -Kx_0 \tag{5.11}$$

To overcome the tradeoff between the need of better performance of the state using the maximum torque capability of the reaction wheel and the actuator saturation due to operation of the wheel at the maximum torque, operating value of the torque is considered as 50 % of the maximum torque for a given reaction wheel.

ADCOLE Maryland Aerospace company products, MAI-400 attitude determination, and control integrated tool kit specifications are considered and the MAI-400 reaction wheel's maximum torque is 0.635 mN.m [54]. To handle the tradeoff between performance and actuator saturation, 50% of the maximum torque was used as an operating torque point in the controller design which is approximated as 0.3 mN.m [54].

Choosing the maximum torque from the three individual reaction wheel's maximum torque

$$C_1 = Max(abs(torque_{max})) \tag{5.12}$$

Max- represents the maximum of the three-reaction wheel's maximum torques and abs- represents absolute value or magnitude of reaction wheel's maximum torques.

The second objective is based on the concept of avoiding any deviation from the reaction wheel's assumed operating torque value which is 0.3 mN.m. The square torque deviation from the given operating torque limit is the second objective function.

$$f_2(x) = (C_1 - (3 \times 10^{-4}))^2 \tag{5.13}$$

The MATLAB script code combining the two objectives function is available in appendix-I.

5.1.2 Constraint Function

Constraint function in any optimization avoids unrealistic or off-limit values during optimization. One of the constraint considerations is actuator saturation. The controller might generate signals that are not addressed by the actuator at hand.

This thesis considers constrained attitude torque in the genetically tuned LQR controller design. The maximum torque in the attitude dynamics is considered.

Taking ADCOLE Maryland Aerospace company products, MAI-400 attitude determination, and control integrated tool kit specifications, the maximum reaction wheel torque which is 0.635 mN.m is used as actuator torque limit to avoid actuator saturation. The constraint function will be

$$\begin{gathered} torque_{max} \leq torque\ limit \\ torque_{max} = -Kx_0 \leq 0.635\ \text{mN.m} \end{gathered} \tag{5.14}$$

The constraint function MATLAB script code used in the genetic algorithm optimization of the LQR controller is shown in the appendix (see appendix-II).

5.2 Fuzzy Logic Controller Design

Fuzzy logic mimics the intuitive and vague way of human uncertain reasoning and decision making to solve engineering problems. The fuzzy controller applies a fuzzy logic concept with control theory which gives a chance of easy controlling means using simple linguistics of "if and then rules".

The fuzzy logic controller in this thesis consists of fuzzification, fuzzy rule base, and TSK fuzzy inference system. ANFIS will develop the fuzzification, fuzzy rule base, the output functions in the TSK inference system from the training data.

5.2.1 Neuro-Fuzzy System

The neuro-fuzzy system considered in this thesis is the adaptive neuro-fuzzy controller system (ANFIS). ANFIS will be used to optimal adjustment of the input membership function parameter and develop a fuzzy rule base and TSK inference system. The consequents of the TSK inference system are linear functions that their parameters are modified through learning methods in ANFIS.

The adaptive neuro-fuzzy controller system (ANFIS) is used for optimal adjustment of the input membership function and output function parameters and to drive a fuzzy rule base. ANFIS commonly uses either a backpropagation or hybrid (backpropagation and least square estimator) learning methods. In the hybrid learning method, the least square learning will be used to develop optimal consequent parameters of ANFIS while the back-propagation learning rule will be used to develop optimal premise parameters of ANFIS. Hybrid learning is considered for this thesis.

Samples are taken from optimally tuned LQR controllers using a genetic algorithm. Three state inputs which are torque inputs of the system and the corresponding six states are sampled from the simulation of genetically tuned LQR controllers. The states and state inputs are considered as the inputs and the output of the training and testing data. These multiple outputs sampled training data will be organized into three distinct single output training data since ANFIS supports single output training. Distinct three groups training data given to ANFIS and it returns optimally tuned input membership function, output function, and fuzzy rule base for each group of training data. Thus, three different fuzzy systems will be developed from the final results. These three separated fuzzy systems become a fuzzy controller used to control the three reaction wheels.

5.2.2 Fuzzification

The fuzzy logic controller (FLC) for the CubeSat attitude control used in this thesis uses a state-based fuzzy logic controller. The fuzzification process tries to group each feed-back state to input membership functions. The universe of discourse of feedback states is the range of states' feed-back values.

The sampled data from genetically tuned LQR simulation of CubeSat attitude kinematics and dynamics plant are used to develop a membership function for each state using ANFIS. First grid partitioning is used to group each sampled state values into three general bell-shaped membership functions based on characters that the sampled data shows. The parameters of the ball-shaped

membership function will be further optimized using hybrid training in ANFIS such that fine-tuned bell-shaped input membership functions are developed at the end.

Each state is considered to have three input membership functions. Thus, each of the three fuzzy systems has 18 input membership functions that are developed for the six states.

Thus, every state's membership functions are considered to be the input membership function in the FLC. The bell-shaped membership function is selected because they have three parameters to express bell shape which makes it adaptive during learning and is free from sharp corners so that they are continuous and differentiable at any point.

5.2.3 Fuzzy Rule Base

The rules of the fuzzy logic controllers are based on if and then premise and consequence relations. The premise consists of all the six states' membership function (MF) combinations while the consequent consists of the action that will be applied if the premise is satisfied. For the six states that each state has three membership functions, there are three the power of six fuzzy rule bases which yields 729 rules. ANFIS will develop those rules. The three fuzzy systems that will be used to control three reaction wheels have their own 729 rules.

The form of the rules looks like

$$If\ q_1\ is\ MF1\ \ and\ \ q_2\ is\ MF1\ \ and\ \ q_3\ is\ MF1\ \ and\ \ \dot{q}_1 is\ MF1\ \ and\ \ \dot{q}_2\ is\ MF1\ \ and\ \ \dot{q}_3 is\ MF1$$

$$then\ output\ memberhsip\ function\_1$$

$$\vdots$$

$$\vdots$$

$$If\ q_1\ is\ MF3\ \ and\ \ q_2\ is\ MF3\ \ and\ \ q_3\ is\ MF3\ \ and\ \ \dot{q}_1 is\ MF3\ \ and\ \ \dot{q}_2\ is\ MF3\ \ and\ \ \dot{q}_3 is\ MF3$$

$$then\ output\ memberhsip\ function\_n$$

5.2.4 Fuzzy Inference System

The Takagi-Sugeno-Kang (TSK) inference system is applied to the FLC. The antecedents are combinations of state membership functions and the consequents are linear equations. The TSK inference system is dictated as follows

$$If\ q_1\ is\ MF1\ \ and\ \ q_2\ is\ MF1\ \ and\ \ q_3\ is\ MF1\ \ and\ \ \dot{q}_1 is\ MF1\ \ and\ \ \dot{q}_2\ is\ MF1\ \ and\ \ \dot{q}_3 is\ MF1$$

$$then\ z_1 = f_1(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3)$$

$$\vdots$$

$$\vdots$$

$$If\ q_1\ is\ MF3\ \ and\ \ q_2\ is\ MF3\ \ and\ \ q_3\ is\ MF3\ \ and\ \ \dot{q}_1 is\ MF3\ \ and\ \ \dot{q}_2\ is\ MF3\ \ and\ \ \dot{q}_3 is\ MF3$$

$$then\ z_1 = f_n(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3)$$

The consequents are a linear function that has a form as shown

$$f_i(x, y) = m_1 q_1 + m_2 q_2 + m_3 q_3 + m_4 \dot{q}_1 + m_5 \dot{q}_2 + m_6 \dot{q}_3 + b_1$$

Where: i=1, 2, 3… n

Each output function might be different from the other, therefore; there will be a maximum of 729 distinct functions that are the results of the TSK inference system.

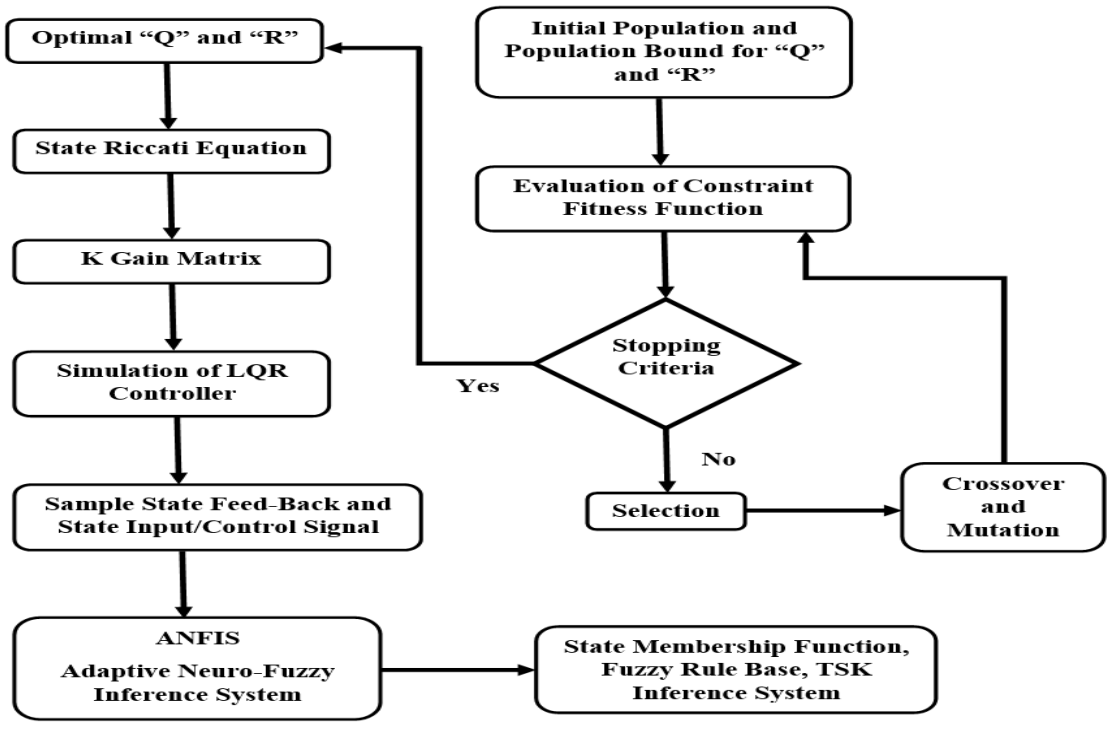


Figure 5.2 ANFIS Based FLC Design Flow Chart

5.2.5 Fuzzy Logic Controller

There are six states: the three quaternions vector components and their corresponding rates. Every state has 3 membership functions. As a result, 18 state membership functions are used in the single fuzzy logic controller as an input membership function. Also, three torques are control signals (state inputs) of the system which are output membership function in the fuzzy rule base and linear output function TSK inference system.

Three fuzzy systems developed from ANFIS. Each fuzzy system controls each reaction wheel in the CubeSat.

- The first fuzzy system is developed from ANFIS using the sampled data of six states and the first state input or control signal. It approximates the input state one of genetically tuned LQR and will control the reaction wheel one.
- The second fuzzy system is developed from ANFIS using the sampled data of six states and the second state input or control signal. It approximates the input state two of genetically tuned LQR and will be used to control the reaction wheel two.
- The third fuzzy system FLC is developed from ANFIS using the sampled data of six states and the third state input or control signal. It approximates the input state three of genetically tuned LQR and controls the reaction wheel three.

Finally, each developed three different fuzzy systems are the fuzzy logic controllers that can be used to control the CubeSat attitude.

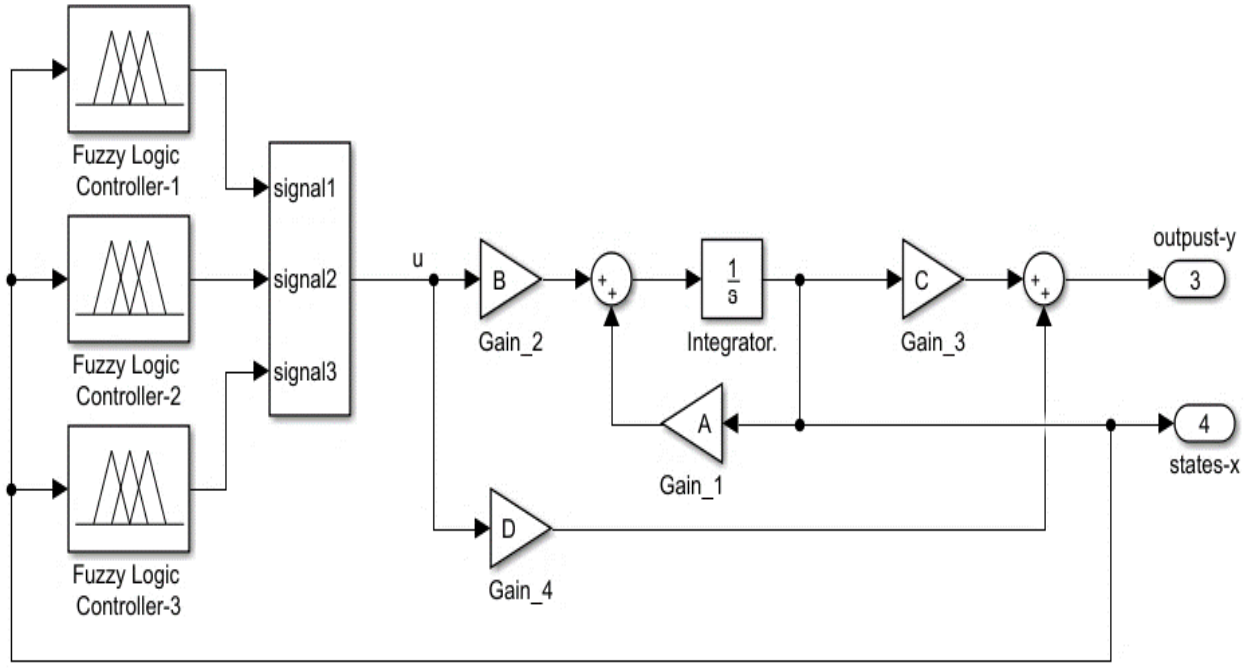


Figure 5.3 FLC Designed Using ANFIS

6 Attitude Control Simulation

In this section, the control methods developed in the previous chapter are simulated by applying them to the attitude mathematical model. The simulation results are presented using graphical results.

Matlab software is used for the simulation of the genetically tuned LQR and also FLC which is developed by ANFIS using training sampled data taken form the simulation of genetically tuned LQR controller. Matlab script codes, built-in tools, and applications, such as optimization toolbox for genetic algorithm optimization and Neuro-fuzzy designer app for ANFIS and SIMULINK for controller simulations are used.

6.1 Simulation Input Parameter

The simulation runs based on input parameters. These parameters include the mass inertia, orbit's orbital elements, and initial conditions of the states.

- Initial Conditions State, X_0

$$X_0 = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.5 \\ 0.3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Mass inertial Matrix in Kg.m^2

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} 0.0026 & 0 & 0 \\ 0 & 0.0024 & 0 \\ 0 & 0 & 0.0022 \end{bmatrix}$$

- Standard Values

No.	Standard Parameters	Standard Values
1	Universal Gravitational Constant (Km^3/s^2), μ	398,600
2	The radius of Earth (Km)	6378

Table 6.1 Scientific Standard Values

- Necessary Orbital Elements Orbit

No.	Parameters	Values
1	The radius of the Orbit (Km), a (considering CubeSat from 400 km from Earth's surface). (i.e. $\mathrm{a} = 400 + \mathrm{Raduis\ of\ Earth}$)	6778
2	The eccentricity of the Orbit, $e \approx 0$ (assuming circular orbit)	0
3	Orbital Inclination, i (Considering Deployment from international Space Station (ISS)).	51.6^0

Table 6.2 Orbit Parameters

6.2 Linear Quadratic Regulator

The linear quadratic regulator (LQR) control method stated in the previous chapter is used in this subsection and it is genetically optimized.

6.2.1 LQR Optimization Using Genetic Algorithm

The optimization is conducted by applying a genetic algorithm based on the objective and the constraint functions discussed in the previous chapter. The Q and R matrix are assumed to be a diagonal matrix with identical elements.

$$Q = \begin{bmatrix} q & 0 & 0 & 0 & 0 & 0 \\ 0 & q & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & 0 & 0 \\ 0 & 0 & 0 & q & 0 & 0 \\ 0 & 0 & 0 & 0 & q & 0 \\ 0 & 0 & 0 & 0 & 0 & q \end{bmatrix}, \quad R = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}$$

6.2.2 Genetic Algorithm Input Parameters

The genetic optimization to tune LQR tries to find optimal values of gain matrix K by searching for optimal state penalizing matrix, Q and state input penalizing matrix, R within the given searching space. The following table shows the input values and the techniques used in each type of genetic operators.

No.	Genetic Optimization Setups		Techniques and Values Selected
1.	Problem Setup	Fitness Function	See Appendix-I
		Number of variables	Two variables q and r
		Non-Linear Constraint	See Appendix-II
		Bounds	Lower Bound- and Upper Bound- (i.e. the bounds only consider the positive values for q and r to handle the conditions of positive definiteness of R matrix and positive definiteness matrix or semi-positive definiteness Q matrix). The bound assumed for q is [0 1000] and that of r is [0 1000].
2.	Population Size		50
3.	Creation Function		Constrained Dependent
4.	Initial population		The default values that are initialized from the creation Function.
5.	Initial Score		Default which means score will be specified by the score of the initial population using the fitness function.
6.	Initial Range		[0,20]
7	Fitness Scaling		Rank Scaling
8.	Selection		Tournament Selection with a tournament size 10
9.	Elite Count		0.05 times population size
10.	Reproduction through Crossover Fraction		0.8
11.	Mutation		Constrained Dependent based on Adaptive Feasible mutation
12.	Crossover		Constrained Dependent based on Scattered Crossover
13.	Migration		Forward direction
			0.2 fraction of the total population
			Migration Occurs in 20 generation intervals

14.	Pareto Front Population Fraction	Considering the default value 0.35	
15.	Stopping Criteria	Number of Generations	500
		Time Limit	Inf
		Fitness Limit	0.00001
		Stall Generations	450
		Stall Time Limit	Inf
		Function Tolerance	0.000001
		Constrained Tolerance	0.000001

Table 6.3 Genetic Algorithm Input Value and Their Description

6.2.3 Genetic Algorithm Optimization Results

Multi-objective Genetic algorithm optimization is applied using the optimization tool usually retrieved from the MATLAB command window by typing 'optimtool'. Taking the input considerations and values from the above table and applying appendix I and II MATLAB scripts code, the simulation results are obtained. The genetic algorithm results are discussed as follows:

- **Objective Function and Tuned Parameter Values**

20 different simulations are made and each simulation has been running over 500 generations to find the values of q and r within the bound of [0 1000]. The best value is selected combining objective function one and two.

Simulation Attempts	Objective Function-1	Objective Function-2	q	r	Maximum Torque
1	0.6333	5.29×10^{-09}	7.38×10^{-06}	19.0814	0.000373
2	0.6191	7.93×10^{-09}	5.65×10^{-06}	13.4153	0.00038903
3	0.6426	3.90×10^{-09}	6.51×10^{-06}	17.7811	0.000348
4	0.5061	7.08×10^{-08}	1.58×10^{-05}	17.7287	0.000566
5	0.526	5.16×10^{-08}	1.52×10^{-05}	19.7005	0.000527
6	0.5824	1.86×10^{-08}	6.65×10^{-06}	12.5522	0.000436
7	0.4839	9.88×10^{-08}	1.56×10^{-05}	14.8788	0.00061425
8	0.7102	7.08×10^{-17}	4.50×10^{-06}	17.9447	0.00029999
9	0.5436	3.84×10^{-08}	4.92×10^{-06}	7.1836	0.000496
10	0.6361	4.86×10^{-09}	1.55×10^{-05}	40.5938	0.00037
11	0.7102	1.99×10^{-16}	4.07×10^{-06}	16.2314	0.00029999
12	0.7109	3.28×10^{-13}	4.09×10^{-06}	16.398	0.000299
13	0.4763	1.10×10^{-07}	2.24×10^{-05}	20.1277	0.000632
14	0.7102	1.29×10^{-17}	2.56×10^{-06}	10.2185	0.0003
15	0.0195	0.0021	0.0742	12.4582	0.0464
16	0.5141	6.24×10^{-08}	1.17×10^{-05}	13.951	0.00055
17	0.7102	4.20×10^{-18}	4.08×10^{-06}	16.2816	0.0003
18	0.0253	0.0012	0.0645	18.9558	0.035
19	0.7098	9.63×10^{-14}	7.59×10^{-06}	30.2184	0.0003
20	0.5853	1.7496×10^{-08}	1.053×10^{-05}	20.2422	0.00043232

Table 6.4 Simulation Attempts

The 10 best optimal values using the objective function-1(the minimum values) are selected and again the best optimal values from these ten fittest values are selected using objective function-2 (the minimum value). Thus, simulation attempt 20 is selected as the optimal attempt. The following table states these optimal results.

No.	Parameters		Numerical Evaluation
1.	Fitness Functions Results	Objective Function-1	0.5853
		Objective Function-2	1.7496×10^{-8}
2.	q		0.1053×10^{-4}
3.	R		20.2422

Table 6.5 Genetic Algorithm Optimization Results

- **Pareto Plot**

The plot shows sample non-inferior values in the last generation which indicates two objective function value distributions. The last generation non-inferior values of the two fitness objectives function considered in the genetic algorithm optimization are shown in the plot below.

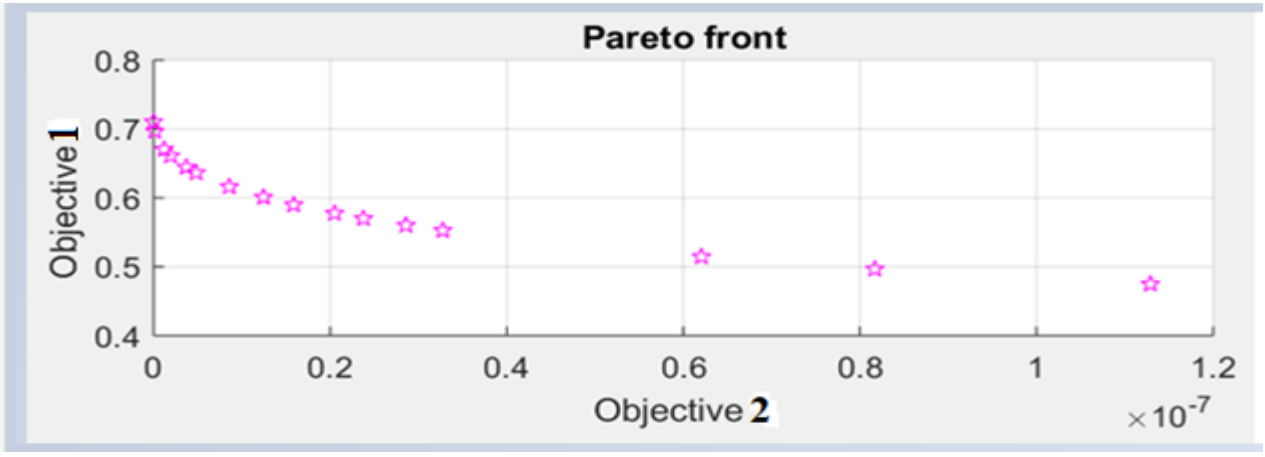


Figure 6.1 Pareto Plot between Fitness Function or Objective Function One and Two

- **The score of the Last Generation**

The score of the last generation evaluated in two fitness objective functions is:

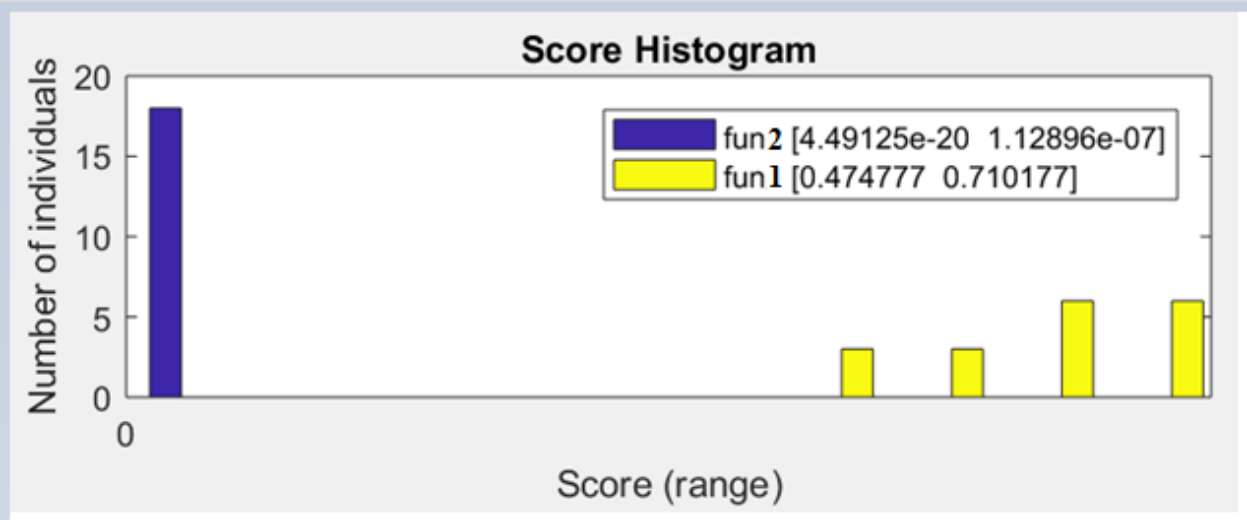


Figure 6.2 Score Range of Different Individuals in the Last Generation

The score plot of the individuals implies the last generation individuals' fitness function value ranges. The final optimal individuals' fitness values are within the ranges shown in the plot.

Stopping Criteria

Genetic algorithm execution terminated as the maximum number of generations is met with the one set on the stopping criteria.

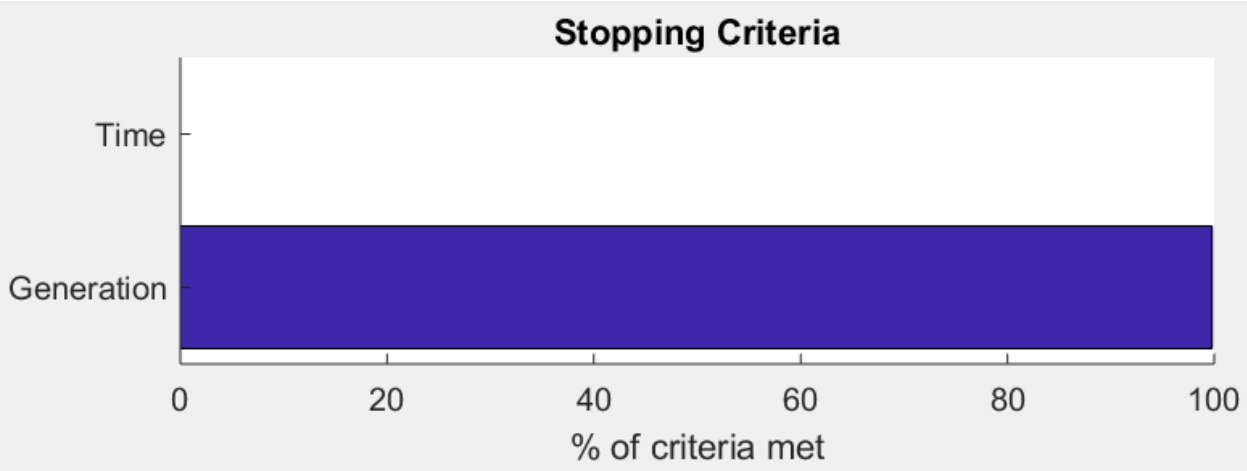


Figure 6.3 Percentage of Stopping Criteria Met

- **Average Pareto Spread**

The average values for the r and q parameters are indicated in the spread plot shown below. The plot indicates that the early generation average values of r and q values are around zero that makes

state and state input penalizing matrix small. But the average values of r and q values are changed over the upcoming generations that lead to the optimal values.

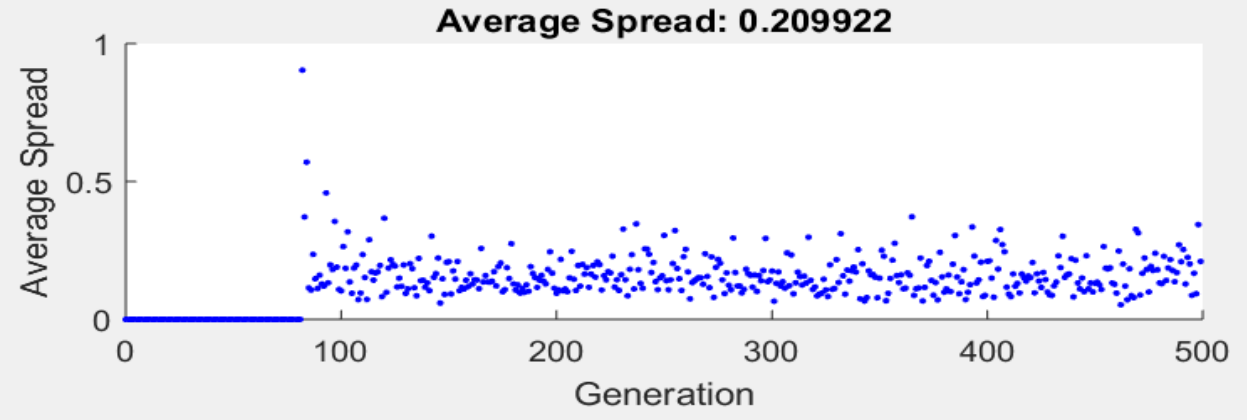


Figure 6.4 Average Pareto Spread over All Generations

- **Mutation and Cross-Over**

Mutation and cross-over across the whole generations are shown as

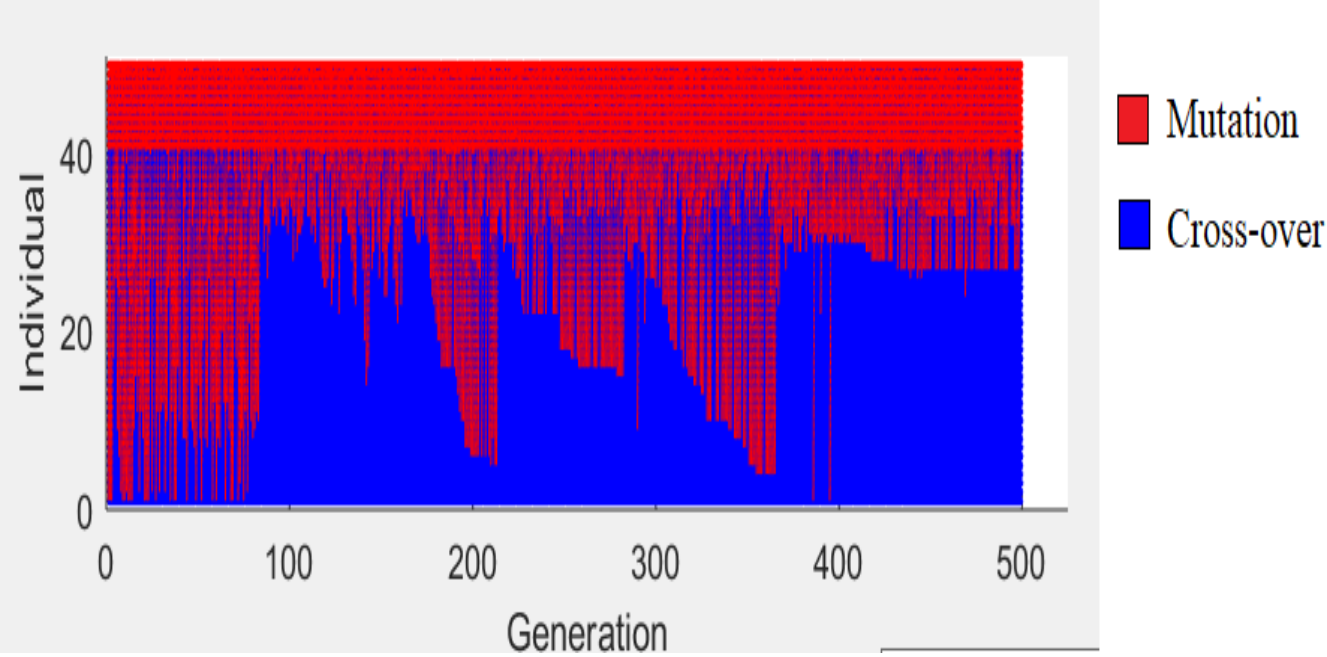


Figure 6.5 Mutation and Cross-Over over Generations

- **State Penalizing Matrix Q and The Input Penalizing Matrix R**

Finally, the values of the state penalizing matrix Q and the input penalizing matrix R are

$$Q = \begin{bmatrix} q & 0 & 0 & 0 & 0 & 0 \\ 0 & q & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & 0 & 0 \\ 0 & 0 & 0 & q & 0 & 0 \\ 0 & 0 & 0 & 0 & q & 0 \\ 0 & 0 & 0 & 0 & 0 & q \end{bmatrix} =$$

$$1 \times 10^{-5} \begin{bmatrix} 1.053 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.053 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.053 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.053 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.053 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.053 \end{bmatrix},$$

$$R = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} = \begin{bmatrix} 20.2422 & 0 & 0 \\ 0 & 20.2422 & 0 \\ 0 & 0 & 20.2422 \end{bmatrix}$$

A. Linear Quadratic Regulator Simulation

The values of the state penalizing matrix Q and the input penalizing matrix R have been found out using a genetic algorithm. Applying the 'lqr (A, B, Q, R)' command in MATLAB the gain matrix K is computed.

$$K = \begin{bmatrix} -7.2125 \times 10^{-4} & -3.8858 \times 10^{-20} & 1.4367 \times 10^{-6} & -2.8320 \times 10^{-3} & -2.0192 \times 10^{-19} & 4.1151 \times 10^{-12} \\ -3.1223 \times 10^{-20} & -7.2125 \times 10^{-4} & -2.6680 \times 10^{-19} & -2.1875 \times 10^{-19} & -2.7000 \times 10^{-3} & -5.4709 \times 10^{-19} \\ -1.4367 \times 10^{-6} & -1.3968 \times 10^{-19} & -7.2125 \times 10^{-04} & 4.8633 \times 10^{-12} & -5.9683 \times 10^{-19} & -2.600 \times 10^{-3} \end{bmatrix}$$

The simulation results of state and state inputs are shown in the following consecutive figures as states Vs time and state input Vs time plots.

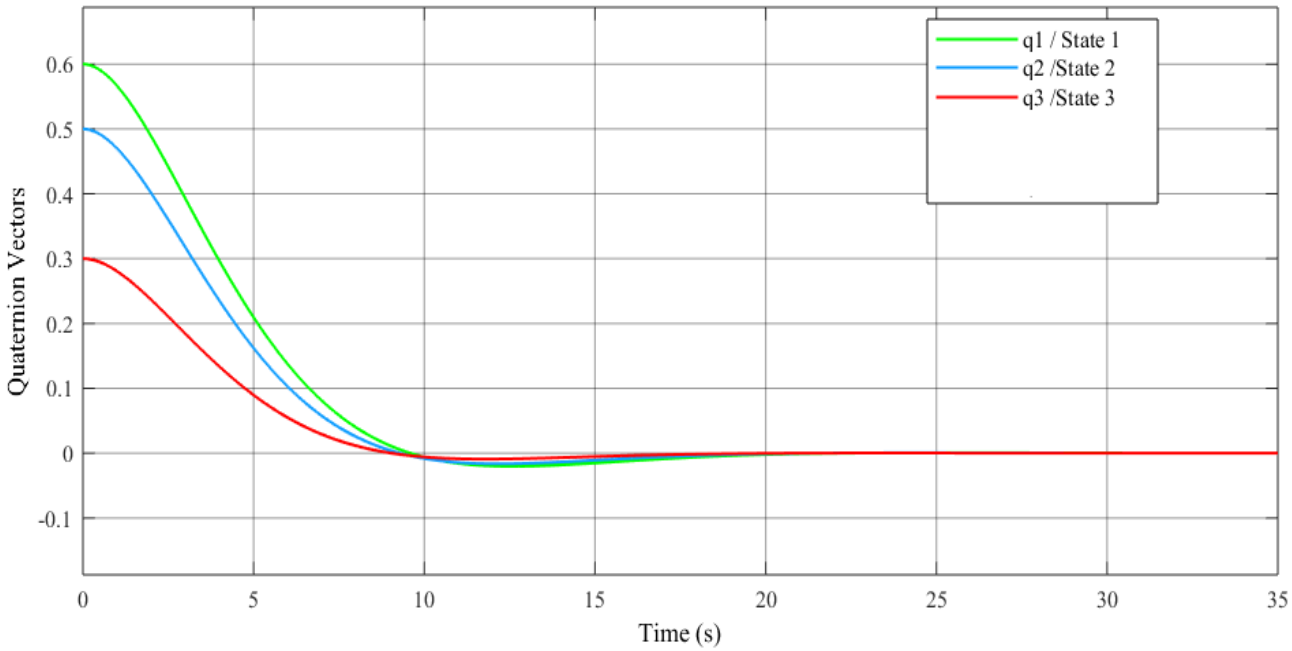


Figure 6.6 Simulation Plot of State One to State Three Vs Time.

The first three states (quaternion vector one, quaternion vector two, and quaternion vector three) plots over time are shown in the above plots. The three quaternion vectors diminish to zero within a finite time range and their steady-state errors are zero. They settle around 12 seconds. States one settles 12.171 seconds while state two and state three has a settling time 11.798 seconds and 11.440 seconds, respectively.

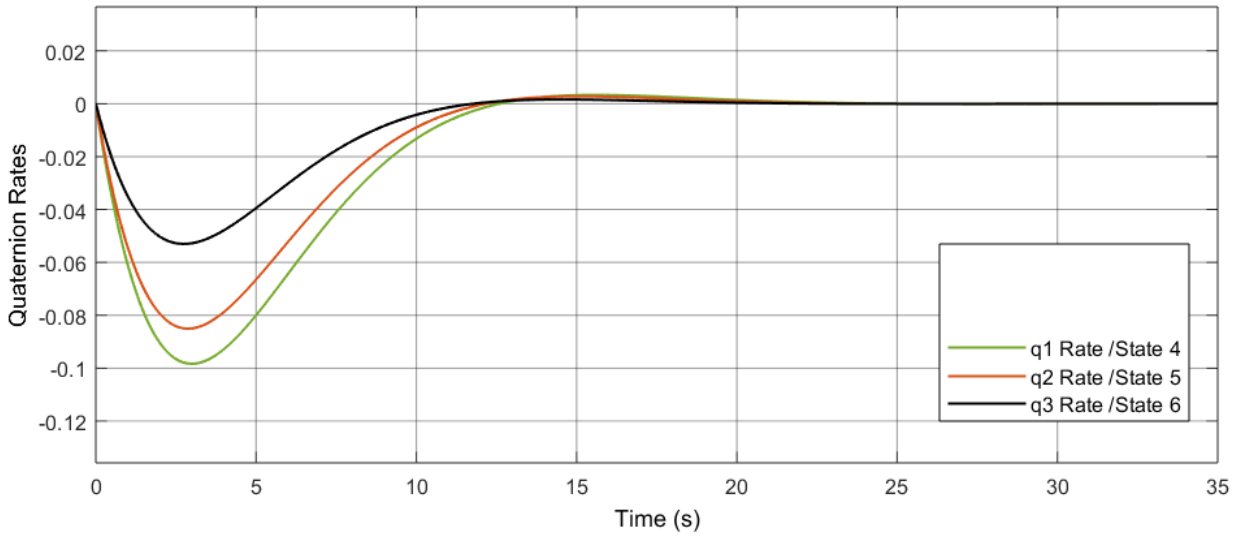


Figure 6.7 Simulation Plot of State Four to State Six Vs Time.

The states four, five, and six (three quaternion vector rates) plots in the above figure show that their steady-state errors are zero. State four settles in 12.167 seconds. State five has a settling time of 11.796 seconds and the last state settles in 11.430 seconds. The following table summarizes the steady-state and transient properties of the states.

States	**Rise Time (s)**	**Fall Time (s)**	**+Over shoot**	**-Under shoot**	**Settling time (s)**	**Steady-State error**
State one/q_1	-	5.925	-	3.646 %	12.171	0
State two/q_2	-	5.701	-	3.646 %	11.798	0
State three/q_3	-	5.462	-	3.646 %	11.440	0
State four/q_1 rate	5.924	1.719	3.646 %	0.521 %	12.167	0
State five/q_2 rate	5.700	1.650	3.646 %	0.521 %	11.796	0
State six/q_3 rate	5.460	1.572	3.646 %	0.521 %	11.430	0

Table 6.6 Transient and Steady-State Performance Measures for LRQ

The steady-state error for all states ($q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3$) are zero and the system regulates to zero states. Since the sum of the square of all quaternion elements ($q_0, q_1, q_2\ and q_3$) equals to one (see equation 3.2) and the values of the vector components of the quaternion (q_1, q_2, q_3) are zero at steady state, the scalar component of the quaternion (q_0) is one. These quaternion values develop the identity transformation matrix which indicates every axis of the body-fixed body reference frame lays over every corresponding axis of the body-centered orbit reference frame. Finally, the Z-axis of the body-fixed body reference frame points to the Earth's center and the CubeSat becomes a nadir pointing satellite.

The state inputs are reaction wheel torques. The peak values of these torques are 0.4323 x 10^{-3}, 0.3606 x $10^{-3}$and 0.2172 x 10^{-3} N.m. These values are constraint submissive that is less than the maximum torque limit of 0.635 mN.m in constraint function. The plots of the state inputs are shown below.

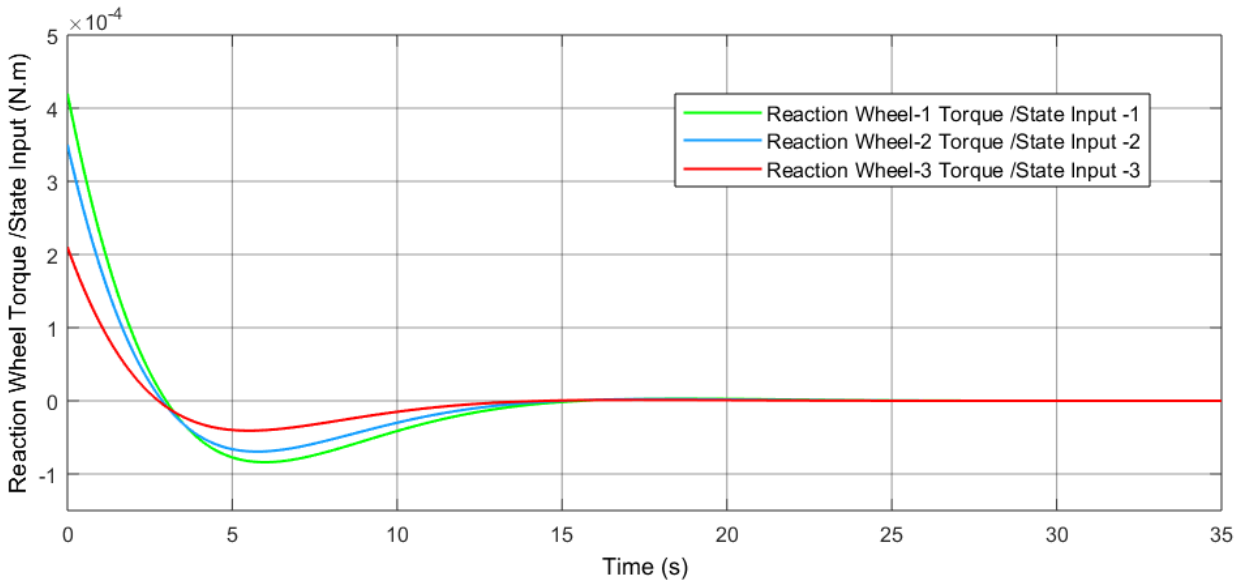


Figure 6.8 Simulation Plot of State Inputs Vs Time

6.3 Adaptive Neuro-Fuzzy Inference System Based Fuzzy Logic Controller

In this last subsection, the fuzzy logic controller (FLC) will be designed and applied to the attitude mathematical model. The FLC will be designed using the adaptive neuro-fuzzy inference system (ANFIS). The state and control signal values at specific sampling time are taken from genetically tuned LQR simulation. Sampled data for the training and testing purposes are separately prepared. The FLC design will be developed using ANFIS by applying hybrid learning that combines the backpropagation learning algorithm and the least square estimator based on the training data. The testing data will be conducted to check the reliability of the ANFIS hybrid training.

6.3.1 Adaptive Neuro-Fuzzy Inference System Input Parameter

The ANFIS is employed to develop well-tuned input membership from the given input membership and linear output membership function using training data. Also, the fuzzy rules combining those input and output membership functions are developed.

ANFIS uses hybrid learning. The hybrid learning method that uses two learning methods combining the backpropagation learning algorithm and the least square estimator.

MATLAB Neuro-Fuzzy Design Application Toolbox is used as an ANFIS platform to develop the fuzzy logic system. The following table shows the input for MATLAB Neuro-Fuzzy Design Application Toolbox.

No.	Input	Description
1.	Input membership function	***Three Generalized bell-shaped functions for each state:*** The generalized ball-shaped membership function is advantageous since it has three parameters to describe the function that makes it more adaptive for learning and it does not include any means of the sharp corner which ease differentiation steps in the learning methods.
2.	Output Membership Function	***Linear function:*** ANFIS uses TSK fuzzy inference system. The linear function is selected for the Output Membership Function
3.	Training Epoch	**6**
4.	Learning Rule	***Hybrid learning:*** It combines the backpropagation learning algorithm and the least square estimator. The least-square method will be used to develop optimal consequent parameters through sequential forward updates while the backpropagation will be used to develop optimal premise parameters through the sequential back pass.
5.	General Fuzzy Inference System fuzzification	***Grid Partitioning:*** It is helpful to cluster the data to different generalized bell-shaped membership functions which will be further tuned through training.
6.	Training data sampling	532 rows by 9 columns data are sampled at every 0.047 seconds from genetically tuned LQR 25 second simulation. The 9 columns are six state signals and three control signals or state input.
7.	Testing data sampling	109 rows by 9 columns data are sampled at every 0.23 second from genetically tuned LQR 25 second simulation.
8.	Training Data	See Appendix-III
9.	Testing Data	See Appendix-IV

Table 6.7 Inputs for Matlab Neuro-Fuzzy Designer

Since ANFIS does not support multiple output data instead it supports single output data, the training and testing data set that consists of six states as input, and three state inputs as output are classified into three class single output data groups. These three classes are:

- The first group training data is prepared from six states and the first state input by taking 1, 2, 3, 4, 5, 6, and 7 columns of the table available at appendix III and IV. This data will be used to develop a fuzzy system that mimics the first state input of genetically tuned LQR.
- The second group training data is prepared from six states and the second state input by taking 1, 2, 3, 4, 5, 6 and 8 columns of the table available at Appendix IV and V. This data will be used to develop a fuzzy system that mimics the second state input of genetically tuned LQR.
- The third group training data is prepared from six states and the third state input by taking 1, 2, 3, 4, 5, 6 and 9 columns of the table available at Appendix IV and V. This data will be used to develop a fuzzy system that approximates the third state input of genetically tuned LQR.

These three training data groups are separately given to MATLAB Neuro-Fuzzy Design Application Toolbox and three distinct fuzzy systems are developed form them.

6.3.2 ANFIS Result

MATLAB Neuro-Fuzzy Design Application Toolbox which is the ANFIS platform takes the inputs mentioned earlier and develops a fuzzy system that can be used in the fuzzy logic controller. The fuzzy systems developed from ANFIS are three fuzzy systems using the three classes of data described earlier. Each fuzzy logic system mimics the state inputs.

- Fuzzy Logic System one approximates state-input one of genetically tune LQR and controls reaction wheel one.
- Fuzzy Logic System two approximates state-input two of genetically tune LQR and controls reaction wheel two.
- Fuzzy Logic System three approximates state-input three of genetically tune LQR and controls reaction wheel three.

A. Training and Testing Errors

The training errors shows how much the fuzzy system developed from ANFIS mimics the state inputs of genetically tuned LQR for the given training data. The testing error shows the generalization ability of the developed fuzzy systems for state inputs values of genetically tuned LQR that are not included in the training.

Error	Fuzzy logic system One	Fuzzy logic system Two	Fuzzy logic system Three
Training Error	3.2951×10^{-8}	2.2418×10^{-8}	2.1895×10^{-8}
Testing Error	7.1294×10^{-8}	6.3248×10^{-8}	4.3193×10^{-8}

Table 6.8 Training and Testing Average Error

The fuzzy logic system one approximates the state input one of genetically tuned LQR with 3.2951×10^{-8} average error while the fuzzy logic system two mimics the state input two of genetically tuned LQR with 2.2418×10^{-8} average error. Also, the fuzzy logic system three approximates the state input three of genetically tuned LQR by average error deviation of 2.1895×10^{-8}.

The three fuzzy systems have relatively similar generalization error for values other than the training values taken from genetically tuned LQR.

B. Fuzzification and States' Membership Signals.

General bell-shaped is selected as the fuzzy input membership function since it has three parameters to describe the function that makes it more adaptive for learning and it does not include any means of sharp corners that ease differentiation steps in the learning methods. They are developed by fuzzifying the state feed-back based on the universe of discourse and the desired number of fuzzy membership functions for each state. Three input membership functions for each feed-back state are considered to handle the trade-off fuzzy generalization and rule bulkiness. Grid partitioning method available in ANFIS changes the crisp state feed-back data to different general bell-shaped membership functions which will be further tuned through backpropagation and least square estimator training methods available in the ANFIS. Three generalized bell-shaped membership functions are developed for each state. Thus, 18 input membership functions are developed for each fuzzy system that consists of six states as an input.

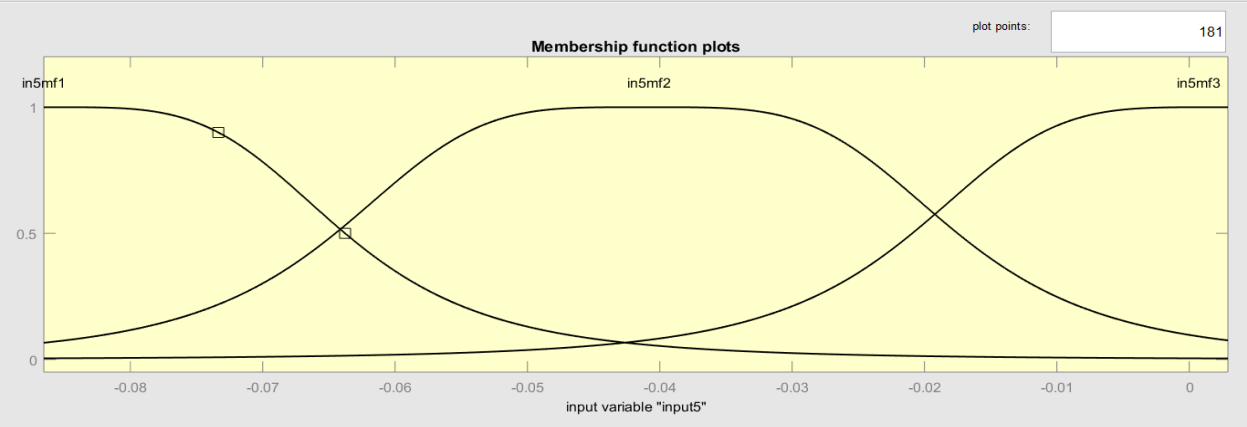


Figure 6.9 Generalized Bell-Shaped Input Membership Function

The parameters of input general bell-shaped input membership developed from ANFIS are shown in the following table.

No.	States	Membership Function (MF)	Parameters (a, b, c) for fuzzy system One	Parameters (a, b, c) for fuzzy system Two	Parameters (a, b, c) For fuzzy system Three
1.	q_1	q_1 MF one	(0.1552 2 -0.02078)	[0.155 0.593 -0.0209]	[0.1552 2 -0.02076]
		q_1 MF two	(0.1549 2 0.2895)	[0.154 2 0.141]	[0.1552 2 0.2896]
		q_1 MF three	(0.1552 2 0.6)	[0.155 2 0.6487]	[0.1552 2 0.6]
2.	q_2	q_2 MF one	(0.1292 2 -0.01693)	[0.1291 2 -0.01702]	[0.1292 2 -0.01691]
		q_2 MF two	(0.1294 2 0.2414)	[0.1296 2 0.2412]	[0.1292 2 0.2415]
		q_2 MF three	(0.1292 2 0.5)	[0.1293 2 0.4999]	[0.1292 2 0.5]
3.	q_3	q_3 MF one	(0.07743 2 -0.009795)	[0.07741 2 -0.00979]	[0.07743 2 -0.009762]
		q_3 MF two	(0.0786 2 0.1447)	[0.07956 2 0.146]	[0.07742 2 0.1452]
		q_3 MF three	(0.07745 2 0.3)	[0.07803 2 0.2997]	[0.07743 2 0.3]
4.	$\dot{q}_1$	$\dot{q}_1$ MF one	(0.02592 2 -0.1002)	[0.02497 2 -0.1014]	[0.02551 2 -0.1002]
		$\dot{q}_1$ MF two	(0.02676 2 -0.04934)	[0.03038 2 -0.04921]	[0.02422 2 -0.04996]
		$\dot{q}_1$ MF three	(0.02608 2 0.002248)	[0.02191 2 0.004907]	[0.02595 2 0.001763]
5.	$\dot{q}_2$	$\dot{q}_2$ MF one	(0.02284 2 -0.08652)	[0.02319 2 -0.08708]	[0.02266 2 -0.08644]
		$\dot{q}_2$ MF two	(0.0262 2 -0.03969)	[0.02837 2 -0.04434]	[0.02351 2 -0.04101]
		$\dot{q}_2$ MF three	(0.02444 2 0.0006489)	[0.02454 2 0.003517]	[0.02325 2 0.002331]
6.	$\dot{q}_3$	$\dot{q}_3$ MF one	(0.01476 2 -0.05308)	[0.01466 2 -0.05341]	[0.01495 2 -0.05359]
		$\dot{q}_3$ MF two	(0.01505 2 -0.02354)	[0.01932 2 -0.02125]	[0.02098 2 -0.02031]
		$\dot{q}_3$ MF three	(0.02055 2 0.0004801)	[0.02119 2 9.58e-05]	[0.01504 2 0.003171]

Table 6.9 Parameters of General Bell-Shaped Parameter Values

ANFIS also develops the universe of discourse of the state feed-back. The universe of discourse for the three fuzzy systems is the same since they use the same state feed-back.

No.	States feed-back	Universe of Discourse
1.	q_1(state one)	[-0.02076 0.6]
2.	q_2 (state two)	[-0.0169 0.5]
3.	q_3 (state three)	[-0.009759 0.3]
4.	$\dot{q}_1$ (state four)	[-0.09992 0.003457]
5.	$\dot{q}_2$(state five)	[-0.08654 0.002926]
6.	$\dot{q}_3$(state six)	[-0.05419 0.001765]

Table 6.10 Universe of Discourse of the Input Membership Functions

C. Output Membership Function

There are 729 output membership functions for each fuzzy system that are generated from ANFIS. The output membership functions are linear function with the six states ($q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3$) as input variables and seven parameters designated as p_n, n= 0,2,3…6. The six parameters are the coefficients to the six variables and the last parameter is a constant.

$$output\ membership\ function = p_1 q_1 + p_2 q_2 + p_3 q_3 + p_4 \dot{q}_1 + p_5 \dot{q}_2 + p_6 \dot{q}_3 + p_0$$

All 729 different membership functions are developed by changing the parameters of the above linear functions. The parameters for the sample membership function of the three fuzzy logic systems are available in appendix-V.

The universe of discourse of the Output membership function for the three fuzzy is shown in the table below.

No.	Output Membership Function	Universe of Discourse
1.	Output membership function of fuzzy system one	$[-8.6700 \times 10^{-5} \quad 4.3232 \times 10^{-4}]$
2.	Output membership function of fuzzy system two	$[-7.1900 \times 10^{-5} \quad 3.6062 \times 10^{-4}]$
3.	Output membership function of fuzzy system three	$[-4.2700 \times 10^{-5} \quad 2.1724 \times 10^{-4}]$

Table 6.11 Universe of Discourse of the Output Membership Functions

D. Fuzzy Rule Bases

Each fuzzy system has its own 729 rule bases. The rule bases are so bulky. Consequently, sample rule bases are shown in the figure below

1. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf1) (1)
2. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf2) (1)
3. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf3) then (output is out1mf3) (1)
4. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf4) (1)
5. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf5) (1)
6. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf3) then (output is out1mf6) (1)
7. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf3) and (input6 is in6mf1) then (output is out1mf7) (1)
8. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf3) and (input6 is in6mf2) then (output is out1mf8) (1)
9. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf3) and (input6 is in6mf3) then (output is out1mf9) (1)
10. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf1) then (output is out1mf10) (1)
11. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf2) then (output is out1mf11) (1)
12. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf3) then (output is out1mf12) (1)
13. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf1) then (output is out1mf13) (1)
14. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf2) then (output is out1mf14) (1)
15. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf2) and (input6 is in6mf3) then (output is out1mf15) (1)
16. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf3) and (input6 is in6mf1) then (output is out1mf16) (1)
17. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf3) and (input6 is in6mf2) then (output is out1mf17) (1)

Figure 6.10 Sample Fuzzy Rule Base

6.3.3 Fuzzy Logic Controller Simulation

Previously, the fuzzy system is developed using training data taken from genetically tuned LQR simulation. The training data is given to the ANFIS system and in return, three fuzzy systems that can separately control three reaction wheels are obtained. Thus, the fuzzy system one, two, and three control reaction wheel one, two, and three, respectively. In the overall process, the fuzzy control system that mimics LQR is developed. This is confirmed in the simulations shown below.

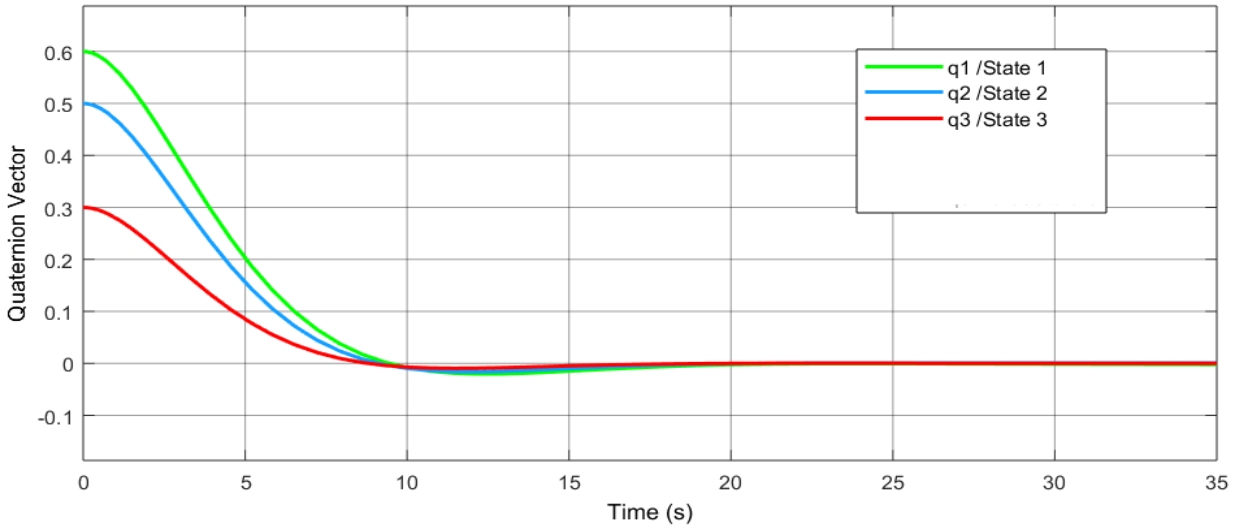


Figure 6.11 Simulation Plot of State One to State Three Vs Time.

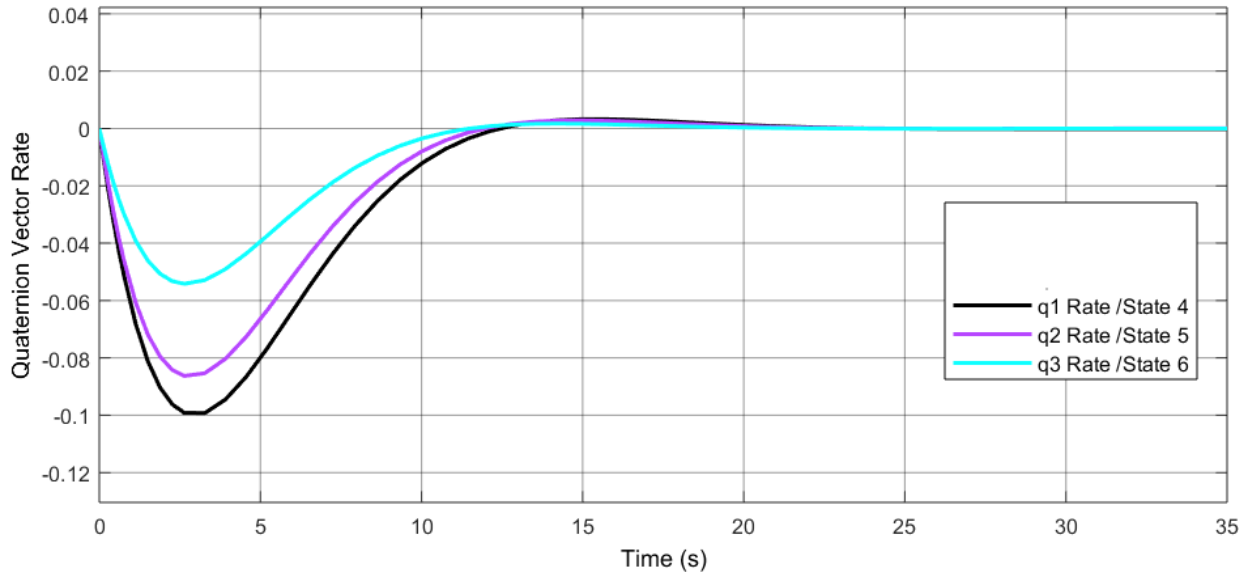


Figure 6.12 Simulation Plot of State Four to State Six Vs Time.

The six states (the three quaternion vectors and their corresponding rates) plots of the proposed FLC that mimics genetically tuned LQR is shown above. The FLC brings the states to zero steady-state error within finite time. The steady-state and transient response properties of simulation states are summarized in the table shown.

States	Performance Measures					
	Rise Time (s)	Fall Time (s)	Overshoot	Undershoot	Settling time (s)	Steady-State error
State one/q_1	-	5.938	-	3.646 %	12.445	0
State two/q_2	-	5.715	-	3.646 %	11.930	0
State three/q_3	-	5.478	-	3.646 %	11.595	0
State four/q_1 rate	5.921	1.698	3.646 %	0.469 %	12.339	0
State five/q_2 rate	5.713	1.651	3.646 %	0.521 %	12.012	0
State six/q_3 rate	5.466	1.590	3.646 %	0.521 %	11.637	0

Table 6.12 Transient and Steady-State Performance Measures for FLC

All the vector components of quaternion elements settle to zero state values. This indicates the transformation matrix between the body-fixed body reference frame and the body-centered orbit reference frame is the identity matrix. Thus, the Z-axis of the body-fixed body reference frame points to the Earth's center and the CubeSat becomes a nadir pointing satellite.

The fuzzy logic controllers that approximate the state inputs in the genetically tuned LQR obey the maximum torque constraint limit of 0.635 mN.m. The maximum control torques generated by the FLC controllers are 0.4323 mN.m, 0.3606 mN.m, and 0.2172 mN.m. The plots of the control signal developed by the three FLC are shown below.

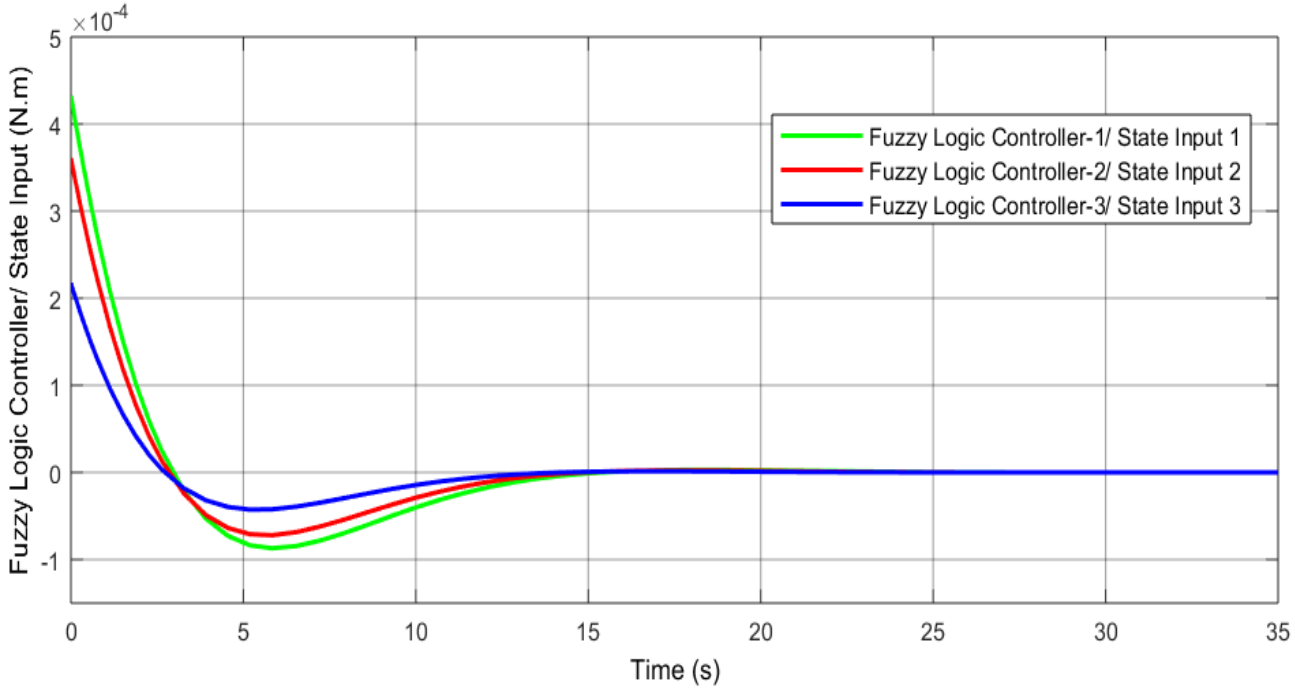


Figure 6.13 Simulation Plot of FLC Inputs Vs Time.

6.4 Computational Resources and Computational Time

The fifth-generation Dell laptop is used for both genetic algorithm LQR optimization and the three fuzzy system design using ANFIS. This laptop has 2.40 GHz Intel(R) Core (TM) i5-6200U CPU processor and 8.00 GB installed RAM. The optimized LQR and the three distinct fuzzy systems can be loaded to onboard computers in the satellite once they are developed. MATLAB R2017a version is used for ANFIS based fuzzy system development, genetic algorithm optimization, and simulation.

The tables shown below summarizes the computational time that the aforementioned laptop takes to develop the fuzzy systems using ANFIS and to optimize the LQR using genetic algorithm.

Computation	Simulation Attempts	Computation Time for Each Attempt	Total Computational Time for Total Attempts
Genetic Algorithm Optimization of LQR	20	24 minutes and 19 seconds	8 hours, 6 minutes and 20 seconds

Table 6.13 Genetic Algorithm Computational Time

Computation		Training Computational Time
Fuzzy System Design Using ANFIS	Fuzzy system-I	3 hours, 25 minutes and 8 seconds
	Fuzzy system-II	3 hours, 25 minutes and 8 seconds
	Fuzzy system-II	3 hours, 25 minutes and 8 seconds

Table 6.14 ANFIS Computational Time

Neuro-Fuzzy application, optimization toolbox, and Simulink environment available in the MATLAB are used for learning, optimization, and simulation.

Resources Used from MATLAB	Description
"Optimtool"	Optimization Tool Box
Neuro-fuzzy Designer	Application environment for ANFIS
Simulink	Graphical Simulation Environment

Table 6.15 Resources Used form MATLAB

6.5 Comparison between Genetically Tuned LQR and FLC Developed Using ANFIS

The LQR and the FLC controllers are compared using transient performance characteristics. Both controllers show very similar transient performance characteristics as shown in table 6.6 and table 6.12. The rise time of the quaternion rates of both LQR and FLC controllers are similar. Besides, the falling time of the quaternion vectors and their rates are almost identical both in LQR and FLC.

Moreover, relatively the same over-shoot and under-shoot are observed in both LQR and FLC. Also, the settling time of the quaternion and their rates are less than 12.2 seconds for LQR while the settling time of quaternion and the quaternion rates are less than 12.5 seconds for FLC.

Generally, both genetically tuned LQR and FLC developed using ANFIS have similar transient response characteristics with zero steady-state errors for each state. These similarities are expected from the simulations because the FLC is developed employing ANFIS that uses the sampled training and testing data taken from the simulation of genetically tuned LQR.

The simulation results obtained in this thesis show relatively similar results with others' literature. Some of these are

- The first literature is [55]. It investigates matrix gain PD controller to control aerodynamically disturbed single CubeSat. The attitude is represented using Euler angles. The system model actuated by tetrahedral configured 4 reaction wheels. The maximum control torque is below 2.5×10^{-3} N.m which is less than the maximum control torque in this thesis that uses three perpendicularly configured reaction torques. The plant settles within 0.003 orbits (it considers 1 orbit period as 5732 seconds**)** which is 17.196 seconds. Comparatively, the better settling time less than 12.5 seconds is achieved in this thesis.
- The second literature reference is [56]. It studies a 27-unit CubeSat actuated by a four-wheel pyramid reaction wheel array. The attitude is modeled using quaternion parametrization and the attitude controller used in this satellite design were PID and time-optimal controllers. The system settles less than 21 seconds for PID and less than 17 seconds for time-optimal control. Also, the better settling time is achieved in this thesis which is less than 12.5 seconds compared to PID and the time-optimal controllers.
- The third literature reference is [46]. The satellite is a 500 kg satellite that uses Euler angles (roll, pitch, and roll) as a parametrization. Though it is a large satellite compared to CubeSat, its attitude is controlled using LQR that the system settles less than 10.2 seconds and it can be compared with the LQR attitude control of a CubeSat. The attitude system settles 2 seconds faster than the system in this thesis.

7 Conclusion and Recommended Future Work

7.1 Conclusion

The problems described in the literature review and problem justification as the research gap were optimization problem for optimal LQR tuning and challenge of FLC design in state-space form for multiple input multiple output CubeSat attitude control due to lots of rules that integrates many membership functions.

The genetic algorithm to tune LQR is investigated as the solution for optimal LQR tuning. Two fitness functions and one constraint function are used for this optimization. The final results show that the values of the control torque are below the value of the actuator saturation level. A settling time less than 12.2 seconds and zero steady-state error is achieved within the system. Thus, LQR based reaction actuation can be used for pointing attitude control.

The ANFIS is considered as a solution for FLC design. The training and testing data that consists of 6 states and 3 state inputs sampled from genetically tuned LQR simulation at different sampling times are taken. This data is feed to ANFIS to develop the fuzzy system that can be difficult if it is done with expert knowledge and intuitive guess. The fuzzy logic controllers developed is quite good in approximating the genetically tuned LQR. This was demonstrated in the small testing and training average errors found in the ANFIS training and testing. The developed FLC has zero steady-state error and settling time less than 12.5 seconds. Therefore, this FLC control can be used for pointing attitude control, and also, it can replace the genetically tuned LQR.

7.2 Recommended Future Work

The thesis considers only reaction wheels as CubeSat control actuators with genetically tuned LQR and FLC that mimics genetically tuned LQR. The attitude control further could be expanded considering other means of optimizations like particle swarm optimization (POS) with magnetic actuators and combining magnetic actuator and reaction wheels.

References

[1] (2018, December 2). *Satellite* Available: https://en.m.wikipedia.org/wiki/Satellite

[2] S. R. Starin and J. Eterno, "Attitude Determination and Control Systems," 2011.

[3] K. L. Makovec, "A nonlinear magnetic controller for three-axis stability of nanosatellites," Virginia Tech, 2001.

[4] M. J. Sidi, "Orbit dynamics," in *Spacecraft dynamics and control: a practical engineering approach*, vol. 7: Cambridge Univ. Press, 2000.

[5] R. Sutherland, I. Kolmanovsky, and A. R. J. I. T. o. C. S. T. Girard, "Attitude Control of a 2U Cubesat by Magnetic and Air Drag Torques," 2018.

[6] I. Belokonov and I. J. P. E. Timbai, "The selection of the design parameters of the aerodynamically stabilized nanosatellite of the CubeSat standard," vol. 104, pp. 88-96, 2015.

[7] W. H. Steyn, V. J. A. S. Lappas, and Technology, "Cubesat solar sail 3-axis stabilization using panel translation and magnetic torquing," vol. 15, no. 6, pp. 476-485, 2011.

[8] H. S. Ousaloo, "Attitude control of a small satellite with uncertainly dynamic model using fuzzy logic strategy," in *The 2nd International Conference on Control, Instrumentation and Automation*, 2011, pp. 68-73: IEEE.

[9] T. Inamori, K. Otsuki, Y. Sugawara, P. Saisutjarit, and S. J. A. A. Nakasuka, "Three-axis attitude control by two-step rotations using only magnetic torquers in a low Earth orbit near the magnetic equator," vol. 128, pp. 696-706, 2016.

[10] R. Wisniewski and F. L. J. I. P. V. Markley, "Optimal magnetic attitude control," vol. 32, no. 2, pp. 7991-7996, 1999.

[11] R. J. J. o. G. Wisniewski, Control, and Dynamics, "Linear time-varying approach to satellite attitude control using only electromagnetic actuation," vol. 23, no. 4, pp. 640-647, 2000.

[12] R. Wisniewski, *Sliding mode attitude control for magnetic actuated satellite*. Department of Control Engineering, Aalborg University, 1998.

[13] R. Wisniewski, "Satellite attitude control using only electromagnetic actuation," Aalborg University. Department of Control Engineering, 1996.

[14] M. Abdelrahman, I. Chang, and S.-Y. J. I. J. o. N.-L. M. Park, "Magnetic torque attitude control of a satellite using the state-dependent Riccati equation technique," vol. 46, no. 5, pp. 758-771, 2011.

[15] M. Abdelrahman and S.-Y. J. A. A. Park, "Integrated attitude determination and control system via magnetic measurements and actuation," vol. 69, no. 3-4, pp. 168-185, 2011.

[16] M. Y. Ovchinnikov, D. Roldugin, V. Penkov, S. Tkachev, and Y. J. A. A. Mashtakov, "Fully magnetic sliding mode control for acquiring three-axis attitude," vol. 121, pp. 59-62, 2016.

[17] S. Shao, Q. Zong, B. Tian, and F. J. J. o. t. F. I. Wang, "Finite-time sliding mode attitude control for rigid spacecraft without angular velocity measurement," vol. 354, no. 12, pp. 4656-4674, 2017.

[18] F. J. A. A. Celani, "Robust three-axis attitude stabilization for inertial pointing spacecraft using magnetorquers," vol. 107, pp. 87-96, 2015.

[19] Y. Li, D. Ye, Z. J. A. S. Sun, and Technology, "Robust finite time control algorithm for satellite attitude control," vol. 68, pp. 46-57, 2017.

[20] Y. Bai, J. D. Biggs, X. Wang, and N. J. E. J. o. C. Cui, "A singular adaptive attitude control with active disturbance rejection," vol. 35, pp. 50-56, 2017.

[21] Q. Hu, Y. Shi, X. J. A. S. Shao, and Technology, "Adaptive fault-tolerant attitude control for satellite reorientation under input saturation," vol. 78, pp. 171-182, 2018.

[22] K. D. Kumar, N. Abreu, and M. J. A. A. Sinha, "Fault-tolerant attitude control of miniature satellites using reaction wheels," vol. 151, pp. 206-216, 2018.

[23] D. Ran, X. Chen, A. de Ruiter, and B. J. A. A. Xiao, "Adaptive extended-state observer-based fault tolerant attitude control for spacecraft with reaction wheels," vol. 145, pp. 501-514, 2018.

[24] D. Calvo, T. Avilés, V. Lapuerta, and A. J. E. S. w. A. Laverón-Simavilla, "Fuzzy attitude control for a nanosatellite in low Earth orbit," vol. 58, pp. 102-118, 2016.

[25] M. L. J. J. o. G. Psiaki, Control, and Dynamics, "Magnetic torquer attitude control via asymptotic periodic linear quadratic regulation," vol. 24, no. 2, pp. 386-394, 2001.

[26] M. Reyhanoglu and J. R. Hervas, "Three-axis magnetic attitude control algorithms for small satellites," in *Proceedings of 5th International Conference on Recent Advances in Space Technologies-RAST2011*, 2011, pp. 897-902: IEEE.

[27] M. Lovera, E. De Marchi, and S. J. I. T. o. C. S. T. Bittanti, "Periodic attitude control techniques for small satellites with magnetic actuators," vol. 10, no. 1, pp. 90-95, 2002.

[28] L. C. G. J. M. b. d. o. s. De Souza and machines, "Design of satellite control system using optimal nonlinear theory," vol. 34, no. 4, pp. 351-364, 2006.

[29] A. R. Walker, P. T. Putman, K. J. J. o. S. Cohen, and Rockets, "Solely magnetic genetic/fuzzy-attitude-control algorithm for a CubeSat," vol. 52, no. 6, pp. 1627-1639, 2015.

[30] S. Kukreti, A. Walker, P. Putman, and K. Cohen, "Genetic Algorithm Based LQR for Attitude Control of a Magnetically Actuated CubeSat," in *AIAA Infotech@ Aerospace*, 2015, p. 0886.

[31] C.-H. Cheng, S.-L. Shu, and P.-J. J. E. S. w. A. Cheng, "Attitude control of a satellite using fuzzy controllers," vol. 36, no. 3, pp. 6613-6620, 2009.

[32] A. Walker, P. Putman, and K. Cohen, "Fuzzy logic attitude control of a magnetically actuated cubesat," in *AIAA Infotech@ Aerospace (I@ A) Conference*, 2013, p. 5059.

[33] Y. M. Yassin, A. El Mahallawy, A. J. T. E. J. o. R. S. El-Sharkawi, and S. Science, "Real time prediction and correction of ADCS problems in LEO satellites using fuzzy logic," vol. 20, no. 1, pp. 11-19, 2017.

[34] A. Heydari, S. H. Pourtakdoust, and H. Heydari, "Magnetic attitude control using fuzzy logic," in *2009 IEEE Control Applications,(CCA) & Intelligent Control,(ISIC)*, 2009, pp. 456-460: IEEE.

[35] A. Skullestad, K. Olsen, S. Rennehvammen, and H. Fløystad, "Control of a gravity gradient stabilised satellite using fuzzy logic," 2001.

[36] M. Abdelrahman, S.-Y. J. A. S. Park, and Technology, "Spacecraft attitude control via a combined state-dependent Riccati equation and adaptive neuro-fuzzy approach," vol. 26, no. 1, pp. 16-28, 2013.

[37] S.-W. Kim, S.-Y. Park, and C. J. A. i. S. R. Park, "Spacecraft attitude control using neuro-fuzzy approximation of the optimal controllers," vol. 57, no. 1, pp. 137-152, 2016.

[38] W. M. Van Buijtenen, G. Schram, R. Babuska, and H. B. J. I. T. o. F. S. Verbruggen, "Adaptive fuzzy control of satellite attitude by reinforcement learning," vol. 6, no. 2, pp. 185-194, 1998.

[39] P. Guan, X.-J. Liu, and J.-Z. J. E. A. o. A. I. Liu, "Adaptive fuzzy sliding mode control for flexible satellite," vol. 18, no. 4, pp. 451-459, 2005.

[40] A. R. Fazlyab, F. F. Saberi, and M. J. A. i. S. R. Kabganian, "Adaptive attitude controller for a satellite based on neural network in the presence of unknown external disturbances and actuator faults," vol. 57, no. 1, pp. 367-377, 2016.

[41] J. L. Junkins and H. Schaub, *Analytical mechanics of space systems*. American Institute of Aeronautics and Astronautics, 2009.

[42] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. Springer, 2014.

[43] J. B. Kuipers, "Quaternions and Rotation Sequences."

[44] A. H. De Ruiter, C. Damaren, and J. R. Forbes, *Spacecraft dynamics and control: an introduction*. John Wiley & Sons, 2012.

[45] H. D. Curtis, *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013.

[46] C. U. Eze, C. C. Mbaocha, and J. O. Onojo, "Design of Linear Quadratic Regulator for the Three-Axis Attitude Control System Stabilization of Microsatellites."

[47] G. S. Sandhu and K. S. Rattan, "Design of a neuro-fuzzy controller," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 1997, vol. 4, pp. 3170-3175: IEEE.

[48] J.-S. Jang and C.-T. J. P. o. t. I. Sun, "Neuro-fuzzy modeling and control," vol. 83, no. 3, pp. 378-406, 1995.

[49] Y. Wu, B. Zhang, J. Lu, K. J. I. J. O. A. I. Du, and E. SYSTEMS, "Fuzzy Logic and Neuro-fuzzy Systems: A Systematic," p. 47.

[50] A. T. J. I. J. o. I. S. T. Azar and Applications, "Adaptive neuro-fuzzy system as a novel approach for predicting post-dialysis urea rebound," vol. 10, no. 3, pp. 302-330, 2011.

[51] J.-S. J. I. t. o. s. Jang, man, and cybernetics, "ANFIS: adaptive-network-based fuzzy inference system," vol. 23, no. 3, pp. 665-685, 1993.

[52] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[53] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.

[54] M. A. inc. (2019, may 25). *MAI-400 DataSheet*. Available: https://www.adcolemai.com/wp-content/uploads/2019/02/AMA-MAI-400-Datasheet.pdf

[55] A. M. Oluwatosin, Y. Hamam, and K. Djouani, "Attitude control of a CubeSat in a circular orbit using reaction wheels," in *2013 Africon*, 2013, pp. 1-8: IEEE.

[56] K. H. Gross, R. Patrick, E. Swenson, and J. S. Agte, "Optimal attitude control of a 6u cubesat with a four-wheel pyramid reaction wheel array and magnetic torque coils," in *AIAA Modeling and Simulation Technologies Conference*, 2016, p. 0174.

Appendix-I

Fitness Function

```
% This script code is GA optimization function is used in the MATLAB optimization toolbox,
  optimtool.
% Symbols used in the script
        % μ       - Universal gravitational constant (km^3/s^2)
        % I_XX      - Mass inertia in the CubeSat x-axis of body-fixed body frame (K g.m^2)
        % I_yy      - Mass inertia in the CubeSat y-axis of body-fixed body frame (Kg.m^2)
        % I_zz      - Mass inertia in the CubeSat z-axis of body-fixed body frame (Kg.m^2)
        % a         - Orbital radius (Km)
        % ω_c       - Orbital angular velocity (rad/s)
        % altitude  - Distance from the Earth's surface to the CubeSat center of gravity (Km)
        % Q         - State punishment matrix
        % R         - Input punishment matrix

function Z = GA_optimization_Objective_Function (x)
        μ =3 98600;
        raduis_of_earth = 6378;
        altitude = 400;
        a = raduis_of_earth + altitude;
        ω_c = sqrt (Ut/a^3);

% Initialization of mass inertia and variable declarations

        I_xx = 0.0026;
        I_yy = 0.0024;
        I_zz = 0.0022;
        G_1 = (I_zz-I_yy)/I_xx;
        G_2 = (I_xx-I_zz)/I_yy;
        G_3 = (I_yy-I_xx)/I_zz;
```

% State matrix 'A' and 'B' definitions

A= [0, 0, 0, 1, 0, 0; 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 1; (4*G_1*ω_c^2), 0, 0, 0, 0, (G_1*ω_c + ω_c); 0,
(-3*G_2*ω_c^2), 0, 0, 0, 0; 0, 0, (-G_3*ω_c^2), (G_3*ω_c - ω_c), 0, 0];

B= [0, 0, 0; 0, 0, 0; 0, 0, 0; (-1/2*I_{xx}), 0, 0; 0, (-1/2*I_{yy}), 0; 0, 0, (-1/(2*I_{zz})];

% Initial condition of the CubeSat

x_0 = [0.6;0.5;0.3;0;0;0];

% Input argument assigned to new variables and assigning diagonal matrix 'Q' & 'R'

q = x (1);
r = x (2);
Q = diag ([q q q q q q])
R = diag ([r r r])

% Calculating the 'k' gain matrix using the linear quadratic regulator command

K = lqr (A, B, Q, R)

% Calculating sum of the magnitude of the real part of Eigenvalue of the characteristic equation

e = eig(A-B*k)
f = real(e);
h = abs(f);
g = h (1,1) + h (2,1) + h (3,1) + h (4,1) + h (5,1) + h (6,1);

% The state, X can be expressed as $X = e^{(A-B*K)t}X_0$. State input, u which is a control torque will be defined as $u = torque = K * X = K*e^{(A-B*K)t}X_0$. For a stable system that eigenvalues of

A-B*K are negative, the input, u is maximum at a time, t=0. Thus, the maximum input or control torque is $u_{max} = Torque_{max} = K * X_0$ (see fitness function discussed in chapter five).

Torque$_{max}$ = -K*X$_0$

c_1 = max (abs (Torque$_{max}$));

% 50% of the maximum torque of the reaction wheel of ADCOLE Maryland Aerospace product, MAI-400 which is 0.3 mN.m is considered. This value is taken as operating point reaction wheel torque to avoid a possibility of actuator saturation (taken from specification of Maryland Aerospace product, MAI-400 Attitude determination, and control integrated toolkits). Torque square deviation from the given operating torque limit.

j= (c_1-(3*1.000e-04)) ^2;

% the objective function

Z (1) = (1/g);

Z (2) = j

end

Appendix-II

Constraint Function

```
% This script code is constraint function is used in the MATLAB optimization toolbox, optimtool.

% Symbols used in the script
        % μ          - Universal gravitational constant (km^3/s^2)
        % IXX        - Mass inertia in the CubeSat x-axis of body-fixed body frame (K g.m^2)
        % Iyy        - Mass inertia in the CubeSat y-axis of body-fixed body frame (Kg.m^2)
        % Izz        - Mass inertia in the CubeSat z-axis of body-fixed body frame (Kg.m^2)
        % a          - Orbital radius (Km)
        % ωc         - Orbital angular velocity (rad/s)
        % altitude   - Distance from the Earth's surface to the CubeSat center of gravity (Km)
        % Q          - State punishment matrix
        % R          - Input punishment matrix

function [c, c_eq] = Constraint_Function(x)

% universal gravitational constant (in km3/s2)

        Ut = 398600;
        raduis_of_earth = 6378;
        altitude = 400;
        a = raduis_of_earth + altitude;
        ωc = sqrt (Ut/a^3);

% Initialization of mass inertia and variable declarations

        I1 = 0.0026;
        I2 = 0.0024;
        I3 = 0.0022;
        G1 = (Izz-Iyy)/Ixx;
```

G_2 = (Ixx-Izz)/Iyy;

G_3 = (Iyy-Ixx)/Izz;

% State matrix 'A' and 'B' definitions

A= [0, 0, 0, 1, 0, 0; 0, 0, 0, 0, 1, 0; 0, 0, 0, 0, 0, 1; (4*G_1*ω_c^2), 0, 0, 0, 0, (G_1*ω_c + ω_c); 0, (-3*G_2*ω_c^2), 0, 0, 0, 0; 0, 0, (-G_3*ω_c^2), (G_3*ω_c - ω_c), 0, 0];

B= [0, 0, 0; 0, 0, 0; 0, 0, 0; (-1/2*I_{xx}), 0, 0; 0, (-1/2*I_{yy}), 0; 0, 0, (-1/(2*I_{zz})];

% Initial condition of the CubeSat

x_0 = [0.6;0.5;0.3;0;0;0];

% Input argument assigned to new variables and assigning diagonal matrix 'Q' & 'R'

q = x (1);

r = x (2);

Q = diag ([q q q q q q])

R = diag ([r r r])

% calculating the 'k' gain matrix using the linear quadratic regulator command

K= lqr (A, B, Q, R);

% The state, X expressed as $X = e^{(A-B*K)t}X_0$. State input, u which is a control torque will be defined as $u = torque = K * X = K*e^{(A-B*K)t}X_0$. For a stable system that eigenvalues of A-B*K are negative, the input, u is maximum at a time, t=0. Thus, the maximum input or control torque is $u_{max} = Torque_{max} = K * X_0$ (see the constraint function discussed in chapter five).

$Torque_{max}$ =K*X_0

c_1 = [$Torque_{max}$ (1); $Torque_{max}$ (2); $Torque_{max}$ (3)]';

```
        c_2    = max(abs(c_1));

% Considering maximum control torque, the maximum torque of the reaction wheel which is 0.
  635 mN.m to avoid a possibility of actuator saturation (taking the specification of Maryland
  Aerospace product, MAI-400 Attitude determination and control integrated toolkits).

        c      = c_2-(0.635*1.000e-03);
        c_eq   = [];
end
```

Appendix-III

Training Data

The training data is 532 rows by 9 columns data that is sampled at every 0.047 seconds from genetically tuned LQR 25 second simulation. The first 6 columns are six state signals which are inputs data to the ANFIS. The last three columns are state inputs that are considered to be the outputs data to ANFIS.

Since ANFIS only supports multiple output data, the training data is grouped into three class single output data groups for the application of training.

- The first group of training data is consisting of six states and the first state's input by taking 1, 2, 3, 4, 5, 6, and 7 columns of the training data table. The first 6 columns are input data and the seventh column is output data. It used to develop the fuzzy system that mimics LQR state-input one.
- The second group of training data is consisting of six states and the second state's input by taking 1, 2, 3, 4, 5, 6, and 8 columns of the training data table. The first 6 columns are input data and the eighth column is output data. It used to develop the fuzzy system that mimics LQR state-input two.
- The third group of training data is consisting of six states and the third state's input by taking 1, 2, 3, 4, 5, 6, and 9 columns of the training data table. The first 6 columns are input data and the ninth column is output data. It used to develop the fuzzy system that mimics LQR state-input three.

The training data in the following table is the sample of training data.

States						State Inputs		
1	2	3	4	5	6	1	2	3
0.4013	0.324781	0.188162	-0.09987	-0.08654	-0.05408	6.33E-06	-1.87E-06	-5.44E-06
0.49233	0.404277	0.238307	-0.0911	-0.08017	-0.05103	9.67E-05	7.28E-05	3.89E-05
0.20826	0.160591	0.087802	-0.08121	-0.06757	-0.04021	-7.99E-05	-6.85E-05	-4.18E-05
0.20446	0.157431	0.085923	-0.08049	-0.06689	-0.03976	-8.06E-05	-6.90E-05	-4.19E-05
0.55049	0.455715	0.271251	-0.0717	-0.06376	-0.0411	0.000194	0.000155	8.87E-05
0.00028	0.000442	0.000308	0.000431	0.000194	3.46E-05	1.43E-06	8.48E-07	3.14E-07
0.1359	0.101064	0.05285	-0.0652	-0.05295	-0.03061	-8.67E-05	-7.16E-05	-4.19E-05
0.15825	0.119295	0.063445	-0.07066	-0.05788	-0.0338	-8.61E-05	-7.19E-05	-4.26E-05
-0.01275	-0.00875	-0.00403	0.003444	0.002805	0.00155	5.63E-07	1.34E-06	1.13E-06
-0.01324	-0.00915	-0.00426	0.003455	0.002841	0.001586	2.43E-07	1.15E-06	1.07E-06
0.08665	0.061511	0.030297	-0.05133	-0.0407	-0.02284	-8.29E-05	-6.67E-05	-3.79E-05
0.19696	0.151207	0.082228	-0.07902	-0.06553	-0.03885	-8.19E-05	-6.97E-05	-4.22E-05
-0.00101	-0.00026	9.24E-05	0.00107	0.000635	0.000232	2.30E-06	1.55E-06	6.73E-07
-0.01163	-0.00785	-0.00354	0.003388	0.0027	0.001459	1.21E-06	1.71E-06	1.25E-06
0.2606	0.20444	0.114115	-0.08988	-0.07577	-0.04582	-6.68E-05	-5.93E-05	-3.74E-05
0.00071	0.000557	0.000271	3.07E-05	-4.35E-05	-4.98E-05	6.01E-07	2.83E-07	6.63E-08
-0.01674	-0.01215	-0.006	0.003208	0.002893	0.001758	-2.98E-06	-8.73E-07	2.55E-07
0.00069	0.00057	0.000291	8.51E-05	-1.48E-05	-4.18E-05	7.38E-07	3.71E-07	1.01E-07
-0.00091	-0.0002	0.000114	0.001029	0.000605	0.000218	2.26E-06	1.51E-06	6.51E-07
0.57188	0.474774	0.283569	-0.05773	-0.05157	-0.03343	0.000249	0.000202	0.000118
-0.0152	-0.01441	-0.00919	-0.0075	-0.00444	-0.00158	-3.22E-05	-2.25E-05	-1.08E-05
-0.00292	-0.00146	-0.00039	0.001751	0.001153	0.000495	2.85E-06	2.09E-06	1.01E-06
-0.02065	-0.01632	-0.00886	0.000745	0.00149	0.00132	-1.28E-05	-7.71E-06	-2.96E-06
0.36379	0.292377	0.16799	-0.09938	-0.08555	-0.05305	-1.93E-05	-2.25E-05	-1.73E-05
0.24806	0.193881	0.107741	-0.08802	-0.07399	-0.04458	-7.05E-05	-6.20E-05	-3.88E-05
-0.02033	-0.0169	-0.00964	-0.00172	-0.00019	0.000562	-1.95E-05	-1.27E-05	-5.51E-06
-0.00571	-0.00338	-0.00126	0.002499	0.00178	0.000846	2.96E-06	2.42E-06	1.30E-06
0.05057	0.033262	0.014687	-0.0392	-0.03024	-0.0164	-7.46E-05	-5.85E-05	-3.23E-05
0.01486	0.00631	0.000468	-0.02467	-0.01808	-0.00918	-5.92E-05	-4.48E-05	-2.37E-05
0.41536	0.336981	0.195798	-0.09954	-0.08646	-0.05419	1.74E-05	7.14E-06	-2.01E-07
-0.01969	-0.01673	-0.00975	-0.00285	-0.001	0.000174	-2.23E-05	-1.48E-05	-6.61E-06
0.00054	0.000543	0.000317	0.00025	8.03E-05	-9.59E-06	1.09E-06	6.11E-07	2.04E-07
-0.00106	-0.00029	8.13E-05	0.001091	0.00065	0.000239	2.32E-06	1.56E-06	6.84E-07
0.59858	0.498716	0.299158	-0.01484	-0.01338	-0.00877	0.000389	0.000323	0.000194
-0.00748	-0.00466	-0.00189	0.00286	0.002112	0.001048	2.71E-06	2.40E-06	1.38E-06
-0.00052	2.79E-05	0.000193	0.000854	0.00048	0.000159	2.05E-06	1.33E-06	5.56E-07
-0.00182	-0.00075	-9.62E-05	0.00138	0.000866	0.000345	2.60E-06	1.82E-06	8.33E-07
0.58405	0.485662	0.290639	-0.04553	-0.0408	-0.02654	0.000292	0.000239	0.000141
0.00043	0.000503	0.000317	0.000334	0.000132	1.01E-05	1.25E-06	7.24E-07	2.56E-07
0.03843	0.023957	0.00968	-0.03462	-0.02636	-0.01407	-7.04E-05	-5.47E-05	-2.98E-05
0.05243	0.034697	0.015466	-0.03988	-0.03082	-0.01675	-7.52E-05	-5.91E-05	-3.27E-05
0.00067	0.000571	0.000298	0.000113	5.63E-07	-3.70E-05	8.03E-07	4.14E-07	1.19E-07

Appendix-IV

Testing Data

Testing data is 109 rows by 9 columns data are that are sampled at every 0.23 second from genetically tuned LQR 25 second simulation. As the training data, testing data is grouped into three class data groups.

- The first group of testing data is consisting of six states and the first state's input by taking 1, 2, 3, 4, 5, 6, and 7 columns of the testing data table. It used to test the approximation ability of the fuzzy system that tries to mimic LQR state-input one.
- The second group of testing data is consisting of six states and the second state's input by taking 1, 2, 3, 4, 5, 6, and 8 columns of the testing data table. It used to test the approximation ability of the fuzzy system that tries to mimic LQR state-input two.
- The third group of testing data is consisting of six states and the third state's input by taking 1, 2, 3, 4, 5, 6, and 9 columns of the testing data table. It used to test the approximation ability of the fuzzy system that tries to mimic LQR state-input three.

The testing data shown in the following table is the sample of testing data.

States						State Inputs		
1	2	3	4	5	6	1	2	3
0.236344	0.184043	0.10182	-0.0862	-0.07225	-0.04338	-7.38E-05	-6.44E-05	-3.99E-05
-0.01014	-0.01125	-0.00794	-0.01125	-0.00732	-0.00311	-3.92E-05	-2.81E-05	-1.39E-05
0.069073	0.047643	0.022562	-0.04569	-0.0358	-0.0198	-7.96E-05	-6.33E-05	-3.55E-05
-0.01625	-0.01503	-0.00941	-0.00664	-0.00379	-0.00124	-3.05E-05	-2.12E-05	-1.00E-05
0.000375	0.000483	0.000316	0.000374	0.000157	1.95E-05	1.33E-06	7.77E-07	2.80E-07
0.000684	0.000574	0.000296	9.83E-05	-7.82E-06	-3.99E-05	7.71E-07	3.92E-07	1.10E-07
0.117806	0.086415	0.044416	-0.06044	-0.04871	-0.02789	-8.63E-05	-7.06E-05	-4.09E-05
-0.02068	-0.01634	-0.00887	0.000774	0.001512	0.001332	-1.27E-05	-7.66E-06	-2.94E-06
0.538795	0.445329	0.264568	-0.07725	-0.06854	-0.04406	0.000169	0.000134	7.61E-05
0.320927	0.255629	0.145313	-0.0968	-0.08266	-0.05076	-4.29E-05	-4.12E-05	-2.77E-05
-0.00042	8.02E-05	0.000211	0.000814	0.000451	0.000146	2.00E-06	1.29E-06	5.34E-07
-0.00044	-0.00468	-0.00496	-0.01706	-0.01191	-0.00565	-4.86E-05	-3.59E-05	-1.84E-05
0.013711	0.005458	2.90E-05	-0.02416	-0.01766	-0.00893	-5.85E-05	-4.43E-05	-2.34E-05
-0.00734	-0.00456	-0.00183	0.00284	0.002092	0.001035	2.75E-06	2.42E-06	1.38E-06
-0.00412	-0.00723	-0.00615	-0.01498	-0.01026	-0.00472	-4.54E-05	-3.32E-05	-1.68E-05
-0.00396	-0.00215	-0.00069	0.00206	0.001404	0.00063	2.98E-06	2.28E-06	1.14E-06
-0.01486	-0.01051	-0.00502	0.003432	0.002925	0.001691	-9.95E-07	4.01E-07	7.86E-07
0.591941	0.492746	0.295255	-0.03368	-0.03026	-0.01975	0.000331	0.000273	0.000162
-0.00735	-0.00941	-0.00714	-0.01305	-0.00873	-0.00388	-4.22E-05	-3.06E-05	-1.53E-05
-0.01407	-0.00984	-0.00464	0.003461	0.002896	0.001645	-3.39E-07	8.05E-07	9.43E-07
-0.0087	-0.00558	-0.00234	0.003069	0.002318	0.001182	2.42E-06	2.30E-06	1.39E-06
-0.01911	-0.01441	-0.00743	0.002472	0.002552	0.001716	-6.77E-06	-3.43E-06	-8.94E-07
5.10E-05	0.000332	0.000285	0.000571	0.000286	7.26E-05	1.65E-06	1.02E-06	3.96E-07
0.00072	0.000551	0.000264	1.10E-05	-5.37E-05	-5.25E-05	5.50E-07	2.51E-07	5.35E-08
0.104342	0.075598	0.038246	-0.05665	-0.04536	-0.02577	-8.52E-05	-6.92E-05	-3.98E-05
-0.01955	-0.0167	-0.00978	-0.00311	-0.00118	8.60E-05	-2.29E-05	-1.53E-05	-6.86E-06
0.049635	0.032531	0.014284	-0.03889	-0.02997	-0.01624	-7.44E-05	-5.83E-05	-3.22E-05
-0.00164	-0.00064	-5.08E-05	0.001317	0.000818	0.000321	2.55E-06	1.77E-06	8.03E-07
-0.00267	-0.0013	-0.00032	0.001674	0.001092	0.000462	2.82E-06	2.05E-06	9.76E-07
0.058963	0.039752	0.018218	-0.04223	-0.03283	-0.01798	-7.71E-05	-6.09E-05	-3.39E-05
0.256532	0.201006	0.112037	-0.08933	-0.07524	-0.04545	-6.81E-05	-6.03E-05	-3.79E-05
-0.02044	-0.01595	-0.00855	0.001299	0.00185	0.001469	-1.11E-05	-6.46E-06	-2.34E-06
-0.01872	-0.01634	-0.00976	-0.00417	-0.00196	-0.0003	-2.53E-05	-1.71E-05	-7.85E-06
0.180432	0.137528	0.074138	-0.07565	-0.06242	-0.03679	-8.42E-05	-7.11E-05	-4.27E-05
-0.01327	-0.00918	-0.00427	0.003464	0.002849	0.001591	2.42E-07	1.15E-06	1.07E-06
0.079992	0.056232	0.027336	-0.04926	-0.03889	-0.02171	-8.19E-05	-6.56E-05	-3.71E-05
-0.0023	-0.00106	-0.00022	0.001551	0.000996	0.000412	2.74E-06	1.96E-06	9.18E-07
0.041063	0.025953	0.01074	-0.03567	-0.02724	-0.01459	-7.14E-05	-5.56E-05	-3.04E-05
0.000282	0.000443	0.00031	0.000435	0.000196	3.52E-05	1.43E-06	8.54E-07	3.16E-07

Appendix-V

Output Membership Function

There are a total of 729 output membership functions for each fuzzy systems that are generated from ANFIS. The output membership functions are linear function with the six states as input variables and seven parameters designated as p_n, n= 0,2,3…6. The six parameters are the coefficients to the six variables and the last parameter is a constant.

$$output\ membership\ function = p_1 q_1 + p_2 q_2 + p_3 q_3 + p_4 \dot{q}_1 + p_5 \dot{q}_2 + p_6 \dot{q}_3 + p_0$$

A. Output Membership Function for the Fuzzy System One

Output membership function for the fuzzy system one that mimics LQR state-input one has a universe of discourse $[-8.6700 \times 10^{-5} \quad 4.3232 \times 10^{-4}]$. The parameters of the sample output function for the fuzzy system one is shown below.

Output MF	Function type	Output membership function parameters [p_1, p_2, p_3, p_4, p_5, p_6, p_0]
'out1mf1'	'linear'	[-1.79577228200378e-06,-1.37607297536001e-06,-7.46583126808716e-07,7.30500494655699e-07,6.05256990703541e-07,3.58419401594744e-07,-9.31382001653686e-06]
'out1mf2'	'linear'	[-2.97630056402817e-06,-2.25750573726622e-06,-1.20972277984421e-06,1.28462340206517e-06,1.05647519715323e-06,6.20024673861661e-07,-1.75616749229760e-05]
'out1mf3'	'linear'	[-3.83269344565281e-07,-2.90419785194646e-07,-1.55455555467827e-07,1.71841030318485e-07,1.40861193426221e-07,8.23934128741963e-08,-2.67073982602828e-06]
'out1mf4'	'linear'	[-2.66513340819117e-06,-2.02140411558853e-06,-1.08314219339780e-06,1.15088197608470e-06,9.46435896131153e-07,5.55409515433729e-07,-1.57462690578899e-05]
'out1mf5'	'linear'	[-6.28129763175224e-06,-4.73268782731637e-06,-2.51465522129433e-06,2.84085184866772e-06,2.32545948227108e-06,1.35710243139092e-06,-4.10241185245672e-05]
'out1mf6'	'linear'	[-6.15395530691737e-07,-5.20057944494250e-07,-3.07178752733244e-07,2.83922481868017e-07,2.48944569292109e-07,1.56823942952791e-07,-7.41838197258486e-06]
'out1mf7'	'linear'	[-7.35332579521020e-08,-5.58456918023714e-08,-2.98814435010901e-08,3.69555625944390e-08,3.01547931759958e-08,1.75374840927598e-08,-7.66995957010629e-07]
'out1mf8'	'linear'	[-1.90283838267908e-08,-6.55253550217398e-08,-6.06188147403200e-08,-1.68130325626391e-08,4.77846029691913e-09,1.45975063349059e-08,-1.91995971930155e-06]
'out1mf9'	'linear'	[4.31249416446470e-06,3.82148173043008e-06,2.38086197105028e-06,5.37661050319484e-07,1.64592777383247e-07,-8.98838014312723e-08,-5.45930325976848e-05]
'out1mf10'	'linear'	[-1.75431501002420e-06,-1.33016017991267e-06,-7.12513296256484e-07,7.58781413764907e-07,6.23781290270649e-07,3.65925832181255e-07,-1.04242776599863e-05]
'out1mf11'	'linear'	[-4.52300182663266e-06,-3.40962113402273e-06,-1.81297003712335e-06,2.03654296341018e-06,1.66776211039621e-06,9.73802122847798e-07,-2.91733402645938e-05]
'out1mf12'	'linear'	[-4.86322491710565e-07,-3.90686546206889e-07,-2.20845566858052e-07,2.47161890846984e-07,2.08308413144529e-07,1.25998191444189e-07,-5.41649831604152e-06]
'out1mf13'	'linear'	[-4.08055361921075e-06,-3.07196258524612e-06,-1.63106048509597e-06,1.86049574751210e-06,1.52118556015822e-06,8.86714453456366e-07,-2.73450682503220e-05]
'out1mf14'	'linear'	[-4.89224762570292e-06,-3.58634834064163e-06,-1.88302805026223e-06,3.56306127296450e-06,2.77458031014138e-06,1.54341278518996e-06,-0.000130115837911830]
'out1mf15'	'linear'	[-6.67417507756270e-06,-5.15084445390229e-06,-2.69814323515065e-06,6.04005897903741e-06,4.96303143041149e-06,2.90613314167606e-06,-0.000114097105153992]
'out1mf16'	'linear'	[-3.85905452666462e-08,-6.77562211363110e-08,-5.62227496440663e-08,3.20837748574436e-09,1.58843440781772e-08,1.80306667458292e-08,-7.89362021931605e-08]
'out1mf17'	'linear'	[-8.23565359747066e-06,-6.28933152229412e-06,-3.25097655782952e-06,6.04743727069776e-06,5.02374497213724e-06,2.96103966552120e-06,-9.98033871354249e-05]
'out1mf18'	'linear'	[7.32409569169192e-05,5.32006050251410e-05,2.71026072417423e-05,-2.03125835823408e-05,-1.69174653411870e-05,-9.96214367954881e-06,-0.000251659458113302]
'out1mf19'	'linear'	[-1.88377933866908e-08,-1.51015665846355e-08,-8.42103883477618e-09,1.04368006005472e-08,8.73906278983895e-09,5.20058933165926e-09,-3.43979245147719e-07]
'out1mf20'	'linear'	[2.30330242453491e-07,7.78074870302461e-08,-9.57791172439493e-09,-2.84761284800575e-07,-1.94580966770738e-07,-8.90915252465776e-08,5.50421780283155e-06]
'out1mf21'	'linear'	[4.07595105350026e-06,3.66452164178472e-06,2.32636361998169e-06,5.50029556423375e-07,1.88522499360839e-07, -7.39092346769004e-08,-4.64315374741693e-05]

B. Output Membership Function for the Fuzzy System Two

Output membership function for the fuzzy system two that mimics LQR state-input two has a universe of discourse $[-7.1900 \times 10^{-5} \;\; 3.6062 \times 10^{-4}]$. The parameters of the sample output function for fuzzy system two is shown below.

Output MF	Function type	Output membership function parameters [p_1, p_2, p_3, p_4, p_5, p_6, p_0]
'out1mf1'	'linear'	[-8.54986442554238e-07,-6.54821261923755e-07,-3.55013939427238e-07,3.48907279149972e-07,2.89050993826592e-07,1.71132852714589e-07,-4.43301691788761e-06]
'out1mf2'	'linear'	[-1.49074776535082e-06,-1.13299902306712e-06,-6.08461994885012e-07,6.33300023834855e-07,5.22040507333179e-07,3.07161533632306e-07,-8.25195445912306e-06]
'out1mf3'	'linear'	[-1.32207832591941e-07,-1.02906214427156e-07,-5.71734133797747e-08,4.58157404052383e-08,3.82853230487155e-08,2.29835060410312e-08,-4.91652734768608e-07]
'out1mf4'	'linear'	[-1.41971066605774e-06,-1.08312959248873e-06,-5.84444993729193e-07,5.90944292700445e-07,4.88233562289571e-07,2.88085136488529e-07,-7.63019054106326e-06]
'out1mf5'	'linear'	[-1.80926444352764e-06,-1.44320282672115e-06,-8.23928485457772e-07,5.18507539713912e-07,4.45231793198821e-07,2.75746682711364e-07,-5.65550514151870e-06]
'out1mf6'	'linear'	[-4.10551521428720e-07,-1.55823301537804e-07,7.04596445742204e-09,5.86788003717399e-07,4.21117268012070e-07,2.06448536893745e-07,-1.84987020287644e-05]
'out1mf7'	'linear'	[-2.63896248277089e-08,-1.89053646522499e-08,-9.72506709381578e-09,1.13337089214405e-08,8.75385303466948e-09,4.81496091382332e-09,-1.00545157200615e-07]
'out1mf8'	'linear'	[-1.95801210783441e-07,-5.18307365254681e-08,1.52650253226217e-08,1.09454249589727e-07,4.89458968784174e-08,4.05975564388689e-09,2.66950158260542e-06]
'out1mf9'	'linear'	[1.84260699079531e-07,4.56060904756030e-07,3.85903346798542e-07,3.69484894914461e-07,1.50377948894314e-07,-2.70363677741137e-09,1.04364086611826e-06]
'out1mf10'	'linear'	[-1.13515269607815e-06,-8.63295384067273e-07,-4.63954141307407e-07,4.80866785151693e-07,3.96584137416736e-07,2.33477499597183e-07,-6.25392160813977e-06]
'out1mf11'	'linear'	[-1.59024472962842e-06,-1.33284032165253e-06,-7.99130021806280e-07,2.25514094980510e-07,2.26897941134503e-07,1.62808420187426e-07,3.14851851618700e-06]
'out1mf12'	'linear'	[2.35041931416424e-08,1.23849250638240e-07,1.25335511313402e-07,2.12388301306291e-07,1.32422971269439e-07,5.01544362993180e-08,-1.01210493848361e-05]
'out1mf13'	'linear'	[-1.53315392225975e-06,-1.20797783050968e-06,-6.79196205679998e-07,4.99516027930402e-07,4.22626759062150e-07,2.57047696420541e-07,-6.21475759413375e-06]
'out1mf14'	'linear'	[-1.04693132641894e-05,-6.46968237676541e-06,-2.51539021246300e-06,6.76825364020579e-06,5.19391825456039e-06,2.76259259519051e-06,-9.26697709589025e-05]
'out1mf15'	'linear'	[-5.91250216870833e-05,-4.47810652816690e-05,-2.39122992146720e-05,2.04913140929927e-05,1.72250020546353e-05,1.03137598996311e-05,-2.91771353392578e-05]
'out1mf16'	'linear'	[-1.64417913665686e-07,-7.64954260654375e-08,-1.82718591357090e-08,8.33639250820586e-08,4.95427853681483e-08,1.73351001950275e-08,-1.38032928465018e-06]
'out1mf17'	'linear'	[-2.19286015270386e-05,-1.33836231146970e-05,-5.57804412743284e-06,8.55580476997350e-06,5.92541544956401e-06,2.76756527252522e-06,-5.34445164127437e-05]

C. Output Membership Function for the Fuzzy System Three

Output membership function for the fuzzy system three that mimics LQR state-input three has a universe of discourse $[-4.2700 \times 10^{-5} \quad 2.1724 \times 10^{-4}]$. The parameters of the sample output function for fuzzy system-three is shown below.

Output MF	Function type	Output membership function parameters [p_1, p_2, p_3, p_4, p_5, p_6, p_0]
'out1mf1'	'linear'	[-8.54722017281475e-07,-6.54374968754450e-07,-3.54590489167889e-07,3.49512764842023e-07,2.89511784207985e-07,1.71369190396236e-07,-4.43384820087416e-06]
'out1mf2'	'linear'	[-1.20722747570407e-06,-9.24940553448515e-07,-5.02001565026321e-07,4.91075323941496e-07,4.06379915626363e-07,2.40378180279670e-07,-6.51860310897808e-06]
'out1mf3'	'linear'	[-4.65673427878593e-08,-3.28748545812366e-08,-1.61897884789352e-08,2.61467808898142e-08,2.05601102880062e-08,1.14340087079874e-08,-4.91431553247631e-07]
'out1mf4'	'linear'	[-9.48649279659152e-07,-7.25665100395949e-07,-3.93134282259661e-07,3.90559081870118e-07,3.22683303097055e-07,1.90527684807714e-07,-5.32047533448335e-06]
'out1mf5'	'linear'	[-1.83786899944884e-06,-1.08506312310619e-06,-3.92107396925090e-07,1.79748496873413e-06,1.35673449337268e-06,7.14074155816130e-07,-5.15191007638007e-05]
'out1mf6'	'linear'	[-7.98246559101326e-07,-5.59038614123042e-07,-2.71081403408043e-07,4.46338695096734e-07,3.51942027573171e-07,1.95893095817133e-07,-8.02122833622935e-06]
'out1mf7'	'linear'	[-3.54209470385056e-08,-2.47159784052091e-08,-1.20333004614719e-08,1.97559380622898e-08,1.54111598877642e-08,8.48940497081933e-09,-3.36358377043087e-07]
'out1mf8'	'linear'	[-7.35100605622243e-07,-4.52412266756782e-07,-1.86481991776805e-07,4.15836923025409e-07,3.03581785083820e-07,1.52592995875415e-07,-3.78423949429705e-06]
'out1mf9'	'linear'	[2.92180690976363e-07,3.81134045772327e-07,2.7491727258747 6e-07,5.94171991255591e-08,-2.77702091638404e-08,-6.26081365678732e-08,2.29738986892185e-06]
'out1mf10'	'linear'	[-5.52703679069876e-07,-4.25595175929000e-07,-2.32408856693884e-07,2.18569817805613e-07,1.81475193162832e-07,1.07789982902956e-07,-2.86347328104252e-06]
'out1mf11'	'linear'	[-5.41901925416274e-07,-2.15613187038754e-07,4.26608938295031e-09,8.68179568498194e-07,6.37044842973684e-07,3.21808346057737e-07,-2.89295154008225e-05]
'out1mf12'	'linear'	[-4.47347874881290e-07,-3.11409260786839e-07,-1.49965209061416e-07,2.57214064161944e-07,2.02020281880038e-07,1.11960955989169e-07,-5.25436201546321e-06]
'out1mf13'	'linear'	[-1.09640349254376e-06,-6.96920687561903e-07,-2.87980489816362e-07,8.97795092185649e-07,6.90172990031507e-07,3.71543289299980e-07,-2.06397416728185e-05]
'out1mf14'	'linear'	[-6.00222843216729e-05,-4.65927532348603e-05,-2.54596572492136e-05,2.11670519783189e-05,1.81524043505549e-05,1.11023164297692e-05,-2.91161117923783e-05]
'out1mf15'	'linear'	[1.45368921093379e-05,1.09975759282210e-05,5.78653810434031e-06,-6.31760194753668e-06,-5.31005581303616e-06,-3.16854913531716e-06,-2.20386752752547e-05]
'out1mf16'	'linear'	[-3.55478188204207e-07,-2.45196469205083e-07,-1.16923456094240e-07,1.96869651735706e-07,1.53787855457970e-07,8.45985198942370e-08,-3.72706362245352e-06]
'out1mf17'	'linear'	[-9.17460280086895e-06,-5.92975524257284e-06,-2.61382004299757e-06,3.96954597175104e-06,2.95240054716230e-06,1.51003779693430e-06,-7.47842714489248e-05]