Dynamic Modeling and ANFIS controller design to Enhance Stability of Self-Balancing Two-Wheeled Electric Scooter on Uneven Terrain

Wendwsen Birhane

A Thesis Submitted to

Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

Sep, 2024

Adama, Ethiopia

Dynamic Modeling and ANFIS controller design to Enhance Stability of Self-Balancing Two-Wheeled Electric Scooter on Uneven Terrain

Wendwsen Birhane

Advisor: Dr. Endalew Ayenew (Assistant professor)

A Thesis Submitted to

Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

Sep, 2024

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "Dynamic Modeling and ANFIS controller design to Enhance Stability of Self-Balancing Two-Wheeled Electric Scooter on Uneven Terrain" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Wendwsen Birhane	______________________	______________
Name	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Dynamic Modeling and ANFIS controller design to Enhance Stability of Self-Balancing Two-Wheeled Electric Scooter on Uneven Terrain**" prepared under my guidance by Wendwsen Birhane, submitted in partial fulfillment of the requirements for the degree of Masters of science in Control Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Endalew Ayenew	________________	________________
Advisor	Signature	Date

APPROVAL PAGE

I, the advisor of the thesis entitled, "**Dynamic Modeling and ANFIS controller design to Enhance Stability of Self-Balancing Two-Wheeled Electric Scooter on Uneven Terrain"** by Wendwsen Birhane, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Endalew Ayenew	______________	______________
Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Wendwsen Birhane We, have read and evaluated the thesis entitled "Dynamic Modeling and ANFIS control to Enhance Stability of Self-Balancing Two-Wheel Electric Scooter on Uneven Terrain" and examined the candidate during open defense. This is, therefore to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Control System Engineering.

______________	______________	______________
Chairperson	Signature	Date
______________	______________	______________
Internal Examiner	Signature	Date
Dr. Riessom Weldegiorgis	Riessom	19/09/2024
External Examiner	Signature	Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC)

______________	______________	______________
Department Head	Signature	Date
______________	______________	______________
School Dean	Signature	Date
______________	______________	______________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I am deeply grateful to God for granting me the strength, health, and perseverance to complete this thesis. Without His blessings, this achievement would not have been possible.

I would also like to express my sincere appreciation to my advisor, Dr. Endalew Ayenew, for his invaluable support and guidance throughout my studies. His insights and encouragement have been instrumental to the success of this work.

Finally, I extend my heartfelt thanks to my friends and colleagues for their unwavering support during this journey. Your encouragement and camaraderie have been greatly appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION AND ACRONYMS

ANFIS	Adaptive-Network Fuzzy Inference System
ASTU	Adama Science and Technology University
HT	Human Transport
IAE	Integral Absolute Error
ISE	Integral Square Error
ITAE	Integral Time-Weighted Absolute Error
NN	Neural Network
PID	Proportional, Integral, and Derivative
PSO	Particle Swarm Optimization
SBTWES	Self-Balancing Two-Wheeled Electric Scooter
STSMC	Super Twisting Sliding Mode Controller
TLBO	Teaching Learning based Optimization

LIST OF SYMBOLS

D	Distance b/n the two wheels contact
dx, dy	disturbance due to uneven terrain affecting the horizontal and vertical motion
$d\delta$	disturbance affecting the yaw angle dynamics due to uneven terrain
G	Gravitational acceleration
H_L, H_R	Reaction force b/n the wheels and the ground
F_{fL}, F_{fR}	Friction force b/n the wheels and the ground
i_a	Current flowing through Armature Coil
I_R	Motor Inertia
J_B	Inertia of the body chassis
J_{TL}, J_{TR}	Rotating moment of inertia
K_e	DC motor back EMF coefficient
K_m	Torque constant of DC motor
L	Distance b/n the center of gravity and the z-axis
L_a	Inductance of armature
M_B	Mass of the body
M_W	Mass of the wheels
τ_L, τ_R	Torques of the left and right
V_a	Voltage supplied to the Armature
Ve	Counter EMF of the motor
θ	Balance angle
δ	Direction angle
ΔHL, ΔHR	Horizontal Ground Reaction Forces HL and HR due to uneven ground

ABSTRACT

This thesis looks at the dynamic modeling and design of an optimal controller for a Segway, focusing on important features like self-balancing and steering control, especially on uneven surfaces. Self-balancing is crucial for stability during movement, while steering control is essential for accurate and responsive direction changes on rough terrain. The study starts with a detailed analysis of how the Segway's mechanical, electrical, and control systems interact under different ground conditions. A mathematical model is created to understand the Segway's behavior in various situations. To improve these functions, STSMC is developed to handle balancing and steering. The STSMC parameters are fine-tuned using Teaching-Learning-Based Optimization (TLBO). The results from these tuned parameters are then used to train ANFIS for pitch angle control and a Neural Network (NN) for yaw angle control. The ANFIS_STSMC achieves a settling time of 0.1802 seconds and a rise time of 0.2538 seconds for balance angle control, with no steady-state error. For yaw angle control, the NN controller trained with STSMC data has a settling time of 0.2994 seconds and a rise time of 0.3495 seconds, also with no steady-state error and shows an overshoot of 0.16% for balance angle and 0.05% for direction angle. The performance of these controllers is tested using MATLAB/Simulink. The results show that ANFIS_STSMC and NN controllers greatly enhance the Segway's stability and steering ability, especially on uneven terrain.

Keywords: ANFIS, Segway, TLBO, STSMC, NN, Uneven terrain.

CHAPTER ONE

INTRODUCTION

1.1 Background

Several individuals are choosing environmentally friendly options in response to climate change and global warming. They have begun switching from conventional energy sources to alternative ones like solar and wind power. Transportation, however, is the main issue that needs to be addressed because it produces thousands of tons of carbon dioxide every day. Therefore, in order to protect the environment, switching to clean electric vehicles like electric scooters instead of fuel-burning ones should be performed.

Over the last two decades, there has been a consistent rise in the demand for and advancement of personal transportation systems. Within the realm of personal mobilities (PM), the Segway, created by American inventor Dean Kamen in 2001(*Super Segway*, n.d.), presents a distinct structure and operational modes compared to traditional transport systems. Due to its lightweight and portable nature, the Segway has the potential to serve as an alternative for short-distance travel, akin to a bicycle or motorcycle. The widespread adoption of the Segway hinges on its control stability and driving safety from a practical standpoint.

The Segway Company responded to the growing need for personal transportation vehicles in today's rapidly expanding transportation market by introducing self-balancing scooters. Nowadays, a lot of industries, including the travel and tourism, film industry, and manufacturing sectors, use these personal transporters, which perfectly capture the idea of personal mobility. Enhancing the effectiveness of human transportation while cutting expenses is the goal.

Urban transportation has undergone a paradigm shift as a result of the widespread adoption of electric mobility options, with two-wheel electric scooters emerging as adaptable and sustainable alternatives. These electric scooters are becoming an essential part of contemporary transportation, and their control systems and dynamic behavior are crucial to maintaining rider safety and vehicle performance.

Likely you won't fall on your face if you stand up and tilt forward, throwing off your equilibrium. Because the fluid in your inner ear changes when you are unbalanced, your brain detects this and sends a signal to your leg to halt you from falling. Your brain will continue pushing your legs forward to maintain your upright posture if you continue to lean forward. You move forward step by step rather than collapsing. Much of the same functions are performed by the Segway HT, with the exception of having wheels in place of legs, a motor in place of muscles, a group of microprocessors in place of a brain, and an advanced set of tilt sensors in place of an inner-ear balancing system. The Segway recognizes when you are bending forward, much like your brain does. It moves the wheels at precisely the appropriate speed to maintain equilibrium, allowing you to advance. The Segway HT can balance on just two wheels thanks to a patented mechanism that the company refers to as dynamic stabilization.

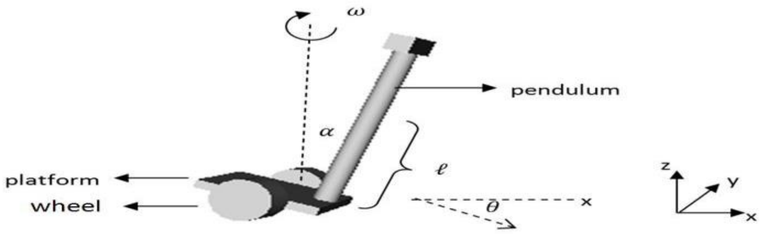


Figure 1.1: self-balancing two-wheeled robot (Asali et al., 2017a)

While the idea of electric scooters is not new, their applicability is growing worldwide, and they are becoming more important in the current environment.

Designing a controller for a Self-Balancing Two-Wheel Electric Scooters is a complex task due to various inherent challenges associated with the system's dynamics. Significant obstacles in designing controllers for Self-Balancing Two-Wheel Electric Scooters include:

<u>Nonlinearity:</u> The system’s motion is described by highly nonlinear, which makes the system intrinsically difficult to control.

<u>High Dimensionality:</u> The system involves multiple interconnected links, resulting in a high-dimensional state space.

<u>Instability:</u> The self-balancing two-wheel scooter is inherently unstable, and small disturbances can lead to chaotic and unpredictable behavior.

<u>Controller Tuning:</u> The system’s sensitivity to the system to controller gains and the interplay between different control parameters requires careful tuning to achieve stable and satisfactory performance.

<u>Energy Constraints:</u> Designing controllers that optimize energy consumption while ensuring stability and performance is a significant obstacle.

Overcoming these obstacles typically involves a combination of advanced control strategies, including nonlinear, optimal, and adaptive control, as well as thorough modeling and simulation studies to understand the system dynamics. As a result, sophisticated controller designs are essential to manage the yaw angle for steering the robot left and right, controlling the pitch angle for balance, and regulating the overall position of the robot.

1.2 Statement of the Problem

This private transporter vehicle offers advantages improved accessibility and fuel-free, potentially making it a key solution to the growing global challenges of traffic congestion and environmental pollution. In our country, where there are numerous tourist attractions, a convenient mode of transport is needed to move people from one spot to another. The scooter is easy to operate and suitable for everyone, regardless of age. Moreover, this vehicle is designed with inclusivity in mind, allowing it to be driven using just one leg or even an artificial limb, making it particularly valuable for individuals who have lost a leg in war or due to other circumstances. The scooter's adaptive control system ensures that those with physical limitations can confidently navigate it, providing greater mobility and independence.

Additionally, the scooter is engineered to handle uneven terrain, making it ideal for varied landscapes, including parks, trails, and other natural environments. Its robust design and stability features enable it to traverse rough or uneven surfaces with ease, ensuring a smooth ride even in challenging conditions. This versatility further enhances its appeal, offering a practical and empowering solution for both everyday transportation and recreational use, especially in areas where traditional vehicles cannot go.

1.3 Objectives

1.3.1 General Objective

The overall goal of this research work is to develop a Dynamic Modeling and ANFIS control design for Enhanced Stability: Self-Balancing and Yaw Angle Control of Two-Wheel Electric Scooters on Uneven Terrain.

1.3.2 Specific Objectives

- To develop a detailed mathematical model for the Self-Balancing Two-Wheeled Electric Scooters.
- To perform the simulation of a segway based on a mathematical model and capture the dynamic behavior of the system under many scenarios.

- To design and tune parameters for STSMC controller using optimization algorithms.
- To design an ANFIS Controller for balancing and NN for direction using data from STSMC.
- To evaluate the generated Segway's performance in MATLAB/Simulink.

1.4 Scope

The scope of this research work is to derive a mathematical model of the Self-Balancing Two-Wheel Electric Scooters dynamics considering external disturbances and parametric uncertainties and design an advanced control strategy (ANFIS and NN) for trajectory control of a Segway using MATLAB/SIMULINK model.

1.5 Limitation

This study is confined to the specific application of Self-Balancing and Yaw Angle Control of Two-Wheel Electric Scooters on Uneven Terrain and limited to mathematical design and using MATLAB/SIMULINK, simulate the system, While the outcomes may offer valuable insights, the study acknowledges certain limitations, including assumptions made in the modeling process and the potential sensitivity of results to specific system parameters.

1.6 Motivation

This investigation is inspired by the widespread recognition of the two-wheeled electric scooter as a transportation device with sought-after characteristics like minimal noise, steady speed, simplicity of usage, and minimal energy consumption, and environmental friendliness. The extensive range of applications, from navigating densely populated and polluted cities to providing noiseless and eco-friendly solutions in hospitals, shopping malls, airports, parks, and entertainment settings, underscores the potential societal impact of advancing the stability and control of such scooters. Recognizing their comparative advantages over four-wheeled vehicles, with reduced expenses and smaller dimensions, simplified operation, and enhanced parking convenience, fuels the motivation to delve into dynamic modeling and ANFIS control design. By enhancing the scooter's performance on uneven terrain, this study aims to contribute not only to the field of vehicle dynamics but also to the broader objectives of sustainable and efficient urban mobility, thus motivating a comprehensive exploration of these technologies.

1.7 Significance

The thesis " Dynamic Modeling and ANFIS control design for Enhanced Stability: Self-Balancing and Yaw Angle Control of Two-Wheel Electric Scooters on Uneven Terrain." aims to develop a design model and advanced control strategies for two-wheel electronic scooters to enhance their stability on uneven terrain. The study is significant because it addresses the challenges of self-balancing and yaw angle control of two-wheel electronic scooters on uneven terrain, which is a common problem faced by riders. The study will contribute to the development of two advanced control strategies ANFIS and NN for two-wheel electronic scooters, which can improve their stability and safety on uneven terrain.

Because an electric scooter runs on human movement, it offers a reliable and secure mode of transportation. Its versatility makes it useful not only in the travel business but also in airports, shopping malls, and offices. Because of its inherent properties, this vehicle is perfect for traveling across cities and provides a wearier-free substitute for typical tourism methods, eliminating the need for lengthy walking. The safety and environmental benefits of electric power make it an even more appealing zero-emission mode of transportation.

1.8 Thesis Organization

The work for the thesis can be arranged as follows:

Chapter 2: Review the existing literature on control systems developed for Robots, Wheeled Mobile Robots (WMR), and Two-Wheeled Mobile Robots (TWMR).

Chapter 3: This section outlines the materials used, the methods to be followed, the mathematical modeling, and the controller design for the system. Additionally, it introduces the optimization algorithms, specifically TLBO and PSO.

Chapter 4: This section presents the results and discussions related to the proposed controller and optimization techniques.

Chapter 5: Summarizes the contributions of this research, concludes, and the recommendation for further works.

CHAPTER TWO

2. REVIEW OF RELATED LITERATURE

2.1 Summary

This section summarizes previous attempts to create self-balancing gadgets. The information gained from reading different publications is quite helpful in the initial design of a segway.

2.2 Introduction

The Segway is a two-wheeled, self-balancing electric scooter invented by Dean Kamen and brought to market in 2001(*Segway - Wikipedia*, n.d.). It is a personal transport device, making it the first of its kind. The Segway PT (Personal Transporter) utilizes gyroscopic sensors and balance sensors to enable the rider to control the scooter by shifting their body weight. It is powered by lithium-ion batteries and uses brushless DC motors in each wheel. The Segway PT is designed for short-distance travel in urban and suburban areas, offering a unique and intuitive way to move around. Despite initial hype, Segway faced challenges in finding a widespread market due to its limited practicality in certain settings. However, it remains a significant innovation in the field of personal transportation.

The two-wheeled self -balancing electric scooter known as a Segway operates on a complex network of sensors, control systems, and motors that are intended to maintain dynamic equilibrium. The following are the main elements and how they work:

- Accelerometer: Accelerometers measure the rate of acceleration or deceleration. They provide information about the scooter's movement, helping to determine if the rider is leaning forward, backward, or remaining stationary.
- Control System: The control system processes data from the gyroscopic sensors and accelerometers in real-time. It calculates the required adjustments to maintain balance and stability based on the rider's movements.
- DC Motors: Electric motors, one for each wheel, are responsible for propelling the Segway. The control system adjusts the torque and rotation direction of these motors to counteract the rider's movements and maintain balance.
- Leaning: When a rider leans forward, the Segway detects this inclination through the sensors. The control system responds by accelerating the motors, propelling the scooter

forward. Similarly, when the rider leans backward, the motors decelerate to move the Segway in reverse.

- Turning: To turn, the rider shifts their weight to one side. The control system adjusts the speed of the motors on one side, causing the Segway to pivot in the desired direction.
- User Interface: The Segway is typically equipped with a user interface, such as handlebars or a steering grip, allowing the rider to control the segway's speed and direction intuitively.

The continuous interaction between sensors, control systems, and motors creates a seamless and dynamic self-balancing mechanism. As the rider shifts their weight or leans in a certain direction, the Segway responds instantly to maintain stability, providing an efficient and intuitive mode of personal transportation.

Figure 2.1: segway (*Segway* - *Wikipedia*, n.d.)

2.3 Related Work of Studies

Current research aims to improve their ability to move, see their environment, and behave autonomously. Making control systems smarter is a key goal in order to improve their ability to manage unknown causes and interruptions. Among the techniques being used to drive and stabilize Two-Wheel Electric Scooters is system modeling (Ahmed & Saleh Alshandoli, 2020a; Asali et al., 2017b; Kim & Kwon, 2015; Mudeng et al., 2020) ,PID control(Mudeng et al., 2020) ,neural network (Ahmed & Saleh Alshandoli, 2020a),LQR (Mg et al., 2020),Adaptive and Backstepping (Son & Anh, 2014a), Sliding mode (Yang et al., 2019) , etc.

The primary objective of this SBTWES is to maintain the balance of its rider on two wheels aligned with each other. Both wheels receive power from a DC motor.

According to a study conducted by Ahmed and Saleh in 2020 (Ahmed & Saleh Alshandoli, 2020a) the use of a Neural Network (NN) controller and a traditional PID controller, for Segway showed results in terms of controlling the position and handle bear angle. The study concludes that Neural Network (NN) controller offers performance and accuracy in controlling the segway compared to the PID controller. However, a novel limitation of this paper is it does not address how the NN controller performs under varying environmental conditions or dynamic loads.

(Chakrabarti, n.d.) examines the creation and management of the segway that employes LQR adjustable gain. It discusses the difficulties of managing the segway when the driver's weight is unpredictable and offers the use of an altered LQR to solve this problem. The study presents experimental and simulated findings that shows how successful the control method is. However, it does not explore the long-term adaptability of the altered LQR controller to continuous or abrupt ground disturbance and weight changes over time.

AN adaptive backstepping control strategy for a self-balancing two-wheel electric scooter is presented in the paper (Son & Anh, 2014a). The adaptive backstepping self-balancing control system for a two-wheel electric scooter is shown in this publication. It combines adaptive backstepping control with feedback control that complies with Lyapunov stability. A feedback control rule is constructed to efficiently regulate the scooter's self-balancing feature by utilizing the regulated function and uncertain parameter estimation are done using a recursive structure. In this simulation, the adaptive backstepping control shows good performance and resilience. It's important to keep in mind, though, that backstepping control requires exact model parameters, and the suggested solution makes no mention of its shortcomings, like how uneven the ground or outside influences the scooter's stability.

The paper (Sevekari et al) Presents a Robust Control System to Ensure Safe and Consistent Functioning of a Two-Wheeled Human Transporter. In order to improve the two-wheeled human transporter's (TWHT) stability and safety, a higher order sliding mode control (HOSMC) is proposed. Higher-order sliding mode differentiators are used to estimate states that cannot be directly monitored in the control development process, which takes into account a reduced order model. The suggested controllers are put to the test using simulations and contrasted with a linear state feedback controller (SFC) to determine how effective they are. They vary the rider's weight to test the robustness of HOSMC. However, the research does not explore particularly how the controller performs under actual operating conditions, such as uneven terrain, sensor noise, or sudden external disturbances. Moreover, while STSMC is known for its robustness against uncertainties and disturbances, it can suffer from issues such as chattering, increased computational complexity, and sensitivity to parameter variations.

The paper (Ahmed & Saleh Alshandoli) explores replacing the traditional PID controller in the industrial use of the segway, a personal transportation device, with a neural network controller. The study utilizes a basic inverted pendulum model to evaluate the segway's performance by managing the cart position and pendulum angle to enhance stability, response, and system handling. The results from both PID and neural network controllers are compared on systems output and analysis. Overall, the comparison indicates that the neural network controller delivers superior performance compared to PID. However, this approach may not fully capture the complexities of real-world conditions, such as rider dynamics, or environmental disturbances.

The "modeling, simulation, and control of a dynamically unstable two-wheeled mobile robot (TWBMR)" are investigated in work of (Chawla et al). It improves tracking performance and system stability by creating clever control algorithms to handle nonlinear behaviors. The robot was accurately represented by feed-forward (FF) and cascade-forward (CF) neural networks with different hidden layers, resulting in a low Mean Squared Error (MSE). ANFIS was used to change the control parameters to their ideal values. Experiments using various reference signals showed that the intelligent controller outperformed the fuzzy logic controller (FLC) in terms of tracking accuracy, performing well in both transient and steady-state scenarios.

Segway-like systems, known for their complex dynamics and high nonlinearity, are traditionally designed for urban environments, which limits their performance on uneven terrain. Ensuring stability and maneuverability in such conditions remains a significant challenge. While various advanced control methods have been proposed, they often necessitate the replacement of conventional controllers with more intelligent systems, and few have been extensively tested or optimized for off-road scenarios. Furthermore, many existing approaches focus either on balance or direction control, rarely addressing both in a unified framework. This research addresses these gaps by implementing an ANFIS-based Super Twisting Sliding Mode Control (ANFIS-STSMC) for balance and a Neural Network (NN) for direction control, tailored specifically for challenging terrains. This integrated approach not only improves stability and control precision but also offers a more sustainable and adaptable solution for electric scooters in diverse environments, advancing the field beyond the current urban-focused applications.

CHAPTER THREE

3 METHODOLOGY AND SYSTEM MODELING

3.1 Introduction

This chapter presents, the methodology, which includes materials and software tools used and methods followed to achieve the settled goal. creation of a mathematical model that includes a segway's “body, wheel,”(Son & Anh, 2014b) and actuator. STSMC is developed to handle balancing and steering. The STSMC parameters are fine-tuned using TLBO and PSO algorithms, the parameters of both controllers have been adjusted in relation to an objective function (J). An ANFIS for pitch angle control and a Neural Network (NN) for yaw angle control are then trained using the outcomes of these adjusted parameters. Next, a super twisted sliding mode controller (STSMC) and an adaptive neuro-fuzzy inference system (ANFIS) are used to balance the segway. In addition, the direction of the segway is managed by a neural network controller.

3.2 Used Materials

For this thesis, MATLAB 2023a/SIMULINK is used to simulate and study the system model. This software is really handy and has lots of tools. It helps us solve math problems and visualize solutions easily. Also, Microsoft Office 2021 used to write the thesis. Instead of typing out equations directly, we'll use MATHTYPE to make them look better and for drawings some blocks are generated with draw.io software.

3.3 Followed Methods

Generally, the following figure lists the methods used in this study to fulfill the necessary goals. It shows how this thesis was completed, beginning with a review of relevant literature to identify gaps in knowledge and potential remedies. The mathematical modeling for the thesis is established by extensive research using many sources. Following the acquisition of the overall dynamic modeling, controller design is completed and the controller's parameters are adjusted by the application of metaheuristic algorithms (TLBO and PSO). The results from these tuned parameters are then used to train an Adaptive Neuro-Fuzzy Inference System (ANFIS) for pitch angle control and a Neural Network (NN) for yaw angle control. Next, MATLAB/Simulink is used to implement the simulation.

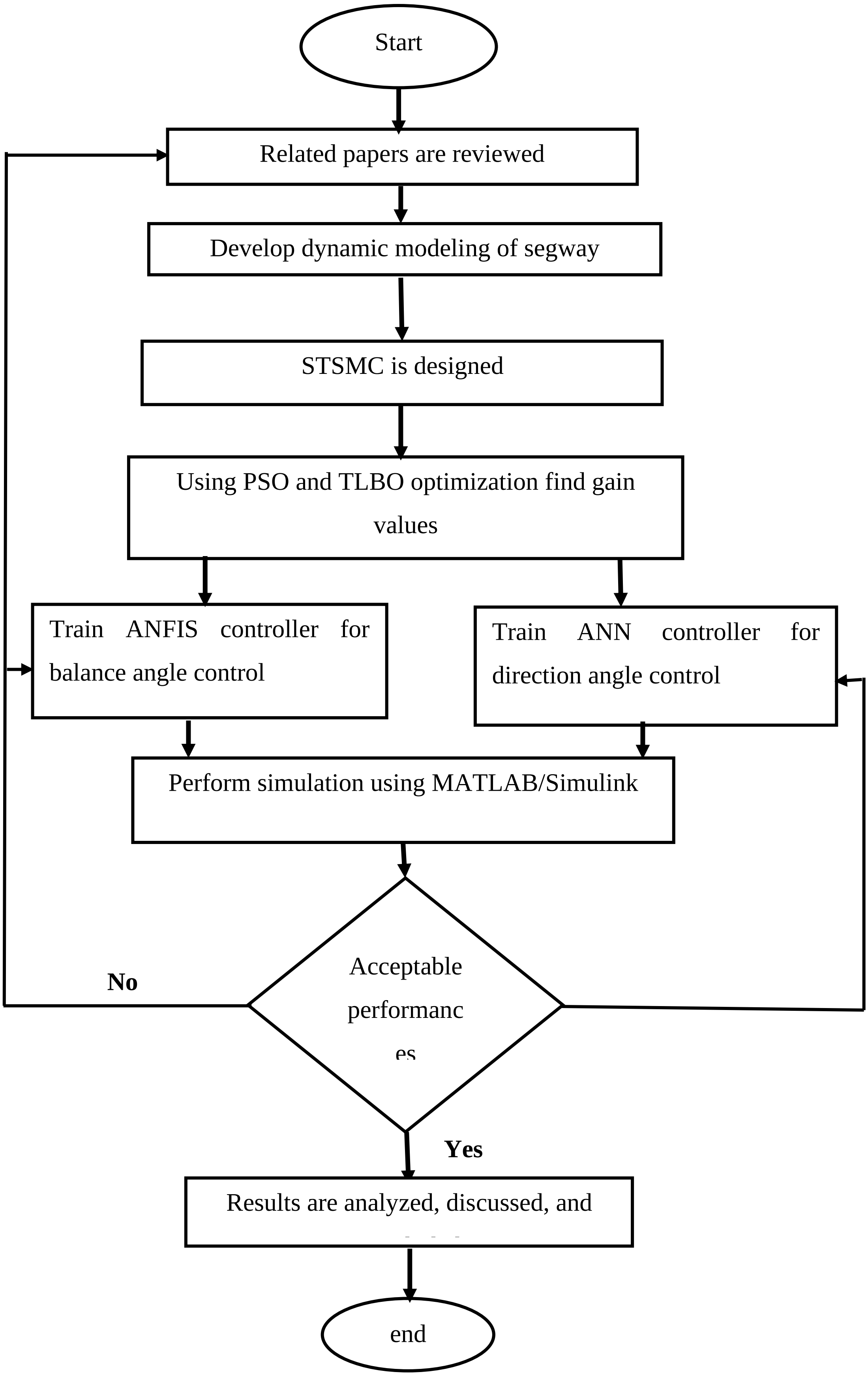


Figure 3.1: Work flow chart

3.4 Segway's Operating Principle

The first of its kind, the Segway is a self-balancing, portable personal mobility device. The Segway's cutting-edge technology allows it to move you over most terrains quickly and silently while emitting no harmful emissions into the atmosphere. It's enjoyable and simple to use.

The self-balancing feature of the suggested scooter is its main benefit. This characteristic aids in the scooter's constant stability.

The Segway's dynamics are exactly like those of the inverted pendulum, a well-known control problem. For propulsion or balance, the servos can be used to rotate the wheels forwards or backwards. The rider only needs to lean forward or backward in the desired direction to accelerate or decelerate. "Gliding" on a Segway is easy and quick to pick up.

3.5 Block Diagram for Segway

As illustrated in Figure 3.3, block diagram depicts the system for a segway, which operates as a Multiple Input Multiple Output (MIMO) system. Due to its inherent characteristics, this system is both nonlinear and unstable. To rectify this instability, a controller is necessary. The system features two inputs, both representing angles. The first one governs the scooter's balance, while the second angle controls its direction (either turning left or right). These angles are compared with a reference angle, and any deviation forms an error angle. This error angle is then directed to the respective controller, which works to stabilize the scooter's balance and control its direction.

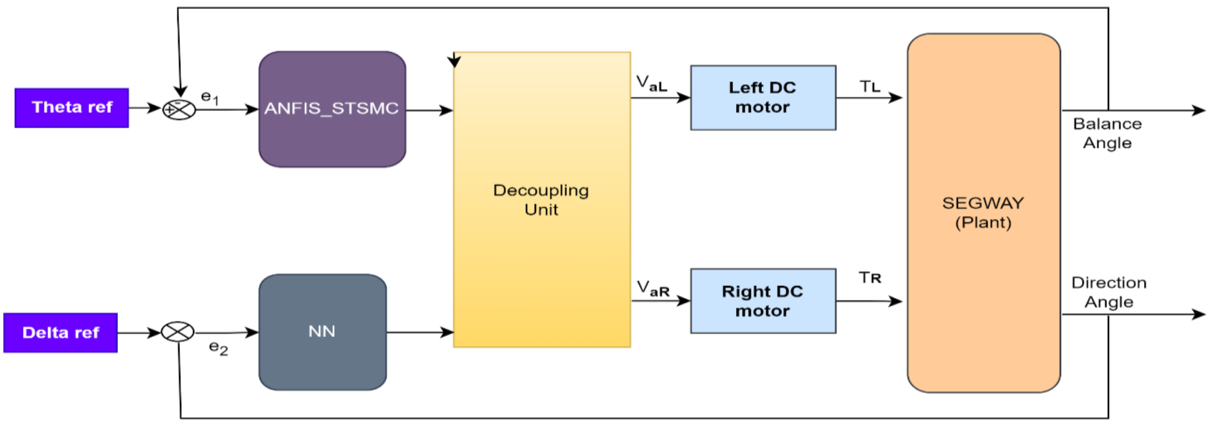


Figure 3.2: Proposed Block Diagram for Segway.

3.6 Modeling of a Segway System

To effectively create a control system guiding the Segway to a desired position and integrating various behaviors with its inputs, we first need a mathematical model of the Segway. The Segway's movements involve rotation around its vertical axis and forward/backward motion along the axis between its two wheels.

The Segway is an electrical actuator-driven mechanical system that is underactuated, therefore modeling it involves taking into account both its electrical and mechanical subsystems. There are three parts to the mathematical model:

a) Wheel Modeling: This part focuses on how the wheels rotate around the Segway's vertical axis and move along the central axis between them.

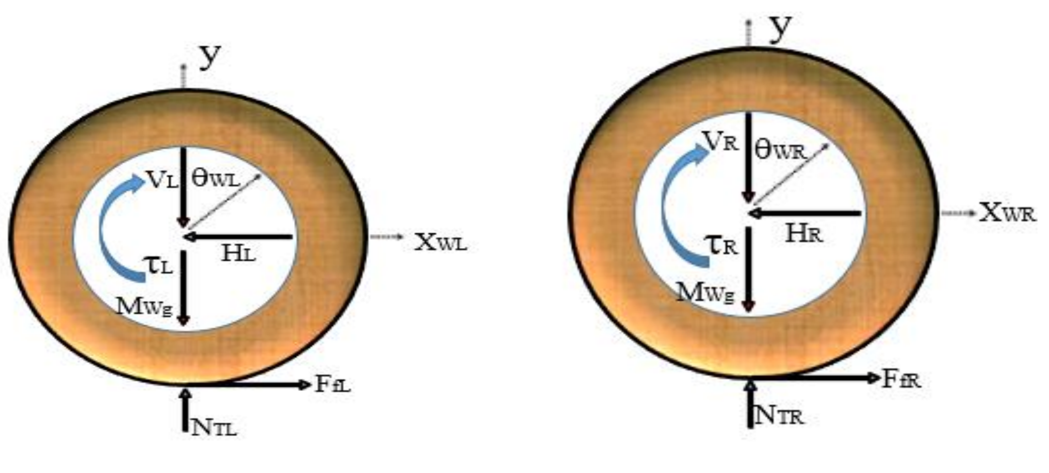
b) Body Modeling: Here, we examine the dynamics of the Segway's body, including its balance and response to external forces.

c) DC Motor (Actuator) Modeling: This component deals with modeling the electric actuator that drives the Segway.

To derive the mathematical model, we apply Newton's second law method and design each part of the model separately.

3.6.1 Dynamics of Wheel

According to Newton's second law, the forces acting on both left and right wheels are analyzed. The dynamic force analysis around the X-axis for the left wheel is detailed, and it is noted that the right wheel exhibits similar characteristics.



(A) (B)

Figure 3.3: wheel free body for left (A) and right (B)

The forces operating on the wheels are examined by using Newton's second rule of motion along the horizontal (X-axis):

$$\sum F_{net,x} = ma \tag{3.1}$$

$$M_w \ddot{X}_{WL} = F_{fL} - H_L \tag{3.2}$$

The forces operating on the wheels are examined by using Newton's second rule of motion along the vertical (Y-axis):

$$\sum F_{net,y} = m.a_y \tag{3.3}$$

$$M_w \, \ddot{y}_{WL} = N_{TL} - V_L - M_w g \tag{3.4}$$

The total of the moments around the center of the wheel is equal to:

$$\sum M_o = I\alpha \tag{3.5}$$

$$I_w \, \ddot{\theta}_{WL} = \tau_L - F_{fL} R \tag{3.6}$$

The correlation between the angular and linear motion components is provided by:

$$X_{WL} = \theta_{WL} R \tag{3.7}$$

Left wheel moment of inertia:

$$I_{WL} = \frac{1}{2} M_{WL} R^2 \tag{3.8}$$

Add the equation for the left and right wheels:

$$M_W\left(\ddot{X}_{WL} + \ddot{X}_{WR}\right) = \left(F_{fL} + F_{fR}\right)_{H_L + H_R} \tag{3.9}$$

$$\begin{cases} I_{WL}\ddot{\theta}_{WL} = \tau_L - F_{fL} R & \text{for the left wheel and} \\ I_{WR}\ddot{\theta}_{WR} = \tau_R - F_{fR} R & \text{for the right wheel} \end{cases} \tag{3.10}$$

Solve Eq. (3.10) as follow

$$\begin{cases} F_{TL} = \dfrac{\tau_L - I_{WL}\ddot{\theta}_{WL}}{R} & \text{for the left wheel and} \\ F_{TR} = \dfrac{\tau_R - I_{WR}\ddot{\theta}_{WR}}{R} & \text{for the right wheel} \end{cases} \tag{3.11}$$

3.6.2 Dynamics of Body

The scooter may be shown rotating around its tilt angle (θ) and angular velocity (ω) in the below picture. It can also rotate along its vertical axis (δ) and corresponding $\dot{\delta}$. In addition, the location x and its matching speed $\dot{x}$ are related to the wheels' linear motion.

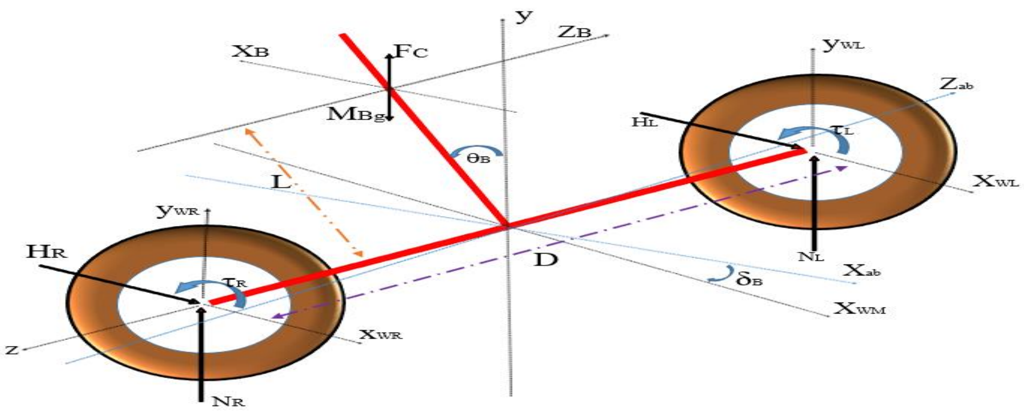


Figure 3.4: Force analysis of body of the segway

The forces operating on the wheels are examined by using Newton's second rule of motion along the horizontal (X-axis):

$$\sum F_{net,x} = m.a_x \quad (3.12)$$

$$M_B\, \ddot{X}_B = H_L + H_R \quad (3.13)$$

The forces operating on the wheels are examined by using Newton's second rule of motion along the vertical (Y-axis):

$$\sum F_{net,y} = m.a_y \quad (3.14)$$

$$M_B\, \ddot{y}_B = V_L + V_R - M_B g + \frac{\tau_L + \tau_R}{L}\sin(\theta_B) \quad (3.15)$$

The total of the moments around the center of the wheel is equal to:

$$\sum M_o = I\alpha \quad (3.16)$$

$$I_\theta \ddot{\theta}_B = (V_L + V_R)L\sin(\theta_B) - (H_L + H_R)L\cos(\theta_B) - (\tau_L + \tau_R) \quad (3.17)$$

$$X_B = L\sin(\theta_B) + \frac{X_{WL} + X_{WR}}{2} \quad (3.18)$$

$$y_B = -L\big(1 - \cos(\theta_B)\big) \quad (3.19)$$

Moment of inertia of chassis

$$I_B = \frac{1}{3} M_B L^2 \tag{3.20}$$

Defining tilt angles as

$$\theta = \theta_B = \theta_W = \theta_{WL} = \theta_{WR} \tag{3.21}$$

Average X position of the wheel

$$X_{WM} = \frac{X_{WL} + X_{WR}}{2} \tag{3.22}$$

And for the rotation

$$\delta = \frac{X_{WL} - X_{WR}}{D} \tag{3.23}$$

For the chassis rotation

$$I_\delta \ddot{\delta}_{WL} = \frac{D}{2}(H_L - H_R) \tag{3.24}$$

After solving Eq. (3.15) for $V_L + V_R$ and inserting Eq. (3.13) and Eq. (3.21) into Eq. (3.17)

$$I_\theta \ddot{\theta}_B = M_B L(\ddot{y}\sin(\theta) - \ddot{X}_B\cos(\theta)) + M_B g L(\ddot{y}\sin(\theta) - (\tau_L + \tau_R)(1 + \sin^2(\theta)) \tag{3.25}$$

Substitute Eq. (3.22) into Eq. (3.18) and differentiate it 2 time:

$$\ddot{X}_B = L\ddot{\theta}\cos(\theta) - L\dot{\theta}^2\sin(\theta) + \ddot{X}_{MW} \tag{3.26}$$

Similarly, differentiate y_B of Eq. (3.19) than

$$\ddot{y}_B = -L\ddot{\theta}\sin(\theta) - L\dot{\theta}^2\cos(\theta) \tag{3.27}$$

Multiply Eq. (3.26) by cos(θ) and multiply Eq. (3.27) by $\sin(\theta)$ and then subtract resulting equation from both equation:

$$\ddot{y}_B\sin(\theta) - \ddot{X}_B\cos(\theta) = -L\ddot{\theta} - \ddot{X}_{WM}\cos(\theta) \tag{3.28}$$

Substitute Eq. (3.28) and Eq. (3.20) into Eq. (3.25):

$$\frac{4}{3} M_B L^2 \ddot{\theta} + M_B L\cos(\theta)\ddot{X}_{MW} = M_B g L\sin(\theta) - (\tau_L + \tau_R)\left(1 + \sin^2(\theta)\right) \tag{3.29}$$

Substitute Eq. (3.11), Eq. (3.13) and Eq. (3.21) into Eq. (3.9):

$$M_W\left(\ddot{X}_{WL} + \ddot{X}_{WR}\right) = \left(\frac{\tau_L + \tau_R - \left(I_{WL}\ddot{\theta} + I_{WR}\ddot{\theta}\right)}{R}\right) - M_B\ddot{X}_B \tag{3.30}$$

Whereas

$$I_{WL} + I_{WR} = 2I_W \quad \text{then} \\ \ddot{X}_{WL} + \ddot{X}_{WR} = 2\ddot{X}_{WM} \tag{3.31}$$

$$2M_W\ddot{X}_{WM} = -M_B\ddot{X}_B + \frac{\tau_L + \tau_R}{R} - \frac{2I_W\ddot{\theta}}{R} \tag{3.32}$$

After solving for the left and right wheels, add the two equations from Equation (3.8).

$$I_{WL} + I_{WR} = R^2(M_{WL} + M_{WR}) \tag{3.33}$$

$$2I_W = R^2M_W \tag{3.34}$$

Substitute Eq. (3.26) and Eq. (3.34) into Eq. (3.32) yields

$$(M_BL\cos(\theta) + RM_W)\ddot{\theta} + (2M_W + M_B)\ddot{X}_{WM} = M_BL\dot{\theta}^2\sin(\theta) + \frac{(\tau_L + \tau_R)}{R} \tag{3.35}$$

Solve Eq. (3.29) for $\ddot{X}_{WM}$ and substitute into Eq. (3.35) than

$$\left((2M_W + M_B) - \frac{0.75(M_BL\cos(\theta) + M_WR)\cos(\theta)}{L}\right)\ddot{\theta} = \frac{0.75g(2M_W + M_B)\sin(\theta)}{L} - \\ 0.75M_B\sin(\theta)\cos(\theta)\,\dot{\theta}^2 - \left(\frac{0.75(2M_W + M_B)\left(1 + \sin^2(\theta)\right)}{M_BL^2} + \frac{0.75\cos(\theta)}{RL}\right)(\tau_L + \tau_R) \tag{3.36}$$

Solve Eq. (3.29) for $\ddot{\theta}$ and substitute into Eq. (3.35) than

$$\left(-0.75\frac{\cos(\theta)\,M_BL\cos(\theta) + M_WR}{L} + (2M_W + M_B)\right)\ddot{X}_{WM} \\ = -0.75(M_BL\cos(\theta) + M_WR)\frac{g\sin(\theta)}{L} + M_BL\sin(\theta)\dot{\theta}^2 - \\ \left(0.75\frac{(M_BL\cos(\theta) + M_WR)\left(1 + \sin^2(\theta)\right)}{M_BL^2} + \frac{1}{R}\right)(\tau_L + \tau_R) \tag{3.37}$$

Solve F_{fL} from Eq. (3.2)

$$F_{fL} = M_W\ddot{X}_{WL} + H_L \tag{3.38}$$

Differentiate Eq. (3.7)

$$\ddot{\theta}_{WL} = \frac{\ddot{X}_{WL}}{R} \tag{3.39}$$

Substitute Eq. (3.38) into Eq. (3.6) than

$$\begin{cases} H_L = \frac{\tau_L}{R} - \ddot{X}_{WL}\left(M_W + \frac{I_W}{R^2}\right) \text{ for the left wheel} \\ H_R = \frac{\tau_R}{R} - \ddot{X}_{WR}(M_W + \frac{I_W}{R^2} \text{ for the right wheel} \end{cases} \tag{3.40}$$

Subtract one Equation from another of Eq. (3.40) and then substitute in Eq. (3.23) than

$$H_L - H_R = \frac{\tau_L + \tau_R}{R} - D\ddot{\delta}\left(M_W + \frac{I_W}{R^2}\right) \tag{3.41}$$

Substitute Eq. (3.41) into Eq. (3.24) then

$$\left(I_\delta + \frac{1}{2}D^2\left(M_W + \frac{I_W}{R^2}\right)\right)\ddot{\delta} = \frac{1}{2}D\frac{\tau_L + \tau_R}{R} \tag{3.42}$$

Then

$$I_W = \frac{1}{3}M_B R^2 = I_\delta = \frac{1}{3}M_B\left(\frac{D}{2}\right)^2 = \frac{1}{12}M_B D^2 \tag{3.43}$$

Substitute Eq. (3.43) into Eq. (3.42) then

$$\ddot{\delta} = \frac{6}{(M_B + 9M_W)DR}(\tau_L + \tau_R) \tag{3.44}$$

Let

$$\begin{cases} U_1 = \tau_L + \tau_R \text{ and} \\ U_2 = \tau_L - \tau_R \end{cases} \tag{3.45}$$

Eq. (3.36), (3.37) and (3.44) are the fundamental system equations:

Let x1= = θ, x2 = $\dot{\theta}$, x3 = x, x4 = $\dot{x}$, x5 = δ and x6 = $\dot{\delta}$. State equations of the segway is rewritten as

$$\begin{cases} \dot{X}_1 = X_2 \\ \dot{X}_2 = f_1(X_1) + f_2(X_1, X_2) + g_1(X_1)(\tau_L + \tau_R) \\ \dot{X}_3 = X_4 \\ \dot{X}_4 = f_3(X_1) + f_4(X_1, X_2) + g_2(X_1)(\tau_L + \tau_R) \\ \dot{X}_5 = X_6 \\ \dot{X}_6 = g_3(\tau_L - \tau_R) \end{cases} \tag{3.46}$$

Where,

$$f_1(X) = \frac{\frac{(-0.75g\sin x_1)}{L}}{\left(0.75\frac{(M_W R + M_B L\cos x_1)(\cos x_1)}{(2M_W + M_B)L} - 1\right)}$$

$$f_1(X_1, X_2) = \frac{\frac{0.75M_B L(\sin x_1)(\cos x_1)}{(2M_W + M_B)L}(X_2)^2}{\left(0.75\frac{(M_W R + M_B L\cos x_1)(\cos x_1)}{(2M_W + M_B)L} - 1\right)}$$

$$g_1(X_1) = \frac{\left(\frac{0.75(1 + (\sin x_1)^2)}{M_B L^2} + \frac{0.75(\cos x_1)}{(2M_W + M_B)RL}\right)}{\left(0.75\frac{(M_W R + M_B L\cos x_1)(\cos x_1)}{(2M_W + M_B)L} - 1\right)}$$

$$f_3(X_1) = \frac{\frac{-0.75g\left(M_W R + M_B L(\cos x_1)\right)\sin(x_1)}{L}}{\left(2M_W + M_B - 0.75\frac{(M_W R + M_B L\cos x_1)(\cos x_1)}{L}\right)}$$

$$f_4(X_1, X_2) = \frac{\frac{M_B\sin(x_1)(X_2)^2}{L}}{\left(2M_W + M_B - 0.75\frac{(M_W R + M_B L\cos x_1)(\cos x_1)}{L}\right)}$$

$$g_2(X_1) = \frac{\left(\frac{0.75(M_W R + M_B L(\cos x_1))(1 + \sin(x_1)^2)}{M_B L^2} + \frac{1}{R}\right)}{\left(2M_W + M_B - 0.75\frac{(M_W R + M_B L\cos x_1)(\cos x_1)}{L}\right)}$$

$$g_3 = \frac{6}{(M_B + 9M_W)DR}$$

3.6.3 Model with Uneven Terrain

Modeling Terrain-Induced Variations in Forces on the Body

Consider disturbance Terms in the Eq. (3.15) Vertical Force Balance (Y-axis) updated as:

$$M_B\,\ddot{y}_B = V_L + V_R - M_B g + \frac{\tau_L + \tau_R}{L}\sin(\theta_B) + dy(t) \tag{3.47}$$

where dy(t) represents the disturbance due to uneven terrain affecting the vertical motion of the body.

Horizontal Force Balance (X-axis) Eq. (3.13) With Disturbance Term Updated as:

$$M_B\,\ddot{X}_B = H_L + H_R + dx(t) \tag{3.48}$$

where dx(t) represents the disturbance due to uneven terrain affecting the horizontal motion of the body

Modeling Terrain-Induced Variations in Forces on the Wheels

Vertical Ground Reaction Forces VL and VR updated hence Terrain affects the ground reaction forces on the wheels:

$$\begin{aligned} V_L &\rightarrow V_L + \Delta V_L(t) \\ V_R &\rightarrow V_R + \Delta V_R(t) \end{aligned} \tag{3.49}$$

Where $\Delta V_L(t)$ and $\Delta V_R(t)$ represent variations due to the uneven terrain affecting the wheels.

Also, for Horizontal Ground Reaction Forces H_L and H_R:

$$\begin{aligned} H_L &\rightarrow H_L + \Delta H_L(t) \\ H_R &\rightarrow H_R + \Delta H_R(t) \end{aligned} \tag{3.50}$$

Where $\Delta H_L(t)$ and $\Delta H_R(t)$ are variations caused by the terrain.

Similarly, for the chassis rotation update Eq. (3.24)

$$I_\delta \ddot{\delta}_{WL} = \frac{D}{2}\left[\left(H_L + \Delta H_L(t)\right) - \left(H_R + \Delta H_R(t)\right)\right] + d\delta(t) \tag{3.51}$$

where $d\delta(t)$ is the disturbance affecting the yaw angle dynamics due to uneven terrain.

Finally arranging and substituting those equations the overall system model with uneven terrain will be:

$$\begin{cases} \dot{X}_1 = X_2 \\ \dot{X}_2 = f_1(X_1) + f_2(X_1, X_2) + g_1(X_1)(\tau_L + \tau_R) + dy(t) \\ \dot{X}_3 = X_4 \\ \dot{X}_4 = f_3(X_1) + f_4(X_1, X_2) + g_2(X_1)(\tau_L + \tau_R) + dx(t) + \dfrac{\Delta H_L(t) + \Delta H_R(t)}{M_W} \\ \dot{X}_5 = X_6 \\ \dot{X}_6 = g_3[(\tau_L - \tau_R) + \left(\Delta H_L(t) - \Delta H_R(t)\right)] + d\delta \end{cases} \tag{3.52}$$

Table 3.1: Parameter of segway values and unit (khan et al., 2022; Son & Anh, 2014b)

Symbol	Value	Description
θ	Degree	Balance angle
β	Degree	Direction angle
M_w	7kg	Wheel mass
M_r	90kg	Rider's mass
R	0.2m	wheel radius
L	1m	Distance b/n the segway 's center of gravity and Z-axis
D	0.6m	Distance b/n contact of the wheels
g	9.81m/s^2	acceleration due to gravity

This updated model reflects the impact of uneven terrain on both the body and wheel dynamics of the Segway. The terrain-induced disturbances are now incorporated into the equations governing the vertical and horizontal motion, as well as the yaw dynamics, ensuring that both the body and wheels respond appropriately to variations in terrain.

3.6.4 DC Motor Modeling

In control systems, DC motors are commonly used actuators that provide direct rotary motion. When combined with wheels, they can also generate translational motion. The free-body diagrams for the rotor and armature circuits are shown in Figure 3.6.

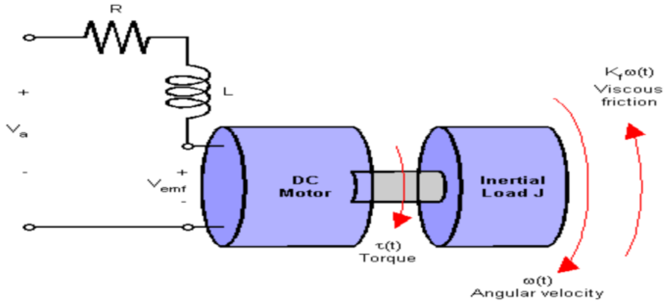


Figure 3.5: DC motor(*DC Motor Control - MATLAB & Simulink Example*, n.d.)

The dynamics of DC motor can be expressed as the following equations by applying Kirchoff’s Voltage law

$$V_a = R_a i_a + L_a \frac{d}{dt} i_a + V_e \tag{3.53}$$

Armature current and magnetic field strength are directly correlated with electric motor torque.as shown below, the armature current determines the motor torque alone, with a constant factor Km.

$$\tau_m = K_m i_a \tag{3.54}$$

K_e is constant factor determines how the back emf is related to the shaft's angular motion.

$$V_e = K_e \omega_m \tag{3.55}$$

By apply Newton's law to the motor system

$$\tau_m = I_R \frac{d}{dt} \omega_m + K_f \omega_m + \tau_a \tag{3.56}$$

Substitute Eq. (3.55) into Eq. (3.53) and solve ia of Eq. (3.54) and substitute again into Eq. (3.53) then

$$V_a = \frac{R_a}{K_m} \tau_m + \frac{L_a}{K_m} \frac{d}{dt} \tau_m + K \omega_m \tag{3.57}$$

Then taking the Laplace transform of Eq. (3.56) and Eq. (3.57)

$$V_a(s) = \frac{R_a}{K} \tau_m(s) + \frac{L_a}{K} S \tau_m(s) + K \omega_m(s) \tag{3.58}$$

$$\tau_m = I_R S \omega_m(s) + K_f \omega_m(s) + \tau_a \tag{3.59}$$

Finaly rearranging and substituting Eq. (3.59) to Eq. (3.58)

$$V_a(s) = \frac{R_a}{K} \tau_m(s) + \frac{L_a}{K} S \tau_m(s) + \frac{K}{I_R S + K_f} \tau_m(s) \tag{3.60}$$

This can be the voltage for the left and right motor as

$$V_{aL}(s) = \frac{R_a}{K} \tau_{mL}(s) + \frac{L_a}{K} S \tau_{mL}(s) + \frac{K}{I_R S + K_f} \tau_{mL}(s) \tag{3.61}$$

$$V_{aR}(s) = \frac{R_a}{K} \tau_{mR}(s) + \frac{L_a}{K} S \tau_{mR}(s) + \frac{K}{I_R S + K_f} \tau_{mR}(s) \tag{3.62}$$

Assuming

$$\begin{aligned} U_1 &= \tau_{mL}(s) + \tau_{mR}(s) \\ U_2 &= \tau_{mL}(s) - \tau_{mR}(s) \end{aligned} \tag{3.63}$$

Then relating the motor dynamics with segway resulting

$$\begin{cases} U_1 = \dfrac{K(I_R S + K_f)}{R_a(I_R S + K_f) + L_a S(I_R S + K_f) + K^2}(V_{aL} + V_{aR}) \\ U_2 = \dfrac{K(I_R S + K_f)}{R_a(I_R S + K_f) + L_a S(I_R S + K_f) + K^2}(V_{aL} - V_{aR}) \end{cases} \tag{3.64}$$

Table 3.2: DC Motor parameter (9.5 Inch Self Balancing; Segway - Wikipedia, n.d.)

DC Motor Parameter	Description	Unit
Ra	Armature Resistance	2.5ohms
La	Armature Inductance	0.00055H
Va	Voltage on Armature	V
Ve	Back Emf	V
K_m	torque constant	0.58Nm/A
K_e	Motor back Emf constant	0.58Nm/A
Ir	Moment inertia of the motor	$0.02Kgm^2/s^2$

3.7 System Decoupling

System decoupling in control engineering refers to the process of designing control systems such that the influence or coupling between different variables or components of the system is minimized or eliminated. This is achieved by implementing control strategies or mechanisms that ensure changes or disturbances in one part of the system do not significantly affect other parts, thus enhancing system stability, performance, and robustness.

The state-space representation of the segway suggests that it is a heavily coupled nonlinear system. the system's separation into two independent subsystems. To translate U_θ and U_β into the voltages of the wheel motors on the left (V_{aL}) and right (V_{aR}), use the decoupling transformation below.

Assume

$$\begin{cases} V_\theta = V_L + V_R \\ V_\beta = V_L - V_R \end{cases}$$

This shows that

$$\begin{cases} V_L = \frac{1}{2}(V_\theta + V_\beta) \\ V_R = \frac{1}{2}(V_\theta - V_\beta) \end{cases} \tag{3.65}$$

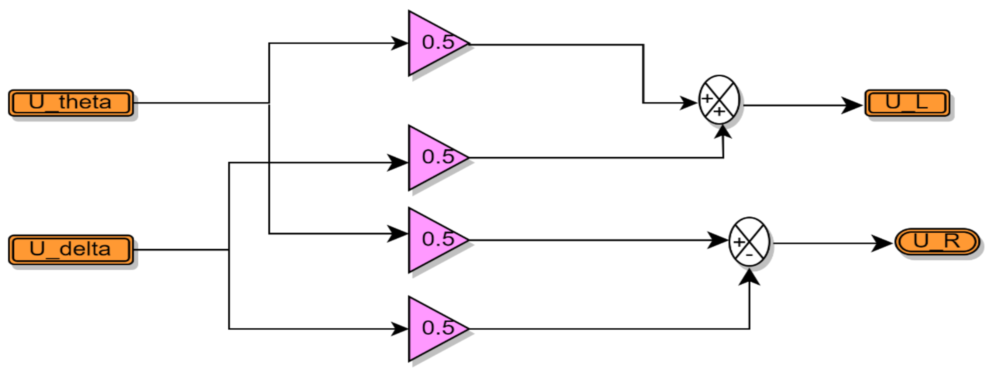


Figure 3.6: Decoupling circuit for the System

3.8 Controller Design

3.8.1 Sliding Mode Control

An appealing feedback control that can ensure the stability of nonlinear systems is sliding mode control. The fundamental principle of SMC is to use discontinuous control input to push a nonlinear system's state trajectory along a predefined surface in the state space. This surface, known as a sliding manifold, symbolizes the required plant dynamics' characteristics, like tracking ability and stability to the origin(Bin Xing et al., 2016).

SMC can happen in a dynamic system that is driven by right-hand discontinuous ordinary differential equations. It is possible for the control to fluctuate rapidly as a result of the system state. We refer to this motion as sliding mode.

Sliding mode control has two key advantages: it makes the closed loop response entirely immune to certain particular uncertainties and allows for the modification of the system's dynamic behavior by choosing a certain sliding function. This theory also applies to bounded non-linearities, disturbances, and uncertainty in model parameters (VADIM I. UTKIN, n.d.). Following the first reaching stage, the system states move in a linear fashion along S = 0. The specific S = 0 surface was selected because, when restricted to it, it exhibits favorable reduced-order dynamics.

There are two steps in the sliding mode controller design.

1. creating a sliding surface with the intention of eliciting a desired system response from the plant that is constrained to it. This suggests that in order to meet the switching surface equation, the state variables of the plant dynamics are confined.

2.. Drive the plant's state trajectory towards the sliding surface by building switched feedback gains. The generalized Lyapunov stability theory serves as the foundation for these creations.

The four classical reaching laws are:

a. constant rate reaching law

b. Exponential reaching law

c. Power rate Reaching law and

d. General reaching law

The sliding mode controller is based on the reaching law and includes both the reaching and sliding phases. The reaching phase drive system to stable manifold while the sliding phase turn system slides to equilibrium.

By applying the general equation proposed by Slotine and Li (Slotine & Li, 2005), we can derive the sliding surface as follows:

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \tag{3.66}$$

λ - is sliding gain, its value is a positive real and n - is the dimension of sliding surface.

3.8.1.2 Super Twisting Sliding Mode Control

HOSM technique offers an alternative approach to eliminating chattering. Typically, a sliding mode control signal comprises two components: one that focuses on the dynamical systems and sliding surfaces, and another that ensures the system dynamics remain on the sliding surface through switching control.

As a result, the sliding surface is defined as the following:

$$s = \lambda e + \dot{e} \tag{3.67}$$

The SMC is given by:

$$u = u_{eq} + u_s \tag{3.68}$$

Where $U_{eq,}$ the equivalent control law, when system uncertainty and external disturbances are not considered, aims to maintain control over the sliding surface variables.

3.8.1.3 **Design Super Twisted Sliding Mode Control Law**

First for balancing of segway and then direction the equivalent term is designed

Considering that the sliding surface as (3.67)

$$\begin{cases} s = \lambda e + \dot{e} \\ \dot{s} = \lambda \dot{e} + \ddot{e} \end{cases} \tag{3.69}$$

Where s-is sliding surface and e as error.

$$\begin{cases} e_\theta = x_1 - x_{1ref} \\ e\delta = x_5 - x_{5ref} \end{cases} \tag{3.70}$$

First let's solve for tilt angle there fore

$$\begin{cases} \dot{e}_\theta = \dot{x}_1 - \dot{x}_{1ref} = x_2 - \dot{x}_{1ref} \\ \ddot{e}_\theta = \ddot{x}_1 - \ddot{x}_{1ref} = \dot{x}_2 - \ddot{x}_{1ref} \end{cases} \tag{3.71}$$

This shows that

$$\begin{cases} S_\theta = i\left(x_1 - x_{1ref}\right) + \left(\dot{x}_1 - \dot{x}_{1ref}\right) \\ \dot{S}_\theta = i\left(\dot{x}_1 - \dot{x}_{1ref}\right) + \left(\dot{x}_2 - \ddot{x}_{1ref}\right) \end{cases} \tag{3.72}$$

Hence

$$\dot{x}_2 = f_1(x_1) + f_2(x_1, x_2) + g_1(x_1)U_1 + dy(t) \tag{3.72}$$

$$\dot{S}_\theta = i\left(\dot{x}_1 - \dot{x}_{1ref}\right) + f_1(x_1) + f_2(x_1, x_2) + g_1(x_1)U1 + dy(t) - \ddot{x}1ref \tag{3.74}$$

Then to ensure stability by applying Lyapunov

$$\begin{cases} V_\theta = \dfrac{1}{2} S_\theta^2 \\ \dot{V}_\theta = S_\theta \dot{S}_\theta \end{cases} \tag{3.75}$$

Substituting equation (3.74) in to (3.75) and solving for $S\dot{S}$

$$\dot{V}_{\theta} = S_{\theta}\dot{S}_{\theta} = S_{\theta}\left(i\left(\dot{x}_1 - \dot{x}_{1ref}\right) + f_1(x_1) + f_2(x_1, x_2) + g_1(x_1)U1 + dy(t) - \ddot{x}1ref\right) \tag{3.76}$$

To ensure the balancing of segway stable on the surface, the Lyapunov stability criterion must be met, meaning the condition $\dot{V} = S_{\theta}\dot{S}_{\theta} < 0$ must be satisfied. The equivalent term is designed accordingly.

$$U_{1eq} = \frac{-i\left(\dot{x}_1 - \dot{x}_{1ref}\right) - f_1(x_1) - f_2(x_1, x_2) - dy(t) + \ddot{x}1ref}{g_1(x_1)} \tag{3.77}$$

Second for direction control defining error and sliding surface as

$$\begin{cases} \dot{e}_{\delta} = \dot{x}_5 - \dot{x}_{5ref} = x_6 - \dot{x}_{5ref} \\ \ddot{e}_{\delta} = \dot{x}_5 - \dot{x}_{5ref} = \dot{x}_6 - \dot{x}_{5ref} \end{cases} \tag{3.78}$$

This shows that

$$\begin{cases} s_{\delta} = j(x_5 - x_{5ref}) + (\dot{x}_5 - \dot{x}_{5ref}) \\ \dot{s}_{\delta} = j(\dot{x}_5 - \dot{x}_{5ref}) + (\ddot{x}_5 - \ddot{x}_{5ref}) \end{cases} \tag{3.79}$$

But

$$\dot{x}_6 = g_3(x_1)[(U_2 + \Delta H_L(t) - \Delta H_R(t)] + d\delta(t) \tag{3.80}$$

$$\dot{S}_{\delta} = j\left(x_6 - \dot{x}_{5ref}\right) + \left(g_3(x_1)[U_2 + \Delta H_L(t) - \Delta H_R(t)]\right) + d\delta(t) - \ddot{x}_{5ref} \tag{3.81}$$

Then to ensure stability by applying Lyapunov

$$\begin{cases} V_{\delta} = \frac{1}{2}S_{\delta}^2 \\ \dot{V}_{\delta} = S_{\delta}\dot{S}_{\delta} \end{cases} \tag{3.82}$$

Substituting equation (3.81) in to (3.82) and solving for $S\dot{S}$

$$\dot{V}_{\delta} = S_{\delta}\dot{S}_{\delta} = S_{\delta}\left[j\left(x_6 - x_{5ref}\right) + \left(g3(x1)\left[(U2) + \left(\Delta H_L(t) - \Delta H_R(t)\right)\right] + d\delta - \ddot{x}_{5ref}\right)\right] \tag{3.83}$$

To ensure the direction control of segway stable on the surface, the Lyapunov stability criterion must be met, meaning the condition $\dot{V}_{\delta} = S\dot{S} < 0$ must be satisfied. The equivalent term is designed accordingly.

$$U_{2eq} = \frac{-j\left(x_6 - \dot{x}_{5ref}\right) - g_3(x_1)\left(\Delta H_L(t) - \Delta H_R(t)\right) - d\delta + \ddot{x}_{5ref}}{g_3(x_1)} \tag{3.84}$$

The STSMC switching term given as

$$\begin{cases} U_s = -\lambda_1 |S|^{\frac{1}{2}} \mathrm{sgn}(s) + v \\ \dot{v} = -\lambda_2 \mathrm{sgn}(S) \end{cases} \tag{3.85}$$

Adding this into the equivalent terms the overall control input signals can be expressed as

$$\begin{cases} U_1 = \dfrac{-i(\dot{x}_1 - \dot{x}_{1ref}) - f_1(x_1) - f_2(x_1, x_2) - dy(t) + \ddot{x}1ref}{g_1(x_1)} \\ \qquad -i_1 |S_\delta|^{\frac{1}{2}} \mathrm{sgn}(S_\theta) - i_2 \int \mathrm{sgn}(S_\theta) \\ U_2 = \dfrac{-j(x_6 - x_{5ref}) - g3(x1)(\Delta H_L(t) - \Delta H_R(t))] - d\delta + \ddot{x}_{5ref})}{g3(x1)} \\ \qquad -j_1 |S_\delta|^{\frac{1}{2}} \mathrm{sgn}(S_\delta) - j_2 \int \mathrm{sgn}(S_\delta) \end{cases} \tag{3.86}$$

The controller developed for the segway is tested using MATLAB 2023a/Simulink. The parameters of the STSMC, namely i, i1, i2, j, j1 and j2 are optimized using PSO, and TLBO. The STSMC is used to manage the balance and direction of the Segway.

3.9 Objective Function

An objective function is a mathematical formula used in optimization problems to describe the goal that needs to be achieved. The crucial first step in applying any optimization method is choosing the objective functions that will evaluate the fitness of each population member (particle). Here are some examples of objective functions.

$$IAE = \int_0^t |e(\tau)| \, d\tau \tag{3.87}$$

$$ITAE = \int_0^t \tau * |e(\tau)| \, d\tau \tag{3.88}$$

$$ISE = \int_0^t e(\tau)^2 \, d\tau \tag{3.89}$$

Hence the proposed system has two state variables to be controlled. There will be 2 error signals as balance angle and direction angle error. Therefore, the above equation can be rewritten as

$$ITAE = \int_0^t \tau * (|e_1(\tau) + e_2(\tau)|) d\tau \tag{3.90}$$

Where e_1 and e_2 are the error signals of balancing angle and direction angle difference respectively.

3.10 Optimization Algorithms

This section provides a brief introduction to TLBO and PSO. These algorithms are used to tune the STSMC controllers for balancing and steering the Segway. Optimal controller design is a process in engineering and control systems that focuses on developing a controller to achieve the best possible performance for a given system. The aim is to devise control strategies that ensure the system behaves as desired, reducing errors and enhancing efficiency.

In this research 3 d/t optimization techniques are used to find the best values for gains namely i1, i2, i3, j1, j2 and j3.

3.10.1 Teaching Learning Based Optimization

The TLBO algorithm is a teaching-learning process inspired algorithm proposed by Rao et al(Rao, 2016). and Rao and Savsani developed the algorithm based on the impact of a teacher's influence on the performance of learners in a class. It describes two fundamental modes of learning:

- via teacher (sometimes referred to as teacher phase) and
- via communication with fellow students (sometimes referred to as the learner phase).

Figure 3.7: teacher phase (left)and Learner phase (right) (Https://Www.Dreamstime.Com.)

The population of this optimization method is a group of learners, and the diverse subjects that are made available to them are considered as distinct design variables in the optimization problem. The 'fitness' value of the optimization problem is correlated with the learner's outcome. The teacher is chosen as the finest option out of all the candidates. The objective function of the optimization problem is represented by the design variables, and the optimal value of that objective function is the best solution.(Taheri et al., 2021).

A metaheuristic that was first presented eleven years ago, the TLBO algorithm works well for resolving optimization issues. Extending its application areas and improving its performance have been the main research goals. Various hybrid and variant forms have been developed to increase its efficacy. The original TLBO algorithm is still quite effective and has been successfully used in many different optimization fields despite these advancements (Gómez Díaz et al., 2022).

Teacher Phase

In the Teacher Phase of the TLBO (Teaching-Learning-Based Optimization) algorithm, the following steps are typically performed:

1. Select the Teacher: The individual with the best fitness value in the current population is selected as the "teacher." This individual represents the optimal solution so far and will guide the other learners in improving their solutions.
2. Compute the Mean Knowledge: The mean knowledge level of the population is calculated. This is the average of the fitness values (or objective function values) of all learners in the population. It provides a benchmark for the overall performance of the group.
3. Update Learners' Solutions: For each learner, a new solution is generated by adjusting their current solution based on the teacher's solution and the mean solution. The update formula is:

$$X_{new} = X + rand * \left(X_{best} - T_f * X_{mean}\right) \tag{3.91}$$

Here:

X is the learner’s existing solution.

X_{best} is the best solution found so far (the teacher).

X_{mean} is the average solution of the entire population.

r is a random number between 0 and 1, introducing randomness into the update process.

T_f is the teaching factor, which can be either 1 or 2. It controls how strongly the teacher influences the learners. A higher T_f means a stronger influence from the teacher.

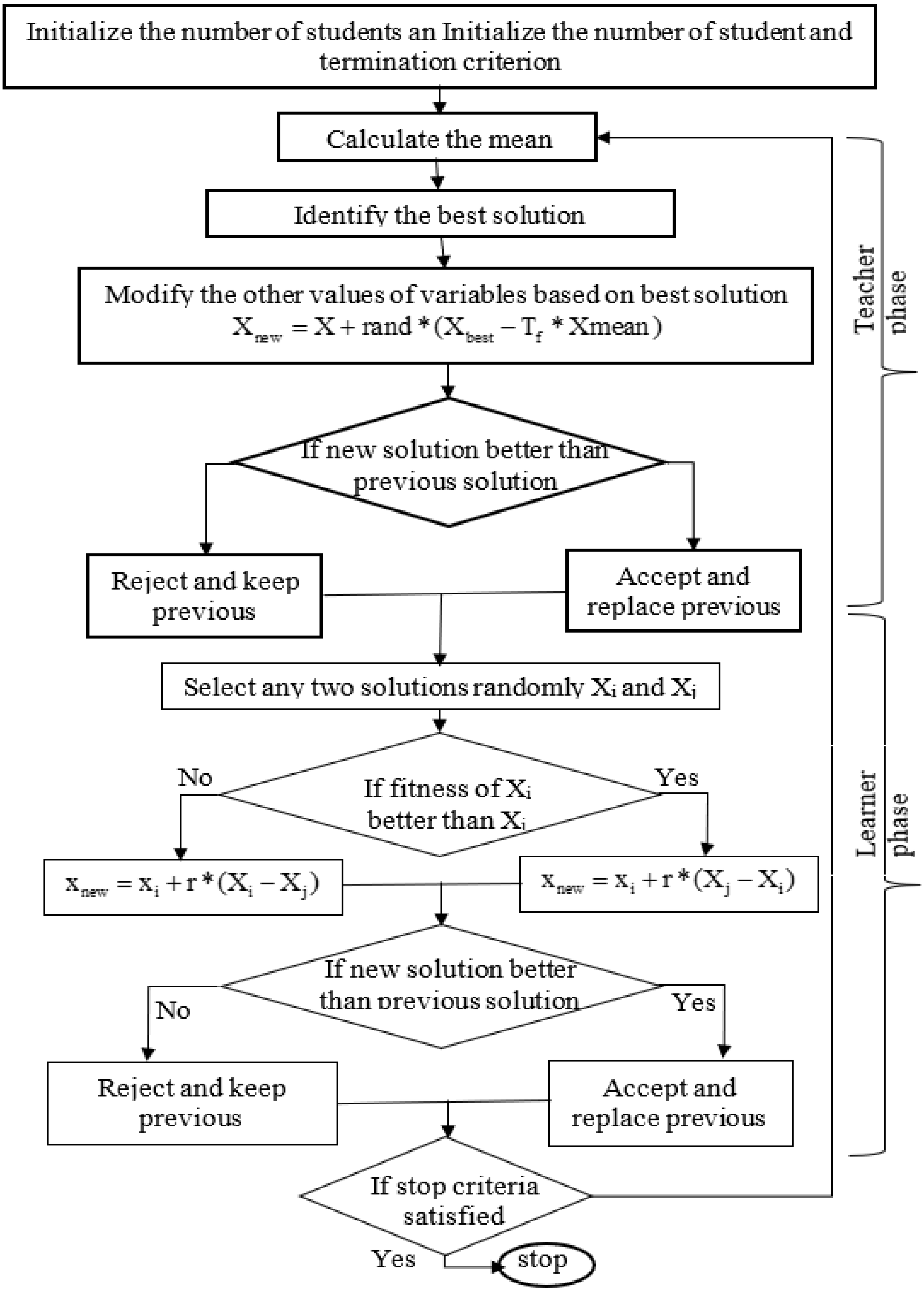


Figure 3.8: TLBO flow chart

This update process aims to improve each learner's solution by moving it closer to the teacher's solution, while also considering the mean knowledge of the population to ensure diversity and avoid premature convergence.

4. Evaluate New Solutions: After generating new solutions through the update process, they are evaluated to determine their fitness values. The learners' solutions are then updated based on whether these new solutions are better than the previous ones.
5. Selection of Improved Solutions: If the new solution of a learner improves upon their previous solution, it replaces the old one. This helps in refining the solutions over successive iterations.

The Teacher Phase is crucial in the TLBO algorithm because it leverages the best solution in the population to guide and enhance the performance of other learners. This collective learning approach helps in exploring the search space more effectively and drives the population towards optimal solutions by continuously improving their knowledge based on the best-performing individual.

Learner Phase

In the Learner Phase of the TLBO (Teaching-Learning-Based Optimization) algorithm, the focus shifts from the influence of the teacher to the interaction between learners. Here’s a detailed description of the Learner Phase:

1. Pairwise Interaction: Each learner in the population is paired with another learner. The pairing is done randomly, ensuring diverse interactions between learners.
2. Generate New Solutions: For each learner pair, the new solutions are generated using the following steps:

   a. Determine the Better Learner: For each pair, compare the fitness values of the two learners to identify the better-performing one.

   b. Update the Learner’s Solution: Each learner in the pair updates their solution based on the difference between their own solution and the solution of the better learner. The update formula is:

$$X_{new} = X \pm rand\left(X - X_p\right) \tag{3.92}$$

Here:

X is the solution of the learner being updated.

X_p is the solution of the learner in the pair randomly selected student.

rand is a random number between 0 and 1, introducing randomness in the update process.

The goal is to improve each learner's solution by moving it towards the solution of a better-performing peer. This promotes diversity and helps explore different regions of the search space.

3. Evaluate New Solutions: The updated solutions are then evaluated to determine their fitness values. The learner's solution is updated if the new solution provides a better fitness value.
4. Selection of Improved Solutions: Similar to the Teacher Phase, if the newly generated solution is better than the previous one, it replaces the old solution. This process helps in refining the solutions based on peer interactions.

The Learner Phase enhances the optimization process by encouraging collaboration among learners. It allows each learner to benefit from the knowledge of better-performing peers, which helps in exploring and exploiting the search space more effectively. This phase complements the Teacher Phase by focusing on peer-to-peer learning and further improves the overall performance of the algorithm.

3.10.2 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm is a computational method used to solve optimization problems by simulating the social behavior of birds or fish. It was introduced by Eberhart and Kennedy in 1995. In recent years, the PSO algorithm has been applied to a wide variety of fields.(Dziwiński et al., 2020; Önen, 2017)

Finding locations in search space that are close to the global minimum or maximum solution (s) is accomplished by an algorithm. The PSO algorithm contains numerous parameters. Each particle's current position is used to calculate the fitness value at that location. The three characteristics for each particle are position (xi ∈ Rn), velocity (i.e., rate of position change; vi), and previous best positions (pi). Moreover, the particle with the highest fitness value is said to be in the global best location.

Every particle is represented by a set of coordinates in the search space called xi. During the search phase, the fitness function is utilized to assess each particle's current position. The best location is then saved in the previous best positions (pi), and each particle's fitness value is compared to its current position after that. The following equation was used to update the particle's velocity and compute its position (vi, xi).

$$V_i(t+1) = \omega V_i^{\,t} + c_1 r_1({p_i}^t - {x_i}^t) + c_2 r_2({g_i}^t - {x_i}^t) \tag{3.93}$$

$$x_i^{\,t+1} = x_i^{\,t} + V_i^{\,t+1} \tag{3.94}$$

Where, $V_i^{\,t}$ is the individual velocity at iteration r, ω is the weight of inertia parameter, c_1, c_2 are the acceleration coefficient, r_1, r_2 are the random number within [0,1], $x_i^{\,t}$ is individual position i at iteration t, and p_i, g_i are best individual position and global best position at iteration t.

Determine all PSO parameters;
Initialize position and velocity randomly;
Assign p_{best} and g_{best} in the particles;
While ($t < M$) (M is maximum iteration)
    Find fitness value (J) from new particle x_i;
    If cost of x_i is better than p_{best}
        Substitute p_{best} by x_i;
    End
    If V_i is better than g_{best}
        Replace g_{best} by x_i;
    End
    For i : S (S is number of particles)
        Update x_i and V_i of particle using equation;
        $V_i(t+1) = \omega V_i^{\,t} + c_1 r_1({p_i}^t - {x_i}^t) + c_2 r_2({g_i}^t - {x_i}^t)$;
        $x_i^{\,t+1} = x_i^{\,t} + V_i^{\,t+1}$;
    End
    $t = t + 1$;
End

Figure 3.9: PSO algorithm pseudo-code

To modify the control parameters in relation to the suggested cost function (J), the TLBO and PSO are used. Table 3.3 shows the set tuning ranges for the parameters, and Table 3.4 lists the parameters of the suggested methods.

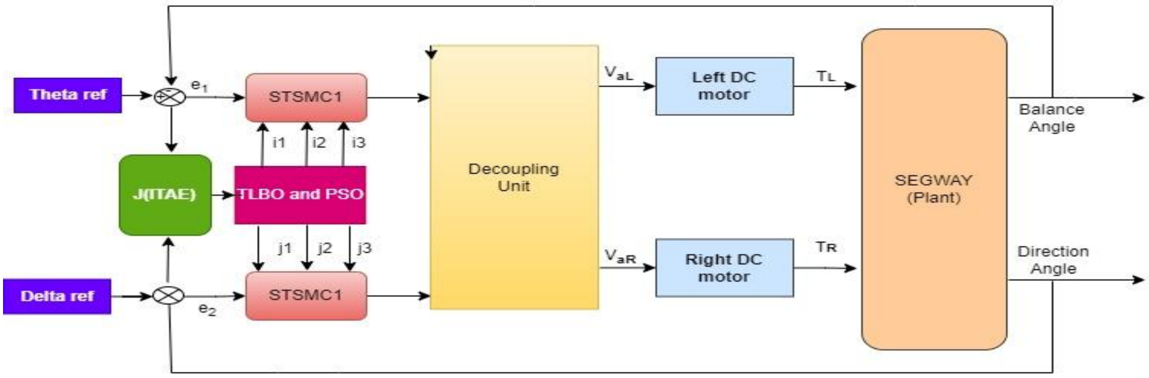


Figure 3.10: system structure for optimization.

Table 3.3: Proposed optimization algorithm parameters

Algorithm	Parameters		
TLBO	No Learners		50
	Max Generations		20
PSO	No particles		50
	Maximum iteration		20
	Inertia weight	ωinit	0.2
		ωfinal	0.9
	Personal best value (c1)		2
	Neighborhood best value (c2)		2

Table 3.4: Boundary of STSMC Gains

Boundary	**STSMC1**			**STSMC2**		
	i1	i2	i3	j1	j2	j3
Lower bound (LB)	0.05	0.05	0.05	0.05	0.05	0.05
Upper bound (UB)	15	15	15	15	15	15

Figure below shows the convergence of TLBO and PSO.TLBO is superior at the beginning and the end as this graph demonstrates. PSO converges more slowly than TLBO in both balance direction control. Consequently, the TLBO outperforms the PSO in response by minimizing the ITAE more.

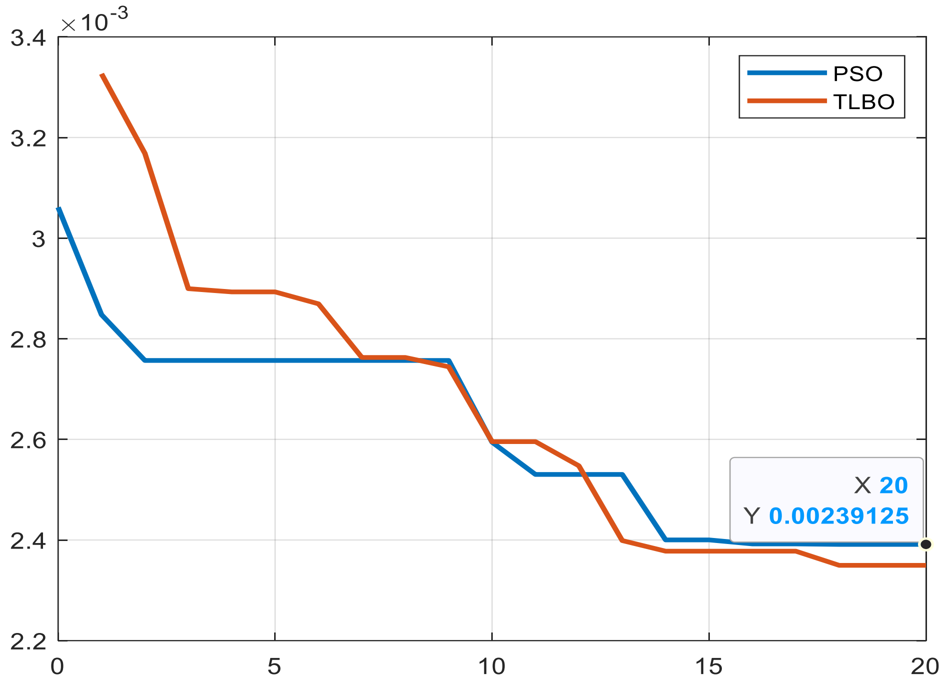
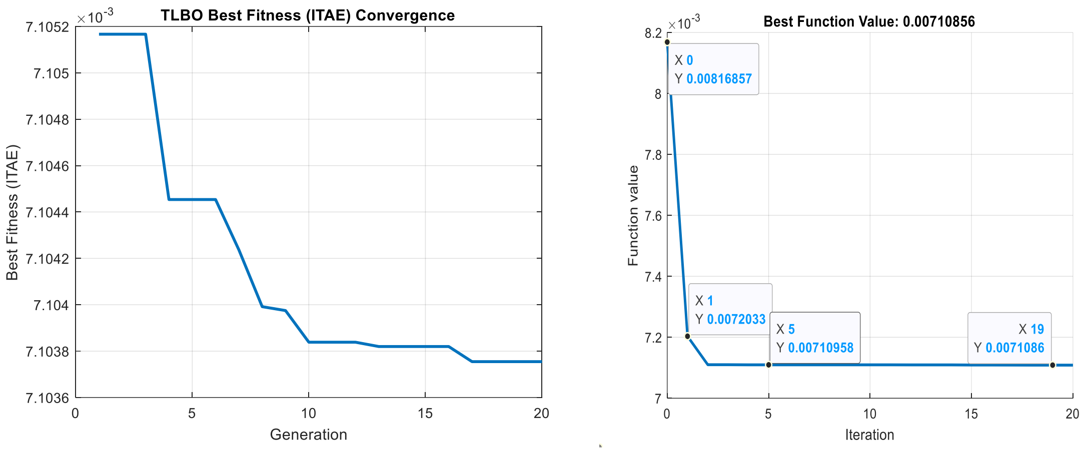


Figure 3.11: Convergence curve of TLBO and PSO for balance angle



(a) (b)

Figure 3.12: Convergence curve of TLBO and PSO for direction control

Table 3.5: Tuned gain values

Controller Gain Coefficients		TLBO	PSO
STSMC1	i_1	1.0417	1.5390
	i_2	13.7703	13.3895
	i_3	14.9872	14.9484
STSMC2	j_1	9.4074	9.4191
	j_2	15	15
	j_3	15	15
Optimal value (ITAE)		**(2.34964+7.1035) e-3**	**(2.39125+7.10856) e-3**
Elapsed time		8148.437543	5790.863257

The TLBO is superior to PSO in response to best fitness value (minimum ITAE) while PSO has better elapsed time to reach optimal value.

3.11 Fuzzy Logic Controller

A fuzzy logic controller is a way of representing, manipulating, and executing a human's heuristic understanding on how to control a system. Fuzzification, inference system, rules, and defuzzification are the four key components of the FLC (Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, n.d.) as shown in below diagram.

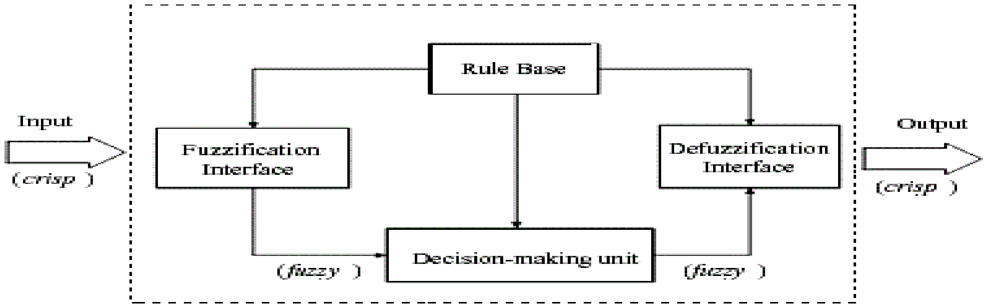


Figure 3.13: Basic component and architecture of FLC system (P.k.Mani KANNAN |)

3.12 ANN Control

An artificial neural network (ANN) is a computational model inspired by the human brain, consisting of interconnected layers of nodes (neurons) that process information. Each neuron receives inputs, applies a mathematical function, and passes the output to the next layer. ANNs can learn from data through training, making them effective for tasks like pattern recognition, classification, and prediction. Their structure allows them to model complex relationships and find patterns in large datasets, making them essential tools in machine learning and artificial intelligence.

An artificial neural network is a system composed of interconnected neurons arranged in various layers, each performing basic mathematical operation(Zupan, 1994). However, when a large number of neurons are networked together, the system gains enhanced capabilities and can accomplish tasks that would otherwise be impossible.

The ANN controller has the following main advantages (Callinan, 2003).

• It can perform tasks that a linear program can't.

• Because of its parallel nature, when the neural network element fails, it can continue without any problem.

• It learns from experience and does not need to be reprogrammed.

• It can be implemented in any application.

• It may require less tuning effort as compared to conventional controller.

Architecture of ANN

An artificial neural network is made up of several artificial neurons, also known as units, stacked one layer on top of the other. Artificial neural networks have many kinds of layers, but the input, hidden, and output layers are the three main types. Which are shown in figure 3-15 (Zupan, 1994).

Input Layer: The input layer communicates with the external environment and gives a pattern to the neural network. Each input neuron should represent some independent variable that affects the neural network's output.

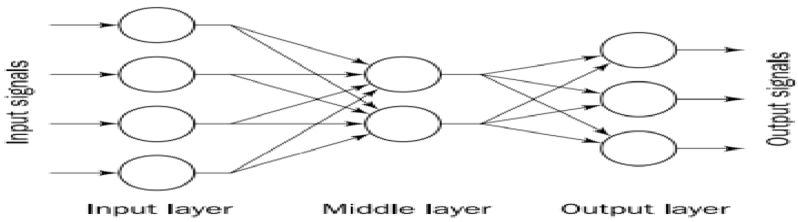


Figure 3.14: Architecture of artificial neural network

Hidden Layer: This layer lies in between the input and output layers and is made up of a group of neurons that have an activation function applied to them. Processing the inputs derived from the input layer is its purpose.

Output Layer: An artificial neural network's output layer gathers and sends data in the manner for which it was intended to. It is possible to link the output layer's pattern directly to the input layer.

The smallest processing unit in an artificial neural network is called an artificial neuron. It computes the outcome after applying a function to a variety of inputs. As seen in the picture below, the fundamental components of an artificial neuron are an activation function and a weighted adder. (Zupan, 1994).

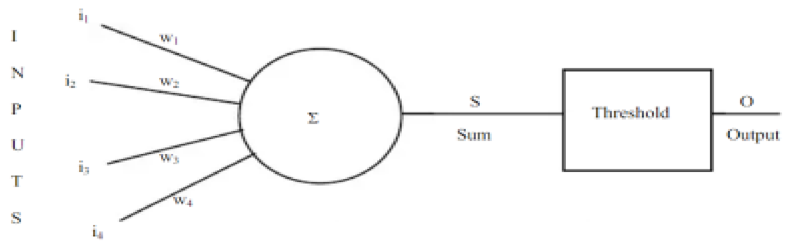


Figure 3.15: Basic artificial neuron structure.

3.12.1 Neural Network Model for Estimating Direction Control STSMC

Using MATLAB software, the neural network model demonstrates several layers of neurons with nonlinear transfer functions, enabling the network to learn both nonlinear and linear correlations between the input and matching output values. The two-layer feedforward network configuration with predefined 2 inputs and 20 hidden neurons is shown in figure 3-16.

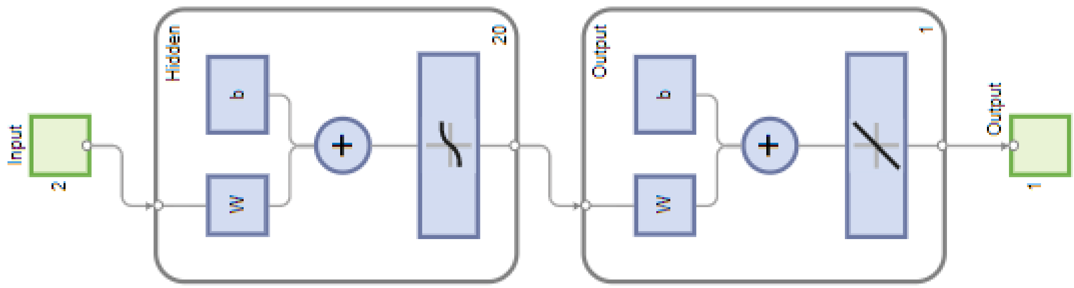


Figure 3.16: Artificial neural network MATLAB model.

The network is ready for training, and to perform the training process, a set of data is required. In this thesis work, the training data are x6, and angle error 2, which are obtained after simulating the system in MATLAB software. The input dataset is split into three classes: training data used for training, validation data for measuring network generalization and testing data for an independent measure of network performance. The biases and weights of the network are adjusted iteratively during the training process to minimize the network output error. The mean square error for different epochs at the time of the training process is shown in figure 3-17.

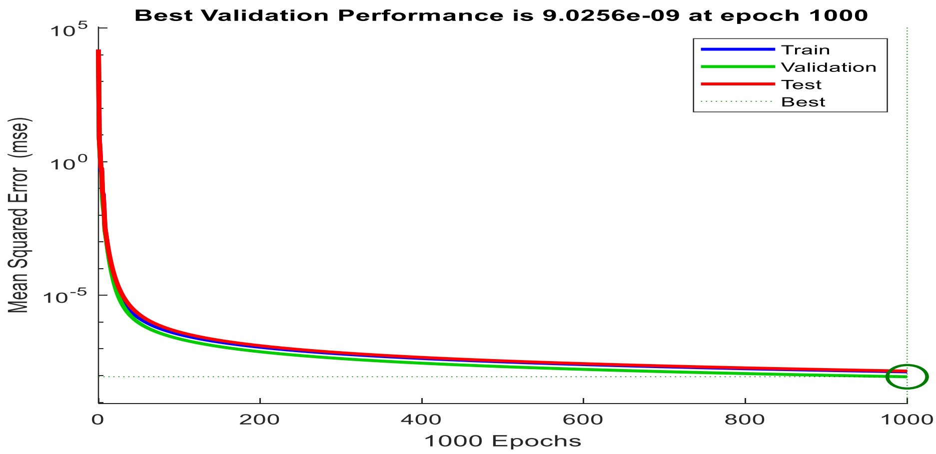


Figure 3.17: ANN training result in MATLAB

To test the performance of the network response, the testing datasets are put through the network and will perform a linear regression between the network outputs and the corresponding targets.

In a regression plot, the value of R indicates the relationship between the network's expected output and the desired output. Figure 3.18 shows the designed ANN controller regression plot, and the R value of the figure indicates a very good relationship between the target and predicted output of the network.

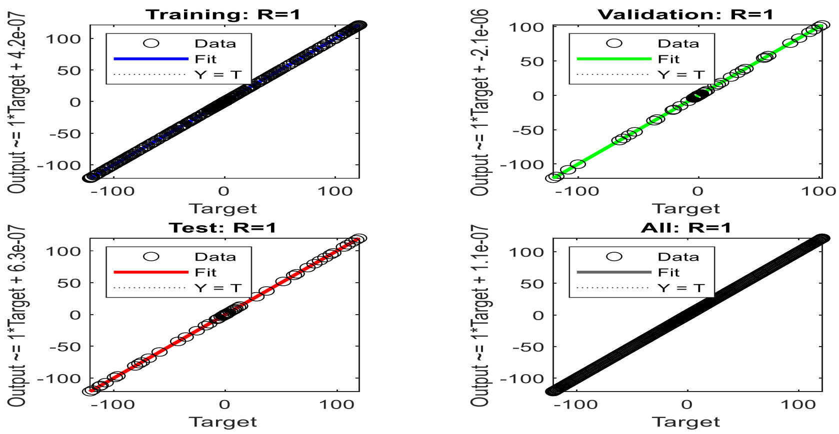


Figure 3.18: Network performance analysis

After obtaining the best regression values through several retraining’s, generate the Simulink block of the neural network controller for simulating the system in MATLAB Simulink.

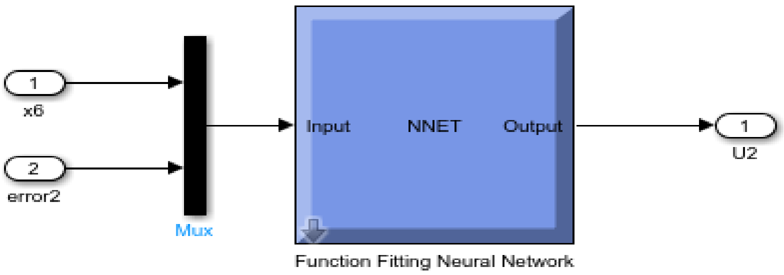


Figure 3.19: NN controller Simulink diagram

3.13 Neuro-Fuzzy System

Before fuzzy systems gained popularity in industrial applications, it was widely believed that developing a high-performing fuzzy system was a challenging task. The process of determining membership functions and suitable rules often involved a lengthy trial-and-error approach.

Neural networks, with their efficient learning capabilities, were introduced as an alternative to automate the tuning process of fuzzy systems.

Fuzzy logic systems and neural networks can be combined to combine their advantages and treat each disease separately. Neural networks gain interpretation and clarity of system representation from fuzzy systems and introduce computational learning features into them. Here, fuzzy systems' drawbacks are offset by neural networks' capacity for learning. The complimentary nature of the Neuro-Fuzzy approaches refreshes their combined application.

3.13.1 ANFIS

ANFIS is a hybrid intelligent system that combines the strengths of fuzzy logic concepts as well as neural networks. It is especially useful for modeling complex; nonlinear systems where conventional methods might struggle.

This architecture utilizes a Takagi-Sugeno fuzzy inference system with five layers. The first hidden layer maps the input variables to their respective membership functions. In the second hidden layer, the T-norm operator is applied to compute the antecedents of the rules. The third hidden layer normalizes the rule strengths, while the fourth hidden layer calculates the consequences of the rules. Finally, the output layer computes the overall output by summing all the signals it receives (Al-Mekhlafi et al., 2018; Saifizul Bin Abdullah et al., 2006).

Sugino fuzzy model is most commonly used in various fuzzy inference model because of high interpretability, increased computational efficiency, built in optimal and adaptation techniques (Saifizul Bin Abdullah et al., 2006), (Al-Mekhlafi et al., 2018).

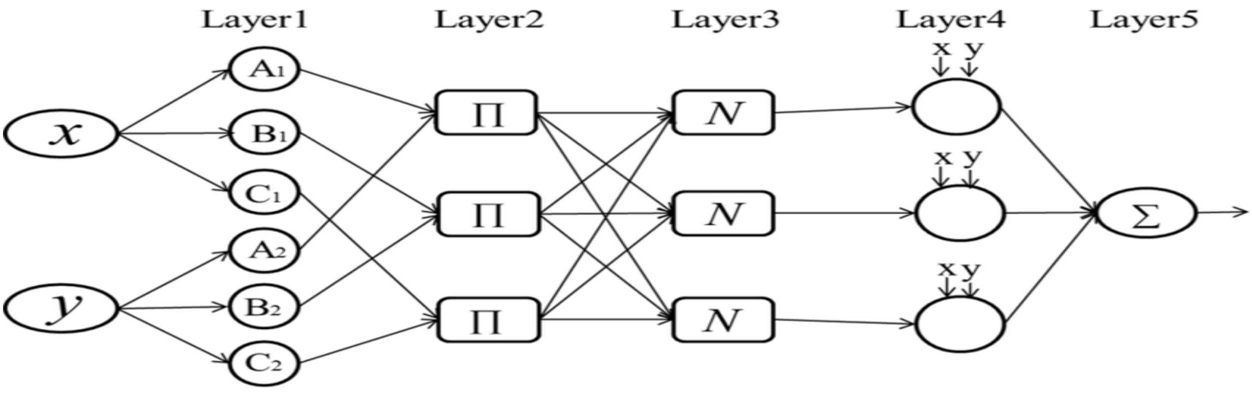


Figure 3.20: Architecture of ANFIS

For this inference system, the inputs are x1and x2 and the output is U1eq for balancing control.

The five layers that make up ANFIS are the product layer, defuzzification layer, normalization layer, fuzzification layer, and overall output layer (Saifizul Bin Abdullah et al., 2006).

For this thesis work, the ANFIS designed using two linguistic variables x1 and x2 and with seven linguistic values for each linguistic variable. Depending on these linguistic values (membership functions) there are 49 rules used here.

a. Function of Each ANFIS Layer:

Layer 1: For the input vectors x1 and x2, every single adaptive node in this layer determines the membership range. This layer's nodes have a square form and the following node function:

$$\begin{aligned} O_{1,i} &= \mu_{x1_i}(x_1),\ i = 1,2,3 \ldots \\ O_{1,i} &= \mu_{x2_i}(x_2),\ i = 1,2,3 \ldots \end{aligned} \tag{3.95}$$

Where x1 and x2 is the i^{th} node input, x1and x2 is the linguistic level associated this node's operation. In the other hand, O1, i is the membership function of x1 and x2 and i identify the degree to which the given variable satisfies the quantifier (i = 1, 2, 3…).

Layer 2: the node in this layer is circular which is labelled by Π. Every node in this layer multiplies all incoming signals before sending the result out. This result illustrates a rule's firring strength.

$$O_{2,i} = w_i = \mu_{e1_i}(e1) * \mu_{x1_i}(x_1) * \mu_{x2_i}(x_2),\ i = 1,2 \ldots \tag{3.96}$$

Layer 3: In this layer, each node is denoted by a circle and given the number N. This layer divides each incoming entry by the sum of all incoming entries.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{\sum_1^{343} w_i} \tag{3.97}$$

Layer 4: the node in this layer represented by square. Using the node function, this layer calculates the i^{th} rule's contribution to the output:

$$O_{4,i} = \overline{w}_i z_i = \overline{w}_i(p_i x_1 + q_i x_2 + r_i e1 + d_i) \tag{3.98}$$

where (p_i, q_i, r_i ,d_i) are the subsequent parameter sets of the i^{th} node in this layer and $\overline{w_i}$ is the normalized output of layer 3.

Layer 5: in this layer there is a single node which is represented by P. At a given input, the node in this layer gives the network response as:

$$O_{5,i} = \sum \overline{w}_i z_i = \frac{\sum_1^{49} w_i z_i}{\sum_1^{49} w_i} \tag{3.99}$$

b. Hybrid Learning Algorithm

The hybrid learning algorithm trains and adjusts the FIS by combining the least mean square method and the back propagation algorithm (BP). The premise parameters in layer 1 and the consequent parameters in layer 4 are two significant groups of configurable parameters in ANFIS, according to the layer description. A linear combination of consequent parameters can be used to express the ANFIS's overall output. The expression for the Ac2 output in figure 3.18 is as follows:

$$\begin{gathered} A_{c2} = \bar{w}_1 z_1 + \bar{w}_2 z_2 = \sum_{i=1}^{2} \bar{w}_1 (p_i S + q_i \Delta S + r_i) \\ = (\bar{w}_1 S) p_1 + (\bar{w}_1 \Delta S) q_1 + \bar{w}_1 r_1 + (\bar{w}_2 S) p_2 + (\bar{w}_2 \Delta S) q_2 + \bar{w}_2 r_2 \end{gathered} \tag{3.100}$$

These premises and their ensuing parameters were identified using the hybrid learning technique. The least mean square technique is used by ANFIS In order to determine the results that follow parameters, while backpropagation training is used to determine the input parameters for the membership function. Each phase of the iterative learning process contains two phases. While the parameters of the premises are fixed, the input patterns are propagated, and the parameters of the consequents are computed utilizing the procedure of the iterative minimum squared method in the first stage. The second section involves the propagation of the input patterns once more, with each iteration using the learning algorithm's backpropagation to change the premises' parameters while keeping the consequents constant.

c. Procedure to Design ANFIS_ STSMC Based on STSMC

Step1: Determine the Simulink model of the plant and control it with STSMC.

Step2: x1 and x2 and the equivalent term of STSMC are sent to workspace using to workspace block and MUX in Simulink.

Step3: After obtaining data from step 2, I train the ANFIS inputs and output using this data. The inputs are x1 and x2, and the output is the output of step 2.

Step4: Substitute the Equivalent term of STSMC by ANFIS controller.

3.9.4 Training of ANFIS Controller Using Data From STSMC

ANFIS based modeling combines the learning capacity of NN with the transparent linguistic representation of fuzzy systems, allowing them to be trained to perform an input/output mapping. This is demonstrated when I train the ANFIS using x1, x2, and Ueq.

The x1 and x2 are inputs, and Equivalent term of U1 is the result from ANFIS. In the proposed controller, the Seguno fuzzy model with five-layer ANN is used. Figure 3.21 depicts the relationship between the designed ANFIS's input and output.

Figure 3.21: ANFIS controller input/output

To generate the ANFIS system, ten thousand one (10,001) data has been considered. 70% of the total data (7,001 items) has been designated as training data, with the remaining 15% (1,500 items) designated as check data and validating data. ANFIS combines the properties NN and fuzzy logic. It offers a way to learn about a data set so that the fuzzy modeling process may calculate the membership function parameters needed for the associated fuzzy inference system (FIS) to track the input/output data. A network type structure akin to a neural network, which maps inputs through input membership functions and related parameters and outputs through output membership functions and associated parameters to outputs, can be taken into consideration in order to create the FIS structure.

The membership functions' corresponding parameters will be changed through the learning (training) process. In this work, a hybrid learning algorithm is used to compute these parameters. Seven membership functions were taken into consideration during the creation of ANFIS, 300 epochs were employed, 0 error tolerance was set, and triangular functions were the type of input membership functions used. The output type that is chosen is the linear membership function type. 49 rules are created from the chosen set of membership functions.

The RMSE was used to guide the selection of training and validation data sets. The system was trained with various parameter configurations, including data set samples, types and numbers of membership functions, and the number of training epochs, to achieve optimal performance and develop the most effective ANFIS model. The MATLAB toolbox ANFIS editor builds a fuzzy inference system whose membership function parameters are adjusted via a hybrid learning algorithm technique, using the input/output data set that was obtained by STSMC.

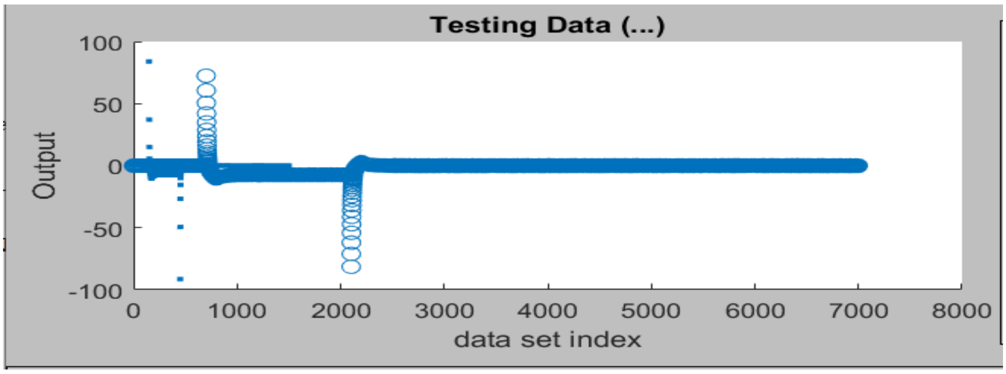
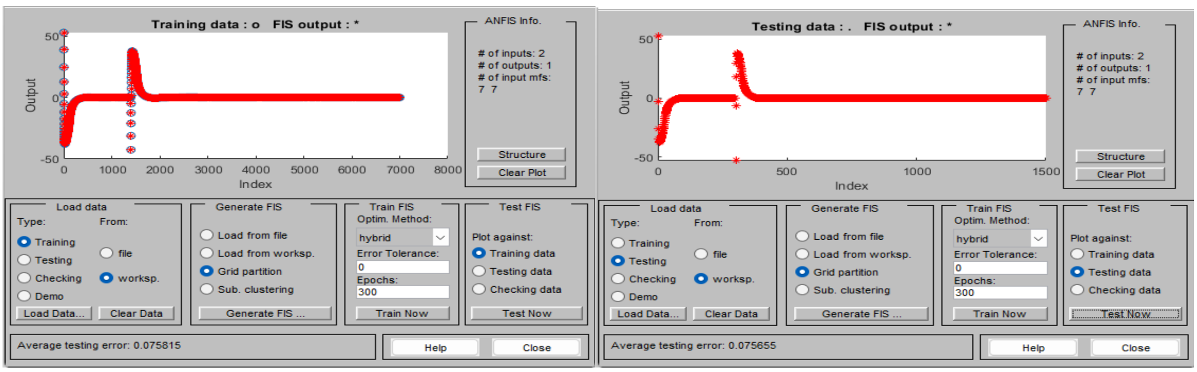


Figure 3.22: ANFIS sets of data for training and checking

The generalization capacity of the fuzzy inference system is tested at each epoch using checking data. The format of the training and verifying sets of data is the same. The fuzzy inference model is validated using this data set. This validation is carried out by feeding the model with the testing data and then observing how well the model responds to it. When the checking data option is selected in the ANFIS Editor GUI, it is used to the model at every training epoch. The FIS membership function specifications are determined using the ANFIS editor GUI after both checking and training data have been loaded. The checking data is similar enough to the training data that the checking data error decreases as the training progresses. The checking error is the difference between the value of the checking data output and the value associated with that checking data output value, which is the output of the fuzzy inference system corresponding to the same checking data input value. The RMSE for the checking data at each epoch is recorded by the checking error. As the system is trained, the checking error versus epochs curve is plotted on the ANFIS Editor GUI. 0.075815 is the least training RMSE.



(a) (b)

Figure 3.23: a) Training and checking error b) Testing of FIS with checking data

Figure 3.24 below illustrates the structure of the proposed ANFIS controller. This figure provides a comprehensive overview of the architecture and components of the ANFIS controller. It highlights the arrangement of the various layers within the ANFIS framework

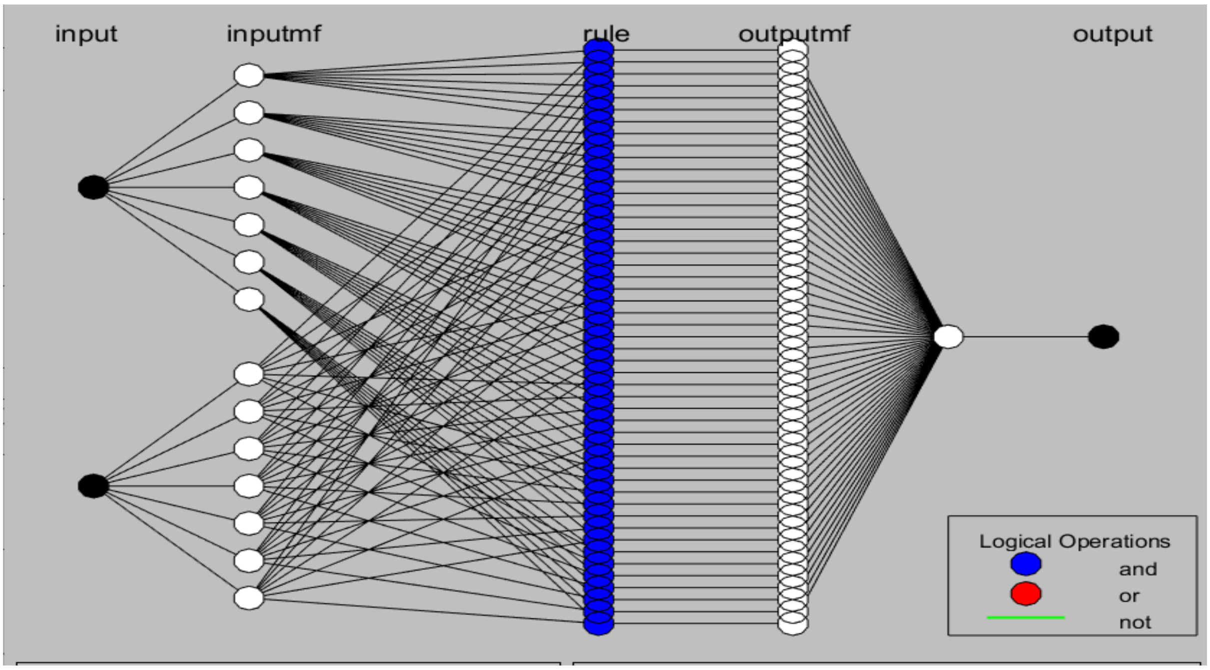


Figure 3.24: ANFIS controller Schematic diagram

CHAPTER FOUR

4 RESULT AND DESCUTIONS

4.1 Introduction

In this chapter, the simulation of the Segway is conducted using MATLAB/Simulink software to test and verify the performance of the designed ANFIS controller for balance angle and the neural network controller for direction control, comparing the simulation results with those of STSMC.

The simulation was conducted using the parameters of the segway detailed in Tables 3.1 and 3.2. An ANFIS controller is implemented for balancing the scooter, while a neural network controller is employed for managing steering of wheels. The parameters for these controllers are optimized using TLBO and PSO to achieve optimal system performance. The MATLAB/Simulink model for the two-wheel electric scooter is illustrated in Figure 4.1 model with STSMC and Figure 4.2 model with ANFIS and NN.

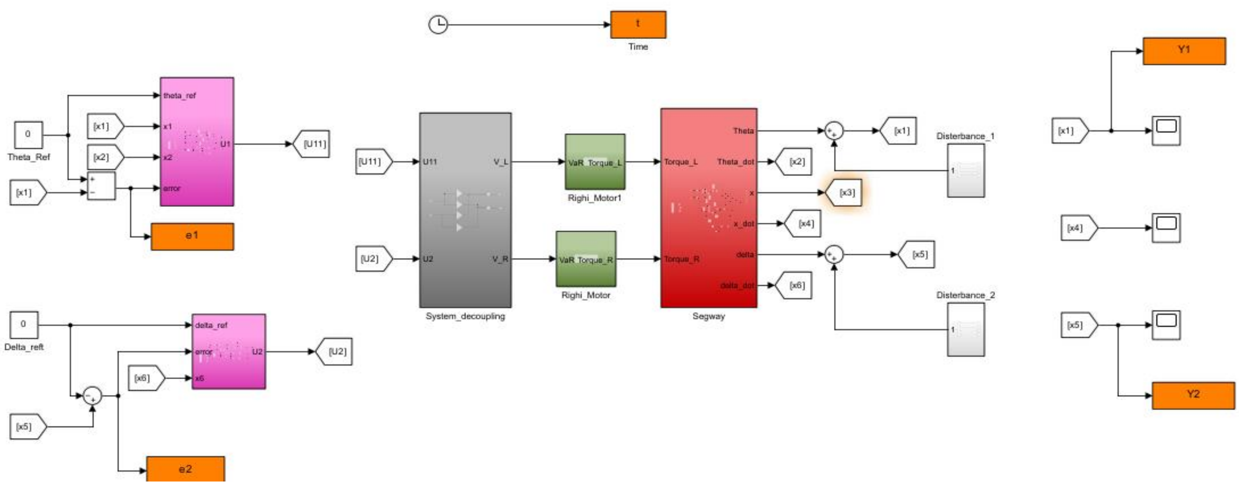


Figure 4.1: Segway model with STSMC

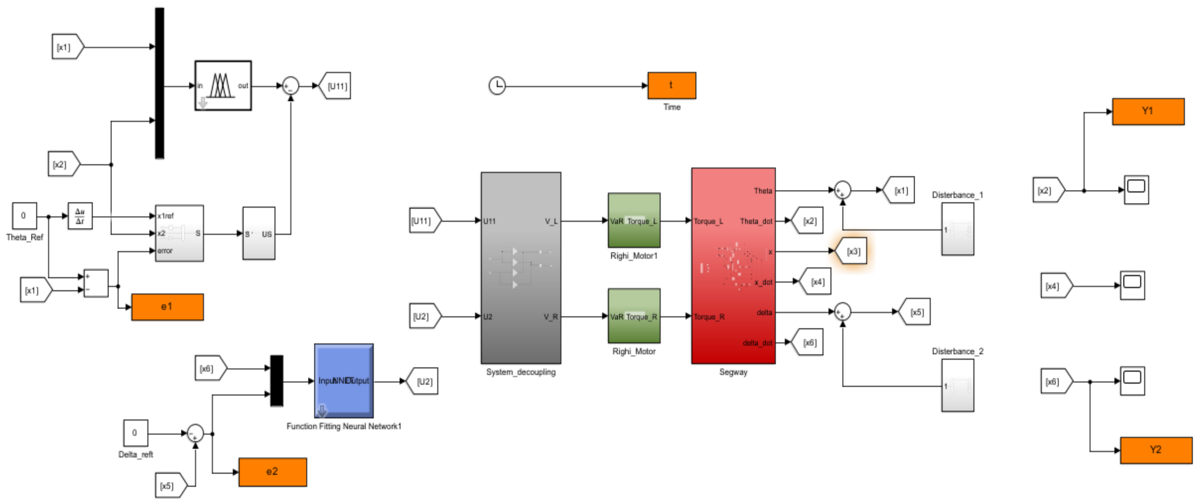


Figure 4.2: MATLAB/Simulink model of Segway

4.2 Simulation Results

The parameters of the designed STSMC controller are optimized using two algorithms: PSO and TLBO. Figures 4.3 and 4.4 present a comparison of these two optimization techniques for the STSMC controller. Based on the rider's input command for direction control and pitch angle for balance, the results demonstrate that TLBO performs better than PSO with respect to time domain criteria, as shown in Table 4.1 below. The results for direction control and balance both show a 0% steady-state inaccuracy, indicating that the TLBO-optimized STSMC controller is effective for balancing and directional control of segway.

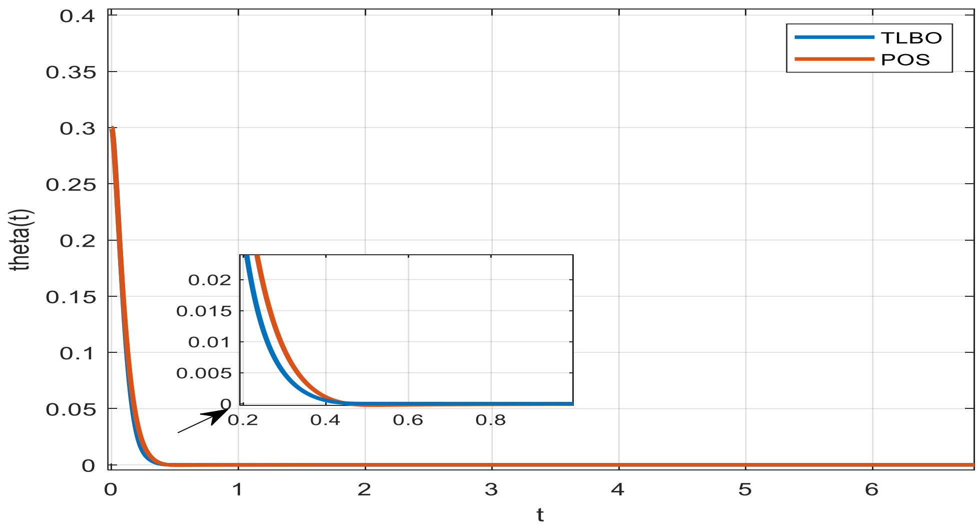


Figure 4.3: Balance angle response to rider's tilt 0.3 rad for TLBO and PSO

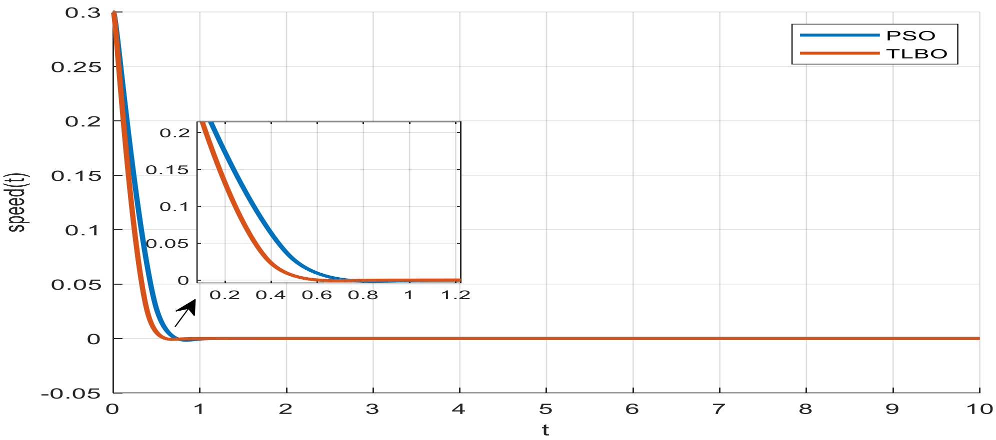


Figure 4.4: Direction angle response to rider's direction input 0.3 rad for TLBO and PSO

Table 4.1: Performance of system with TLBO and PSO Optimization technique

Algorithm	TLBO		PSO	
	Balance angle	Direction angle	Balance angle	Direction angle
Rise Time(sec)	0.1845	0.255	0.2065	0.2711
Settling Time(sec)	0.3157	0.4209	0.3638	0.5337
Overshoot %	0.2%	0.08%	0.23%	0.15%
Steady State error	0	0	0	0

Figure 4.5 and 4.6 shows the comparison of system response of optimal STSMC controller and replaced controllers the ANFIS_STSMC (for balance control) and NN (direction control) of segway. With an initial pitch angle for balancing (0.3) and a yaw angle for direction control (0.2), the results show that intelligent controllers ANFIS integrated STSMC and NN outperforms STSMC in terms of time specifications, as shown in Table 4.2 below. Both balancing and direction control responses exhibit a 0% steady-state error, indicating that the controllers ANFIS integrated STSMC and NN is effective for balancing and directional control of two-wheel electric scooter.

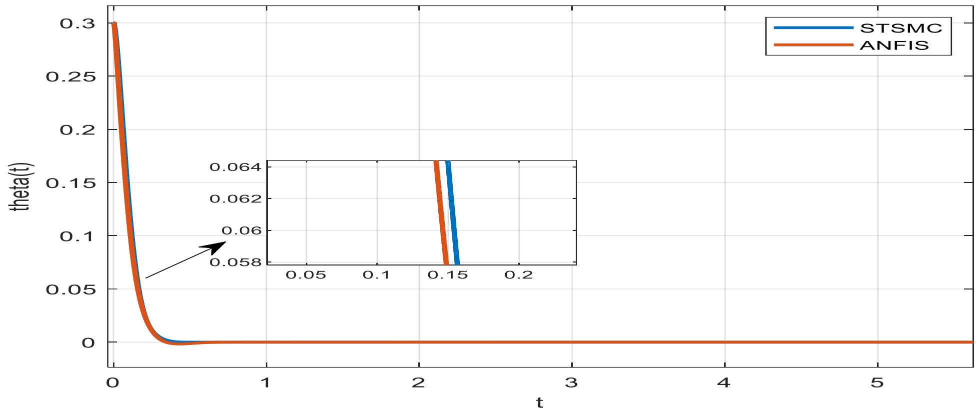


Figure 4.5: Balance angle response of system for rider's tilt of 0.3 with ANFIS_STSMC and STSMC

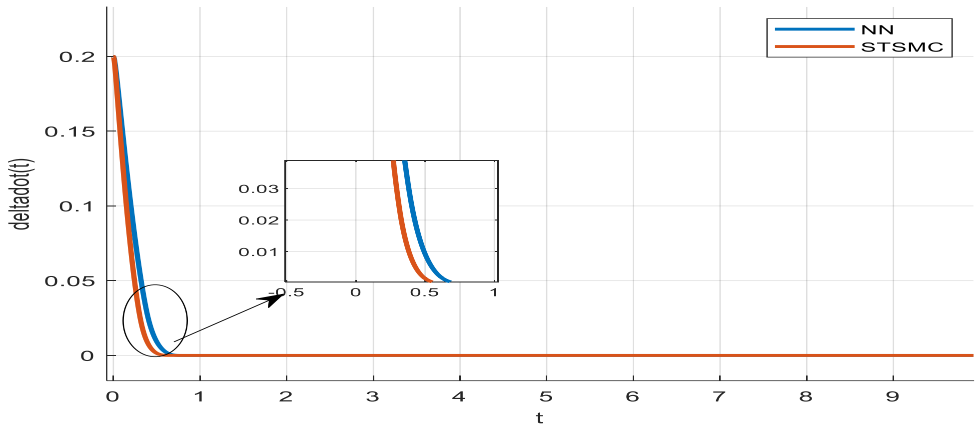
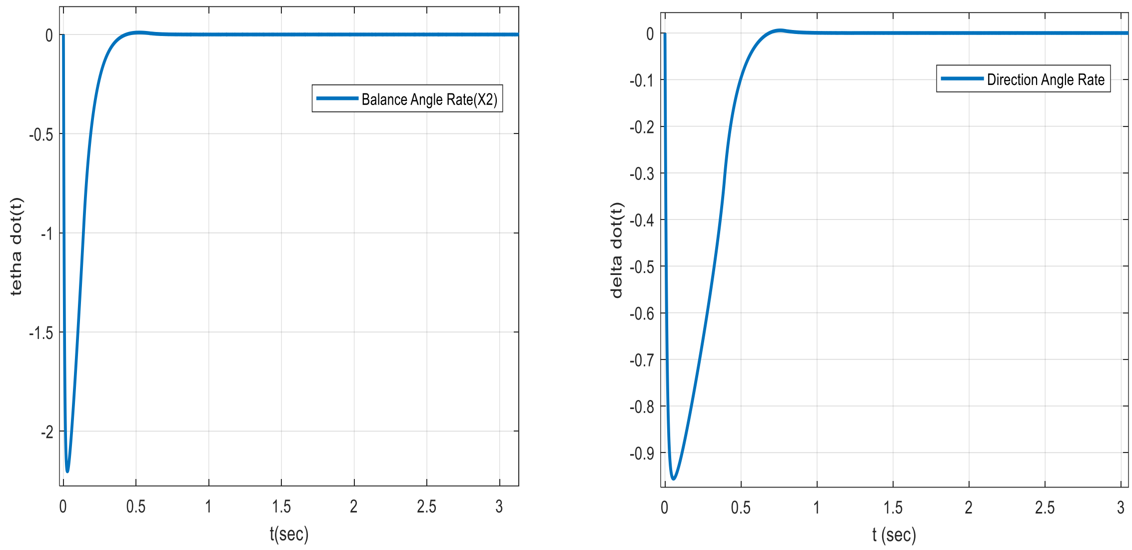


Figure 4.6: Direction angle response of system for rider's direction input of 0.2 with NN and STSMC

Table 4.2: Performance of system with ANFIS_STSMC and NN with STSMC

Controller	ANFIS_STSMC and NN		STSMC	
	Balance angle	Direction angle	Balance angle	Direction angle
Rise time	0.1802	0.2994	0.1845	0.3157
Settling time	0.2538	0.3495	0.255	0.4209
Overshoot %	0.16%	0.05	0.2%	0.08%
Steady State error	0	0	0	0

The corresponding balance angle rate(x2) and direction angle rate(x6) for the above input are shown in Figure 4.7 below. Angle rate of the scooter converge to zero for the two conditions.



(a) (b)

Figure 4.7: corresponding rate of balance angle and direction angle for rider’s tilt of 0.3 with ANFIS_STSMC and NN

Figure 4.8 displays how the Segway responds to a tilt angle of θ = -0.2 rad from a rider, while Figure 4.9 illustrates its response to a direction input angle of δ = -0.2 rad. In both scenarios, the Segway’s balance and direction angles stabilize at zero. These findings show that the Segway effectively maintains balance and accurately adjusts direction in response to various rider input angles.

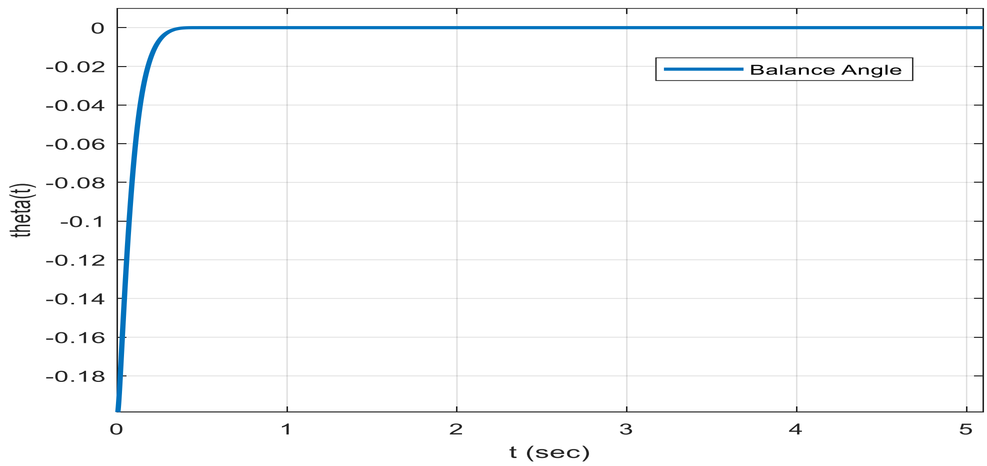


Figure 4.8: Balance angle response of system for rider’s tilt of -0.2 with ANFIS_STSMC

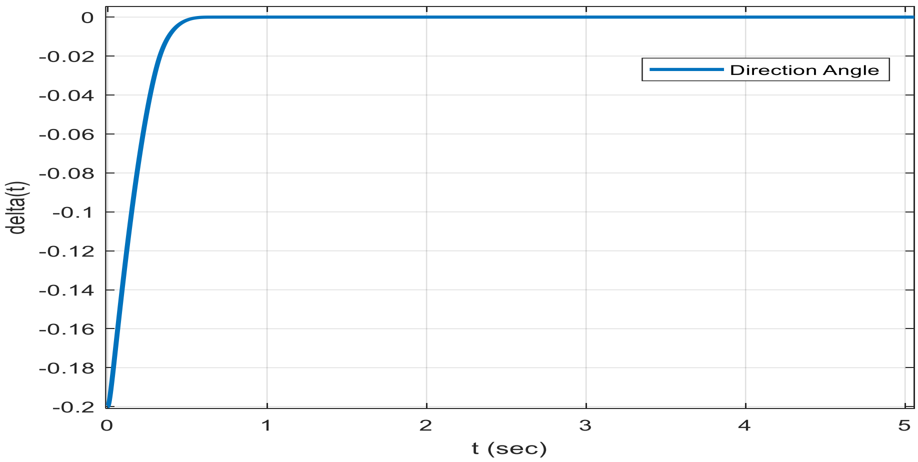
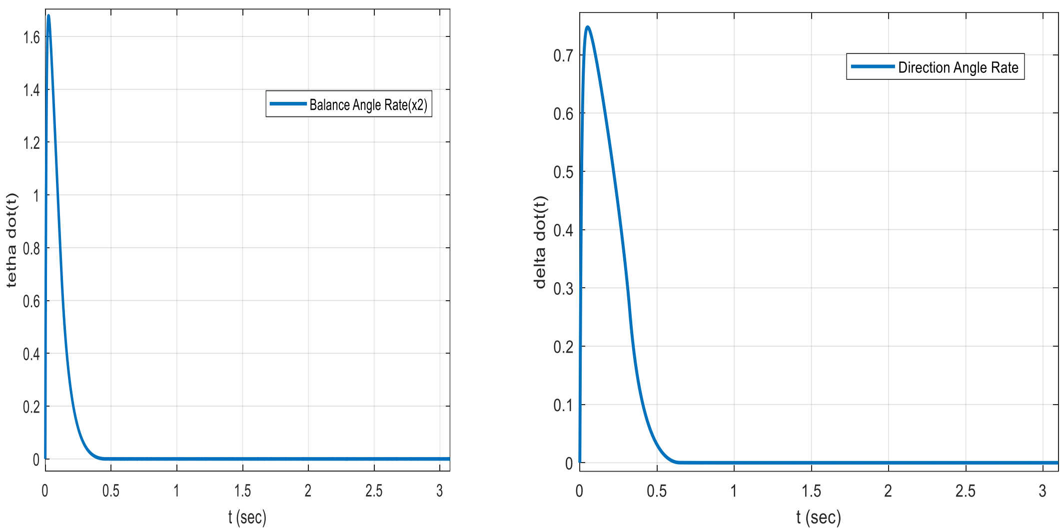


Figure 4.9: Direction angle response of system for rider's direction input of -0.2 with NN

The corresponding balance angle rate(x2) and direction angle rate(x6) for the above input are shown in Figure 4.10 below. Angle rate of the scooter converge to zero for the two conditions.



(a) (b)

Figure 4.10: corresponding balance angle rate(a) and direction angle rate(b)

In this work, the segway can carry a human load of up to 120 kg. Figure 4.11 shows the response to human load varying from 60kg to 120kg. The mass of 90kg has a rising time of 0.1802sec and a settling time of 0.2538sec for initial pitch angle $\theta = 0.3$ rad and has a smaller rising time and settling time than 60kg, and 120kg loads. How ever for 60kg it has no overshot.

Figure 4.12 shows the response to a human load varying from 60kg to 120kg. The mass of 90kg has a rising time of 0.2994sec and a settling time of 0.3495sec for an initial yaw angle of

$\delta = 0.2$ rad and has a smaller rising time and settling time than 60kg, and 100kg loads. The results show that for different mass variations, the scooter balance and direction control have good dynamic performance for initial pitch angle $\theta = 0.3$ rad and for yaw angle $\delta = 0.2$ rad. One of the benefits of using intelligent control techniques is that it has low sensitivity to plant parameter variations, like the mass of humans.

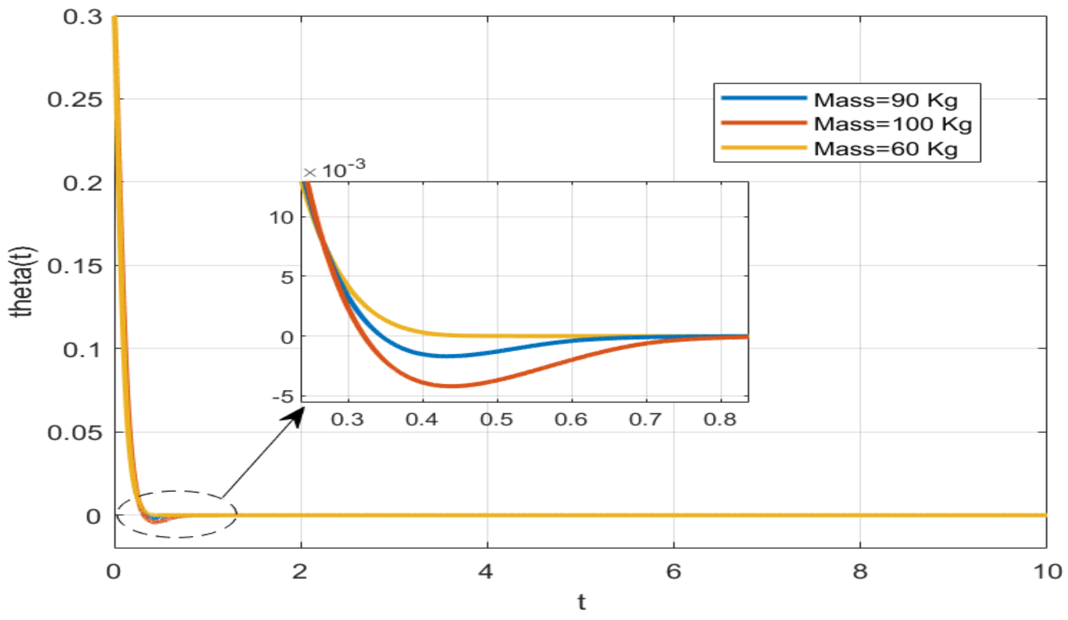


Figure 4.11: Balance angle response with initial $\theta = 0.3$ rad for mass variation from 60kg to 100kg.

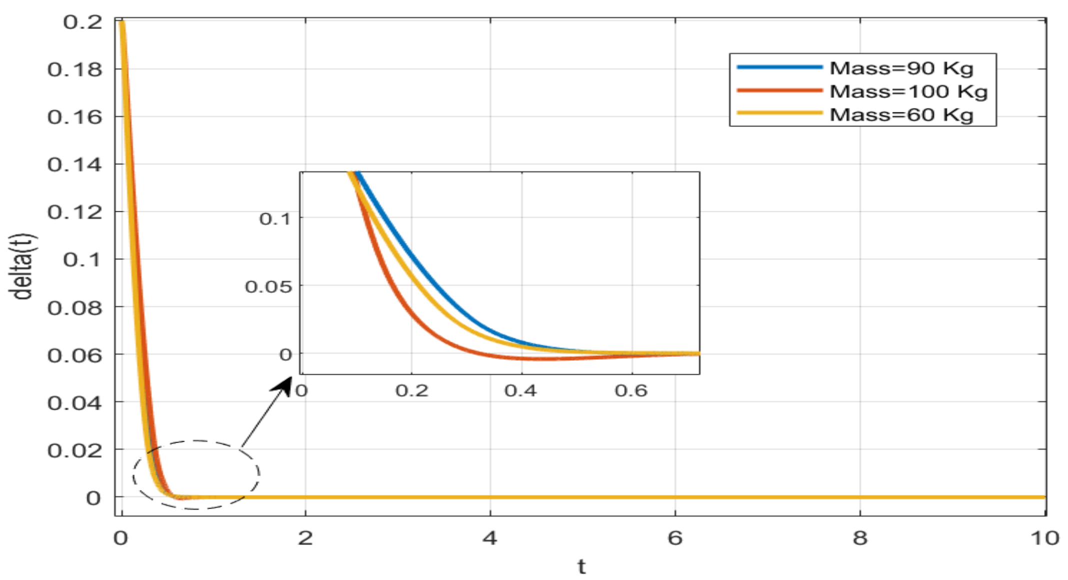


Figure 4.12: Direction angle response with initial $\theta = 0.2$ rad for mass variation from 60kg to 100kg.

Figure 4.13 and 4.14 shows the response of the system with a tilt angle of θ = 0.2 rad from a rider and a direction input angle of δ = 0.2 rad considering uneven terrain and taking the impact of the disturbance on the ground b/t time 2-4 sec and with defined values of dx, dy, dHL, dHR and dδ as below and the result shows that it has low sensitivity to small variations on the terrain.

$$\begin{cases} dy = 0.2 * \sin(2t) \\ dx = 0.2 * \cos(2t) \\ dHL = 0.03 * \cos(2t) \\ dHR = 0.03 * \cos(2t) \\ d\delta = 3 * \exp(-1.2 * t) \end{cases}$$

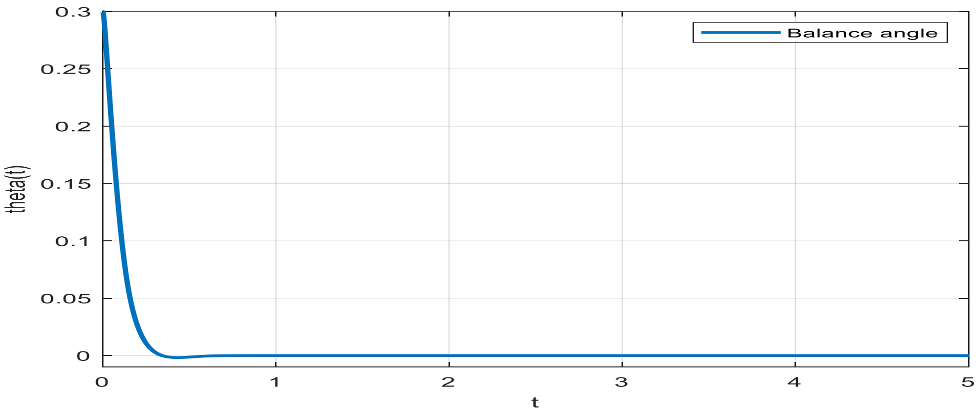


Figure 4.13: Balance angle response of system for rider's tilt of 0.3 with ANFIS_STSMC and defined disterbance

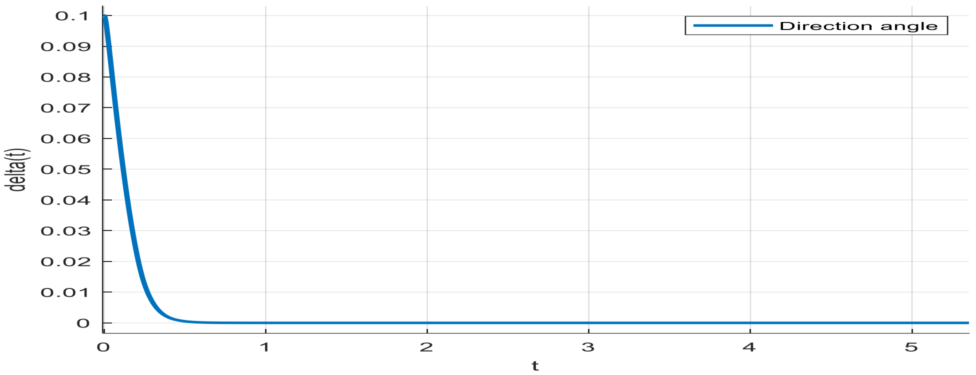


Figure 4.14: Direction angle response of system for rider's direction input of 0.1 with NN and defined disturbance

4.3 Simulation Result for Balance and Direction Angel Variation

The rider's orders are generated using step signals and combining them to generate pulse signal. Figure 4.15 show balance angle response for rider's tilt order pitch angle θ =0.3$[U(t) - U(t-2) - U(t-4) + U(t-6)]$ and the results demonstrate that the segway can keep balance well for different pitch angle change.

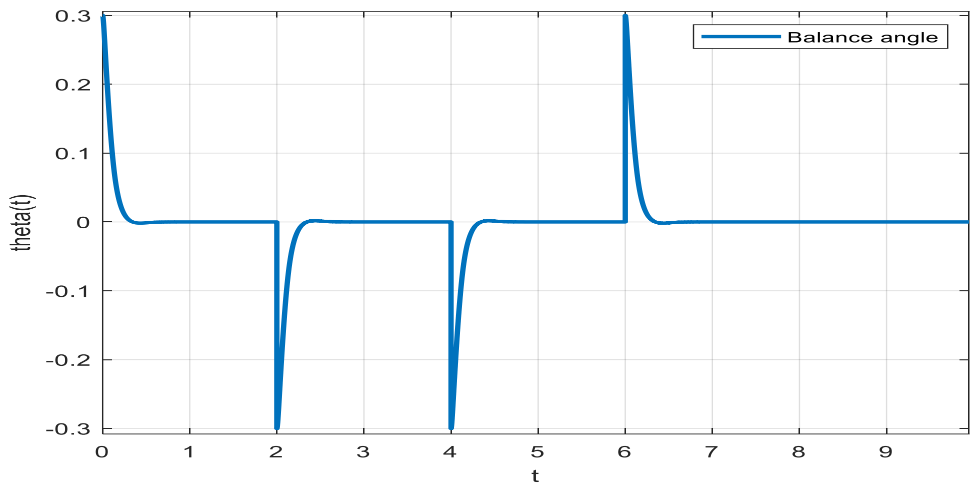


Figure 4.15: response of balance angle for the rider's tilt order

Figure 4.16 show direction angle response for rider's order angle δ=0.2$[U(t-1) - U(t-3) - U(t-5) + U(t-7)]$ and the results shows the designed system has better performance with d/t yaw angle changes and can keep the segway in desired direction.

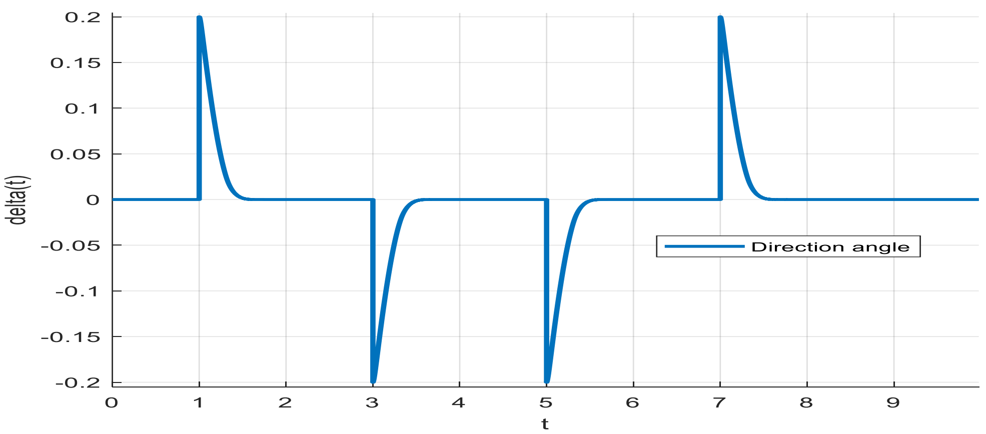


Figure 4.16: response of direction angle for the rider's direction order

Figure 4.17 and 4.18 shows response of Applying those signals on uneven terrain with the predefined values used for testing in Figure 4.15 and 4.16.the result shows that the proposed controllers can withstand the disturbance and keep balance and direction angle in the desired value.

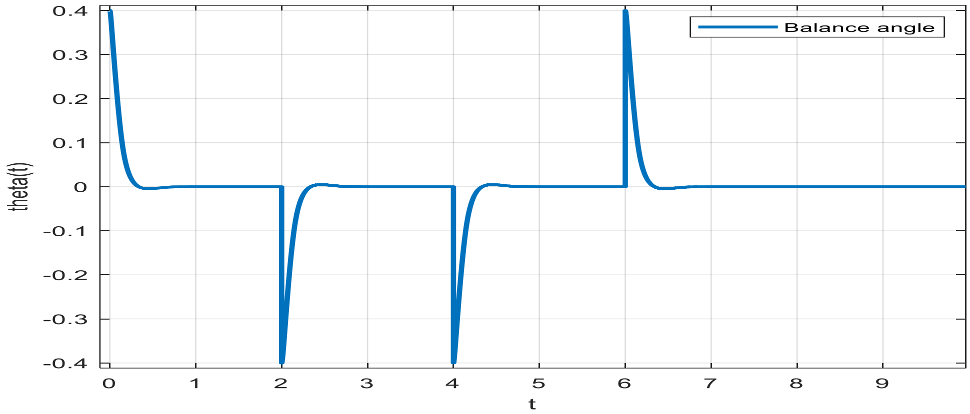


Figure 4.17: response of balance angle for the rider's tilt order with uneven terrain

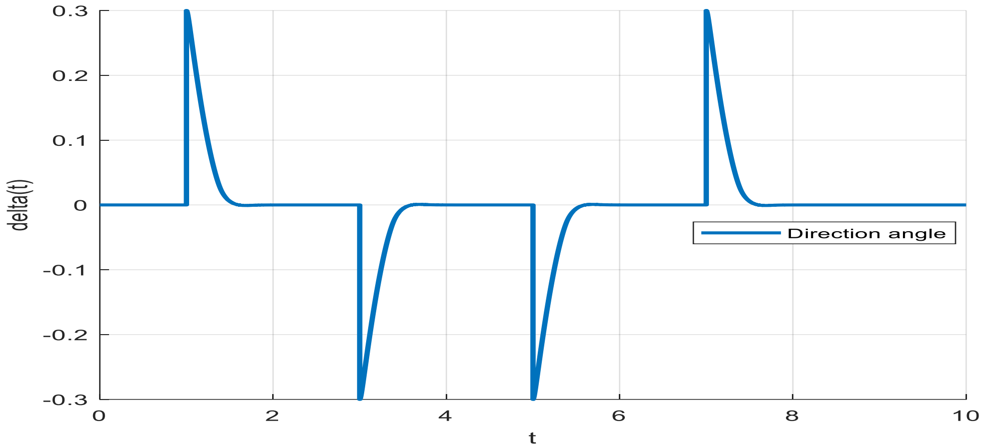


Figure 4.18: response of balance angle for the rider's direction order on uneven terrain

The speed of the segway for the above two cases is shown in the figure 4.19.as it is seen in the figure below the segway changed its speed to keep balance and direction in desired value.

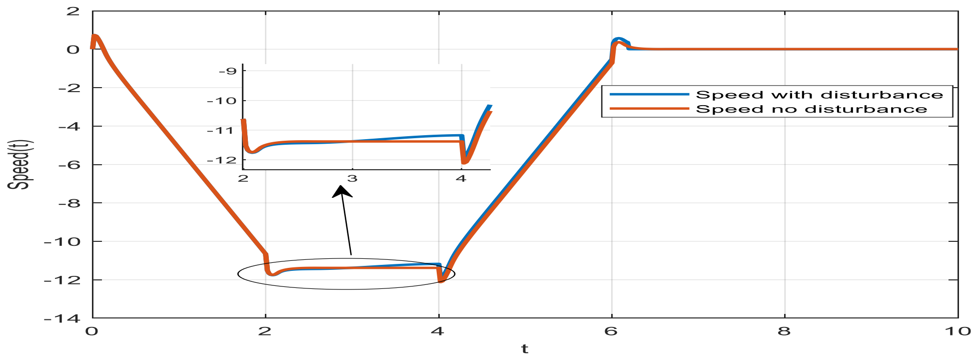


Figure 4.19: speed response of segway

The two controlling current signals from ANFIS_STSMC and NN controller for balance angle and direction angle respectively shown in the figure 4.17 and 4.18.

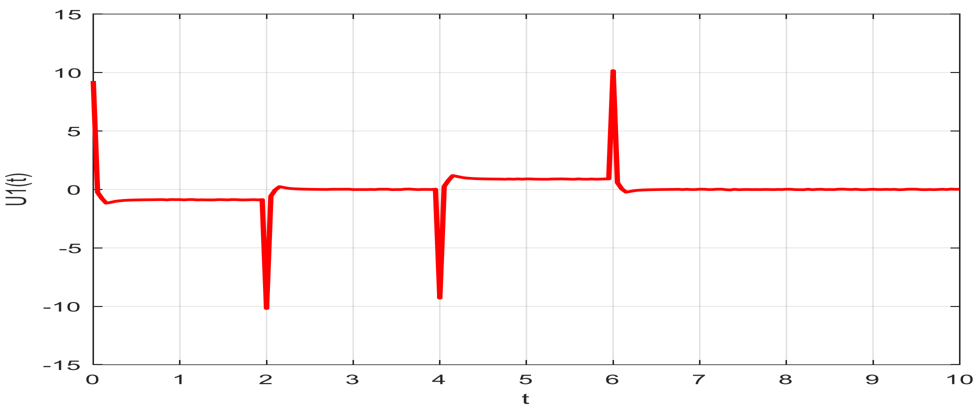


Figure 4.20: balance angle controller output U1

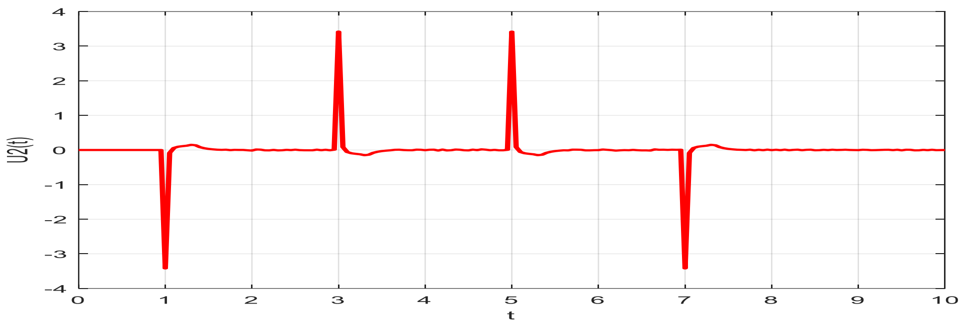


Figure 4.21: direction angle controller output U2

Figure 4.22 show Pitch angle response for the initial pitch angle θ =0.3$[U(t) - U(t-2) - U(t-4) + U(t-6)]$ as show in Figure and the results demonstrate that the segway can keep balance (stabilize) well for different pitch angle change.

In the same way Figure 4.23 show the yaw angle response, for initial yaw angle of δ=0.2$[U(t-1) - U(t-3) - U(t-5) + U(t-7)]$ in the Figure and the result shows that scooter direction control is done in the perfect manner for different yaw angle change

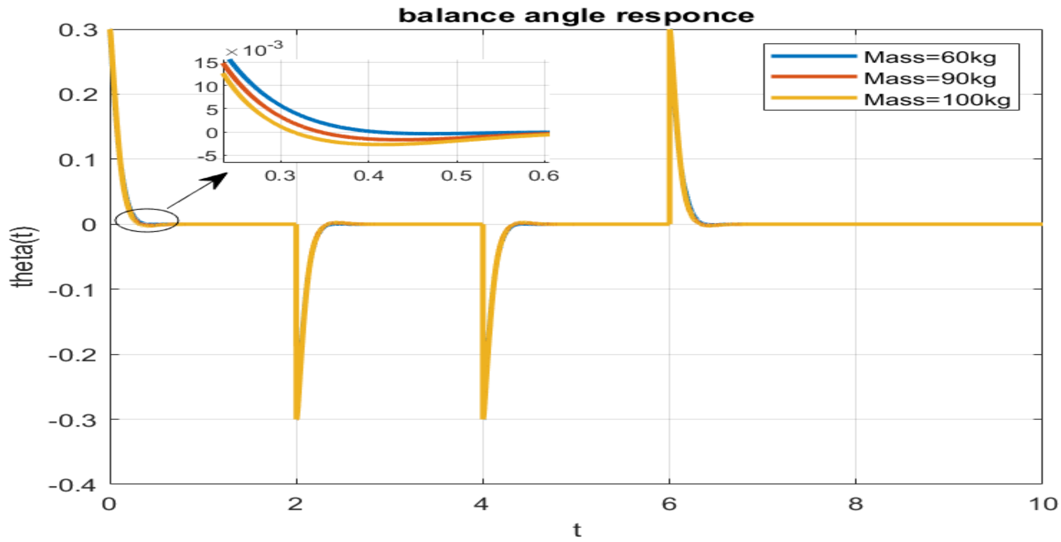


Figure 4.22: Balance angle response for different initial angle

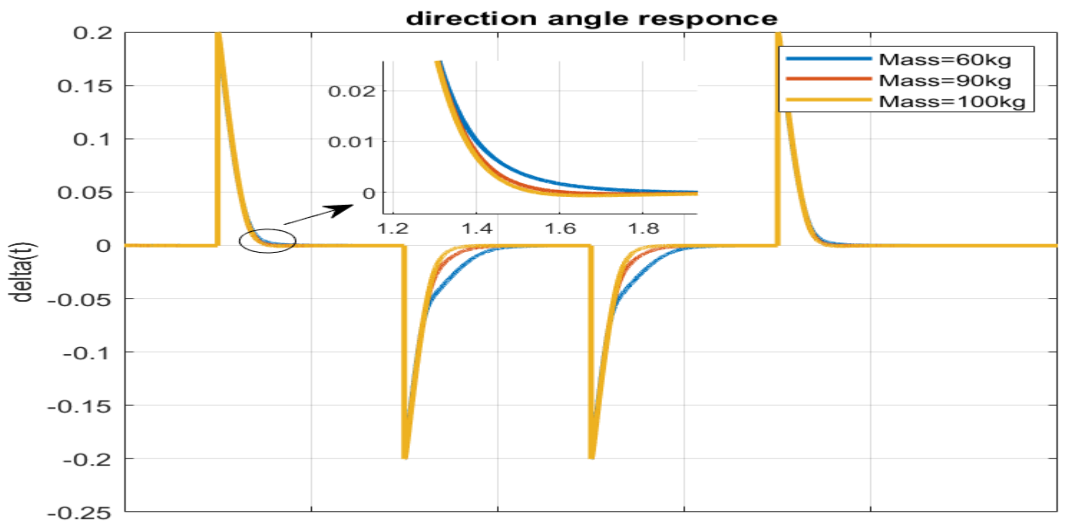


Figure 4.23: Direction angle response for different initial angle

In general, segway system efficiently controlled through different tilt and direction order, and all simulation result shows the two wheeled electric Scooter can operate stably and with good performance.

CHAPTER FIVE

5 CONCLUSION AND RECOMMENDATION

5.1 Conclusions

The self-balancing scooter, resembling an inverted pendulum, is a highly nonlinear and unstable system, prone to rapid instability with even minor input disturbances in an open-loop configuration. In this paper, an ANFIS-based Super Twisted Sliding Mode Controller (STSMC) for balance control and a Neural Network (NN) for direction control were proposed, leveraging the principles of inverted pendulum dynamics to stabilize the system.

Key parameters such as the mass of the body, mass of the wheels, wheel radius, and DC motor characteristics were meticulously identified and modeled in MATLAB/Simulink to accurately replicate the system's dynamics.

The controllers were optimized using Teaching-Learning-Based Optimization (TLBO) and Particle Swarm Optimization (PSO), with a comparative analysis based on computation time and fitness values. The results indicate that TLBO outperforms PSO, offering faster computation times and better fitness values.

The simulation results show that using ANFIS-based STSMC for balance control, combined with a Neural Network for direction control, effectively keeps the segway balanced and under control, much like balancing an Inverted pendulum. The system works well even with a person weighing up to 120 kg. It maintains precise balance and direction, even from different starting angles. This proves that the ANFIS-STSMC and Neural Network controller, especially when tuned with TLBO for STSMC, are highly effective for a two-wheel self-balancing electric scooter.

5.2 Recommendations

In the rapidly evolving field of transportation, self-balancing scooters have emerged as a popular solution for personal mobility. This study demonstrates the effectiveness of an ANFIS-based Super Twisted Sliding Mode Controller (STSMC) and Neural Network (NN) for balance and direction control in a two-wheel self-balancing scooter, using simulations conducted in MATLAB/Simulink.

For future work, transitioning from simulation to hardware implementation is recommended to validate and enhance the proposed control strategies in real-world conditions. Implementing the ANFIS-based STSMC and NN controllers on an actual scooter would provide valuable insights into their performance and reliability.

However, it is important to acknowledge some limitations of the ANFIS and NN approaches. ANFIS can be computationally intensive and may struggle with real-time adaptability in highly dynamic environments. Neural Networks, while powerful, require extensive training data and may exhibit limited generalization to novel conditions not encountered during training. These factors could impact the overall robustness and efficiency of the control system in practical applications.

Given these limitations, exploring adaptive control strategies may offer significant advantages. Adaptive controllers are designed to adjust their parameters in real-time based on changing system dynamics and external conditions, potentially providing more robust and flexible control compared to static models like ANFIS and NN.

Additionally, the self-balancing scooter could be enhanced with features to support various applications beyond personal mobility. For instance, integrating cargo-carrying capabilities or transforming the scooter into a versatile utility vehicle for last-mile delivery could expand its use. Adding modular attachments for carrying goods, tools, or even small equipment could make the scooter a valuable asset in urban logistics or for small business operations.

REFERENCES

9.5 inch Self Balancing Segway mono shaft motor - uumotor. (n.d.). Retrieved September 17, 2024, from https://www.uumotor.com/9-5-inch-self-balancing-segway-mono-shaft-motor.html

2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC) : 26-28 September, 2013, Jaypee University of Information Technology, Waknaghat, Shimla, H.P., India. (2013). IEEE.

Ahmed, A. A., & Saleh Alshandoli, A. F. (2020a, September 25). On replacing a PID controller with Neural Network controller for Segway. *2020 International Conference on Electrical Engineering, ICEE 2020*. https://doi.org/10.1109/ICEE49691.2020.9249811

Al-Mekhlafi, M. A. A., Wahid, H., & Aziz, A. A. (2018). Adaptive neuro-fuzzy control approach for a single inverted pendulum system. *International Journal of Electrical and Computer Engineering*, *8*(5), 3657–3665. https://doi.org/10.11591/ijece.v8i5.pp3657-3665

Asali, M. O., Hadary, F., & Sanjaya, B. W. (2017a). Modeling, simulation, and optimal control for two-wheeled self-balancing robot. *International Journal of Electrical and Computer Engineering*, *7*(4), 2008–2017. https://doi.org/10.11591/ijece.v7i4.pp2008-2017

Bin Xing, Lei Guo, Shimin Wei, & Yuan Song. (2016). *2016 IEEE International Conference on Information and Automation (ICIA) : 1-3 Aug. 2016*. IEEE.

Callinan, T. (2003). *Artificial Neural Network identification and control of the inverted pendulum*.

Chakrabarti, S. (n.d.). *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) : 1st-3rd November, 2018, University of British Columbia, Vancouver, Canada*.

Chawla, I., Chopra, V., & Singla, A. (2020). Performance comparison of PID and ANFIS controller for stabilization of x and x-y inverted pendulums. *Advances in Intelligent Systems and Computing*, *940*, 182–192. https://doi.org/10.1007/978-3-030-16657-1_17

DC Motor Control - MATLAB & Simulink Example. (n.d.). Retrieved April 19, 2024, from https://www.mathworks.com/help/control/ug/dc-motor-control.html?searchHighlight=state%20space%20model%20of%20dc%20motor%20volt age%20and%20torque&s_tid=srchtitle_support_results_7_state%2520space%2520model%2520of%2520dc%2520motor%2520voltage%2520and%2520torque

Dziwiński, P., Bartczuk, Ł., & Paszkowski, J. (2020). A New Auto Adaptive Fuzzy Hybrid Particle Swarm Optimization and Genetic Algorithm. *Journal of Artificial Intelligence and Soft Computing Research*, *10*(2), 95–111. https://doi.org/10.2478/jaiscr-2020-0007

Gómez Díaz, K. Y., De León Aldaco, S. E., Aguayo Alquicira, J., Ponce-Silva, M., & Olivares Peregrino, V. H. (2022). Teaching–Learning-Based Optimization Algorithm Applied in Electronic Engineering: A Survey. In *Electronics (Switzerland)* (Vol. 11, Issue 21). MDPI. https://doi.org/10.3390/electronics11213451

https://www.dreamstime.com/teacher-asking-kid-raising-hand-to-answer-question-lesson-school-kids-listening-answering-teachers-questions-classes-image224099528. (n.d.).

Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems. (n.d.).

khan, S. I., Choudhry, M. A., Ali, A., Shaikh, I. U. H., & Saleem, F. (2022). A Hybrid Technique for Upward Stabilization and Control of Two Wheeled Self-Balancing Segway. *Mehran University Research Journal of Engineering and Technology*, *41*(1), 169–179. https://doi.org/10.22581/muet1982.2201.17

Kim, S., & Kwon, S. J. (2015). Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot. *International Journal of Control, Automation and Systems*, *13*(4), 926–933. https://doi.org/10.1007/s12555-014-0564-8

Mg, L., Thwin, M., & Chan, Y. (2020). APPLICATION OF LQR CONTROL FOR TWO-WHEEL SELF-BALANCING ROBOT. In *J. Myanmar Acad. Arts Sci* (Vol. 2).

Mon, Y.-J. (2013). *ANFIS ANFIS ANFIS ANFIS Transient Transient Transient Transient Control Control Control Control for Double Inverted Pendulum Sy for Double Inverted Pendulum Sy for Double Inverted Pendulum Sy for Double Inverted Pendulum Sys s s stem tem tem tem*. www.IJCSI.org

Mudeng, V., Hassanah, B., Tonce, Y., Priyanto, K., & Saputra, O. (2020). Design and Simulation of Two-Wheeled Balancing Mobile Robot with PID Controller. In *International Journal of Sustainable Transportation Technology* (Vol. 3, Issue 1).

Önen, Ü. (2017). Particle Swarm Optimization Based LQR Control of an Inverted Pendulum. *Engineering and Technology Journal*, *2*(5). https://doi.org/10.18535/etj/v2i5.01

p.k.mani KANNAN | Professor (Associate) | D.E.E.E.,B.E(EEE).,M.E(PSE).,Ph.D(EEE) | Veltech Multitech Dr. Rangarajan Dr.Sagunthala Engineering College, Chennai | Department of Electrical and Electronics Engineering | Research profile. (n.d.). Retrieved August 9, 2024, from https://www.researchgate.net/profile/Pkmani-Kannan

Rao, R. V. (2016). Teaching-Learning-Based Optimization Algorithm. In *Teaching Learning Based Optimization Algorithm* (pp. 9–39). Springer International Publishing. https://doi.org/10.1007/978-3-319-22732-0_2

Saifizul Bin Abdullah, A., Saifizul, A., Zainon, Z., Abu Osman, N., Azlan, C., & Ungku Ibrahim, U. (2006). Intelligent Control for Self-erecting Inverted Pendulum Via Adaptive Neuro-fuzzy Inference System. *American Journal of Applied Sciences*, *3*(4), 1795–1802.

Segway - Wikipedia. (n.d.). Retrieved February 15, 2024, from https://en.wikipedia.org/wiki/Segway

Sevekari, P., Tamhane, B., & Kurode, S. (2020). Robust control for stable and safe performance of a two wheeled human transporter. *IFAC-PapersOnLine*, *53*(1), 616–621. https://doi.org/10.1016/j.ifacol.2020.06.103

Slotine, J.-J. E. ., & Li, Weiping. (2005). *Applied nonlinear control*. Prentice Education Taiwan Ltd.

Son, N. N., & Anh, H. P. H. (2014a). Adaptive backstepping self-balancing control of a two-wheel electric scooter. *International Journal of Advanced Robotic Systems*, *11*. https://doi.org/10.5772/59100

Super Segway. (n.d.). Retrieved December 28, 2023, from https://www.supersegway.com/en_closerlook.php

Taheri, A., RahimiZadeh, K., & Rao, R. V. (2021). An efficient Balanced Teaching-Learning-Based optimization algorithm with Individual restarting strategy for solving global optimization problems. *Information Sciences*, *576*, 68–104. https://doi.org/10.1016/j.ins.2021.06.064

VADIM I. UTKIN. (n.d.). *utkin1977*.

Yang, Y., Yan, X., Sirlantzis, K., & Howells, G. (2019). Application of sliding mode trajectory tracking control design for two-wheeled mobile robots. *Proceedings - 2019 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2019*, 109–114. https://doi.org/10.1109/AHS.2019.00012

Zupan, J. (1994). Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them*. In *Acta Chimica Slovenica* (Vol. 41).

APPENDICES

Appendix A: Simulink Model

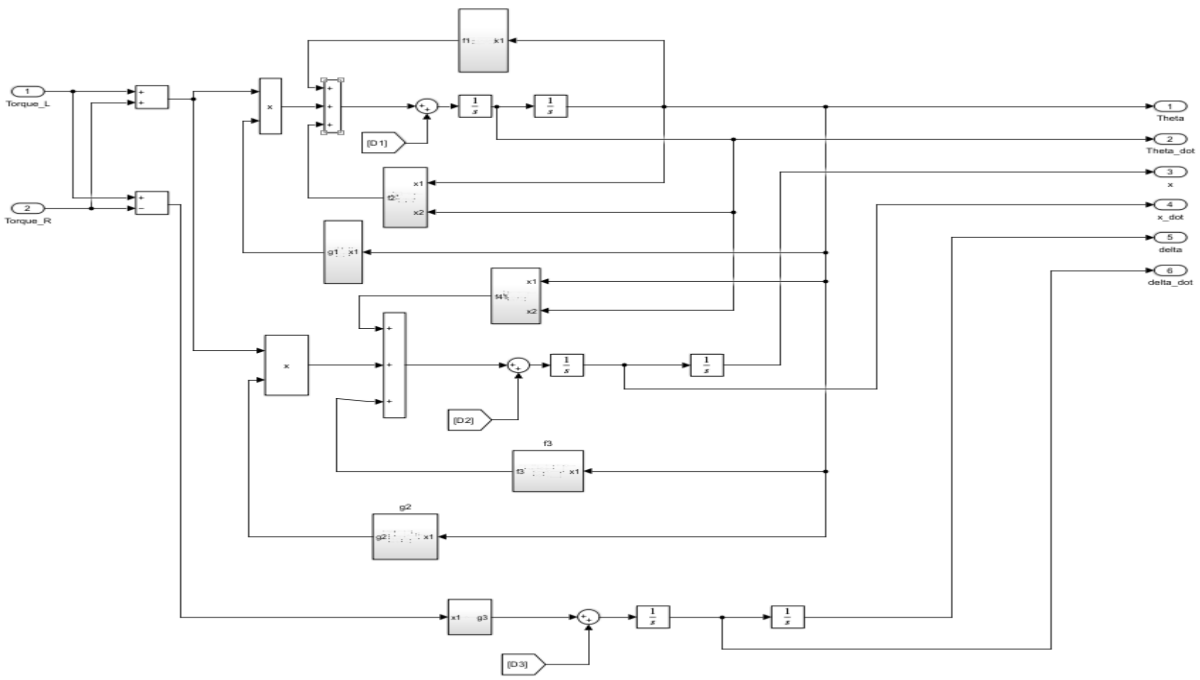


Figure A.1: Simulink model of segway with uneven terrain

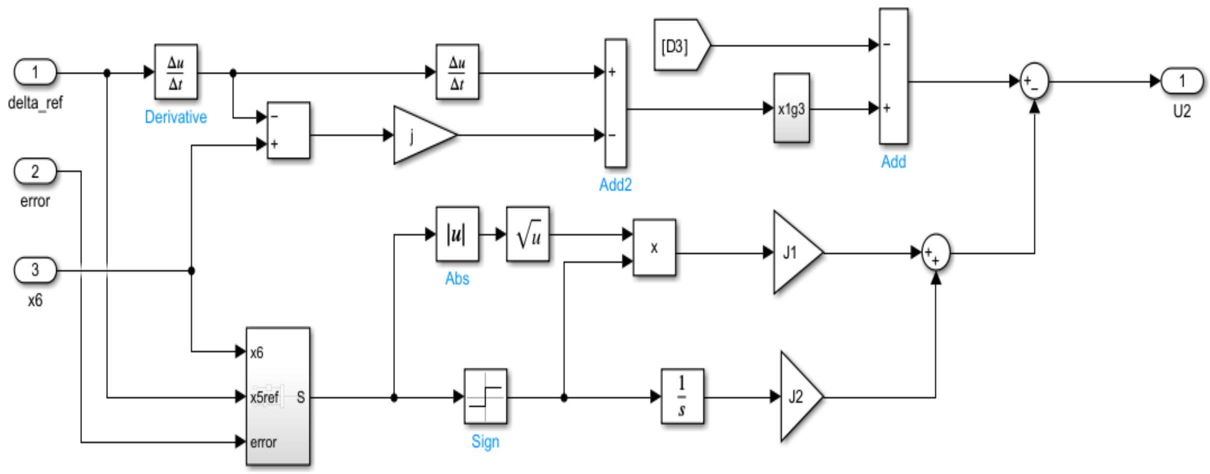


Figure A.2: Simulink model of STSMC for direction control

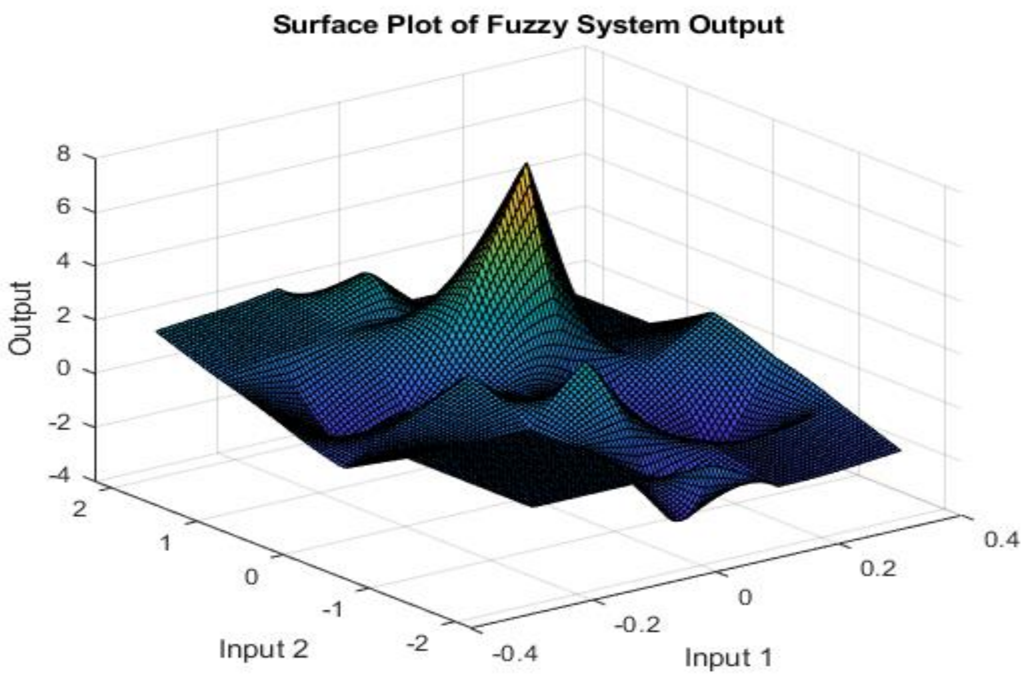


Figure A.3: Surface Plot of ANFIS System Output

Appendix B: TLBO Optimization Algorithm

```
function [bestSolution, bestFitness] = TLBO_ITAE(objectiveFunction, numVariables, numLearners, maxGenerations, lowerBounds, upperBounds)
   % Teaching-Learning-Based Optimization Algorithm (TLBO) for ITAE
   % Initialize the population of learners
   learners = repmat(lowerBounds, numLearners, 1) + ...
           rand(numLearners, numVariables) .* repmat((upperBounds - lowerBounds), numLearners, 1);
   fitness = zeros(numLearners, 1);
   % Evaluate initial population
   for i = 1:numLearners
      fitness(i) = objectiveFunction(learners(i, :));  % Call the ITAE objective function
   end
   % Initialize the best solution
   [bestFitness, bestIndex] = min(fitness);
```

```
bestSolution = learners(bestIndex, :);
% Store best fitness value for each generation
bestFitnessHistory = zeros(maxGenerations, 1);
bestFitnessHistory(1) = bestFitness;
% Main optimization loop
for gen = 1:maxGenerations
    % Teacher Phase
    % Identify the teacher (best learner)
    [~, teacherIndex] = min(fitness);
    teacher = learners(teacherIndex, :);
    % Calculate the mean of the learners
    meanLearners = mean(learners);
    % Teaching factor randomly set to 1 or 2
    Tf = randi([1, 2]);
    % Update learners towards the teacher
    for i = 1:numLearners
        newSolution = learners(i, :) + rand(1, numVariables) .* ...
                      (teacher - Tf * meanLearners);
        % Ensure the new solution is within bounds
        newSolution = max(min(newSolution, upperBounds), lowerBounds);
        % Evaluate the new solution
        newFitness = objectiveFunction(newSolution);
        % Replace if the new solution is better
        if newFitness < fitness(i)
            learners(i, :) = newSolution;
            fitness(i) = newFitness;
```

```
    end
end
% Learner Phase
for i = 1:numLearners
    % Randomly select another learner
    partnerIndex = randi([1, numLearners]);
    while partnerIndex == i
        partnerIndex = randi([1, numLearners]);
    end
    partner = learners(partnerIndex, :);
    % Learning mechanism
    if fitness(i) < fitness(partnerIndex)
        newSolution = learners(i, :) + rand(1, numVariables) .* ...
                     (learners(i, :) - partner);
    else
        newSolution = learners(i, :) + rand(1, numVariables) .* ...
                     (partner - learners(i, :));
    end
    % Ensure the new solution is within bounds
    newSolution = max(min(newSolution, upperBounds), lowerBounds);
    % Evaluate the new solution
    newFitness = objectiveFunction(newSolution);
    % Replace if the new solution is better
    if newFitness < fitness(i)
        learners(i, :) = newSolution;
        fitness(i) = newFitness;
```

```
        end
    end
    % Update the best solution found so far
    [currentBestFitness, currentBestIndex] = min(fitness);
    if currentBestFitness < bestFitness
        bestFitness = currentBestFitness;
        bestSolution = learners(currentBestIndex, :);
    end
    % Store best fitness value
    bestFitnessHistory(gen) = bestFitness;
    % Display iteration information
    fprintf('Generation %d: Best Fitness (ITAE) = %.6f\n', gen, bestFitness);
  end
  % Plot best fitness over generations
  figure;
  plot(1:maxGenerations, bestFitnessHistory, 'LineWidth', 2);
  xlabel('Generation');
  ylabel('Best Fitness (ITAE)');
  title('TLBO Best Fitness (ITAE) Convergence');
  grid on;
end
```