SYSTEM IDENTIFICATION AND CONTROL OF AUTOMATIC CAR PEDAL PRESSING SYSTEM FOR LOW-SPEED DRIVING IN A ROAD TRAFFIC DELAY

LAI CHONG JIN

B.ENG (HONS.) ELECTRICAL ENGINEERING (ELECTRONICS)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : LAI CHONG JIN

Date of Birth : 23/2/1998

Title : System Identification and Control of Automatic Car Pedal Pressing System for Low -Speed Driving in a Road Traffic Delay

Academic Session : 2021/2022 Semester II

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

LaiCJ	
(Student's Signature)	(Supervisor's Signature)
980223-04-5609	Normaniha Binti Abd Ghani
New IC/Passport Number	Name of Supervisor
Date: 17/6/2022	Date: 27/6/2022

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name	Lai Chong Jin
Thesis Title	System identification and control of automatic car pedal pressing system for low-speed driving in a road traffic delay
Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date: 27/6/2022

Stamp: ASSOC. PROF. TS. DR. NOR MANIHA ABDUL GHANI
ASSOCIATE PROFESSOR
DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
UNIVERSITI MALAYSIA PAHANG
LEBUHRAYA TUN RAZAK
26300 GAMBANG, KUANTAN, PAHANG
TEL : +609-424 6087 FAX : +09-549 2689

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis, and, in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Engineering in Electrical and Electronics Engineering.

____

(Supervisor's Signature)

Full Name :

Position :

Date :

ASSOC. PROF. TS. DR. NOR MANIHA ABDUL GHANI
ASSOCIATE PROFESSOR
DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
UNIVERSITI MALAYSIA PAHANG
LEBUHRAYA TUN RAZAK
26300 GAMBANG, KUANTAN, PAHANG
TEL : +609-424 6087 FAX : +09-549 2689

____

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

LaiCJ

____

(Student's Signature)

Full Name : LAI CHONG JIN

ID Number : EA18150

Date : 17/6/2022

SYSTEM IDENTIFICATION AND CONTROL OF AUTOMATIC CAR PEDAL PRESSING SYSTEM FOR LOW-SPEED DRIVING IN A ROAD TRAFFIC DELAY

LAI CHONG JIN

Thesis submitted in fulfillment of the requirements
for the award of the
B.Eng. (Hons.) Electrical Engineering (Electronics)

College of Engineering

UNIVERSITI MALAYSIA PAHANG

JUNE 2022

ACKNOWLEDGEMENTS

Throughout this completion of this Final Year Project, I have gained an opportunity to learn new skills and apply the knowledge related to studies in Electrical and Electronics Engineering field.

First and foremost, I very grateful to my supervisor of this Final Year Project, Associate Professor Ts. Dr. Normaniha Binti Abdul Ghani. I extremely thankful and indebted to her for sharing expertise, and sincere and valuable guidance and encouragement extended to me. Her willingness to motivate me contributed remarkably to my project. She did not hesitate to answer my questions and help me to solve the problem. I truly appreciate the time spent on refine and correcting my mistake that will happen in my project.

Besides, my sincere thanks also go to my subject coordinator, Dr. Nur Huda Binti Ramlan for coordination and guidance of this subject BEE4724. Moreover, I would like to thank the authorities of University Malaysia Pahang (UMP) for providing me with a good environment, facilities and information source that assisted me to complete my Final Year Project.

My sincere appreciation to lecturers of Faculty of Electrical and Electronic Engineering Technology who had put in effort to the lectures and always nurture and guide me with precious advice. Thank you for sharing those experiences. Without any assistance from them, I could not have done my project successfully.

Finally, I want to express a very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout this semester of study and through the process of researching and writing this report. This accomplishment would not have been possible without them. Thank you.

ABSTRAK

Kesesakan di bandar-bandar utama telah menjadi perkara biasa di kalangan masyarakat. Terperangkap dalam lalu lintas selama berjam-jam dalam keadaan duduk memerlukan kerja berulang iaitu menekan pemecut secara manual dan menghentikan pedal secara berlebihan, yang, jika dilakukan tanpa postur duduk yang betul, boleh mengakibatkan keletihan yang cepat, terutamanya untuk kaki dan belakang pemandu. Oleh itu, penyelidikan ini adalah untuk mengautomasikan mekanisme menekan dengan memodelkan penggerak untuk pedal menekan menumpukan untuk pemanduan kelajuan rendah dalam kelewatan lalu lintas jalan raya. Projek ini membentangkan pengenalan sistem dan kawalan sistem penekan pedal kereta automatik untuk pemanduan kelajuan rendah dalam kelewatan lalu lintas jalan raya. Dua parameter seperti daya dan kelajuan rendah akan diperhatikan untuk kedua-dua pengawal dan keupayaan untuk mengurangkan gangguan akan disimulasikan. Output pengawal akan menentukan daya brek kereta. Pautan selari disambungkan kepada penggerak supaya ia memberikan tindak balas yang bertentangan dengan pedal kereta. Pengenalpastian sistem diambil daripada tingkah laku dinamik sistem dan dimodelkan menggunakan data input-output yang diperoleh terus daripada pelantar eksperimen. Kerja ini menggunakan rangkaian saraf untuk memodelkan sistem kerana sistem itu mempamerkan tingkah laku yang sangat tidak linear. Makalah ini membandingkan kawalan kelajuan rendah dengan kedua-dua pengawal: PID konvensional dan pengawal logik kabur berkenaan dengan ralat overshoot dan keadaan mantap. Pengawal PID dan pengawal Fuzzy telah direka bentuk, disimulasikan dan dibandingkan dalam keupayaan mereka untuk mengawal kelajuan kereta. Kedua-dua pengawal kemudiannya dilaksanakan dan diuji dalam sistem penekan pedal kereta automatik. Keuntungan pengawal ditala menggunakan algoritma metaheuristik iaitu Algoritma Swarm Partikel (PSO) untuk nilai optimum parameter pengawal kabur. Parameter pengawal dioptimumkan berdasarkan Ralat Integral Squared (ISE), Kesilapan Mutlak Integral (IAE), Kesalahan Absolute Time Integral (ITAE) dan Kesalahan Mean Squared) MSE. Penilaian perbandingan kedua-dua pengawal dilaporkan dan dibincangkan. Ini menunjukkan bahawa pengawal logik Fuzzy berprestasi lebih baik daripada pengawal PID dengan pengurangan 1.2724% dalam ralat keadaan mantap dan 3.64% dalam overshoot masing-masing.

ABSTRACT

Congestion in major cities has been a typical occurrence among communities. Being stuck in traffic for hours in a sitting position necessitates recurrent chores of manually pressing the accelerator and stop pedals excessively, which, if performed without the proper sitting posture, can result in quick weariness, especially for the driver's leg and back. Therefore, this research is to automate the pressing mechanism by modelling an actuator for pedals pressing concentrating for low-speed driving in a road traffic delay. This project presents system identification and control of automatic car pedal pressing system for low-speed driving in a road traffic delay. Two parameters such as force and low speed will be observed for both controllers and the ability to attenuate disturbance will be simulated. Output of the controller will determine the force of the car brake. The parallel linkage is connected to the actuator such that it provides an opposite reaction to the car pedal. The system identification is taken from the dynamic behavior of the system and modelled using input-output data acquired directly from the experimental rig. The work utilized a neural network to model the system as the system exhibits highly nonlinear behavior. This paper compares the low-speed control with both controllers: conventional PID and fuzzy logic controller with respect to overshoot and steady state error. A PID controller and a Fuzzy controller were designed, simulated, and compared in their ability to control the speed of the car. Both controllers were then implemented and tested in automatic car pedal pressing system. The controller gains were tuned using metaheuristic algorithm which is Particle Swarm Algorithm (PSO) for optimal values of fuzzy controller parameters. The controller parameters are optimized based on Integral Squared Error (ISE), Integral Absolute Error (IAE), Integral Time Absolute Error (ITAE) and Mean Squared Error) MSE. The comparative assessment of both controllers was reported and discussed. It shown that Fuzzy logic controller performs better than PID controller with 1.2724% reduction in steady state error and 3.64% in overshoot respectively.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PID	Proportional Integral Derivative
ANN	Artificial neural networks
NN	Neural network
AI	Artificial Intelligence
MLP	Multilayer perceptron
MIMO	Multiple-input multiple-output
NARX	Nonlinear autoregressive exogenous inputs
MPC	Model predictive control
OE	Output error
ARMAX	Autoregressive Moving Average with exogenous inputs
MLPNN	Multilayer perceptron neural network
MSE	Mean square error
FLC	Fuzzy logic control
SISO	Single-input/single-output
LM	Levenberg-Marquardt
FIS	Fuzzy Inference System Editor
OSA	One-step-ahead
PSO	Particle swarm optimization
IAE	Integral Absolute Error
ISE	Integral Square Error
ITAE	Integral Time Absolute Error
GA	Genetic Algorithm
FA	Firefly Algorithm
ABC	Artificial Bee Colony Algorithm

CHAPTER 1

INTRODUCTION

1.1 Project Background

Vehicle traffic congestion occurs when there are too many vehicles on the road and the traffic flow is impeded. Traffic congestion disrupts users' regular activities and produces pandemonium on the road. Time spent on the road has numerous negative implications on productivity, social behaviour, the environment, and economic costs. Congestion is worsening, resulting in situations where traffic flow is always unpredictable and uncontrollable, such as floods, accidents, and road repair [1].

There is no doubt that an automatic car would not have made sense without an effective and real-time responsive car pedal pressing system. The cause of many of these accidents is driver distraction and failing to react in a timely manner. An advanced system of auxiliary functions has been devised to aid in the avoidance of such an accident and the reduction of the effects of a collision should one occur. This is accomplished by shortening the overall stopping distance [2]. As a result, the car pedal should have a solid software system to aid a driver when driving.

Road congestion studies show that about half of traffic delay is bottlenecks (traffic demand exceed roadway capacity). The other half is attributable to temporary disruptions of the transportation system like traffic incidents, work zones, poor signal timing and bad weather [3].

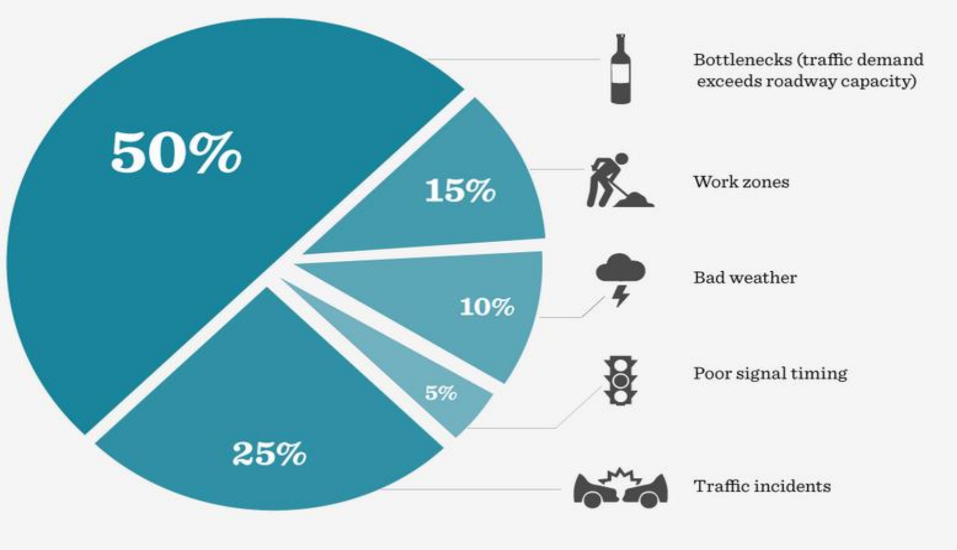

Figure 1-1 Percentage of road traffic congestion in 2019

Most of the drivers' actions are based on their prior experience rather than precise mathematical calculations. The model of car pedal pressing is a highly nonlinear function, therefore finding a correct model is difficult. Many researchers have developed fuzzy logic systems for autonomous driving controllers because they mimic the performance of a skilled human operator in language tulles without requiring the use of a mathematical model [4]. Because of the increasing traffic density, car pedal pressing control systems for passenger cars are becoming less useful. It is rarely possible to travel at a predetermined speed. An intelligent car pedal pressing control system, on the other hand, must behave similarly to a skilled human driver to get approval. As a result, the following distance and control dynamics must be adjusted to meet the needs of each unique driver. Because the driver's experience may be easily translated into rules, applying fuzzy logic to intelligent car pedal pressing control appears to be an appropriate technique to achieve this human behaviour [5].

Artificial neural networks (ANN) system identification is also used as a learning system, well recognized by the computer science community and with many applications. The controller uses ANNs to create a model of the tracks to estimate the car's trajectory and goal speed. Data received from a human was used to train the ANNs. The results of this research are promising in terms of forecasting trajectory on new tracks; nonetheless, the target speed is slower than that of humans on the same tracks [6].

1.2 Problem Statement

Congestion in major cities has been a typical occurrence among communities. Even though the government and state agencies have made numerous steps to address traffic congestion, the situation appears to be worsening as car production and sales continue to rise year after year, regardless of traffic congestion. Being stuck in traffic for hours in a sitting position necessitates recurrent chores of manually pressing the accelerator and stop pedals excessively, which, if performed without the proper sitting posture, can result in quick weariness, especially for the driver's leg and back. It will have a long-term harmful impact on the driver's health, especially in terms of the physical, psychological, and emotional health. Therefore, this research is to automate the pressing mechanism by modelling an actuator for pedals pressing concentrating for low-speed driving in a road traffic delay.

It is known that the intelligent techniques have a strong capability of learning and cognition, as well as a good tolerance to uncertainty and imprecision. Due to these properties, they can be applied successfully to Intelligent Vehicle Systems. Fuzzy Logic is very adequate to build qualitative (or linguistic) models, of many kinds of systems without an extensive knowledge of their mathematical models. The car pedal is one of the most important actuators for driving. Even if there are few examples of successful PID control of an automatic car pedal pressing system, the most of them are study cases. In real-world circumstances, the complexity and unpredictability of an actual system become crucial problems for keeping the system stable. To maintain system stability, intelligent controller is suggested for the automatic car pedal pressing system.

Other than that, optimization is often done to tune the input and output scaling factors of controller because they have significant effects on the dynamic of controller. The performance of controller was increased by scaling the input and output factors of controller. The system in this project is about automatic car pedal pressing, PID Fuzzy logic controllers and optimization used in Particle Swarm Optimization (PSO) process. The presence of proportionality constant, integral constant and derivative which are found in regular PID controller along with integral order and derivative order found in fractional-order PID controller increases the difficulty of designing controller. Thus, PSO is implemented to obtain the control gain parameters.

1.3 Objective

This project aims to control the car pedal pressing system using fuzzy logic control technology with the aid of artificial neural network algorithms and techniques. To achieve the main goal, the objectives of this project are determined:

- To develop system identification based on input and output data of pedal pressing system hardware rig by using Artificial Neural Network (ANN)

- To develop intelligent controller for the automatic car pedal pressing system by using fuzzy logic and compare with PID controller in term of system performance

- To implement PSO algorithm for control gain parameters to achieve optimum system performance

1.4 Project Scope

There are several scopes acquired to be fulfilled in this work. In general words, these scopes are covering up the objectives to achieve the expected outcome while accomplishing the project.

- Develop system modelling for testing, validation of neural network

- Provide performance of neural network with different number of delay signals

- To optimize input and output gains (scale factor) for designed controllers using Particle Swarm Optimization

- Provide good input-output graph for this project

- To compare performance of different fitness function between PID and Fuzzy Logic controller

- Conduct simulation in MATLAB Simulink.

1.5 Thesis Outline

This thesis covers five chapters including Introduction, Literature Review, Methodology, Result and Discussion and Conclusion.

Chapter 1 is introducing the overall project that is automatic car pedal pressing system. In this chapter, the project background, problem statement, objective as well as project scope will be showcased.

Chapter 2 presents the findings based on literature review of this project that includes the type of controllers and optimization method. The reference sources from the research papers, books, articles, journals, and websites.

Chapter 3 is about the methodology used in this project. It shows the layout for the entire project. It starts with flow chart to give an overview of project provide procedure and information in the steps of project.

Chapter 4 shows and discusses the result obtained in this project. Result including system identification, optimization performance and the optimized controller performance indices.

Chapter 5 provides the conclusion for the whole project. The limitation and future improvements will be discussed in this chapter. It is used to enhance the system so that is feasible in real-world situation.

CHAPTER 2

LITERATURE REVIEW

2.1 Automatic Car Pedal Pressing System

The automatic car pedal pressing is a self-driving car that uses intelligent decision-making and control technologies to transfer all real-time driving functions to the automobile automation system. The development and implementation of self-driving automobiles has piqued the interest of leading automotive and technology companies. Despite this, the majority of research has been focused on automating specific driving operations such as parking, braking, and cruise control [7]. Few studies and research have focused on producing semi-autonomous or fully autonomous vehicles, but most of these designs ignore key aspects that influence driving automation, such as vehicle and weather conditions.

Using the fully parameterized technique, the brake pedal dynamic behaviour can be chosen nearly arbitrarily in terms of pedal force and pedal movement. The advantage is that regardless of the wheel braking mechanism used, automakers can apply their preferred braking experience at the pedal. After numerically solving the necessary dynamic model for the brake pedal, the time indexed reference position of the pedal in relation to the braking force is obtained [8].

The car pedal pressing control problem is highly non-linear and has been used to manifest the ability to control techniques. Since car pedal pressing is a non-linear problem, it may be managed with the right strategy. Various control techniques, both conventional and intelligent, have been incorporated in the system, all of which can stabilise the car pedal pressing.

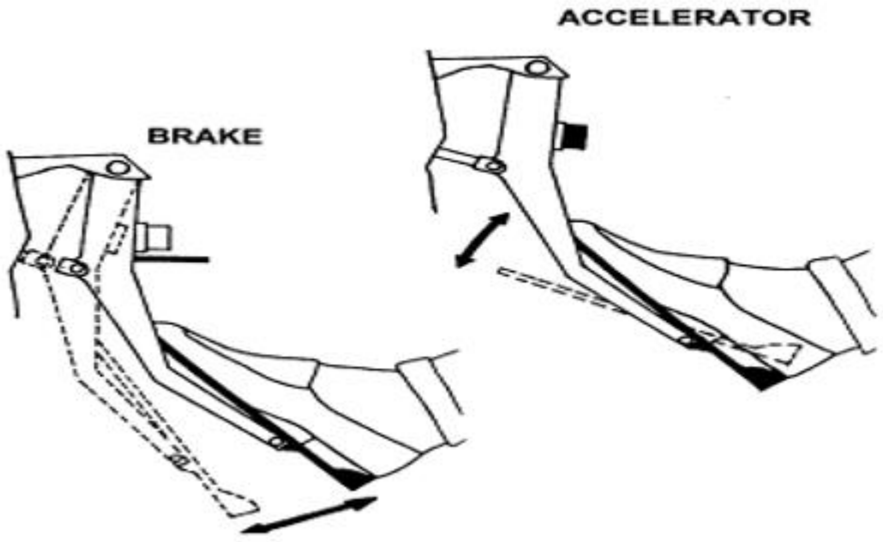

Figure 2-1 Schematic diagram of car pedal pressing

2.2 System Model

As the one of the strategies in any system study, system modelling is a vital stage in conducting research. Iterating between modelling and simulation will improve the system's design quality early on, lowering the number of faults discovered later in the design phase. Due to the very non-linear behaviour of car pedal pressing, developing an appropriate mathematical dynamic model is a difficult challenge. Differential equations, which are difficult to acquire, are used to explain the system dynamics of car pedal pressing, or in other words, the car braking system. Additionally, there is another approach that simply requires experimental input and output data to demonstrate system dynamics.

2.2.1 Mathematical Modelling

A system model of car pedal pressing system can be created by using mathematical approach. The mathematical model of dynamic stabilization system for autonomous car in [9] shows equations of linear movement and limit speeds, equation system for wheel speeds on the turn, virtual sensor algorithms for car speed, turn angle, and the additional component of yaw rate, for identifying top values of friction coefficients between wheels and road surface, and for controlling brakes by actuator. Next, reference in [4] shows the engine dynamic, skidding, slip and friction of the car is disregard. Therefore, they are using Newton's second law of motion, the force *F* causes acceleration of the car. The car brake system is described by differential equations that explain the system behaviour related to force and acceleration of the car. In this paper, the approach is about Newton's second law of motion is used to model the car pedal pressing system because it can provide precise mathematical model of the car.

2.2.2 System Identification

During the system identification stage, which is a function approximation method, the dynamic model of the system is built based on observable input-output data [10]. This is because to its ease of use and ability to discover nonlinear relationships from a set of data. Studies show that system identification can be performed either in open-loop or closed-loop depending on the characteristics of the system. System identification on a control in self-driving car is conducted by collecting the input-output data from open-loop system and carry out [11]. Due to its capacity to handle restrictions and operate near to state boundaries in multiple-input multiple-output (MIMO) systems, model predictive control (MPC) has proven to be an efficient solution for trajectory reference tracking, particularly in self-driving cars.

The input layer of the network receives training inputs in a forward sweep, allowing the output of each element to be computed layer by layer. Backpropagation training is the process of teaching the network to correlate input vectors with appropriate output vectors using input and target vectors. The neural network's usefulness in building dynamic models that are realistic of genuine nonlinear plants based on the interactions between the inputs and outputs is exploited, according to the study. Experimentation, model structure selection, model estimate, and model validation are the four processes in the system identification process.

2.2.3 Neural Network

For the model structure selection, the neural network nonlinear autoregressive exogenous inputs (NARX) model has been proven too readily represent any nonlinear, discrete, time-invariant system. Although high system order is good, increasing it may have an impact on some dynamic features such as stability. It's also easier to use, non-recursive (unlike nonlinear models-based output error (OE) and Autoregressive Moving

Average with exogenous inputs (ARMAX), which rely on current and future outputs), and more stable because it doesn't require feedback [12].

For the model estimation, the neural network structures are selected for use in the network training of the model. The model estimation method is guided by two factors: the NN structure's simplicity and computing ease. Therefore, an input layer, a hidden layer, and an output layer were created in a feedforward multilayer perceptron neural network (MLPNN) structure. The Levenberg-Marquardt method was used to train the network because of its rapid convergence and durability. This is since it has the smallest mean square error (MSE) and the most epochs. The Levenberg-Marquardt training algorithm is preferred above the other approaches because it improves with time [13].

For the model validation, the fitness analysis of three one-step ahead predictions for sigmoidnet, wavenet, and neuralnet structures to the validation data is presented in [14], which shows that the performance of the trained network based on the validation data with the quality of the identification is indicated by the mean square error.

2.3 Strategic Control Approach

The car pedal pressing system is essential in the evaluation and comparison of various control theories. Control system deals with the dynamic behaviour of systems, and it modifies the system's inputs to produce the desired effect on the output. A proper control system, such as a conventional controller or an intelligent controller, is necessary to control the car pedal pressing system.

2.3.1 Conventional Controller

The Proportional-Integral-Derivative (PID) control offers various advantages, including a simple structure, good overall control performance, robust design, and easy implementation [15]. In contrast, the PID approach necessitates the use of supporting algorithms to discover and finetune its hyper-parameters. The difficulty of obtaining excellent, ideal values for these hyper-parameters that appropriately fit the environment is due to the complexity of vehicle dynamics, uncertainty of external disturbances, and the vehicle's nonholonomic restriction [16].

Reference in [17] shows that a further step towards the prospects of autonomous driving is presented. The major objective is to create a car that can autonomously follow a pre-defined trajectory or route determined by the autonomous car's projected path planner in both highway and urban traffic while maintaining the best level of comfort. Sensors are used in the suggested control system to provide the real speed of the controlled car as well as the relative position of the car in relation to the provided trajectory. The controller then calculates the optimal steering wheel motions. These actions regulate the vehicle's direction, and they're computed using a Proportional-Integral-Derivative (PID) control algorithm.

2.3.2 Intelligent Controller

Because of the autonomous car's complexity and nonlinearity, developing a controller using traditional methods is particularly difficult when the autonomous car's effective parameters and inputs are unknown. Fuzzy logic control (FLC) approaches, on the other hand, are well-known for their applicability and capability in the language description of complex systems, and they can be utilised to create and convert linguistically conveyed human experience into appropriate automatic control strategies [18].

Reference in [19] shows car brake pedal system is the most important system in a car. The automated car braking system's goal is to provide an automated control system that keeps a safe driving distance from objects while in traffic. At a given range, the system properly recognises a barrier ahead and devises a strategy to avoid a collision by slowing the car. Driving becomes more enjoyable and less stressful as a result. The system is created with MATLAB's fuzzy logic toolbox and then simulated to test how well the car's braking system works.

2.4 Summary

The discussion for system modeling and strategies control approaches are summarized in the tables below

Table 2-1 Summary of system modelling method

Authors	Year of Publication	Plant	Modelling Method
Bassey, E.F. Udofia, K.M.	2019	Automatic car braking system	Mathematical modelling
A. M. Saikin, S. E. Buznikov, N. S. Shabanov, and D. S. Elkin	2018	Autonomous vehicle dynamic stabilisation system	Mathematical modelling
Dalimus, Zaini	2014	Car braking system	Mathematical modelling
Salt Ducaju, Julian M, Tang Chen, Tomizuka Masayoshi, Chan Ching Yao	2020	Self -driving car system	System Identification
Albelihi, K. Vrajitoru, D.	2019	Autonomous car driving system	System Identification with neural network
Dahunsi O. A, Pedro J. O, Nyandoro O. T.	2010	Servo- Hydraulic vehicle suspension System	System Identification with neural network
Germann St, Isermann R.	1995	Autonomous car speed control system	Neural network

Table 2-2 Summary of control approaches

Authors	Year of Publication	Plant	Controller
Widaa,Abdulrahman H.A. Talha, Waddah Abdelbagie	2017	Autonomous (self-driving) car system	Fuzzy Logic
Muller, R. Nocker, G.	1992	Automated car speed system	Fuzzy Logic
Mamat, M. Ghani, N. M.	2009	Automated car braking system	Fuzzy Logic
H.L Zhan, L. Deng, S.J Xue	2012	Smart Car Speed Control System	Fuzzy Logic and PID
Hirulkar, Sachin Damle, Manish Rathee, Vishal Hardas, Bhalchandra	2014	Automated car braking system	Fuzzy Logic and PID
P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, T. Mei	2012	Autonomous vehicle system	PID
Farag, Wael	2019	Self-driving cars system	PID

CHAPTER 3

METHODOLOGY

3.1 Car Pedal Pressing Hardware Configuration

The car pedal is pressing or releasing depend on the distance between the cupboard (front car) and speed of the actuator. The components that used to set up this car pedal hardware are Arduino Uno, car pedal with linear actuator, 12V power supply, ultrasonic sensor, and L298 motor driver. Figure 3.1 shows the car pedal pressing testing with all components used to build up this hardware.

Figure 3-1 Car pedal pressing testing

3.1.1 Data Collected in Simulation and Hardware System

Since the car pedal pressing is open-loop unstable in simulation, it is recommended to have a controller for stabilization to increase system run time. Data collected method is used to collect the data from input and output. The input of car pedal pressing is force when pressing the pedal whereas the low speed of the car is collected as the output data in simulation. The input and output data are labelled as shown in figure 3.2.

In the car pedal pressing hardware system, the cupboard (front car) is held and maintained at 1 meter away from ultrasonic sensor. After that, the cupboard (front car) will start to go toward ultrasonic sensor, the actuator starts to retract, and car pedal is pressing. When the cupboard (front car) goes away from the ultrasonic sensor, the actuator starts to extend, and car pedal is releasing. Then, the system is always repeat the same action in a few minutes to collect the input and output data in form of distance and speed of the actuator when retracting and extending with applying the PID or fuzzy logic controller in this process.

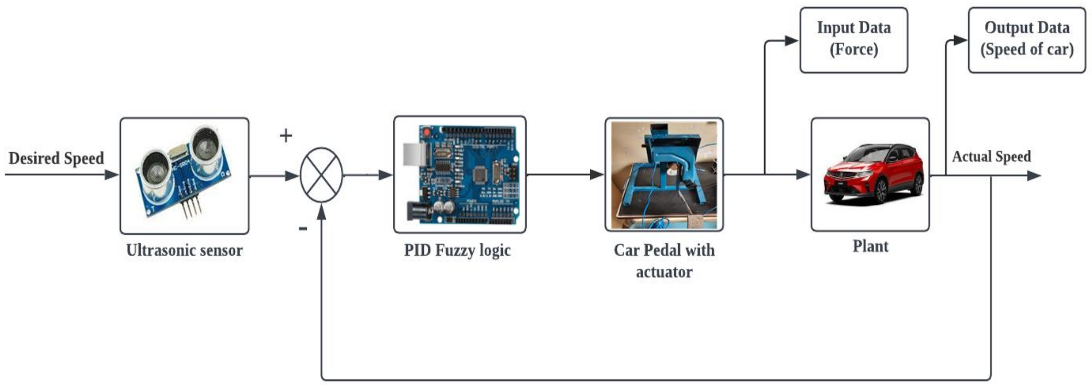

Figure 3-2 Input-Output collected from car pedal pressing hardware

3.2 Block Diagram

Figure 3.3 presents the illustration in simulation using MATLAB Simulink; the reference input is set between 2 m/s (8km/h) to 7m/s (25km/h) in low speed during traffic jam. Once on the traffic jam road and at our desired output, pressing a pedal that switches the controller from manual mode to automatic mode.

The low speed of the car now we close the loop and switch from manual to automatically becomes the set point. The controller then continually computes and transmits corrective actions to the pedal to maintain measured low speed at set point. PID and Fuzzy logic controller is implemented to control the car pedal with actuator. Since the scaling of membership function in PID fuzzy logic controller is uncertain, particle swarm optimization is added to help the tuning of membership function scaling.

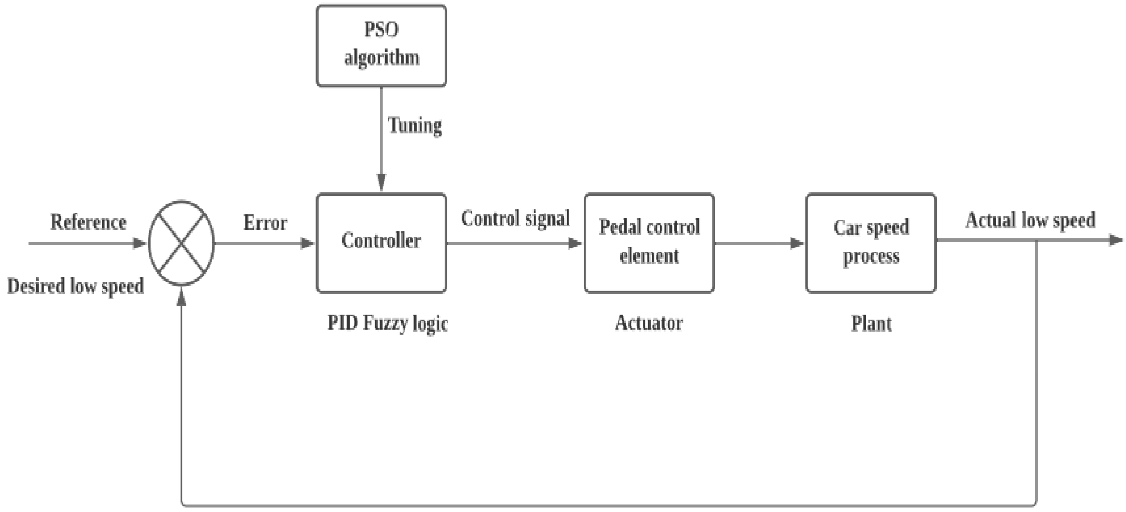

Figure 3-3 Block Diagram of Car Pedal Pressing System in Simulation

3.3 Flow Chart

Figure 3.4 shows the system flow chart of this project. It begins with experimental data acquired from car pedal pressing system that was previously constructed and used as the project's system model. The input and output data of car pedal pressing system are the force when pressing the car pedal and low speed of the car respectively.

Following that, these experimental data are sent into system identification analysis to do further model estimation. Black-box identification is used to estimate and validate nonlinear models from single-input/single-output (SISO) experimental data to find the one that best represents the system dynamics. Moreover, construct the artificial neural network for system identification. The experimental data will be trained and tested for various configurations to provide an accurate system with the lowest MSE.

Lastly, fuzzy logic and PID controller are designed and constructed to control car pedal pressing system. In this project, both controllers are designed with normalized membership function and compensate with scale factor to scale the range of gain range. Particle Swarm Optimization was used to optimal gains for both controllers. Performance indices such as MSE, IAE, ISE and ITAE will be accessed to determine the optimization of controllers. Both controllers have similarity which is to minimise the error.

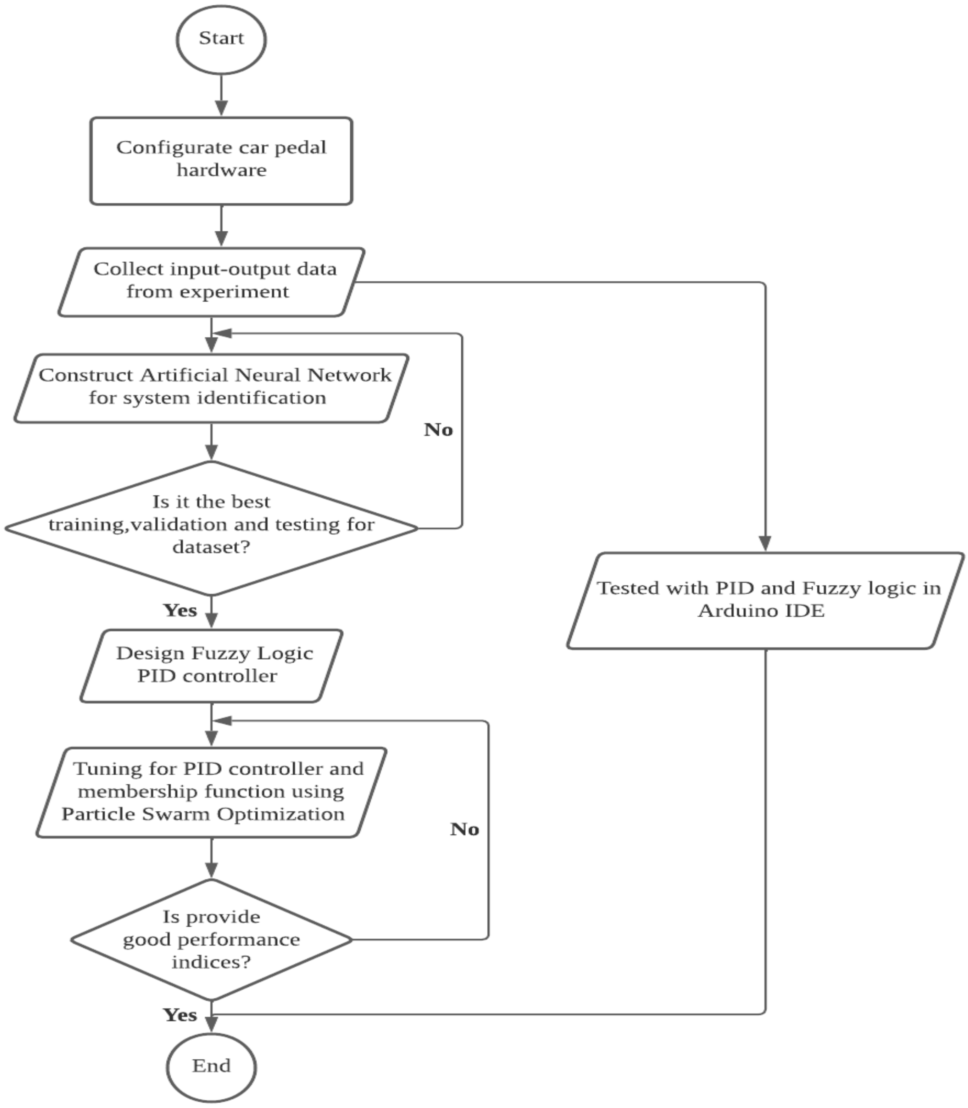

Figure 3-4 Flow Chart of the project

3.4 System Modelling

Due of its simplicity and effectiveness in non-linear system identification, the system identification approach was used for producing the system model of the car pedal pressing mechanism. To create an accurate plant model, input and output data are collected, then the data will be sent to neural network system for training, testing, and validation. The trained network may not always respond the same way as the actual system, so that network testing and validation are essential for determining the trained network's accuracy.

3.4.1 System Identification

The ability of system identification to find an accurate model of dynamical systems has gotten a lot of attention. As a result, there was a strong desire to apply the system identification technique to create a dynamic model that characterised the car pedal pressing system using data from a real plant. Using these methods, the produced models were able to develop the system's dynamic characteristics while avoiding the complications of mathematical and physical model development.

There are a few phases involved in system identification in general. Data collection, model structure selection, model estimation, and model validation are the four steps. Data acquisition plays a vital part in modelling the dynamic system since it allows for the collection of several sets of data. The major aim of identification is to estimate the model parameters after the model structure has been identified. The estimated model must have properties that are comparable to those of the true model and be able to predict future output values. After obtaining a model of the system, it is necessary to verify the model. Model validity tests are processes for determining if a fitted model is adequate [20].

In the figure 3.5, data obtained from a previously built car pedal pressing system that was used as the project's system model. The input and output data of car pedal pressing system are the force when pressing the car pedal and low speed of the car respectively. The Non-linear Autoregressive with External Input (NARX), state space and transfer function model had been estimated in the process of system identification. The highest fit of estimation to system is NARX model which is 88.1%, followed by state space model which is 87.09% and the lowest fit of estimation to system is transfer function which consist of 31.35%.

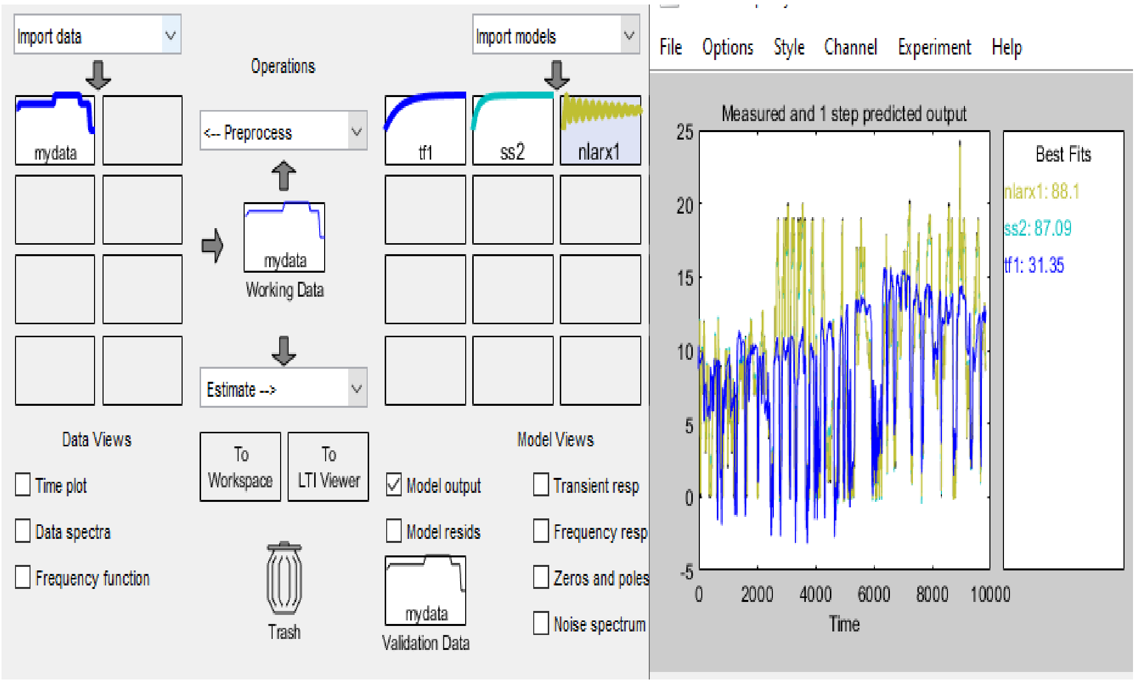

Figure 3-5 Black-Box Identification

3.4.2 Artificial Neural Network

System modelling generally applies two basic processing elements when using the perceptron-based neural networks and the neuron-based function. The nonlinear model of a neuron is called the perceptron. The basic neural model is the two elementary components which are a linear combiner and a non-linear activation function. A linear combiner calculates the product of input vector, x of the neuron and the parameter vector, w. And a non-linear activation function was subject to the output of the linear combiner. the ANN identification technique has been increasingly used to a variety of nonlinear systems. Non-linear Autoregressive with External Input (NARX) has an algebraic link between prediction and past data only, and so has a predictor without feedback, which simplifies the model. Though the NARX model structure exhibited reasonable approximation, the model's predictor with feedback could lead to overfitting estimation.

Due to its simplicity and high fit of estimation after identifying the system, the NARX model is used as the model framework in this project since car pedal pressing is extremely nonlinear. The nonlinear generalisation of the ARX model, which is a standard tool in linear black-box identification, is the NARX model. The neural network is used to estimate the nonlinear component of the ARX structure. Neural Networks are mathematically designed to mimic the biological neurons in the brain.

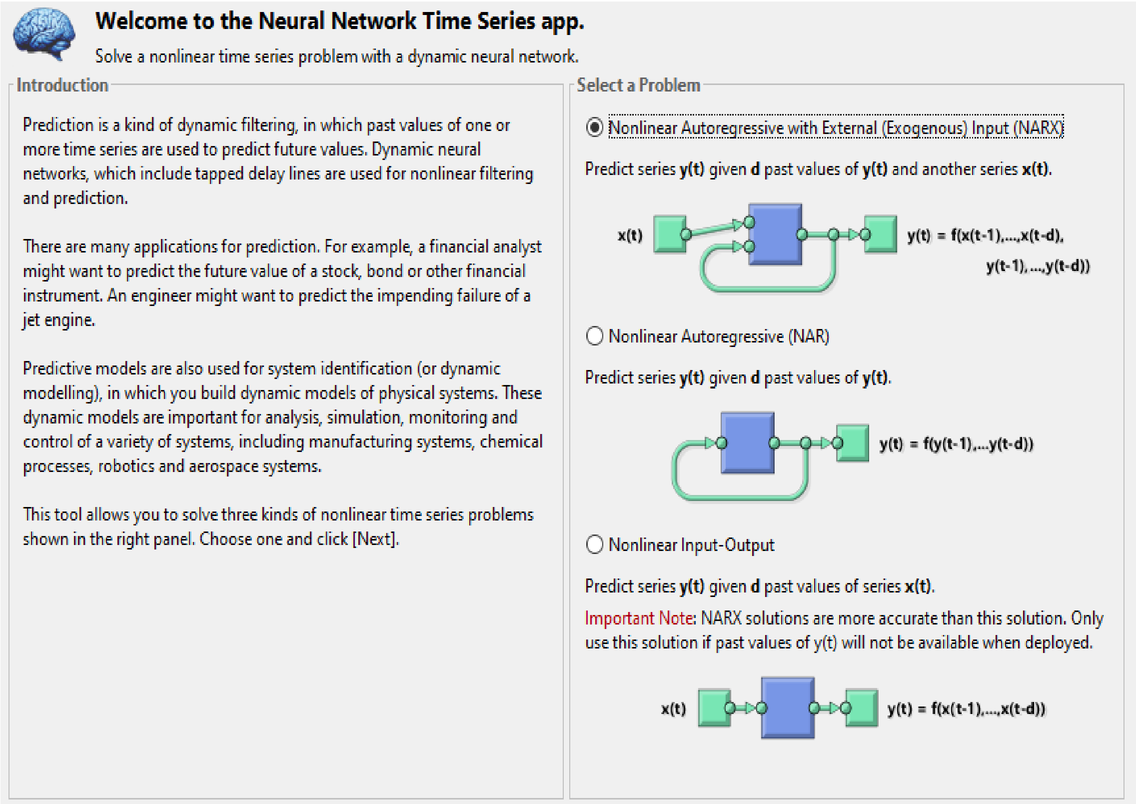

Figure 3-6 Neural Network Time Series Toolbox in MATLAB

Total of 9842 samples of datasets which collected from the car pedal pressing is used for neural network training, testing, and validating. The data is divided into three parts where training data consists of 70% of the dataset, testing and validating data are 15% each. This step is critical for determining the model's accuracy in simulating actual car pedal pressing. After the network has been trained, it will be tested and validated, and the model with the lowest MSE will be determined.

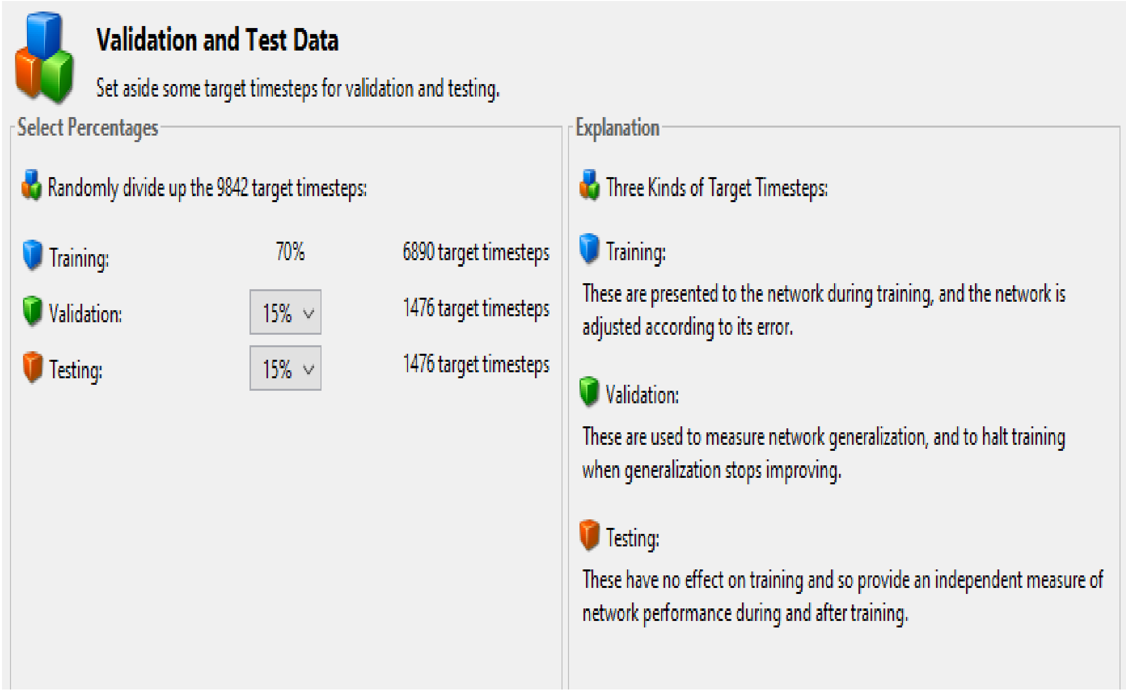

Figure 3-7 Data for Training, Validation dan Testing

In NARX neural network structure, the number of delay signals, the number of neurons in the hidden layer, and the error are the three essential elements to consider during the procedure. The third factor is assessed along the process of getting the best number of delay signals and the structure for each model.

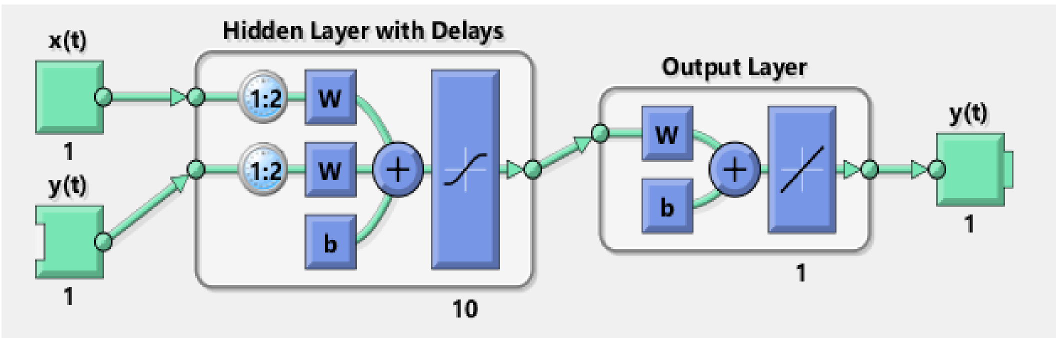

Figure 3-8 NARX neural network with 10 neurons in hidden layer and 2 number of delay signals.

3.4.2.1 Multi-layer perceptron

The back propagation for multi-layer perceptron (MLP) neural network for modelling three sets of a Single Input Single Output (SISO) car pedal pressing system.

Because of its capacity to offer a basic model and estimate a highly sophisticated formula linkage, the MLP is the most popular of the neural network family.

The MLP is made up of one layer of nodes that serves as the input layer and a second layer that serves as the NN's output layer, with multiple intermediate or hidden layers in between. By its faster convergence time, Levenberg-Marquardt (LM) is used for network training, despite requiring more memory than other algorithms.

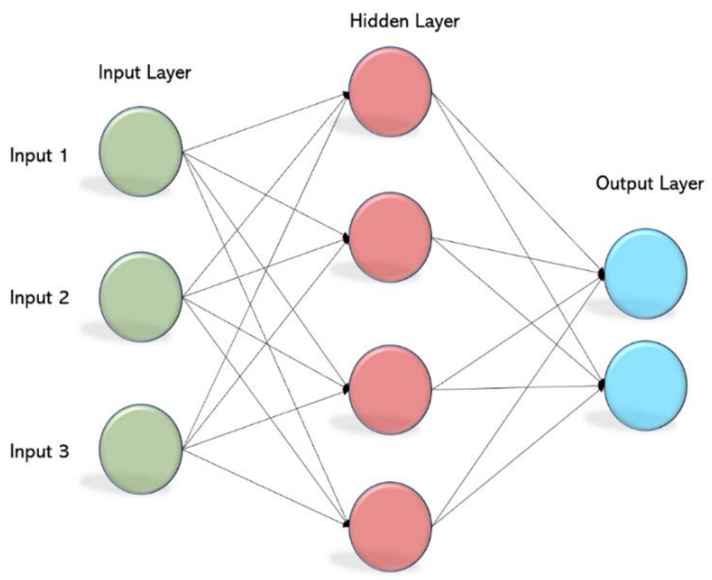

Figure 3-9 Multi-Layer Perceptron (MLP) structure

3.4.2.2 Best Validation Performance

According to Figure 3.10, the best validation performance achieved during the training process is Mean Squared Error (MSE) of 0.44628 at epoch 65. This shows that the neural network has highest accuracy with the smallest error at the epoch of 65.

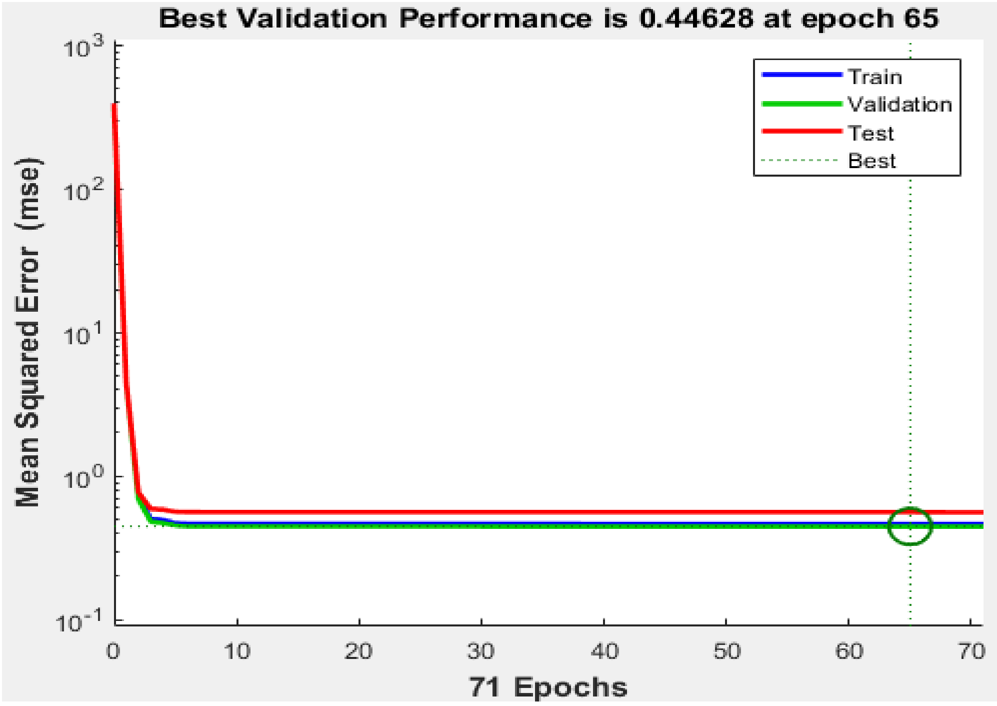

Figure 3-10 Best validation performance achieved during the training process

3.4.2.3 Regression Plot

The regression graphs in Figure 3.11 show the network outputs in relation to targets for training, validation, and test sets. For a perfect fit, the dataset should fall along a 45-degree line to make the network outputs identical to the targets. For all datasets, the fit is reasonable, all R values of 0.90 or higher in each case. The neural network achieves the best fit with the training data sets, with R values of 0.99.

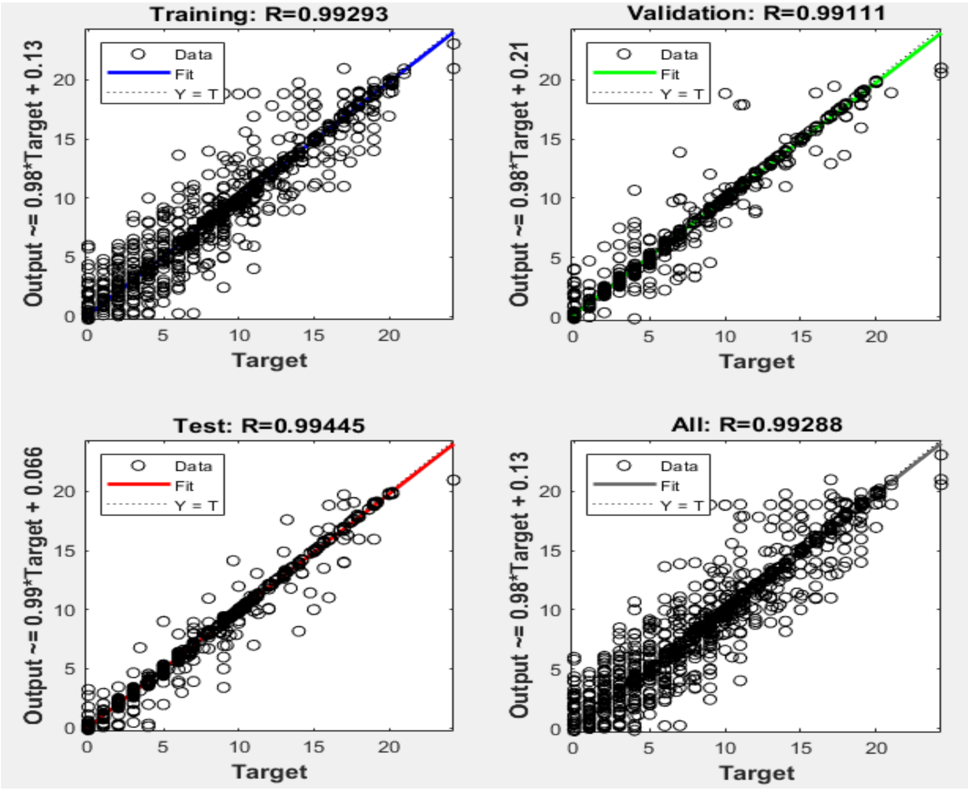

Figure 3-11 Regression plot and R values of the neural network

3.4.2.4 Error Histogram

The error histogram is defined as the histogram of the errors between target and predicted values after training a feedforward neural network, as shown in Figure 3.12. Bins refer to the number of vertical bars shown on the graph. In this case, the whole error range is separated into 20 smaller bins. The number of samples from our dataset that fall into each category is represented on the Y-axis. We have a bin in the middle of the plot that corresponds to an error of 0.1513, and the height of that bin for the training dataset is approximately 6000, and the height of that bin for the validation and testing dataset is between 6000 and 9000. It means that many samples from our different datasets have an inaccuracy in the range illustrated below. In this situation, the zero-error point is contained within the bin with a centre of 0.1513.

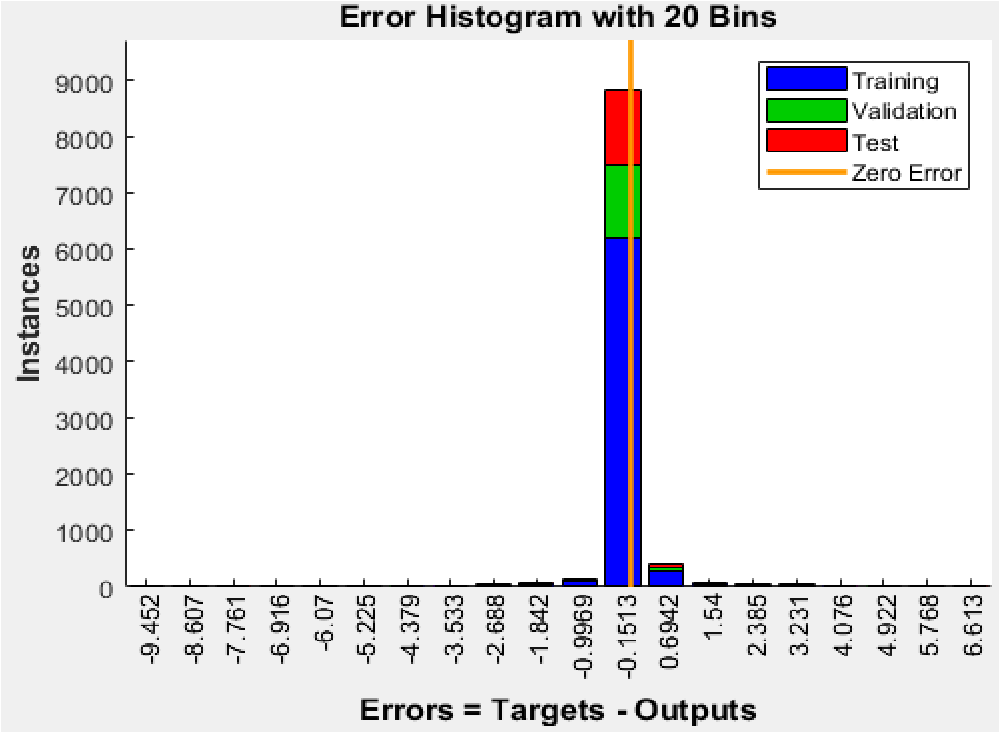

Figure 3-12 Error Histogram of the neural network

3.5 Controller Designing

It will deal with a Fuzzy Logic Controller (FLC) for an automated car pedal pressing mechanism in the control system. The system's response is simulated using MATLAB's Fuzzy Logic and PID tuner Toolbox. This controller's job is to brake an automatic car pedal when it gets too close to another car within a certain range. The Fuzzy Logic Controller is created in MATLAB using the Fuzzy Logic Toolbox.

3.5.1 Conventional PID Controller

The fundamental block diagram of a PID system in Figure 3.13 is used to create a PID. A plant is a system that must be managed. The controller, on the other hand, is the device that provides the plant with stimulation and is designed to control the entire system behaviour.

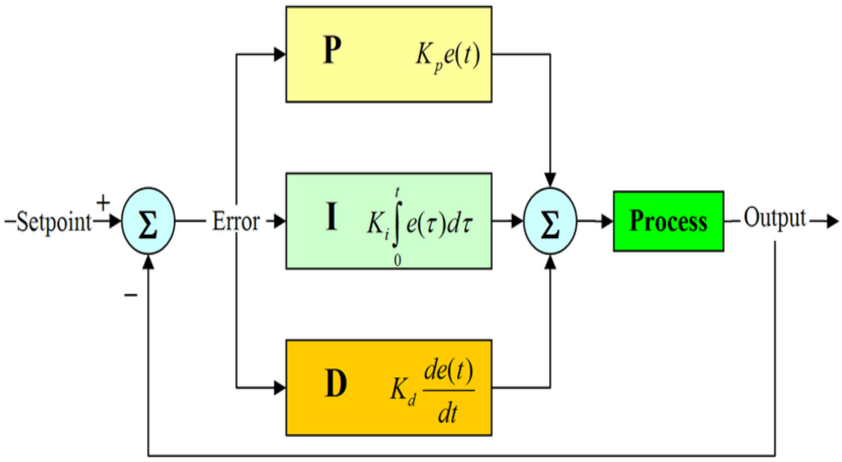

Figure 3-13 Basic block diagram of PID system

This is a technique for improving the controller's performance by auto-tuning it off-line using repeated attempts. Because the controller's convergence cannot be assured throughout the trials, they cannot be carried out on an actual plant. The final tuning of PID controller is given in trial where (*Kp* = 2.2267, *KI* = -0.0565 and *KD* = 19.6979).

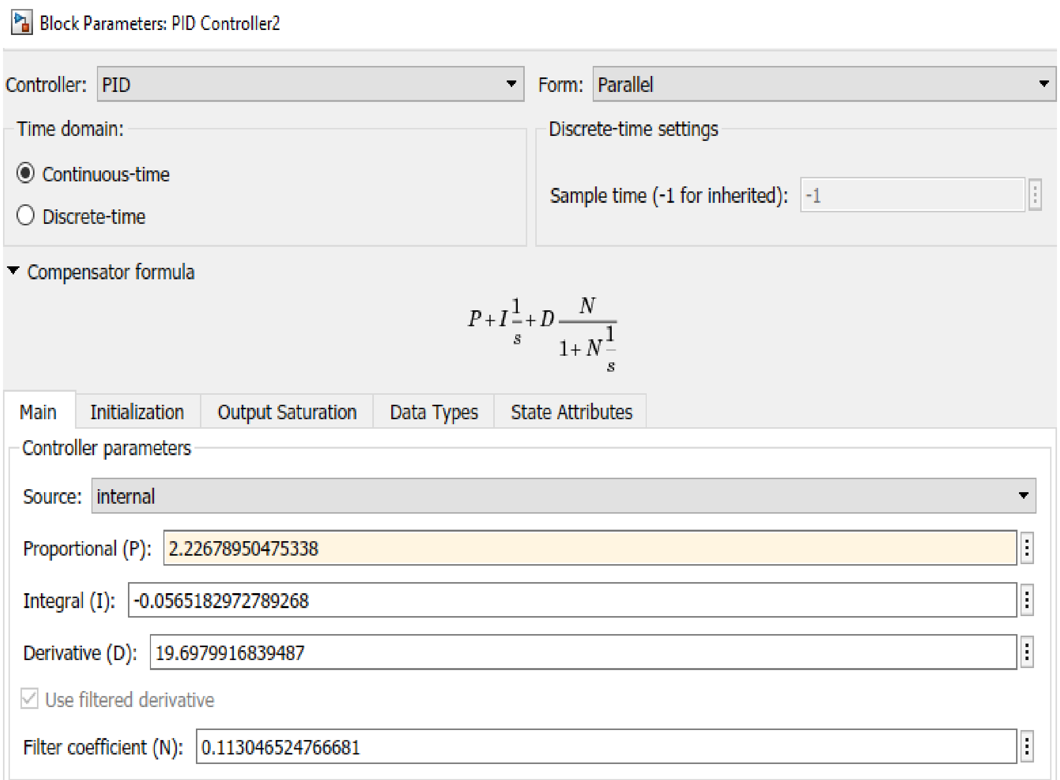

Figure 3-14 PID Tuner Toolbox in MATLAB

3.5.2 Intelligent Fuzzy Logic Controller

The MATLAB Fuzzy logic toolbox is used to create a fuzzy logic controller. The editor includes the FIS editor, membership function editor, and rule base editor when using the Fuzzy Inference System Editor (FIS).

There are two inputs and one output designed through the toolbox for the car pedal pressing design. As seen in Figure 3.16, all membership functions are generalised in the -1 to 1 range. All the input and output crisp data are sent into the fuzzy logic controller and they undergo with gaussian membership. Gaussian membership is chosen to accommodate the fuzzy logic set because it is more flexible and easier to represent and optimize.

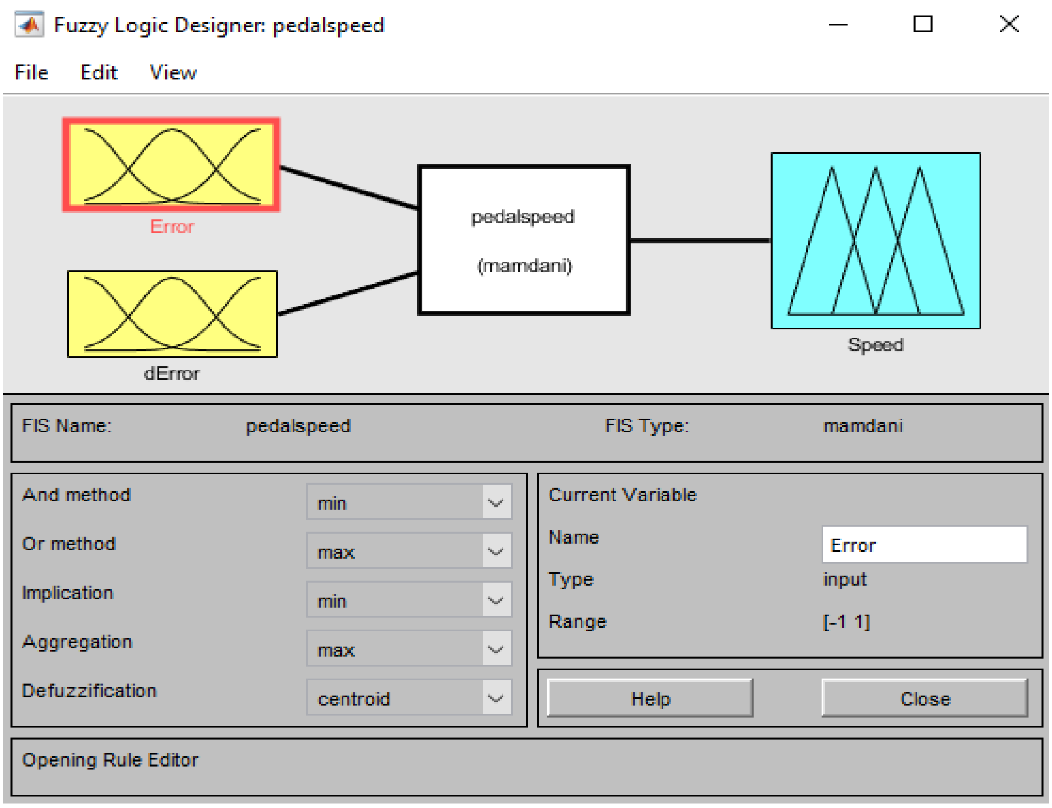

Figure 3-15 FIS Editor

The Membership Function Editor is a tool that displays all the membership functions associated with all the fuzzy inference system's input and output variables and allows users to alter them. It's used to specify the forms of all the membership functions for each variable. In the design of fuzzy logic controllers, the Mamdani type is used so that it creates a scaling factor for optimization to take place.

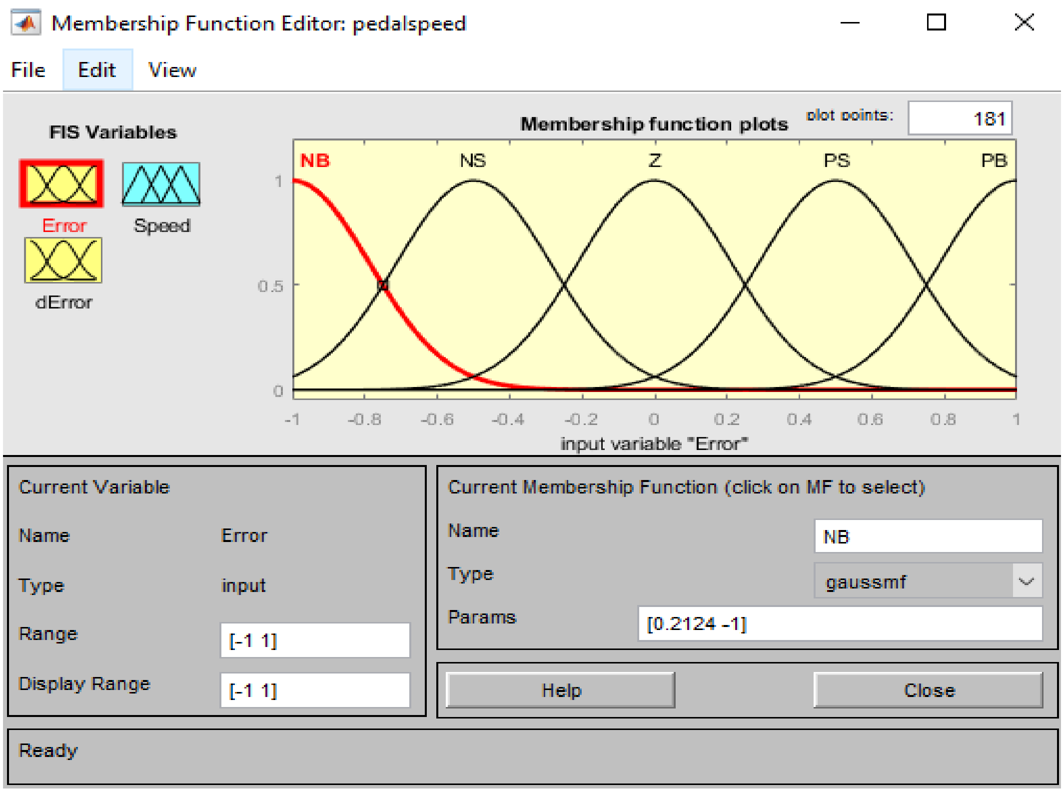

Figure 3-16 Membership Function Editor

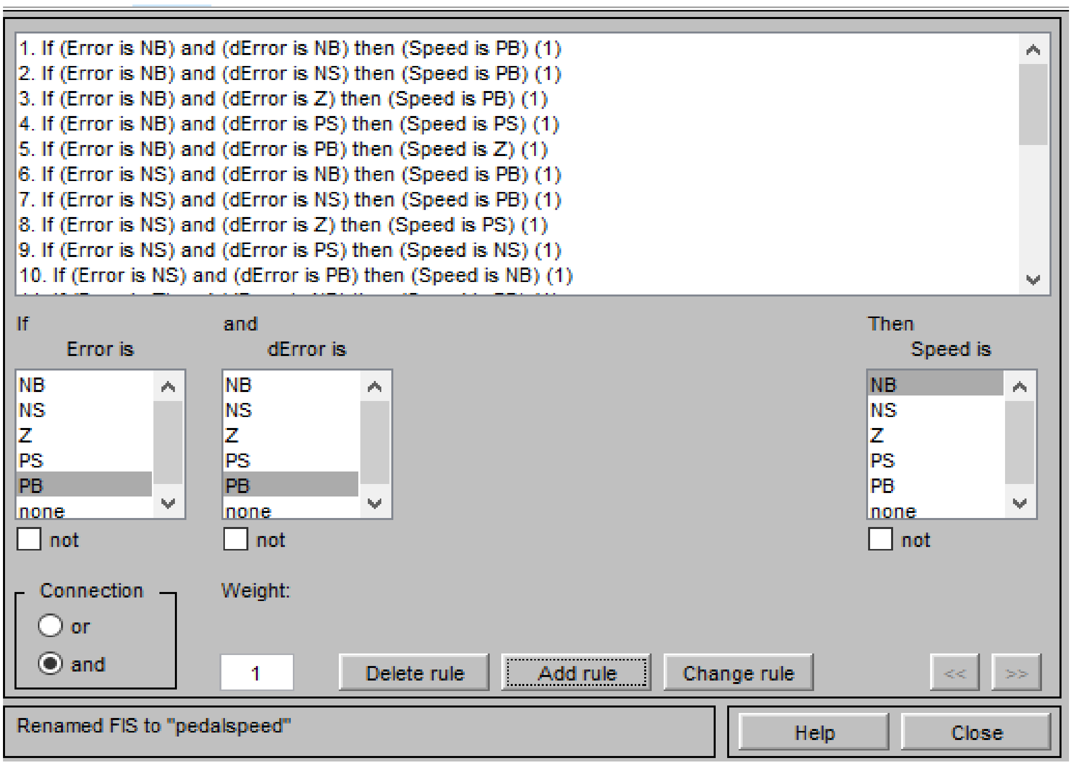

Figure 3-17 Fuzzy Rules Design

Table 3-1 Fuzzy Logic Rule Base

E/dE	**NB**	**NS**	**Z**	**PS**	**PB**
NB	PB	PB	PB	PS	Z
NS	PB	PB	PS	Z	NS
Z	PB	PS	Z	NS	NB
PS	PS	Z	NS	NB	NB
PB	Z	NS	NB	NB	NB

3.6 Simulink Setup

After adjusting the PID controller, the Simulink model of the PID controller is completed, as shown in Figure 3.18. If the PID controller parameters, which are the gains of the proportional and derivative components, are chosen wrong, the regulated process input may become unstable. Its output diverges regardless of whether it oscillates and is only limited by saturation or mechanical breaking. The adjustment of a control loop's control parameters to the optimum values for the desired control response is known as tuning.

When the fuzzy logic controller is finished, MATLAB Simulink is used to put it up. This is done to imitate the automobile pedal controller's interaction with the system. As a result, the car brake system's performance is assessed. Figure 3.19 shows the fuzzy logic Simulink model.

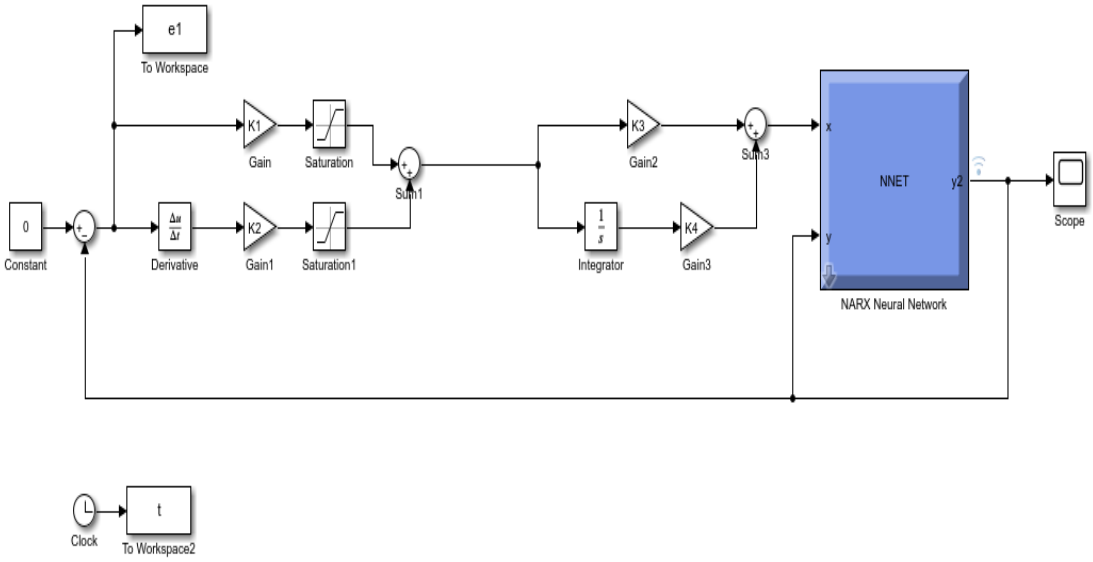

Figure 3-18 PID Simulink Model

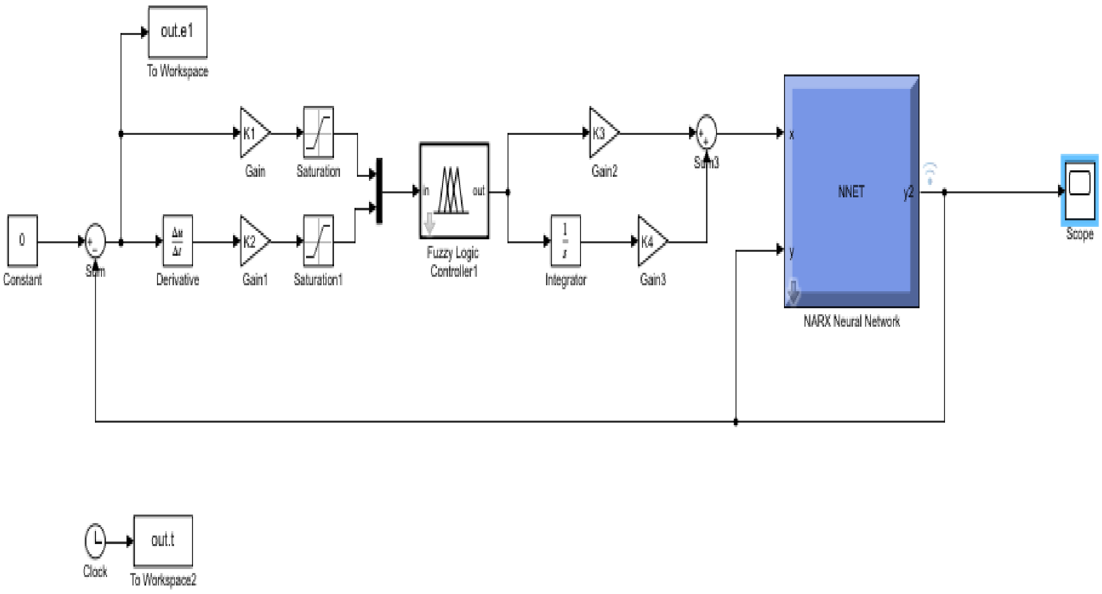

Figure 3-19 Fuzzy Logic Simulink Model

3.7 Controller Optimization

The controller optimization process will be carried out by particle swarm optimization. Previously, this project required designed controller; PID and Fuzzy Logic controller are normalised with membership function, it seems that a scaling factor must be found to scale the membership functions. To obtain the best scaling factor, PSO process considerably improves the efficiency of finding the scale factor due to fact that searching them by trial-and-error.

3.7.1 Particle Swarm Optimization

Swarm intelligence and theory in general, such as bird flocking, fish schooling, and even human social behaviour, inspired this optimization technique[21].The two most used methods are known as gbest model and pbest model in particle swarm optimization. In iteration, pbest and gbest are obtained after the end of first iteration and if the new pbest value is smaller than current gbest value in later iteration, it will replace it. This process is repeated until the iteration end and the final gbest value is the optimal value that wanted.

Table 3.2 shows the PSO parameter that is used for the optimization process, Table 3.3 shows the range of scaling factors for both controllers

Table 3-2 Setting for PSO algorithm

PSO Parameter	Value
Number of particles	100
Number of iterations	30
Learning factor 1	0.12
Learning factor 2	0.2
Minimum weight	0.4
Maximum weight	0.9

Table 3-3 Range of scale factor for PID and Fuzzy Logic controller

Scale Factor	Value
K1	[0.001 1]
K2	[0.001 0.1]
K3	[1 1000]
K4	[1 1000]

3.7.2 Performance Indices

A performance index is a quantitative measure of the performance of a system and is chosen so that emphasis is given to the important system specifications. When the system parameters are altered to the point where the index hits an extreme value, usually a minimal value, the system is considered an optimum control system.[22]. Various performance indexes such as ISE, IAE, ITAE and MSE are accessed with the optimization of PID Fuzzy logic controller for the car pedal pressing system.

We can try to alter the control system settings to minimise some performance index of our choosing to improve the performance of a closed-loop control system. There are some common equations for ISE, IAE, ITAE, and MSE performance indicators[23], and they will be used in the optimization of controller for car pedal pressing system. These equations are implemented as the criterion for controller optimization individually.

$$MSE = \int_0^\infty te^2\,(t)\,dt$$

$$\mathrm{IA}E = \int_0^\infty |e\,(t)|\,dt$$

$$\mathrm{I}SE = \int_0^\infty e^2\,(t)\,dt$$

$$\mathrm{ITA}E = \int_0^\infty t|e\,(t)|\,dt$$

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Input-Output Data Collected in MATLAB Simulink

The Figure 4.3 and 4.4 show the input-output with both controllers. From the result, the speed driving in the road traffic should be low, the reference input will be set between 2 m/s (8km/h) to 7m/s (25km/h) in speed. Since the input of the PID and fuzzy logic controller (pedal position signal)., the controllers see the error is growing when there is traffic jam. The speed of the car will be lagging a few seconds when traffic jam occurs. After traffic jam disappears, it increases the signal to the actuator which in turn increases the engine force and the speed of the car. It means that the car pedal is pressed to increase the speed. The input and output in figures 4.3 and 4.4 are desired speed of the car and actual speed of the car.

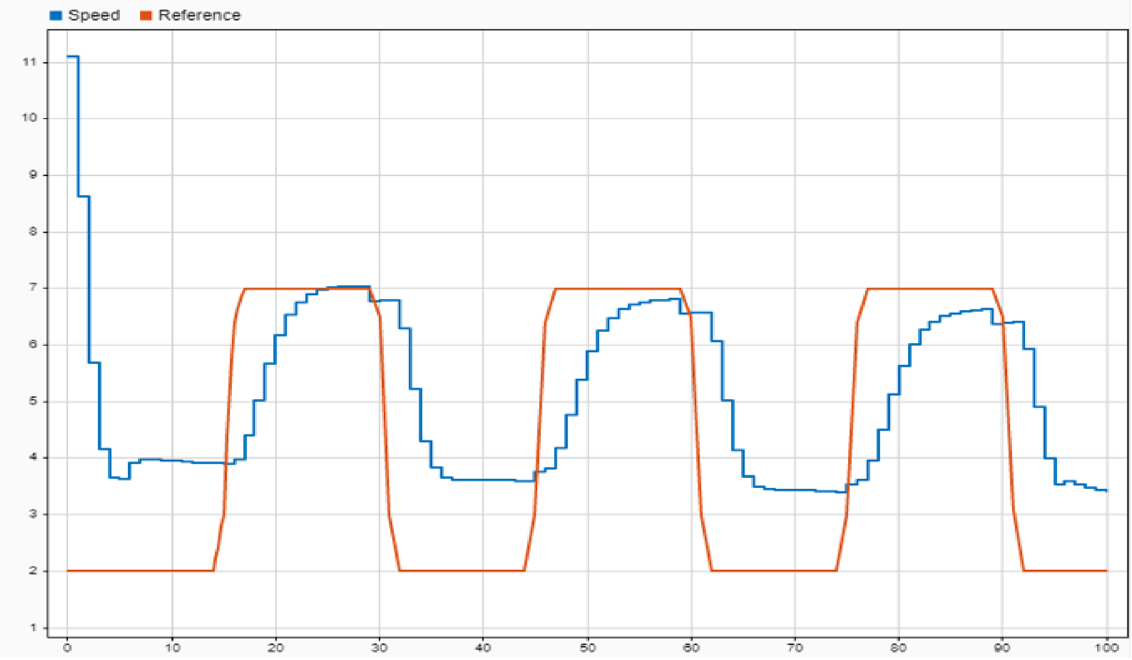

Figure 4-1 Input-Output graph with Conventional PID Controller

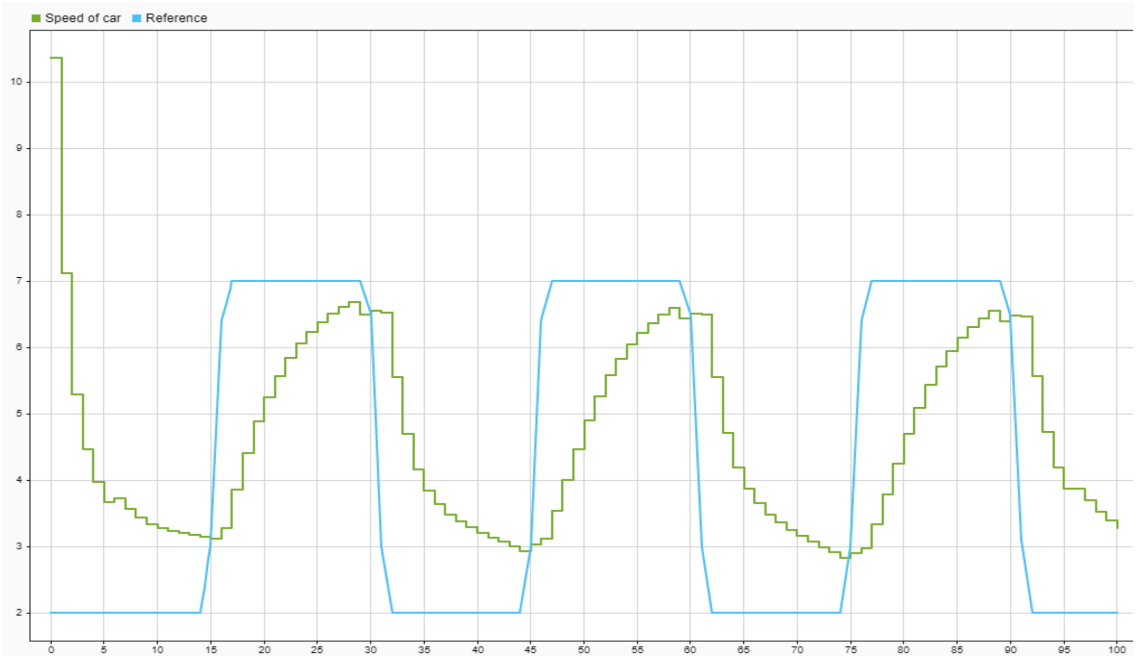

Figure 4-2 Input-Output graph with Fuzzy Logic Controller

4.2 Neural Network for System Modelling

The device identification process by changing the number of delay signals was investigated to predict the one-step-ahead (OSA) reaction. In the meantime, the model structure in the hidden layer is set to 10 neurons. The summary of performance for neural network system identification is shown in Table 4.1 and Table 4.2. While the relationship between number of delay or hidden neuron and mean square error for neural network are shown in Figure 4.5 and Figure 4.6

Table 4-1 Performance of NN with different number of delay signals

Number of Delay	EPOACH	Mean Square Error (MSE)
1	23	0.43721
2	65	0.44628
3	7	0.46296
4	8	0.46678
5	6	0.48319
6	11	0.51659
7	5	0.62752
8	10	0.50075
9	8	0.43983
10	18	0.42550

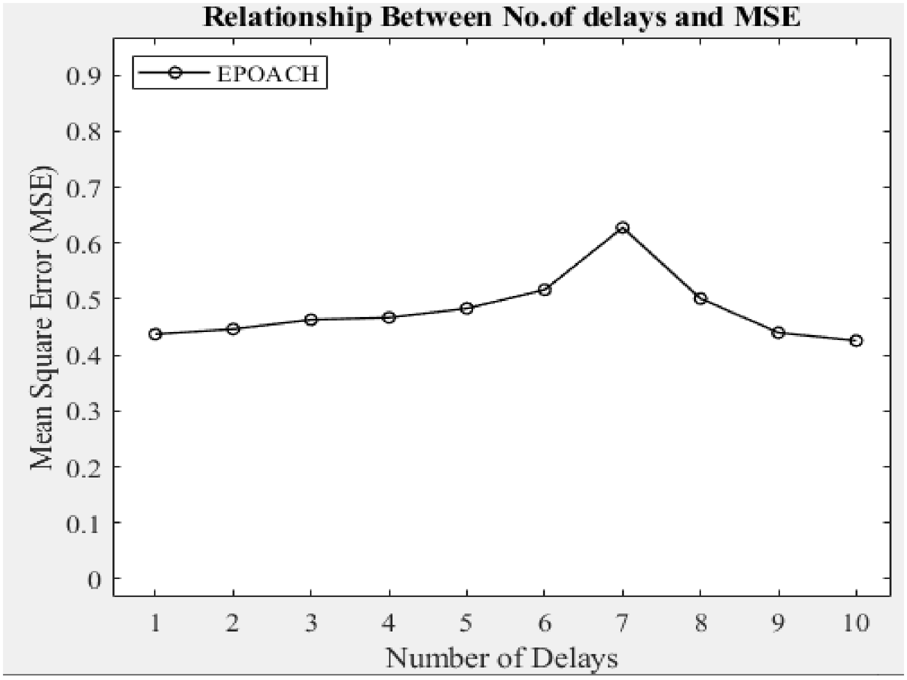

Figure 4-3 Relationship Between Number of Delays and MSE

From Table 4.1, it can be observed that as the number of delays increases until 7 delay then MSE decreases the system input and output signals from 7 delay. For the last 3 delay signal setting, the value of MSE and the amount of convergence iteration were transformed into an increasing trend. Therefore, it can be inferred that using the input and output signal of 2 delays, the excellent one-step-ahead prediction is acquired.

Table 4-2 Performance of NN with different number of hidden neurons

Number of Hidden Neurons	**EPOACH**	**Mean Square Error (MSE)**
2	9	0.61249
4	7	0.42800
6	17	0.55152
8	8	0.47994
10	6	0.41389
12	8	0.48265
14	8	0.40715
16	7	0.51921
18	18	0.50297
20	8	0.41471

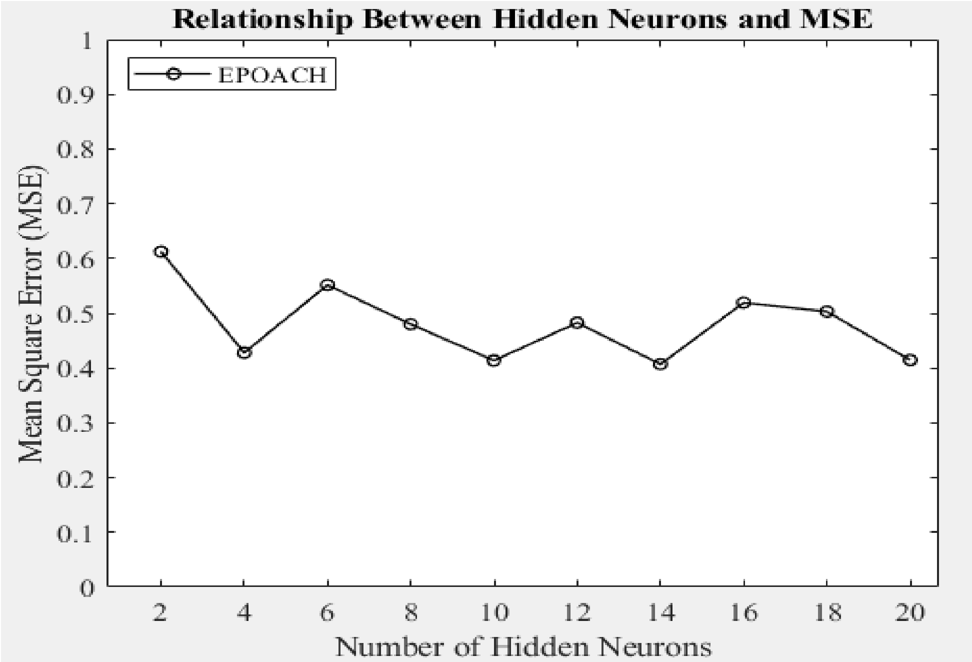

Figure 4-4 Relationship Between Number of Hidden Neurons and MSE

Neural network is checked with different number of neurons in the hidden layer. Table 4.2 displays the results of the system prediction as the number of neurons in the hidden layers advanced.

4.3 Controller After Optimization

In this project, PID and Fuzzy Logic controller design were performed and manipulated with scaling factors by using optimization method. Optimal controller parameters that provide the best control from the PID and Fuzzy Logic controller parameters that make the system stable by using the optimization method are obtained. The performance indices for both controllers are very different as table 4.3 and 4.4 below.

Table 4-3 Scaling factor of PID controller

	K1	**K2**	**K3**	**K4**	**Cost**
MSE	0.5733	-0.1001	459.2665	-16.1983	7.0326
IAE	0.3491	-0.0762	623.8969	-29.9177	9.0568
ISE	0.4836	-0.0481	636.8295	-18.0069	5.6286
ITAE	0.4465	-0.1066	467.2691	-23.9742	14.0564

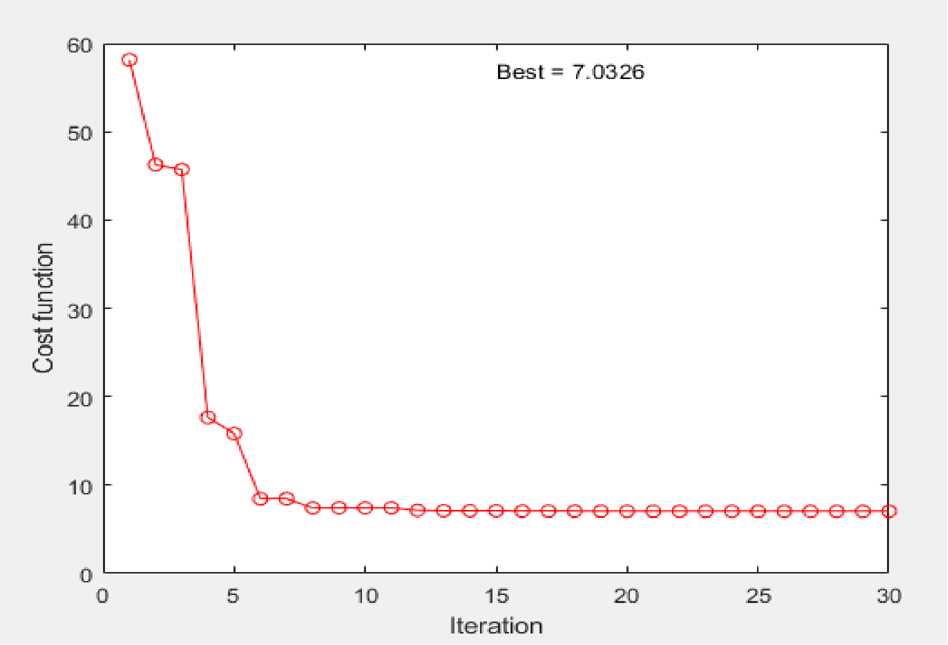

Figure 4-5 MSE performance for PID controller

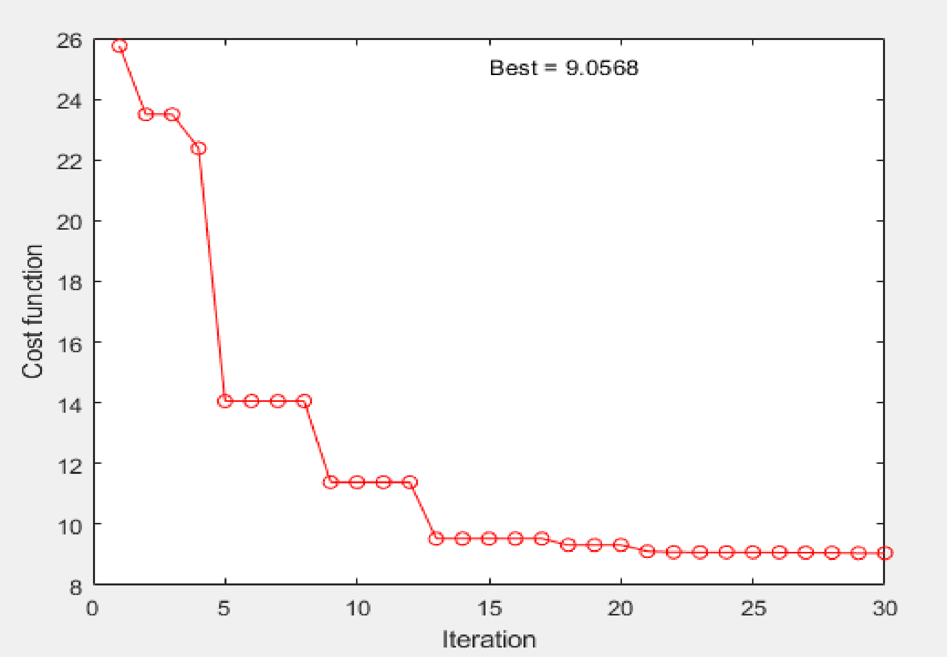

Figure 4-6 IAE performance for PID controller

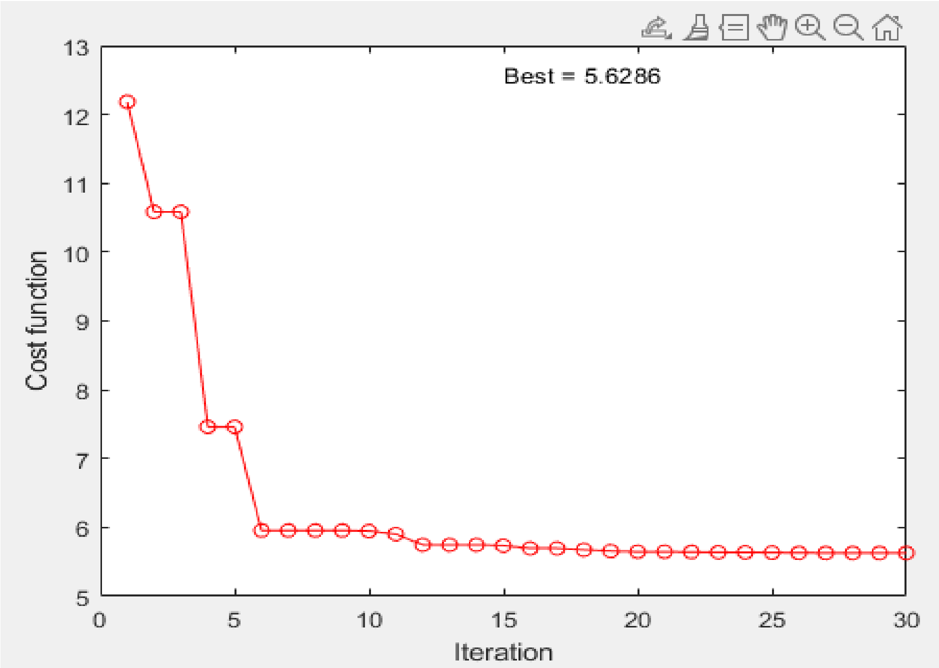

Figure 4-7 ISE performance for PID controller

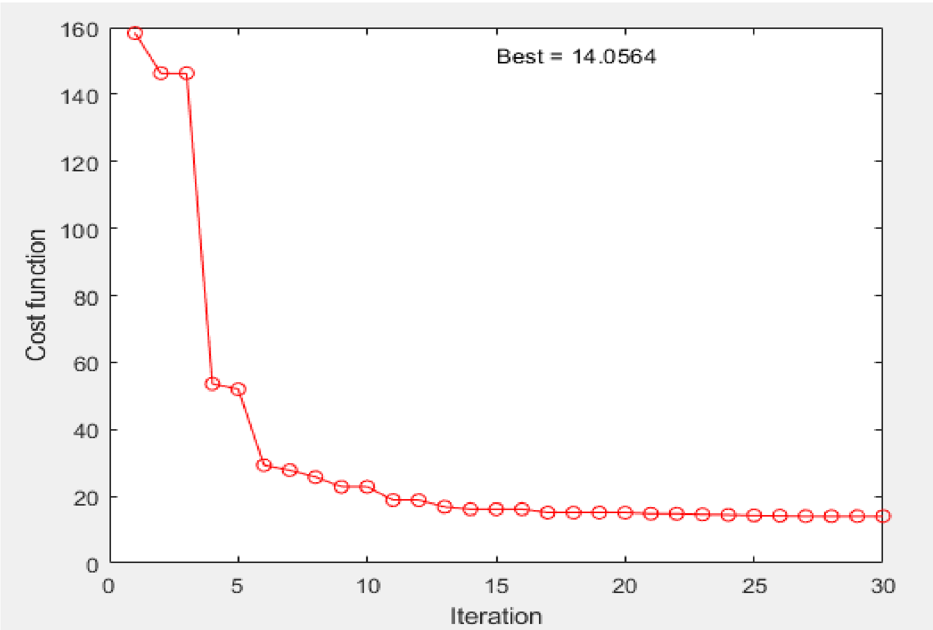

Figure 4-8 ITAE performance for PID controller

Table 4.4 shows the scaling factors with the minimum cost for each performance index for Fuzzy Logic controller. The cost function decreased as the iteration increased because the new gbests value tends to produce smaller error hence the cost function become smaller. Figure 4.11, Figure 4.12, Figure 4.13, and Figure 4.14 show the convergence of PSO and lead to their minimum cost value respectively.

Table 4-4 Scaling factor of Fuzzy Logic controller

	K1	K2	K3	K4	Cost
MSE	-0.0261	-0.0011	786.0065	234.7067	2.7115
IAE	-0.0470	-0.0036	507.4040	129.6444	14.4872
ISE	-0.4014	-0.0071	1.1909	-27.6994	5.2540
ITAE	-0.0334	-0.0024	954.31	183.8143	4.1064

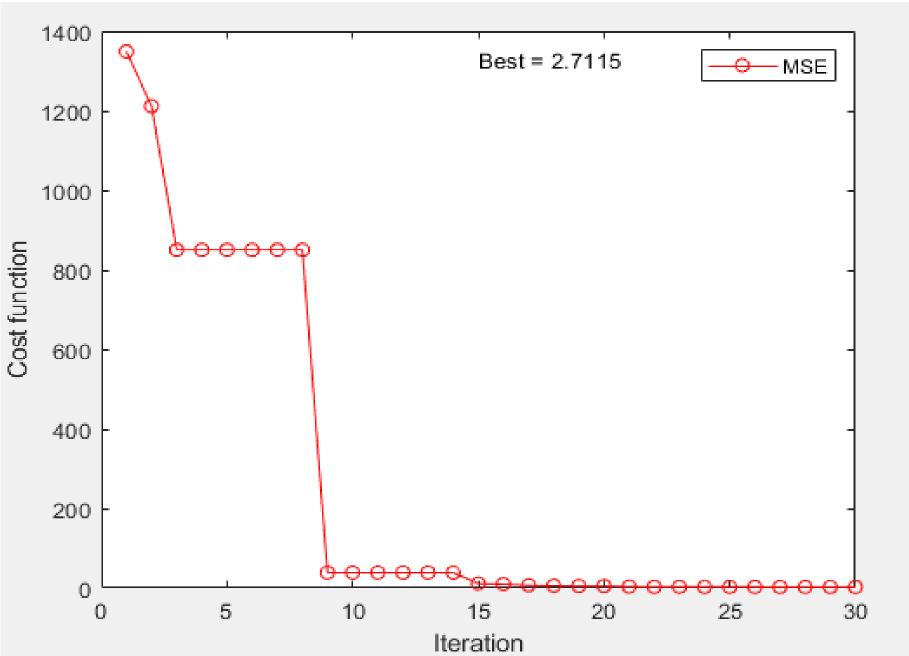

Figure 4-9 MSE performance for Fuzzy Logic controller

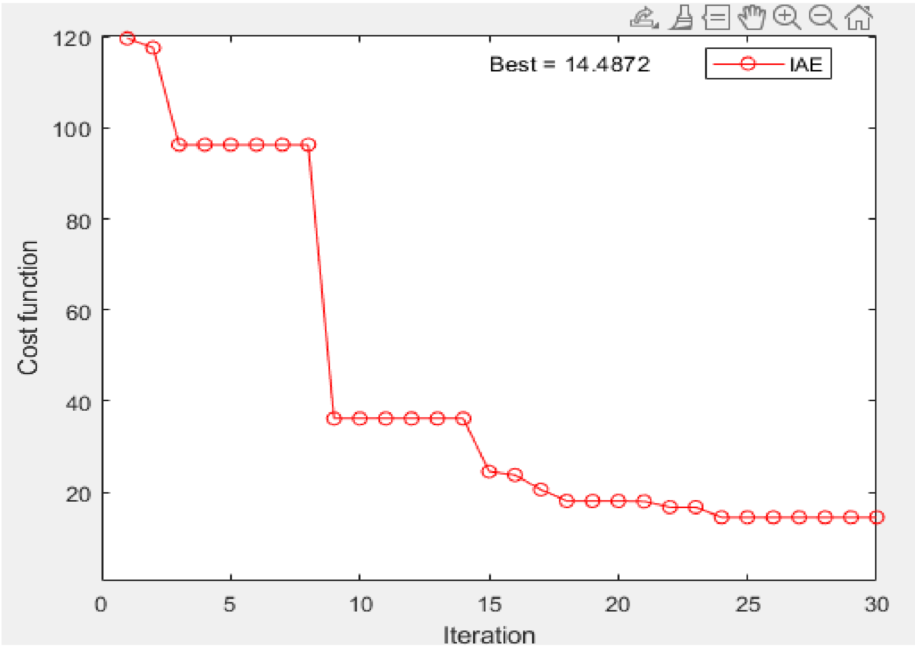

Figure 4-10 IAE performance for Fuzzy Logic controller

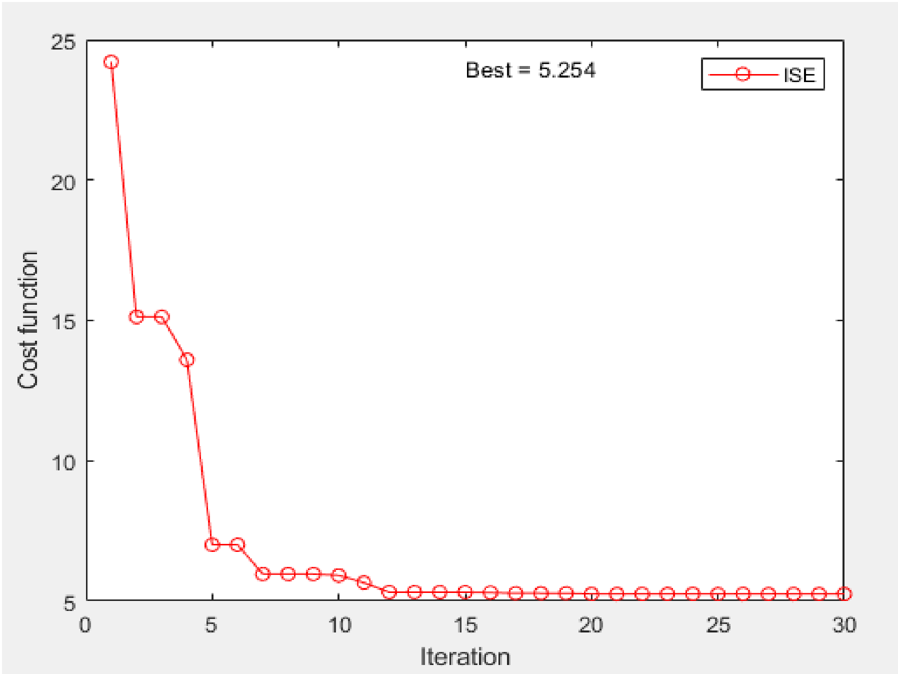

Figure 4-11 ISE performance for Fuzzy Logic controller

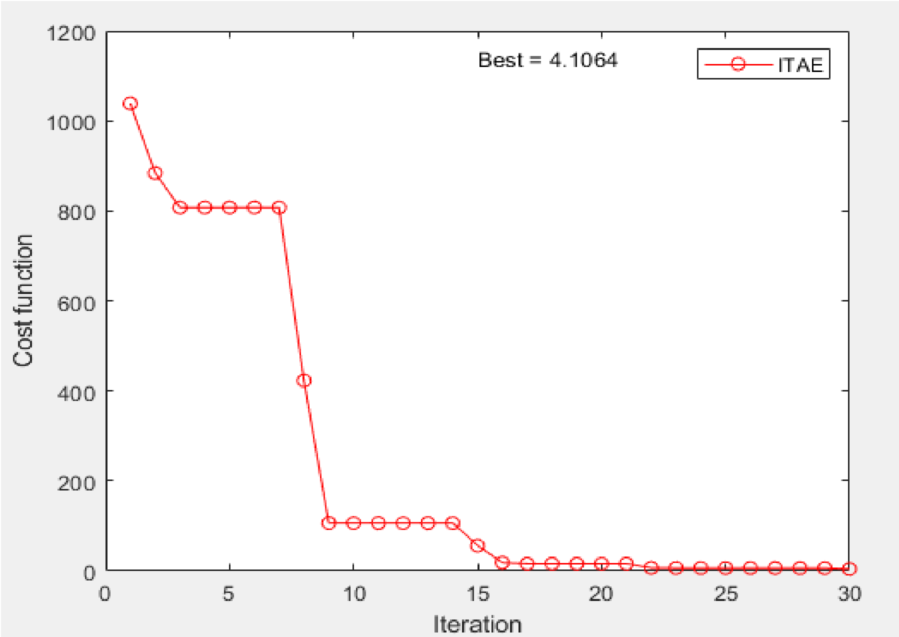

Figure 4-12 ITAE performance for Fuzzy Logic controller

4.4 System Performance with Applying Both Controllers

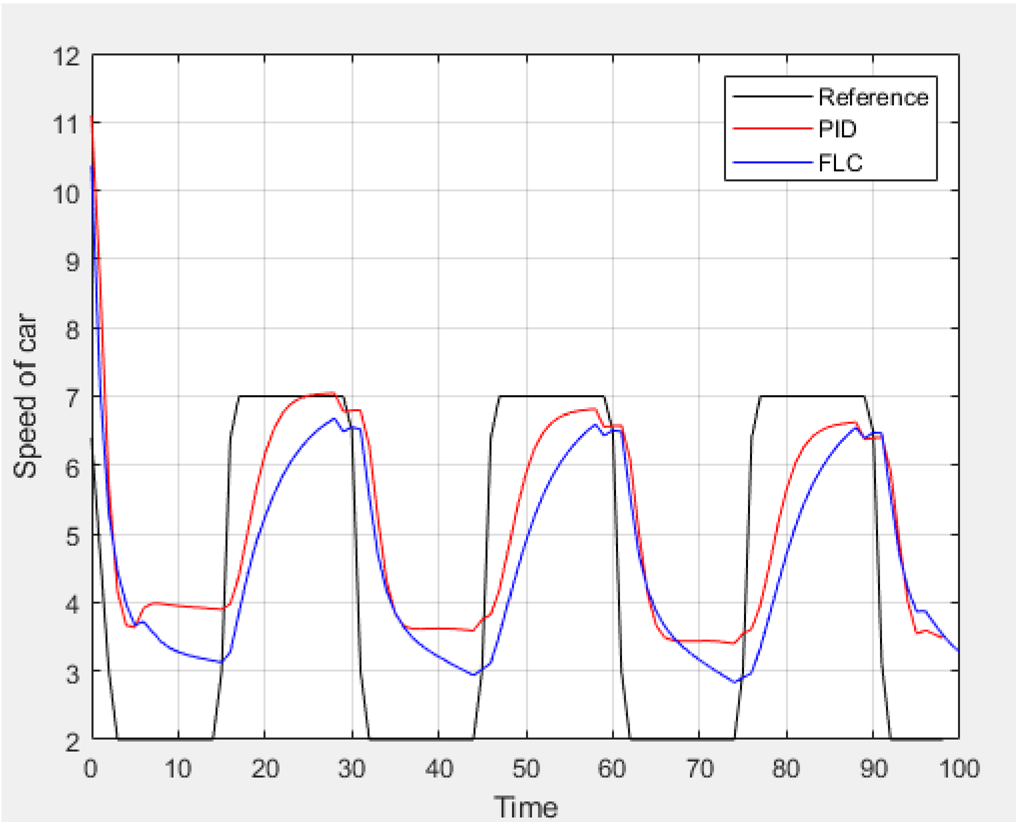

Figure 4-13 System Performance with applying both controllers

Table 4-5 Comparison System Performance with applying both controllers

	PID Controller	**Fuzzy Logic Controller**
Rise Time (s)	0.3556	0.0897
Settling Maximum (s)	8.6300	7.1144
Settling Minimum (s)	3.3998	2.8339
Overshoot (%)	10.9625	3.6438
Steady State Error (%)	1.4821	1.2724

Table 4.5 shows the result performance for PID Controller and Fuzzy Logic Controller. For rise time, Tr, PID Controller achieved in 0.3556s which are also less low than the Fuzzy Logic Controller. But, for overshoot, Fuzzy Logic was much better than PID Controller which Fuzzy Logic Controller had 3.64% compared to PID Controller which had 10.96% overshoot. It means, it has proved that Fuzzy Logic was a better controller than PID Controller based on the performances of overshoot parallel with objective where the objectives are to reduce or eliminate overshoot. Compared to FLC, less overshoot occurred meaning that the system follows the response with less error. Even though the rise time for Fuzzy Logic Controller is quite slower than PID Controller, but it was not worst, because the delay time has been just 0.3s only.

CHAPTER 5

CONCLUSION

5.1 Introduction

In the nutshell, the project accomplished the objectives discussed in chapter 1 successfully. System identification of nonlinear automatic car pedal pressing has been done using neural network. A detuned controller was built for identifying purposes, allowing more of the car pedal pressing system dynamics to be viewed. A feedforward network with varied numbers of hidden layer neurons was tested. The MSE between the system and the neural model is low, and the neural network model can predict car pedal pressing. It was discovered that while utilising a closed loop controller, some specific hidden neuron counts have less of an impact on model correctness than when using a detuned controller.

System identification technique is opted to complete the first step of system control analysis. NARX neural network system identification is useful because input and output data of plant are all it needs. Then, two different controllers (Conventional PID and Intelligent fuzzy logic) are designed to achieve the desired result. Conventional PID and intelligent fuzzy logic controller, both utilized and showed promising results in the stabilization control of car pedal pressing system with system performance.

Finally, the taxing process of tuning controller manually is omitted with the implementation of PSO. Optimization technique is extremely helpful in the progress of determining optimal scaling factors for controllers. Moreover, optimization based on different criteria also impacts the effectiveness of controllers and it opens a lot of options in choosing the suitable controller for automatic car pedal pressing system.

5.2 Future Recommendation

This project can be done especially the settling time for the using both controllers. Both controllers were able to track the reference well though with overshoots and both controllers had risen time values that were less than the required. From the project, MSE obtained for the decided system model is acceptable and it represents the actual system doubtlessly, but simulation is still the result on paper but not on the real world, the goal of project is meant for real-world usage after all.

Next, the optimization method will be done for both controllers in the future. The optimization method including genetic (GA), firefly (FA) and artificial bee colony algorithm (ABC) to optimize the controller error. Moreover, the membership function and the rules can be optimized as well for fuzzy logic controller. There is a countless possibility for the proposed controller to performs even better.

Lastly, various performance index such as ISE, IAE, ITAE and MSE will be accessed with the optimization of PID-Fuzzy logic controller for the control of automatic car pedal pressing system. Error and time are very important factors that must be considered at the same time. A performance index is a single measure of the performance of a system that highlights the response characteristics that are considered important.

5.3 Impact to Society and Economic

The method discussed in this project creates a new point of view to the problem, as the technology and knowledge advanced, intelligent control such as neural network able to replace the conventional mathematical approach in system modelling which save times and reduce mistake. Fuzzy logic controller on the other hand also able to handle the non-linear and uncertainties of system which classic controller such as PID controller incapable of doing before system linearization. Furthermore, the PSO approach reduces the time of tuning controller significantly and increase controller reliability. The intelligent control helps to reduce human workload and provide an unswerving outcome.

Furthermore, as automatic cars are expected to be safer than car with human control, the chance of getting an accident will be reduced. This reduction in car accidents will impact transport-related sectors such as insurance industries and car repair centres, again impacting millions of jobs.

Automatic cars can help lower- and middle-income people get access to mobility by lowering transportation costs: According to some analysts, this mode of transportation will become over 50% cheaper compared to present expenses, making it potentially cheaper than public transportation. Public transport might become redundant, as automatic cars will become cheaper than public transport, safer in terms of less accidents, and could be more comfortable in terms of privacy and hygiene. In fact, in the event of a global pandemic, where public transportation is scarce, autonomous cars could be a viable option for maintaining economic activity while lowering the risk of virus spread.

REFERENCES

[1] S. Manogaran, M. Ali, K. M. Yusof, and R. Suhaili, "Analysis of vehicular traffic flow in the major areas of Kuala Lumpur utilizing open-traffic," *AIP Conf. Proc.*, vol. 1883, no. September 2017, 2017, doi: 10.1063/1.5002031.

[2] V. Car, "Geneva 2005 – Volvo Cars focuses on preventive safety," [Online]. Available: https://www.media.volvocars.com/global/en-gb/media/pressreleases/5032.

[3] A. Info, "The tried (and not so tried) methods cities are considering to reduce traffic," 2019. https://www.ameinfo.com/tech-and-mobility/reduce-traffic-congestion-solutions/.

[4] M. Mamat and N. M. Ghani, "Fuzzy logic controller on automated car braking system," *2009 IEEE Int. Conf. Control Autom. ICCA 2009*, no. 0, pp. 2371–2375, 2009, doi: 10.1109/ICCA.2009.5410301.

[5] R. Muller and G. Nocker, "Intelligent cruise control with fuzzy logic," *IEEE Intell. Veh. Symp. Proc.*, no. 49, pp. 173–178, 1992, doi: 10.1109/IVS.1992.252252.

[6] K. Albelihi and D. Vrajitoru, "An application of neural networks to an autonomous car driver," *Proc. 2015 Int. Conf. Artif. Intell. ICAI 2015 - WORLDCOMP 2015*, pp. 716–722, 2019.

[7] A. M. T. Amr O. Abdel-rhman, "Car braking using fuzzy control," *B.Sc. Proj.*, 2012.

[8] A. Hildebrandt, O. Sawodny, R. Trutschel, and K. Augsburg, "Nonlinear control design for implementation of specific pedal feeling in brake-by-wire car design concepts," *Proc. Am. Control Conf.*, vol. 2, pp. 1463–1468, 2004, doi: 10.23919/acc.2004.1386782.

[9] A. M. Saikin, S. E. Buznikov, N. S. Shabanov, and D. S. Elkin, "The mathematical model of dynamic stabilization system for autonomous car," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 315, no. 1, pp. 0–8, 2018, doi: 10.1088/1757-899X/315/1/012023.

[10] L. K. H. M.Norgaard, O.Ravn,N.K.Poulsen, "Neural Networks for Modelling and Control of Dynamic Systems," *A Pract. Handb.*, 2000.

[11] J. M. Salt Ducaju, C. Tang, M. Tomizuka, and C. Y. Chan, “Application Specific System Identification for Model-Based Control in Self-Driving Cars,” *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 384–390, 2020, doi: 10.1109/IV47402.2020.9304586.

[12] J. D. B. and D. E. Seborg, “Determination of Model Order for NARX Models Directly from Input - Output Data,” *J. Process Control*, vol. 8, p. 5, 2009.

[13] H. D. and M. Beale, “Neural Networks Toolbox User’s Guide: For use with MATLAB,” *MathWorks, Inc, Massachusetts*, 2002.

[14] O. A. Dahunsi, J. O. Pedro, and O. T. Nyandoro, “System identification and neural network based pid control of servo- Hydraulic vehicle suspension System,” *SAIEE Africa Res. J.*, vol. 101, no. 3, pp. 93–105, 2010, doi: 10.23919/saiee.2010.8531554.

[15] W. Q.-G. H. C. C. K.K. Tan, “Advances in PID Control, London, UK,” *Springer-Verlag*, 1999.

[16] and T. M. P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, “Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID,” *Int. J. Adv. Robot. Syst.*, vol. 9, 2012.

[17] W. Farag, “Track Maneuvering using PID Control for Self-driving Cars,” *Recent Adv. Electr. Electron. Eng. (Formerly Recent Patents Electr. Electron. Eng.*, vol. 13, no. 1, pp. 91–100, 2019, doi: 10.2174/2352096512666190118161122.

[18] C. Morgenstern, “Autonomous Cars: Should We Continue On This Path?,” 2012.

[19] S. Hirulkar, M. Damle, V. Rathee, and B. Hardas, “Design of automatic car breaking system using fuzzy logic and PID controller,” *Proc. - Int. Conf. Electron. Syst. Signal Process. Comput. Technol. ICESC 2014*, pp. 413–418, 2014, doi: 10.1109/ICESC.2014.81.

[20] A. Jamali, I. Z. M. Mat Darus, P. M. Samin, and M. O. Tokhi, “Intelligent modeling of double link flexible robotic manipulator using artificial neural network,” *J. Vibroengineering*, vol. 20, no. 2, pp. 1021–1034, 2018, doi: 10.21595/jve.2017.18575.

[21] H. Liu, B. Li, Y. Ji, and T. Sun, “Applied Soft Computing Technologies: The Challenge of Complexity,” *Appl. Soft Comput. Technol. Chall. Complex.*, no. January 2004, 2006, doi: 10.1007/3-540-31662-0.

[22] T. I. Signals, “Chapter 5 The Performance of Feedback Control Systems.”

[23] R. Dorf, M. C. Systems, and P. Prentice-hall, “ME 360 Control Systems Performance Indices for Closed-Loop Control[1] R. Dorf, M. C. Systems, and P. Prentice-hall, ‘ME 360 Control Systems Performance Indices for Closed-Loop Control,’ pp. 5–7, 2005.,” pp. 5–7, 2005.

APPENDIX A
SYSTEM IDENTIFICATION WITH NEURAL NETWORK

```
X = mydata(in_new,false,false);
T = mydata(out_new,false,false);

trainFcn = 'trainlm';
inputDelays = 1:7;
feedbackDelays = 1:7;
hiddenLayerSize = 8;

net =
narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.inputs{2}.processFcns = {'removeconstantrows','mapminmax'};
[x,xi,ai,t] = preparets(net,X,{},T);
net.divideFcn = 'divideblock';
net.divideMode = 'time';

net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net.performFcn = 'mse';

net.plotFcns = {'plotperform','plottrainstate', 'ploterrhist',
'plotregression', 'plotresponse', 'ploterrcorr', 'plotinerrcorr'};
[net,tr] = train(net,x,t,xi,ai);

y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)
trainTargets = gmultiply(t,tr.trainMask);
valTargets = gmultiply(t,tr.valMask);
testTargets = gmultiply(t,tr.testMask);
netc = closeloop(net);

netc.name = [net.name ' - Closed Loop'];
[xc,xic,aic,tc] = preparets(netc,X,{},T);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(net,tc,yc)
numTimesteps = size(x,2);
knownOutputTimesteps = 1:(numTimesteps-5);
predictOutputTimesteps = (numTimesteps-4):numTimesteps;

X1 = X(:,knownOutputTimesteps);
T1 = T(:,knownOutputTimesteps);
[x1,xio,aio] = preparets(net,X1,{},T1);
[y1,xfo,afo] = net(x1,xio,aio);
x2 = X(1,predictOutputTimesteps);
[netc,xic,aic] = closeloop(net,xfo,afo);
[y2,xfc,afc] = netc(x2,xic,aic); 49
```

```
multiStepPerformance = perform(net,T(1,predictOutputTimesteps),y2)

if (false)
genFunction(net,'myNeuralNetworkFunction');
y = myNeuralNetworkFunction(x,xi,ai);
end

if (false)
genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
x1 = cell2mat(x(1,:));
x2 = cell2mat(x(2,:));
xi1 = cell2mat(xi(1,:));
xi2 = cell2mat(xi(2,:));
y = myNeuralNetworkFunction(x1,x2,xi1,xi2);
end

if (false)
gensim(net);
end
```

APPENDIX B
PARTICLE SWARM OPTIMIZATION

```
%Initialization

clear all
close all
clc

rng default

global K1 K2 K3 K4;

global K1_min K1_max K2_min K2_max K3_min K3_max K4_min K4_max;


warning ('off','all');

n = 100;                          %no. of agent
bird_setp = 30;                   %no. of iteration
dim = 4;                          %no.of problem
c1 = 0.12;
c2 = 0.2;
wmax = 0.9;
wmin = 0.4;

GBestFitness = [];
current_position=[];
fitness=0*ones(n,bird_setp);
                             %------------------------------%
                             %    initialize the parameter %
                             %------------------------------%
R1 = rand(dim, n);
R2 = rand(dim, n);
current_fitness =0*ones(n,1);

                      %------------------------------------------------%
                      % Initializing swarm and velocities and position %
                      %------------------------------------------------%
K1_min=0.001;
K1_max=1;
K2_min=0.001;
K2_max=0.01;
K3_min=1;
K3_max=1000;
K4_min=1;
K4_max=1000;

for m=1:n
current_position(1,m)=abs (K1_min+(K1_max - K1_min)*rand(1,1));
current_position(2,m)=abs (K2_min+(K2_max - K2_min)*rand(1,1));
current_position(3,m)=abs (K3_min+(K3_max - K3_min)*rand(1,1));
current_position(4,m)=abs (K4_min+(K4_max - K4_min)*rand(1,1));
end

velocity = .3*randn(dim,n) ;
local_best_position  = current_position ;
```

```
                    %--------------------------------------------%
                    %     Evaluate initial population            %
                    %--------------------------------------------%

for i = 1:n
current_fitness(i) = trackspid(current_position(:,i));
end

local_best_fitness  = current_fitness ;
[global_best_fitness,g] = min(local_best_fitness) ;

GBestFitness(1) = global_best_fitness;

plot((GBestFitness),'ro'); xlabel('iteration'); ylabel ('Cost
function');
text (0.5,0.95,['Best = ', num2str(GBestFitness)],
'Units','normalized');
drawnow;

for i=1:n
    globl_best_position(:,i) = local_best_position(:,g) ;
end
                           %--------------------%
                           %  VELOCITY UPDATE   %
                           %--------------------%

velocity = rand*wmax*velocity + c1*(R1.*(local_best_position-
current_position)) + c2*(R2.*(globl_best_position-current_position));

                           %------------------%
                           %   SWARMUPDATE    %
                           %------------------%

current_position = current_position + velocity ;

                        %-------------------------%
                        %  evaluate a new swarm   %
                        %-------------------------%

%% Main Loop

iter = 0 ;                 % Iterations'counter
while  ( iter < bird_setp )
iter = iter + 1;
inertia=(wmax-((wmax-wmin)/bird_setp)*iter);

for i = 1:n,
current_fitness(i) = trackspid(current_position(:,i)) ;
end

for i = 1 : n
        if current_fitness(i) < local_best_fitness(i)
           local_best_fitness(i)  = current_fitness(i);
           local_best_position(:,i) = current_position(:,i)   ;
        end
 end
```

```
 [current_global_best_fitness,g] = min(local_best_fitness);

if current_global_best_fitness < global_best_fitness
   global_best_fitness = current_global_best_fitness;

    for i=1:n
        globl_best_position(:,i) = local_best_position(:,g);
    end

end

GBestFitness(iter)=global_best_fitness;
plot((GBestFitness),'ro-'); xlabel('Iteration'); ylabel('Cost
function');
text(0.5,0.95,['Best = ' num2str(GBestFitness(iter))],
'Units','normalized');
drawnow;

 velocity = inertia *velocity + c1*(R1.*(local_best_position-
current_position)) + c2*(R2.*(globl_best_position-current_position));
 current_position = current_position + velocity;

 sprintf('The value of interation iter %3.0f ', iter )

 global_best_fitness
 globl_best_position(:,n)

end

global_best_fitness
globl_best_position(:,n)

function F = trackspid(pid)

global K1 K2 K3 K4;

global K1_min K1_max K2_min K2_max K3_min K3_max K4_min K4_max;

       K1 = pid(1);
       K2 = pid(2);
       K3 = pid(3);
       K4 = pid(4);
                simopt =
simset('solver','ode4','SrcWorkspace','Current','DstWorkspace','Curren
t');
       [tout,xout,yout] = sim('PID_fuzzy_test',[0 100],simopt);
       [n,~]=size(e1);

       cost_value=0;
       for i=1:n
       %cost_value=cost_value+(e1(i))^2 ;  % ISE
       cost_value=cost_value+abs(e1(i));  % IAE
       % cost_value=cost_value+t(i)*abs(e1(i));  % ITAE
       % cost_value=cost_value+t(i)*(e1(i))^2;  % MSE
        end

       F = cost_value;
    end
```