VEHICLE SPEED CONTROL USING NEURAL NETWORK PREDICTIVE CONTROLLER

by

Muhammad Baizuri Bin Ramli

Dissertation submitted in partial fulfillment of

the requirement for the

Bachelor of Engineering (Hons)

(Electrical and Electronic Engineering)

DECEMBER 2010

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31759 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Vehicle Speed Control Using Neural Network Predictive Controller

by

Muhammad Baizuri Bin Ramli

A project dissertation submitted to the
Electrical and Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,

…………………………………………….
(Pn. Noor Hazrin Hany Bt Mohamad Hanif)
`

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
December 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and that the original work combined herein have not been undertaken or done by unspecified sources or persons.

……………………………………

Muhammad Baizuri Bin Ramli

ABSTRACT

Nowadays, most vehicles are integrated with autonomous control system such as Stability Control System (SCS) and Traction Control System (TCS) which aim to improve the car safety by compensating the error made by the drivers. This thesis focuses on the enhancement of the previous work on path following ability of a linear car model. The path following is the ability of the vehicle to follow the vehicle's path and keeping the vehicle in its lane as accurately as possible. When the car maneuvers itself to follow the prescribed path, the car tends to off track when it undergoes cornering. This is due to the constant speed which the car is traveling. The vehicle is unable to adjust its speed to reduce its lateral speed when it undergoes cornering. As a result, the car will take cornering at high speed and consequently introduces greater path following error to the vehicle. The speed of the vehicle needs to be varied by introducing the speed controller. The speed controller of the vehicle functions to vary the speed of the vehicle to achieve certain speed trajectories when it approaching cornering. This can be well understood by the natural behavior of the driver which tend to reduce the speed of the vehicle when approach cornering. This project utilizes the potential of the neural network predictive controller as the information processing element to control the forward speed of the vehicle when the vehicle approaching cornering.

ACKNOWLEDGMENT

First and foremost, I would like to praise Allah the Almighty for with His blessings and guidance I am able to successfully complete my final year project. My appreciation goes to my Supervisor, Pn Noor Hazrin Hany Binti Mohamad Hanif for the advice, plans, and also for the knowledge and experiences shared during my attachment under her.

My gratitude is extended to all lecturers for their respective professionalism and all the knowledge shared. Also, a million thanks to all staffs and technicians at lab for the continuous help, support and guidance from the beginning of the final year project until the end.

Finally, my acknowledgement is considered incomplete without thanking all my fellow colleagues, and family who have been giving great support and encouragement for me to complete the project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of the Project

The increment of road accidents over past few years have drew automakers' attention to produce cars with more safety features. Road accidents typically occur due to the careless of the drivers and their inability to efficiently control the car. Due to that reasons, we have seen major improvement in the vehicle safety systems starting from the development of three-point seatbelts to airbag system to Anti-Lock Braking Systems (ABS) which have proved to save many lives on the road.

The advance in microchip technology has accelerated the development of active vehicle safety system. The aim for this system is to improve the driving safety by introducing the automated control systems such as Traction Control System (TCS), Acceleration Slip Regulation (ASR) traction control, Electronic Stability Control (ESP), Autonomous Cruise Control System and many more. The introduction if these systems have made driving more relaxing, less stressful and the driver can concentrate on other important factors.

The purpose of these systems is to enhance the stability of the vehicle especially when facing difficult situation such as loss of control due to slip road or when there is sudden change of the car's direction due to obstacles avoidance. The systems will help drivers to maneuver the cars to the correct position by applying autonomous control over the engine speed, wheels rotation and the brakes.

Autonomous Cruise Control System helps drivers to automatically maneuver the cars on its prescribed path based on the information it received from the sensor unit. However, being autonomous, the degree of safety and the reliability of this technology on the roadways are inevitably debated. Therefore, this thesis employing neural network predictive controller as the speed controller for the vehicle would demonstrate the ability of the autonomous cruise control to provide safer journey to the drivers as the cars travelling safely, accurately and eventually give a comfortable ride to the drivers autonomously.

1.2 Problem Statement

With the number of vehicle are dramatically increased over past few years, greater attention need to be given on the safety aspects and comfortable driving experiences. This phenomenon also is supported by the lower cost for producing high reliability electronics equipment for various functions in the vehicle.

1.2.1 Improve Safety

A vehicle travel at high speed is more likely to loss steerability and controllability especially when the vehicle is approaching sharp turn or corner. The phenomenon arises due to the vehicle wheel slip during vehicle acceleration maneuver. At high speed, the vehicle requires no margin of errors as any of it would introduce high risk to the vehicle's passengers and road users. The condition become more critical when cruise control system is engaged as the vehicle would travel at preset speed which normally at speed higher than 80 km/h.

1.2.2 Driving Comfort

Since the cruise control system aims to relieve the drivers from constantly adjusting the vehicle's speed, the system should provide automatic speed adjustment to reduce the speed of the vehicle when needed and resume to the preset speed when the road is clear from any obstacles or curvature. Even though the Adaptive Cruise Control (ACC) has made possible for the vehicle to automatically adjust its speed when it detects any obstacles in its view, the system do not offer speed adjustment when the vehicle approaching cornering or curvature. Therefore, it is a much interest in this project to develop such system which primary interest is paid upon longitudinal control of the vehicle. Furthermore, cornering at high speed would results in uncomfortable situation felt by the driver and passengers which eventually could reduce less stressful driving experiences.

This situation is well simulated in the previous work done by [1] which the linear car model employed tends to get off track when the vehicle undergone cornering and the error became more significant at high speed. This is due to the constant speed that was set to the linear car model when it is following the predetermined path. Therefore, when the linear car model approaching sharp curves or cornering, it does not have the ability to reduce the speed to enable the car to grip to its path properly.

1.3 Objective and Scope of Study

This project is an extension of current cruise control system. However, the system does not only take care of the prescribed speed set by the driver but at the same time, has the ability to reduce its preset speed whenever necessary. The system would then resume its previous prescribed speed after the vehicle left the triggering situation such as after the vehicle leaving the road curvature or cornering. It is an interest in this project to maintain both lateral and longitudinal control of the vehicle in order to have more control over vehicle behavior during driving activities.

Lateral control of the vehicle has been well presented in the previous works done by [2,3] .This project can be assumed as the extension of the previous works with great attention given on controlling the longitudinal displacement of the vehicle. In order to achieve this, a suitable speed controller needs to be designed so that the car can achieve its desired speed trajectories.

1.3.1 Controller Selection

With the advance of control theory and technology, there are numbers of choices that could be considered whenever it comes to choose the right controller. For example, Proportional-Integral-Derivatives (PID) controller is well establish and suitable for many kind of application. Since the controller is already recognized, it is easier to convince interested parties whenever the responsible parties came to commercialize the application. However, attention also needs to be given on the limitation of the controller itself as it would really reflect the future performance of the controller.

One of the areas that always come into consideration of the control engineers is the non-linearity nature of the system. Therefore, it is important to determine whether the chosen controller has the capability to operate on non-linear system. As for vehicle dynamics, many non-linear factors present in the system such as the road conditions, road friction, drag force and many more. Since PID controller cannot well operate under non-linear system, another suitable controller which could cater the limitation of the PID should be considered.

Neural network which the nature of operation is different from conventional controller has gained attention to control engineers. The nature of the controller which could work on non-linear system makes it suitable to replace conventional controller. However, since computation time takes longer than conventional controller, many approaches were tested to counter the drawbacks. One of the techniques employed is by integrating predictive control algorithm into existing neural network architecture. The resulting system should produce more satisfactory performance. Hence, the term neural network predictive controller came into consideration which will be extensively used throughout this project.

1.3.2 Longitudinal Vehicle Control

The scope of this project will be revolved around controlling longitudinal displacement of the vehicle using neural network predictive controller. The project will always started with simpler application first since it would be much easier to measure the performance of the outcome. For example, instead of starting with complex dynamics of the car model, simpler version of the vehicle dynamics representation would be employed without significantly degrades the key performance of the tested system. More explanation on the vehicle model will be described on the later chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Relevant Research Topics

Vehicle speed control is becoming crucial area of research in autonomous vehicle control. With the success implementation of intelligent vehicle transmission such as automatic transmission and continuous variable transmission (CVT), focus has been shifted into developing intelligent controller which capable of controlling the mechanical part of the transmission system efficiently. Proper management of transmission system could result in efficient fuel usage and better driving experience.

In [1], an automotive speed controller based on neural network has been developed which work on low and high speed using throttle and brake control input. The network itself consists of simple multilayer feedforward perceptron network which was trained with dynamic vehicle model. The work suggested that the accuracy of the vehicle model and the choice of cost function are important in determining the practical use of the controller. However, compromise had to be made between model accuracy and computation time as more complex vehicle model could lead to increase computation time.

Tight control on the vehicle speed can also improve lateral stability of the vehicle especially during the vehicle cornering. It has been shown in [2] that significant lateral error occurs during the cornering of the vehicle. This is due to the deployment of constant speed for the vehicle which in practical should be reduced when the vehicle approach cornering.

Vehicle speed control is used in stability control which aims in reducing the risk of vehicle collision due to the obstacle avoidance under emergency. In [6], effort has been put on developing control strategy for high speed obstacle avoidance under emergency. The developed control strategy was aimed in preventing vehicles from spinning and drifting out on high speed obstacle avoidance under emergency. The strategy which is known as Vehicle Dynamics Control (VDC) was achieved by applying counter braking at individual wheels as needed until steering control and vehicle stability is regained. The system integrates Anti-Lock Braking System (ABS) and traction control to stabilize the vehicle when it changes direction from that intended by the drivers.

Study has shown that there are increasing tendency of serious traffic accidents while cornering. A major cause of such accident is due to the excessive vehicle speed due to driver's misjudgment or incorrect recognition of the road condition. Therefore, study that was conducted in [7], try to investigate the base technology for producing self-reliant cornering vehicle speed system without relying on outside facilities such as dedicated traffic monitoring system on roadsides. The paper suggests that information obtained from GPS system which already installed in most vehicles could be used to predict the oncoming road curvature. This information together with vehicle speed data could be used by the judgment unit to decide the appropriate cornering speed for the vehicle. Finally, algorithm for better judgment method was developed in order produce more reliable and accurate control system.

Another method for controlling longitudinal displacement of the vehicle was done in [8]. The work stressed on intelligent communication between infrastructure and vehicle (I2C). This system relies on the implementation of RFID that enable vehicles to communicate and receive signal from infrastructure on the roadside such as speed limit zone, road signs and maintenance work on the road. The system will adjust the speed of the vehicle depending on the road circumstances.

In order to achieve more satisfactory vehicle dynamic control, both lateral and longitudinal control should be introduced on the vehicle. The work done in [9] attempt to combine both longitudinal and lateral control of a vehicle. The paper proposed and evaluated an integrated longitudinal and lateral control for vehicle low speed automation. The focus of the system is on congested sub-urban area. A simplified coupled longitudinal/lateral model was employed and the solution for vehicle following problem was presented using first and second order sliding mode controls.

A simple vehicle model and multi-body vehicle model have been deployed in [3] to show the application of optimal preview control to speed tracking of road vehicles. The work shows that the appropriate feedback gains need to be obtained in order for the vehicle to accurately track the speed demand. Since the optimal gain is employed, gain scheduling may not be necessary as the effect is small on the vehicle performance. In the nutshell, the work can be seen as a preliminary work before the integration of the lateral and longitudinal control of a vehicle which is important in order to simulate more realistic vehicle dynamics problem.

2.2 Linear Car Model

Two vehicle models have been considered to represent lateral and longitudinal translation dynamics of the vehicle. For the lateral control, the car model from [4] has been considered and the parameters of the model are shown in Table 1. The equation of motion of the linear car model is as follows:

$$\dot{x} = Ax + B\delta_{sw} , \tag{1}$$

with the state vector, $x = [\ x_1\ x_2\ x_3\ x_4\ ]^T$

where, x_1 = global lateral position, y

x_2 = global lateral speed, $\dot{y}$

x_3 = global attitude angle, ψ

x_4 = global attitude rate, $\dot{\psi}$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -(Cf + Cr)/Mu & (Cf + Cr)/M & (bCr - aCf)/Mu \\ 0 & 0 & 0 & 1 \\ 0 & (bCr - aCf)/Izu & (aCf - bCr)/Iz & -(a * aCf + b * bCr)/Izu \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ Cf/MG \\ 0 \\ aCf/IzG \end{bmatrix}$$

Table 1: Parameters for Lateral Linear Car Parameters

Parameters	Values
Body Mass (M)	1200 Kg
Yaw Inertia (I_z)	1500 Kgm^2
Distance from center of gravity to front axle (a)	0.92
Distance from center of gravity to rear axle (b)	1.38
Cornering stiffness of front axle tyres (C_f)	$1.2x10^5$ $Nrad^{-1}$
Cornering stiffness of rear axle tyres (C_r)	$8x10^4$ $Nrad^{-1}$
Fixed Steering Ratio (Hand wheel/ road wheel) (G)	17

The model which based on standard yaw/sideslip model and it is assumed that the car is a rigid body, moves on a flat path with three degree of freedom which is forward, lateral (side) and yaw (side to side) motions. Four types of forces which react on the car model are:

- Front axle longitudinal force
- Front axle lateral force

- Rear axle longitudinal force
- Rear axle lateral force

The suspension, aerodynamic force, tire aligning moment and lateral weight shift are neglected in this vehicle model in order to simplify the simulation as it is not much affected the normal car at normal speed. On the other hand, the dynamic model for longitudinal translation of the vehicle was obtained from [3] and the state space equation is given by the following equation. The diagram of the model is shown in Appendix B, figure 25.

The equation of motion of the linear car model is as follows:

$$\dot{x} = Ax + Bt_c\ , \tag{2}$$

with the state vector, $x = [\ x_1\ x_2\ x_3\ x_4\ ]^T$

where, x_1 = longitudinal position, x

x_2 = longitudinal speed, u

x_3 = front wheel spin angle, w_f

x_4 = rear wheel spin angle, w_r

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -(Cf+Cr)/Mu & (CfxRf)/(Mu) & (CrxRr)/Mu \\ 0 & (CfxRf)/(Ifu) & (CfxRf\text{^}2)/(Ifu) & 0 \\ 0 & (CrxRr)/(Iru) & 0 & -(CrxRr\text{^}2)/(Iru) \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ Tf/If \\ 0 \end{bmatrix}$$

Table 2 shows the parameters for the longitudinal car model.

Table 2: Parameters for Longitudinal Car Model

Parameters	Values
Body Mass (M)	1200 Kg
Front Wheel Inertia (I_f)	2 kgm^2
Rear Wheel Inertia (I_r)	2 kgm^2
(m_f)	100 kg
(m_r)	100 kg
Front Longitudinal Slip Stiffness (C_{fx})	80,000N
Rear Longitudinal Slip Stiffness (C_{rx})	80,000N
Front and Rear Wheel Radius (R_f & R_r)	0.3m

The model has a main body with longitudinal translational freedom only and front and rear axles with spinning wheel. The tyres interact with the ground with a simple force slip relationship independent of tyres loading. The car model is assumed to be driven by front axle with a torque that is reacted to the main body. The car model also is assumed to have three degrees of freedom with tyres longitudinal forces proportional to longitudinal slip.

2.3 Speed Preview Model

Based on the work done in [2], five road models were used in the study. The path can be described by lateral deviation, y_r from a fixed straight line at sampling time kT . If the forward speed is constant, the lateral deviation at time kuT meters ahead of the car can be represented as:

$$y_{red}(k) = [y_0 \; y_1 \; \dots \; y_m]^T$$

The uT is the x spacing in which u is the speed of the vehicle. The x spacing is the interval between the consequence path at instance k.

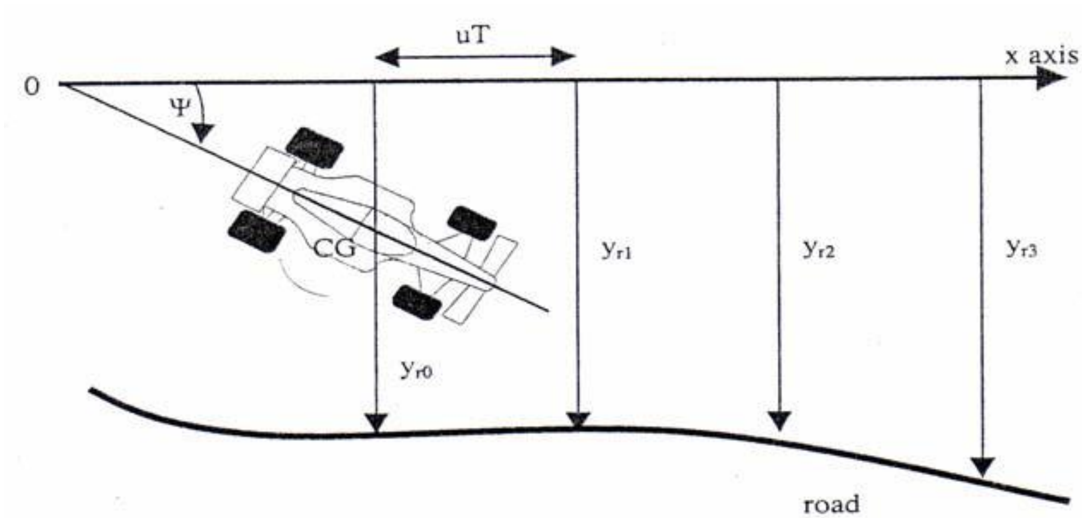

Figure 1: Road Preview Model [2]

The state space equation of the road preview model is

$$y_{ref}(k+1) = D.y_{ref}(k) + E.y_{ri} \tag{3}$$

and the vector of D and E are:

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

However, referring to the work done in [3], the road preview model can be used as speed preview model for vehicle speed tracking. Therefore, in this project, the speed preview model can be described by the same formula as road preview model.

By replacing the notation in equation 3, the new formula for speed preview model is,

$$u_{ref}(k+1) = D.\, u_{ref}(k) + E.\, u_{ri} \tag{4}$$

2.4 Optimal Controller

Referring to the work done in [3], optimal preview controller was employed for the purpose of connecting both linear car model and road preview model. Optimal preview controller can be assumed as the human representation to represent the vision of the driver so that the vehicle can follow the path as accurately. In short, optimal controller is used to aid the car to be driven along the path. As shown in [3], optimal controller is used to combine speed preview model and linear car model which is quite similar to solve the problem involving road preview model and linear car model. The state space equation of the car and the speed (having no connection between both) is:

$$\begin{bmatrix} x(k+1) \\ yr(k+1) \end{bmatrix} = \begin{bmatrix} Ad & 0 \\ 0 & D \end{bmatrix} . \begin{bmatrix} x(k) \\ yr(k) \end{bmatrix} + \begin{bmatrix} 0 \\ E \end{bmatrix} . y_{ri} + \begin{bmatrix} Bd \\ 0 \end{bmatrix} . \delta_{sw}$$

which takes the standard discrete time form,

$$z(k+1) = A_z(k) + B\tau(k) + E\eta_{ri}(k) \tag{5}$$

$$y(k) = Cz(k) \tag{6}$$

As the work done in [3] suggests, optimal preview controller also can be used to solve quite similar problem involving longitudinal displacement of the car model. Instead of the reference paths, the problem can now be used to combine vehicle-speed target system states.

2.5 Neural Network Controller

The purpose of the neural network controller is to represent the driver information processing structure and to output the desired response to the given input. The previous work done by [2, 5] used neural network controller to optimize the path following of the vehicle. However, in this study, the same neural network is used to mimic the driver control of the speed of the vehicle. This neural network utilized the speed profile information from the speed preview model.

The controller is set to be a linear, single processing neuron. The input to the first controller is the augmented state $z = [x \;\; u_r]^T$ where x is the equation of motion of the car model and u_r is the speed preview error. The output of the second controller is the torque, t_c applied on the front wheel of the linear car model since the car is driven by front wheel.

2.6 Neural Network Predictive Controller

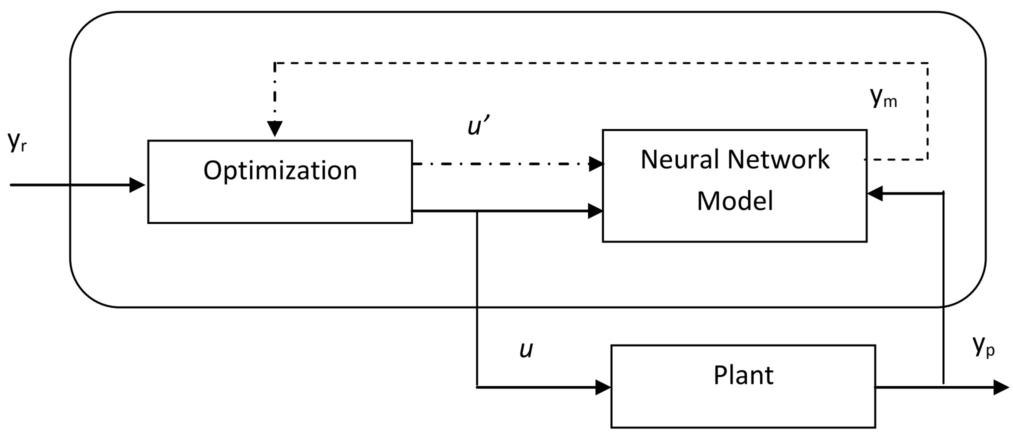

Figure 2: Neural Network Predictive Controller [10]

Neural network predictive controller is a controller that has the ability to predict future plant performance. The controller will calculate control signal values that will optimize plant performance over specified time horizon. The controller will utilize neural network plant model in predicting plant future performance.

2.6.1 System Identification

The initial stage in model predictive control is to develop neural network and trains it to represent the forward dynamics of the plant. Depending on the training algorithm used, the different between plant output and the neural network output will be used as the training data to the neural network.

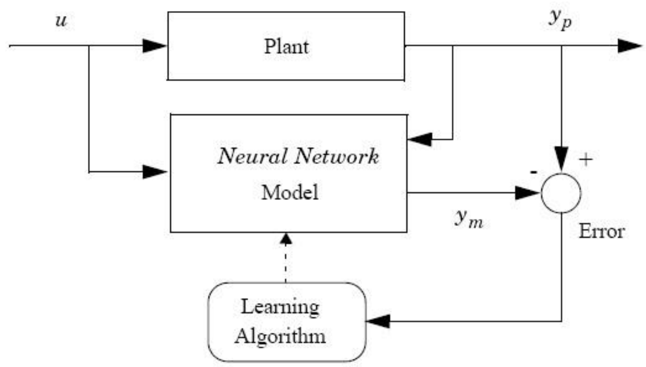

Figure 3: Plant Identification [10]

There are three important parameters that need to be identified for system identification which is network architecture, training data and training parameters. Feedforward multilayer neural network was used for this project with its parameters were shown in Table 3. Training data parameters are shown in Table 4 while the training parameters are shown in Table 5.

Table 3: Network Architecture

Parameters	Values
Size of hidden layer	7
Sampling interval (sec)	0.2s
No. delayed plant inputs	2s
No. delayed plant outputs	2s

Table 4: Training Data Parameters

Parameters	Values
Training samples	2000
Maximum plant input	2
Minimum plant input	-2
Maximum interval values (sec)	20s
Minimum interval values (sec)	5s

Table 5: Training Parameters

Parameters	Values
Training epochs	200
Training function	Lavenberg-Marquardt (trainlm)

2.6.2 Predictive Control

The predictive control is used to predict plant future performance based on the horizon receding technique. The prediction will be used to determine the control signal that will minimize the following performance criterion function over the specified time horizon.

$$J = \sum_{j=N1}^{N2} \left(yr(t+j) - ym(t+j)\right)^2 + \rho \sum_{j=1}^{Na} \left(u'(t+j-1) - u'(t+j-2)\right)^2 \quad (3)$$

Where *yr =the desired response*

u'= tentative control signal

y_m=*network model response*

y_r = *desired response*

Table 6 shows the controller parameters for the neural network predictive controller.

Table 6: Controller Parameters

Parameters	Values
Cost horizon (N_2)	7
Control horizon (N_u)	2
Control weighting factor (p)	0.05
Search parameter (a)	0.001
Minimization routine	csrchbac
Iteration per sample time	2

CHAPTER 3

METHODOLOGY

3.1 Procedure Identification

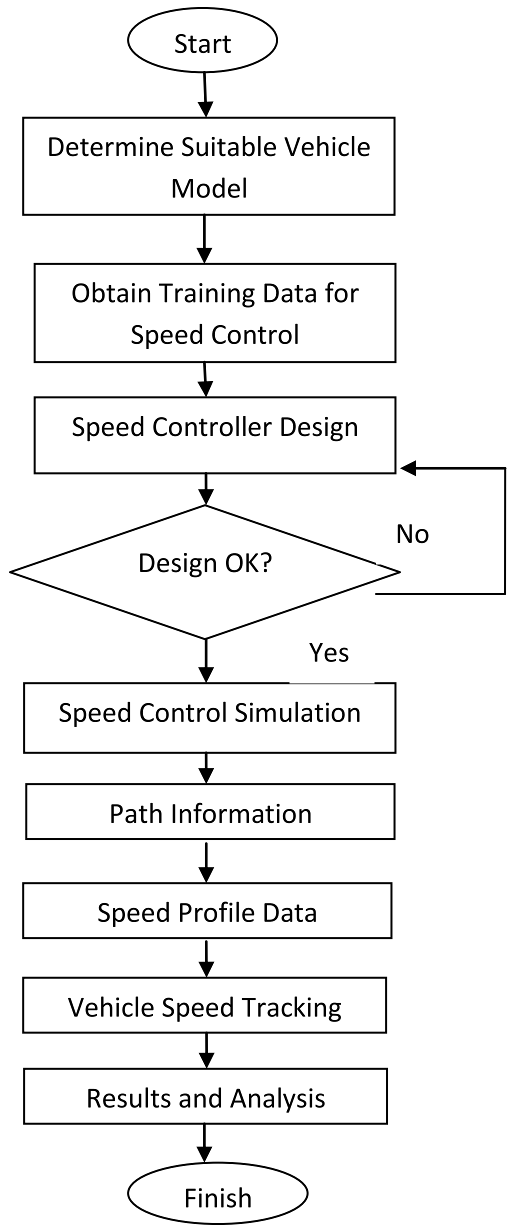

Figure 4: Project Methodology

3.2 Detail Procedure

3.2.1 Determine suitable vehicle model

Linear car model from the work in [2] only consider lateral displacement of the vehicle about its center of gravity (CG). In order to control the longitudinal translation of the linear car model, appropriate model needs to be determined. In order to reduce complexity in the design, another linear car model has been considered which based on simple trolley model [3]. The dynamics of the model describes its important parameters for longitudinal displacement such as longitudinal position, longitudinal speed, front wheel spin angle and rear wheel spin angle. The model as shown in Figure 20 Appendix B is subjected to small velocity perturbation which been superimposed on the trim velocity of the linear car model.

3.2.2 Obtain training data for speed control

Lateral control of the linear car model was re-simulated in order to obtain training data for the speed controller. From the information obtained, speed trajectories which within the ranges of perfect tracking of the speed controller were developed as the reference signal of the controller.

3.2.3 Speed controller design

3.2.3.1 Neural network predictive controller

The choice of the training algorithm and search routine for the neural network predictive controller is quite crucial at this stage as it would determine the performance of the speed controller to track the speed trajectories. However, compromise needs to be made between precise tracking and computing time since perfect speed tracking would require more complex training algorithm and hence increasing computing time. For this project, the parameters for the controller have been described in previous chapter. However, it is worth to draw the architecture of the speed controller to further enhance the understanding.

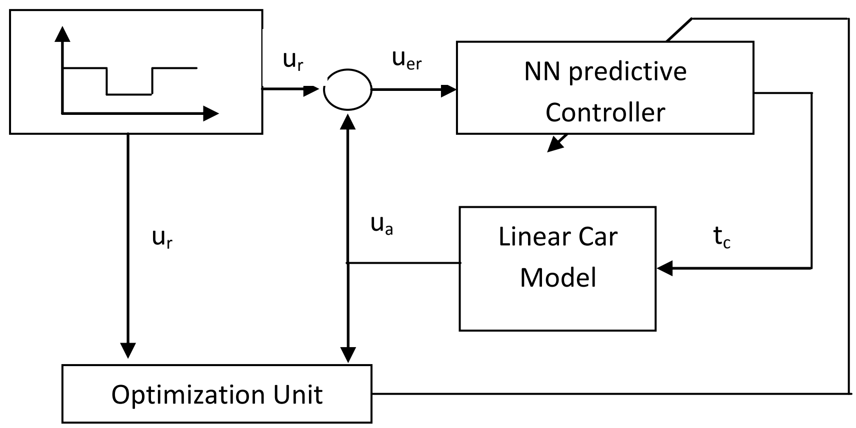

Figure 5: Speed Controller Design

Where u_r is the reference speed trajectories, u_a is the actual speed from the linear car model and u_{er} is the difference between reference speed trajectories and actual speed trajectories of linear car model. The output of the controller is the front wheel drive torque, t_c , which is the input to the linear car model.

3.2.3.2 Neural network and optimal controller

Referring to the work done in [3], the speed tracking system was simulated using optimal controller but the work done in [2] suggested that the tracking performance of the controller could be improved with the use of neural network controller together with optimal controller. Therefore, this project also considers this controller configuration in order to compare its speed tracking performance with the neural network predictive controller. Figure 5 shows the algorithm for neural network and optimal controller configuration,

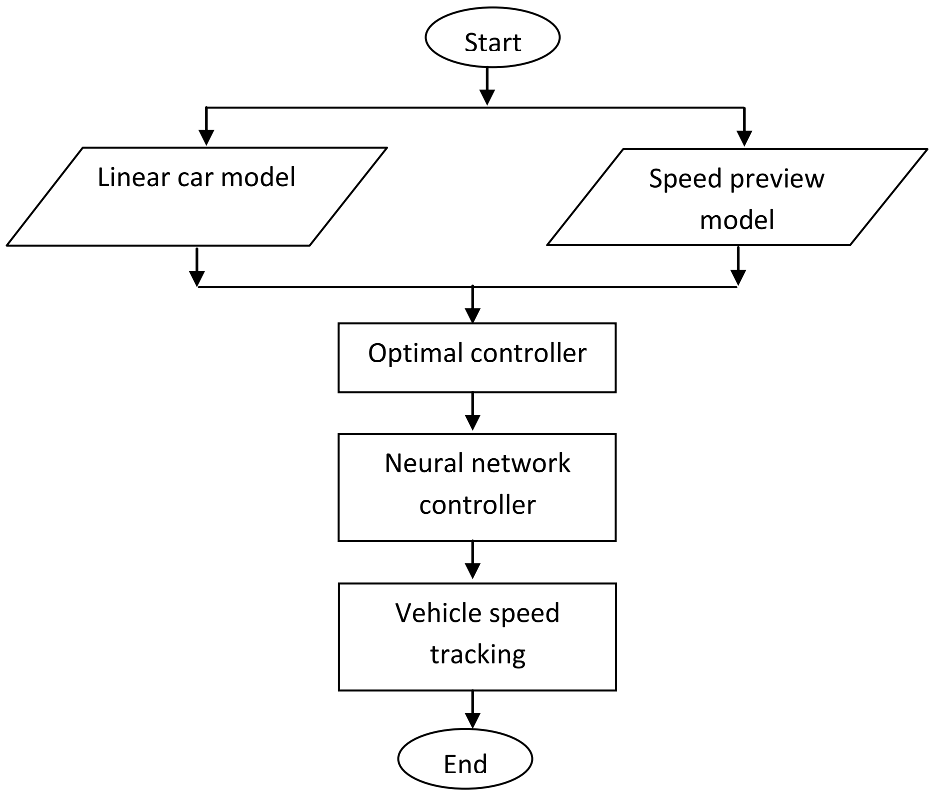

Figure 6: Neural Network with Optimal Controller Algorithm

3.2.4 Speed control simulation

Once the parameters for the speed controller have been determined, the performance of the controller is simulated by inputting appropriate command signal such as random reference signal. After the controller shows satisfactory performance, the desired speed trajectories are fed to the controller and the corresponding results were recorded.

3.2.5 Speed Profile

Speed profile which has been generated from lateral path errors is feed into the speed controller input. The speed controller will track the desired speed trajectories with the help of neural network predictive controller. The tracking performance is dependent on the learning algorithm used, number of the hidden layer and also the search routine used for the controller.

3.3 Tools and Software

The main software used in this project is MATLAB and the toolboxes used in this software are:

- Neural network tool
- Simulink
- M-file

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Neural Network Predictive Controller

4.1.1 Plant Identification

Prior to employing neural network predictive controller, linear car model (plant) needs to be identified by the system. Below are the result obtained from plant identification with 2000 sampling data. Sampling data was taken from random signal in order to train the forward dynamics of the plant. Increasing sampling data will improve the accuracy of the neural network model but too many sampling data could lead to increase in computing time. Therefore, 2000 sampling data was taken as it is optimal to the system.

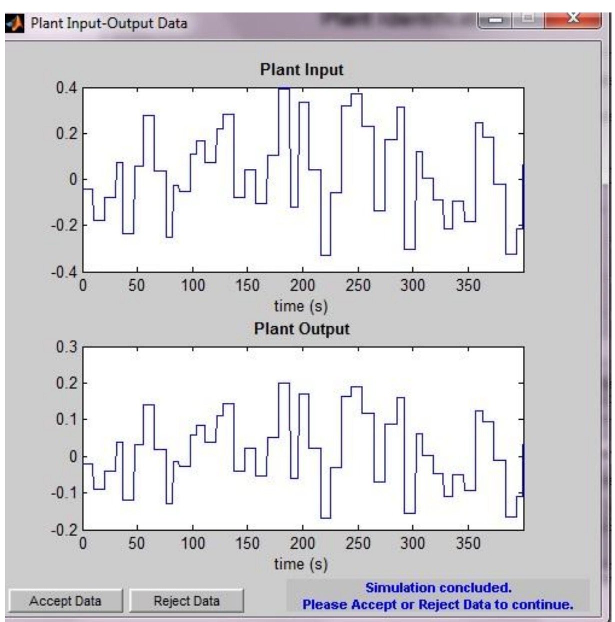

Figure 7: Plant Identification Result

4.1.2 Neural Network Training

Once the plant has been identified, the controller is then trained with the training data. Figure below shows the results from the training. The errors between plant output and NN output are significantly small which indicate that the plant has been well indentified and can approximate actual plant output quite well.

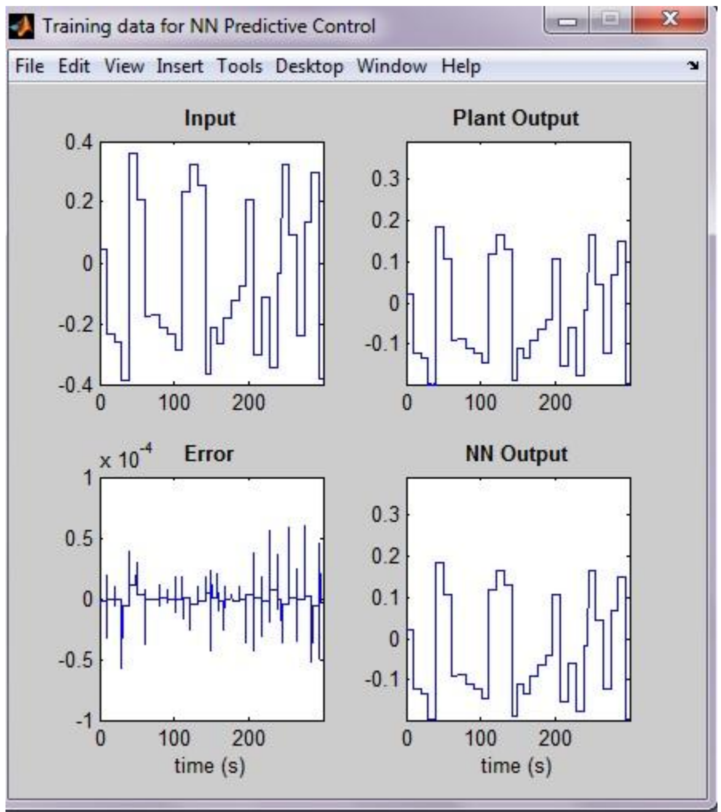

Figure 8: Neural Network Training

Once neural network training was completed, the neural network model was again being validated by randomly supplied with input values. The output of the neural network plant and actual plant were compared to measure the accuracy of the neural network model. The errors shown in model validation window are quite small and acceptable.

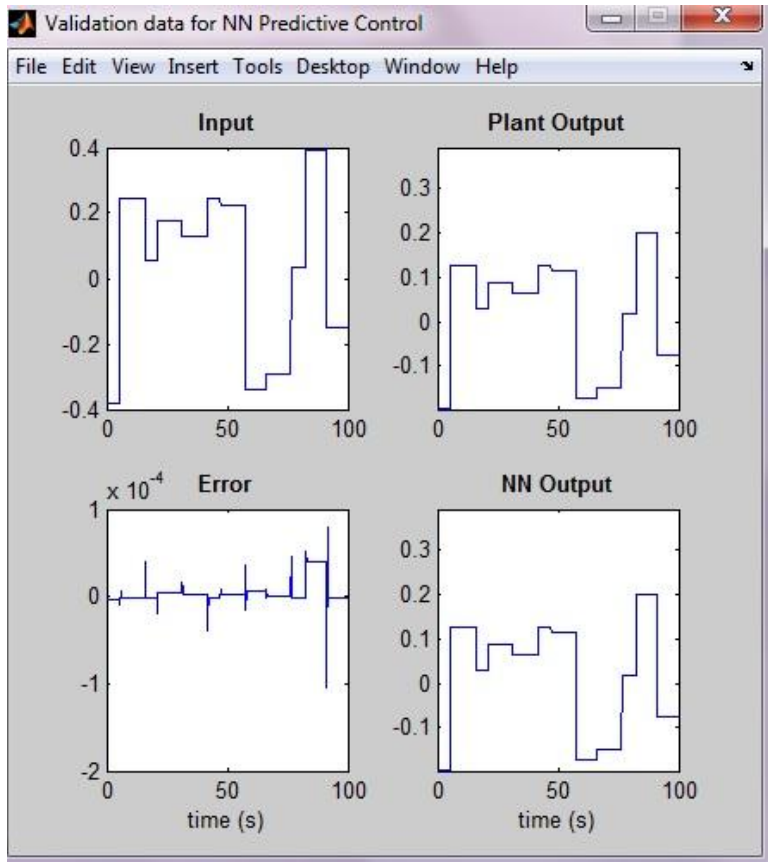

Figure 9: Model Validation

4.2 Simulation Results

4.2.1 Path and Speed Profiles

Path profiles from lane change path and sudden change of direction have been considered for the simulation. These paths are perfect for the purpose of monitoring the lateral path errors both at straight and curve profile.

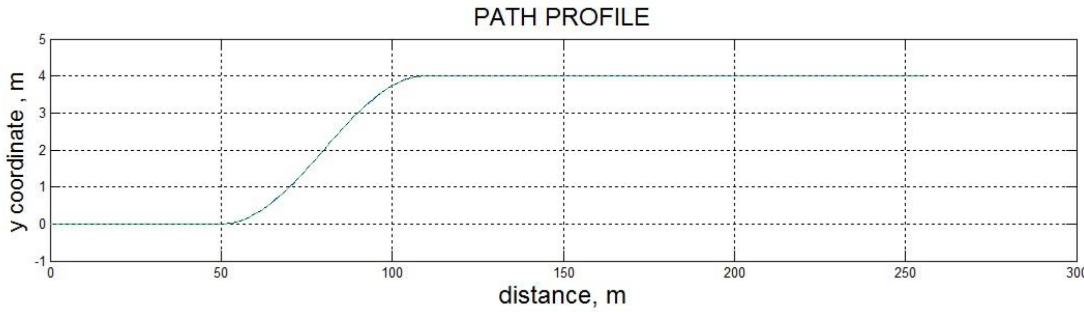

Figure 10: Lane Change Path Profile

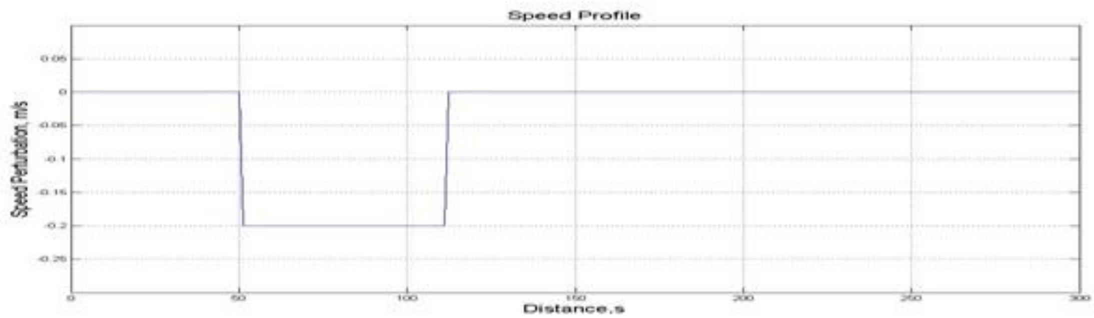

Figure 11: Speed Profile for Lane Change Path

From Figure 8 and Figure 9, speed reduction occurred at 50m away from the initial position which correspond to the first curve on the path profile. The speed resumes to its original value after 100m away from the initial position which correspond to the second turning on the path profile.

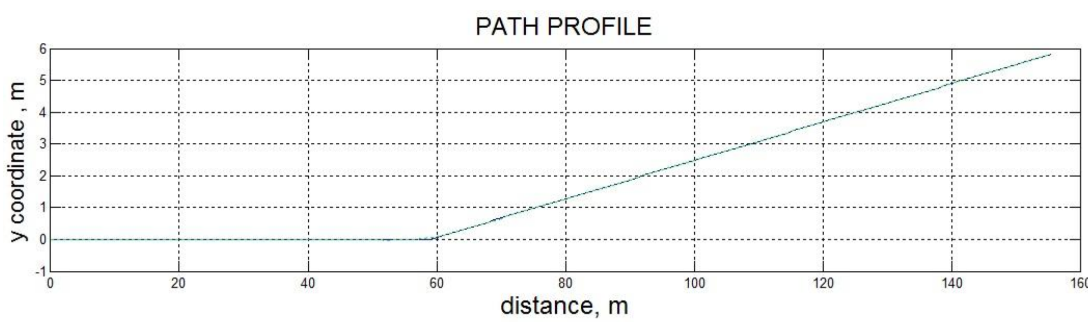

Figure 12: Sudden Change of Lane Profile

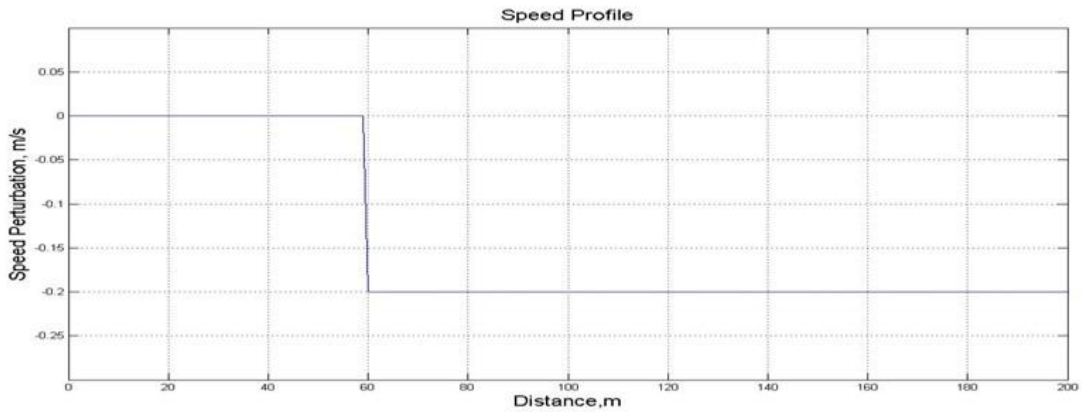

Figure 13: Speed Profile for Sudden Change of Lane

The path profile for sudden change of direction is shown in Figure 10 and the resulting speed profile is shown in Figure 11. For this path profile, speed reduction occurred at 60m away from the initial position and was maintained throughout the path profile.

4.2.2 Speed Tracking Profile

4.2.2.1 Neural Network Predictive Controller

Speed tracking using neural network predictive controller was simulated using two different trim velocities; 20 m/s and 30 m/s. The purpose of introducing trim velocity is to simulate the steady state speed of the linear car model when disturbance is not present. The following results were obtained for two different trim velocities.

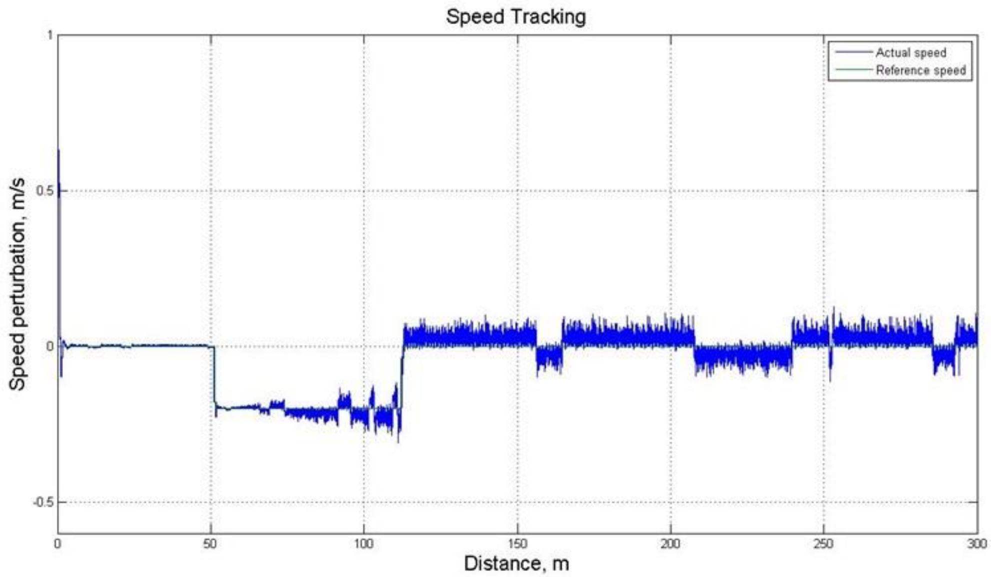

Figure 14: Speed Tracking for Lane Change Path at 20 m/s Trim Velocity

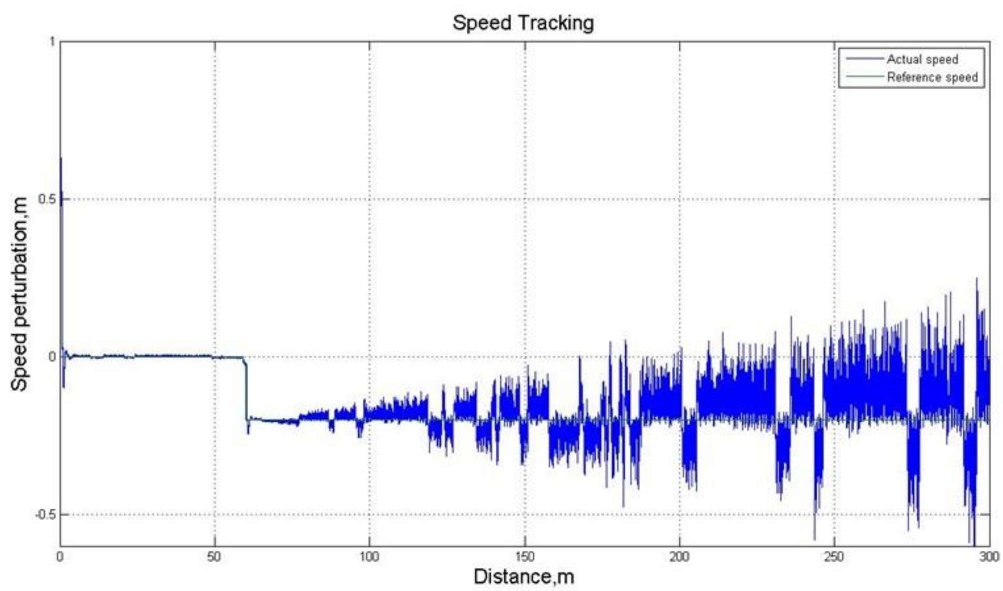

Figure 15: Speed Tracking for Sudden Change of Lane at 20 m/s Trim Velocity

From figure 14 and figure 15, the speed tracking performance experiences excessive fluctuation particularly after the speed change took place. For reduced speed profile, the actual speed seems increasing downward along the reference speed.

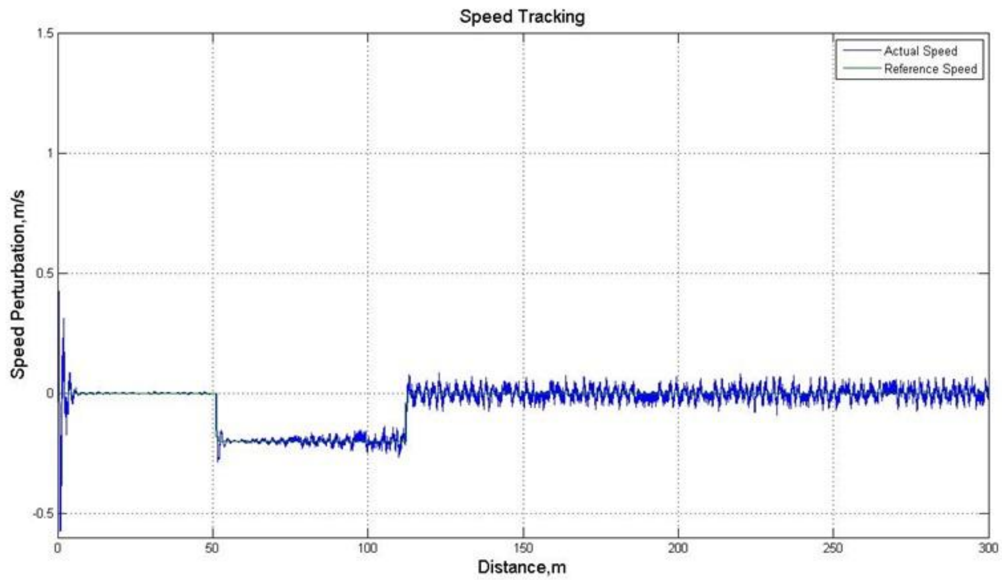

Figure 16: Speed Tracking for Lane Change Path at 30 m/s Trim Velocity

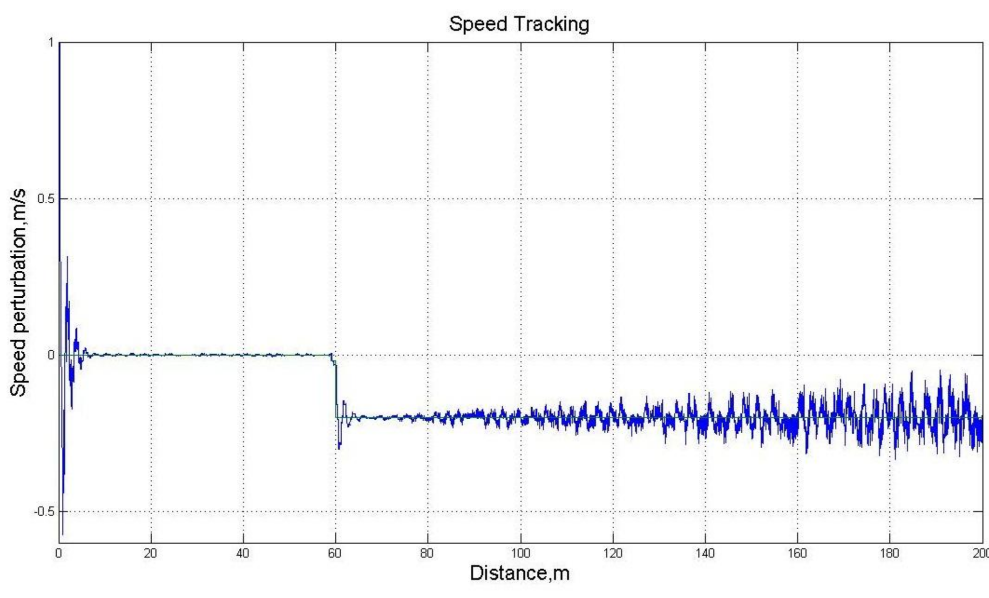

Figure 17: Speed Tracking for Sudden Change of Lane at 30 m/s Trim Velocity

The controller achieves satisfactory performance which can be seen in figure 14 and figure 15. However, it can be seen that the controller experience slight overshoot when the speed reduction occurred.

4.2.2.2 Neural Network and Optimal Controller

Speed tracking also was simulated using neural network and optimal controller. The simulation was conducted using two different trim velocities, 20 m/s and 30 m/s. The desired speed profile is the same as in neural network predictive controller.

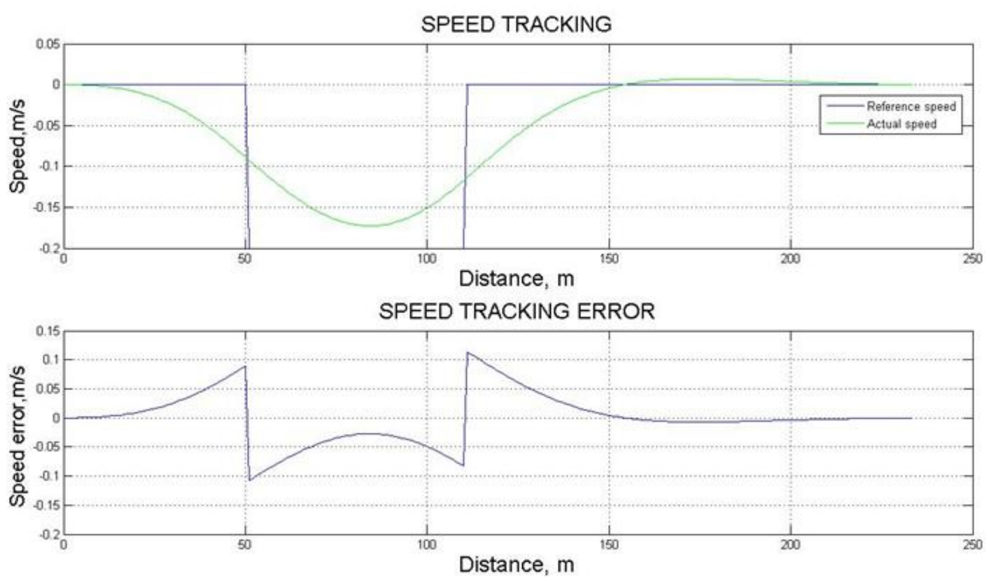

Figure 18: Speed Tracking for Lane Change Path at 20 m/s Trim Velocity (top) and Corresponding Speed Tracking Error (bottom)

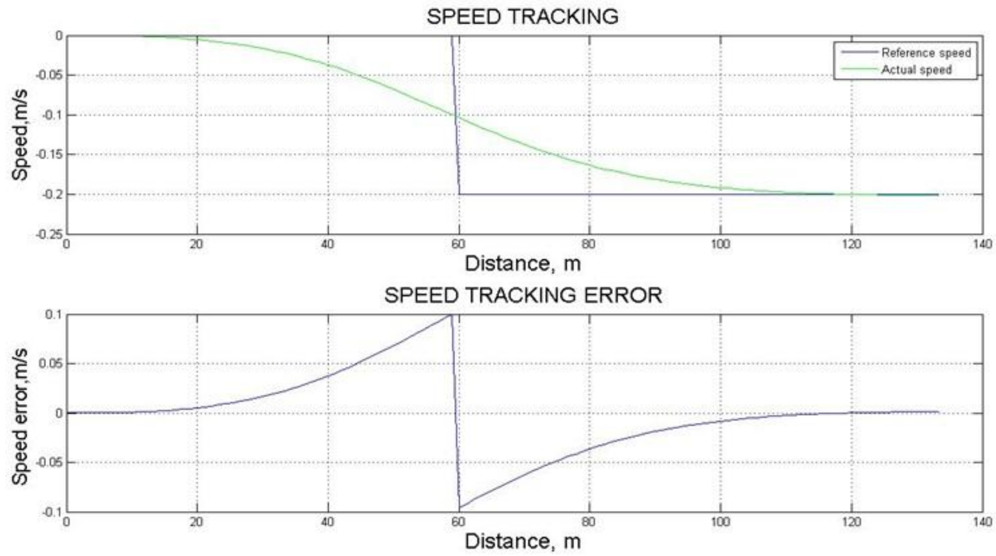

Figure 19: Speed Tracking for Sudden Change of Lane at 20 m/s Trim Velocity (top) and Corresponding Speed Tracking Error (bottom)

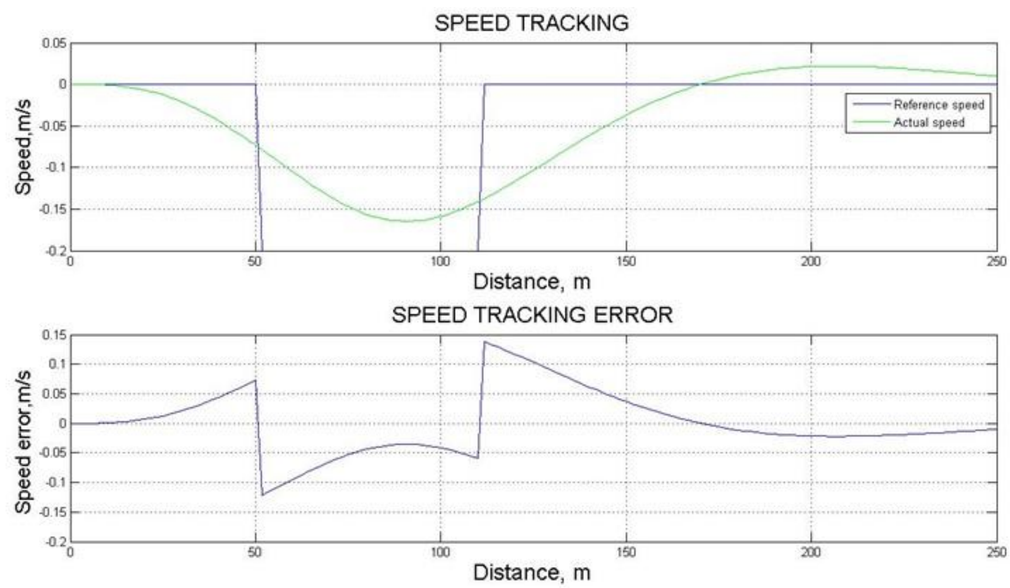

Figure 20: Speed Tracking for Lane Change Path at 30 m/s Trim Velocity (top) and Corresponding Speed Tracking Error (bottom)

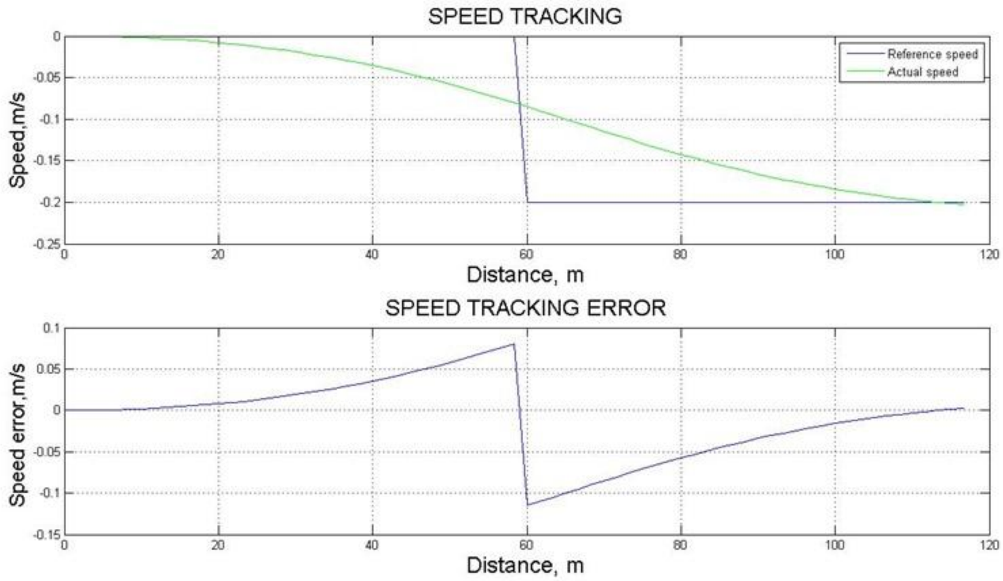

Figure 21: Speed Tracking for Sudden Change of Lane at 30 m/s Trim Velocity (top) and Corresponding Speed Tracking Error (bottom)

The speed tracking for this controller configuration is smoother compared to the neural network predictive controller. The controller also has successfully track the desired speed profile give zero offset at the end of the speed tracking.

4.2.3 Controller Performance

In order to evaluate the performance of the speed controller, the deviation of the actual speed (linear car model's speed) with the reference speed is observed.

Table 7: Maximum Speed Error for Neural Network Predictive Controller

Trim Velocity (m/s)	Path Profile	Maximum Speed Error (m/s)
20	Lane change	0.10
	Sudden change of lane	0.40
30	Lane change	0.10
	Sudden change of lane	0.10

Table 8: Maximum Speed Error for Neural Network and Optimal Controller

Trim Velocity (m/s)	Path Profile	Maximum Speed Error (m/s)
20	Lane change	0.10
	Sudden change of lane	0.10
30	Lane change	0.14
	Sudden change of lane	0.12

The maximum speed errors show the maximum speed error when the linear car model changes its speed from its trim velocity or steady state velocity to the new velocity. Even though the controller performance is not very good but it shows the ability of the controller to track the desired speed profile. Both controller configuration show quite similar maximum speed error, however, since neural network and optimal controller give the smoother speed tracking response, it is the preferred controller configuration. Even though the comparison is done between two different simulation environment (simulink and m-file), the results obtained should give a rough idea on stabilizing the linear car model. The results obtained in simulink environment (nn predictive controller) contain ripple in speed tracking response which might indicate the instability of the controller or the car model.

However, in m-file simulation (neural network and optimal controller), the ripple is not present in the speed tracking response and that shows the stability of both controller and car model in speed tracking.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Neural network predictive controller for vehicle speed control has been developed and the performance of the controller was simulated using two different trim velocities; 20 m/s and 30 m/s. Speed tracking simulation was also conducted using neural network controller that coupled together with optimal controller in order to compare the speed tracking stability with neural network predictive controller.

Great emphasis was given on the choice of the linear car model whereby the model itself should have the component of longitudinal displacement apart from the lateral displacement. To reduce the complexity in this project while preserving the main objective of the project, longitudinal displacement of the vehicle is separately evaluated from its lateral displacement counterpart. The task is important prior to the integration of both components (lateral and longitudinal displacement) in order to represent more realistic vehicle maneuvering problems.

All in all, the resulting speed controller can be applied to control the longitudinal displacement of a vehicle such as in cruise control application. The controller also could enhance existing traction control system which only considers lateral control of the vehicle.

5.2 Recommendations

5.2.1 Improvement on the Reference Speed

In order to simulate more realistic speed trajectories generation, real-time or online speed trajectories translation can be employed. The speed trajectories can be directly input to the speed controller without involving offline calculation / speed generation. In another words, the path information can be directly translated by the system to produce desired speed trajectories for the speed controller.

5.2.2 Integration between Lateral and Longitudinal Displacement

The ultimate aims for this project and previous project is to dynamically control both lateral and longitudinal translation of the vehicle. Both components (lateral and longitudinal displacement) have direct effect to each other. For example, excessive forward speed during cornering could increase the lateral path error of the vehicle from its intended path. Furthermore, by incorporating both dynamics in single state space representation, more realistic linear car model could be developed.

REFERENCES

[1] H. Fritz, *Neural Speed Control For Autonomous Road Vehicles*, Daimler Benz AG, Research Dept. F1M/1A, T728, D-70546, Stuttgart, Germany, June 1995.

[2] N.H.H. Mohamad Hanif, *Path Following Using a Learning Neural Network Controller*, M.Sc. Thesis, Imperial College London, September 2004.

[3] R.S Sharp, *Application of Optimal Preview Control to Speed-Tracking of Road Vehicle*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Professional Engineering Publishing, Volume 221, Number 12/2007, July 2007.

[4] R.S Sharp and V.Valtetsiosis, *Optimal Preview Car Steering Control*, ICTAM : Selected papers from the 20th International Congres of Theoretical and Applied Mechanics, Swets & Zeitlinger, Chicago, 28 August – 1 September 2000

[5] Tan Zhang Yaw, *Neural Network Based Controller For High Speed Vehicle Following Predetermined Path*, B.Sc. Thesis, Universiti Teknologi PETRONAS, December 2006

[6] Shuwen Zhou, Siqi Zhang, Lixin Guo, and Chuanyin Tang, *Vehicle Control Strategy on High Speed Obstacle Avoidance under Emergency*, Berlin Heidelberg: Springer-Verlag, 2008.

[7] Kazuya Tamura, Yukinobu Nakamura, Hiroshi Sekine, and Nobuyoshi Asaanuma, *A Study of Self-Reliant Cornering Speed Control System,* Honda R&D Co.,Ltd, Tochigi R&D Center, 4630 Shimotakanezawa Haga-machi Haga-gun Tochigi 321-33, Japan, 1994

[8] Joshue Perez, Fernando Seco,Vicente Milanes,Antonio Jimenez,Julio C. Diaz and Teresa de Pedro, *An RFID-Based Intelligent Vehicle Speed Controller Using Active Traffic Signals,* Centro de Automatica y Robotica, UPM-CSIC, 28500 Arganda del Rey, Madrid,Spain,2010

[9] Said Mammar and Mariana, *Intergrated Longitudinal and Lateral Control for Vehicle Low Speed Automation*, Proceeding of the 2004 IEEE, International Conference on Control Application, Taipei, Taiwan, September 2-4,2004

[10] Howard Demuth, Mark Beale, and Martin Hagan, *Neural Network Toolbox 6 User's Guide*, The Mathwork Inc., 3 Apple Hill Drive, Natrick, MA, September 2009.

APPENDICES

APPENDIX A

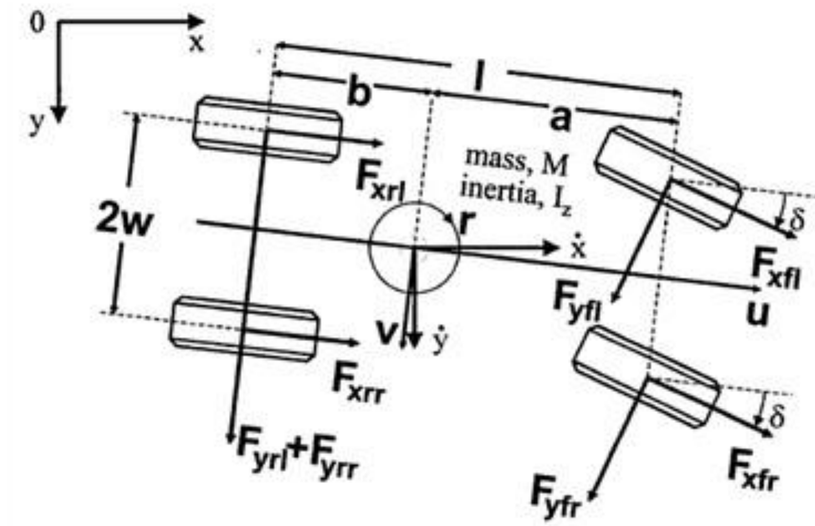

Figure 22: Plan View of Linear Car Model with its Lateral Dynamics

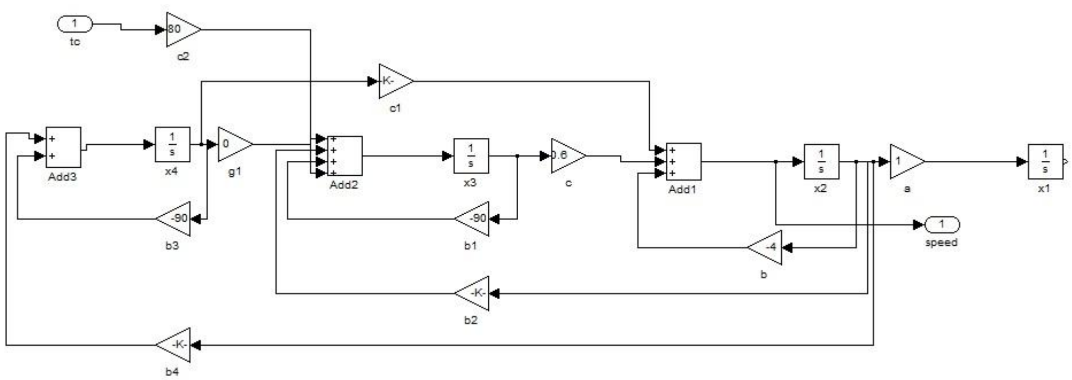

Figure 23: Representation of Linear Car Model in Simulink

APPENDIX B

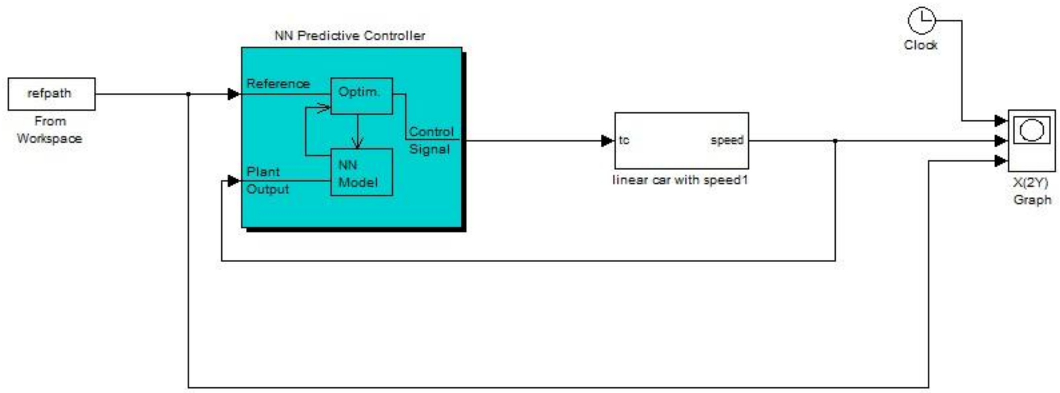

Figure 24: NN Predictive Controller Block

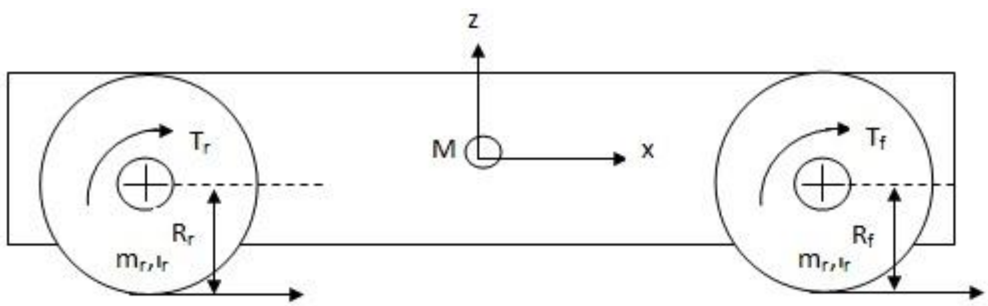

Figure 25: Longitudinal Car Model

APPENDIX C

Main Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %                          LNN.m                            %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


disp('     ----------------------------------------------------------
---------')
disp('       One Single Processing Element, Linear Car Model   ')
disp('     ----------------------------------------------------------
---------')
disp('')
clear all;close all;
% forward speed
    u=input('Base speed ?  (using 20 par (default))');
    u=u*1000/3600; %convert km/h to m/s

    if isempty(u), u=20; disp('Using u=20m/s (default)'), end
% sampling period T
    T=0.2;

% number of preview points
    n=input('how many preview points (using 20 par (default))');
    if isempty(n), n=2/T; disp('Using a number corresponding to 1sec
ahead (default)'), end

%car parameters definition
Cfx=80000;
Crx=80000;
Rf=0.30;
Rr=0.30;
M=800;
If=2;
Ir=2;
mf=100;
mr=100;
Tf=0.37;
Mt=M+mf+mr;

%%%%%%%%%%%%%%%%%%%% Speed Model Matrices %%%%%%%%%%%%%%%%%%%

%speed shift operators

D=[zeros(n,1) eye(n); zeros(1,n+1)];
%D=[0 1 0 0 0 0 0 0 0 0 ;
%   0 0 1 0 0 0 0 0 0 0 ;
%   0 0 0 1 0 0 0 0 0 0 ;
%   0 0 0 0 1 0 0 0 0 0 ;
%   0 0 0 0 0 1 0 0 0 0 ;
```

```
%   0 0 0 0 0 0 1 0 0 0 ;
%   0 0 0 0 0 0 0 1 0 0 ;
%   0 0 0 0 0 0 0 0 1 0 ;
%   0 0 0 0 0 0 0 0 0 1 ;
%   0 0 0 0 0 0 0 0 0 0 ];

E=[zeros(n,1); 1];
%E=[0;0;0;0;0;0;0;0;0;1];

%%%%%%%%%%%%%%%%%%%% Linear Car Model %%%%%%%%%%%%%%%%%%%%%%
longitudinal_car_model_mfile
disp('')

%%%%%%%%%%%%%%%%% Linear control gain calculaton %%%%%%%%%%%%%%

Q=[100 0 ;       %Q=[q1 0
   0  1 ];       %   0 q2]
R2=1;

LQRgain

%%%%%%%%%%%%%%%%%%%%%% Linear cost parameters %%%%%%%%%%%%%%

%the cost to be minimised is the folowing one :
%J=Z(:,k)'*R1cost*Z(:,k)+delta(k)'*R2cost*delta(k)

R1cost=R1;
R2cost=R2;

tic %Start a stopwatch timer
disp('  Loading speed information......')

for epoch=1:5

    RefSpeedProfile     %speed perturbation

    [K,nb]=size(uref'); %array size of uref


%%%%%%%%%%%%  State definition & initialisation %%%%%%%%%%%%%

% At each time step, a new global frame is defined
% The state is based on a frame comprising the local x
% and y-axes of the vehicle

%     Z=[  local lateral displacement v   ]
%       [  vdot                           ]
%       [  local angle phi                ]
%       [  phidot                         ]
%       [  local lateral preview errors   ]

Z = zeros(4+n+1,K-n-1);
Z(2,1) = uref(1);         % Longitudinal speed initialization
```

```
Z(3,1) = uref(1)/Rf;      % Init. of front and rear wheel spin
velocity
Z(4,1) = uref(1)/Rr;
Z(4+1:4+n+1,1) = uref(1:n+1)'; % Local Speed Preview Errors

%augmented E matrx

Ebis=[zeros(4,1); E];

%%%%%%%%%%%%%%%%%%%%%%% Paramaters Initialisation %%%%%%%%%%%%%%%%%%

%sensitivity functions initialized to 0

        dzdw = zeros(n+5,n+5);
        dudw = zeros(1,n+5);
        dJdw = zeros(1,n+5);   %to be multiplied with gama
        prevdJdw = zeros(1,n+5);
        deltaw = zeros(1,n+5);   %to be added to w to obtain w(k+1)
        prevdeltaw = zeros(1,n+5);

    %other parameters
        phi(1)=uref(1)/Rf;  %current front wheel spin vel.

        tc(1)=0;            %Initialisation of front wheel torque

        global_speed(1)=Z(2,1); %Initialisaion of speed tracking

        Zinit = zeros(4+n+1,1);

        Zstep = zeros(4+n+1, 1);

 %%%%%%%%%%%%%%% Neural network implementation%%%%%%%%%%%%

disp('               neural network implementation.....')

%choose an input layer with n+4 (number of states) neurons
input=[-50*ones(n+5,1) 50*ones(n+5,1)];
net=newlin(input,1);

%Initialize the vector W(:) containing all weights and biases.
if epoch==1
    for jg=1:4+n+1
        W(jg)=Kt(jg); %Weight based coeff obtained from optimal ctrl
theory (LQRgain.m), initial weighting parameters
        W_init=W;
    end

    %fixed learning rate
    gama=0.1;
    gama_init=gama; %Storing the initial learning rate
    gama_next(1)=gama;

else
    W=W_last; %Last updated weight from previous epoch
```

```
    gama=gama_last; %last updated learning rate from previous epoch
    gama_next(1)=gama;
end

%Initialize neural network weightings
net.IW{1,1}=W;
net.b{1}=[0];

toc %Read the stopwatch timer
disp('  main loop....')
tic %start another stopwatch timer


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%                                   MAIN LOOP
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


for k = 1:(K-n-1) % 300-preview point-1
%definition of a new global frame based on the local x and y axes of
the
%car
           % definition of the state of the car
            Zinit = Z(:,k);
            Ydot = Z(2,k);


        % due to the choice of the frame, absolute positions become
zero
            Zinit(2) = 0;
            Zinit(3) = 0;
            Zinit(4) = 0;

        % absolute to relative road data transformation

            local_urefs = uref(k:k+n+1);

                for j = 1:(n+2),

            local_urefs(j) =  local_urefs(j) - global_speed(k)- ...
                    (j-1)*phi(k)*u*T;


                end


        % definition of the remaining states (preview speed errors)
            Zinit(4+1:4+n+1) = local_urefs(1:n+1);

    %%%%%%%%%%%%%%%%%state error%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        epsB=Zinit;
```

```
    %%%%%%%%%%%%front wheel torque %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

       tc(k) = sim(net,-epsB); %Simulate speed tracking

    %%%%%%%%%%%%%state update%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Zstep=A*Zinit+B*tc(k)+Ebis*local_urefs(n+2);

    %%%%%%%%%%%%%%%Weighting update%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            %dudw(k) calculation
                dudw= -( Zstep' + W*dzdw);
% u is for front wheel torque

%dJdw(k) calculation and keeping the previous derivative of the cost
                prevdJdw=dJdw;
                dJdw=2*Zstep'*R1cost*dzdw+2*tc(k)*R2cost*dudw;

            %dzdw(k+1) calculation
                dzdw=A*dzdw+B*dudw;

            %adaptive learning rate

                if dJdw/prevdJdw<1.000     % Cost ratio
                    gama=1.05*gama;
                end
                if dJdw/prevdJdw>1.005
                    gama=0.7*gama;
                end
%difference calculation Polak Ribiere or Gradient method
                deltaw=-gama*dJdw;   %value for deltaw
               gama_next(k+1) = gama;

            %weighting update
            W=W+deltaw;
            net.IW{1,1} = W;

%%%%%%%%%%%%%%%%%% END OF WEGHTINGS UPDATE   %%%%%%%%%%%%%%%%%%%

    %update absolute speed
        global_speed(k+1) = global_speed(k) + u*T*phi(k) +
Zstep(2,1);

        phi(k+1) = phi(k) + Zstep(3,1);

   %store the state
    Z(:,k+1) = Zstep;


end

toc

gama_last=gama_next(K-n);
```

```
gama_end(epoch)=gama_last;
W_last=W;
W_end(epoch,1:4+n+1)=W_last(1,1:4+n+1);
end

Ws=W_init+0.0001;
xyz=[];
for r=1:jg;
    xyz(r)=abs((net.IW{1,1}(r)-Ws(r))/Ws(r))*100;
end
weight_change=(sum(xyz))/jg;

%%%%%%%%%%%%%%%%%%END OF MAIN LOOP %%%%%%%%%%%%%%%%%%%%%%%

PlottingSpeed

%%%%%%%%%%%%%%End of Program%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Speed Profile Generation

```
%%%%%% RefSpeedProfile.m %%%%%%%%%

    xr1=[0:u*T:50-u*T];
    xr2=[50:u*T:50+60];
    xr3=[110+u*T:u*T:300];
    xref=[xr1 xr2 xr3];
    yr1=0*xr1;
    yr2=2-2*cos((pi/60)*xr2-50*pi/60);
    yr3=4*ones(size(xr3));
    uref=[yr1 yr2 yr3];

     ur(1)=0;
    [Kr,dr]=size(uref');
    for t=2:Kr
        rx=uref(t)-uref(t-1);
        if rx==0
            ur(t)=0;
        else
            ur(t)=-0.2;
        end
    end

    uref=ur;

%%%%% End of RefSpeedProfile.m %%%%%%%%
```

Optimal Controller

```
%%%%%%%%%%%%%%%%%%%%%%%%% LQRgain.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Compute the Linear control gain obtained with the LQR theory

    C=[ 1 0 0 0    -1         0       zeros(1,n-1) ;
        0 0 1 0 1/(u*T) -1/(u*T)    zeros(1,n-1)   ];

    R1=C'*Q*C;

    A=[ Ad             zeros(4,n+1);                 % Car model
        zeros(n+1,4)       D       ];                % Road model
    B=[     Bd;
       zeros(n+1,1)];

%non-preview gain using the DLRMI function
%We could directly have used [Kt,Sbis,Ebis] = DLQR(A,B,R1,R2)
%but the time to compute would be much greater

    [K1,P11] = dlqr(Ad,Bd,R1(1:4,1:4),R2);     % K1 is gain matrix :
feedback gain
                                                % P11 is Riccati
equation solution

% Use non-preview results to solve the preview problem.

    FC = Ad - Bd*K1;             % xdot

    P12 = zeros(4,n+1);
    P12(:,1) = R1(1:4,4+1);      % replaces the whole row and first
column of P12 with first four rows
                                 % and fifth column of R1

    for i=2:n+1
        P12(:,i) = R1(1:4, 4+i) + FC'*P12(:,i-1);
    end

    K2 = inv(Bd'*P11*Bd + R2) * Bd' * P12 * D;

    Kt=[K1 K2];

%%%%%%%%%%%%%%%%%%End of LQRgain.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Linear Car Model

```
%%%%% Longitudianl_car_model_mfile.m %%%%%%%%%%%%%%%
%Linear car model with variable forward speed

Acar = [0   1                   0                    0;
        0 -(Cfx+Crx)/(Mt*u)  (Cfx*Rf)/(Mt*u)        (Crx*Rr)/(Mt*u);
        0 (Cfx*Rf)/(If*u)     -(Cfx*Rf^2)/(If*u)      0;
        0 (Crx*Rr)/(Ir*u)         0                -
(Crx*Rr^2)/(Ir*u);];

Bcar = [0;
    0;
    Tf/If;
    0;];

Ccar = [0 1 0 0];
Dcar = [0];

%discrete state space of the car model
[Ad Bd]=c2d(Acar,Bcar,T);
```