Lane Keep Assist System for Chevrolet Blazer

Xinyu Cao

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Mechanical Engineering

University of Washington

2020

Committee:

Robert Bruce Darling

Brain Fabien

Sawyer Fuller

Program Authorized to Offer Degree:

Mechanical Engineering

University of Washington

Abstract

Lane Keep Assist System for Chevrolet Blazer

Xinyu Cao

Chair of the Supervisory Committee:

Robert Bruce Darling

Department of Electrical & Computer Engineering

This thesis presents the design process of a lane keep assist (LKA) system for a hybrid electric vehicle, Chevrolet Blazer. The vehicle is implemented with a P4 Split-Parallel hybrid architecture, the vehicle's front axle is powered by a GM 2.0L I-4 Turbo engine coupled with a 9-spd automatic transmission and an eRAD motor is used to propel the rear axle. The primary goal of this thesis is to present the construction of a hybrid electric vehicle system model, the development of an LKA system, and the tests and validations of the system. The vehicle system is modeled according to the P4 Split-Parallel hybrid architecture first, and then the LKA system is developed based on the model predictive control algorithm. Afterwards, tests and simulations are conducted through the whole system, which is a combination of the vehicle model and the LKA system. The test

results show that the LKA system has the ability to tracking a path with small errors, especially at low and medium speed.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Research Background

The University of Washington (UW) EcoCAR team is currently working on the EcoCAR Mobility Challenge. The EcoCAR Mobility Challenge is the current four-year Advanced Vehicle Technology Competition (AVTC) that challenges 12 universities. The UW EcoCAR team is following a real-world vehicle development process to design, integrate and refine a new, advanced technology, energy-efficient mobility solution for the competition vehicle, Chevrolet Blazer. The team is currently working in Phase 2 of the competition; the team has selected a hybrid electric architecture and completed the baseline vehicle evaluation in year one, and the team is working on the subsystem design, including the lane keep assist (LKA) system.

The primary goal of this thesis is to present the design process of the LKA system, including the construction of the vehicle model, the development of the LKA system, and the test and validation of the system. The vehicle model is used as the simulation platform to facilitate the development and testing of the LKA system. In order to most effectively introduce the topics covered in this thesis, some background information about the hybrid electric vehicle and LKA system are provided in the following sections, which is followed by a section about the thesis outline.

1.2 Hybrid Electric Vehicle Architecture Description

A Hybrid Electric Vehicle is a specific type of Hybrid Vehicle that powered by a conventional internal combustion engine (ICE) subsystem in combination with another

electric motor subsystem [1]. These two subsystems compose a more complicated and integrated propulsion system. The ICE subsystem is usually powered by gasoline, diesel, or biofuels stored in the fuel tank, and its structure is similar to a conventional vehicle's propulsion system. In terms of the electric motor subsystem, one or more motors that use energy stored in the battery are used for propulsion. The extra power provided by the electric motor may allow a smaller ICE and the battery can also power auxiliary loads and reduce engine idling when the vehicle is stopped. Together, the existence of the electric subsystem results in better fuel economy and better performance than a conventional vehicle. The propulsion system and some basic components of a hybrid electric vehicle are shown in Figure 1.1.

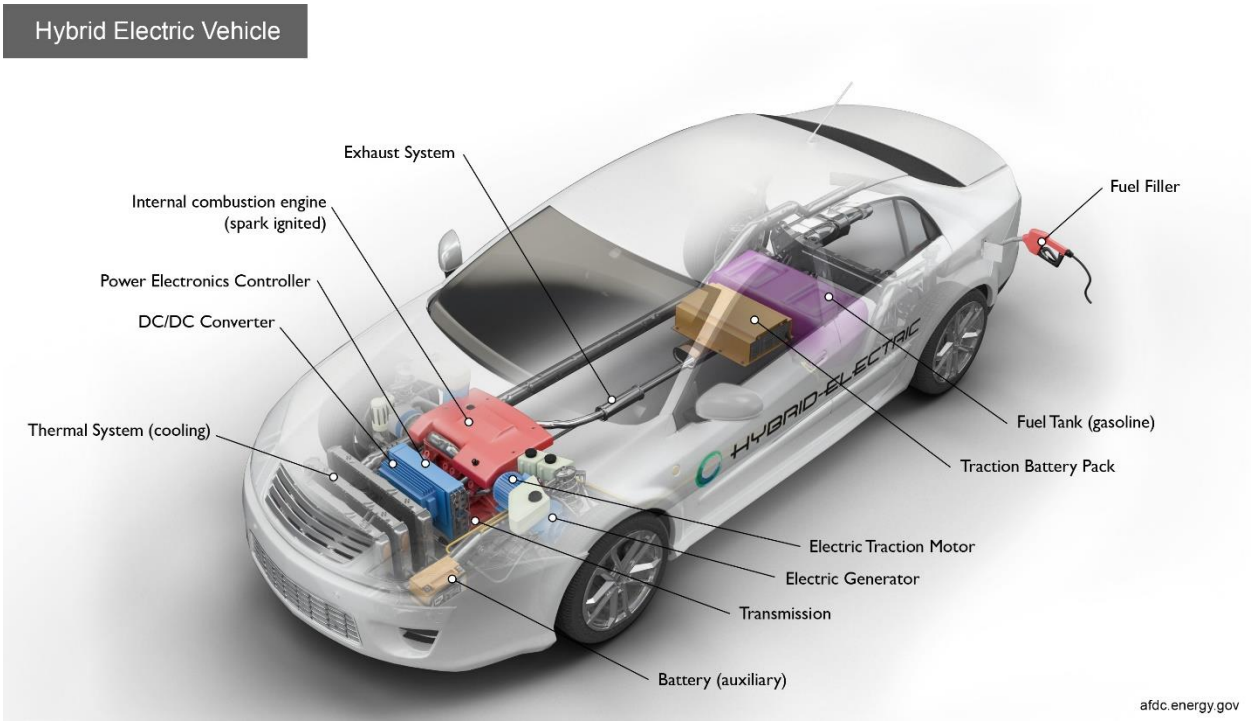

Figure 1.1: Basic Structure of Hybrid Electric Vehicle[3]

Hybrid electric vehicles can be classified into several types according to the way in which power is delivered to the road surface. Regardless of the type of hybrid electric

vehicle, its powertrain is usually made up of an ICE, an electric motor, an energy storage system (ESS), and a transmission system[2]. These components can be integrated with different ways and sizes which results in variation in the vehicle's architecture. Based on the component integration and the way in which power from the engine and the electric motors is delivered to the drivetrain, the vehicle's architecture can be classified into three categories, which are parallel hybrid, series hybrid, and series-parallel hybrid.

Parallel hybrid is the most common design for hybrid electric vehicles, its ICE and the electric motor are connected to the wheels through the mechanical coupling, which can deliver the power to the wheels simultaneously through a conventional transmission. There is no separate generator in a parallel hybrid, the electric motor not only provides torque to the driven wheels but also functions as a generator when using regenerative braking. A smaller battery pack is used in the parallel hybrid, and it usually relies on the regenerative braking to recharge. The structure of a parallel hybrid architecture is shown in Figure 1.2.

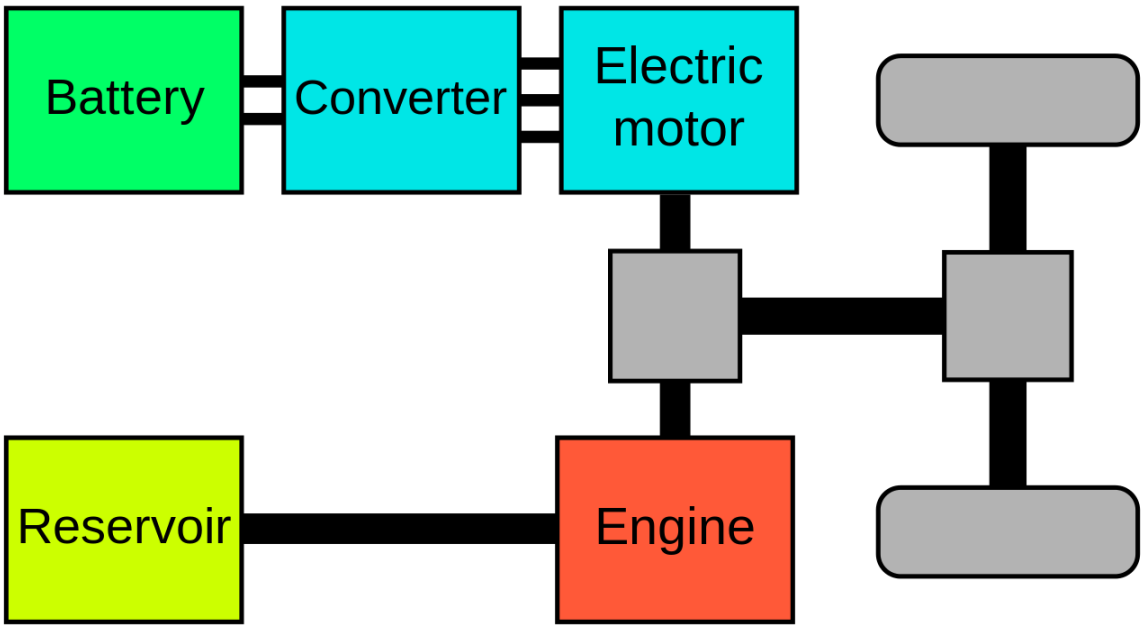

Figure 1.2: Structure of a parallel hybrid electric vehicle[4]

Series hybrids the simplest hybrid configuration, it is also referred to as an extended-range electric vehicle or range-extended electric vehicle. In series hybrid architecture, only the electric motor is used to drive the wheels, which is powered by a larger battery pack. The entire mechanical transmission between the ICE and the wheels is replaced by an electric generator, and the ICE only works as a generator to charge the battery pack or power the electric motor. The electric motor receives power from the battery or directly from the ICE and provides torque to the driven wheels. The structure of a series hybrid architecture is shown in Figure 1.3.

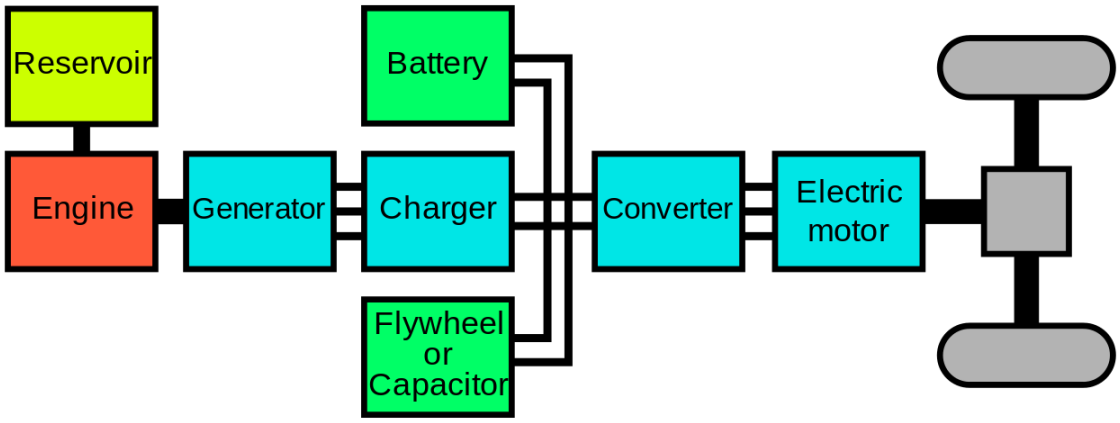

Figure 1.3: Structure of a series hybrid electric vehicle[4]

Power-split hybrid, also called series-parallel hybrid, combines the features of both parallel and series hybrid architecture. Both the engine and the electric motor can drive the wheels directly, and the power distribution between the engine and the electric motor is designed so that the engine can run in its optimum operating range as much as possible. Depending on the different driving conditions, the vehicle uses the power from the electric motor or the power from both the electric motor and the engine. Compared with a pure parallel hybrid, a series-parallel hybrid system requires a separate generator and a larger

battery pack, which makes the system more complicated. The structure of a series-parallel hybrid architecture is shown in Figure 1.4.

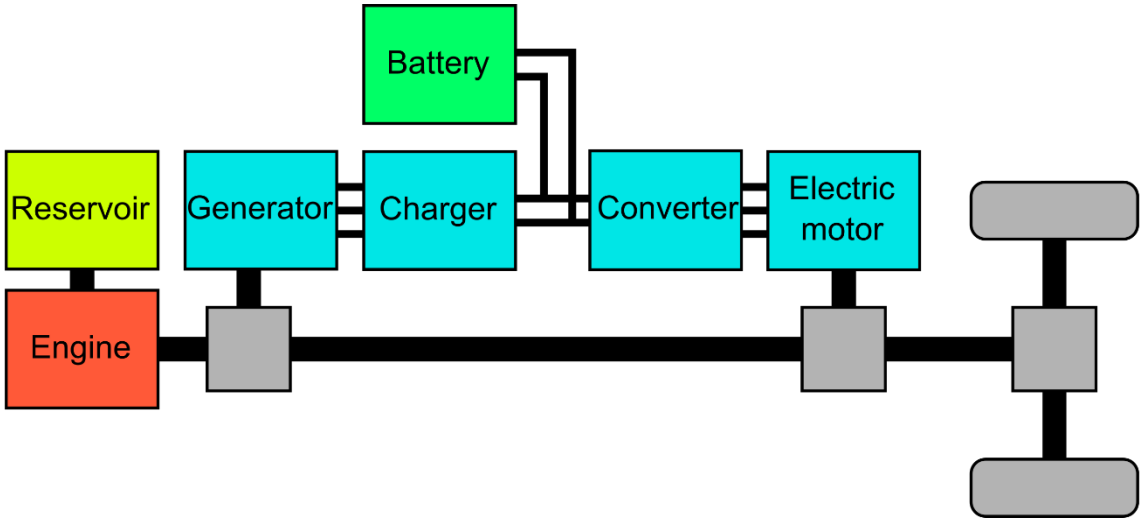

Figure 1.4: Structure of a series-parallel hybrid electric vehicle[4]

1.3 Lane Keep Assist System

A lane keep assist system is a type of driver assistance system, and it is a further development of the lane departure warning (LDW) system. Both systems provide electronic driving aids to help the driver keep the vehicle in the driving lane. Lane departure warning system is a straightforward warning system that merely alerts the driver when the vehicle is leaving the driving lane, whereas the LKA system is able to provide automatic steering to support the driver in staying within the lane. These systems are often bundled with techniques that perform other functions, such as adaptive cruise control and autonomous emergency braking.

1.4 Thesis Outline

After the introduction chapter, the thesis focuses on the development of a hybrid electric vehicle model for Chevrolet Blazer in Simulink. It is followed by a chapter discussing the approach and methodology to build the LKA system. Afterwards, simulations and tests are discussed. Finally, the conclusion and outlook for future work are discussed.

Chapter 2 Development of Vehicle Model

This chapter presents the construction of the hybrid electric vehicle model. It begins with a description of a hybrid vehicle model structure, including its overall structure, vehicle powertrain system, braking system, steering system, and the supervisory controller. Afterwards, it moves on to describe the vehicle dynamics via dynamics equations, followed by the model construction in MATLAB/Simulink platform.

2.1 Vehicle Model Structure

The overall vehicle model can be separated into different subsystems, to guarantee basic driving functions, four major subsystem models need to be built are the following:

- Powertrain System Model
- Braking System Model
- Steering System Model
- Supervisory Controller

This section first focuses on the vehicle's powertrain architecture, and then the braking system and steering system are studied to make the model have the basic driving functions. At last, a supervisory controller containing several individual control systems is connected to the three subsystem models mentioned before.

2.2 Powertrain System Model

Compared with a conventional vehicle, a hybrid electric vehicle has a more complicated powertrain system. As mentioned before, a hybrid electric vehicle is usually powered by an internal combustion engine (ICE) propulsion system in combination with another electric motor propulsion system[2]. In the second propulsion system, electric motors can be placed in various positions throughout the propulsion system, which results in different configurations. Figure 2.1 shows five major positions where electric motors can be placed.

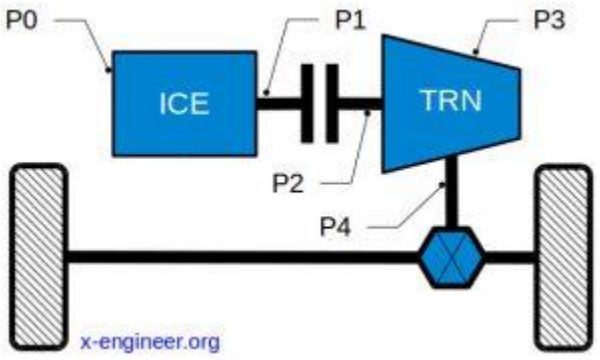

Figure 2.1: Electric Motor Positions[5]

The powertrain system is composed of these two propulsion systems and other components that deliver the power to the road. The powertrain takes the acceleration pedal position (AP) and brake pedal position (BP) as inputs to generate and deliver corresponding power to the wheels.

In the first year of competition, the team proposed two split-parallel hybrid electric vehicle architecture options with different components. Architecture 1 is a P4 Split-Parallel Hybrid Electric Vehicle architecture. In this architecture, two sub-propulsion systems are operated in parallel to deliver torque to the road, with the ICE on the front axle and the P4

Traction Motor on the rear axle. Torque can flow from the rear wheels to the P4 Traction Motor through regenerative braking to generate electricity and store the energy in the ESS[6]. Architecture 1 is described through the diagrams and tables below.

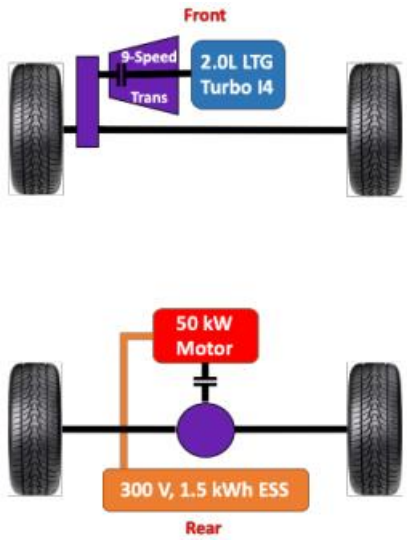

Figure 2.2: Structure of P4 Split-Parallel HEV[6]

Table 2.1: P4 Split-Parallel HEV Propulsion System Components[6]

Component	Specification
Fuel	E10, 93 Octane
Engine	GM 2.0L LTG, 191 kW, 400 Nm
Transmission	GM M3E (9T50)
P4 Traction Motor	Magna eRAD, 50 kW, 200 Nm with Disconnect
Motor Gear Reduction	Single-speed 9.1727:1
ESS	GM Li-on (NMC), 62 kW, 80s1p

The second architecture is a P0-P4 Split-Parallel Hybrid Electric Vehicle, its structure and operation are a little different from the Architecture 1. A Belted Alternator Starter (BAS) Motor is installed at the front of the vehicle, which allows torque not only to flow from the rear wheels to the real reaction motor but also from the engine to the BAS

Motor (during regenerative braking). Architecture 2 is described through the diagrams and tables below.

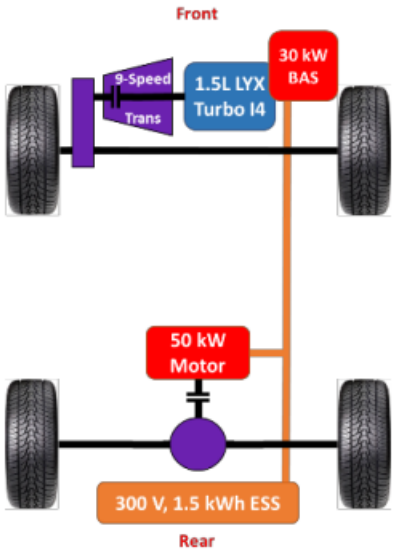

Figure 2.3: Structure of P0-P4 Split-Parallel HEV[6]

Table 2.2: P0-P4 Split-Parallel HEV Propulsion System Components[6]

Component	Specification
Fuel	E10, 87 Octane
Engine	GM 1.5L LYX, 126 kW, 282 Nm
Transmission	GM M3U (9T45)
P4 Traction Motor	Magna eRAD, 50 kW, 200 Nm with Disconnect
Motor Gear Reduction	Single-speed 9.1727:1
P0 BAS Motor	Denso ISG, 30 kW, 60 Nm
P0 Belt Reduction Ratio	2.57:1
ESS	GM Li-on (NMC), 62 kW, 80s1p

The UW EcoCAR team has planned to implement the P4 Split-Parallel hybrid electric vehicle architecture on the Chevrolet Blazer given less vehicle mass and structure change. On the front axle of the vehicle, the original Chevrolet Blazer engine is replaced

by a General Motors 2.0L I-4 Turbo engine paired with a 9-Spd Automatic transmission. On the rear axle, a Magna eRAD drive unit powered by the General Motors 1.5kWh HESS battery is installed. Thus, the powertrain system model is composed of the following four component models:

- Engine Model
- 9-Spd Automatic Transmission Model
- Electric Motor Model
- Energy Storage System Model

Each component model is built based on the component's specification.

2.2.2 Braking System Model

The braking system for a hybrid electric vehicle with P4 Split-Parallel architecture is different from the conventional vehicle. Besides the friction braking system installed on the front wheels, the electric motor is also used for regenerative braking to slow down the vehicle. In most conditions, both friction and regenerative braking are used to produce the required total braking. Thus, the braking system model is composed of a friction braking model and the electric motor model.

2.2.3 Steering System

The steering system converts the rotation of the steering wheel into a swiveling movement of the road wheels in such a way that the steering-wheel rim turns a long way to move the road wheels a short way. Thus, only a converter needs to be added between

the steering wheel and the front wheels. The converter represents the steering mechanism connecting the steering wheel and the front wheels.

2.2.4 Supervisory Controller

The supervisory controller is a compact and robust in-vehicle prototyping system that contains different control systems. These systems take driver's commands or the feedbacks of the vehicle system as inputs and then process these inputs to send control commands to different components. Three main control systems of the supervisory controller in this research are:

- Lane Keep Assist System
- Propulsion Control System
- Braking Control System

Lane keep assist system is the main content of this research, which will be presented in the next chapter. Propulsion control system is used to perform power split to optimize energy consumption with a simplified Equivalent Consumption Minimization Strategy (ECMS). It also converts driver inputs to corresponding output power commands. Braking control system takes the braking pedal position and vehicle speed as inputs to decide the braking forces generated by friction braking and regenerative braking.

2.3 Vehicle Dynamics

This section begins with the establishment of three coordinate systems, these three coordinate systems will be used throughout this thesis. Afterwards, the vehicle longitudinal dynamics are processed based on the forces acting about the vehicle, and the specific value

or the expression of each force is determined by the dynamic equations of each component model[7]. At last, a Vehicle Body 3DOF Single Track block is used to calculate the vehicle lateral dynamics.

2.3.1 Coordinate system

Before calculating vehicle dynamics, two coordinate systems were defined according to the right-hand rule. The first coordinate system is an Earth-Fixed Inertial Coordinate System E with origin O_E, which is shown in Figure 2.4. The X_E axis is in the forward direction of the vehicle, and the Y_E axis extends to the right of the vehicle, both X_E and Y_E are parallel to the ground plane. For Z_E axis, the positive Z_E axis points downward.

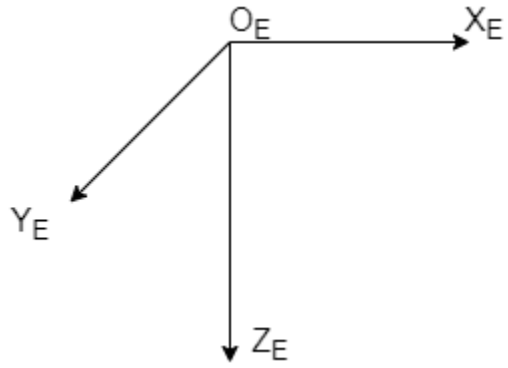

Figure 2.4: Earth-Fixed Inertial Coordinate System

The second coordinate system is the vehicle coordinate system shown in Figure 2.5. The system is a vehicle-fixed frame, with its origin at the vehicle's center of mass, and the $X_V, Y_V \; and \; Z_V$ axes correspond to the vehicle longitudinal, lateral, and vertical direction, respectively. Afterwards, the derivation of the vehicle dynamics needs to be conducted.

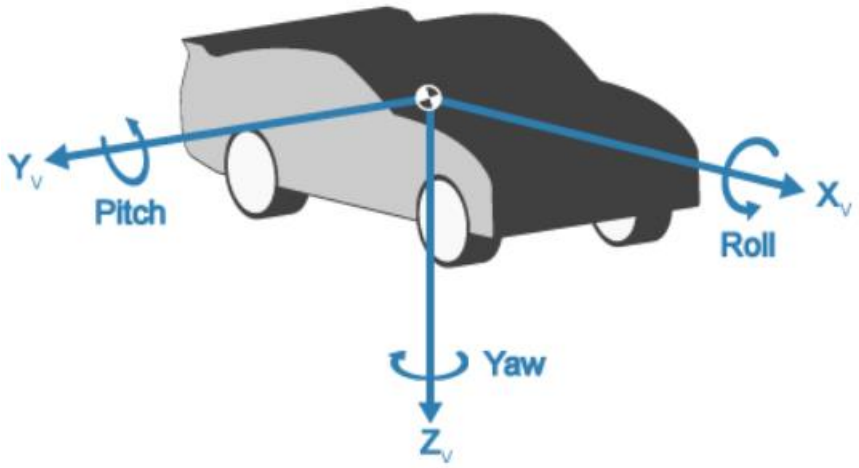

Figure 2.5: Vehicle Coordinate System[8]

2.3.2 Longitudinal Model

In this section, the vehicle longitudinal dynamics are processed based on the assumptions made on the three degrees-of-freedom (3DOF) single-track vehicle body. In order to gain a simple model for simulation, a longitudinal vehicle model is developed first.

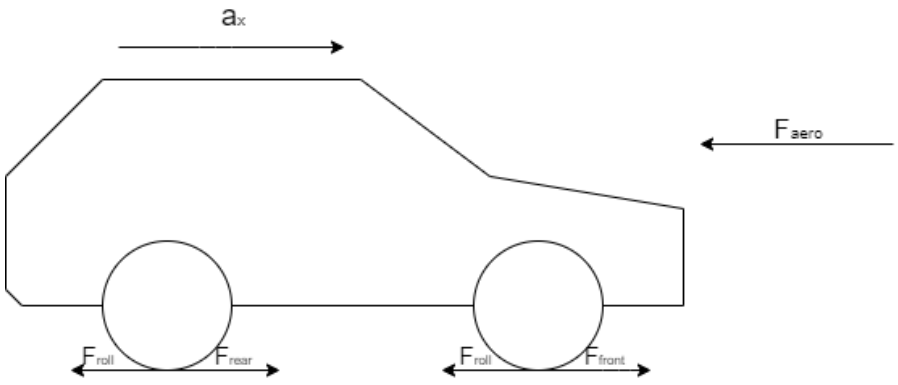

Figure 2.6: Longitudinal force balance for single track model

Figure 2.6 shows all the longitudinal forces and equations of motion in longitudinal direction can be derived from Newton's law; the equations are given as:

$$F_x = F_{front} + F_{rear} - F_{roll} - F_{aero} \tag{2.1}$$

$$F_{roll} = \mu_r N \tag{2.2}$$

$$F_{aero} = \frac{1}{2}\rho C_f S v_x^2 \tag{2.3}$$

$$F_x = m a_x \tag{2.4}$$

where m is the vehicle mass, a_x is the vehicle acceleration in the longitudinal direction. F_x denotes the total force applied on the vehicle in the longitudinal direction, F_{front} and F_{rear} are traction or braking forces on the front and rear wheels respectively. F_{roll} is the rolling friction between the vehicle's tires and the road surface, and μ_r represents the rolling friction coefficient. F_{aero} is the aerodynamic drag, ρ is the air density, C_f is the drag coefficient, S is the reference area and v_x is vehicle longitudinal velocity.

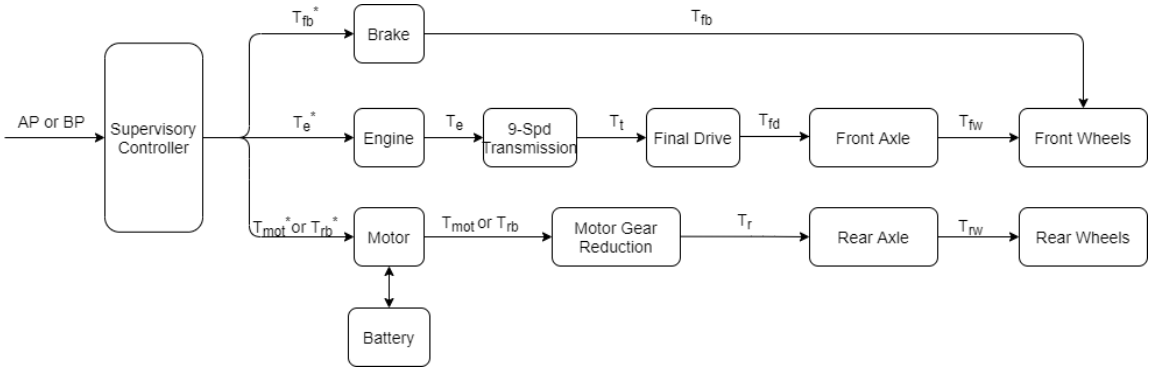

Figure 2.7: Longitudinal Model Block Diagram

As shown in Figure 2.7, a longitudinal model block diagram is built to present the power flow and corresponding torques. When the acceleration pedal is pressed, the propulsion control system in the supervisory controller determines the engine torque command T_e^* and the motor torque command T_{mot}^* based on the ECMS. Considering all the components need to take a transient process to reach the command value, a first-order

system is added to each component to simulate the transient process. Afterwards, the actual engine torque T_e will be transmitted by the transmission unit to the final drive, and the torque from the transmission T_t is equal to T_e times the transmission gear ratio η_t. At last, to get the torque from the final drive to the front axle T_{fd}, T_t is multiplied by a fixed axle ratio η_d. Thus, the final torque applied on the front wheels is given by:

$$T_{fw} = T_{fd} = T_t \eta_t = T_e \eta_d \eta_t \tag{2.5}$$

Similarly, the actual motor torque T_{mot} is transmitted by a motor gear reduction to the rear axle, and the torque from the reduction T_r is equal to T_{mot} times a fixed axle ratio η_r. The final torque applied on the rear wheels is given by:

$$T_{rw} = T_r = T_{mot} \eta_r \tag{2.6}$$

When the brake pedal is pressed, the braking control system in the supervisory controller decides the friction braking torque command T_{fb}^* the regenerative braking torque command T_{rb}^*. The actual friction braking torque T_{fb} is directly transmitted to the front wheels. In this process, the motor performs regenerative braking function instead of the driving function, thus the value of actual regenerative braking torque T_{rb} is equal to T_{mot}, but in the opposite direction. The final torque applied on the rear wheels T_{rw} is also transmitted through the gear reduction, its expression is the same as equation 2.6.

To calculate the forces applied on the wheels, the details of the front and rear wheel models are shown in Figure 2.8.

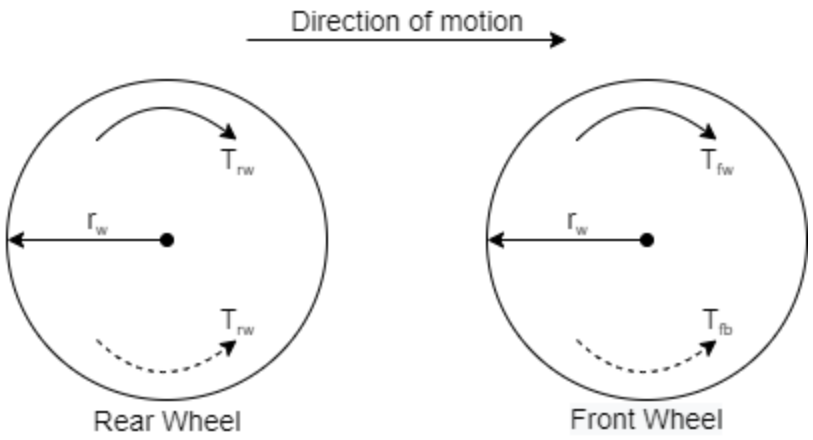

Figure 2.8: Wheel torque equilibrium

The solid curves represent the propulsion torques applied on the wheels, and the dashed curves represent the braking torques and r_w is the wheel radius. For the front wheels, the longitudinal force is given by:

$$F_{front} = \frac{T_{fw} - T_{fb}}{r_w} \tag{2.7}$$

For the rear wheels, the longitudinal force is given by:

$$F_{rear} = \frac{T_{rw}}{r_w} \tag{2.8}$$

Following the above steps, vehicle longitudinal acceleration can be calculated, and then vehicle longitudinal velocity is given by:

$$v_x = \int_{t_0}^{t_f} a_x \; dt \tag{2.9}$$

2.3.3 Lateral Model

In this section, the vehicle dynamics are processed based on the assumptions made on the three degrees-of-freedom (3DOF) single-track vehicle body model. The following lateral dynamics equations are obtained based on Newton's law.

$$\ddot{y} = -\dot{x}r + \frac{F_{y,front} + F_{y,rear} + F_{y,ext}}{m} \tag{2.10}$$

$$\dot{\omega} = \frac{l_f F_{y,front} - l_r F_{y,rear} + M_{z,ext}}{I_{zz}} \tag{2.11}$$

$$\omega = \dot{\theta} \tag{2.12}$$

where $\ddot{y}$ denotes vehicle lateral velocity, ω denotes vehicle angular velocity, and θ denotes rotation of vehicle-fixed frame about earth-fixed z-axis. $F_{y,front}$ and $F_{y,rear}$ are lateral forces applied to front and rear wheels, $F_{y,ext}$ denotes the external force applied to the center of mass, along the vehicle-fixed y-axis, and $M_{z,ext}$ denotes the external moment about the vehicle-fixed z-axis. l_f and l_r are longitudinal distances from the center of mass to the front and rear axles, respectively.

In order to calculate the lateral dynamics, a Vehicle Body 3DOF Single Track block is used. The block takes the front wheels steering angle and vehicle velocity in the longitudinal direction as inputs to calculate vehicle lateral and yaw motion. Vehicle body mass and aerodynamic drag between the axles due to acceleration in longitudinal direction and steering angle are also accounted by the block.

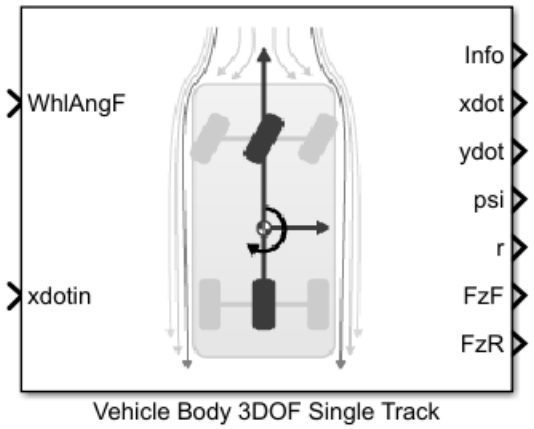

Figure 2.9: Vehicle Body 3DOF Single Track Block

The block takes the front wheels steering angle and vehicle velocity in longitudinal direction as inputs to calculate vehicle lateral and yaw motion. Vehicle body mass and aerodynamic drag between the axles due to acceleration in longitudinal direction and steering angle are also accounted by the block.

2.4 Whole Vehicle Model

The whole vehicle system is modeled in Simulink environment by combining all the component models mentioned above. The top level of the Simulink model is represented in Figure 2.10.

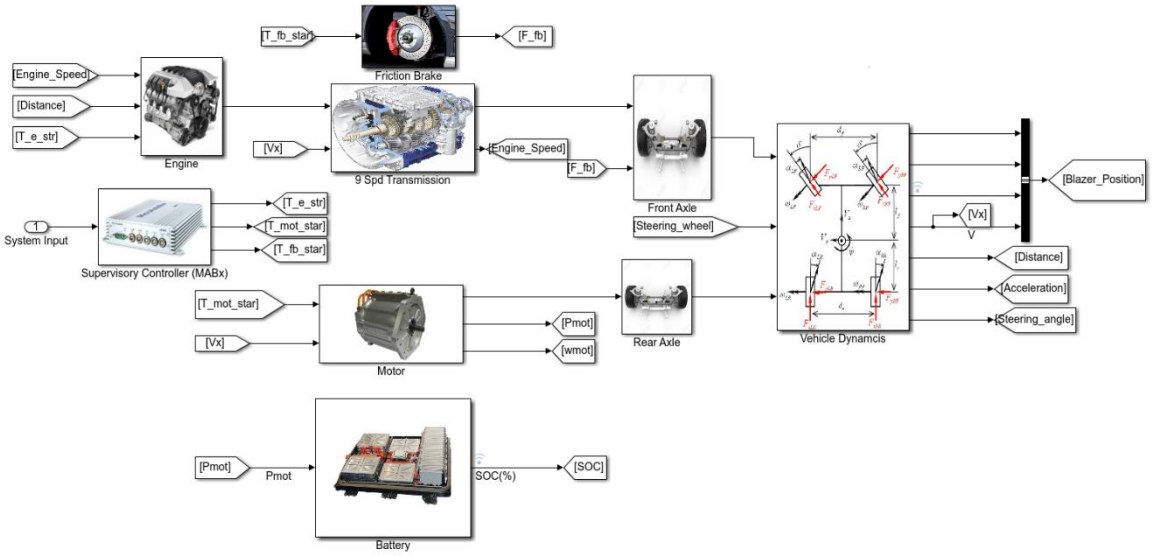

Figure 2.10: Vehicle Simulink Model

Chapter 3 Development of Lane Keep Assist System

The lane keep assist (LKA) system is required to have the ability to provide automatic steering to help the vehicle track the reference path with small tracking errors. The first step in the development of the LKA system is to select a suitable control strategy, two strategies are proposed, the first one is the proportional-integral-derivative (PID) controller and the second one is the model predictive control (MPC) controller. Compared with PID controller, the MPC controller can handle multi-input multi-output (MIMO) systems that have interactions between their inputs and outputs, at the same time, it can also handle constraints to limit output results. Thus, MPC is selected as the control strategy for the system. The following sections present the development of the MPC controller, including its design process and tuning process.

3.1 Overview of MPC controller

Model Predictive Control (MPC) is an advanced method of process control that is used to control a process while satisfying a set of constraints. It is also known as Dynamical Matrix Control (DMC) or Generalized Predictive Control (GPC). It has been widely used in different industries, such as chemical plants, power system balancing models, and power electronics. The MPC controller is based on iterative, finite-horizon optimization of a plant model[9][10]. The plant model is used to minimize the difference between a system's predicted responses and the target responses. The model is intended to represent the behavior of a complex dynamical system, and it is generally used in the form of state-space

representation, which is derived from differential equations of a linear system or linearized differential equations of a nonlinear system about equilibrium points. The finite-horizon optimization problem is usually solved by linear programming (LP) or quadratic programming (QP), thus a linear or quadratic objective function is constructed and optimized, subject to linear equality or linear inequality constraints. The theory behind the MPC controller is described in Figure 3.1.

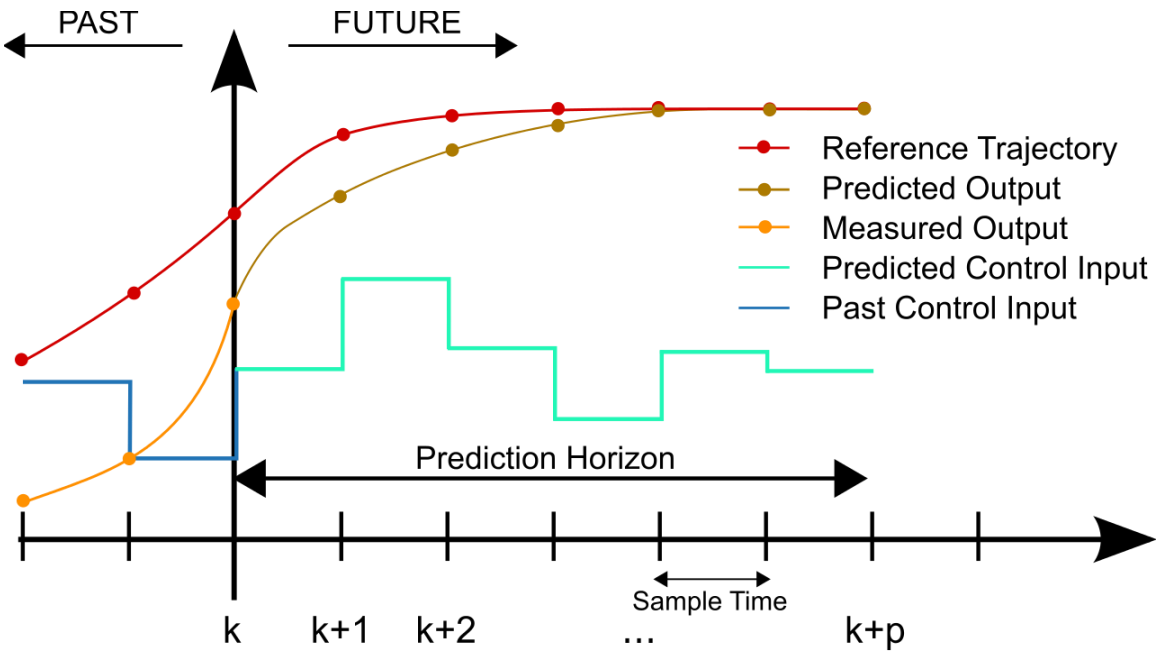

Figure 3.1: Theory Behind MPC Controller[10]

At each control instant k, the current plant state is sampled, and the MPC controller computes future outputs based on the current states and inputs. Afterwards, a desired sequence of control inputs is determined by minimizing the objective function. Only the first pair of the control inputs is implemented to the system, then the plant state is sampled again and the calculations are repeated starting from the new state, yielding a new control sequence and new predicted state path.

3.2 Design of MPC Controller

As shown in Figure 3.2, the block diagram shows the basic structure of the MPC controller. The MPC controller is composed of a predictor and an optimizer. The predictor contains a state-space model which is built based on the vehicle's kinematic model, it predicts the future states and outputs of the vehicle with current states and inputs. The states and inputs are represented up to prediction horizon N_p and control horizon N_c, respectively. The optimizer will generate an objective function to find the optimal solution for each step.

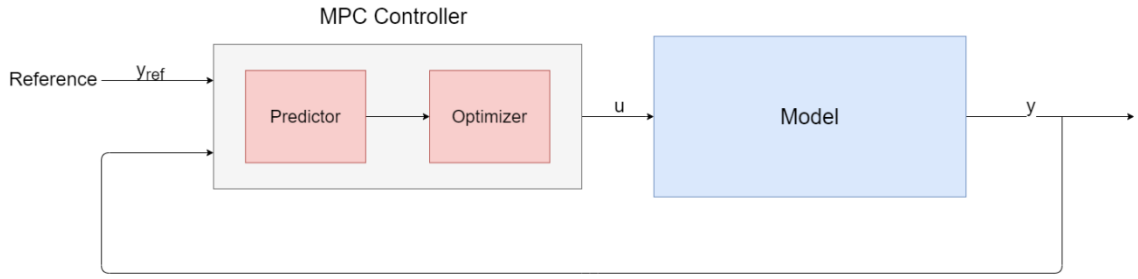

Figure 3.2: Structure of MPC Controller

3.2.2 Predictor

For the predictor, a state-space model is built for the vehicle's kinematics to describe the physical interpretation of the states X, inputs U and outputs Y. A kinematic bicycle model with a rear axle reference point is used to build the plant model, and its visual representation is shown in Figure 3.3. The state of the model is defined by $X = [x \quad y \quad \theta]^T$. x and y represent the coordinates of the center of the front axle in the world coordinate system. θ denotes the heading angle of the vehicle, which is the rotation angle of the vehicle-fixed frame about the world-fixed Z-axis. The input of the model is defined

by $U = [v \quad \delta]^T$, where v is the vehicle speed and δ is the front wheel steering angle. The output of the model is also defined by $Y = [x \quad y \quad \theta]^T$.

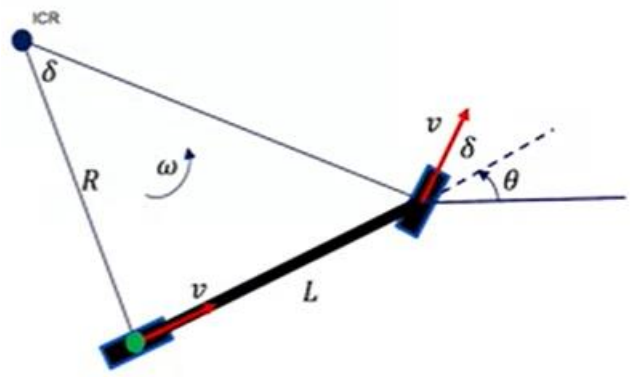

Figure 3.3: Bicycle Model Visual Representation

The nonlinear continuous time equations that describe the kinematic bicycle model represented in Figure 3.3 are given by:

$$\dot{X} = f(X, U) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} vcos(\theta) \\ vsin(\theta) \\ \frac{vtan\delta}{l} \end{bmatrix} \tag{3.1}$$

$$Y = h(X, U) = X \tag{3.2}$$

where l represents the distance from the center of the front axle to the center of the rear axle.

The first step in modeling the state observer is to linearize the continuous equations to get the following continuous state space model:

$$\dot{X} = AX + BU \tag{3.3}$$

$$Y = CX + DU \tag{3.4}$$

where:

$$A = \frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial v\cos(\theta)}{\partial x} & \frac{\partial v\cos(\theta)}{\partial y} & \frac{\partial v\cos(\theta)}{\partial \theta} \\ \frac{\partial v\sin(\theta)}{\partial x} & \frac{\partial v\sin(\theta)}{\partial y} & \frac{\partial v\sin(\theta)}{\partial \theta} \\ \frac{\partial}{\partial x}\frac{v\tan\delta}{l} & \frac{\partial}{\partial y}\frac{v\tan\delta}{l} & \frac{\partial}{\partial \theta}\frac{v\tan\delta}{l} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v\sin(\theta) \\ 0 & 0 & v\cos(\theta) \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial U} = \begin{bmatrix} \frac{\partial v\cos(\theta)}{\partial v} & \frac{\partial v\cos(\theta)}{\partial \delta} \\ \frac{\partial v\sin(\theta)}{\partial v} & \frac{\partial v\sin(\theta)}{\partial \delta} \\ \frac{\partial}{\partial v}\frac{v\tan\delta}{l} & \frac{\partial}{\partial \delta}\frac{v\tan\delta}{l} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ \frac{\tan\delta}{l} & \frac{v\sec^2(\delta)}{l} \end{bmatrix}$$

$$C = \frac{\partial h}{\partial X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = \frac{\partial h}{\partial U} = 0$$

The next step is to discretize the continuous system with sampling period T_s. The continuous model can be approximated as

$$X(t + T_s) = X(t) + \dot{X}(t)T_s \tag{3.5}$$

$$= X(t) + \big(AX(t) + BU(t)\big)T_s$$

$$Y(t) = CX(t) + DU(t) \tag{3.6}$$

Compute $X(t)$ and $Y(t)$ at $t = kT_s$ for $k \in N_p$, then the formula can be written as:

$$X\big((k+1)T_S\big) = X(kT_S) + \big(AX(kT_s) + BU(kT_s)\big)T_s \tag{3.7}$$

$$= (I + AT_s)X(kT_s) + BU(kT_s)T_S$$

$$Y(kT_s) = CX(kT_s) + DU(kT_s) \tag{3.8}$$

Set $X_k = X(kT_s), X_{k+1} = X\big((k+1)T_S\big)$, and $Y_k = Y(kT_s)$, then the discrete time state variable model can be described by:

$$X_{k+1} = (I + AT_s)X_k + BT_sU_k \tag{3.9}$$

$$= A_dX_k + B_dU_k$$

$$Y_k = C_dX_k + D_dX_k \tag{3.10}$$

where

$$A_d = I + AT_s = \begin{bmatrix} 1 & 0 & -vsin(\theta)T_s \\ 0 & 1 & vcos(\theta)T_s \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_d = BT_s = \begin{bmatrix} \cos(\theta)\,T_s & 0 \\ \sin(\theta)\,T_s & 0 \\ \dfrac{tan\delta}{l}T_s & \dfrac{vsec^2(\delta)}{l}T_s \end{bmatrix}$$

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

$$D_d = D = 0$$

To predict the states for the next N_p steps ahead, the sequence state vectors is predicted as follows:

$$X_{k+2} = A_dX_{k+1} + B_dU_{k+1} \tag{3.11}$$

$$= A_d(A_dX_k + B_dU_k) + B_dU_{k+1}$$

$$= A_d^2X_k + A_dB_dU_k + B_dU_{k+1}$$

$$X_{k+3} = A_dX_{k+2} + B_dU_{k+2}$$

$$= A_d(A_d^2X_k + A_dB_dU_k + B_dU_{k+1}) + B_dU_{k+2}$$

$$= A_d^3X_k + A_d^2B_dU_k + A_dB_dU_{k+1} + B_dU_{k+2}$$

$$\vdots$$

$$X_{k+N} = A_d^{N_p} X_k + \sum_{j=0}^{N_p-1} A_d^{N_p-1-j} B_d U_{k+j}$$

Rewrite the equation in state-space form

$$X_{vec} = \hat{A}_d X_k + \hat{B}_d U_{vec} \tag{3.12}$$

where state vector and control input vector are given by:

$$X_{vec} = \begin{bmatrix} X_{k+1} \\ X_{k+2} \\ \vdots \\ X_{k+N_p-1} \\ X_{k+N_p} \end{bmatrix}; \quad U_{vec} = \begin{bmatrix} U_k \\ U_{k+1} \\ \vdots \\ U_{k+N_p-2} \\ U_{k+N_p-1} \end{bmatrix}$$

and the model matrices are:

$$\hat{A}_d = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{N_p-1} \\ A_d^{N_p} \end{bmatrix}_{4N_p \times 4} \quad ; \quad \hat{B}_d = \begin{bmatrix} B_d & 0 & \cdots & 0 & 0 \\ A_d B_d & B_d & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ A_d^{N_p-2} B_d & A_d^{N_p-3} B_d & \cdots & B & 0 \\ A_d^{N_p-1} B_d & A_d^{N_p-2} B_d & \cdots & AB & B \end{bmatrix}_{4N_p \times 2N_p}$$

The output vector is given by:

$$Y_{vec} = \hat{C}_d X_{vec} = G X_k + H U_{vec} \tag{3.13}$$

where

$$Y_{vec} = \begin{bmatrix} Y_{k+1} \\ Y_{k+2} \\ \vdots \\ Y_{k+N_p-1} \\ Y_{k+N_p} \end{bmatrix}, \hat{C}_d = \begin{bmatrix} C \\ C \\ \vdots \\ C \\ C \end{bmatrix}, G = \hat{C}_d \hat{A}_d, H = \hat{C}_d \hat{B}_d$$

3.2.3 Optimizer

The main task of the optimizer is to find the optimal solution for the control inputs U_{vec} in the neighborhood of the current position to some finite horizon. To solve this optimization problem, an objective function is defined to represent the performance in this neighborhood, and then the optimal solution is solved by searching a feasible solution that minimizes the cost function. A general objective function for MPC (MPC formulation) is given by:

$$J = \sum_{n=k+1}^{k+N_p} \hat{Y}_n^T P \hat{Y}_n + \sum_{n=k}^{k+N_c-1} \widehat{U}_n^T Q \widehat{U}_n \tag{3.14}$$

where

$$\hat{Y}_n = Y_n - Y_{ref,n}$$

$$\widehat{U}_n = U_n - U_{ref,n}$$

$$\Delta U_n = U_n - U_{n-1}$$

N_p and N_c is the control horizon, respectively. $Q = diag\{w_x, w_y, w_\theta\}$ and $R = diag\{w_v, w_\delta\}$ are weighting matrices. $w_x, w_y, w_\theta, w_v,$ and w_δ are weighting factors corresponding to output and input variables.

Rewrite the function into state-space representation:

$$J = \hat{Y}^T \hat{Q} \hat{Y} + \widehat{U}^T \hat{R} \widehat{U} \tag{3.15}$$

where

$$\hat{Y} = Y_{vec} - Y_{ref} = \begin{bmatrix} Y_{k+1} \\ Y_{k+2} \\ \vdots \\ Y_{k+N_p-1} \\ Y_{k+N_p} \end{bmatrix} - \begin{bmatrix} Y_{ref,k+1} \\ Y_{ref,k+2} \\ \vdots \\ Y_{ref,k+N-1} \\ Y_{ref,k+N} \end{bmatrix}$$

$$\hat{U} = U_{vec} - U_{ref} = \begin{bmatrix} U_{k+1} \\ U_{k+2} \\ \vdots \\ U_{k+N_p-1} \\ U_{k+N_p} \end{bmatrix} - \begin{bmatrix} U_{ref,k} \\ U_{ref,k+1} \\ \vdots \\ U_{ref,k+N-2} \\ U_{ref,k+N-1} \end{bmatrix}$$

$$\hat{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & Q \end{bmatrix}_{4N_p \times 4N_p}$$

$$\hat{R} = \begin{bmatrix} R & 0 & \cdots & 0 & 0 \\ 0 & R & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & R & 0 \\ 0 & 0 & \cdots & 0 & R \end{bmatrix}_{2N_c \times 2N_c}$$

Rewrite the output vector Y_{vec} into the following form:

$$\begin{aligned} Y_{vec} &= GX_k + HU_{ref} + H\left(U_{vec} - U_{ref}\right) \\ &= \left(GX_k + HU_{ref}\right) + H\hat{U} \end{aligned} \tag{3.16}$$

Let $Y_{opt} = GX_k + HU_{ref}$ and $U_{opt} = H\hat{U}$, then

$$Y_{vec} = Y_{opt} + U_{opt} \tag{3.17}$$

Substitute $\hat{Y} = Y_{vec} - Y_{ref}$, $\hat{U} = U_{vec} - U_{ref}$ and $Y_{vec} = Y_{opt} + U_{opt}$ into the objective function, the objective function becomes:

$$J = \left(Y_{vec} - Y_{ref}\right)^T \hat{Q}\left(Y_{vec} - Y_{ref}\right) + \hat{U}^T \hat{R}\hat{U} \tag{3.18}$$

$$= \left(Y_{opt} - Y_{ref} + U_{opt}\right)^T \hat{Q}\left(Y_{opt} - Y_{ref} + U_{opt}\right) + \hat{U}^T \hat{R} \hat{U}$$

$$= \left(Y_{opt} - Y_{ref}\right)^T \hat{Q}\left(Y_{opt} - Y_{ref}\right) + 2\left(Y_{opt} - Y_{ref}\right)^T \hat{Q} U_{opt}$$

$$+ U_{opt}^T \hat{Q} U_{opt} + \hat{U}^T \hat{R} \hat{U}$$

Then the objective function $J(X, U)$ is minimized using control input $\hat{U}$ as the free variable. The term $\left(Y_{opt} - Y_{ref}\right)^T \hat{\mathrm{Q}}\left(Y_{opt} - Y_{ref}\right)$ is not affected by the optimization, thus a simplified objective function is given by:

$$\hat{J} = \left(Y_{opt} - Y_{ref}\right)^T \hat{Q} U_{opt} + \frac{1}{2} U_{opt}^T \hat{\mathrm{Q}} U_{opt} + \frac{1}{2} \hat{U}^T \hat{R} \hat{U} \tag{3.19}$$

Substitute $U_{opt} = H\hat{U}$ into the simplified objective function, and the objective function can be converted to the quadratic form:

$$\hat{J} = \frac{1}{2} \hat{U}^T h \hat{U} + f^T \hat{\mathrm{U}} \tag{3.20}$$

where

$$h = H^T \hat{Q} H + \hat{\mathrm{R}}$$

$$f^T = \left(Y_{opt} - Y_{ref}\right)^T \hat{Q} \mathrm{H}$$

To solve the optimization problem, some hard and soft constraints need to be created on the system variables. The hard constraints for this problem are lower and upper limits on the inputs, which are given in Table 3.1.

Table 3.1: Constraints of Control Inputs

Control Inputs	Lower Limit	Upper Limit
Vehicle Speed	0	Reference Speed
Steering Angle	-33°	33°

For the soft constraints, the upper constraint is given by:

$$Y_{vec} \leq Y_{max,vec} \tag{3.21}$$

where

$$Y_{vec} = Y_{opt} + U_{opt}$$

Then, the constraints on U_{opt} is given by:

$$U_{opt} \leq Y_{max,vec} - Y_{opt} \tag{3.22}$$

Substitute $U_{opt} = H\widehat{U}$ into equation 3.22, then

$$H\widehat{U} \leq Y_{max,vec} - Y_{opt} \tag{3.23}$$

Similarly, the lower constraint is given by:

$$-H\widehat{U} \leq -Y_{min,vec} + Y_{opt} \tag{3.24}$$

Substitute $U_{opt} = \hat{B}_d \Delta U$ into the two constraints, the constraints on ΔU can be obtained:

$$L_x \Delta U \leq b_x \tag{3.25}$$

where

$$L_x = \begin{pmatrix} \hat{C}\hat{B} \\ -\hat{C}\hat{B} \end{pmatrix}; \quad b_x = \begin{pmatrix} b_{x1} \\ b_{x2} \end{pmatrix}$$

$$b_{x1} = Y_{max,vec} - \hat{C}X_{opt}; \quad b_{x2} = -Y_{min,vec} + \hat{C}X_{opt}$$

Then the *quadprog* function is used to solve the optimization problem, and the first pair of solutions will be used as the control inputs of the vehicle model.

3.3 Tune MPC Parameters

MPC algorithms incorporate several parameters that give enormous flexibility to the configuration of the control system, a successful implementation of MPC requires parameters appropriately tuned to achieve desired control performance. Basically, five main parameters that need to be tuned in the MPC controller include the sampling time T_s, the prediction horizon N_p, the control horizon N_c, and the MPC weighting matrices Q and R. The last two matrices are diagonal matrices with output and input weighting factors on their main diagonals. This section focuses on the selection and tuning process of each parameter.

3.3.1 Sampling Time

The sampling time T_s is used for the discretization of the system. Generally, a smaller sampling time T_s results in higher controller accuracy, but at the same time, the computational time also increases. Thus, the optimal choice for T_s is a balance of control accuracy and computational time. For automotive control, set the value of sampling time $T_s = 0.1s$ first, and then hold it constant while tuning other controller parameters. If the controller performance after the whole tuning process is still poor, the sampling time T_s needs to be revised, and other parameters also need to be re-tuned.

3.3.2 Prediction Horizon

The prediction horizon N_p is the number of output predictions that are used in the optimization calculation at each control interval. In general, a short prediction horizon may lead to an unstable controller, a long prediction horizon improves the control performance,

however, it also increases the computational time[11]. To find the appropriate value of the prediction horizon, the reference vehicle speed is set as 60 km/h, and a step signal with an amplitude of 5 meters is set as the target lateral trajectory of the vehicle, as shown by the dashed line in Figure 3.4. The default value of the prediction horizon is $N_p = 1$, hold other parameters constant, and then vary the value of N_p until further increases have a minor impact on the comprehensive performance of the controller. The solid lines in Figure 3.4 show the output using different prediction horizons, it can be noticed that the increase of the value of the prediction horizon reduces the overshoot of the output response, but it also leads to more frequent oscillation. Apart from that, continue to increase the value of N_p, and the system needs to spend a longer time to achieve the steady state. Therefore, $N_p = 1$ is chosen as the value of the prediction horizon, considering faster response, its output response, and corresponding control input are shown in Figure 3.5 and Figure 3.6.

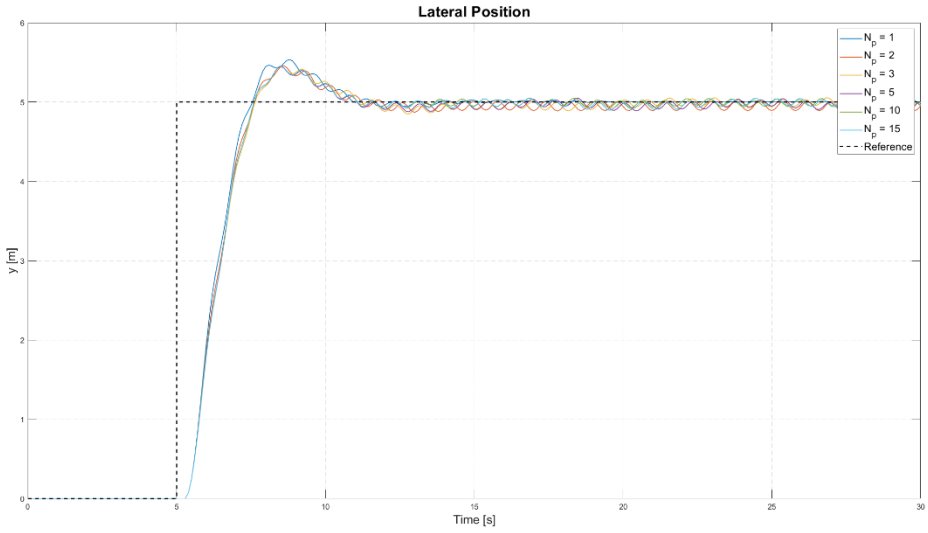

Figure 3.4: Vehicle Later Position with Different Prediction Horizon

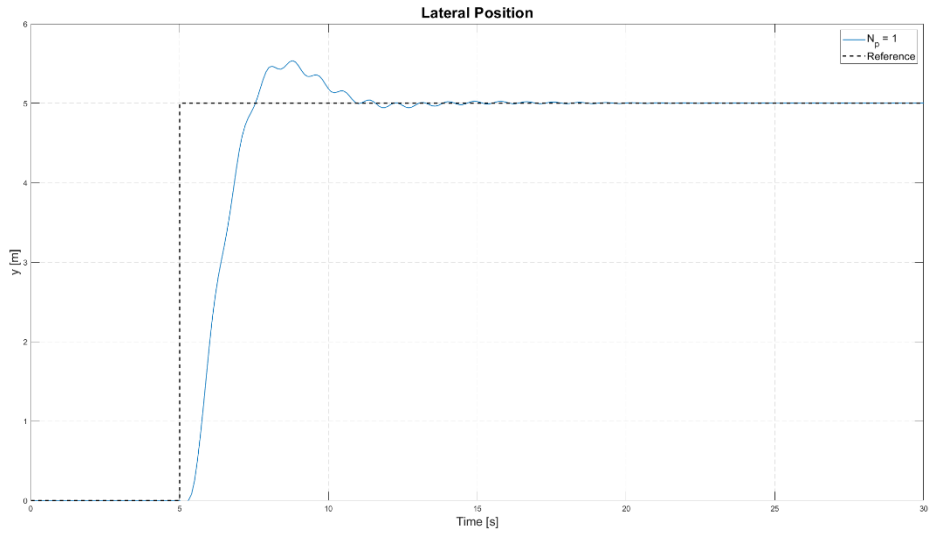

Figure 3.5: Vehicle Later Position with $N_p = 1$

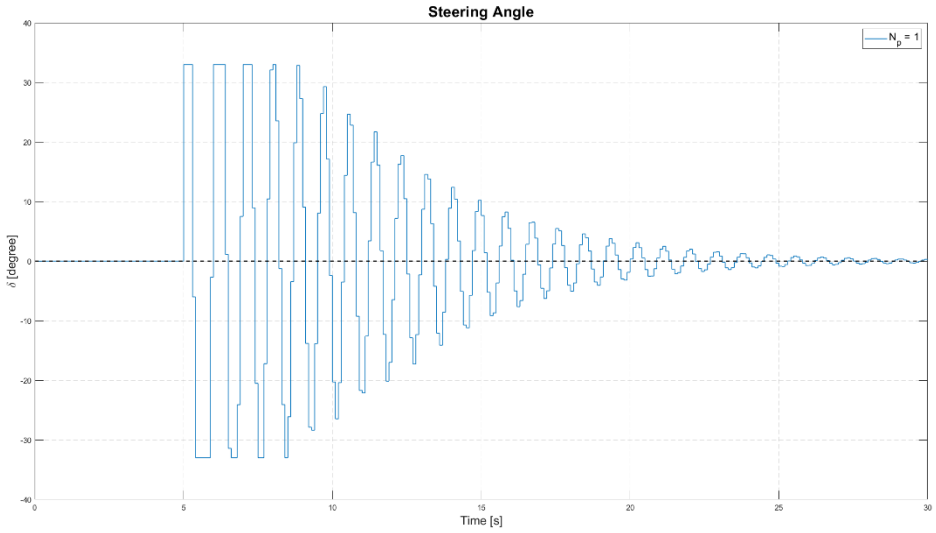

Figure 3.6: Vehicle Steering Angle with $N_p = 1$

3.3.3 Control Horizon

The control horizon N_c is the number of control inputs that are optimized at each control interval, it always falls between 1 and the prediction horizon N_p. In general, short control horizon promotes an internally stable controller, whereas a long control horizon

leads to more control actions and longer computational time[11]. To select a suitable control horizon, fix the prediction horizon and vary the control horizon N_c. Since the value of the prediction horizon is selected as $N_p = 1$, thus, the value of the prediction horizon is also selected as $N_c = 1$.

3.3.4 Weighting Matrices

The MPC objective function weights are composed of two weighting matrices, output weighting matrix Q and input weighting matrix R. The value of the output weighting matrix Q reflects the relevant importance of the variables. The output weighting matrix Q consists of four output weighting factors w_x, w_y and w_θ. These factors allow the output variables to be weighted according to their relevant importance. A higher value of output weighting factor represents a small tracking error desired[11][12]. If both outputs cannot reach the value of the desired outputs, the controller would care more about the tracking error for the higher weight output over the tracking error for the lower wight output.

Similarly, the input weight matrix R is composed of two input weighting factors w_a and w_δ. These two factors define the importance of the input variables. A higher value of input weighting factor penalizes excessive incremental control actions at the cost of output tracking error, and it also slows down the response. To tune the weighting factors, the prediction horizon and control horizon is fixed at $N_p = 1$ and $N_c = 1$, respectively. In order to adapt to different driving conditions, three sets of weighting factors are tuned according to different vehicle speeds, which are suitable for low, medium, and high speeds, and the three sets are tuned at a speed of 30km/h, 60km/h and 100km/h, respectively. The step signal with an amplitude of 5 meters is still set as the target lateral trajectory of the

vehicle. Considering offsets and deviations are common in the actual driving process, a tolerance band is used as a reference to tune the weighting factors. To tune the weighting factors, the overshoots of the system are required to keep within the tolerance band. The width of the tolerance band w_t is determined based on the traffic lane width w_l and vehicle's width w, which is given by:

$$w_t = w_l - w \tag{3.26}$$

In general, a traffic lane has a width of 3.05 meters (10 feet), and the width of Chevrolet Blazer is 1.96 meters (77 inches), thus the width of the tolerance band is 1.09 meters. Figure 3.7 illustrates a comparison between the vehicle's lateral position and heading angle with different reference speed, and the corresponding actual control actions are shown in Figure 3.8.

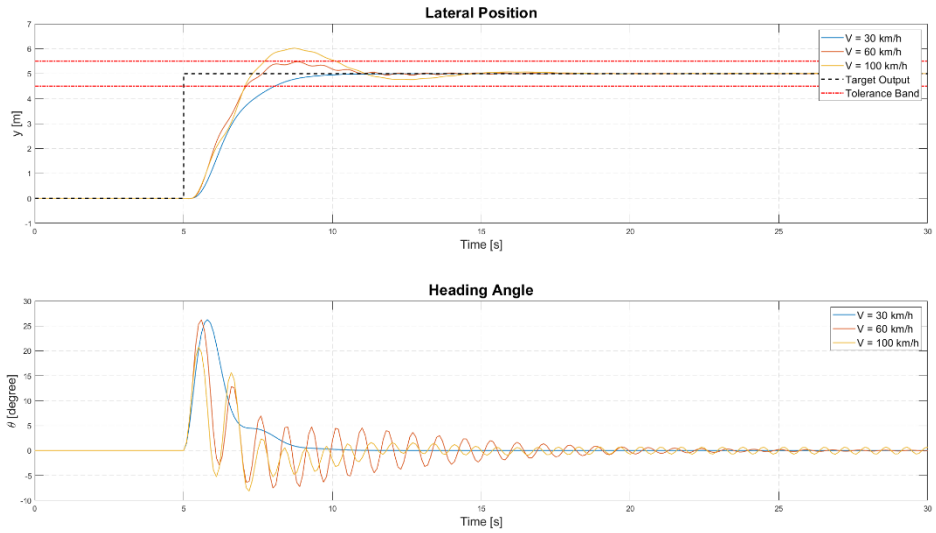

Figure 3.7: Output Responses with Different Speeds

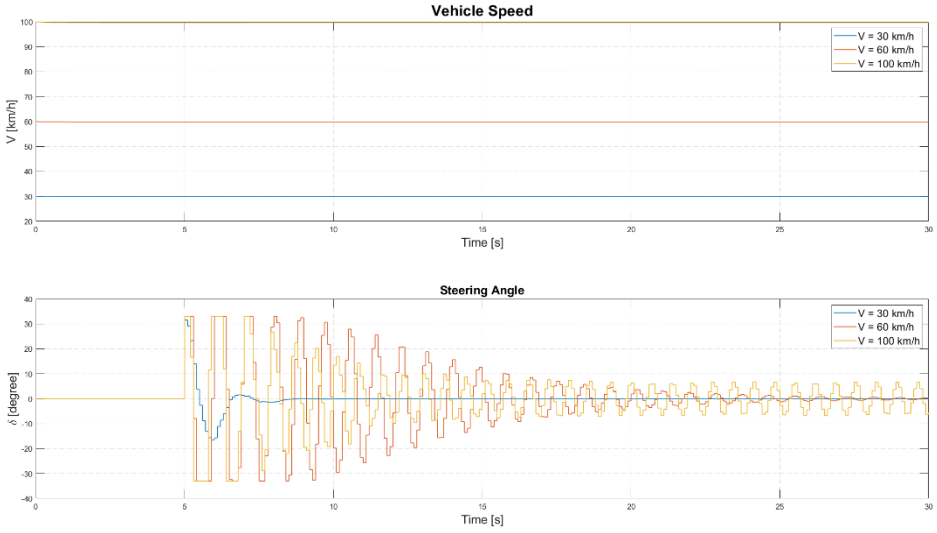

Figure 3.8: Control inputs with Different Speeds

It can be noticed that the larger overshoots and deviations can be observed when the vehicle travels at medium and high speeds, especially at high speed, the vehicle's overshoot in lateral position even exceeds the tolerance band. Apart from that, the actual heading angle fluctuates more severely. For all three sets of weighting factors, the value of the lateral position weighting factor needs to be increased to shorten the settling time, and the value of heading angle weighting factor also needs to be increased to prevent fluctuations of the heading angle. Besides, other weighting factors may also need to be tuned to improve the control performance. The final weighting factors for the three groups are given in Table 3.2.

Table 3.2: Weighting Factors at Different Speeds

Group	**Q**			**R**	
	w_x	w_y	w_θ	w_v	w_δ
Low Speed	1	1.2	1	1	1
Medium Speed	1	1.2	5	1	1
High Speed	1	1	40	1	1

Figure 3.9 and Figure 3.10 illustrate the system outputs and corresponding control actions after tuning weighting factors. It can be noticed that the overshoots are reduced, the actual lateral responses remain within the tolerance band. Apart from that, the fluctuations of the heading angle and the control actions of steering angle at medium and high speeds are largely reduced.

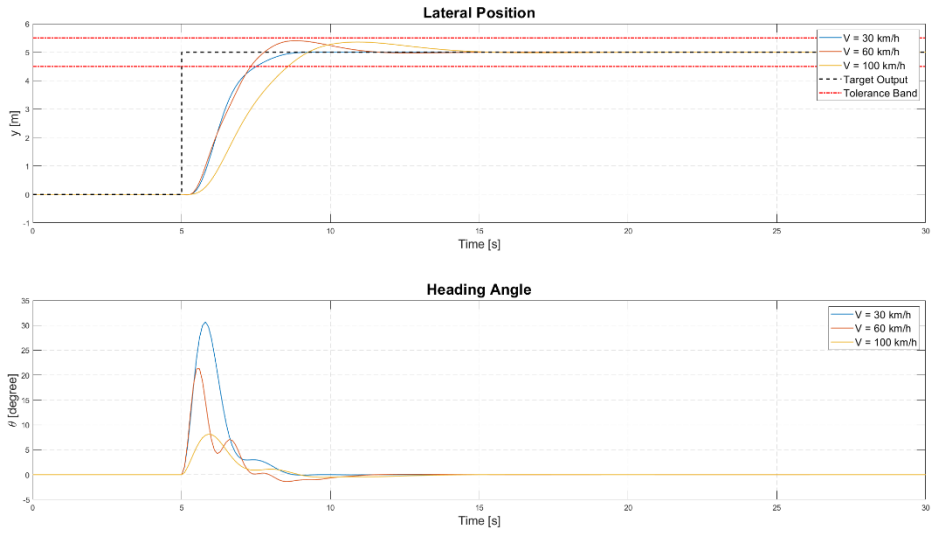

Figure 3.9: Output Responses after Tuning

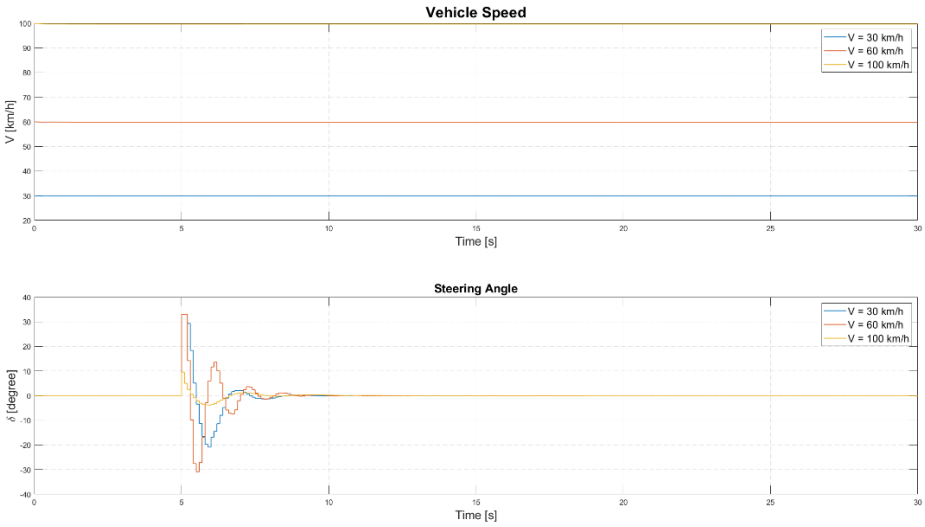

Figure 3.10: Control Inputs after Tuning

Chapter 4 Tests and Validations

This chapter focuses on the tests and validations of the designed LKA system. The vehicle model built in Chapter 2 is used as the simulation platform, and three test cases are conducted to investigate the general performance of the MPC based LKA system.

4.1 Lane Change Test

The first test case is a lane change test. The test aims to evaluate the lane changing performance of the vehicle at different speeds. In the test, the vehicle model is required to follow a reference path with a total length of 2100 meters. A lane changing request is sent to the supervisory controller every 300 meters, which is a step signal with an amplitude of 3.05 meters (10 foot), the width of a typical traffic lane. Apart from that, a Highway Fuel Economy Driving Schedule (HWFET) velocity drive cycle is used as the vehicle reference speed, and the speed tracking was achieved using a PID controller. To evaluate the controller's performance, the tolerance band used in the weighting factors tuning process used again as a reference. The vehicle's overshoots and undershoots in lateral position are required to keep within the tolerance band.

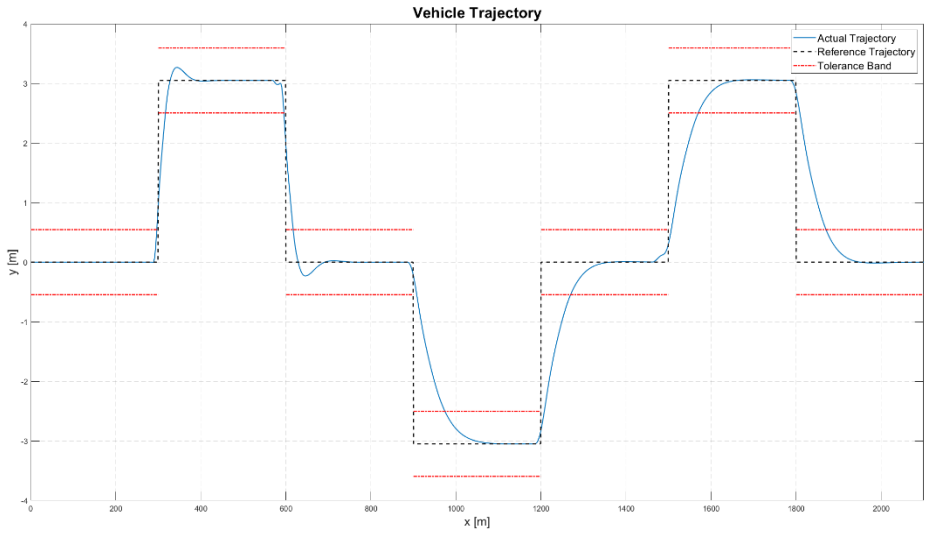

Figure 4.1: Vehicle Trajectory for Lane Change Test

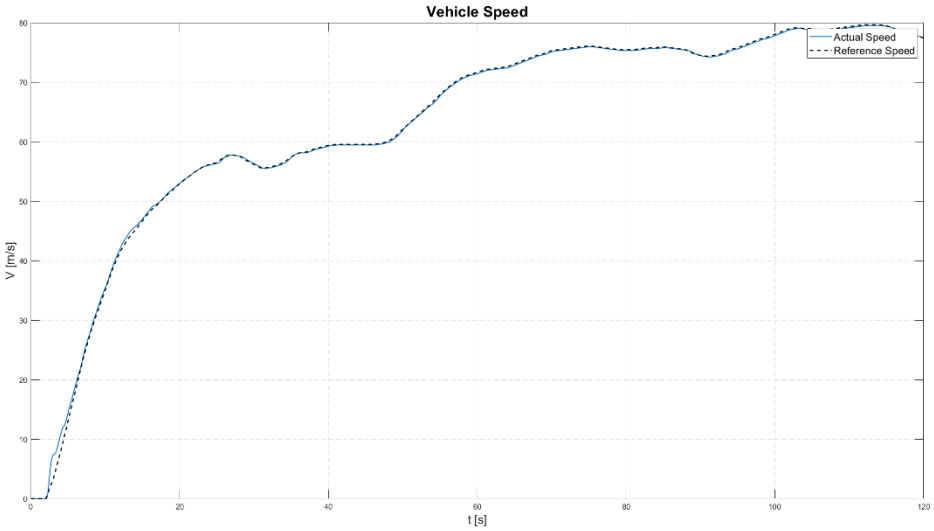

Figure 4.2: Vehicle Speed for Lane Change Test

Figure 4.1 illustrates the result with a graphical comparison between the vehicle actual trajectory and reference trajectory, and Figure 4.2 compares the vehicle's actual speed and reference speed. It can be noticed that the vehicle can complete the lane changing in a short distance at different speeds. Although at some speeds, small overshoots still exist, none of them exceeds the tolerance band, and the system achieves the steady state quickly.

4.2 Path Tracking Test at Different Speeds

In order to verify the controller's path tracking performance, a complex curve is designed as the reference path for this test. In the test, the vehicle model is required to track the reference path at low, medium, and high speeds. Figure 4.3 to Figure 4.5 show the simulation results of the path tracking at a speed of 40 km/h, 70 km/h, and 110 km/h, respectively. For each test, the vehicle actual trajectory is indicated by the blue solid line while the reference trajectory is indicated by a black dashed line.

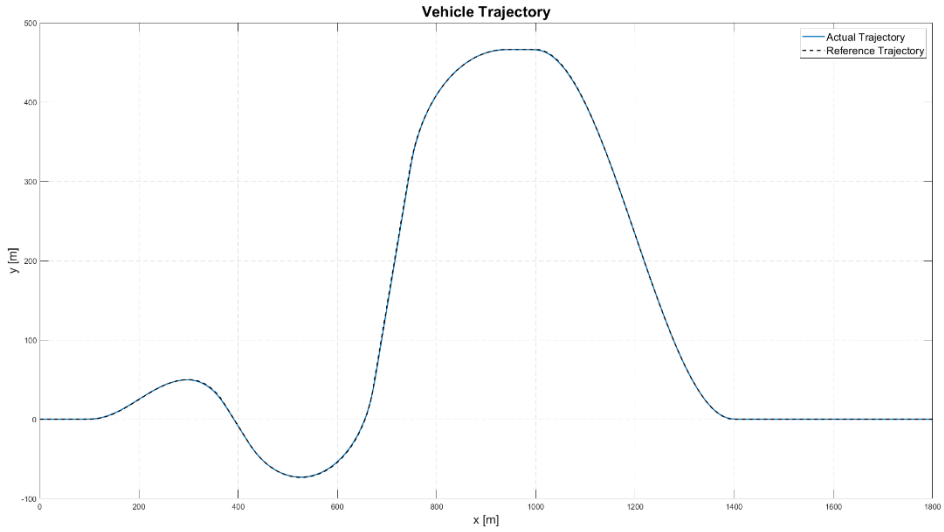

Figure 4.3: Path Tracking Test at 40km/h

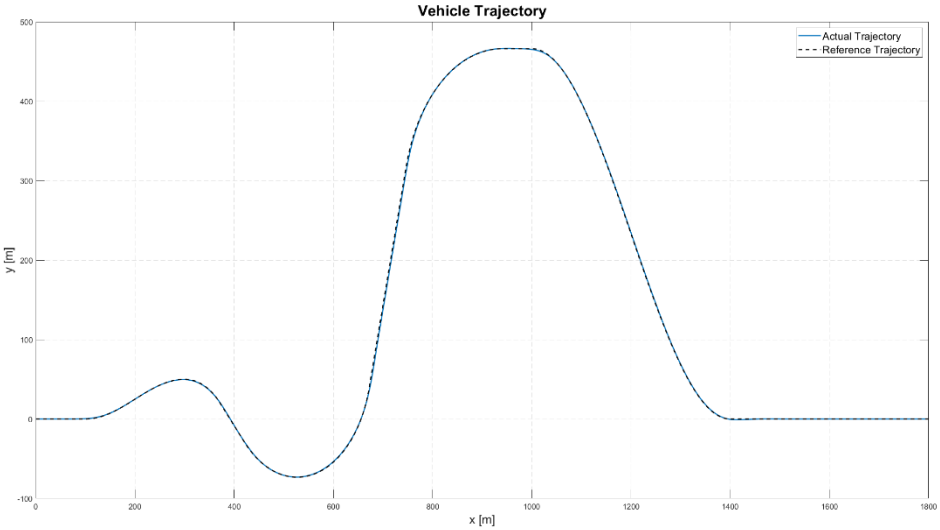

Figure 4.4: Path Tracking Test at 70km/h

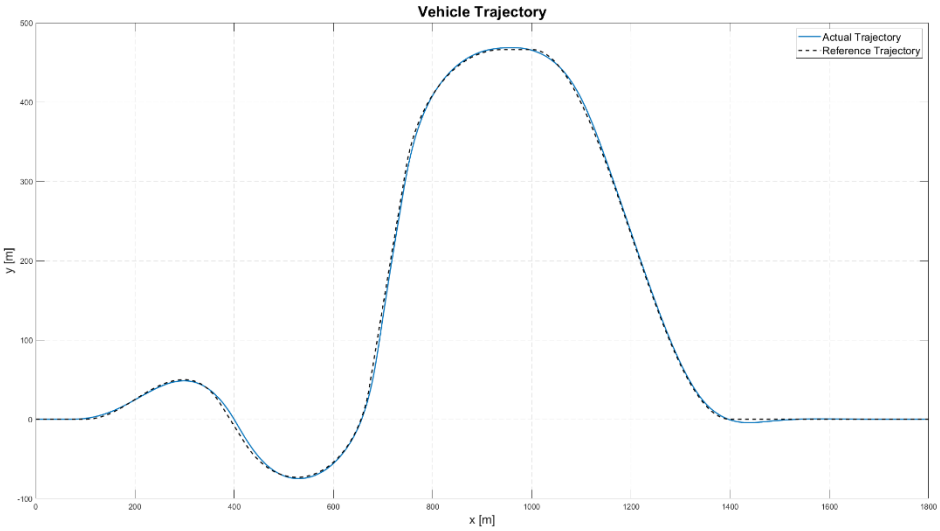

Figure 4.5: Path Tracking Test at 110km/h

From Figure 4.3 and Figure 4.4, it can be noticed that the MPC controller can track the reference path precisely when the vehicle travels at low and medium speeds. However, Figure 4.5 shows that when the vehicle travels at high speed, large deviations can be observed, especially near the position where the curvature changes greatly. One of the reasons that leads to the poor performance for high speed test is the mismatch between the

vehicle model and the reference path. In other words, when the vehicle travels at a high speed, the vehicle model is in lack of sufficient steering ability at some large turning angles. Thus, another test needs to be conducted to study this effect.

4.3 High Speed Path Tracking Test

In the previous test, it can be noticed that compared with low speed and medium speed, when the vehicle travels at high speeds, a larger tracking error can be noticed. In order to eliminate the effect of the reference path, a specific reference path is designed for high speed test, which is shown in Figure 4.6.

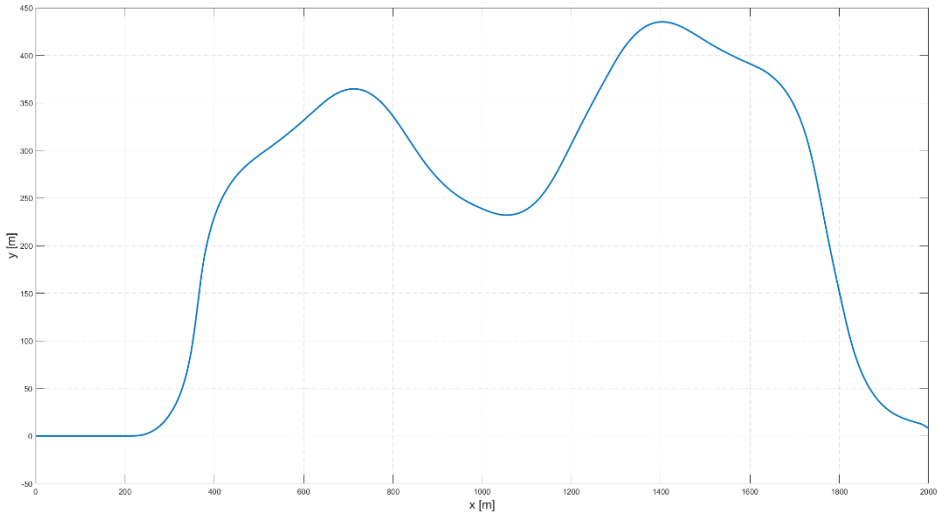

Figure 4.6: Reference Path

The path was designed by recording the vehicle actual positions at 110 km/h. The LKA system was temporarily disabled throughout the process, and the steering angle was changed manually. Then, the recorded positions are used as the reference path positions, reactivate the LKA system, the simulation results are shown in Figure 4.7.

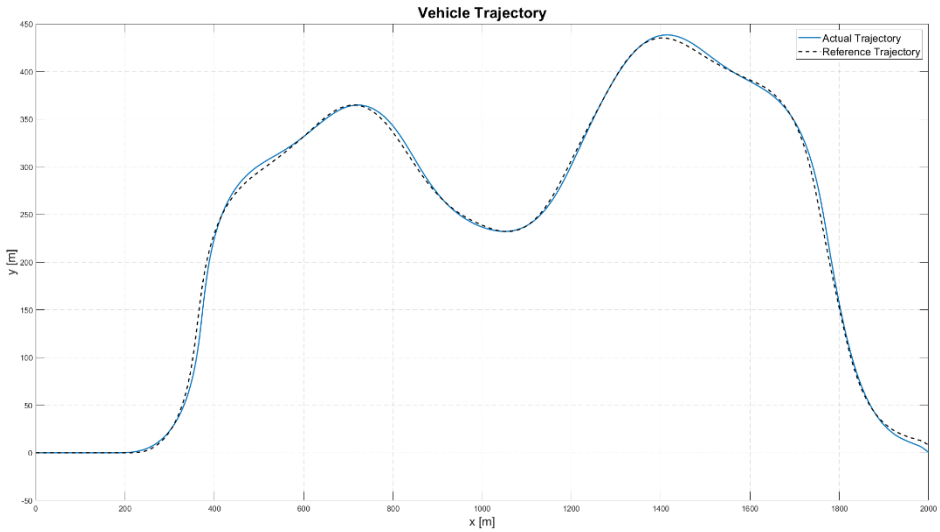

Figure 4.7: Vehicle Trajectory

In the figure, different degrees of deviations can be noticed, deviations are especially obvious after turning. The corresponding steering angles and reference steering angles are shown in Figure 4.8. The actual steering angles are relatively small compared to the reference steering angles, which indicates when the vehicle travels at high speeds, the LKA system cannot provide enough steering angle to the vehicle model.

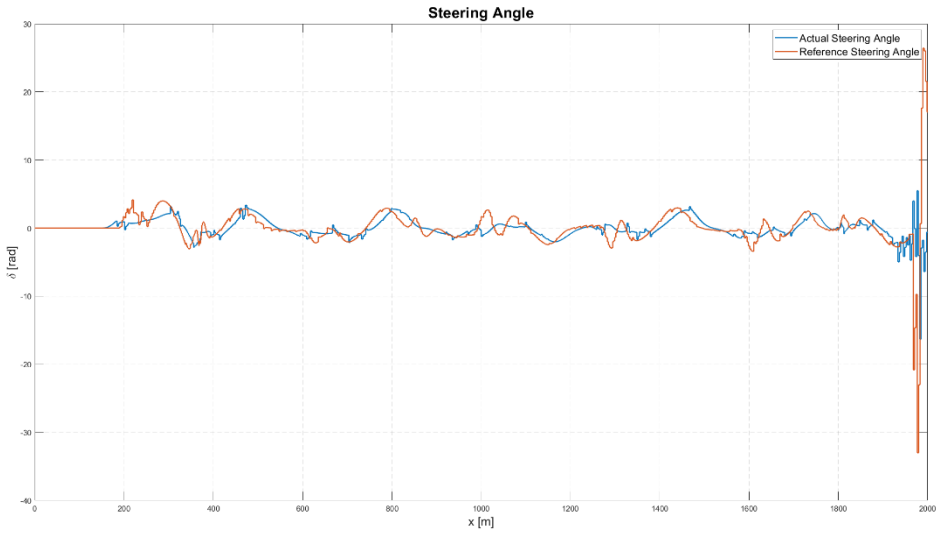

Figure 4.8: Vehicle Steering Angle

4.4 Summary

In this chapter, three different test cases were performed to verify and evaluate the control performance of the LKA system. From the first test, the system's ability to change lanes at different speeds is validated. With the help of the LKA system, the vehicle can change lanes smoothly at different speeds. Although overshoot may occur at some speeds, it still far less than the upper bound of the tolerance band, the vehicle can achieve the steady state in a short distance.

However, problems are found after performing the second and third tests. Both test cases focus on the system's path tracking ability. On the one hand, when the vehicle travels at low or medium speeds, good path tracking performance is achieved, but on the other hand, when the vehicle travels at high speed, large tracking errors are observed in both tests, and the LKA system cannot provide enough steering angle compensation.

Two potential reasons result in the system's poor performance at high speed. First, the error may be caused by the linearized MPC controller. In the MPC controller, the plant model is built by linearizing the nonlinear plant at each control interval, the linearization may produce the error. When the vehicle travels at low and medium speeds, the error is small, but when the vehicle travels at high speeds, the error is enlarged, which leads to the large tracking error shown in Figure 4.7. Another reason causing the large error is the simplified vehicle model. The vehicle model used in the test is a simplified model, although it has the basic functions of a vehicle, its incompleteness will also lead to errors during the simulation.

Chapter 5 Conclusion and Outlook

5.1 Conclusion

This thesis described the design process of an LKA system for Chevrolet Blazer, including the construction of a simplified hybrid electric vehicle model and the development of an LKA system.

By reviewing the team's previous research, the thesis established the overall structure and modeled the main components of the simplified hybrid electric vehicle model based on the P4 Split-Parallel HEV architecture. Afterwards, vehicle dynamics were derived to compose all the component models to a complete vehicle system model.

The development of the LKA system focused on the development of the MPC controller and its parameters' tuning process. The parameters were selected and tuned by summarizing and analyzing previous research. The selection can have an immense impact on the overall performance of the controller; in other words, wrong selection or tuning may result in system's low accuracy and large tracking errors.

5.2 Outlook

The LKA system designed in the thesis has the ability to change lanes at different speeds smoothly, and its low speed and medium speed path tracking ability is also validated. However, the system has a poor performance when the vehicle travels at high speed. To reduce the deviation and the tracking error at high speed, future research needs to be conducted, including the modification of the vehicle model and the advanced tuning

approach for the MPC parameters. Apart from that, more test cases and simulations need to be performed to validate and evaluate the system's performance. The supervisory controller and the vehicle model will be piped into MicroAutoBox (MABx) and the dSpace simulator, respectively. All the test cases will be tested and validated through the MABx controller and dSpace simulator before tested on the vehicle.

References

[1] Singh, K.V., Bansal, H.O. & Singh, D. A comprehensive review on hybrid electric vehicles: architectures and components. J. Mod. Transport. 27, 77–107 (2019). https://doi.org/10.1007/s40534-019-0184-3.

[2] Boyd, Steven. (2006). Hybrid Electric Vehicle Control Strategy Based on Power Loss Calculations.

[3] "Key Components of a Hybrid Electric Car", Alternative Fuels Data Center, https://afdc.energy.gov/vehicles/how-do-hybrid-electric-cars-work..

[4] Wikipedia contributors. (2020, June 19). Hybrid vehicle drivetrain. In Wikipedia, The Free Encyclopedia. Retrieved 02:56, June 25, 2020, from https://en.wikipedia.org/w/index.php?title=Hybrid_vehicle_drivetrain&oldid=963460034.

[5] "Mild Hybrid Electric Vehicle (MHEV) - architectures," X-Engineer, 2019. Retrieved from https://x-engineer.org/automotive-engineering/vehicle/hybrid/mild-hybrid-electric-vehicle-mhev-architectures/

[6] "UW_EMC Y1 Architecture Selection Report", 2019.

[7] Rill, Georg. (2006). Vehicle modeling by subsystems. Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG. 28. 10.1590/S1678-58782006000400007.

[8] MathWorks. (2019, August 31). Coordinate Systems in Vehicle Dynamics Blockset. Retrieved from https://www.mathworks.com/help/vdynblks/ug/coordinate-systems-in-vehicle-dynamics-blockset.html

[9] Serale, G.; Fiorentini, M.; Capozzoli, A.; Bernardini, D.; Bemporad, A. Model Predictive Control (MPC) for Enhancing Building and HVAC System Energy Efficiency: Problem Formulation, Applications and Opportunities. Energies 2018, 11, 631.

[10] Wikipedia contributors. (2020, June 3). Model predictive control. In Wikipedia, The Free Encyclopedia. Retrieved 02:53, June 25, 2020, from https://en.wikipedia.org/w/index.php?title=Model_predictive_control&oldid=960451937.

[11] Ramasamy, V.; Sidharthan, R.K.; Kannan, R.; Muralidharan, G. Optimal Tuning of Model Predictive Controller Weights Using Genetic Algorithm with Interactive Decision Tree for Industrial Cement Kiln Process. Processes 2019, 7, 938.

[12] Wojsznis, Willy & Gudaz, John & Blevins, Terry & Mehta, Ashish. (2003). Practical approach to tuning MPC. ISA transactions. 42. 149-62. 10.1016/S0019-0578(07)60121-9.