Development of a Model-based Control Strategy for Autonomous Vehicle Collision Avoidance

by

Ben MacCallum

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2022

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Human inattention is the leading cause of traffic accidents in many regions around the world. Autonomous vehicle technologies are rapidly emerging with the aim to remove the human factor in key driving procedures, such as perception, decision-making, path planning, and control. These technologies are subject to technological, ethical, and social scrutiny; therefore, extensive work is required to instill confidence in the reliability of these automated driving features. One key responsibility of automated driving is in planning and tracking a trajectory to avoid collisions with obstacles, such as other vehicles. One of the foremost challenges in the formulation of a feasible path is considering the dynamics and constraints of the vehicle and the environment.

Model predictive control (MPC) is one of the most common control techniques for its ability to handle constraints. For this reason, MPC has been widely studied for path planning and tracking for autonomous vehicles and mobile robots. MPC relies upon an accurate vehicle dynamics model which enables accurate state predictions, thereby resulting in effective control actions to achieve the desired objective. It is challenging, however, to capture all of the details and uncertainties of the dynamics associated with a vehicle. In particular, modeling tire dynamics requires detailed nonlinear models to fully reflect the vehicle behavior. One common technique for motion planning using MPC is to employ artificial potential fields (PFs) which generate an artificial repulsive force from obstacles or road boundaries to influence the controller to track the vehicle along a safe trajectory. Some state-of-the-art PF-based techniques include the PF intensity directly in the MPC objective function, thereby considering the vehicle constraints and dynamics as part of the path planning.

In this thesis, an enhanced PF-based motion controller is presented. The control design uses MPC with a detailed dynamics model; the model considers the combined-slip effect on tire forces, nonlinearities, and actuator dynamics. Therefore, it offers an improvement upon prior studies which rely upon simplified dynamics models. Moreover, the PF intensity is included in the objective function, like prior studies, although the PF approximation is further simplified by only considering the lateral component of the repulsive force as part of the latera controller. A separate, novel longitudinal control policy uses the longitudinal component of the PF gradient to regulate the speed setpoint when approaching an obstacle in the same lane; subsequently, proportional-

integral-derivative (PID) controllers command axle torque and brake pressure to track the reference speed. The developed controller and dynamics model are validated in both simulation and physical vehicle tests. To emulate the various driving scenarios where avoidance or stopping is required, a virtual driving environment is employed: simulated obstacles are placed in the roadway, the detections of which are sent to the controller. The controller performance is demonstrated in various evasive maneuvers, and in different road conditions.

Acknowledgements

I would foremost like to acknowledge the guidance and support of my supervisor, Professor Amir Khajepour. I would also like to recognize the advisement and support I received from Professor Ehsan Hashemi in my coursework and research. Additionally, the support and influence of my numerous lab mates has been monumental in my learning throughout my master's program. I would like to particularly recognize Ted Ecclestone, from whom I learned a lot when working alongside him during our various courses and research projects. Furthermore, I would like to thank the technical staff in the Mechatronics Vehicle Systems lab – namely, Aaron Sherratt, Jeff Graansma, Adrian Neil, and Michael Duthie – for their assistance with my development and experiments. Lastly, I would like to acknowledge the support of my family and friends, without whom I would not have realized this incredible opportunity.

Dedication

To my brother, Jake, for always being by my side and inspiring me to do what I love.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1. Motivation

The National Highway Traffic Safety Administration reported that drivers are the critical factor in 94% of crashes from 2005 to 2007 [1]. In many other countries around the world, such as the Czech Republic, for example, human inattention is the leading cause of traffic accidents [2]. The advent of autonomous driving is therefore fueled in part by the need to reduce traffic accidents by removing the human factor from the perception, decision making, and control of vehicles. Although autonomous driving has been recognized for its benefits with regard to reliable and safe transportation, it poses many technological, ethical and social challenges which must be addressed [3]. One outstanding technological challenge facing autonomous driving is the development of reliable motion planning and control techniques.

The key problem of planning and control for autonomous driving is determining a collision-free trajectory which navigates obstacles and reaches a goal state [4]. Planning the trajectory is challenging because vehicles are subject to dynamical and mechanical limits, rendering some maneuvers infeasible, such as those which are necessary to avoid a collision [5]. Many different planning and control techniques have been studied, some of which directly address the problem of satisfying the kinematic and dynamic limits of the vehicle. Model-predictive control (MPC) is the most widely adopted method of systematically handling constraints [6]; therefore, MPC is a common solution for path planning control [5], [7], [8]. MPC may be paired with one of many possible planning algorithms: some approaches generate the path directly in the control objective, while others use a standalone program to generate a reference trajectory which is then tracked by the MPC.

Potential field (PF)-based path planning has been widely studied for mobile robotics and autonomous driving [5], [7], [9]–[12]. Potential fields prevent collisions with obstacles by emulating a repulsive force away from the obstacle [13]. Moreover, potential fields may be generated for different types of features, including vehicles and road and lane boundaries [14]. The

advantage of using PFs for obstacle avoidance is that the intensity of the field may be integrated directly into the control objective function such that the controller prevents collisions while considering other objectives and constraints. With such an approach, no reference trajectory is generated for obstacle avoidance, but rather the controller generates an optimal trajectory spanning the prediction horizon at each discrete control step.

The effectiveness of MPC is highly dependent on an accurate vehicle dynamics model for prediction. It is not feasible, however, to capture all the complexities and uncertainties of the vehicle dynamics. Tire dynamics, for instance, are highly complex and uncertain, especially in extreme maneuvers [15]. Therefore, many developments have focused on either increasing the complexity of dynamics models or, instead, adapting to the vehicle dynamical uncertainties by training a learning model. Both approaches provide a higher level of prediction accuracy over a traditional, fixed linear model. There is a need, however, to explore the effects of a higher fidelity dynamics model for motion planning and control, such as that which considers nonlinear, combined-slip tire response. A complex tire model is particularly advantageous when the vehicle is engaged in harsh maneuvers, where the lateral and longitudinal tire slip is significant.

1.2. Objectives

One principal objective of this thesis is to demonstrate the application of a high accuracy dynamics model to an MPC-based motion planning and control routine. An existing planning approach based on artificial PFs is adapted using an improved prediction model. This thesis outlines a dynamics model which features consideration for combined-slip tire forces, actuator dynamics, and nonlinearities in the body dynamics. Simulation results are studied to show the prediction accuracy of various key states, such as tire forces, in various collision-avoidance maneuvers.

The second main objective is to reduce computational complexity of the MPC problem, particularly as it relates to the approximation of the PF intensity. Unlike the prior developments, the presented control strategy is decoupled: MPC handles lateral control and planning based on PFs, while the longitudinal control uses a feedback loop based on the PF gradient to slow and stop the vehicle when driving behind a lead vehicle. Additional steps are proposed to simplify the PF approximation, such as ignoring the longitudinal gradient of the potential fields in the lateral control policy.

The third objective is to construct a simulation environment to validate the performance of the controller. This simulation environment must feature a vehicle dynamics model to represent the plant, as well as a means of generating driving scenarios and reading sensor information based on those scenarios. The vehicle plant model is developed using a collection of library blocks provided in MATLAB and Simulink. The model is adequately detailed for representing the lateral and longitudinal vehicle dynamics; it considers combined-slip tire forces and aerodynamics. In this thesis, the dynamics of the plant model are directly compared to the controller model, particularly regarding the estimated tire forces of each.

1.3. Outline

Chapter 2 describes the relevant literature – particularly, existing approaches for motion planning and tracking, as well as vehicle dynamics modeling. The described literature focuses on using MPC for path planning and tracking. Many of the studies use PFs in combination with MPC. The path planning approach described in this thesis is based on the methodology of one study by Rasekhipour *et al.,* which is outlined in detail in this section.

Chapter 3 describes the vehicle dynamics model developed for the MPC motion planning controller. The models of each dynamical aspect of the vehicle are outlined, including the nonlinear bicycle model of the vehicle body, Burckhardt combined-slip tire model, wheel model, and steering actuator model. The dynamics equations for each component are introduced. Moreover, these equations are constructed into a linear model, where the system is linearized about the operating point. Then, the system is discretized in order to make discrete state predictions within the context of the MPC controller.

Chapter 4 describes the overall design and evaluation of the path planning and control algorithm. This includes an overview of the PF-based approach adapted from the literature [7], where the PF approximation is convexified for solution with quadratic problem. Next, the design of the MPC control objective is shown, where the PF intensity, path tracking error, and steering effort are penalized to optimize the steering action for collision avoidance and lane centering. Additionally, the stability constraints are described, where the yaw rate and side-slip angle are limited to prevent oversteering. The longitudinal control policy is also described in this section, which uses the

longitudinal potential field gradient as feedback to cruise or stop the vehicle for a lead vehicle. For estimating unmeasurable vehicle states, a pair of state observers are also evaluated in this section.

Chapter 5 describes the simulations generated for validating the controller design. The results are generated using a simulation environment featuring a vehicle dynamics model, which is compared directly to the controller model. Additionally, virtual driving scenarios are integrated in-the-loop, where vehicle and lane boundary detections are generated. A set of simulation results are presented which demonstrate the performance of the controller in various driving scenarios. This includes the avoidance of static and moving vehicles, performing adapting cruise and stopping for a lead vehicle, and performing an evasive maneuver in adverse road conditions.

Chapter 6 includes a set of experimental results on a physical vehicle. A test vehicle featuring automated electric drive motors and steering is used to validate the motion controller in various collision-avoidance maneuvers. A virtual driving scenario allows for placement of dummy obstacles in the environment, simulated detections of which are sent to the controller. The experimental tests reflect those executed in simulation, but at lower speeds.

Chapter 7 includes a set of conclusions and a plan for future work to develop and evaluate the control method described in this thesis. The conclusions highlight the advantages of higher fidelity vehicle and tire models for the controller, as the state predictions are shown to closely match those of the vehicle plant. The future work includes suggestions to improve the accuracy of the controller model, as well as to expand the scope of this control method to other vehicle control techniques, such as torque vectoring.

CHAPTER 2

Literature Review and Background

2.1. Motion Planning and Tracking

Vehicle motion control requires an advanced control approach which considers numerous constraints and control objectives. A common control approach, MPC, relies upon a dynamics model to optimize the control inputs, such as steering or acceleration, over a given time horizon. The optimization is performed with respect to certain control objectives associated with path tracking, including lane centering and speed tracking. Additionally, MPC solves the problem with regard for dynamical or mechanical constraints of the vehicle. Therefore, MPC been widely studied for path tracking of autonomous vehicles and mobile robots [5], [8], [16]–[19]. For example, Yuan *et al.* developed an MPC-based mixed motion planning and tracking routine for longitudinal and lateral control [17]. Other studies rely upon MPC for motion tracking, often paired with a path planning program such as a graph search-based or artificial potential field-based planner.

In conjunction with MPC, studies have demonstrated techniques to include collision avoidance as part of the control objective. For instance, Brown *et al.* present an MPC controller for combined path planning and tracking [16]. In this study, the authors employed a safe driving area constrained by road boundaries and the bounds of obstacles in the roadway. One objective of the controller is to track a reference path, as defined by the center of the desired driving lane. However, when an obstacle is present in the lane, the constraints influence the control input to result in the vehicle safely deviating from the lane to avoid a collision. Additionally, vehicle stability is considered by constraining states such as the yaw rate. Similarly, Schulman *et al*. propose an approach for mobile robot path planning where the control actions are selected to achieve the optimal collision-free trajectory [20]. The algorithm therein handles obstacles with various forms, including thin and irregularly shaped obstacles. Constraints on the no-collision conditions are generated according to the exact geometry of the robots and obstacles.

Artificial potential fields (PFs) are another technique for achieving collision avoidance in conjunction with MPC. PFs generates an artificial repulsive force away from an obstacle, the magnitude of which can be used as a heuristic for path planning. When applied in conjunction with MPC, the PF intensity may be used to generate a reference path which is tracked by the controller. Ji *et al.* have presented a path planning and tracking routine where the reference trajectory is generated using potential fields [5]. Although this approach to potential field-based path planning considers a variety of obstacles, the planned path does not directly consider the dynamical constraints of the vehicle [7]. Therefore, several studies have included potential fields directly in the MPC objective function alongside the vehicle dynamics terms [7], [11], [21], [22]. This approach ensures that the planned path is feasible according to the dynamical constraints of the vehicle. Huang *et al.* have proposed a path planning program for vehicles to avoid obstacles using a resistance network and model predictive control (MPC) [23]. This method utilizes a two-tiered approach to motion planning and control: the upper layer consists of a path-planner based on the resistance network, while the lower layer features separate MPCs for lateral and longitudinal control based on a point mass model. Rasekhipour *et al.* introduce a method of convexifying potential field functions so that the MPC optimization problem may be solved using quadratic programming [7]. This solution demonstrates good obstacle avoidance behavior, but further work is required to enhance the accuracy of the vehicle dynamics model, such as consideration for the combined-slip effect on tire forces. Additionally, methods of further reducing the computational complexity of the MPC problem should be explored, particularly for the potential field approximation.

Other approaches feature a standalone planning algorithm which generates an optimal path for the MPC to track. Dolgov *et al.* propose a path planning algorithm for autonomous vehicles which employs the A-star algorithm [24]. The planner obtains a kinematically feasible trajectory, as the search algorithm is applied to the kinematic state space of the vehicle. The optimal path is selected based on two heuristics: the non-holonomic cost, which considers the shortest path in the absence of obstacles, and the holonomic-with-obstacles cost which considers the cost of collisions. This study compares the effectiveness of free-space planning to semi-structured planning guided by lane networks. Erke *et al.* show a similar approach to local planning using the A-star algorithm [25]. The A-star is typically problematic when applied to autonomous driving; it may plan a path which is optimal in terms of finding the shortest distance around an obstacle, but at the cost of

deviating from the natural trajectory of a human driver. Therefore, this study uses a global guideline which matches the best path in the absence of obstacles. The algorithm is shown to be robust and stable and, moreover, it results in the avoidance of obstacles much earlier than a traditional A-star planner.

The state-of-the-art approaches to path tracking with regard to highway lane following vary in nature. Some developments use an end-to-end approach where a single model is trained to learn the dynamics between detection and control. For example, Chen *et al.* describe an end-to-end learning approach using convolutional neural networks (CNNs) to steer the vehicle for lane centering based directly on the vision detections of the lane boundaries [26]. This approach aims to mimic the black-box nature of a human driver, where perception and sensory information informs the control actions without consideration for the intermediate steps therebetween. Conversely, others opt for a modular approach where clear interfaces between functions, such as perception, decision-making, and control, are established. For example, Liu *et al.* describe a technique where lane boundaries are detected using a lightweight network which processes camera data [27]. Then, a separate Kalman filter-based controller tracks the lane center using the output of the lane detection module. This type of approach allows for analysis of intermediate signals in the autonomous driving stack, thereby enabling easier troubleshooting and validation of each function.

2.2. Dynamics Modeling

One important step to implement a reliable MPC controller for path planning and tracking is the design of a reference model to represent the vehicle dynamics. The reference model should capture the actual dynamics of the vehicle with a high degree of accuracy so that the control actions are optimized according to realistic state predictions. Many studies which employ MPC for path planning and tracking use a simplified dynamics model of the vehicle as a reference. Although the complex dynamics of a real vehicle cannot be fully captured, some studies increase the level of detail in the reference model to achieve a better representation of the vehicle dynamics. This includes the consideration of nonlinear tire dynamics or steering actuator dynamics.

The bicycle model is the most commonly used vehicle model for its simplicity and ease of implementation [28]. The model features 3 degrees-of-freedom (DOF) – the lateral and

longitudinal translations and yaw rotation. Moreover, it assumes a single-track system, featuring one tire in the front and one tire in the rear. A kinematics- or dynamics-based bicycle model may be used for representing the vehicle. Some studies have shown successful results using only a kinematic vehicle model [7], [22]; however, a dynamics model provides a higher level of accuracy, as it affords consideration for the effects of tire forces. The dynamic bicycle model is shown to capture the vehicle dynamics adequately well for the purpose of trajectory planning, but its estimates deviate significantly from a higher DOF model in harsh driving maneuvers, such as those where high lateral acceleration is exhibited [29]. Adaptations may be made to improve the modeling accuracy of the bicycle model by pairing it with a nonlinear tire force model and actuator dynamics model, for example.

Tire force modeling has a significant effect on the accuracy of any vehicle dynamics model. To reduce complexity in the model, several studies have developed MPC-based lateral controllers which assume linear tire dynamics [7], [11], [22], [30]. However, the simplification by assuming linear tire dynamics leads to significant prediction inaccuracies due to normalized tire forces at large slip angles [31]. Moreover, ignoring the combined-slip effect on tire forces impairs prediction accuracy at large longitudinal slip ratios. Therefore, numerous approaches have opted for including nonlinear and combined-slip tire models [17], [31]–[34]. These approaches provide more accurate estimates of tire forces, particularly in scenarios where both lateral and longitudinal slip is significant, such as in poor road conditions, or evasive maneuvers. These tire models are fit to empirical data by tuning a set of parameters. Common nonlinear tire models include the Pacejka magic formula [35] and the Burckhardt model, the latter of which has a much more simple form [36]. Furthermore, both models may be formulated in terms of pure lateral-, pure longitudinal-, or combined-slip. It is shown in literature that these two models perform similarly in a variety of road conditions [36].

A simplified model which ignores actuator dynamics results in the false assumption that the optimal control actions by the MPC are immediately applied at the lowest level of the vehicle mechanics. For example, the steering request applied by the controller is not immediately reflected as the actual steering angle at the wheel; there is some transient response which should be considered. Therefore, the accuracy of the bicycle model is shown to improve when steering actuator dynamics are included [18]. Kim *et al.* have proposed a MPC for path tracking which

features a 2nd order steering dynamics model in the controller [18]. Other actuators, such as the accelerator and brakes, may also be modeled as first or 2nd order systems in order to improve the prediction accuracy of the MPC.

Despite the efforts to model the many complexities of a vehicle, there remains numerous sources of error that cannot be easily represented with physical models. Therefore, many studies have explored learning-based models, resulting in the advent of learning-based model predictive control (LBMPC). LBMPC features a dynamics model which includes an unknown error term that is updated based on the learned behavior of the system. For example, Kabzan *et al.* have developed a LBMPC for autonomous racing which uses Gaussian Process Regression (GPR) to learn the error between the reference model and vehicle plant [37]. Ostafew *et al.* have shown similar developments using GPR-based LBMPC for mobile robot path tracking [38], [39]. GPR is a favorable technique as it not only provides an estimate of the model inaccuracy, but also provides the variance of the estimate based on the extent to which the model has been trained in a particular state space. Other studies use parametric models such as neural networks to characterize the reference model error. [40], [41]. Although LBMPC using a neural network is shown to be safe and robust [41], the reliability of the network output is uncertain, as it does not provide the variance of its estimate.

CHAPTER 3

Prediction Model for Vehicle and Wheel Dynamics

A detailed vehicle dynamics model is developed for the MPC prediction model by combining state-of-the-art models for the vehicle body, wheels, tires, and actuators. This model considers combined-slip tire forces according to the Burckhardt tire model. Moreover, the wheel dynamics and steering actuator dynamics are included. This dynamics model enhances dynamics models presented in similar prior studies, many of which ignore the nonlinear and combined-slip effects on tire forces.

3.1. Bicycle Model

The bicycle model simplifies the lateral vehicle dynamics by modeling the vehicle body as a single-track system with one front tire and one rear tire. By using such model as a reference for MPC, the controller complexity is reduced, thereby improving computational load. The bicycle model may be approximated as a linear system; however, in doing so, the accuracy is compromised. Therefore, a nonlinear formulation is selected, which shall be linearized at the operating point for the MPC controller.

This formulation of the bicycle includes the following states: the global X-coordinate, X; the longitudinal speed, v_x; the global Y-coordinate, Y; the lateral speed, v_y; the yaw angle, ψ; and the yaw rate, r, as shown in Figure 3.1. The following differential equations characterize the vehicle body dynamics as

$$f_1 = \dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \tag{1}$$

$$f_2 = \dot{v}_x = \frac{F_{x,f} \cos\left(\delta_{f,\mathrm{act}}\right) - F_{y,f} \sin\left(\delta_{f,\mathrm{act}}\right) + F_{x,r}}{m} + v_y r \tag{2}$$

$$f_3 = \dot{Y} = v_y \cos(\psi) + v_x \sin(\psi) \tag{3}$$

$$f_4 = \dot{v}_y = \frac{F_{x,f} \sin(\delta_{f,\text{act}}) + F_{y,f} \cos(\delta_{f,\text{act}}) + F_{yr}}{m} - v_x r \tag{4}$$

$$f_5 = \dot{\psi} = r \tag{5}$$

$$f_6 = \dot{r} = \frac{l_f (F_{x,f} \sin(\delta_{f,\text{act}}) + F_{y,f} \cos(\delta_{f,\text{act}})) - l_r F_{y,r}}{I_z} \tag{6}$$

where $F_{x,f}$ and $F_{x,r}$ are the total front and rear longitudinal tire forces in the respective wheel frames, respectively, and $F_{y,f}$ and $F_{y,r}$ are the total front and rear lateral tire forces in the respective wheel frames, respectively. Moreover, m is the vehicle mass and I_z is the yaw moment of inertia. The actual front steering angle is denoted as $\delta_{f,\text{act}}$, where the steering request affects the actual steering angle according to the actuator dynamics model described in 0.

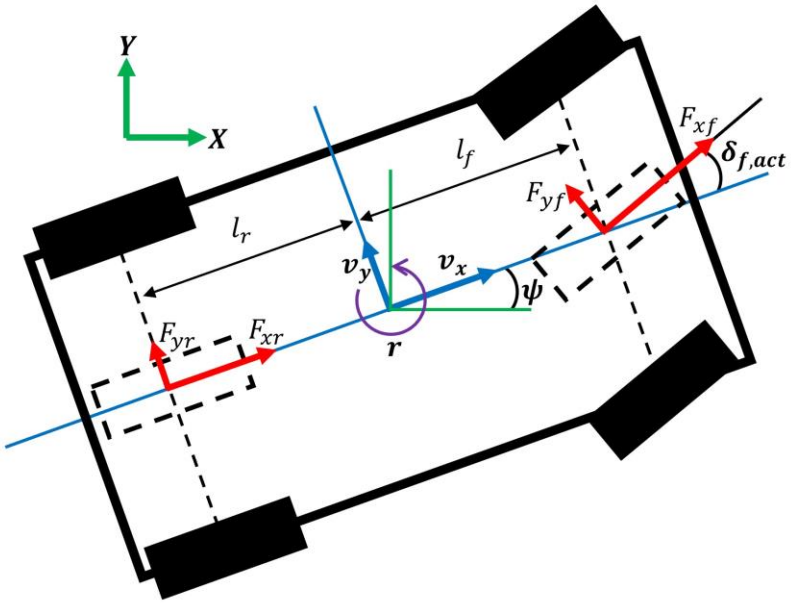

Figure 3.1: Bicycle model for vehicle body dynamics

3.2. Wheel Dynamics

The wheel dynamics are also included as part of the reference model to characterize the longitudinal slip, as it affects the estimation of both lateral and longitudinal tire forces. Figure 3.2 shows the wheel states and parameters for each of the two wheels in a single-track system. The time derivatives of the front and rear wheel angular velocities are

$$f_7 = \dot{\omega}_f = \frac{2T_{\text{axle},f} - F_{x,f}R_{\text{eff}}}{2I_w} \tag{7}$$

and

$$f_8 = \dot{\omega}_r = \frac{-F_{x,r}R_{\text{eff}}}{2I_w}, \tag{8}$$

respectively, where $T_{\text{axle},f}$ is the axle torque applied to each front wheel, R_{eff} is the effective tire rolling radius, and I_w is the rolling moment of inertia of each wheel.

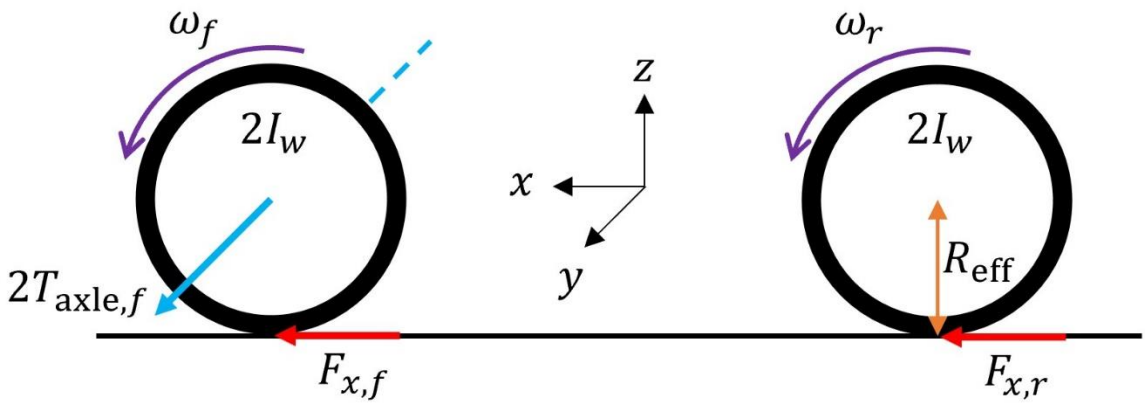

Figure 3.2: Single-track wheel dynamics model

3.3. Actuator Dynamics

The incorporation of steering actuator dynamics in the prediction model improves the accuracy of the bicycle model [18]. Therefore, the steering actuator dynamics are modeled as a first order system,

$$f_9 = \dot{\delta}_{f,\text{act}} = \frac{\delta_{f,\text{req}} - \delta_{f,\text{act}}}{\tau_{\delta_f}},$$

(9)

where $\delta_{f,\text{req}}$ is the steering request and $\tau_{\delta_f} = 50$ ms is the estimated steering actuator time constant. By including the actuator dynamics, the actual steering angle, $\delta_{f,\text{act}}$ may be added to the state vector, and $\delta_{f,\text{req}}$ may be considered as a control input to be optimized by MPC.

3.4. Combined-Slip Tire Forces

Combined-slip tire models consider the interaction of longitudinal and lateral slip in estimating the tire forces, whereas pure slip models map the lateral slip to lateral forces and longitudinal slip to longitudinal forces. Therefore, combined-slip models afford more accurate force estimates when both longitudinal and lateral slip are significant, such as during evasive maneuvers. In this study, the tire forces are estimated using the Burckhardt combined-slip tire model. The Burckhardt model is simpler in its expression than the more widely used Pacejka model, whereas the numerous parameters of the latter are set according to empirical data. Therefore, the Burckhardt model is easily tunable by least-squares (LS) fitting or by trial and error. Moreover, the simpler formulation of the Burckhardt model is less computationally intensive.

According to the combined-slip Burckhardt model, the lateral force on each tire in the single-track model, where $i \in \{f, r\}$, is defined as

$$F_{y,i} = F_{z,i} \frac{\alpha_i}{S_{\text{res},i}} \left[C_{1y,i} (1 - e^{-C_{2y,i} S_{\text{res},i}}) - C_{3y,i} S_{\text{res},i} \right],$$

(10)

where $C_{1y,i}$, $C_{2y,i}$, and $C_{3y,i}$ are the tunable model parameters, and F_{zi} is the total downward force from the vehicle body on each axle. The tire slip angle, α_i, is computed for the front and rear tires, respectively, as,

$$\alpha_f = \delta_{f,\text{act}} - \tan^{-1}\left(\frac{v_{y,f}}{v_x}\right) \tag{11}$$

and

$$\alpha_r = -\tan^{-1}\left(\frac{v_{y,r}}{v_x}\right) \tag{12}$$

where v_f and v_r are the lateral speeds at the front axle, $v_{y,f} = v_y + l_f r$, and rear axle, $v_{y,r} = v_y - l_r r$, respectively.

According to the Burckhardt combined-slip model, the longitudinal force on each tire is defined as

$$F_{x,i} = F_{z,i}\frac{\lambda_i}{S_{\text{res},i}}\left[C_{1x,i}\left(1 - e^{-C_{2x,i}S_{\text{res},i}}\right) - C_{3x,i}S_{\text{res},i}\right], \tag{13}$$

where $C_{1x,i}$, $C_{2x,i}$, and $C_{3x,i}$ are the tunable model parameters, and λ_i is the longitudinal tire slip ratio, computed for the front and rear tires, respectively, as

$$\lambda_f = \frac{-\left(v_x\cos\delta_f + v_{y,f}\sin\delta_f - \omega_f R_{\text{eff}}\right)}{\omega_f R_{\text{eff}}} \tag{14}$$

and

$$\lambda_r = \frac{-(v_x - \omega_r R_{\text{eff}})}{\omega_r R_{\text{eff}}}, \tag{15}$$

where R_{eff} is the effective rolling radius of the tires. The combined-slip ratio, $S_{\text{res},i}$, is defined at each axle as

$$S_{\text{res},i} = \sqrt{\lambda_i^2 + \alpha_i^2},$$

(16)

which characterizes the total slip as it affects both the longitudinal and lateral tire forces. In each of the respective force equations ((10), (13)), the fraction of the slip in the respective direction over the total slip is multiplied by the Burckhardt model equation computed using the combined slip ratio.

To accurately represent the force response of the tires, the model parameters must be tuned to fit the empirical force data. The Burckhardt tire model parameters in (10) and (13) are tuned by fitting the force response by LS to 205/55R16 tire data for pure lateral and longitudinal slip conditions, respectively, as shown in Figure 3.3d. The data is fit for $F_z = 8090.49$ N for the front tires and $F_z = 4045.24$ N for the rear tires, as these closely match the actual respective values of F_z in the simulated vehicle plant. The resultant parameters are shown in Table 3.1.

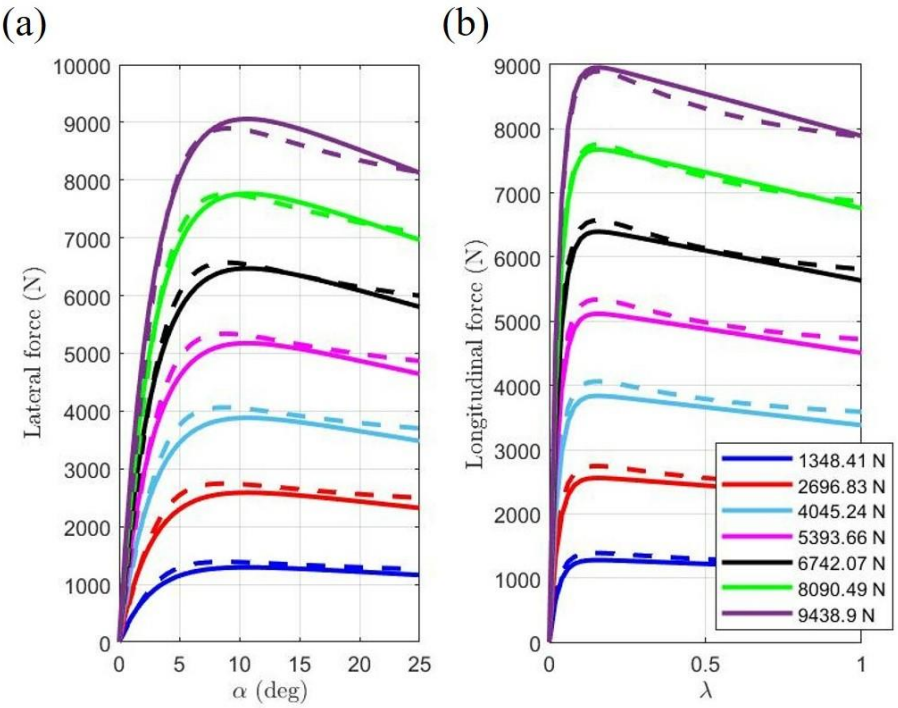

Figure 3.3: (a) Lateral and (b) longitudinal tire forces using the Burckhardt model for pure-slip conditions. The parameters are tuned by LS regression fitting to tire data for 205/55R16 tires at $F_z = 8090.49$ N.

Table 3.1: Tuned Burckhardt tire model parameters

Parameter	Value	Parameter	Value
C_{1xf}	0.9744	C_{1yf}	1.075
C_{2xf}	35.42	C_{2yf}	20.45
C_{3xf}	0.1387	C_{3yf}	0.4902
C_{1xr}	1.02	C_{1yr}	1.121
C_{2xr}	36.16	C_{2yr}	21.16
C_{3xr}	0.143	C_{3yr}	0.5077

3.5. Model Linearization and Discretization

3.5.1. Linearization at Operating Point

MPC generally requires a linear prediction model of the form, $\dot{x} = Ax + Bu$, where A and B are the system matrices, x is the state vector, and u is the control vector. Due to the incorporation of the combined-slip Burckhardt tire model, however, the vehicle dynamics model used for prediction is nonlinear. Therefore, it is necessary to linearize the model at each control step, thereby ignoring nonlinearities over the duration of the prediction horizon.

The state vector for the prediction model is $x = \left[X\ v_x\ Y\ v_y\ \psi\ r\ \omega_f\ \omega_r\ \delta_{f,\text{act}}\right]^\mathsf{T}$ and the single control input is the front steering angle request, $u = \delta_{f,\text{req}}$. The set of state equations in (1) - (9) comprise the vector, $f(x,u) = [f_1, f_2, \ldots f_9]$, which characterizes the continuous-time dynamics of the vehicle system. By linearizing about the operating point, the system may be approximated linearly as

$$\dot{x} = f(x,u) \approx Ax + Bu + W,$$

(17)

where

$$A = \left[\begin{matrix} \frac{\partial f_1}{\partial X} & \cdots & \cdots & \frac{\partial f_1}{\partial \delta_{f,\text{act}}} \\ \vdots & \frac{\partial f_2}{\partial v_x} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_9}{\partial X} & \cdots & \cdots & \frac{\partial f_9}{\partial \delta_{f,\text{act}}} \end{matrix}\right]\Bigg|_{x=x_t,\delta_{f,\text{req}}=\delta^*_{f,\text{req},t-1}},$$

(18)

$$B = \left[\begin{matrix} \frac{\partial f_1}{\partial \delta_{f,\text{req}}} \\ \vdots \\ \frac{\partial f_9}{\partial \delta_{f,\text{req}}} \end{matrix}\right]\Bigg|_{x=x_t,\delta_{f,\text{req}}=\delta^*_{f,\text{req},t-1}},$$

(19)

and where the matrix, W, includes the residual linearization terms, as

$$W = f_t - Ax_t - B\delta^*_{f,\text{req},t-1}.$$

(20)

The operating point is characterized by the current state, x_t, and the previous optimal steering angle, $\delta^*_{f,\text{req},t-1}$. Additionally, the vector characterizing the system dynamics, f, is shown as f_t when evaluated at the current time step.

3.5.2. Discretization

To implement the dynamics model for MPC, the continuous-time formulation must be converted to a discrete-time formulation. Thereby, state predictions are generated for a set of discrete time steps into the future. The continuous-time linearized model is discretized with a time step of length, $\tau_\text{s} = 100$ ms. At each time step, t, each future state is predicted as

$$x_{t+k+1} = A_d x_{t+k} + B_d \delta_{f,\text{req},t+k} + W_d,$$

(21)

where A_d, B_d, and W_d are the discretized versions of the system matrices, A, B, and W, respectively as [42]

$$A_d = e^{A\tau_\text{s}}$$

(22)

$$B_d = \left(\int_0^{\tau_\text{s}} e^{At}\, dt\right) B = \sum_{j=1}^{N} \frac{A^{j-1}\tau_\text{s}^j}{j!} B$$

(23)

$$W_d = \left(\int_0^{\tau_\text{s}} e^{At}\, dt\right) W = \sum_{j=1}^{N} \frac{A^{j-1}\tau_\text{s}^j}{j!} W$$

(24)

The solutions to the integrals in (23) and (24) are approximated with Taylor series expansions, as shown. The number of terms in each expansion is determined by trial and error where additional terms are added until the state predictions from a nominal state converge to consistent values. For each, B_d and W_d, the number of terms, N, is 4.

3.5.3. Prediction Horizon Formulation

At each discrete time step, t, a sequence of state predictions is made over the prediction horizon with N_p time steps, each with a length of τ_s, as

$$x_{\text{pred}} = \begin{bmatrix} x_{t+1} \\ \vdots \\ x_{t+N_p} \end{bmatrix} = \Phi x_t + \Theta \begin{bmatrix} \delta_{f,\text{req},t} \\ \vdots \\ \delta_{f,\text{req},t+N_c-1} \end{bmatrix} + Z$$

(25)

where the matrices, Φ, Θ, and Z are defined as,

$$\Phi = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{N_p} \end{bmatrix},$$

(26)

$$\Theta = \begin{bmatrix} B_d & 0 & \dots & 0 \\ A_d B_d & B_d & 0 & \vdots \\ \vdots & & \ddots & 0 \\ \vdots & & & B_d \\ A_d^{N_p-1} B_d & \dots & \dots & \sum_{k=0}^{N_p-N_c} A_d^{N_p-N_c-k} B_d \end{bmatrix},$$

(27)

and

$$Z = \begin{bmatrix} W_d \\ (A_d + 1)W_d \\ \vdots \\ \left(A_d^{N_p-1} + A_d^{N_p-2} + \cdots + 1\right) W_d \end{bmatrix}.$$

(28)

The state prediction vector, x_{pred}, represents a matrix-based formulation of the prediction horizon, generated by repeating the discrete state updates from (21). This form is used for formulating the MPC problem which optimizes the set of steering actions, $\{\delta_{f,\text{req},t}, \dots, \delta_{f,\text{req},t+N_c-1}\}$, as described in 4.3.

The prediction horizon length is a key parameter for MPC, as it affects the length of time that the model generates state predictions. In this study, the prediction horizon length, N_p, is set to 30 steps with a time step length, $T_s = 50$ ms; therefore, the prediction horizon spans 1.5 s. With a nominal speed of 80 km/h, the prediction horizon spans a future distance of 33.33 m.

CHAPTER 4

Development of a Potential Field-based Motion Planning Controller

A comprehensive planning and control system is developed which employs MPC and PF-based path planning for collision avoidance. Based on prior studies which use PFs for path planning, this method includes an approximation of the PF intensity directly in the MPC objective function such that the proximity to each obstacle is penalized during the prediction horizon. In this study, however, the potential field approximation is further simplified by only considering the repulsive force from the PFs which acts in the lateral road direction. Additionally, this approach decouples the lateral and longitudinal controllers: the lateral control is handled by MPC according to PFs and other stability objectives, while the longitudinal control is handled by a feedback-based controller using the potential field gradient in the longitudinal direction. Thereby, the MPC-based control problem is simplified, and the speed control is handled much like a traditional adaptive cruise control (ACC) system, but with the employment of artificial PFs.

4.1. Potential Fields

Artificial potential fields are generated for each feature in the environment which requires some path planning on the part of the ego vehicle to avoid collision or to avoid undesired lane departures. The potential field intensity is penalized in the controller cost function such that the optimal control solution should safely navigate the ego vehicle in the presence of obstacles and lane or road boundaries. Figure 4.1 shows an example of the potential field intensity generated around a non-crossable obstacle, such as a vehicle.

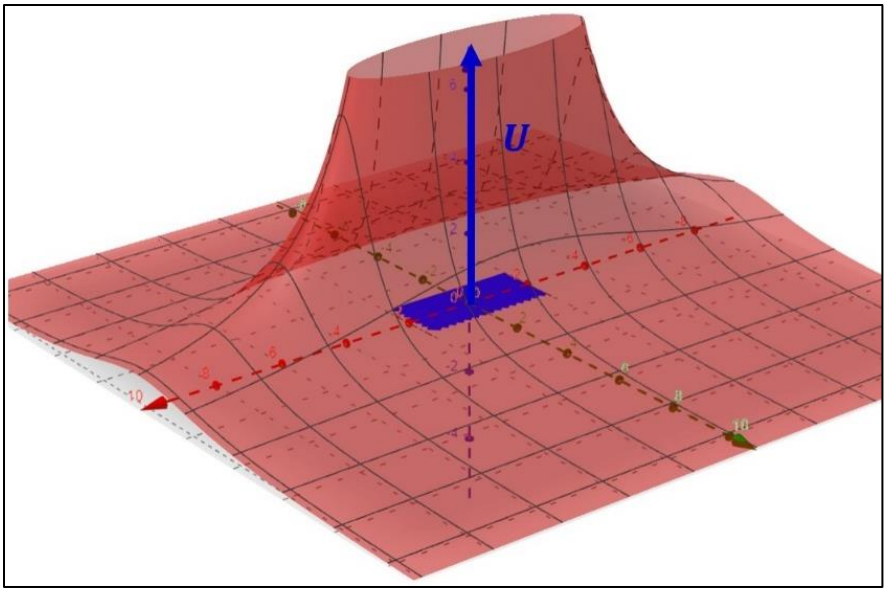

Figure 4.1: Potential field intensity surrounding a non-crossable obstacle.

4.1.1. Non-crossable Obstacles

For a non-crossable object, such as a vehicle, pedestrian, or barrier, the potential field intensity in its proximity is generated according to [7]

$$h_i = \frac{a_{NC}}{s_i^{b_{NC}}},$$

(29)

where a_{NC} and b_{NC} are intensity and shape parameters, respectively, and s_i is the signed distance between the ego vehicle and i^{th} obstacle. The signed distance is scaled by safe distances, $\Delta X_{s,i}^r$ and $\Delta Y_{s,i}^r$, to adjust the shape of the field in the lateral and longitudinal directions as

$$s_i = \sqrt{s_{X^r,i}^2 + s_{Y^r,i}^2} = \sqrt{\left(\frac{\Delta X_i^r}{\Delta X_{s,i}^r}\right)^2 + \left(\frac{\Delta Y_i^r}{\Delta Y_{s,i}^r}\right)^2},$$

where ΔX_r and ΔY_r are the longitudinal and lateral distances between the ego vehicle and i^{th} obstacle in the road coordinate frame. The safe distances in the longitudinal and lateral directions are defined as

$$\Delta X_{s,i}^r = \Delta X_{s0}^r + v_x T_0 + \frac{\left(v_x - v_{x,i}\right)^2}{2a_{\text{nom}}}$$

(30)

and

$$\Delta Y_{s,i}^r = \Delta Y_{s0}^r + v_x T_0 \sin(\psi - \psi_i) + \frac{\left(v_y - v_{y,i}\right)^2}{2a_{\text{nom}}},$$

(31)

respectively, where ΔX_{s0} and ΔY_{s0} are the minimum safe distances when stationary, T_0 is a safe following time, and a_{nom} is the comfortable nominal acceleration of the ego vehicle.

4.1.2. Convexification and simplification of potential fields

The potential field function for an obstacle or lane boundary must be approximated as a convex function at the operating point, t, to enable a quadratic solution for the MPC objective function. A second order Taylor series is therefore used to approximate the field intensity, h, as

$$\begin{aligned} g_{\text{i}}^r(X_{t+k}, Y_{t+k}) = g_i^r(X_t^r, Y_t^r) + \nabla {g^r}_i^{\top}\big|_{X_t,Y_t} \begin{bmatrix} X_{t+k}^r - X_t^r \\ Y_{t+k} - Y_t \end{bmatrix} \\ + \frac{1}{2}\begin{bmatrix} X_{t+k}^r - X_t^r \\ Y_{t+k}^r - Y_t^r \end{bmatrix}^{\top} \nabla^2 g_i^r|_{X_t,Y_t} \begin{bmatrix} X_{t+k}^r - X_t^r \\ Y_{t+k}^r - Y_t^r \end{bmatrix}, \end{aligned}$$

(32)

This approximation is quadratic in nature; therefore, it is solvable by quadratic programming. However, convexity only holds true in this approximation if the diagonal terms of the Hessian matrix, $\nabla^2 g_i^r$, are positive. To achieve this condition, a transformation into the (ζ, η) frame is performed, as shown in Figure 4.2, according to the methodology presented in Rasekhipour *et al.* [7]. The gradient and Hessian of the Taylor series approximation are thereby rotated by γ – the angle between the scaled signed distance and the road heading angle, ϕ_r, as

$$\nabla g_{T,i} = \begin{bmatrix} \dfrac{\partial h_i}{\partial \zeta} \\ \dfrac{\partial h_i}{\partial \eta} \end{bmatrix}$$

(33)

and

$$\nabla^2 g_{T,i} = \begin{bmatrix} \dfrac{\partial^2 h_i}{\partial \zeta^2} & \dfrac{\partial^2 h_i}{\partial \zeta\, \partial \eta} \\ \dfrac{\partial^2 h_i}{\partial \zeta\, \partial \eta} & \dfrac{\partial^2 h_i}{\partial \eta^2} \end{bmatrix},$$

(34)

respectively. In the (ζ, η) frame, a non-zero gradient only exists in the ζ direction because η acts orthogonally to the direction of repulsion. Moreover, the off-diagonal terms in the Hessian matrix are zero. However, the diagonal terms are not guaranteed to be positive in the current form. Therefore, an eigenvalue decomposition is performed to ensure convexity, where the negative eigenvalues are removed; next, the Hessian matrix is recomposed with the modified eigenvalue matrix as

$$\nabla^2 g_{T,\text{conv},i} = V \Lambda_p V^{-1},$$

(35)

where V is a square 2×2 matrix whose j^{th} column is the eigenvector, q_j, and Λ_p is a 2×2 matrix whose first diagonal element is the single positive eigenvalue, λ_1, of $\nabla^2 g_{T,i}$. Consequently, the revised Hessian matrix, $\nabla^2 g_{T,\text{conv},i}$, is positive-definite and, thus, the potential field approximation is convex.

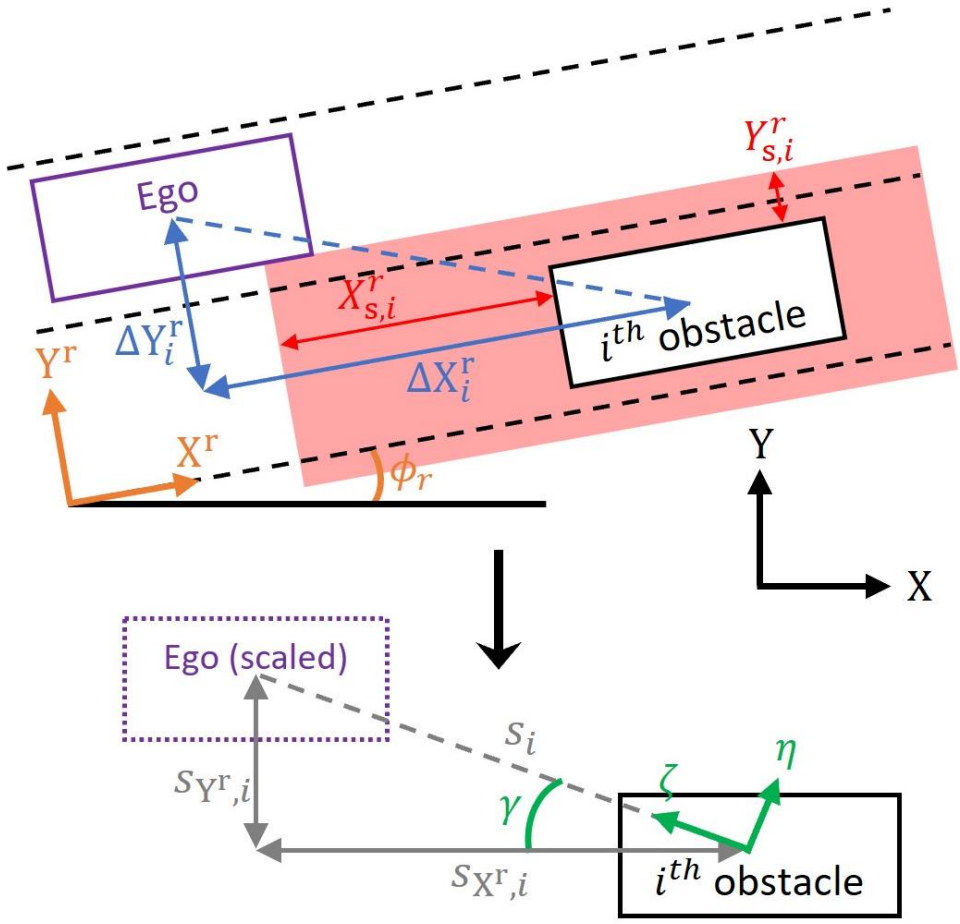

Figure 4.2: The X- and Y- distances between the ego vehicle and the i^{th} obstacle are scaled by the safe distances, $X_{s,i}$ and $Y_{s,i}$, and then rotated ζ, η coordinate frame.

Once convexified, the Hessian and gradient are rotated into the road coordinate frame as $\nabla g_i^{\mathrm{r}} = R_T \nabla g_{T,i}$ and $\nabla^2 g^r = R_T^{\mathrm{T}} \nabla^2 g_{T,\mathrm{conv},i} R_T$, respectively, where R_T is the 2-by-2 rotation matrix with rotation angle, γ.

The component of the repulsion emanating from each potential field in the longitudinal direction in the lane is not necessary for a pure lateral control policy. Therefore, the components of the gradient and Hessian which are parallel to the road heading are set to zero. Conversely, the prior study by Rasekhipour *et al.* keeps both the lateral and longitudinal components in the approximation because the lateral and longitudinal control are coupled [7]. Thus, only the lateral repulsive force of the field is included in the objective function. The resultant PF gradient and Hessian in the road frame, (X^r, Y^r), are defined as

$$\nabla g_i^{Y^r,r} = \begin{bmatrix} 0 & \frac{\partial h_i}{\partial Y_i^r} \end{bmatrix}^\top$$

(36)

and

$$\nabla^2 g_i^{Y^r,r} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{\partial^2 h_i}{\partial (Y_i^r)^2} \end{bmatrix},$$

(37)

respectively, where Y^r is the lateral direction of the road at the location of the i^{th} feature.

To achieve the form of the approximation shown in (32), where the expansion is defined with respect to the (X, Y) coordinate frame, a final rotation is performed. The lateral components of the gradients and Hessians are transformed back into the global coordinate frame as $\nabla g_i^{Y^r} = R_{\phi_{r,i}} \nabla g_i^{Y^r,r}$ and $\nabla^2 g_i^{Y^r} = R_{\phi_{r,i}}^\top \nabla g_i^{Y^r,r} R_{\phi_{r,i}}$, respectively, where $R_{\phi_{r,i}}$ is the rotation matrix for the road heading angle at the i^{th} feature location, $\phi_{r,i}$. Then, the PF approximation with respect to the (X, Y) frame is defined as

$$\begin{aligned} g_i^{Y_r}(X_{t+k}, Y_{t+k}) = g_i(X_t, Y_t) + \nabla g_i^{Y^r\top}\Big|_{X_t,Y_t} \begin{bmatrix} X_{t+k} - X_t \\ Y_{t+k} - Y_t \end{bmatrix} \\ + \frac{1}{2} \begin{bmatrix} X_{t+k} - X_t \\ Y_{t+k} - Y_t \end{bmatrix}^\top \nabla^2 g_i^{Y^r}\Big|_{X_t,Y_t} \begin{bmatrix} X_{t+k} - X_t \\ Y_{t+k} - Y_t \end{bmatrix}. \end{aligned}$$

(38)

Finally, the approximations of all PFs are summed as $G^y = \sum_{i=1}^N g_i^{Y^r}$, which is finally included in the MPC objective function, as described in 4.3.1. Each processing step of the PF approximation described in this section, such as the rotations, convexification, and simplifications, are performed once offline, whereas the final approximation shown in (38) is computed online. Therefore, the numerous aforementioned processing steps do not affect the computational expense of the controller.

4.2. Vehicle Stability Constraints

Due to the mechanical limitations of the vehicle, it is necessary to bound the front steering angle as $-\delta_{f,max} \leq \delta_f \leq \delta_{f,max}$, where $\delta_{f,\max}$ is set to 10 degrees. Additionally, the yaw rate, r, and side-slip angle, β, are constrained to prevent over-steering as [43]

$$\underline{\overline{\beta}} = \frac{r l_r}{v_x} \pm \tan\alpha_{\mathrm{p}},$$

(39)

and

$$\underline{\overline{r}} = \pm \frac{\mu_{r,\mathrm{est}} g}{v_x},$$

(40)

where $\mu_{r,\mathrm{est}}$ is the estimated road-tire friction coefficient, g is the acceleration due to gravity, and α_{p} is slip angle of the rear tires corresponding to the peak lateral tire force. Since β is not in the state space, $\underline{\overline{\beta}}$ is applied by bounding the lateral speed, v_y, as $\underline{\overline{v_y}} = v_x \tan^{-1}\left(\underline{\overline{\beta}}\right)$. The estimated road-tire friction coefficient, $\mu_{r,\mathrm{est}}$, is set naively as 0.8 to reflect dry road conditions. Therefore, in the case of poor road conditions, the yaw rate constraints will be too wide. The effect of adjusting the estimated friction coefficient to accurately reflect poor road conditions is therefore studied in Chapter 5.

4.3. Optimal Control Problem

4.3.1. Objective Function

The control objective considers several objectives with regard for path planning, path tracking, lateral stability, and steering effort. This includes penalizing tracking error with respect to the nominal path, which is the center of the desired driving lane. Additionally, the approximated potential field intensity is penalized. To improve vehicle stability and reduce unnecessary steering action, both the magnitude of the steering actions and the difference between each steering angle and the previous optimal steering angle, $\delta^*_{f,\mathrm{req},t-1}$, are penalized.

The optimal problem for lateral control is characterized by an objective function where the set of steering actions, $\{\delta_{f,\mathrm{req},t}, \dots, \delta_{f,\mathrm{req},t+N_c-1}\}$, and the set of constraint slack variables, $\{\epsilon_1, \dots, \epsilon_{N_p}\}$, are optimized. The control objective by which the control inputs are optimized is defined as

$$\min_{\delta_f,\epsilon} \sum_{k=1}^{N_p} \left(\left\| Y_{t+k} - Y_{des,t+k} \right\|_Q^2 + \left\| \delta_{f,t+k-1} \right\|_R^2 + G_{Y,t+k} + \|\epsilon_k\|_S^2 \right) + \|\delta_{f,\mathrm{req},t+k-1} - \delta_{f,\mathrm{req},t-1}\|_P^2,$$

(41)

where Q, R, S, and P are weighted identity matrices for the Y-coordinate tracking error, steering angle, difference between the current and previous steering angles, and the constraint slack variables, respectively. The desired Y-coordinate, Y_{des}, is the center of the desired driving lane.

4.3.2. Constraints

Certain vehicle states are constrained, such as those which affect vehicle stability, as outlined in 0. Additionally, the Y-coordinate, Y, is constrained according to the road boundaries. Within the MPC policy, the inequality constraint, $A_c u \leq b_c$, constrains the vehicle states, where

$$A_c = \begin{bmatrix} C_c \Theta \\ -C_c \Theta \end{bmatrix}$$

(42)

and

$$b_c = \begin{bmatrix} \bar{x} - C_c \Phi x_t \\ -\bar{x} + C_c \Phi x_t \end{bmatrix},$$

(43)

and where the constrained states, Y, v_y and r, are indexed in the state prediction by the block diagonal matrix, C_c. The upper and lower bounds of Y, v_y and r comprise $\overline{x}$ and $\underline{x}$ respectively as $\overline{x} = \begin{bmatrix} \overline{Y} & \overline{v_y} & \overline{r} & \dots & \overline{Y} & \overline{v_y} & \overline{r} \end{bmatrix}^\top$ and $\underline{x} = \begin{bmatrix} \underline{Y} & \underline{v_y} & \underline{r} & \cdots & \underline{Y} & \underline{v_y} & \underline{r} \end{bmatrix}^\top$.

Slack in the state constraints is necessary to avoid infeasibility while solving the quadratic programming problem. This slack allowance is characterized by a set of slack variables, $\epsilon_k = \left[\epsilon_{\overline{Y},t+k}\ \epsilon_{\underline{Y},t+k}\ \epsilon_{\overline{v_y},t+k}\ \epsilon_{\underline{v_y},t+k}\ \epsilon_{\bar{r},t+k}\ \epsilon_{\underline{r},t+k}\right]^\top$.

4.3.3. Quadratic Programming Formulation

The control objective function is solved using quadratic programing in MATLAB. To do so, the cost function in (41) is converted into the quadratic programming problem,

$$\tilde{J} = \frac{1}{2}\tilde{u}^\top \tilde{H}\tilde{u} + \tilde{f}^\top \tilde{u},$$

(44)

where the augmented control vector,

$$\tilde{u} = \left[\delta_{f,\mathrm{req},t} \quad \delta_{f,\mathrm{req},t+1} \quad \cdots \quad \delta_{f,\mathrm{req},t+N_c-1} \quad \epsilon_1 \quad \cdots \quad \epsilon_{N_p}\right]^\top,$$

(45)

comprises the set of steering actions for the control horizon and the set of slack variables for the state constraints. The Hessian matrix and gradient vector of $\tilde{J}$ are defined as

$$\tilde{H} = \begin{bmatrix} H & 0_{N_c \times 6N_p} \\ 0_{6N_p \times N_c} & S_{6N_p \times 6N_p} \end{bmatrix}$$

(46)

and

$$\tilde{f} = \begin{bmatrix} f \\ 0_{6N_p \times 1} \end{bmatrix},$$

(47)

respectively. The matrix, H, and the vector, f, characterize the quadratic problem in the absence of slack variables, as

$$H = 2[(C_Y\Theta)^\top Q C_Y\Theta + R + P] + \left(C_{X,Y}\Theta\right)^\top \nabla^2 G_Y|_{X_t,Y_t} C_{X,Y}\Theta$$

(48)

and

$$f^\top = 2\left[E^\top Q C_Y\Theta - P\delta_{f,t-1}\right] + \nabla G_Y|_{X_t,Y_t} C_{X,Y}\Theta + \Omega^\top \nabla^2 G_Y|_{X_t,Y_t} C_{X,Y}\Theta.$$

(49)

The block diagonal matrix, C_Y, indexes Y in the state vector, whereas $C_{X,Y}$ indexes X and Y in the state vector. Moreover, $E = C_Y\phi x_t - Y_{des}$ and $\Omega = C_{X,Y}\phi x_t - \Gamma$, where $\Gamma = [X_t\ Y_t]^\top$ extended over $2N_p$ rows. The objective, $\tilde{J}$, is subject to the augmented inequality constraint, $\tilde{A}_c\tilde{u} \leq b_c$, where

$$\tilde{A}_c = \begin{bmatrix} A_c & -I_{6N_p} \end{bmatrix}.$$

(50)

4.4. State Estimation

A state observer estimates non-measurable states according to the control inputs in u and the measurable state outputs in y. A Kalman filter is a particular type of state observer for a stochastic system, where measurement and process noises are considered [44]. Conversely, a Luenberger observer is a state observer for a deterministic system. To estimate some states in x which are typically unmeasurable on a production vehicle, such as the lateral speed, v_y, a pair of state observers are tested. At each discrete time step with step size, τ_s, the state, x, is estimated as $\hat{x}$, with the measurable output, $y = Cx = \left[X\ v_x\ Y\ \psi\ \omega_f\ \omega_r\ \delta_{f,act}\right]^\top$.

4.4.1. Kalman filter

For the Kalman filter, the first step of state estimation is to make the a priori prediction, $\hat{x}^-$, as

$$\hat{x}_t^- = A_d|_{x=\hat{x}_{t-1}}\hat{x}_{t-1} + B_d|_{x=\hat{x}_{t-1}}u_t + W_d|_{x=\hat{x}_{t-1}}$$

(51)

and

$$P_t^- = A_d|_{x=\hat{x}_{t-1}} P_{t-1} A_d^\top + Q_n,$$

(52)

where Q_n is the covariance matrix of the process noise, modeled as Gaussian white noise, and P_t^- is the a priori estimate covariance matrix. Next, the prediction, $\hat{x}^-$, is corrected to $\hat{x}$, and the estimate covariance, P^-, is corrected to P, according to

$$K_k = P_t^- C^\top (C P_t C^\top + R_n)^{-1},$$

(53)

$$\hat{x}_t = \hat{x}_t^- + K_k(y_t - C\hat{x}_t^-),$$

(54)

and

$$P_t = (I_9 - K_k C) P_t^-,$$

(55)

where K_k is the Kalman filter gain matrix and R_n is the covariance matrix of the sensor noise, modeled as Gaussian white noise. As the controller is run in simulation, there is no sensor noise present; therefore, the sensor noise covariance matrix is set to zero. The process noise covariance matrix is set to a diagonal matrix, which is then tuned by trial-and-error until the estimate, $\hat{x}$, converges to the actual full-state, x.

4.4.2. Luenberger observer

For the Luenberger observer, the state estimate is updated according to a simple feedback loop, as shown in Figure 4.3, where the estimation error, $y - \hat{y}$, is used to correct the estimate produced by the linear model as

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d u_k + G_d + L_d(y - C_d \hat{x}_k),$$

(56)

where L_d is the discrete-time observer gain matrix, which is tuned to set the eigenvalues of $(A_d - L_d C_d)$ to be inside the unit circle.

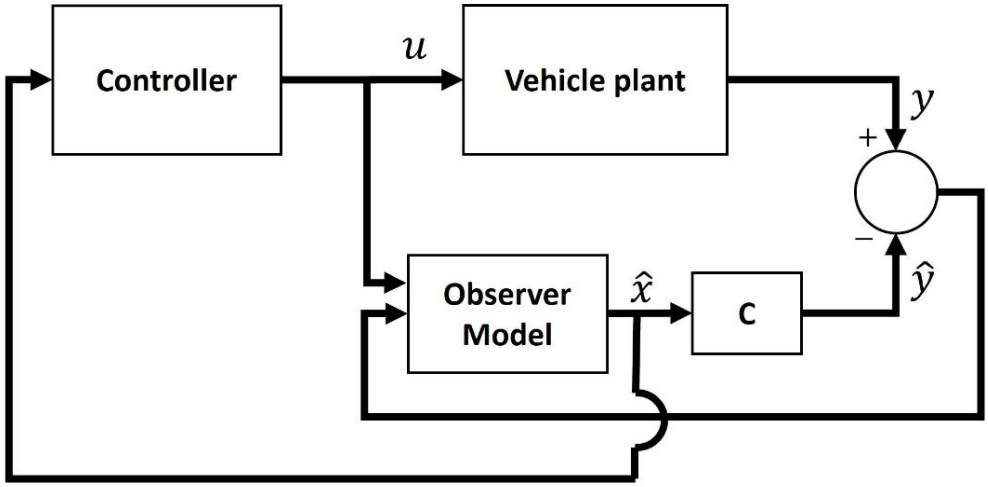

Figure 4.3: Luenberger observer block diagram

4.4.3. Comparison of observers

Figure 4.4 shows the results of the state observers when the lateral speed, v_y, is unmeasurable in the state output, y. In this scenario, the vehicle follows a double lane change maneuver while traveling at 100 km/h. It is shown that both the Luenberger and Kalman observers track the actual state trajectory with a high level of accuracy. However, the Kalman filter tracks the actual state more closely than the Luenberger observer due to the consideration for process noise with the former technique.

To assess the stability of the Luenberger observer, the eigenvalues of the matrix, $(A_d - L_d C_d)$, are recorded. The fourth eigenvalue, λ_4, corresponds to the estimate of the unknown state, v_y; during this maneuver, λ_4 has the highest magnitude of all eigenvalues, but does not exceed a magnitude of 0.664 after tuning L_d. Therefore, the full-state observation by this approach is stable.

The Kalman filter also exhibits a high level of stability and rapid convergence to the actual value of v_y. This is demonstrated by the posteriori estimate covariance matrix, P, all elements of which exhibit magnitudes no greater than 0.02 during the entire simulation. Therefore, the coefficient of variation (CV = σ/μ), is approximately 4% at the instances where the lateral speed is the greatest,

which indicates a high level of confidence in the state estimates provided by the Kalman filter, even during rapid steering maneuvers.

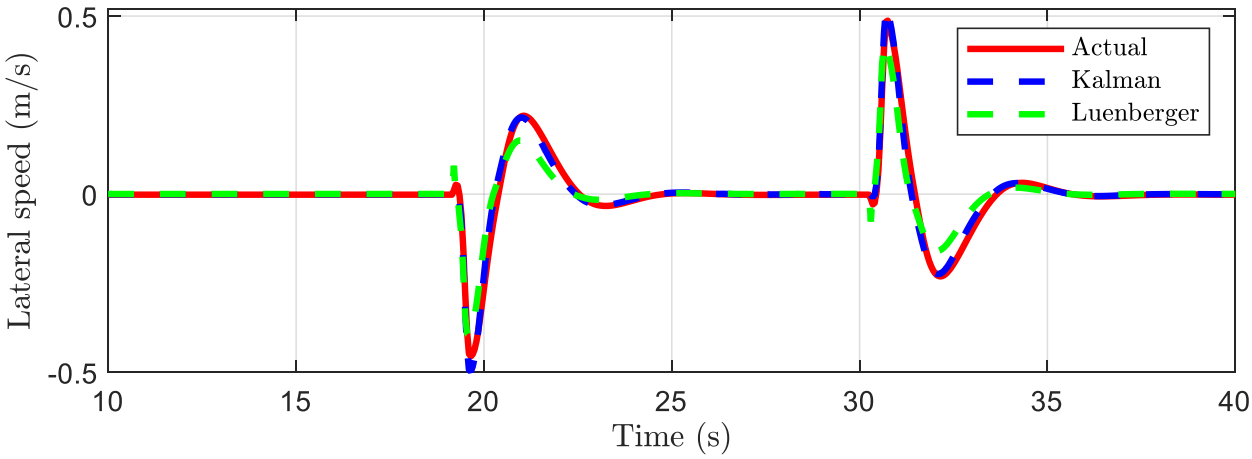

Figure 4.4: Lateral speed during a double lane change maneuver at 100 km/h, where the state is estimated by Kalman and Luenberger observers

4.5. Longitudinal Velocity Control

While MPC handles the lateral control associated with obstacle avoidance and lane-keeping, the longitudinal control strategy is independent. The objective of the longitudinal controller is to track the desired longitudinal speed, $v_{x,\text{des}}$. In the nominal case, where there are no obstacles in the vicinity of the ego vehicle, $v_{x,\text{des}} = v_{x,\text{max}}$, where $v_{x,\text{max}}$ is the speed limit. If an obstacle is within both safe distances, X_s and Y_s, the speed is reduced to avoid a collision. In this case, the desired speed is set according to the PF gradient resulting from the respective obstacle,

$$v_{x,\text{des}} = v_{x,\text{max}} \left(1 - k_X \frac{\left.\frac{\partial g_i^*}{\partial X}\right|_{X_t}}{\left.\frac{\partial g_i^*}{\partial X}\right|_{X_{s,t}}} \right), \tag{57}$$

where g_i^* is the PF approximation of the obstacle in question, k_X is a tunable factor of safety, and $v_{x,\text{max}}$ is the speed limit. By this policy, the longitudinal gradient of the PF is leveraged only for longitudinal control, while it is completely ignored in the lateral control policy. Therefore, the repulsive force from each PF is divided into its longitudinal and lateral components to exclusively affect the control policies for the respective directions.

The speed is controlled by a pair of low-level PID controllers which command the front axle torque, $T_{\text{axle},f}$, and brake pressure, P_{b}, as shown in Figure 4.5. Additionally, an anti-lock braking system (ABS) is included, which releases the brakes using a bang-bang controller if either the front or real longitudinal slip ratios decrease below -0.2.

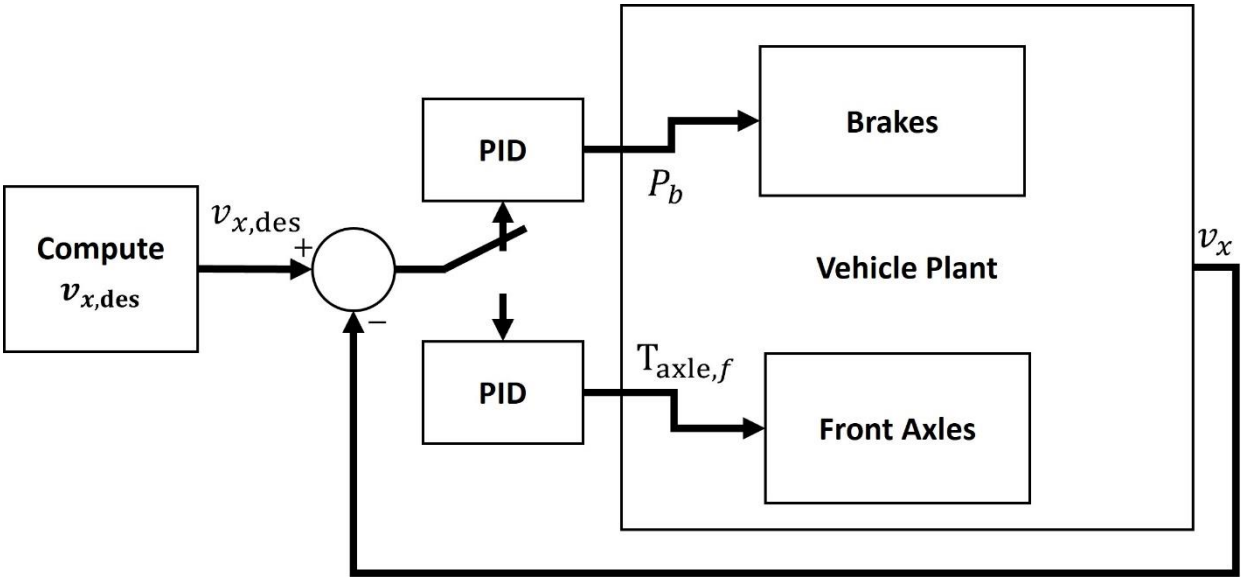

Figure 4.5: Block diagram of longitudinal control loop

4.6. Reference State Formulation

4.6.1. Localization on Path

The vehicle is located on a pre-recorded global path according to the method exhibited in Figure 4.6. The objective of this localization routine is to define the position of the vehicle in a path-based coordinate frame, with path-coordinate, s, and lateral displacement, Δ. In the diagram shown, the *Bus* represents the vehicle, where the CG location is denoted by the East and North coordinates, (E_B, N_B). The reference path, as shown in green, consists of a set of discrete waypoints, shown to be connected by imaginary line segments. The vehicle is localized in the path frame by select the nearest path waypoint in terms of Euclidian distance and checking whether an orthogonal vector, $\left(\frac{dN}{dE}\right)_{normal}$ exists on that segment with one end at the vehicle coordinates, (E_B, N_B) and such that

$$\left(\frac{dN}{dE}\right)_{road} \times \left(\frac{dN}{dE}\right)_{normal} = -1.$$

(58)

By this definition, the coordinates of the intersection point of the orthogonal vector and the path segment are

$$E_e = \frac{\left(\frac{dN}{dE}\right)_{road}\left[N_B + \left(\frac{dN}{dE}\right)_{road} E_1 - N_1\right] + E_B}{1 + \left(\frac{dN}{dE}\right)^2_{road}}$$

(59)

$$N_e = \left(\frac{dN}{dE}\right)_{road} (E_e - E_1) + N_1$$

(60)

The one-dimensional path coordinate of the vehicle, s_b, is therefore defined as the known path coordinate of the waypoint, (E_1, N_1) plus the Euclidian distance between (E_1, N_1) and (E_e, N_e).

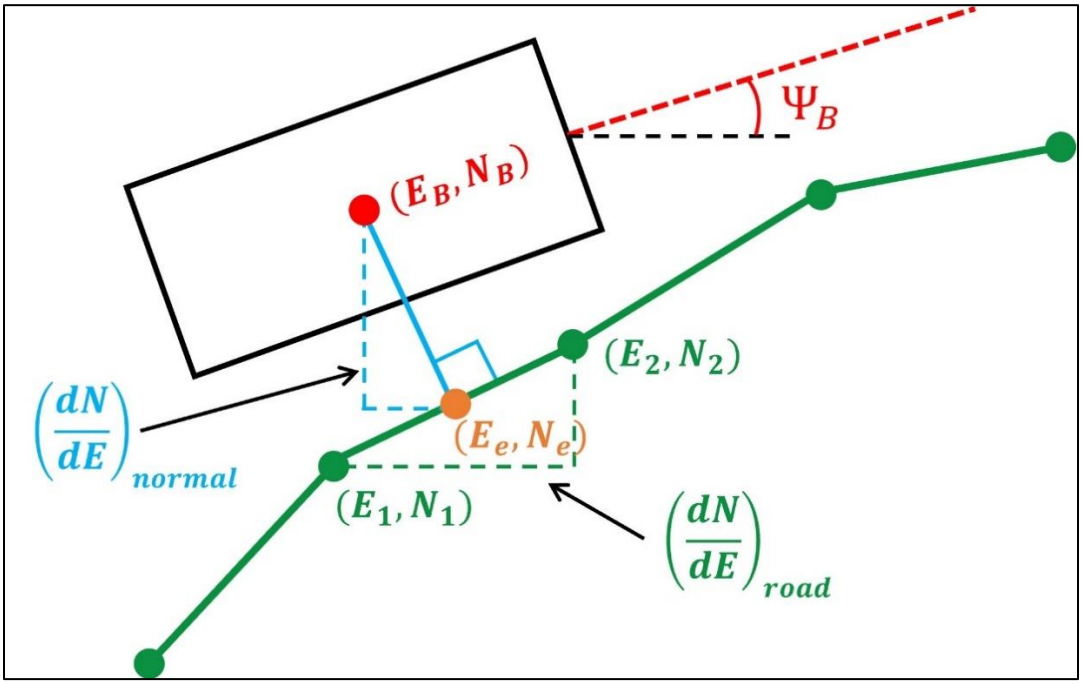

Figure 4.6: Localization on 1-dimensional path coordinate

In an iterative routine, the validity of the localization step is executed by checking whether the resultant orthogonal vector lies between the points (E_1, N_1) and (E_2, N_2); the orthogonal vector may intersect an extrapolation of this path segment in either direction. The localization routine is repeated by incrementally checking each path segment in the appropriate direction until an orthogonal vector is computed which lies within the bounds of the respective path segment.

4.6.2. Reference Coordinates and Heading

After localizing the vehicle on the path coordinate, the reference trajectory for the prediction horizon is constructed. Each k^{th} discrete step is assigned the reference coordinates, $X_{des,t+k}, Y_{des,t+k}$, and the reference heading angle, $\psi_{des,t+k}$. These reference values are determined according to the estimated waypoint locations at each predicted step. For each step, the corresponding path waypoint is defined as

$$n_{t+k} = n_{(E_2,N_2)} + \text{floor}\left(\frac{\tau_s v_x k}{d}\right) \tag{61}$$

where $n_{(E_2,N_2)}$ is the index of the next closest path waypoint after the vehicle, as shown in Figure 4.6, and d is the Euclidian distance between each waypoint in the reference path. By this estimation, the vehicle speed is assumed to remain constant throughout the prediction horizon, and the vehicle is assumed to track directly on the reference path. Thereby, rough approximations of the desired coordinates and heading for each k^{th} step in the prediction horizon are given as

$$X_{des,t+k} = E(n_{t+k}), \tag{62}$$

$$Y_{des,t+k} = N(n_{t+k}), \tag{63}$$

and

$$\psi_{des,t+k} = \psi(n_{t+k}), \tag{64}$$

where $\psi(n_{t+k})$ is the road heading angle, ψ_r, at the $(n_{t+k})^{th}$ path waypoint, as evaluated by the mean of the two path segment headings adjacent to the respective point, as

$$\psi(n_{t+k}) = \text{mean}\left(\text{atan}\left(\frac{N(n_{t+k}+1) - N(n_{t+k})}{E(n_{t+k}+1) - E(n_{t+k})}\right), \text{atan}\left(\frac{N(n_{t+k}) - N(n_{t+k}-1)}{E(n_{t+k}) - N(n_{t+k}-1)}\right)\right). \quad (65)$$

CHAPTER 5

Simulation Results

A virtual test environment is constructed within MATLAB and Simulink to validate the performance of the control algorithm described in Chapter 4. This simulation environment features a medium-fidelity dynamics model to represent the vehicle plant. Additionally, driving scenarios are generated and integrated with the vehicle plant; detections of actor vehicles and lane boundaries are output from the scenario and read by the controller.

5.1. Simulation Environment

5.1.1. Vehicle Model

To validate the controller performance, a simulated vehicle plant is designed using the MATLAB & Simulink Vehicle Dynamics Blockset. As shown in Figure 5.1, a vehicle body block is used in conjunction with four wheel and tire blocks. The states vector, x, is read directly by the controller, in addition to the tire normal forces, F_{zf} and F_{zr}, and effective rolling radius, R_{eff}. The vehicle body parameters, including m, I_z, l_f, and l_r, are defined in Table 5.1. The tire model parameters are set to the default values for Light Passenger Car 205/60R15 within the Combined Slip 2DOF Wheel blocks. The road surface friction for the plant is set by adjusting the lateral and longitudinal friction coefficients, P_{DY1} and P_{DX1}, respectively, from the Pacejka Magic formula [35]. The control inputs, including the front steering angle, $\delta_{f,\text{req}}$, axle torque to each front wheel, $T_{\text{axle},f}$, and brake pressure, P_{b}, are input to the appropriate wheel blocks in the vehicle plant.

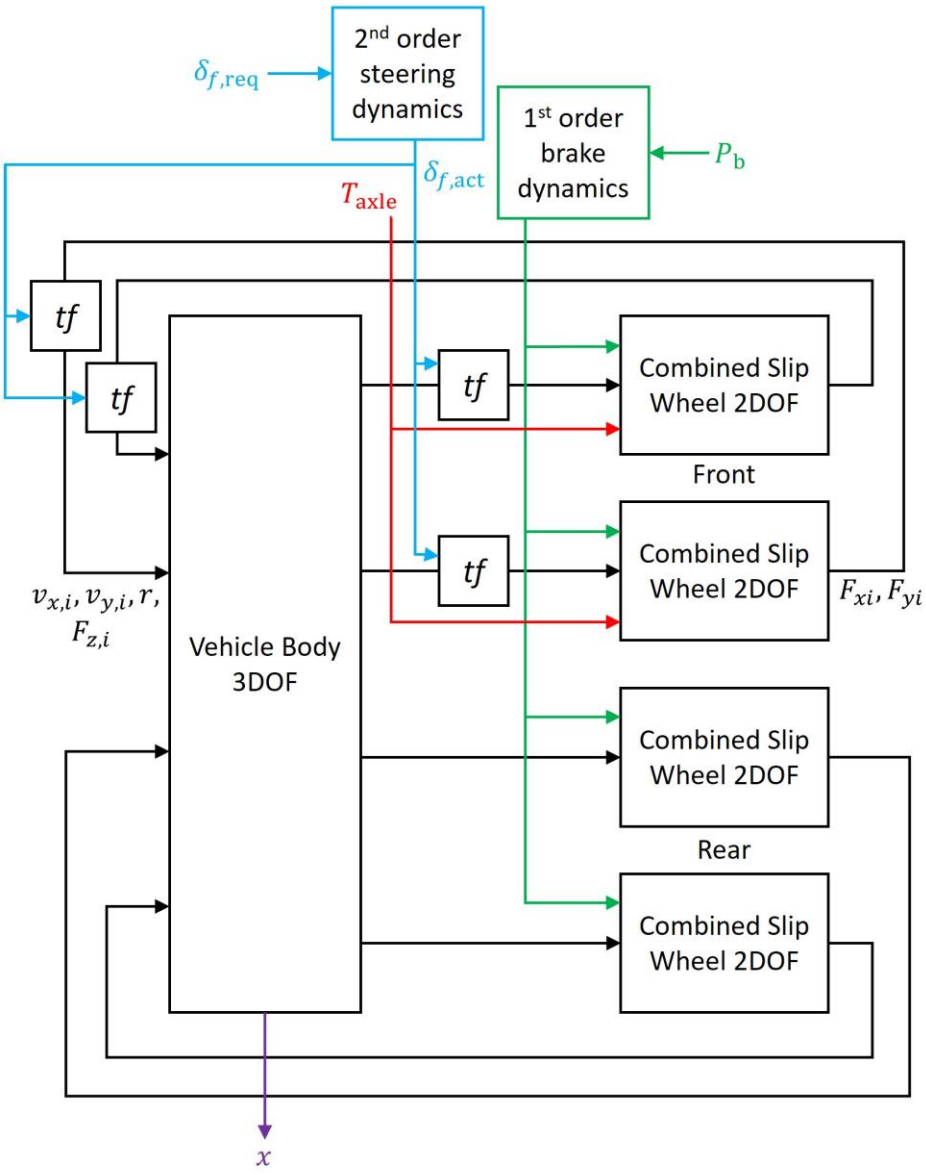

Figure 5.1: Block diagram of vehicle model, featuring a vehicle body dynamics model and four wheel and tire models which consider the combined slip effect on tire forces.

The tire model blocks (Combined Slip Wheel 2DOF) use the Pacejka tire formula [35] and consider the combined-slip effect. The Pacejka tire formula and the Burckhardt model, the latter of which estimates the tire forces in the MPC prediction model, exhibit similar behavior for a variety of road conditions [36]. Therefore, the reference model can reasonably represent the plant tire dynamics, assuming the model parameters are tuned properly. Figure 5.2 shows the lateral force generated from each model, both in low- and high-longitudinal slip conditions. For this comparison, the Burckhardt model is tuned by least-squares (LS) fitting to 205/55R16 tire data at

$F_z = 4045.24$ N. The advantage of using a combined-slip model is exhibited, as the pure-slip model significantly over-estimates the lateral tire force in the case of high longitudinal slip (Figure 5.2(b)). Moreover, the Burckhardt combined-slip model is shown to accurately estimate the lateral tire forces in a range of slip conditions. However, the difference between the two models is significant at very high slip angles (> 20 degrees).

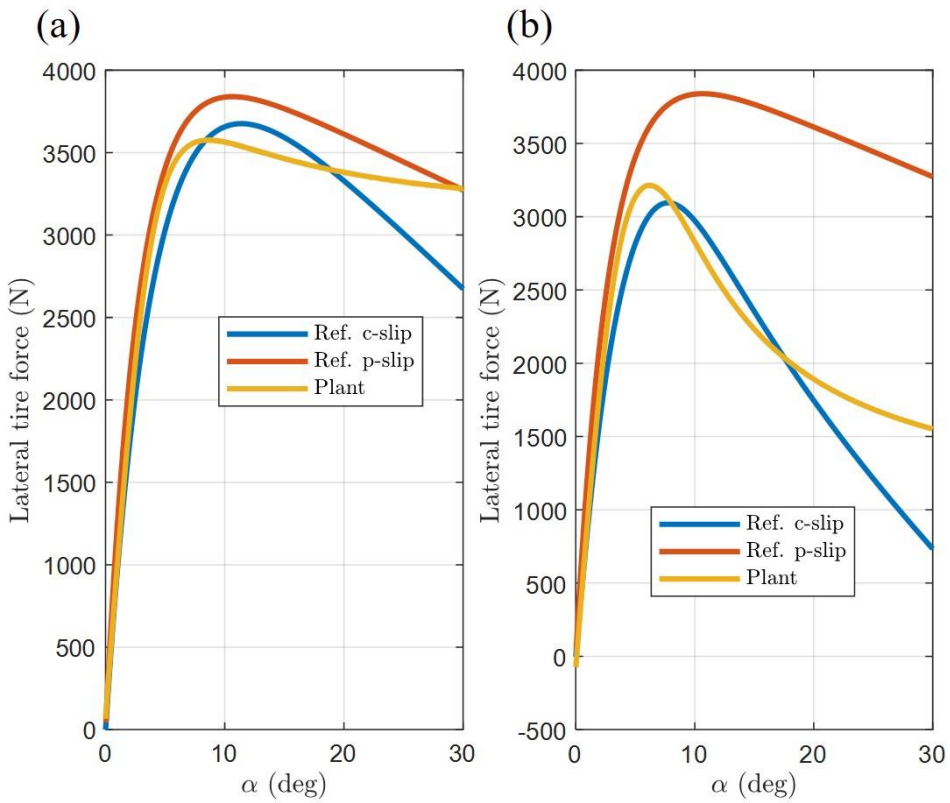

Figure 5.2: Estimated lateral tire forces for combined- and pure-slip Burckhardt models, compared to those generated by the Pacejka model. (a) Low slip ratio: 0.003. (b) High slip ratio: 0.3.

First order braking dynamics are included as part of the plant model with a time constant, τ_b, as defined in Table 5.1. The steering dynamics are modeled with a second order transfer function,

$$\delta_{f,\text{act}} = \delta_{f,\text{req}} \frac{\omega_{n,s}^2}{s^2 + 2\omega_{n,s}\zeta_s s + \omega_{n,s}^2} \quad (66)$$

where $\omega_{n,s}$ and ζ_s are the natural frequency and damping ratio of the steering system, respectively, as defined in Table 5.1. Additionally, the steering rate of change saturates at $\left|\frac{d\delta_{f,\text{act}}}{dt}\right| \leq 1$ deg/s.

Table 5.1: Simulated vehicle model parameters

Symbol	Description	Value	Units
m	Mass	1270	kg
I_z	Yaw moment of inertia	1536.7	kg m^2
l_f	Front axle to CG	1.015	m
l_r	Rear axle to CG	1.895	m
I_w	Wheel rolling moment of inertia	1.084	kg m^2
R_e	Effective tire rolling radius	0.325	m
$\omega_{n,s}$	Steering natural frequency	$\frac{2\pi}{0.04}$	rad/s
ζ_s	Steering damping ratio	1	-
τ_b	Braking time constant	0.01	s

5.1.2. Scenario Design

The vehicle model is simulated in various driving scenarios generated using the MATLAB Driving Scenario Designer application. Figure 5.3 shows an example of a 2-dimensional driving scenario, where the ego vehicle is shown in blue, and an actor vehicle is shown in orange. In this scenario, a three-lane roadway is also included. The trajectory of the actor vehicle is set in the scenario designer as a set of waypoints with a pre-determined velocity at each point. The ego vehicle is given an initial pose in the scenario designer, but its trajectory is determined based on the controller input during simulation. The scenario is also shown in a 3-dimensional environment which uses the Unreal gaming engine to generate realistic graphics of the environment.

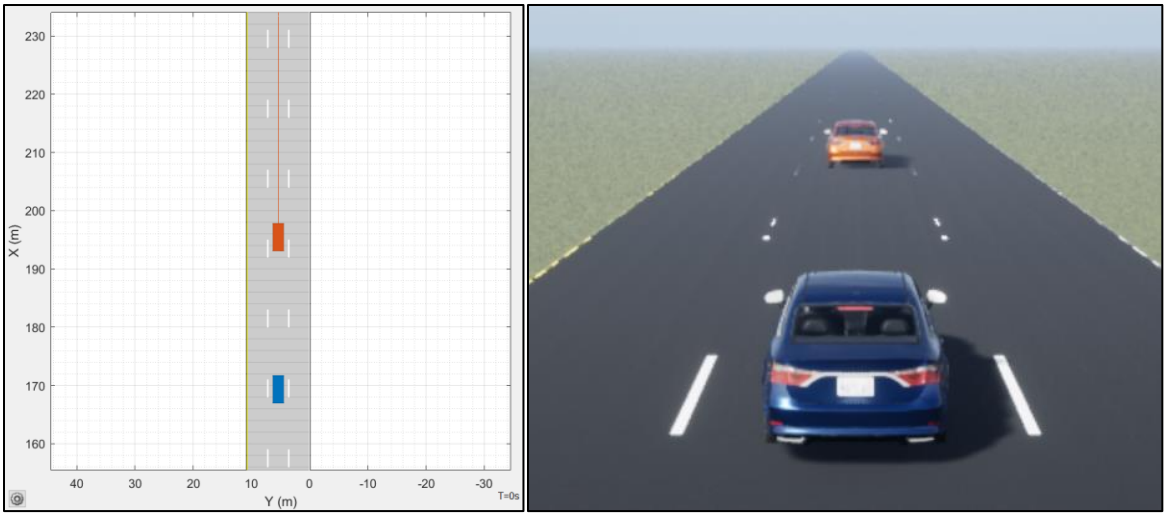

Figure 5.3: Example of driving scenario generated using MATLAB Driving Scenario Designer

In the simulation environment, features such as lane boundaries and obstacles are detected by virtual sensors on the ego vehicle. These sensors are configured using the Driving Scenario Designer tools provided in MATLAB. Lane boundaries are detected by a simulated vision sensor for a specified horizon ahead of the ego vehicle. Obstacles such as actor vehicles are detected by a simulated radar sensor.

5.2. Simulation Results

5.2.1. Obstacle avoidance

Figure 5.5 shows the controller performance when avoiding a vehicle in its driving lane while traveling at 80 km/h. In Figure 5.5(a), the vehicle to be avoided is stationary; in Figure 5.5(b), the vehicle to be avoided is traveling at 50 km/h. It is shown in both scenarios that the vehicle avoids the obstacle by performing a lane change maneuver. Particularly, Figure 5.4 shows the ego vehicle trajectory to smoothly navigate around the potential field of the stopped vehicle. Moreover, each maneuver is comfortable and stable, as exhibited by low side-slip angles (Figure 5.5 (e) and Figure 5.5(f)). This highlights the ability of the controller to perform avoidance maneuvers at high speeds while maintaining rider comfort and vehicle stability.

Both avoidance maneuvers shown in Figure 5.5 exhibit a very low longitudinal slip ratio due to high road surface friction and low axle torque input. Therefore, the combined-slip effect is negligible in both scenarios. Figure 5.5(g) and (h) show the lateral tire forces estimated by the

MPC reference model, compared to the actual forces, for the avoidance maneuvers featuring static and moving vehicles, respectively. The actual and estimated forces follow a very similar trend throughout the maneuver, highlighting good tire force estimates in pure lateral slip conditions. However, at the point where the tire slip angles are the highest, the lateral forces are generally overestimated. This discrepancy is a result of model mismatch due to sub-optimal tuning of the Burckhardt parameters.

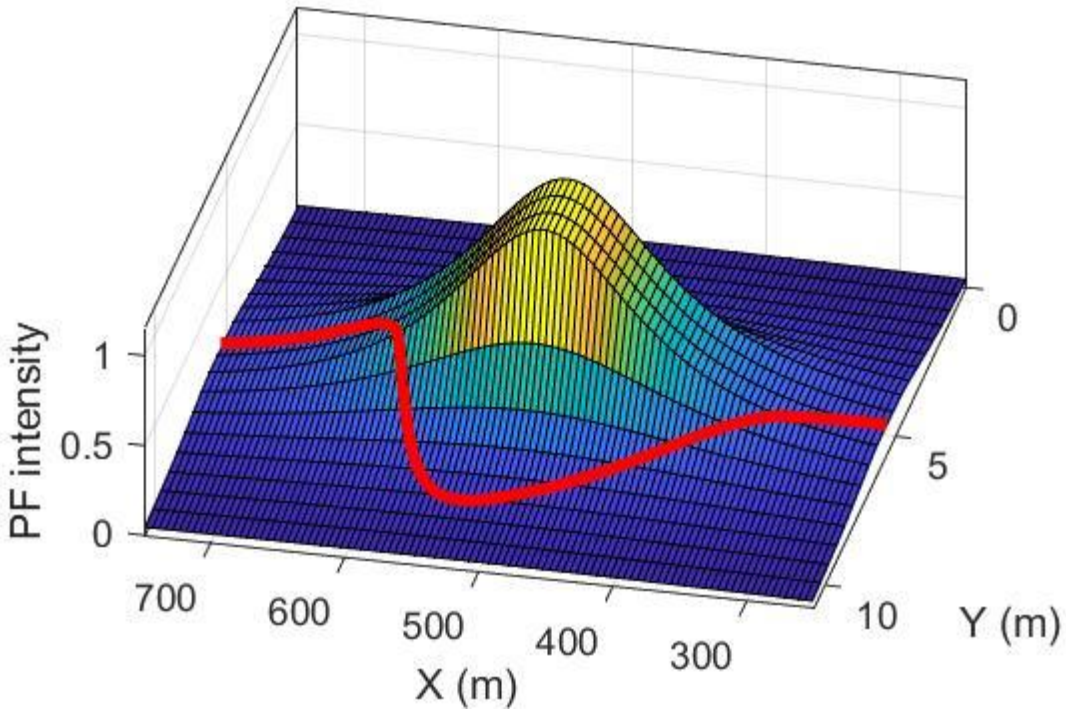

Figure 5.4: Potential field intensity and ego vehicle trajectory

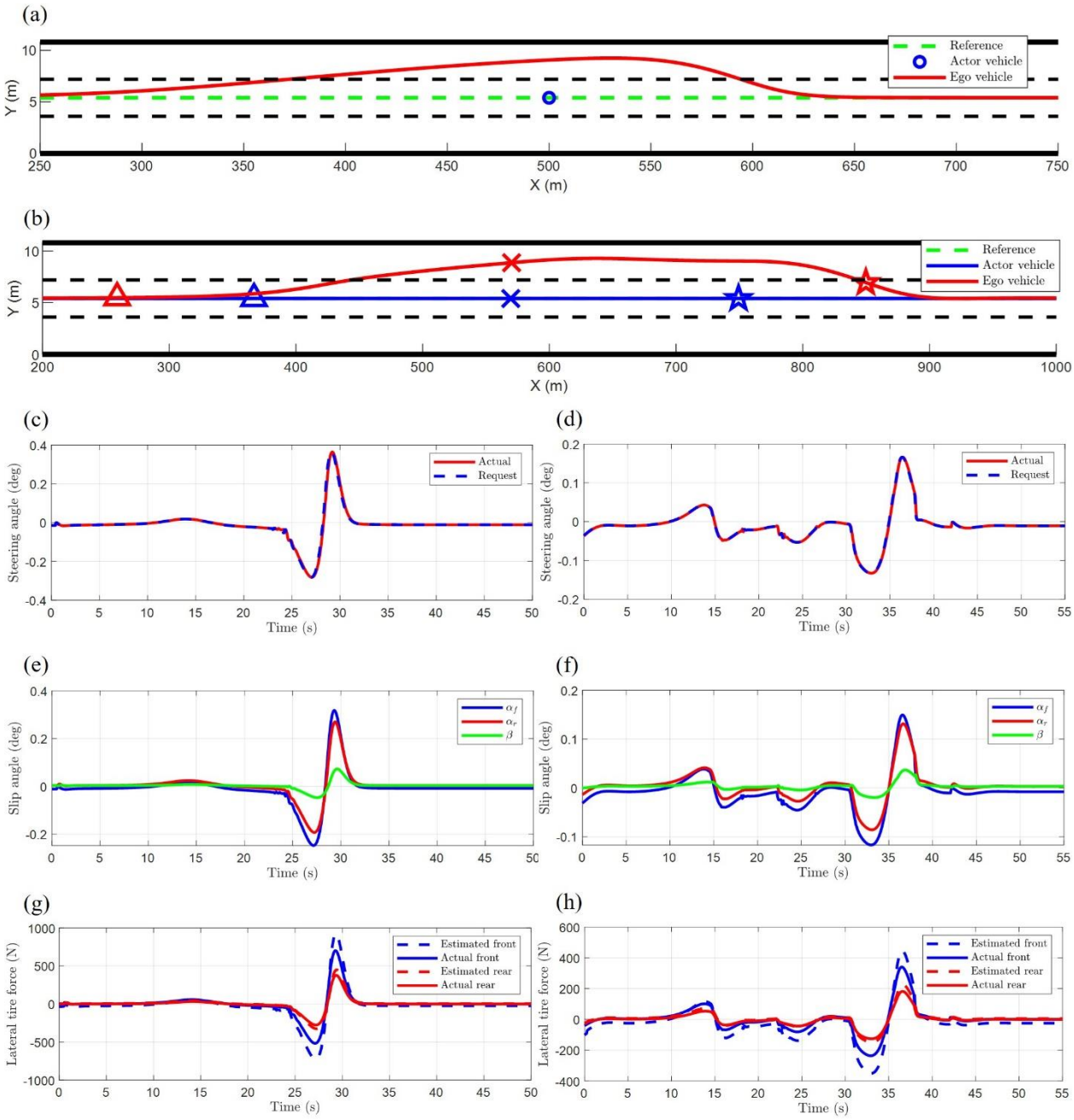

Figure 5.5: (a, c, e, g) Avoidance of a static vehicle while traveling at 80 km/h in dry road conditions. (b, d, f, h) Traveling at 80 km/h, avoiding a vehicle traveling at 50 km/h. (a, b) Trajectories of ego vehicle and actor vehicle. (c, d) Steering angle. (e, f) Side-slip angle, β, and tire slip angles, α_f and α_r. (g, h) Actual lateral tire forces compared to those estimated by reference model.

5.2.2. Stopping for Obstacles

Figure 5.6 shows the controller performance when slowing from 80 km/h to 50 km/h to follow the actor vehicle ahead, and then stopping completely behind that vehicle, according to the speed control policy in (57). Figure 5.6(a) shows that the ego vehicle comes to a complete stop approximately 6 m behind the actor vehicle. Moreover, it is shown in Figure 5.6 (c) that the desired speed profile is smooth as it converges toward the speed of the actor vehicle, at both 50 km/h and 0 km/h. This indicates that the longitudinal control policy is effective in safely following and stopping at a safe distance from obstacles. The longitudinal slip ratio in the stopping maneuver is very low, even when braking to stop for the actor vehicle, as shown in Figure 5.6(d). The low slip ratio is due to high road surface friction and low axle torque and brake pressure (Figure 5.6(b)). This results in good rider comfort and vehicle stability. Moreover, Figure 5.6(e) shows the estimated longitudinal tire forces during the stopping maneuver. The estimated tire forces closely match the actual longitudinal tire forces, indicating accuracy on the part of the Burckhardt model in pure longitudinal slip conditions.

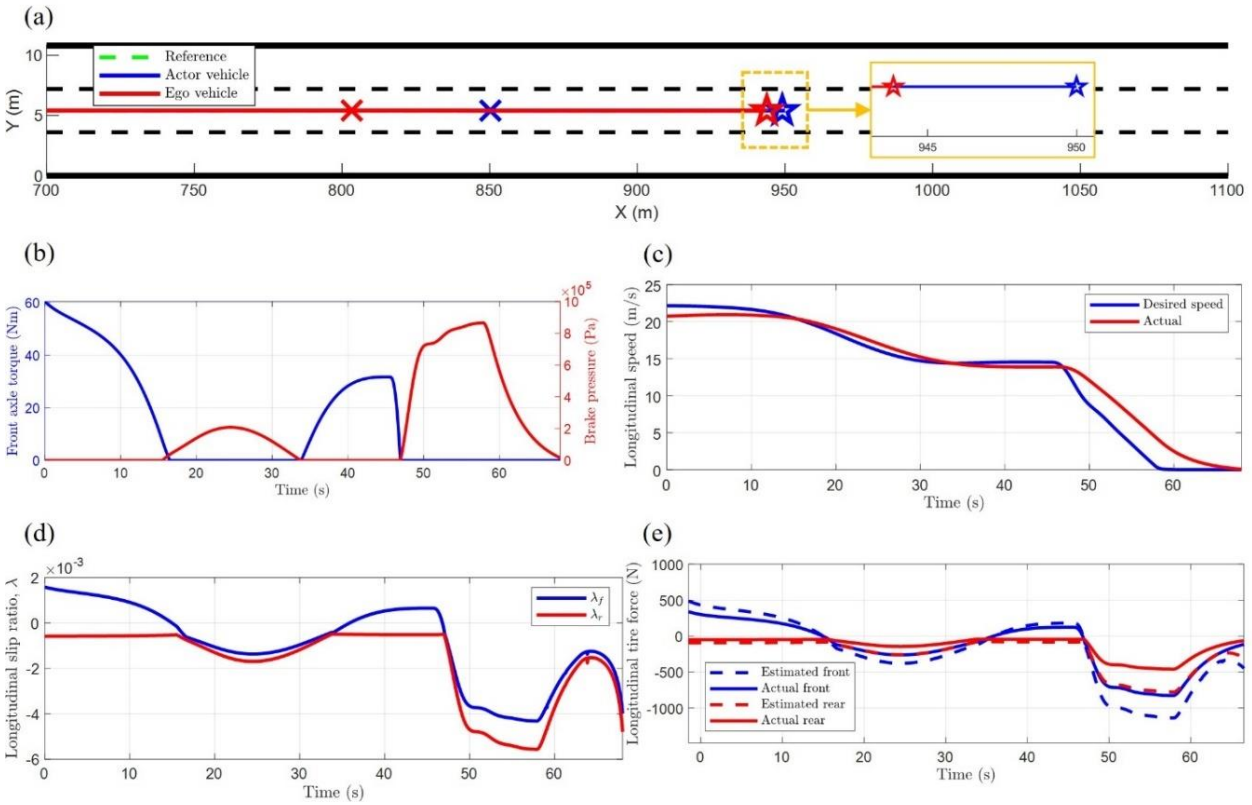

Figure 5.6: Traveling at 80 km/h, slowing to 50 km/h for vehicle ahead, then stopping for vehicle ahead in dry road conditions ($\mu_{r,\mathrm{est}} = 0.8$). The actor vehicle begins decelerating to a stop at X = 850 m and comes to a full stop at X = 950 m.

5.2.3. Evasive maneuver

The capability of the path planning controller to avoid a collision are tested under conditions which feature high lateral and longitudinal tire slip. Figure 5.8 demonstrates a harsh avoidance maneuver in wet road conditions with road surface friction, $\mu_{\mathrm{r}} = 0.5$. In this scenario, the ego vehicle follows behind an actor vehicle at 100 km/h until the actor vehicle stops abruptly. To avoid a collision, the path planning controller and speed controller are both enabled, resulting in a combination of avoidance and slowing.

It is shown in Figure 5.8(a) that the ego vehicle successfully performs a left lane change to avoid the stopped vehicle. Despite the low road surface friction, the vehicle remains stable, as exhibited by the low side-slip angle shown in Figure 5.8(f). Furthermore, although a high longitudinal acceleration is exhibited due to harsh braking, the lateral acceleration remains under 1 m/s^2 throughout the maneuver, as shown in Figure 5.8(e). Moreover, the lateral tire forces, as estimated by the reference model, maintain good accuracy, as shown in Figure 5.8(g). The lateral tire forces

are generally overestimated due to the low road surface friction; however, the estimates follow an analogous trend to the actual forces throughout. This result demonstrates good performance on the part of the path planning controller in poor road conditions, where the tire forces are overestimated. The desired longitudinal speed decreases as the ego vehicle approaches the vehicle ahead, as shown in Figure 5.8, according to (57).Once the vehicle has achieved the safe lateral distance, Y_s, the ego vehicle resumes driving at $v_{x,\max}$. The speed control is achieved by regulating axle torque, $T_{\text{axle,f}}$, and brake pressure, P_{b}. As shown in Figure 5.8(c), the brake pressure increases sharply to decelerate the vehicle. The longitudinal slip ratios of both front and rear tires, therefore, exhibit values approaching -6% during maximum braking, as shown in Figure 5.8(g). This result demonstrates the importance of considering the combined-slip effect for estimating tire forces, particularly in harsh braking maneuvers where the longitudinal slip ratio is high. Moreover, the longitudinal control law is shown to work well in collaboration with the path planning controller to maintain vehicle stability during evasive maneuvers.

Figure 5.7 shows the side-slip angle and yaw rate trajectory for the emergency avoidance maneuver presented in Figure 5.8. The effect of the estimated tire friction coefficient, $\mu_{\text{r,est}}$, is studied by changing it from the naive estimate of 0.8, to the correct road friction coefficient, 0.5. Both trajectories stay within the strictest constraint according to $\mu_{\text{r,est}} = 0.5$. This result exhibits the capability of the controller to maintain vehicle stability despite low road surface friction. Moreover, the two trajectories are virtually identical. This demonstrates that a naive estimate of the road surface friction is acceptable, even when road conditions are actually poor; therefore, online road surface friction estimation is not necessary for the purpose of calculating state constraints.

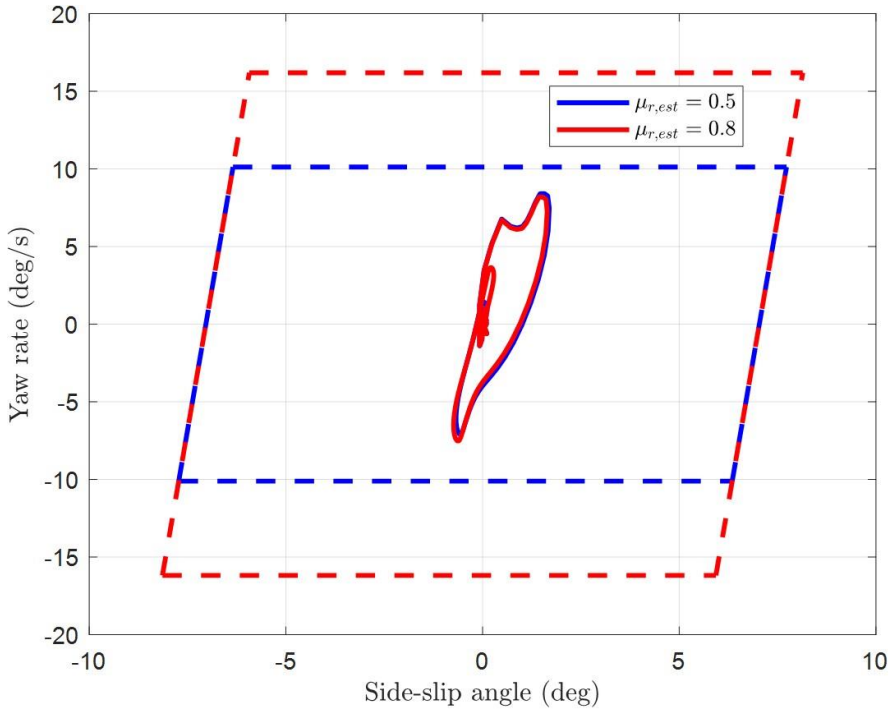

Figure 5.7: Phase portrait of yaw rate and side-slip angle for emergency avoidance maneuver

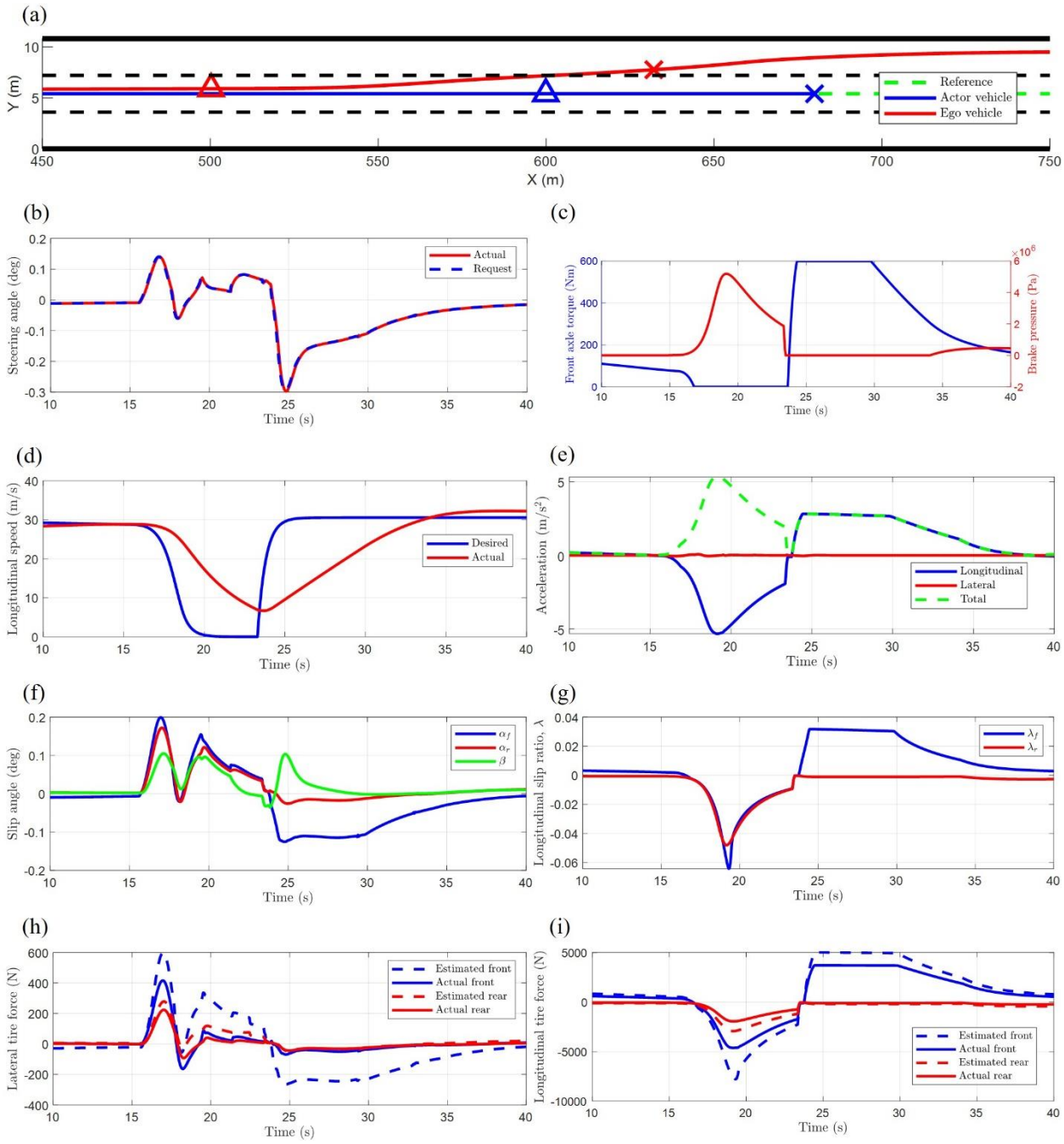

Figure 5.8: Following vehicle at 100 km/h, followed by the actor vehicle stopping within a distance of 80 m. Road conditions are wet (μ_r = 0.5). (a) Trajectories of ego and actor vehicles. The triangle markers denote the time when the actor vehicle begins decelerating to a stop and the 'X' markers denote the time when the actor vehicle comes to a full stop. (b) Requested and actual front steering angles. (c) Longitudinal control inputs, including front axle torque and brake pressure. (d) Desired and actual longitudinal speeds. (e) Lateral, longitudinal, and total resultant acceleration. (f) Side-slip angle and slip angles of front and rear tires. (g) Longitudinal slip ratios of front and rear tires. (h) Actual and estimated lateral tire forces. (i) Actual and estimated longitudinal tire forces.

.

CHAPTER 6

Experimental Results

6.1. Vehicle Platform

Experimental validation of the controller and motion planner described in Chapter 5 is performed on a test vehicle – a Chevrolet Equinox SUV equipped with independent four-wheel electric drive and automatic steering, as shown in Figure 6.1. The dynamics and kinematics parameters of the vehicle are shown in Table 6.1, each of which are set accordingly within the controller reference model. No modeling noise is considered in the model; therefore, the parameters of the reference model are assumed to match those of the actual vehicle plant.

Figure 6.1: Chevrolet Equinox used for experimental validation of motion controller

The motion controller is run in MATLAB/Simulink with a CAN interface for receiving state signals and sending control signals. An onboard global navigation satellite system (GNSS) reports the vehicle coordinates, velocity, heading angle, yaw rate, and acceleration. Moreover, speed sensors report the angular velocity of each wheel. The control signals, including the front steering angle, $\delta_{f,\mathrm{req}}$, and the front axle torque, T_{axle}, are sent by the controller to be activated by the steering actuator and the front wheel motors, respectively. No states are estimated in the

experimental trials, nor is sensor noise accounted for in the controller. Therefore, it is assumed that the noise from each sensor is negligible.

Table 6.1: Chevrolet Equinox parameters

Symbol	Description	Value	Units
m	Mass	2271	kg
I_z	Yaw moment of inertia	4600	kg m^2
l_f	Front axle to CG	1.43	m
l_r	Rear axle to CG	2.855	m
I_w	Wheel rolling moment of inertia	1.084	kg m^2
R_e	Effective tire rolling radius	0.325	m

6.2. Scenario Design

A virtual driving scenario is designed according to a real-world test environment to test the motion controller on the physical vehicle. Figure 6.2 shows an overhead view of the test environment in a parking lot on the University campus. To set up the scenario, the vehicle is driven in a straight line while the coordinates from the GNSS are recorded. Then, the recorded points are used to construct the reference path, as shown. Furthermore, this reference path is imported into the MATLAB Driving Scenario Designer to construct a driving scenario with three lanes, where the middle lane corresponds to the reference path. In this driving scenario, an actor vehicle is also added to represent an obstacle to be avoided. The three-lane roadway represents the area that is considered acceptable for avoidance – the ego vehicle may avoid on either side of the obstacle and subsequently return to the center lane. The resultant scenario is shown overlaid on the map in Figure 6.2.

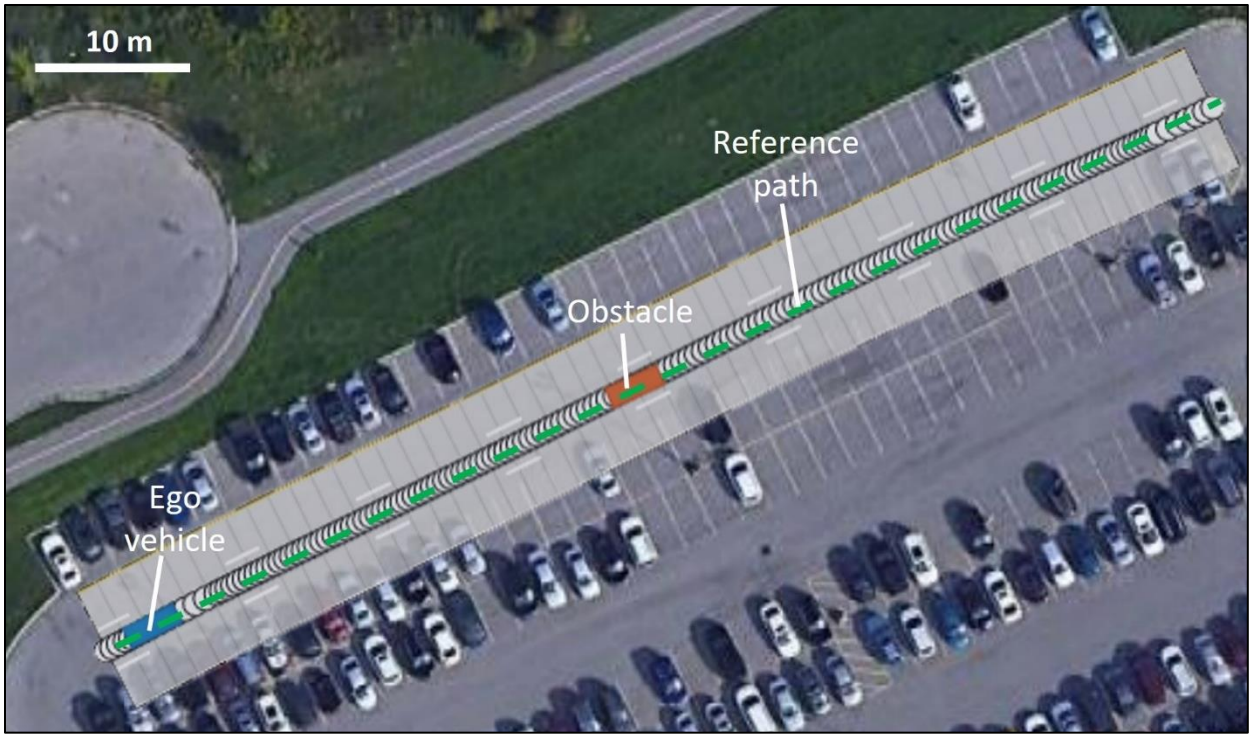

Figure 6.2: Obstacle avoidance and path tracking scenario in parking lot

6.3. Results

6.3.1. Path tracking

The first experimental test aims to validate the path tracking performance of the motion controller in the absence of any obstacles. In this test, the ego vehicle is parked at the beginning of the reference path and then the controller is activated. Meanwhile, a human operator controls the motor torque to achieve a longitudinal speed of approximately 20 km/h. Only the lateral controller is activated in this case to achieve path tracking by steering while cruising at a constant speed. Figure 6.3 shows the path tracking result, where the ego vehicle trajectory exhibits good tracking of the reference path. The steering angle request is smooth and tracked closely by the actual steering angle, as shown in Figure 6.3(b). Moreover, Figure 6.3(e) shows the lateral tracking error, which converges toward zero after correcting for the error due to the initial positioning of the vehicle. This result demonstrates the ability of the controller to track the vehicle along a reference path while commanding a steering angle which is achievable by real steering actuator dynamics. Furthermore, it demonstrates that the lateral position in the lane may be held even when the speed is not constant due to the accelerator input by a human operator.

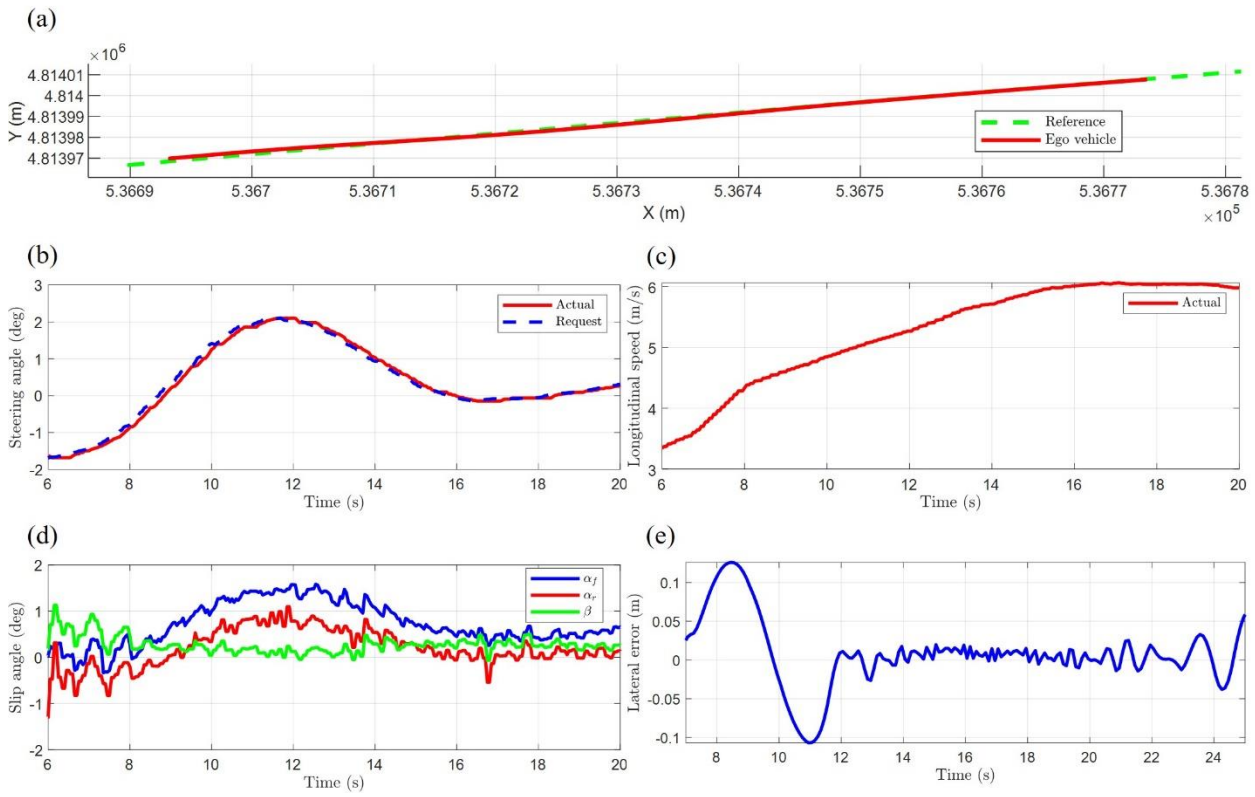

Figure 6.3: Path tracking performance on Chevrolet Equinox. (a) Trajectory and reference path. (b) Requested and actual steering angles. (c) Longitudinal speed. (d) Side-slip angle and tire slip angles. (e) Lateral tracking error.

6.3.2. Obstacle avoidance

The driving scenario shown in Figure 6.2 is executed to demonstrate the obstacle avoidance performance of the motion controller, where a virtual obstacle is placed on the reference path. In this case, the ego vehicle is parked at the beginning of the reference path and then the lateral controller is activated while a human operator controls the acceleration to achieve a speed of approximately 27 km/h. In this case, the longitudinal control is not important, as only the steering will be affected by the PF from the obstacle. Figure 6.5 shows the resultant behaviour while avoiding the obstacle: the trajectory is shown to first follow the reference path before deviating to the right of the path around the obstacle, and then finally, returning to the reference path. Furthermore, Figure 6.4 shows the 3D contour of the PF, which demonstrates that the ego vehicle trajectory circumvents the peak of the PF emanating from the obstacle and returns to the path where the PF intensity is significantly lower. This trajectory reflects the simulation results presented in 5.2.1, albeit at a much lower speed than the simulated maneuver.

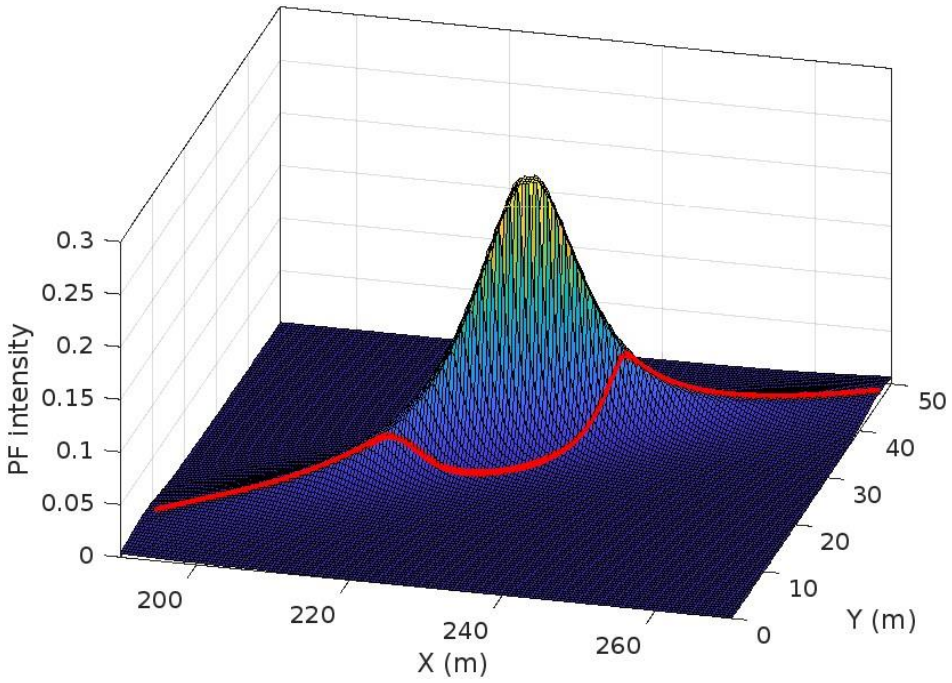

Figure 6.4: Potential field intensity surrounding obstacle, where the ego vehicle trajectory is indicated by the red line.

The successful obstacle avoidance is the result of the steering request shown in Figure 6.5(b), where the steering angle alternates between consecutive right-, left-, and right-hand turns, with a maximum steering angle of approximately 8 degrees. The steering angle request is tracked closely, but some lag is observed, particularly when switching from a right-hand turn to a left-hand turn after deviating to the right side of the obstacle. Furthermore, the steering angle saturates at 8 degrees during the left-hand turn due to hard constraints set on the steering angle. This angle constraint is set to prevent unsafe behavior due to a harsh steering angle. When compared to the simulation result shown in 5.2.1, the steering request herein is much higher and changing from one direction to another more sharply. This rapid shift in the steering angle request may be caused by lag in the steering actuator, as well as a lower speed which allows for a higher steering angle without violating the stability constraints. Ultimately, this steering request results in a smooth and comfortable trajectory.

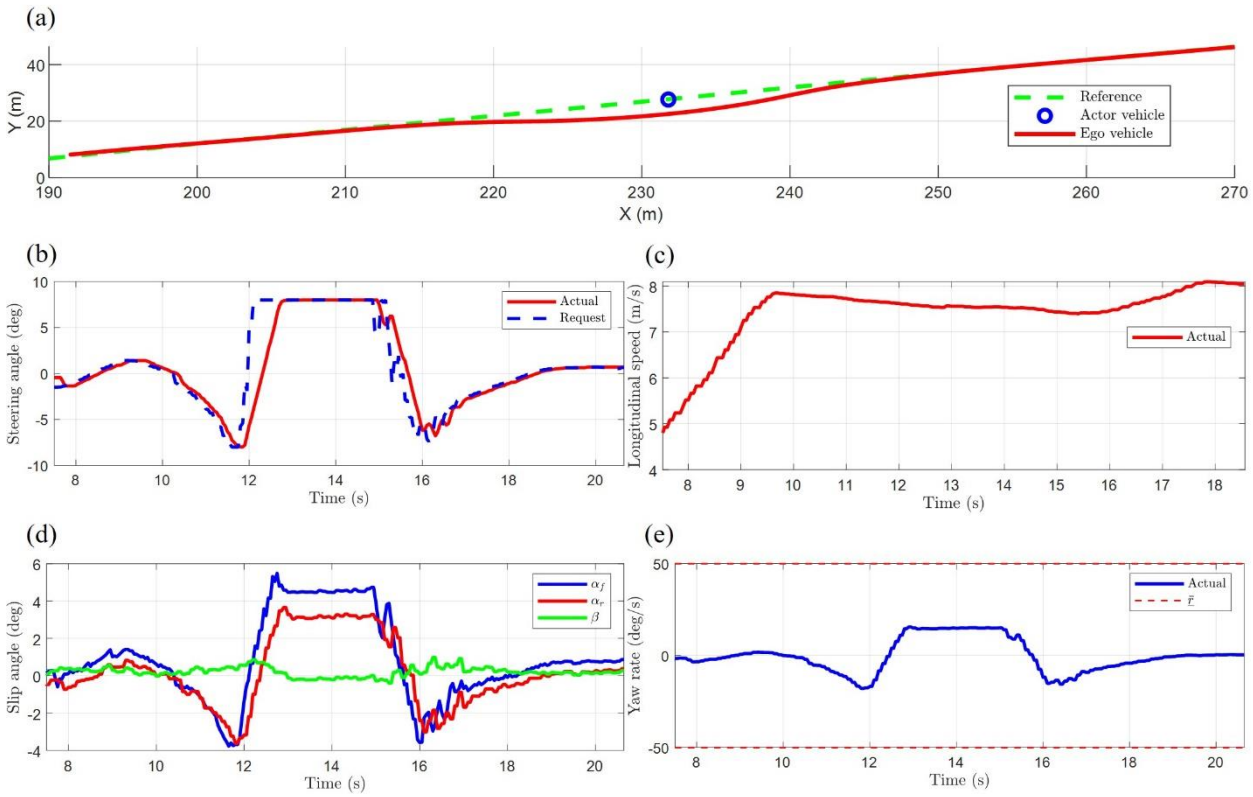

Figure 6.5: Result of obstacle avoidance maneuver on Chevrolet Equinox. (a) Trajectory of ego vehicle and location of obstacle. (b) Actual and requested steering angle at the front wheels. (c) Longitudinal speed. (D) Side-slip angle and tire slip angles. (e) Yaw rate and the constraints thereof.

During the maneuver, rider comfort and vehicle stability are maintained, as exhibited by a low side-slip angle and yaw rate, as shown in Figure 6.5(d) and Figure 6.5(e), respectively. Furthermore, Figure 6.6 shows the phase portrait of the side-slip angle and yaw rate trajectory; the trajectory stays well within the stability constraints defined in (39) and (40). Ultimately, this result validates the effectiveness of the motion controller in controlling a real vehicle to avoid a static obstacle in the driving lane and safely return to the lane thereafter while maintaining vehicle stability and rider comfort.

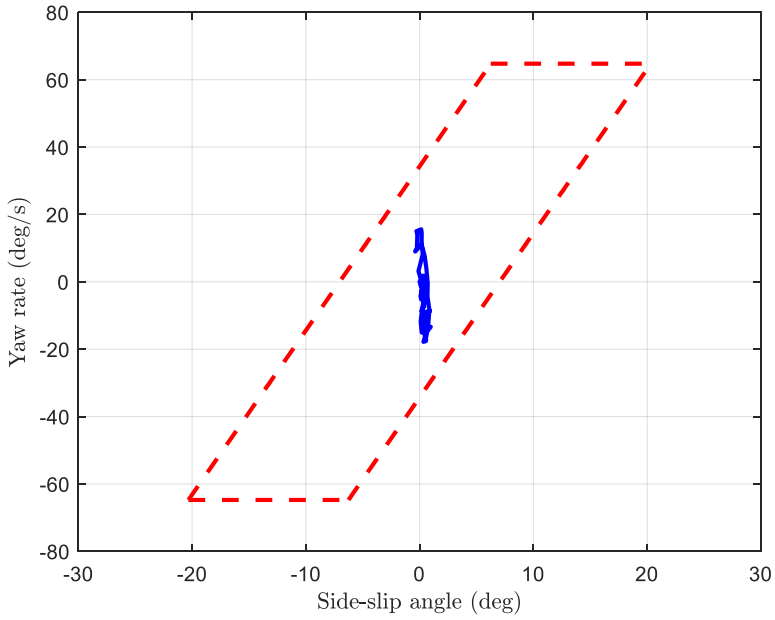

Figure 6.6: Phase portrait of yaw rate and side-slip angle during an obstacle avoidance maneuver. The constraints are indicated by the red dotted boundary.

6.3.3. Stopping for obstacle

To test the longitudinal control policy defined in (57), an experimental scenario is set up where the ego vehicle is required to stop for a lead vehicle placed in the driving lane. In this case, the virtual obstacle is placed on the far end of the reference path. The ego vehicle is parked at the start of the reference path, then the lateral and longitudinal controllers are both activated. However, the obstacle avoidance feature is deactivated; therefore, the longitudinal controller is responsible for stopping the vehicle to avoid a collision. Unlike the prior experimental trials, the human operator has no influence on the speed in this case. The resultant trajectory is shown in Figure 6.8(a), where the ego vehicle travels along the reference path until it comes to a stop approximately 7 m ahead of the lead vehicle. Furthermore, Figure 6.7 shows the 3D contour of the PF intensity surrounding the obstacle, where the ego vehicle trajectory is shown to approach the peak of the PF but stop beforehand due to the increasing longitudinal gradient of the field.

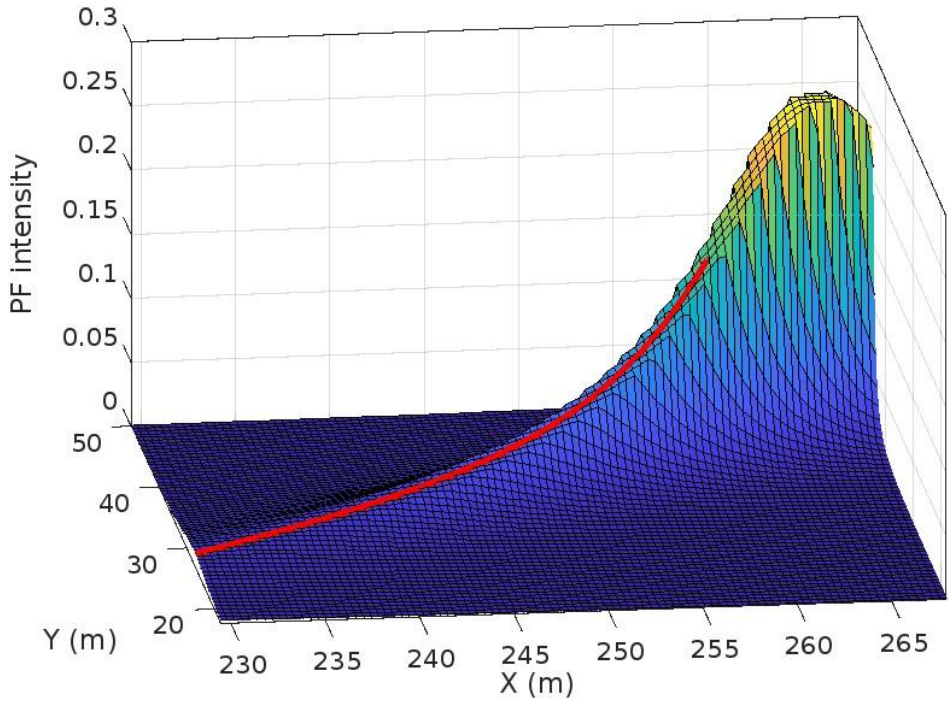

Figure 6.7: Potential field intensity surrounding obstacle during stopping maneuver, where the ego vehicle trajectory is indicated by the red line

The longitudinal controller commands the front axle torque to achieve the desired speed, and the lateral controller commands the steering to track the reference path, as shown in Figure 6.8(b) and Figure 6.8(c), respectively. Figure 6.8(d) shows the longitudinal speed request throughout the trial, where the request starts at the setpoint of $v_{x,\max} = 25$ km/h and then decreases according to the potential field gradient, as computed in (57). Consequently, the vehicle comes to a gradual and complete stop while achieving a safe distance behind the obstacle. This reflects the simulated result presented in 5.2.2 and ultimately demonstrates the effectiveness of this novel longitudinal control policy in adapting the vehicle speed to avoid a collision with a lead vehicle. Additionally, this result demonstrates that the lateral controller can keep the vehicle centered in the driving lane while the vehicle slows to a stop.

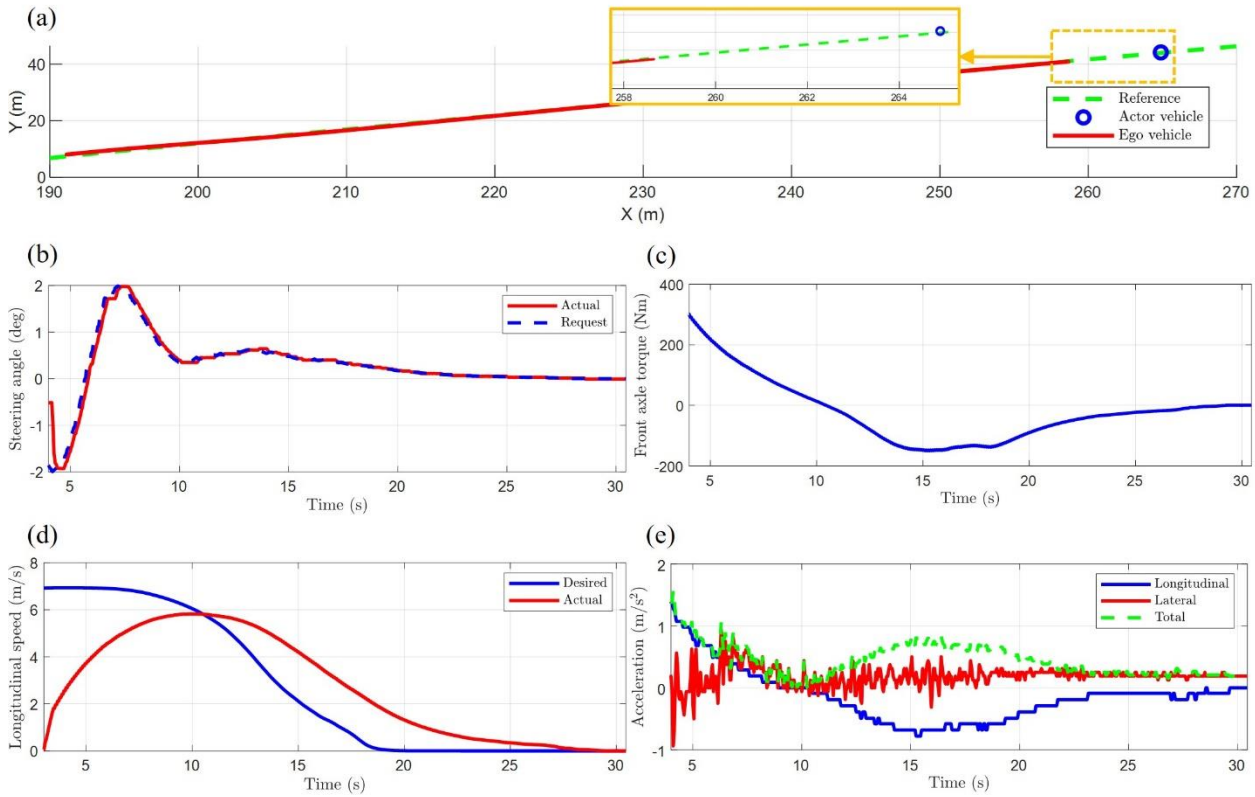

Figure 6.8: Stopping performance on Chevrolet Equinox. (a) Ego vehicle trajectory, where the ego vehicle stops ~6 m behind the obstacle. (b) Steering angle. (c) Front axle torque, where negative torque acts to slow the vehicle. (d) Requested and actual longitudinal speeds. (e) Lateral, longitudinal, and resultant acceleration.

CHAPTER 7

Conclusions and Future Work

7.1. Conclusions

This thesis describes an autonomous vehicle motion controller which employs MPC and potential fields for autonomous vehicle path planning control. The described method advances prior works by augmenting the accuracy of the vehicle dynamics model. The controller design presented herein considers the combined-slip effect on tire forces: the tire forces are estimated by the Burckhardt combined-slip model, which are shown to model those generated by the Pacejka model in various slip conditions with reasonable accuracy. Additionally, other key state predictions are shown to match the actual states with a high degree of accuracy. Therefore, the dynamics model is an effective representation of vehicle dynamics for lateral motion control.

The study also presents steps to reduce the computational complexity of the controller. A novel longitudinal control policy, which slows the vehicle to avoid collisions according to the potential field gradient, removes the longitudinal aspect from the optimization problem. The longitudinal control policy is shown to slow the vehicle to perform adaptive cruise control as well as stop for another vehicle at a safe distance. Moreover, the potential field approximation in the MPC objective is simplified to only consider the repulsive force in the lateral direction, which further simplifies the problem. Furthermore, unlike prior studies, the potential field approximation considers the estimated road heading angle to adapt the PF repulsive force according to the lateral direction in the road coordinate frame.

The presented motion control design is validated through both simulation and physical vehicle tests. The experimental tests showed good performance of the controller in two maneuvers: avoiding and stopping for an obstacle in the driving lane. Each of these tests showed that the longitudinal and lateral controllers maintain a safe distance from the obstacle with regard for vehicle stability and rider comfort. In simulation, these maneuvers are repeated, but at higher speeds which are not feasible during the physical tests. Therein, the system is shown to maintain vehicle stability, even during highly evasive maneuvers where road conditions are diminished.

Moreover, the longitudinal and lateral controllers are demonstrated in concert to perform such an evasive maneuver: the brakes are applied to slow the vehicle, while the steering is activated to perform a lane change.

7.2. Future Work

The design of the controller may be further improved by adapting to varying road surface friction. It is shown that the controller model overestimates the tire forces in simulation when the road surface friction is diminished due to poor conditions. The yaw rate and side slip angle may be used to train a data-driven model in select road conditions to provide an estimation model for road conditions. By such an estimate, the tire model parameters may be adapted according to better reflect the actual road conditions, thereby improving the prediction accuracy of the controller model. Other uncertain parameters in the controller model, such as vehicle mass, for example, may also be adapted rather than assuming fixed values.

A detailed dynamics model is presented in this thesis, which characterizes the vehicle, tire, wheel, and actuator dynamics. While the benefits of such a model are demonstrated, there remains some uncertainty in the vehicle which is not represented by the controller model. Therefore, a learning-based model should be explored in the future. Such a learning-based model may be trained on the error of the fixed dynamics model to improve prediction accuracy. A comparison of the learning-based approach to a fixed model like the one shown in this thesis would be valuable.

This thesis presents a decoupled control approach: MPC handles only lateral control using active front steering, and an error-based longitudinal controller tracks the desired speed by commanding axle torque and braking. Although the detailed wheel and tire model in the lateral controller improves the prediction accuracy, it should be further leveraged for longitudinal control objectives, such as traction control. To do so, the MPC controller may perform torque vectoring in addition to active steering, according to the total tire slip ratio. Thereby, the lateral stability of the vehicle as well as the traction of each tire would be considered as part of the optimal control policy. Furthermore, the vehicle may perform more effective evasive maneuvers, particularly where road conditions are diminished.

References

[1] S. Singh, “Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey,” Washington, DC, 2015.

[2] K. Bucsuházy, E. Matuchová, R. Zůvala, P. Moravcová, M. Kostíková, and R. Mikulec, “Human factors contributing to the road traffic accident occurrence,” *Transp. Res. Procedia*, vol. 45, no. 2019, pp. 555–561, 2020, doi: 10.1016/j.trpro.2020.03.057.

[3] I. Yaqoob, L. U. Khan, S. M. A. Kazmi, M. Imran, N. Guizani, and C. S. Hong, “Autonomous Driving Cars in Smart Cities: Recent Advances, Requirements, and Challenges,” *IEEE Netw.*, vol. 34, no. 1, pp. 174–181, 2020, doi: 10.1109/MNET.2019.1900120.

[4] K. Berntorp, B. Olofsson, K. Lundahl, and L. Nielsen, “Models and methodology for optimal trajectory generation in safety-critical road-vehicle manoeuvres,” *Veh. Syst. Dyn.*, vol. 52, no. 10, pp. 1304–1332, 2014, doi: 10.1080/00423114.2014.939094.

[5] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, “Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952–964, 2017, doi: 10.1109/TVT.2016.2555853.

[6] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-Based Model Predictive Control: Toward Safe Learning in Control,” *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 3, no. 1, pp. 269–296, 2020, doi: 10.1146/annurev-control-090419-075625.

[7] Y. Rasekhipour, A. Khajepour, S. K. Chen, and B. Litkouhi, “A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1255–1267, 2017, doi: 10.1109/TITS.2016.2604240.

[8] D. Chu, H. Li, C. Zhao, and T. Zhou, “Trajectory Tracking of Autonomous Vehicle Based on Model Predictive Control With PID Feedback,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, 2022, doi: 10.1109/TITS.2022.3150365.

[9] E. Galceran, R. M. Eustice, and E. Olson, “Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2015-Augus, no. Iv, pp. 304–309, 2015, doi: 10.1109/IVS.2015.7225703.

[10] F. Bounini, D. Gingras, H. Pollart, and D. Gruyer, “Modified artificial potential field method for online path planning applications,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 180–185, doi: 10.1109/IVS.2017.7995717.

[11] M. Li *et al.*, “A Two-Layer Potential-Field-Driven Model Predictive Shared Control Towards Driver-Automation Cooperation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4415–4431, 2020, doi: 10.1109/TITS.2020.3044666.

[12] C. W. Warren, “Global path planning using artificial potential fields,” in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 316–321 vol.1, doi: 10.1109/ROBOT.1989.100007.

[13] Y. K. Hwang and N. Ahuja, “A Potential Field Approach to Path Planning,” *IEEE Trans. Robot. Autom.*, vol. 8, no. 1, pp. 23–32, 1992, doi: 10.1109/70.127236.

[14] M. T. Wolf and J. W. Burdick, “Artificial potential functions for highway driving with collision avoidance,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 3731–3736, 2008, doi: 10.1109/ROBOT.2008.4543783.

[15] T. Hsiao, “Robust wheel torque control for traction/braking force tracking under combined longitudinal and lateral motion,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1335–1347, 2015, doi: 10.1109/TITS.2014.2361515.

[16] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, “Safe driving envelopes for path tracking in autonomous vehicles,” *Control Eng. Pract.*, vol. 61, pp. 307–316, 2017, doi: 10.1016/j.conengprac.2016.04.013.

[17] K. Yuan, H. Shu, Y. Huang, Y. Zhang, A. Khajepour, and L. Zhang, “Mixed local motion planning and tracking control framework for autonomous vehicles based on model predictive control,” *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 950–959, 2019, doi: 10.1049/iet-its.2018.5387.

[18] E. Kim, J. Kim, and M. Sunwoo, “Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics,” *Int. J. Automot. Technol.*, vol. 15, no. 7, pp. 1155–1164, 2014, doi: 10.1007/s12239-014-0120-9.

[19] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, “MPC-based approach to active steering for autonomous vehicle systems,” *Int. J. Veh. Auton. Syst.*, vol. 3, no. 2–4, pp. 265–291, 2005, doi: 10.1504/IJVAS.2005.008237.

[20] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, “Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization,” *Robot. Sci. Syst.*, 2013, doi: 10.15607/rss.2013.ix.031.

[21] H. Wang, Y. Huang, A. Khajepour, D. Cao, and C. Lv, “Ethical Decision-Making Platform in Autonomous Vehicles with Lexicographic Optimization Based Model Predictive Controller,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8164–8175, 2020, doi: 10.1109/TVT.2020.2996954.

[22] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, “Crash Mitigation in Motion Planning for Autonomous Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3313–3323, 2019, doi: 10.1109/TITS.2018.2873921.

[23] Y. Huang, H. Wang, A. Khajepour, H. Ding, K. Yuan, and Y. Qin, “A novel local motion planning framework for autonomous vehicles based on resistance network and model predictive control,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 55–66, 2020, doi: 10.1109/TVT.2019.2945934.

[24] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path planning for autonomous vehicles in unknown semi-structured environments,” *Int. J. Rob. Res.*, vol. 29, no. 5, pp. 485–501, 2010, doi: 10.1177/0278364909359210.

[25] S. Erke, D. Bin, N. Yiming, Z. Qi, X. Liang, and Z. Dawei, “An improved A-Star based path planning algorithm for autonomous land vehicles,” *Int. J. Adv. Robot. Syst.*, vol. 17,

no. 5, pp. 1–13, 2020, doi: 10.1177/1729881420962263.

[26] Z. Chen and X. Huang, “End-To-end learning for lane keeping of self-driving cars,” *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 1856–1860, 2017, doi: 10.1109/IVS.2017.7995975.

[27] M. Liu, X. Deng, Z. Lei, C. Jiang, and C. Piao, “Autonomous Lane Keeping System: Lane Detection, Tracking and Control on Embedded System,” *J. Electr. Eng. Technol.*, vol. 16, no. 1, pp. 569–578, 2021, doi: 10.1007/s42835-020-00570-y.

[28] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2015-Augus, pp. 1094–1099, 2015, doi: 10.1109/IVS.2015.7225830.

[29] P. Polack, F. Altche, B. DAndrea-Novel, and A. De La Fortelle, “The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?,” *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 812–818, 2017, doi: 10.1109/IVS.2017.7995816.

[30] J. Guo, P. Hu, and R. Wang, “Nonlinear Coordinated Steering and Braking Control of Vision-Based Autonomous Vehicles in Emergency Obstacle Avoidance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3230–3240, 2016, doi: 10.1109/TITS.2016.2544791.

[31] E. Hashemi, M. Jalali, A. Khajepour, A. Kasaiezadeh, and S. K. Chen, “Vehicle Stability Control: Model Predictive Approach and Combined-Slip Effect,” *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 6, pp. 2789–2800, 2020, doi: 10.1109/TMECH.2020.2993792.

[32] Y. Qin, E. Hashemi, and K. Amir, “Integrated Crash Avoidance and Mitigation Algorithm for Autonomous Vehicles,” *IEEE Trans. Ind. Informatics*, vol. 3203, no. c, pp. 1–10, 2021, doi: 10.1109/TII.2021.3058948.

[33] M. Jalali, E. Hashemi, A. Khajepour, S. ken Chen, and B. Litkouhi, “A combined-slip predictive control of vehicle stability with experimental verification,” *Veh. Syst. Dyn.*, vol. 56, no. 2, pp. 319–340, 2018, doi: 10.1080/00423114.2017.1376751.

[34] H. Lee and S. Choi, “Development of Collision Avoidance System in Slippery Road Conditions,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–13, 2022, doi: 10.1109/TITS.2022.3168668.

[35] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. .

[36] M. Dousti, s. C. Baslamısli, e. T. Onder, and S. Solmaz, “Design of a multiple-model switching controller for ABS braking dynamics,” *Trans. Inst. Meas. Control*, vol. 37, no. 5, pp. 582–595, 2015, doi: 10.1177/0142331214546522.

[37] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, “Learning-Based Model Predictive Control for Autonomous Racing,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3363–3370, 2019, doi: 10.1109/LRA.2019.2926677.

[38] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, “Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4029–4036, 2014, doi: 10.1109/ICRA.2014.6907444.

[39] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, “Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking,” *Int. J. Rob. Res.*, vol. 35, no. 13, pp. 1547–1563, 2016, doi: 10.1177/0278364916645661.

[40] X. Zhang, M. Bujarbaruah, and F. Borrelli, “Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework,” *arXiv*, pp. 1–25, 2019, doi: 10.1109/tcst.2020.3024571.

[41] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, “Provably safe and robust learning-based model predictive control,” *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013, doi: 10.1016/j.automatica.2013.02.003.

[42] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. John Wiley & Sons, Inc., 2012.

[43] C. G. Bobier and J. C. Gerdes, “Staying within the nullcline boundary for vehicle envelope control using a sliding surface,” *Veh. Syst. Dyn.*, vol. 51, no. 2, pp. 199–217, 2013, doi: 10.1080/00423114.2012.720377.

[44] B. Alsadik, “Chapter 10 - Kalman Filter,” in *Adjustment Models in 3D Geomatics and Computational Geophysics*, vol. 4, B. Alsadik, Ed. Elsevier, 2019, pp. 299–326.