Design a Reliable Model Predictive Control for Path Following with Application to the Autonomous Vehicle and Considering Different Vehicle Models

by

Behnaz Ahmadi

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, 2021

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

Despite many advances that aim to bring autonomous cars to market, several challenges remain to overcome before such technology is widely adopted. Among these is the need for a reliable cornering control that can work efficiently in real-time.

This thesis will focus on the successful development, implementation, and validation of reliable control algorithms for motion control of autonomous vehicles, which will give the University of Waterloo Watanomous team a competitive edge in the annual Auto-Drive competition. A reliable path following controller based on Model Predictive Control (MPC) and using different types of vehicle models are developed to 1) minimize the lateral distance between the vehicle and a reference path, 2) minimize the heading error of the vehicle, and 3) limit the rate of steering inputs to their saturation values and produce smooth motions. The control algorithm proposed in this thesis will rely on a known path as the desired input. Such input will be used to define a horizon over which reliable control actions can be computed to command the vehicle along the desired path with minimum lateral position error.

The proposed MPC is implemented in MATLAB in which the associated Optimal Control Problem (OCP) is discretized using the direct Multiple Shooting method. MPC is tested using the CarSim vehicle simulator to validate its performance for double lane change (DLC) vehicle maneuver at constant low/high forward speeds and roads with different known friction coefficients. The effects of changing prediction horizon, sampling time, and weighting factors on path-following controllers' performance are also discussed.

Simulation results show that designed controllers pulled the system back to the predefined reference path, and tracking performances were satisfactory. It is also observed that MPC designed using a linear dynamic model and a combined single-track model, both had comparable performances and less lateral tracking error than MPC designed using the nonlinear kinematic model for various speeds on the dry road, and at the speeds below 80 *kph* on the wet road. Moreover, MPCs using linear dynamic and combined single-track vehicle models both worked in the shorter horizon. Overall, MPC using a combined single-track model showed enhanced performance by comparing the other two schemes for all the tests conducted on wet and dry roads at various vehicle speeds.

Acknowledgements

I would like to thank my supervisor, Prof. William Melek, for his invaluable advice, continuous support, and patience during my Master's study. His immense knowledge and plentiful experience have encouraged me in my academic research and daily life. It is a great honor to work under his supervision.

I would like to thank Prof. Amir Khajepour for his support, encouragement, creative and comprehensive advice during my study and this work.

I would like to thank Prof. Naser Lashgarian Azad and Prof. Soo Jeon for their valuable comments.

I would like to thank Dr. Mohamed Mehrez and Mehdi Abroshan for their help and valuable comments during this work.

Last but not least, I must express my very profound gratitude to my parents, my daughter, my sisters, and their husbands for providing me with unfailing support and continuous encouragement throughout my years of study and throughout the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Dedication

This is dedicated to my daughter and my parents.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Numerous accidents result from human errors and traffic congestion, so converting cars to autonomous (smart) vehicles has been proposed to reduce these problems. Autonomous vehicles can carry outmaneuvers, navigate through an environment, and make decisions without human supervision. An autonomous car needs the best awareness of its surrounding environment to make the best decisions and recognize the correct pathway and all objects around it. This information can be acquired by using various sensors like Radar, Light Detection and Ranging (LIDAR); Global Positioning System (GPS); Inertial Measurement Unit (IMU), and cameras. An advanced control system interprets sensory information for tasks such as path following, lane changing, and obstacle avoidance. Therefore, the autonomous vehicle can move safely and appropriately through unknown environments by using sensors to detect objects in the environment and controllers to track the desired path using the feedback obtained from these sensors.

This Master thesis focuses on designing a path following controller for the Watanamous electric vehicle used in the Auto-Drive competition. The proposed controller should enable the vehicle to closely follow a reference path at various high/low speeds on wet/dry roads. The results of this research will be implemented on the Auto-Drive Chevy Bolt vehicle to assist the University of Waterloo Watonomous team in executing all path planning challenges during the annual Auto-Drive challenge.

1.1 Motivation

Despite the many advances in bringing autonomous cars to the market, several challenges remain to overcome before such technology is widely adopted. Among these is the need for reliable cornering control that can run efficiently in real-time.

Existing path following control methods in the literature mostly lack experimental validation [15, 13, 40, 41, 45]. Furthermore, the most common control methods for vehicle path following are optimal techniques [32, 44, 23, 30]. However, most of the proposed optimal control methods in the literature have not validated on EV platforms to ensure reliable performance in various maneuvers. Therefore, this thesis will focus on the successful development, implementation, and validation of reliable control algorithms for motion control of autonomous vehicles, which will give the University of Waterloo Watanomous team a competitive edge in the annual Auto-Drive competition. Furthermore, the design and experimental validation of real-time control methods for autonomous cars will be crucial. To advance this technology, make it safer, and gain public acceptance, all difficulties should be solved before seeing wider-scale deployment as a future solution for human and goods transportation.

The proposed path following control algorithm will need to be integrated into the Chevy Bolt 2017 software pipeline. It should be tested to ensure satisfactory real-time performance to allow the vehicle to execute various maneuvers autonomously effectively, i.e., double-lane change, break-in-turn. The control algorithm proposed in this thesis will rely on a known path the vehicle has to follow as the desired input (computed via reliable path planning methods, which are outside the scope of this work). Such input will be used to define a horizon over which reliable control actions can be computed to command the vehicle along the desired path with minimal lateral position errors. This primary objective will be used to design the control algorithms proposed in this thesis.

1.2 Proposed Contribution

This thesis studies the path following problem for autonomous vehicles. The main objective of this thesis is to design a reliable path following controller which can: 1) minimize the lateral distance between the vehicle and defined path, 2) minimize the headings of the vehicle and its defined path, and 3) limit the rate of steering inputs to produce smooth motions.

The thesis proposes the design of both linear and nonlinear Model Predictive Controller (MPC) schemes to achieve the above objective. The MPC-based path following controller aims to minimize the norm of the error between the control output and the reference path that the vehicle needs to follow. MPC schemes are among the most popular optimal control methods often used to achieve desired performance by minimizing multi-constrained objective functions that govern closed-loop control performance [25].

The first contribution in MPC development involves designing a lateral controller for path following using the nonlinear kinematic bicycle model. Constraints are introduced in the design process of this controller to ensure desired performance. The steering angle is introduced as the control input to the system, and the control objective is to minimize the lateral distance between the vehicle position and the predefined path.

The second and third contribution involves swapping the bicycle model above with a linear lateral dynamic and combined single-track vehicle model. The path following controller has been implemented in MATLAB/Simulink. CasADi has been used as an open-source tool for nonlinear optimization and algorithm differentiation. Additionally, the Multiple Shooting Method [4] is applied to breakdown the system integration into shorter time intervals, solve the optimal control problem, and use the system model as a state constraint in each optimization step. The path following controllers from all contributions will be tested using a vehicle simulator, CarSim, to validate its performance at various constant longitudinal forward speeds and known friction coefficients on wet/dry roads under a double lane change (DLC) scenario. In this case, the C-Class Hatchback vehicle model is utilized in CarSim for simulation. The designed controllers using different models will be tested to compare their tracking performance.

1.3 Thesis Outline

Chapter 2 presents a literature review on existing vehicle lateral control algorithms. A summary table 2.3 including the papers related to this control method is also provided.

In chapter 3, the nonlinear kinematic, linear dynamic, and combined single-track state-space models of the vehicle are presented. All controllers based on the model predictive controller (MPC) are proposed to track a DLC predefined path at various speeds and on the road with known friction coefficients. The Multiple Shooting Method is also discussed in this section.

In chapter 4, the MPC schemes based on mentioned vehicle models introduced in Chapter 3 are simulated in MATLAB/Simulink using the CasADi scheme. Moreover, they are also implemented on the high-fidelity CarSim Model, and their performances are compared.

Chapter 5 summarizes the comparison results of the three control methods proposed in this thesis. Conclusions of the thesis and directions for future research are also presented in this chapter.

Chapter 2

Literature Review

This chapter reviews the literature on different techniques and vehicle models used in previous works to design a path following controller for autonomous vehicles.

2.1 Autonomous Vehicle Literature Review

Autonomous vehicles (AVs) or driverless cars have been the subject of noticeable attention by various research teams worldwide. An AV equipped with various sensors such as cameras and LIDAR is designed to travel between destinations without human supervision. Theoretically, when the vehicle's automated driving system can effectively perform dynamic driving tasks in the various driving scenarios and environments, the vehicle is referred to as autonomous, Level 4 [39]. Several companies in the automotive industry have set goals for achieving full AV technology shortly. For instance, all Tesla EV models are equipped with self-driving capabilities in 2020 [31].

Historically, the AV's technological development and main milestones started a few decades ago, as early as 1918 [5, 31]. Of course, AV technology had changed drastically since the 1920s when the first radio-controlled vehicles were created [5]. In the 1960s, many AVs were designed with similar electronic guidance systems [5]. By the 1980s, a vision-guided system enabled the introduction of some semi-autonomous functions in high-end vehicles, such as lane-keeping, automatic braking, and adaptive cruise control [5]. General Motors revealed the first concept of an AV in 1939 and began the first phase of AV research and development in the 1950s [31]. Furthermore, the Defense Advanced Research Projects Agency's (DARPA) Grand Challenges Program accelerated research in the AV through the mainly made investments in this technology within the US since 2004 [31].

Although, by definition, an AV can also refer to any mobile robot platform that can navigate autonomously in unstructured environments, such systems differ depending on the environment they are deployed in and their motion models and dynamic constraints. Flying and aerial vehicles known as unmanned aerial vehicles (UAVs) are a class of autonomous vehicles that operate above ground at higher altitudes. There are also autonomous vehicles that operate below sea level, i.e., unmanned underwater vehicles (UUVs). This research focuses on autonomous ground vehicles (AGVs). AGVs can be regarded as the main subset of the unmanned ground vehicle category (UGV). Besides, AGVs are labeled as non-holonomic systems because they have fewer controllable degrees of freedom (DoF) than the total number of DoFs. The four DoFs of AV are motions in the two Cartesian coordinates direction, orientation, and heading in specific terms. They only have two controllable DoF, which are longitudinal direction torque (forward and backward), and lateral direction (bounded) steering input [20]. The five levels of classification for autonomous vehicles are described in Table 2.1[10].

Table 2.1: Classification of autonomous system

Level 0	**Level 1**	**Level 2**	**Level 3**	**Level 4**
No Automation	Assisted Automation	Partial Automation	High Automation	Full Automation

According to Deshpande et al., worldwide, nearly 3000 deaths happen daily because of road accidents [10], mostly caused by human error. Therefore, switching to fully autonomous vehicles will decrease this number significantly. Moreover, autonomous vehicle technology has more merits, such as reducing traffic congestion, increasing roadway capacity, improving fuel economy, reducing stress, and reducing the need for parking space by up to $\frac{1}{4}$ of the current capacity [5]. The merits and demerits of AV technology are discussed in depth in [22]. Table 2.2 provides a comparison of the benefits of AV versus a non-AV.

Passenger(s) safety, as a crucial factor in modern intelligent transportation systems, is of interest to all auto manufacturers. Vehicle safety technologies are available in automotive products to reduce fatalities and protect people from accidents. The safety subsystems in the vehicle today fall into the categories of passive and active technologies. Passive safety technologies such as seat belts and airbags have enhanced passenger safety and significantly reduce the risk of injury or death during an accident. Active safety and driver assistance systems have been introduced since the 1990s [5]. In contrast to the active safety system, the design of autonomous vehicles that are safe to operate for passengers and others within the environment of the AV (pedestrians, cyclists, other vehicles, etc.) under various road conditions is still an endeavor pursued by many researchers and industries around the globe. Among other objectives, these efforts are aiming to develop AVs that can carry out complex maneuvering and cornering tasks in unstructured environments and still achieve their best performance in terms of path following, collision avoidance, energy efficiency, etc., in the presence of disturbances.

Table 2.2: Advantages of AV versus non-AV [22]

Parameters	**Autonomous Car**	**No Autonomous Car**
Avoid drinking drunk	Driving drunk	Avoid drinking drunk
Independent mobility	Independent mobility for younger, elder, ad disable	No mobility for younger and disable
Fuel efficient	Improve fuel-efficient	Fuel utilization more
Utilize driving time	Driving time utilized for other activities or task rest	Driver concentrates on driving

In [12], a systematic literature review covers the potential capability and impacts of AVs, as well as the planning and policies governing their development. Moreover, the review provides the trajectories of the technological development of AVs and the disruptive effects caused by such development. This work also introduces strategies to address the disruptions and probable gaps in the literature.

In [29], the authors focus on the methodologies for intelligent intersection management systems and the production of autonomous vehicles.

Mobile robotics technology has improved significantly in the past decade. These systems can now move around safely in cluttered surroundings, understand natural speech, recognize a real object, locate themselves, plan their path, and generally think independently. Subsequently, they can perform many functions, including localization, perception, planning, control, and management, required for safe navigation within an unknown environment. Therefore, different disciplines such as computer science, mechanical, electrical, and control engineering should be combined to design an AV [9].

In [27], the authors presented the challenges inherent in developing an autonomous system, such as dealing with the environment's dynamism complexity, carrying out a high workload, and developing risk assessment measures. This paper answers questions about the main requirements for building an adjustable autonomous and ways to improve this technology with an eye on safety and reliability. Many key hardware and software capabilities needed to develop reliable autonomous vehicle technology, such as accurate GPS, obstacle avoidance, accurate sensor fusion algorithms, path tracking controllers, and the need for situation awareness, etc., are also described in [38].

2.2 Lateral Control of Autonomous Vehicle Literature Review

One of the most active fields of research and development related to automated driving is driver assistant systems. Developing an AV platform requires design, testing, and integration of different modules, including sensing and perception, planning, and control. The sensing and perception module processes real-time vehicle sensors to localize the vehicle within its environment. The planning module computes a safe and collision-free path for the vehicle to follow based on the sensing and perception module's sensory information. The control module contains the actuators to move the AV along the desired path safely. Many research works in the literature addressed developing these modules for AV applications; however, creating fully autonomous vehicles that require no human supervision to operate safely and avoid accidents on the road remains a complex task [37, 7, 34, 31]. Nevertheless, significant research progress on path planning and tracking of unmanned aerial vehicles (UAV) and other types of robots have been reported. Unfortunately, none of these methods developed to-date can be easily adopted for AGVs because of the complexity of the high-fidelity vehicle models, and the uncertainty in the environment due to co-existence with humans and other objects in the neighborhood of the AV [19].

This thesis focuses on the third module, which is the development of a reliable control approach guiding vehicles autonomously along a predefined geometric reference path. The objective function of the controller is designed to 1) minimize the lateral distance between the vehicle and defined path, 2) minimize the headings of the vehicle and its defined path, and 3) limit the rate of steering inputs in order to produce smooth motions. Additionally, in this control scheme, correctional steering input is introduced to adjust the vehicle's lateral direction position. This work's motivation aligns with the fact that all levels of automation expressed by SAE-J3016 [37] require vehicles to follow a specific and desired path or trajectory.

Frequently, trajectory generation and tracking are named as two parts of the path following problems. A controller designed explicitly for path following, or trajectory tracking leads the vehicle to follow a globally defined geometric path. This goal can be achieved by determining and calculating the desired input, which can be a correctional steering input to adjust the vehicle's position in the lateral direction or correctional braking or throttle setting to adjust the vehicle's motion in the longitudinal direction.

The design of control strategies for wheel-based vehicles, in general, is a complex task because of the multi-variable and nonlinear characteristics of the path following problem. Therefore, an accurate vehicle model as a requirement for control design plays a vital role in this case. The importance of the vehicle model in designing the controller is highlighted in two main development steps. The first step is related to simulating the vehicle system, as doing so plays a crucial role in gaining insight into the performance of the control design and improving it to meet the path following in practice. During the simulation, the controller properties are examined, and controller parameters are tuned to reach the best performance. Therefore, the path following performance is evaluated with the controller applied to the vehicle model in simulations. The second development step focuses on the control laws used, which govern the path following controller's performance and depend on the kinematics and dynamics of the vehicle system [3].

In general, vehicle models come in three types: geometric, kinematic, and dynamic. The geometric vehicle model is represented in the vehicle's geometrical dimensions, with no kinematic or dynamic properties. The geometric vehicle model's importance is apparent when the vehicle's dimensions are related to the turn radius and radius of curvature of the road undertaken by the vehicle during cornering. This model can be developed based on Ackerman steering configurations, for instance, without specifying vehicle velocity and acceleration. In contrast, the kinematic model contemplates the vehicle's motion in terms of acceleration, velocity, and position related to the whole vehicle's geometry. The dynamic model contemplates the vehicle's motion in terms of internal forces, inertia, and energy properties. In the literature, the path following problem has been addressed using four different approaches: geometric modeling only [24, 42], kinematic modeling only [36], dynamic modeling [47], or considering both kinematic and dynamic models [32].

Over the past decades, many studies on control design have focused on path-following controllers for AVs. In [2], the authors proposed a robust fractional-order proportional integral derivative (FOPID) controller optimized using a particle swarm algorithm with both kinematic and dynamic models. The FOPID showed improved performance compared to the conventional PID controller in terms of minimizing tracking errors [2]. In [46] an adaptive PID controller is presented by using the two DOF vehicle dynamic model. The Pure Pursuit and Stanley methods are adopted in [11] for designing a path following controller based on the vehicle's geometry. The control command is generated to ensure a smooth path and a minimum turning radius. A robust state feedback controller is proposed in [43] for lateral control of the vehicle considering parameter uncertainties and external disturbances. In [17] a hierarchically improved fuzzy dynamical sliding-model control (HIFDSMC) is compared with the hierarchical fuzzy decentralized path tracking controller. In [18], a path following control law based on multi-variable sliding mode is used to achieve global boundness and convergence of the position tracking error even in the presence of defined parametric uncertainty. State of the art, challenges in modeling and control strategies in the subject of path following are reviewed in [3]. The fact that vehicles are composed of mechanical and electrical subsystems presents a complexity layer to the design of controllers, which take into account factors such as saturation limits of actuators, physical constraints, nonlinear characteristics of the vehicle, and its interactions with the road. In this case, model predictive control (MPC) has emerged as an attractive approach that can be used to design a vehicle control system while incorporating all the above factors explicitly in the control design formulation. MPC has been used extensively in the control of constrained linear and nonlinear systems. This optimal control method computes a future control sequence in a defined horizon to predict the plant output as close as possible to the reference by minimizing the multi-stage cost function concerning future control actions and using a set of constraints. MPC also uses the predefined references in control law calculation [32]. MPC has been used for multiple inputs-multiple outputs dynamics systems to solve the tracking problems while guaranteeing stability [44]. MPC has also been used since the 1980s in practical applications taking into account system model uncertainties, constraints, plant-model mismatch, and disturbances [26]. Due to the MPC's advantages mentioned above, it has also been implemented in several automotive and other transportation active safety systems such as active traction, active braking, active steering, active suspension systems to improve vehicle handling and ride comfort [44].

Table 2.3 summarizes some of the most notable works in the area of AVG path following problem. The Table lists the vehicle model adopted in each reference and assumptions made on control design and implementation.

Table 2.3: Relevant works on designing different types of path following controller

Ref.No	**Method and Vehicle Model**	**Assumption and Summary**
[32]	MPC with both linear kinematic and dynamic vehicle model	First strategy is based on the successive linearization concept, and the second one is based on combining a local reference frame with an approaching path strategy.
[44]	MPC with both simple and complex linear bicycle models with and without the roll vehicle dynamics	A comparative study on 2WS and 4WS, known path, the crosswind effect is used as a step response, simulation test at both low/high forward speed and road friction surfaces, MPC eliminates the crosswind effect while maintaining stability.
[23]	Explicit MPC, LQR, with the linear dynamic model including position and orientation error	Explicit MPC is used to reduce the computational complexity caused by online optimization with three types of reference path (straight, circular, and clothoid).
[30]	MPC and PID with both nonlinear kinematic and dynamic models	The vehicle's speed is controlled by PID; LMPC is more time-expensive and has less error than PID; A real-time test on the wheeled mobile robot was carried out.
[15]	Fuzzy factor and LTV-MPC with both linear kinematic and dynamic models and included external disturbances	Presented a coordinated path following system and direct yaw moment control of the autonomous electric vehicle; A new fuzzy factor is used based on the magnitude of longitudinal velocity. Also, the wheel steering angle and external yaw moment were calculated by LTV-MPC; GPs and INS are used to estimate the vehicle's sideslip angle; the controller is validated in simulations only.

[13]	Robust MPC combined with multi-points preview controller and using the linear dynamic error models	The MPC acts as a feedback term, and the preview controller acts as a feedforward term; The controller is tested for double lane change maneuver with external disturbances in simulations.
[6]	MPC with the nonlinear vehicle dynamic model and an online linear model	The controller was tested on an icy road with a speed of up to 21m/s during double lane change maneuver in simulations with a known trajectory.
[21]	MPC with nonlinear kinematic and dynamic vehicle models	Comparison between dynamic and kinematic models at low speed; The simulated states are compared to the measured state of the real vehicle to evaluate the accuracy of both models; The results show that the kinematic model open-loop prediction mean error is greater than the dynamic model.
[14]	MPC with the linear kinematic model	Only the steering control was studied; The experimental results based on Hongqi AGV; Road boundaries detect the region of path tracking for AV, and then MPC was used to consider actuator and road boundary constraints.
[40]	MPC with the linear dynamic model	Design a path tracking controller to minimize lateral path tracking deviation at high speed and large lateral acceleration conditions; MPC minimized course-direction deviation instead of handing deviation and also reduced the lateral deviation.

[41]	MPC with vehicle lateral dynamic model	MPC was used to reduce the lateral tracking deviation and maintain vehicle stability at both high and low speeds; A fuzzy-logic-based condition classifier was used to maintain the regime condition of a vehicle maneuver and the switching instant; Velocity heading direction was chosen a tracking error
[45]	MPC with linear dynamic vehicle model	The controller with longitudinal speed compensation was presented to reduce the tracking error of the desired path; The longitudinal speed is a variable in the prediction horizon; The speed changes caused tracking error and changed the stability constraints; Tested just by simulation in high-speed maneuvers.

This thesis focuses on developing path following controller for autonomous vehicles, one of the main capabilities required to ensure such technology is reliable and safe. Existing path following control methods in the literature mostly lack experimental validation on an actual AV platform [44, 23, 13, 6, 21]. Furthermore, the most commonly used control methods for vehicle path following rely on optimal structures such as MPC methods [32, 44, 13, 6, 21, 40]. However, most of the proposed optimal control methods in the literature have required assumptions that can be prohibitive in practical applications. Most of these methods have not been validated on an AV platform to ensure reliable performance in various maneuvers and under different operating conditions.

Chapter 3

Vehicle Modeling and Model Predictive Controller Design

This chapter introduces the nonlinear kinematic, linear dynamic, and combined single-track bicycle models of the vehicle used in this thesis. The vehicle models introduced herein are used to design and evaluate the path following controller proposed in this thesis. Three controllers based on the model predictive controller (MPC) are proposed to track a predefined DLC path at low/high constant longitudinal speeds and on different known road friction coefficients. Multiple Shooting technique is also discussed in this chapter.

3.1 Nonlinear Kinematic Bicycle Model

Many studies on the path following control for AVs have employed either a linear or a nonlinear handling model, also known as a bicycle model, to design the steering controller. The bicycle model has been used to represent the vehicle's kinematics because of its simplicity and ability to represent the nonholonomic constraints. This thesis adopts the vehicle bicycle model, in which the right and left tires are merged and represented by one tire at the front and rear. The front-wheel orientation can be controlled relative to the heading of the vehicle. Figure 3.1 shows a representation of the bicycle vehicle model used in this thesis.

The kinematic bicycle model is analyzed using the reference points x_c and y_c, located at the vehicle's center of gravity (CG). The vehicle's velocity vector is denoted V and points in the same direction as the front wheel and has the slip angle β relative to the

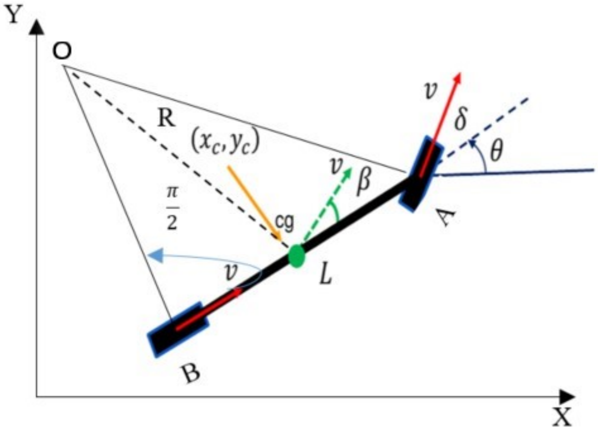


Figure 3.1: The bicycle kinematics of lateral vehicle motion

longitudinal axis. The steering angle for the front wheel is δ and is measured relative to the bicycle's longitudinal direction. In this model, just the front wheels can be steered. This assumption is referred to as the "no-slip" condition, which assumes no lateral and longitudinal movements of wheels. Moreover, the slip angles at both wheels are zero [33]. Distance from the center of the rear wheels to CG is l_r, and distance from the CG to the center of the front wheel is l_f. The wheelbase of the vehicle is $L = l_f + l_r$.

In this model, the coordinates axis of the CG in the inertial frame are denoted by X and Y while θ is the yaw angle, and $\dot{\theta}$ the yaw rate. The course angle for the vehicle is defined in equation 3.1.

$$\gamma = \theta + \beta \tag{3.1}$$

By applying the sine rule to the triangle OCA

$$\frac{sin(\delta - \beta)}{l_f} = \frac{sin(\frac{\pi}{2} - \delta)}{R} \tag{3.2}$$

Also, by applying the sine rule to the triangle OCB

$$\frac{sin(\beta)}{l_r} = \frac{sin(\frac{\pi}{2})}{R} = \frac{1}{R} \tag{3.3}$$

where R is the radius of the vehicle's path and is defined by the length of the line OC.

Substituting equation 3.1 into equation 3.2, we get

$$\frac{sin(\delta)cos(\beta) - sin(\beta)cos(\delta)}{l_f} = \frac{cos(\delta)}{R} \tag{3.4}$$

Multiplying both sides of the equation 3.4 by $\frac{l_f}{cos(\delta)}$, we obtain

$$tan(\delta)cos(\beta) - sin(\beta) = \frac{l_f}{R} \tag{3.5}$$

and multiplying both side of the equation 3.3 by l_r gives

$$sin(\beta) = \frac{l_r}{R} \tag{3.6}$$

Adding equations 3.5, and 3.6, we get

$$tan(\delta)cos(\beta) = \frac{l_f + l_r}{R} \tag{3.7}$$

By assuming that the radius of the vehicle path changes slowly due to the low vehicle speed, the rate of the vehicle's change of orientation will be equal to the angular velocity [33]. In other words,

$$\dot{\theta} = \frac{V}{R} \tag{3.8}$$

By using the equations 3.8 in the equation 3.7, we get

$$\dot{\theta} = \frac{V cos(\beta)}{l_f + l_r} tan(\delta) \tag{3.9}$$

where

$$\dot{X} = V cos(\theta + \beta) \tag{3.10}$$

$$\dot{Y} = V sin(\theta + \beta) \tag{3.11}$$

Therefore, the overall equations of motion of the vehicle are given by equations 3.9, 3.10, and 3.11 where the slip angle β can be obtained using equations 3.5 and 3.6 as follows

$$\beta = \tan^{-1}(\frac{l_r tan\delta}{l_f + l_r}) \tag{3.12}$$

3.2 Linear Lateral Dynamic Model

This section introduces the vehicle dynamic model based on the bicycle model approach in Figure 3.2. In the full dynamic bicycle model, two components of the motion are maintained: 1) the longitudinal direction is the heading direction, and 2) the lateral direction is perpendicular to the heading. Specifically, for the lateral vehicle model, the main part of the modeling defines the vehicle's rotation rate based on the moments that affect the vehicle while moving. In the lateral dynamics bicycle model, the following assumptions will be made: 1) The forward longitudinal velocity is assumed to be constant in order to decouple the lateral and longitudinal dynamic models, which simplifies the modeling task greatly but does lead to modeling inaccuracies when accelerating or decelerating out of curves, 2) The left and right wheels of both front and rear axles are lumped into a single wheel to convert the four wheels to two wheels bicycle model. Finally, 3) The nonlinear effects such as suspension movement, road inclination, and aerodynamic influences are neglected. Figure 3.2 shows the bicycle model used to represent the vehicle dynamics.

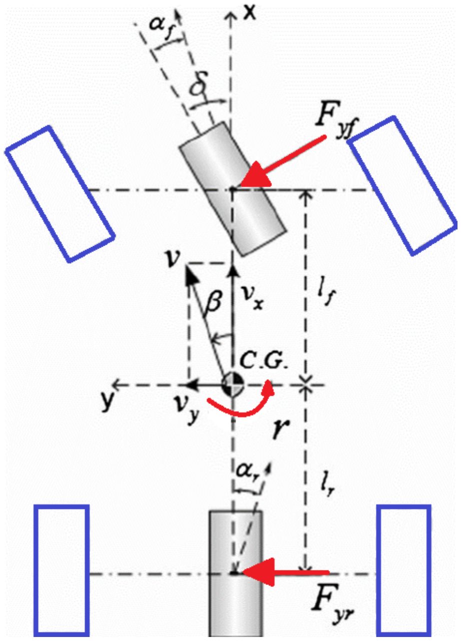


Figure 3.2: The lateral vehicle dynamics bicycle model

In this model, the vehicle center of gravity is used as the reference point to simplify Newton's second law application. The total acceleration in the inertial frame is defined in equation 3.13 and includes the lateral acceleration in the body frame.

$$a_y = \ddot{y} + w^2 R = V\dot{\beta} + V\dot{\psi} \tag{3.13}$$

The vehicle lateral dynamics expression can be formulated using the lateral forces on the front and rear tires by equation 3.14. In other words,

$$mV(\dot{\beta} + \dot{\psi}) = F_{yf} + F_{yr} \tag{3.14}$$

which m is the vehicle mass, V is vehicle velocity, $\dot{\beta}$ is slip side rate and $\dot{\psi}$ is the yaw rate.

The angular accelerations, $\ddot{\psi}$ is defined as

$$I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \tag{3.15}$$

where I_z is vehicle inertia, and l_f and l_r are the CG's distance from the center of the front and rear tires, respectively.

3.2.1 Tire Model

One of the most critical components of dynamic vehicle modeling is the tire. Generally, the exact prediction of tire forces is challenging, and tire models tend to be nonlinear and empirically defined. Fortunately, for normal driving conditions, a simple linear approximation of the tire model is valid for small slip angles and can be used to model tire forces. To use these linear tire models, the front and rear side slip angles (α_f, α_r) should be defined relative to the wheel's direction and the vehicle velocity at the wheel center. The cornering stiffness of a tire is defined by its ability to resist deformation while the vehicle corners. Figure 3.3 shows the relationship between the lateral tire force and the tire slip angle.

The slope of the curve in the linear region is known as the cornering stiffness coefficients, C_y. As shown in Figure 3.3 for small slip angles, typically less than eight degrees, the relationship between lateral forces and the wheel slip angle is almost linear. The parameters C_f and C_r are defined as the linearized cornering stiffness of the front and rear wheels in the vehicle's bicycle model, respectively. Therefore, the relation between the lateral forces acting on the front and rear wheels can be defined as:

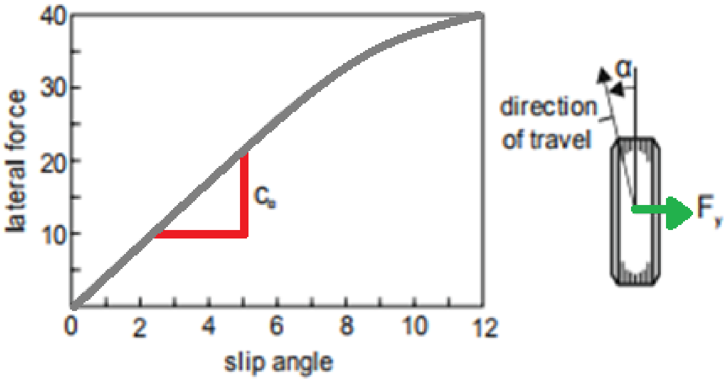


Figure 3.3: The relationship between the lateral tire force and the tire slip angle

$$F_{yf} = C_f\alpha_f = C_f(\delta - \beta - \frac{l_f\dot{\psi}}{V}) \tag{3.16}$$

$$F_yr = C_r\alpha_r = C_r(-\beta + \frac{l_r\dot{\psi}}{V}) \tag{3.17}$$

Substituting equations 3.16 and 3.17 in equations 3.14 and 3.15 produces the following two ordinary differential equations that govern the dynamics of the vehicle in the lateral direction.

$$\dot{\beta} = -\frac{C_r + C_f}{mV}\beta + (\frac{C_r l_r - C_f l_f}{mV^2} - 1)\dot{\psi} + \frac{C_f}{mV}\delta \tag{3.18}$$

$$\ddot{\psi} = \frac{C_r l_r - C_f l_f}{I_z}\beta - \frac{C_r l_r^2 + C_f l_f^2}{I_z V}\dot{\psi} + \frac{C - fl_f}{I_z}\delta \tag{3.19}$$

The dynamic lateral vehicle model in equations 3.18 and 3.19 is linear.

3.2.2 State-Space Representation of the Vehicle Model

This section presents the state-space representation of the vehicle dynamic model, which will be used in designing the path following controller. A state vector can be defined as $X = \begin{bmatrix} y & \beta & \psi & \dot{\psi} \end{bmatrix}^T$ which includes the lateral position, sideslip angle, yaw angle, and yaw rate. The model can be expressed in standard state-space form as follows

$$\dot{X} = AX + B\delta \tag{3.20}$$

$$A = \begin{bmatrix} 0 & V & V & 0 \\ 0 & -\dfrac{C_r + C_f}{mV} & 0 & \dfrac{C_r l_r - C_f l_f}{mV^2} - 1 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{C_r l_r - C_f l_f}{l_z} & 0 & -\dfrac{C_r l_r^2 + C_f l_f^2}{l_z V} \end{bmatrix} B = \begin{bmatrix} 0 \\ \dfrac{C_f}{mV} \\ 0 \\ \dfrac{C_f l_f}{I_z} \end{bmatrix}$$

The system matrices are A and B are time-invariant with assuming constant forward speed. The main input to the system in equation 3.20 is the driver steering angle command, δ.

3.3 Combined Single-Track Model with Switching

In sections 3.1 and 3.2, we introduced two vehicle models, namely the kinematic and dynamic bicycle models. A kinematic bicycle model according to [33] is introduced by assuming a steerable front wheel and zero slip angles at the front and rear wheels as given in equation 3.1. These assumptions are reasonable at the low-speed motion of the vehicle [33]. At higher speeds, the slip angles at both front and rear wheels can be more significant than zero because of the increase in lateral tire forces. Therefore, a dynamic bicycle model is introduced in section 3.2 to describe the lateral vehicle motion. Based on equations 3.18 and 3.19, the dynamic bicycle model when $V \to 0$ becomes unstable because of the vehicle velocity term at the denominator. In other words, the dynamic model cannot be used to control vehicles at low speeds, which happens in stop and go and urban driving scenarios [36]. A modified model is developed by combining both the vehicle's dynamic and kinematic motion equations to define a functional vehicle model that can work at both low and high speeds.

To avoid the design of two control structures, one for each vehicle model, a switchable bicycle model is introduced in this section. By designing the proposed controller to utilize the kinematic bicycle model at speed $0 \leq V \leq 50\ kph$ and a switching parameter $\kappa = 1$ which allows switching to the kinematic model, we can define:

$$\dot{X} = \kappa V cos(\psi + \beta) \tag{3.21}$$

$$\dot{Y} = \kappa V sin(\psi + \beta) \tag{3.22}$$

$$\dot{\theta} = \frac{V cos(\beta)}{l_f + l_r} \kappa tan(\delta) \tag{3.23}$$

$$\beta = \kappa tan^{-1}(\frac{l_r tan\delta}{l_f + l_r}) \tag{3.24}$$

Also, the dynamic bicycle model, which assumes small slip angles at the speed (50kph¡ V≤100kph) and the associated switching parameter $\kappa = 0$ is defined as:

$$\dot{X} = (1 - \kappa)(AX + B\delta) \tag{3.25}$$

$$A = \begin{bmatrix} 0 & V & V & 0 \\ 0 & -\dfrac{C_r + C_f}{mV} & 0 & \dfrac{C_r l_r - C_f l_f}{mV^2} - 1 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{C_r l_r - C_f l_f}{l_z} & 0 & -\dfrac{C_r l_r^2 + C_f l_f^2}{l_z V} \end{bmatrix} B = \begin{bmatrix} 0 \\ \dfrac{C_f}{mV} \\ 0 \\ \dfrac{C_f l_f}{I_z} \end{bmatrix}$$

The three vehicle models introduced in sections 3.1, 3.2, and 3.3 will be used in the next section to design the proposed model predictive path following controller.

3.4 Model Predictive Control Design

This section introduces the proposed Model Predictive Control (MPC) structure for vehicle path following. AVs must be able to make decisions at each sampling time in varying traffic and road conditions. The vehicle must also operate within road boundaries. Therefore, an MPC approach is introduced to design the path following controller while capturing all the dynamics which govern the vehicle response given the real-time control requirements and cornering/road constraints.

MPC, often referred to as receding horizon control, numerically solves an optimization problem in which the prediction discretized dynamic model is solved over a finite preview time [13]. A certain number of control actions are obtained for this finite horizon to minimize the objective performance index, even though only the first control action is applied, and all subsequent ones computed a priori are discarded. At the next time step, a new set of state variables is obtained, and the whole process is repeated to determine a new control action [1]. MPC is proposed to design the path following controller for autonomous vehicles because of its ability to handle multi-variable/multi-objective problems and explicitly consider hard constraints related to the vehicle dynamics and the environment [8].

The vehicle models mentioned in sections 3.1, 3.2, and 3.3 are discretized using the Euler method as follow

$$X_(k+1) = X_k + \triangle T.f_k(X_k, u_k) = f^{dt}(X_k, u_k) \tag{3.26}$$

where $\triangle T$ is sampling time.

Given the current state X_0 at the current time step t, MPC computes the optimal control sequence u_k with the receding horizon principle, which solves the following optimization problem:

$$Minimise J_{mpc}([X_k]_k^N, [u_k]) \tag{3.27}$$

$$\text{Subject to: } X_{k+1} - f^{dt}(X_k, u_k) = 0 \tag{3.27.1}$$

$$|u_k| - u_{sat} \leq 0 \tag{3.27.2}$$

$$|\dot{\psi}| - \dot{\psi}_{max} \leq 0 \tag{3.27.3}$$

where k = 0,...,N; N is prediction horizon, k is time step within the prediction horizon and u_{sat} denotes the saturated magnitude value of the control input. Equation 3.27.1 is the equality constraints related to the dynamics of the vehicle, equations 3.27.2 and 3.27.3 are inequality constraints on the control input and yaw rate, respectively. A constraint on the maximum yaw rate, $\dot{\psi}_{max}$, is defined in equation 3.28 to ensure lateral stability of the vehicle:

$$|\dot{\psi}_k| \leq \dot{\psi}_{max} = \frac{\mu g}{V_x} \tag{3.28}$$

where parameters μ, g, V_x are the friction coefficient, gravity, and longitudinal velocity, respectively.

The cost function to be minimised in the equation 3.27 is defined as follows

$$J_{mpc} = \frac{1}{2} \triangle X_k^T P \triangle X_k + \sum_{k=0}^{N-1} (\frac{1}{2} \triangle X_k^T Q \triangle X_k + \frac{1}{2} u_k^T R u_k) \tag{3.29}$$

where $\triangle X_k$ is the deviation of the actual state from the reference state and u_k represents the control input. P and Q are positive semi-definite matrices, and R is a positive definite constant weighting matrix.

The proposed MPC as an optimal control technique for autonomous driving can be adopted for a wide range of operating conditions and driving scenarios. This flexibility and convenience associated with the use of MPC in vehicle path following comes with an increase in the computational effort needed to compute the u_k steering command within the frequency range of the feedback control loop to ensure satisfactory and stable performance [8].

3.4.1 Direct Multiple Shooting

Optimal control problem (OCP) formulated in the form of differential equations as a part of dynamic optimization aims to determine control variables. Finding the optimal control problem's analytical solution is not easy because of the nonlinear system's existence. Therefore, the numerical method, Multiple Shooting Method, could solve the optimal control problem. Multiple Shooting method as one of the numerical methods was firstly found to solve the boundary value problem in differential equation scope [28]. By applying the Multiple Shooting method, the system of differential equations will be transformed into the numerical formulation, and OCP is transferred into a nonlinear programming problem (NLP).

In this work, the concept of Multiple-Shooting is applied in designing an MPC to break down the system integration into shorter time intervals, and the system model is used as a state constraint at each optimization step. Multiple Shooting as a Lifted Single Shooting technique [4] is used to reformulate a function to make it less nonlinear. Multiple Shooting is superior to the Single Shooting method. In this method, both the control input and state are initialized by solving an optimal control problem. Both control input and state act as the optimization variable in this case [35] [4]. The algorithm of Multiple Shooting method is defined as follow

Algorithm 1: Algorithm of Multiple Shooting Method

```
Discritize the horizon $[t_0, t_f]$ into equal subintervals $[t_i, t_{i+1}]$, such that
    $t_0 < t_1 < ... < t_n = t_f$
Parameterize the control function $u(t)$ for each subinterval
    $u(t) = v_i fort \in [t_i, y_{i+1}], i = 0, 1, ..., n-1$
Parametrize the initial condition of the state vector for each subinterval
    $x(t_i) = h_i, i = 0, 1, ..., n-1$
Evaluate the state trajectories in each subintervals and the value of $h_i$ from the
    final state subinterval considering the parameterized state initial value in the
    previous step
    $\dot{x}_i(t) = f(x_i(t), v_i, t), t \in [t_i, t_{i+1}], x_i = h_i$
Define the continuity constraints
    $h_{i+1]-x_{[i}}(t_{i+1}; h_i, v-i) = 0$
Compute the objective function for each subinterval
```

Chapter 4

Simulation Setup

In this chapter, the MPC schemes based on the kinematic, dynamic, and combined single-track models introduced in Chapter 3 are executed in Matlab using the CasADi. The numerical results based on the Carsim-Simulink co-simulation platform are obtained at various constant forward longitudinal speeds on the roads with known and varying friction coefficients.

4.1 CasADi and IPOPT

To design and simulate the proposed MPC controller for vehicle path following, an open-source tool, CasADi, is used within MATLAB and Simulink. CasADi implements algorithmic differentiation on user-defined symbolic expressions. It also provides standardized interfaces to various simulations, optimizations, and solutions of linear and nonlinear equations [35]. CasADi is written in C++, although the user input can be provided from other programming languages like MATLAB and Python. However, the internal handling of all symbolic expressions as well as the numerical computations performed in the compiled environment makes the speed of processing of this method in simulation comparable to directly compiling and running the C-code version [35]. A robust optimization solver, IPOPT (Interior Point Optimizer), is interfaced with CasADi, as an open-source software package that facilitates solving large-scale nonlinear optimization problems.

4.2 Vehicle Parameters

The vehicle model used in the CarSim simulation is a C-Class Hatchback 2017. Table 4.1 lists the parameters for that vehicle model.

Table 4.1: Parameters of vehicle model

Vehicle Parameter	Description	Value (units)
m_1	Vehicle sprung mass	$1270(kg)$
m_2	Vehicle unsprung mass(rear suspension)	$88.6(kg)$
m_3	Vehicle unsprung mass (front suspension)	$54.4(kg)$
a	Distance from center to front axis	$1.015(m)$
b	Distance from center to rear axis	$1.895(m)$
I_z	Vehicle yaw inertia	$1536.7(kg.m^2)$
C_f	Front tire cornering stiffness	$48377(N/rad)$
C_r	Rear tire cornering stiffness	$26075(N/rad)$

To find the length from the Center of Gravity (CG) to the front and rear axles first, the CG location should be determined. In this thesis, it is assumed that the CG is positioned in the center of the vehicle.

Among these parameters, the front and rear cornering stiffness values, C_f and C_r are not defined explicitly in CarSim as they vary according to tire slip angle. To calculate these values based on the tire model, $215/55R17$, and use them in the linear time-invariant system, the normal static load on each tire is calculated by equations 4.1 and 4.2.

$$F_{zf} = \frac{mg * b}{2(a+b)} = \frac{(1270 + 88.6 + 54.4) * 9.81 * 1.9}{2(1.9 + 1.01)} = 4525.2N \tag{4.1}$$

$$F_{zr} = \frac{mg * a}{2(a+b)} = \frac{(1270 + 88.6 + 54.4) * 9.81 * 1.01}{2(1.9 + 1.01)} = 2405.5N \tag{4.2}$$

Figure 4.1 shows the corresponding lateral tire force as a function of the tire's slip angle with different vertical tire loads. This plot can calculate the cornering tire stiffness by calculating the graph's slope and assuming it is linear up to 4 degrees of lateral slip angle. Table 4.2 is used to calculate the absolute longitudinal force based on the vertical tire load and absolute slip ratio. Therefore, based on the calculated front and rear lateral tire force obtained from equations 4.1 and 4.2, the interpolations between the red, blue, and green lines are needed to find the longitudinal force.

The rear and front cornering stiffness are calculated base on equations 4.3 and 4.4 as:

$$C_{yf} = \frac{\triangle F}{\triangle \alpha} = \frac{3878.7 - 1345.7}{(4-1) * \frac{\pi}{180}} = 48377(N/rad) \tag{4.3}$$

$$C_{yr} = \frac{\triangle F}{\triangle \alpha} = \frac{2098.9 - 733.6}{(4-1) * \frac{\pi}{180}} = 26075(N/rad) \tag{4.4}$$

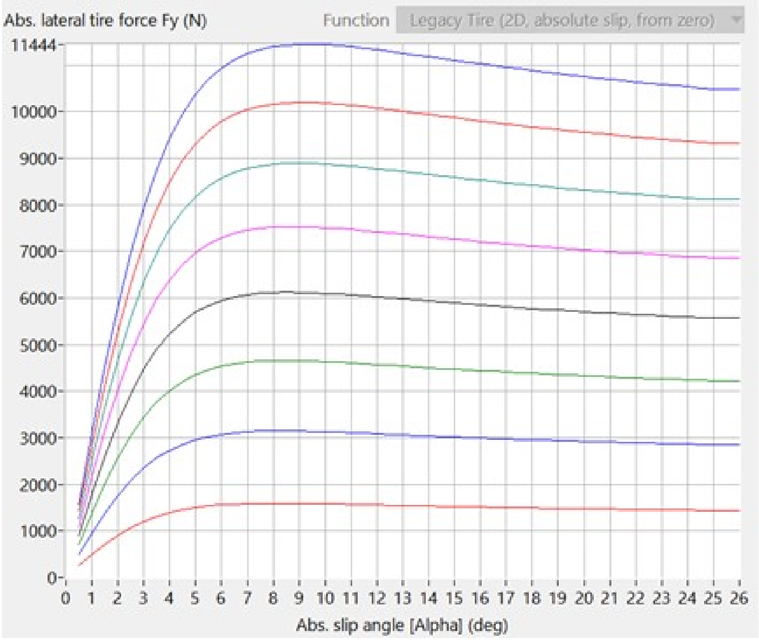


Figure 4.1: Corresponding lateral tire force as a function of the tire slip angle with different vertical tire loads

Table 4.2: Top row: vertical tire load, x-axis: absolute slip ratio and rest: absolute longitudinal force

	X Axis	**1**		**2**		**3**
0	0	1593.58	2405.5	3187.16	4525.2	4780.74
1	0.5	248.08		483.5		704.46
2	1	486	733.6	947.74	1345.7	1381.71
3	1.5	704.93		1357.9		2007.9
4	2	898.46		1755.66		2565.35
5	2.5	1063.18		2080.29		3044.12
6	3	1198.64		2348.57		3441.97
7	3.5	1306.65		2563.69		3762.95
8	4	1390.46	2098.9	2731.63	3878.7	4015.21
9	4.5	1453.91		2859.62		4208.83
10	5	1500.82		2954.92		4354.16
11	5.5	1534.64		3024.19		4460.73
12	6	1558.28		3073.11		4536.82
13	6.5	1574.12		3106.35		4589.27
14	7	1584.05		3127.63		4623.58

4.3 Lateral Tracking Error

Numerical simulations with MPC have been conducted using CarSim and MATLAB/Simulink to investigate the performance of the path following controllers proposed in chapter 3. Figure 4.2 shows the block diagram of the controller simulation implementation.

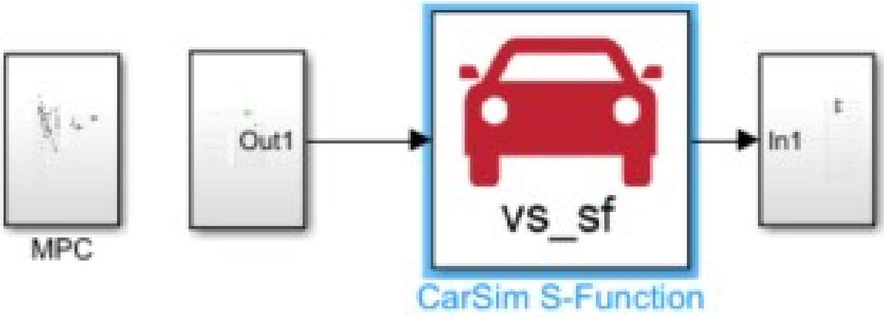


Figure 4.2: Architecture of path following controller

The MPC constructed by using the CasADi block is used to run a closed-loop steering maneuver to follow the predefined reference trajectory introduced in Figure 4.3 The simulation is conducted on both dry and wet roads with the known road friction coefficients (μ) of 0.85 and 0.5 at various low/high speeds without braking and accelerating. The speed limit is divided into three categories shown in Table 4.3 to define safe driving rules and speeds consistent with urban driving limits.

Table 4.3: Speed limit for various types of driving

Area of Driving	Speed Limit for Various Types of Area
School zones and residential streets	$30 \leq V(kph) < 50$
Major roads in urban and suburban areas	$50 \leq V(kph) < 70$
Most 2-lane highways outside cities and towns, and multi-lane highways	$70 \leq V(kph) \leq 100$

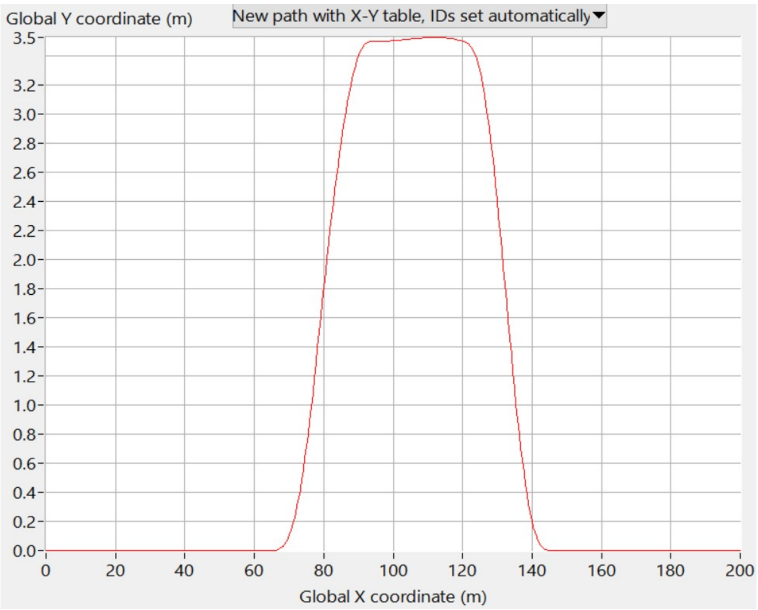


Figure 4.3: The predefined reference DLC path

Lateral tracking error is calculated based on the VS Math Models of CarSim that can provide output variables for externally defined driver steering controller. The output variables, lateral errors, are also based on the ISO/SAE intermediate X-axis points. The lateral distance error from the target path to point i, in Figure 4.4, is calculated by applying interpolation between nine points across the prediction horizon, at each sampling time using the vehicle forward speed V.

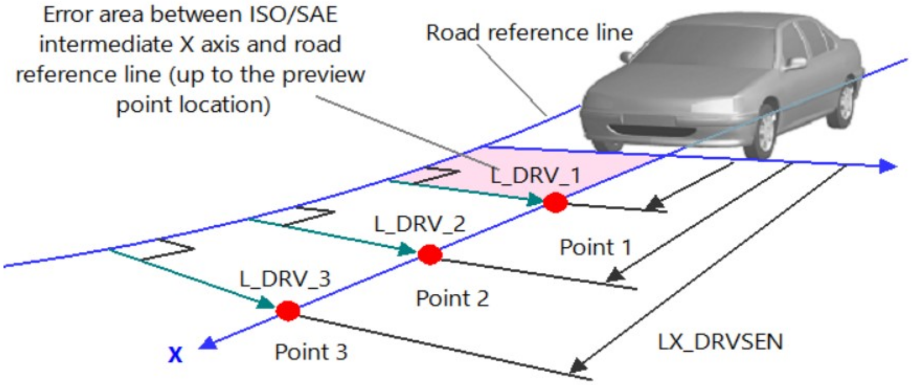

Error area between ISO/SAE intermediate X axis and road reference line (up to the preview point location)
Road reference line
L_DRV_1
L_DRV_2
Point 1
L_DRV_3
Point 2
LX_DRVSEN
X
Point 3

Figure 4.4: Preview points for external driver control

4.4 MPC Controller Design Using the Kinematic Bicycle Model

This subsection summarizes the proposed MPC system's simulation results in chapter 3 using the kinematic bicycle model. The controller's weighting gains are selected using the trial-and-error method, where selection criteria are based on the minimum lateral error tracking and smooth path following. Table 4.4 lists the upper and lower bounds of equality and inequality constraints. Experiments are performed at speeds 30 kph, 50 kph, 70 kph, and 100 kph, which are defined with the speed bounds in Table 4.3.

At each sampling time $\triangle T = 0.05s$, the MPC controller solves the constrained optimal control problem 3.27. The prediction of the vehicle state at each time step acts as an important step in the design of MPC for path following. Performance index will be minimized based on states' future prediction to compute the optimal control inputs [19]. The prediction horizon is assumed to be $N = 30$, which gives the MPC total preview time of $N * \triangle T = 0.05 * 30 = 1.5s$.

Table 4.4: The upper and lower bands of equality and inequality constraints for MPC using the kinematic vehicle model

Constraints	**Parameter**	**Lower Bound**	**Upper Bound**
Equality Constraint	Based on Equation 3.27.1	0	0
Inequality Constraints	Steering angle (δ)	-30°	30°
	Longitudinal Distance (X)	-1m	200m
	Lateral Distance (Y)	-1m	6m
	Yaw Angle (θ)	-180°	180°

The weighting matrices Q, R, and P of the proposed controller are shown in Table 4.5, where P is defined as an identity matrix.

Table 4.5: The value of weighting factors Q, R, and tracking error at different speeds in MPC using kinematic model, $\mu = 0.85$

$\mathbf{V(\frac{km}{h})}$	**R**	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$	**Error (cm)**
30	10	1	0	15	9
50	40	1	0	50	29.1
70	100	1	0	150	58.4
100	100	1	0	300	84.7

4.4.1 Simulation Results

The simulation results of the MPC controller with the kinematic bicycle vehicle model on the road with $\mu = 0.85$ and at various speeds are shown in figures 4.5, 4.6, 4.7, 4.8, 4.9, and 4.10. The simulation results show that the controller enables the vehicle to track the predefined path with small lateral tracking error listed in the last column of Table 4.5.

The controller's tracking accuracy drops at higher speeds. Therefore, relatively faster control of the steering wheel and more weights on the heading angle may be required by comparison to the lower speed cases. Results indicate the fulfillment of the MPC controller's constraints by showing that all variables are within the boundaries.

Selecting the weighting matrix R below 5 makes the vehicle unable to follow the path at the low-speed, 30 kph, satisfactorily. The weighting factor Q_3, the weight on the heading angle, under 15 results in oscillatory steering, more lateral error, and inaccurate path following at the low test speed, 30 kph. Increasing Q_1 decreases the lateral error at a low-speed. Yet, this change makes the vehicle unstable at a higher speed. Therefore, Q_1 is chosen as constant for all speeds. Nevertheless, lower values of N decrease the lateral error at the low-speed, inaccurate path following causes at higher speeds. Increasing the sampling time, e.g., $0.1s$, requires more weights on the control input and heading angle and increases the lateral error in low/high speeds.

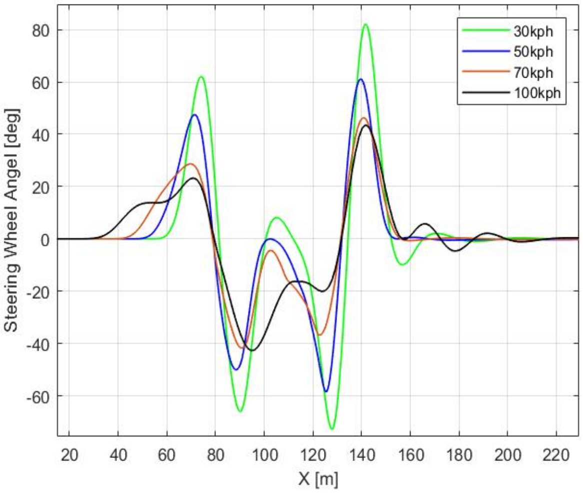


Figure 4.5: Steering handwheel angle based on kinematic model: External 4WS, DLC, $\mu = 0.85$

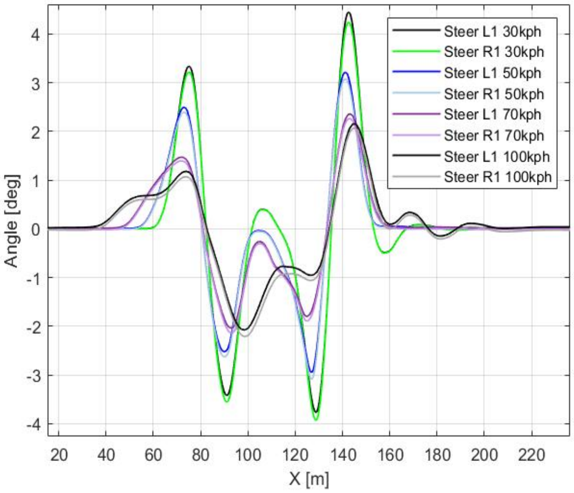


Figure 4.6: Road wheel steer angle - front based on kinematic model: External 4WS, DLC, $\mu = 0.85$

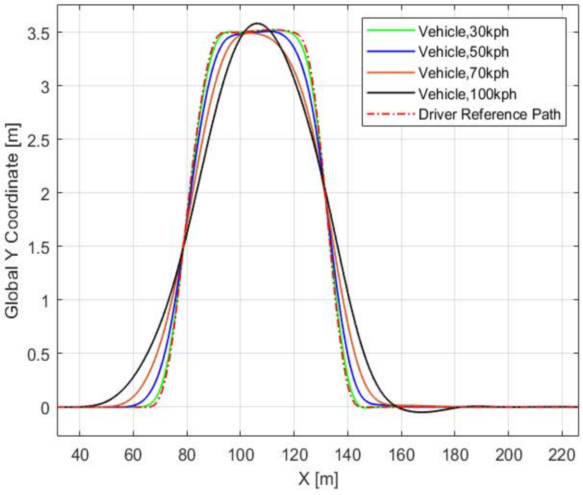


Figure 4.7: Y vs. X – trajectory based on kinematic model: External 4WS, DLC, $\mu = 0.85$

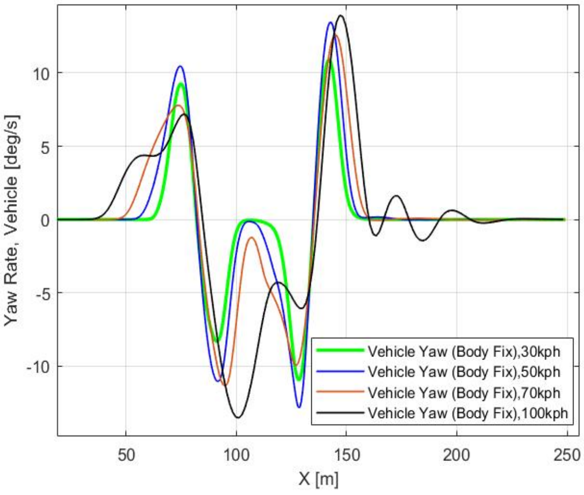


Figure 4.8: Yaw rate of sprung masses vs. time based on kinematic model: External 4WS, DLC, $\mu = 0.85$

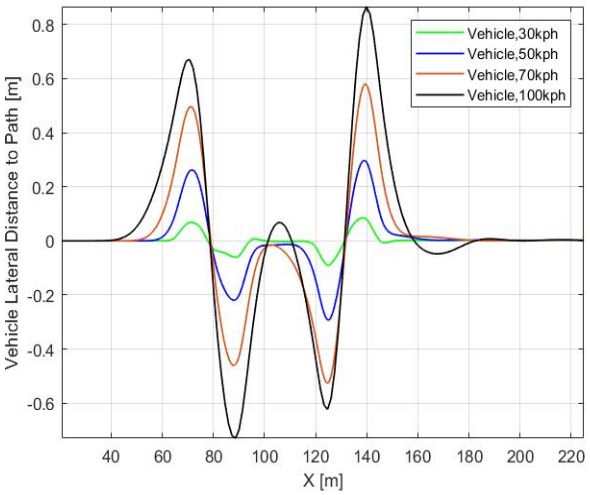


Figure 4.9: Lateral tracking vs.time based on kinematic model: External 4WS, DLC, $\mu = 0.85$

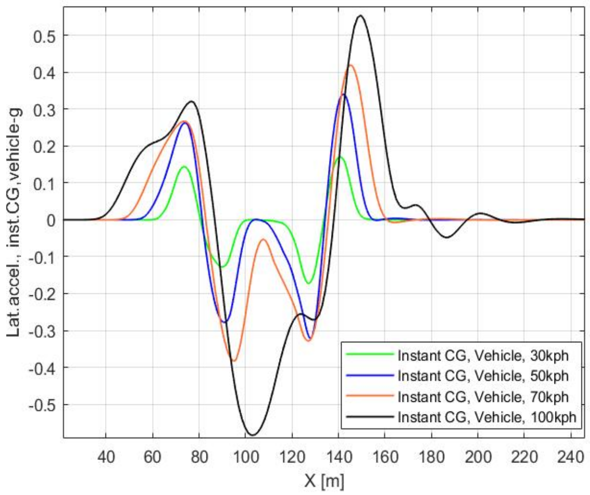


Figure 4.10: Lateral accel. of CG's vs.time based on kinematic model: External 4WS, DLC, $\mu = 0.85$

Moreover, the MPC controller with the kinematic bicycle model is tested on the wet road with $\mu = 0.5$. At speeds lower than 70 kph, the controller performance on the wet road is comparable to that on the dry road. The vehicle follows the path with the same lateral tracking error and weighting factors mentioned in Table 4.6. By increasing the speed to 80 kph, the weighting factor Q_3 should be increased to 500 to add more weights to the heading angle. Although the vehicle follows the path, the lateral error reaches $92cm$ at a speed of 100 kph. Figures 4.11, 4.12, 4.13, 4.14, 4.15, and 4.16 illustrate the controller performance at various speeds test on a wet road, $\mu = 0.5$. The prediction horizon and sampling time are chosen 30 and $0.05s$ respectively. Using a prediction horizon equals 20 decreases the lateral tracking accuracy at low-speed tests; however, the vehicle cannot follow the path accurately at the higher speed tests in this case.

Table 4.6: The value of weighting factors Q, R, and tracking error at different speeds in MPC using kinematic model, $\mu = 0.5$

$\mathbf{V}(\frac{\mathbf{km}}{\mathbf{h}})$	**R**	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$	**Error (cm)**
30	10	1	0	15	9
50	40	1	0	50	29.1
70	100	1	0	150	58.4
100	100	1	0	500	92

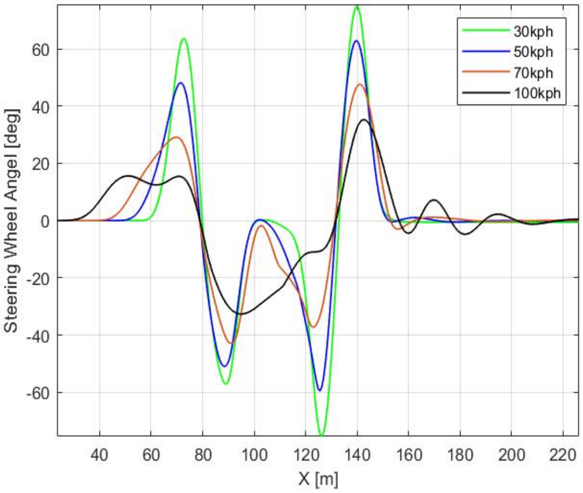


Figure 4.11: Steering handwheel angle based on kinematic model: External 4WS, DLC, $\mu = 0.5$

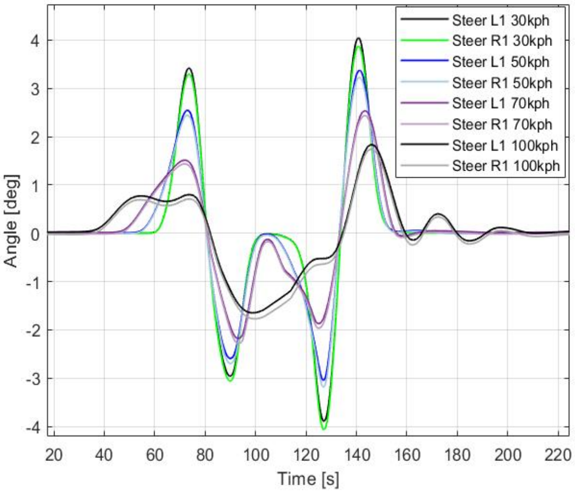


Figure 4.12: Road wheel steer angle - front based on kinematic model: External 4WS, DLC, $\mu = 0.5$

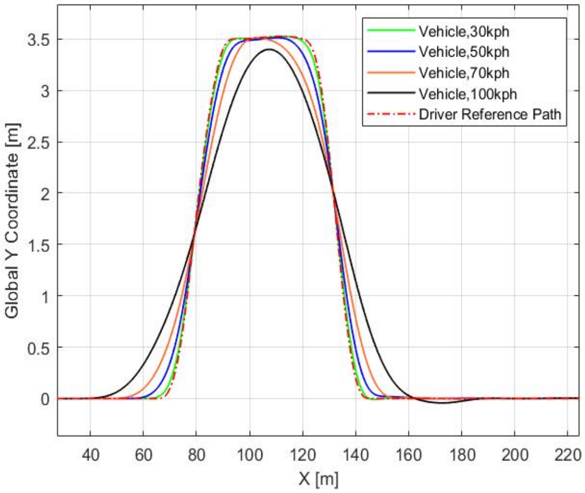


Figure 4.13: Y vs. X – trajectory based on kinematic model: External 4WS, DLC, $\mu = 0.5$

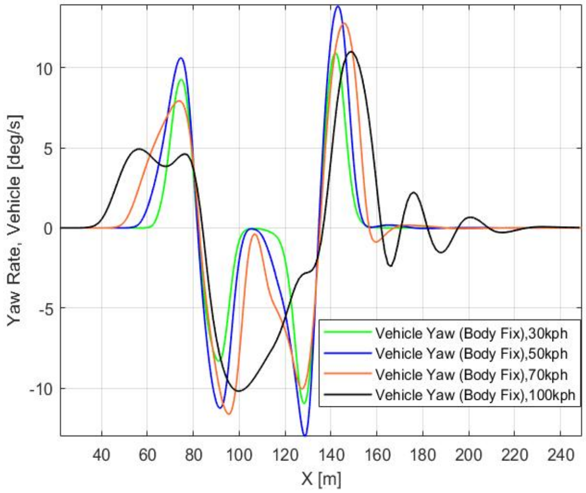


Figure 4.14: Yaw rate of sprung masses vs. time based on kinematic model: 4WS, DLC, $\mu = 0.5$

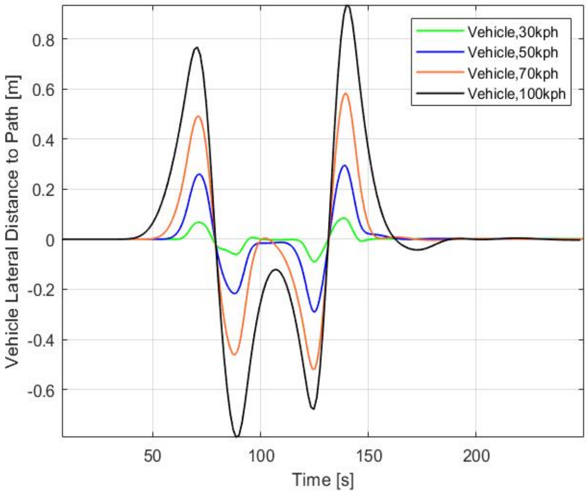


Figure 4.15: Lateral tracking vs.time based on kinematic model: External 4WS, DLC, $\mu = 0.5$

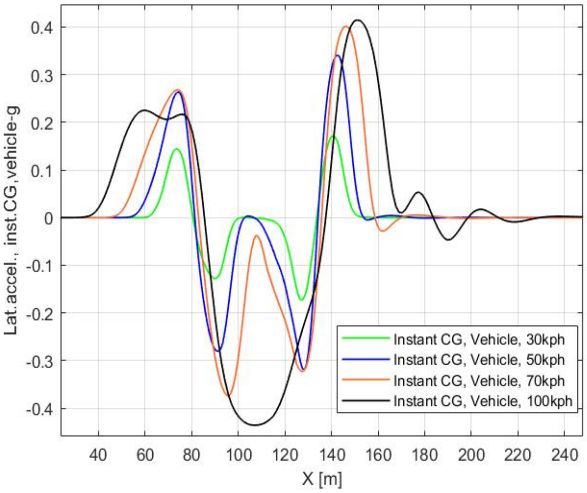


Figure 4.16: Lateral accel. of CG's vs.time based on kinematic model: External 4WS, DLC, $\mu = 0.5$

The controller in both road conditions and at various speeds follows the path.

More simulation plots for the MPC using the nonlinear kinematic model also are provided in appendices A.1.

4.5 MPC Controller Design Using the Dynamic Bicycle Model

This subsection shows the simulation results of the proposed MPC system in chapter 3 using the dynamic bicycle model; the equality constraints, which is related to the dynamics of the vehicle and inequality constraints, control input, and state constraints, on the steering angle of the front wheel, lateral position, sideslip angle, yaw angle, and yaw rate respectively. Table 4.7 lists the upper and lower bounds of equality and inequality constraints.

A constraint on the maximum yaw rate ($\dot{\psi}_{max}$) is defined based on the equation 3.28 to ensure lateral stability of the vehicle.

Table 4.7: The upper band and lower band of equality and inequality constraints for MPC with the dynamic vehicle model

Constraints	**Parameter**	**Lower Bound**	**Upper Bound**
Equality Constraint	Based on Equation 3.27.1	0	0
Inequality Constraints	Steering angle (δ)	-30°	30°
	Lateral position (Y)	-1m	6m
	Sideslip angle (β)	-8°	8°
	Yaw Angle (ψ)	-180°	180°
	Yaw rate ($\dot{\psi}$)	-25degree/sec	$\frac{\mu * g}{V_x}$

The designed MPC with the dynamic model is tested at speeds 30 kph, 50 kph, 70 kph, and 100 kph. At each sampling time $\triangle T = 0.05s$, the MPC controller solves the constrained optimal control problem 3.27. The prediction horizon is $N = 20$, which gives the MPC total preview time of $N * \triangle T = 0.05 * 20 = 1s$. The weighting matrices Q, R, and P of the proposed controller are selected as shown in, Table 4.8, where P is defined as an identity matrix.

Table 4.8: The value of weighting factors Q, R and tracking error at different speeds in MPC using dynamic model, $\mu = 0.85$

$\mathbf{V(\frac{km}{h})}$	**R**	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$	$\mathbf{Q_4}$	**Error (cm)**
30	5	1	0	1	0	3.8
50	30	1	0	1	0	21
70	120	1	0	1	0	38.5
100	200	1	0	100	0	68

4.5.1 Simulation Results

The simulation results of the MPC controller using the dynamic bicycle vehicle model on the dry road, $\mu = 0.85$, are shown in Figures 4.17, 4.18, 4.19, 4.20, 4.21, and 4.22 at the vehicle speeds 30 kph, 50 kph, 70 kph, and 100 kph respectively. The simulation results indicate that the controller can make the vehicle track the predefined path with a relatively small lateral tracking error as shown in the results in the last column of Table 4.8.

By increasing the speed in the DLC maneuver, the weighting matrix R will increase to add more weights to the control input. Although at the higher speeds, more weighting on yaw angle, Q_3, is required to prevent the steering angle and yaw rate from being oscillatory at the end of the DLC maneuver, this change also decreases the lateral tracking error. On the contrary, increasing the Q_3 at the lower speeds causes higher tracking errors during the DLC maneuver. Increasing Q_1, the weight on the lateral position makes the vehicle unstable at low/high speeds. Therefore, Q_1 is chosen as a constant for all test speeds.

However, the lower value of N decreases the lateral error at both low and high speeds; it causes oscillatory steering angle at high speed. Moreover, high values of N increase the computation time of MPC and also result in higher lateral errors in low/high speeds.

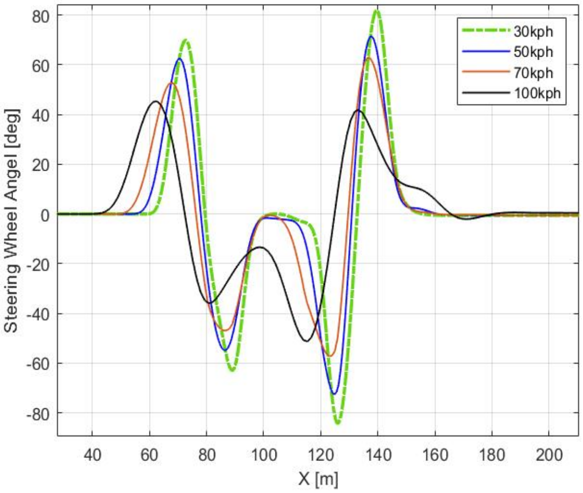


Figure 4.17: Steering handwheel angle based on dynamic model: External 4WS, DLC, $\mu = 0.85$

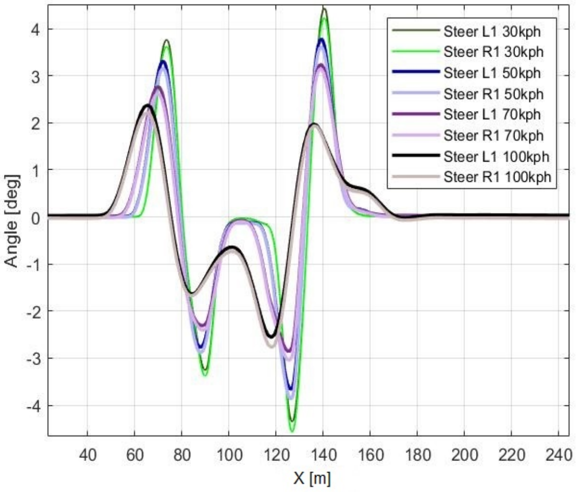


Figure 4.18: Road wheel steer angle - front based on dynamic model: External 4WS, DLC, $\mu = 0.85$

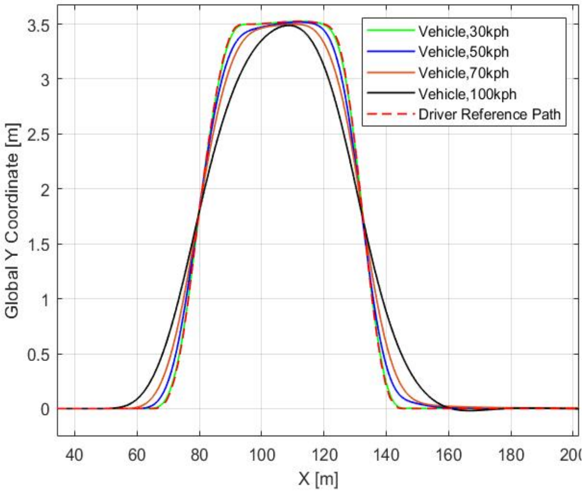


Figure 4.19: Y vs. X – trajectory based on dynamic model: External 4WS, DLC, $\mu = 0.85$

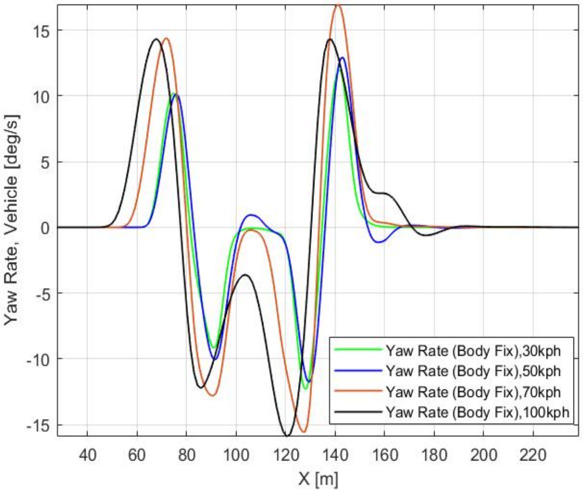


Figure 4.20: Yaw rate of sprung masses vs. time based on dynamic model: External 4WS, DLC, $\mu = 0.85$

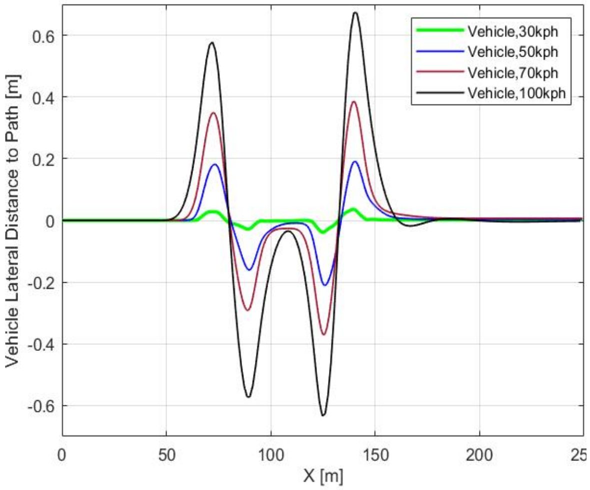


Figure 4.21: Lateral tracking vs.time based on dynamic model: External 4WS, DLC, $\mu = 0.85$

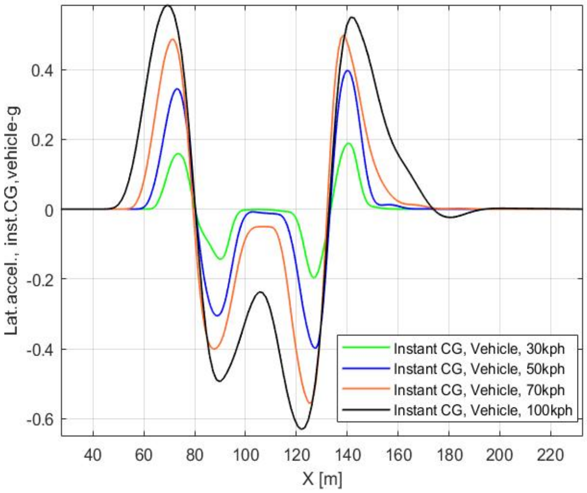


Figure 4.22: Lateral accel. of CG's vs.time based on dynamic model: External 4WS, DLC, $\mu = 0.85$

Moreover, the MPC controller is tested on wet road, $\mu = 0.5$, at various speeds, Figures 4.23, 4.24, 4.25, 4.26, 4.27, and 4.28. The results illustrate that the vehicle follows the path with a relatively small lateral tracking error based on the results in the last column of Table 4.9. By increasing the speed, the weighting factor Q_3 should increase to ensure better path following. Also, the weighting matrix R increases to add more weight to the control input compared to the case on the dry road. The prediction horizon, N, can be changed to 30 at higher speeds to have a smooth path following, Figure 4.29; however, this issue does not affect the lateral tracking error. During the DLC maneuver, at speeds more than 70 kph, overshooting also happens, attenuated by tuning the weighting factor Q_3.

Table 4.9: The value of weighting factors Q, R and tracking error at different speeds, in MPC using dynamic model, $\mu = 0.5$

$\mathbf{V(\frac{km}{h})}$	**R**	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$	$\mathbf{Q_4}$	**Error (cm)**
30	5	1	0	5	0	5.5
50	30	1	0	15	0	24.2
70	120	1	0	20	0	40
100	300	1	0	300	0	126

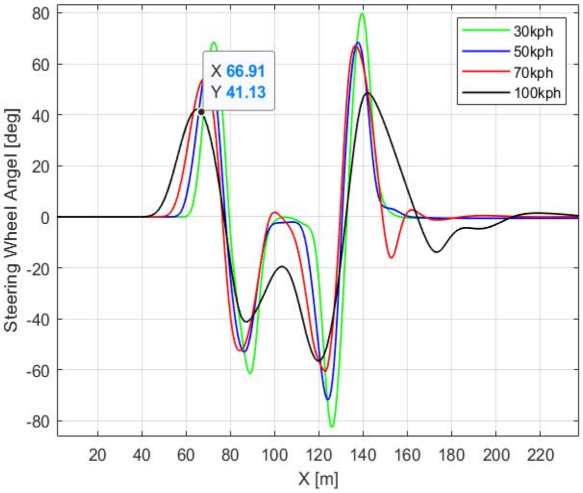


Figure 4.23: Steering handwheel angle based on dynamic model: External 4WS, DLC, $\mu = 0.5$

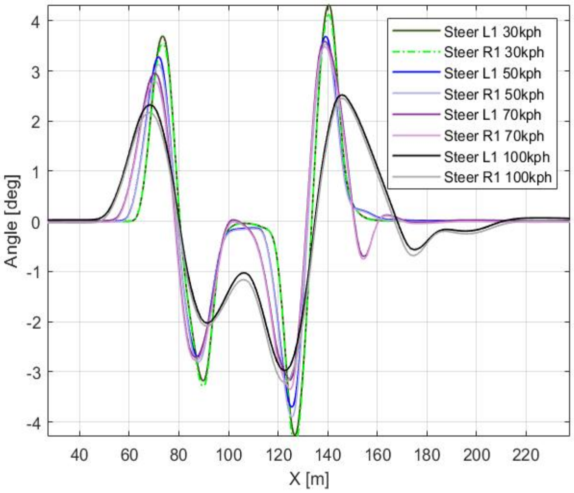


Figure 4.24: Road wheel steer angle - front based on dynamic model: External 4WS, DLC, $\mu = 0.5$

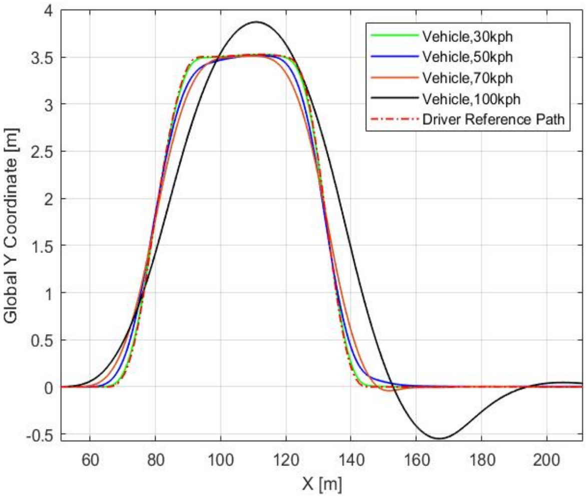


Figure 4.25: Y vs. X – trajectory based on dynamic model: External 4WS, DLC, $\mu = 0.5$

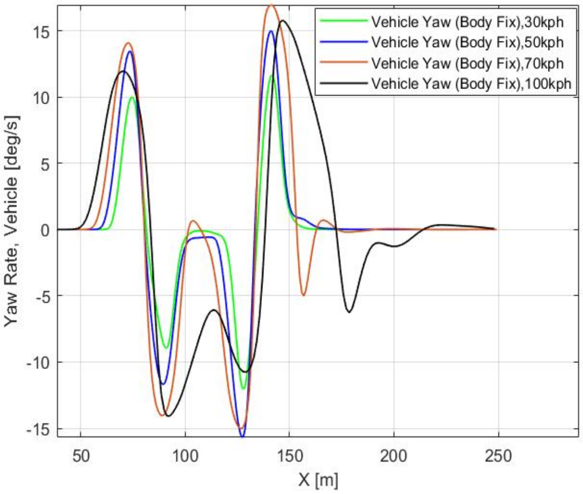


Figure 4.26: Yaw rate of sprung masses vs. time based on dynamic model: External 4WS, DLC, $\mu = 0.5$

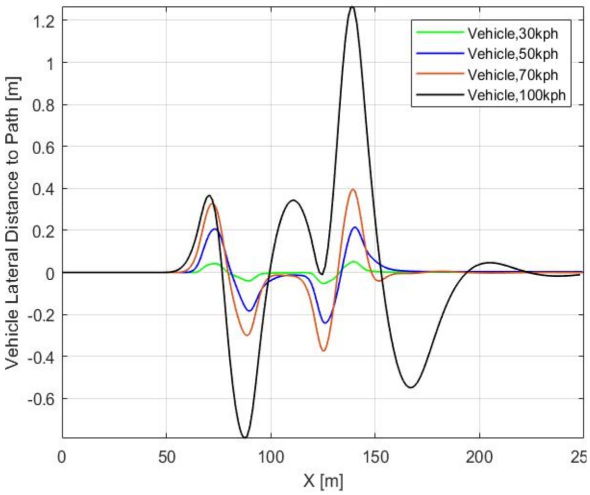


Figure 4.27: Lateral tracking vs.time based on dynamic model: External 4WS, DLC, $\mu = 0.5$

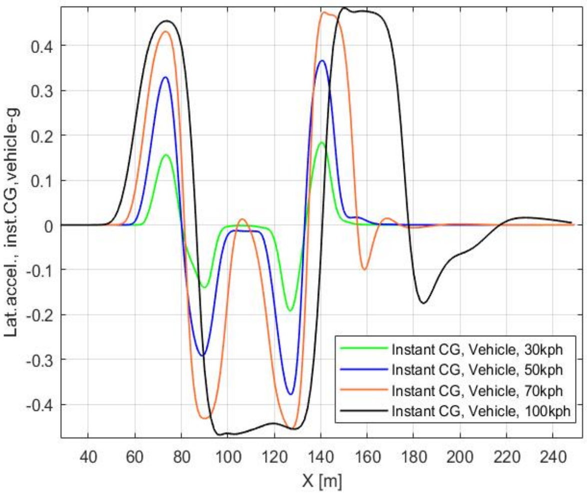


Figure 4.28: Lateral accel. of CG's vs.time based on dynamic model: External 4WS, DLC, $\mu = 0.5$

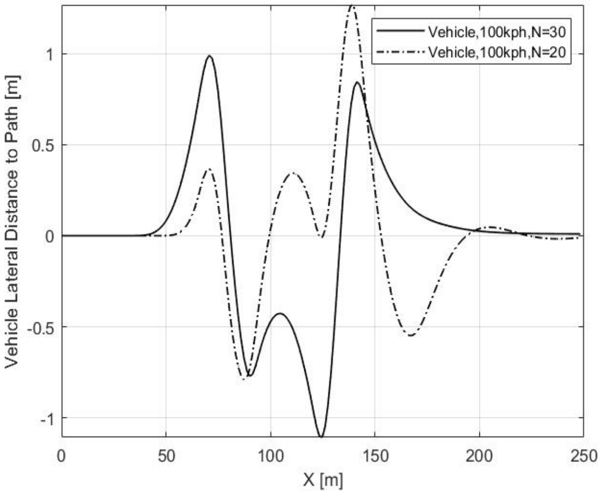


Figure 4.29: The effect of increasing N to 30 on the lateral tracking Error, based on dynamic model: External 4WS, DLC, $\mu = 0.5$

Based on the simulation results in Figures 4.17- 4.29, the vehicle closely follows the predefined DLC path with minimal lateral error as shown in the last column of Tables 4.8 and 4.9. There is higher lateral tracking errors on the wet road, $\mu = 0.5$, compared to the dry road, $\mu = 0.85$. The controller works well with setting the prediction horizon equals to $N = 20$ on both road conditions and at low/high speeds, However, increasing the prediction horizon to $N = 30$ on the wet road prevents the car from slipping during lane changes at speeds > 80 kph, which allows for smoother path following.

More simulation plots for the MPC with the dynamic model are provided in appendices A.2.

4.6 MPC Controller Design Using the Combined Single-Track Bicycle Model

This subsection summarizes the proposed MPC system's simulation results in chapter 3 using the combined single-track vehicle model. The equality constraints related to the dynamics of the vehicle, and inequality constraints on the steering angle of the front wheel as a control input, longitudinal position, lateral position, sideslip angle, yaw angle, and yaw rate as states of the system are defined in Table 4.10 respectively.

Simulations are performed for speeds 30 kph, 50 kph, 70 kph, and 100 kph in order to abide by the speed limits in Table 4.3.

At each sampling time $\triangle T = 0.05s$, the MPC controller solves the constrained optimal control problem in equation 3.27. The prediction horizon is assumed to be $N = 20$, which gives the MPC total preview time of $N * \triangle T = 0.05 * 20 = 1s$.

The weighting factors Q, R, and P of the proposed controller are selected as shown in Table 4.11, where P is defined as an identity matrix.

Table 4.10: The upper band and lower band of equality and inequality constraints for MPC using the combined single-track vehicle model

Constraints	**Parameter**	**Lower Bound**	**Upper Bound**
Equality Constraint	Based on Equation 3.27.1	0	0
Inequality Constraints	Steering angle (δ)	-30°	30°
	Lateral position (Y)	-1m	6m
	Longitudinal Distance (X)	-1m	200m
	Sideslip angle (β)	-8°	8°
	Yaw Angle (ψ)	-180°	180°
	Yaw rate ($\dot{\psi}$)	-25degree/sec	$**\frac{\mu * g}{V_x}$

Table 4.11: The value of weighting factors Q, R and tracking error at different speeds, in MPC using combined single-track model, $\mu = 0.85$

V($\frac{\text{km}}{\text{h}}$)	**R**	**Q$_1$**	**Q$_2$**	**Q$_3$**	**Q$_4$**	**Q$_5$**	**Error (cm)**
30	1	0	1	0	10	0	3.7
50	1	0	1	0	200	0	23
70	100	0	1	0	1	0	37
100	200	0	1	0	100	0	68

4.6.1 Simulation Results

The simulation results of the MPC controller using the combined single track model on the dry road, $\mu = 0.85$, are shown in Figures 4.30, 4.31, 4.32, 4.33, 4.34, and 4.35 at vehicle speeds of 30 *kph*, 50 *kph*, 70 *kph*, and 100 *kph*, respectively. The simulation results indicate that the controller can make the vehicle track the predefined path with relatively small lateral tracking error based on the results in the last column of Table 4.11.

At speeds 30 *kph* and 50 *kph*, the controller switches to the kinematic model, and the weighting factor Q_4 has a significant role in tuning the controller. In contrast, at the speeds

70 *kph* and 100 *kph*, the controller switches to the dynamic model, and the weighting factor R acts as an important weight in controller tuning.

Moreover, increasing the weighting factor Q_4 prevents the steering angle from being oscillatory at the end of the DLC path for the speed over 90 *kph*. Increasing Q_2, the weight on the lateral error makes the vehicle unstable and decreasing it causes overshooting at low/high speeds. Therefore, Q_2 is chosen as a constant at all test speeds. Increasing N results in increasing lateral tracking error and also the computation time. Besides, by decreasing N to 10, lateral tracking error increases at the low-speed, and the vehicle cannot follow the path at high speeds. Increasing the sampling time, e.g., $0.1s$, will require adding more weights on the control input and heading angle and increases the lateral error in low/high speeds.

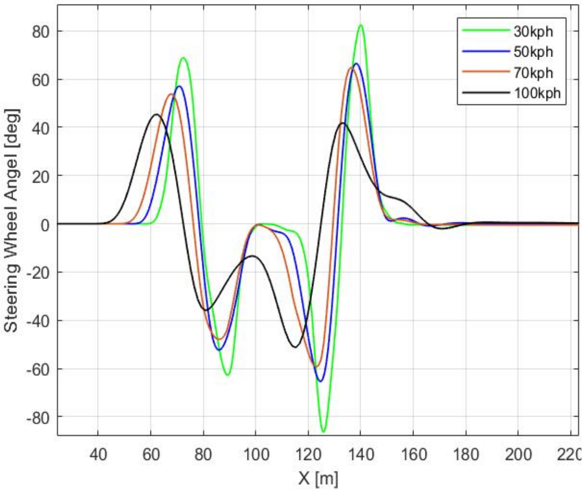


Figure 4.30: Steering handwheel angle based on combined single-track model: External 4WS, DLC, $\mu = 0.85$

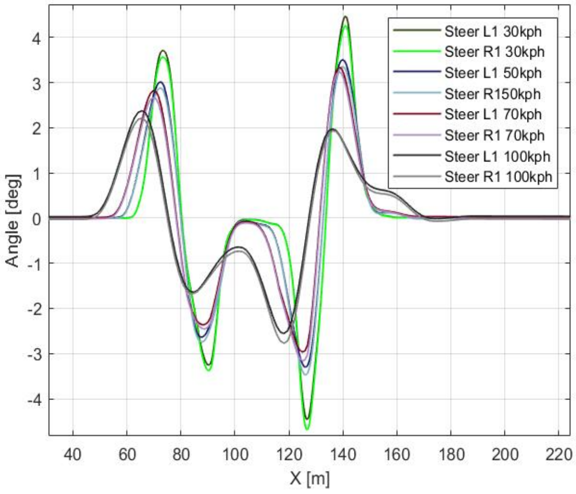


Figure 4.31: Road wheel steer angle - front based on combined single-track model: External 4WS, $\mu = 0.85$

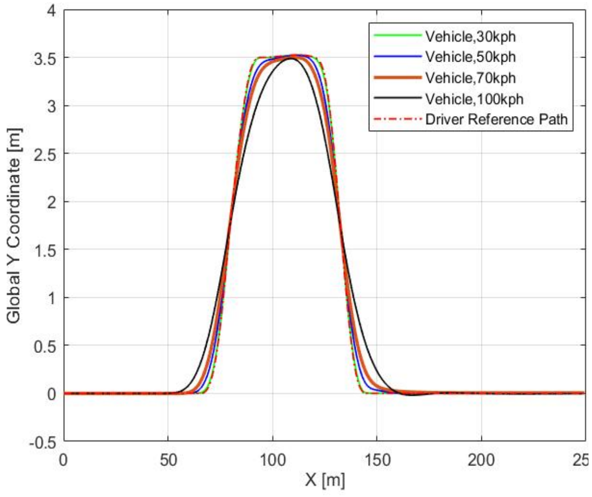


Figure 4.32: Y vs. X – trajectory based on combined single-track model: External 4WS, DLC, $\mu = 0.85$

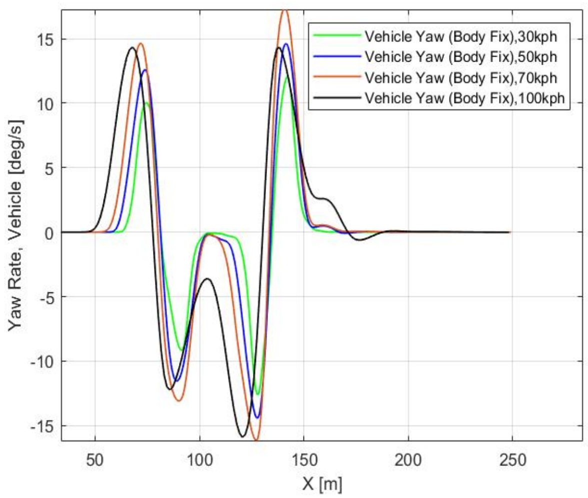


Figure 4.33: Yaw rate of sprung masses vs. time based on combined single-track model, DLC, $\mu = 0.85$

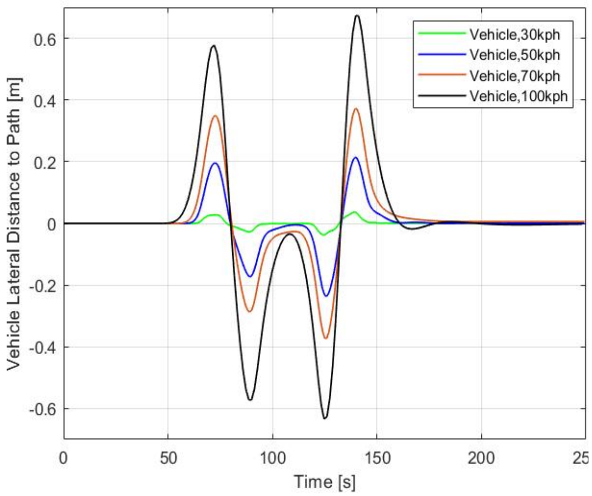


Figure 4.34: Lateral tracking vs.time based on combined single-track model: External 4WS, DLC, $\mu = 0.85$

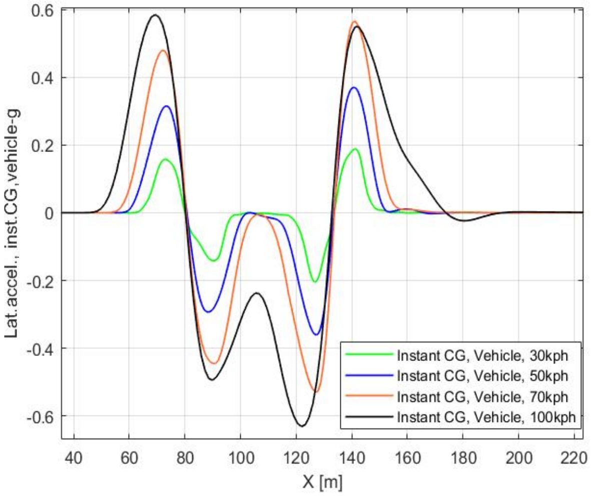


Figure 4.35: Lateral accel. of CG's vs.time based on combined single-track model: External 4WS, $\mu = 0.85$

Moreover, the MPC controller is tested on the wet road, $\mu = 0.5$, at various speeds, Figures 4.36, 4.37, 4.38, 4.39, 4.40, and 4.41. The results illustrate that the vehicle follows the path with a relatively small lateral tracking error based on Table 4.12. By increasing the speed, the weighting factor Q_4 should increase for better path following. At speeds of more than $80kph$, the weighting factor R increases to add more weights on the control input compared to the test case on the dry road. During the DLC maneuver, at speeds higher than 70, overshoot also happens, controlled by tuning the weighting factor Q_4. Prediction horizon N can be changed to 30 at higher speeds to have a smooth path following and decrease the lateral tracking error. Figures 4.42 and 4.43 illustrate the effect of increasing prediction horizon, N on the lateral tracking error and steering angle at the speed of 100 kph. However, at higher speeds increasing the prediction horizon to 30 decreases the lateral tracking error to $76cm$ for $R = 400$ and $Q_4 = 1$.

Table 4.12: The value of weighting factors Q, R and tracking error at different speeds, in MPC with combined single-track model, $\mu = 0.5$

$\mathbf{V(\frac{km}{h})}$	**R**	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$	$\mathbf{Q_4}$	$\mathbf{Q_5}$	**Error (cm)**
30	1	0	1	0	10	0	3.7
50	1	0	1	0	200	0	23
70	100	0	1	0	50	0	40
100	400	0	1	0	100	0	100

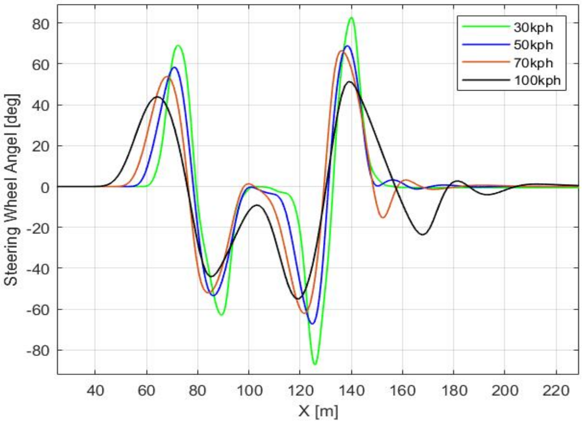


Figure 4.36: Steering handwheel angle based on combined single-track model: External 4WS, DLC, $\mu = 0.5$

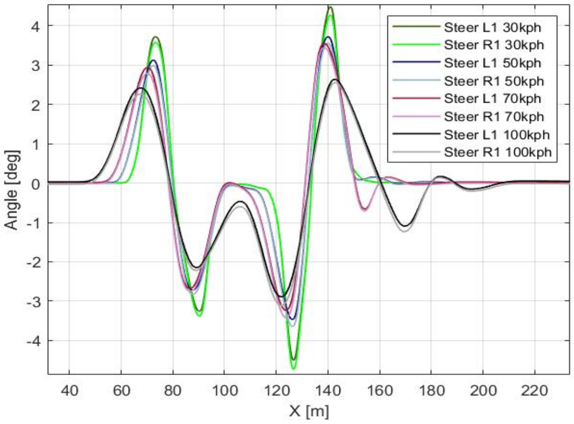


Figure 4.37: Road wheel steer angle front based on combined single-track model: External 4WS, DLC, $\mu = 0.5$

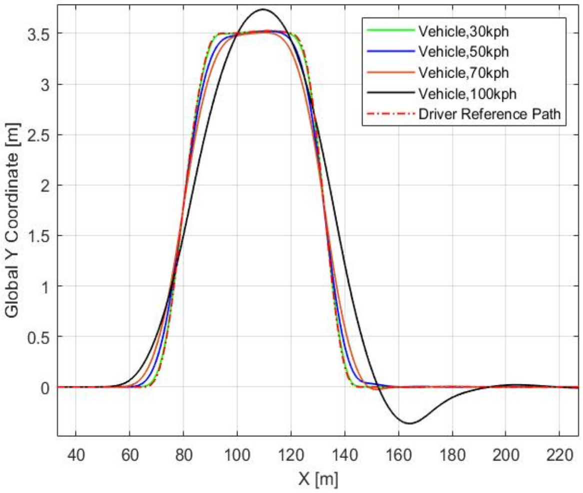


Figure 4.38: Y vs. X – trajectory based on combined single-track model: External 4WS, DLC, $\mu = 0.5$

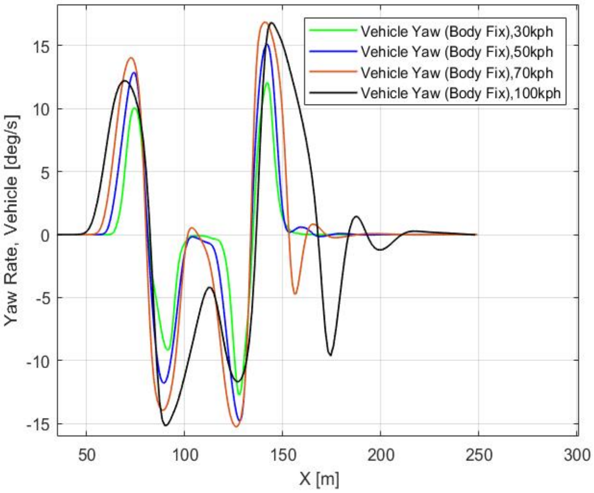


Figure 4.39: Yaw rate of sprung masses vs. time based on combined single-track model, DLC, $\mu = 0.5$

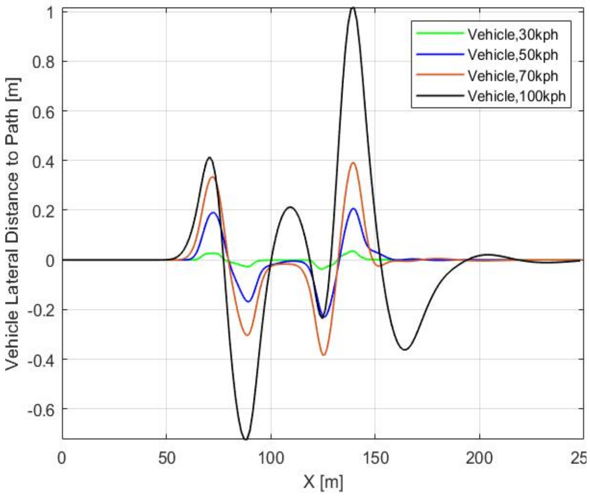


Figure 4.40: Lateral tracking vs.time based on combined single-track model: External 4WS, DLC, $\mu = 0.5$

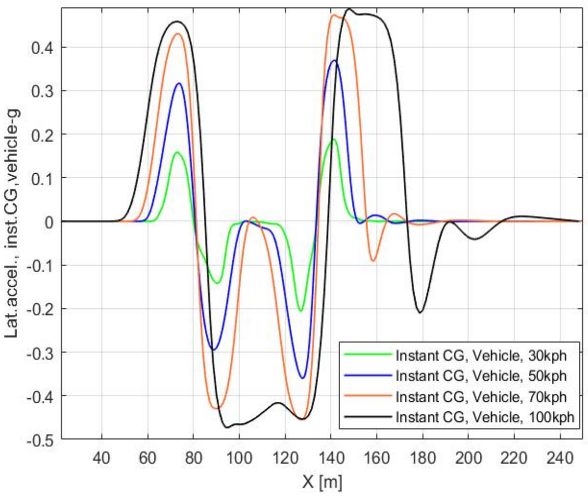


Figure 4.41: Lateral accel. of CG's vs.time based on combined single-track model: External 4WS, $\mu = 0.5$

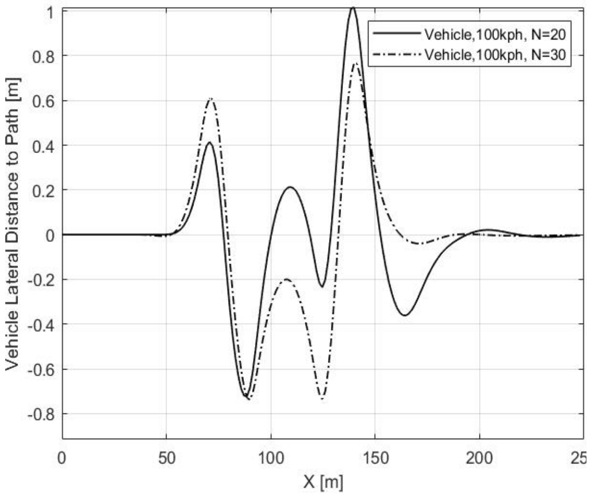


Figure 4.42: Comparing lateral tracking vs.time based on combined single-track model for N=20 and N=30, $\mu = 0.5$

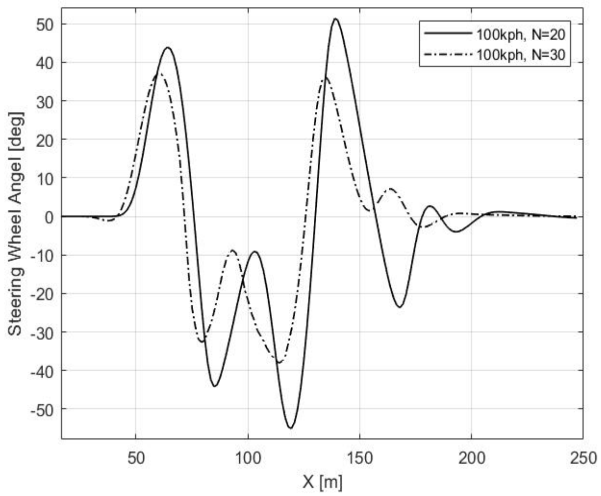


Figure 4.43: Comparing steering handwheel angle based on combined single-track model: for N=20 and N=30, $\mu = 0.5$

For all speeds and road friction coefficients under which the proposed MPC's performance is tested, the controller pulls the system back to the predefined path, and the DLC tracking performance is satisfactory. In all simulations conducted in this chapter, it is assumed that tire forces operate within the linear region of the tire model adopted in Chapter 3.

More plots related to simulation of the proposed MPC using the combined single-track model are available in appendices A.3.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, a reliable path following controller was designed using MPC and different vehicle models to minimize the lateral distance between the vehicle and predefined geometric reference path in the DLC maneuver at constant low/high longitudinal speeds on dry/wet roads. MPCs were also designed to minimize the vehicle's heading and its defined path and limit steering inputs.

The path following controller had been implemented in MATLAB by using an open-source tool, CasADi. The Multiple Shooting technique was applied to breakdown the system integration into shorter time intervals and use the system model as a state constraint in each optimization step. The vehicle models were discretized using the Euler method. The performance of MPC was investigated using CarSim and Simulink. The lateral tracking error was calculated based on the VS Math Models of CarSim and applying interpolation between points across the prediction horizon and at each sampling time using the vehicle forward speed.

The first MPC was designed by using the nonlinear kinematic vehicle model. Based on the simulation results, MPC using the kinematic model worked well for a longer prediction horizon; however, this issue will increase the optimization's computational complexity. MPC using the kinematic model also had less lateral tracking error at high speeds (over 80 *kph*) on the wet road. Decreasing prediction horizon decreased lateral tracking error at low speed test and caused inaccurate path following at higher speeds on the dry road.

In designing the second and third MPC control schemes, the linear dynamic and combined single-track vehicle models were used, respectively. MPCs using the dynamic and combined single-track models worked in a shorter horizon and had less lateral tracking error at all speed tests on the dry road than the MPC with the kinematic model.

In MPC using dynamic model, decreasing the prediction horizon decreased lateral error at low/high speeds and caused oscillatory steering angle at high speed on the dry road. Furthermore, increasing prediction horizon increased computation time and lateral tracking error in high/low speeds on the wet road. Decreasing the prediction horizon in MPC using the combined single-track model increased lateral tracking error at low speed and no accurate path following happened at high speeds on the dry road. On the other hand, increasing the prediction horizon caused more lateral tracking error and computation time on the dry road. However, in this controller, increasing the prediction horizon to 30 caused smooth path following and decreased the lateral tracking error at higher speed on the wet road.

In all various speeds and road friction coefficients for which the controllers' performance was tested, the controllers pulled the system back to the predefined path, and tracking performance was satisfactory.

Overall, MPC with the combined single-track model showed enhanced performance by comparison to the other two schemes for all the tests conducted on wet and dry roads at various vehicle speeds.

5.2 Future Work

The future direction of this research can include:

- Experimental validation on the Chevy bolt
- Other tire models (nonlinear)
- Testing with new maneuvers, break-in-turn, etc
- Adding the speed control inner loop into the MPC design
- Comparison of proposed MPCs with other optimal control schemes for path following
- Test controller in the purpose of Obstacle avoidance
- Computing desired input via reliable path planning methods

References

[1] Mehdi Abroshan, Reza Hajiloo, Ehsan Hashemi, and Amir Khajepour. Model predictive-based tractor-trailer stabilisation using differential braking with experimental verification. *Vehicle System Dynamics*, pages 1–24, 2020.

[2] Auday Al-Mayyahi, Weiji Wang, and Phil Birch. Design of fractional-order controller for trajectory tracking control of a non-holonomic autonomous ground vehicle. *Journal of Control, Automation and Electrical Systems*, 27(1):29–42, 2016.

[3] Noor Hafizah Amer, Hairi Zamzuri, Khisbullah Hudha, and Zulkiffli Abdul Kadir. Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges. *Journal of intelligent & robotic systems*, 86(2):225–254, 2017.

[4] Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[5] Keshav Bimbraw. Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, volume 1, pages 191–198. IEEE, 2015.

[6] Francesco Borrelli, Paolo Falcone, Tamas Keviczky, Jahan Asgari, and Davor Hrovat. Mpc-based approach to active steering for autonomous vehicle systems. *International journal of vehicle autonomous systems*, 3(2-4):265–291, 2005.

[7] Alberto Broggi, Michele Buzzoni, Stefano Debattisti, Paolo Grisleri, Maria Chiara Laghi, Paolo Medici, and Pietro Versari. Extensive tests of autonomous driving technologies. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1403–1415, 2013.

[8] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control.* Springer Science & Business Media, 2013.

[9] Riccardo Coppola and Maurizio Morisio. Connected car: technologies, issues, future trends. *ACM Computing Surveys (CSUR)*, 49(3):1–36, 2016.

[10] Pawan Deshpande. Road safety and accident prevention in india: a review. *Int J Adv Eng Tech*, 64(2):68, 2014.

[11] Ádám Domina and Viktor Tihanyi. Path following controller for autonomous vehicles. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–5. IEEE, 2019.

[12] Asif Faisal, Md Kamruzzaman, Tan Yigitcanlar, and Graham Currie. Understanding autonomous vehicles. *Journal of transport and land use*, 12(1):45–72, 2019.

[13] Hongyan Guo, Dongpu Cao, Hong Chen, Zhenping Sun, and Yunfeng Hu. Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification. *Mechanical Systems and Signal Processing*, 118:41–60, 2019.

[14] Hongyan Guo, Feng Liu, Ru Yu, Zhenping Sun, and Hong Chen. Regional path moving horizon tracking controller design for autonomous ground vehicles. *Science China Information Sciences*, 60(1):013201, 2017.

[15] Jinghua Guo, Yugong Luo, Keqiang Li, and Yifan Dai. Coordinated path-following and direct yaw-moment control of autonomous electric vehicles with sideslip angle estimation. *Mechanical Systems and Signal Processing*, 105:183–199, 2018.

[16] Peng Hang, Xinbo Chen, and Fengmei Luo. Lpv/h_∞ controller design for path tracking of autonomous ground vehicles through four-wheel steering and direct yaw-moment control. *International Journal of Automotive Technology*, 20(4):679–691, 2019.

[17] Chih-Lyang Hwang, Chang-Chia Yang, and John Y Hung. Path tracking of an autonomous ground vehicle with different payloads by hierarchical improved fuzzy dynamic sliding-mode control. *IEEE Transactions on Fuzzy Systems*, 26(2):899–914, 2017.

[18] Daxiong Ji, Jian Liu, Hongyu Zhao, and Yiqun Wang. Path following of autonomous vehicle in 2d space using multivariable sliding mode control. *Journal of Robotics*, 2014, 2014.

[19] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Transactions on Vehicular Technology*, 66(2):952–964, 2016.

[20] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[21] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099. IEEE, 2015.

[22] X Krasniqi and E Hajrizi. Use of iot technology to drive the automotive industry from connected to full autonomous vehicles. *IFAC-PapersOnLine*, 49(29):269–274, 2016.

[23] Junho Lee and Hyuk-Jun Chang. Analysis of explicit model predictive control for path-following control. *PloS one*, 13(3):e0194110, 2018.

[24] Hao Li, Jie Luo, Shu Yan, Min Zhu, Qianyang Hu, and Zhennan Liu. Research on parking control of bus based on improved pure pursuit algorithms. In *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 21–26. IEEE, 2019.

[25] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[26] Prashant Mhaskar. Robust model predictive control design for fault-tolerant control of process systems. *Industrial & engineering chemistry research*, 45(25):8565–8574, 2006.

[27] Salama A Mostafa, Mohd Sharifuddin Ahmad, and Aida Mustapha. Adjustable autonomy: a systematic literature review. *Artificial Intelligence Review*, 51(2):149–186, 2019.

[28] Said Munzir, Vera Halfiani, et al. An optimal control solution using multiple shooting method. *Bulletin of Mathematics*, 4(02):143–160, 2012.

[29] Elnaz Namazi, Jingyue Li, and Chaoru Lu. Intelligent intersection management systems considering autonomous vehicles: a systematic literature review. *IEEE Access*, 7:91946–91965, 2019.

[30] Lluis Pacheco and Ningsu Luo. Testing pid and mpc performance for mobile robot local path-following. *International Journal of Advanced Robotic Systems*, 12(11):155, 2015.

[31] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H Ang. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1):6, 2017.

[32] Guilherme V Raffo, Guilherme K Gomes, Julio E Normey-Rico, Christian R Kelber, and Leandro B Becker. A predictive controller for autonomous vehicle path tracking. *IEEE transactions on intelligent transportation systems*, 10(1):92–102, 2009.

[33] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[34] Abhijeet Ravankar, Ankit A Ravankar, Yukinori Kobayashi, Yohei Hoshino, and Chao-Chung Peng. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors*, 18(9):3170, 2018.

[35] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[36] Robert Ritschel, Frank Schrödel, Juliane Hädrich, and Jens Jäkel. Nonlinear model predictive path-following control for highly automated driving. *IFAC-PapersOnLine*, 52(8):350–355, 2019.

[37] Juan Rosenzweig and Michael Bartl. A review and analysis of literature on autonomous driving. *E-Journal Making-of Innovation*, pages 1–57, 2015.

[38] Supriya B Sarkar and B Chandra Mohan. Review on autonomous vehicle challenges. In *First International Conference on Artificial Intelligence and Cognitive Computing*, pages 593–603. Springer, 2019.

[39] Steven Smith. Preparing for a driverless future. *Admin. & Reg. L. News*, 42:4, 2016.

[40] Chuanyang Sun, Xin Zhang, Lihe Xi, and Ying Tian. Design of a path-tracking steering controller for autonomous vehicles. *Energies*, 11(6):1451, 2018.

[41] Chuanyang Sun, Xin Zhang, Quan Zhou, and Ying Tian. A model predictive controller with switched tracking error for autonomous vehicle path tracking. *IEEE Access*, 7:53103–53114, 2019.

[42] Hua Wang, Xi Chen, Yu Chen, Baoming Li, and Zhonghua Miao. Trajectory tracking and speed control of cleaning vehicle based on improved pure pursuit algorithm. In *2019 Chinese Control Conference (CCC)*, pages 4348–4353. IEEE, 2019.

[43] Rongrong Wang, Hui Jing, Chuan Hu, Fengjun Yan, and Nan Chen. Robust $\mathcal{H}_\infty$ path following control for autonomous ground vehicles with delay and data dropout. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2042–2050, 2016.

[44] Fitri Yakub, Aminudin Abu, Shamsul Sarip, and Yasuchika Mori. Study of model predictive control for path-following autonomous ground vehicle control under crosswind effect. *Journal of Control Science and Engineering*, 2016, 2016.

[45] Qiangqiang Yao and Ying Tian. A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking. *Applied Sciences*, 9(22):4739, 2019.

[46] Pan Zhao, Jiajia Chen, Yan Song, Xiang Tao, Tiejuan Xu, and Tao Mei. Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems*, 9(2):44, 2012.

[47] Hongliang Zhou, Levent Güvenç, and Zhiyuan Liu. Design and evaluation of path following controller based on mpc for autonomous vehicle. In *2017 36th Chinese Control Conference (CCC)*, pages 9934–9939. IEEE, 2017.

APPENDICES

Appendix A

Simulation Plots

A.1 Simulation Plots of MPC Using Kinematic Model

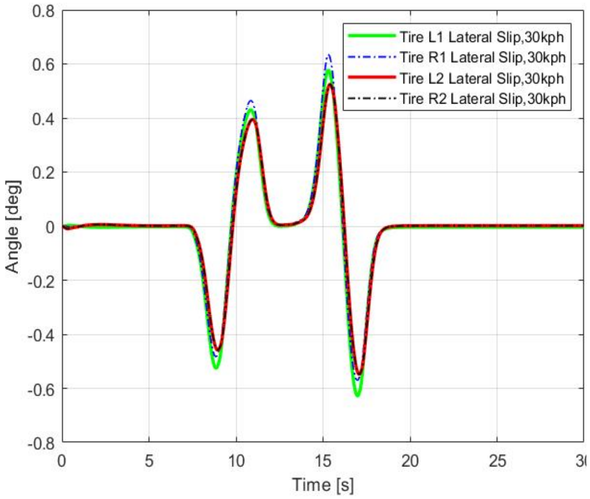


Figure A.1: Slip angles based on kinematic model, External 4WS, DLC, $V = 30\ kph$, $\mu = 0.85$

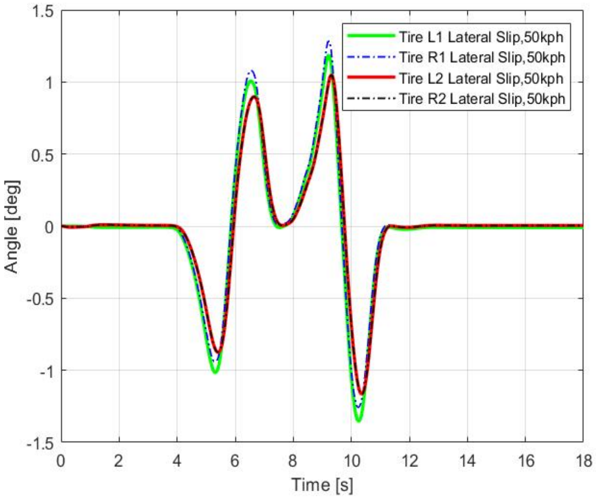
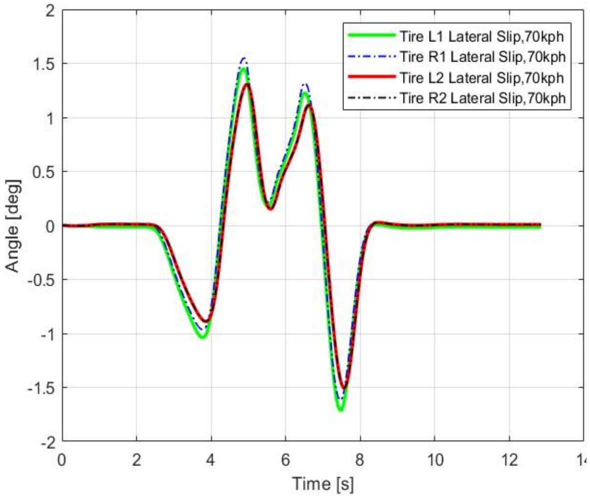


Figure A.2: Slip angles based on kinematic model: External 4WS, DLC, $V = 50\ kph$, $\mu = 0.85$



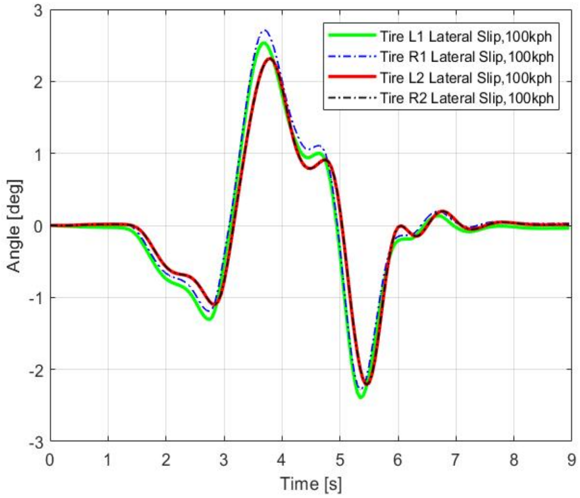
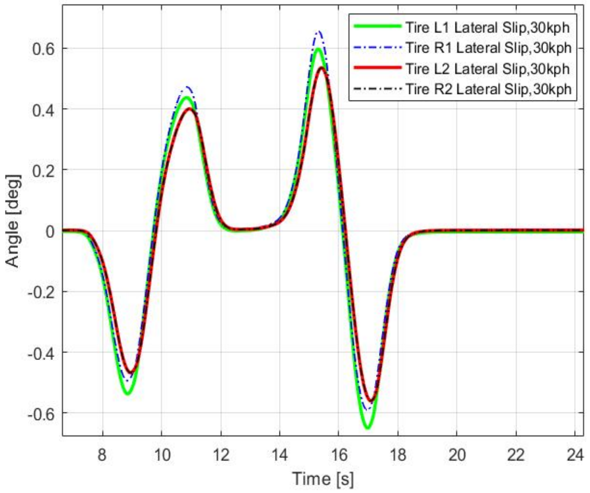


Figure A.3: Slip angles based on kinematic model:External 4WS, DLC, $V = 70\ kph$, $\mu = 0.85$

Figure A.4: Slip angles based on kinematic model:External 4WS, DLC, $V = 100\ kph$, $\mu = 0.85$



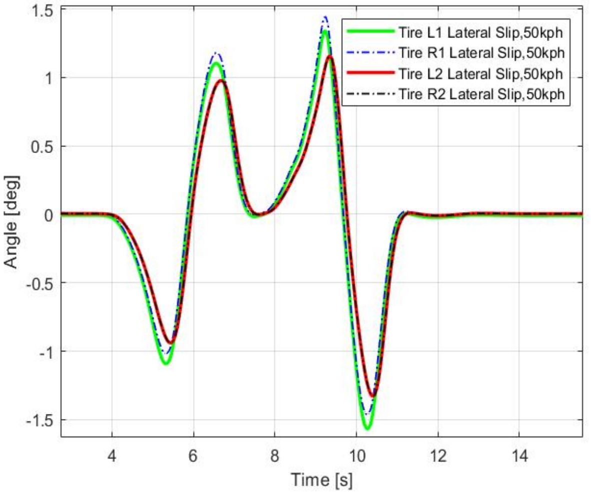


Figure A.5: Slip angles based on kinematic model:External 4WS, DLC, $V = 30\ kph$, $\mu = 0.5$

Figure A.6: Slip angles based on kinematic model:External 4WS, DLC, $V = 50\ kph$, $\mu = 0.5$

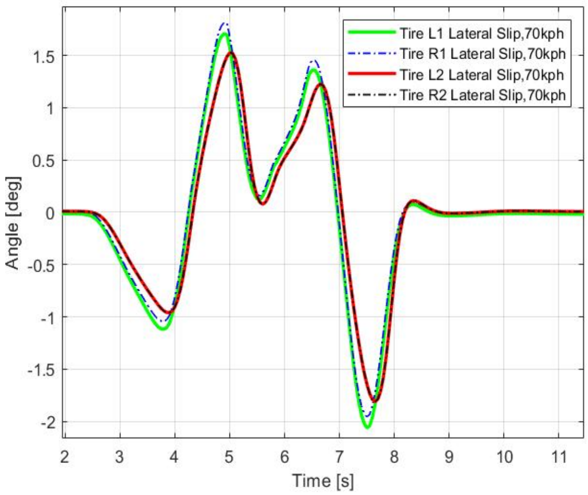


Figure A.7: Slip angles based on kinematic model:External 4WS, DLC, $V = 70\ kph$, $\mu = 0.5$

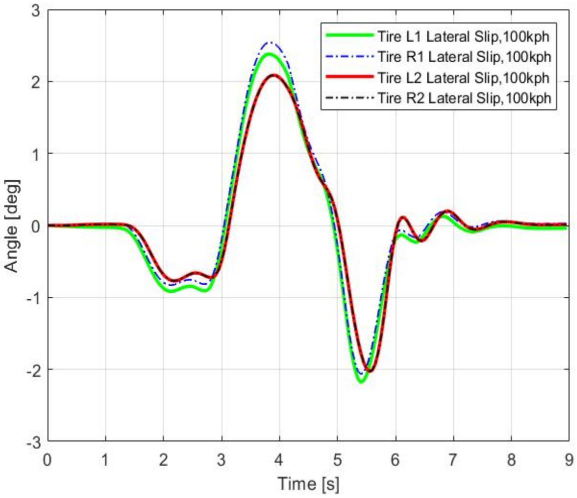


Figure A.8: Slip angles based on kinematic model:External 4WS, DLC, $V = 100\ kph$, $\mu = 0.5$

A.2 Simulation Plots of MPC Using Dynamic Model

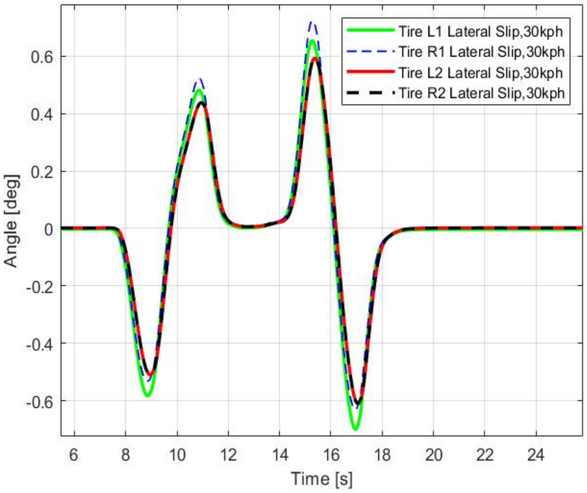


Figure A.9: Slip angles based on dynamic model:External 4WS, DLC, $V = 30\ kph$, $\mu = 0.85$

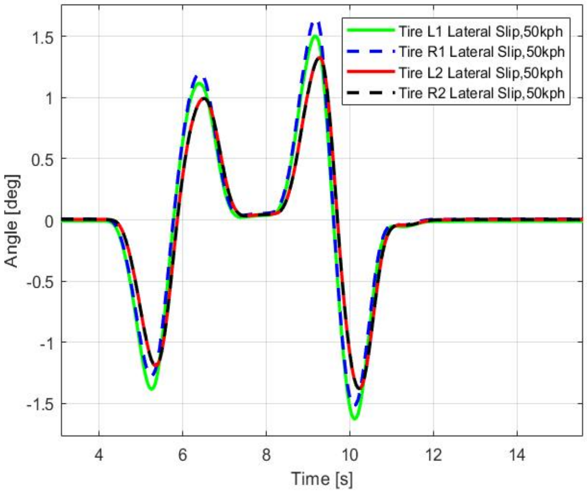


Figure A.10: Slip angles based on dynamic model:External 4WS, DLC, $V = 50\ kph$, $\mu = 0.85$

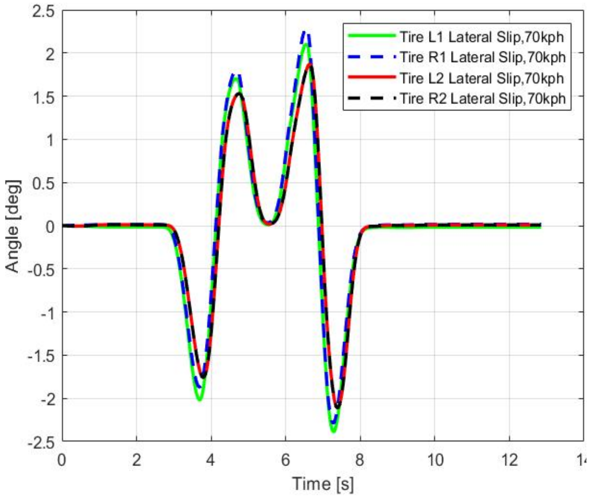


Figure A.11: Slip angles based on dynamic model:External 4WS, DLC, $V = 70\ kph$, $\mu = 0.85$

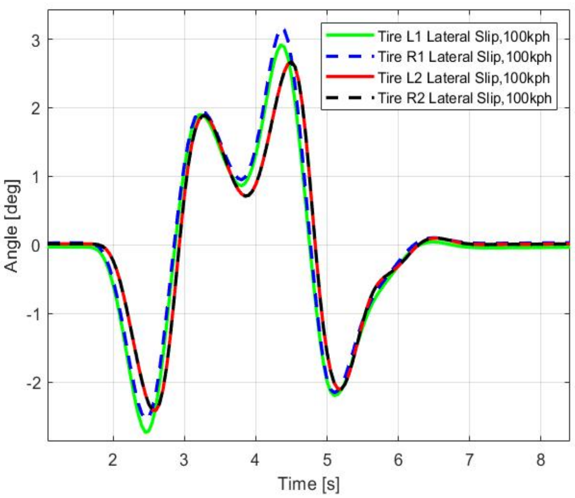


Figure A.12: Slip angles based on dynamic model:External 4WS, DLC, $V = 100\ kph$, $\mu = 0.85$

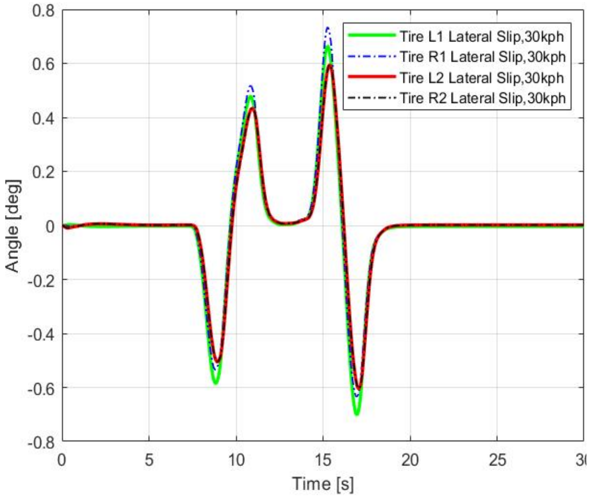


Figure A.13: Slip angles based on dynamic model:External 4WS, DLC, $V = 30\ kph$, $\mu = 0.5$

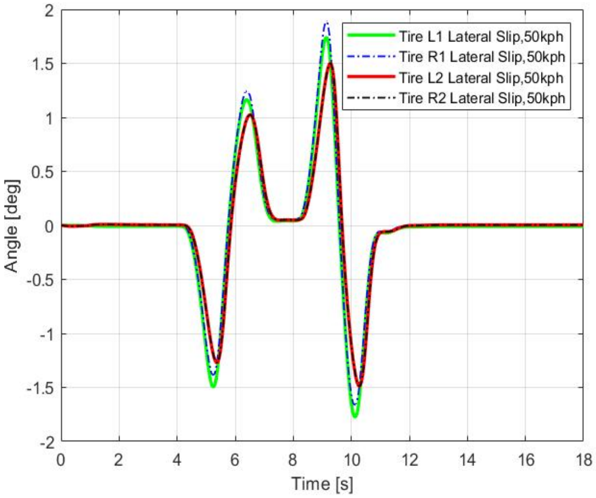


Figure A.14: Slip angles based on dynamic model:External 4WS, DLC, $V = 50\ kph$, $\mu = 0.5$

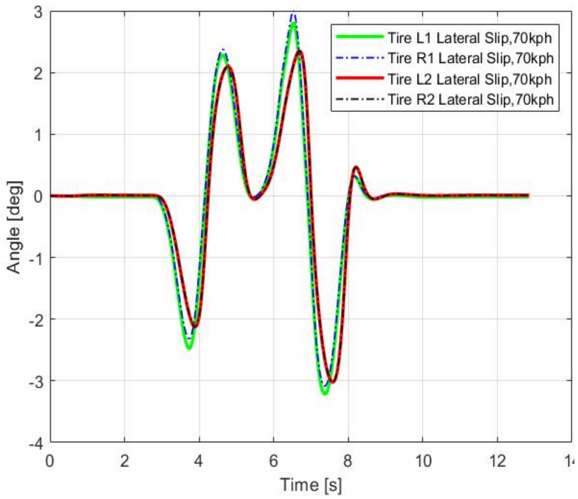


Figure A.15: Slip angles based on dynamic model:External 4WS, DLC, $V = 70\ kph$, $\mu = 0.5$

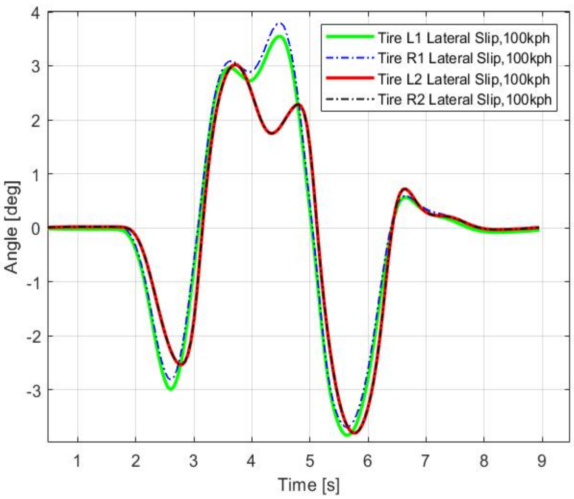


Figure A.16: Slip angles based on dynamic model:External 4WS, DLC, $V = 100\ kph$, $\mu = 0.5$

A.3 Simulation Plots of MPC Using Combined Single-Track Model

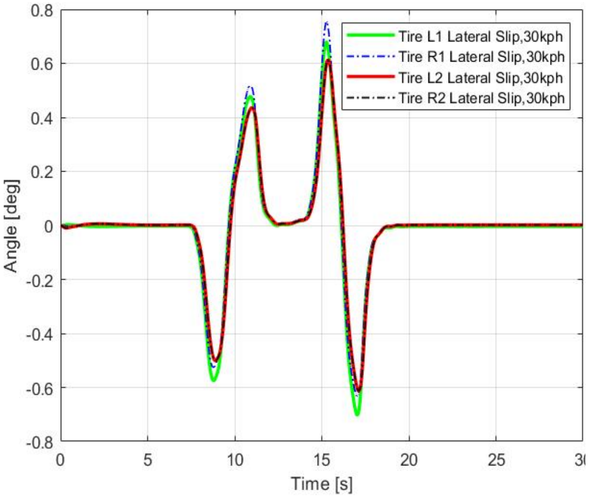


Figure A.17: Slip angles based on combined single-track model:External 4WS, DLC, $V = 30\ kph$, $\mu = 0.85$

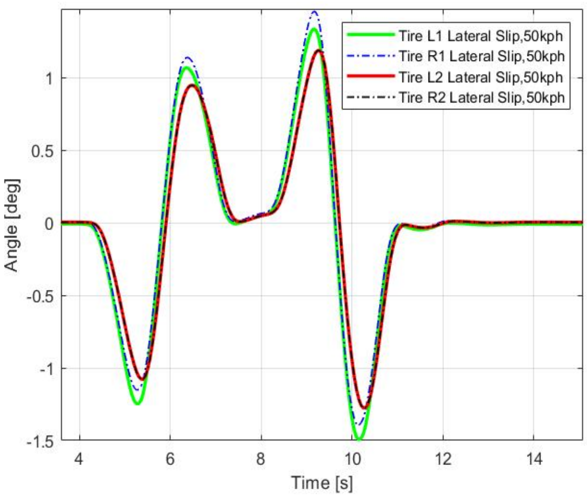


Figure A.18: Slip angles based on combined single-track model:External 4WS, DLC, $V = 50\ kph$, $\mu = 0.85$

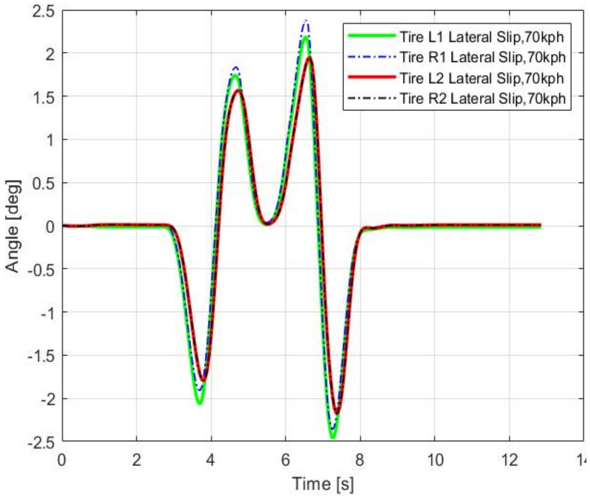


Figure A.19: Slip angles based on combined single-track model:External 4WS, DLC, $V = 70\ kph$, $\mu = 0.85$

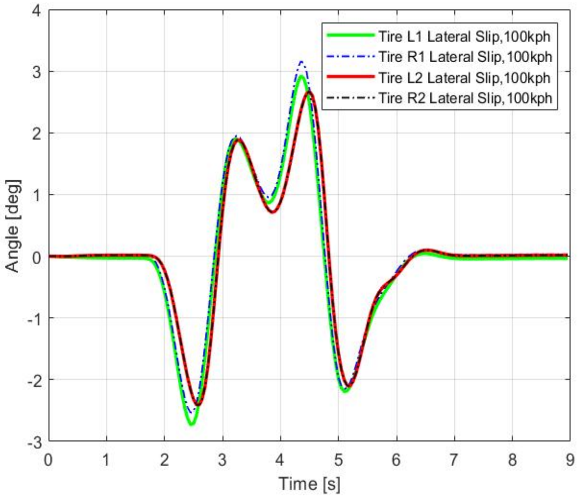


Figure A.20: Slip angles based on combined single-track model:External 4WS, DLC, $V = 100\ kph$, $\mu = 0.85$

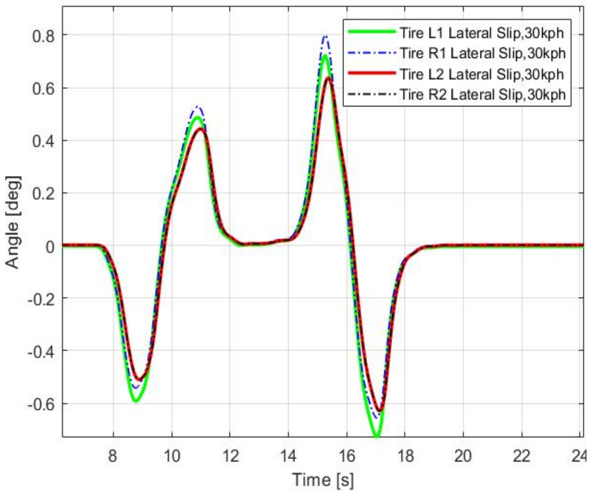


Figure A.21: Slip angles based on combined single-track model:External 4WS, DLC, $V = 30\ kph$, $\mu = 0.5$

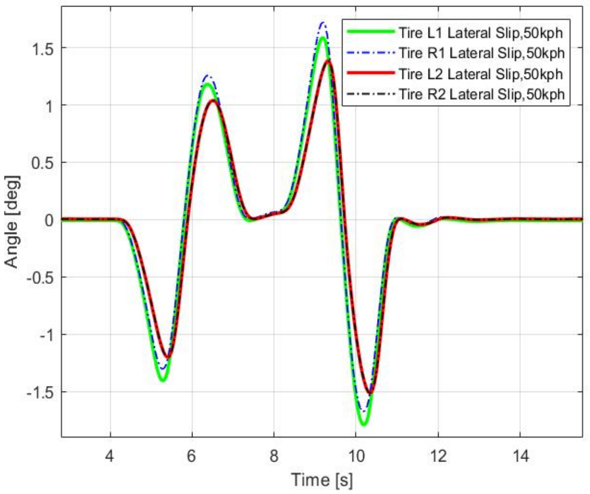


Figure A.22: Slip angles based on combined single-track model:External 4WS, DLC, $V = 50\ kph$, $\mu = 0.5$

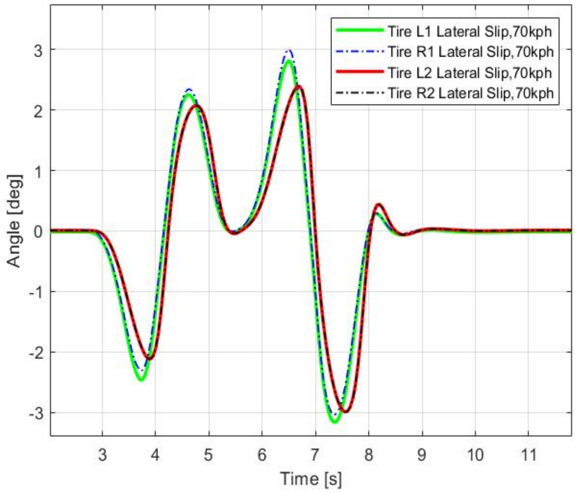


Figure A.23: Slip angles based on combined single-track model:External 4WS, DLC, $V = 70\ kph$, $\mu = 0.5$

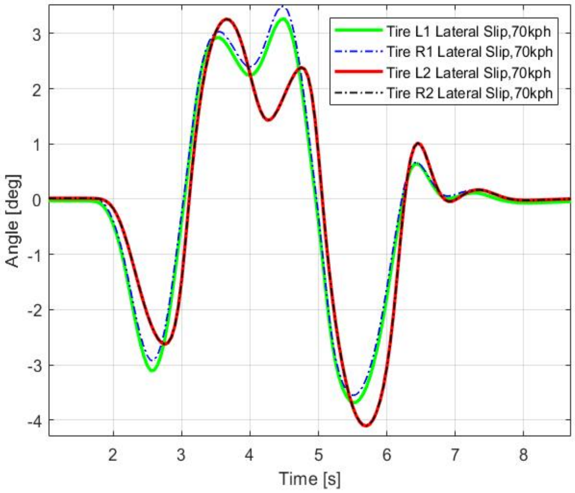


Figure A.24: Slip angles based on combined single-track model:External 4WS, DLC, $V = 100\ kph$, $\mu = 0.5$