Prioritized Obstacle Avoidance in Motion Planning of Autonomous Vehicles

by

Yadollah Rasekhipour

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2017

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Farrokh Jenbi-Sharifi Professor, Mechanical and Industrial Engineering
Supervisor(s)	Amir Khajepour Professor, Mechanical and Mechatronics Engineering
Internal Member	Baris Fidan Associate Professor, Mechanical and Mechatronics Engineering
Internal Member	William Melek Professor, Mechanical and Mechatronics Engineering
Internal-external Member	Krzysztof Czarnecki Professor, Electrical and Computer Engineering

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

Driver errors are a critical factor of the majority of car crashes. Autonomous vehicles take drivers and driver errors out of the equation, so they are being developed to reduce car crashes. However, in some situations, a crash is unavoidable even for an autonomous vehicle. An autonomous vehicle is expected to behave properly in such a situation. Crashing into different obstacles have different costs based on the injury or damage the crash might cause. In an imminent crash situation, an autonomous vehicle is expected to consider these costs and plan a trajectory that avoids the obstacles with the highest priorities.

In this thesis, a motion planning Model Predictive Controller (MPC) has been developed that plans the vehicle's trajectories based on the obstacle's priorities. Motion planning MPCs usually use potential fields or obstacle constraints for obstacle avoidance. However, they treat all the obstacles in the same way. Two methods have been developed in this thesis to prioritize obstacles in motion planning. The first method prioritizes obstacles based on their avoidance necessities. It categorizes obstacles as crossable and non-crossable, and assigns a potential function to each category corresponding to its avoidance necessity. The second method prioritizes obstacles based on their corresponding crash costs. It applies lexicographic optimization on the MPC to prioritize the non-crossable obstacles according to their crash costs by prioritizing their corresponding constraints.

A motion planning MPC problem is generally a nonlinear MPC problem. It is usually approximated by a quadratic MPC problem to become implementable in real time. In this thesis, a quadratic motion planning MPC has been developed. This MPC has a linear vehicle model and linear vehicle and obstacle constraints. The linear vehicle model along with the linear vehicle constraints should be able to model the nonlinear vehicle behavior. A linear bicycle model has been utilized, and linear tire constraints have been developed such that they can model the nonlinear vehicle behavior at the tire force limits. Moreover, a linear obstacle constraint set misses some of the feasible trajectories in the process of convexifying the obstacle-free area. An iterative obstacle avoidance method has been developed in this thesis to reduce the number of feasible trajectories missed due to the convexification.

The performance of the developed motion planning MPC has been evaluated in a computer simulation with a high fidelity vehicle model. The MPC has been simulated for test scenarios to evaluate its performance in autonomous driving and prioritizing obstacles. The capabilities of the developed tire constraints and the iterative obstacle avoidance method have also been observed. The motion planning MPC has also been implemented on an autonomous test vehicle platform to show that it is implementable in real time and to validate the simulation results.

Acknowledgements

Though only my name is on the cover of this thesis, many people have contributed to its production. I owe my gratitude to all who have made this work possible.

Foremost, I would like to thank my supervisor, Prof. Amir Khajepour, for the continuous support of my PhD study and related research, for his patience, motivation, and genuine concern.

My sincere thanks also goes to Prof. JianWei Gong in Beijing Institute of Technology who provided me an opportunity to join their team as an exchange student, and who gave access to the laboratory and research facilities. I would also like to thank Qi Wang and Boyang Wang for their technical support and their help in making experimental tests possible.

I also would like to acknowledge the financial support of Automotive Partnership Canada, Ontario Research Fund, and General Motors.

I thank my fellow labmates Milad Jalali, Ehsan Heshemi, Mohammad Pirani, Soheil Mohagheghi Fard, Saeid Khosravani, and Iman Fadakar for the stimulating discussions, and for all the fun we have had over the last four years.

Last but not the least, I would like to thank my family for supporting me spiritually throughout writing this thesis and my life in general.

Dedication

This dissertation is dedicated to my family, especially my parents, for their constant support and encouragement throughout my life.

Table of Contents

List of Figures

List of Tables

Nomenclature

α_f	Sideslip angle of front tires
α_r	Sideslip angle of rear tires
Δu_{a_i}	Approaching velocity of i^{th} non-crossable obstacle in longitudinal direction
Δv_{a_i}	Approaching velocity of i^{th} non-crossable obstacle in lateral direction
ΔX_0	Longitudinal distance threshold
ΔY_R	Lateral offset of road
δ	Vehicle's steering angle
$\Delta\delta$	Change of steering angle in one step
$\Delta\delta_{max}$	Maximum change of steering angle in one step
δ_{max}	Maximum steering angle
η_i	Rotated signed distance coordinate Y-axis
γ	Signed distance angle
$\hat{\boldsymbol{x}}$	Predicted state vector
$\mathcal{P}_l$	Iteration set of l^{th} obstacle priority order
$\boldsymbol{\lambda}$	Slack variable weighting matrix

$\boldsymbol{\lambda}_i$	Slack variable weighting matrix of i^{th} objective function
μ	Tire-road friction coefficient
ψ	Obstacle's constraint angle
τ_i	i^{th} binary variable
$\boldsymbol{Q}$	Tracking weighting matrix
$\boldsymbol{R}$	Input weighting matrix
$\boldsymbol{S}$	Input change weighting matrix
$\boldsymbol{C}_s$	Constraint output matrix
$\boldsymbol{D}_s$	Constraint feedforward matrix
$\boldsymbol{E}_s$	Constraint binary matrix
$\boldsymbol{f}$	Vector of objective functions
θ	Vehicle's orientation
θ_e	Maximum heading prediction error
$\tilde{\boldsymbol{x}}_d$	Estimated delayed state vector
$\tilde{\boldsymbol{x}}_d$	Predicted delayed state vector
$\tilde{\boldsymbol{x}}_r$	Estimated non-delayed state vector
ξ_i	Rotated signed distance coordinate X-axis
a_i	Intensity parameter of potential function of i^{th} non-crossable obstacle
a_j	Intensity parameter of potential function of j^{th} crossable obstacle
a_n	Normal acceleration

a_q	Intensity parameter of potential function of q^{th} lane marker
a_x	Vehicle's longitudinal acceleration
a_{max}	Maximum acceleration
b_i	Shape parameter of potential function of i^{th} non-crossable obstacle
b_j	Shape parameter of potential function of j^{th} crossable obstacle
C_f	Cornering stiffness of front tires
C_r	Cornering stiffness of rear tires
D_a	Allowed distance from lane marker
f_i	i^{th} objective function
f_i^*	Optimal value of i^{th} objective function
f_l^p	Optimal value of objective function of l^{th} obstacle priority order at p^{th} iteration of IQMPC
F_{x_T-max}	Maximum total longitudinal tire force
F_{x_T}	Total longitudinal force of tires
F_{y_f-max}	Maximum total front lateral tire force
F_{y_f0-max}	Nominal maximum total front lateral tire force
F_{y_f}	Total lateral force of front tires
F_{y_r-max}	Maximum total rear lateral tire force
F_{y_r0-max}	Nominal maximum total rear lateral tire force
F_{y_r}	Total lateral force of rear tires
F_{z_f0}	Nominal front vertical tire force

F_{z_f}	Front vertical tire force
F_{z_r0}	Nominal rear vertical tire force
F_{z_r}	Rear vertical tire force
g	Double-variable potential function
g_T	Transferred potential function
h	Height of vehicle's center of gravity
h	Single-variable potential function
I_z	Vehicle's momentum of inertia around its vertical axis
k	Prediction time step
k_a	Actuation delay
k_e	Estimation delay
L_0	Obstacle's added length
l_f	Distance from vehicle's center of gravity to front axle
l_r	Distance from vehicle's center of gravity to rear axle
L_w	Lane width
l_{des}	Desired lane index
m	Vehicle's mass
n	Number of obstacle priorities
N_c	Control horizon
N_p	Prediction horizon

N_{rc}	Repetition control horizon
p^*	Optimal iteration
P_b	Brake pressure
R	Radius of constraint arc
r	Vehicle's yaw rate
R_{eff}	Effective wheel radius
s	Signed distance of vehicle and obstacle
s_c	Collision signed distance
s_i	Signed distance of vehicle and i^{th} non-crossable obstacle
s_j	Signed distance of vehicle and j^{th} crossable obstacle
s_l	Signed distance of vehicle and left lane marker
s_q	Signed distance of vehicle and q^{th} lane marker
s_r	Signed distance of vehicle and right lane marker
s_{X_i}	Signed distance coordinate X-axis
s_{Y_i}	Signed distance coordinate Y-axis
t	Current time step
T_0	Safe time gap
T_m	Motor torque
T_{max}	Maximum propelling torque
U	Potential field

u	Vehicle's longitudinal velocity
u_0	Initial vehicle longitudinal velocity
U_{acc}	Accident potential parameter
U_{C_j}	Potential function of j^{th} crossable obstacle
u_{des}	Desired longitudinal velocity
U_{lma}	Lane marker potential parameter
u_{max}	Maximum allowed vehicle speed
U_{NC_i}	Potential function of i^{th} non-crossable obstacle
u_{o_i}	Longitudinal velocity of i^{th} non-crossable obstacle
U_{R_q}	Potential function of q^{th} lane marker
U_{saf}	Safe potential parameter
U_{unc}	Uncomfortable potential parameter
v	Vehicle's lateral velocity
W	Vehicle's weight
W_0	Obstacle's added width
X	Vehicle's longitudinal position
X_0	Minimum safe longitudinal distance
X_1	Longitudinal position of left side of contact rectangle
X_2	Longitudinal position of right side of contact rectangle
X_{c_i}	Longitudinal collision distance of i^{th} non-crossable obstacle

X_{o0_i}	Initial obstacle longitudinal position
X_{o_i}	Longitudinal position of i^{th} non-crossable obstacle
X_{s_i}	Safe longitudinal distance of i^{th} non-crossable obstacle
X_{s_j}	Safe longitudinal distance of j^{th} crossable obstacle
Y	Vehicle's Lateral Position
Y_0	Minimum safe lateral distance
Y_1	Lateral position of lower side of contact rectangle
Y_2	Lateral position of upper side of contact rectangle
Y_{c_i}	Lateral collision distance of i^{th} non-crossable obstacle
Y_{des}	Desired lateral position
Y_{o_i}	Lateral position of i^{th} non-crossable obstacle
Y_{s_i}	Safe lateral distance of i^{th} non-crossable obstacle
Y_{s_j}	Safe lateral distance of j^{th} crossable obstacle
$\boldsymbol{\tau}$	Vector of binary variables
$\boldsymbol{\varepsilon}$	Vector of slack variables
$\boldsymbol{\xi}$	Vector of optimization variables
$\boldsymbol{\xi}^*$	Optimal solution
$\boldsymbol{A}$	State matrix
$\boldsymbol{A}_d$	Discrete state matrix
$\boldsymbol{B}$	Input matrix

$\boldsymbol{B}_d$	Discrete input matrix
$\boldsymbol{C}$	Output matrix
$\boldsymbol{D}$	Feedforward matrix
$\boldsymbol{u}_c$	Input vector
$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Output vector
$\boldsymbol{y}_{des}$	Desired output vector
$\boldsymbol{y}_{s-max}$	Vector of constraint bound
$\boldsymbol{y}_s$	Vector of constraint variables

Chapter 1

Introduction

1.1 Motivation

A large percentage of car crashes are caused by driver errors. The National Highway Traffic Safety Administration reported the driver as the critical factor of 94% of crashes involving light vehicles from 2005 to 2007 [1]. Of the crashes caused by the driver, 41% are because of recognition errors like inattention, 33% are because of decision errors like driving too fast, 11% are because of performance errors like poor directional control, 7% are because of non-performance errors like sleeping, and 8% are caused by other driver errors. Autonomous vehicles do not become inattentive, do not speed, and do not sleep; they are being developed in hopes of reducing the number of crashes by removing the main cause of crashes, the driver.

Autonomous vehicles can reduce the number of crashes, but they cannot totally eradicate crashes. Autonomous vehicle crashes can be grouped into three types of causes [2],[3]. First, an autonomous vehicle system is imperfect and can occasionally fail. Failures can happen because of hardware failures, software bugs, perceptual errors, or reasoning errors [4]. Second, even if autonomous vehicles are perfect, they will drive amongst human-driven vehicles. Human drivers have unpredictable driving behaviors, and avoiding all of their possible movements is impossible [5]. Third, even on a road with only perfect autonomous vehicles, the vehicles would face wildlife, pedestrians, and bicyclists, all of which also have unpredictable behaviors. Because of the mentioned reasons, there will be unavoidable crash scenarios for autonomous vehicles.

Autonomous vehicles are expected to respond properly in a situation where a crash is imminent. Drivers might panic in such a situation, but autonomous vehicles cannot use this excuse. One example of such a situation is when a deer is on the middle of the road at such a distance that the vehicle cannot stop behind the deer if it brakes, but it can swerve to avoid the deer [6]. In this situation, a driver might decide to brake, which would result in a crash and a possible injury of the passenger. But, he would not be blamed for this decision since it was made in an occasion of panic. However, a brake decision is not acceptable for an autonomous vehicle. The vehicle can be programmed to swerve instead of braking, and not programming it to do so is construed as negligence with legal ramifications [7].

Many factors like the type of the objects around the vehicle, the road structure, and the conditions of the road sides are important in making decisions for a scenario with an imminent crash. In the deer scenario mentioned above, the deer is on the road, but the vehicle has enough space on the road beside the deer to swerve. Therefore, swerving is less costly compared to braking and is a better response in this situation. An alternative scenario is when there is not enough space beside the deer for the vehicle to swerve safely, and there are objects on the road sides that can damage the vehicle. In this scenario, swerving and moving to the road sides only damages the vehicle and most probably does not result in any injury. Therefore, going to the side is less costly and is more reasonable in this situation. Another alternative is a similar scenario when the deer is replaced by a squirrel. In this situation, crossing over the squirrel is more reasonable since it has no harm or damage to the passengers and the vehicle so it is less costly.

There are many scenarios where a crash is imminent, and even unavoidable. In these scenarios, autonomous vehicles are expected to consider priorities of the obstacles and find the maneuver with the minimum cost based on these priorities. The obstacles' priorities should be considered in the motion planning module since it is the module in autonomous vehicles that considers obstacles in its planning. Therefore, it is expected that the motion planning module implements obstacles' priorities in planning vehicle's trajectories.

A motion planning module should consider vehicle dynamics in its planning procedure. If the vehicle's trajectory is planned without considering the vehicle's dynamics and limitations, it might not be feasible for tracking. However, a motion planning module that considers the dynamics and limitations of the vehicle in its planning procedure plans trajectories that can be tracked by the vehicle. Such a motion planning system knows the vehicle's capacity and can utilize this capacity in reducing the crash cost. Moreover, it can perform as both the motion planning module and the trajectory tracking module since it also covers the tasks of the trajectory tracking module.

Many motion planning techniques are developed for autonomous road vehicles; interpolating curve planners, graph-search planners like A*, sample-based planners like Rapidly-exploring Random Tree (RRT), and optimization planners like a Model Predictive Controller (MPC). An MPC handles the future predictions and system constraints in a unified manner to find the optimal solution [8]. Therefore, among the motion planning techniques, an MPC has the advantage of systematically handling vehicle future predictions and constraints of vehicle dynamics, actuators, and obstacles in planning the optimal trajectory.

A nonlinear MPC optimizes a nonlinear objective function with a nonlinear model and nonlinear constraints. This MPC is non-convex and has a high calculation time. A motion planning MPC has a vehicle model for longitudinal and lateral vehicle motions. It also includes constraints on vehicle dynamics, actuators, and obstacles. A vehicle model is generally nonlinear because of its nonlinear equations of motions and extremely nonlinear tire behavior. Obstacle avoidance is also performed either through adding potential functions, which are nonconvex, to the objective function or through constructing an obstacle-free area by constraints, which are generally nonlinear. Therefore, a motion planning MPC is generally nonlinear.

A nonlinear MPC has a high calculation time and is difficult to solve in real-time. In order to be appropriate for real-time applications, the general nonlinear motion planning MPC can be approximated by a quadratic motion planning MPC. A quadratic MPC optimizes a quadratic convex objective function with a linear model and linear constraints. Therefore, for a quadratic motion planning MPC, the vehicle behavior should be modeled by a linear vehicle model and linear constraints. It is important that the model and the constraints model the vehicle behavior appropriately over a wide range of operating conditions, e.g. high speed and tire force limit. Moreover, the nonconvex obstacle-free area should be approximated by linear constraints. Some parts of the obstacle-free area are removed in the process of approximation, which causes the MPC to lose some feasible trajectories. It is important to develop approximation methods that do not cause losing the optimal trajectories.

1.2 Objectives

The first objective of this project is to plan the vehicle's trajectory according to the obstacles' priorities, i.e. in an imminent crash situation, the vehicle avoids the obstacles with the highest priorities. A quadratic MPC will be used for motion planning, and two obstacle avoidance methods will be utilized to apply obstacles' priorities in the MPC problem.

The first method is the potential field method, which prioritizes obstacles based on the need to avoid them. In this method, obstacle potential functions are included in the MPC cost function for obstacle avoidance. Obstacles will be categorized as crossable and non-crossable, and a potential function will be assigned to each category corresponding to its avoidance necessity to prioritize the obstacles.

The second method is to utilize lexicographic optimization to implement priorities on the non-crossable obstacles. Constraints will be included in the MPC problem for the non-crossable obstacles, and the obstacles will be prioritized by prioritizing their corresponding constraints. Lexicographic optimization will be used to prioritize the constraints in the MPC problem.

The second objective is to reduce the number of useful trajectories removed by convexification of the obstacle-free area. An obstacle-free area is generally nonconvex and cannot be constructed in a quadratic MPC problem. For a quadratic motion planning MPC, it is usually convexified by one set of linear constraints. This process of convexification removes some of the feasible trajectories. In this project, an iterative MPC will be introduced that solves the motion planning problem for three sets of obstacle constraints to reduce the number of trajectories missed due to the convexification.

The third objective is to develop a vehicle model and vehicle constraints for the MPC such that the model is valid over a wide range of operating conditions. Since the MPC is quadratic, the model and constraints should be linear. A linear bicycle model will be used as the vehicle model. Linear tire models can model lateral tire forces if they accompany constraints on tire sideslip angles. Moreover, in large tire forces, combined tire slip and load transfer have great effects. In this project, tire constraints will be developed that consider the combined tire slip and longitudinal load transfer and cover the tire sideslip angle constraints. With these constraints, the vehicle model considers its capacity and remains valid at tire force limits.

Moreover, the developed motion planning MPC should be implementable in real time.

1.3 Thesis Outline

In the second chapter, the background of motion planning MPCs is presented. The architecture of autonomous vehicles is explained, and different motion planning techniques are discussed. The literature of motion planning MPC with an emphasis on the vehicle model and the obstacle model is reviewed, and the background on lexicographic optimization method is expressed.

In the third chapter, a potential-field-based motion planning MPC is developed. A vehicle bicycle model with linear tire models is presented, and tire constraints are introduced to keep the model valid by keeping the tire in its linear force region. The tire constraints also consider the combined tire slip and the longitudinal force transfer so that the model considers its limitations and remains valid at tire force limits. The potential field including the potential functions of the obstacles and lanes are expressed. The obstacles are prioritized as crossable and non-crossable by assigning them potential functions corresponding to their characteristics. Next, the MPC problem is introduced with the vehicle model as its model, vehicle constraints as its constraints, and the potential field included in its objective function. Then, the simulation results of some test scenarios are presented to evaluate the performance of the motion planning MPC in autonomous driving and prioritizing obstacles.

In the fourth chapter, the MPC introduced in the third chapter is developed to prioritize non-crossable obstacles through lexicographic optimization. Instead of using one linear obstacle constraint set, three linear obstacle constraint sets are introduced to reduce the number of removed trajectories. The constraint sets are included in the MPC problem to generate Iterative Quadratic MPC (IQMPC); IQMPC includes three MPCs, one MPC for each constraint set. Lexicographic optimization is applied on IQMPC to prioritize the non-crossable obstacles. Then, the motion planning method is simulated for some test scenarios to evaluate its performance in obstacle avoidance and prioritizing obstacles.

In the fifth chapter, the developed motion planning MPC from the third chapter is implemented on an autonomous test vehicle platform. The MPC is modified to become compatible with the test vehicle software platform. It is also modified to compensate for the delays of the test platform by using predicted vehicle states as its initial states. Then, the experimental results for some test scenarios are presented to show that the MPC is implementable in real time and to validate the simulation results of the MPC.

In the sixth chapter, conclusions of the thesis are presented, and suggestions are expressed for continuation of this project.

Chapter 2

Background and Literature Review

2.1 Autonomous Vehicles

More than 90% of car crashes are caused by driver errors [1]. Autonomous vehicles are being developed in hopes of reducing the number of crashes by removing the main cause of crashes, the driver. However, there are situations when a crash is imminent, or even unavoidable, even for an autonomous vehicle. In such a situation, an autonomous vehicle is expected to respond properly. Crashing into different obstacles have different costs based on the injury and damage they may cause. In an imminent crash situation, an autonomous vehicle is expected to consider these costs and plan a maneuver that avoids the obstacles with the highest priorities. The main focus of this thesis is to develop a platform for autonomous vehicles that can consider the priority of the obstacles and find the maneuver with the minimum cost based on the obstacles' priorities. The architecture of autonomous vehicles is studied in this section to determine the vehicles' modules in which the priorities can be applied.

2.1.1 Architecture of Autonomous Vehicles

Autonomous vehicle systems have different architectures. Figure 2.1 demonstrates a general architecture for them presented in [9]. Although there is no clear line between the modules of an autonomous vehicle, this general architecture can demonstrate the essence of an autonomous vehicle. A similar architecture is presented in [10] for the decision stage of an autonomous vehicle, which is the combination of the decision and control stages of Fig. 2.1.

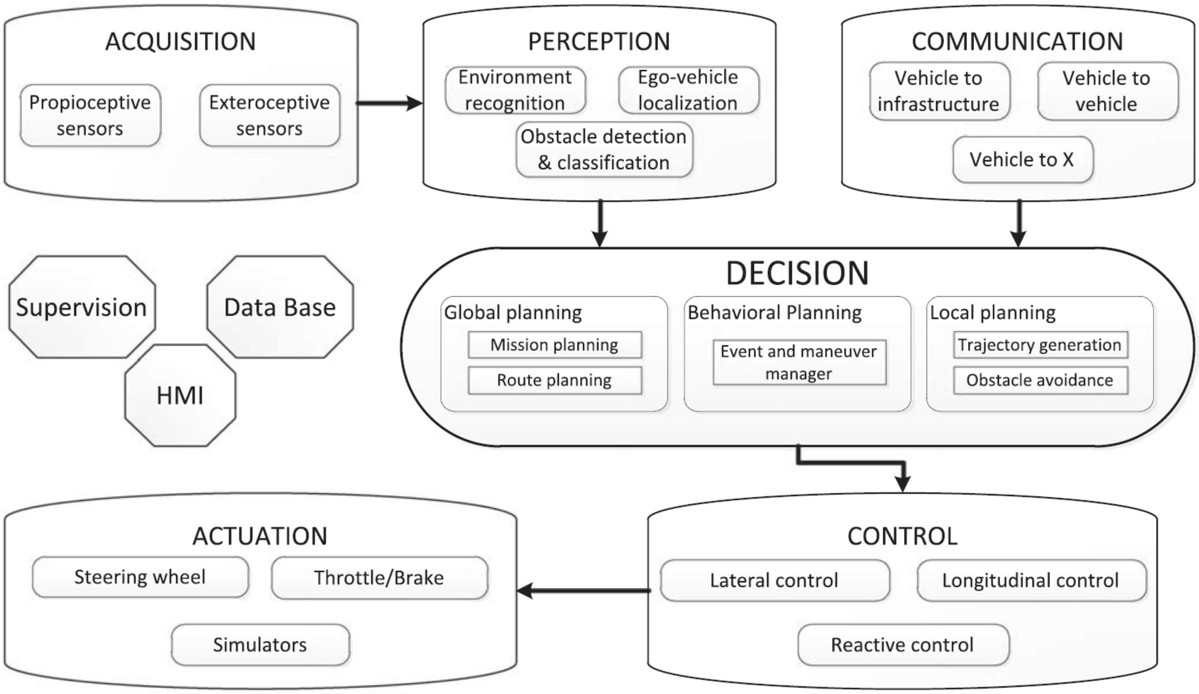


Figure 2.1: A general architecture for autonomous vehicles [9].

In general, the vehicle receives data from sensors such as GPS/INS sets, LIDARs, vision cameras, and radars. It may also receive data from infrastructures and others vehicles via communication sets. The perception stage provides knowledge about the road structure [11],[12], road regulations [13],[14], obstacles [15],[16], and vehicle states [17],[18] based on the data. The decision and control stages decide the vehicle's path based on the information generated in the perception stage and generate the actuation commands.

The decision stage consists of global planning, behavioral planning, and local planning modules. The global planning module finds an efficient route from the current position to the destination [19],[20]. The behavioral planning module determines an appropriate driving behavior, like stopping at a position or changing the lane, to follow the route [21],[22]. The local planning module or motion planning module plans the vehicle's trajectory such that the vehicle performs the planned behavior while avoiding the obstacles. Then, the control module or trajectory tracking module generates the vehicle's actuator commands to follow the planned trajectory while maintaining the vehicle's stability.

The obstacle avoidance task is performed in the motion planning module. Therefore, this module is expected to plan a trajectory that minimizes the cost of an imminent crash. In this thesis, a motion planning module is developed that is capable of planning trajectories based on the obstacles' priorities.

Trajectory tracking modules usually consider vehicle dynamics to keep the vehicle stable and improves the ability of the vehicle in tracking the trajectory [23],[24]. However, it is possible that the planned trajectory is not feasible for tracking by the vehicle since the vehicle dynamics and its limitations are not considered in trajectory generation procedure [25]. Therefore, a motion planning technique should consider vehicle dynamics and limitations so that the trajectory that it generates can be tracked by the vehicle. Furthermore, if the vehicle dynamics and limitations are considered in the motion planning technique appropriately, the motion planning technique can utilize the vehicle's entire capacity to reduce the crash cost.

A motion planning module and a trajectory tracking module can be combined if the motion planning module covers the tasks of the trajectory tracking module. A motion planning module that considers the vehicle's dynamics and limitations in the process of generating the trajectory and generates the actuator inputs as its outputs can be used in place of the two modules [26]. This way, the tasks of both modules can be performed in an integrated manner. Such a motion planning module is developed in this thesis.

2.2 Motion Planning Techniques

The motion planning module should be capable of planning an appropriate trajectory considering the road's structure and regulation, obstacles' configurations and dynamics, and the vehicle's current states. As mentioned, the module should consider the vehicle dynamics and limitations appropriately to find the best feasible trajectory in the situation of an imminent crash. Moreover, if the module also generates the vehicle inputs, it can be used as both the motion planning module and the trajectory tracking module in a unified manner.

Figure 2.2 presented in [9] illustrates the motion planning techniques used in the autonomous vehicle demonstrations over time. The main motion planning techniques developed for autonomous road vehicles can be categorized as interpolating curve planners, graph-search planners, sample-based planners, and optimization planners.

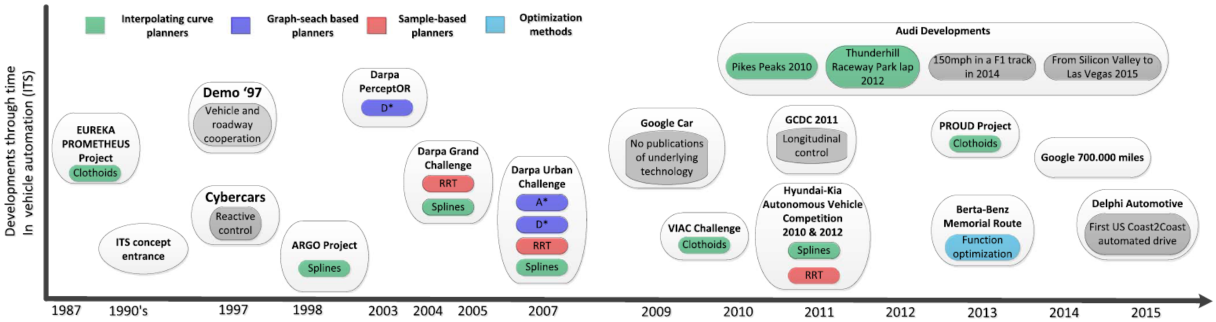


Figure 2.2: Autonomous vehicle demonstrations over time and their associated motion planning techniques [9].

Interpolating curve planners take a set of path waypoints and generate a local smooth trajectory while considering vehicle constraints and dynamic environment. Different techniques are utilized to generate the trajectories: lines and circles [27], clothoid curves [28], polynomial curves [29], Bezier curves [30], and spline curves [31]. They can consider curvature constraints in the motion planning process, but they do not consider the vehicle dynamics.

Graph-search planners generate a grid map and find a path from the initial state to the final state through the grids. The basic algorithm of this category is the Dijkstra algorithm, which is a simple node search to find the shortest path from initial position to the final position [32]. An A* algorithm [33] and similar algorithms like D* [34] implement heuristics to make the node search fast. The state lattice algorithm generates a grid of states instead of positions and uses the other graph search algorithms to find the best path from the initial state to the final state [35]. Since the calculation time of the algorithm increases by increasing the dimension of the problem, the method cannot be solved in real-time if the vehicle dynamics states are considered.

Sample-based algorithms sample the search state space and search for connections between the samples. Rapidly-exploring Random Tree (RRT) [36] and similar algorithms like RRT* [37] are the main algorithms in this category. They can quickly find a path between the initial and final states while considering vehicle constraints . These methods quickly find a path between the initial and final states but finding the optimal path can take longer.

Optimization planners optimize an objective function of the vehicle states with state and input constraints. The main optimization motion planning technique is the Model Predictive Controller (MPC) technique. This technique has the vehicle dynamics as its model and predicts the vehicle behavior in the process of optimizing the objective function. It can optimize the vehicle performance, constrain the vehicle states and inputs, and find the optimal obstacle-free trajectory for the vehicle in a unified manner. The MPC problem can also be solved in a short time.

The potential field technique is another motion planning technique which is not usually used for autonomous road vehicles but will be used in this thesis. It generates a potential field based on the Potential Functions (PFs) of obstacles, road structures, and goal. It plans the path by moving in the descent direction of the field. A field is generated by a combination of different PFs. Wang et al. [38] use exponential PFs for lanes, hyperbolic PFs for static obstacles, and products of exponential and hyperbolic functions as PFs for moving obstacles. Wolf et al. [39] use quadratic PFs for lanes, hyperbolic PFs for road boundaries, and exponential PFs for cars. Ji et al. [40] use cosine PFs for lanes and exponential PFs for cars. The idea of assigning different PFs to different types of obstacles is used in this thesis to prioritize obstacles.

The MPC technique is used as the motion planning technique in this thesis since it is the technique that can systematically consider a vehicle's future behavior along with the vehicle, actuator, and obstacle constraints. Moreover, it can perform motion planning in a timely manner. Furthermore, it generates actuator inputs, and the motion planning module using this technique can act as both the motion planning module and the trajectory tracking module in a unified manner. Potential fields are also implemented in the motion planning MPC as a method of obstacle avoidance, and their capabilities in prioritizing obstacles is used to apply obstacles' priorities in the motion planning MPC.

Obstacles are not prioritized in the literature of motion planning of autonomous vehicles. However, there are papers in the literature of robotics that prioritize obstacles for motion planning [41],[42]. Although these papers prioritize obstacles, they are developed to avoid all obstacles. However, it is necessary to develop motion plannings that prioritize obstacles based on their crash costs for situations that avoiding all obstacles is not possible. Moreover, there are obstacles that avoiding them is not necessary, and can be crossed if required.

2.3 MPC Motion Planning

The literature of the motion planning MPC is investigated in this section. The nature of motion planning MPC of an autonomous ground vehicle is non-convex. A vehicle model for the vehicle longitudinal and lateral motions are used as MPC's model, which is generally nonlinear because of its nonlinear dynamics equations and nonlinear tire behavior. The obstacle avoidance is also performed either by including potential functions, which are non-convex, in the cost function or adding obstacle constraints, which are non-convex in essence.

Some researchers consider the nonlinear nature of the problem and solve nonlinear MPCs. However, the high calculation cost of a nonlinear MPC makes it difficult for real-time solution. Therefore, several researchers have worked on simplifying the nonlinear MPC problem to a quadratic MPC problem, which can be solved in a short period of time and is appropriate for real-time applications. A quadratic MPC has a quadratic cost function, a linear model, and linear constraints. The researchers work on finding a linear vehicle model with linear constraints that predicts the vehicle's behavior as close to the nonlinear vehicle model as possible. They also study a set of linear obstacle constraints that construct a convex area including the most useful section of the non-convex obstacle-free area. In this thesis, a quadratic MPC motion planning problem is developed for real-time implementation. In the following sections the vehicle models and obstacle avoidance methods used in the MPC motion planning literature are reviewed.

2.3.1 Vehicle Models

The vehicle models used for the motion planning of autonomous vehicles can be categorized as point mass vehicle models, kinematics vehicle models, and dynamics vehicle models. Point mass models are linear models modeling the vehicle as a particle with the vehicle mass capable of moving in longitudinal and lateral directions. A point mass model does not consider a tire model and vehicle geometry, and subsequently, it can cause large tracking errors. Therefore, some state constraints can be added to make the generated trajectory more feasible. Longitudinal and lateral accelerations can be constrained by the acceleration corresponding to the maximum tire force capacity [43],[44],[45]. Moreover, the vehicle sideslip angle cannot be large for a vehicle in a non-drifting maneuver, and can be constrained [46],[45]. A point mass model cannot predict the vehicle's behavior adequately even with these constraints.

Kinematics models are nonlinear models modeling the vehicle based on its geometry. It does not consider tire models. However, to consider the passenger comfort and to avoid skidding, constraints can be applied on the lateral acceleration to limit it to normal driving values [47],[48],[49]. Vehicle dynamics models consider tire model in their model. Carvalho et al. [48] and Zhang et al. [50] compare open loop behaviors of a vehicle kinematics model and a vehicle dynamics model. The results show that, in low velocities, both models have almost the same performance in modeling the vehicle behavior. At higher velocities, specially higher than $15m/s$, the dynamics model performs noticeably better, when the maneuver includes steering angles larger than 1.5° [50]. Therefore, if an autonomous vehicle is supposed to perform high speed maneuvers with large lateral accelerations, a vehicle dynamics model is preferred to a vehicle kinematics model for use as the MPC model.

Vehicle dynamics models are obtained based on Newton's second law by considering tire models as the maneuvering forces. The vehicle dynamics equations are nonlinear regardless of the tire model, but the main source of nonlinearity in vehicle behavior is tire behavior.

Tires have limited capacity and become saturated. Nonlinear tire models like Pacejka and Brush tire model are used to model the nonlinearity in the tire model. Linear tire models model the tire behavior in the linear region. A linear tire model usually accompanies a constraint on the sideslip angle to limit the model to the linear region of the tire force where it is valid.

Frasch et al. [51] present a four wheel vehicle dynamics model with equations of longitudinal, lateral and yaw motions at the vehicle's center of gravity based on the four tire forces. They use a Pacejka tire model and considers the wheels' dynamics in the model. Moreover, they consider load transfer resulting from the longitudinal and lateral accelerations in the tire model to generate a more accurate vehicle model. The wheel dynamics increases the number of vehicle states by four but has a very small effect on the accuracy of the vehicle model [52]. Gao et al. [43] use a four wheel vehicle dynamics model similar to [51] without wheel dynamics. They also do not consider load transfer effects in the model.

A four wheel vehicle dynamics model with no wheel dynamics can be simplified by a bicycle model. For a bicycle model, the tires of each axle are modeled as one resulting tire. A bicycle model is nonlinear in essence. Yoon et al. [53] consider a bicycle model with a Pacejka tire model. They constrain the front and rear tire sideslip angles since large tire sideslip angles are not favorable. Park et al. [54] and Zhang et al. [50] consider a linear tire model for the vehicle while the vehicle's equations of motions are nonlinear. They also constrain the tire sideslip angles to keep the tire in its linear force region and keep the vehicle model valid.

Linear Bicycle Model

The previously mentioned vehicle dynamics models are nonlinear, and the MPC using these models should be nonlinear. However, a linear bicycle model can be used in a quadratic MPC to handle high speed maneuvers with large lateral accelerations since it considers vehicle dynamics [55],[56],[57],[58]. Yi et al. [59] develop a nonlinear bicycle model with a Pacejka tire model and with longitudinal load transfer. They then linearize the model around the operating point. They also constrain the total vehicle acceleration to stay in the friction circle. The circle is approximated by half-spaces so that the quadratic constraint is approximated by linear constraints to be used in a quadratic MPC.

Turri et al. [52] develop a four wheel vehicle model for vehicle lateral motion with a Pacejka tire model where the longitudinal motion of the vehicle is known. They also consider load transfer in the model. They then linearize the vehicle model. They calculate the rear and front tire models as functions of the total vehicle longitudinal force and linearize them. This way, they consider the load transfer as well as the combined slip effects.

Gao et al. [26] present a nonlinear bicycle model with a Pacejka tire model to model the lateral motion of the vehicle, and then linearize the model. They use a linear tire model for the rear tire force and constrains the tire sideslip angle. They calculate the tire cornering stiffness and the maximum tire sideslip angle such that it has the best approximation of the tire behavior and can generate a lateral force close to the maximum lateral force. For the front tire, they use the tire force in the motion equations and derive the steering angle by the inverse Pacejka model. They also constrain the tire sideslip angles to keep the tires in their linear force regions.

Erlien et al. [60] present a nonlinear bicycle model with a brush tire model for the lateral vehicle motion of a race vehicle. Similar to [26], they use a tire inverse model for the front tire. They assume the nominal path curvature to be the road curvature. Therefore, instead of using a linear tire model for the rear tire, they linearize the brush model around the nominal sideslip angle corresponding to the nominal path curvature. This work is for race vehicles working in large path curvatures at high speeds, which need large tire sideslip angles to track the path. For road vehicles, small tire sideslip angles are required to track the path, and the resultant tire model would be similar to a linear tire model. The paper also constrains the front lateral force and applies a stability envelope for the rear tire instead of constraining the tire sideslip angles. The envelope limits the yaw rate to its maximum steady state value corresponding to the road tire-road friction and the rear sideslip angles to the limits corresponding to the tire linear force region.

Funke et al. [61] present a nonlinear bicycle model with a brush tire model for the lateral vehicle motion of a race vehicle. For both the front and rear tires, they linearize the brush tire model around the nominal sideslip angles corresponding to the nominal path curvature. They use a stability envelope similar to that of [60]. For the front lateral tire force limitation, they consider the combined slip effect. They assume that the longitudinal force commanded by driver remains constant, and they limit the lateral force to the remaining tire capacity.

As mentioned, for a road vehicle, the linearized tire model in [60] and [61] are similar to a linear tire model. Moreover, the tire sideslip angle constraints keep the tires in its linear force range, so they keep the linear tire model valid. Several works use a vehicle bicycle model with linear tire models, and constrain the rear and front sideslip angles [62],[63],[48],[64]. In this thesis, a vehicle bicycle model with linear tire models is used to model the longitudinal and lateral motions of the vehicle for the MPC motion planning. The cornering stiffness values and maximum sideslip angles are calculated similar to [26]. Instead of constraining the front and rear sideslip angles, the lateral forces are constrained in combined tire slip constraints such that the generated constraints cover the sideslip angle constraints.

The combined slip is important in large tire forces, e.g, the tire has no lateral force capacity if the the whole tire capacity is used for the longitudinal force. Load transfer is also significant in large accelerations, e.g, harsh braking. In this thesis, the combined slip and load transfer effects are considered as constraints. The longitudinal load transfer is applied in the calculation of the normal tire forces similar to [59]. Funke et al. [61] and Turri et al. [52] only model the lateral vehicle motion and apply the combined slip as a constraint on the lateral force. In this thesis, the longitudinal motion is also modeled, and the combined tire slip should be applied as a constraint on the longitudinal force and the lateral force. The constraint is an ellipse, if load transfer is not considered. In this thesis, load transfer equations are applied on the ellipse constraint equation to derive the tire constraint. The resultant constraint is nonlinear and cannot be used in a quadratic MPC. However, the constraint constructs a close convex space, which can be approximated by half-spaces similar to [59].

2.3.2 Obstacle Model

Obstacle avoidance is the main task of a motion planning system. One way to perform this task is to generate a repulsive force that keeps the vehicle away from the obstacle. This method is performed by adding a repulsive PF to the optimization cost function. Abbas et al. [65] and Gao et al. [43] include hyperbolic potential functions of the distance from the obstacle, and Park et al. [54] and Yoon et al. [53] include parallax PFs in the MPC cost function. The resulting cost functions are nonlinear and nonconvex, and they require the solution of nonlinear optimization problems.

Another way to perform the obstacle avoidance task is constraining the vehicle to remain in the obstacle-free area. The essence of an obstacle-free area is nonconvex, and the area can be generated by nonconvex constraints. Liu et al. [66] generate a safe area in the LIDAR detection area. The safe area is the semicircle detection area cut by obstacles. Gotte et al. [67] constrain the vehicle out of the circle around each obstacle. Gao et al. [26] constrain the vehicle out of the ellipse around each ellipsoidal obstacle. Liao et al. [64] consider the obstacles as rectangles and use mixed integer constraints to keep the vehicle in the obstacle-free area. Frasch et al. [51] also consider obstacles as rectangles but uses nonlinear constraints to generate the obstacle-free area. Qian et al. [47] generate a quadratic nonconvex constraint for an obstacle on the side of the lane. The constraint keeps the vehicle out of the portion of the lane containing the obstacle. For an obstacle on the middle of the lane, they keep the vehicle behind the obstacle with a convex linear constraint.

MPC problems with nonconvex constraints are nonlinear and have high calculation costs. Several works investigate convex alternatives for the problem. Some researches only control the lateral motion of the vehicle for obstacle avoidance and assume to know the longitudinal motion prior to obstacle avoidance. They grid the obstacle-free space for prediction time steps based on the longitudinal motion. For each prediction time step, they constrain the vehicle's lateral position to an available convex lateral space at the corresponding grid [55],[62],[63],[61],[68]. The method is useful for situations where only the lateral motion is planned by the motion planning module, e.g, a driving assistance system where the driver controls the longitudinal motion. However, it does not plan the longitudinal and lateral motions simultaneously, and cannot plan maneuvers like stopping behind an obstacle. Erlien et al. [60] solve a similar MPC problem several times for different brake values. This way, they can plan the longitudinal motion at the same time as the lateral motion. However, this method can consider only specific brake values and also increases the calculation time.

Some papers generate a convex safe envelope based on the driving mode, and plan the longitudinal and lateral motions to keep the vehicle in the safe envelope [69],[70]. Schildbach et al. [49] plan a time for lane change and performs obstacle avoidance based on this time. They keep the vehicle in the rectangular safe envelope of the current lane for time steps less than the planned time and in the rectangular safe envelope of the current lane and the intended lane for time steps larger than the planned time. These methods consider a predefined envelope structure for each driving mode. They keep the structure of the envelope, and therefore, lose a large portion of the obstacle-free area.

Some other papers consider a linear constraint for each obstacle. Nilsson et al. [46] generate a linear constraint with a constant slope for each obstacle ahead of the vehicle. The constraint line is determined such that the constraint keeps longitudinal and lateral safety distances from the obstacle. A similar constraint is also generated for an obstacle behind the vehicle. However, for these constraints, if the longitudinal safe distance is small, the constraint causes difficulties for a vehicle passing an obstacle on its side, and if it is large, a section of the space behind the obstacle cannot be used for a maneuver of stopping behind the obstacle. Jalalmaab [45] use similar constraints but considers a horizontal constraint for the situation that the obstacle is on a different lane from the vehicle, and it is close to the vehicle in the longitudinal direction. This variation cannot solve the mentioned problem for the obstacles on the same line as the vehicle.

Carvalho et al. [58] generate a constraint based on the signed distance of the vehicle and the obstacle. The signed distance of two objects is their minimum distance if they are not in contact. The constraint constrains the signed distance to be non-negative, which is non-convex. The paper linearizes the constraint around the predicted states. This method generates a linear constraint with slopes based on the relative position of the vehicle and the obstacle, and solves the problems existing for the constraints presented by [46]. However, for an obstacle in front of the vehicle, this method generates a constraint limiting the vehicle to stop behind the obstacle and does not allow a swerving maneuver. An obstacle is defined to be in front of the vehicle if it is ahead of the vehicle with no lateral distance from the vehicle.

In this thesis, different PFs are included in the MPC cost function for different kinds of obstacles to put priority on obstacles based on the their avoidance necessity. The obstacles are categorized as crossable and non-crossable obstacles; a crossable obstacle, like a bump, can be crossed but avoidance is preferred, and a non-crossable obstacle, like a car, should be avoided. A convex quadratic approximation is performed on the PFs so that they can be used in a quadratic MPC.

Obstacle constraints are applied on the non-crossable obstacles. As mentioned, the linearized signed distance obstacle constraints do not allow the vehicle to swerve when there is an obstacle in front of it. However, there are situations such that stopping behind an obstacle is not feasible but avoiding the obstacle by swerving is possible. Therefore, the vehicle misses its optimal solution if it uses the constraint set generated by this method of convexification. In this thesis, in addition to this constraint set, two constraint sets are generated, which include swerving maneuvers. IQMPC is introduced to solve the motion planning MPC for the union of the three constraint sets to fix the previously mentioned problem. The non-crossable obstacles are also prioritized in IQMPC using lexicographic optimization explained in the following section.

2.4 Lexicographic Optimization

Lexicographic Optimization (LO) is a method to prioritize objective functions of an optimization problem. Generally, an optimization problem with multiple objective functions does not have a solution that minimizes all the objective functions. A weighted sum of the objective functions can be solved to find a pareto-optimal solution of the problem [71]. If an objective function has priority over another objective function, this method is not appropriate, since it does not necessarily minimize the objective function with the higher priority order. Using LO, it is possible to consider priorities on the objective functions [72]. It finds the optimal solution set of an objective function in the optimal solution set of the objective function with the higher priority order. The optimal solution of the objective function with the lowest priority order is the optimal solution of the problem.

In an MPC problem, where constraints can cause infeasibility, slack variables are added to the constraints to avoid infeasibility. Terms containing slack variables are also added to the objective function to penalize constraint violations. Priorities can exist on the constraints, i.e. violating some constraints can be less favorable than violating other constraints. LO can include the priority orders of the constraints in the MPC problem by prioritizing the penalizing terms of the constraint violations [73]. In this thesis, LO is applied on the motion planning MPC problem to prioritize non-crossable obstacles by prioritizing the obstacle constraints based on the priority orders of their corresponding obstacles.

2.5 Summary

In this chapter, the background on a motion planning MPC was presented. The main focus of this thesis is to develop a platform for autonomous vehicles that plans the vehicles' trajectories based on the obstacles' priorities for near-crash situations. Such a platform has not been developed in the literature. In an autonomous vehicle, the motion planning module is the module that plans the vehicles' trajectory and performs the obstacle avoidance task. Therefore, in this thesis, the platform is developed by developing the motion planning module; a motion planning module is developed that plans vehicle's trajectories based on obstacles' priorities.

As explained in Section 2.1, a motion planning module should consider vehicle dynamics in its planning so that it can utilize all the vehicle capacity to plan the best feasible trajectory. If the motion planning module also generates the actuator commands, it covers the tasks of the trajectory tracking module. Such a motion planning module can perform the tasks of both the motion planning and trajectory tracking modules in a unified manner. In this thesis, such a motion planning method is used.

Section 2.2 discussed different motion planning techniques presented for autonomous road vehicles. As explained, among the techniques, the MPC technique can systematically consider the vehicle model and the vehicle's and obstacles' constraints. It can also be quickly solved. Therefore, this technique is used for motion planning in this thesis.

Section 2.3 presented the background on MPC motion planning for autonomous road vehicles. As explained, a motion planning MPC is generally a nonconvex problem since the vehicle model is nonlinear and the obstacle avoidance PFs and constraints are nonconvex. A nonconvex MPC problem has a high calculation time. If the vehicle model is linear, the PFs are quadratic, and the obstacle's and vehicle's constraints are linear, the MPC becomes a quadratic MPC, which can be solved in real-time. In this thesis, a quadratic MPC is developed for motion planning.

Section 2.3.1 explained the vehicle models used in motion planning MPCs. As discussed, unlike a vehicle point-mass model and a vehicle kinematics models, a vehicle dynamics model can predict the vehicle's longitudinal and lateral behavior appropriately in all ranges of speed and lateral acceleration. It is because a vehicle dynamics model includes the tire model in the vehicle model. For a quadratic MPC, a linear vehicle model should be used. A linear bicycle model is linear and is used as a vehicle dynamics model. As explained, the model can use a linear tire model accompanying constraints on tire sideslip angles. Moreover, for large tire forces, the effects of load transfer and combined tire slip on vehicle dynamics are significant. However, in the literature, there is no linear bicycle model modeling longitudinal and lateral motions for a motion planning MPC that considers combined tire slip. In this thesis, combined tire slip constraints are developed for the vehicle model, which is a linear bicycle model modeling the longitudinal and lateral motions. The tire constraints also cover the constraints on tire side slip angles. Longitudinal load transfer effect is also applied on the constraints.

Section 2.3.2 discussed that obstacle avoidance in motion planning MPCs is performed by PFs or obstacle constraints. As mentioned, a repulsive PF can be used for an obstacle to keep the vehicle away from it. In this thesis, different types of functions are used for obstacles to prioritize non-crossable obstacles over crossable obstacles; hyperbolic PFs are assigned to non-crossable obstacles to avoid crossing them, and exponential PFs are assigned to crossable obstacles to allow crossing them.

As discussed in Section 2.3.2, the obstacles' constraints can also be utilized for obstacle avoidance to generate the obstacle-free area. An obstacle-free area is nonconvex in general. Several methods were discussed that approximate the obstacle-free area by linear constraints to be used for a quadratic MPC. To plan the longitudinal and lateral motions simultaneously, a safe envelope can be developed or a linear constraint can be used for each obstacle. It is important that the constructed area covers the useful parts of the obstacle-free area. The safe envelopes and the linear constraints with constant slopes fail to do so since their predefined structure limit their covering. An appropriate presented method is the linearized signed distance method since its slope varies based on the obstacle's relative position. However, this constraint limits the vehicle to stay behind an obstacle in front of the vehicle, and does not allow swerving. There are situations where stopping behind the obstacle is not possible, but obstacle avoidance through swerving is possible. In these situations, the constraint set developed by the signed distance method fails to cover the optimal trajectory. In this thesis, in addition to this constraint set, two constraint sets are generated for swerving. IQMPC is introduced to solve the motion planning MPC for the union of the three constraint sets to fix the mentioned problem.

As discussed in Section 2.4, LO can be used to prioritize the objective terms of an optimization problem. It can apply priority on the constraints of an MPC problem. In this thesis, constraints are applied on a non-crossable obstacle. The obstacles are prioritized by prioritizing their corresponding constraints through LO. Using this method, in a situation where a crash is unavoidable, the vehicle avoids the obstacles with the highest priorities.

Chapter 3

Potential-Field-based Motion Planning MPC

3.1 Introduction

In this chapter, a motion planning MPC is developed that utilizes the potential field obstacle avoidance method to prioritize obstacles. It prioritizes obstacles by assigning different potential functions to them corresponding to their avoidance necessity. This chapter is mainly developed based on the work of Rasekhipour et al. [74].

The motion planning MPC introduced in this chapter is quadratic. Therefore, its objective function should be quadratic, and its model and its constraints should be linear. A linear bicycle vehicle model is used to model the vehicle behavior in the MPC. The model uses linear tire models, and tire constraints are included in the MPC to keep the model valid at different operating conditions. The tires have limited capacities, which are considered in the constraints by including tire combined slip limitations and the effect of longitudinal load transfer on the limitations. With these constraints, the MPC can generate feasible trajectories on tire force limits since the vehicle model considers its limitations through the constraints. The constraints also keep the tire in its linear lateral force region to keep the linear tire models, and subsequently, the vehicle model, valid.

In this chapter, first, the correlations of the motion planning module with other modules of an autonomous vehicle is discussed. Next, the equations of motion of a bicycle model are presented. The equations are linearized and discretized for use in the MPC problem. The tire constraints are also presented and linearized for use in the MPC problem. Next, potential functions are presented for the road structure and different kinds of obstacles to generate the potential field. Then, the MPC problem is constructed based on the vehicle model, constraints, and potential field. The motion planning MPC is evaluated with a high fidelity CarSim simulation under complicated scenarios, and the results are presented and discussed.

3.2 Autonomous Vehicle Architecture

The correlations of the motion planning module with other modules of an autonomous vehicle are discussed in this section. Figure 3.1 illustrates the correlations through the architecture of an autonomous vehicle. The motion planning module plans the vehicle's trajectory so that it avoids obstacles, complies with road regulations, follows the desired commands, and provides the passengers with a smooth ride. It is assumed that the module receives information of the obstacles, road, vehicle, and desired commands from the other modules.

The obstacle information includes the position, velocity, size, and category of each obstacle. The road information consists of the road profile, the number of lanes, and the widths of the lanes. The vehicle information includes the vehicle's position, heading angle, longitudinal velocity, lateral velocity, yaw rate, and normal tire forces. The obstacle, road, and vehicle information are provided for the motion planning module from the perception module [75],[76] and the estimation module [77],[78]. Moreover, the desired commands including the desired lane and speed are generated in the behavioral planning module.

It is assumed that each obstacle moves with the same longitudinal and lateral velocities as its current velocities for predicting its position in MPC. The risk due to uncertain behaviors of the obstacle as well as errors in the estimation of the vehicle's and the obstacle's states is considered in generating the safe distance of the potential functions. Moreover,

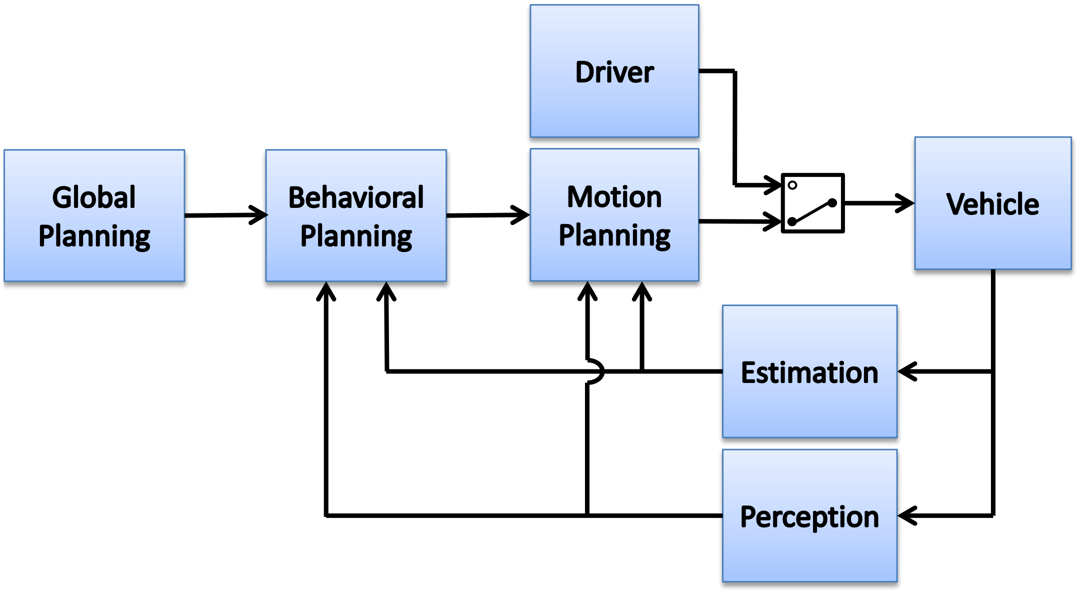


Figure 3.1: Architecture of an autonomous vehicle.

the running step time of the MPC is $50ms$, and any change in the behavior of the obstacle is considered in the planned motion of the vehicle rapidly.

The motion planning module calculates the driving commands including the front steering angle and the total longitudinal force commands. These choices of commands correspond to the driver commands, which include steering wheel angle and the gas/brake pedal positions, so for a semi-autonomous vehicle, switching between the autonomous system and the driver can be performed simply.

The vehicle model used for the predictions of MPC is a linear model, which models only longitudinal, lateral, and yaw motions of the vehicle, and does not consider roll, pitch and bounce motions of the vehicle. However, roll, pitch, and bounce motions of the vehicle do not correspond to the motion of the vehicle on the road, and only correspond to the vertical tire forces, which are assumed to be available for the motion planning module from the estimation module. Furthermore, the vehicle parameters of the vehicle are assumed to be constant. However, if their values change, and the estimation module estimates the parameters, the vehicle model can be updated by the estimated parameters easily. The perception, estimation, and robustness analysis are out of scope of this thesis.

3.3 Vehicle Dynamics Model

A bicycle model is used to model the vehicle dynamics. The notation used in the vehicle model is shown in Fig. 3.2. The equations of motion of the bicycle model are [79]:

$$\dot{X} = u\cos(\theta) - v\sin(\theta), \tag{3.1a}$$

$$\dot{Y} = v\cos(\theta) + u\sin(\theta), \tag{3.1b}$$

$$\dot{\theta} = r, \tag{3.1c}$$

$$m(\dot{u} - vr) = F_{x_T}, \tag{3.1d}$$

$$m(\dot{v} + ur) = F_{y_f} + F_{y_r}, \tag{3.1e}$$

$$I_z\dot{r} = l_f F_{y_f} + l_r F_{y_r}, \tag{3.1f}$$

in which X, Y, and θ are the vehicle's longitudinal position, lateral position, and heading angle in the global coordinate, u, v, and r are the vehicle's longitudinal velocity, lateral velocity, and yaw rate at its center of gravity, m is the vehicle's mass, I_z is the vehicle's momentum of inertia around its vertical axis, F_{x_T} denotes the total longitudinal force of tires, F_{y_f} and F_{y_r} are the total lateral forces of the front and rear tires, and l_f and l_r are the distances from the vehicle's center of gravity to the front and rear axles, respectively.

The vehicle is assumed to have a front steering system. A linear tire model is used for the lateral tire forces [79]:

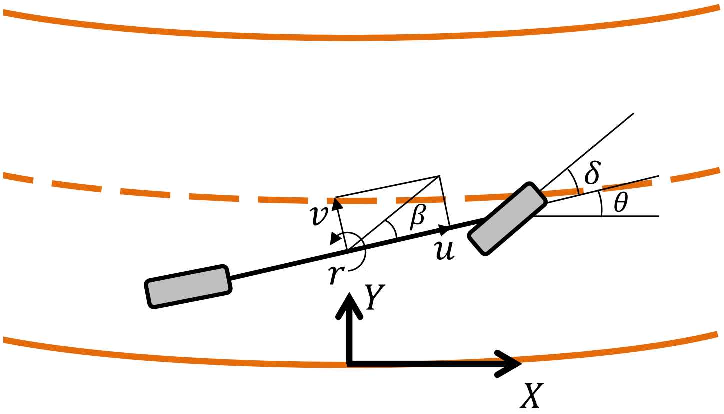


Figure 3.2: Vehicle bicycle model.

$$F_{y_f} = C_f \alpha_f = C_f \left(\delta - \frac{v + l_f r}{u} \right), \tag{3.2a}$$

$$F_{y_r} = C_r \alpha_r = C_r \left(-\frac{v - l_r r}{u} \right), \tag{3.2b}$$

where δ is the steering angle, α_f and α_r are the sideslip angles of the front and rear tires, and C_f and C_r are the cornering stiffness values of the front and rear tires. Figure 3.3 illustrates the tire's nonlinear behavior and its linear approximation with a linear tire model. The tire's sideslip angles should be limited to keep the linear approximation valid. The cornering stiffness values and the sideslip angle limit are calculated considering three criteria: 1) the error between the linear model and the nonlinear behavior should be small in the limited sideslip angle range to keep the tire model valid, 2) the lateral tire force at the sideslip angle limit should be close to the maximum lateral tire force so that most of the lateral tire force capacity is available for the vehicle model, 3) the lateral tire force calculated by the linear tire model at the sideslip angle limit should be equal to the maximum lateral tire force so that the tire force ellipse constraint explained in the next section constrains the tire force to remain within its linear region [26].

Equations (3.1) and (3.2) are linearized around the vehicle's operating point to obtain the linear vehicle dynamics:

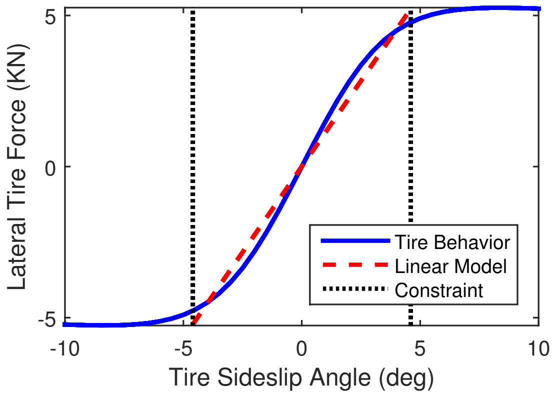


Figure 3.3: Linear tire model approximation.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}_c, \tag{3.3a}$$

$$\boldsymbol{x} = [X \quad Y \quad \theta \quad u \quad v \quad r]^T, \tag{3.3b}$$

$$\boldsymbol{u}_c = [F_{x_T} \quad \delta]^T, \tag{3.3c}$$

in which $\boldsymbol{x}$ and $\boldsymbol{u}_c$ denote the state and input vectors, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are the state and input matrices. The linear model is discretized through the zero order hold method for use as the model of the MPC.

3.3.1 Vehicle constraints

A road vehicle has limitations on the actuator capacities and the tire force capacities. Actuator capacities are considered as constraints:

$$|\delta| \leq \delta_{max}, \tag{3.4}$$

$$F_x \leq \frac{T_{max}}{R_{eff}}, \tag{3.5}$$

where δ_{max} is the maximum steering angle, R_{eff} is the effective radius of the wheels, and T_{max} is the maximum propelling torque. Since the propulsion system of the vehicle in this thesis includes four in-wheel motors, T_{max} is the total sum of the maximum motor torques. It is notable that in (3.5), wheel dynamics is neglected. Constraints are also applied on the rate of change of steering angle:

$$|\Delta\delta| \leq \Delta\delta_{max}, \tag{3.6}$$

where $\Delta\delta$ is the change of the steering angle in one step, and $\Delta\delta_{max}$ is its capacity.

Moreover, since the tire's longitudinal and lateral forces cannot exceed the friction ellipse, the model predictive controller should consider this limitation in its prediction to have an accurate prediction:

$$\left(\frac{F_{x_T}}{F_{x_T-max}}\right)^2 + \left(\frac{F_{y_*}}{F_{y_*-max}}\right)^2 \leq \mu^2, \tag{3.7}$$

where F_{x_T-max} is the maximum total longitudinal tire force , F_{y_*-max}, for $* = f, r$, is the maximum front or rear lateral tire force, and μ is the tire-road friction coefficient. It is notable that these constraints also limit the lateral tire forces to remain in their linear region.

The maximum forces in the constraint equations of (3.7) are dependent on the load transfer. Since the bicycle vehicle dynamics considers the total forces of the tires on the same wheel track, the lateral load transfer is ignored. The effects of longitudinal load transfer on the vertical front and rear tire forces is:

$$F_{z_f} = \frac{Wl_r - F_{x_T}h}{l_f + l_r}, \tag{3.8a}$$

$$F_{z_r} = \frac{Wl_f + F_{x_T}h}{l_f + l_r}. \tag{3.8b}$$

where F_{z_*}, for $* = f, r$, is the front or rear vertical tire force, h is the height of the vehicle's center of gravity from the ground, and W is the vehicle's weight. The maximum total longitudinal tire force is not affected by load transfer. However, the longitudinal load transfer affects the maximum lateral force in (3.7). Assuming that the lateral tire force capacity changes linearly with respect to the vertical tire force:

$$F_{y_*-max} = F_{y_{*0}-max}\frac{F_{z_*}}{F_{z_{*0}}}. \tag{3.9}$$

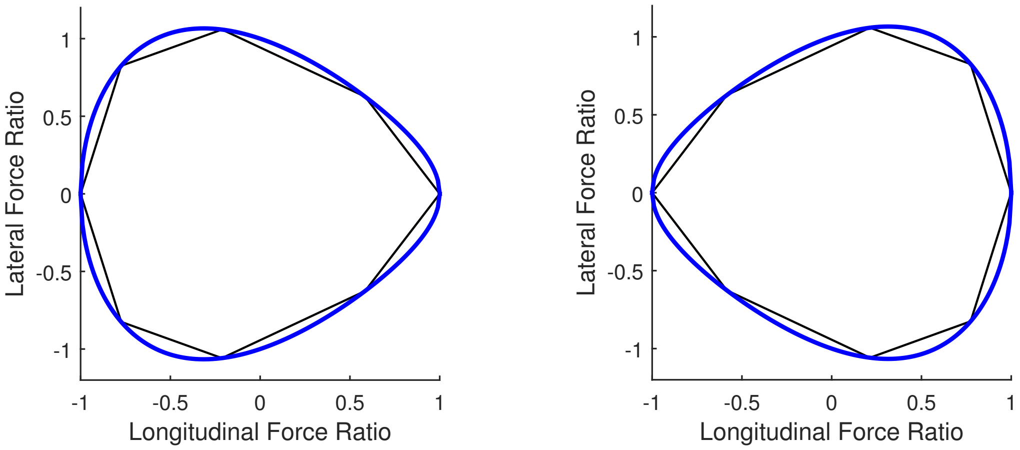
in which $F_{y_{*0}-max}$ and $F_{z_{*0}}$, for $* = f, r$, are the nominal maximum lateral front or rear tire force and the nominal vertical front or rear tire force, where nominal forces are the forces with no load transfer.

The longitudinal load transfer effect is included in the tire force ellipse constraints by applying (3.8) and (3.9) on (3.7):

$$\left(\frac{F_{x_T}}{F_{x_T-max}}\right)^2 + \left(\frac{F_{y_f}}{F_{y_{f0}-max}}.\frac{Wl_r}{Wl_r - F_{x_T}h}\right)^2 \leq \mu^2, \tag{3.10a}$$

$$\left(\frac{F_{x_T}}{F_{x_T-max}}\right)^2 + \left(\frac{F_{y_r}}{F_{y_{r0}-max}}.\frac{Wl_f}{Wl_f + F_{x_T}h}\right)^2 \leq \mu^2. \tag{3.10b}$$

The resultant constraints are illustrated in Fig. 3.4. In these figures, the longitudinal/lateral force ratio is the ratio of the longitudinal/lateral tire force to the nominal maximum longitudinal/lateral force. The constraints are nonlinear but convex. They are approximated by linear constraints so that they can be used in the quadratic MPC problem. Each constraint is approximated by a octagon inscribed in it. The hexagon is calculated by minimizing the area between the original constraint and its hexagon approximation to maximize the available tire force for the vehicle model.



(a)

(b)

Figure 3.4: Tire force constraints and their polyhedral approximations, a) front tire force constraint, b) rear tire force constraint

Moreover, the vehicle speed should not exceed the maximum speed limit. The limit is considered by a constraint:

$$u \leq u_{max}, \tag{3.11}$$

where u_{max} denotes the maximum allowed vehicle speed.

3.4 Potential Field

A potential field is a field generated by the obstacle and goal Potential Functions (PFs) to lead the vehicle toward the goal while keeping it away from the obstacles. A goal PF has a minimum at the goal so that the goal attracts the vehicle, and an obstacle PF has a maximum at the obstacle position so that the obstacle repulses the vehicle. In this thesis, the task of leading the vehicle towards its goal is performed by the tracking terms in the objective function of the motion planning MPC. Therefore, the potential field generated here is repulsive only, and is constructed of obstacle PFs. A PF, U_{R_q}, is defined for the lane markers to prevent the vehicle from going out of the lane. The obstacles are categorized as crossable and non-crossable obstacles, and a PF is defined for each category based on it avoidance necessity, U_{NC_i} for a non-crossable obstacle and U_{C_j} for a crossable obstacle. The potential field, U, is the sum of the PFs:

$$U = \sum_i U_{NC_i} + \sum_j U_{C_j} + \sum_q U_{R_q}, \tag{3.12}$$

where indices i, j, and q denote the i^{th} non-crossable obstacle, the j^{th} crossable obstacle, and the q^{th} lane marker, respectively. The presented functions below are some sample functions; other functions can be used for modeling other road regulations and obstacles.

3.4.1 Non-crossable Obstacles

Non-crossable obstacles are obstacles that should not be crossed since they are either important themselves, like a pedestrian, or can cause a damage to the vehicle, like a car. A hyperbolic function of the distance between the vehicle and the obstacle is used to generate the potential field caused by this kind of obstacle. The rate of change of the function strictly increases as the distance decreases, and it approaches to infinity, which prevents the vehicle from crossing the obstacle. Schulman et al. [80] use the Signed Distance (SD) between the vehicle shape and the obstacle shape for collision avoidance. Figure 3.5 shows the signed distance of two shapes. The SD is the minimum distance of the shapes if there is no contact between the shapes, and the negative of the penetration distance if there are contact points. More information on the signed distance can be found in [81].

The PF is generated as a function of the SD, s_i:

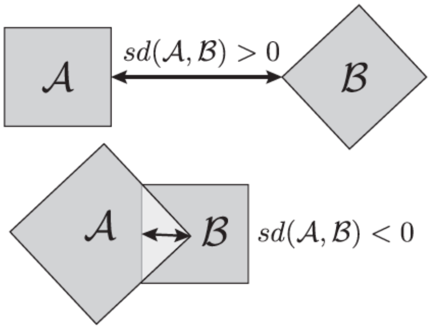
$$U_{NC_i}(X, Y) = \frac{a_i}{s_i(\frac{X}{X_{s_i}}, \frac{Y}{Y_{s_i}})^{b_i}}, \tag{3.13}$$

where a_i and b_i are intensity and shape parameters of the PF, respectively. A SD is a function of the position of the vehicle and the obstacle and their shapes. But, since the position of the obstacle and the shapes of the vehicle and the obstacles are know prior to generation of the potential function, s_i is demonstrated as a function of vehicle position only in (3.13). In addition, the vehicle needs to have a larger distance to the obstacle in the longitudinal direction than the lateral direction. Therefore, the SD is normalized by the safe longitudinal and lateral distances from the obstacle, X_{s_i} and Y_{s_i}, which are defined as:

$$X_{s_i} = X_0 + uT_0 + \frac{\Delta u_{a_i}^2}{2a_n}, \tag{3.14a}$$

$$Y_{s_i} = Y_0 + (u \sin\theta_e + u_{o_i} \sin\theta_e)T_0 + \frac{\Delta v_{a_i}^2}{2a_n}, \tag{3.14b}$$

In (3.14a) for the safe longitudinal distance, the first term is the minimum safe longitudinal distance, X_0, and the second term is the distance spanned by the vehicle during the safe time gap included to consider the risk due to vehicle speed. The third term of this equation is the distance corresponding to the longitudinal velocity difference between the vehicle and the obstacle included to consider the risk due to the speed difference [82]. In (3.14b) for safe lateral distance, the first term is the minimum safe lateral distance, Y_0, and the second term is the lateral distance spanned by the vehicle and the obstacle during the safe time gap if they have the constant heading angles of θ_e toward each other, which is included to consider the risk due to vehicle and obstacle speed. The third term of this equation is the distance corresponding to the lateral velocity difference between the vehicle and the obstacle included to consider the risk due to lateral velocity difference. The safe time gap, denoted by T_0, compensates for the vehicle response time, and its value is assigned accordingly. Furthermore, u_{o_i} is the longitudinal velocity of the i^{th} obstacle, a_n is the normal acceleration, and Δu_{a_i} and Δv_{a_i} are the approaching velocities in the longitudinal and lateral directions. In each direction, the approaching velocity is set to the velocity difference between the vehicle and the obstacle if they are approaching, and it is set to zero otherwise. The SD is normalized due to the desire that the distance of the vehicle and the obstacle in longitudinal orientation is more than the safe longitudinal distance and in lateral orientation is more than the safe lateral distance.



$\mathcal{A}$
$sd(\mathcal{A},\mathcal{B}) > 0$
$\mathcal{B}$
$\mathcal{A}$
$\mathcal{B}$
$sd(\mathcal{A},\mathcal{B}) < 0$

Figure 3.5: Signed distance of two shapes [58].

Moreover, a zero SD results in an infinite PF. In addition, with this PF, the vehicle would have no longitudinal response to the obstacle approaching from the side if the longitudinal component of the SD is zero but a driver would brake in this situation. These issues are resolved with a modification in the calculation of the SD; if the longitudinal distance between the vehicle and the obstacle is less than a threshold, ΔX_0, it is set to ΔX_0 with the obstacle being ahead.

If the vehicle and the obstacle are approaching each other, there is a region around the obstacle where the vehicle cannot avoid a collision. The longitudinal and lateral collision distances, X_{c_i} and Y_{c_i}, are defined as the maximum distances from the obstacle in the longitudinal and lateral directions at which the collision cannot be avoided. In each direction, the collision distance is the distance required to change the approaching velocity to zero by modifying the vehicle velocity with the maximum acceleration, a_{max}:

$$X_{c_i} = \frac{\Delta u_{a_i}^2}{2a_{max}}, \tag{3.15a}$$

$$Y_{c_i} = \frac{\Delta v_{a_i}^2}{2a_{max}}, \tag{3.15b}$$

The intensity and shape parameters of (3.13) are calculated by assigning the safe potential parameter, U_{saf}, and the accident potential parameter, U_{acc}, to the PF at the safe distance and the collision distance, respectively:

$$U_{NC_i} = \begin{cases} U_{saf} & s_i = 1 \\ U_{acc} & s_i = s_c \end{cases} . \tag{3.16}$$

It is notable that in order to be at the safe distance from the obstacle, the vehicle just needs to be at the safe distance in either lateral or longitudinal direction. The same expression holds for the collision distance. Therefore, the collision SD, s_c, is the maximum of the corresponding SD of the longitudinal collision distance and the corresponding SD of the lateral collision distance. The potential field of an obstacle vehicle located at $(X_{o_i}, Y_{o_i}) = (20, 3.5)m$ and moving at the same speed as the vehicle at $80Km/h$ is shown in Fig. 3.6.

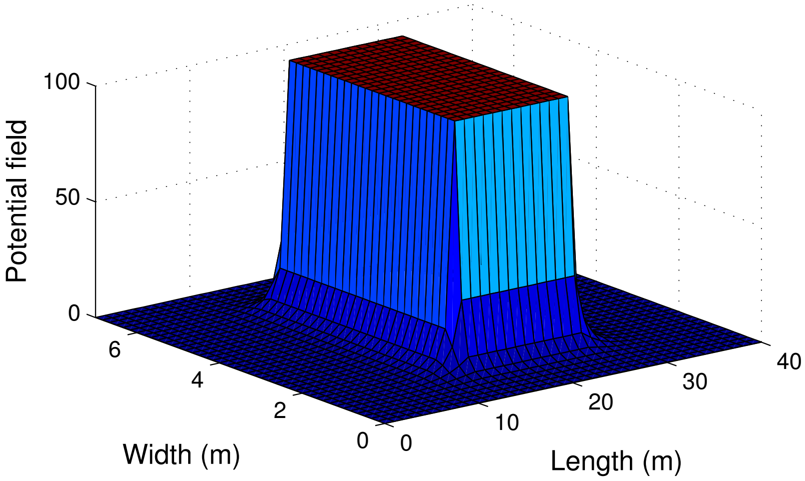


Figure 3.6: Non-crossable obstacle potential field.

3.4.2 Crossable Obstacles

Crossable obstacles are obstacles that can be crossed without any damage, but it is preferred not to cross them, if possible, like a low profile object or a bump on the road. The PF of such an obstacle is defined with an exponential function:

$$U_{C_j}(X, y) = a_j e^{-b_j s_j(\frac{X}{X_{s_j}}, \frac{Y}{Y_{s_j}})}, \tag{3.17}$$

where s_j is the normalized SD between the vehicle and the obstacle calculated similar to (3.13),(3.14). a_j and b_j are also the intensity and shape parameters, which are calculated similar to (3.14)-(3.16) except that the uncomfortable potential parameter, U_{unc}, is assigned to the PF at the collision distance.

The exponential function repulses the vehicle from the obstacle everywhere because of its positive gradient. But, at positions close to the obstacle, the gradient decreases as the distance to the obstacle decreases, which allows the vehicle to cross the obstacle. Figure 3.7 shows the potential field generated by this function for a similar situation to that of Fig. 3.6.

3.4.3 Lane boundaries

In a structured road, the vehicle should not cross the road lane markers unless a lane change is desired. To avoid undesirable lane marker crossings, PFs are defined for lane markers:

$$U_{R_q}(X, Y) = \begin{cases} a_q \left(s_q(X, Y) - D_a\right)^2 & s_q(X, Y) \leq D_a \\ 0 & s_q(X, Y) > D_a \end{cases}, \tag{3.18}$$

where s_q is the SD of the vehicle from the lane marker, D_a is the allowed distance from the lane marker, index $q = r, l$ denotes the right or left lane marker, and a_q, is the intensity

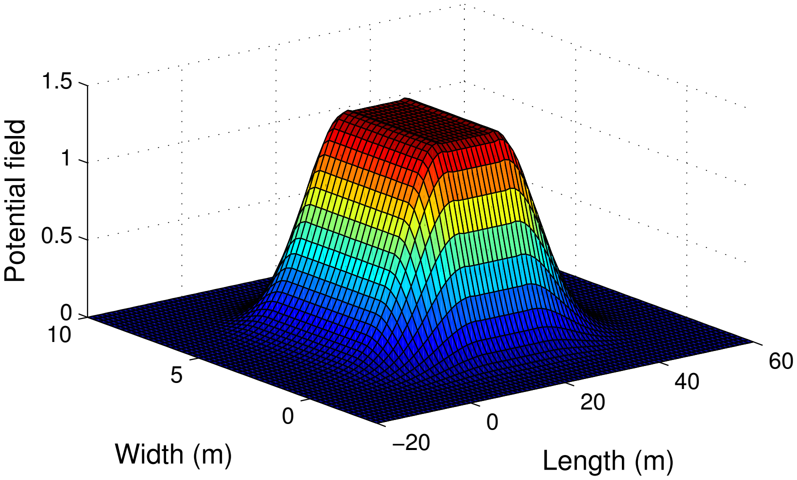


Figure 3.7: Crossable obstacle potential field.

parameter calculated by assigning the lane marker potential parameter, U_{lma}, to the PF at zero SD.

If a lane keeping is intended, the right and left lane markers are the ones on which the PFs are implemented. If a lane change is intended, the PF is not implemented on the lane marker that can be crossed for the lane change. It is implemented on the next lane marker instead. The lane marker PFs are defined with quadratic functions, and their gradients increase linearly as the SD decreases. Therefore, the vehicle can cross the lane markers to any extent, but the farther the vehicle goes from the middle of the lane, the harder the PF pushes it toward there. Figure 3.8 shows the road PF for a lane change maneuver on a two lane road.

3.5 MPC Problem

In this section, a motion planning MPC is developed with the presented vehicle dynamics model and constraints. The presented potential field for obstacles and road regulations is added to the controller objective function to include obstacle avoidance and road regulation

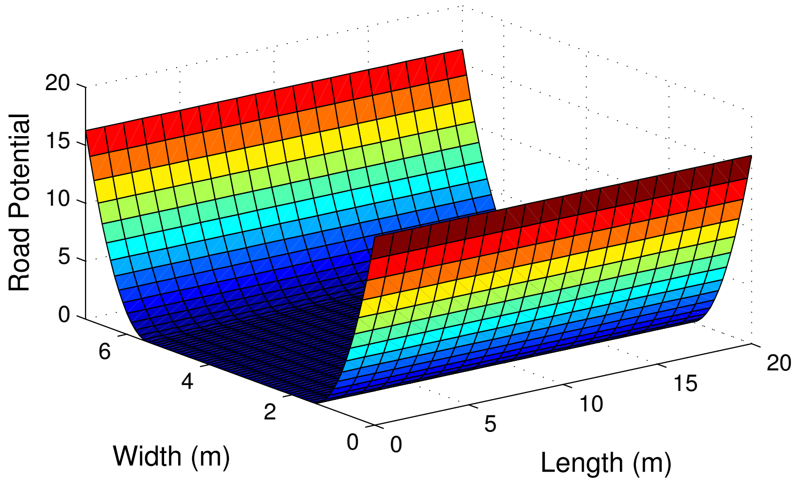


Figure 3.8: Lane potential field for a lane change.

observation to the MPC. The model predictive controller predicts the response of the vehicle up to a horizon, and optimizes the vehicle dynamics, command following, obstacle avoidance, and road regulations observation up to that horizon based on the predicted values.

It is assumed that the desired lane and speed are predefined. Therefore, the desired lateral position, which is the center of the desired lane, and the desired longitudinal velocity are the outputs to be tracked:

$$\boldsymbol{y} = [Y \quad u]^T, \tag{3.19a}$$

$$\boldsymbol{y}_{des} = [Y_{des} \quad u_{des}]^T, \tag{3.19b}$$

$$Y_{des} = (l_{des} - 1/2)L_w + \Delta Y_R, \tag{3.19c}$$

where $\boldsymbol{y}$ is the output vector tracking the desired output vector, $\boldsymbol{y}_{des}$, Y_{des} is the desired lateral position, u_{des} is the desired vehicle speed, L_w is the lane width, ΔY_R is the lateral offset of the road compared to a straight road, and l_{des} is the index number of the desired lane counted from the right.

The constraints of (3.10) and (3.11) are applied in the optimal control problem as soft constraints. A soft constraint can be violated, but its violation is penalized. A slack variable is added to the constraint equation to allow some violation, and it constructs a penalty term in the objective function of the MPC to penalize the violation. It is notable that although surpassing the tire ellipses is physically impossible, the constraints on the tire forces are considered soft. It is because the constraints are models of the actual tire limitation and might differ from the exact limitations. Moreover, the errors in the estimated states might also cause constraint violations. Therefore, the tire constraints are considered as soft constraints to avoid infeasiblity due to constraint violation.

The motion planning MPC problem is:

$$\min_{\boldsymbol{u}_c,\boldsymbol{\varepsilon}} \sum_{k=1}^{N_p} \left(U^{t+k,t} + \|\boldsymbol{y}^{t+k,t} - \boldsymbol{y}_{des}^{t+k,t}\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_c^{t+k-1,t}\|_{\boldsymbol{R}}^2 + \|\boldsymbol{u}_c^{t+k-1,t} - \boldsymbol{u}_c^{t+k-2,t}\|_{\boldsymbol{S}}^2 \right) + \|\boldsymbol{\varepsilon}\|_{\boldsymbol{\lambda}}^1, \tag{3.20a}$$

$$s.t.(k = 1, \dots, N_p)$$

$$\boldsymbol{x}^{t+k,t} = \boldsymbol{A}_d \boldsymbol{x}^{t+k-1,t} + \boldsymbol{B}_d \boldsymbol{u}_c^{t+k-1,t}, \tag{3.20b}$$

$$\boldsymbol{y}^{t+k,t} = \boldsymbol{C}\boldsymbol{x}^{t+k,t} + \boldsymbol{D}\boldsymbol{u}_c^{t+k,t}, \tag{3.20c}$$

$$\boldsymbol{y}_s^{t+k,t} = \boldsymbol{C}_s \boldsymbol{x}^{t+k,t} + \boldsymbol{D}_s \boldsymbol{u}_c^{t+k,t}, \tag{3.20d}$$

$$\boldsymbol{y}_s^{t+k,t} \leq \boldsymbol{y}_{s-max}^{t+k,t} + \boldsymbol{\varepsilon}, \tag{3.20e}$$

$$\boldsymbol{\varepsilon} \geq 0, \tag{3.20f}$$

$$\boldsymbol{u}_c^{t+k,t} = \boldsymbol{u}_c^{t+k-1,t}, \quad k > N_c, \quad k \neq c_2 N_{rc} + N_c, \quad c_2 = 1, \dots, (N_p - N_c)/N_{rc}, \tag{3.20g}$$

$$\boldsymbol{u}_c^{t-1,t} = \boldsymbol{u}_c(t-1), \tag{3.20h}$$

$$\boldsymbol{x}^{t,t} = \boldsymbol{x}(t), \tag{3.20i}$$

where $t+k,t$ index denotes the predicted value at k steps ahead of the current time t, N_p is the prediction horizon, and $\boldsymbol{\varepsilon}$ is the vector of slack variables. The objective function includes the predicted potential field defined in (3.12), quadratic terms of tracking, inputs, and changes in inputs with weighting matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{S}$, respectively, and first norm of slack variables weighted with $\boldsymbol{\lambda}$. The quadratic terms of the input and the changes in the input are included to minimize the consumption energy and the jerk, respectively. The states are predicted through (3.20b), which is obtained by discretizing (3.3a) to obtain $\boldsymbol{A}_d$ and $\boldsymbol{B}_d$ as the discrete state and input matrices. Equation (3.20c) calculates the tracking outputs, where $\boldsymbol{C}$ and $\boldsymbol{D}$ are the output and feedforward matrices. The constraint on the actuators, (3.4)-(3.6), on the vehicle speed, (3.11), and corresponding linear constraints of the tire capacity constraint, (3.10), are presented in (3.20e), where $\boldsymbol{y}_s$ is the vector of constraint variables and is bounded by $\boldsymbol{y}_{s-max}$, the vector of constraint bounds, and the slack variable vector is included to allow violation of the bounds. The slack variables corresponding to actuator constraints are set to zero since they cannot be violated. The constraint variables are linearized around the operating point, which are to be written as a function of states and inputs in (3.20d), where $\boldsymbol{C}_s$ and $\boldsymbol{D}_s$ are the constraint output and feedforward matrices. The computation cost is reduced by reducing the number of control inputs in (3.20g); after the first N_c prediction steps, the control inputs change every N_{rc} steps.

The presented optimization problem can be solved for any PF. However, because of the nonlinear nonconvex PFs, the problem is nonlinear and nonconvex, and its solution is expensive. Its approximated quadratic convex problem can be solved noticeably faster. Thus, to reduce the calculation time, the problem is converted into a quadratic convex problem. To do so, the PFs are first approximated by convex functions.

The PFs are defined on (X, Y). Olfati-Saber [83] defines the obstacle PF only in the SD's direction to generate the repellent force. For each obstacle and at each prediction step, the PFs defined in this chapter are transformed to a coordinate, (ξ_i, η_i), that has one axis (ξ_i) in the direction of the SD:

$$\begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} s_{X_i} \\ s_{Y_i} \end{bmatrix}. \tag{3.21}$$

Figure 3.9 illustrates the coordinate transformation. The black coordinate is the road coordinate and the red coordinate is the SD coordinate, which is normalized with the safe distances. The red rectangle is the vehicle in this coordinate, and s_i is the SD. The vehicle position at the prediction step k is anticipated based on the vehicle speed and heading angle at time step t. The angle between the SD at this position and s_{X_i}-axis is γ. The blue coordinate, (ξ_i, η_i), is obtained by rotating the SD coordinate by this angle.

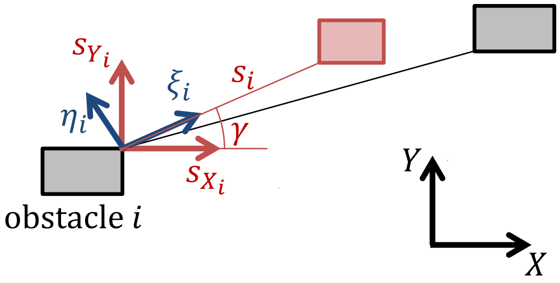


Figure 3.9: Coordinate Transformation.

The PFs defined in Section 3.4 can all be written as a function of s_i instead of (X, Y). In other words, for a PF, $g : \mathbb{R}^2 \to \mathbb{R}$, there is a function, $h : \mathbb{R} \to \mathbb{R}$, that $h(s_i) = g(X, Y)$. Moreover, the PF, g, can be transformed from (X, Y) to (ξ_i, η_i) by (3.21) to obtain the transformed PF, $g_T : \mathbb{R}^2 \to \mathbb{R}$, where $g_T(\xi_i, \eta_i) = g(X, Y)$. Considering the definition of the SD, the gradient and Hessian of g_T are:

$$s_i = \begin{cases} (\xi_i^2 + \eta_i^2)^{1/2} & \textit{no contact} \\ -(\xi_i^2 + \eta_i^2)^{1/2} & \textit{in contact} \end{cases}, \tag{3.22a}$$

$$\nabla g_T = \begin{bmatrix} \frac{\xi_i}{s_i} h' & \frac{\eta_i}{s_i} h' \end{bmatrix}^T, \tag{3.22b}$$

$$\nabla^2 g_T = \begin{bmatrix} \frac{\xi_i^2}{s_i^2}h'' + \frac{s_i^2-\xi_i^2}{s_i^3}h' & \frac{\xi_i\eta_i}{s_i^2}h'' - \frac{\xi_i\eta_i}{s_i^3}h' \\ \frac{\xi_i\eta_i}{s_i^2}h'' - \frac{\xi_i\eta_i}{s_i^3}h' & \frac{\eta_i^2}{s_i^2}h'' + \frac{s_i^2-\eta_i^2}{s_i^3}h' \end{bmatrix}, \tag{3.22c}$$

where h' and h'' are the first and second derivatives of function h with respect to s_i. From (3.22b), it can be seen that, at the anticipated vehicle position, the gradient is in ξ_i direction, i.e. the repellent force is only in the direction of the SD, as it is in [83]. Moreover, due to (3.22c), the Hessian matrix is uncorrelated at the anticipated vehicle position in the new coordinate. Therefore, the function is convex at this position if both diagonal elements are non-negative. If any diagonal element is negative, the function is linearized at the corresponding direction of the element, using the first order Taylor series. The resulting function is a convex function convexified around the anticipated operating point.

The convex function is then transformed to the original coordinate, (X, Y). Since convexity holds for a linear transformation, the transformed function is also a convex function. The whole process is equivalent to an eigenvalue decomposition process that only keeps the positive eigenvalues. Therefore, the Hessian of the resulted function is the closest positive definite matrix to the Hessian of the original function in terms of the Frobenius norm [84].

The resulting convex function is then approximated by a quadratic function through the second order Taylor series. The quadratic function is a close convex quadratic approximation of the original function around the nominal point; its gradient equals the original function's gradient, and its Hessian matrix is the closest positive definite matrix to the original function's Hessian matrix in terms of the Frobenius norm. The quadratic approximation adds calculation time spent on transformations, first and second derivatives, and Taylor series approximations. However, the added time is negligible compared to the calculation time of the optimization problems. Using the resulting PFs, the optimal control problem is a convex quadratic optimization problem. The problem is similar to a corresponding nonlinear problem solved by Sequential Quadratic Programming (SQP) in one sequence. Boggs et al. [85] derive an upper bound for the optimization error of each sequence of SQP, where the optimization error is the difference between the result of the sequence and the local minimum of the nonlinear problem in the neighborhood of the problem's initial value. Based on this upper bound, for the quadratic problem, the closer the problem's initial value is to the minimum, which is equivalent to the anticipated vehicle point being closer to the vehicle position at the minimum, the smaller the optimization error. Moreover, the closer the calculated Hessian matrices of the PFs to their Hessian matrices at the minimum, the smaller the optimization error. Therefore, a PF with a smaller convex quadratic approximation error and a smaller variation of the Hessian matrix in the neighborhood of the problem's initial value result in a smaller optimization error. In the next section, the performance and the calculation time of the nonlinear problem and the quadratic problem are compared for a scenario. The other scenarios are simulated only for the quadratic problem.

3.6 Results

3.6.1 Test Scenarios

Roads are dynamic environments with obstacles moving at different speeds in different lanes and positions. The roads themselves might be curved, and a lane might end or begin. Moreover, a vehicle might be required to change its lane or stay in the lane to take an exit or turn. For any combination of the obstacles, road, and intended lane, the undertaken maneuver might be different. In this section, some test scenarios are defined to evaluate the performance of an autonomous driving system. Some normal scenarios for an autonomous driving system are:

- Lane keeping on curved roads,
- Lane changing with no obstacle in the vicinity,
- Keeping a desired distance from the vehicle in front of the ego vehicle (adaptive cruise control).

Other more complicated scenarios that an autonomous driving system should be able to perform include:

- Lane changing while there are vehicles on the intended lane,
- Merging into a highway while there are vehicles on the right lane,
- A vehicle carelessly approaching the ego vehicle from the side,
- Non-crossable static obstacle on the lane,
- Crossable static obstacle on the lane.

The aforementioned complicated maneuvers are only some of the many cases that might happen when driving on a road. However, they can evaluate the performance of motion planning systems in observing safety and road regulations. The first and second cases test the vehicle in observing safety and road regulations in a lane change. The vehicle should change the lane as soon as it is safe and keep its lane if it is not safe. In the second case, the current lane is ending and the vehicle may need to reduce its speed or even stop before the lane ends. The corresponding maneuvers of these situations include normal maneuvers such as lane changing and modifying speed to keep distance from the obstacles.

The third case tests the motion planning system in predicting the lateral movement of the obstacles and taking action in emergency situations while observing the road regulations. The vehicle should be able to predict the obstacle's path and avoid the accident while keeping its lane, which is performed by keeping some space from the obstacle via accelerating or decelerating. It includes simple maneuvers such as lane keeping and keeping a safe distance from the obstacles.

The fourth and fifth cases test the motion planning system for observation of the road regulations. The vehicle should keep its lane; if there is enough lateral space on the lane, it should pass the obstacle on the side; otherwise, it should stop behind the obstacle or cross it. It also tests the motion planning system for prioritizing obstacles. In the situation that there is not enough lateral space for passing the obstacle on the side, if the obstacle is not crossable, the vehicle should stop behind it, and if it is crossable, the vehicle should cross it.

Altogether, these cases are appropriate for evaluating the performance of motion planning systems in observing the safety and road regulations, obstacle avoidance, and longitudinal and lateral maneuverability. They also evaluate how the motion planning system performs in prioritizing obstacles based on their avoidance necessities. The following test scenarios are defined based on the mentioned cases:

Scenario 1: The vehicle is merging onto a highway and its lane ends in $150m$. It should change its lane from Lane 1 to Lane 2 while there are three vehicles in Lane 2. There is not enough space between these vehicles for the ego vehicle to merge safely between them.

Scenario 2: The vehicle starts in Lane 1 and is commanded to change its lane while there are three vehicles in Lane 2. There is enough space between these vehicles for the ego vehicle to go in between them. The road is curved with a radius of $300m$ for $X = \begin{bmatrix} 200 & 250 \end{bmatrix} m$ and a radius of $-300m$ for $X = \begin{bmatrix} 250 & 300 \end{bmatrix} m$.

Scenario 3: The vehicle starts in Lane 1 and is commanded to stay in Lane 1. There is a vehicle in Lane 2 at the same longitudinal position and with the same speed as the ego vehicle. It moves laterally from the center of Lane 2 towards the center of Lane 1 with a constant lateral velocity in the time interval of $t = \begin{bmatrix} 1 & 6 \end{bmatrix} m$. The ego vehicle should make enough space for it to avoid collision.

Scenario 4: The vehicle starts in Lane 1 and is commanded to stay in Lane 1. There is a static non-crossable obstacle on Lane 1 located at $0.5m$ from the right boundary of the lane. The obstacle is assumed to be a square obstacle with $0.5m$ length, and there is enough lateral space on the lane for the vehicle to pass it.

Scenario 5: The scenario is the same as Scenario 4 except that the obstacle is crossable.

Scenario 6: The scenario is the same as Scenario 4 except that the obstacle is located at $1.5m$ from the right boundary of Lane 1, and therefore, there is not enough lateral space on the lane for the vehicle to pass the obstacle.

Scenario 7: The scenario is the same as Scenario 6 except that the obstacle is crossable.

The initial vehicle speed, u_0, the desired vehicle speed, u_{des}, the speed of obstacles, u_{o_i}, and initial position of the obstacles relative to the vehicle, X_{o0_i}, are listed in Table 3.1.

Table 3.1: Test Scenario Parameters

	u_0	u_{des}	V_{o_1}	V_{o_2}	V_{o_3}	X_{o0_1}	X_{o0_2}	X_{o0_3}
	(Km/h)	(Km/h)	(Km/h)	(Km/h)	(Km/h)	(m)	(m)	(m)
Scenario 1	100	100	100	100	100	-40	0	40
Scenario 2	80	100	100	100	100	-25	0	25
Scenario 3	80	80	80	–	–	0	–	–
Scenarios 4-7	80	80	0	–	–	80	–	–

3.6.2 Simulation

The proposed motion planning MPC is simulated on a vehicle system to evaluate the performance of the controller. The vehicle system used in the simulation is a model of a Chevrolet Equinox in CarSim software. The vehicle parameters used in the MPC are extracted from this vehicle model. The controller parameters are shown in Table 3.2 for a dry road. The vehicle is electric with four wheel electric motors. The motor torques and brake torques that generate the total force are calculated and applied to each wheel. The torques are distributed proportional to the wheels' vertical force. A slip controller is also applied on each wheel to avoid large slip ratios. The upper bound on the longitudinal force is based on the motors' torque capacities, which are determined from the motors' specifications. The bound varies with the vehicle speed, and the one at the vehicle speed of $80Km/h$ is presented in the table. Also, the maximum speed, u_{max}, is assumed to be 10% over the desired speed.

In this section, the controller is simulated for the scenarios presented in the previous section so that its performance in observing the road regulations, obstacle avoidance, maneuverability, and prioritizing obstacles is evaluated. The controller time step is $50ms$, and the prediction horizon is 20 steps, making the prediction time to be $1s$. The optimization problem is solved on an Intel Core-i7 3.4GHz CPU with QPOASES [86], which is a quadratic programming solver.

At each time step, a potential function is generated for each obstacle at each prediction step. The obstacle position at each prediction step is predicted assuming that the obstacle moves with the same longitudinal and lateral speed of the current time step. The vehicle position is also predicted at each prediction time step assuming that the vehicle moves with the same longitudinal and lateral speed of the current time step. For each prediction time step, a potential function is generated for each obstacle based on the predicted obstacle position, and is approximated by a quadratic convex function around the predicted vehicle position. Then, the vehicle current state is used as the MPC initial state, and matrices required for quadratic programming problem of QPOASES are generated based on the objective function and constraints defined in (3.20) using the state, input, output, and feedforward matrices. The MPC problem is solved by QPOASES using the generated matrices to calculate the vehicle inputs. Next, the CarSim vehicle model is simulated for one step. The input of the CarSim vehicle model is the calculated vehicle inputs, and its outputs are the vehicle states, which are used to as the vehicle current states in the calculations of the next step.

Scenario 1 is a merging maneuver when there are moving obstacles on the other lane and the current lane is ending. The scenario is simulated for the nonlinear and quadratic motion planning problems, and the simulation results are shown in Fig. 3.10. The paths of the ego vehicle and obstacles are shown in Fig. 3.10a. In this figure, at some sample times, markers are used to demonstrate the position of the vehicle and obstacles; each shape represents a sample time, and each color represents the vehicle or each of the obstacles. As shown, the vehicle waits for all the obstacles to pass; the potential fields of the obstacles keep the vehicle away from Lane 2 when there are obstacles occupying it. Moreover, a potential field of a static obstacle located at the end of Lane 1 is added to the existing potential field to keep the vehicle from passing the end of the lane. Due to this potential field, the vehicle reduces its speed and avoids passing the end of the lane. After all the obstacles pass, the vehicle changes its lane safely. At the end of the lane change, the potential field of the left lane boundary keeps the vehicle from going out of the road.

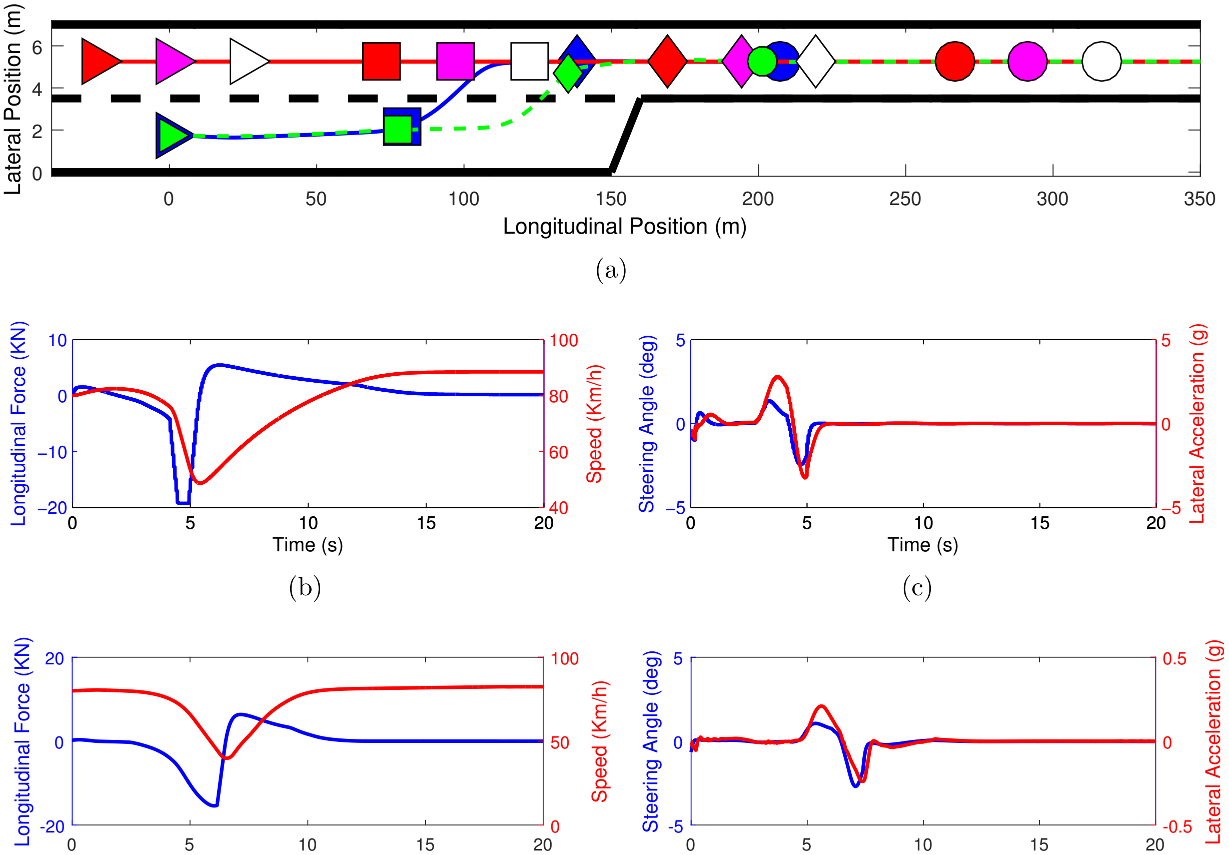
The scenario is simulated for the nonlinear and quadratic problems. As it can be seen, the quadratic motion planning system imitates the behavior of the nonlinear motion planning system. The difference between the simulation results is noticed closer to the end of the lane. At this location, the required large deceleration causes an error in the anticipated longitudinal vehicle position. Moreover, since the anticipated vehicle position

Table 3.2: Controller Parameters

Parameter	Value	Unit	Parameter	Value	Unit
m	2270	Kg	T_0	0.25	s
I_z	4600	Kgm^2	R_{eff}	0.351	m
l_f	1.421	m	D_a	0.5	m
l_r	1.434	m	L_w	3.5	m
C_f	127000	N	ΔX_0	1	m
C_r	130000	N	F_{x_T-max}	21400	N
W	22268.7	N	$F_{y_{f0}-max}$	10400	N
h	0.647	m	$F_{y_{r0}-max}$	10600	N
μ	0.9	-	T_{max}	3000	Nm
a_{max}	9	m/s^2	δ_{max}	10	deg
a_n	1	m/s^2	$\Delta\delta_{max}$	0.5	deg
U_{saf}	1	N_p	20	-	
U_{acc}	10	-	N_c	5	-
U_{unc}	2	-	N_{rc}	5	-
U_{lma}	2	-	$\boldsymbol{Q}$	$\begin{bmatrix}0.2 & 0.01\end{bmatrix}$	-
X_0	1.5	m	$\boldsymbol{R}$	$[2e-9 \quad 100]$	-
Y_0	1	m	$\boldsymbol{S}$	$[5e-8 \quad 500]$	-

is too close to the end of the lane, the error in approximating the hyperbolic PF of the end of the lane by a quadratic convex function becomes more noticeable. These two sources cause the differences in the results of the quadratic problem. Despite the differences, the performance of the quadratic problem is comparable to that of the nonlinear problem. On the other hand, the average calculation time of the nonlinear problem for a time step of this simulation is $21.03s$ while that of the quadratic problem is $0.0094s$. It is notable that since the step time is $0.05s$, the quadratic problem can be solved in real time. The other scenarios are simulated for the quadratic problem.

Scenario 2 is a lane change while there are moving obstacles on the intended lane. Figure 3.11 shows the simulation results for this scenario. Since there is a moving obstacle



(a)

(b)

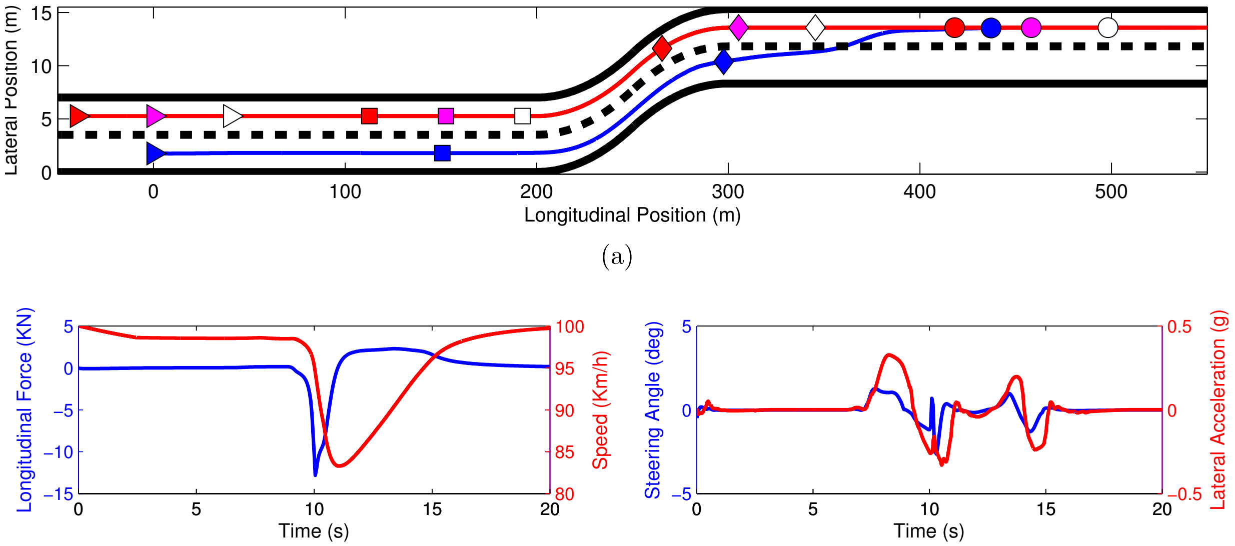
(c)

(d)

(e)

Figure 3.10: Scenario 1 for nonlinear and quadratic problems, a) paths of vehicle and obstacles, blue: vehicle for nonlinear problem, green: vehicle for quadratic problem, red: Obstacle 1, purple: Obstacle 2, white: Obstacle 3, b) longitudinal force command and vehicle speed for the nonlinear problem, c) steering angle command and lateral acceleration for the nonlinear problem, d) longitudinal force command and vehicle speed for the quadratic problem, e) steering angle command and lateral acceleration for the quadratic problem.

on the side of the vehicle, the vehicle cannot proceed with the lane change immediately; the potential fields of the obstacles keep the vehicle away from Lane 2. The vehicle slightly reduces its speed, and waits for the obstacle on its side to pass. When there is enough



(a)

(b)

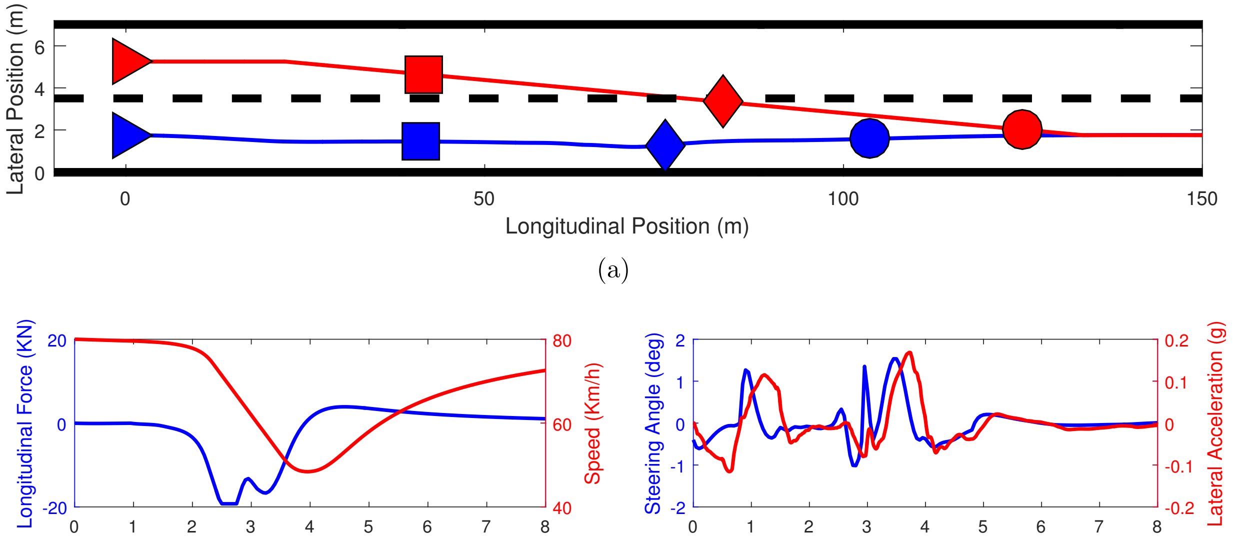
(c)

Figure 3.11: Scenario 2, a) paths of vehicle and obstacles, blue: vehicle, red: Obstacle 1, purple: Obstacle 2, white: Obstacle 3, b) longitudinal force command and vehicle speed, c) steering angle command and lateral acceleration.

distance between the obstacles and the vehicle, it moves to the other lane while keeping its distance from the both obstacles by adjusting its speed. The lateral movements of the vehicle and its speed changes are according to the PFs keeping the vehicle away from the obstacles. It can also be seen that the motion planning system can handle the maneuvers on a curved road.

In this scenario, the vehicle merges in between the obstacles since there is enough space. In Scenario 1, there was less space between the obstacles, and the vehicle's speed was largely different from obstacles' speeds. Therefore, going in between the obstacles was not safe enough and the potential fields of the obstacles kept the vehicle in Lane 1 until all the obstacles passed the vehicle and the lane change was safe.

The third scenario is when a moving obstacle beside the vehicle carelessly changes its lane to the vehicle's current lane. The simulation results for Scenario 3 are shown in Fig. 3.12. Due to the potential field of the obstacle, the vehicle reduces its speed to make some



(a)

(b)

(c)

Figure 3.12: Scenario 3, a) paths of vehicle and obstacle, blue: vehicle, red: obstacle b) longitudinal force command and vehicle speed, c) steering angle command and lateral acceleration.

space for the obstacle, and moves to the right to keep its lateral distance from the obstacle and avoid collision. The potential field of the right boundary lane, on the other hand, leads the vehicle towards the middle of the lane and keeps the vehicle in the lane. By the time the obstacle is on the middle lane marker, the vehicle has made approximately $10m$ of longitudinal space to keep a safe distance from the obstacle. The vehicle goes back towards the center of the lane, due to the right lane boundary PF, after making enough longitudinal space for obstacle avoidance.

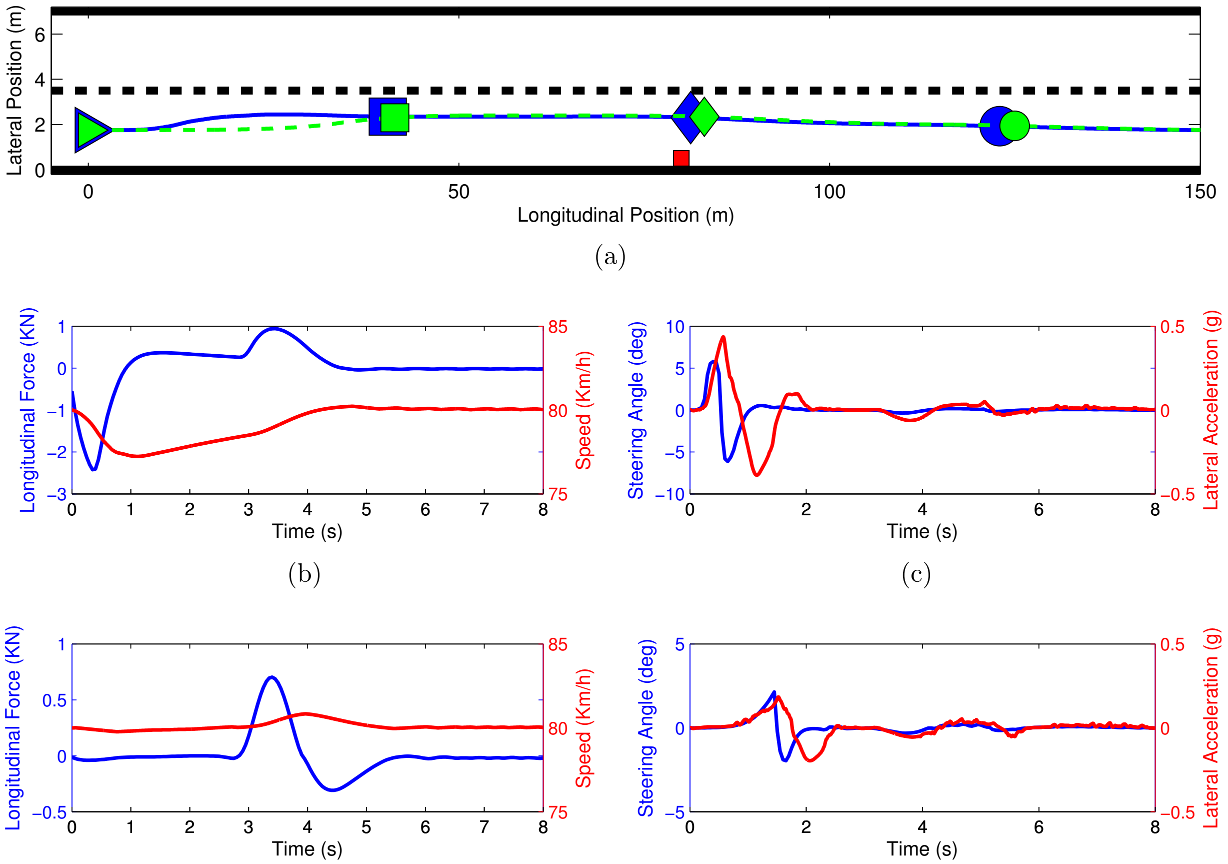
Scenarios 4-7 are designed to show the different responses of the motion planning MPC to different kinds of obstacles. Two kinds of obstacles are considered: crossable obstacles and non-crossable obstacles. Scenarios 4 and 5 are when there is a crossable or non-crossable obstacle in the current lane of the vehicle, but there is enough lateral space to pass the obstacle on the side. The simulation results for these scenarios are shown in Fig. 3.13. The PFs of the obstacles lead the vehicle to the left of the lane, and the road potential field leads the vehicle to the right. As a result, the vehicle moves slightly to the left to pass the obstacle while it stays in the lane. At the time that the vehicle passes the obstacle, the lateral distance between the boundary of the obstacle and that of the vehicle is around $0.6m$ for both Scenarios 4 and 5. After the vehicle passes the obstacle, the road potential field leads the vehicle back to the lane center. Moreover, the vehicle speed does not change noticeably in any of the cases, as expected. It is notable that the obstacle of Scenario 4 is static, so its potential field is sharper than the PFs corresponding to the moving obstacle of Scenarios 1-3, which allows the vehicle to pass it on the side with a smaller margin.

Scenarios 6 and 7 are where there is a crossable or non-crossable obstacle on the current lane of the vehicle and there is not enough lateral space to pass the obstacle on the side. The simulation results of these scenarios are shown in Fig. 3.14. As the results show, the potential field of the non-crossable obstacle leads the vehicle to stop behind the obstacle. The crossable obstacle, however, is crossed while the vehicle does not change its speed considerably, which reflects the appropriate choice of the crossable obstacle PF. Moreover, for both cases, the vehicle does not move noticeably in the lateral direction.

3.7 Summary

In this chapter, a motion planning MPC was introduced that utilizes the potential field obstacle avoidance method for prioritizing obstacles. A vehicle dynamics model along with actuator and tire constraints were presented. Different PFs were introduced for different obstacles and road structures based on their characteristics. A motion planning MPC was presented with the vehicle model as its model and actuator and tire constraints as its constraints so that it predicts the vehicle behavior appropriately and generates feasible maneuvers. The PFs were included in the MPC's objective function for observing road regulations and obstacle avoidance based on the obstacle's priorities.

The MPC problem was nonlinear and to reduce the computational time, the problem was approximated by a quadratic convex problem. The calculation time and the performance of the nonlinear and quadratic problems were compared by simulation. The results showed that although the approximation can cause errors in the result of the quadratic problem, the performance of the quadratic problem was acceptable with a fraction of time



(a)

(b)

(c)

(d)

(e)

Figure 3.13: Scenarios 4 and 5, a) vehicles path and obstacles position, blue: vehicle of Scenario 4, purple: vehicle of Scenario 5, red: obstacle b) longitudinal force command and vehicle speed in Scenario 4, c) Steering angle command and lateral acceleration in Scenario 4, d) longitudinal force command and vehicle speed in Scenario 5, e) steering angle command and lateral acceleration in Scenario 5.

needed to solve the nonlinear problem.

Some complex test scenarios were defined to evaluate the performance of the proposed motion planning MPC. The simulations used high fidelity vehicle models in CarSim, but the vehicle model of the MPC was a linear bicycle model. The results showed the capability of the introduced motion planning method in performing the appropriate maneuvers in

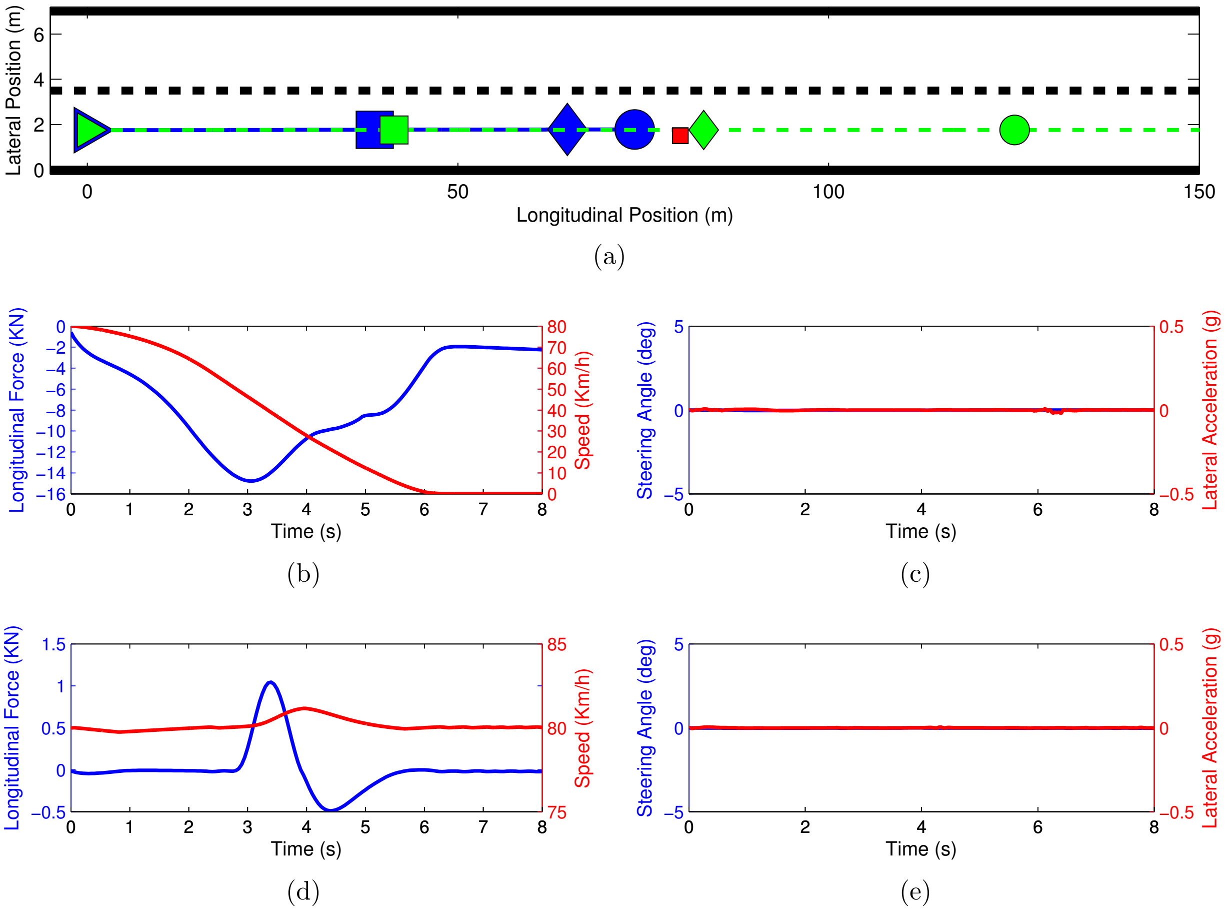


Figure 3.14: Scenarios 6 and 7, a) vehicles path and obstacles position, blue: vehicle of Scenario 6, purple: vehicle of Scenario 7, red: obstacle b) longitudinal force command and vehicle speed in Scenario 6, c) steering angle command and lateral acceleration in Scenario 6, d) longitudinal force command and vehicle speed in Scenario 7, e) steering angle command and lateral acceleration in Scenario 7.

complicated scenarios. When a lane change is commanded from the behavior planning module, the vehicle does not change its lane unless it is safe to do so. The vehicle merges in between two vehicles if there is enough space between them and it is safe to merge. If the current lane is ending, and a lane change is not safe, the vehicle reduces its speed or even stops before the lane ends, and changes its lane only when it is safe to do so. If a vehicle is approaching the vehicle from the side carelessly, the vehicle makes space for it as much as possible while staying on the road. For all these complicated scenarios, potential fields keep the vehicle away from the obstacles and road boundaries, and the tracking terms of the objective functions guide the vehicle toward their desired speed and lane. It is notable that the shape and intensity parameters of the developed potential field are tuned such that the vehicle behaves appropriately in different situations, which is a time consuming process. Besides, becoming trapped in the local minimum is one of the problems of potential field methods, which is addressed by tuning the potential field parameters in this project.

Moreover, the MPC was developed to prioritize the obstacles based on their avoidance necessity. Appropriate PFs were assigned to crossable and non-crossable obstacles to treat them based on their characteristics. The results showed the capability of the developed MPC in prioritizing the obstacles. The vehicle stops behinds a non-crossable obstacle if there is not enough space to pass the obstacle on its side. On the other hand, the vehicle crosses a crossable obstacle without a noticeable speed change if there is not enough space to pass the obstacle on its side. For both kinds of obstacles, the vehicle passes them on their sides, if possible.

Chapter 4

Prioritizing Obstacles using Lexicographic Optimization

4.1 Introduction

A motion planning MPC was introduced in Chapter 3 that avoids obstacles by including a potential field to its cost function. The MPC prioritizes the crossable and non-crossable obstacles through assigning them PFs corresponding to their avoidance necessity. In this chapter, the MPC developed in the previous chapter is modified to apply priority on non-crossable obstacles, which have a wide range of crash costs. This chapter is mainly developed based on the work of Rasekhipour et al. [87].

Obstacle avoidance constraints are generated in this chapter for the non-crossable obstacles, and are included in the MPC problem. As mentioned in Chapter 2, the obstacle constraint set introduced in [58] is appropriate for most driving situations. However, the constraint set limits the vehicle to stop behind an obstacle in front of the vehicle while in some situations, stopping behind the obstacle is not feasible but passing the obstacle on its side is feasible. This method fails to avoid the obstacle in such a situation. In this chapter, in addition to this constraint set, two constraint sets are introduced: one for passing the obstacle on its right, and one for passing it on its left. Iterative Quadratic MPC (IQMPC) is introduced consisting of three iterations of the developed MPC problem; one iteration for each constraint set. IQMPC fixes the aforementioned problem by expanding the feasible set through including the other two constraint sets.

LO is applied on the MPC to prioritize non-crossable obstacles. It prioritizes the obstacles through prioritizing their corresponding constraints. It is notable that, in a motion planning problem, the vehicle model should remain valid so that the planned trajectories based on the model can be tracked by the vehicle. Therefore, the vehicle dynamics constraints have the highest priority order. Obstacle constraints are ordered based on the obstacles' crash costs after the vehicle constraints. LO applies the priority orders on IQMPC. Using this method, in a situation where avoiding all obstacles is not possible, the MPC finds the solution that preserves the vehicle dynamics capacities, and avoids the obstacles with the highest priority orders.

The rest of the chapter is organized as follows. First, the obstacle avoidance constraint sets are presented, and IQMPC is introduced for the constraint sets. Next, the LO approach for prioritizing constraints in an MPC problem is explained, and the LO-based motion planning MPC is introduced to prioritize the obstacles. A mixed integer MPC is also presented as a benchmark to evaluate the performance of IQMPC in obstacle avoidance. Then, simulation results are illustrated for some test scenarios to evaluate the performance of IQMPC in obstacle avoidance and prioritizing obstacles.

4.2 Obstacle constraints

Carvalho et al. [58] present obstacle constraints based on the signed distance of the vehicle and an obstacle:

$$s(X, Y) > 0. \tag{4.1}$$

Then, they linearize the constraint around the vehicle operating point, which makes the obstacle avoidance problem a convex problem.

This approach handles most of the driving situations. However, in a situation where an obstacle is in front of the vehicle, the corresponding constraint of the obstacle is vertical and keeps the vehicle behind the obstacle. An obstacle is defined to be in front of the vehicle if it is ahead of the vehicle along the road and its signed distance with the vehicle has no component in the Y-direction. There are situations where stopping behind an obstacle that is in front of the vehicle in not possible because of the velocity difference between the obstacle and the vehicle, but it is feasible for the vehicle to pass the obstacle by moving to its side. The approach presented in [58] cannot handle these situations. Therefore, in this section, constraints are presented for an obstacle in front of the vehicle so that the vehicle can also pass the obstacle on its side.

For driving on a road, there are three options for the vehicle when there are obstacles in front of the vehicle: to stop behind the obstacles, to pass the obstacles on their left side, and to pass them on their right side. Therefore, three sets of constraints are defined in this section for obstacles in front of the vehicle. The first set of constraints are the linearized constraints of (4.1), which generates the available area for the vehicle to stop behind the obstacle. The second set of constraints is defined so that the vehicle can move to the left side of the obstacles. A crash rectangle is defined for each obstacle as the locus of the vehicle when the signed distance of the vehicle and the obstacle is zero. It is notable that in this thesis, the vehicle and the obstacles are assumed to be of rectangular shapes parallel to the X-Y axes. The area outside the crash rectangle is the available area for the vehicle. However, the area is non-convex and should be convexified by linear constraints to be used in the quadratic MPC. A line passing through the upper left corner of the crash rectangle, (X_1, Y_2), is used as an approximation of the area for passing the obstacle on its left, as shown in Fig. 4.1:

$$(X - X_1) \leq (Y - Y_2) \cot \psi, \tag{4.2}$$

where ψ is the constraint angle, which is defined as the angle of the constraint line with the X-axis. As demonstrated in the figure, the angle is obtained by connecting the vehicle position to the upper left edge of the crash rectangle with a circular arc. The angle of the arc at the vehicle position equals the vehicle heading. Therefore, the arc represents a path for the vehicle if it goes on a circle. The angle of the arc at the rectangle's edge is set as the constraint angle. Using this geometry, the radius of the arc, R, and the constraint angle are:

$$R = \frac{(X_1 - X)^2 + (Y_2 - Y)^2}{2(Y_2 - Y)\cos\theta - 2(X_1 - X)\sin\theta}, \tag{4.3}$$

$$\psi = \cos^{-1}\left(\cos\theta - \frac{Y_2 - Y}{R}\right). \tag{4.4}$$

If the vehicle's heading angle is large enough that the vehicle can pass the obstacle by going straight, the constraint angle is set to the vehicle's heading angle. Furthermore, for the potential function of the obstacle, the signed distance is set to the signed distance of the vehicle and the constraint line instead of the signed distance of the vehicle and the obstacle. It is notable that the obstacle position is predicted at each prediction step by assuming that it continues its motion with the same longitudinal and lateral velocity as the current moment.

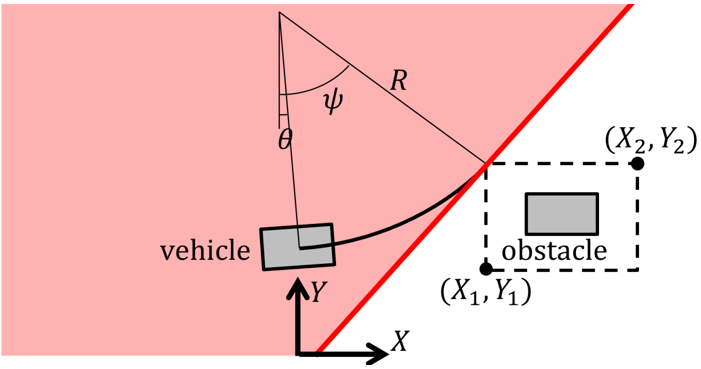


Figure 4.1: Linear Constraint Approximation.

It is also notable that if stopping behind the obstacle is not feasible, there are prediction steps such that the vehicle cannot be behind the obstacle even with the maximum brake acceleration. These steps satisfy the following equation:

$$-\frac{1}{2}a_{max}k^2 + u_0 k \geq X_1, \tag{4.5}$$

where, a_{max} is the maximum possible acceleration, u_0 is the current vehicle longitudinal velocity, and k is the prediction step. For these prediction steps, the vehicle should be on the side of the obstacle to avoid the obstacle, and the constraint is changed to:

$$Y \geq Y_2, \tag{4.6}$$

Moreover, because of the uncertainties of the road-tire friction, the vehicle's model, and the obstacle's motion the vehicle cannot be planned to pass the obstacle at zero distance. Therefore, a length of L_0 and a width of W_0 are added to each obstacle in calculating the constraints, which are set to $1.5m$ and $1m$, respectively. It also implies that the constraints should be soft constraints, since they might be violated because of the previously mentioned causes.

A similar approach is also used in calculation of the third set of constraints which corresponds whto passing the obstacles in front of the vehicle on their right side. Moreover, although potential functions are used to lead the vehicle within the lanes, constraints are implemented on the road boundaries to ensure that the vehicle stays within the boundaries. The constraints of all the obstacles and the road boundaries, except for the obstacle in front of the vehicle, are the same for the three constraint sets and are obtained from (4.1).

4.3 Iterative Quadratic MPC

IQMPC is introduced in this section for motion planning. It consists of three iterations of MPC, one for each constraint set presented in Section 4.2. By solving the three iterations, three solutions are found for stopping behind the obstacles in front of the vehicle, moving to their left, and moving to their right. The solution of IQMPC is the best one of the three solutions. The MPC problem for each iteration is presented in the following.

The MPC problem optimizes the vehicle dynamics behavior in its prediction horizon since the vehicle dynamics model presented in Section 3.3 is its model and constraints presented in that section are included in the MPC. The potential functions presented in Section 3.4 are also included in the objective function of the MPC to keep the vehicle at a safe distance from the obstacles. The motion planning MPC should also track the desired lane and speed; as explained in Section 3.5, the output vector $\boldsymbol{y}$ should track the desired value vector, $\boldsymbol{y}_{des}$. The MPC problem is:

$$\min_{\boldsymbol{u}_c,\boldsymbol{\varepsilon}} \sum_{k=1}^{N_p} \left(U^{t+k,t} + \|\boldsymbol{y}^{t+k,t} - \boldsymbol{y}_{des}^{t+k,t}\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_c^{t+k-1,t}\|_{\boldsymbol{R}}^2 + \|\boldsymbol{u}_c^{t+k-1,t} - \boldsymbol{u}_c^{t+k-2,t}\|_{\boldsymbol{S}}^2 \right) + \|\boldsymbol{\varepsilon}\|_{\boldsymbol{\lambda}}^1, \quad (4.7a)$$

$$s.t.(k = 1, \dots, N_p)$$

$$\boldsymbol{x}^{t+k,t} = \boldsymbol{A}_d \boldsymbol{x}^{t+k-1,t} + \boldsymbol{B}_d \boldsymbol{u}_c^{t+k-1,t}, \quad (4.7b)$$

$$\boldsymbol{y}^{t+k,t} = \boldsymbol{C}\boldsymbol{x}^{t+k,t} + \boldsymbol{D}\boldsymbol{u}_c^{t+k,t}, \quad (4.7c)$$

$$\boldsymbol{y}_s^{t+k,t} = \boldsymbol{C}_s\boldsymbol{x}^{t+k,t} + \boldsymbol{D}_s\boldsymbol{u}_c^{t+k,t}, \quad (4.7d)$$

$$\boldsymbol{y}_s^{t+k,t} \leq \boldsymbol{y}_{s-max}^{t+k,t} + \boldsymbol{\varepsilon}, \quad (4.7e)$$

$$\boldsymbol{\varepsilon} \geq 0, \quad (4.7f)$$

$$\boldsymbol{u}_c^{t+k,t} = \boldsymbol{u}_c^{t+k-1,t}, \quad k > N_c, \quad k \neq c_2 N_{rc} + N_c, \quad c_2 = 1, \dots, (N_p - N_c)/N_{rc}, \quad (4.7g)$$

$$\boldsymbol{u}_c^{t-1,t} = \boldsymbol{u}_c(t-1), \quad (4.7h)$$

$$\boldsymbol{x}^{t,t} = \boldsymbol{x}(t), \quad (4.7i)$$

The MPC variables are the input vector, $\boldsymbol{u}_c$ and the slack variable vector, $\boldsymbol{\varepsilon}$. The objective function of the MPC includes the quadratic potential field. The objective function also includes the quadratic terms of the output tracking, the input, and the change of the input and the first norm of the slack variable. (4.7b) predicts the states and (4.7c) calculates the tracking outputs. The vehicle and obstacle constraints are presented in (4.7e); the vector of constraint variables is bounded by the vector of constraint bounds. Also, the slack variable vector is added to the constraints to make them soft constraints. The corresponding slack variable of the input constraints (3.4)-(3.6) are set to zero since they cannot be violated physically. The tire capacity constraints of (3.10) cannot be violated physically either, but the slack variables are included in them to avoid possible infeasibility due to estimation errors. The constraint variables are predicted in (4.7d). The number of the control input is reduced as in (4.7g) to reduce the calculation time.

As explained, three iterations of the MPC problem presented in (4.7) are solved, one for each obstacle constraint set presented in Section 4.2. For each iteration, the obstacle constraints of (4.7e) are the corresponding obstacle constraint set of the iteration. The solution of IQMPC is the best solution of the three iterations. The best solution is the solution with the smallest slack variables, which is the solution that is more likely to avoid obstacles. The iteration with the best solution is called the optimal iteration. In the case that the slack variables are equal, a predefined iteration preference is used to find the optimal iteration. The iteration preference is determined based on the road regulation; e.g, staying on the lane is preferred to leaving the lane, and passing the obstacle on its left is preferred to passing it on its right. In this thesis, the iteration preference is: 1) stopping behind the obstacles in front of the vehicle, 2) passing the obstacles on their left sides, and 3) passing the obstacles on their right sides. The procedure of finding the optimal iteration is explained in detail in the next section.

4.4 Lexicographic Optimization (LO)

4.4.1 LO-based MPC

In this section, the LO-based MPC for solving the aforementioned multi-objective optimization problems is briefly presented [88]. Assume a multi-objective optimization problem:

$$\min_{\boldsymbol{\xi} \in \Xi} \boldsymbol{f}(\boldsymbol{\xi}), \tag{4.8a}$$

$$\boldsymbol{f}(\boldsymbol{\xi}) = \begin{bmatrix} f_1(\boldsymbol{\xi}) & f_2(\boldsymbol{\xi}) & \dots & f_q(\boldsymbol{\xi}) \end{bmatrix}^T, \tag{4.8b}$$

where $\Xi \subseteq \mathbb{R}^d$, and $f_i(\boldsymbol{\xi}) : \Xi \to \mathbb{R}$. The optimization problem has a minimum if:

$$\exists \boldsymbol{\xi}^* : \quad f_i(\boldsymbol{\xi}^*) = \min_{\boldsymbol{\xi} \in \Xi} f_i(\boldsymbol{\xi}) \quad \forall i, \tag{4.9}$$

The optimal point, $\boldsymbol{\xi}^*$, does not exist for general objective functions. Therefore, other approaches are used to find pseudo-optimal solutions for the problem.

One approach to finding the pseudo-optimal point is to solve the weighted sum of the objective functions. This method finds a compromised solution that does not optimize any of the objectives in general. It is the method used in the MPC problems introduced in the previous section. The LO method is another approach for finding the pseudo-optimal point. It is used when the objectives have priority over each other, i.e. optimizing one objective has priority over optimizing another objective. If the i^{th} objectives have priority over the j^{th} objective for $i < j$, $\boldsymbol{\xi}^*$ is the lexicographic minimum of (4.8) if:

$$f_1^* = \min_{\boldsymbol{\xi} \in \Xi} f_1(\boldsymbol{\xi}), \tag{4.10a}$$

$$f_i^* = \min_{\boldsymbol{\xi} \in \Xi} \left\{ f_i(\boldsymbol{\xi}) | f_j(\boldsymbol{\xi}) \leq f_j^*, j = 1, \ldots, i-1 \right\}, \quad \forall i \in \{2, \ldots, q\}, \tag{4.10b}$$

$$\boldsymbol{\xi}^* = \left\{ \boldsymbol{\xi} \in \Xi | f_j(\boldsymbol{\xi}) \leq f_j^*, j = 1, \ldots, q \right\}. \tag{4.10c}$$

As expressed in [72], the LO can be used for MPC problems with prioritized constraints and objectives. Holding constraints usually has priority over minimizing the regulation and tracking terms of the objective function. Moreover, holding some constraints may have priority over holding other constraints. In an MPC problem, when it is possible to violate a constraint, a slack is added to the constraint to avoid infeasibility due to constraint violation, and the violation is penalized by adding a term containing the slack value to the MPC objective function. Therefore, the objective function terms can be prioritized as follows: the penalizing terms of the objective function are prioritized based on the priority order of their corresponding constraint, and the tracking and regulating terms are prioritized based on their priority orders after the penalizing terms.

4.4.2 Obstacles Priority Order

The main focus of this thesis is to develop a platform to prioritize the obstacles based on injuries and damages they may cause. In Chapter 3, obstacles were categorized as crossable and non-crossable obstacles, and each category was assigned an appropriate potential function to be treated according to its avoidance necessity. Crossing a crossable obstacle has no crash cost. Crossable obstacles were modeled by exponential functions that penalize approaching the obstacle but let the vehicle cross the obstacle. Failing to avoid a non-crossable obstacle, on the other hand, causes a crash cost. Non-crossable obstacles were modeled by hyperbolic functions that do not allow crossing the obstacles.

In this chapter, the non-crossable obstacles are prioritized since their crash costs vary widely. The method used here can consider any number of priority orders. The following priority orders are applied on the obstacles and road boundaries in this project: 1) occupied sidewalk, 2) pedestrian, 3) car, 4) rock, 5) empty sidewalk. The priority orders are determined based on the possible corresponding crash costs. For example, hitting a pedestrian could cause injury, hitting a car might only cause car damage, and going on an empty sidewalk most probably does not cause any damage or injury. Therefore, a pedestrian is superior to a car, and a car is superior to an empty sidewalk. Moreover, a crossable obstacle does not required avoidance, so it has the lowest obstacle priority order. Its potential function included in the objective function of the MPC generates a force repulsing the vehicle from the obstacle.

4.4.3 LO-based Motion Planning MPC

The MPC introduced in (4.7) has objective terms including the potential functions, the tracking and regulation terms, the penalizing terms for vehicle constraint violations, and the penalizing terms for the obstacles constraint violations. If the priority of the objective terms is determined, the MPC can be solved using a lexicography method.

As explained, the tire force capacity cannot be physically violated, but it is considered as a soft constraint to avoid the possible infeasibility due to estimation errors. Holding this constraint is important so that the MPC's predictions based on the vehicle model are correct. Therefore, the penalizing term corresponding to this constraint has the first priority order. Violating the obstacles' constraints causes injury and damage cost, and their corresponding objectives are considered as objectives with the next priority orders. Their priority order are determined as explained in Section 4.4.2. The potential functions, tracking and regulation terms, and the penalizing term of the vehicle speed have the same priority, and the weighted sum of their objectives is considered as the objective with the lowest priority order. Therefore, assuming there are n obstacle priority orders, the priority orders of the objectives are:

1) $f_0(\boldsymbol{u}_c, \boldsymbol{\varepsilon}) = \|\boldsymbol{\varepsilon}\|^1_{\boldsymbol{\lambda}_0}$,

2) $f_1(\boldsymbol{u}_c, \boldsymbol{\varepsilon}) = \|\boldsymbol{\varepsilon}\|^1_{\boldsymbol{\lambda}_1}$,

$\vdots$

n+1) $f_n(\boldsymbol{u}_c, \boldsymbol{\varepsilon}) = \|\boldsymbol{\varepsilon}\|^1_{\boldsymbol{\lambda}_n}$,

n+2) $f_{n+1}(\boldsymbol{u}_c, \boldsymbol{\varepsilon}) = \sum_{k=1}^{N_p} \left(U^{t+k,t} + \| \boldsymbol{y}^{t+k,t} - \boldsymbol{y}_{des}^{t+k,t} \|_{\boldsymbol{Q}}^2 + \| \boldsymbol{u}_c^{t+k-1,t} \|_{\boldsymbol{R}}^2 + \| \boldsymbol{u}_c^{t+k-1,t} - \boldsymbol{u}_c^{t+k-2,t} \|_{\boldsymbol{S}}^2 \right) +$
$\| \boldsymbol{\varepsilon} \|_{\boldsymbol{\lambda}_{n+1}}^1$,

where the terms of $\boldsymbol{\lambda}_0$ corresponding to the tire capacities, the terms of $\boldsymbol{\lambda}_l$ corresponding to the obstacles with the l^{th} priority order, and the terms of $\boldsymbol{\lambda}_{n+1}$ corresponding to the speed constraint set to 1, and the other terms are set to 0. The MPC iterations of IQMPC can be solved with LO using these priority orders on the objective functions. Using this method, the tire capacities are held as much as possible as the first priority order so that the controller has a correct prediction. Next, obstacles with the first priority are avoided as much as possible, while it is assured that the prediction model is as correct as possible. Then, obstacles with the second priority are avoided as much as possible, while the prediction model is as correct as possible and the first priority obstacles are avoided as much as possible. The same procedure is applied for the rest of the obstacle priorities. Finally, the vehicle finds an appropriate trajectory based on the potential functions, the tracking and regulation terms, and speed violation penalizing term, while the vehicle model is as correct as possible and the non-crossable obstacles are avoided as much as possible.

Solving an LO problem increases the calculation time, as each MPC is split to $n + 2$ optimization problems. IQMPC has three MPC iterations, and LO is applied on each iteration. The first priority objective is the same for the three iterations. For the obstacle priority objectives, each of the n optimization problems should be solved three times to find the best iteration. Then, the quadratic problem of $f_{n+1}(\boldsymbol{u}_c, \boldsymbol{\varepsilon})$ is solved for the best iteration to find the final answer. Therefore, the motion planning problem, at the worst case, needs to solve $3n + 1$ linear optimization problems and 1 quadratic optimization problem.

The optimal iteration is determined after solving the n obstacle priority problems for the three iterations. It is determined based on the objective values and the iteration preferences:

$$\mathcal{P}_0 = \{1, 2, 3\}, \tag{4.11a}$$

$$\mathcal{P}_l = \arg\min_{p \in \mathcal{P}_{l-1}} f_l^p(\boldsymbol{u}_c, \boldsymbol{\varepsilon}), \quad l = 1, \ldots, n, \tag{4.11b}$$

$$p^* = \min_{p \in \mathcal{P}_n} p, \tag{4.11c}$$

where $f_l^p(\boldsymbol{u}_c, \boldsymbol{\varepsilon})$ denotes the objective function corresponding to the obstacle with the l^{th} priority order at the p^{th} iteration. As mentioned, there are three iterations with the preference from 1 to 3. First, the objective values are used in descending order of priority to determine the best iterations in (4.11b). Then, in (4.11c), the iteration preferences determine the optimal iteration, p^*, among the best iterations determined by (4.11b).

4.5 Mixed Integer MPC

A Mixed Integer MPC called MIMPC is introduced in this section, which uses mixed integer constraints to generate a complete non-convex obstacle-free area. It is used as a benchmark to evaluate the obstacle avoidance performance of IQMPC. MIMPC is similar to the MPC presented in (4.7) for IQMPC, with the difference that it utilizes mixed integer obstacle constraints to generate the obstacle-free area. For each obstacle, the area outside the crash rectangle is the obstacle-free area. The area outside of a rectangle can be constructed by the union of four constraints [89]:

$$\begin{aligned} X &\leq X_1 \quad \textit{or} \\ X &\geq X_2 \quad \textit{or} \\ Y &\leq Y_1 \quad \textit{or} \\ Y &\geq Y_2, \end{aligned} \tag{4.12}$$

where (X_1, Y_1) and (X_2, Y_2) denote the positions of the lower left and upper right edges of the rectangle, respectively. The above "*or*" constraints can be written as "*and*" constraints using mixed integer constraints [89]:

$$\begin{aligned} X &\leq X_1 + M\tau_1 \quad && \textit{and} \\ -X &\leq -X_2 + M\tau_2 && \textit{and} \\ Y &\leq Y_1 + M\tau_3 && \textit{and} \\ -Y &\leq -Y_2 + M\tau_4 && \textit{and} \\ \textstyle\sum_{i=1}^{4} \tau_i &\leq 3, && \end{aligned} \tag{4.13}$$

where M is the upper bound of the distance between the vehicle and the obstacle and τ_i, for $i = 1, ..., 4$, is a binary variable.

MIMPC utilizes the mixed integer obstacle constraints presented in (4.13) for obstacle avoidance. Except for the obstacle constraints, the optimization problem of MIMPC is similar to that of IQMPC presented in (4.7). Therefore, the mixed integer quadratic optimization problem of MIMPC is:

$$\min_{\boldsymbol{u}_c, \boldsymbol{\varepsilon}, \boldsymbol{\tau}} \sum_{k=1}^{N_p} \left(U^{t+k,t} + \| \boldsymbol{y}^{t+k,t} - \boldsymbol{y}_{des}^{t+k,t} \|_{\boldsymbol{Q}}^2 + \| \boldsymbol{u}_c^{t+k-1,t} \|_{\boldsymbol{R}}^2 + \| \boldsymbol{u}_c^{t+k-1,t} - \boldsymbol{u}_c^{t+k-2,t} \|_{\boldsymbol{S}}^2 \right) + \| \boldsymbol{\varepsilon} \|_{\boldsymbol{\lambda}}^1, \tag{4.14a}$$

$$s.t.(k = 1, \dots, N_p)$$

$$\boldsymbol{x}^{t+k,t} = \boldsymbol{A}_d \boldsymbol{x}^{t+k-1,t} + \boldsymbol{B}_d \boldsymbol{u}_c^{t+k-1,t}, \tag{4.14b}$$

$$\boldsymbol{y}^{t+k,t} = \boldsymbol{C} \boldsymbol{x}^{t+k,t} + \boldsymbol{D} \boldsymbol{u}_c^{t+k,t}, \tag{4.14c}$$

$$\boldsymbol{y}_s^{t+k,t} = \boldsymbol{C}_s \boldsymbol{x}^{t+k,t} + \boldsymbol{D}_s \boldsymbol{u}_c^{t+k,t} + \boldsymbol{E}_s \boldsymbol{\tau}^{t+k,t}, \tag{4.14d}$$

$$\boldsymbol{y}_s^{t+k,t} \leq \boldsymbol{y}_{s-max}^{t+k,t} + \boldsymbol{\varepsilon}, \tag{4.14e}$$

$$\boldsymbol{\varepsilon} \geq 0, \tag{4.14f}$$

$$\boldsymbol{\tau} \in \{0,1\}^n, \tag{4.14g}$$

$$\boldsymbol{u}_c^{t+k,t} = \boldsymbol{u}_c^{t+k-1,t}, \quad k > N_c, \quad k \neq c_2 N_{rc} + N_c, \quad c_2 = 1, \ldots, (N_p - N_c)/N_{rc}, \tag{4.14h}$$

$$\boldsymbol{u}_c^{t-1,t} = \boldsymbol{u}_c(t-1), \tag{4.14i}$$

$$\boldsymbol{x}^{t,t} = \boldsymbol{x}(t), \tag{4.14j}$$

In addition to the control inputs and the slack variables, the binary variables are also optimization variables of MIMPC; $\boldsymbol{\tau}$ is the vector of binary variables. The mixed integer obstacle constraints of (4.13) are included in constraint equations of (4.14d) where the binary variables are added to the equations through the binary matrix, $\boldsymbol{E}_s$.

4.6 Results

In this chapter, IQMPC and MIMPC are simulated on the CarSim vehicle model as explained in the previous chapter to evaluate the performance of IQMPC. The prediction horizon is 40, and the time step is $50ms$, which makes the prediction time $2s$. The calculation time of the solver for each of the quadratic MPC problems is less than $2ms$, which makes IQMPC problem solved in less than $34ms$. Therefore, IQMPC can be implemented in real time.

For IQMPC, at each time step, three obstacle constraints and three potential functions are generated for each obstacle at each prediction step. The obstacles' and vehicle's position at each prediction step are predicted as explained in Section 3.6.2. For each prediction time step, three obstacle constraints and three potential functions are generated for each obstacle based on the predicted obstacle position as explained in Section 4.2. Each potential function is approximated by a quadratic convex function around the predicted vehicle position. Then, the MPC problem of the tire constraint objective function is solved by QPOASES to find the optimal value of the objective function. Next, for each iteration of IQMPC, starting from the highest priority order, the quadratic programming problem of MPC is solved for each obstacle priority order using QPOASES to find the optimal value of the corresponding objective function. Each of the MPC problems is constrained such that the value of each of the objective functions with a higher priority order than that of the MPC problem is less than or equal its calculated optimal value. They are also constrained such that the tire objective function has a value less than or equal its optimal value. This procedure is performed for the three iterations, where for each iteration, its corresponding constraints and potential functions are used. Next, the iteration that avoids the obstacles with the highest priority orders better is selected, as explained in Section 4.4.3. The quadratic MPC problem corresponding to that iteration is solved by QPOASES to find the vehicle inputs. The MPC is constrained such that the value of each of the tire and obstacle objective functions is less than or equal its corresponding optimal value. Next, the CarSim vehicle model is simulated for one step. The input of the CarSim vehicle model is the calculated vehicle inputs, and its outputs are the vehicle states, which are used to as the vehicle current states in the calculations of the next step. A similar procedure is performed for MIMPC with the difference that one iteration is solved using the mixed-integer constraints as explained in Section 4.5.

4.6.1 Scenario 1: Passing an Obstacle

As mentioned in Section 4.3, IQMPC has the ability to pass an obstacle on its side when stopping behind the obstacle is not feasible, but passing it on its side is feasible. In this section, a scenario is designed to show the capability of IQMPC in this situation. The vehicle is assumed to move on the first lane of a two-lane road with a speed of $60Km/h$. The desired speed is $60Km/h$ and the desired lane is the first lane. It is assumed that the road has a $1.5m$ margin on each side. There are sidewalks on the road boundaries. The right sidewalk is occupied with pedestrians, and the left sidewalk is empty. A static obstacle appears in the middle of the first lane at a distance of $18m$ ahead of the vehicle. The obstacle is square-shaped with the sides of $1m$, representing a pedestrian. The priority orders assigned to the obstacles are presented in Table 4.1. The stopping distance of the vehicle at this speed is $16.38m$. Considering the vehicle length of $5m$, the vehicle cannot stop behind the obstacle since it only has $15m$ obstacle-free space ahead of the

Table 4.1: Priority Orders of Obstacles in Different Scenarios

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Right Road Boundary	1	1	4	1	1
Left Road Boundary	3	1	1	1	1
Static Obstacle	2	2	2	3	3
Moving Obstacle	–	3	3	2	2

vehicle. Therefore, if the MPC problem is solved only using the convexified signed distance constraints presented in [58], it cannot avoid the obstacle. But, IQMPC can avoid the obstacle.

Figures 4.2-4.5 show the maneuver performed by the autonomous vehicle IQMPC, MIMPC, ad MPC with signed distance constraints similar to [58]. Figure 4.2 illustrates the path of the vehicle. In this figure, the vehicle at the initial position is shown by a blue rectangle, and the obstacle is shown by a red square. The path of the vehicle is demonstrated by the purple line for MPC with signed distance constraints, by the blue line for IQMPC, and by the green line for MIMPC. The vehicle position is marked on the path by small squares at each $1.5s$. A contact rectangle is drawn around the obstacle with dashed lines. The rectangle represents the area that if the vehicle position is located within it, the vehicle and the obstacle are in contact. Therefore, a path that does not enter the contact rectangle is obstacle-free. It can be seen that, as expected, the motion planning method with the signed distance constraints similar to [58] cannot avoid the obstacle, but IQMPC avoids the obstacle.

The procedure of choosing the optimal iteration and calculating the optimal solution is explained for IQMPC at the start of Scenario 1. First, f_0, the objective function of the tire capacities, is optimized. f_0 is the same for the three iterations, and therefore, is solved once. $f_0 = 0$ at this moment, meaning that the tire capacities are not violated. Then, f_1^p is optimized for $p = 1, 2, 3$ while keeping $f_0 \leq 0$, which results in $\begin{bmatrix} f_1^1 & f_1^2 & f_1^3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. Therefore, based on (4.11b), $\mathcal{P}_1 = \{1, 2, 3\}$. This means that the obstacles with the first priority order can be avoided for the three iterations while tire capacity constraints are hold. Then, f_2^p is optimized for $p = 1, 2, 3$ while keeping $f_0 \leq 0$ and $f_1^p \leq 0$, which

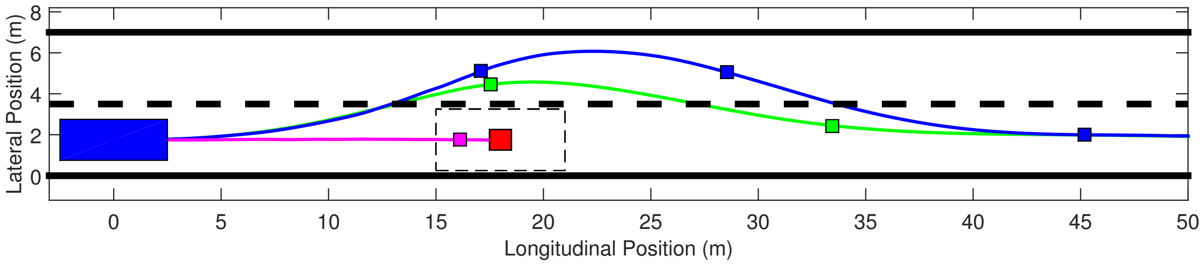
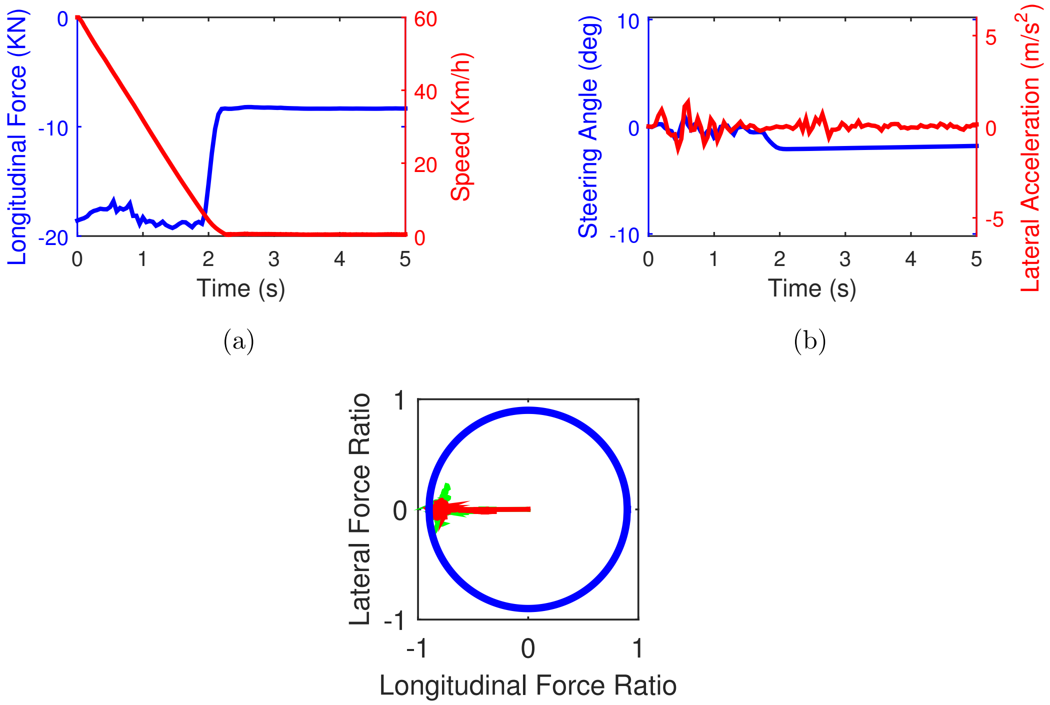


Figure 4.2: Scenario 1 vehicle's and obstacle's path- blue: vehicle for IQMPC- green: vehicle for MIMPC- red: static obstacle.

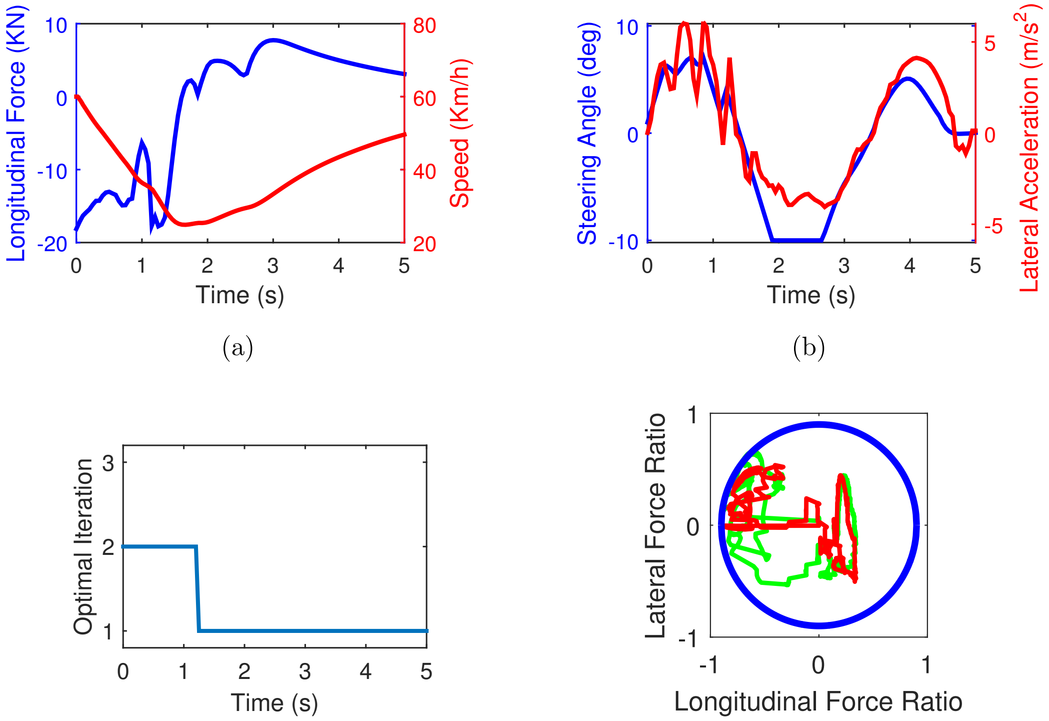


(a)

(b)

(c)

Figure 4.3: Scenario 1 simulation results for MPC with signed distance constraints: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) tire friction circle.



(a)

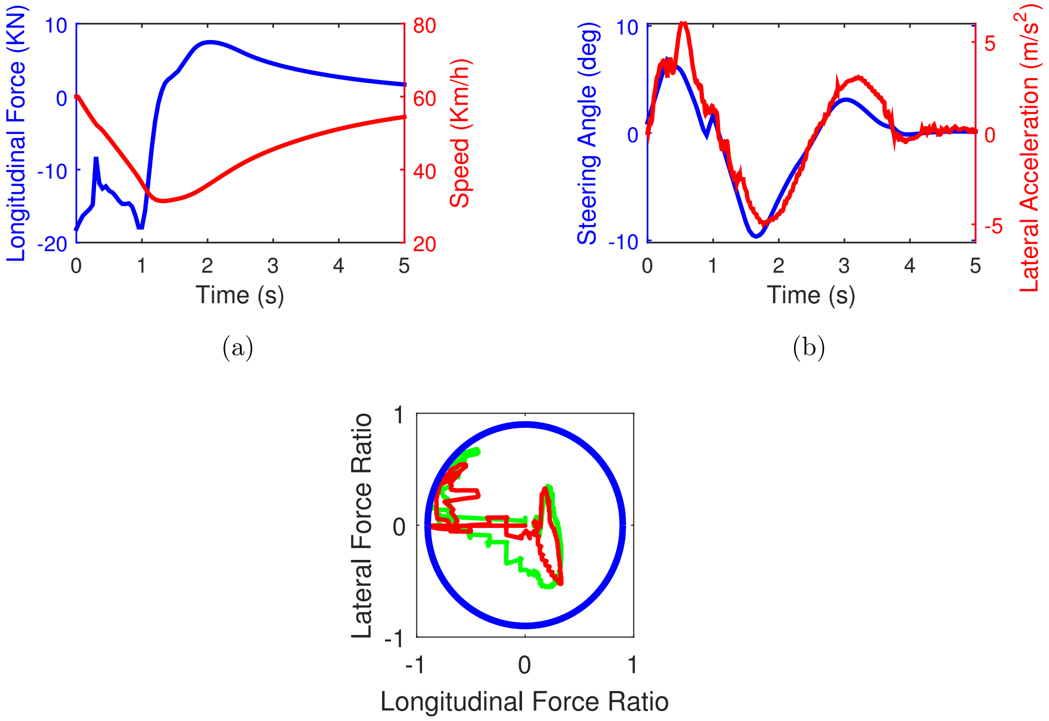
(b)

(c)

(d)

Figure 4.4: Scenario 1 simulation results for IQMPC: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) optimal iteration over time, d) tire friction circle.

results in $\begin{bmatrix} f_2^1 & f_2^2 & f_2^3 \end{bmatrix} = \begin{bmatrix} 2.40 & 0.28 & 4.78 \end{bmatrix}$. Therefore, based on (4.11b), $\mathcal{P}_2 = \{2\}$. This means that the best iteration for avoiding the obstacle with the second priority order is Iteration 2. It is clear that since $\mathcal{P}_2$ has only one value of 2, based on (4.11b), $\mathcal{P}_3 = \{2\}$. Therefore, based on (4.11c), $p* = 2$, meaning that Iteration 2 is the optimal iteration for this maneuver. It is notable that the optimal iteration is determined based on f_2^p for $p = 1, 2, 3$, and there is no need to solve f_l^p for $l > 2$ and $p \neq 2$. Hence, to find the best solution of the optimization problem, f_3^2 is optimized while keeping $f_0 \leq 0$, $f_1^2 \leq 0$, and $f_2^2 \leq 0.28$, which results in $f_3^2 = 5.06$. Then, f_4, which is a quadratic objective function, is optimized while keeping $f_0 \leq 0$, $f_1^2 \leq 0$, $f_2^2 \leq 0.28$, and $f_3^2 \leq 5.06$, which calculates the optimal solution of the motion planning problem. It is also notable that if



(a)

(b)

(c)

Figure 4.5: Scenario 1 simulation results for MIMPC: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) tire friction circle.

(4.11b) does not determine the optimal iteration, (4.11c) does so based on the iteration preferences. For example, at time $t = 2s$, $f_l^p = 0$ for $l = 1, 2, 3$ and $p = 1, 2, 3$, which means that $\mathcal{P}_3 = \{1, 2, 3\}$. Therefore, based on (4.11c), the optimal iteration is calculated to be Iteration 1.

As seen, both IQMPC and MIMPC can pass the obstacle. The calculated optimal iteration of IQMPC is shown in Fig. 4.4c. While the vehicle is behind the obstacle, Iteration 2, which corresponds to passing the obstacle on the left, is the optimal iteration. Then, Iteration 1 is the optimal iteration as it has the first iteration preference.

Figures 4.4 and 4.5 also show that the vehicle applies large steering and braking values simultaneously to perform the maneuver, using a large amount of the tires' and actuators' capacities. The tire friction circles are demonstrated in Fig. 4.4d and 4.5c, where the blue circles are the tire capacity circles on a dry road with a friction of 0.9. The horizontal axis is the ratio of the total longitudinal tire force to the total force capacity. The vertical axis is the ratio of the front/rear lateral tire force to the front/rear force capacity. The front/rear force capacity is calculated as the front/rear vertical tire force times the friction, and the total force capacity is the sum of front and rear force capacities. The green line corresponds to the front tires and the red line corresponds to the rear tires. As it can be seen, both the front and rear tire forces can be on the edge of the tire friction circle, i.e. the motion planning MPCs are capable of utilizing the tire capacities in their planning. The steering angle capacity is also reached in performing the maneuvers (Fig. 4.4b). Since the vehicle model and tire constraints are considered in the MPCs, the vehicle can perform the maneuvers, utilizing the tires' and actuators' capacities. Moreover, an oscillation with a frequency of almost $2Hz$ is noticed in the lateral acceleration when the maneuver is harsh. The oscillation happens because of the vehicle's pitch and roll dynamics, which causes the vertical forces on the wheels to oscillate.

IQMPC avoids the obstacle, but compared to MIMPC, it performs a harsher maneuver. It is because the process of convexification reduces the available area, which makes the maneuver harsher. However, IQMPC takes around $9.1ms$ by average to be solved at each step time, while MIMPC takes around $19.2s$. The scenario is repeated when the initial distance of the obstacle and the vehicle is reduced by increments of $1m$. For both controllers, the smallest distance such that they can avoid the obstacle is $15m$. Therefore, although IQMPC does not perform as well as MIMPC, it can avoid the obstacle with much less calculation time. The next scenarios are simulated only for IQMPC.

4.6.2 Scenarios 2-5: Obstacle Priority

Scenarios 2-5 are designed to observe the performance of the presented motion planning system in implementing priority on obstacles. These scenarios are similar to Scenario 1, except that, in these scenarios, there is also a moving obstacle. The obstacle is moving with a velocity of $25Km/h$ in the middle of the second lane, and is initially $10m$ ahead of the vehicle in X-direction. The obstacle is of the same size of the vehicle, which is $5m$ in

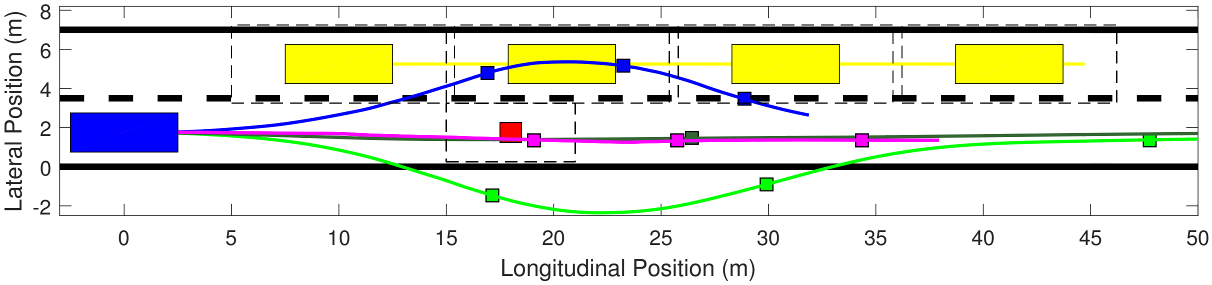


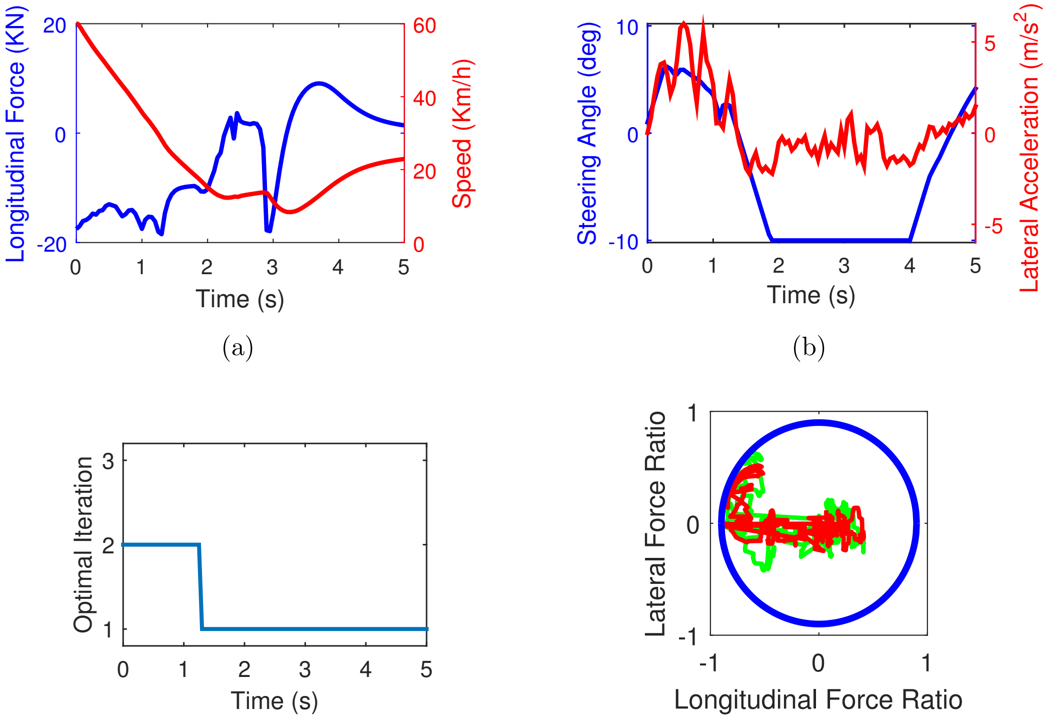
Figure 4.6: Scenario 2-5 vehicle's and obstacle's path- blue: vehicle in Scenario 2- green: vehicle in Scenario 3- purple: vehicle in Scenario 4- dark green: vehicle in Scenario 5- red: static obstacle- yellow: moving obstacle.

length and $2m$ in width. The situation is designed such that the vehicle cannot avoid the both obstacles while it stays within the road boundaries.

Four scenarios are defined for this situation with different obstacle priority orders to show how the priority implementation changes the vehicle's maneuver. In Scenario 2, sidewalks occupied with pedestrians exist over the road boundaries on both sides, the static obstacle is a pedestrian, and the moving obstacle is a car. Scenario 3 is similar to Scenario 2 except that the right sidewalk is empty. Scenario 4 is similar to Scenario 2 except that the static obstacle is a rock. Scenario 5 is similar to Scenario 2 except that the static obstacle is a bump, which is categorized as a crossable obstacle. The priority orders of the obstacles for each scenario are presented in Table 4.1.

The simulation results for these three scenarios are shown in Fig. 4.6-4.10. The paths of Scenarios 2, 3, 4, and 5 are shown by the blue, green, purple, and dark green lines, respectively, in Fig. 4.6. The vehicle's position along each path is marked by a square every $1.5s$. The static obstacle is shown with a red square, and the dashed rectangle around it is the contact rectangle similar to Fig. 4.2. The moving obstacle is also illustrated by yellow rectangles at the initial position and at its position every $1.5s$, and its path is demonstrated by a yellow line. Contact rectangles are also drawn around the moving obstacle at each occasion. If the marker on the vehicle path is outside the contact rectangle at the corresponding occasion, the obstacle is avoided at that occasion.

In Scenario 2, the vehicle remains on the road and avoids the static obstacle, but it



(a)

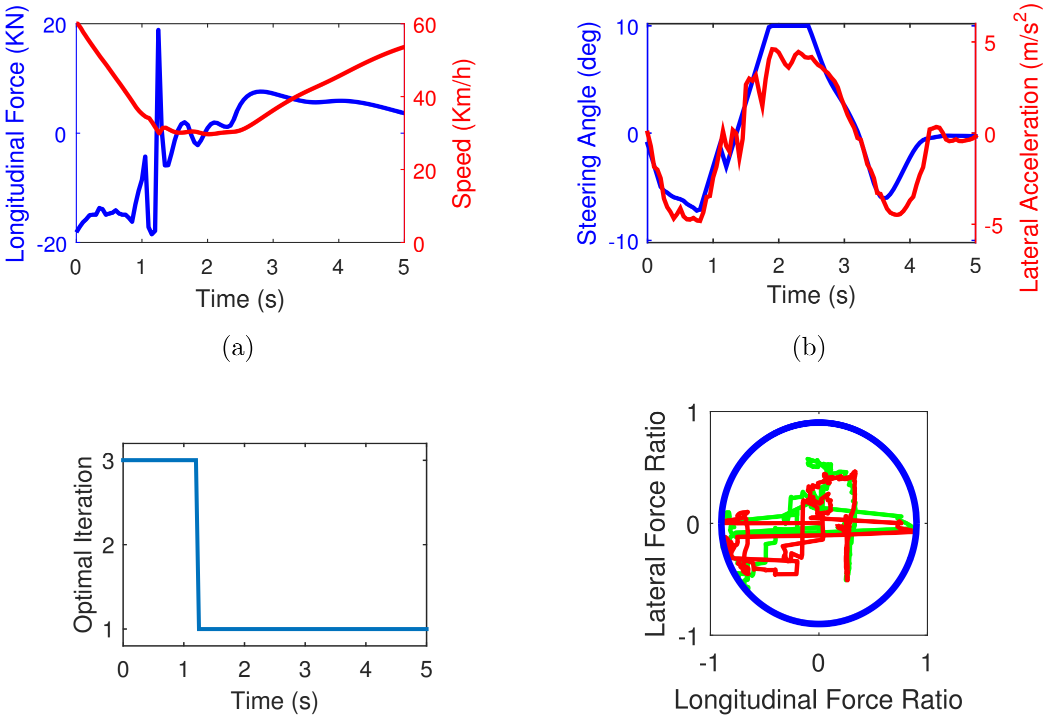
(b)

(c)

(d)

Figure 4.7: Scenario 2 simulation results: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) optimal iteration over time, d) tire friction circle.

hits the moving obstacle, which has the lowest priority order. The controller calculates the optimal iteration to be Iteration 2, which corresponds to moving to the left of the obstacle in front of the vehicle. It is notable that, although the vehicle hits the obstacle, it tries to avoid the obstacle by reducing its speed. In Scenario 3, the vehicle avoids the static and moving obstacles, but it crosses the right road boundary, which has the lowest priority order. The optimal iteration is calculated to be Iteration 3, which corresponds to moving to the right of the obstacle in front of the vehicle. The vehicle moves back to its lane as soon as it passes the static obstacle. In Scenario 4, the vehicle avoids the moving obstacle and stays on the road, but it hits the static obstacle, which has the lowest priority order. The optimal iteration is Iteration 1, which corresponds to stopping behind the obstacle in



(a)

(b)

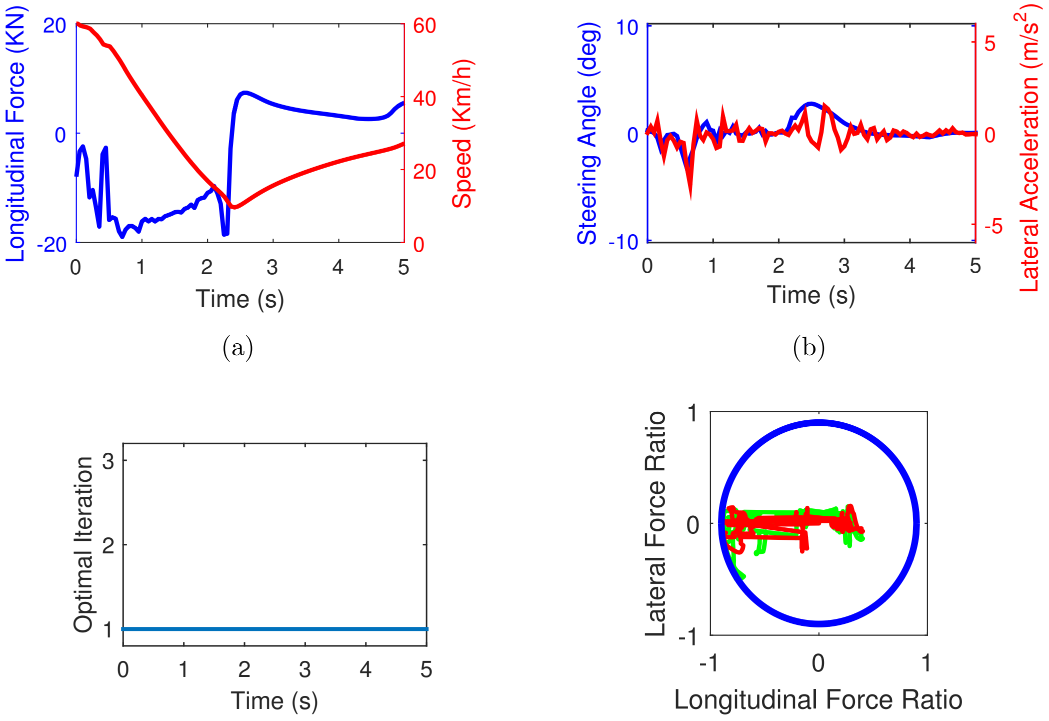
(c)

(d)

Figure 4.8: Scenario 3 simulation results: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) optimal iteration over time, d) tire friction circle.

front of the vehicle. Although the vehicle hits the static obstacle, it tries to stop behind the obstacle by reducing its speed. In Scenario 5, the vehicle avoids the moving obstacle, stays on the road, and crosses the static obstacle, which has the lowest priority order. Since the static obstacle is crossable, the vehicle crosses it without changing its velocity noticeably.

In all the scenarios, the vehicle tries to avoid all the non-crossable obstacles and uses the tires' and actuators' capacities to do so. Since the vehicle constraints have the highest priority order, IQMPC plans the trajectory for the lowest violation of the vehicle limitations, and its prediction of the vehicle behavior remains valid. Therefore, it is capable of planning a trajectory performable by the vehicle. In all the scenarios, the vehicle is successful in avoiding the obstacles with the highest priority orders, and hits the obstacles



(a)

(b)

(c)

(d)

Figure 4.9: Scenario 4 simulation results: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) optimal iteration over time, d) tire friction circle.

with the lowest priority orders if avoiding them is not feasible.

4.7 Summary

In this chapter, IQMPC was introduced for motion planning, and the LO was applied on it to prioritize obstacles. A quadratic motion planning MPC misses some feasible trajectories because it uses just one linear obstacle constraint set. IQMPC was introduced to reduce the number of missed feasible trajectories by using three linear obstacle constraint sets instead of one. It uses the linear signed distance obstacle constraint set as well as two

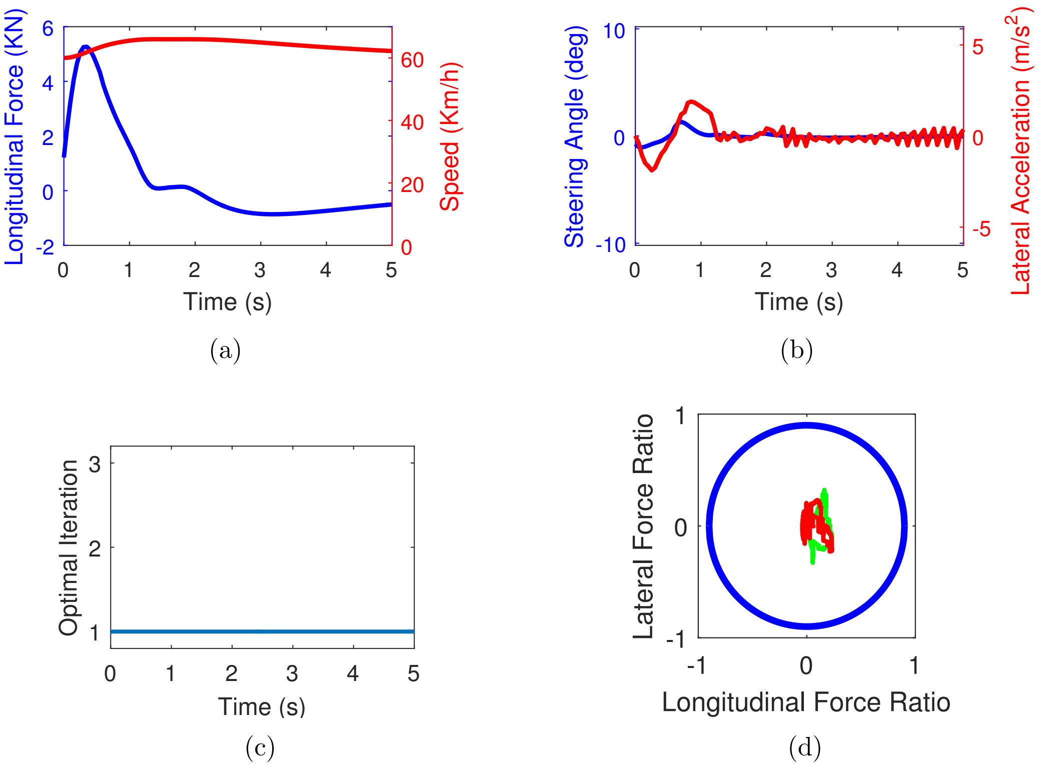


Figure 4.10: Scenario 5 simulation results: a) longitudinal force command and vehicle speed over time, b) steering angle command and lateral acceleration over time, c) optimal iteration over time, d) tire friction circle.

constraint sets allowing the vehicle to swerve to the right and left of an obstacle in front of it. IQMPC was simulated on a high fidelity CarSim vehicle model in some test scenarios. In Scenario 1, there is a static obstacle in front of the vehicle at a distance such that the vehicle cannot stop behind it, but can avoid it by moving to its side. A motion planning MPC using the linearized signed distance constraint set fails to avoid the obstacle in this situation. However, as the simulation results showed, the vehicle avoids the obstacle by moving to the obstacle's side if it uses IQMPC since it can use the union of the feasible trajectories of the three constraint sets.

The scenario was also simulated for MIMPC considering the obstacles with mixed integer constraints. Using this method, the vehicle can avoid the obstacle with a smoother maneuver compared to IQMPC. The reason is although IQMPC covers more feasible trajectories than a quadratic MPC, it does not cover all the feasible trajectories, while MIMPC can consider all the obstacle-free area. However, IQMPC solves the problem noticeably faster than MIMPC. Moreover, the results showed that the smallest distance between the vehicle and the obstacle for which the vehicle can avoid the obstacle is the same for both methods. Therefore, although IQMPC does not find a smoother trajectory of the vehicle, it has the advantage of solving the problem fast and being implementable in real time over MIMPC.

The LO was applied on IQMPC to prioritize obstacles. Scenarios 2-5 were designed to show the performance of IQMPC in prioritizing obstacles. There is a static obstacle and a moving obstacle on the road, such that both the obstacles cannot be avoided while the vehicle stays on the road. Different priorities are designed for the obstacles and the road margins. The results of Scenarios 2-5 show that the vehicle avoids the obstacles with the highest priority orders and hits the obstacle with the lowest priority order. For Scenario 2, there are pedestrians on the road sidewalks, the static obstacle is a pedestrian and the moving obstacle is a car. In this scenario, the vehicle avoids the pedestrians, stays on the road, but hits the car. Scenario 3 is the same as Scenario 2 except that there is no pedestrian on the right road margin. In this scenario, the vehicle avoids the pedestrian and the car by moving to the right sidewalk. Scenario 4 is the same as Scenario 2 except that the static obstacle is a rock. In this scenario, the vehicle avoids the car, stays on the road, and hits the rock while reducing its velocity. Scenario 5 is the same as Scenario 2 except that the static obstacle is a bump, which is a crossable obstacle. In this scenario, the vehicle avoids the car, stays on the road, and crosses the bump without changing its velocity noticeably.

The results of Scenarios 2-5 show that with the presented method, the vehicle avoids the obstacles with the highest priority orders successfully. It utilizes the tires' and actuators' capacities to perform the best obstacle avoidance maneuver. Since the vehicle constraints have the highest priority, their violation is minimum, which keeps the vehicle model valid. Therefore, the planned trajectory is performable by the vehicle while the tires' and actuators' capacities can be reached. It is also notable that the method can be applied for any number of obstacle categories. Increasing the number of priorities increases the calculation time. However, possible non-crossable obstacles on a road can be categorized in a handful number of priority categories allowing for the real-time implementation of the proposed method.

Chapter 5

Experimental Results

5.1 Introduction

In this chapter, the motion planning MPC developed in Chapter 3 is implemented on a test autonomous vehicle platform. The motion planning MPC is modified to become compatible with the test vehicle software platform, and be used as the motion planning module of the software platform. The test platform has large delays, which causes oscillatory vehicle behavior. The delays are compensated by predicting the vehicle states and using the predicted states as the MPC's initial states. Experimental tests are performed on the test vehicle to validate the simulation results of the motion planning MPC and show that the MPC is implementable in real-time.

The rest of the chapter is organized as follows. First, the test vehicle and its equipments and modules are introduced. Next, the setup of the motion planning module for experimental tests is explained. The MPC inputs including vehicle states and virtual road and obstacles are obtained based on the data received from the other modules. Actuation mappings are calculated, and actuator inputs are generated based on the MPC outputs using the mappings. The platform delays are also compensated through a state predictor. Then, the experimental results of some test scenarios are presented to validate the simulation results of the motion planning MPC.

5.2 Test Vehicle

The experimental tests are performed on a BYD Tang test vehicle belonging to Intelligent Vehicle Research Center of Beijing Institute of Technology (Fig. 5.1). The vehicle is equipped with a Velodyne 32 LIDAR, two cameras, a GPS/INS navigation system, and two ARK-2000 on-board computers. The communications between the hardware sets are performed with TCP/IP protocol through EKI-2528PAI routers. The vehicle is also equipped with an active steering system, an electric motor, and a hydraulic brake system.

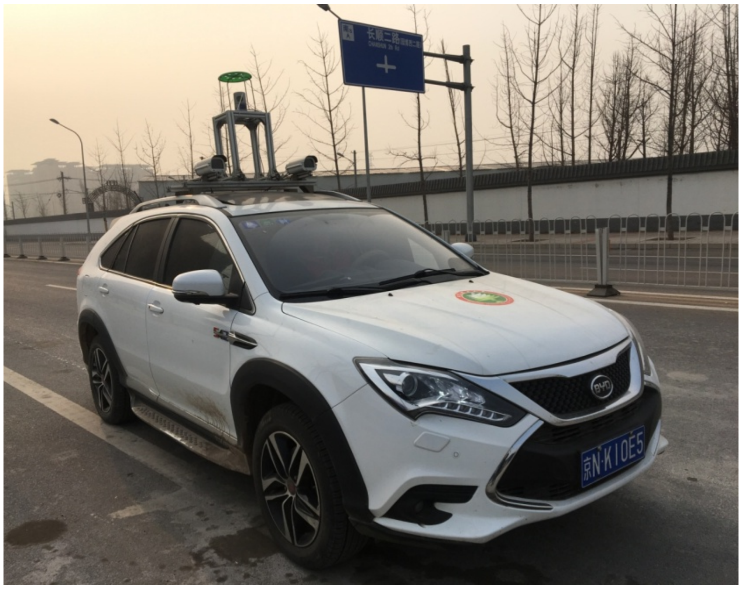


Figure 5.1: Test Vehicle.

The test vehicle software platform is developed on C++. The platform consists of a perception module, a route planner module, and a motion planning module. The perception module uses a Simultaneous Localization And Mapping (SLAM) method to generate an occupation map and localize the vehicle based on the LIDAR and GPS/INS data. The SLAM has a precision of $20cm$ and localizes the vehicle on a $20cm \times 20cm$ grid map. The route planner module generates a predefined path profile, e.g, a straight path or a circular path, for the vehicle. Since the platform is not capable of detecting lanes or road margins, the path profile generated by this module is assumed to be the road profile in the motion planning module. The motion planning MPC developed in Chapter 3 is utilized in the motion planning module of the platform.

5.3 Motion Planning Module Setup

The motion planning MPC developed in Chapter 3 should be adjusted for use in the test vehicle software platform. Since the platform is developed on C++, the MPC developed in MATLAB/Simulink is converted to C++. It is notable that since QPOASES is based on C++, it can be used in the platform to solve the quadratic programming problem of the MPC.

The motion planning MPC should also be modified based on its inputs and outputs from the software platform. The developed motion planning MPC receives the vehicle states, the road and obstacle data, and the desired velocity and lane and generates the driving commands. In this section, the adjustments required in the MPC for it to be compatible with the software platform are presented.

The experimental results of the test platform with the motion planning MPC show an oscillatory vehicle behavior because of the delays in the vehicle platform. The delays originate from the localization system, the motion planning system, and the actuation system. In this section, the delays are compensated through a state predictor to remove the oscillations.

5.3.1 Vehicle States

The motion planning MPC requires initial vehicle states, and receives them from the vehicle platform. The vehicle states include the longitudinal velocity, lateral velocity, yaw rate, position, and heading angle. The yaw rate and heading angle are received from the vehicle navigation system. The vehicle speed in the north and east direction are also received from the navigation system. The vehicle longitudinal velocity can be calculated as the vector sum of the vehicle speed in north and east directions. The vehicle lateral velocity cannot be obtained directly from the navigation system, so it needs to be estimated, which is out of the scheme of this work. This velocity is set to zero for these experiments. This approximation does not affect the predictions of the MPC significantly if no harsh lateral maneuver is performed.

The vehicle position is received from the perception module, which localizes the vehicle using a SLAM method. The module has a large calculation time, which is shown for a maneuver in Fig. 5.2. The calculation time is around 300ms on average, which causes an average delay of around 300ms on the position data. The delay is compensated by a state predictor as introduced in the following subsection.

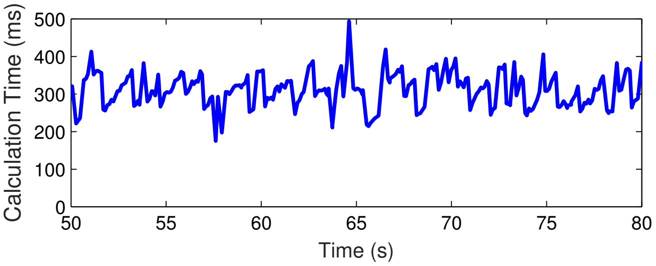


Figure 5.2: Calculation time of the perception module.

5.3.2 Road and Obstacles

The motion planning MPC requires road data including the lane widths, profile, and number, and the obstacles data including the position of the obstacles, their size, and their type. The obstacles, lanes, and road margins are not detected in the software platform, and detecting them is out of the scope of this thesis.

The software platform has a route planner module, which generates a virtual path profile. For the experimental tests, the path profile is assumed to be the current lane profile. Moreover, the other road data including the width and number of lanes, and the obstacle data are defined virtually. The virtual road and obstacle data, as well as the desired speed and lane, are defined in the motion planning module for each scenario.

5.3.3 Actuation System

The motion planning MPC generates the driving commands including the total longitudinal force and the front steering angle. These commands are applied on the test vehicle through its active steering system, electric motor, and hydraulic brake system. The active steering system receives the steering wheel angle, the electric motor receives the motor torque, and the hydraulics brake system receives the brake pressure. Therefore, the front steering angle command should be mapped onto the steering wheel angle, and the total longitudinal force command should be mapped onto the motor torque and the brake pressure.

For low velocities, the steady state yaw rate has the following relation with the front steering angle:

$$r = \frac{u}{l_f + l_r}\delta, \tag{5.1}$$

By knowing the steering wheel angle feedback and the vehicle yaw rate and speed for a low speed maneuver, the mapping from the front steering angle to the steering wheel angle can be calculated. The map is calculated to be a linear map with a proportional factor of 17.5. Figure 5.3 illustrates the steering wheel angle over time for an experimental test comparing the actual feedback value with the estimated value from (5.1) with the obtained map to show the accuracy of the map.

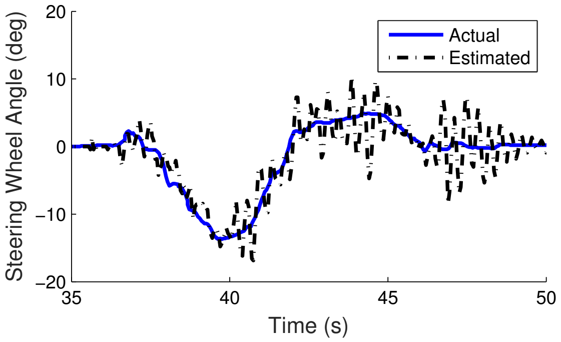


Figure 5.3: Steering wheel angle mapping.

The total longitudinal force can also be calculated by Newtons second law using the longitudinal acceleration:

$$F_{x_T} = ma_x, \tag{5.2}$$

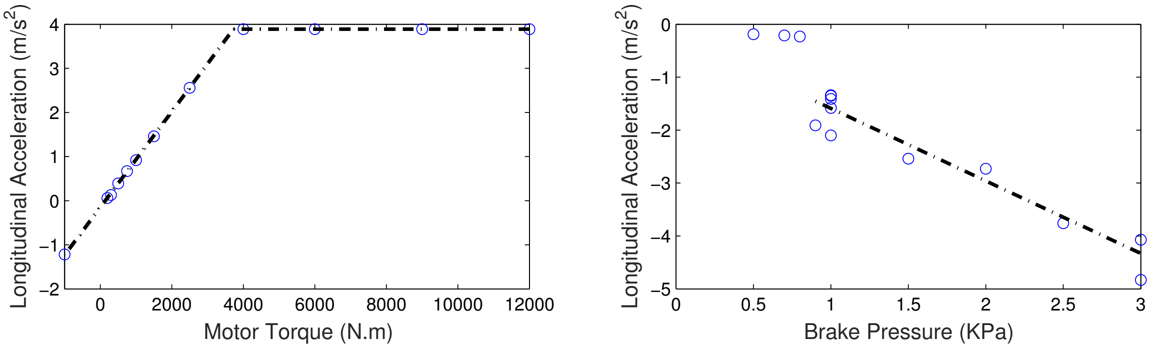
where a_x is the longitudinal acceleration. By knowing the brake pressure, motor torque command, and the vehicle longitudinal acceleration, the mappings from the total longitudinal force to the brake pressure and to the motor torque can be obtained. Figure 5.4 illustrates mappings from the longitudinal acceleration to the brake pressure and to the motor torque which can be transformed to the desired mappings by (5.2). In this figure, the circles demonstrate the cloud maps obtained from experimental results, and the lines represent the approximated map to be used in the motion planning module. As seen in Fig. 5.4a, the maximum longitudinal acceleration generated by the motor is $3.89m/s^2$. The mapping for the motor is approximated by two lines, which are shown in the figure. Moreover, as seen in Fig. 5.4b, for the brake pressures less than $0.9KPa$, no brake is applied on the vehicle. The mapping for the brake is approximated by a line for brake pressures larger than this value. Therefore, the total longitudinal force is mapped to the brake pressure and motor torque by the following equations:

$$T_m = \begin{cases} 922\frac{F_{x_T}}{m} + 136 & F_{x_T} \geq -1.5m \\ 0 & F_{x_T} < -1.5m \end{cases}, \tag{5.3}$$

$$P_b = \begin{cases} 0 & F_{x_T} \geq -1.5m \\ -0.73\frac{F_{x_T}}{m} - 0.16 & F_{x_T} < -1.5m \end{cases}, \tag{5.4}$$

where T_m is the motor torque and P_b is the brake pressure. Moreover, the maximum total longitudinal force is also constrained to consider the limitation of the motor:

$$F_{x_T} \leq 3.89m, \tag{5.5}$$



(a) (b)

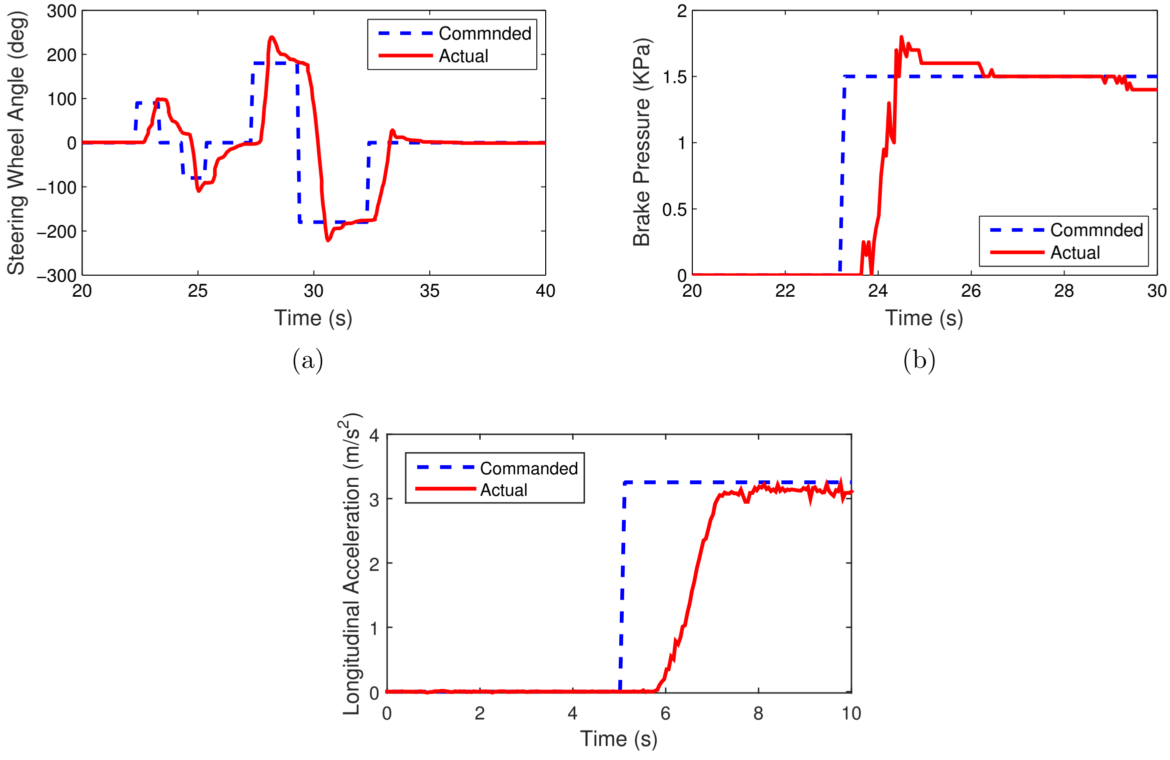
Figure 5.4: Mapping from the longitudinal acceleration to actuator inputs, a) mapping to motor torque, b) mapping to brake pressure.

Comparing the steering wheel angle and brake pressure commands to the steering wheel angle and brake pressure feedback show that there are also delays from the sent commands to the command implementation (Fig. 5.5a,5.5b). There is no motor torque feedback. However, comparing the equivalent longitudinal acceleration of the motor torque command to the measured longitudinal acceleration shows delays on the motor torque (Fig. 5.5c). As the results show, the delay is around $600ms$ for all the actuation systems. The following section predicts the states to compensate for these delays and the other delays in the system.

5.3.4 Delay Compensation

There are delays on the test vehicle platform that makes the vehicle's behavior oscillatory. Figure 5.6 shows the performance of the motion planning MPC in keeping a straight lane in speeds less than $25Km/h$. The route planner module generates a straight path in the direction of the vehicle heading to set the lane profile straight along the heading. As the results show, the vehicle needs very large steering angles to follow the straight path, which is inappropriate. The reason for this behavior is the delays of the test platform.

As explained in Section 5.3.1, the calculation time of the perception module is around



(a)

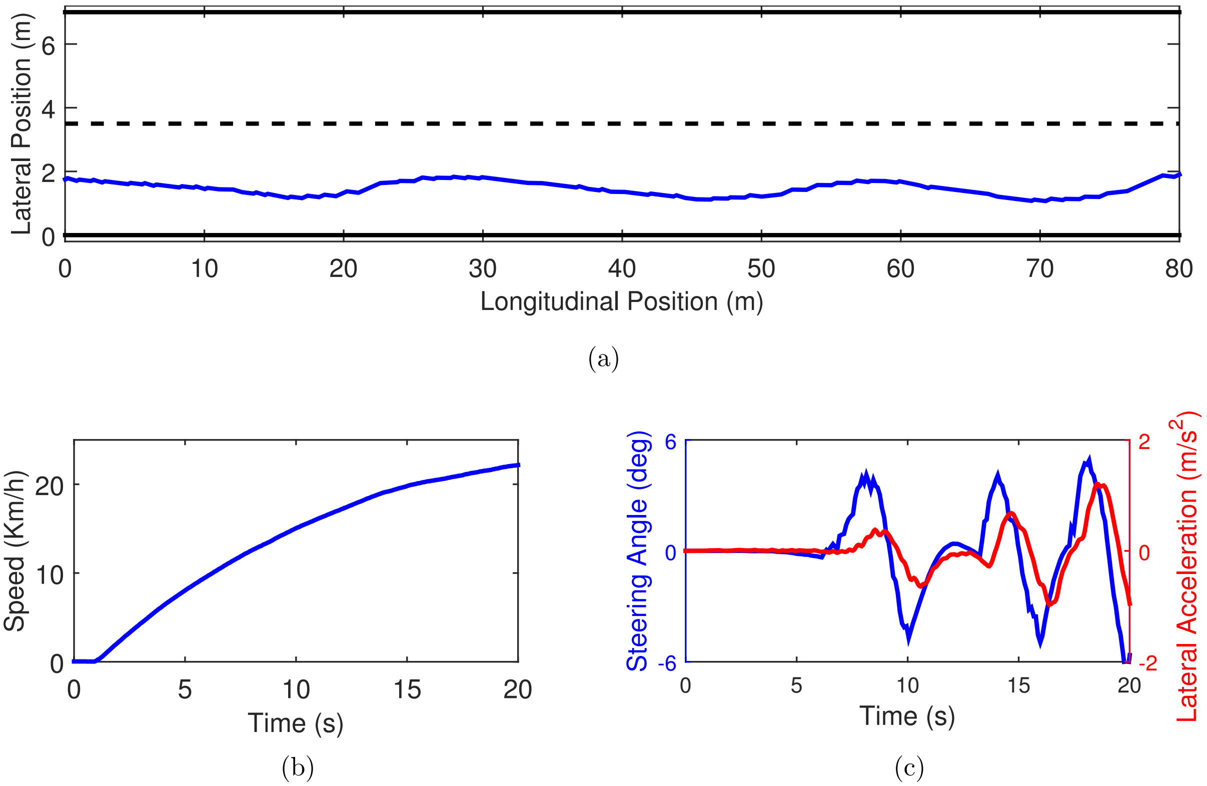
(b)

(c)

Figure 5.5: Actuation delays, a) steering delay, b) brake delay, c) motor delay.

300ms, which makes the vehicle position approximately 300ms old. Moreover, as explained in Section 5.3.3, the actuators implement the driving commands with around 600ms delay. It is also notable that the step time of the MPC is 100ms for the experimental tests, which causes a 100ms delay on the vehicle commands. Therefore, approximately, a total delay of 1000ms exists from the time that the vehicle states are measured to the time that the commands calculated based on the states are applied on the vehicle. As seen in Fig. 5.6, because of this large delay in the test platform, the test vehicle performs the lane keeping maneuver with an oscillatory behavior.

If a delay of k_e steps exists on the estimated vehicle states, the available estimated states at the current time t are not the current vehicle states and pertain to vehicle states



(a)

(b)

(c)

Figure 5.6: Experimental results for a lane keeping maneuver without delay compensation, a) vehicle path, b) vehicle speed, c) steering angle command and lateral acceleration.

at $t - k_e$. If this delay is the only source of delay in the system, the current vehicle states can be predicted based on the previous control inputs, $u(t-i)$ for $1 \leq i \leq k_e$, to be used as the initial states of MPC [90],[91]. Moreover, if the system delay is only because of the MPC step time, the current control input, $u(t)$, is implemented on the vehicle at the next step time. Therefore, $u(t)$ should be calculated for the states pertaining to $t+1$ instead of the states pertaining to t. The control input calculated at the previous step, $u(t-1)$, is applied on the vehicle during this delay time, and the vehicle states can be predicted based on this input to be used as the MPC's initial states [92]. The same statement holds for the actuation delay. If the only source of delay in the system is the actuation system, the control input calculated at the current time, $u(t)$, is applied on the vehicle at $t + k_a$ where the actuation delay is of k_a steps. If there is no disturbance on the vehicle, the vehicle states can be predicted at $t + k_a$ based on the previous control inputs, $u(t - i)$ for $1 \leq i \leq k_a$, to be used as the MPC initial states [93].

In the test vehicle platform, all three mentioned delays exist. If some states are estimated with no delay, $\tilde{\boldsymbol{x}}_r$, and some states are estimated with a delay, $\tilde{\boldsymbol{x}}_d$, the predicted states, $\hat{\boldsymbol{x}}$, at $t - k_e$ is:

$$\hat{\boldsymbol{x}}(t - k_e) = \begin{bmatrix} \tilde{\boldsymbol{x}}_d(t) & \tilde{\boldsymbol{x}}_r(t - k_e) \end{bmatrix}^T, \tag{5.6}$$

The vehicle dynamics equations presented in (3.1) can be written as $\boldsymbol{x}(t{+}1) = f(\boldsymbol{x}(t), \boldsymbol{u}_c(t))$ using zero order hold method. This equation is used to predict the vehicle states. Because of the controller step time delay and the actuation delay, the control input applied on the vehicle at the current time is old for $k_a + 1$ steps. Therefore, the vehicle states can be predicted using the following equation:

$$\hat{\boldsymbol{x}}(t + i + 1) = f(\hat{\boldsymbol{x}}(t + i), \boldsymbol{u}_c(t + i - k_a - 1)), \quad -k_e \leq i \leq k_a, \tag{5.7}$$

In these experimental tests, a delay of 3 steps exists on the perception module. Therefore, the position states are estimated with a 3 steps delay. The longitudinal velocity, yaw rate, and yaw angle are assumed to be estimated with no delay. The lateral velocity is set to zero, and can be assumed to be either a delayed state or a state with no delay. It is assumed to be a delayed state so that it is predicted for the 3 steps to approach its steady state value. Therefore, k_e is set to 3, and the estimated state matrices are $\tilde{\boldsymbol{x}}_d = [X \quad Y \quad v]^T$ and $\tilde{\boldsymbol{x}}_r = [u \quad r \quad \theta]^T$. The actuation delay, k_a, is also 6 steps. Starting from the states obtained by (5.6), the vehicle states can be predicted using (5.7).

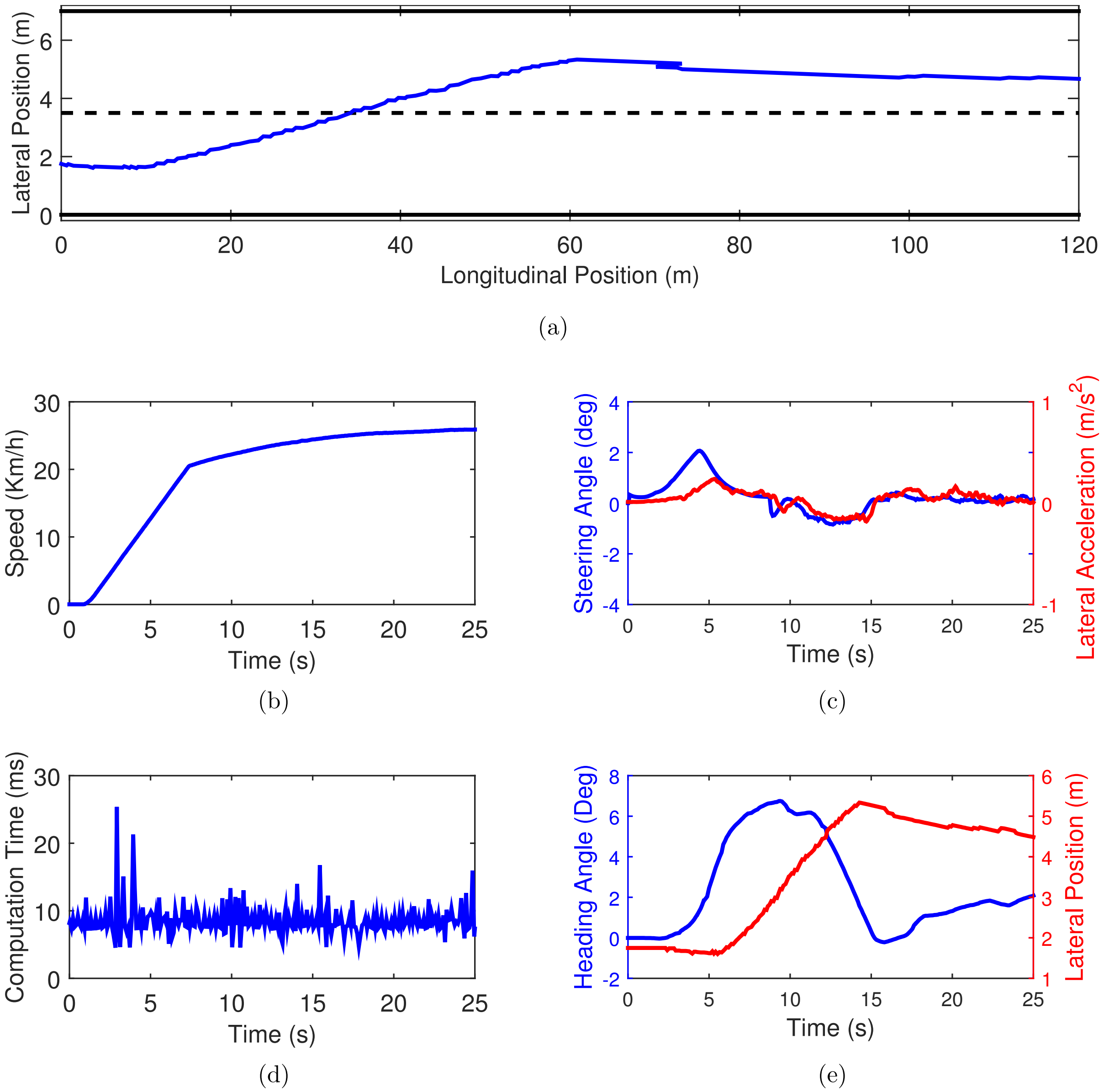
In the experimental tests, the predicted vehicle states at $t + k_a + 1$ are used as the MPC initial states instead of the estimated states to consider the effect of the delays. It is notable that the states estimated with no delay are available at t. Therefore, in the process of prediction, their estimated values at t replace their corresponding predicted states at t for a more accurate prediction:

$$\hat{\boldsymbol{x}}(t) = \begin{bmatrix} \hat{\boldsymbol{x}}_d(t) & \tilde{\boldsymbol{x}}_r(t) \end{bmatrix}^T . \tag{5.8}$$

Moreover, the future desired values of the MPC are calculated for $t + k_a + 1$ to factor in the delays in the setup of MPC problem [94].

The results of a lane change maneuver to the left with the predicted states is illustrated in Fig. 5.7. As the results show, the vehicle can change its lane with an appropriate behavior. Moreover, after the lane change, the vehicle can follow its lane with a considerably smaller steering angle compared to the maneuver shown in Fig. 5.6. It can be concluded that, using the predicted states as the MPC initial states, the motion planning module generates more reasonable inputs by considering the effect of the delay. For the rest of the experimental tests, the motion planning MPC uses the predicted states as its initial states. The computation time of the motion planning module is also illustrated in Fig. 5.7d. It shows that the calculation time of the motion planning module is less than $30ms$, and the module can plan the vehicle trajectory for real-time applications with step times over $30ms$.

As mentioned in Section 5.2, the perception module uses a SLAM method to localize the vehicle. The SLAM method localizes the vehicle with large errors. The heading angle and the lateral position of the lane change maneuver are plotted over time in Fig. 5.7e to show the errors. The heading angle is relative to the initial heading angle. As the results show, from the time $17s$ to $25s$, the heading angle is positive. Moreover, The steering angle is also less than $0.3°$ during this time, and the vehicle sideslip angle is very small. Therefore, it is expected that the vehicle moves toward the left side of the lane. However, the plot of the lateral position shows that the vehicle moves toward the right side of the lane during this time, which is because of the localization error. This error avoids the motion planning MPC for perfect path tracking. Moreover, the accumulation of this error causes large lateral offsets from the desired path, which makes performing maneuvers with large lateral movements difficult.



(a)

(b)

(c)

(d)

(e)

Figure 5.7: Experimental results for a lane keeping maneuver without delay compensation, a) vehicle path, b) vehicle speed, c) steering angle command and lateral acceleration, d) heading angle and lateral position, e) calculation time of motion planning module.

5.4 Test Scenarios

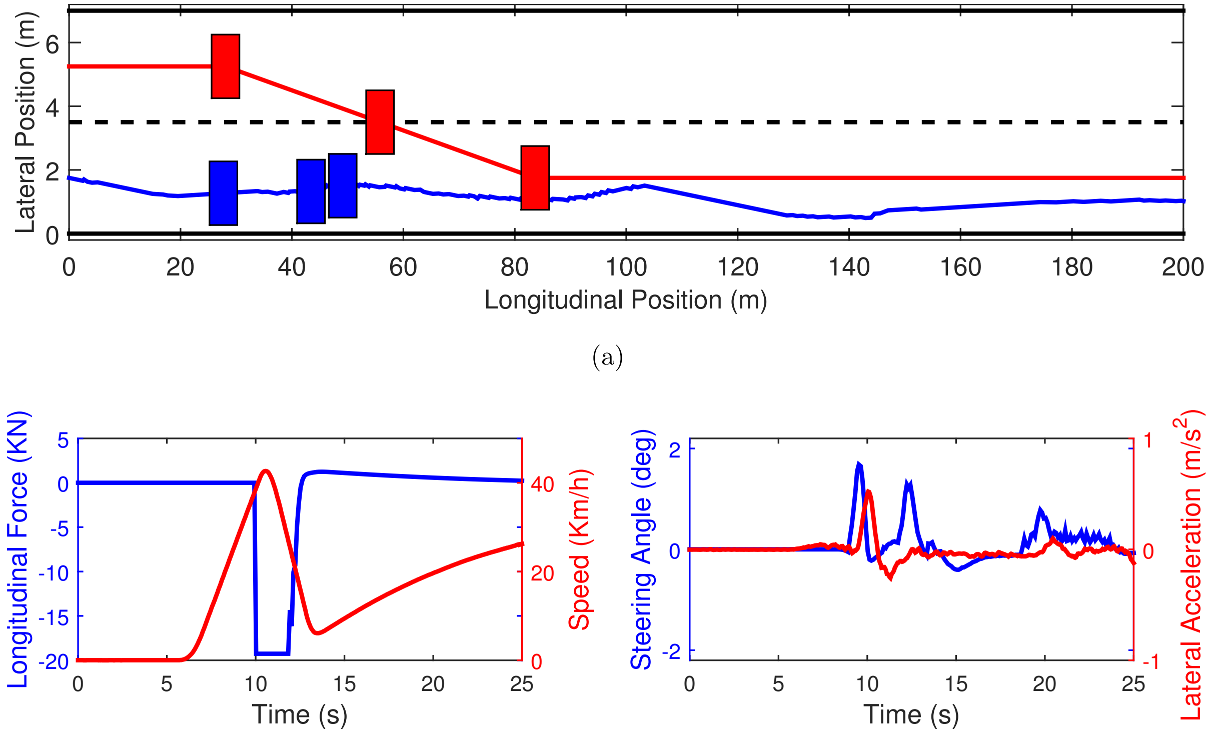
In this section, some test scenarios are performed on the test vehicle. The test scenarios are obstacle avoidance scenarios similar to Scenarios 3-7 of Chapter 3. It is notable that the road profile is generated based on the predefined path of the route planner module regardless of the actual road profile, and SLAM has noticeable position errors accumulating over time. Since the tests are performed on the streets, the lateral movements are limited. Therefore, the test scenarios expressed in this section are scenarios designed for keeping a straight lane.

The desired longitudinal velocity of the scenarios is $40Km/h$. Each scenario starts with $5s$ of being static. Then, the vehicle accelerates for $5s$ with constant motor torque of $3000Nm$ while the lane should be kept. The torque is applied to increase the vehicle speed in a short time. Then, the obstacle avoidance scenario is performed.

5.4.1 Car approaching from the side

This scenario is similar to Scenario 3 of Chapter 3; a car is on the left lane and carelessly changes its lane to the test vehicle's lane. The test vehicle is commanded to keep its lane. So, it should reduce its speed and go to the right of the lane to avoid the obstacle. In this scenario, a virtual obstacle is included, which is moving on the left lane with a speed of $40Km/h$. It has the same longitudinal position as the test vehicle at the time of $10s$. It starts changing its lane at the time of $11s$ and changes its lane in $5s$.

The experimental results of this scenario are illustrated in Fig. 5.8. The obstacle and the test vehicle are plotted in Fig. 5.8a by rectangles in actual sizes at the times of $11s$, $13.5s$, and $16s$. As the results show, the vehicle reduces its speed rapidly to make space for the obstacle, as expected. After the obstacle passes, the vehicle increases its speed. The results also show that the vehicle moves to the right of the lane, but because of the large localization errors, the movement due to obstacle avoidance cannot be distinguished from the errors.



(a)

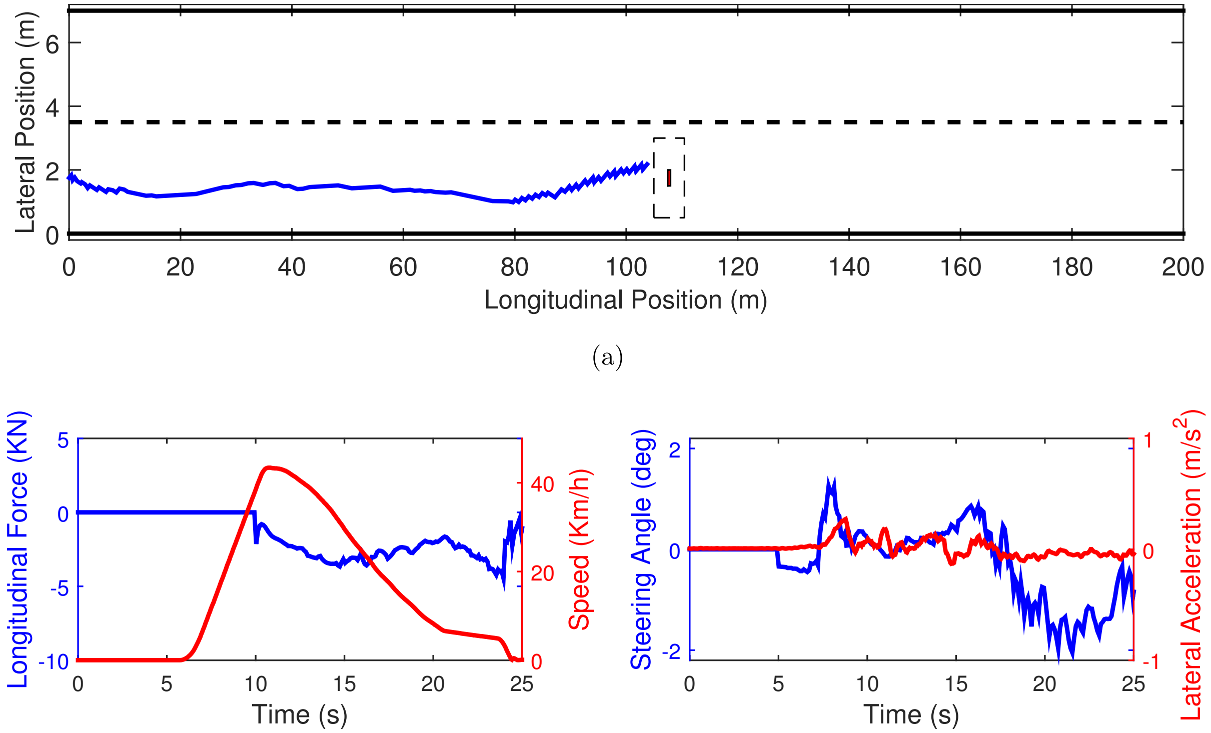
(b)

(c)

Figure 5.8: Experimental results for car approaching from the side, a) paths of vehicle and obstacle, blue: test vehicle, red: obstacle, b) longitudinal force command and vehicle speed, c) steering angle command and lateral acceleration.

5.4.2 Non-crossable obstacle on the middle of the lane

This scenario is similar to Scenario 6 of Chapter 3; a non-crossable obstacle is in the middle of the lane while lane keeping is commanded. So, the vehicle should stop behind the obstacle. In this scenario, a virtual obstacle is placed in the middle of the lane at $90m$ ahead of the vehicle at time $10s$, which is when the obstacle avoidance starts. The obstacle is $0.5m \times 0.5m$. Fig. 5.9 shows the experimental results of this scenario. The obstacle is shown with a red rectangle, and a contact rectangle is drawn around it with dashed lines. The contact rectangle represents the area that if the vehicle position is located in it, the vehicle and the obstacle are in contact. Therefore, a path that does not enter the contact



(a)

(b)

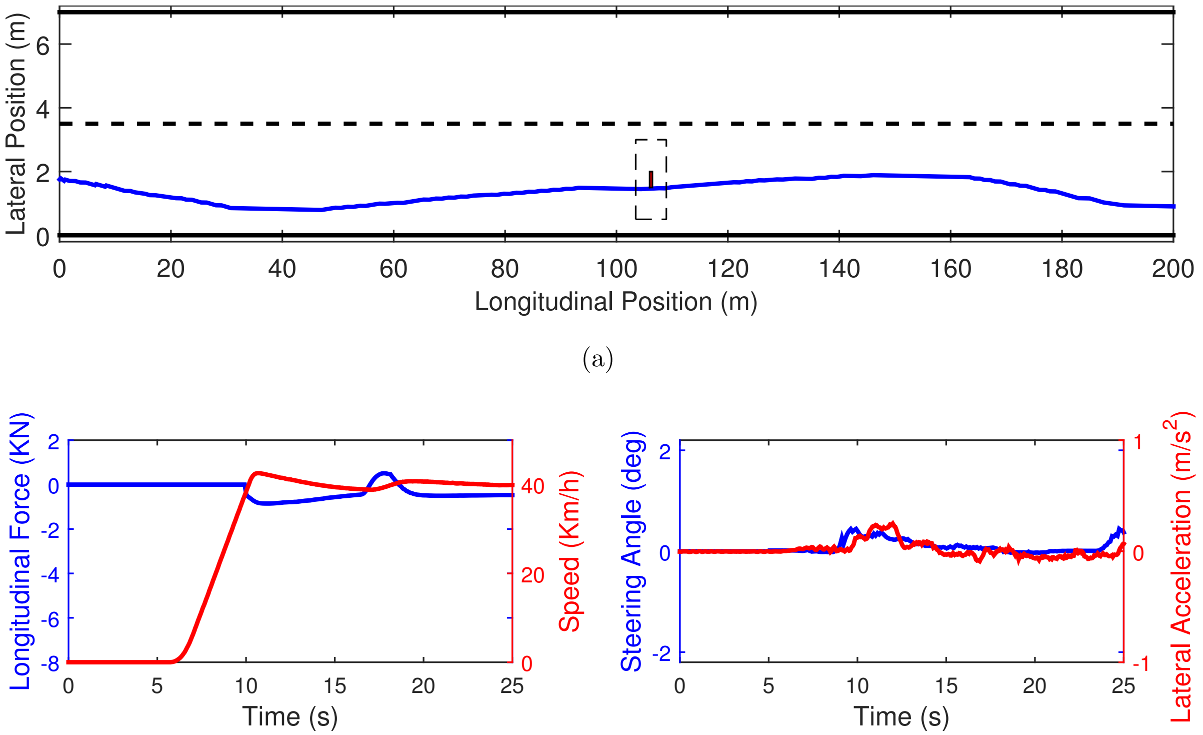
(c)

Figure 5.9: Experimental results for non-crossable obstacle on the middle of the lane, a) vehicle path and obstacle position, blue: test vehicle, red: obstacle, b) longitudinal force command and vehicle speed, c) steering angle command and lateral acceleration.

rectangle is obstacle-free. As seen, the vehicle stops behind the obstacle while keeping the lane, as expected.

5.4.3 Crossable obstacle in the middle of the lane

This scenario is similar to Scenario 7 of Chapter 3; a crossable obstacle is in the middle of the lane while lane keeping is commanded. Therefore, the vehicle should cross the obstacle. The scenario is set similar to the scenario of Section 5.4.2 except that the virtual obstacle is set to be crossable. Fig. 5.10 shows the experimental results of this scenario. As seen, the vehicle crosses the obstacle while keeping the lane and its desired speed, as expected.



(a)

(b)

(c)

Figure 5.10: Experimental results for crossable obstacle on the middle of the lane, a) vehicle path and obstacle position, blue: test vehicle, red: obstacle, b) longitudinal force command and vehicle speed, c) steering angle command and lateral acceleration.

5.4.4 Non-crossable Obstacle on the side of the lane

Two scenarios similar to Scenario 4 of Chapter 3 are performed; a non-crossable obstacle is on the side of the lane, placed once on the right side of the lane and once on the left side of the lane. Lane keeping is also commanded. So, the vehicle should pass the obstacle that is on the right side of the lane on the obstacle's left and the obstacle that is on the left side of the lane on the obstacle's right. The scenarios are set similar to the scenario of Section 5.4.2 except that the virtual obstacle is placed once at 1.5m on the right side of the center of the lane and once at 1.5m on the left side of the center of the lane. The experimental results of these scenarios are demonstrated Fig. 5.11.

Figure 5.11a shows the vehicle's path and the obstacle's positions for these scenarios. The obstacle on the right side of the lane is demonstrated by a red rectangle, and the vehicle's path corresponding to its avoidance is in blue. Moreover, the obstacle on the left side of the lane is demonstrated by an orange rectangle, and the vehicle's path corresponding to its avoidance is in green. As the results show, when the obstacle is on the right side, the vehicle slightly moves to the left to avoid the obstacle while keeping the lane and the desired velocity. Moreover, when the obstacle is on the left side, the vehicle stays on the right side of the lane to avoid the obstacle while keeping the lane and the desired velocity, as expected.

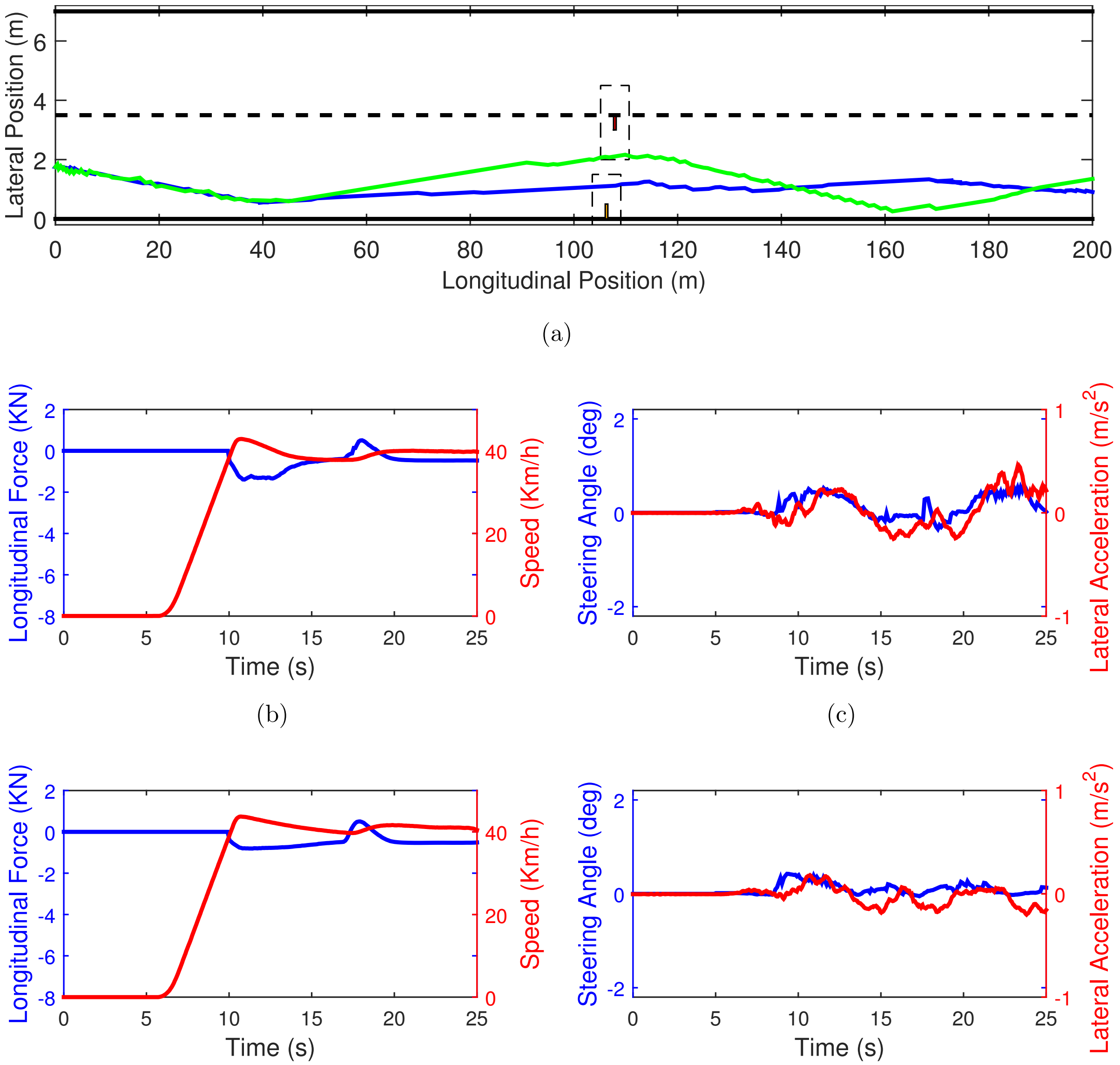
5.4.5 Crossable obstacle on the side of the lane

Two scenarios similar to Scenario 5 of Chapter 3 are preformed; a crossable obstacle is on the side of the lane, placed once on the right side of the lane and once on the left side of the lane. These scenarios are set similar to the scenario of Section 5.4.4 except that the virtual obstacles are crossable. Therefore, the vehicle should pass the obstacle that is on the right side of the lane on the obstacle's left and the obstacle that is on the left side of the lane on the obstacle's right.

Fig. 5.12 shows the experimental results of this scenario. As the results show, when the obstacle is on the right side, the vehicle moves slightly to the left to avoid the obstacle while keeping the lane and the desired velocity. Moreover, when the obstacle is on the left side, the vehicle stays on the right side of the lane to avoid the obstacle while keeping the lane and the desired velocity, as expected.

5.5 Summary

The motion planning MPC presented in Chapter 3 was implemented on an autonomous test vehicle platform in this chapter. The MPC was modified to be compatible with this platform, and some test scenarios were performed to validate the simulation results of the motion planning MPC with experimental results in real time.



(a)

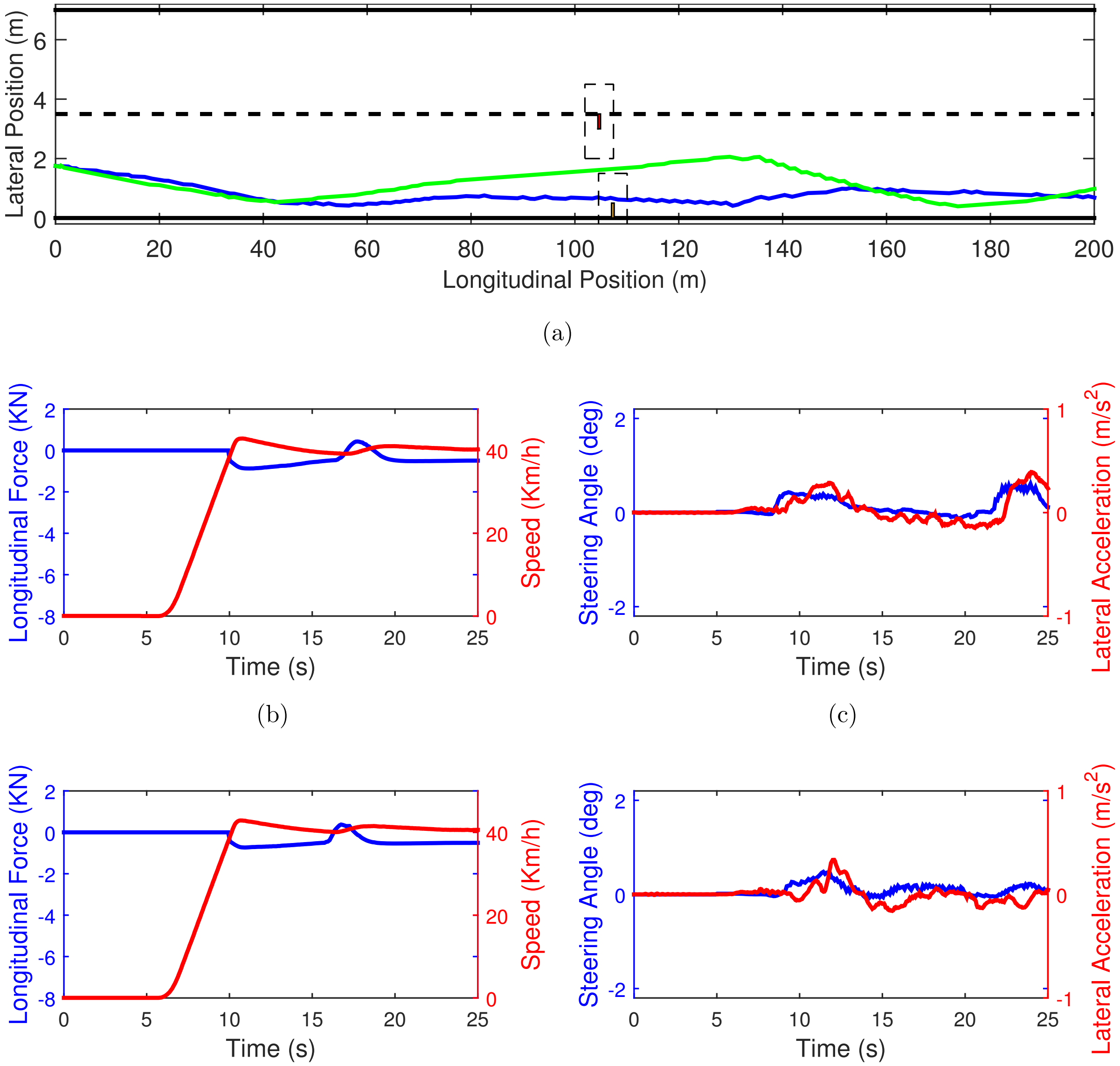
(b)

(c)

(d)

(e)

Figure 5.11: Experimental results for non-crossable obstacle on the side of the lane, a) vehicle path and obstacle position, blue: test vehicle for right obstacle, red: right obstacle, green: test vehicle for left obstacle, orange: left obstacle, b) longitudinal force command and vehicle speed for right obstacle, c) steering angle command and lateral acceleration for right obstacle, d) longitudinal force command and vehicle speed for left obstacle, e) steering angle command and lateral acceleration for left obstacle.



(a)

(b)

(c)

(d)

(e)

Figure 5.12: Experimental results for crossable obstacle on the side of the lane, a) vehicle path and obstacle position, blue: test vehicle for right obstacle, red: right obstacle, green: test vehicle for left obstacle, orange: left obstacle, b) longitudinal force command and vehicle speed for right obstacle, c) steering angle command and lateral acceleration for right obstacle, d) longitudinal force command and vehicle speed for left obstacle, e) steering angle command and lateral acceleration for left obstacle.

There were delays on the test platform. The vehicle position was obtained by the perception module, which had a calculation time of around $300ms$. The results also showed a delay of around $600ms$ on the actuation system. The step time of the motion planning module of the test vehicle was also $100ms$. Therefore, a total delay of around $1000ms$ existed on the test platform. The experimental results showed that if the delay was not compensated for, the test vehicle could not follow a path smoothly, even at low speeds. The delay was compensated by using predicted vehicle states as the MPC's initial states. Using this method, a path could be followed smoothly, and a lane change could also be performed appropriately.

Some test scenarios were also performed to validate the simulation results of the motion planning MPC. The test vehicle stopped behind a non-crossable obstacle when passing on its side was not possible, and passed in on its side when possible. The vehicle also crossed a crossable obstacle, when passing on its side was not possible, and passed in on its side, when possible. Moreover, when a car approached the test vehicle carelessly from the side, the vehicle reduced its speed to make space for the car and avoid the car. Therefore, although there was a large delay in the platform, and the localization system had large errors, the motion planning MPC was modified adequately to work in this platform and performed appropriately. The experimental results validated the simulation results of the motion planning MPC. They also showed that the MPC is implementable in real time.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, a motion planning MPC was designed for an autonomous vehicle that plans the vehicle's trajectory according to obstacle priority orders. This motion planning MPC categorizes the obstacles as crossable and non-crossable, and treat each category according to its characteristics. It also prioritizes the non-crossable obstacles based on their possible crash cost, and plans its trajectory based on their priority order. Moreover, it utilizes a vehicle model using a bicycle model and tire constraints that remain valid at tire force limits. Furthermore, the iterative obstacle avoidance method presented in this thesis allows for a more optimal motion planning by reducing the number of feasible trajectories removed by convexification. The major findings and contributions of this thesis are as follows.

The obstacles were prioritized based on their avoidance necessity by utilizing a potential field in the motion planning MPC. Potential fields have been used for obstacle avoidance in motion planning MPCs in literature. However, one potential function has been used for all the kinds of obstacles in each MPC. This approach treats all the obstacles similarly. However, obstacles can be prioritized by assigning each category of obstacles a different potential function that corresponds to the obstacles' characteristics. This approach treats obstacles based on their characteristics. In this thesis, the obstacles were categorized as crossable and non-crossable using this approach. Assigning an appropriate function to each category makes the non-crossable obstacles to be avoided at any condition, and the crossable obstacles to be avoided when it is comfortable.

Non-crossable obstacles were prioritized based on their possible crash costs by applying lexicographic optimization on the motion planning MPC. It is common to use obstacle constraints for obstacle avoidance in motion planning MPCs. This method considers the same priority for all obstacles. However, the obstacles have different crash costs and should be prioritized. In this thesis, obstacle constraints were included in the MPC for non-crossable obstacles. The obstacles were prioritized by prioritizing their corresponding constraints through lexicographic optimization. Therefore, in a situation where a crash is unavoidable, the motion planning MPC plans the vehicle's trajectory such that the vehicle avoids the obstacles with the highest priority orders.

A vehicle model was developed for the MPC that is valid at tire force limits. A linear vehicle bicycle model has been used in the literature of motion planning MPC. The model uses linear tire models and accompanies constraints on tire sideslip angle to keep the model valid by keeping the tire in its linear force region. On the other hand, considering the limitations caused by combined tire slip and load transfer in the vehicle model makes the model valid at tire force limits. However, in the literature, there is no linear bicycle model that considers combined tire slip in modeling both longitudinal and lateral motions of a motion planning MPC. In this thesis, tire constraints were presented that consider the limitations of the combined tire slip and the effect of longitudinal load transfer on these limitations. The tire constraints were used in the motion planning MPC to generate a vehicle model that considers tire capacities. The tire constraints also cover the tire sideslip angle constraint, and keep the tire in its linear force region to keep the vehicle model valid.

An iterative obstacle avoidance method was presented that reduces the number of trajectories removed by convexification. If the obstacle avoidance is performed through obstacle constraints, and the motion planning MPC is quadratic, the obstacle-free area should be convexified by linear obstacle constraints. The convexification process removes some of the feasible trajectories. One of the best convexification methods presented for autonomous road vehicles is the linearized signed distance method. However, this method constrains the vehicle to stop behind an obstacle in front of it, and does not allow a swerving maneuver in such a situation. In this thesis, in addition to generating a constraint set based on this method, two constraint sets corresponding to swerving to the right and to the left were also generated. The union of these three constraint sets has less removed trajectories than the first constraint set. IQMPC was presented to solve the problem for the union of the three sets; it consists of three MPCs, one for each constraint set. IQMPC increases the calculation time, but it increases the available feasible trajectories.

The performance of the developed MPC was evaluated through computer simulations in MATLAB/Simulink and CarSim. A high fidelity vehicle model of an electric Chevrolet Equinox in CarSim was utilized to simulate the vehicle behavior. The MPC was implemented in Simulink, and a quadratic programming solver called QPOASES was used to solve the MPC quadratic problem. The MPC was simulated for some test scenarios. The simulation results showed that the MPC plans the vehicle's trajectory appropriately in complex driving situations. The results also showed that the MPC treats obstacles based on their characteristics; the vehicle avoids the crossable obstacles only when it is comfortable to do so, but it tries to always avoid the non-crossable obstacles. They also showed that the MPC prioritizes the obstacles; in situations that avoiding all the obstacles is not possible, the vehicle avoids the obstacles with the highest priority orders and crashes into the obstacles with the lowest priority orders.

It can also be seen from the results that the vehicle generates large tire forces on the tire force limits, when required; i.e. the MPC used the tire capacity in generating the vehicle's trajectories. Furthermore, the results showed that the proposed IQMPC increases the available feasible trajectories; in a situation where stopping behind an obstacle in front of the vehicle is not possible, but swerving is possible, the IQMPC avoids the obstacle by swerving while the MPC based on the linearized signed distance constraint set cannot avoid the obstacle.

The proposed MPC was also implemented on a test vehicle platform to validate the simulation results of the MPC with experimental results. The MPC was modified to become compatible with the test vehicle software platform. It was also modified to compensate for the test platform delays by using predicted vehicle states as the MPC initial states. The modified MPC was implemented on the test vehicle for some test scenarios. The results showed that the MPC was modified adequately to work in this platform and performed appropriately despite the large delays in the platform and large localization errors. The experimental results also validated the simulation results of the MPC and showed that the MPC is implementable in real time.

6.2 Future Work

The following are a few suggestions for the continuation of the work done in this thesis to improve the performance of the motion planning MPC.

- Improving obstacle priority orders: In this thesis, the priority orders were assigned to each obstacle based on the possible crash costs of the obstacle's category; e.g. cars were prioritized over rocks. However, the speed and the angle of the crash also have great impacts on the crash costs; e.g. a high speed crash with a rock is more costly than a low speed crash with a car. These factors can be considered in predicting the possible crash cost of obstacles to obtain more accurate obstacle priority orders.
- Obtaining the optimal lane by utilizing IQMPC: The desired lane is usually planned in the behavioral planning stage of an autonomous vehicle. However, this stage does not consider the vehicle dynamics and obstacle avoidance in planning the desired lane. IQMPC can be modified to find trajectories for lane keeping, lane change to the left, and lane change to the right. The trajectories can be sent to the behavioral planning stage as feedback so that this stage decides the desired lane by considering the vehicle dynamics and obstacle avoidance performance of the optimal trajectory corresponding to each lane.
- Considering ride comfort: The weighting matrices in the objective function of the proposed motion planning MPC were tuned for a smooth driving condition, but the ride comfort was not optimized. Passenger ride comfort indexes can be included in the MPC objective function so that the optimality in terms of passengers' comfort is also considered in planning the vehicle's trajectories.
- Improving IQMPC: IQMPC was proposed to improve the obstacle avoidance performance by covering more feasible trajectories. However, it does not cover all the

feasible trajectories. The number of iterations and the constraint set of each iteration can be modified so that more feasible trajectories are covered, and the obstacle avoidance performance is improved.

- Controlling tire slips: In this thesis, tire slips were not controlled in the motion planning MPC; they were controlled in a slip controller. Tire slips can be controlled in the MPC by including the wheel dynamics in the vehicle model of the MPC. This way, the vehicle's trajectory is planned based on a more accurate vehicle model. It is notable that to control the tire slips, the torques on each wheel should be considered as a control input, which increases the number of control inputs, and consequently, the calculation time. However, the tire slip should be considered only at a few first prediction steps. Therefore, to reduce the number of control inputs, the wheel torques can be replaced by the total longitudinal force after these steps.

- Planning trajectories with large yaw angles: The vehicle model used in the proposed motion planning MPC is a linear bicycle model linearized for small yaw angles. However, there are some maneuvers that require large yaw angles in a prediction horizon, e.g. U-turn. In these maneuvers, the proposed vehicle model is invalid, and therefore, the motion planning MPC cannot plan these maneuvers appropriately. These maneuvers are performed at low speeds. Therefore, a switching MPC can be developed that uses the dynamics model at high speeds and a kinematics model at low speeds. This MPC can plan trajectories with large yaw angles at low velocities while it considers the vehicle dynamics at high speeds where vehicle dynamics consideration is required.

References

[1] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report DOT HS 812 115, 2015.

[2] Noah J Goodall. Machine ethics and automated vehicles. In *Road vehicle automation*, pages 93–102. Springer, 2014.

[3] Noah Goodall. Ethical decision making during automated vehicle crashes. *Transportation Research Record: Journal of the Transportation Research Board*, (2424):58–65, 2014.

[4] Thierry Fraichard and James J Kuffner. Guaranteeing motion safety for robots. *Autonomous Robots*, 32(3):173–175, 2012.

[5] Rodrigo Benenson, Thierry Fraichard, and Michel Parent. Achievable safety of driverless ground vehicles. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 515–521. IEEE, 2008.

[6] Patrick Lin. Why ethics matters for autonomous cars. In *Autonomous Driving*, pages 69–85. Springer, 2016.

[7] Noah J Goodall. Can you program ethics into a self-driving car? *IEEE Spectrum*, 53(6):28–58, 2016.

[8] Milad Jalali, Amir Khajepour, Shih-ken Chen, and Bakhtiar Litkouhi. Integrated stability and traction control for electric vehicles using model predictive control. *Control Engineering Practice*, 54:256–266, 2016.

[9] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016.

[10] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

[11] Jose M Álvarez, Antonio M López, Theo Gevers, and Felipe Lumbreras. Combining priors, appearance, and context for road detection. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1168–1178, 2014.

[12] Qingquan Li, Long Chen, Ming Li, Shih-Lung Shaw, and Andreas Nuchter. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. *IEEE Transactions on Vehicular Technology*, 63(2):540–555, 2014.

[13] Vijay John, Keisuke Yoneda, Zheng Liu, and Seiichi Mita. Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching. *IEEE Transactions on Computational Imaging*, 1(3):159–173, 2015.

[14] Tao Chen and Shijian Lu. Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression. *IEEE Transactions on Vehicular Technology*, 65(6):4006–4015, 2016.

[15] Beomseong Kim, Baehoon Choi, Seongkeun Park, Hyunju Kim, and Euntai Kim. Pedestrian/vehicle detection using a 2.5-d multi-layer laser scanner. *IEEE Sensors Journal*, 16(2):400–408, 2016.

[16] Javier Hernandez-Aceituno, Rafael Arnay, Jonay Toledo, and Leopoldo Acosta. Using kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE Sensors Journal*, 16(10):3603–3610, 2016.

[17] Ehsan Hashemi, Alireza Kasaiezadeh, Saeid Khosravani, Amir Khajepour, Nikolai Moshchuk, and Shih-Ken Chen. Estimation of longitudinal speed robust to road conditions for ground vehicles. *Vehicle System Dynamics*, 54(8):1120–1146, 2016.

[18] Alberto Y Hata and Denis F Wolf. Feature detection for vehicle localization in urban environments using a multilayer lidar. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):420–429, 2016.

[19] Robert Geisberger, Peter Sanders, Dominik Schultes, and Christian Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46(3):388–404, 2012.

[20] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. Route planning in transportation networks. In *Algorithm Engineering*, pages 19–80. Springer, 2016.

[21] Michael Ardelt, Constantin Coester, and Nico Kaempchen. Highly automated driving on freeways in real traffic using a probabilistic framework. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1576–1585, 2012.

[22] Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, and Lydie Nouveliere. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):589–606, 2010.

[23] Paolo Falcone, H Eric Tseng, Francesco Borrelli, Jahan Asgari, and Davor Hrovat. Mpc-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(S1):611–628, 2008.

[24] Toshihiro Hiraoka, Osamu Nishihara, and Hiromitsu Kumamoto. Automatic path-tracking controller of a four-wheel steering vehicle. *Vehicle System Dynamics*, 47(10):1205–1227, 2009.

[25] Stephen M Erlien. *Shared vehicle control using safe driving envelopes for obstacle avoidance and stability*. PhD thesis, Stanford University, 2015.

[26] Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.

[27] Ming Feng Hsieh and Umit Ozguner. A parking algorithm for an autonomous vehicle. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1155–1160. IEEE, 2008.

[28] Misel Brezak and Ivan Petrovic. Real-time approximation of clothoids with bounded error for path planning applications. *IEEE Transactions on Robotics*, 30(2):507–515, 2014.

[29] Christoph G Keller, Thao Dang, Hans Fritz, Armin Joos, Clemens Rabe, and Dariu M Gavrila. Active pedestrian safety by automatic braking and evasive steering. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1292–1304, 2011.

[30] Zhao Liang, Guoqiang Zheng, and Jishun Li. Automatic parking path optimization based on bezier curve fitting. In *Automation and Logistics (ICAL), 2012 IEEE International Conference on*, pages 583–587. IEEE, 2012.

[31] Tomas Berglund, Andrej Brodnik, Håkan Jonsson, Mats Staffanson, and Inge Soderkvist. Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles. *IEEE Transactions on Automation Science and Engineering*, 7(1):167–172, 2010.

[32] Rahul Kala and Kevin Warwick. Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization. *Journal of Intelligent & Robotic Systems*, 72(3-4):559, 2013.

[33] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, 29(5):485–501, 2010.

[34] Maxim Likhachev and Dave Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research*, 28(8):933–945, 2009.

[35] Tianyu Gu, Jarrod Snider, John M Dolan, and Jin-woo Lee. Focused trajectory planning for autonomous on-road driving. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 547–552. IEEE, 2013.

[36] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.

[37] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1478–1483. IEEE, 2011.

[38] Jianqiang Wang, Jian Wu, and Yang Li. The driving safety field based on driver–vehicle–road interactions. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2203–2214, 2015.

[39] Michael T Wolf and Joel W Burdick. Artificial potential functions for highway driving with collision avoidance. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3731–3736. IEEE, 2008.

[40] 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, 2017.

[41] Erwin Prassler, Jens Scholz, and Paolo Fiorini. Navigating a robotic wheelchair in a railway station during rush hour. *The international journal of robotics research*, 18(7):711–727, 1999.

[42] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[43] Yiqi Gao, Theresa Lin, Francesco Borrelli, Eric Tseng, and Davor Hrovat. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In *ASME 2010 dynamic systems and control conference*, pages 265–272. American Society of Mechanical Engineers, 2010.

[44] Julia Nilsson, Mattias Brännström, Jonas Fredriksson, and Erik Coelingh. Longitudinal and lateral control for automated yielding maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1404–1414, 2016.

[45] Mehdi Jalalmaab, Barış Fidan, Soo Jeon, and Paolo Falcone. Model predictive path planning with time-varying safety constraints for highway autonomous driving. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 213–217. IEEE, 2015.

[46] Julia Nilsson, Paolo Falcone, Mohammad Ali, and Jonas Sjöberg. Receding horizon maneuver generation for automated highway driving. *Control Engineering Practice*, 41:124–133, 2015.

[47] Xiangjun Qian, Arnaud De La Fortelle, and Fabien Moutarde. A hierarchical model predictive control framework for on-road formation control of autonomous vehicles. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 376–381. IEEE, 2016.

[48] Ashwin Carvalho, Stéphanie Lefévre, Georg Schildbach, Jason Kong, and Francesco Borrelli. Automated driving: The role of forecasts and uncertaintya control perspective. *European Journal of Control*, 24:14–32, 2015.

[49] Georg Schildbach and Francesco Borrelli. Scenario model predictive control for lane change assistance on highways. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 611–616. IEEE, 2015.

[50] Kun Zhang, Jonathan Sprinkle, and Ricardo G Sanfelice. Computationally aware control of autonomous vehicles: a hybrid model predictive control approach. *Autonomous Robots*, 39(4):503–517, 2015.

[51] Janick V Frasch, Andrew Gray, Mario Zanon, Hans Joachim Ferreau, Sebastian Sager, Francesco Borrelli, and Moritz Diehl. An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles. In *Control Conference (ECC), 2013 European*, pages 4136–4141. IEEE, 2013.

[52] Valerio Turri, Ashwin Carvalho, Hongtei Eric Tseng, Karl Henrik Johansson, and Francesco Borrelli. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 378–383. IEEE, 2013.

[53] Yongsoon Yoon, Jongho Shin, H Jin Kim, Yongwoon Park, and Shankar Sastry. Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, 17(7):741–750, 2009.

[54] JM Park, DW Kim, YS Yoon, HJ Kim, and KS Yi. Obstacle avoidance of autonomous vehicles based on model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 223(12):1499–1516, 2009.

[55] Alexander Liniger, Alexander Domahidi, and Manfred Morari. Optimization-based autonomous racing of 1: 43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.

[56] Sterling J Anderson, Steven C Peters, Tom E Pilutti, and Karl Iagnemma. Design and development of an optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *Robotics Research*, pages 39–54. Springer, 2011.

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

[58] Ashwin Carvalho, Yiqi Gao, Stéphanie Lefevre, and Francesco Borrelli. Stochastic predictive control of autonomous vehicles in uncertain environments. In *12th International Symposium on Advanced Vehicle Control*, 2014.

[59] Boliang Yi, Stefan Gottschling, Jens Ferdinand, Norbert Simm, Frank Bonarens, and Christoph Stiller. Real time integrated vehicle dynamics control and trajectory planning with mpc for critical maneuvers. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 584–589. IEEE, 2016.

[60] Stephen M Erlien, Susumu Fujita, and Joseph Christian Gerdes. Shared steering control using safe envelopes for obstacle avoidance and vehicle stability. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):441–451, 2016.

[61] Joseph Funke, Matthew Brown, Stephen M Erlien, and J Christian Gerdes. Prioritizing collision avoidance and vehicle stabilization for autonomous vehicles. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 1134–1139. IEEE, 2015.

[62] Andrew Gray, Yiqi Gao, J Karl Hedrick, and Francesco Borrelli. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 208–213. IEEE, 2013.

[63] Andrew Gray, Yiqi Gao, Theresa Lin, J Karl Hedrick, and Francesco Borrelli. Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2329–2334. IEEE, 2013.

[64] Yi-Wen Liao and J Karl Hedrick. A discrete-time integral sliding model predictive control for obstacle avoidance of ground vehicles. In *ASME 2015 Dynamic Systems and Control Conference*, pages V003T44A002–V003T44A002. American Society of Mechanical Engineers, 2015.

[65] Muhammad Awais Abbas, Ruth Milman, and J Mikael Eklund. Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–6. IEEE, 2014.

[66] Jiechao Liu, Paramsothy Jayakumar, Jeffrey L Stein, and Tulga Ersal. A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles. *Vehicle System Dynamics*, 54(11):1629–1650, 2016.

[67] Christian Gotte, Martin Keller, Carsten Hass, Karl-Heinz Glander, Alois Seewald, and Torsten Bertram. A model predictive combined planning and control approach for guidance of automated vehicles. In *Vehicular Electronics and Safety (ICVES), 2015 IEEE International Conference on*, pages 69–74. IEEE, 2015.

[68] Stephen M Erlien, Susumu Fujita, and J Christian Gerdes. Safe driving envelopes for shared control of ground vehicles. *IFAC Proceedings Volumes*, 46(21):831–836, 2013.

[69] Jongsang Suh, Kyongsu Yi, Jiyeol Jung, Kyungjun Lee, Hyokjin Chong, and Bongchul Ko. Design and evaluation of a model predictive vehicle control algorithm for automated driving using a vehicle traffic simulator. *Control Engineering Practice*, 51:92–107, 2016.

[70] Jongsang Suh, Beomjun Kim, and Kyongsu Yi. Stochastic predictive control based motion planning for lane change decision using a vehicle traffic simulator. In *Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2016 IEEE Conference and Expo*, pages 900–907. IEEE, 2016.

[71] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[72] Eugene C Freuder, Robert Heffernan, Richard J Wallace, and Nic Wilson. Lexicographically-ordered constraint satisfaction problems. *Constraints*, 15(1):1–28, 2010.

[73] Eric C Kerrigan and Jon M Maciejowski. Designing model predictive controllers with prioritised constraints and objectives. In *Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 33–38. IEEE, 2002.

[74] Yadollah Rasekhipour, Amir Khajepour, Shih-Ken Chen, and Bakhtiar Litkouhi. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1255–1267, 2017.

[75] Neeraj Kumar, Jong-Hyouk Lee, and Joel JPC Rodrigues. Intelligent mobile video surveillance system as a bayesian coalition game in vehicular sensor networks: Learning automata approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1148–1161, 2015.

[76] Haiyan Guan, Jonathan Li, Yongtao Yu, Zheng Ji, and Cheng Wang. Using mobile lidar data for rapidly updating road markings. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2457–2466, 2015.

[77] Ehsan Hashemi, Saeid Khosravani, Amir Khajepour, Alireza Kasaiezadeh, Shih-Ken Chen, and Bakhtiar Litkouhi. Longitudinal vehicle state estimation using nonlinear and parameter-varying observers. *Mechatronics*, 43:28–39, 2017.

[78] Ehsan Hashemi, Mohamamd Pirani, Amir Khajepour, Alireza Kasaiezadeh, Shih-Ken Chen, and Bakhtiar Litkouhi. Corner-based estimation of tire forces and vehicle velocities robust to road conditions. *Control Engineering Practice*, 61:28–40, 2017.

[79] Amir Khajepour, M Saber Fallah, and Avesta Goodarzi. *Electric and Hybrid Vehicles: Technologies, Modeling and Control-A Mechatronic Approach*. John Wiley & Sons, 2014.

[80] John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: science and systems*, volume 9, pages 1–10. Citeseer, 2013.

[81] Christer Ericson. *Real-time collision detection*. CRC Press, 2004.

[82] Arne Kesting, Martin Treiber, Martin Schönhof, and Dirk Helbing. Adaptive cruise control design for active congestion avoidance. *Transportation Research Part C: Emerging Technologies*, 16(6):668–683, 2008.

[83] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420, 2006.

[84] Mirai Tanaka and Kazuhide Nakata. Positive definite matrix approximation with condition number constraint. *Optimization Letters*, 8(3):939–947, 2014.

[85] Paul T Boggs and Jon W Tolle. *Sequential quadratic programming*, volume 4. Cambridge University Press, 1995.

[86] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[87] Yadollah Rasekhipour, Iman Fadakar, Amir Khajepour, Shih-Ken Chen, and Bakhtiar Litkouhi. Prioritizing obstacles in path planning of autonomous road vehicles using lexicographic optimization. *IEEE Transactions on Intelligent Transportation Systems*, 2017 (submitted).

[88] Jostein Vada, Olav Slupphaug, Tor A. Johansen, and Bjarne A. Foss. Linear {MPC} with optimal prioritized infeasibility handling: application, computational issues and stability. *Automatica*, 37(11):1835 – 1843, 2001.

[89] Arthur Richards, Tom Schouwenaars, Jonathan P How, and Eric Feron. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002.

[90] Andres Hernandez, Harold Murcia, Cosmin Copot, and Robin De Keyser. Towards the development of a smart flying sensor: Illustration in the field of precision agriculture. *Sensors*, 15(7):16688–16709, 2015.

[91] Milad Jalali, Amir Khajepour, Shih-ken Chen, and Bakhtiar Litkouhi. Handling delays in yaw rate control of electric vehicles using model predictive control with experimental verification. *Journal of Dynamic Systems, Measurement, and Control*, 139(12):121001, 2017.

[92] Yinlong Hu, Michael ZQ Chen, and Zhongsheng Hou. Multiplexed model predictive control for active vehicle suspensions. *International Journal of Control*, 88(2):347–363, 2015.

[93] Daniel Martins Lima, Tito Luís Maia Santos, and Julio Elias Normey-Rico. Robust nonlinear predictor for dead-time systems with input nonlinearities. *Journal of Process Control*, 27:1–14, 2015.

[94] Galina Mirzaeva, Graham Clifford Goodwin, Brendan P McGrath, Carlos Teixeira, and Marco E Rivera. A generalized mpc framework for the design and comparison of vsi current controllers. *IEEE Transactions on Industrial Electronics*, 63(9):5816–5826, 2016.