Model Predictive Control of Highway Emergency Maneuvering and Collision Avoidance

by

Mohammadmehdi Jalalmaab

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	Professor Ya-Jun Pan
Supervisor(s)	Associate Professor Baris Fidan Associate Professor Soo Jeon
Internal Members	Professor Jan P. Huissoon Associate Professor Hyock Ju Kwon
Internal-external Member	Associate Professor Nasser Lashgarian Azad

Author’s Declaration

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

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

Abstract

Autonomous emergency maneuvering (AEM) is an active safety system that automates safe maneuvers to avoid imminent collision, particularly in highway driving situations. Uncertainty about the surrounding vehicles' decisions and also about the road condition, which has significant effects on the vehicle's maneuverability, makes it challenging to implement the AEM strategy in practice. With the rise of vehicular networks and connected vehicles, vehicles would be able to share their perception and also intentions with other cars. Therefore, cooperative AEM can incorporate surrounding vehicles' decisions and perceptions in order to improve vehicles' predictions and estimations and thereby provide better decisions for emergency maneuvering.

In this thesis, we develop an adaptive, cooperative motion planning scheme for emergency maneuvering, based on the model predictive control (MPC) approach, for vehicles within a vehicular network. The proposed emergency maneuver planning scheme finds the best combination of longitudinal and lateral maneuvers to avoid imminent collision with surrounding vehicles and obstacles. To implement real-time MPC for the non-convex problem of collision free motion planning, safety constraints are suggested to be convexified based on the road geometry. To take advantage of vehicular communication, the surrounding vehicles' decisions are incorporated in the prediction model to improve the motion planning results.

The MPC approach is prone to loss of feasibility due to the limited prediction horizon for decision-making. For the autonomous vehicle motion planning problem, many of detected obstacles, which are beyond the prediction horizon, cannot be considered in the instantaneous decisions, and late consideration of them may cause infeasibility. The conditions that guarantee persistent feasibility of a model predictive motion planning scheme are studied in this thesis. Maintaining the system's states in a control invariant set of the system guarantees the persistent feasibility of the corresponding MPC scheme. Specifically, we present two approaches to compute control invariant sets of the motion planning problem; the linearized convexified approach and the brute-force approach. The resulting computed control invariant sets of these two approaches are compared with each other to demonstrate the performance of the proposed algorithm.

Time-variation of the road condition affects the vehicle dynamics and constraints. Therefore, it necessitates the on-line identification of the road friction parameter and implementation of an adaptive emergency maneuver motion planning scheme. In this thesis, we investigate cooperative road condition estimation in order to improve collision avoidance performance of the AEM system. Each vehicle estimates the road condition individually, and disseminates it through the vehicular network. Accordingly, a consensus estimation algorithm fuses the individual estimates to find the maximum likelihood estimate of the road condition parameter. The performance of the proposed cooperative road condition estimation has been validated through simulations.

Acknowledgements

First, I would like to express my thanks to my supervisors Prof. Baris Fidan and Prof. Soo Jeon for their endless support of my Ph.D. study. I would also like to thank Prof. Ya-Jun Pan, Prof. Jan P. Huissoon, Prof. Nasser Lashgarian Azad and Prof. Hyock Ju Kwon for taking time to be in my committee and guide me with their helpful comments.

In addition, I would like to thank Prof. Paolo Falcone and Prof. Steven Waslander for their support and guidance during my Ph.D. study. I would like to thank to NSERC and Nuvation Inc., the industrial partner in the NSERC CRD project "Autonomous Driving Strategies for Urban and Highway Environments," which financially supported this Ph.D. thesis study. Finally, I would like to extend my sincere thanks to Feyyaz Emre Sancar for being a great teammate in all works we have collaborated on.

Dedication

To my parents, my wife, and my son

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human driving is prone to accident due to the driver's misjudgement, tiredness, panic, etc. Statistics clearly show that more than 90 percent of all vehicle accidents are due to human mistakes [97]. Autonomous driving is a new technology trend to achieve more comfort and safety, by sensing the vehicle's surrounding environment and navigating through it without human intervention. With recent advances in artificial intelligence, vehicular sensors, communication, and control technologies, the problem of safe autonomous driving for passenger cars is highly focused among industrial and academic researchers, now.

Autonomous vehicles are typically equipped with active safety systems such as adaptive cruise control (ACC), autonomous emergency braking (AEB), lane keeping systems (LKS), etc., to provide collision-free driving at different driving situations. Obviously, the collision avoidance approach and policy are different for highways than for other driving situations and environments such as residential areas, intersections, and off-road driving. The highway collision avoidance problem is concerned about driving in high-speed, multi-lane roads with numerous surrounding vehicles, which may form platoons together to drive more efficiently. Autonomous emergency maneuvering (AEM) is an active safety system that can be used for highway autonomous driving and other situations, to determine and follow safe emergency maneuvers in critical situations, as well as avoid imminent collisions. These emergency maneuvers may include braking, lane shifting, or combination of them.

Vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication can provide a more effective decision-making framework for AEM and other collision avoidance systems. In a vehicular network, each vehicle ideally has access to surrounding vehicles' states (position, heading angle, speed, etc.) and decisions (intended steering and acceleration commands). Communication provides information to each vehicle about the objects that are located outside the range of in-vehicle sensors. Additionally, having access to surrounding vehicles' decisions enables the controller to predict the trajectories of the neighbouring vehicles more accurately. Altogether, vehicles in the network can perform the planning tasks cooperatively. Centralized or leader-based cooperative control techniques impose certain requirements on many driving situations. In particular, low penetration rate of autonomous vehicles, non-homogeneous autonomous controlling systems, and uncertainty about the quality of communication service are some of the barriers for the design and implementation of centralized cooperative collision avoidance. Therefore, decentralized cooperative control schemes are considered more practical and feasible for autonomous collision avoidance systems.

System identification is an important part of control schemes for plants with unknown parameters. Maximum road friction coefficient or *road condition parameter* is an unknown time-varying parameter that plays a critical role in motion planning for emergency maneuvering tasks. Therefore, real-time estimation of this parameter is essential for the development of control strategy for emergency maneuvering. Parameter identification techniques, to provide a reliable estimation, require persistently exciting measurements with limited noise variance, which may not always be possible in real-life applications. In the presence of a vehicular network, which is established by V2V or V2I communication technologies, cooperative estimation techniques such as consensus-based ones can improve estimation accuracy. Moreover, storing estimation data from other agents in the network can be used for generating a look-up table of road condition values along the road to predict upcoming road condition, for example, to be better prepared for low friction areas.

Model predictive control (MPC) is a well-studied optimal control approach for real-time applications and thus is a proper candidate for AEM planning and control. Despite many advantages, the MPC technique has some implementation issues that should be addressed for practical AEM application. Linear dynamics with convex constraints are usually preferred for MPC problem formulation due to the computational efficiency of the optimal control solution, but vehicular maneuver planning is essentially a non-convex constrained problem with non-holonomic and non-linear dynamics. Additionally, MPC problems may encounter feasibility issues due to the limited foresight in comparison to infinite horizon optimal control approaches. As a result, a feasible trajectory may not be found by the controller during the operation of the vehicle. These challenges should be addressed for the obstacle avoidance control system.

1.1 Contributions

In this thesis, we develop a cooperative AEM control strategy and study its performance in different scenarios and road conditions. In brief, the main contributions of this thesis are as follows:

(i) Convexification of the cooperative MPC problem for emergency maneuvering: The allowable domain of maneuver planning is non-convex due to the presence of obstacles and surrounding vehicles. Non-convex constraints are not easy to handle. Convexification of the domain of the road structure can expedite the computation speed of the MPC solution. A new MPC convexification methodology is developed for multi-lane roads and multiple obstacles in this thesis (Chapter 3) [40].

(ii) Determination of persistent feasibility conditions for the model predictive emergency maneuvering motion planning problem with obstacles: The MPC provides a fast suboptimal solution by considering only a finite time horizon for the original optimal control problem. However, it may cause feasibility issues due to ignoring the constraints beyond the prediction horizon. This thesis proposes a set of persistent feasibility criteria for autonomous driving (Chapter 4) [41].

(iii) Developing a consensus-based cooperative road condition estimation scheme to improve the MPC decisions: On-line parameter identification suffers from measurement noise. Within a vehicular network, sharing the estimation outcome of a same parameter would help each agent to have better identification. Road friction coefficient is an unknown time-varying parameter that has significant effects on motion prediction and control constraints. A cooperative framework for road condition estimation is proposed based on a consensus algorithm (Chapter 5) [42, 43].

(iv) Integration of cooperative estimation and emergency maneuver planner: The proposed cooperative motion planning and road condition estimation schemes is integrated. This study provides a new formulation of the emergency maneuver motion planning that is adaptive to road condition variations. It has been shown that the cooperatively estimated road condition leads to proper motion planning while individual estimates may not be accurate enough for collision avoidance (Chapter 5) [43].

Chapter 2

Literature Review and Background

2.1 Highway Emergency Maneuvering and Collision Avoidance Problem

The first vehicular collision avoidance systems were developed around mid-nineties [79]; while it lasted about a decade to become notably available in market by the major manufacturers [93,102]. The primitive collision avoidance systems were mostly radar-based and generated only warning messages to drivers in dangerous situations. Later, these systems were upgraded with autonomous partial/full braking and various sets of sensors including cameras, laser scanners, and radars to avoid or mitigate the collision if the driver did not react to impending collision situations [95]. Today, active safety systems such as AEB and ACC are mature collision avoidance systems that provide safety and comfort for high-end passenger cars [17,89]. The successful performance of these collision avoidance systems is dependent on several factors including correct detection of static and dynamic obstacles, robust situation assessment and threat analysis algorithms, feasible trajectory planning approach, and effective actuators control system [2,17,36].

Emergency maneuvering that considers both braking and steering maneuvers to avoid collision is an advanced technology investigated in the last few years [20,23,26,32,45,98]. Safe longitudinal collision avoidance requires large enough *relative stopping distance* with respect to the front vehicle/obstacle. Relative stopping distance increases by the relative velocity with respect to the front vehicle/obstacle. Therefore, at situations where the relative distance of two objects are not sufficiently large, lane shifting with lateral maneuvering is an alternative solution for collision avoidance [20]. Highway driving is one of the situations that emergency maneuvering would be more useful due to the high speed of the vehicles and probable availability of a free lane to shift. While longitudinal collision avoidance systems such as ACC and AEB were broadly developed and used in commercial products, two dimensional collision avoidance systems with emergency maneuvering are still in the predevelopment stage. Therefore, further investigation is needed for the development of integrated lateral and longitudinal collision avoidance systems [20].

Most of today's collision avoidance systems are developed for semi-autonomous vehicles and referred as advanced driver assist systems (ADAS). These systems are intended to work in parallel with the driver to assist him or her in dangerous situations by warning or proper actuation. An ADAS consists of many safety and comfort subsystems such as driver monitoring system, blind spot monitor, intersection assist, collision avoidance system, lane keeping system and lane change assist, ACC, and AEB. These subsystems collaborate with each other to deliver a safe drive for the passengers. However, the driver is still in the driving control loop and high-level trajectory planning are mostly instructed by the driver [65].

Emergency maneuvering and collision avoidance systems for fully autonomous vehicles require a new design framework without any human intervention. Although the development of autonomous driving technologies has been investigated by many research facilities from the nineties [99, 100], there has been no fully autonomous car available in the market yet. Meanwhile, DARPA urban challenge 2007 [44, 74, 101, 106], Google's driverless car project [105], and the VisLab intercontinental autonomous challenge 2010 [8], have demonstrated the feasibility of full autonomous driving in structured and unstructured off-road environments.

Autonomous collision free driving requires completely functional perception, planning, and control systems to continuously operate during the driving. Perception and estimation systems gather necessary information and process them to achieve a reliable understanding of the driving situation [3,4,15,19,24,27,37,49,55,58–60,64,68,69,73,84,85,103,104]. Planning and control are responsible to determine the optimal high-level path to the destination, generate collision-free low-level trajectories based on the perceptional information [53,54,114,115], and determine optimal and effective control inputs for the actuators, i.e. steering, throttle, and brakes, to follow the desired trajectory successfully [66]. In this thesis, we discuss the most important techniques and challenges for perception, planning, and control, from the control engineering perspective, for safe autonomous driving, in 2.1.1 and 2.1.2 respectively.

V2V and V2I communication technologies have been developed and standardized in the last decade to enable autonomous and semi-autonomous vehicles to share their information within the vehicular network. Autonomous vehicles are equipped with communication devices such as dedicated short range communication (DSRC) to have reliable communication with short latency [112]. Vehicular network improves perception and planning approaches [38]. Therefore, cooperative estimation and planning techniques are covered in 2.1.1 and 2.1.2, as well.

2.1.1 Perception and Estimation

Semi- and full-autonomous vehicles need to be aware of surrounding objects, detect hazardous situations, and predict probable collisions using perceptional techniques and algorithms. Perception unit may contain algorithms for localization and states estimation [60, 73, 85], surrounding vehicles and obstacles detection and tracking [49,69], lanes and road boundaries and road condition estimation [64, 68], traffic lights and signs recognition [19, 24, 59], pedestrians and cyclists detection [3, 15], and prediction of moving obstacles' motion in oncoming seconds with proper level of uncertainty [4,27,37,55,58,84,103,104].

The majority of perceptional features are provided by signal and image processing techniques that are outside the scope of this study. However, road condition estimation is a perception task that is relevant to this study, and therefore covered in this thesis. Specifically, incorporation of the time-varying road condition in the collision avoidance control problem has been identified as an appealing indirect adaptive control problem and is addressed in Section 2.3 as a part of this study.

Vehicular network can be used to improve the performance of the perception and estimation. Besides the ability of vehicular network to share the vehicles' states and characteristics, such as position, heading, speed, and size of the vehicle, the agents may act as a *sensor network* and broadcast their perception and estimation about the driving environment, e.g. road geometry and friction, object detection, and hazard identification. Online road condition estimation by a single agent may not be reliable enough for critical problems of collision avoidance control, particularly in circumstances in which measurement noise is large and sensor malfunction is probable. Therefore, cooperative estimation can be used to minimize the road condition estimation error and eliminate the effect of a faulty agent in the network. This problem is addressed in 2.3 in more detail.

2.1.2 Planning and Control

Planning is a decision-making process that the autonomous vehicle uses to reach its goals. Recent advances in perception technologies and on-board computational capabilities have been contributing to the development of new planning algorithms for autonomous vehicles, but it is still an open problem for autonomous driving due to the complexity of driving environment.

Planning systems are responsible for different levels of decision-making, from high-level strategies to low-level tactical ones. High-level navigation, alternatively called mission planning or global navigation, computes fastest or shortest routes, using environmental data such as preloaded street and geographical maps and a cost function associated with various factors such as travel time, travel distance, etc. It would be updated during the driving if new information about the traffic and road closures are received, which may not occur frequently. On the other hand, the low level motion planning and obstacle avoidance, use the real-time sensor data, to determine safe and feasible trajectories among surrounding obstacles and vehicles. This layer of planning performs tactical decisions for determining proper driving lane, speed, distance to front and rear vehicle, time to brake or shift the lane, etc., to follow the high-level navigation waypoints successfully. Contrary to high-level navigation system, this layer requires high frequency updating rate and high precision sensor data. While static obstacles such as buildings and one-way roads would be handled in high-level long-term navigation, low-level collision avoidance planning is concerned about dynamic obstacles and unexpected events over the short term of driving [6, 66, 96].

A typical driving trip may contain various navigational situations that require different motion planning approaches. In unstructured situations, such as off-road driving and driving in areas like parking zones, there may be no specific road boundaries, and static terrains and obstacles mainly form the feasible routes and trajectories. There have been many heuristic approaches that provide feasible motion planning for such situations [7, 56, 86, 117]. Structured environments, including highways, intersections, and residential areas involve moving traffic, lanes, and road boundaries and driving regulations. Here, smoothness, dynamics, optimality, and the ability to treat moving obstacles become more important [114]. A *behavioural executive* scheme should be utilized between high-level and low-level planning algorithms to choose a proper tactical planning and collision avoidance approach, e.g. the intersection planner, the highway planner, the parking zone planner, etc.

This study is focused on highway motion planning, collision avoidance, and emergency maneuvering techniques. Therefore, the study on the planning is limited to the tactical layer. The high-level navigation and behaviour executive algorithms are out of the scope of this study. As it was mentioned before, tactical motion planning requires high frequency updating rate to be able to react properly to surrounding moving obstacles. In particular, highway collision avoidance and emergency maneuver planning should be equipped with a reliable real-time planning scheme.

Similar to unstructured environment planning, a wide range of heuristic and combinatorial planning approaches have been suggested for highway trajectory planning in the literature. Some of them use graph-based *state lattice* [52, 71, 81, 87, 90, 109, 116] or spline-based search tree [22] techniques to take into account kinematic and dynamic constraints of the problem. The state lattice space may include the position, the orientation and the steering angle of a vehicle to form a graph, edges of which represent the trajectories [52, 81, 87]. To find the optimal constrained trajectory, any real-time graph search approaches such as pre-computed lookup tables [90, 109] can be used. Using Frenet frame attached to a reference curve like the center of a lane is suggested in [71] to avoid oscillations between two orientation samples caused by the discretization of heading angles in state lattice lookup table.

Recently, large attention has been paid to numerical optimization approaches due to the onboard computation advances. Model predictive control (MPC) is a planning and control technique based on numerical optimization, which incorporates the vehicle's motion model and constraints, to propagate the states over a prediction horizon and find the optimal decisions [1, 2, 11–13, 16, 25, 28, 40, 76, 91]. Obstacle avoidance objectives can be formulated as state constraints for minimum allowable relative distance, speed, etc., [40, 76] or the cost function term that increases when the obstacles are in close proximity [107, 108]. The parametric control models like curvature polynomials are known to reduce the solution space by allowing fast planning, but the resulting solution may be sub-optimal [5, 46, 106].

2.2 Highway Emergency Maneuvering MPC

2.2.1 Main Concept

MPC, alternatively known as the receding horizon control (RHC) or the embedded optimal control (EOC) in the literature, is an advanced optimal control strategy that has been widely used in the process industries since the 1980s. Recent advances in theory and computation algorithms have enlarged the range of applications of MPC. MPC relies on the system's dynamic model to predict the propagation of the states in the future, and take control actions accordingly. The main advantage of MPC is the fact that it allows the optimization of a performance index under the operating constraints, while taking the system's future states into account. This is achieved by solving a constrained finite-time optimal control (CFTOC) problem over the prediction horizon, to find a sequence of optimal control inputs while implementing only the control signal for the current time step. In the next time step, a new optimization problem is solved over a shifted prediction horizon and this process repeats at each sample time. The capability of handling hard constraints in an explicit way makes MPC a very attractive control technique, especially for applications where it is required to handle the plant close to the boundary of the set of admissible states and inputs.

The basis for the MPC problem is a dynamic model that describes the evolution of the states with time:

$$x[k+1] = g(x[k], u[k]), \tag{2.1}$$

where $g(x, u)$ is a function that can be linear, nonlinear, discrete or continuous, etc., and $x[k]$ and $u[k]$ are the state and input vectors, respectively, subject to the constraints:

$$x[k] \in \mathcal{X}[k] \subseteq \mathbb{R}^n, \quad u[k] \in \mathcal{U}[k] \subseteq \mathbb{R}^m, \quad \forall k \geq 0. \tag{2.2}$$

where $\mathcal{X}[k]$ and $\mathcal{U}[k]$ are the states and inputs admissible regions. The goal of MPC at time k is to find the input vector $U[k \to k + N_p|k] = \left[u[k|k]^T, ..., u[k + N_p - 1|k]^T\right]^T$ over a prediction horizon N_p, that optimizes a given cost function involving the states and input vectors:

$$J_k = p(x[k + N_p|k]) + \sum_{j=0}^{N_p-1} q(x[k + j|k], u[k + j|k]) \tag{2.3}$$

where the terms $q(x, u)$ and and $p(x)$ are referred as stage cost and terminal cost, respectively, $x[k+j|k]$ is the state vector of time step $k+j$ that predicted by simulating the system (2.1) from initial conditions at time k. The problem formulation for MPC can be summarized as [10]:

$$J_k^*(x[k]) = \min_{U[k \to k+N_p|k]} J_k(x[k], U[k \to k+N_p|k])$$

subj. to

$$x[k+j+1|k] = g(x[k+j|k], u[k+j|k]) \tag{2.4a}$$

$$x[k+j|k] \in \mathcal{X}, \quad u[k+j|k] \in \mathcal{U}, \quad j = 0, ..., N_p - 1 \tag{2.4b}$$

$$x[k+N_p|k] \in \mathcal{X}_f \tag{2.4c}$$

$$x[k|k] = x[k] \tag{2.4d}$$

where (2.4c) specifies the terminal states constraint, $\mathcal{X}_f$ is the terminal admissible set and (2.4d) defines the initial condition. The cost function optimization is solved in each sampling time and the first column of the resulting input vector, $u[k] = u^*[k|k]$ is applied to the system. At the next time step, the optimal input vector is computed based on the new states measurement over a shifted horizon.

MPC design and implementation require decision-making about the specifications of the controller structure including cost function, prediction horizon, terminal set, etc. The squared 2-norm cost function is employed more often than the 1- or ∞-norm cost functions since 1- or ∞-norm formulations make it difficult to formulate the optimal control problem and to choose the weights [10]. The longer horizon N_p provides a longer prediction span, making the solution closer to that of the infinite-horizon optimal control problem. On the other hand, the larger prediction horizon increases the computational effort and would be a barrier for real-time implementation.

Mainly, there are two possible approaches available for the implementation of MPC. In the first approach, a mathematical program is solved at each sampling time for the current initial states. In the second approach, the explicit piecewise affine feedback policy (that provides the optimal control for all states) is precomputed off-line. This reduces the on-line computation of the MPC law to a function evaluation, thus avoiding the on-line solution of a mathematical program.

Major Technical Issues

For the special case of infinite horizon MPC, if the optimization problem is feasible at the beginning, then the closed-loop trajectories will be feasible for all times. When MPC optimization problem is solved over a finite horizon repeatedly at each time step, at least two problems may occur. First of all, the controller may lead to a situation where after a few steps the finite horizon optimal control problem becomes infeasible, i.e. there does not exist a sequence of control inputs for which the constraints are obeyed. Second, even if the feasibility problem does not occur, the generated control inputs may not lead to stable trajectories.

In general, stability and feasibility are not ensured by the MPC law (2.4) and are difficult to analyze. One way to handle the stability and persistent feasibility for MPC is to impose some conditions on the terminal constraint set $\mathcal{X}_f$ such that closed-loop stability and feasibility are ensured. Such conditions are reviewed in 2.2.3. Then, the implementation of these conditions for the collision avoidance problem is studied in Chapter 4.

2.2.2 Control Design

To design a reliable MPC algorithm for emergency maneuver planning, it is necessary to have simple and accurate control-oriented model and admissible regions for states and inputs. After that, we can make a decision on prediction horizon, cost function parameters, and terminal states region.

System Modeling

The challenge in choosing models suitable for control design is the trade-off between accuracy and complexity. Models that accurately capture the dynamics of the system tend to yield computationally intensive algorithms. Vehicle models can be broadly classified into three categories:

1. Point-mass models treat the vehicle as a particle that yields large tracking errors when used for path planning due to their inability to account for dynamic feasibility [30].

2. Kinematic models are a function of vehicle geometry, and can represent the vehicle motion in a range of conditions that does not involve highly dynamic maneuvers and/or tire force saturation [50].

3. Dynamic models rely on tire models to describe the interaction between the vehicle and the road. In this case, the complexity arises from the non-linear relationship between the tire forces and the vehicle states and inputs. Bicycle models, wherein the left and right wheels are represented by a single wheel, are a common approach in developing models suitable for control design [88].

Besides the host vehicle (the controlled vehicle) dynamic model, it is necessary to model the surrounding obstacles' and vehicles' motion to predict their trajectory in order to plan safe trajectories. While the simpler trajectory prediction approaches use kinematic or dynamic vehicle models with some assumptions on the control inputs, maneuver-based or interaction-aware models should be employed for longer trajectory predictions. Maneuver-based models use collection of basic maneuvers to represent various maneuvers that the vehicle can perform. Interaction-aware models improve their predictions by taking into account the interaction between vehicles. An overview of existing approaches of vehicle motion modelling for trajectory prediction can be found in [58].

Constraints

Physical limitations on the actuators impose bounds on the control inputs and input change rates. Additionally, the objective of avoiding surrounding obstacles and road departure can be expressed as constraints on the vehicle's states. These constraints may make the admissible region non-convex and non-differentiable. This increases the complexity of the resulting on-line program.

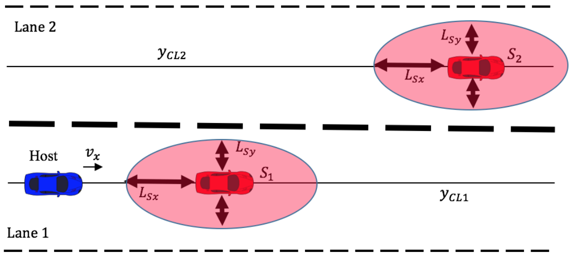


Figure 2.1: Safety constraints over surrounding vehicles.

Various approaches have been proposed in the literature for formulating the collision avoidance constraints. Gao et al. [29] described obstacles as ellipses to have smooth and differentiable constraints (see Fig. 2.1). Polytopic descriptions of obstacles require mixed-integer programming [9, 94], which leads to prohibitive computational complexity and prevents the real-time execution of the trajectory planning algorithm.

As depicted in Fig. 2.1, safety constraints may lead to non-convexity in the state space. Non-convexity of the problem would be a burden for real-time implementation of the planner scheme, since it demands higher computation cost. For structured environments such as highways, a simplified approach based on sampling the boundaries of the safe region on the road is proposed by Anderson et al. [2] and Carvalho et al. [13]. Anderson et al. defined a locally convex driving corridor by a lateral position constraint vector over the prediction horizon by use of sensor data and estimated behaviour of hazards and the vehicle [2]. Convexification of the collision avoidance constraints for a particular type of road geometry, i.e. one-way, two-lane roads, is investigated in [76]; the collision avoidance constraints are formulated as linear combinations of the vehicle states and inputs, to provide a convex linear constraint. This constraint formulation eliminates the need of mixed-integer inequalities and the optimization problem results in a standard quadratic program.

2.2.3 Feasibility and Persistent Feasibility

MPC is a finite-time horizon optimal control problem, therefore the solution it produces may lead to infeasibility at future time instances, which means there is no solution for the problem any more [10, 31, 61, 63]. It is always desired to make MPC persistently feasible. Necessary conditions to guarantee persistent feasibility are discussed in [10, 31, 63]. It is shown that if the MPC final set $\mathcal{X}_f$ is control invariant, then the controller is persistently feasible. Alternatively, persistent feasibility can be achieved if the prediction horizon of the controller is larger than a specific value, the determinedness index of the situation [10]. In [31, 63], the initial states that may cause infeasibility in subsequent time steps are presented. In [61], a sufficient condition for recursive feasibility of nonlinear stochastic MPC for semi-autonomous driving is presented based on a technique introduced in [51]. Interestingly, persistent feasibility directly leads to Lyapunov stability of the MPC schemes [70]. Thus, persistent feasibility conditions are also considered as the stability conditions for the MPC.

To the best of the author's knowledge, the implementation of the persistent feasibility for autonomous emergency maneuver planner systems has not been studied in the literature. Without persistent feasibility conditions, a model predictive planner may ignore obstacles beyond the prediction horizon and generate trajectories that are optimal at the time but become infeasible at a later time. Assume the situation where the vehicle is driving at a high speed, the prediction horizon is relatively short and a fully stopped obstacle exists in the vehicle's lane (Fig. 2.2). If the planner is not aware of the stopped vehicle until the prediction horizon recedes, a collision may be inevitable. Persistent feasibility conditions enforce the planning scheme to be prepared for upcoming, out-of-horizon events.

As preliminaries, basic concepts of invariant set theory are explained in the following. Subsequently, existing theoretical results to guarantee persistent feasibility of a controlled system (2.1)–(2.2) are presented.

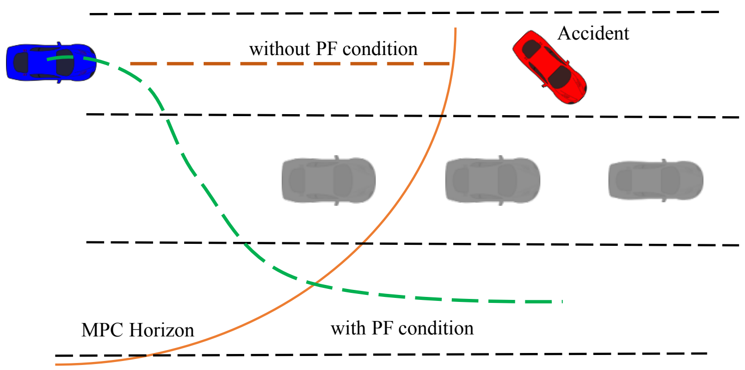


Figure 2.2: A driving scenario with unfeasibility issue in near future.

Invariant Set Theory

The following definitions, algorithm, theorem, and corollary are based on the set theory presented in [10], and are stated for the system definition (2.1), (2.2), with the allowable state set $\mathcal{X}$ and the allowable input set $\mathcal{U}$.

Definition 1. *One-step controllable set* for (2.1), (2.2) to a given target set $S \subseteq \mathcal{X}$ is the set of all states that can be propagated to S in one time step by an admissible input $u \in \mathcal{U}$:

$$\mathcal{K}_1(S) = \{x \in \mathbb{R}^n : \exists u \in \mathcal{U} \ \ s.t. \ \ g(x,u) \in S\}. \tag{2.5}$$

Definition 2. *N-step controllable set* $\mathcal{K}_N(S)$ for $N = 2, 3, ...$ is defined recursively:

$$\mathcal{K}_j(S) = Pre(\mathcal{K}_{j-1}(S)) \cap \mathcal{X}. \tag{2.6}$$

Definition 3. *One-step reachable set* for (2.1), (2.2) from given set $S \subseteq \mathcal{X}$, is denoted with $\mathcal{R}_1(S)$, and is defined as the set of all states that can be reached from S in one step by an admissible input $u \in \mathcal{U}$:

$$\begin{aligned}&\mathcal{R}_1(S) = \{x[k+1] \in \mathbb{R}^n : \\ &\exists x[k] \in S, \exists u[k] \in \mathcal{U} \ \ s.t. \ \ x[k+1] = g(x[k], u[k])\}.\end{aligned} \tag{2.7}$$

Definition 4. *N-step reachable set* $\mathcal{R}_N(S)$ is defined recursively:

$$\mathcal{R}_j(S) = Reach(\mathcal{R}_{j-1}(S)). \tag{2.8}$$

In the literature, *controllable set* is also named as *backward reachable set*, while *reachable set* is called *forward reachable set*. Interestingly, reachable and controllable set calculation for convex sets are fast and computationally efficient.

Definition 5. *Maximal controllable set* $\mathcal{K}_\infty(S)$ for a target set $S \subseteq \mathcal{X}$ is the union of N-step controllable sets $\mathcal{K}_N(S)$, for all $N = 1, 2, ...$, contained in $\mathcal{X}$.

Definition 6. A set $C \subseteq \mathcal{X}$ is said to be a *control invariant set* if there are admissible inputs for all future time steps to keep the states inside S:

$$x(k) \in C \Rightarrow \exists u(k) \in \mathcal{U} \ \ s.t. \ \ g(x[k], u[k]) \in C, \ \ \forall k \in \mathbb{N}_+. \tag{2.9}$$

Definition 7. The set $C_\infty \subseteq \mathcal{X}$ is said to be the *maximal control invariant set*, if it is control invariant set and contains all control invariant sets contained in $\mathcal{X}$. Computation procedure of C_∞ is demonstrated in Algorithm 2.1.

Definition 8. Considering Algorithm 2.1, the set C_∞ is *finitely determined* if and only if $\exists\, k \in \mathbb{N}$ such that $\Omega_{k+1} = \Omega_k$. The smallest element $d_i \in \mathbb{N}$ such that $\Omega_{d_i+1} = \Omega_i$ is called the *determinedness index*.

Definition 9. Persistent Feasibility is a property for MPC law that guarantees the feasibility of MPC for all future time, assuming the feasibility of the initial set $\mathcal{X}_0$.

Theorem 2.1. *Considering an MPC law with $N \geq 1$, if $\mathcal{X}_f$ is a control invariant set for the control system, then the MPC is persistently feasible.*

Corollary 2.1. *If the prediction horizon of the MPC law is greater than the determinedness index of $\mathcal{K}_\infty(\mathcal{X}_f)$, then the MPC problem is persistently feasible.*

Based on Theorem 2.1, persistent feasibility can be achieved by proper selection of $\mathcal{X}_f$. In another way, Corollary 2.1 presents a condition for the prediction horizon N_p of the MPC to achieve persistent feasibility.

Algorithm 2.1 Computation of C_∞

```
input: $g$, $\mathcal{X}$ and $\mathcal{U}$
output: $C_\infty$

let $\Omega_0 \leftarrow \mathcal{X}$
repeat
$k = k + 1$
$\Omega_{k+1} \leftarrow \mathcal{K}_1(\Omega_k) \cap \Omega_k$
until $\Omega_{k+1} = \Omega_k$
$C_\infty \leftarrow \Omega_{k+1}$
```

2.3 Road Condition Estimation for Emergency Maneuvering

Road condition dramatically affects the maneuverability of the autonomous vehicle. The proper steering and acceleration inputs for snowy roads would be considerably different from those for dry roads. In emergency situations, it may be necessary to employ maximum available acceleration and steering without losing control. Therefore, road condition determination is crucial for emergency maneuver planning and it would help to determine the real bounds of actuators. In its simplest form, the road condition can be represented by the maximum road-tire fiction coefficient in the vehicle's equations of motion and control law.

In typical passenger cars, there is no instrument available to measure the maximum road friction force, and therefore relying on estimation approaches is inevitable. On-line parameter identification (PI) is a class of estimation approaches that employs a parametric model of the plant, adaptive estimation law and measurement data to regulate the estimation error of the unknown parameters [14, 33, 34, 57]. A general recursive least square online PI approach is explained in 2.3.1. In the presence of the measurement noise and potential for sensor malfunction, parameter identification by a single agent may not be reliable for critical problems of collision avoidance control. Additionally, parameter identification convergence requires persistent excitation of the system [39], which may not be available in many instances.

By communication technologies, connected vehicles can share their states and characteristics, e.g. position, heading, speed, and size of the vehicle. Additionally, the agents may act as a *sensor network* and disseminate their perception and estimation about the driving environment like road's geometry, friction, traffic light's state, etc. Cooperative estimation can play an important role to minimize the estimation error and eliminate the effect of a faulty agent in the network.

With the establishment of a vehicular sensor network, a sensor fusion scheme would be required to estimate the unknown parameters, cooperatively. In the networks with central data fusion topology, each agent sends its data to the fusion center, where the cooperative estimation is computed. Road-side units (RSU) can provide centralized topology for the vehicular networks, but the availability of RSU is not always guaranteed and the vehicle network should be robust to situations with no fusion center. Therefore, decentralized sensor fusion schemes will be advantageous. In this type of data fusion, each agent exchanges data only with its neighbours and computes the local average [111].

Consensus estimation algorithm is a decentralized cooperative estimation approach to reach an agreement regarding a certain parameter that is measured by all agents [72, 77, 111]. It specifies how the information should be exchanged between an agent and its neighbours in a network, to converge to the agreement efficiently [77]. Cooperative or multi-agent estimation within the vehicular network has been studied for applications such as traffic flow estimation [75] and vehicle positions [21, 62, 82]. Cooperative road condition estimation can increase the reliability of the estimation and enable the vehicles to predict the road condition for the future time steps. The general form of the consensus algorithm is explained in 2.3.2.

2.3.1 Single Agent Estimation

In this part, a widely used online parameter identification approach, recursive least square and its convergence criteria are presented. The basic idea behind recursive least square parameter identification (LSPI) is fitting a mathematical model to a sequence of observed data by minimizing the square of the difference between the observed and computed data. This method is simple to apply and analyze in the case where the unknown parameters appear in a linear form of parametric model [39].

The first step for parameter identification is to form the parametric model modified from the system equation in such a way that the unknown parameters are lumped in one side separated from known signals. Linear static parametric model (SPM) is in the following form:

$$f(t) = \theta^*(t)\phi(t), \tag{2.10}$$

where θ^* is the unknown scalar parameter to be identified and $f(t)$ and $\phi(t)$ are known signals of the system. The estimation model takes the same form as the parametric model with θ^* replaced by θ_{PI}:

$$\hat{f}(t) = \theta_{PI}(t)\phi(t), \tag{2.11}$$

Accordingly, the normalized estimation error is formulated as the normalized difference between these two signals:

$$\varepsilon(t) = \frac{f(t) - \hat{f}(t)}{m_s^2(t)} \tag{2.12}$$

$$m_s^2(t) = 1 + \phi^T(t)\phi(t) \tag{2.13}$$

where $\varepsilon(t)$ is estimation error and $m_s^2(t)$ is the normalization signal. Then, the recursive least square algorithm with forgetting factor can be formulated as an adaptive law to regulate the estimation error as follows [39]:

$$\dot{\theta}_{PI}(t) = P(t)\varepsilon(t)\phi(t), \tag{2.14}$$

$$\dot{P}(t) = \beta P(t) - P(t)\frac{\phi(t)\phi^T(t)}{m_s^2(t)}P(t), \tag{2.15}$$

where $\theta_{PI}(0) = \theta_{PI0}$ and $P(0) = P_0 = Q_0^{-1}$. According to Theorem 3.7.1 in Ref. [39], if ϕ/m_s is *persistently exited*, then the recursive least-square algorithm with forgetting factor guarantees that $\theta_{PI}(t) \to \theta^*$ as $t \to \infty$. The convergence is exponential if $\beta \geq 0$.

For discrete model of:

$$f[k] = \theta^*[k]\phi[k], \qquad \text{parametric model} \tag{2.16}$$

$$\hat{f}[k] = \theta_{PI}[k-1]\phi[k], \qquad \text{estimation model} \tag{2.17}$$

$$\varepsilon[k] = \frac{f[k] - \hat{f}[k]}{m_s^2[k]}, \qquad \text{estimation error} \tag{2.18}$$

The recursive least square algorithm is given by:

$$P[k] = P[k-1] - \frac{P[k-1]\phi[k]\phi^T[k]P[k-1]}{m^2[k] + \phi^T[k]P[k-1]\phi[k]}, \tag{2.19}$$

$$\theta_{PI}[k] = \theta_{PI}[k-1] + P[k]\phi[k]\varepsilon[k], \tag{2.20}$$

where $P[0] = P_0 = P_0^T > 0$ and $\theta_{PI}[0] = \theta_{PI0}$. Based on the Theorem 4.6.1 of [39], by using this least square algorithm if $\frac{\theta[k]}{m[k]}$ is persistently excited, then $\theta_{PI}[k] \to \theta_{PI}^*$ as $k \to \infty$.

2.3.2 Cooperative Estimation

A sensor network can be represented by an undirected graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is a set of nodes, i.e. agents, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, i.e. communication links between agents. The neighbours of node $v_i \in \mathcal{V}$ are given by the set $\mathcal{N}_i = \{v_j \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}\}$. The degree of node v_i is denoted by d_i. In Fig. 2.3, a communication graph with 5 nodes and 5 edges is shown where $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_3 = \{1, 2, 4\}$, $d_4 = 2$ and $d_5 = 1$.

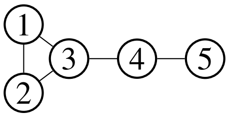


Figure 2.3: A communication graph example.

Consensus algorithms for sensor fusion require of each agent only a fixed small memory and simple isotropic exchange information with its neighbours. It diffuses information across the network, by updating the data from each node with a weighted average of its neighbours. At each step, every node can compute a local weighted average estimate, which eventually converges to the global weighted average solution.

In the special case that each sensor makes one noisy measurement of:

$$\theta_i = \theta^* + v_i, \tag{2.21}$$

where θ^* is the unknown constant variable to be estimated cooperatively, the distributed linear iterative average consensus method at sampling time $l = 0$, initializes each node with its measurement:

$$\theta_{\text{cons}_i}[0] = \theta_i, \tag{2.22}$$

where $\theta_{\text{cons}_i}[l]$ is the consensus estimate of the agent i at step l. At each following step, each node updates its consensus estimate with a linear combination of its own estimate and its neighbours' [111]:

$$\theta_{\text{cons}_i}[l+1] = w_{ii}[l]\theta_{\text{cons}_i}[l] + \sum_{j \in \mathcal{N}_i} w_{ij}[l]\theta_{\text{cons}_j}[l], \tag{2.23}$$

where w_{ij} is the linear weighting coefficient on θ_{const_j} at node i. Consensus weights can be incapsulated in the weighting matrix of W_{cons}. If all sensors' noises are independent and identically distributed (i.i.d) Gaussian, then the average consensus estimate of θ^* is equal to the maximum-likelihood estimates of the measurements:

$$\hat{\theta}_{ML} = \hat{\theta}_{avg} = \frac{1}{n}\sum_{i=1}^{n}\theta_i. \tag{2.24}$$

To guarantee convergence of the all nodes consensus estimates to the network's average estimate of θ^*, consensus algorithm weighings should be selected appropriately. For networks with static topology, the necessary and sufficient conditions for convergence of the distributed estimates from all the nodes to the networks' average estimate of θ^* for any initial set of θ_i are [72, 111]:

$$\mathbf{1}^T W_{\text{cons}} = \mathbf{1}^T, \tag{2.25}$$

$$W_{\text{cons}}\mathbf{1} = \mathbf{1}, \tag{2.26}$$

$$\rho(W_{\text{cons}} - \mathbf{1}\mathbf{1}^T/n) < 1, \tag{2.27}$$

where **1** is a vector that all elements are equal to 1, and ρ denoted the spectral radius of the W_{cons} matrix.

There are a wide range of consensus weighting schemes with different network topology information requirements and convergence rates. For example *constant edge weight* [78] is a weighting scheme with good performance. In this scheme all the edge weights are set to a constant α, and the self-weight is chosen to satisfy convergence conditions:

$$w_{ij} = \begin{cases} \alpha & i, j \in \mathcal{E}, \\ 1 - d_i\alpha & i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.28}$$

As an example, constant edge weight scheme for network of Fig. 2.3 is as follows:

$$W_c = \begin{bmatrix} 1-2\alpha & \alpha & \alpha & 0 & 0 \\ \alpha & 1-2\alpha & \alpha & 0 & 0 \\ \alpha & \alpha & 1-3\alpha & \alpha & 0 \\ 0 & 0 & \alpha & 1-2\alpha & 0 \\ 0 & 0 & 0 & \alpha & 1-\alpha \end{bmatrix}. \tag{2.29}$$

Due to convergence criteria (2.25)-(2.27), α should be bounded:

$$\alpha < \max_i d_i. \tag{2.30}$$

Therefore, each node needs to know $\max_i d_i$, which is a global topology property, or α should be chosen very conservatively. Another weighting scheme is *Metropolis weight*, which is defined as:

$$w_{ij} = \begin{cases} \frac{1}{1+\max d_i + d_j} & i,j \in \mathcal{E}, \\ 1 - \Sigma_{i,k\in\mathcal{E}} w_{i,k} & i = j, \\ 0 & \text{otherwise}. \end{cases} \tag{2.31}$$

One advantage of the Metropolis approach is that the nodes do not need any global knowledge of the communication graph, or even the number of nodes n [111].

Chapter 3

The Proposed Model Predictive Planning Design

In this chapter, a model predictive planning strategy is developed for highway emergency maneuvering in the presence of obstacles. Due to the time-varying nature of the driving environment, the planning scheme is formulated in an adaptive form and on-line estimation of the road condition is incorporated as a part of the planning strategy.

The closed-loop block diagram of the proposed planning and estimation scheme is demonstrated in Fig. 3.1. As depicted in this figure, longitudinal and lateral accelerations $\{a_{cx}, a_{cy}\}$ are applied by the planning and control unit, to avoid or mitigate collisions. The road-tire friction coefficient is considered as the unknown system parameter $\mu_{\max}$. The cooperative estimate of this parameter for the i-th agent denoted by $\hat{\mu}^{\text{cons}}_{\max i}$, comes from a consensus algorithm, which fuses the individual parameter identification $\hat{\mu}^{PI}_{\max i}$ with the road condition estimates of the surrounding vehicles $\hat{\mu}^{\text{cons}}_{\max j}, j \in \mathcal{N}_i$. Accordingly, the collision avoidance system uses $\hat{\mu}^{\text{cons}}_{\max}$ besides other measurements and perception data, to compute the proper acceleration commands. Additionally, computed $\hat{\mu}^{\text{cons}}_{\max}$ is broadcasted within the vehicular network, to be used by other vehicles, in a similar manner. The MPC planning technique and its simulation studies are presented in this chapter and the cooperative estimation approach is explained in Chapter 5.

The objective of emergency maneuvering is to keep the vehicle at a safe and optimal distance from the surrounding objects and road boundaries while tracking the road's center-line and avoiding exceeding the speed limit as possible. Therefore, the relative distance and speed of the surrounding vehicles, and the absolute position and speed of the host vehicle are incorporated in the decision-making process by using various sensing technologies such as LiDAR, INS, and GPS.

Real-time implementation of the planning algorithm is challenging. It needs to be updated more frequently to perform timely reactions to sudden changes of the situation. Accordingly, simplified linear motion model and quadratic cost function are employed for the controller to increase the speed of the MPC algorithm. Collision avoidance objective is enforced by states and input constraints, which will be termed here as safety constraints.

The proposed planning approach is evaluated in MATLAB/Simulink environment for two critical scenarios. Simulation results confirm successful motion planning by using the proposed emergency maneuver planner.

3.1 Modelling and Problem Definition

3.1.1 System Dynamics

A kinematic point-mass vehicle model [18, 47, 48] is considered as a control-oriented model for the collision avoidance problem studied in this thesis. However, the nonholonomic characteristic of the vehicle is imposed by defining a constraint for lateral velocity, as explained in 3.1.2. The discrete-time version of the control-oriented equations of motion of the vehicle are described as

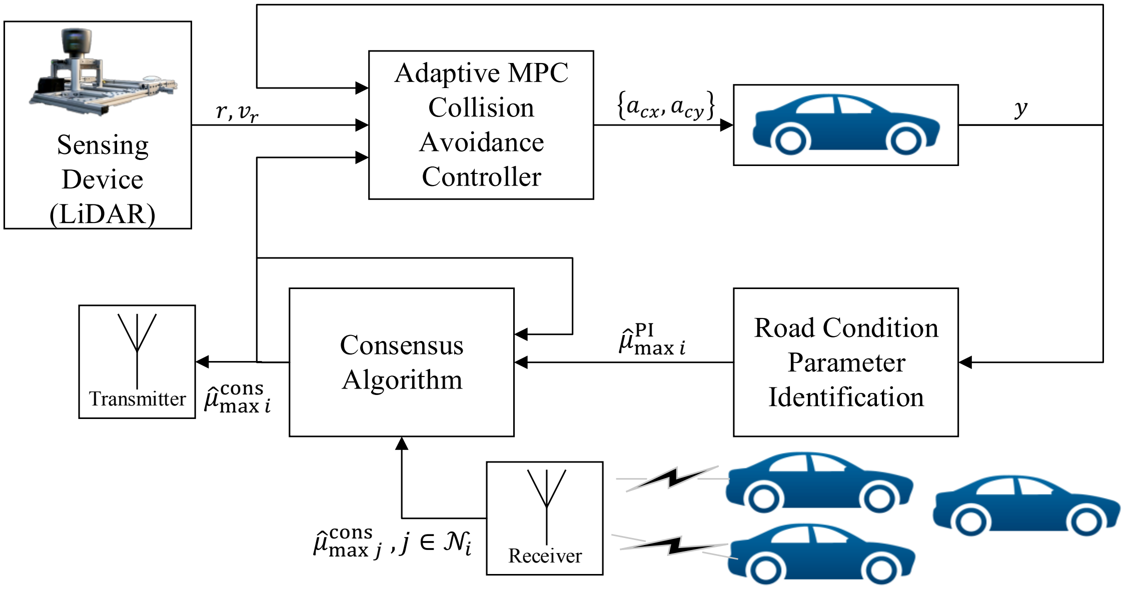

Sensing Device (LiDAR)
r, v_r
Adaptive MPC Collision Avoidance Controller
$\{a_{cx}, a_{cy}\}$
y
Road Condition Parameter Identification
$\hat{\mu}^{\mathrm{PI}}_{\max i}$
Consensus Algorithm
Transmitter
$\hat{\mu}^{\mathrm{cons}}_{\max i}$
Receiver
$\hat{\mu}^{\mathrm{cons}}_{\max j}, j \in \mathcal{N}_i$

Figure 3.1: Adaptive model predictive collision avoidance with cooperative estimation.

follows:

$$v_x[k+1] = v_x[k] + a_{cx}[k]T_{md}, \tag{3.1}$$

$$x[k+1] = x[k] + v_x[k]T_{md}, \tag{3.2}$$

$$v_y[k+1] = v_y[k] + a_{cy}[k]T_{md}, \tag{3.3}$$

$$y[k+1] = y[k] + v_y[k]T_{md}, \tag{3.4}$$

where x and y denote the longitudinal and lateral coordinates, respectively. Here, T_{md} is the model discretization time step, a_{cx} and a_{cy} are the commanded acceleration inputs and v_x and v_y are the speed of the vehicle, in longitudinal and lateral direction, respectively. Since the safety constraints are defined over the relative distances, equations for relative motion with respect to surrounding vehicles are also included in the vehicle model:

$$v_{rx_m}[k+1] = v_{rx_m}[k] + (a_{cx}[k] - a_{x_m}[k])T_{md}, \tag{3.5}$$

$$x_{r_m}[k+1] = x_{r_m}[k] + v_{rx_m}[k]T_{md}, \tag{3.6}$$

$$v_{ry_m}[k+1] = v_{ry_m}[k] + (a_{cy}[k] - a_{y_m}[k])T_{md}, \tag{3.7}$$

$$y_{r_m}[k+1] = y_{r_m}[k] + v_{ry_m}[k]T_{md}. \tag{3.8}$$

The subscript m denotes the m-th surrounding vehicle and the subscript r denotes the relative coordinate system. Accordingly, v_{rx_m}, and x_{r_m} represent the relative velocity and the position of the m-th surrounding vehicle in x direction:

$$v_{rx_m}[k] = v_x[k] - v_{x_m}[k], \tag{3.9}$$

$$x_{r_m}[k] = x[k] - x_m[k]. \tag{3.10}$$

Similarly, v_{ry_m}, and y_{r_m} are defined in y direction. Surrounding vehicle acceleration decisions, $\{a_{x_m}, a_{y_m}\}$, are received through V2V communication, and assumed to remain constant until new decisions are made. By defining the state vector as

$$X = [v_x, x, v_y, y, v_{rx_1}, x_{r_1}, v_{ry_1}, y_{r_1}, ..., v_{rx_M}, x_{r_M}, v_{ry_M}, y_{r_M}]^T_{1\times(4+4M)}, \tag{3.11}$$

the system dynamic (3.1)–(3.8) can be rewritten in a compact form as:

$$X[k+1] = AX[k] + BU[k] + W[k], \tag{3.12}$$

where M is the number of surrounding vehicles, $X \in R^{4+4M}$ is the state vector, $A \in R^{(4+4M)\times\ (4+4M)}$, $U \in R^2$ is the control inputs vector, $B \in R^{(4+4M)\ \times\ 2}$ and $W \in R^{4+4M}$ is the disturbance vector. The surrounding vehicles' acceleration commands are considered as disturbances in this model. As an example, if we have two surrounding vehicles, then $A \in R^{12\times12}$, $B \in R^{12\times2}$.

3.1.2 Constraints

Control Input Constraints

The vehicle's capability to produce acceleration and deceleration in lateral and longitudinal directions is limited. Thus, a_{cx} and a_{cy} are considered bounded. Friction circle concept describes the relation of the acceleration limits with maximum road friction coefficient [80]:

$$a_{cx}[k]^2 + a_{cy}[k]^2 \leq g^2 \mu_{\max}[k]^2, \tag{3.13}$$

where g is the acceleration due to gravity. This inequality is nonlinear and it would deteriorate the mathematical programming performance. Therefore, (3.13) can be replaced conservatively by two linear inequality constraints:

$$-\frac{1}{2} g \mu_{\max}[k] \leq a_{cx}[k] \leq \frac{1}{2} g \mu_{\max}[k], \tag{3.14}$$

$$-\frac{1}{2} g \mu_{\max}[k] \leq a_{cy}[k] \leq \frac{1}{2} g \mu_{\max}[k]. \tag{3.15}$$

State Constraints

The vehicle's longitudinal velocity v_x, should be positive and smaller than the vehicle's maximum feasible speed:

$$0 \leq v_x[k] \leq v_{x_{\max}}. \tag{3.16}$$

A set of constraints for vehicle's lateral velocity can be used to restrict equations of motion in order not to violate the nonholonomic constraints of the vehicle motion [76]:

$$- v_x[k] \tan(\beta_{\max}) \leq v_y[k] \leq v_x[k] \tan(\beta_{\max}). \tag{3.17}$$

where $\beta_{\max}$ is the vehicle's maximum admissible slip angle. Vehicle lateral position, y, is limited by the road limits:

$$y_{\mathrm{rd}_{\min}}[k] + \frac{L_y}{2} \leq y[k] \leq y_{\mathrm{rd}_{\max}}[k] - \frac{L_y}{2}, \tag{3.18}$$

where L_y is the width of the vehicle, $y_{\mathrm{rd}_{\max}}$ and $y_{\mathrm{rd}_{\min}}$ are the upper and lower road boundaries, respectively.

Safety Constraints

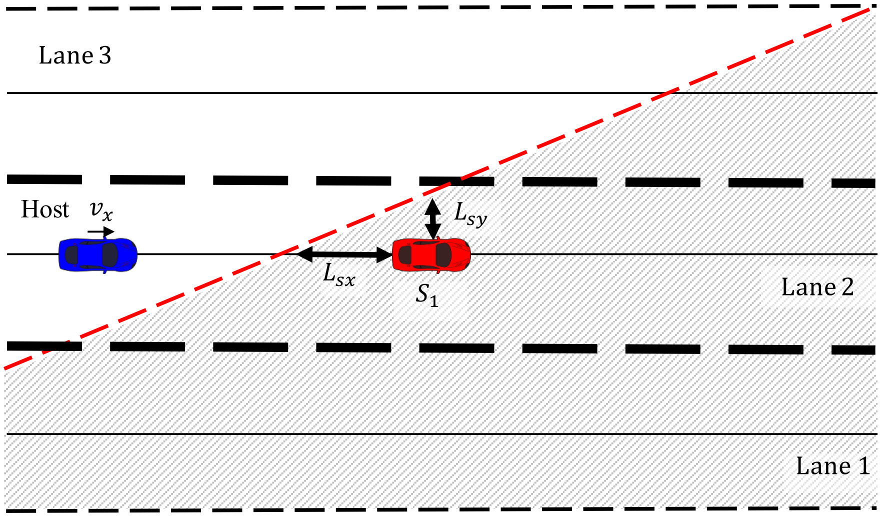
Safety constraints are used to avoid collision with surrounding obstacles. These constraints are defined differently for the obstacle in the same lane and obstacles in the adjacent lanes. For the obstacles in the same lane, e.g. S_1 in Fig. 3.2, the safety constraints are defined in convexified form to deviate the vehicle to adjacent lanes:

$$\begin{cases} \frac{1}{L_{sx}} x_{r_m}[k] - \frac{1}{L_{sy}} y_{r_m}[k] + \epsilon_{sc}[k] \leq -1 & \text{deviation to left} \\ \frac{1}{L_{sx}} x_{r_m}[k] + \frac{1}{L_{sy}} y_{r_m}[k] + \epsilon_{sc}[k] \leq -1 & \text{deviation to right} \end{cases} \tag{3.19}$$

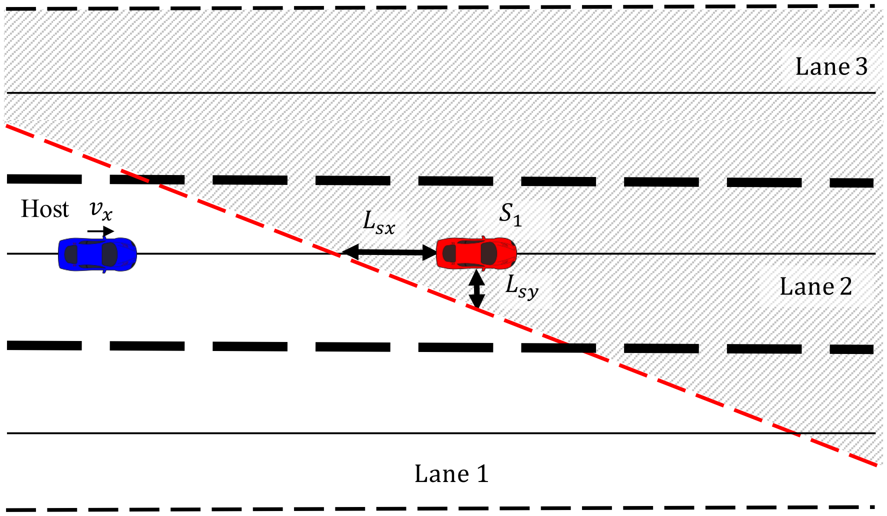
where ϵ_{sc} is a slack variable to soften the safety constraints. L_{sx} and L_{sy} are the safe distances i.e. minimum allowable distances between two vehicles in longitudinal and lateral direction, respectively. In Fig. 3.2, the red dashed line specifies the boundary of allowable area (white-coloured) for the planning.

The model predictive planner provides a trajectory for a certain prediction horizon; therefore the computed trajectory may violate the safety constraint at later instances of the prediction horizon. Accordingly, constraints are soften by a slack variable ϵ_{sc} to avoid infeasibility of the planning problem and to allow the planner to displace the safety constraints, by increasing the cost of the solution.

For obstacles in the adjacent lanes, defining safety constraints in the form of (3.19) makes the planning problem unnecessarily sophisticated. The inadmissible area for an adjacent-lane



(a) To left lane deviation constraint.



(b) To right lane deviation constraint.

Figure 3.2: Safety constraint for obstacles in the vehicle's lane.

obstacle is expanded by (3.19) to areas in the host lane. Therefore, it may cause unnecessary speed reduction to avoid constraint violation in the posterior instances of the prediction horizon. This issue is depicted in Fig. 3.3(a).

In the proposed planning strategy, the vehicles in adjacent lanes, e.g. S_2 in Fig. 3.3, are ignored until their longitudinal distance become less than or equal to L_{sx}. At that time, y bounds are changed to exclude unsafe areas around the adjacent vehicle. This convex constraint can be easily handled by the MPC solvers. For example, if the surrounding vehicle is placed in the left adjacent lane, y_{max} is changed when the longitudinal distance to the surrounding vehicle is less than or equal to L_{sx}, as depicted in Fig. 3.3(b). Thus the time-varying bounds of y constraint is:

$$y_{\mathrm{max}}[k] = \begin{cases} y_m[k] - L_{sy} & \text{if } |x_{r_m}[k]| \leq L_{sx} \ \& \ y_{r_m}[k] < 0 \\ y_{\mathrm{rd}_{max}} & \text{otherwise} \end{cases} \tag{3.20}$$

$$y_{\mathrm{min}}[k] = \begin{cases} y_m[k] + L_{sy} & \text{if } |x_{r_m}[k]| \leq L_{sx} \ \& \ y_{r_m}[k] > 0 \\ y_{\mathrm{rd}_{min}} & \text{otherwise} \end{cases} \tag{3.21}$$

For L_{sx}, it is common to define it as a function of v_x [47]:

$$L_{sx}[k] = d_0 + v_x[k]h_0, \tag{3.22}$$

where d_0 and h_0 are constant safety distance and headway time. In (3.22), time variation of road condition is not considered, thus the safety distance between two consequent vehicles on icy road would be the same as the safety distance on dry road, which is not reliable.

It is clear that if L_{sx} is greater than the minimum travel distance required to reach an obstacle's speed, $d_{x_{\mathrm{min}}}$, then the collision avoidance to that obstacle is guaranteed. $d_{x_{\mathrm{min}}}$ can be formulated in terms of the maximum deceleration magnitude $g\mu_{\mathrm{max}}$ by [67]:

$$d_{x_{\mathrm{min}}}[k] = (v_x[k]^2 - v_{x_{Si}}[k]^2)/(2g\mu_{\mathrm{max}}[k]), \tag{3.23}$$

This term can be added to (3.22) to make an extra gap between the vehicle and front/rear obstacle:

$$L_{sx}[k] = d_{0_x} + v_x h_0[k] + \max\{d_{x_{\mathrm{min}}}[k], 0\}. \tag{3.24}$$

Note that when two objects are divergent, $d_{x_{\min}}$ is negative; to avoid the negative effect in this case, term $\max\{.,0\}$ is used in (3.24).

In real-world application, relative distances and speeds can be determined by radar and/or LiDAR system. Road boundaries and lateral position can be determined by GPS data, vision sensors, and stored maps.

3.1.3 Cost Function

A certain quadratic cost function is defined to minimize unnecessary driving accelerations and enforce the vehicle to track the road's center-lines:

$$J[k] = \sum_{j=0}^{N_p-1} \Big(q_y (y[k+j|k] - y_{\text{ref}}[k+j])^2 + U[k+j|k]^T Q_u U[k+j|k] \Big) + \rho_{sc}^2 \epsilon_{sc}[k]^2, \quad (3.25)$$

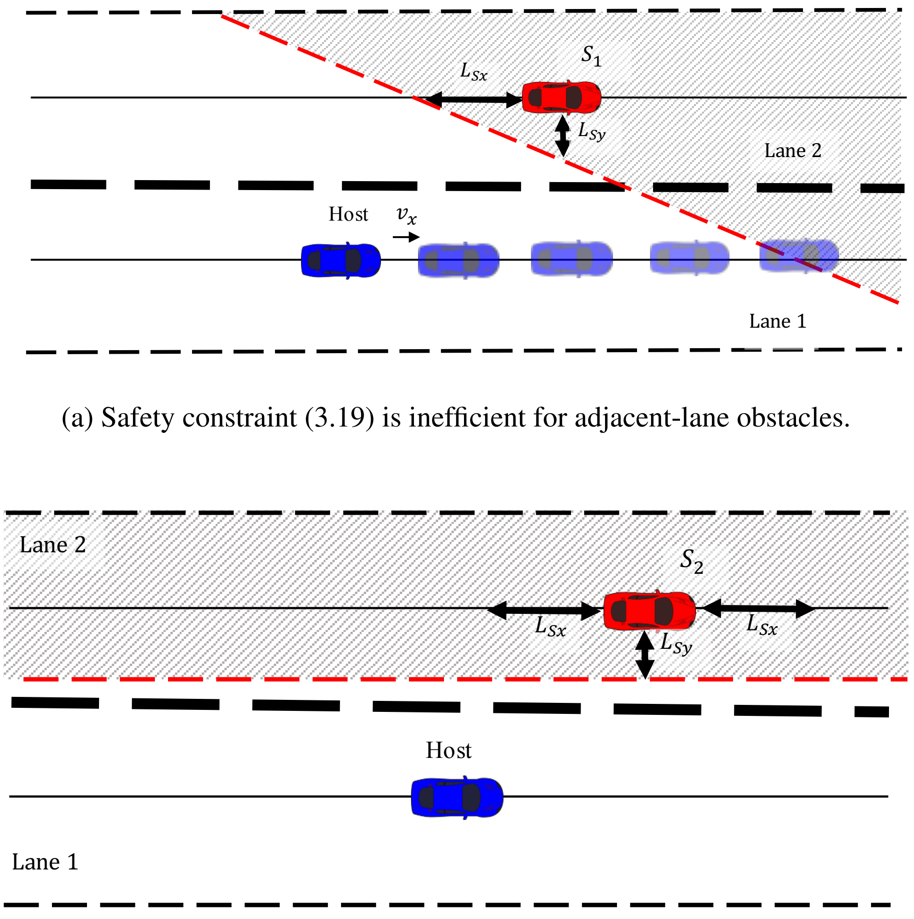
where q_y and ρ_{sc} are the weighting factors for tracking error y and slack variable ϵ_{sc}, respectively. Q_u is the weighting matrix for control inputs and $y_{\text{ref}}[k+j]$ is the lateral position of the closest center-line to the vehicle:

$$y_{\text{ref}}[k] = \arg \min_{y_{CL_i} \in \{y_{CL_1}, y_{CL_2}, \dots\}} (y[k] - y_{CL_i}), \quad (3.26)$$

where y_{CL_i} is the lateral coordination of each lane's center-line.

3.2 Simulation Results

Autonomous driving in a two- lane road is simulated in the Simulink/Matlab environment. Two surrounding vehicles are considered where S_1 is on the same lane and S_2 is on the adjacent lane. Two scenarios are considered for evaluation of the path planning strategy:



(a) Safety constraint (3.19) is inefficient for adjacent-lane obstacles.

(b) Changing $y_{\max}$ to avoid collision with adjacent-lane obstacles.

Figure 3.3: Safety constraint for obstacles in the vehicle's adjacent lane.

Table 3.1: Scenarios Description

Scenario	$x_{0_1}[m]$	$x_{0_2}[m]$	$v_0[m/s]$	$a_{x_1}[m/s^2]$	$a_{x_2}[m/s^2]$
1	100	30	20	-4	0
2	100	100	20	-4	-4

1. Lane shifting collision avoidance maneuvering, where S_1 suddenly stops while S_2 is driving with constant speed.

2. An emergency braking maneuver, in which both lanes are suddenly blocked.

Table 3.1 shows the obstacles' initial condition for each scenario. Although there exist many more scenarios that may be encountered as dangerous situations in highway driving, the above scenarios can be used as a baseline to demonstrate the performance of collision avoidance algorithms for major dangerous situations. Problem parameters are specified in Table 3.2.

The vehicle's trajectory for Scenario 1 is depicted in Fig. 3.4. According to this figure, the proposed planning scheme successfully computes a collision-free trajectory for the first scenario. The trajectory is inside the road boundaries, and the safety distance to moving obstacles are maintained during the driving. At the beginning of the scenario, none of the obstacles are close to the vehicle, thus safety constraints do not affect the MPC solution. When the vehicle reaches to the vicinity of surrounding vehicle S_2, the safety constraints for obstacles on adjacent lanes, i.e. (3.20) and (3.21) are triggered and restrict lateral movement. By overtaking S_2 and passing its safety distance, the safety constraint due to obstacles on the adjacent lane is excluded and the vehicle can freely move laterally and shift its lane. Accordingly, by becoming closer to the same-lane obstacle S_1, its corresponding safety constraint, i.e. (3.19), forces to shift its lane to the left. Acceleration inputs are depicted in Fig. 3.5. Based on these figures, the required longitudinal and lateral acceleration of the generated trajectory are inside the admissible range. Therefore, the planning outcome is physically feasible.

Table 3.2: Problem Parameters

$y_{\mathrm{rd}_{\min}} = 0$ [m]	$y_{\mathrm{rd}_{\max}} = 10$ [m]
$y_{\mathrm{CL}_1} = 2.5$ [m]	$y_{\mathrm{CL}_2} = 7.5$ [m]
$v_{x_{\max}} = 40$ [m/s]	$\beta_{\max} = 5$ [deg]
$a_{c_x \min} = -4$ [m/s^2]	$a_{c_x \max} = 1$ [m/s^2]
$a_{c_y \min} = -2$ [m/s^2]	$a_{c_y \max} = 2$ [m/s^2]
$L_{s_x} = 30$ [m]	$L_{s_y} = 1$ [m]
$L_x = 5$ [m]	$L_y = 2.5$ [m]

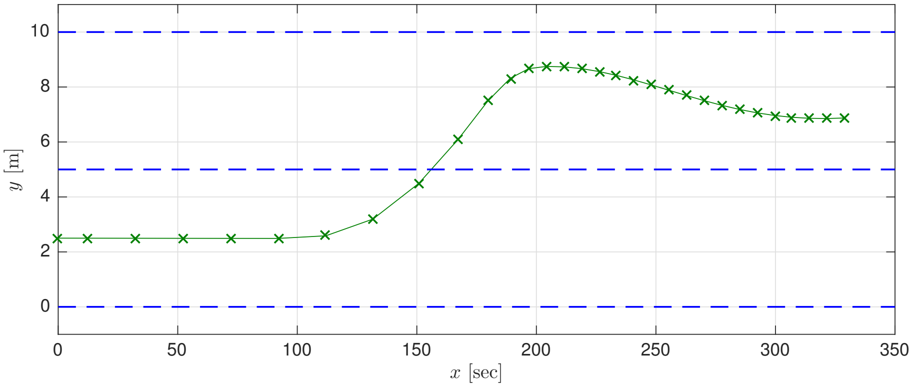


Figure 3.4: The vehicle's trajectory for Scenario 1.

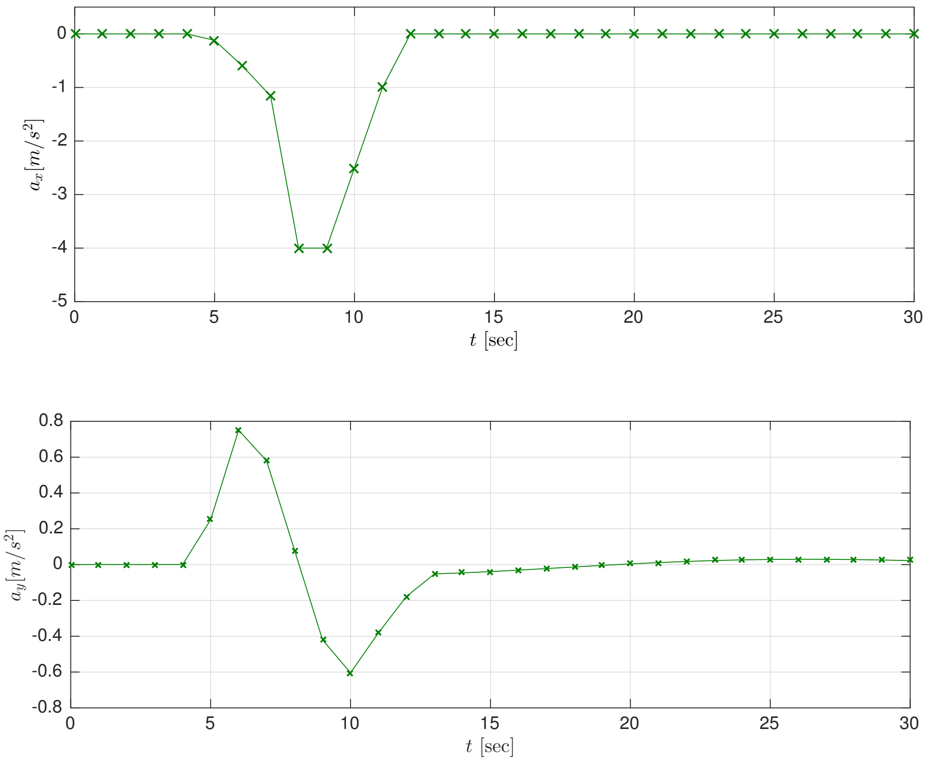


Figure 3.5: Longitudinal and lateral accelerations, a_{c_x} , a_{c_y} for Scenario 1.

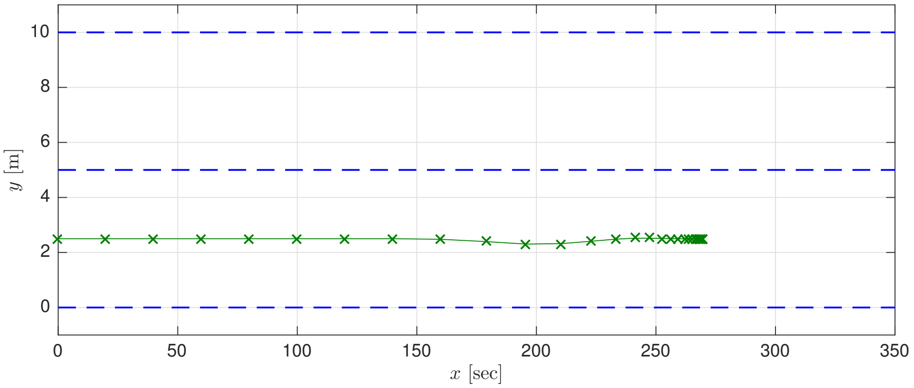


Figure 3.6: The vehicle's trajectory for Scenario 2.

The simulation results for Scenario 2 are plotted in Figs. 3.6 and 3.7. In this scenario, there is not enough distance between two surrounding vehicles. Therefore, the feasible decision is full braking to stop the vehicle completely before hitting the obstacles. The resulting trajectory in Fig. 3.6 shows that the vehicle is stopped around $x = 270[m]$. Accordingly, considerable longitudinal deceleration should be employed, as depicted in Fig. 3.7.

The linearity of the system equations and convexity of the constraints provide us a quadratic programming (QP) problem that can be solved in a fast and computationally efficient way. The solver presented in [92] is used in Matlab/Simulink environment. As a result, the computation time for path planning of a 30 sec scenario with a typical Intel Core i5 laptop was less than 4–5 seconds.

3.3 Summary

A linear MPC with time-varying constraints is used for emergency maneuver planning in a multi-lane road. Deviation from the lane center-line and control inputs is considered as quadratic cost terms, safety constraints are defined by time-varying convexified inequality constraints. Sur-

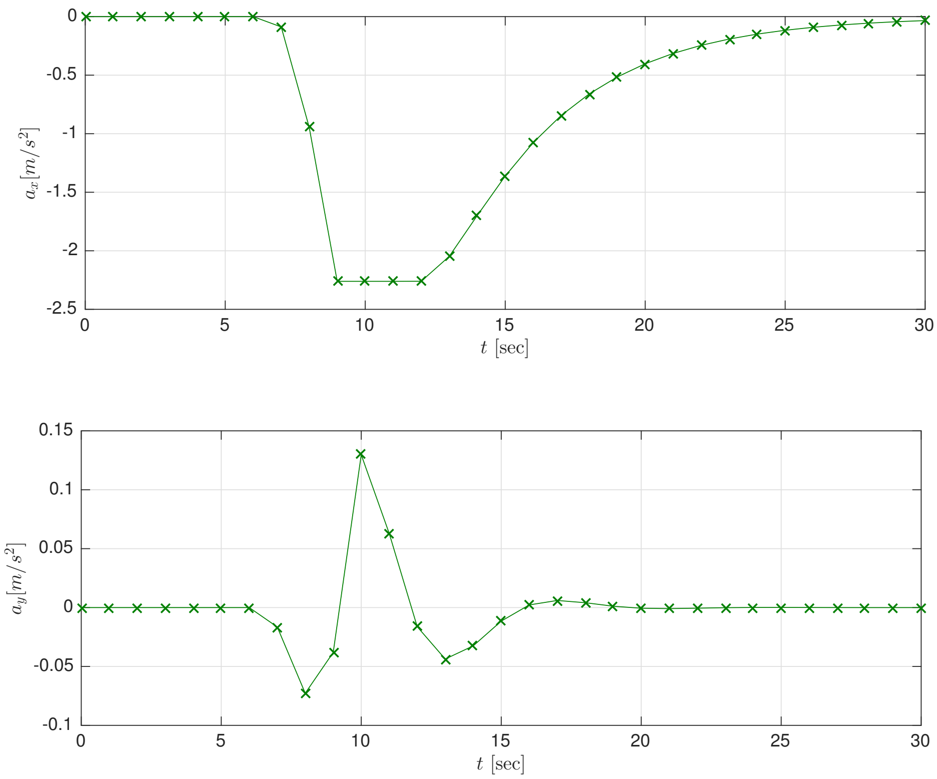


Figure 3.7: Longitudinal and Lateral accelerations, a_{c_x} , a_{c_y} for Scenario 2.

rounding vehicles' accelerations are incorporated in the model predictive decision making to have better trajectory prediction of these vehicles. The set of quadratic cost functions and convex constraints for states, control inputs, and safety concerns provide a real-time implementable collision avoidance strategy.

Two scenarios including lane shifting and complete stopping are evaluated for the proposed control scheme. Simulation results confirm the proper activation of collision avoidance commands. Convexified safety constraints for the same-lane obstacles successfully enforce the vehicle to shift the lane, if no obstacle is in the next lane. The simulation studies confirm that, changing $y_{\max}$ and $y_{\min}$ is a simple and efficient way to avoid adjacent-lane obstacles.

Chapter 4

Persistent Feasibility of Model Predictive Motion Planning

In this chapter, we investigate the real-time approaches to guarantee the persistent feasibility of a model predictive motion planning scheme. As stated in Chapter 3, model predictive control (MPC) can provide real-time and near-to-optimal decisions for autonomous vehicle motion planning, considering the vehicle's dynamics and constraints and predicting the state propagation for a specific time horizon.

The vehicle's onboard sensors provide the perceptional information required for the motion planning, including the obstacles' localization and road boundaries. In addition, vehicular communication can be used to extend the autonomous vehicle's perception range. Nevertheless, the prediction horizon of the motion planner is limited and cannot incorporate all the perception information. It is mainly because of limited on-board computation resources and the requirement on the computational speed of motion planning updates. Therefore, some of the detected obstacles would be ignored temporarily and it is always possible that the planning scheme encounters an obstacle that enters the prediction horizon belatedly and causes infeasibility. For instance, in a high-speed driving situation when an obstacle (e.g. another vehicle) is in the vehicle's lane (Fig. 4.1), if the motion planning prediction horizon is short, the motion planning scheme ig-

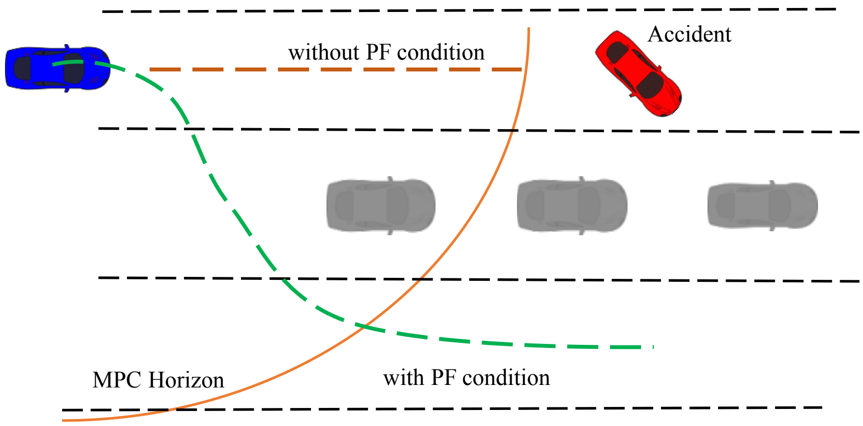


Figure 4.1: A driving scenario with unfeasibility issue in near future.

nores the stopped obstacle until the prediction horizon recedes and covers the stopped obstacle, which may cause collision. The persistent feasibility conditions enforce the planning scheme to be prepared for upcoming, out of horizon events.

In Section 2.2.3, we explained that the solution produced by MPC may lead to infeasibility at the upcoming time instances, i.e. non-existence of a solution that satisfies the system constraints. It is shown in [10] that if the MPC terminal states are kept inside a *control invariant set* of the system, then the controller is persistently feasible.

In this chapter, we investigate two real-time implementable approaches for computation of a near-to-maximal control invariant set of the constrained motion planning problem to maintain persistent feasibility of autonomous motion planning. The first approach to our computation of control invariant set is based on linearization and convexification of the motion planning problem, noting that the collision avoidance constraints, as depicted by red regions in Fig. 2.1, make the admissible domain of the optimization problem *non-convex*. Convexification helps the control invariant set computation to be fast and efficient. The downside of convexification is that it considerably reduces the admissible domain and thus may limit the range of feasible solutions in certain cases. The second approach utilizes a brute-force search algorithm, which is employed offline to extract look-up tables to determine the control invariant sets in real-time.

This chapter is organized as follows: Section 4.1 presents the dynamic model and constraints that we consider for the determination of persistently feasible conditions. Section 4.2 presents the convexification approach for control invariant set computation while Section 4.3 4.3 introduces the offline brute force search approach for real-time incorporation of the persistent feasibility condition. The simulation studies and conclusions are presented in Section 4.4 and Section 4.5.

4.1 Dynamic Model and Constraints

A linear vehicle model, same as (3.1)–(3.4), is used in this chapter for persistent feasibility analysis. Acceleration constraints and states constraints are also considered, in a similar way as they have been defined by (3.14)–(3.18). The set of linear equations of motion and corresponding constraints are listed as below:

$$
\begin{aligned}
v_x[k+1] &= v_x[k] + a_{cx}[k]T_{md}, && (4.1)\\
x[k+1] &= x[k] + v_x[k]T_{md}, && (4.2)\\
v_y[k+1] &= v_y[k] + a_{cy}[k]T_{md}, && (4.3)\\
y[k+1] &= y[k] + v_y[k]T_{md}, && (4.4)\\
a_{cx_{\min}}[k] \le& a_{cx}(k) \le a_{cx_{\max}}[k], && (4.5)\\
a_{cy_{\min}}[k] \le& a_{cy}(k) \le a_{cy_{\max}}[k], && (4.6)\\
v_{x_{\min}} \le& v_x[k] \le v_{x_{\max}}, && (4.7)\\
y_{\min}[k] \le& y[k] \le y_{\max}[k], && (4.8)\\
-v_x[k]\tan(\beta_{\max}) \le& v_y[k] \le v_x[k]\tan(\beta_{\max}), && (4.9)
\end{aligned}
$$

Safety constraints that guarantee collision avoidance with obstacles also affect the control invariant sets. As we discussed in Chapter 2, quadratic constraints (see Fig. 2.1) can be used for defining an obstacle's safety constraint:

$$\frac{1}{L_{Sx}^2}(x - x_{Si})^2 + \frac{1}{L_{Sy}^2}(y - y_{Si})^2 \geq 1, \tag{4.10}$$

where L_{Sx} and L_{Sy} are longitudinal and lateral safety distances, respectively, x_{Si} and y_{Si} are the coordinates of S_i. For obstacles in front of the vehicle, having a slower speed makes the obstacle more likely to collide. Therefore, a worst-case scenario is full-stopping of the front vehicle/obstacle in a very short time duration, which requires a very fast and severe response to avoid the collision. Accordingly, in this chapter, our approach to the computation of control invariant set is focused on the worst-case scenarios, the situations with a fully stopped vehicle/obstacle.

4.2 Convexification Approach

Control invariant computation for linear and convex systems is simple and efficient [10], but safety constraints make the motion planning optimization problem non-convex. Therefore, convexification of the safety constraint as introduced in [40,76,83] would facilitate the computation. In this technique, the non-convex state admissible set $\mathcal{X}$ is divided into multiple convex subsets of $\mathcal{X}$. For the motion planning problem, $\mathcal{X}$ can be divided into 4 convex subsets $\{\mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{rr}},\ \mathcal{X}_{\text{fl}},\ \mathcal{X}_{\text{fr}}\}$ as depicted in Fig. 4.2, which are defined as:

$$(x - x_{Si}) \pm \frac{L_{Sx}}{L_{Sy}}(y - y_{Si}) \leq -L_{Sx}, \tag{4.11}$$

$$(x - x_{Si}) \pm \frac{L_{Sx}}{L_{Sy}}(y - y_{Si}) \geq L_{Sx}. \tag{4.12}$$

Then, the control invariant set of each convex set can be computed separately, by using Algorithm 2.1. Afterwards, the union of the resulting control invariant sets can be calculated to determine an estimate of the maximal control invariant set of the system's admissible set, as depicted in Algorithm 4.1.

The resulting control invariant set of Algorithm 4.1 is always a subset of the $C_\infty(\mathcal{X})$ by the following theorem:

Algorithm 4.1 Convexification Approach

```
**input:** $g$, $\mathcal{X}$ and $\mathcal{U}$
**output:** $C(\mathcal{X})$

**Convexify** $\mathcal{X}$ into $m$ convex set $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_m | \cup_{i=1}^{m} \mathcal{X}_i \in \mathcal{X}\}$
**for** $i = 1 : m$
  **Compute** $C_\infty(\mathcal{X}_i)$ Using Algorithm 2.1
$C_\infty(\mathcal{X}) \leftarrow \cup_{i=1}^{m} \{C(\mathcal{X}_i\}$
```

Theorem 4.1. *Given a group of sets $S_1, ..., S_N$, and a control invariant set for each of them, $C(S_1), ..., C(S_N)$, then the union of the control invariant sets $\bigcup_{i=1}^{N} C(S_i)$ is a subset of the maximal control invariant set of $\bigcup_{i=1}^{N} S_i$:*

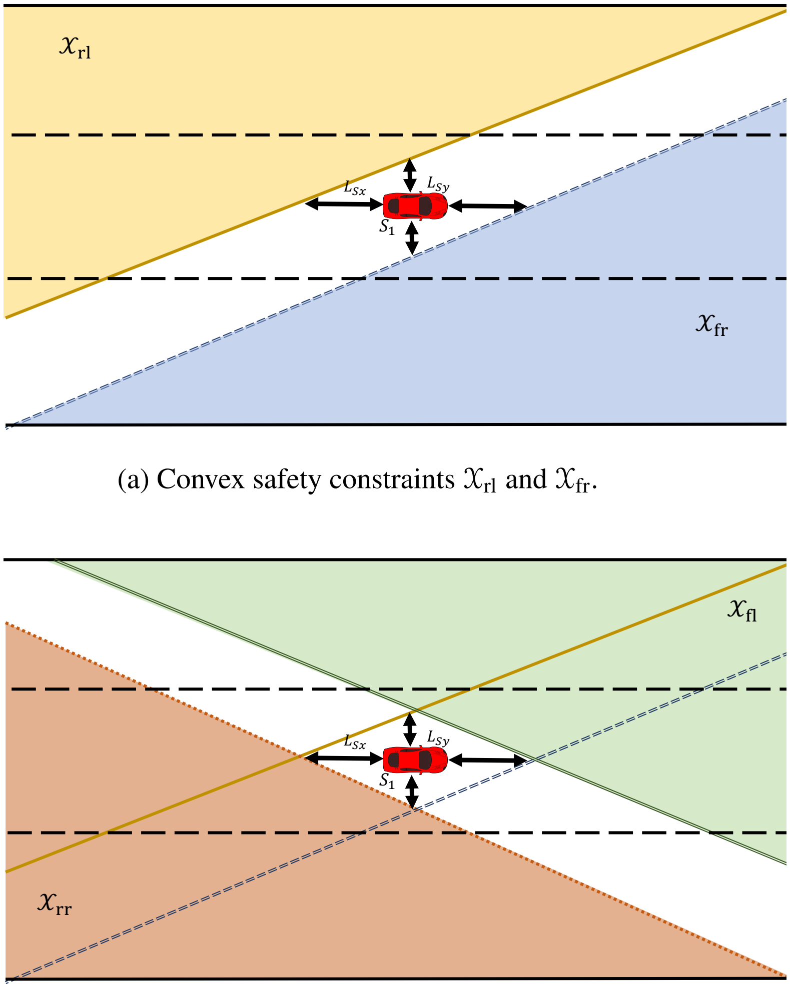
$$C(S_1) \cup ... \cup C(S_N) \subset C_\infty(S_1 \cup ... \cup S_N) \tag{4.13}$$

Proof. *Based on the definition of control invariant set, if $x \in C(S_i)$, then, there are admissible control inputs to keep x inside S_i forever. Since $S_i \subset \bigcup_{i=1}^{N} S_i$, therefore, that admissible control input will maintain x inside $\bigcup_{i=1}^{N} S_i$. Thus, $x \in C_\infty(\bigcup_{i=1}^{N} S_i)$.*

By Theorem 4.1, control invariant sets $C(\mathcal{X}_{\text{rl}}),\ C(\mathcal{X}_{\text{rr}}),\ C(\mathcal{X}_{\text{fl}}),\ C(\mathcal{X}_{\text{fr}})$ are subsets of $C_\infty(\mathcal{X})$ and $C_\infty(\mathcal{X})$ can be estimated by the union of them.

4.3 Offline Brute-Force Search Approach

A brute-force search algorithm can be used to compute the controllable sets of a constrained problem without convexification or linearization of the problem. By this approach, $\mathcal{X}$ is discretized and converted to a finite set of points P_0 in the state space $\mathbb{R}^n$. Then, the intersection of controllable set $\mathcal{K}_1(P_0)$ with $\mathcal{X}$ is considered as Ω_1 of Algorithm 2.1. For Ω_2, the same process is repeated for points remained in Ω_1 and checked if there is an admissible control input to maintain the one step propagation of these states within the Ω_1 boundaries. This recursive algorithm is


$\mathcal{X}_{\mathrm{rl}}$
L_{Sx}
L_{Sy}
S_1
$\mathcal{X}_{\mathrm{fr}}$
$\mathcal{X}_{\mathrm{fl}}$
L_{Sx}
L_{Sy}
S_1
$\mathcal{X}_{\mathrm{rr}}$

(a) Convex safety constraints $\mathcal{X}_{\mathrm{rl}}$ and $\mathcal{X}_{\mathrm{fr}}$.

(b) Convex safety constraints $\mathcal{X}_{\mathrm{fl}}$ and $\mathcal{X}_{\mathrm{rr}}$.

Figure 4.2: Four convexified safety constraints over an obstacle.

repeated until it converges to maximal control invariant set $\mathcal{C}_\infty(\mathcal{X})$. The proposed algorithm for computation of C_∞ by brute-force technique is demonstrated in Algorithm 4.2.

Algorithm 4.2 Brute-Force Search Approach

```
input: $g$, $\mathcal{X}$ and $\mathcal{U}$
output: $C(\mathcal{X})$

Discretize $\mathcal{X}$ into finite set $P_0$ of points
Set $\Omega_0 \leftarrow \alpha(P_0)$, $\alpha$-hull of $P_0$
Repeat
  $P_k \leftarrow \mathcal{K}_1(P_{k-1}) \cap \Omega_{k-1}$
  $\Omega_k \leftarrow \alpha(P_k)$
  $k \leftarrow k + 1$
Until $P_k = P_{k-1}$
$C_\infty \leftarrow \Omega_k$
```

In using the brute-force search approach for computation of controllable sets of a constrained dynamic system, it is necessary to numerically verify if the given set of points is inside or outside of Ω_k since Ω_k is defined only by a set of n-D points. This operation can be done using Delaunay triangulation and alpha-shapes computation techniques. The alpha shape of a set of points is a generalization of the convex hull. Alpha shapes have a parameter that controls the level of detail, or how tightly the boundary fits around the point set, therefore it can be used to generate non-convex volumes. *InShape* function of MATLAB's computational geometry toolbox checks if a 2-D or 3-D point is inside an *alpha shape* or not. Since the employed dynamic system's state space is 4-D, some modifications are applied to the MATLAB's InShape function to be able to process 4-D state vectors.

The brute-force search approach is computationally intensive and cannot be used in real-time. Therefore, it is suggested to extract a table of information from the computed $\mathcal{C}_\infty(\mathcal{X})$ and use it as a look-up table in real-time. For example, it is possible to determine non-invariant area around the obstacle for a wide range of longitudinal and lateral velocities. Therefore, in real-time application, the model predictive motion planning scheme can use this information to adapt the safety distances L_{s_x} and L_{s_y} based on the current speed of the vehicle:

$$L_{Sx_{PF}} = L_{Sx_{PF}}(v_x, v_y), \tag{4.14}$$

$$L_{Sy_{PF}} = L_{Sy_{PF}}(v_x, v_y), \tag{4.15}$$

By this way, the MPC scheme does not use a constant longitudinal and lateral safety distance, and update them based on the instantaneous speed of the vehicle.

4.4 Simulation Results

In this section, control invariant set computation of the constrained motion planning problem in presence of an obstacle, which is assumed to perform a stop, is computed using two approaches explained in this chapter. The vehicle parameters and characteristic parameters for the scenarios are presented in Table 4.1.

4.4.1 Convexification Approach

Based on the approach explained in Section 4.2, the admissible set is divided into four convex subsets, $\mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{rr}},\ \mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{fr}}$, and accordingly, C_∞ is computed for each subset. The union of these control invariant sets provides an estimate of $C_\infty(\mathcal{X})$. Multi Parametric Toolbox (MPT) [35] is used to determine C_∞ for each subset and also to compute the union of the control invariant sets. MPT is an open-source, Matlab-based toolbox for parametric optimization, computational geometry, and MPC. The command *invariantSet* of this toolbox computes the maximal control invariant set for a linear convex system.

Fig. 4.3 shows the computed C_∞ for $\mathcal{X}_{\text{rl}}$ set for $v_y = 0$. MPT's *slice* command is used to slice $\mathcal{X}_{\text{rl}}$ for $v_y = 0$. From this figure, it is clear the computed control invariant area is shrinking as v_x increases. The estimate of $C_\infty(\mathcal{X})$, i.e. union of $\mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{rr}},\ \mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{fr}}$, for $v_y = 0$ is illustrated

Table 4.1: Simulation parameters.

$a_{cx_{\min}} = -6$ [m/s^2]	$a_{cx_{\max}} = 6$ [m/s^2]
$a_{cy_{\min}} = -3$ [m/s^2]	$a_{cy_{\max}} = 3$ [m/s^2]
$v_{x_{\min}} = 0$ [m/s]	$v_{x_{\max}} = 30$ [m/s]
$v_{y_{\min}} = -3$ [m/s]	$v_{y_{\max}} = 3$ [m/s]
$\beta_{\max} = 0.2$	$T_{md} = 0.5$ [sec]
$y_{\min} = 0$ [m]	$y_{\max} = 12$ [m]
$L_{Sx} = 3$ [m]	$L_{Sy} = 2$ [m]
$x_{obs} = 40$ [m]	$y_{obs} = 6$ [m]

in Fig. 4.4. The resulting union of these subsets is depicted in Fig. 4.5 for different longitudinal speeds while $v_y = 0[m/s]$. Fig. 4.5 clearly verifies the earlier observation that the computed control invariant set is smaller for higher v_x values. In other word, the vehicle should maintain larger distance from the obstacle at larger v_x values to have persistently feasible motion planning solution.

4.4.2 Brute-Force Search Results

Considering motion planning constraints (4.5), (4.6), (4.9), (4.10), and their corresponding parameters in Table 4.1, the maximal control invariant set of this problem is computed by brute-force technique. Fig. 4.6 demonstrates the slices of $C_\infty(\mathcal{X})$ for $v_x = 10[m/s]$ and $v_x = 20[m/s]$ respectively while $v_y = 0[m/s]$ for both figures. As it can be seen from this figure, the non-invariant set behind the obstacle is expanded longitudinally for larger v_x, which is in accordance

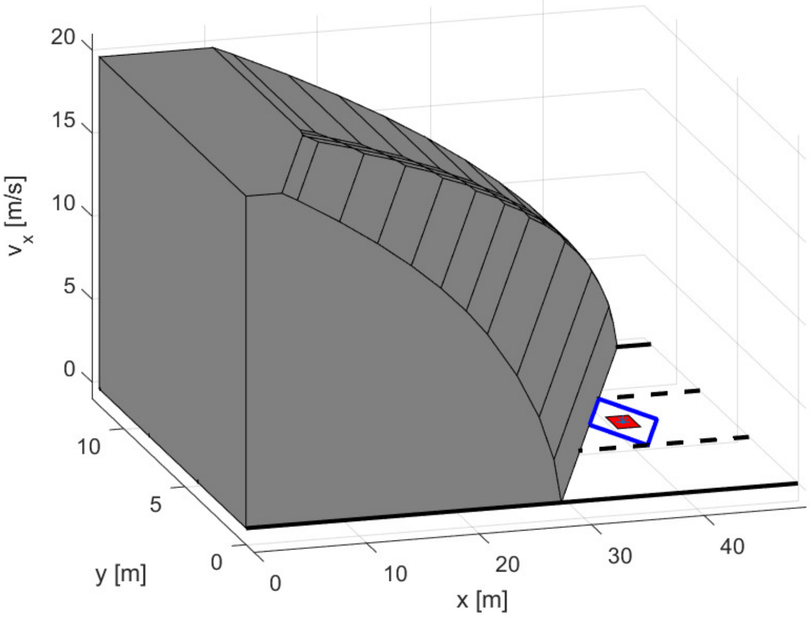


Figure 4.3: $\mathcal{X}_{\text{rl}}$ Control invariant set slice for $v_y = 0$.

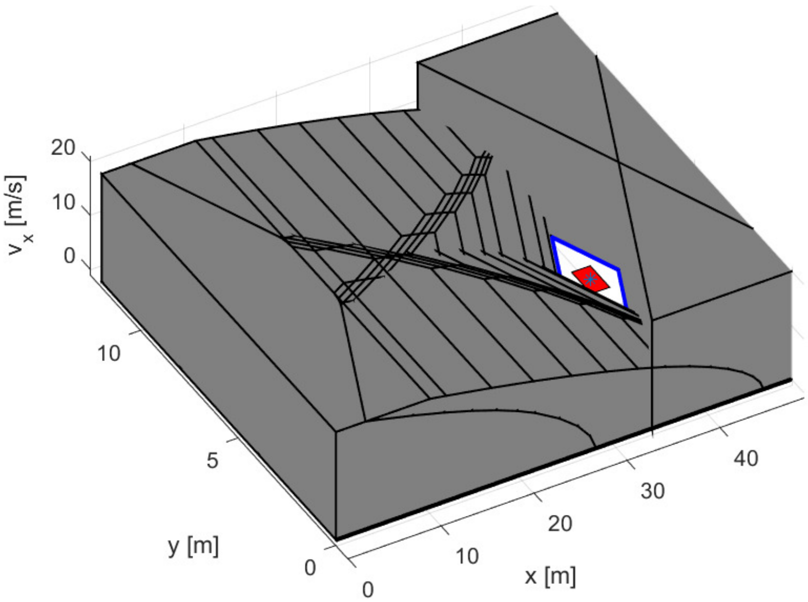
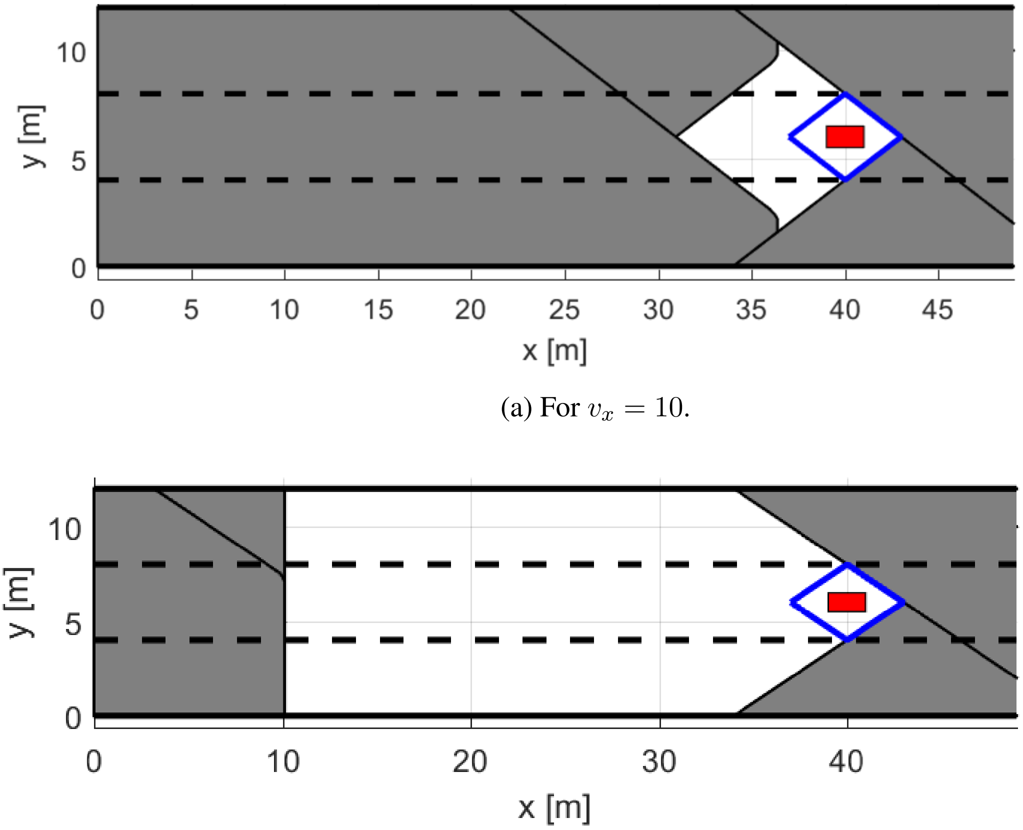


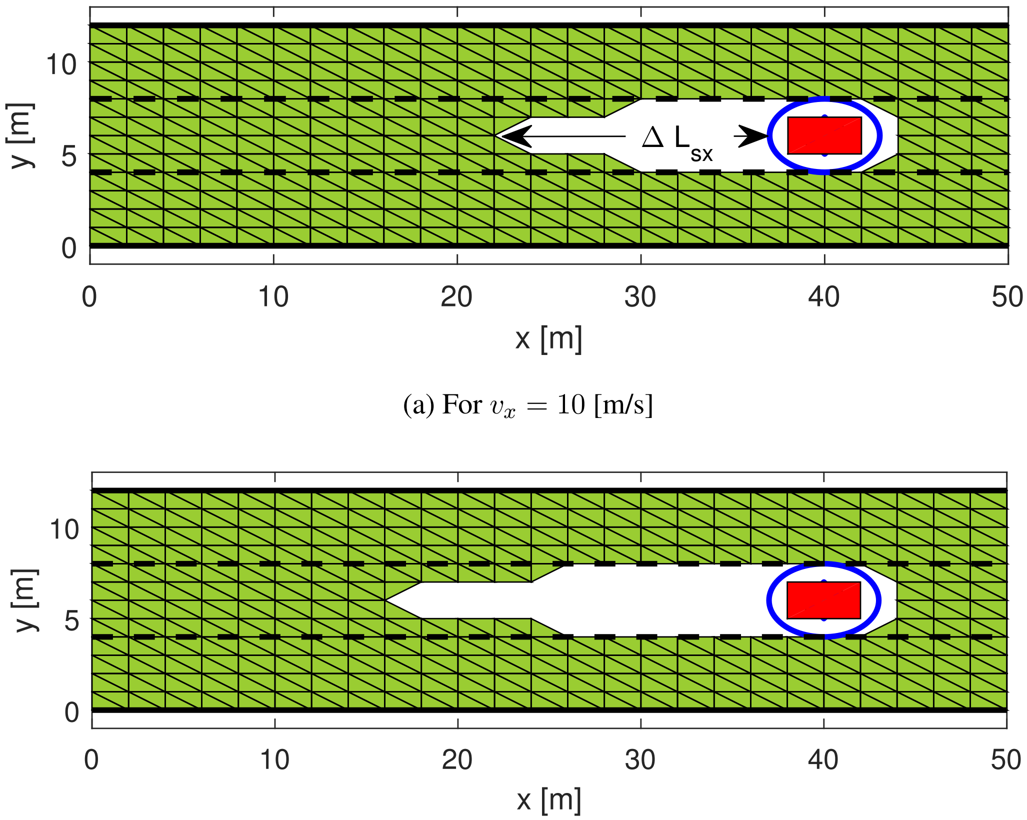
Figure 4.4: Union of $\mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{rr}},\ \mathcal{X}_{\text{rl}},\ \mathcal{X}_{\text{fr}}$, sliced for $v_y = 0$.



(a) For $v_x = 10$.

(b) For $v_x = 20$

Figure 4.5: Slice of C_∞ on $x - y$ plane for $v_y = 0$ and different longitudinal speeds.



(a) For $v_x = 10$ [m/s]

(b) For $v_x = 20$ [m/s]

Figure 4.6: $C_\infty(\mathcal{X})$ by brute-force algorithm for $v_y = 0$ and different longitudinal speeds.

with the results from convexification approach.

Fig. 4.7 illustrates $C_\infty(\mathcal{X})$ for a situation that vehicles' lateral velocity is $v_y = 2[m/s]$ and compares it with a scenario with zero lateral velocity $v_y = 0$. It shows a little expansion of non-invariant set to the right side of the vehicle.

To provide better comparison between the two studied approaches, results from convexification and brute-force approaches are combined in Fig. 4.8. It is clear that the convexification approach is more conservative and enforces more restriction on the admissible states.

ΔL_{Sx} and ΔL_{Sy} can be defined as the maximum distance of the non-invariant area from the

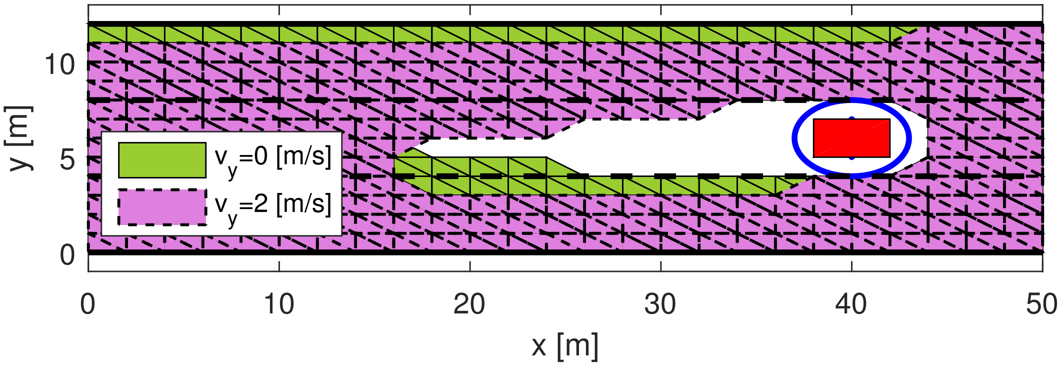


Figure 4.7: Lateral speed effect on $C_\infty(\mathcal{X})$ for $v_x = 20$ [m/s].

obstacle's safety boundary. For example, in Fig. 4.6(a), ΔL_{Sx} is depicted by an arrowed line and ΔL_{Sy} is zero. Therefore, $L_{Sx_{PF}} = L_{Sx} + \Delta L_{Sx}$ and $L_{Sy_{PF}} = L_{Sy} + \Delta L_{Sy}$. ΔL_{Sx} and ΔL_{Sy} can be extracted from the computed control invariant sets and stored in arrays like below:

$$\Delta L_{Sx}(v_x) = [(0,0),(4,5),(10,15),(14,17),(20,21)], \tag{4.16}$$

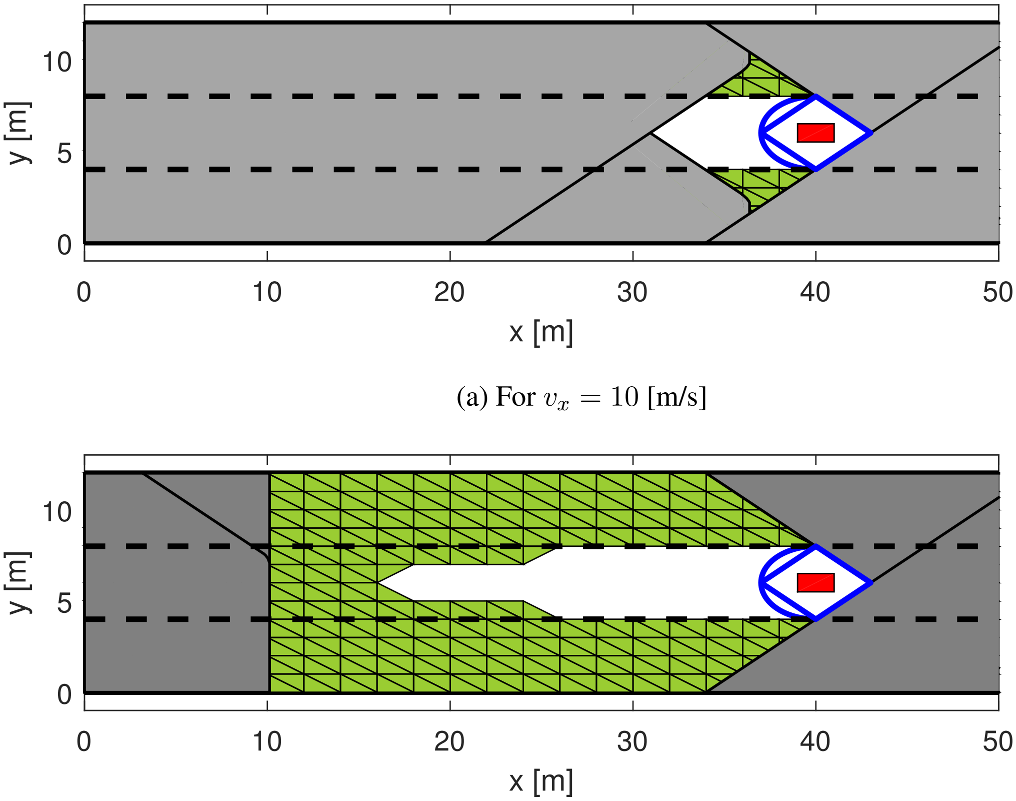
$$\Delta L_{Sy}(v_y) = [(0,0),(1,1),(2,1),(3,2)], \tag{4.17}$$

which means that when for example $v_x = 10[m/s]$, ΔL_{Sx} should be $15[m]$ or more and when $v_y = 1[m/s]$, ΔL_{Sy} should be more than $1[m]$ to maintain the vehicle in the system's control invariant set.

The extracted L_{Sx} and L_{Sy} can be used by the model predictive motion planning scheme. In this manner, in each time step $L_{Sx_{PF}}$ and $L_{Sy_{PF}}$ safety distances are calculated by the stored look-up arrays and used by the MPC scheme. Therefore, the MPC scheme does not use a constant safety distance, and updates the safety distances based on the instantaneous speed of the vehicle.

4.5 Summary

In this chapter, two efficient control invariant set computation approaches, convexification approach and brute-force search approach are investigated to guarantee persistent feasibility of the



(a) For $v_x = 10$ [m/s]

(b) For $v_x = 20$ [m/s]

Figure 4.8: Comparison of computed $C_\infty(\mathcal{X})$ by convexification and brute-force techniques for different longitudinal speeds.

model predictive motion planning problem for autonomous driving vehicles. It is shown that the convexification approach is more conservative and eliminates many feasible maneuvers for the motion planning. On the other hand, using the brute-force technique, an efficient technique is suggested to compute the control invariant sets offline, then the resulting non-invariant area around an obstacle is characterized by two parameters that can be stored in look-up tables for real-time application.

Chapter 5

Cooperative Road Condition Estimation for Emergency Maneuvering

Time-variation of the road condition affects the vehicle dynamics and constraints. Therefore, there is a substantial need for the estimation of the road friction coefficient [14, 34, 57], which can be coupled with the implementation of adaptive planning and control techniques [88]. In this chapter, we propose a new cooperative estimation technique based on a consensus algorithm for estimation of the road condition within the vehicular network. It is assumed that each vehicle estimates the road condition individually, and disseminates it through the network. Then, the employed consensus algorithm fuses the individual estimates to find the average of network's road conditions. To the best of the author's knowledge, there has been no investigation in cooperative road condition estimation that incorporates the online identification of the road friction coefficient.

The proposed cooperative road condition estimation algorithm, in the network of n connected vehicles, runs a dual-rate estimation scheme composed of (i) a high-rate individual recursive least-squares on-line parameter identification algorithm (explained in 5.1), and (ii) a low-rate consensus-based cooperative estimation scheme (explained in 5.2). The individual parameter identification generates the high-rate estimate using the measured signals of longitudinal force and velocity together with the wheel speed. The low-rate consensus-based cooperative estimation scheme is fed by individual estimates and is used to converge to a common estimate between the vehicles.

Simulation studies for a small network of communicating vehicles are performed in 5.3, to evaluate the performance of the proposed estimation scheme. Measurement noise is incorporated in the estimation loop to achieve more realistic simulation results. It is concluded that the cooperative consensus scheme improves the estimation performance significantly.

5.1 Single-agent parameter identification

On-line parameter identification of the road condition coefficient requires a parametric model of the tire dynamics. LuGre tire dynamics model is adopted for the study in this thesis. It requires relatively few parameters to characterize tire dynamics and also it can capture the transient dynamics of the tire. Since the implementation of the online parameter identification schemes are done using digital computers, it requires discrete representation of the system. The discrete representation of the LuGre tire model is first introduced in 5.1.1. In 5.1.2, an online parameter identification method is developed to the road condition parameter identification problem. An adaptive Luenberger observer is developed in 5.1.3 to determine the internal state of the discrete LuGre tire model since it is required by the online identification scheme of 5.1.2

5.1.1 Tire Model

For highway driving, it can be assumed that there are no significant lateral forces most of time. Therefore, longitudinal LuGre tire model is used for agents' individual road condition estimations. LuGre tire model dynamics is described as below [110]:

$$\dot{z}(t) = v_{r\omega}(t) - \sigma_0 \frac{|v_{r\omega}(t)|}{\mu_{\max}(t) g(v_{r\omega}(t))} z(t), \tag{5.1}$$

$$\mu(t) = \sigma_0 z(t) + \sigma_1 \dot{z}(t) + \sigma_2 v_{r\omega}(t), \tag{5.2}$$

$$F_x(t) = \mu(t) F_n(t), \tag{5.3}$$

where $z(t)$ is the internal state of LuGre, σ_0, σ_1 and σ_2 are nonnegative parameters that describe the tire characteristics, $\mu_{\max}(t)$, the maximum road friction coefficient, represents the road condition and its capacity to provide friction forces, and ω_w is the tire's rotational speed. Here, $\mu(t)$ is the instantaneous road-tire friction coefficient, i.e. the normalized tire traction force corresponding to the tire's normal load. Function $g(v_{r\omega})$ in (5.1) is represented as:

$$g(v_{r\omega}(t)) = \mu_c + (\mu_s - \mu_c) e^{-|\frac{v_{r\omega}(t)}{v_s}|^{0.5}}, \tag{5.4}$$

where μ_c, μ_s and v_s are the tire's parameters, assumed to be constant and known. In (5.1), $v_{r\omega}(t)$ denotes the difference between the tire surface velocity and the vehicle longitudinal velocity. It is formulated as:

$$v_{r\omega}(t) = r_{\text{eff}}\, \omega_w(t) - v_x(t), \tag{5.5}$$

where ω_w is the wheel rotational speed and r_{eff} is the effective tire radius.

Remark 5.1. *In the steady state phase of highway driving, $v_{r\omega}$ changes are relatively small and it can be assumed as a constant parameter.*

The maximum road friction coefficient or *road condition* parameter is the unknown parameter to be estimated. Simply, by substituting (5.1) in (5.2), an algebraic equation for friction can be determined, the discrete form of which can be written as:

$$\mu[k] = \left[\sigma_0 - \sigma_1\sigma_0 \frac{|v_{r\omega}|}{\mu_{\max}[k] g(v_{r\omega})}\right] z[k] + [\sigma_1 + \sigma_2]\, v_{r\omega}[k]. \tag{5.6}$$

The identification scheme relies on (5.6) to form the standard parametric model. A Luenberger observer is used to estimate the internal state $z[k]$.

5.1.2 Parameter Identification

A least square parameter identification algorithm is employed by each vehicle to identify the road friction coefficient. By assuming LuGre tire model parameters $\{\sigma_0, \sigma_1, \sigma_2, \mu_s, \mu_c\}$ and r_{eff} are known, the only unknown parameter in (5.6) is $\mu_{\max}$. To transform (5.6) into the standard parametric model of:

$$f[k] = \mu_{\max}[k]\phi[k], \tag{5.7}$$

we need to reorder (5.6) in order to lump $\mu_{\max}$ in one side of the equation. By multiplying $\mu_{\max}$ into both side of (5.6), we have:

$$\mu[k]\mu_{\max}[k] = \left[\sigma_0\mu_{\max}[k] - \frac{\sigma_1\sigma_0|v_{r\omega}|}{g(v_{r\omega})}\right] z[k] + [\sigma_1 + \sigma_2]\mu_{\max}[k] v_{r\omega}. \tag{5.8}$$

By moving all terms with unknown parameter $\mu_{\max}[k]$ to the right side, we have:

$$\sigma_1\sigma_0 \frac{|v_{r\omega}|}{g(v_{r\omega})} z[k] = \left[\sigma_0 z[k] + [\sigma_1 + \sigma_2] v_{r\omega} - \mu[k]\right]\mu_{\max}[k] \tag{5.9}$$

Accordingly, the parametric models' regressors are:

$$f[k] = \sigma_1 \sigma_0 \frac{|v_{r\omega}|}{g(v_{r\omega})} z[k], \tag{5.10}$$

$$\phi[k] = \sigma_0 z[k] + [\sigma_1 + \sigma_2] v_{r\omega}[k] - \mu[k]. \tag{5.11}$$

Now, by having the parametric model components, we can use a discrete least square parameter identification algorithm for online identification of $\mu_{\max}[k]$.

Discrete Recursive Least Square Identification Algorithm

In 2.3.1 the least square algorithm is introduced. To identify the road friction coefficient for agent i-th, the least square parameter identification scheme is used as [39]:

$$P[k] = P[k-1] - \frac{P^2[k-1]\phi^2[k]}{m^2[k] + \phi^2[k]P[k-1]}, \tag{5.12}$$

$$\varepsilon[k] = \frac{f[k] - \hat{\mu}^{PI}_{\max}[k-1]\phi[k]}{m^2[k]}, \tag{5.13}$$

$$\hat{\mu}^{PI}_{\max}[k] = \hat{\mu}^{PI}_{\max}[k-1] + P[k]\phi[k]\varepsilon[k]. \tag{5.14}$$

where $\hat{\mu}^{PI}_{\max}[0]$ is the initial guess of the unknown parameter. Based on Theorem 4.6.1 in Ref. [39], if $\phi[k]/m[k]$ is *persistently exited*, then the above adaptive law guarantees that $\hat{\mu}^{PI}_{max}[k] \to \mu_{\max}$ as $k \to \infty$.

5.1.3 An Observer for $z[k]$

$z[k]$ is the internal state of the LuGre tire model that is present in $f[k]$ and $\phi[k]$ equations of the identification scheme, therefore a discrete observer is necessary to estimate $z[k]$. To discretize (5.1), an Euler approximation is applied:

$$\dot{z}(t) = \frac{z[k+1] - z[k]}{T_{PI}}, \tag{5.15}$$

where T_{PI} is the sampling time for the identification system. Substituting (5.15) in (5.1) results in:

$$z[k+1] = \left[1 - \sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{\max}[k] g(v_{r\omega}[k])} T_{PI}\right] z[k] + T_{PI} v_{r\omega}[k]. \tag{5.16}$$

The Luenberger observer equations are derived below:

$$\begin{aligned}\hat{z}[k+1] = {} & \left[1 - \sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{\max}[k] g(v_{r\omega}[k])} T_{PI}\right] \hat{z}[k] \\ & + T_{PI} v_{r\omega}[k] + K_{obs}\left[\mu[k] - \hat{\mu}[k]\right], \qquad (5.17) \\ \hat{\mu}[k] = {} & \left[\sigma_0 - \sigma_1 \sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{\max}[k] g(v_{r\omega}[k])}\right] \hat{z}[k] + [\sigma_1 + \sigma_2] v_{r\omega}[k], \qquad (5.18)\end{aligned}$$

which results in:

$$\begin{aligned}\hat{z}[k+1] = {} & \Big[1 + \sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{\max}[k] g(v_{r\omega}[k])} (K_{\text{obs}} \sigma_1 - T_{PI}) - K_{\text{obs}} \sigma_0\Big] \hat{z}[k] \\ & + \big[T_{PI} - K_{\text{obs}}[\sigma_1 + \sigma_2]\big] v_{r\omega}[k] + K_{\text{obs}} \mu[k]. \qquad (5.19)\end{aligned}$$

Theorem 5.1. *For LuGre model* (5.1)-(5.3)*, observer* (5.19) *guarantees that* $\lim_{k\to\infty} \hat{z}[k] = z[k]$ *if* $K_{obs} = \frac{T_{PI}}{\sigma_1}$ *and* $T_{PI} < \frac{2\sigma_1}{\sigma_0}$.

Proof. *Defining observer error as* $\tilde{z}[k] = z[k] - \hat{z}[k]$*, then:*

$$\tilde{z}[k+1] = z[k+1] - \hat{z}[k+1] \tag{5.20}$$

by substituting (5.6), (5.16), (5.19) *and* K_{obs} *value into above equation, we have:*

$$
\begin{aligned}
\tilde{z}[k+1] = & \left[1 - \sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{max}[k] g(v_{r\omega}[k])} T_{PI}\right] z[k] + T_{PI} v_{r\omega}[k] \\
& - \left[1 + \sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{max}[k] g(v_{r\omega}[k])} (K_{obs}\sigma_1 - T_{PI}) - K_{obs}\sigma_0\right] \hat{z}[k] \\
& - \left[T_{PI} - K_{obs}[\sigma_1 + \sigma_2]\right] v_{r\omega}[k] \\
& - K_{obs} \left[\sigma_0 - \sigma_1\sigma_0 \frac{|v_{r\omega}[k]|}{\mu_{max}[k] g(v_{r\omega}[k])}\right] z[k] \\
& - K_{obs} \left[\sigma_1 + \sigma_2\right] v_{r\omega}[k]. \qquad (5.21)
\end{aligned}
$$

Therefore, we have:

$$
\tilde{z}[k+1] = \left[1 - \frac{T_{PI}}{\sigma_1}\sigma_0\right] \tilde{z}[k]. \qquad (5.22)
$$

From discrete system stability analysis, it is known that, the necessary and sufficient condition that the above equation converges to zero is :

$$
-1 < 1 - \frac{T_{PI}}{\sigma_1}\sigma_0 < 1, \qquad (5.23)
$$

which leads to these two conditions:

$$
0 < \frac{T_{PI}}{\sigma_1}\sigma_0 < 2 \qquad (5.24)
$$

Since σ_0 and σ_1 are nonnegative, by having $T_{PI} < \frac{2\sigma_1}{\sigma_0}$, $\tilde{z}[k]$ will converges to zero.

5.2 Multi-agent Consensus Estimation

Consensus schemes improves the individual estimations of a common unknown parameter, e.g. road friction coefficient. However, the road condition is non-constant in the real world and the static consensus law of (2.23) that does not incorporate the agents' updates cannot be applied for cooperative estimation of the road condition. Accordingly, we employed a *dynamic average consensus* [113] scheme for each vehicle to track the average of individually estimated time-varying road condition parameters by local communication with neighbours.

By considering the updating rate of the consensus scheme q times slower than the parameter identification's one, $k = ql$, the dynamic consensus policy of agent i-th is in the following form:

$$\mu^{\text{cons}}_{\max i}[l+1] = w_{ii}\mu^{\text{cons}}_{\max i}[l] + \sum_{j \in \mathcal{N}_i} w_{ij}\mu^{\text{cons}}_{\max j}[l] + (\hat{\mu}^{PI}_{\max i}[ql] - \hat{\mu}^{PI}_{\max i}[ql-q]) \tag{5.25}$$

where $\mu^{\text{cons}}_{\max i}[l]$ is the consensus estimate of $\mu_{\max}$ for agent i at $t = l\ T_c$, T_c is the consensus sampling time, $\mu^{\text{cons}}_{\max i}[0] = \hat{\mu}^{PI}_{\max i}[0]$ and w_{ij} is the weight of $\mu^{\text{cons}}_{\max j}$ at node i.

It is shown in [113], for a periodically strong connected network, the above dynamic consensus (5.25) with a weighting matrix that satisfies (2.25)–(2.27) conditions, would track the average of the agents' individually estimated road condition parameter, $\bar{\mu}^{PI}_{\max}[ql] = \frac{1}{N}\Sigma^{N}_{i=1}\hat{\mu}^{PI}_{\max i}[ql]$ with some bounded steady-state error.

A *constant edge weight* scheme, which is introduced in Section 2.3.2, is used for edges weighting. We can assume the maximum degree of each agent in the vehicular network is less than $d_{\max}$ due to limited communication channels and other limitations. Therefore, we can assign constant weight $\alpha = \frac{1}{1+d_{\max}}$:

$$w_{ij} = \begin{cases} \frac{1}{1+d_{\max}} & i, j \in \mathcal{E}, \\ 1 - d_i \frac{1}{1+d_{\max}} & i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{5.26}$$

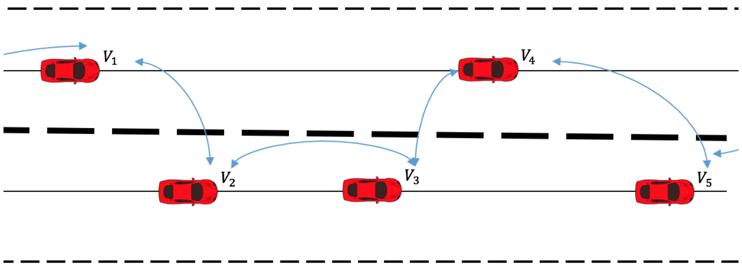


Figure 5.1: A typical configuration of 5 intelligent vehicles driving in a two-lane road with communication links between them.

5.3 Simulation Results

In this section, first, the performance of the proposed cooperative road condition estimation technique is studied; then, the simulation results for the adaptive model predictive path planner with this cooperative road condition estimator are presented.

5.3.1 Cooperative road condition estimation

A simulation scenario with 5 connected vehicles $\{V_1, V_2, V_3, V_4, V_5\}$ is considered. All vehicles are equipped with V2V communication and road condition estimation systems. Due to network constraints such as communication range and receivers limitation, it is assumed that each vehicle only communicates with the two nearest vehicles, e.g. one in front and one at rear. Each vehicle disseminates its data at 10 Hz. A typical configuration of this group of intelligent vehicles with their communication links is depicted in Fig. 5.1.

For online parameter identification, a high-fidelity vehicle model in CarSim is used, which involves a full-vehicle multi-body dynamics model, including dynamics of tires, wheels, powertrain, and chassis. The front left tire, represented by fL, is considered for the road condition identification and $v_{x_{fL}}$, $\omega_{w_{fL}}$ and μ_{fL} are measured for this tire/wheel. It is assumed that $v_{x_{fL}}$ is estimated by integrating the output of the inertial measurement unit (IMU). To represent real-world application scenarios, random noise with Gaussian distribution is added to the IMU's measurement.

In section 5.1, we assumed the LuGre tire parameters $\{\sigma_0, \sigma_1, \sigma_2, \mu_c, \mu_s, v_s\}$ are known for online road condition estimation. Accordingly, these tire parameters should be identified in advance. For this purpose, for a road with the known road condition coefficient, $v_{x_{fL}}$, $\omega_{w_{fL}}$ and μ_{fL} data and LuGre model equations are used by a nonlinear least-squares curve fitting scheme to identify the CarSim's tire parameters. The resulting LuGre tire parameters are shown in Table 5.1.

Table 5.1: Identified LuGre tire parameters.

$\sigma_0 = 125$	$\sigma_1 = 0.85$
$\sigma_2 = 0.106$	$\mu_c = 2.35$
$\mu_s = 2.35$	$v_s = 21.0$ [m/s]

To satisfy the persistent excitation condition, we assumed that vehicles are equipped with an active transfer case system that allows flexible distribution of the engine torque between front and rear axles. Accordingly, a relatively high frequency, low amplitude sinusoidal signal is added to the transfer case controller. As a result, we can persistently excite the front and rear wheels' rotational speed without changing the vehicle longitudinal velocity. The front and real wheels' rotational speed of V_1 are shown in Fig. 5.2a. It is clear that due to different torque distributions to front and rear axles, rotational speeds at these two wheels are not equal and thereby fluctuate. In Fig. 5.2b, v_x profile for V_1 is demonstrated. We can see that, the employed torque distribution does not affect the general dynamics of the vehicle.

It should be noted that persistently rich excitation of a tire for system identification application may affect the life span of the tire. The effect of high-frequency low-amplitude torque distribution of front and rear wheels on the tire life is out of the scope of this study and should be investigated separately. We can only argue that this probable effect is the cost that should be considered for accurate road condition identification for active safety application in all road conditions.

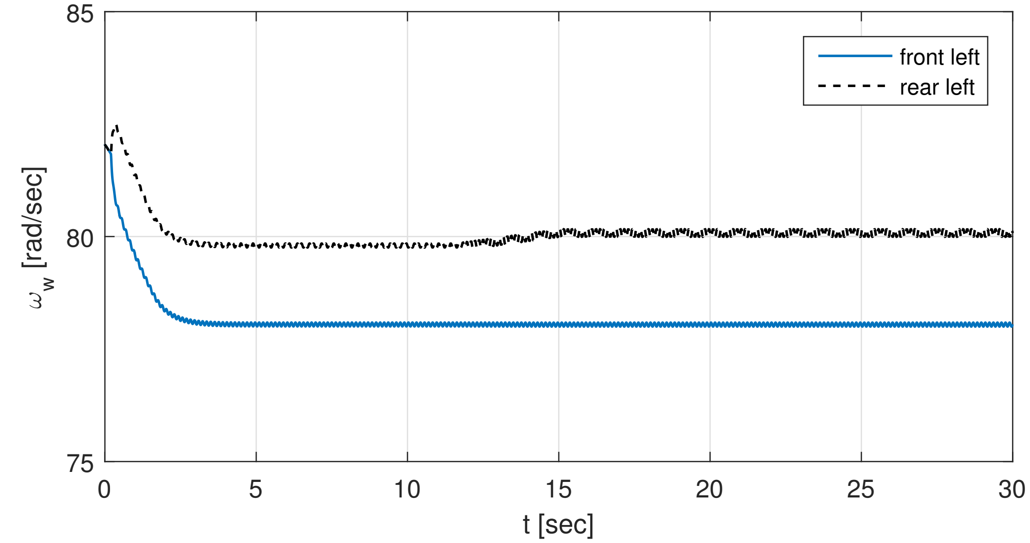
A situation with time-varying road condition is considered for the simulation analysis to see better the performance of the cooperative road condition estimation system presented. It is assumed that the road condition is changed in the middle of the simulation from $\mu_{\max} = 0.8$ to $\mu_{\max} = 0.5$; thus vehicles need to update their estimates to determine the new $\mu_{\max}$ value.

The simulation results of the observer used in vehicle V_1 to estimate the internal state of the LuGre tire model is shown in Fig. 5.3. It illustrates that the estimated $\hat{z}$ perfectly follows the true value of the internal state. Thus, the effect of the initial observer error on the road condition identification is negligible. Vehicle V_1's local least square road condition identification results are illustrated in Fig. 5.4 for both the ideal situation, with no noise in measurements, and the real one with noises. The results are also compared with the true value of the road condition parameter. We can see that in the ideal situation, the online identification perfectly follows the true value of $\mu_{\max}$, but incorporation of measurement noise would cause a considerable amount of error in the identification. All five vehicles' local estimates are shown in Fig. 5.5. In addition, the instantaneous average of all agents' measurements is shown in Fig. 5.6. Clearly, in presence of independent random noises, the average of agents' parameter identification is significantly closer to the true value of the road condition parameter.

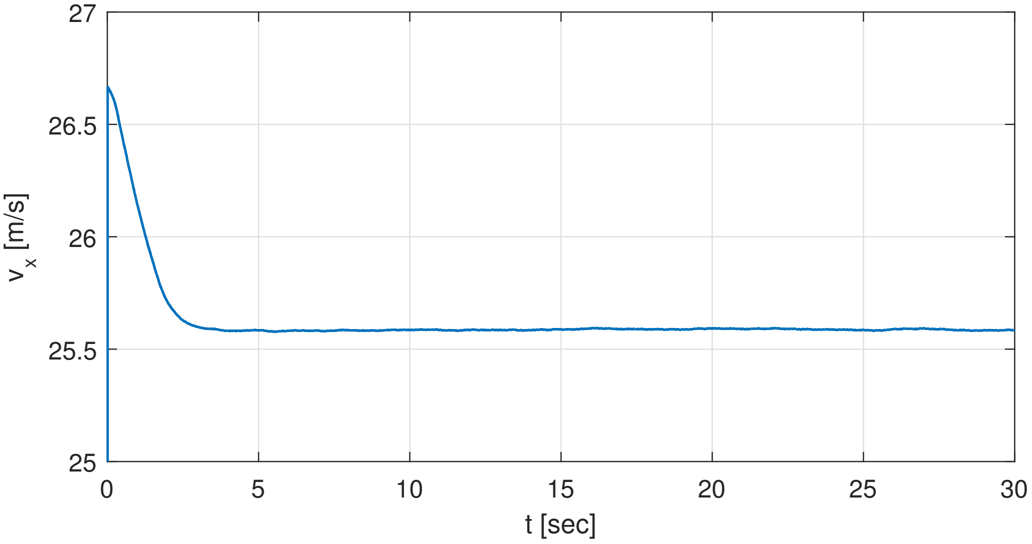
The cooperative road condition estimation result for vehicle V_1 is presented in Fig. 5.7. In this figure, the cooperative estimate somehow follows the network's average. It is clear that the cooperative estimation result is much closer to the actual road condition value than single-agent estimation. The cooperative $\mu_{\max}$ estimates of all five agents are shown in Fig. 5.8. We can see in this figure that the proposed cooperative estimation scheme improves the $\mu_{\max}$ estimation for all agents in the vehicular network.

As a comparison index, the summation of absolute estimation errors is considered:

$$J_{est} = \sum_{l=0}^{l_{sim}} |\hat{\mu}_{\max}[l] - \mu^*_{\max}[l]| \tag{5.27}$$



(a) Left wheels rotational speed for vehicle V_1.



(b) Vehicle longitudinal velocity for vehicle V_1.

Figure 5.2: Vehicle V_1's measured states for road condition identification.

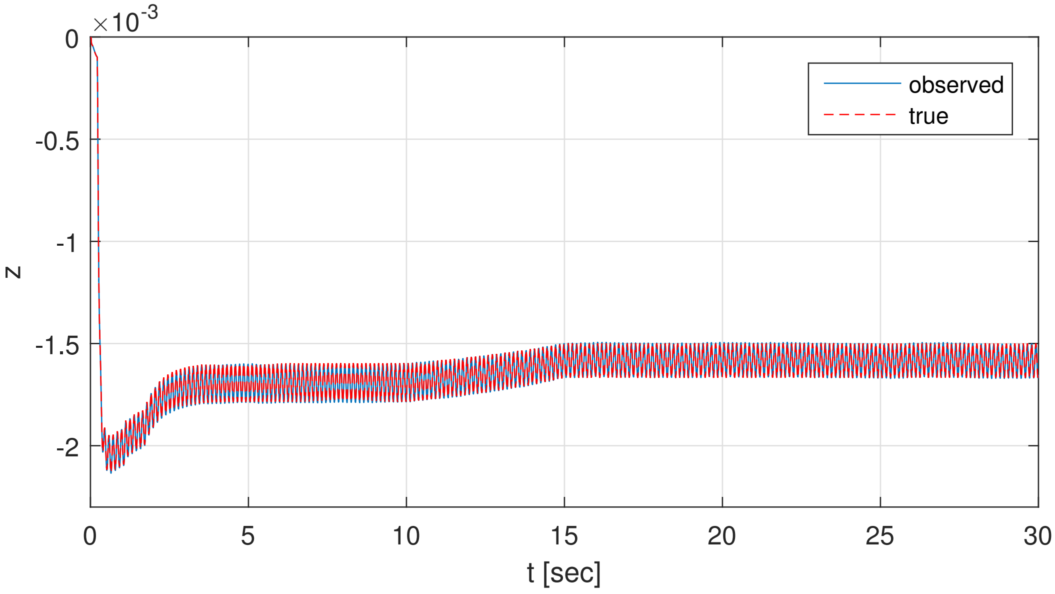


Figure 5.3: Internal state of the LuGre model for vehicle V_1.

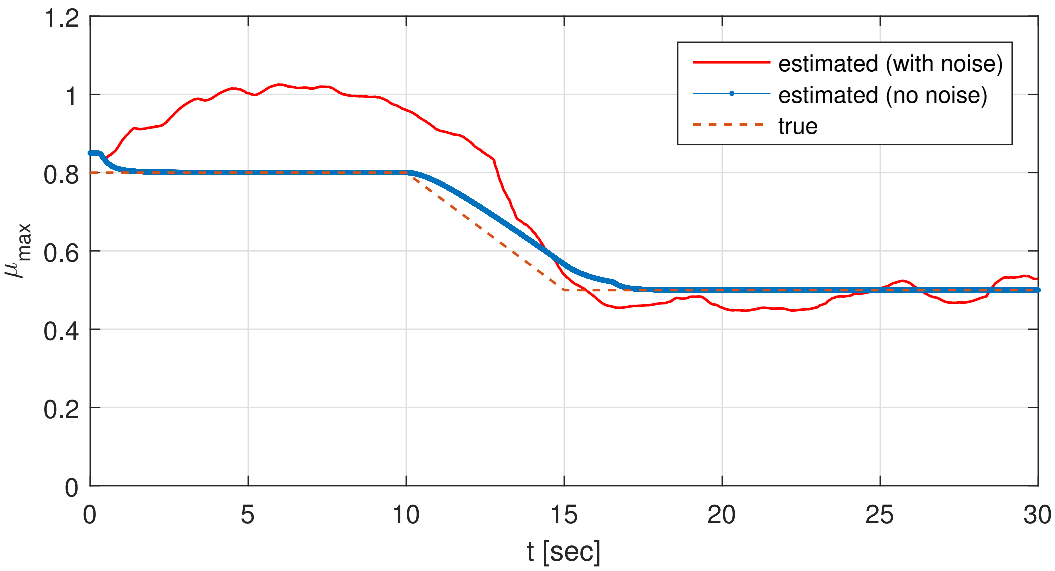


Figure 5.4: The comparison of the real road friction coefficient and estimated one for vehicle V_1.

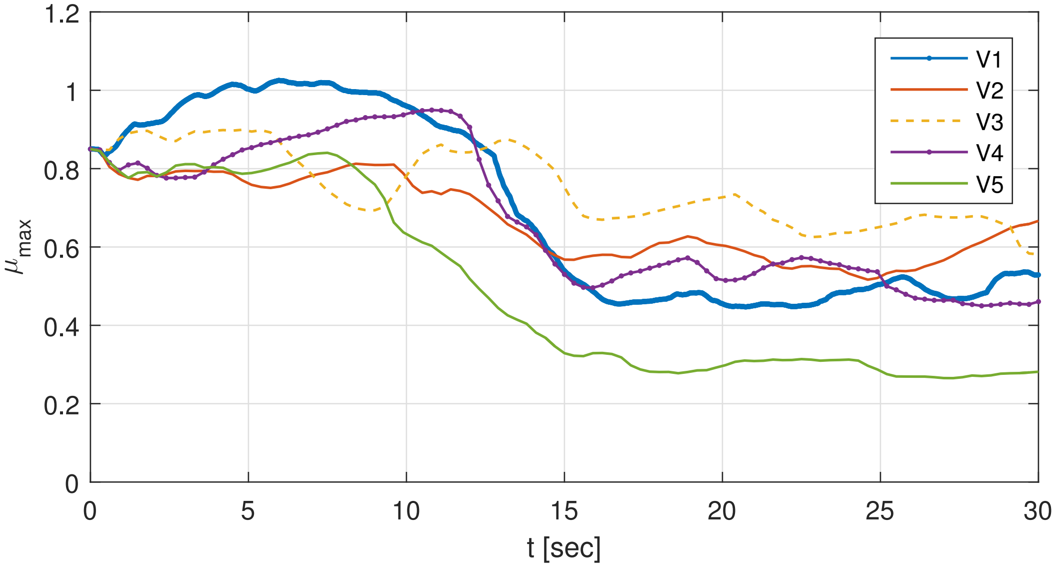


Figure 5.5: The road condition estimation simulation results for all vehicles in the network.

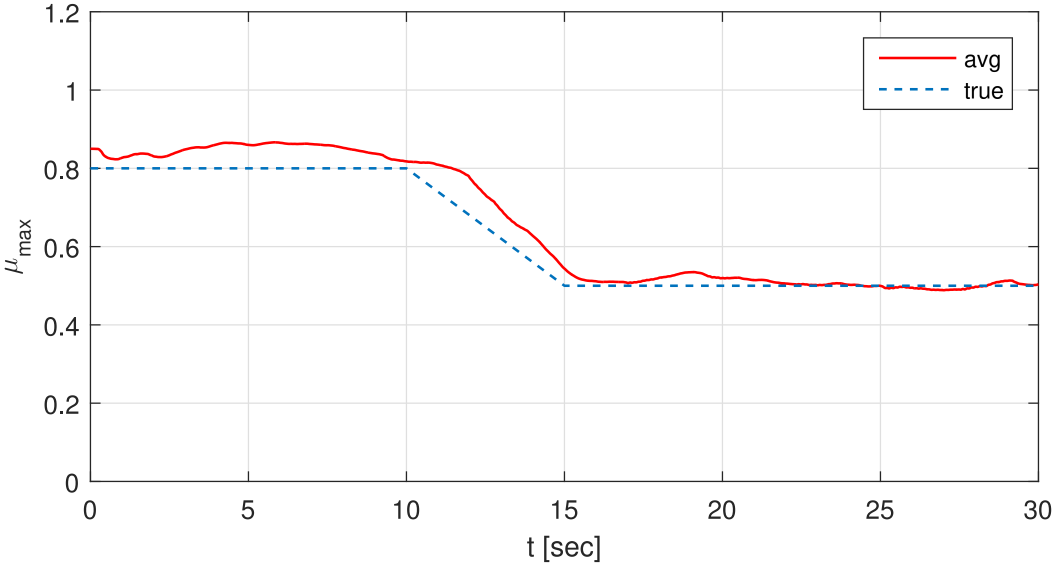


Figure 5.6: The comparison of the real road friction coefficient and the average of all agents' estimates.

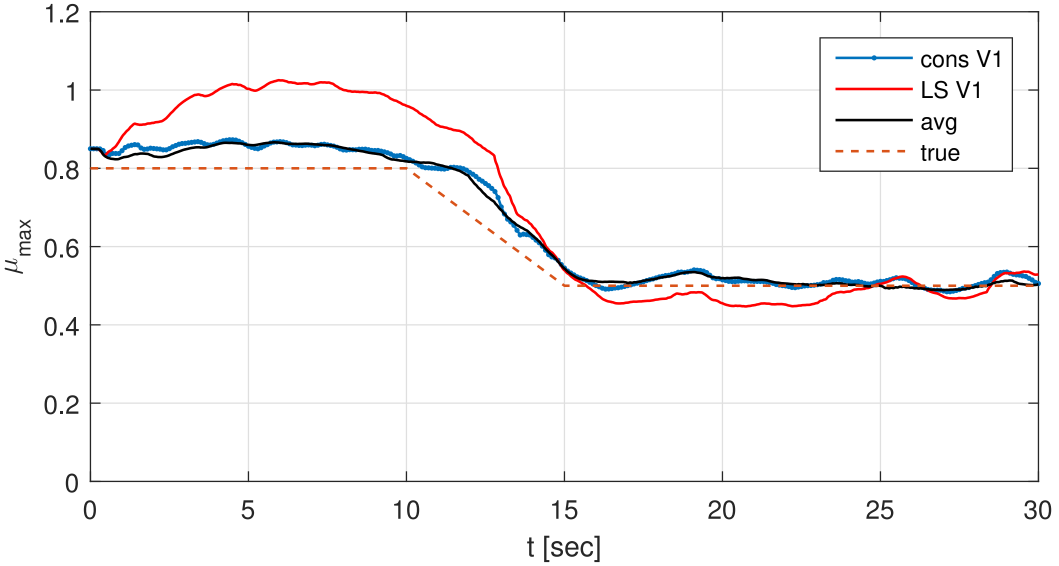


Figure 5.7: The vehicle $V1$'s cooperative road condition estimate, in comparison with results of V_1's individual least square (LS) estimate and also averages of all agents' estimates.

where l_{sim} is simulation time span, $\hat{\mu}_{\max}[k]$ is the estimated road condition, and $\mu^*_{\max}[k]$ is the true value of road condition at step k^{th}. J_{est} for single-agent (vehicle V_1, dashed blue) and cooperative (red) estimation, together with the average of all vehicle estimates are shown in Fig. 5.9. This figure shows distinctly the ascendancy and significance of the consensus approach to eliminate unbiased white noise effects from on-line estimation. Cooperative estimation's J_{est} profile is very similar to the network average, and it is 34% less than single-agent estimation.

The effect of increasing the number of network nodes is demonstrated in Fig. 5.10. In a network with 10 agents the average of all agents' estimates is closer to the true parameter value and each agent's cooperative estimation is more accurate in comparison to the network with 5 agents, but the improvement is not significant, at least not for this set of simulation results.

Network topology can also change the quality of consensus cooperative estimation. If the number of connected agents increases, then the consensus will converge faster to the network's average. In Fig. 5.11 the comparison between the consensus estimation of the two networks is

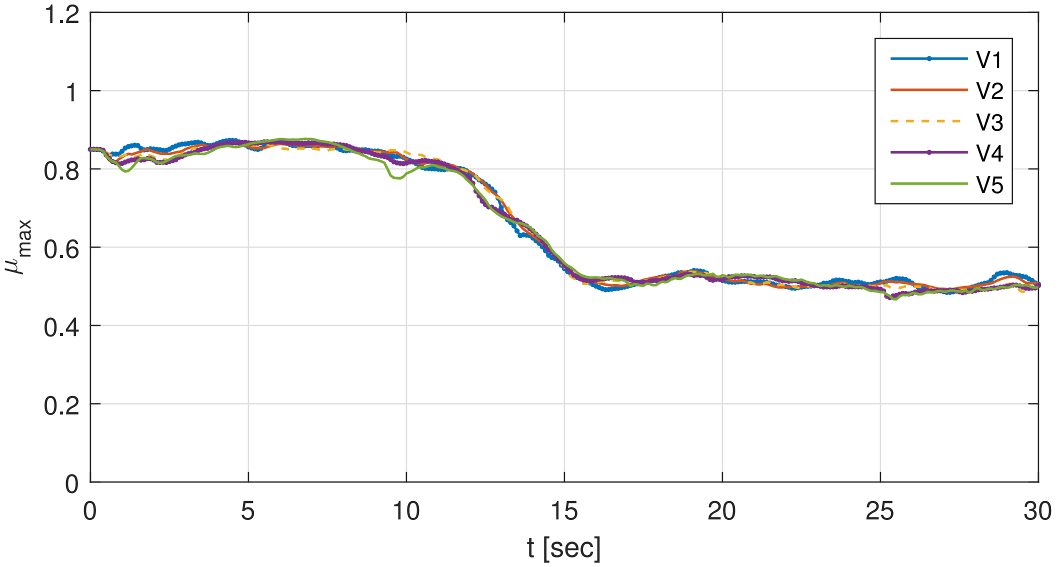


Figure 5.8: The cooperative road condition estimation results for all vehicles in the network.

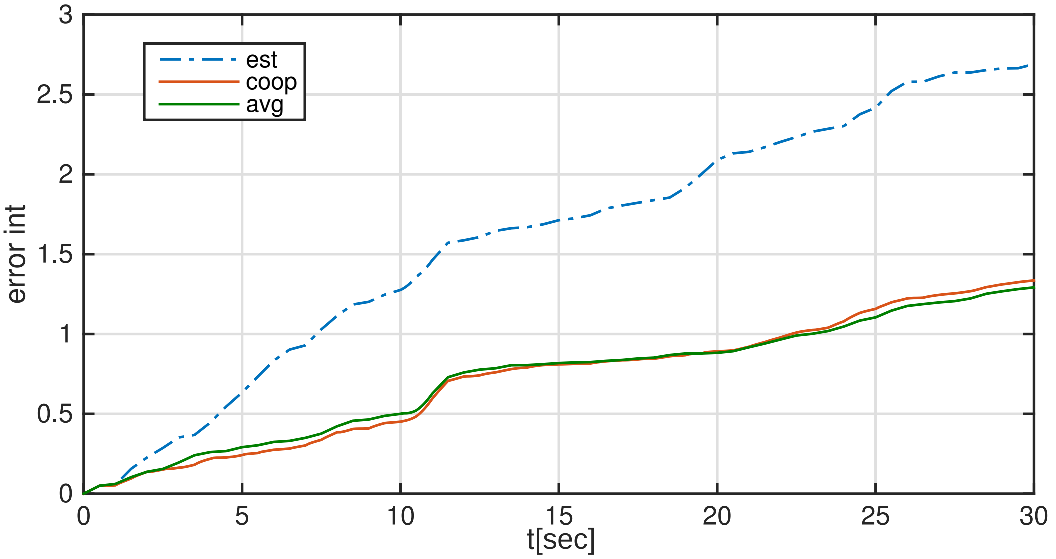


Figure 5.9: Estimation error integrals for single-agent estimation, consensus, and total averages.

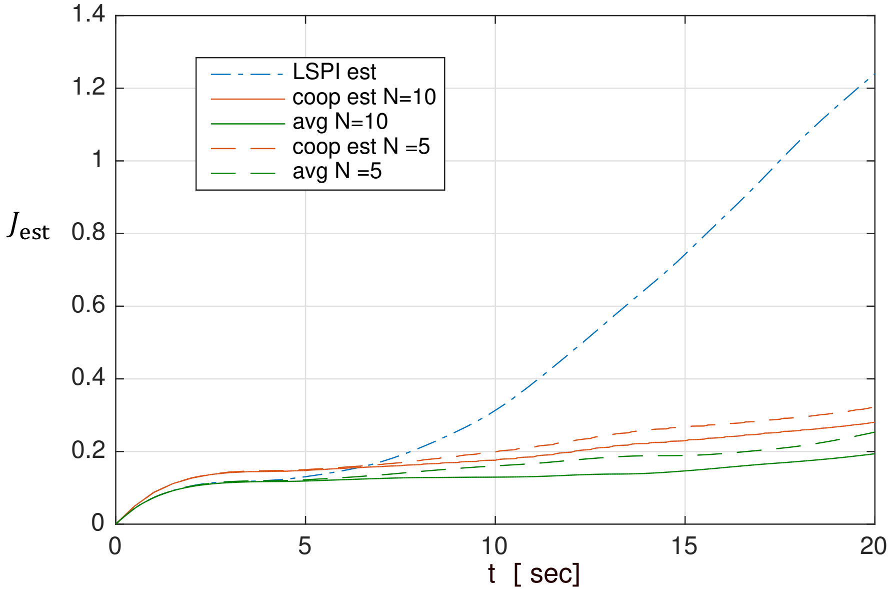


Figure 5.10: Changing the number of nodes in the network.

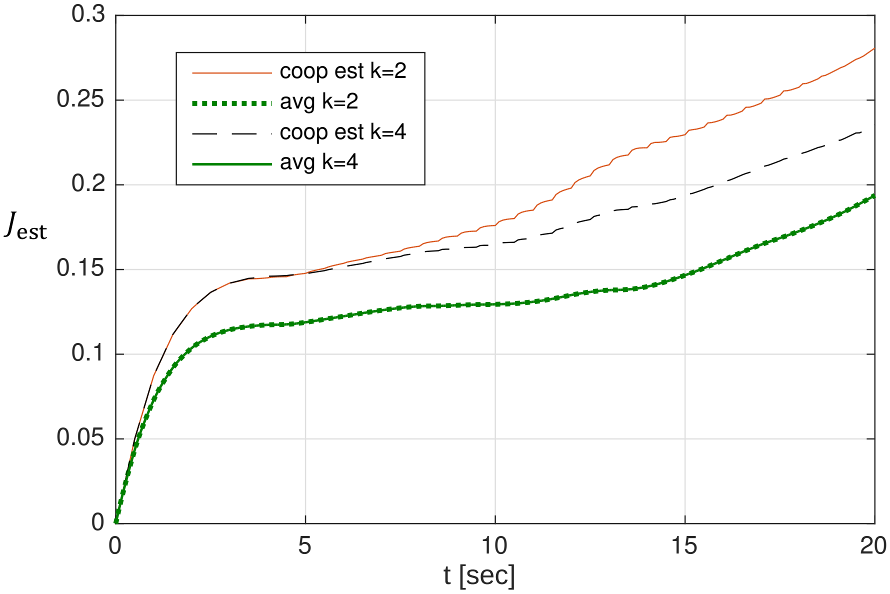


Figure 5.11: Changing the number of connected neighbours.

presented. In one network each agent is connected to two adjacent agents and in the other one, each agent is connected to four surrounding vehicles. Although the networks' averages are the same, the accumulative error of the cooperative estimation of the second network is less than the first one.

5.3.2 Adaptive Model Predictive Collision Avoidance Control Simulation Results

In this section, we study the simulation results of the emergency maneuver planning that uses the proposed cooperative road condition estimation scheme presented in Chapter 5. It is investigated whether the accuracy of single-agent estimation is satisfying for the emergency maneuver planning or if we need cooperative estimation to avoid obstacles.

A challenging driving scenario is defined to test the proposed parameter estimation. It is assumed that in a two-lane road, there is a stationary obstacle in each lane with a rather narrow distance between the obstacles. Similar to Section 5.3.1, the cooperative estimation scheme of Section 5.2 is applied in a network of 5 communicating vehicles.

The simulation parameters are specified in Table 5.2. The true road friction value for this scenario is $\mu_{\max} = 0.5$, which is slightly low and referring to wet road condition. The model predictive emergency maneuver planning has to to decide how to react when it confronts obstacles, based on factors such as the vehicle speed, obstacles distance, road condition, and vehicle maneuvering capabilities. The resulting estimation and trajectory profiles for the first agent are shown in Fig. 5.12. In Figs. 5.12(b) and 5.12(c) the obstacles are represented by green rectangles, and the safety boundaries around them are depicted by ellipses.

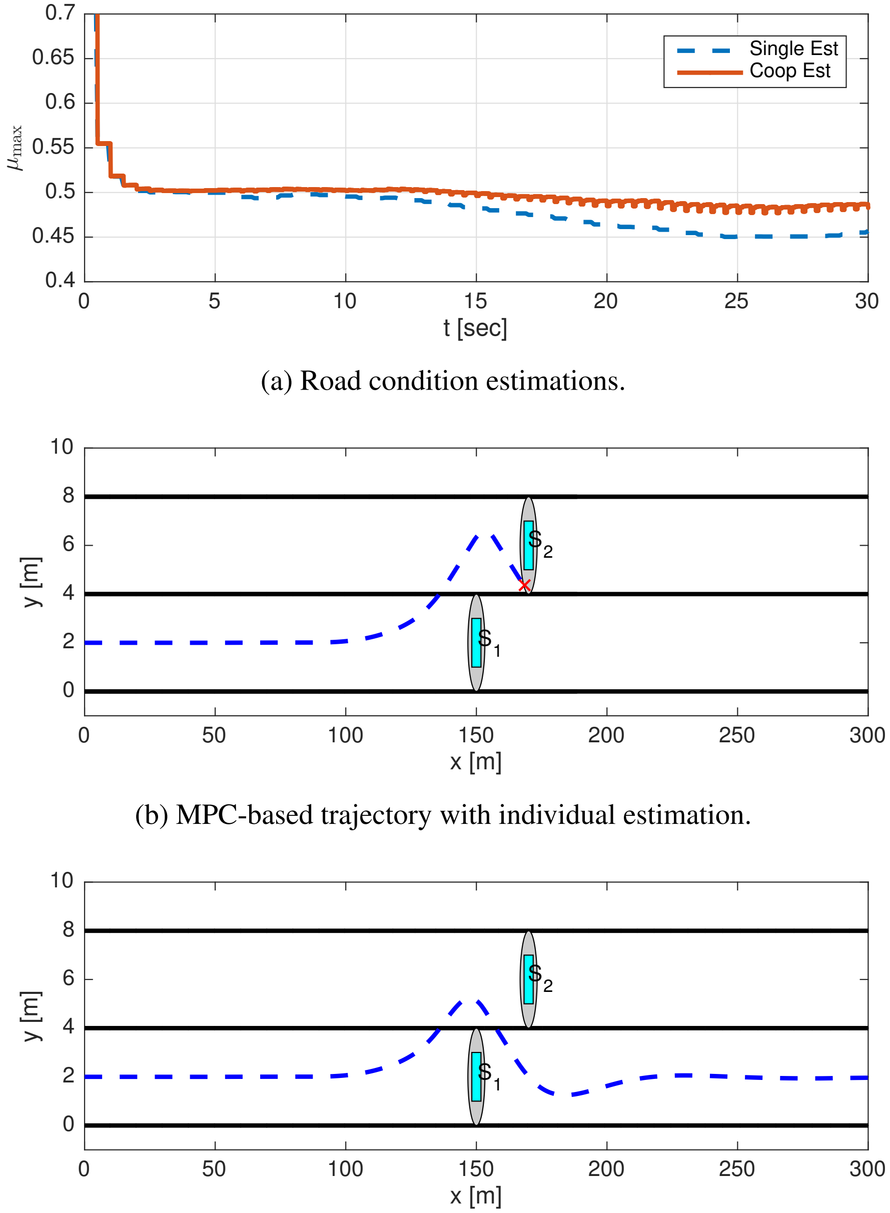
As seen in Fig. 5.12(a), single-agent road condition estimation results in less accurate estimation. While the cooperative estimation error is less than 5%, single-agent road condition estimation is about 10% in this case. This inaccuracy is critical for maneuver planning in some scenarios. In Figs. 5.12(b) and 5.12(c) all simulation characteristics are the same except that in Fig. 5.12(b) single-agent road condition identification technique is used, and in Fig. 5.12(c) we

Table 5.2: Simulation parameters.

$y_{\text{road}_{\min}} = 0$ [m]	$y_{\text{road}_{\max}} = 8$ [m]
$y_{\text{lane}_1} = 2$ [m]	$y_{\text{lane}_2} = 6$ [m]
$v_{x_{\min}} = 0$ [m/s]	$v_{y_{\max}} = 30$ [m/s]
$v_{x_0} = 20$ [m/s]	$v_{y_0} = 0$ [m/s]
$x_0 = 0$ [m]	$y_0 = 2$ [m]
$x_{obs1_0} = 150$ [m]	$y_{obs1_0} = 2$ [m]
$x_{obs2_0} = 170$ [m]	$y_{obs2_0} = 6$ [m]

used cooperative estimation for road condition estimation scheme. It can be observed that the model predictive maneuver planning with single-agent estimation ends in a collision with the second obstacle, while the maneuver planning with cooperative estimation successfully shifts between lanes and overtakes the obstacles.

The resulting maneuver commands of the employed maneuver planning strategy are shown in Fig. 5.13. This figure indicates that the non-accurate single agent estimation causes the adaptive MPC to generate acceleration commands that fluctuate. The speed profiles of the autonomous vehicles are shown in Fig. 5.14. It can be observed in Fig. 5.14(a) that maneuver planning scheme in both cases make speed reduction during lane shifting and overtaking the obstacles, but this speed reduction is significantly larger for the case with single-agent estimation, which finally decided to have hard braking around $t = 15$ sec to mitigate collision with the obstacle in the left lane.

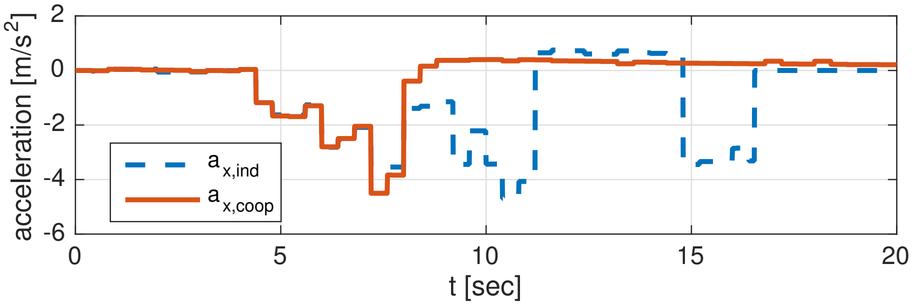


(a) Road condition estimations.

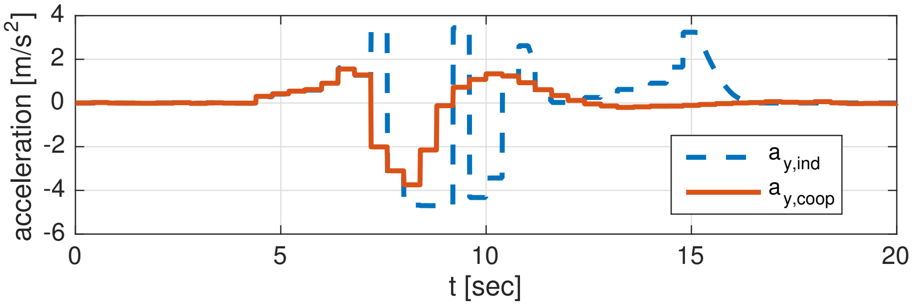
(b) MPC-based trajectory with individual estimation.

(c) MPC-based trajectory with cooperative estimation.

Figure 5.12: Trajectories of host vehicle in presence of obstacles and unknown road condition.

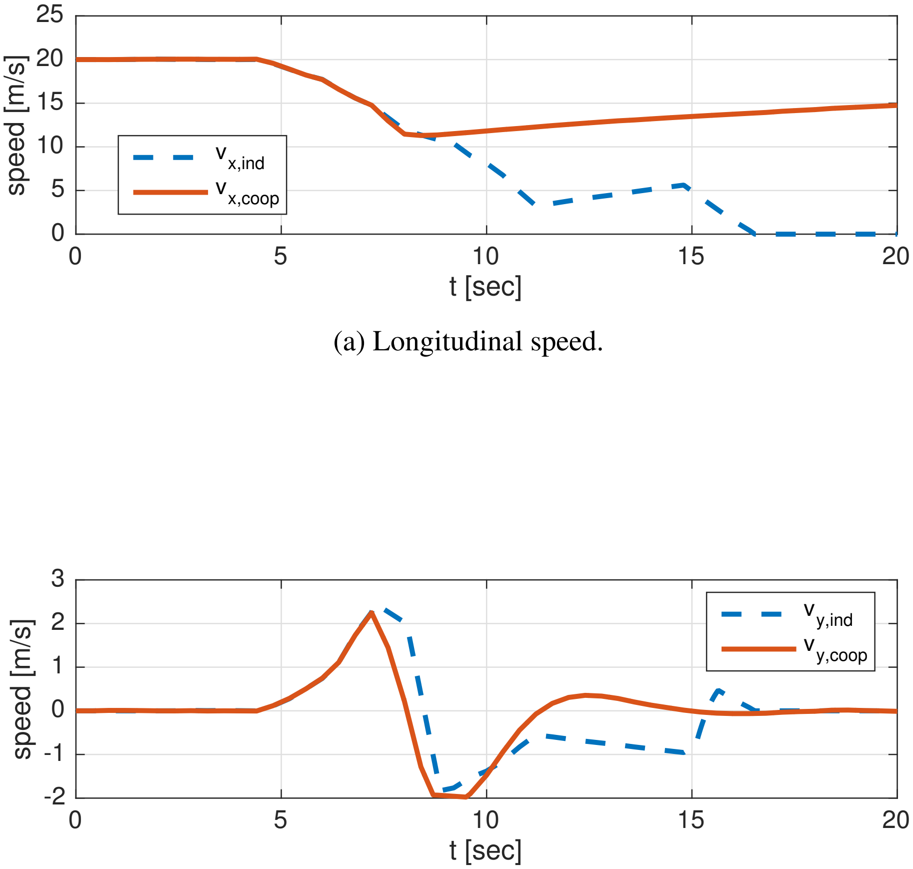


(a) Longitudinal accelerations.



(b) Lateral accelerations.

Figure 5.13: MPC-commanded accelerations based on the individual and cooperative estimates.



(a) Longitudinal speed.

(b) Lateral speed.

Figure 5.14: Resulting vehicle speed profiles due to employing the adaptive MPC with individual or cooperative estimators.

5.4 Summary

An on-line cooperative estimation technique for the vehicular networks is presented in this study, including a pair of least square parameter identification and constant edge weight consensus algorithm for each vehicle. The approach is used for cooperative estimation of the road condition in a small group of vehicles connected with V2V communicating devices.

The simulation results show significant improvement of cooperative estimation compared to individual estimation in presence of measurement noises and network communication latency. Additionally, the simulation results confirm the effect of a network's node and edge on convergence of the consensus algorithm.

Chapter 6

Conclusion and Future Work

In this thesis, we developed an adaptive, cooperative motion planning scheme for emergency maneuvering, based on the model predictive control (MPC) approach, for vehicles within a vehicular network. The proposed emergency maneuver planning scheme finds the best combination of longitudinal and lateral maneuvers to avoid imminent collision with surrounding vehicles and obstacles. To provide real-time implementable MPC for the non-convex problem of collision free motion planning, safety constraints are suggested to be convexified based on the road geometry. Surrounding vehicles' accelerations are incorporated in the model predictive decision-making to have better trajectory prediction of these vehicles. Two scenarios including lane-shifting and complete stopping are evaluated for the proposed control scheme. Simulation results confirm the proper activation of collision avoidance commands. Convexified safety constraints for the same lane obstacles successfully enforce the vehicle to shift the lane, if no obstacle is in the next lane.

The conditions that guarantee persistent feasibility of a model predictive motion planning scheme are studied in this thesis. Two efficient control invariant set computation approaches, convexification approach, and brute-force search approach are investigated to guarantee persistent feasibility of the model predictive motion planning problem for autonomous driving vehicles. It is shown that the convexification approach is more conservative and eliminates many feasible maneuvers for the motion planning. On the other hand, using the brute-force technique, an efficient technique is suggested to compute the control invariant sets offline, then the resulting non-invariant area around an obstacle is characterized by two parameters that can be stored in look-up tables for real-time application.

The cooperative road condition estimation advantages to improve collision avoidance performance of the AEM system is investigated. Accordingly, a consensus estimation algorithm is employed to fuse the individual vehicles' estimates to find the maximum likelihood estimate of the road condition parameter. The simulation results show significant improvement of cooperative estimation compared to individual estimation in the presence of measurement noises and network communication latency. Additionally, the simulation results confirm the effect of a network's node and edge on convergence of the consensus algorithm.

A potential future work direction is extension of the proposed MPC based maneuver planner to incorporate the intended future trajectory and acceleration of surrounding vehicle to provide more accurate prediction about them and therefore, decide more cooperatively.

Another future research direction to address the effects of variations in the road condition is improvement of the constant-weighting based consensus algorithm studied in Chapter 5, using a more sophisticated weighting logic based on the inter-vehicle distance. This approach will reduce the effect of agents that have different road condition but are far from the vehicle. The proposed weighting logic has to meet the convergence conditions introduced in Section 2.3.2.

Road condition prediction is an advantage provided by cooperative estimation and we plan to study its effect on MPC planning performance. Storing the position and estimation data of connected agents in the network can be used for generating a mapping of road condition versus agents positions to predict upcoming road condition.

References

[1] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma. Constraint-based planning and control for safe, semi-autonomous operation of vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 383–388, 2012.

[2] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems*, 8(2-4):190–216, 2010.

[3] A. Angelova, A. Krizhevsky, and V. Vanhoucke. Pedestrian detection with a large-field-of-view deep network. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 704–711, 2015.

[4] G. Aoude, J. Joseph, N. Roy, and J. How. Mobile agent trajectory prediction using bayesian nonparametric reachability trees. In *Proceedings of AIAA Infotech@ Aerospace*, pages 1587–1593, 2011.

[5] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, et al. Odin: Team victortango's entry in the darpa urban challenge. *Journal of Field Robotics*, 25(8):467–492, 2008.

[6] C. R. Baker, D. I. Ferguson, and J. M. Dolan. Robust mission execution for autonomous urban driving. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, pages 178–184, 2008.

[7] J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10(2-4):121–155, 1993.

[8] M. Bertozzi et al. Viac: An out of ordinary experiment. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 175–180, 2011.

[9] L. Blackmore, M. Ono, and B. C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011.

[10] F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[11] F. Borrelli, P. Falcone, T. Keviczky, and J. Asgari. MPC-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2-4):265–291, 2005.

[12] M. Brännström, E. Coelingh, and J. Sjöberg. Decision-making on when to brake and when to steer to avoid a collision. *International Journal of Vehicle Safety 1*, 7(1):87–106, 2014.

[13] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli. Predictive control of an autonomous ground vehicle using an iterative linearization approach. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 2335–2340, 2013.

[14] Y. Chen and J. Wang. Adaptive vehicle speed control with input injections for longitudinal motion independent road frictional condition estimation. *IEEE Transactions in Vehicular technology*, 60:839–848, 2011.

[15] H. Cho, P. E. Rybski, A. Bar-Hillel, and W. Zhang. Real-time pedestrian detection with deformable part models. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1035–1042, 2012.

[16] C. Choi, Y. Kang, and S. Lee. Emergency collision avoidance maneuver based on nonlinear model predictive control. In *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6, 2012.

[17] E. Coelingh, A. Eidehall, and M. Bengtsson. Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 155–160, 2010.

[18] G. de Campos, P. Falcone, and J. Sjoberg. Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 1456–1461, 2013.

[19] A. De la Escalera, J. M. Armingol, and M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and vision computing*, 21(3):247–258, 2003.

[20] A. Eckert, B. Hartmann, M. Sevenich, and P. Rieth. Emergency steer & brake assist: a systematic approach for system integration of two complementary driver assistance systems. In *Proceedings of the International Technical Conference on the Enhanced Safety of Vehicles*, 2011.

[21] M. Efatmaneshnik, A. Kealy, N. Alam, and A. Dempster. A cooperative positioning algorithm for dsrc enabled vehicular networks. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 22, 2011.

[22] A. Eidehall. Multi-target threat assessment for automotive applications. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 433–438, 2011.

[23] A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark. Toward autonomous collision avoidance by steering. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):84–94, 2007.

[24] N. Fairfield and C. Urmson. Traffic light mapping and detection. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5421–5426, 2011.

[25] P. Falcone, M. Ali, and J. Sjöberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1352–1361, 2011.

[26] M. Fausten. Accident avoidance by evasive manoevres. In *Proceedings of the Tagung Sicherheit durch Fahrerassistenz*, 2010.

[27] D. Ferguson, M. Darms, C. Urmson, and S. Kolski. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1149–1154, 2008.

[28] Y. Gao. *Model Predictive Control for Autonomous and Semiautonomous Vehicles*. PhD thesis, University of California, Berkeley, 2014.

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

[30] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In *Proceedings of ASME Dynamic Systems and Control Conference*, pages 265–272, 2010.

[31] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[32] A. Gurov, A. Sengupta, M. Jonasson, and L. Drugge. Collision avoidance driver assistance system using combined active braking and steering. In *Proceedings of the International Symposium on Advanced Vehicle Control*, 2014.

[33] F. Gustaffson. Slip-based tire–road friction estimation. *Automatica*, 33:1087–1099, 1997.

[34] J. O. Hahn, R. Rajamani, and L. Alexander. GPS-based real-time identification of tire–road friction coefficient. *IEEE Transaction on Control Systems Technology*, 10:331–343, 2002.

[35] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proceedings of the European Control Conference*, pages 502–510, Zürich, Switzerland, 2013. `http://control.ee.ethz.ch/~mpt`.

[36] J. Hillenbrand, A. M. Spieker, and K. Kroschel. A multilevel collision mitigation approachits situation assessment, decision making, and performance tradeoffs. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):528–540, 2006.

[37] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 4363–4369, 2013.

[38] R. Hult, F. E. Sancar, M. Jalalmaad, A. Vijayan, , A. Severinson, M. Di Vaio, P. Falcone, B. Fidan, and S. Santini. Design and experimental validation of a cooperative driving control architecture for the grand cooperative driving challenge 2016. *Transactions on Intelligent Transportation Systems*, (Special Issue on Grand Cooperative Driving Challenge), 2017.

[39] P. A. Ioannou and B. Fidan. *Adaptive Control Tutorial*. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2006.

[40] M. Jalalmaab, B. Fidan, S. Jeon, and P. Falcone. Model predictive path planning with time-varying safety constraints for highway autonomous driving. In *Proceedings of the IEEE International Conference on Advanced Robotics*, pages 213–217, 2015.

[41] M. Jalalmaab, B. Fidan, S. Jeon, and P. Falcone. Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 843–848, 2017.

[42] M. Jalalmaab, M. Pirani, B. Fidan, and S. Jeon. Cooperative least square parameter identification by consensus within the network of autonomous vehicles. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 9(2016-01-0149), 2016.

[43] M. Jalalmaab, M. Pirani, B. Fidan, and S. Jeon. Cooperative road condition estimation for an adaptive model predictive collision avoidance control strategy. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1072–1077, 2016.

[44] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. v. Hundelshausen, et al. Team annieway's autonomous system for the 2007 darpa urban challenge. *Journal of Field Robotics*, 25(9):615–639, 2008.

[45] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila. Active pedestrian safety by automatic braking and evasive steering. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1292–1304, 2011.

[46] A. Kelly and B. Nagy. Reactive nonholonomic trajectory generation via parametric optimal control. *The International Journal of Robotics Research*, 22(7-8):583–601, 2003.

[47] R. Kianfar, P. Falcone, and J. Fredriksson. A receding horizon approach to string stable cooperative adaptive cruise control. In *Proceedings of the IEEE Intelligent Transportation Systems*, pages 734–739, 2011.

[48] R. Kianfar, P. Falcone, and J. Fredriksson. Reachability analysis of cooperative adaptive cruise controller. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 1537–1542, 2012.

[49] J. Klappstein, F. Stein, and U. Franke. Monocular motion detection using spatial constraints in a unified manner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 261–267, 2006.

[50] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1094–1099, 2015.

[51] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng. Explicit use of probabilistic distributions in linear predictive control. *Automatica*, 46(10):1719–1724, 2010.

[52] A. Kushleyev and M. Likhachev. Time-bounded lattice for efficient planning in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1662–1668, 2009.

[53] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.

[54] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How. Motion planning in complex environments using closed-loop prediction. In *Proceedings of AIAA Guidance, Navigation, and Control Conf. and Exhibit*, 2008.

[55] C.-C. Lai and W.-H. Tsai. Location estimation and trajectory prediction of moving lateral vehicle using two wheel shapes information in 2-d lateral vehicle images by 3-d computer vision techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 881–886, 2003.

[56] S. M. LaValle, J. J. Kuffner, and Jr. Rapidly-exploring random trees: Progress and prospects, 2000.

[57] C. Lee, K. Hedrick, and K. S. Yi. Real-time slip estimation and maximum tire–road friction coefficient. *IEEE Transactions on Vehicular Technology*, 9:454–458, 2004.

[58] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1(1):1, 2014.

[59] J. Levinson, J. Askeland, J. Dolson, and S. Thrun. Traffic light mapping, localization, and state detection for autonomous vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5784–5791, 2011.

[60] J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4372–4378, 2010.

[61] C. Liu, A. Gray, C. Lee, J. K. Hedrick, and J. Pan. Nonlinear stochastic predictive control with unscented transformation for semi-autonomous vehicles. In *Proceedings of the IEEE American Control Conference (ACC)*, pages 5574–5579, 2014.

[62] K. Liu and H. B. Lim. Positioning accuracy improvement via distributed location estimate in cooperative vehicular networks. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 1549–1554, 2012.

[63] J. Löfberg. Oops! I cannot do it again: Testing for recursive feasibility in mpc. *Automatica*, 48(3):550–555, 2012.

[64] H. Loose and U. Franke. B-spline-based road model for 3d lane recognition. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 91–98, 2010.

[65] M. Lu, K. Wevers, and R. Van Der Heijden. Technical feasibility of advanced driver assistance systems (adas) for road traffic safety. *Transportation Planning and Technology*, 28(3):167–187, 2005.

[66] T. Luettel, M. Himmelsbach, and H.-J. Wuensche. Autonomous ground vehiclesconcepts and a path to the future. *Proceedings of the IEEE*, 100(Special Centennial Issue):1831–1839, 2012.

[67] F. Mannering, W. Kilareski, and S. Washburn. *Principles of highway engineering and traffic analysis*. John Wiley & Sons, 2009.

[68] M. Manz, M. Himmelsbach, T. Luettel, and H.-J. Wuensche. Detection and tracking of road networks in rural terrain by fusing vision and lidar. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 4562–4568, 2011.

[69] M. Manz, T. Luettel, F. Von Hundelshausen, and H.-J. Wuensche. Monocular model-based 3d vehicle tracking for autonomous vehicles in unstructured environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2465–2471, 2011.

[70] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[71] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4889–4895, 2011.

[72] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[73] I. Miller, M. Campbell, and D. Huttenlocher. Map-aided localization in sparse global positioning system environments using vision and particle filtering. *Journal of Field Robotics*, 28(5):619–643, 2011.

[74] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, et al. Junior: the stanford entry in the urban challenge. In *The DARPA Urban Challenge*, pages 91–123. Springer, 2009.

[75] S. Müller, M. Uchanski, and J. Hedrick. Cooperative estimation using vehicle communications. *Vehicle System Dynamics*, 39(2):121–133, 2003.

[76] J. Nilsson, M. Ali, P. Falcone, and J. Sjöberg. Predictive manoeuvre generation for automated driving. In *Proceedings of the IEEE Intelligent Transportation Systems*, pages 418–423, 2013.

[77] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *IEEE Transactions on Automatic Control*, 95:215–233, 2007.

[78] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49:1520–1533, 2004.

[79] R. Olney, R. Wragg, R. Schumacher, and F. Landau. Collision warning system technology. In *Steps Forward. Intelligent Transport Systems World Congress*, number Volume 3, 1995.

[80] H. B. Pacejka and R. S. Sharp. Shear force development by pneumatic tyres in steady state conditions: a review of modelling aspects. *Vehicle system dynamics*, 20(3-4):121–175, 1991.

[81] S. Pancanti, L. Pallottino, D. Salvadorini, and A. Bicchi. Motion planning through symbols and lattices. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3914–3919, 2004.

[82] R. Parker and S. Valaee. Cooperative vehicle position estimation. In *Proceedings of IEEE International Conference on Communications*, pages 5837–5842, 2007.

[83] E. Pérez, C. Ariño, F. X. Blasco, and M. A. Martínez. Explicit predictive control with non-convex polyhedral constraints. *Automatica*, 48(2):419–424, 2012.

[84] A. Petrovskaya and S. Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3):123–139, 2009.

[85] O. Pink, F. Moosmann, and A. Bachmann. Visual features for vehicle localization and ego-motion estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 254–260, 2009.

[86] M. Pivtoraiko and A. Kelly. Efficient constrained path planning via search in state lattices. In *Proceedings of International Symposium on Artificial Intelligence, Robotics, and Automation*, pages 1–7, 2005.

[87] M. Pivtoraiko, R. A. Knepper, and A. Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.

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

[89] M. Rizzi, A. Kullgren, and C. Tingvall. The injury crash reduction of low-speed autonomous emergency braking (aeb) on passenger cars. In *Proceedings of the IEEE International Conference on IRCOBI*, 2014.

[90] M. Rufli and R. Siegwart. On the design of deformable input-/state-lattice graphs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3071–3077, 2010.

[91] F. E. Sancar, B. Fidan, J. P. Huissoon, and S. L. Waslander. MPC based collaborative adaptive cruise control with rear end collision avoidance. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 516–521, 2014.

[92] C. Schmid and L. T. Biegler. Quadratic programming methods for reduced hessian sqp. *Computers & chemical engineering*, 18(9):817–832, 1994.

[93] R. Schöneburg, K.-H. Baumann, and R. Justen. Pre-safe-the next step in the enhancement of vehicle safety. In *Proceedings of 18th ESV International Technical Conference*, number 410, pages 19–22, 2003.

[94] T. Schouwenaars, B. De Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. In *Proceedings of European Control Conference (ECC)*, pages 2603–2608. IEEE, 2001.

[95] A. Shaout, D. Colella, and S. Awad. Advanced driver assistance systems-past, present and future. In *Proceedings of Seventh International Computer Engineering Conference (ICENCO)*, pages 72–82, 2011.

[96] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[97] S. Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report DOT HS 812 115, National Highway Traffic Safety Administration, Washington, DC, 2015.

[98] D. Soudbakhsh and A. Eskandarian. Steering control collision avoidance system and verification through subject study. *Intelligent Transport Systems*, 9(10):907–915, 2015.

[99] C. Thorpe, M. Herbert, T. Kanade, and S. Shafer. Toward autonomous driving: the cmu navlab. i. perception. *IEEE expert*, 6(4):31–42, 1991.

[100] C. Thorpe, M. Herbert, T. Kanade, and S. Shafter. Toward autonomous driving: the cmu navlab. ii. architecture and systems. *IEEE expert*, 6(4):44–52, 1991.

[101] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[102] S. Tokoro, K. Kuroda, A. Kawakubo, K. Fujita, and H. Fujinami. Electronically scanned millimeter-wave radar for pre-crash safety and adaptive cruise control system. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 304–309, 2003.

[103] R. Toledo-Moreo, M. Pinzolas-Prado, and J. M. Cano-Izquierdo. Maneuver prediction for road vehicles based on a neuro-fuzzy architecture with a low-cost navigation unit. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):498–504, 2010.

[104] R. S. Tomar and S. Verma. Neural network based lane change trajectory prediction in autonomous vehicles. In *Transactions on computational science XIII*, pages 125–146. Springer, 2011.

[105] C. Urmson. Progress in self-driving vehicles. In *Proceedings of Symposium on Frontiers of Engineering*, pages 5–9, 2015.

[106] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[107] F. Wang, M. Yang, and R. Yang. Conflict-probability-estimation-based overtaking for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):366–370, 2009.

[108] J. Wei, J. M. Dolan, and B. Litkouhi. A prediction-and cost function-based algorithm for robust autonomous freeway driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 512–517, 2010.

[109] M. Werling, J. Ziegler, S. Kammel, and S. Thrun. Optimal trajectory generation for dynamic street scenarios in a freńet frame. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 987–993. IEEE, 2010.

[110] C. C. Wit, H. Olsson, K. J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40:419–425, 1995.

[111] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of Fourth International Symposium on information Processing in Sensor Networks*, pages 63–70, 2005.

[112] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-vehicle safety messaging in dsrc. In *Proceedings of 1st ACM international workshop on Vehicular ad hoc networks*, pages 19–28. ACM, 2004.

[113] M. Zhu and S. Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.

[114] J. Ziegler, P. Bender, T. Dang, and C. Stiller. Trajectory planning for berthaa local, continuous method. In *Proc IEEE Intelligent Vehicles Symposium*, pages 450–457, 2014.

[115] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, et al. Making bertha drivean autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):8–20, 2014.

[116] J. Ziegler and C. Stiller. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems, IROS*, pages 1879–1884, 2009.

[117] J. Ziegler, M. Werling, and J. Schroder. Navigating car-like robots in unstructured environments using an obstacle sensitive cost function. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 787–791, 2008.