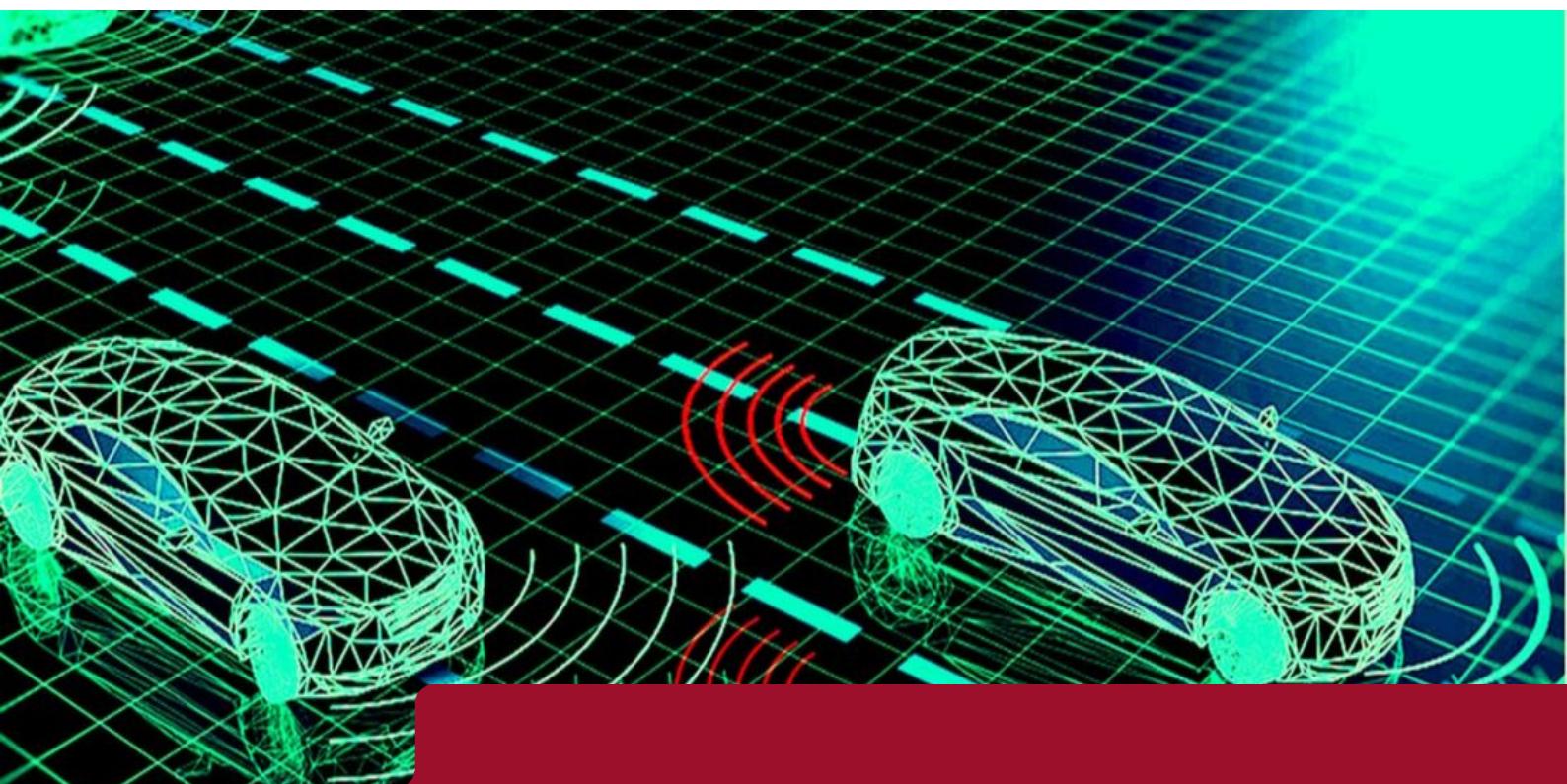




DEGREE PROJECT IN MECHANICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Stochastic Model Predictive Control for Autonomous Emergency Integrated Braking and Steering System

DEKUN WANG



KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Stokastisk modell
förlorutseende
kontroll för
Autonoma
integrerade

Sammanfattning

 KTH VETENSKAP OCH KONST KTH Industriell teknik och management	Examensarbete TRITA-ITM-EX 2021:535 Stokastisk modell förutseende kontroll för Autonoma integrerade bromssystem och styrsystem för nødsituationer Dekun Wang	
Godkänt 2021/09/14	Examinator Ulf Olofsson	Handledare Kaveh Nazem Tahmasebi
	Uppdragsgivare Kaveh Nazem Tahmasebi, Mechatronics, Machine Design Department, KTH	Kontaktperson Peng Su

För att kontrollera både den longitudinella och den laterala dynamiken hos ett självständigt fordon som står inför en eventuell kollision krävs en effektiv uppfattning av omgivningen och en optimal användning av dess manövrerbarhet.I nødfall kan separat styrning eller bromsing inte kunna undvika kollision eller säkerställa fordonets stabilitet.Detta papper utgör en integrerad styrenhet som kombinerar klassisk autonom nødbromsning (AEB), främre styrning och differentialbromsing, som kan besluta antingen att styra, bromsa eller båda för att undvika kollision med den stabilitet som garanteras.med beaktande av bristfällig modellering av det kontrollerade fordonet och osäkerhet i rörelse hos omgivande fordon,Metod med Stochastic Model Predictive Control (SMPC) används för att formulera den föreslagna styrenheten med ett Interactive Multiple Model Kalman Filter (IMM KF) som uppskattar manövrer av andra fordon och ett par smält kamera och radar som mäter deras aktuella tillstånd.Den föreslagna styrkrets effektivitet utvärderas genom simulering av Matlab/Simulink och hårdvara i slingan (HiL).HiL-plattformen är uppbyggd med en verlig styrenhet som körs på Nvidia Jetson Nano och ett virtuellt kontrollerat objekt (en modell för fordon med hög tillförlitlighet i CarSim).Kommunikationen mellan den registeransvarige och det kontrollerade objektet styrks av Robots operativsystem (ROS) och Matlab/Simulink.

Abstract

 KTH VETENSKAP OCH KONST KTH Industriell teknik och management	Examensarbete TRITA-ITM-EX 2021:535 Stochastic Model Predictive Control for Autonomous Emergency Integrated Braking and Steering System Dekun Wang	
Approve 2021/09/14	Examiner Ulf Olofsson	Supervisor Kaveh Nazem Tahmasebi
	Commissioner Kaveh Nazem Tahmasebi, Mechatronics, Machine Design Department, KTH	Contact Person Peng Su

Controlling both longitudinal and lateral dynamics of an autonomous vehicle confronting possible collision requires efficient perception of surrounding environment and optimal use of its maneuverability. In emergency, separate steering or braking may not be able to avoid collision or ensure stability of the vehicle. This paper presents an integrated controller that combines classical Autonomous Emergency Braking(AEB), front steering and differential braking, is capable of deciding either to steer, brake or both to avoid collision with stability ensured. Considering imperfect modelling of the controlled vehicle and uncertainty of motion of surrounding vehicles, Stochastic Model Predictive Control(SMPC) method is used to formulate the proposed controller with an Interacting Multiple Model Kalman Filter(IMM KF) that estimates maneuver of other vehicles and a pair of fused camera and radar that measures current states of them. The effectiveness of the proposed controller is evaluated via simulation on Matlab/Simulink and hardware in the loop(HiL). The HiL platform is constructed with a real controller running on Nvidia Jetson Nano and a virtual controlled object (a high-fidelity vehicle model in CarSim). The communication between controller and controlled object is empowered by Robot Operating System(ROS) and Matlab/Simulink.

Foreword

This thesis was written at the Mechatronics Division at Machine Design Department at KTH in Sweden over a period of eight months. I am very thankful for the opportunity to finish my master's thesis at this department. Here I would like to express my gratitude to the people that made it possible for me to reach this goal.

I am especially grateful for the support of my supervisor Kaveh Nazem Tahmasebi as well as Peng Su and Professor Dejiu Chen. Their extensive knowledge and guidance helped me throughout the whole project. During the early stages of the thesis, Kaveh was incredibly patient and inspired me when I was struggling with some unsolved issues. In our frequent meetings, Kaveh always pointed me in the right direction when I tried to realize some new ideas and advance to our final goals. He was extremely helpful by giving me honest and direct feedback when I was getting off track. This helped me to keep focused on the main goals of the project.

I would like to thank all the professors at the Machine Design Department, where I took most of my courses in the master's program. I would like to thank other professors and staff at KTH as well. You offered great knowledge, skills and services. The program lasted for more than two years, during which I experienced both rich academic atmosphere at KTH and open, free, equal, diverse atmosphere in Sweden that will be the most precious memories of my youth.

Finally, I would like to thank my family and friends who have been encouraging and supporting me through all the years of studying. Especially I would like to thank my parents who have always been my most important backups no matter in normal times or in the time of struggles such as the Covid-19 crisis.

Finishing master thesis is only a milestone for my life but there are more waiting for me to accomplish. We are living in a problematic world where there are wars, diseases, climate changes, injustice, inequality everywhere. The key to those problems is not the strong offering the weak out of mercy from the position of strength but to build a community of common destiny for all mankind. And I will continue to strive for this goal.

Nomenclature

Abbreviations

ADAS Automated Driving Assistance Systems. 8, 33

AEB Autonomous Emergency Braking. 8, 11, 27, 28, 43, 56

BTN Braking threat Number. 15

ESC Electronics Stability Control. 17

FOH First Order Hold. 8, 16, 23, 24, 42, 43

HiL Hardware in the Loop. 12, 51, 57

IMMKF Interacting Multiple Model Kalman Filter. 12, 33, 44, 56

MIO the most important object. 12, 27

MPC Model Predictive Control. 12, 24, 26, 56

ROS Robot Operating System. 12, 51, 57

SMPC Stochastic Model Predictive Control. 12, 56

STN Steering threat Number. 15

TTC Time to Collision. 15

ZOH Zero Order Hold. 8, 16, 22, 23, 42

Contents

	Page
List of Tables	7
List of Figures	8
1 Introduction	11
1.1 Background	11
1.2 Purpose	12
1.3 Delimitations	13
1.4 Method	13
1.4.1 The Engineering Design Process	13
1.4.2 Hardware in the Loop	14
2 Frame of Reference	15
2.1 Collision-free path planning	15
2.2 Tracking controller	16
2.3 Yaw stabilization	17
2.4 Tire model	17
2.5 Stochastic MPC	18
3 Implementation	19
3.1 Vehicle dynamic model	19
3.1.1 Continuous-time model	19
3.1.2 Discrete-time model	22
3.2 Classic MPC controller design	24
3.2.1 Constraints	25
3.2.2 Optimization Problem Formulation	29
3.3 Stochastic MPC controller design	32
3.3.1 Target Vehicle Models and IMMKF	33
3.3.2 SMPC Formulation	36
4 Simulation Results	40
4.1 Classical MPC Controller Results	40

4.2 Stochastic MPC controller Results	44
5 Hardware in the Loop Results	51
5.1 HiL platform setup	51
5.2 HiL Results	51
6 Discussion and Conclusion	55
6.1 Discussion	55
6.2 Conclusion	56
7 Future Work	58
8 Appendix	59
A Gantt Chart	59
B Risk Analysis	59
C Matrix Formulation for QP	60
D Kalman Filter Algorithm [1]	61
E Tire Model Specifications	61
F Code for SMPC algorithm	61
Bibliography	71

List of Tables

Table	Page
4.1 Specifications of the test vehicle model	40
4.2 Specifications of the controller parameters	40
4.3 Specifications of the fixed time step controller parameters	43
5.1 The comparison of communication qualities between two connection methods	52
8.1 Risk Analysis	70
8.2 Specifications of the controller parameters	70

List of Figures

Figure	Page
1.1 Necessary distance to avoid collision by braking or steering [3]	11
1.2 The Engineering Design Process	13
2.1 Potential field at a moment [14]	15
2.2 Environment envelops [15]	15
3.1 Bicyle model diagram [1]	19
3.2 Friction force circle. Applying a limited longitudinal force has little effect on the lateral force capacity	20
3.3 First Order Hold is a better approximation of typical closed loop inputs than Zero Order Hold [15]	23
3.4 Overall control structure. FWD: front wheel drive	25
3.5 The logic decides which mode should the controller be on	26
3.6 The environment envelop. The blue blocks are inaccessible regions	27
3.7 Figure 3.6 is in AES mode while this in AEB mode. By extending or compressing the prediction length, Adaptive Cruise Control and AEB functions can be achieved.	28
3.8 Block diagram of overal ADAS	33
3.9 Nominal position/real position/IMMKF trakcing results of the target vehicle in 100s	36
3.10 Model probabilities variation in 100s	36
4.1 The figure shows the planned trajectory at 5.1s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment (Section 4.1. A)	41
4.2 The figure shows the planned trajectory at 7s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment(Section 4.1. A)	41
4.3 The figure shows the planned trajectory at 10s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment(Section 4.1. A)	41
4.4 The vehicle trajectory in the phase plane of lateral velocity and yaw rate (Section 4.1. A)	42
4.5 The comparison of longitudinal velocity between two controller (Section 4.1. A)	43
4.6 The comparison of lateral velocity and yaw rate between two controller (Section 4.1. A)	44

4.7	The comparison of yaw angle and lateral deviation between two controller (Section 4.1. A)	45
4.8	The comparison of control inputs between two controller (Section 4.1. A) . .	46
4.9	The comparison of solution time between two controller (Section 4.1. A) . .	46
4.10	The figure shows the planned trajectory at 5.1s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment of the fixed time step controller (Section 4.1. A)	47
4.11	The figure shows the planned trajectory at 7s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment of the fixed time step controller (Section 4.1. A)	47
4.12	The figure shows the planned trajectory at 10s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment of the fixed time step controller (Section 4.1. A)	47
4.13	The vehicle trajectory in the phase plane of lateral velocity and yaw rate of the fixed time step controller (Section 4.1. A)	48
4.14	Trajectories of vehicles from the moment 5.1s to 7.02s (Section 4.1. B) . . .	48
4.15	Braking status from 0s to 7.02s (Section 4.1. B)	49
4.16	Longitudinal velocity from 0s to 7.02s (Section 4.1. B)	49
4.17	The figure shows the comparison of trajectories between MPC and SMPC controller when the risk parameter ε is 0.05 (Section 4.2. A)	49
4.18	The figure shows the comparison of trajectories between MPC and SMPC controller when the risk parameter ε is 0.3 (Section 4.2. A)	49
4.19	The model probabilities of IMMKF in 22s (Section 4.2. B)	50
4.20	The planned trajectory and trajectory of the closed-loop at time 5.1s (Section 4.2. B)	50
4.21	The planned trajectory and trajectory of the closed-loop at time 6.6s (Section 4.2. B)	50
5.1	The structure of the designed HiL platform and the real setups	52
5.2	The controlled vehicle is following the lead vehicle in the begining	53
5.3	The lead vehicle brakes and the controlled vehicle starts steering	53
5.4	The controlled vehicle overtakes the lead vehicle	53
5.5	The drive or brake force in front left wheel	53
5.6	The drive or brake force in front right wheel	53
5.7	The drive or brake force in front right wheel	53
5.8	The drive or brake force in rear left wheel	53
5.9	The steering angle in degrees	54
8.1	GANTT chart for time plan	59
8.2	Longitudinal force curves	61

8.3 Lateral force curves	62
------------------------------------	----

1 Introduction

In this chapter, the background for this thesis along with project purpose, delimitation and methodology used for this project are presented.

1.1 Background

The introduction of new sensing technologies like camera and radar along with actuation technologies such as active steering and differential braking are making autonomous driving possible. The reliability and safety of autonomous vehicles are one of the most interesting and important topics for both researchers and automotive industry to ensure their acceptance by customers and regulatory agencies [1]. In fact, expectations on this promising field are more than ensuring reliability and safety in normal situations but go beyond to emergency situations like to avoid or mitigate potential collision. A variety of Automated Driving Assistance Systems (ADAS) have been studied and developed to meet those expectations, which considerably improves vehicle active safety by facilitating better Controllability and stability of vehicles.

The prestigious Anti-lock Braking System(ABS) and Traction Control (TC) could be the earliest driving assistance systems, which could increase the braking efficiency and prevent the wheel from slipping respectively so as to improve stability and steerability. This was followed by Electronic Stability Control (ESC), Active Front Steering (AFS) and Four-wheel steering (4WS). ESC employs differential braking and actively distribute engine torque to control the yaw motion of vehicle at handling limit to stabilize the vehicle. AFS and 4WS use steering (front wheels or full four wheels) to control lateral dynamics of vehicles. [2] Because of improvements of perception technologies, automakers have developed the Autonomous Emergency Braking (AEB) system, which can actively apply longitudinal braking when a possible collision is imminent [3].

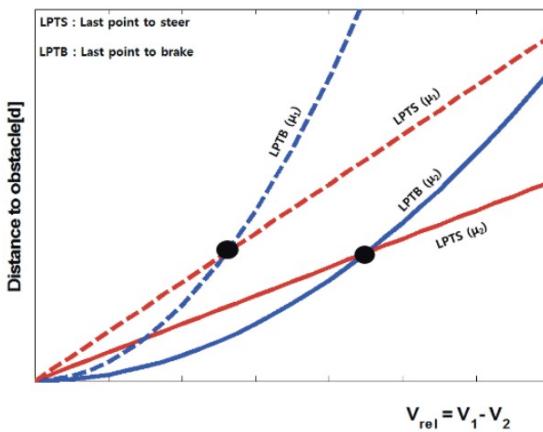


Figure 1.1: Necessary distance to avoid collision by braking or steering [3]

Although rear-end collision avoidance can be achieved by braking, its application is limited to scenarios where the relative speed and position between the vehicle and the obstacle is

fairly low as figure [1.1]. In other words, conventional AEB system alone cannot provide good performance at high relative speeds [3]. In highway scenarios, obstacle avoidance may need more aggressive maneuver in order to make the best of multiple actuators since the vehicle is travelling much faster. Steering and braking should be combined to avoid collision without running into instability.

Model Predictive Control (MPC), a powerful optimal control algorithm, has been widely used in developing autonomous driving systems. It uses a model to predict the future evolution of states with which it is able to find the optimal solution sequence that minimizes the cost function. Because of the 'predictive' nature, it is so suitable for designing collision avoidance system. [4] [5] [6]. In the past decade many Stochastic Model Predictive Control (SMPC) algorithms have been developed for systems subject to stochastic disturbances and model uncertainties [7].

Moreover, autonomous vehicles operating in dynamic urban environments must account for the uncertainty arising from the behavior of other objects in the environment. The environment model is required to be able to estimate and predict motion and maneuver of surrounding vehicles. The uncertain predictions are used to formulate a chance-constrained MPC problem [8].

1.2 Purpose

This thesis aims to formulate a control method using SMPC algorithms for a driving safety assistant system, which is able to efficiently actuate wheel brakes and front steering to avoid or mitigate possible obstacle collision without endangering the safety of passengers. In addition, an Interacting Multiple Model Kalman Filter (IMMKF) needs to be employed to estimate and predict behavior of surrounding vehicles, especially the vehicle ahead of the controlled vehicle (MIO, the most important object). IMMKF is supposed to be able to estimate which kind of maneuver the MIO is in so as to predict the future trajectory based on the information obtained from fused sensors (camera and radar). The SMPC's constraints should be adjusted according to the reliability of perception module.

The following objectives are stated to achieve the thesis aims:

- 1). To formulate a logic that decides either steer the vehicle to adjacent lanes or stay in current lane and brake once a possible collision is detected.
- 2). To design an SMPC controller that guides the vehicle to follow a collision-free trajectory as much as possible without endangering its stability in emergency scenarios by using combined braking and steering.
- 3). To model possible behaviors of surrounding vehicles (braking in current lane, moving in current lane at constant velocity, changing lane to left). To formulate an IMMKF that is able to tell which behavior model fits best to the MIO, which latter helps build up MPC environment-related constraints.
- 4). To test the designed controller performance via simulations in Matlab/Simulink.
- 5). To set up an Hardware in the Loop (HiL) platform. The platform consists of three parts, the virtual controlled object(a high-fidelity vehicle dynamics model from CarSim), the real controller (controller in c++ on Nvidia Jetson Nano), the communication links (Robot Operating System (ROS) and Matlab/Simulink). The effectiveness of the proposed controller is evaluated on the HiL platform.

1.3 Delimitations

The content of work not included in this thesis project is listed below:

- 1). The MPC controller is designed based on a simplified single-track vehicle model, known as 'bicycle model', along with Brush tire model [1], which is used for predicting the evolution of vehicle states. The dynamic vehicle model is nonlinear and linearized successively. It is subject to process and measurement noise that obeys Gaussian distribution. Gaussian distribution is not obtained from experiments. Therefore, simplification of model, linearization and unknown Gaussian noise may make the model deviate from the reality which isn't discussed in this thesis.
- 2). For a vehicle in reality, rollover might happen when the car takes an aggressive maneuver in emergency (i.e. suddenly steer to left at a high longitudinal velocity to avoid an obstacle in current lane). Since MPC controller is designed based on 'bicycle model' and it does not include suspensions and it is single track, the effect of possible rollover on stability is not analyzed.
- 3). The models in IMMKF consist of only three models, 'braking in current lane', 'moving in current lane at constant velocity' and 'changing lane'. In reality, other vehicles may take way more kinds of maneuvers that are not considered in this thesis.
- 4). Controller performance is not tested on a real car in an open dynamic road.

1.4 Method

The academic methods used in this thesis project are presented below:

1.4.1 The Engineering Design Process

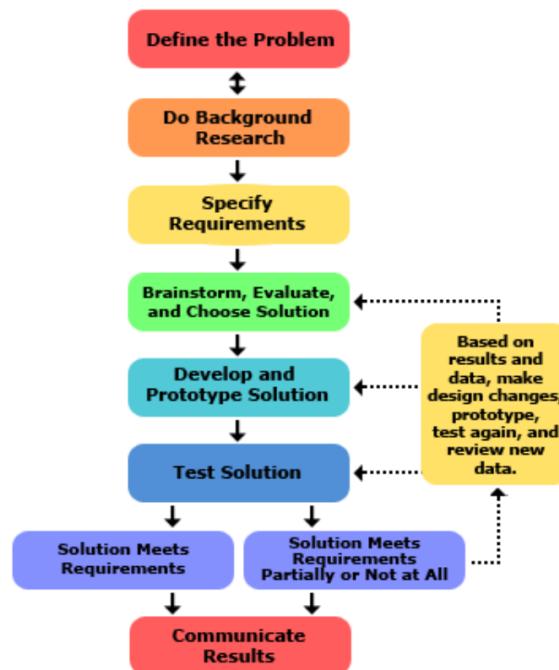


Figure 1.2: The Engineering Design Process

The engineering design process is a series of steps that engineers follow to come up with a solution to a problem, which will be followed to conduct designing, simulations, physical tests during the whole project. Figure 1.2 shows the architecture of this method.

Define the problem and specify requirements have been discussed in details in previous chapters. Background research is mentioned in section 1.1 and will be elaborated in chapter 2. Based on literature research, a brainstorm of possible solutions to the research question is carried out. Then prototype controller is modeled and tested later in simulation. Loops of brainstorm possible solutions, prototype tests are continuously done until solution meets requirements (The designed controller works well in simulation results).

1.4.2 Hardware in the Loop

Hardware in the loop (HIL) is a technique that is used in the development and test of complex real-time embedded systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform. The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems. These mathematical representations are referred to as the “plant simulation”. The embedded system to be tested interacts with this plant simulation.

Finally, an experiment in the HIL platform was conducted to verify the real-time performance and effectiveness of the proposed emergency collision avoidance approach.

2 Frame of Reference

Recent advances in sensor performance and processing technology are empowering vehicles with better perception and tracking of approaching obstacles from any direction around. Sensing technologies like radar, lidar and cameras are integrated which enables vehicle identification of risks more quickly and accurately than human driver could do in emergency situation. Thanks to those breakthroughs, various driving safety assistant systems have been developed for dealing with emergency situations. In general, there are several indispensable pillars required to build up driving assistant systems, which will be introduced one by one in the following.

2.1 Collision-free path planning

Many researchers divided a collision avoidance tasks into collision-free path planning task and tracking control task. Path planner generates a collision-free trajectory and a tracking controller to guide the vehicle along the planned path. Many approaches have been proposed for path planning, such as smooth curve planner [3] [9], arc path planner [10], neural-network-based [11], fuzzy-control-based [12], artificial potential field [13] [14], experiment-based [2] and optimal control method [15] [1].

Smooth curve planning methods implement different techniques for path smoothing and curve generation, such as spline curves and Bezier curves [15]. Cui et al. [10] proposed a method which guides the vehicle for a steering maneuver with a constant centripetal acceleration. Thus, the desired path is formulated as a circular path (an arc). They also proposed a risk monitor that calculates Time to Collision(TTC), Braking threat Number (BTN) and Steering threat Number (STN). With risk monitor, either brake or steer is decided. Fuzzy and neural network-based methods are very similar to the way how human drivers plan path with brains.

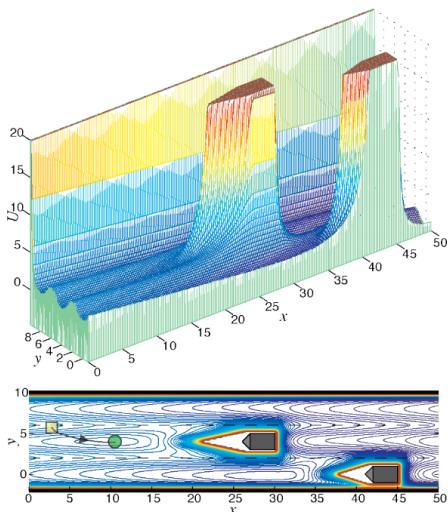


Figure 2.1: Potential field at a moment [14]

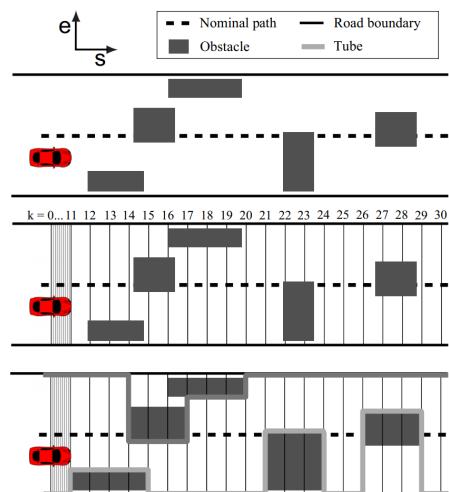


Figure 2.2: Environment envelops [15]

The artificial potential field method [13] [14] models the ego vehicle as a positively-charged body moving in an electric field. Typically, the potential function $U(q)$ includes a negative (attractive) component driving the vehicle towards its reference path and positive (attractive) components around all obstacles and lane boundaries. The resulting potential field (i.e. figure 2.1) is constructed as a superposition of disparate functions for lane-keeping, road-staying, speed preference and vehicle avoidance and passing.

Falcone et al. [2] proposed an experiment-based method. The trajectory is established by simply driving a test vehicle by a human driver slowly along the desired path. The actual path is recorded by differential GPS and then can be used as desired path for other cases at higher speed.

Hajiloo et al. [1] and Brown et al. [15] uses optimal control methods by which obstacles inside prediction horizon are counted into MPC constraints, known as environmental envelopes (figure 2.2). Since the longitudinal velocity is known in prediction horizon, the vehicle distance along the path is known over the horizon. The lateral deviation at each time step in the prediction horizon can be constrained between a max and min value, which are set to be within road boundaries and obstacle-free. With regard to that, this method actually integrates path planning with tracking control in one algorithm.

2.2 Tracking controller

Falcone et al. [2] proposed a control approach based on MPC for actively controlling the front steering angle of vehicles to track a obstacle-free double-lane change path on snow or ice with a given initial forward speed. Two formulations (nonlinear and linear) are mentioned. Nonlinear MPC is used based on nonlinear vehicle model which predicts future evolution of states better. Therefore, it is able to satisfy control objectives at higher entry speed but it is too computationally heavy. Another formulation is based on successive linearization of vehicle model. Since it is linearized around current operating point instead of fixed point, the model mismatch increases as the states and input trajectories deviate from the current operating point, leading to large prediction errors with a consequent instability.

Later he et al. [5] incorporates brake torques at four wheels as actuators into front steering only maneuver. Instead of a single-track vehicle model “bicycle model” used in [2], a four-wheel vehicle model is used. In [4], Falcone et al. takes slip ratios of four wheels instead of brake torques as control inputs along with steering angle. In this way, tire forces are either in linear or nonlinear region and if it is saturated become explicit.

Choi et al. [3] uses a kinematic vehicle model to design its MPC controller for integrated steering and braking. An approach to constrain front steering is proposed.

Brown et al. [15] formulated an alternative MPC control framework which integrates path planning and tracking. Safety envelopes are imposed to enforce stability and tracking which will be discussed in details in section 2.3. The control framework uses the lateral force in front wheel as control input which allows the controller explicitly consider saturation of front tire compared to using steering angle. Hajiloo et al. [1] developed Brown’s framework even further. Yaw moment generated by longitudinal forces (braking) is controlled as well. In order to have longer prediction horizon without computing too long, varying time steps are used. Shorter time steps in the beginning and longer in the end. As a result, Zero Order Hold (ZOH) and First Order Hold (First Order Hold) are together used to discretize the model. In control algorithm, the magnitude of the front lateral force is not penalized, but the magnitude of the yaw moment from differential braking is penalized in cost function.

This measure enables the controller to rely on front steering and activate differential braking only when an additional yaw moment is needed.

Cui et al. [10] proposed two-stage collision avoidance maneuver strategy. Because of the arc reference path, considerable overshoots are always observed when it steers to the new lane.

2.3 Yaw stabilization

In emergency cases, aggressive maneuver is required in order to make all effort to avoid collision, however, which sometimes results in unexpected instability. Falcone et al. [6] proposed two control approaches for yaw and lateral stabilization. The braking yaw moment is generated for yaw stabilization while the front steering is for controlling the lateral dynamics. Two approaches are respectively based on bicycle model and four-wheel vehicle model. The interaction between slip ratio and slip angle results in difficulties in tuning for the four-wheel model but this model does perform better compared to another.

In [2], [4], [5], the designed controllers were used for tracking a reference path generated by a high-level controller. A stability constraint which constrains the slip angles of front and rear wheels, called ad hoc constraint, was considered. But the effect of deviations from the reference path forced by stability constraint is not considered which may result in collision.

In [15] and [16], two safety envelops are proposed. One is for ensuring stability, called stable handling limit which is a control invariant set [16]. Another is the environmental envelop which is imposed to enforce the vehicle to track a collision-free path. Along with safety envelops, slack variables and a hierarchical structure enables the controller prioritize 1) collision avoidance 2) vehicle stability 3) path tracking.

Cui et al. [10] divided collision avoidance maneuver into two stages. In stage 1, a constant lateral acceleration is planned so that the vehicle is guided along a arc path using a feedforward controller. Differential braking (DB) is used to enforce stability, which contains three modes according to the level of lateral acceleration. In the first mode DB is used as a compensator to eliminate the tracking error regardless of the stabilization. The second mode could trade off between the objectives of collision avoidance and stabilization. The third mode, meaning the vehicle has lost stability, goes all out to stabilize the vehicle. In stage 2, the vehicle drives back to the centerline of the new lane. DB now only functions as a traditional Electronics Stability Control (ESC) system.

2.4 Tire model

To be able to control vehicle at its handling limits in emergency situations, appropriate tire model is needed to fully understand the nonlinearity of vehicles, which means to model a vehicle into tire model level, not only chassis level. Therefore, it is important to use a realistic nonlinear tire model.

Falcone et al. [2] used a Pacejka tire model, also known as Magic Formula model. This is a complex, semi-empirical nonlinear model developed by analytical considerations and key parameters depend on experiments data. In most reference papers, the model is used as well.

In [17], The famous Pacejka model is simplified by piecewise linear approximations. A wheel torque controller was developed based on nonlinear MPC which used the simplified tire

model to control the vehicle yaw rate and side slip angle. Its performance was compared to the controller developed based on Pacejka model which turned out to be convincing.

Adireddy et al. [16] refined the simplified tire model proposed in [17]. The simulation results indicate that with the proposed modification on the simplified tire model, the performance of the controller that uses a simplified tire model is close to the controller based on Pacejka tire model and can achieve good control of vehicle behavior in extreme maneuvers but consumes less computational resources.

Hajiloo et al. [1] applied Brush Tyre model and linearize the nonlinear model successively which makes MPC problem still linear quadratic programmable, otherwise nonlinear optimizer is required.

2.5 Stochastic MPC

[7] reviews the development and different methods of SMPC. These methods are motivated by many application fields where a priori knowledge of the stochastic distribution of the uncertainties is available, some degree of constraint violation is allowed, the nominal operation should be defined as close as possible to the operational constraints for optimality reasons. Open loop control, disturbance feedback control and state feedback control strategies are proposed so as SMPC's chance constraints are reformulated to be in form of classical MPC.

[18] [19] present SMPC formulation for semi-autonomous vehicles with a stochastic driver model. Given a probability, the probabilistic safety constraints are converted into tightened constraints accordingly on the states of LTV model by computing the evolution of disturbance. Driver's wrong operation is corrected and minimized.

[20] considered the effect of uncertainties of surround vehicles on the controlled vehicles. Two types of uncertainties are considered, multiple future maneuvers are possible, and within these maneuvers the vehicle can vary from the predicted ideal maneuver path. Brudigam et al. combine Scenario Model Predictive Control (to cope with multiple predicted maneuvers) and Stochastic Model Predictive Control with chance-constraints (to take into account vehicle deviations from the predicted maneuver trajectories of the respective maneuver).

In [8] an Interacting Multiple Model Kalman Filter is used to predict future states of target vehicles. The predicted maneuver with the highest probability is then used to solve a chance-constrained MPC problem.

3 Implementation

In this chapter, the implementation of the methods that were introduced in [1].

3.1 Vehicle dynamic model

3.1.1 Continuous-time model

The control-oriented model is created using a planar bicycle model with five states, the longitudinal velocity u , lateral velocity v , yaw rate r , heading deviation θ and lateral deviation e from the desired path, as illustrated in figure [3.1]. The vehicle is front-wheel driving and

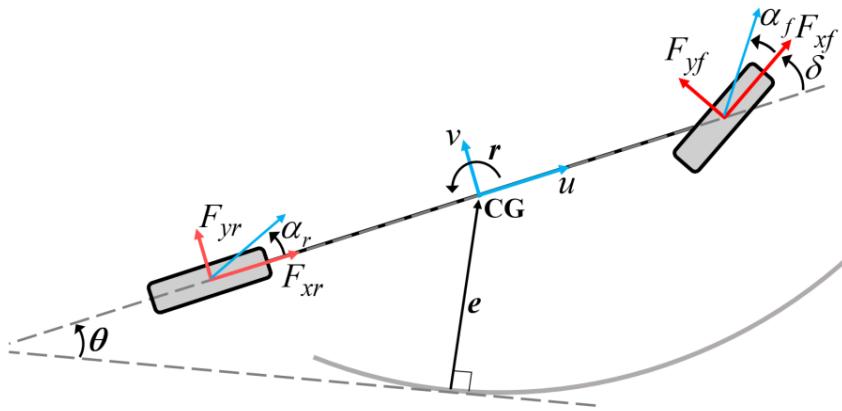


Figure 3.1: Bicyclic model diagram [1]

is equipped with active front steering, longitudinal braking and differential braking. The desired longitudinal velocity of the vehicle along the path is set by the driver, the integrated controller computes the required drive/brake forces in each wheel to follow the reference longitudinal velocity.

Assumption 1: The small angle assumption, which assumes the heading deviation θ small enough so that $\sin\theta$ and $\tan\theta$ can be approximated as θ , $\cos\theta$ as 0.

Under assumption 1, the longitudinal and lateral dynamics of the vehicle are as follows:

$$\dot{u} = vr + \frac{F_{xnet}}{m} \quad (3.1)$$

$$\dot{v} = \frac{F_{yf} + F_{yr}}{m} - ur \quad (3.2)$$

$$\dot{r} = \frac{M_{Fy} + M_{Fx}}{I_z} \quad (3.3)$$

where m and I_z are the vehicle's mass and moment of inertia respectively; F_{xnet} , F_{yf} and F_{yr} are the longitudinal net force, front lateral tire force and rear lateral tire force; M_{Fx} and M_{Fy} are the yaw moment produced by longitudinal forces and lateral forces respectively.

Increasing the lateral agility of a vehicle is important for the ability to change vehicle's heading angle and lateral position in a short time. In normal situations, using only front steering is acceptable since in this case required agility can be satisfied by using only the front steering. Front steering produces lateral forces on tires, producing the yaw moment generated by lateral forces $M_{Fy} = aF_{yf} - bF_{yr}$. However, in emergency situations, M_{Fy} cannot always provide enough maneuverability to avoid collision. It is when additional yaw moment is needed. The yaw moment generated by longitudinal forces M_{Fx} (differential braking in our case) can be added to improve the vehicle's lateral agility and response to emergency faster, which maximizes its agility.

Avoiding collision often requires the vehicle to maximize its lateral maneuverability within the handling limit. Under assumption 2, the maximum limit of lateral acceleration, $a_{y,max}$ can be written as:

$$a_{y,max} = \frac{F_{yf,max} + F_{yr,max}}{m} \quad (3.4)$$

where the maximum lateral forces in front and rear tyres $F_{yf,max} = \sqrt{(\mu F_{zf})^2 - F_{xf}^2}$ and $F_{yr,max} = \sqrt{(\mu F_{zr})^2 - F_{xr}^2}$.

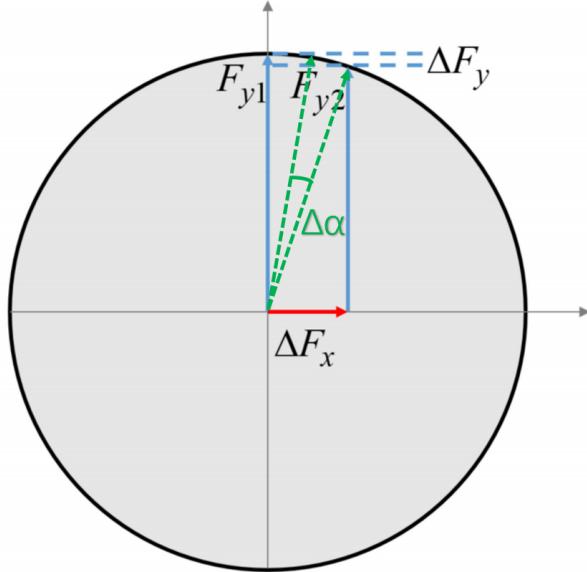


Figure 3.2: Friction force circle. Applying a limited longitudinal force has little effect on the lateral force capacity

Assumption 2: assumes the front and rear tires reach saturation region at the same time. As equation 3.4 shows, the allowed maximum lateral acceleration decreases as the longitudinal forces increase. In figure 3.2, $|\Delta F_x| = \mu F_{zf} \sin \Delta\alpha \approx \mu F_{zf} \Delta\alpha$, $|\Delta F_y| = \mu F_{zf} \cos \Delta\alpha \approx 0$, showing that sacrificing a negligible drop in the lateral forces can produce considerable longitudinal forces which can be used to generate the required additional yaw moment, improving lateral maneuverability. However, incorporating the yaw moment M_{Fx} is a trade-off between increasing the lateral agility and decreasing the allowed maximum lateral acceleration capacity due to tire force coupling effect. [1] Moreover, the longitudinal net force $F_{xnet} = F_{xr} + F_{xf}$ has strong impact on the other two control inputs. Therefore, the going-to-propose MPC controller will have three separated modes 1)steering to avoid collision and 2)braking before the obstacle 3)actuate both by adjusting value of desired control inputs and other conditions.

It should be noted that in reality the assumption 2 is not satisfied in most cases. The saturation in front and rear tires do not necessarily happen at the same time. Given a front drive vehicle as we are interested in this thesis, the front tires may saturate before the rear ones if the longitudinal controller applies drive torque to front axle. In this situation, understeering is about to take place and the front tires are saturated but the remaining capacity of the rear tires can still provide additional yaw moment by longitudinal forces which only sacrifices small share of allowed maximum lateral acceleration. This additional yaw moment can compensate for the lack of steerability when the front tires are saturated.

The additional yaw moment doesn't necessarily ensure vehicle's safety but may endanger its safety, causing instability. This problem can be tackled by constraining the limit of differential braking, especially when the lateral velocities violate a predefined stability constraint.[\[1\]](#) When the vehicle's trajectory exits the stability parallelogram [\[15\]](#), the tires leave the linear region and saturate and the vehicle is prone to instability. Therefore, the limit of differential braking needs to be constrained properly and the use of longitudinal forces should be restricted to avoid instability.

In this thesis, the tire model is a Brush tire model that captures the lateral force drop due to an applied longitudinal force [\[21\]](#):

$$F_y = \begin{cases} -C_\alpha z + \frac{C_\alpha^2}{3\zeta\mu F_z} |z|z - \frac{C_\alpha^3}{27\zeta^2\mu^2 F_z^2} z^3 & |\alpha| < \alpha_{sl} \\ -\zeta\mu F_z sgn\alpha & |\alpha| \geq \alpha_{sl} \end{cases} \quad (3.5)$$

where $z = \tan \alpha$, $\alpha_{sl} = \arctan \frac{3\zeta\mu F_z}{C_\alpha}$, C_α is the tire cornering stiffness; F_z is the normal load on the tire; α is slip angle.

The front and rear tires' slip angles are calculated as follows:

$$\alpha_f = \frac{v + l_f r}{u} - \delta, \quad \alpha_r = \frac{v - l_r r}{u} \quad (3.6)$$

The nonlinearity of the tire model makes the vehicle dynamic model also nonlinear. To reduce the computational complexity and run time of the optimization problem, the nonlinear tire force is linearized successively. At each time step the tire forces are linearized around the current rear slip angle $\bar{\alpha}_r$, which makes rear tire force as an affine function of the rear slip angle.

It should be noted that the front lateral force F_{yf} is considered as the second control input instead of the steering angle δ , which can be later mapped to δ . Finally, substituting linearized equation [3.5](#) and [3.6](#) into equation [3.2](#) and [3.3](#) we get:

$$\dot{v} = \frac{F_{yf} + \overline{F}_{yr} + \overline{C}_{\alpha r}(\alpha_r - \bar{\alpha}_r)}{m} - ur \quad (3.7)$$

$$\dot{r} = \frac{l_f F_{yf} - l_r (\overline{F}_{yr} + \overline{C}_{\alpha r}(\alpha_r - \bar{\alpha}_r))}{I_z} + M_{Fx} \quad (3.8)$$

where $\overline{C}_{\alpha r}$ is the current rear cornering stiffness.

The heading deviation θ and lateral deviation e are local to an estimated lane center with given curvature $\kappa(s)$. So the equations are as follows:

$$\dot{\theta} = r - u\kappa(s) \quad (3.9)$$

$$\dot{e} = u\theta + v \quad (3.10)$$

In general, the continuous-time vehicle dynamic model can be expressed as:

$$\dot{x} = A_c^k x + B_c^k u + d_c^k \quad (3.11)$$

where $x = [u, v, r, \theta, e]^T$, $u = [F_{xnet}, F_{yf}, M_{Fx}]^T$

$$A_c^k = \begin{bmatrix} 0 & r_0 & v_0 & 0 & 0 \\ \frac{\overline{C}_{\alpha r}(l_r r_0 - v_0)}{mu^2} - r_0 & \frac{\overline{C}_{\alpha r}}{mu_0} & -\frac{\overline{C}_{\alpha r} l_r}{mu} - u_0 & 0 & 0 \\ -\frac{l_r \overline{C}_{\alpha r}(l_r r_0 - v_0)}{I_z u_0^2} & -\frac{l_r \overline{C}_{\alpha r}}{I_z u_0} & \frac{\overline{C}_{\alpha r} l_r^2}{I_z u_0} & 0 & 0 \\ -\kappa(s) & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & u_0 & 0 \end{bmatrix}$$

$$B_c^k = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & \frac{l_f}{I_z} & \frac{1}{I_z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad d_c^k = \begin{bmatrix} 0 \\ \frac{\overline{F}_{yr} - \overline{C}_{\alpha r} \overline{\alpha}_r}{m} \\ -\frac{l_r(\overline{F}_{yr} - \overline{C}_{\alpha r} \overline{\alpha}_r)}{I_z} \\ -u_0 \kappa(s) \\ 0 \end{bmatrix}$$

3.1.2 Discrete-time model

The continuous-time vehicle dynamic model described in subsection 3.1.1 has to be discretized to formulate a discrete-time model that can be implemented in an MPC controller. Proper discretization approach enables the controller to effectively predict vehicle behavior along the prediction horizon.

The common method to discretize a continuous-time model is Zero Order Hold with a constant sample time. However, using ZOH to discretize vehicle model which will be implemented in MPC controller has some problems. Autonomous vehicles require the electronic control unit is able to predict the future states in a long enough prediction horizon to response to obstacles in advance and also has a high control frequency to accurately capture the propagation of the lateral velocity and yaw rate which means great computational burden

for CPU. For instance, vehicles need a prediction time of 4s to plan its motion in advance and a small time step of 0.01s to accurately control the vehicle, which makes prediction step to be 400. It becomes a trade-off between having a long prediction horizon and having faster responses. The most intuitive way to work this dilemma out is to use more powerful and faster CPUs on vehicles but it is not always feasible to do so since CPUs' computational capability growth has been developed into a bottle neck due to restriction of the Mole Rule. Therefore, other alternatives must be considered. One of alternatives is using varying time steps and First Order Hold to fit the discretized model to the continuous model better. In this implementation, this method will be implemented and compared with fixed time step and ZOH.

Varying time steps method uses short time steps t_{short} in the beginning to accurately model vehicle and longer time steps t_{long} in the end to efficiently extend the prediction length. In between short and long time steps, transition time steps, called as correction steps are used. The time step at the k th point into the prediction horizon is:

$$t_s^k = \begin{cases} t_{short} & 1 \geq k < N_1 \\ t_{corr} & N_1 \geq k < N_2 \\ t_{long} & N_2 \geq k < N \end{cases} \quad (3.12)$$

where N is the total number of steps in the prediction horizon and N_1 is the number of initial time steps, $N_2 - N_1$ the number of correction time steps, $N - N_2$ the number of terminal time steps. For this implementation, $t_{short} = 0.05s$, $t_{long} = 0.2s$, $N = 70$, $N_1 = 40$, $N_2 = 50$. The correction step t_{corr} is variable whose value ranges from t_{short} to t_{long} linearly.

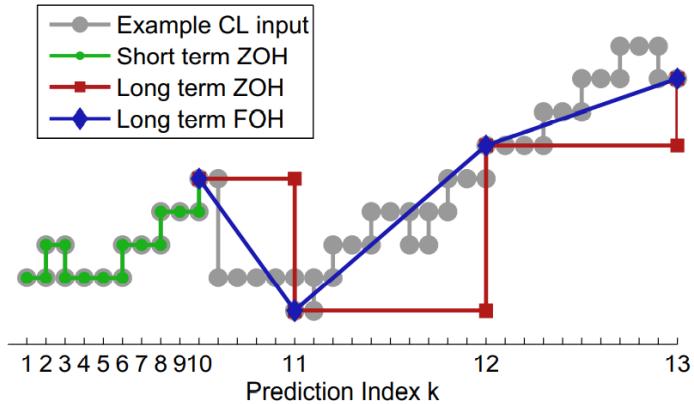


Figure 3.3: First Order Hold is a better approximation of typical closed loop inputs than Zero Order Hold [15]

Every discretization method uses an assumption about the behavior of the input between time steps. ZOH assumes the input is held constant over a time step. This matches typical commands to lower-level controllers to hold a constant value until the next control input is calculated. Figure 3.3 shows one such input sequence in gray. The input assumed by this discrete model is shown in green, matching a typical closed loop input from step $k=0$ to $k=9$. However, for $k=10$ to $k=29$, the discretization assumption does not match closed loop inputs any longer. A Zero Order Hold for these indices would assume the inputs are

constant over one t_{long} period, as shown in red. This is a poor assumption obviously since over one t_{long} period the inputs can vary much more than it do over one t_{short} period. A better approach to those long steps is to discretize with First Order Hold. This assumes that inputs to the system vary linearly between time steps, as illustrated in blue in Figure 3.3. This results in a better assumption that a linear variation is assumed between $u(k)$ and $u(k + 1)$ as $u(k + 1) = u(k) + t_s^k \dot{u}(k)$ [1]. To calculate discrete-time model, an augmented matrix Λ^k is constructed as follows:

$$\Lambda^k = \begin{bmatrix} A_c^k & B_c^k & d_c^k & \mathbf{0}_{5 \times 4} \\ \mathbf{0}_{4 \times 5} & \mathbf{0}_{4 \times 3} & \mathbf{0}_{4 \times 1} & (\frac{1}{t_s^k}) \mathbf{I}_{4 \times 4} \\ \mathbf{0}_{4 \times 5} & \mathbf{0}_{4 \times 3} & \mathbf{0}_{4 \times 1} & \mathbf{0}_{4 \times 4} \end{bmatrix}$$

Next, taking the matrix exponential of $\Lambda^k t_s^k$ gives:

$$e^{\Lambda^k t_s^k} = \begin{bmatrix} \mathbf{A}^k & \Phi_1^k & \Phi_2^k \\ \mathbf{0}_{4 \times 5} & \mathbf{I}_{4 \times 4} & \mathbf{I}_{4 \times 4} \\ \mathbf{0}_{4 \times 5} & \mathbf{0}_{4 \times 4} & \mathbf{I}_{4 \times 4} \end{bmatrix}$$

Defining $\mathbf{B}_1^k = (\Phi_1^k - \Phi_2^k) \times [\mathbf{I}_{3 \times 3}; \mathbf{0}_{1 \times 3}]$, $\mathbf{B}_2^k = \Phi_2^k \times [\mathbf{I}_{3 \times 3}; \mathbf{0}_{1 \times 3}]$, $\mathbf{d}^k = \Phi_1^k \times [\mathbf{0}_{3 \times 1}; 1]$, the discrete-time model can be written as:

$$\mathbf{x}(k + 1) = \mathbf{A}^k \mathbf{x}(k) + \mathbf{B}_1^k \mathbf{u}(k) + \mathbf{B}_2^k \mathbf{u}(k + 1) + \mathbf{d}^k \quad (3.13)$$

3.2 Classic MPC controller design

There are three objectives for the controller to achieve: 1) tracking the desired path 2) enforcing vehicle stability 3) avoiding collisions. To satisfy these objectives, a classical MPC controller has been developed to determine the optimal control inputs of the longitudinal net force F_{xnet} , the front lateral force F_{yf} and the yaw moment M_{Fx} . The controller has three modes: 1) AEB mode: mainly the longitudinal net force F_{xnet} is enabled, the other two are made close to zero. 2) AES mode: the longitudinal net force F_{xnet} is only enabled to keep the longitudinal velocity at the desired velocity. 3) mixed mode: actuate both AEB and AES. Mainly the front lateral force F_{yf} and the yaw moment M_{Fx} are enabled to steer while AEB to brake in order to avoid the collision. A logic is defined to decide which mode the controller should be on in given scenario. The overall control structure is depicted in Figure 3.4. The logic is shown in Figure 3.5. The logic makes decisions depending on the computations of three parameters, TTC(Time To Collision), BTN(Braking Threat Number) and STN(Steering Threat Number). For detailed computation for those parameters readers can refer to [10].

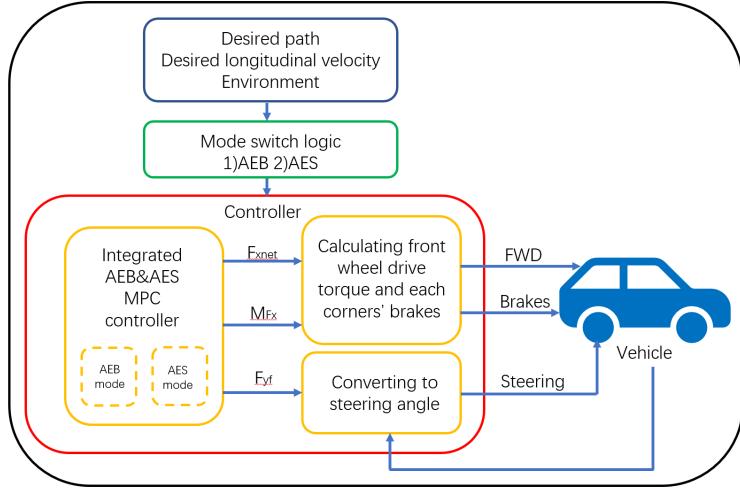


Figure 3.4: Overall control structure. FWD: front wheel drive

3.2.1 Constraints

A. States Constraints

State constraints include three constraints: 1)'Stable Handling limit' 2)'Environment Envelop' 3)'Elastic Length Envelop'.

Enforce lateral stability on the vehicle requires constraining the two dynamic states, yaw rate r and lateral velocity v properly. The steady state assumption can be used for defining the maximum safe yaw rate and lateral velocity. Under such assumption, the steady state limit of the tire's lateral forces determines the maximum safe yaw rate as:

$$r_{max} = \frac{\max(F_{yf} + F_{yr})}{mu} \quad (3.14)$$

If there is no longitudinal force on tires, it can be assumed that the front and rear tires are saturated at the same time, which gives $r_{max} = \frac{g\mu}{u}$. But in reality longitudinal forces are applied so the maximum allowed yaw rate decreases and one axle may become saturated before another. If the lateral force capacity in front axle reduces more than it does in rear axle, the vehicle is prone to understeer condition. If rear axle reduces more, the vehicle is prone to oversteer. However, considering the complexity of saturation in different tyres taking place at different moments can be difficult. In this work, it is assumed that the saturation happens in the same time. Therefore, the saturation of rear axle can be included in the safe maximum yaw rate as:

$$r_{max} = \frac{g\mu}{u} \quad (3.15)$$

The saturation of the rear tires is also important for finding the maximum stable limit of lateral velocity. The maximum stable lateral velocity can be obtained by limiting the rear

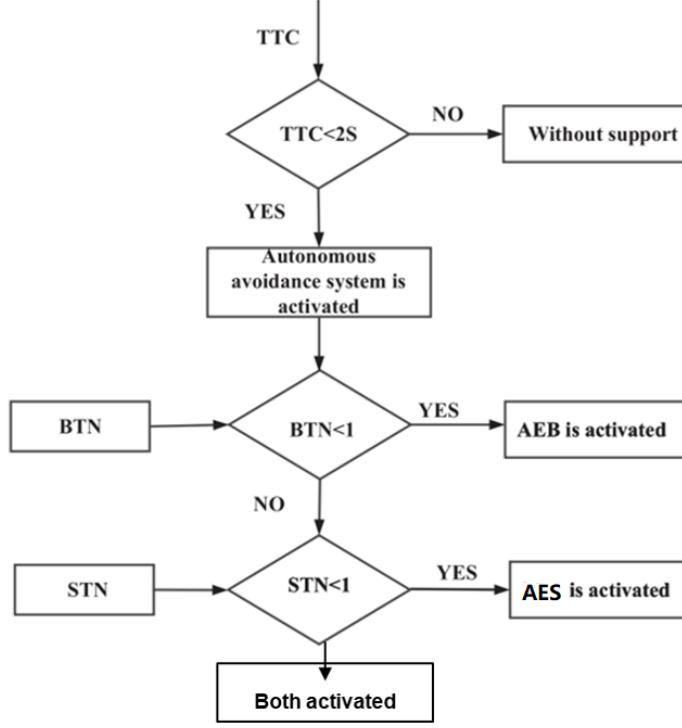


Figure 3.5: The logic decides which mode should the controller be on

slip angle to $|\alpha_r| \leq \alpha_{r,sat}$ and using Equation 3.6 we obtain:

$$l_r r - u\alpha_{r,sat} \leq v \leq l_r r + u\alpha_{r,sat} \quad (3.16)$$

The defined state constraints Equation 3.15 and 3.16 creates a parallelogram in the state space, which is called 'Stable Handling envelop'. It can be included within a Model Predictive Control as a safety constraint to keep the predicted states inside the envelop so as to ensure its stability. The constraint can be represented by the following linear inequality:

$$\Gamma_s \mathbf{x}(k) \leq G_s \quad (3.17)$$

where

$$\Gamma_s = \begin{bmatrix} 0 & 1 & -l_r & 0 & 0 \\ 0 & -1 & l_r & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \quad G_s = \begin{bmatrix} u\alpha_{r,sat} \\ u\alpha_{r,sat} \\ r_{max} \\ r_{max} \end{bmatrix}$$

The second safety constraint is defined for the vehicle in AES mode to plan an obstacle-free path within the road edges, which is called the environment envelop. Since in AES mode, the longitudinal velocity is kept close to the desired velocity which is known over a prediction horizon, so the vehicle distance along the path is known over the horizon. The camera in

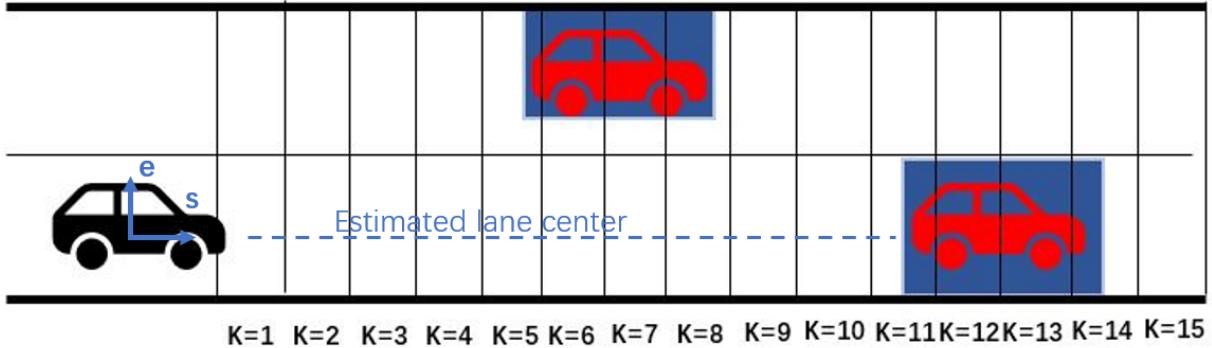


Figure 3.6: The environment envelop. The blue blocks are inaccessible regions

perception module tells the center line of the current lane as well as the lateral deviation of the vehicle from the center line. Given a center line, the relative lateral position of the vehicle is calibrated. By constraining the fifth state, the lateral deviation between an upper and lower bounds, which are set to be within the road edges and free of obstacles (other vehicles). Figure 3.6 shows other two vehicles moving ahead of the ego vehicle. One is moving in the adjacent lane another in our lane. The safety constraint on the lateral deviation is defined as:

$$e_{min}(k) + \left(\frac{w}{2} + e_s\right) < e(k) < e_{max}(k) - \left(\frac{w}{2} + e_s\right) \quad (3.18)$$

where w is the vehicle width and e_s is a minimum safe distance to the boundaries of the safe region. The safe region on position states can be written as:

$$\Gamma_e \mathbf{x}(k) \leq G_e \quad (3.19)$$

where

$$\Gamma_e = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad G_e = \begin{bmatrix} e_{max}(k) - \left(\frac{w}{2} + e_s\right) \\ -e_{max}(k) - \left(\frac{w}{2} + e_s\right) \end{bmatrix}$$

The third constraint is defined for the vehicle in AEB mode to stop before obstacles. Since in AEB mode, mainly the longitudinal net force is put into effect. Therefore, the longitudinal velocity varies over a prediction horizon, resulting in a variable prediction length, which is named as 'elastic length envelop' by the author as shown in Figure 3.7. Firstly, the ego vehicle decides which vehicle is the most important object. Then if the MIO is outside of safe distance but within the prediction length. By constraining the distance traveled by the ego vehicle over a prediction horizon, the ego vehicle can keep a fixed distance from the MIO. If the MIO is within the safe distance, by compressing the prediction length such as to leave the MIO outside and obtained the desired deceleration from the mode switch logic, the ego vehicle can do emergency braking as AEB does in such situations. The constraints

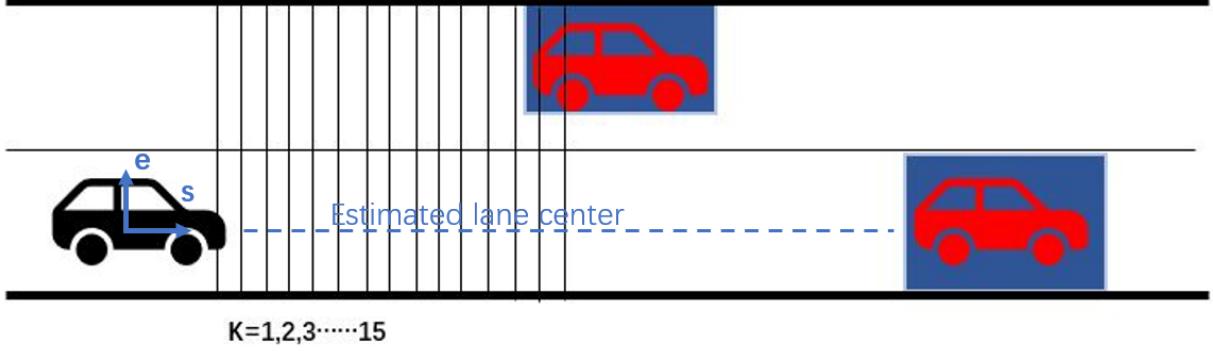


Figure 3.7: Figure 3.6 is in AES mode while this in AEB mode. By extending or compressing the prediction length, Adaptive Cruise Control and AEB functions can be achieved.

can be written as follows:

$$s(k) = \sum_{i=1}^k u(i)t_s^k < G_x(k) \quad (3.20)$$

which can be written as follows:

$$\Omega \Gamma_{x,mat} X T < G_x \quad (3.21)$$

where

$$\Omega = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 \end{bmatrix}_{N \times N}, \quad \Gamma_{x,mat} = \begin{bmatrix} \Gamma_x & 0 & \cdots & 0 \\ 0 & \Gamma_x & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \Gamma_x \end{bmatrix}$$

$$\Gamma_x = [1 \ 0 \ 0 \ 0 \ 0], \quad X = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}, \quad T = [t_s^1 \ t_s^2 \ \cdots \ t_s^N]$$

B. Actuator Constraints

The maximum capacity of the control actuations F_{xnet} , F_{yf} and M_{Fx} are constrained based on the maximum physical capability of the actuators and the maximum friction force capacity between road and tires. These actuator constraints can be written as:

$$\begin{aligned} |u| &\leq u_{max} \\ |\Delta u| &\leq \Delta u_{max}^k \end{aligned} \quad (3.22)$$

where Δu_{max}^k is the maximum allowable increment in the longitudinal net force, front lateral force and the yaw moment generated by differential braking. These maximum slew rates are dependent on the time step size. Namely, the control increments can get larger for longer time steps:

$$\Delta u_{max}^k = \frac{t_s^k}{t_{short}} \Delta u_{max}^0 \quad (3.23)$$

3.2.2 Optimization Problem Formulation

The following optimization problem calculates the optimized control inputs at each time step, generating a trajectory that can best follow a desired path while satisfying the constraints related to stability and position.

$$\begin{aligned} J = \sum_{k=1}^N & [\mathbf{x}^T(k) - x_{ref}^T(k)] \mathbf{Q} [\mathbf{x}(k) - x_{ref}(k)] + [\mathbf{u}^T(k) - u_{ref}^T] \mathbf{R} [\mathbf{u}(k) - u_{ref}] \\ & + \Delta \mathbf{u}^T(k) \mathbf{P} \Delta \mathbf{u}(k) + \rho_s \epsilon_s^2 + \rho_e \epsilon_e^2 + \rho_x \epsilon_x^2 \end{aligned} \quad (3.24)$$

subject to the discrete-time vehicle model Equation 3.13 for ($k=1, \dots, N$), and

$$|\mathbf{u}(k)| \leq \mathbf{u}_{max}^k \quad (3.24a)$$

$$|\Delta \mathbf{u}(k)| \leq \Delta \mathbf{u}_{max}^k \quad (3.24b)$$

$$\mathbf{\Gamma}_s \mathbf{x}(k) \leq \mathbf{G}_s^k + \epsilon_s(k) \quad (3.24c)$$

$$\mathbf{\Gamma}_e \mathbf{x}(k) \leq \mathbf{G}_e^k + \epsilon_e(k) \quad (3.24d)$$

$$\mathbf{\Gamma}_x \mathbf{x}(k) \leq \mathbf{G}_x^k + \epsilon_x(k) \quad (3.24e)$$

$$\epsilon_s(k) \geq 0 \quad (3.24f)$$

$$\epsilon_e(k) \geq 0 \quad (3.24g)$$

$$\epsilon_x(k) \geq 0 \quad (3.24h)$$

where $\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$. The first term in the cost function is for tracking the desired path. This term penalizes heading and lateral deviations with respect to the desired path. The second term penalizes the magnitude of control actuation. The magnitude of the front lateral force F_{fy} is not penalized, but the other two are penalized. It means to make

the controller rely on front steering and longitudinal braking more and only use differential braking when necessary. The first component of u_{ref} is set to be desired brake forces when AEB or mixed mode is on but the other two components are kept zero always. The third term penalizes the increments of control actuation, allowing the vehicle to hold a constant steering angle, zero braking and change smoothly in normal situations. The last three terms penalize for slack variables that are added to ensure feasibility of the optimization problem. The first slack variable ϵ_s is used for the handling limit envelop. The second ϵ_e is for the environment envelop. The last ϵ_x is for the elastic length envelop. The positive semi-definite weight matrices $\mathbf{Q}, \mathbf{R}, \mathbf{P}$ determines the priority between the control objectives. These weights are set such that the priority is given first to collision avoidance and then stability, tracking reference the last.

The proposed MPC controller is solved using quadratic optimization method. This convex optimization has a quadratic cost function $\frac{1}{2}\mathbf{z}^T \mathbf{H} \mathbf{z} + \mathbf{g}^T \mathbf{z}$ with a linear constraint $\mathbf{A} \mathbf{z} \leq \mathbf{B}$, which is capable to execute in real-time. This approach gives an optimized sequence of control inputs for the whole prediction horizon at each execution of the controller. The first set of control inputs in the sequence is taken. The conversion from MPC problem Equation 3.24 to standard quadratic programming is shown as follows:

Here an augmented state is constructed

$$\xi(k|t) = \begin{bmatrix} \mathbf{x}(k|t) \\ \mathbf{u}(k-1|t) \end{bmatrix} \quad (3.25)$$

A new vehicle state-space model is obtained:

$$\xi(k+1) = \tilde{\mathbf{A}}^k \xi(k) + \tilde{\mathbf{B}}^k \Delta \tilde{\mathbf{U}}(k) + \tilde{\mathbf{d}}^k \quad (3.26)$$

$$\eta(k) = \tilde{\mathbf{C}} \xi(k) \quad (3.27)$$

where

$$\tilde{\mathbf{A}}^k = \begin{bmatrix} A^k & B_1^k + B_2^k \\ \mathbf{0}_{3 \times 5} & \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad \tilde{\mathbf{B}}^k = \begin{bmatrix} B_1^k + B_2^k & B_2^k \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}$$

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

$$\tilde{\mathbf{d}}^k = \begin{bmatrix} d^k \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \quad \Delta \tilde{\mathbf{U}}(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix}, \quad k = 1, \dots, N$$

The system output predictions over a horizon can be written in matrix form:

$$\mathbf{X}(t) = \boldsymbol{\Psi}_t \xi(t|t) + \boldsymbol{\Theta}_t \Delta \mathbf{U}(t) + \boldsymbol{\Omega}_t \mathbf{D}_t \quad (3.28)$$

where

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{x}(t+1|t) \\ \mathbf{x}(t+2|t) \\ \vdots \\ \mathbf{x}(t+N_c|t) \\ \vdots \\ \mathbf{x}(t+N|t) \end{bmatrix}, \quad \boldsymbol{\Psi}_t = \begin{bmatrix} \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^k \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{A}}_t^k \\ \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+N_c-1} \dots \tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{A}}_t^k \\ \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+N_p-1} \dots \tilde{\mathbf{A}}_t^{k+2}\tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{A}}_t^k \end{bmatrix}$$

$$\boldsymbol{\Theta}_t = \begin{bmatrix} \tilde{\mathbf{C}}\tilde{\mathbf{B}}_t^k & 0 & 0 & 0 \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{B}}_t^k & \tilde{\mathbf{C}}\tilde{\mathbf{B}}_t^{k+1} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+N_c-1} \dots \tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{B}}_t^k & \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+N_c-2} \dots \tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{B}}_t^{k+1} & \dots & \tilde{\mathbf{C}}\tilde{\mathbf{B}}_t^{k+N_c-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+N_p-1} \dots \tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{B}}_t^k & \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+N_p-2} \dots \tilde{\mathbf{A}}_t^{k+1}\tilde{\mathbf{B}}_t^{k+1} & \dots & \tilde{\mathbf{C}}\tilde{\mathbf{B}}_t^{k+N_c-1} \end{bmatrix}$$

$$\Delta \mathbf{U}(t) = \begin{bmatrix} \Delta \tilde{\mathbf{U}}(k|t) \\ \Delta \tilde{\mathbf{U}}(k+1|t) \\ \vdots \\ \Delta \tilde{\mathbf{U}}(k+N_c|t) \end{bmatrix}, \quad \mathbf{D}_t = \begin{bmatrix} \tilde{\mathbf{d}}_t^k \\ \tilde{\mathbf{d}}_t^{k+1} \\ \vdots \\ \tilde{\mathbf{d}}_t^{k+N_p-1} \end{bmatrix}$$

$$\boldsymbol{\Omega}_t = \begin{bmatrix} \tilde{\mathbf{C}} \mathbf{I}_{8 \times 8} & \mathbf{0}_{5 \times 8} & \dots & \mathbf{0}_{5 \times 8} \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}_t^{k+1} & \tilde{\mathbf{C}} \mathbf{I}_{8 \times 8} & \dots & \mathbf{0}_{5 \times 8} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}\mathbf{A}_t^{k+N_p-1} \dots \mathbf{A}_t^{k+1} & \tilde{\mathbf{C}}\mathbf{A}_t^{k+N_p-1} \dots \mathbf{A}_t^{k+2} & \dots & \tilde{\mathbf{C}}\mathbf{A}_t^{k+N_p-1} \end{bmatrix}$$

Substitute Equation 3.28 into 3.24, we can denote the output error over prediction horizon as $E(t)$:

$$\mathbf{E}(t) = \Psi_t \xi(t|t) - \mathbf{X}_{ref}(t) \quad (3.29)$$

After series of matrix calculation, the optimization problem can be written as:

$$\begin{aligned} \mathbf{J} = & [\Delta \mathbf{U}(t)^T, \epsilon_s, \epsilon_e, \epsilon_x^T] \mathbf{H}_t [\Delta \mathbf{U}(t), \epsilon_s, \epsilon_e, \epsilon_x] \\ & + \mathbf{G}_t [\Delta \mathbf{U}(t)^T, \epsilon_s, \epsilon_e, \epsilon_x^T] + \mathbf{P}_t \end{aligned} \quad (3.30)$$

where

$$\mathbf{H}_t = \begin{bmatrix} \boldsymbol{\Theta}^T \mathbf{Q} \boldsymbol{\Theta} + \boldsymbol{\Phi}^T \mathbf{R} \boldsymbol{\Phi} + \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \rho_s & 0 & 0 \\ \mathbf{0} & 0 & \rho_e & 0 \\ \mathbf{0} & 0 & 0 & \rho_x \end{bmatrix}$$

$$\mathbf{G}_t = [2[\mathbf{E}^T \mathbf{Q} \boldsymbol{\Theta} + \boldsymbol{\Omega} \mathbf{D}_t^T \mathbf{Q} \boldsymbol{\Theta} + \mathbf{U}_t^T \mathbf{R} \boldsymbol{\Phi}] \mathbf{0} \mathbf{0} \mathbf{0}]$$

It should be noted that in H_t and G_t the bold 0 represents zero matrix but the non-bold 0 represents number zero. U_t is the control inputs in last control execution. Φ is constructed to calculate the matrix of the sequence of planned control inputs: $U(t) = U_t + \Phi \Delta U(t)$. The constraints formulation and the derivation of other matrix i.e. Φ can be referred to section C in Appendix.

3.3 Stochastic MPC controller design

In section 3.2.2 a classical MPC problem is formulated which is able to work safely and effectively in many cases. However, classical MPC approaches do not account for system uncertainties that may result in poor decisions of the controller, leading to severe safety problems. Robust MPC and Stochastic MPC approaches are widely used to work these problems out. But robust control approaches that deal with worst-case disturbances may be too conservative and computational expensive. Therefore, this thesis proposes a Stochastic MPC framework for controlling autonomous vehicles where the risk of constraint violations is a tunable design parameter which can be estimated by the perception module and be adjusted as the environment changes.

Moreover, since autonomous vehicles are driving in dynamic and uncertain environment, ego vehicle's sensor inaccuracy and other vehicles' uncertain maneuvers are both required to be considered in controller's decision-making. The challenge is to not only handle uncertainty in the state of the ego vehicle but effectively handle uncertainty in the predicted future motion of surrounding vehicle. Regarding uncertainties of other vehicles, 1) vehicles can make one of multiple possible maneuvers in the future, 2) the precise execution of a future maneuver cannot be predicted with perfect accuracy. It should be noted that the first type

of uncertainties have more serious effect. So in this work only the first type is considered. An Interacting Multiple Model Kalman Filter is used to estimate the type of manuevers of other vehicle in the future, which is later used for constructing the environment envelop and the elastic length envelop. An illustration of how this automated driving assistant system is constructed is shown in Figure

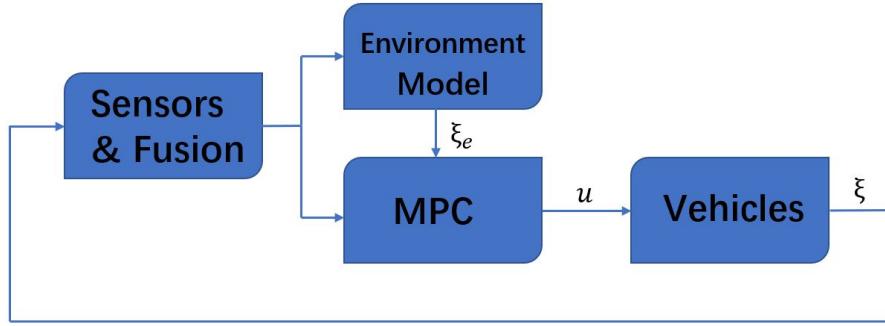


Figure 3.8: Block diagram of overal ADAS

3.3.1 Target Vehicle Models and IMMKF

A. Target Vehicle Models

The dynamics of the i^{th} target vehicle are described by a Markov jump affine system,

$$\begin{aligned} \mathbf{x}_{i|k+1}^t &= \mathbf{F}_k^{(i)} \mathbf{x}_{i|k}^t + \mathbf{G}_k^{(i)} w_k^{(i)} + \mathbf{E}_k^i \\ \mathbf{y}_{i|k}^t &= \mathbf{H}_k^{(i)} \mathbf{x}_{i|k}^t + v_k^{(i)} \end{aligned} \quad (3.31)$$

where $\mathbf{x}_{i|k}^t$ and $\mathbf{y}_{i|k}^t$ denote the target vehicle state and measurement at time k . The process noise $w_k^{(i)}$ and measurement noise $v_k^{(i)}$ are assumed to response to normal distributions $\mathcal{N}(0, \mathbf{Q}_k^{(i)})$ and $\mathcal{N}(0, \mathbf{R}_k^{(i)})$, respectively. The superscript i in Equation 3.31 refers to the model $m^{(i)}$ among three models. The author chose $\mathbf{x}^t = [s, \dot{s}, \ddot{s}, e, \dot{e}]^T$ as states, including longitudinal distance, velocity, acceleration, lateral position and velocity. Three models are as follows:

- 1) lane keeping at constant velocity

$$\mathbf{F}_k^{(1)} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_k^{(1)} = \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \\ 1 \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix} \quad (3.32)$$

2) braking in current lane

$$\mathbf{F}_k^{(2)} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_k^{(2)} = \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \\ 1 \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix} \quad (3.33)$$

3) lane changing

$$\mathbf{F}_k^{(3)} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G}_k^{(3)} = \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \\ 1 \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix}, \quad \mathbf{E}_k^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ T_s K e_{y,ref} \end{bmatrix} \quad (3.34)$$

B. Interacting Multiple Model Kalman Filter (IMMKF)

Transitions between models have fixed probabilities which are denoted as π_{ij} . At each time step k , we estimate the following quantities based on the current measurement and values at the previous time step:

- 1) Model probability $\mu_k^{(i)}$.
- 2) Model-conditioned posterior means and covariances: $\hat{\xi}_{k|k}^{(i)}$ and $\hat{\Sigma}_{k|k}^{(i)}$.
- 3)

Step 1. Evaluate mixing probabilities:

$$\mu_{k-1}^{j|i} = \frac{\pi_{ij} \mu_{k-1}^{(j)}}{\mu_{k|k-1}^{(i)}} \quad (3.35)$$

where

$$\mu_{k|k-1}^{(i)} = \sum_{j=1}^M \pi_{ji} \mu_{k-1}^{(j)}$$

Mixing estimates:

$$\bar{\xi}_{k-1|k-1}^{(i)} = \sum_{j=1}^M \hat{\xi}_{k-1|k-1}^{(j)} \mu_{k-1}^{j|i} \quad (3.36)$$

Mixing covariance:

$$\bar{\Sigma}_{k-1|k-1}^{(i)} = \sum_{j=1}^M [\hat{\xi}_{k-1|k-1}^{(j)} + (\bar{\xi}_{k-1|k-1}^{(i)} - \hat{\xi}_{k-1|k-1}^{(i)}) \cdot (\bar{\xi}_{k-1|k-1}^{(i)} - \hat{\xi}_{k-1|k-1}^{(i)})^T] \mu_{k-1}^{j|i} \quad (3.37)$$

Step 2. For each model $M^{(i)}$, run Kalman Filter with inputs $(\bar{\xi}_{k-1|k-1}^{(i)}, \bar{\Sigma}_{k-1|k-1}^{(i)})$ using Algorithm 1. The model-conditioned estimates are obtained:

$$(\hat{\xi}_{k|k}^{(i)}, \hat{\Sigma}_{k|k}^{(i)}, \hat{y}_k^{(i)}, S_k^{(i)})$$

Step 3. Update the probability of each model,

$$\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)} \mathcal{N}(\hat{y}_k^{(i)}; 0; S_k^{(i)})}{\sum_{j=1}^M \mu_{k|k-1}^{(j)} \mathcal{N}(\hat{y}_k^{(j)}; 0; S_k^{(j)})} \quad (3.38)$$

where $\mathcal{N}(x; \mu; \Sigma)$ is the probability density function of a multivariate normal distribution with mean μ and covariance Σ , evaluated at x .

Step 4. Compute the fused estimates,

$$\hat{\xi}_{k|k} = \sum_{j=1}^M \hat{\xi}_{k|k}^{(j)} \mu_k^{(j)} \quad (3.39)$$

$$\hat{\Sigma}_{k|k} = \sum_{j=1}^M [\Sigma_{k|k}^{(j)} + (\xi_{k|k}^{(j)} - \hat{\xi}_{k|k}^{(j)}) (\xi_{k|k}^{(j)} - \hat{\xi}_{k|k}^{(j)})^T] \mu_k^{(j)} \quad (3.40)$$

C. An Application Example of IMMKF

IMMKF embedded with the target models is assigned to track a target model in 100s. In the first 20s, it travels at constant velocity in the current lane. From 20s to 40s, the vehicle brakes at a constant deceleration of $-0.2m/s^2$. After that, it changes its lane to the adjacent lane after travels at constant velocity for 20s. In the last 20s, the vehicle travels at constant velocity in the new lane as shown in Figure 3.9.

In Figure 3.10, in the very beginning it needs a bit more time to identify which model fits best to the target vehicle's motion since the initial conditions are just set randomly. But it isn't hard to notice that the constant velocity model is the one having the largest probability over this period. In next 20s, the constant acceleration model is identified even without questions. From 40 to 60s the constant velocity model is assigned. From 60s to 80s, 80s to 100s, the lane change model and constant velocity model is identified, respectively. In general, switching from one model to another is quickly enough even in transition except the very beginning due to the well-known reason. Therefore, it is fair to say IMMKF and its corresponding target models are designed properly.

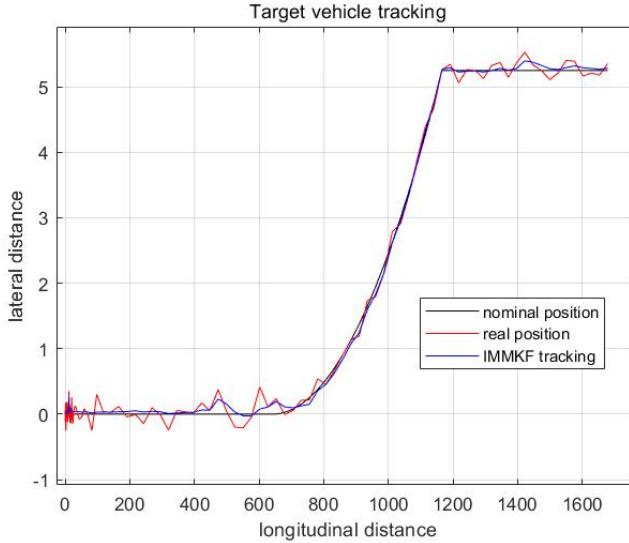


Figure 3.9: Nominal position/real position/IMMKF tracking results of the target vehicle in 100s

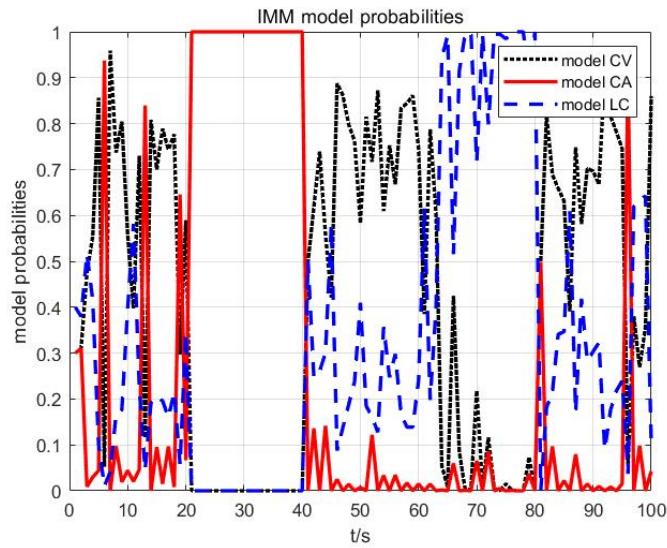


Figure 3.10: Model probabilities variation in 100s

3.3.2 SMPC Formulation

To account for uncertainties due to model mismatch and linearization errors, this work introduces an additive stochastic disturbance in system state-space model Equation 3.13. The disturbed vehicle model can be written as:

$$\xi(k+1) = \mathbf{A}^k \xi(k) + \mathbf{B}_1^k \mathbf{u}(k) + \mathbf{B}_2^k \mathbf{u}(k+1) + \mathbf{d}^k + \mathbf{G}_k \mathbf{w}_k \quad (3.41)$$

where $\mathbf{w}_k \sim \mathcal{N}(0, \Sigma_w)$.

A. Chance Environment Envelop Formulation

In section 3.2.2, Equation 3.19 represents the controlled vehicle can only move inside the

'envelop'. The RHS of Equation 3.19 is uncertain due to the uncertainty in the predicted target vehicle position. This can be dealt with by suitably increasing the dimensions of the 'inaccessible region' such that the target vehicle will be contained with a higher probability. Moreover, the LHS is also uncertain due to the uncertainty of the ego states arising from the additive disturbance in Equation 3.41. The Equation 3.24d of the optimization problem 3.24 in this section is reformulated as a chance constraint which could be satisfied with a specified probability p . That is,

$$\Pr(\boldsymbol{\Gamma}_e \xi(k) \leq \mathbf{G}_e^k + \boldsymbol{\epsilon}_e(k)) \geq p, \quad 0.5 \leq p \leq 1 \quad (3.42)$$

where p denotes the tunable risk parameter which can be adjusted according to the reliability of the perception module. It should be noted that a higher p would lead to conservative behavior, possibly making the chance-constrained optimal control problem infeasible. In the other hand, a smaller p might result in a higher risk of collision. A new optimization problem is now formulated with Equation 3.42 replacing Equation 3.24d in optimization problem 3.24.

B. Open Loop Approach

This work uses the open loop paradigm introduced in [7] to decompose the state and input as,

$$\xi_k = \mathbf{z}_k + \mathbf{e}_k \quad (3.43)$$

where \mathbf{z}_k denotes the deterministic component of ξ_k , and \mathbf{e}_k denotes the stochastic component. Substituting the equation above into Equation 3.41 gives,

$$\mathbf{z}_{k+1} = \mathbf{A}_k \mathbf{z}_k + \mathbf{B}_k \mathbf{u}_k \quad (3.44)$$

$$\mathbf{e}_{k+1} = \mathbf{A}_k \mathbf{e}_k + \mathbf{G}_k \mathbf{w}_k \quad (3.45)$$

It should be noted that in this case the variance of the stochastic component \mathbf{e} evolves in an uncontrolled fashion. Especially in the case the system is unstable, this approach has significant drawbacks, since it may induce serious feasibility problems. However, although the predicted evolution of states cannot ever be accurate Model Predictive Control approach is still able to stabilize vehicles since it solves optimization problem frequently, which gives it a nature of robustness in most cases, capable of eliminating errors or deviations from the reference states. The author has studied the closed loop approach as well. However, calculating the LQR gain \mathbf{K}_k for each pair of $(\mathbf{A}_k, \mathbf{B}_k)$ takes much time and computational resources which makes really hard for the proposed controller to be implemented on the real hardware. Therefore, the computation-saving approach open loop is selected.

C. Uncertainty Propagation

Assumption 3: The system has perfect state in the beginning which can be written as $\xi_0 = \mathbf{z}_0$. It implies $\mathbf{e}_0 = 0$ with probability 1.

The separation of the deterministic and stochastic components of ξ_k allows us to determine the distributions of \mathbf{e}_k , ($k = 1, \dots, H_p$) knowing the distributions of \mathbf{e}_0 and \mathbf{w}_k . Let $\mathbf{e}_k \sim \mathcal{N}(0, \Sigma_k)$. Then, $\mathbf{e}_{k+1} \sim \mathcal{N}(0, \Sigma_{k+1})$, where

$$\Sigma_{k+1} = \mathbf{A}_k \Sigma_k \mathbf{A}_k^T + \mathbf{G}_k \Sigma_w \mathbf{G}_k^T \quad (3.46)$$

The initial condition is $\Sigma_0 = 0$ under Assumption 3.

D. Constraint Tightening

Based on (3.43), the probabilistic constraint (3.42) becomes

$$\Pr(\Gamma_e \mathbf{z}(k) + \Gamma_e \mathbf{e}(k) \leq \mathbf{G}_e^k + \epsilon_e(k)) \geq p \quad (3.47)$$

This is satisfied if [22],

$$\begin{aligned} \Gamma_e \mathbf{z}(k) &\leq \mathbf{G}_e^k + \epsilon_e(k) - \gamma_k \\ \Pr(\Gamma_e \mathbf{e}(k) \leq \gamma_k) &= p \end{aligned} \quad (3.48)$$

Since we know that $\Gamma_e \mathbf{e}(k) \sim \mathcal{N}(0, \Sigma_k)$, we can explicitly compute γ_k from the quantile function of a univariate normal distribution as

$$\gamma_k = \sqrt{2\Gamma_e^T \Sigma_k \Gamma_e} \operatorname{erf}^{-1}(2p - 1) \quad (3.49)$$

where $\operatorname{erf}^{-1}(\cdot)$ is the inverse error function.

E. Modified Optimization Problem

By replacing Equation 3.24d with Equation 3.48, the optimization problem 3.24 is modified to the following problem,

$$\begin{aligned} J = \sum_{k=1}^N & [\mathbf{z}^T(k) - \xi_{ref}^T(k)] \mathbf{Q} [\mathbf{z}(k) - \xi_{ref}(k)] + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k) \\ & + \Delta \mathbf{u}^T(k) \mathbf{P} \Delta \mathbf{u}(k) + \rho_s \epsilon_s^2 + \rho_e \epsilon_e^2 + \rho_x \epsilon_x^2 \end{aligned} \quad (3.50)$$

subject to the discrete-time vehicle model Equation 3.46 substituted into 3.25 to 3.30 for

($k=1,\dots,N$), and

$$|\mathbf{u}(\mathbf{k})| \leq \mathbf{u}_{max}^k \quad (3.50a)$$

$$|\Delta\mathbf{u}(\mathbf{k})| \leq \Delta\mathbf{u}_{max}^k \quad (3.50b)$$

$$\boldsymbol{\Gamma}_s \mathbf{z}(k) \leq \mathbf{G}_s^k + \epsilon_s(k) \quad (3.50c)$$

$$\boldsymbol{\Gamma}_e \mathbf{z}(k) \leq \mathbf{G}_e^k + \epsilon_e(k) - \gamma_k \quad (3.50d)$$

$$\boldsymbol{\Gamma}_x \mathbf{z}(k) \leq \mathbf{G}_x^k + \epsilon_x(k) \quad (3.50e)$$

$$\epsilon_s(k) \geq 0 \quad (3.50f)$$

$$\epsilon_e(k) \geq 0 \quad (3.50g)$$

$$\epsilon_x(k) \geq 0 \quad (3.50h)$$

This problem is a quadratic program(QP) problem. Some derivations are neglected for simplicity. Basically, this formulation is similar to the one in section 3.2.2 where more specific derivation and transformation of states, matrix are included.

4 Simulation Results

In this section, the performance of the designed controller is evaluated numerically using Matlab/Simulink. The test vehicle is a simplified model of Chevrolet Equinox with the specifications listed in Table 4.1. The designed parameters are specified in Table 4.2. The performance of the controller is assessed in several driving scenarios with target vehicles on the road. The tests, carried out on a road with the friction coefficient of $\mu = 0.5$ and the desired longitudinal velocity of 60 kph, are performed first in the classical MPC controller in the first section 4.1. In some situations, the classical MPC controller is not able to ensure the safety so the SMPC controller is assessed to deal with such situations which will be included in the the second section 2.5.

Parameter	Value	Description
m	2272[kg]	Total mass
I_z	4600[$kg \cdot m^2$]	Vehicle moment of inertia
w	1.6[m]	Track width
l_f, l_r	1.11,1.67[m]	Distance from front/rear axle to CG
R_e, l_r	0.351[m]	Effective radius
$C_{\alpha f}$	182200[N/rad]	Front cornering stiffness
$C_{\alpha r}$	182200[N/rad]	Rear cornering stiffness

Table 4.1: Specifications of the test vehicle model

Parameter	Value
N_1, N_2, N_3	40,10,20
N_c	6
t_{short}, t_{long}	0.05, 0.2[s]
Q	$\text{diag}(4 \ 0 \ 0 \ 100 \ 4)$
R	$\text{diag}(4 \times 10^{-5}, 0, 4 \times 10^{-5})$
P	$\text{diag}(0, 3 \times 10^{-5}, 8 \times 10^{-5})$
ρ_s	3×10^3
ρ_e	5×10^4
ρ_x	1×10^6

Table 4.2: Specifications of the controller parameters

4.1 Classical MPC Controller Results

A. Scenario 1

The first scenario is a straight road with road with the centerline as the desired path. A lead vehicle is driving initially 50m ahead of the controlled vehicle. In the begining both vehicles are travelling at a constant velocity of 54 kph. In 3s, the lead vehicle starts braking

at a constant deceleration of $-5m/s^2$ due to some reasons. After 2s, it starts travelling at constant velocity of 5m/s. The top and bottom edges of the road are $e=6m$ and $e=-6m$ respectively. The left side and right side of the lead vehicle is big enough for the controlled vehicle to steer onto.

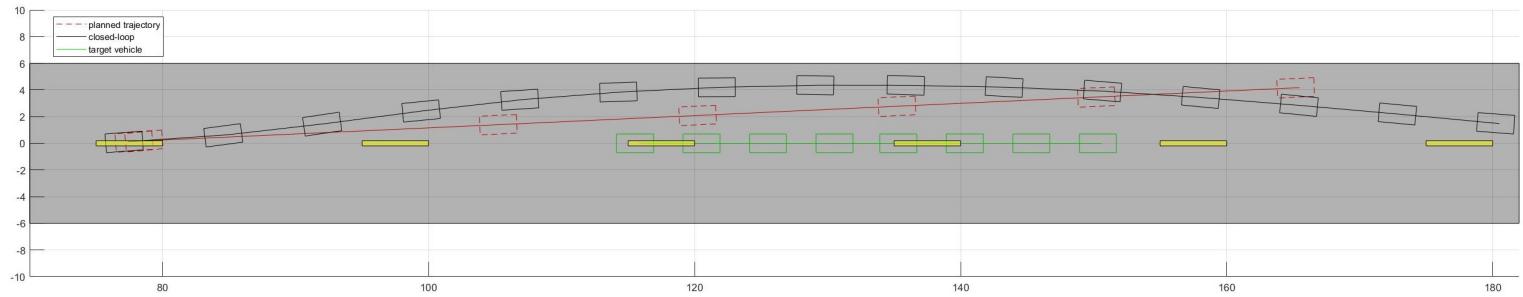


Figure 4.1: The figure shows the planned trajectory at 5.1s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment (Section 4.1, A)

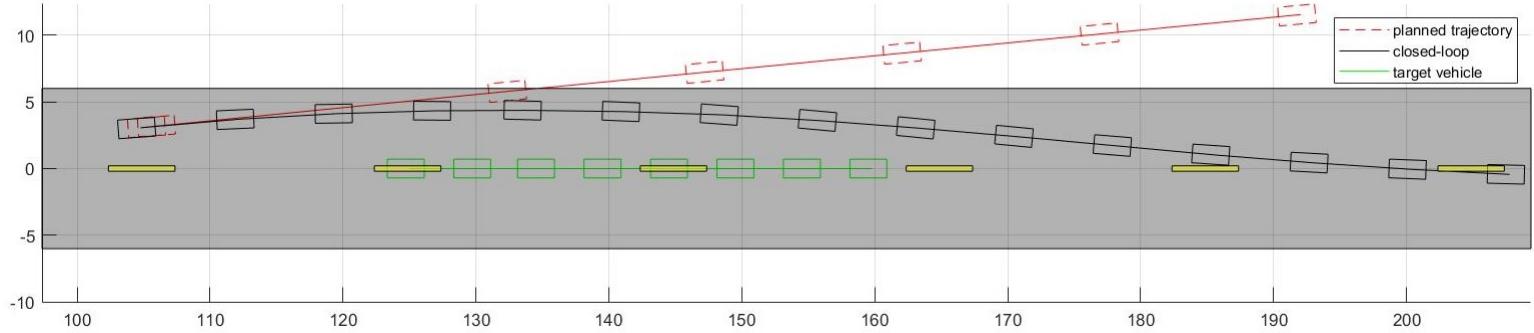


Figure 4.2: The figure shows the planned trajectory at 7s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment(Section 4.1, A)

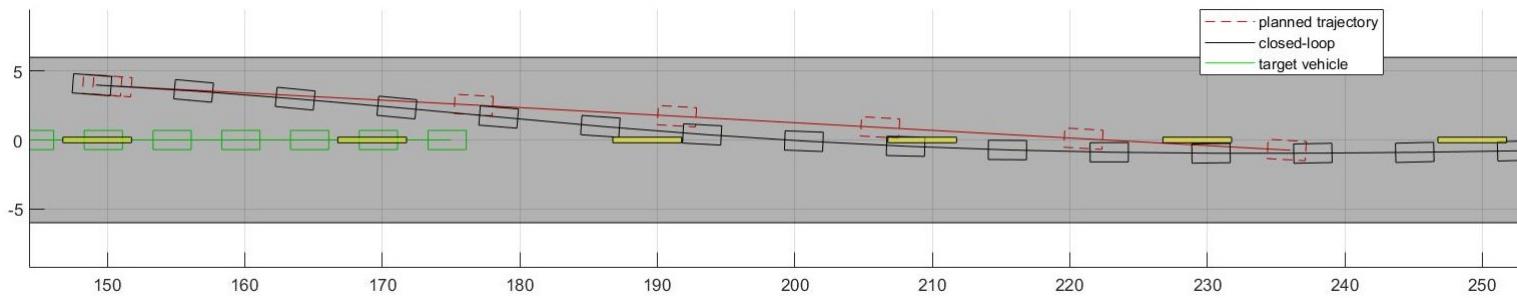


Figure 4.3: The figure shows the planned trajectory at 10s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment(Section 4.1, A)

As Figure 4.1 shows, at 5.1s the controlled vehicle starts planning to steer left. At 7s, the controlled vehicle is already changing the lane but is still required to be further left to keep a safe lateral distance with the lead vehicle. Therefore, the planned trajectory is even outside of the road boundaries. However, the MPC problem is solved continuously over one execution and another. The closed loop trajectory shows the controlled vehicle will not really leave the

road but to move back to the reference path(the center line of the road). In 11s, the vehicle overtook the lead vehicle so starts moving back to the center line.

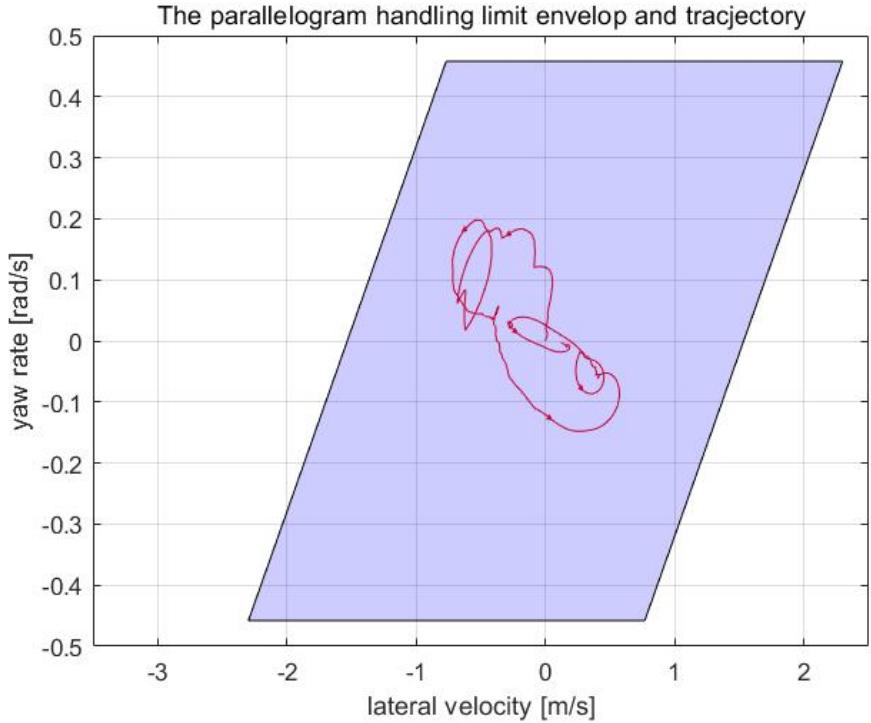


Figure 4.4: The vehicle trajectory in the phase plane of lateral velocity and yaw rate (Section 4.1. A)

Figure 4.4 shows the vehicle trajectory in terms of lateral velocity and yaw rate which is kept inside the handling limit envelop. The Figure 4.8 'FOH' shows the control inputs from 4s to 11s during which maneuvers mainly take place. The front steering is still the main source of the responses. The longitudinal forces are used to keep the longitudinal velocity at the desired value which is similar to the proportional longitudinal controller mentioned in [1]. The longitudinal velocity variations over the duration is shown 'FOH 'in Figure 4.5. The lateral velocity, yaw rate, yaw angle, lateral deviation are all shown in the following figures from 'FOH[4.6 to 4.7]. It should be noted that in Figure 4.9 solution time shows no matter in what situation the solution time is always less than 0.03s which would be good since the smallest sample time of the controller is set to be 0.05s, which means the controller can solve the optimization problem efficiently in time.

In section 3.1.2 the fixed time step discretization approach is compared with the varying time step approach. (ZOH and FOH) In this section, the scenario 1 is also used to test a controller based on the fixed time step approach. Comparisons are made to make it clear that using varying time step is more advantageous than another. The specifications of the controller are same to another, specified in Table 4.2 except the prediction horizon and sample time shown in Table 4.3.

The Figure 4.10 shows at 5.1s the vehicle just started planning to steer left which is later than the varying time step controller, leading to closer distance and less safer situation. The Figure 4.11 and 4.12 draw the same conclusion, which can explain why the trajectory in terms of lateral velocity and yaw rate in Figure 4.13 is much closer to the boundaries of the parallelogram compared to Figure 4.4. For the same reason, Figure 4.5 to Figure 4.7 shows the fixed step controller behaves more aggressively. In Figure 4.8, the control responses of varying step controller happen much earlier than another, leading to a better general

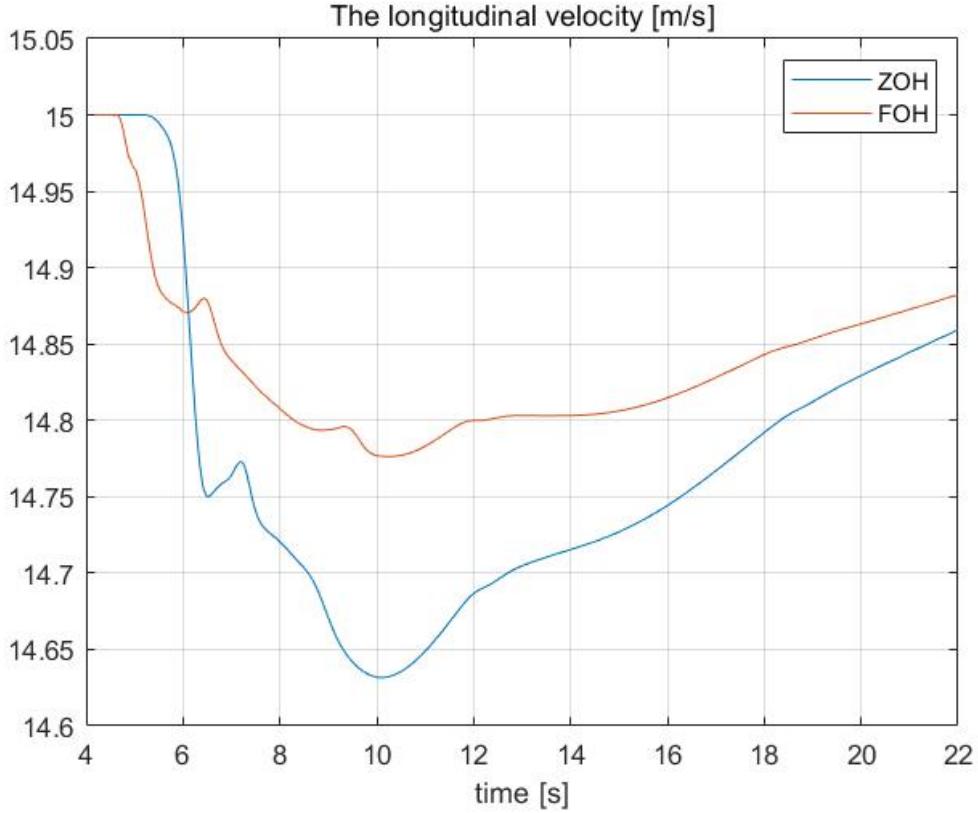


Figure 4.5: The comparison of longitudinal velocity between two controller (Section 4.1. A)

performance. In Figure 4.9 the solution time of varying step controller is a bit longer than another in general which is because it uses 70steps while another uses 60 steps. But anyway, the varying time step controller(denoted as FOH in figures) performs better.

Parameter	Value
N_P	60
t_S	0.05s

Table 4.3: Specifications of the fixed time step controller parameters

B. Scenario 2

Section 3.2.2 presents that the controller can either perform front steering, Autonomous Emergency Braking or integrated braking and steering. Scenario 1 has proven that the controller can steer the vehicle to avoid collision. Scenario 2 is designed to prove the rest of two modes. In scenario 2, there are two lead vehicles ahead, one in the center line another at right side. The left side is an inaccessible area due to some reasons. In the very beginning, the lead vehicle is 20m away from the controlled vehicle. They both travel at constant velocity of 54 kph. At 3s, the lead vehicle starts braking at constant deceleration of $-5m/s^2$ until it stops. The other lead vehicle holds the right side where it is accessible either. To avoid collision, steering mode cannot be used. AEB mode and integrated braking and steering mode are used accordingly. As Figure 4.15 shows, from 0s to around 3.8s the vehicle is traveling at the initial velocity. Then the 1st partial braking is enabled. After a few seconds, 2nd partial braking is also enabled. Before long, the full braking is required but it is not enough to ensure the constraint 3.24e. So it chose to steer a bit and finally successfully stops

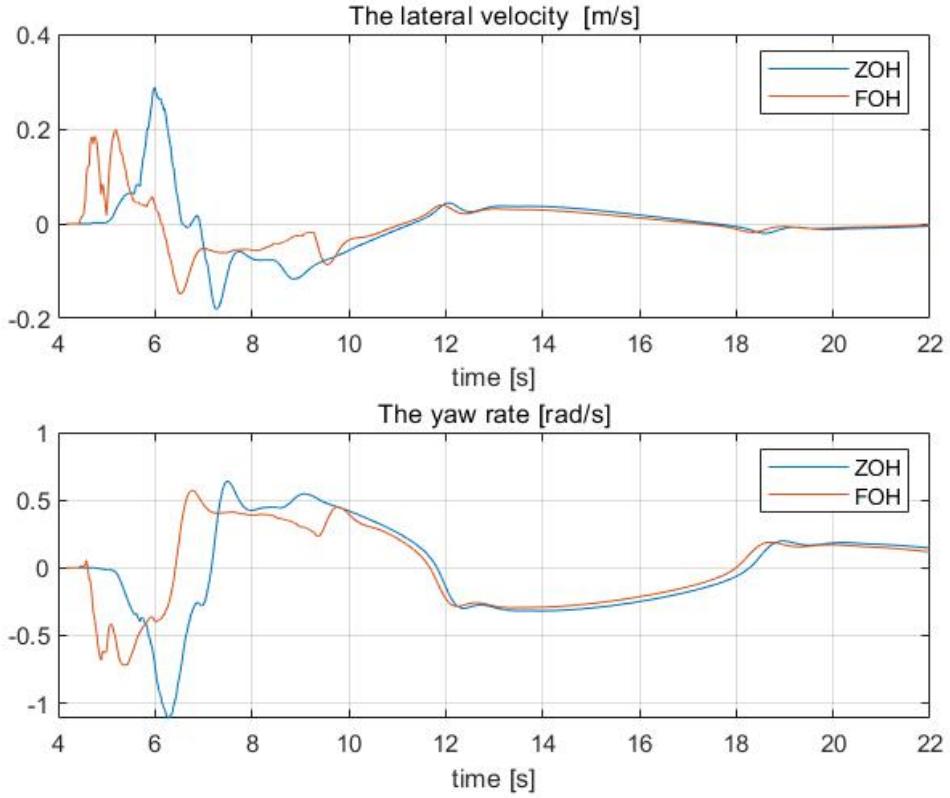


Figure 4.6: The comparison of lateral velocity and yaw rate between two controller (Section 4.1. A)

as Figure 4.14 shows. Collision avoidance in scenario 2 succeeded.

4.2 Stochastic MPC controller Results

In this section, the scenario 1 in last section will be assigned to study the effect of risk parameter ε on the controller's performance. The risk parameter is the probability that the environment envelop chance constraints' failure, which means a larger value of ε leads to higher probability of constraints violation. In this regard, the risk parameter can be used to adjust the controller to conservativeness or aggression accordingly. For instance, if the perception module of the autonomous vehicle is reliable enough then the controller can perform more aggressively. If not, the controller should focus on the safety more. Such risk parameter can be also estimated by some algorithms from the perception module to actively affect the decision-making of the controller, that is also why study on such parameter is interesting. Then, a third scenario is taken into consideration where the target vehicle behaves multiple maneuvers including travelling at constant velocity, braking in the lane and changing the lane. An Interacting Multiple Model Kalman Filter is also used to estimate the type of maneuvers that the target vehicle is taking to assist the controller to set up appropriate constraints.

A. Scenario 1

This scenario is the same to the one mentioned in last section. This time the Stochastic MPC controller is assigned to deal with this scenario. The Figure 4.17 and 4.18 shows the

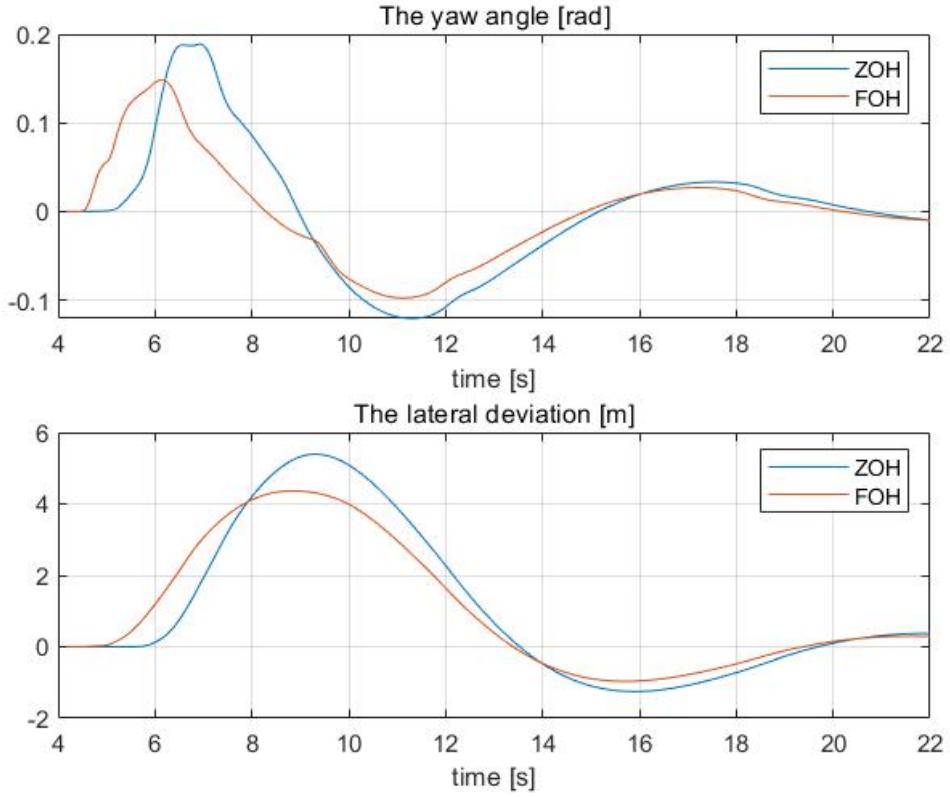


Figure 4.7: The comparison of yaw angle and lateral deviation between two controller (Section 4.1. A)

comparison of planned trajectories, the closed-loop trajectories between the classical MPC and SMPC controller. It should be noted that the two SMPC controllers both planned a trajectory closer to the center line although the one with $\varepsilon = 0.05$ is even closer to another. This is because the environment envelop has both lower and upper boundaries. The existence of chance constraints make the 'allowed region' even smaller such that the vehicle would be less likely to violate the normal constraints, which also fits to our intuition.

B. Scenario 3

In this scenario, the lead vehicle is 50m ahead of the controlled vehicle. They both travels at a constant velocity of 54 kph. In 3s, the lead vehicle starts braking in the lane at a constant deceleration of $-5m/s^2$, which maintains for 2s. Then it changes the lane to left, which take about 2s. After that, it travels at constant velocity of 20 kph. Figure 4.19 shows the variations of the model probabilities of the IMMKF. It should be noted that from 0 to 3s and 7s to 22 the constant velocity model has the highest model probability so the controller predicts the future states of the lead vehicle by this assumption. From 3s to 5s, the braking model (CA, constant acceleration) is identified. From 5s to 7s, the lane change model is identified with some delays. In Figure 4.20 the vehicle planned to steer left to avoid collision at 5.1s although at this moment the lead vehicle already starts changing the lane. But the controller is not aware of lead vehicle's lane change due to the delays of IMMKF. However, the controller reconsiders steering back and keeping the center line to avoid collision with the lead vehicle as Figure 4.21 shows.

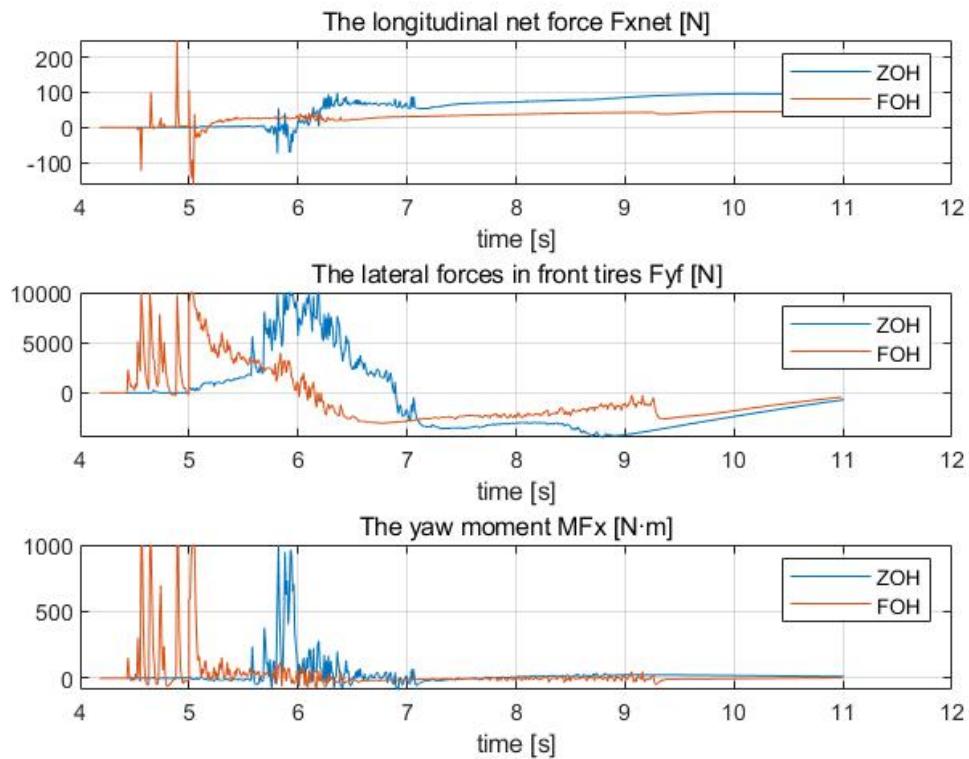


Figure 4.8: The comparison of control inputs between two controller (Section 4.1. A)

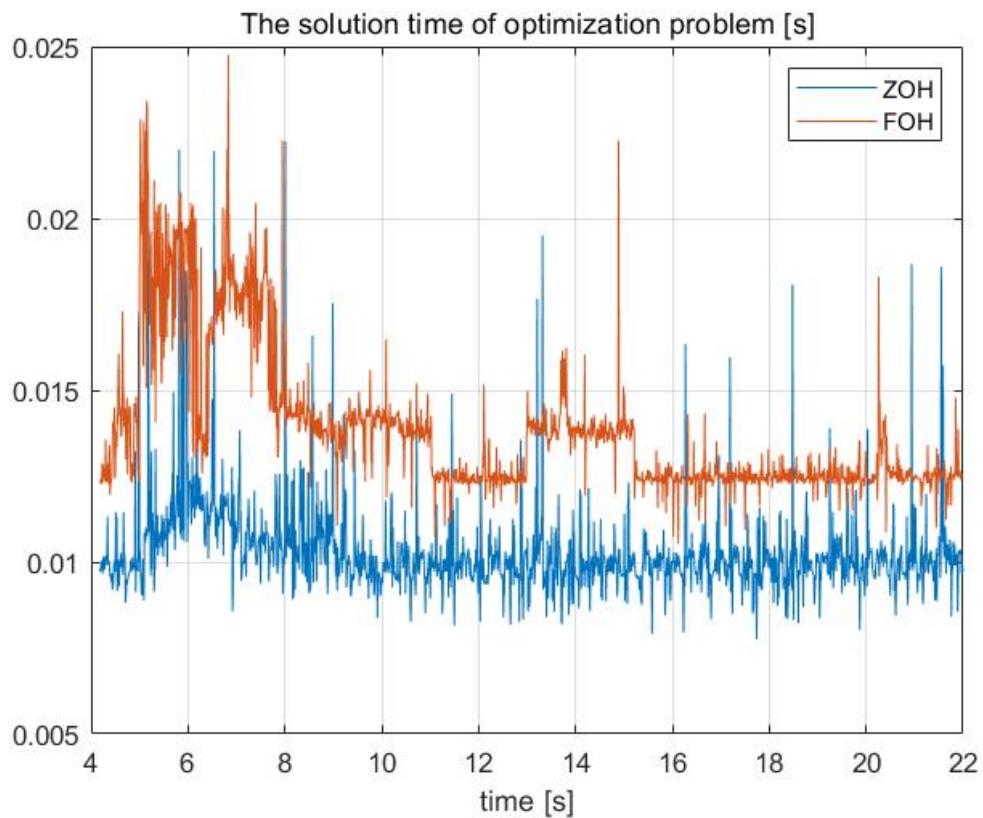


Figure 4.9: The comparison of solution time between two controller (Section 4.1. A)

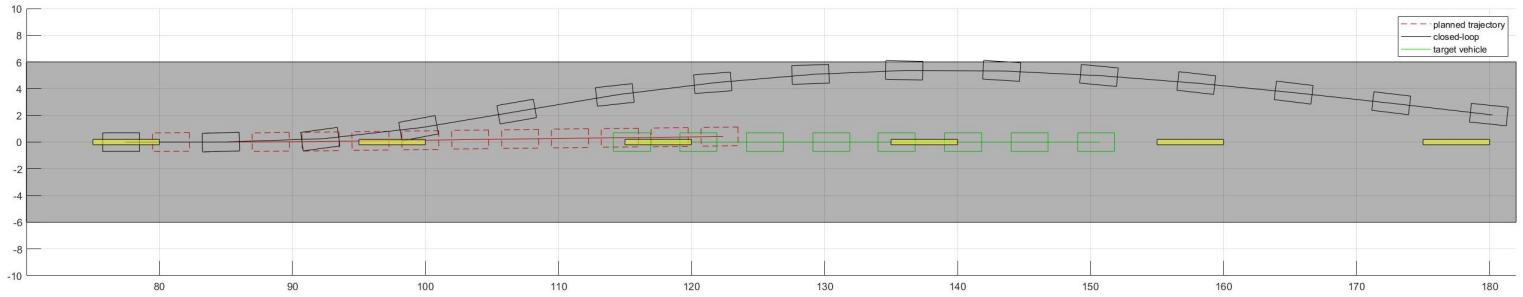


Figure 4.10: The figure shows the planned trajectory at 5.1s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment of the fixed time step controller (Section 4.1. A)

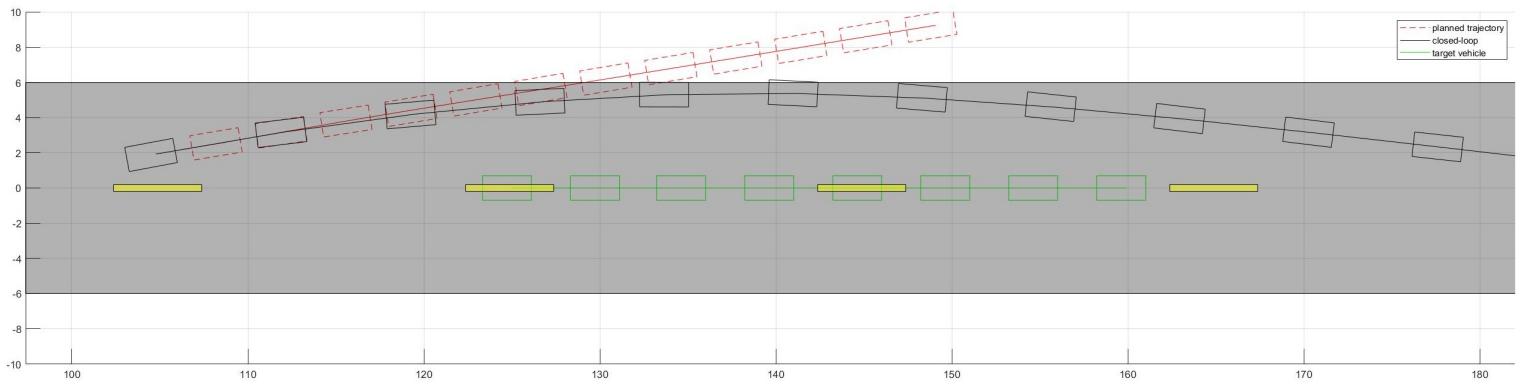


Figure 4.11: The figure shows the planned trajectory at 7s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment of the fixed time step controller (Section 4.1. A)

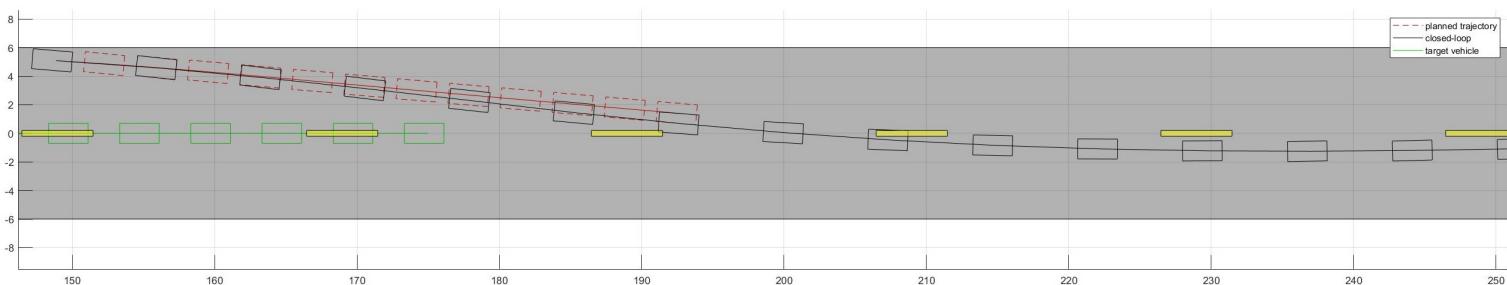


Figure 4.12: The figure shows the planned trajectory at 10s, closed-loop trajectory of the vehicle and trajectory of the lead vehicle around this moment of the fixed time step controller (Section 4.1. A)

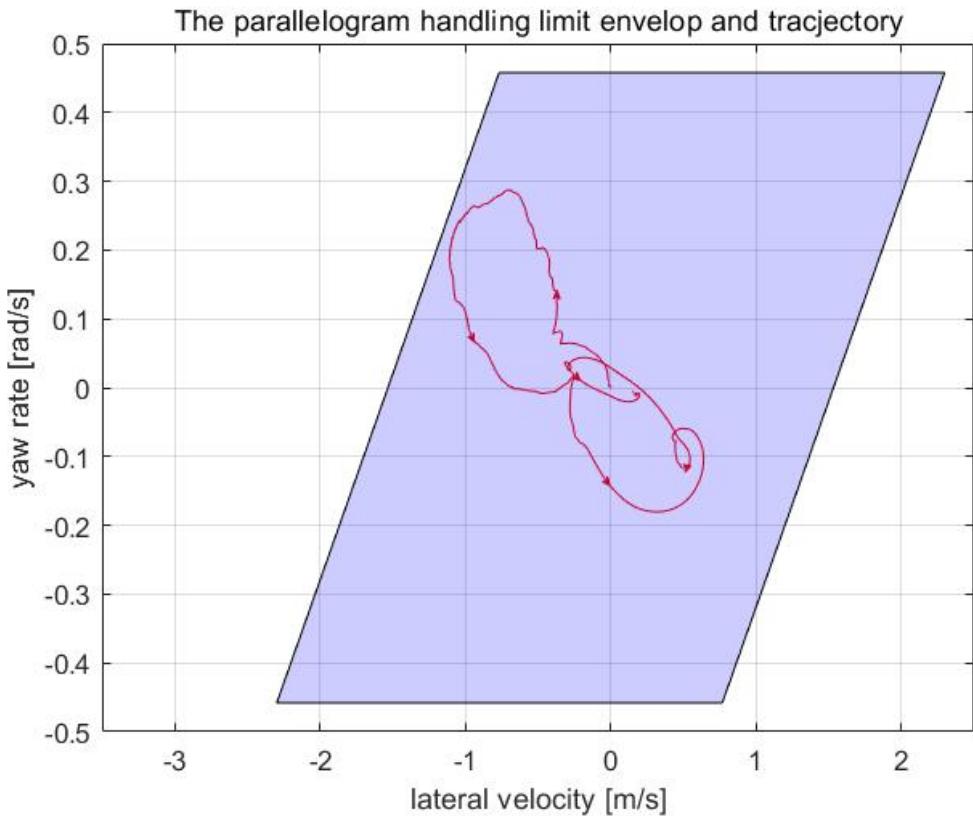


Figure 4.13: The vehicle trajectory in the phase plane of lateral velocity and yaw rate of the fixed time step controller (Section 4.1. A)

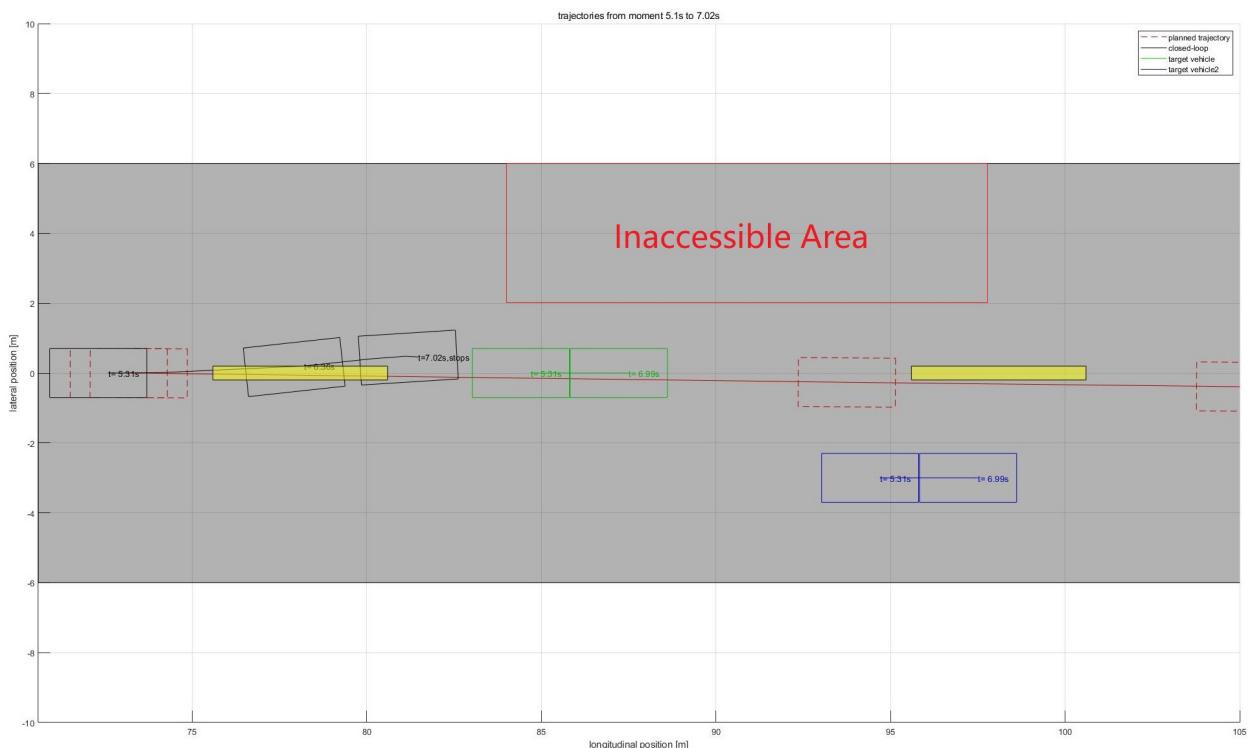


Figure 4.14: Trajectories of vehicles from the moment 5.1s to 7.02s (Section 4.1. B)

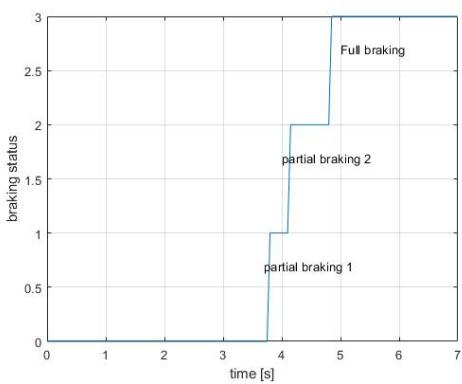


Figure 4.15: Braking status from 0s to 7.02s (Section 4.1, B)

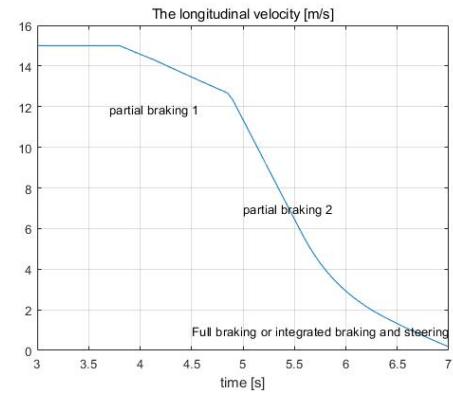


Figure 4.16: Longitudinal velocity from 0s to 7.02s (Section 4.1, B)

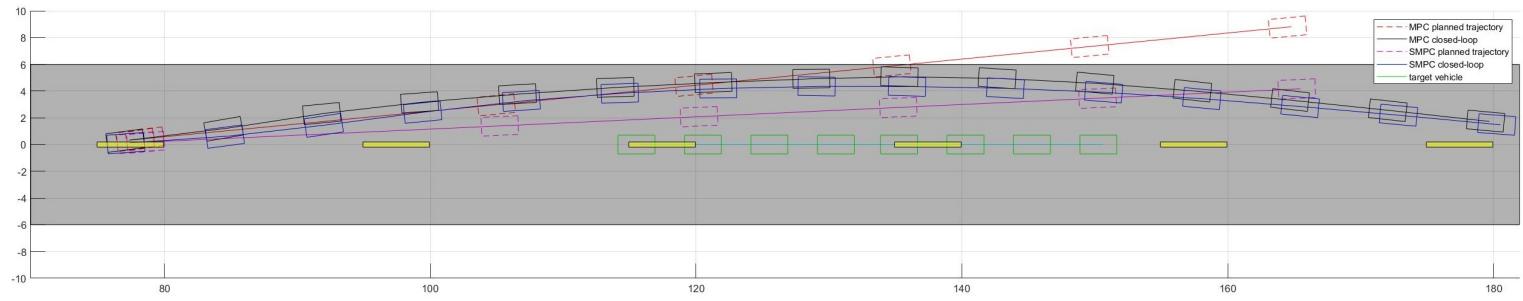


Figure 4.17: The figure shows the comparison of trajectories between MPC and SMPC controller when the risk parameter ε is 0.05 (Section 4.2, A)

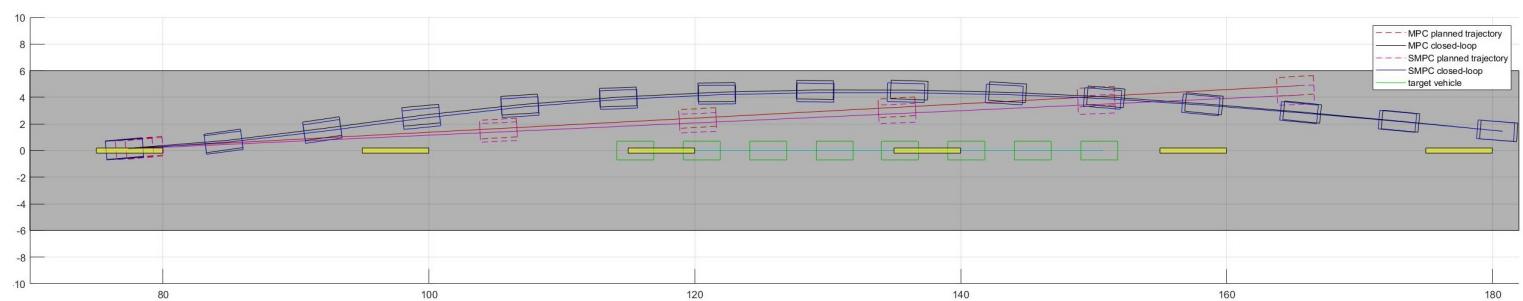


Figure 4.18: The figure shows the comparison of trajectories between MPC and SMPC controller when the risk parameter ε is 0.3 (Section 4.2, A)

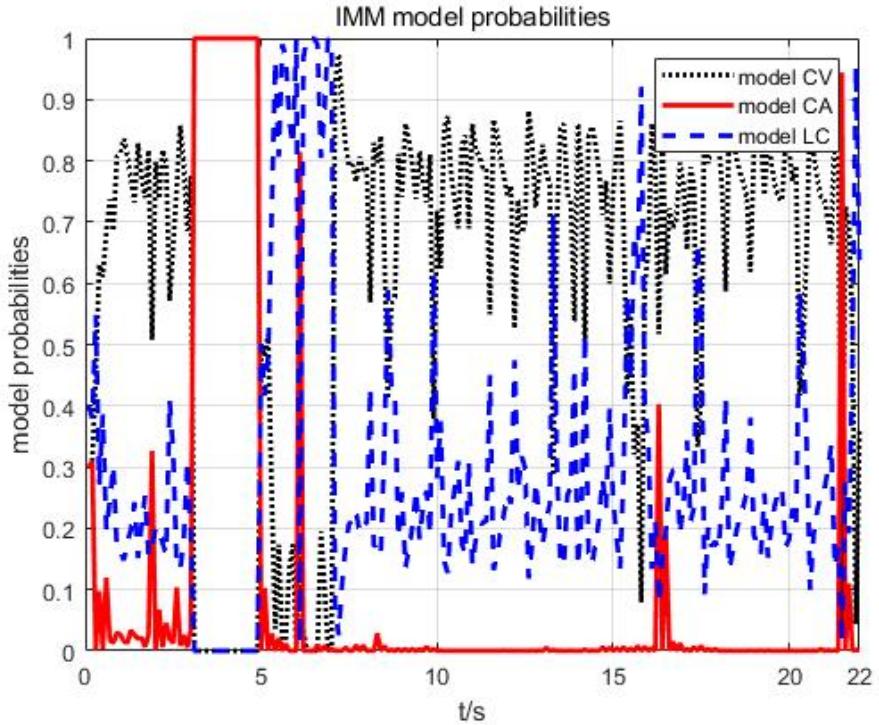


Figure 4.19: The model probabilities of IMMKF in 22s (Section 4.2. B)

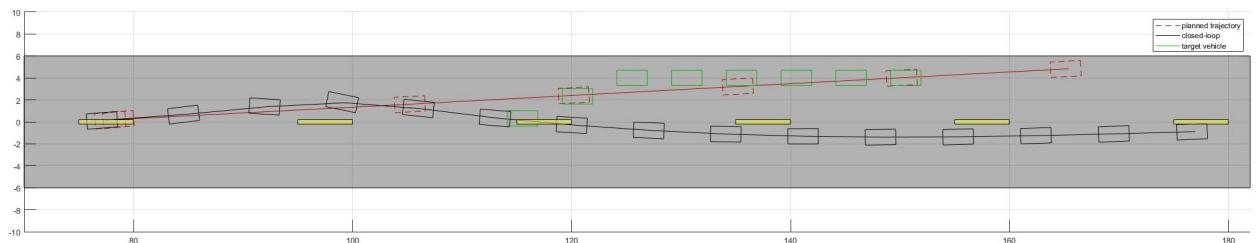


Figure 4.20: The planned trajectory and trajectory of the closed-loop at time 5.1s (Section 4.2. B)



Figure 4.21: The planned trajectory and trajectory of the closed-loop at time 6.6s (Section 4.2. B)

5 Hardware in the Loop Results

In the previous chapters, the related theories of the proposed system are well introduced and later implemented and evaluated in simulations. However, the effectiveness of a controller is always required to be tested in real-time embedded systems. Therefore, in this chapter, a HiL platform is designed and later used for HiL tests.

5.1 HiL platform setup

As Figure 5.1 shows, the designed HiL platform can be mainly divided into three parts, the communication center, the real controller and the virtual controlled object. Robot Operating System is an open-source robotics middleware suite, which can be basically considered as a collection of software frameworks for robot software development. It is used in this platform because it can work as a communication center and provides links to the center for each part of the platform. The real controller is written on Nvidia Jetson Nano in c++ which execute control signals once in every 0.05s. Matlab/Simulink does help a bit in the process of communication while it also function as a part of the virtual controlled object. Since the control-oriented vehicle dynamic model is a bicycle model with the Brush tire model and it was tested by studying the states' evolution of the model itself, a higher fidelity vehicle dynamic model is needed. CarSim is a commercial software package that predicts the performance of vehicles in response to controls in a given environment which contains lots of high fidelity vehicle models. An 'E-class SUV' with specifications in Table 4.1 and a more specified tire model which can be found in Appendix from CarSim is used to play the role of controlled object in HiL process. The inputs from controller to the vehicle are brake forces, drive torques and steering angle. Therefore, based on the inputs calculated by the controller, these decisions are converted into those inputs that can be directly fed into the vehicle model.

It should be noted that those HiL platforms in research level like Speedgoat and dSPACE simulates the virtual controlled objects in real time interacting with the real hardware in the loop. The designed HiL platform in this work does not work in real time but works much slower than the real time which brings doubts on the persuasiveness of the HiL test results. However, by setting up a function call to the controller, the controller can solve the optimization problem once it receives updated states and other information from other parts of the platform and wait until those information is updated in next period. The simulation of the virtual controlled object is slower but can be adapted to.

5.2 HiL Results

The scenario 1 in Section 4.1 is assigned to the HiL test. As Figure 5.2 5.3 5.4 show, a double-lane road environment is established where in the begining the controlled vehicle (white) is following the lead vehicle (blue) at the same velocity of 54 kph. When the lead vehicle decelerates, the controlled vehicle chooses to overtake since the left lane is clear. Intuitively, HiL results agree with the results of simulations.

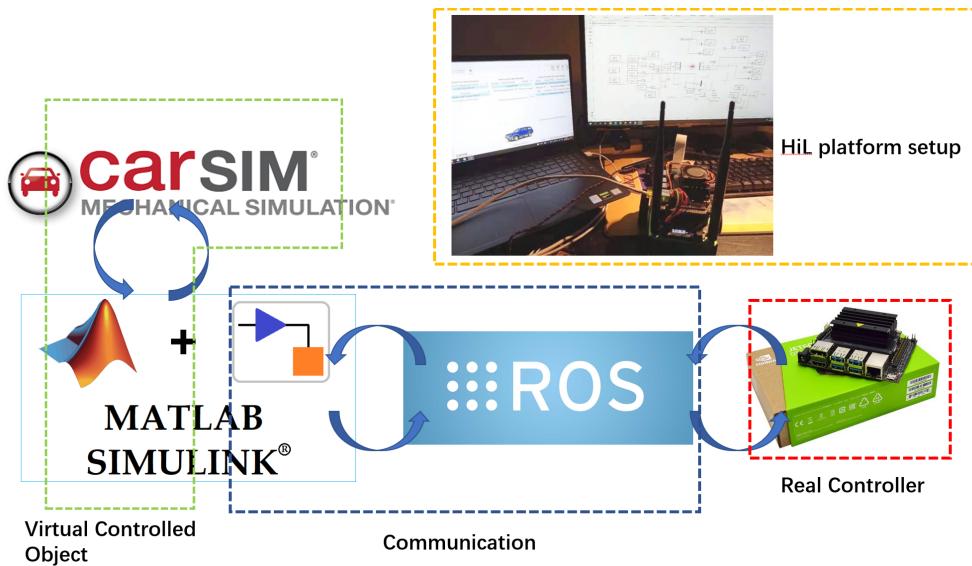


Figure 5.1: The structure of the designed HiL platform and the real setups

Figure 5.5 to 5.8 show that the drive or brake force required in each wheel during the whole process. It should be noted that the rear wheels barely actuate drive or brake forces in order to avoid the possibility of oversteering. The Figure 5.9 shows the steering angle that the controller feeds into the controlled vehicle model. As it could be noted, there is a sharp leap in the steering angle at moments before 5s but it does not result in severe problems since it only lasts for a short time. The connection between the real controller and the virtual controlled object can be done mainly in two methods, via WIFI or Ethernet connection. A comparison between two methods is shown in Table 5.1. Ethernet connection is later chosen for HiL tests due to its reliable performance. In general, the HiL tests on scenario 1 validate the design of the system and the results of simulations.

In this work, other scenarios will not be discussed here since HiL tests' main goal is to study if the controller embedded on a real embedded cpu like Nvidia Jetson Nano is able to solve the optimization problems within a short time in real time. And also, studying the effect of communication between different parts on the performance of the controller is also required.

	Mean ping	Package loss rate
WIFI Connection	93.25ms	33.3%
Ethernet Connection	1ms	0

Table 5.1: The comparison of communication qualities between two connection methods

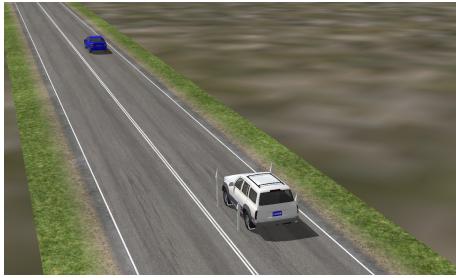


Figure 5.2: The controlled vehicle is following the lead vehicle in the beginning

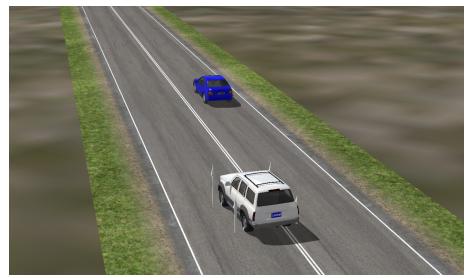


Figure 5.3: The lead vehicle brakes and the controlled vehicle starts steering

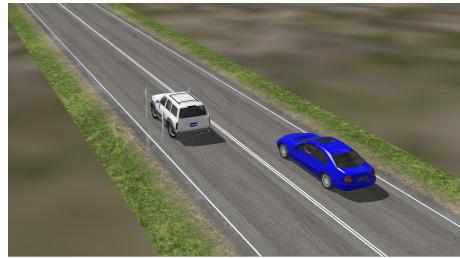


Figure 5.4: The controlled vehicle overtakes the lead vehicle

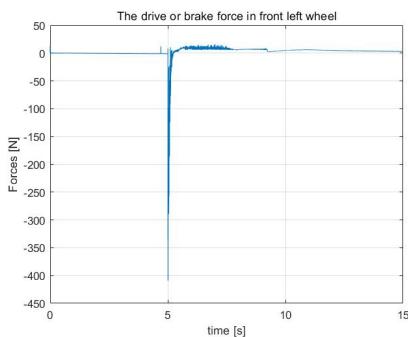


Figure 5.5: The drive or brake force in front left wheel

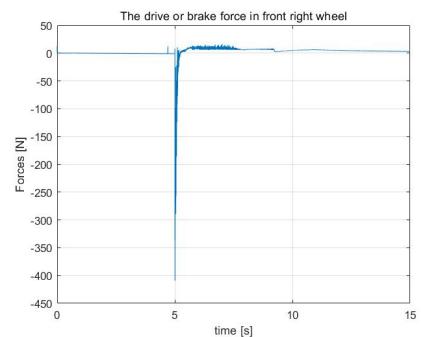


Figure 5.6: The drive or brake force in front right wheel

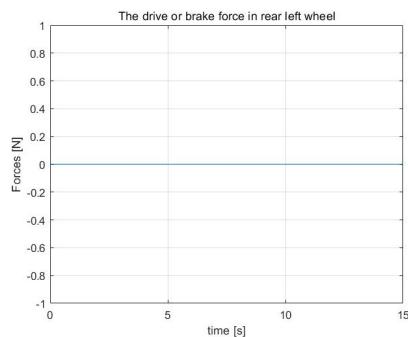


Figure 5.7: The drive or brake force in front right wheel

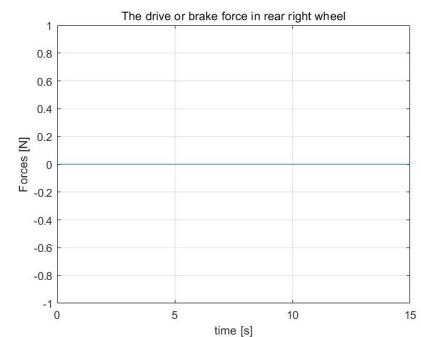


Figure 5.8: The drive or brake force in rear right wheel

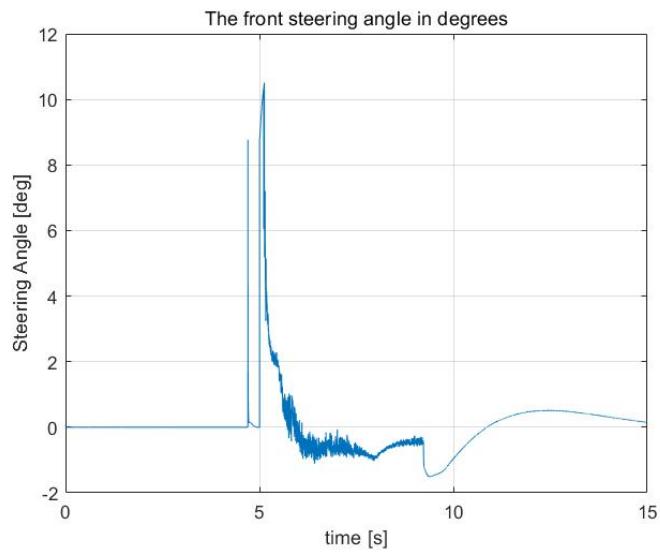


Figure 5.9: The steering angle in degrees

6 Discussion and Conclusion

6.1 Discussion

A. Classical MPC

The controller based on the classical MPC algorithm is presented in Section 3.2.2. It should be noted that the formulation of this controller is based on some assumptions and simplifications and there is also other alternative ways to be formulated.

Firstly, the controller solves the optimization problem to obtain a sequence of optimal control inputs where the solver is the active-set of **quadprog** (quadratic program)@Matlab. However, there are other methods to solve quadratic program problems like 'interior-point-convex', 'trust-region-reflective'. Moreover, in Matlab there are other functions like 'mpcActiveSetSolver' that may differ in solution efficiency, restrictive conditions from **quadprog**.

Secondly, the elastic length envelop 3.20 is constructed under simplifications. It should be noted that the RHS of the equation represents the longitudinal (X relative to the ground) position where the vehicle is allowed to drive into. But the LHS represents the distance that vehicle is predicted to travel in longitudinal (x relative to the CG of the vehicle) position. Therefore, two sides of the equation do not really represent the same thing, which leads to degrading the performance of the controller. In theory, the LHS of the equation should be multiplied by $\cos(\theta)$ (θ is the yaw angle). However, since θ is one of the states the constraint would become nonlinear constraint if $\cos(\theta)$ is taken into account, resulting in that **quadprog** is no more available for the type of optimization problem solving.

Lastly, the controller calculates the lateral forces needed in front tires F_{yf} which is later mapped into the steering angle. One question is once the steering angle is calculated what values should be used in each of two front tires. In reality, the steering angles of each front tires are different in order to steer the vehicle with stability. Since the vehicle model in Section 3.1.2 is not modelled into the drive-line level, how the steering angle in each tire is distributed is unknown. Therefore, in this work, the equal steering angle is used, which may lead into infeasibility on real vehicles.

B. Stochastic MPC

In Section 3.3.2, the vehicle subject to Gaussian noise is studied so that a Stochastic MPC with chance-constraints is formulated. The Gaussian noise is required to be recorded in physical tests on real vehicles which in this work it is impossible. Therefore, in this work the Gaussian noise is modelled out of intuition that is not scientific rigorous. Moreover, in reality vehicle are probably not subject to Gaussian noise. It could be noise that obeys any other distributions. In this regard, the SMPC formulated in Section 3.3.2 could fail to do the job in many cases. To deal with other types of noise, other methods of SMPC and Robust MPC could be used accordingly.

Additionally, in Section 3.3.2 the control strategy is open loop control that computes control sequence as a result of the optimization problem and is independent of the noise. Therefore,

it is deterministically defined as a function of the current state so that the variance of the uncertain component of the states evolves in an uncontrolled fashion. It has many drawbacks especially when it is assigned to deal with an unstable system. Actually, another control strategy, state feedback control is studied in this work. The evolution equations for both the deterministic and uncertain components of the states are shown as follows.

$$\mathbf{z}_{k+1} = \Phi_k \mathbf{z}_k + \mathbf{B}_k \mathbf{c}_k \quad (6.1)$$

$$\mathbf{e}_{k+1} = \Phi_k \mathbf{e}_k + \mathbf{G}_k \mathbf{w}_k \quad (6.2)$$

where $\Phi_k = A_k + B_k K_k, K_k$ is the stabilizing feedback gain for the pair (A_k, B_k) .

It is beautiful in theory but in practice the solution time for computing the stabilizing feedback gain K_k takes too much time in each run, making it inappropriate for implementation.

C. Rationality of Models and Assumptions

In Section 3.2.2 the Brush tyre model is used and the friction coefficient between tyres and the road is set to be 0.7. The problem is those assumptions may not be reasonable enough under other circumstances. For instance, the vehicle drives in a rainy day on a wet road which makes the friction coefficient smaller. Same tyres that work in different conditions (i.e. One in the far north another in south, temperature may differ a lot). These differences are not taken into consideration in our work.

D. Novelty Compared to Others' Work

Reza Hajiloo et al. in [1] presents an integrated lateral controller that computes front forces and the additional yaw moment generated by longitudinal forces while uses a proportional controller controlling the longitudinal dynamics. Two controllers work separately, making it impossible to activate both AEB and AES to avoid collision. However, in my work the two controllers are combined so it can work on either AEB mode or AES mode or even on mixed mode. Qingjia Cui et al. in [10] presents an algorithm to decide either to steer or brake in emergency which is also used in my work. However, the way Cui et al. chooses to steer the vehicle is to guide the vehicle to move in an arc with certain radius which is not consistent with most cases in reality. Moreover, in some cases both AEB and AES can be used together to avoid collision but Cui et al. didn't study the possibility of doing so.

6.2 Conclusion

This thesis has presented an Model Predictive Control design and a Stochastic Model Predictive Control design for an integrated emergency braking and steering automated driving assistance system. The integrated controller uses a combination of front steering, differential braking and classical Autonomous Emergency Braking to avoid collision and follow reference trajectories with stability ensured. In order to avoid safety constraints violation due to uncertainties of the vehicle model and sensor measurements, chance constraints which is based on the assumption that the process and measurement noise of the vehicle model obeys Gaussian distribution are constructed in a SMPC problem. A given risk parameter ε adjusts the trade-off between conservativeness and aggression of controller's behaviors. An Interacting Multiple Model Kalman Filter is used to track other vehicles and estimate which type of maneuvers other vehicles are on. Then it helps the controller builds up the

environment constraints for the MPC problem. The Chapter 3 presents this part of the work.

In Chapter 4 three different scenarios are studied to test the performance of the presented controller. The planned trajectories, the trajectories of the vehicle in closed control loop and target vehicles are plotted at some critical moments from which features of the controllers can be understood in a better manner.

In Chapter 5 an Hardware in the Loop platform is set up with a high-fidelity vehicle model from CarSim, an embedded controller on Nvidia Jetson Nano and Matlab/Simulink and Robot Operating System as communication tools. The first scenario mentioned in Chapter 4 is tested on HiL platform which turns out the controller is able to do the job on real ECU.

7 Future Work

1. HiL should be done on better platforms (i.e. Speedgoat, dSPACE). As it is mentioned in Chapter 5, the self-made HiL platform cannot compute the behaviors of the vehicle model in real time. It computes slower than the real time so that the real controller waits until it finishes each evolution of the vehicle states. Therefore, there could be some unknown problems not been discovered by the self-made HiL platform.
2. The chance constraints for the stochastic mpc problems should be formulated in a more accurate manner. As mentioned in Chapter 6.2, the Gaussian distribution is written by intuition which could not be the case in many situations. Instead, the mean value and standard deviation of the Gaussian distribution should be measured and constructs certain Gaussian distribution which is fed to building up the chance constraints.
3. Physical tests should be conducted to evaluate the design of the controllers. Physical tests can be done in real cars or four-wheel robot cars. The effect of vehicle parts in drive line level on the performance of controllers can be only tested on physical tests.
4. Faults coming out of the perception module should be understood by the algorithms of the controller so that controllers can be fault-tolerant and perception-aware. A certain algorithm for estimating the reliability of sensors after sensor fusion should be constructed to adjust the parameters of the controller.
5. The controller guides the vehicle follow the reference trajectory without violating constraints and with lower costs. However, the reference trajectory is required to be generated before the controller makes a decision. Therefore, a path-planning module should be studied and formulated to empower the controller.

8 Appendix

A Gantt Chart

The time plan for this thesis project is presented in a form of GANTT chart in figure below.

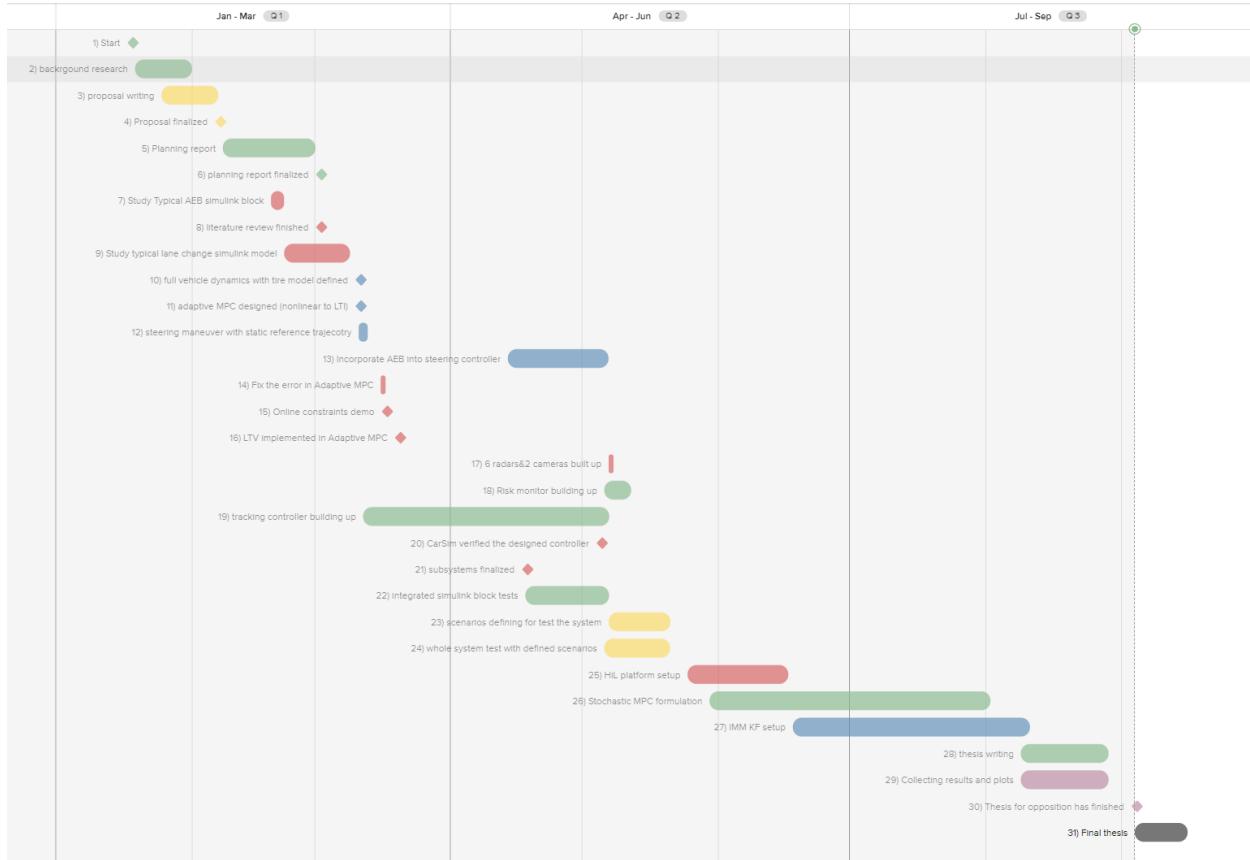


Figure 8.1: GANTT chart for time plan

B Risk Analysis

To evaluate different risk factors to the project, a risk analysis was performed. The severity and likelihood for risk was ranked on the scale 1-3-9. The risk factor is the product of the two. It is considered as low risk (green) when on the scale 1-3, medium risk (yellow) when on the scale 4-6 and high risk (red) when above 6. The risks and their preventive action and action plan in case it does happen are presented in Table 8.1 below.

C Matrix Formulation for QP

(1). Matrix Φ

$$\mathbf{U} = \mathbf{U}_t + \Phi \Delta \mathbf{U} \quad (8.1)$$

where

$$\mathbf{U} = \begin{bmatrix} U(k+1|t) \\ U(k+2|t) \\ \vdots \\ U(k+N_c|t) \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{[N_c \times N_c]} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Note that \otimes is **Kronecker product**.

(2). Matrix for control constraints

$$\begin{bmatrix} A_{ieq1} & \mathbf{0}_{[N_c N_u \times 3]} \\ A_{ieq1} & \mathbf{0}_{[N_c N_u \times 3]} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{U}(t) \\ \epsilon_s \\ \epsilon_e \\ \epsilon_x \end{bmatrix} \leq \begin{bmatrix} \mathbf{U}_{max}^+ - \mathbf{U}_T^+ \\ \mathbf{U}_{min}^+ + \mathbf{U}_T^+ \end{bmatrix}$$

where

$$\mathbf{U}_{min}^+ = \begin{bmatrix} \mathbf{U}_{min} \\ \mathbf{U}_{min} \\ \vdots \\ \mathbf{U}_{min} \end{bmatrix}_{[N_c]}, \quad \mathbf{U}_T^+ = \begin{bmatrix} \mathbf{U}_T \\ \mathbf{U}_T \\ \vdots \\ \mathbf{U}_T \end{bmatrix}_{[N_c]}$$

(3). Matrix for state constraints

$$\Gamma_i^+ \mathbf{X}(t) \leq \mathbf{G}_i + \epsilon_i$$

where

$$\Gamma_i^+ = \begin{bmatrix} \Gamma_i & 0 & \cdots & 0 \\ 0 & \Gamma_i & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \Gamma_i \end{bmatrix}_{[N_p \times N_p]} \quad i = s, e, x$$

D Kalman Filter Algorithm 1

Algorithm 1 Kalman Filter equations for mode(i) [8]

Input: $(\bar{\xi}_{k-1|k-1}^{(i)}, \bar{\Sigma}_{k-1|k-1}^{(i)})$

Prediction step:

$$\begin{aligned}\hat{\xi}_{k|k-1}^{(i)} &= F_{k-1}^{(i)} \bar{\xi}_{k-1|k-1}^{(i)} + E_{k-1}^{(i)} \\ \hat{\Sigma}_{k|k-1}^{(i)} &= F_{k-1}^{(i)} \bar{\Sigma}_{k-1|k-1}^{(i)} + G_{k-1}^{(i)} Q_{k-1}^{(i)} G_{k-1}^{(i)T}\end{aligned}$$

Update step:

$$\begin{aligned}\hat{y}_k^{(i)} &= y_k - H_k^{(i)} \hat{\xi}_{k|k-1}^{(i)} \\ S_k^{(i)} &= H_k^{(i)} \hat{\Sigma}_{k|k-1}^{(i)} H_k^{(i)T} + R_k^{(i)} \\ K_k^{(i)} &= \hat{\Sigma}_{k|k-1}^{(i)} H_k^{(i)T} S_k^{(i)-1} \\ \hat{\xi}_{k|k}^{(i)} &= \hat{\xi}_{k|k-1}^{(i)} + K_k^{(i)} \hat{y}_k^{(i)} \\ \hat{\Sigma}_{k|k}^{(i)} &= \hat{\Sigma}_{k|k-1}^{(i)} - K_k^{(i)} S_k^{(i)} K_k^{(i)T}\end{aligned}$$

Output: $(\hat{\xi}_{k|k}^{(i)}, \hat{\Sigma}_{k|k}^{(i)}, \hat{y}_k^{(i)}, S_k^{(i)})$

E Tire Model Specifications

The tire model is '265/75 R16(2017)' tire model from CarSim. Specifications are shown in the following figures and table.

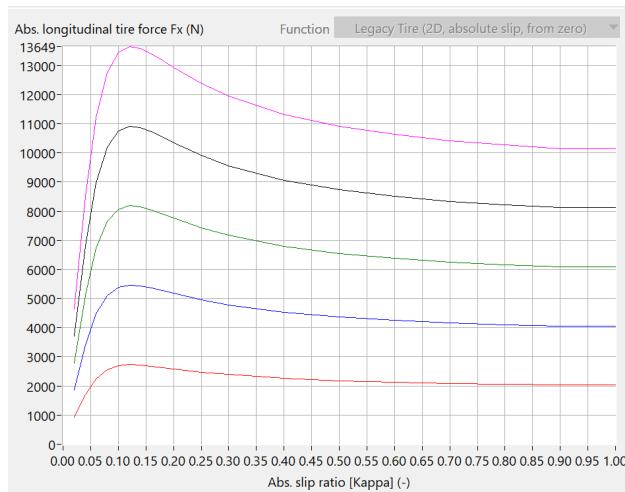


Figure 8.2: Longitudinal force curves

F Code for SMPC algorithm

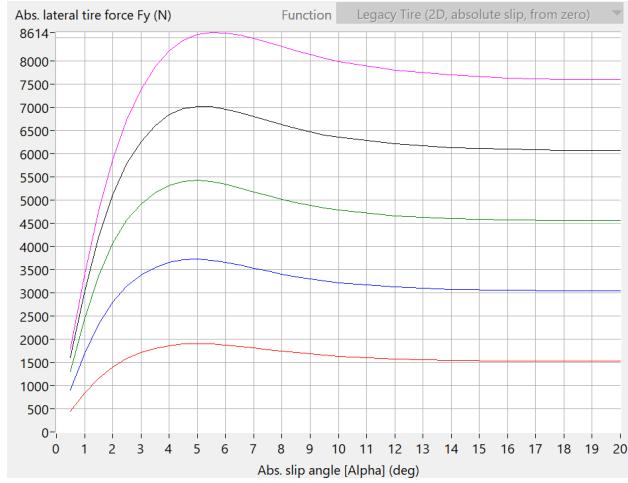


Figure 8.3: Lateral force curves

```

1 function [U1_out,U2_out,U3_out,Useq_,fval,exitflag,dt] = ...
2     fcn(Vxr,Vyr,Yawr,Rater,Yr, ...
3         Fxr,Fyr,Mxr,Vxc,Vyc,Yawc,Ratec,Yc,Ulast1,Ulast2,Ulast3,ks,Useq,emax, ...
4         emin,Gx,epsilon)
5 %Vyr lateral velocity, Yawr yaw angle, Rater yaw rate, Yr lateral position
6 %Fyr lateral force in front tyre,Mxr moment generated by longitudinal force
7 %'r'means reference || 'c'means current
8
9     tstart=tic;%timing on
10
11    if Vxc<0.2
12        Vxc=0.2;%make sure many expressions' denominator isn't 0
13    end
14    lr=1.67;
15    threshold=0.1022;
16    miu=0.7;
17    rmax=9.81*miu/Vxc;
18 %% Initialization
19    Nx=5;%number of state
20    Nu=3;%number of control
21    Np=70;%prediction horizon
22    %N1 cannot be equal to N2
23    Nc=6;%control horizon
24    N1=40;
25    N2=10;
26    %      N3=20;
27
28    %reference
29    r=zeros(Nx,1);%state reference
30    r(1)=Vxr;
31    r(2)=Vyr;
32    r(3)=Yawr;
33    r(4)=Rater;
34    r(5)=Yr;
35    ur1=Fxr;
36    ur2=Fyr;
37    ur3=Mxr;%control reference
38
39    %states
40    x=zeros(Nx,1);%state reference
41    x(1)=Vxc;

```

```

40     x(2)=Vyc;
41     x(3)=Yawc;
42     x(4)=Ratec;
43     x(5)=Yc;
44
45 %control
46 u=zeros(Nu,1);%state reference
47 u(1)=Ulast1;
48 u(2)=Ulast2;
49 u(3)=Ulast3;
50
51 %get augmented states
52 kesi=zeros(Nx+Nu,1); %[X;U]
53 kesi(1)=x(1)-r(1);
54 kesi(2)=x(2)-r(2);
55 kesi(3)=x(3)-r(3);
56 kesi(4)=x(4)-r(4);
57 kesi(5)=x(5)-r(5);
58 kesi(6)=u(1)-url1;
59 kesi(7)=u(2)-ur2;
60 kesi(8)=u(3)-ur3;
61
62 Ts1=0.05;%sample time
63 Ts2=0.2;
64
65 %% Modelling->Discretization->prediction->quadratic programming
66 %Modelling&Linearization&Discretization
67 steps=Np+1;
68 A = zeros(Nx+Nu,Nx+Nu,steps);
69 B = zeros(Nx+Nu,Nu*2,steps);
70 Dc = zeros(Nx+Nu,1,steps);
71 C=[1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 1 0 0 0 ...
    0;0 0 0 0 1 0 0 0];
72
73 for ct = 1:steps
74     uk = Useq(ct,:);
75     if ct == steps
76         uk1 = Useq(ct,:);
77     else
78         uk1= Useq(ct+1,:);
79     end
80     uk=uk';
81     uk1=uk1';
82     if ct≤N1 %in short prediction region, using zero order hold
83         Ts=Ts1;
84         [Ad, Bd, DK, Δ_x] = GetDynamicsZOH(ks,x,uk,Ts);
85         A(:,:,ct)=[Ad,Bd;zeros(Nu,Nx),eye(Nu)];
86         B(:,:,ct)=[Bd zeros(Nx,Nu);eye(Nu) zeros(Nu,Nu)];%in order ...
            to fit the dimension of B from ZOH with B from FOH, ...
            augmentation is done here
87         Dc(:,:,ct)=[DK;zeros(Nu,1)];%constant component
88     elseif ct≤N1+N2 %in correction prediction region, using first ...
        order hold
89         Ts=Ts1+(Ts2-Ts1)*(ct-N1)/(N2-N1);%get this step's ...
            correction sample time
90         [Ad, Bd1,Bd2, DK, Δ_x] = GetDynamicsFOH(ks,x,uk,uk1,Ts);
91         A(:,:,ct)=[Ad Bd1+Bd2;zeros(Nu,Nx) eye(Nu)];
92         B(:,:,ct)=[Bd1+Bd2 Bd2;eye(Nu) zeros(Nu,Nu)];
93         Dc(:,:,ct)=[DK;zeros(Nu,1)];
94     else %in long prediction region, using first order hold

```

```

95 Ts=Ts2;
96 [Ad, Bd1,Bd2, DK, Δ_x] = GetDynamicsFOH(ks,x,uk,uk1,Ts);
97 A(:,:,ct)=[Ad Bd1+Bd2;zeros(Nu,Nx) eye(Nu)];
98 B(:,:,ct)=[Bd1+Bd2 Bd2;eye(Nu) zeros(Nu,Nu)];
99 Dc(:,:,ct)=[DK;zeros(Nu,1)];
100 end
101 x=x+Δ_x; %predict next step's states
102 end
103 %construct important matrix
104 %Y=PHI*kesi+THETA*Δ_U+OMEGA*D
105 coder.varsize('PHI',[Np*Nx,Nx+Nu]);
106 coder.varsize('THETA',[Np*Nx,Nc*Nu]);
107 coder.varsize('D',[Np*(Nx+Nu),1]);
108 coder.varsize('DD',[Np*Nx,1]);
109 coder.varsize('OMEGA',[Nx*Np,(Nx+Nu)*Np]);
110 OMEGA=zeros(Nx*Np,(Nx+Nu)*Np);
111 PHI=zeros(Np*Nx,Nx+Nu);
112 THETA=zeros(Np*Nx,Nc*Nu);
113 D=zeros(Np*(Nx+Nu),1);
114 DD=zeros(Np*Nx,1);
115 product=eye(Nx+Nu);
116 for j=1:Np
117 PHI((j-1)*Nx+1:j*Nx,:)=C*A(:,:,j)*product;
118 product=A(:,:,j)*product;
119 D((j-1)*(Nx+Nu)+1:j*(Nx+Nu),:)=Dc(:,:,j);
120 end
121 for k=1:Nc/2
122 product=eye(Nx+Nu);
123 for j=k:Np
124 THETA((j-1)*Nx+1:j*Nx,(k-1)*Nu*2+1:k*Nu*2)=C*product*B(:,:,k);
125 product=A(:,:,j+1)*product;
126 end
127 end
128 for i=1:Np
129 product=eye(Nx+Nu);
130 for j=i:Np
131 OMEGA((j-1)*Nx+1:j*Nx,(i-1)*(Nx+Nu)+1:i*(Nx+Nu))=C*product;
132 product=A(:,:,j+1)*product;
133 end
134 end
135 DD=OMEGA*D;
136 Sigma = tril(ones(Nc,Nc));
137 Sigma = kron(Sigma,eye(Nu));
138 UT=kron(ones(Nc,1),[u(1);u(2);u(3)]);
139 error=PHI*kesi;
140 Q=kron(eye(Np),diag([4 0 0 100 4]));
141 R=kron(eye(Nc),diag([4e-5 0 4e-5]));
142 P=kron(eye(Nc),diag([0 3e-5 8e-5]));
143 rou1=3e3;
144 rou2=5e4;
145 rou3=1e6;
146 H=2*[THETA'*Q*THETA+Sigma'*R*Sigma+P zeros(Nc*Nu,3);zeros(1,Nc*Nu) ...
147 rou1 0 0;zeros(1,Nc*Nu) 0 rou2 0;zeros(1,Nc*Nu) 0 0 rou3];
148 f=[2*(error'*Q*THETA+(OMEGA*D)'*Q*THETA+UT'*R*Sigma),0,0,0];
149 %% constraints
150 %% inequation
151 A_I = tril(ones(Nc,Nc));
152 A_I = kron(A_I,eye(Nu));
153 Ut=kron(ones(Nc,1),[u(1);u(2);u(3)]);%control in last control step
umin=[-6816; -1e4; -1e3];

```

```

154 umax=[6816; 1e4; 1e3];
155 Umin=kron(ones(Nc,1),umin);
156 Umax=kron(ones(Nc,1),umax);
157 coder.varsize('A_cons', [Np*6+Nc*Nu*2+Np,Nc*Nu+3]);
158 coder.varsize('b_cons', [Np*6+Nc*Nu*2+Np,1]);
159 %add handling envelop constraints & environment envelop &longitudinal
160 %position constraints
161 gamas=[0 1 -lr 0 0;0;-1 lr 0 0;0;0 0 1 0 0;0;0 0 -1 0 0];
162 gamae=[0 0 0 1;0 0 0 0 -1];
163 gamax=[1 0 0 0];
164 Gs=[Vxc*threshold;Vxc*threshold;rmax;rmax];
165 Gamas = kron(eye(Np),gamas);
166 Gamae = kron(eye(Np),gamae);
167 Gsm = kron(ones(Np,1),Gs);
168 A_cons0=[A_I zeros(Nc*Nu,3);-A_I zeros(Nc*Nu,3)];
169 b_cons0=[Umax-Ut;-Umin+Ut];
170 A_cons1 = [Gamas*THETA -ones(4*Np,1) zeros(4*Np,1) zeros(4*Np,1)];
171 b_cons1 = Gsm - Gamas*(PHI*kesi+DD);
172 A_cons2 = [Gamae*THETA zeros(2*Np,1) -ones(2*Np,1) zeros(2*Np,1)];
173 Gse = zeros(2*Np,1);
174 Qw = blkdiag(0.6^2, 0.6^2); %x(k+1)=A*x(k)+B*u(k)+D(k)+G*w
175 G = [0.1 0; 0 0; 0 0; 0 0;0 0.1];
176 var = zeros(Nx,Nx);
177 [emax, emin, ~] = Chance_constraints(A, G, Qw, Np, emax, emin, var, ...
    epsilon);
178 for i=1:Np
179     Gse(2*(i-1)+1,:)=emax(i);
180     Gse(2*(i-1)+2,:)=-emin(i);
181 end
182 b_cons2 = Gse - Gamae*(PHI*kesi+DD);
183 A_cons = [A_cons0;A_cons1;A_cons2];
184 b_cons = [b_cons0;b_cons1;b_cons2];
185 if Vxr==0.1
186     Gamax = kron(eye(Np),gamax);
187     Gamax = tril(ones(Np,Np))*Gamax;
188     A_cons3 = [Ts*Gamax*THETA zeros(Np,1) zeros(Np,1) -ones(Np,1)];
189     b_cons3 = Gx-Ts*Gamax*(PHI*kesi+DD);
190     A_cons = [A_cons;A_cons3];
191     b_cons = [b_cons;b_cons3];
192 end
193
194 %upper and lower boundary
195 delta_umin=[-1e4; -1e4; -1e3];
196 delta_umax=[1e4; 1e4; 1e3];
197 M1=10; %slack variable related to handling limit constraints
198 M2=10; %slack variable related to enviroment constraints
199 M3=10; %slack variable related to longitudinal position constraints
200 delta_Umin=kron(ones(Nc,1),delta_umin);
201 delta_Umax=kron(ones(Nc,1),delta_umax);
202 lb=[delta_Umin;0;0;0];
203 ub=[delta_Umax;M1;M2;M3];
204 %% solve quadratic programming
205 opts = optimoptions('quadprog','Algorithm','active-set');
206 x_start=zeros(Nu*Nc+3,1);%+3 corresponds to slack variables
207 [X,fval,exitflag]=quadprog(H,f',A_cons,b_cons,[],[],lb,ub,x_start,opts);
208
209 %% calculate control signal
210 U1_out=kesi(6)+X(1)+url;%Error plus reference value to get final ...
    output value
211 U2_out=kesi(7)+X(2)+ur2;

```

```

212     U3_out=kesi(8)+X(3)+ur3;
213     U = [kesi(6)+url, kesi(7)+ur2, kesi(8)+ur3];
214     Useq_ = zeros(Np+1,Nu);
215     Xsum = zeros(1,Nu);
216     for i=1:Nc
217         Xsum = Xsum+[X(3*(i-1)+1) X(3*(i-1)+2) X(3*(i-1)+3)];
218         Useq_(i,:)= U + Xsum;
219     end
220     Δ_t=toc(tstart);%timing off
221 end
222
223 function [A, B, DK, Δ_x] = GetDynamicsZOH(ks,x,u,Ts)
224
225 %Plant Model parameters
226 m = 2272; %mass
227 Iz = 4600; %rotational inertia
228 lf = 1.11; % distance from front axle to the mass center
229 lr = 1.67; % distance from rear axle to the mass center
230 C_a = 1.822e5; % rear cornering stiffness
231
232 %Enviroment parameters
233 miu = 0.7; %frction cooeficient 0.7 for dry road and 0.4 for wet
234 g = 9.81; % gravity
235
236 Fz = m*g*lf/(lf+lr); % the load that beared by the rear tire
237
238 %calculate alpha
239 Vx = x(1); %longitudinal velocity
240 Vy = x(2); %lateral velocity
241 r = x(3); %yaw rate
242 alphar_ = Vy/Vx-lr*r/Vx; %calculate linearizing point
243
244 %tire model
245 threshold = atan(3*miu*Fz/C_a); % threshold for the tire linear region
246 if abs(alphar_)<threshold
247     %the rear lateral force value (f(x0)) at the linearizing point
248     Fyr_ = ...
249         -C_a*tan(alphar_)+C_a^2/(3*miu*Fz)*abs(tan(alphar_))*tan(alphar_-
250             -C_a^3/(27*miu^2*Fz^2)*tan(alphar_)^3;
251
252     %the slope of linearized model (f'(x0))
253     if alphar_> 0
254         Cr_ = ...
255             -C_a*sec(alphar_)^2+C_a^2/(3*miu*Fz)*(2*sec(alphar_)^2*tan(alphar_-
256                 -C_a^3/(27*miu^2*Fz^2)*(3*tan(alphar_)^2*sec(alphar_)^2);
257     else
258         Cr_ = ...
259             -C_a*sec(alphar_)^2-C_a^2/(3*miu*Fz)*(2*sec(alphar_)^2*tan(alphar_-
260                 -C_a^3/(27*miu^2*Fz^2)*(3*tan(alphar_)^2*sec(alphar_)^2);
261     end
262 else
263     %the saturation region of tire model
264     Fyr_ = -miu*Fz*sign(alphar_);
265     Cr_ = 0;
266 end
267
268 %Continuous-time model
269 Ac = [0 r Vy 0 0;
270       -r+(Cr_**(lr*r-Vy)/(m*Vx^2)) Cr_/(m*Vx) -lr*Cr_/(m*Vx)-Vx 0 0];

```

```

269     lr*Cr_*(Vy-lr*r)/(Iz*Vx^2) -lr*Cr_/(Iz*Vx) lr^2*Cr_/(Iz*Vx) 0 0;
270     -ks 0 1 0 0;
271     0 1 0 Vx 0];
272 Bc = [1/m 0 0;
273     0 1/m 0;
274     0 lf/Iz 1/Iz;
275     0 0 0;
276     0 0 0];
277 dcc = [0;
278     (Fyr_-Cr_*alphar_)/m;
279     -lr*(Fyr_-Cr_*alphar_)/Iz;
280     -Vx*ks;
281     0];
282 % dcc = dcc-Ac(1:5,1)*Vx;%adjust the Vx's linearization point
283 G=[Ac Bc dcc;zeros(4,9)];
284 M=expm(G*Ts);
285 A=M(1:5,1:5);
286 B=M(1:5,6:8);
287 DK=M(1:5,9);
288 Delta_x=A*x+B*u+DK-x;
289 end
290
291 function [A, B1, B2, DK, Delta_x] = GetDynamicsFOH(ks,x,uk,uk1,Ts)
292
293 %Plant Model parameters
294 m = 2272; %mass
295 Iz = 4600; %rotational inertia
296 lf = 1.11; % distance from front axle to the mass center
297 lr = 1.67; % distance from rear axle to the mass center
298 C_a = 1.822e5; % rear cornering stiffness
299
300 %Enviroment parameters
301 miu = 0.7; %frction cooeficient 0.7 for dry road and 0.4 for wet
302 g = 9.81; % gravity
303
304 Fz = m*g*lf/(lf+lr); % the load that beared by the rear tire
305
306 %calculate alpha
307 Vx = x(1); %longitudinal velocity
308 Vy = x(2); %lateral velocity
309 r = x(3); %yaw rate
310 alphar_ = Vy/Vx-lr*r/Vx; %calculate linearizing point
311
312 %tire model
313 threshold = atan(3*miu*Fz/C_a); % threshold for the tire linear region
314 if abs(alphar_)<threshold
315     %the rear lateral force value (f(x0)) at the linearizing point
316     Fyr_ = ...
317         -C_a*tan(alphar_)+C_a^2/(3*miu*Fz)*abs(tan(alphar_))*tan(alphar_)
318         -C_a^3/(27*miu^2*Fz^2)*tan(alphar_)^3;
319
320     %the slope of linearized model (f'(x0))
321     if alphar_> 0
322         Cr_ = ...
323             -C_a*sec(alphar_)^2+C_a^2/(3*miu*Fz)*(2*sec(alphar_)^2*tan(alphar_))
324             -C_a^3/(27*miu^2*Fz^2)*(3*tan(alphar_)^2*sec(alphar_)^2);
325     else
326         Cr_ = ...
327             -C_a*sec(alphar_)^2-C_a^2/(3*miu*Fz)*(2*sec(alphar_)^2*tan(alphar_))
328             -C_a^3/(27*miu^2*Fz^2)*(3*tan(alphar_)^2*sec(alphar_)^2);

```

```

326     end
327 else
328
329     %the saturation region of tire model
330     Fyr_ = -miu*Fz*sign(alphar_);
331     Cr_ = 0;
332 end
333
334 %Continuous-time model
335 Ac = [0 r Vy 0 0;
336     -r+(Cr_* (lr*r-Vy) / (m*Vx^2)) Cr_/(m*Vx) -lr*Cr_/(m*Vx)-Vx 0 0;
337     lr*Cr_* (Vy-lr*r) / (Iz*Vx^2) -lr*Cr_/(Iz*Vx) lr^2*Cr_/(Iz*Vx) 0 0;
338     -ks 0 1 0 0;
339     0 1 0 Vx 0];
340 Bc = [1/m 0 0;
341     0 1/m 0;
342     0 lf/Iz 1/Iz;
343     0 0 0;
344     0 0 0];
345 dcc = [0;
346     (Fyr_-Cr_*alphar_)/m;
347     -lr*(Fyr_-Cr_*alphar_)/Iz;
348     -Vx*ks;
349     0];
350
351 LAMDA=[Ac Bc dcc zeros(5,4);zeros(4,9) 1/Ts*eye(4);zeros(4,13)];
352 M=expm(LAMDA*Ts);
353 A=M(1:5,1:5);
354 PHI1=M(1:5,6:9);
355 PHI2=M(1:5,10:13);
356 B1=(PHI1-PHI2)*[eye(3);zeros(1,3)];
357 B2=PHI2*[eye(3);zeros(1,3)];
358 DK=PHI1*[0;0;0;1];
359 Delta_x=A*x+B1*u_k+B2*u_k1+DK-x;
360 %%x(k+1)=A*x(k)+B1*u(k)=B2*u(k+1)+DK
361 end
362
363 function [nmax, nmin, numcc] = Chance_constraints(A, G, Qw, Np, emax, ...
364     emin, var, epsilon)
365 % ncc is the number of chance-constrained constraints
366 SIGMA = zeros(Np,1);
367 for i = 1:Np
368     var = A(1:5,1:5,i)*var*A(1:5,1:5,i)' + G*Qw*G'; %var is Sigma(k), ...
369     % variance of e(k)
370     SIGMA(i,1) = sqrt(var(5,5))';
371 end
372 nmax = zeros(Np, 1);
373 nmin = zeros(Np, 1);
374 numcc = 0;
375 for i = 1:Np
376     if (isnan(emax(i,:))) == 0
377         nmax(i,:) = -norminv(1-epsilon, -emax(i,1), ...
378             SIGMA(i,1)); %calculate the new bounds with given ...
379             % probability of invalidation
380             disp(['nmax: ', num2str(nmax(j,i))]);
381             numcc = numcc+1;
382     else
383         nmax(i,:) = NaN;
384     end

```

```
382 if (isnan(emin(i,:))) == 0
383     nmin(i,:) = norminv(1-epsilon, emin(i,1), ...
384         SIGMA(i,1));%calculate the new bounds with given ...
385         probability of invalidation
386         disp(['nmin: ', num2str(nmin(j,i))]);
387         numcc = numcc+1;
388     else
389         nmin(i,1) = NaN;
390     end
391 end
```

Risk	Severity	Likelihood	Risk	Prevention Action
Scope changed/adjusted	3	9	27	Weekly meeting with supervisors. Continuous adjustment of time plan and evaluation against initial project scope
Mismatch in project status with team member	3	3	9	Weekly meeting with team member and self-supervises each other to catch up the time plan
Loss of files or data	9	1	9	Store files on cloud as well as a portable hard drive. Sort files in order and ensure no mess
Calculation errors leading to consequent failures	3	3	9	Double-check the calculation every time and do not miss any possible irregular phenomena that may result from calculation errors
Software and hardware issues	1	6	6	Try to find solutions on related online forums and contact with supervisors and team member
Wrong assumptions	4	2	8	Stay in close contact with supervisors, ask for help if needed
Late realization of inappropriate assumptions	9	1	9	Take the proposal and planning report very serious and elaborate them. Do literature research well
Conflict of requirements between supervisors and examiner	3	1	3	Keep in contact with the examiner and ask his opinions on new changes of scope and contents if necessary
Physical experiments disagree with simulation results	1	3	3	Make sure the model used for simulation is realistic enough so it matches physical experiments
Thesis Crashes	1	3	3	As the project goes, parts of thesis are written. In the end, just need to compile them and finalize the thesis
Sick due to COVID-19	1	3	3	Keep social distance with others and wear mask, sanitize hands frequently when I am out

Table 8.1: Risk Analysis

Parameter	Value
Effective rolling radius	351mm
Unloaded radius	402mm
Spring rate	502N/mm
Maximum allowed force	100000N
Tire width	265mm

Table 8.2: Specifications of the controller parameters

Bibliography

- [1] R. Hajiloo, M. Abroshan, A. Khajepour, A. Kasaiezadeh, and S.-K. Chen, “Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3167–3178, 2021.
DOI: [10.1109/TITS.2020.2984210](https://doi.org/10.1109/TITS.2020.2984210).
- [2] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.
DOI: [10.1109/TCST.2007.894653](https://doi.org/10.1109/TCST.2007.894653).
- [3] C. Choi and Y. Kang, “Simultaneous braking and steering control method based on nonlinear model predictive control for emergency driving support,” *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 345–353, 2017.
- [4] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “A model predictive control approach for combined braking and steering in autonomous vehicles,” In *2007 Mediterranean Conference on Control Automation*, 2007, Pp. 1–6.
DOI: [10.1109/MED.2007.4433694](https://doi.org/10.1109/MED.2007.4433694).
- [5] “Integrated braking and steering model predictive control approach in autonomous vehicles,” *IFAC Proceedings Volumes*, vol. 40, no. 10, pp. 273–278, 2007, 5th IFAC Symposium on Advances in Automotive Control,
ISSN: 1474-6670.
DOI: <https://doi.org/10.3182/20070820-3-US-2918.00038>.
- [6] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, “Mpc-based yaw and lateral stabilisation via active front steering and braking,” *Vehicle System Dynamics*, vol. 46, no. S1, pp. 611–628, 2008.
- [7] M. Farina, L. Giulioni, and R. Scattolini, “Stochastic linear model predictive control with chance constraints – a review,” *Journal of Process Control*, vol. 44, pp. 53–67, 2016,
ISSN: 0959-1524.
DOI: <https://doi.org/10.1016/j.jprocont.2016.03.005>.
- [8] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, “Stochastic predictive control of autonomous vehicles in uncertain environments,” In *12th International Symposium on Advanced Vehicle Control*, 2014,
Pp. 712–719.

- [9] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouvelière, “Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 589–606, 2010.
DOI: [10.1109/TITS.2010.2046037](https://doi.org/10.1109/TITS.2010.2046037).
- [10] Q. Cui, R. Ding, X. Wu, and B. Zhou, “A new strategy for rear-end collision avoidance via autonomous steering and differential braking in highway driving,” *Vehicle system dynamics*, 2019.
- [11] G. Garimella, J. Funke, C. Wang, and M. Kobilarov, “Neural network modeling for steering control of an autonomous vehicle,” In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017,
Pp. 2609–2615.
DOI: [10.1109/IROS.2017.8206084](https://doi.org/10.1109/IROS.2017.8206084).
- [12] D. Fernandez Llorca, V. Milanes, I. Parra Alonso, M. Gavilan, I. Garcia Daza, J. Perez, and M. Á. Sotelo, “Autonomous pedestrian collision avoidance using a fuzzy steering controller,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 390–401, 2011.
DOI: [10.1109/TITS.2010.2091272](https://doi.org/10.1109/TITS.2010.2091272).
- [13] T. Hesse and T. Sattel, “An approach to integrate vehicle dynamics in motion planning for advanced driver assistance systems,” In *2007 IEEE Intelligent Vehicles Symposium*, 2007,
Pp. 1240–1245.
DOI: [10.1109/IVS.2007.4290288](https://doi.org/10.1109/IVS.2007.4290288).
- [14] M. T. Wolf and J. W. Burdick, “Artificial potential functions for highway driving with collision avoidance,” In *2008 IEEE International Conference on Robotics and Automation*, 2008,
Pp. 3731–3736.
DOI: [10.1109/ROBOT.2008.4543783](https://doi.org/10.1109/ROBOT.2008.4543783).
- [15] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, “Safe driving envelopes for path tracking in autonomous vehicles,” *Control Engineering Practice*, vol. 61, pp. 307–316, 2017.
- [16] G. Adireddy, T. Shim, D. Rhode, and J. Asgari, “Model predictive control (mpc) based combined wheel torque and steering control using a simplified tire model,” In *Dynamic Systems and Control Conference*, Vol. 44175, 2010,
Pp. 165–172.
- [17] G. Adireddy, T. Shim, and D. Rhode, “Wheel torque controller development using a simplified tire model for vehicle handling enhancement,” In *Dynamic Systems and Control Conference*, Vol. 48920, 2009,

Pp. 533–540.

- [18] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, and F. Borrelli,
“Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model,”
In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*,
2013,
Pp. 2329–2334.
DOI: [10.1109/ITSC.2013.6728575](https://doi.org/10.1109/ITSC.2013.6728575).
- [19] C. Liu, A. Carvalho, G. Schildbach, and J. K. Hedrick,
“Stochastic predictive control for lane keeping assistance systems using a linear time-varying model,”
In *2015 American Control Conference (ACC)*,
2015,
Pp. 3355–3360.
DOI: [10.1109/ACC.2015.7171850](https://doi.org/10.1109/ACC.2015.7171850).
- [20] T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr,
“Combining stochastic and scenario model predictive control to handle target vehicle uncertainty in an autonomous driving highway scenario,”
In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*,
2018,
Pp. 1317–1324.
DOI: [10.1109/ITSC.2018.8569909](https://doi.org/10.1109/ITSC.2018.8569909).
- [21] H. Pacejka,
Tire and vehicle dynamics.
Elsevier, 2005.
- [22] M. P. Vitus and C. J. Tomlin,
“A probabilistic approach to planning and control in autonomous urban driving,”
In *52nd IEEE Conference on Decision and Control*,
2013,
Pp. 2459–2464.
DOI: [10.1109/CDC.2013.6760249](https://doi.org/10.1109/CDC.2013.6760249).

