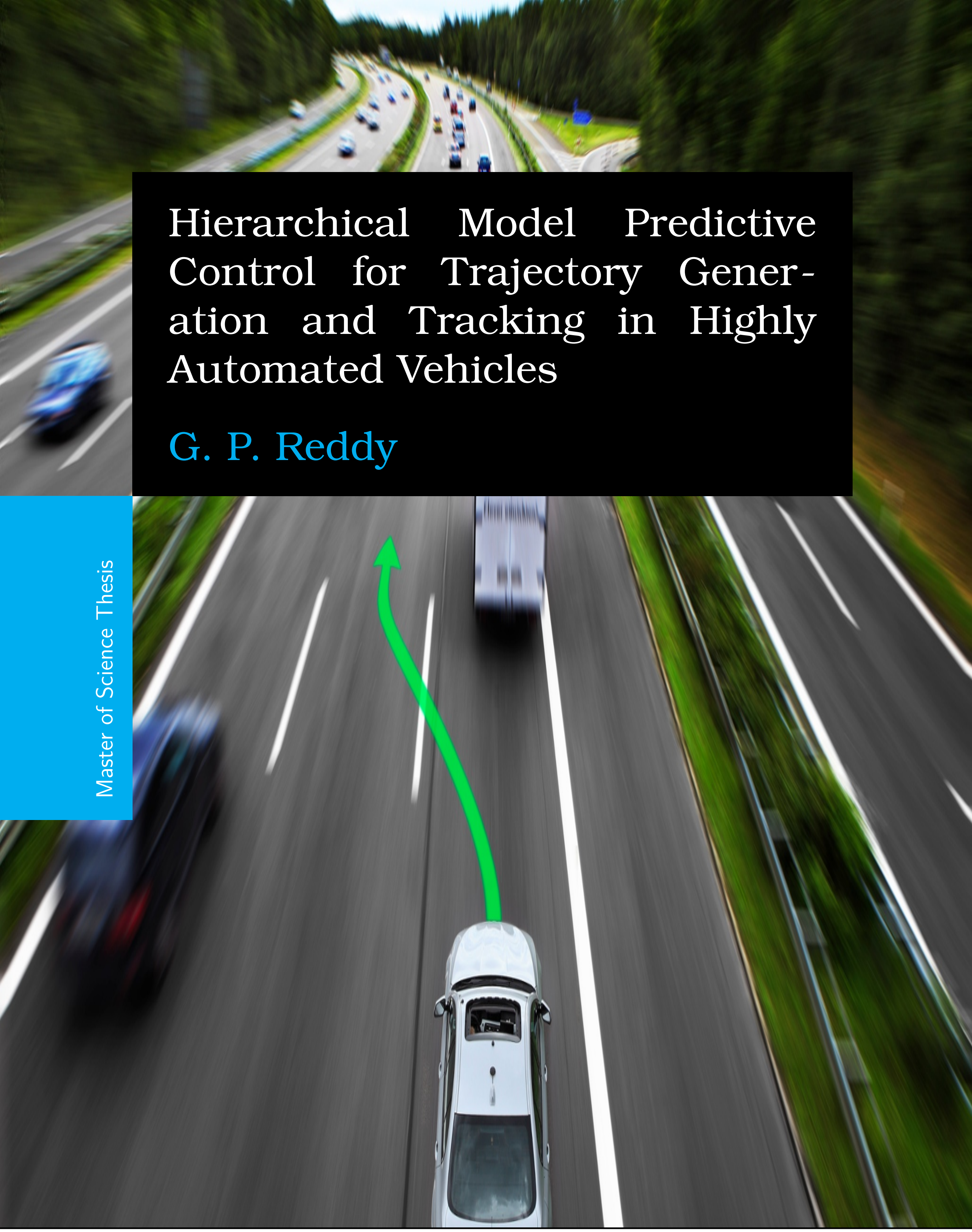

Hierarchical Model Predictive Control for Trajectory Generation and Tracking in Highly Automated Vehicles

G. P. Reddy

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Hierarchical Model Predictive Control for Trajectory Generation and Tracking in Highly Automated Vehicles

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

G. P. Reddy

April 14, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.

Abstract

The research field of autonomous vehicle technology has been growing at an accelerated pace. Improved safety, fuel and commuting efficiencies are the motivating technical and social factors to develop fully autonomous vehicles. Sensor technology, advanced software and intelligent control are the different modules that work in unison to achieve the desired driving result. The system architecture of autonomous vehicles is explored to establish a hierarchy with the planning and control stages. This allows us to focus on the particular topic of intelligent control for both levels of operation.

In this thesis, a hierarchical model predictive control is developed for trajectory generation and tracking of on-road vehicles. The state of the art methods in planning and control are predominantly developed in the robotics domain and the additional challenges of the vehicle such as non-linear dynamics, sampling time and limited computational resources make it a challenging problem. 10 DOF, six state model, and point mass vehicle dynamics models was evaluated to sufficiently represent the dynamics of the systems and allow for efficient operation in the controller development phase.

Model predictive control (MPC) is chosen because of its capability of systematically taking into account non-linearities, future predictions, and operating constraints during the control design stage. In the hierarchical approach, at the high-level, new trajectories are computed on-line, in a receding horizon fashion, based on a simplified point-mass vehicle model in order to avoid the obstacle. The formulation of the collision avoidance constraints to render a quadratic programming (QP) problem from a non-convex optimization problem was crucial in the trajectory generation phase. The parameter values involved in the forward and read collision avoidance constraints defined the feasible driving regions. At the low-level an MPC controller computes the vehicle control inputs steering and acceleration in order to best follow the high level trajectory based on a higher fidelity non-linear vehicle model.

The simulation scenarios defined cases for static obstacle avoidance, car following and special overtake manoeuvres. The effectiveness of the controllers were strongly affected by the parameter tuning of the vehicle, design constraints, and collision avoidance terms. The chosen method implemented a hierarchical controller with a higher level deliberative paradigm and lower level tracking controller to achieve the tasks with respect to highly automated driving.

Table of Contents

Acknowledgements **xi**

1 Introduction **1**

1-1 Aim and motivation 1

1-2 State of the art methods 3

1-2-1 Control systems in automotive industry 4

1-2-2 Model predictive control methods 4

1-2-3 Environment model 6

1-3 Research goals 6

1-4 Organisation of the report 7

2 System model **9**

2-1 Coordinate system 9

2-2 Point Mass model 11

2-3 Tire model 12

2-4 Full Vehicle model 16

2-5 Vehicle model parameters 19

3 Hierarchical control for planning and control **21**

3-1 System architecture and Control Hierarchy 21

3-1-1 Operational autonomous vehicles 24

3-2 Model predictive control 24

3-2-1 Generic Model Predictive Control problem 26

3-3 Hierarchical control system 29

3-4 Concluding remarks 30

4 Model predictive control methods for trajectory generation 31
4-1 Control Objectives . . . 31
4-1-1 Practical constraints . . . 32
4-2 Collision avoidance constraints . . . 33
4-2-1 Longitudinal distance from surrounding traffic . . . 34
4-2-2 Forward collision avoidance constraints . . . 34
4-2-3 Rear collision avoidance constraints . . . 35
4-2-4 Limitations . . . 36
4-3 Optimal control problem formulation for trajectory generation . . . 36
4-3-1 Prediction horizon . . . 37
4-3-2 Tuning weights . . . 37
4-3-3 Cost function . . . 38
4-3-4 Equality and box constraints . . . 38
4-4 Constrained quadratic optimization problem . . . 39
4-4-1 Implementation and Optimization solvers . . . 40
4-5 Concluding remarks . . . 41

5 Model Predictive control for trajectory tracking 43
5-1 Tracking control . . . 43
5-2 Control Objectives . . . 44
5-2-1 Practical constraints . . . 44
5-2-2 Steering angle for lower level controller . . . 44
5-2-3 Lateral tire forces . . . 45
5-3 Optimal control problem formulation . . . 45
5-3-1 Prediction Horizon . . . 45
5-3-2 Cost function and tuning parameters . . . 46
5-3-3 Δu formulation . . . 46
5-4 Controller design . . . 47
5-5 Tracking controller simulation . . . 48
5-5-1 Double lane change manoeuvre . . . 49
5-5-2 Implementation and Optimization . . . 49
5-5-3 Results . . . 50
5-6 Concluding remarks . . . 50

6 Simulation Results 55
6-1 Simulation environment and controller tuning . . . 55
6-2 Scenario 1: Free driving . . . 56
6-2-1 Trajectory generation controller . . . 57
6-2-2 Hierarchical controller . . . 57
6-3 Scenario 2: Car following . . . 57
6-3-1 Trajectory generation controller . . . 62

6-3-2 Hierarchical controller 62
6-4 Scenario 3: Lane change manoeuvre 62
6-4-1 Trajectory generation controller 64
6-4-2 Hierarchical controller 64
6-5 Scenario 4: Static obstacle avoidance 68
6-5-1 Trajectory generation controller 69
6-5-2 Hierarchical controller 69
6-6 Scenario 5: Overtake manoeuvre with slow moving vehicle 69
6-6-1 Trajectory generation controller 72
6-6-2 Hierarchical controller 72
6-7 Concluding remarks 76

7 Conclusion 77
7-1 Thesis contribution 77
7-2 Future work 77
7-2-1 Model mismatch uncertainties 77
7-2-2 Alternative interfacing for Higher level and Lower level controllers 78
7-2-3 Non-convex search spaces 78
7-2-4 Computational time 78
7-2-5 Prediction model for system and surroundings 78

A Appendix 79
A-1 Appendix A 79
A-2 Bicycle model 79

B Appendix -2 83
B-1 Controller implementation in MATLAB 83
B-1-1 Main code 83
B-1-2 Higher level controller function 88
B-1-3 Lower level controller function 90
B-1-4 Standalone lower level controller 92

Bibliography 99

List of Figures

1-1 Number of fatalities and injuries in in EU area [1]. 2
1-2 External costs per mile of driving [2]. 2

2-1 Co-ordinate system of the vehicle defining the longitudinal and lateral motion of the vehicle [3]. 10
2-2 Forces acting on tire model [3]. 12
2-3 Lateral and longitudinal forces for different slip coefficient [3] 15
2-4 Lateral forces and longitudinal forces variation with braking and cornering [4] . . 15
2-5 Friction circle curves [3] 16
2-6 Full vehicle model [5]. 17

3-1 Modules involved in motion autonomy. 23
3-2 Annieway from Karlsruhe University system architecture and sampling optimization to decide optimal trajectories [6] 25
3-3 Boss from Carnegie Mellon University system architecture in the vehicle.[6] . . . 25
3-4 Controller architecture for trajectory generation and tracking. 29

4-1 Forward collision avoidance constraint [7]. 35
4-2 Rear collision avoidance constraint [7]. 36
4-3 Higher level control block. 40

5-1 Lower level control block 47
5-2 Vehicle oversteer and understeer limits with respect to steering angle and lateral acceleration [8]. 48
5-3 Vehicle oversteer and understeer representation. [9] 48
5-4 Double lane change manoeuvre specifications [8]. 49
5-5 Path following performance of vehicle entering at $10[m/s]$ 51

5-6 Path following performance of vehicle entering at $20[m/s]$ 51
5-7 Lateral position Y of the controller vehicle. 52
5-8 Steering and steering rate measures for the double lane change manoeuvre. . . . 52
5-9 Brake torques at the four wheels. 53
5-10 Yaw angle and yaw rate of the vehicle during the double lane change manoeuvre. 53

6-1 Trajectory generation in free highway cruising. 58
6-2 Trajectory generation in free driving at higher speed. 59
6-3 Time taken for controlled vehicle to reach different desired speeds. 60
6-4 Performance of the hierarchical controller. 60
6-5 Performance of the hierarchical controller for non zero initial condition for lateral position Y. 61
6-6 Car following ability when surrounding vehicle is travelling at desired highway velocity. 63
6-7 Car following ability when surrounding vehicle is travelling at desired highway velocity. 64
6-8 Hierarchical controller car following feature 65
6-9 Lane change manoeuvre for vehicle. 66
6-10 Lane change manoeuvre for vehicle with higher desired speed. 67
6-11 Lane change manoeuvre with trajectory generation and tracking. 68
6-12 Overtake manoeuvre with static obstacle in desired lane. 70
6-13 Overtake manoeuvre with static obstacle in desired lane with outside lane centre reference. 71
6-14 Hierarchical controller performing overtake manoeuvre with static obstacle in desired lane. 72
6-15 Overtake manoeuvre with slow moving obstacle in desired lane. 73
6-16 Lane change manoeuvre with slow moving obstacle in desired lane. 74
6-17 Lane change manoeuvre with slow moving obstacle in desired lane. 75

A-1 Bicycle (one track) vehicle model for forces acting on a vehicle body fixed frame [5] 80

List of Tables

2-1 Packeja model coefficients for different road surfaces. 14
2-2 Vehicle parameter values used to define the full vehicle model of the vehicle. . . 19

6-1 Computer hardware details. 56
6-2 Controller tuning parameters 56

Acknowledgements

I would like to thank my supervisor Dr. Meng Wang for his assistance with extensive and helpful feedback. Dr. R. Happee provided helpful insights to guide the direction of the study.

I would like to specially thank Prof. H. Hellendoorn for his support and accomodating nature during the course of the study.

This thesis would not have been possible without the support and help of Lourdes Gallastegui Pujana. It was a patient effort to help me achieve this goal.

Finally, I would like to thank all my friends who have helped me with the process.

Delft, University of Technology G. P. Reddy
April 14, 2016

Dedicated to my family.

Chapter 1

Introduction

Autonomous vehicle **AV!** (**AV!**) technology is set to revolutionize on-road transportation. Vehicles that drive autonomously have the potential to reduce vehicular accidents, and energy costs [2]. The ability of the vehicles to perform multiple automated functions allowing the driver to monitor is a requisite step towards fully autonomous vehicles. There are several promising research directions in the road towards complete autonomy, with respect to sensors, advanced software and intelligent control [10].

The developments in the fields of sensors and control that contribute to the autonomous vehicle technology has grown rapidly in the past decade. The synchronous operation of these systems were evident in the DARPA challenges [11]. These demonstrations encouraged further development of vehicles at universities and industry. The key factors would involve a systematic implementation of autonomous driving functions based on current level of technology. The current systems have been developed to have different system components such as perception, planning, and control. Here, sensor technology is important with the perception component. Another important component, intelligent control has advanced to bridge the gap between lower level operational control towards higher level planning stages. The demands for higher level control algorithms has grown linearly with AV technology.

Particularly interesting, are the planning and control systems of autonomous vehicle technology. The planning systems is synonymous to the motion planning system where trajectory generation is a key focus. The lower level controllers are responsible for tracking the planned trajectory and directions where integrated longitudinal and lateral control take precedence. Planning and control topics are explored in detail in this thesis study to develop a feasible predictive control algorithm for the desired goal of generating safe trajectories and actuate the vehicle to track the trajectory.

1-1 Aim and motivation

Driving technology for self-sufficient vehicles, can enhance the passenger safety, and provide numerous other benefits [4]. The potential to reduce the number of accidents is a strong

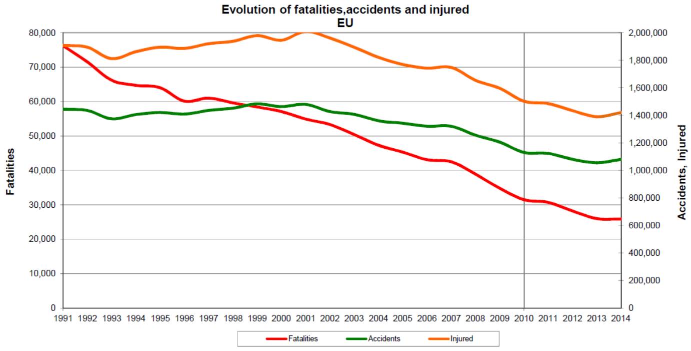

Figure 1-1: Number of fatalities and injuries in in EU area [1].

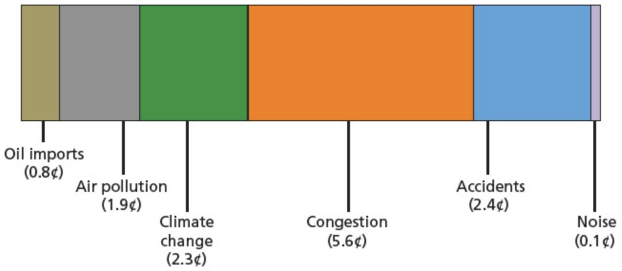

Figure 1-2: External costs per mile of driving [2].

motivating factor to pursue this research direction but the economic benefits are highlighted as well. The advantages of increased mobility, substantial savings in traffic congestion cost, which in turn have an impact on pollution levels.

The figure 1-1 indicate a significant 62% drop in fatalities since 1991 but the number of accidents and injuries still remain high. This leaves sufficient room for development of safety systems that accompany autonomous driving. Figure 1-2indicates the overall costs associated with every mile of driving and amounts the congestion and accident costs as the highest expenses with environmental costs following closely. These economic factors are a major influence on speeding up the life cycle of autonomous vehicle technology.

In this research direction, vehicles capable of achieving autonomous driving feature in the form of extended automated driving capability is important. Two broad areas that fall under autonomous vehicle technology are non-cooperative and co-operative systems. The main reasons to choose the direction of non co-operative systems would be the drawbacks associated with telematics and communication. The need for advanced network protocols and the high penetration rate required as stated in [12] strengthens the proposal. Some general issues with communication networks include,

- Bandwidth
- Latency
- Reliability
- Cost

Vehicle cooperation is a growing field but vehicles that achieve self-sufficient operation is more crucial for AV technology implementation in real time. Another important factor would be the redundancy measures that are meticulously planned in auto-mobiles. The vehicle must be able to rely on technology that can be robust and operate when communication systems are disrupted [13]. Since, non-cooperative vehicles only have sensors to depend on receiving information; intelligent control must take a substantial role. The technology must be able to partially compensate for the range, resolution, accuracy limitations of sensors [14].

One possible approach to compensate for IVC is environment modelling where structured environments can be exploited to describe behaviour of surrounding vehicles. Augmenting the control system with this information can potentially lead to reliable decision making systems at the higher level controller. This is elaborated in the literature review section and the advantages and limitations of such an approach is highlighted.

Another important approach, in the intelligent control section would be to develop reactive control systems which can achieve autonomous driving capability. The current vehicles are capable of automating some tasks, but availability of integrated features is limited. This is represented in the scale of autonomous capability [15] at level 2. The previously described components with sensing, software and control have to be examined closely. With the problems of high safety tolerance in vehicles, limited on board computation capacity pose the need to develop control systems that can satisfy the requirements. Advanced controllers such as MPC have been used in the process industry for several decades [16], [17], [18], but slowly gaining traction with the automotive application. The ability to plan the task and execute the task is noteworthy and a hierarchy can be established with respect to the planning and control systems.

Finally, the problem of planning and control can be subdivided into higher level control with path planning, trajectory generation and lower level trajectory tracking control. The layered architecture with the autonomous vehicle systems is evident and different strategies can be implemented especially with the advancement of control algorithms in the planning stage. The advantages of this structure can be exploited while addressing the limitations.

1-2 State of the art methods

A literature survey was conducted to analyse the different topics under autonomous vehicle technology with respect to vehicle dynamics, environment dynamics, control system architecture, and hierarchical scheme for trajectory generation and tracking. Trajectory generation and tracking problems have been commonly addressed as separate problems in literature. More recently, the interest to combine these features has grown with control systems capable of handling higher level strategic tasks. In this section, the state of the art methods for predictive control of autonomous vehicles are listed, allowing for a constructive problem formulation. The methods for trajectory generation have evolved from robotics applications and primitive with automotive application whereas, trajectory tracking problems have been solved as seen in the widely used driver assistance systems. The additional challenges of our application are the non-linear vehicle dynamics, limited computational resources and speed of operation. The methods reviewed during the literature study were selected to meet the requirements of the process.

1-2-1 Control systems in automotive industry

Automotive control systems enhance vehicle efficiency and safety. Traditionally, the control systems handled specific tasks in maintenance, diagnostic or operation. In vehicle dynamics control, longitudinal and lateral control systems form the base technology for the foray into integrated vehicle dynamics controllers. This technology has been implemented as seen in the number of advanced driver assistance system (ADAS), in longitudinal and more recently lateral control systems. PID controllers, feed-forward controllers, for longitudinal control have been used as seen in [5]. Adaptive cruise control (ACC), anti brake system (ABS), and traction control were first supportive systems operating in the forward moving direction. Lateral direction controllers have been developed since [19] and gained more popularity in the recent years [20]. Lane keep assist (LKA), lane departure warning (LDW) and active front steering (AFS), have been implemented handling vehicle dynamics in the lateral direction. Approaches towards integrated longitudinal and lateral controllers [21] [22] have also been developed in parallel but with increased number of assumptions with respect to infrastructure and vehicle dynamics.

The involvement of control for planning has increased with the demand for intelligent control [23]. The evolution of vehicle control from task specific to including several tasks at once, is an important study to understand the modelling of vehicle dynamics and the appropriate control systems. The classification of autonomy allows for distinction between different features and this thesis covers the semi-autonomous (Level 3) and highly automated levels (Level 4) as stated in [15]. The vehicles capable of this higher level automation were developed for [24], before commercial automotive manufacturers Volvo [25], BMW [26], Tesla [27] started developing their own technology.

The topics of lower level control and higher level planning were reviewed to lay the groundwork for the planning and control problem. Consequently, the model predictive control methods are studied with respect to trajectory generation and tracking.

1-2-2 Model predictive control methods

The use of mathematical models in model based control design is a growing field. In that direction, model predictive control has been implemented in a host of applications such as oil and gas, petrochemical, chemical, paper manufacture [16], with the developments in theory and computational capacity. The use of MPC in predominantly process industry to other industries is fast growing [28]. Optimal control methods for automotive application traction control [29], [30], due to feasible real time implementation.

The challenges now lie with the use of non-linear model predictive control(NMPC) where non-linear systems can be used directly in to develop control laws. Works by [31], [32] and [33] are great sources for theory on NMPC, problem formulation and potential applications.

Trajectory generation problem

Trajectory generation is a subset of motion planning and has been predominantly developed in the robotics field. The distinction between path planning where a geometric path is the focus whereas trajectory planning deals with the system states as it evolves in time or with respect to the road geometry as seen in [34] .

Off-line and On-line methods are available under trajectory planning with each method carrying specific merits. Off-line trajectories primarily allow faster computational times. In this thesis, on-line trajectory planning methods are incorporated as the trajectory might need replanning and obstacles are dynamic in the driving environment. Spline based methods that render smooth paths were studied starting with the [35] that developed time optimal continuous trajectories for robot arms. Cost function based approaches have been used in the past using newton optimization methods in [36]. Cubic and quintic spline based methods were applied in [37] to obtain smooth trajectories during the planning phase. Another commonly used method in robotics is the potential fields method [38] where the controlled system avoids obstacles which are defined by a repulsive force and the target is defined by an attractive force.

Focusing on the automotive application is more complex with vehicle dynamics, fast paced environment and high safety regulations. The recent works include for parametrized curve based approach in [39], manoeuvre planning algorithm [40]. The optimization based approaches used in the study [41] and methods review in [42] are important.

Trajectory tracking problem

Trajectory tracking methods have been used for control of non-holonomic systems as studied in [43] and achieve stabilization, and [44] have worked on methods to guarantee closed loop stability.

In the past, the MPC problems have been formulated as regulation problems driving the system towards the reference. This problem is complex with time varying references [45]. The tracking control problem is applicable with many applications characteristic of a varying references and the especially interesting with constrained non-linear systems.

Nonlinear predictive control [46], system is partially linearized and set point are parametrized but tracking control is given less importance. Many papers have worked with reference governors to handle the constantly changing reference values. But the approach has been disregarded for lack of robustness.

The tracking problem of the vehicle dynamics control is solved employing model predictive control approach that have been successfully tested in [47], [48]. Nonlinear MPC was tested with nonlinear bicycle model in [49]. The conditions for a snowy road were navigated in [7].

Hierarchical control

System architecture for autonomous vehicles have been developed based on different strategies to achieve the goal of autonomous driving. Researchers have leaned towards the deliberative approach involving a higher level planning feature or a more reactive approach which resembles a rule based behaviour scheme [50] . Each of the approaches brings along certain advantages but a hybrid approach has been used to develop the three main modules of autonomous vehicles; perception, planning and control.

A brief view on the system architecture used in autonomous vehicles[11]. The development of GAT, Payton [51] introduced the different higher level planning techniques available. On the other approach, AuRa, SSS [51] architectures were used by vehicles to establish the different manoeuvre based techniques available.

The hybrid approaches seen in Sharp [51] have been developed by different teams which can combine the different modules to achieve the autonomous driving task.

1-2-3 Environment model

Several different approaches were taken to represent the dynamics of the environment. Firstly, driver models [52] with respect to longitudinal, lateral control tasks and situation specific tasks were studied. This approach although applicable in certain scenarios, is limiting due the high variability of the driver behaviour. Although the environment considered is a structured highway environment, nonetheless, driver models cannot be chosen as an appropriate approach to model the surrounding vehicles. In the same direction, microscopic traffic models [53], [54] were explored as they aim to represent the dynamics of the environment locally, and due to the sensor range of the car, this as another potential approach to model the dynamics of the surrounding vehicles. Finally, environment models were considered with respect to physics based, manoeuvre based and interaction aware models [55].

The above methods were chosen in an attempt to represent the dynamics of the surrounding environment but proved to be challenging to model and validate the augmented model. Rather, the rules defined under the studies were extracted to model the behaviour of the controlled car with respect to the surroundings. The heuristic rules set the constraints for acceleration and jerk bounds, car following rules, and lane change rules.

1-3 Research goals

The following research goals were formulated based on the literature survey and the methodology followed during the thesis work. The two main components were environment prediction and controller design, but the focus area of this study was dominated by the hierarchical controller development. The objective to enable the vehicle to operate autonomously to perform lane changes and car following automation functions. The control system is designed to include collision avoidance constraints to navigate safely among surrounding vehicles. The main goal of the thesis is designing a hierarchical model predictive controller with higher level trajectory generation and lower level trajectory tracking. The following way points guided the thesis work during the controller development

- Formulation of the trajectory generation problem as a non-linear problem, and the use of suitable solvers to obtain a collision free trajectory.
- Controller design for a minimal error path tracking controller, by obtaining the references from the higher level planner.
- Analyse the implementation aspects of the hierarchical model predictive controller.
- Evaluate the performance of the trajectory generation and trajectory tracking controllers individually and in the hierarchical scheme.

1-4 Organisation of the report

The following report is structured to introduce the topics, describe the vehicle model, controller development methods, and the simulation results.

Chapter 1 is used to introduce the thesis focus areas of autonomous vehicle technology . The aim and motivation of the chosen topics are elaborated leading to a brief summary of the state of the art methods. The role of control in autonomous vehicles is introduced and then focusing on the model predictive control methods to achieve our control objective. These methods are analysed to formulate the research goals.

Chapter 2 details the vehicle dynamics models with different degrees of freedom. The accuracy of the model differs in the point mass and the full vehicle model. The states and the control variables are clearly established. The non-linear tire model is also included which increases the vehicle dynamic behaviour with the trajectory tracking controller.

Chapter 3 is used to introduce the system architecture of autonomous vehicles with examples in literature and practical applications. Model predictive control theory and optimization problem are highlighted along with the control hierarchy. The controller for the hierarchical design and the method of connecting different controllers that influence the implementation aspects.

The model based controller design is formulated in Chapter 4 with an emphasis placed on model predictive control theory. The trajectory generation controller is described using the objective function, trajectory planner and the obstacle avoidance constraints. The higher level planner is in charge of the obstacle avoidance function to plan safe trajectories. The linear collision avoidance constraints are used for collision free paths in the forward direction and from rear direction. The bounds are also set with respect to the non drivable areas in the highway scenario.

Chapter 5 is used to elaborate on the lower level tracking controller which is using a full vehicle model. The advantages of using a split architecture are highlighted and the several constraints are identified. The lower level controller satisfies the driving tasks at the operational level.

Chapter 6 is for the simulation results of the different scenarios tested to evaluate the performance of the controllers. The selected scenarios evaluate the performance of the trajectory generation controller and the trajectory tracking controller.

Chapter 7 concludes the final conclusion of the thesis with final results, and recommendations of future work.

Chapter 2

System model

This chapter details the vehicle dynamics model with respect to the point mass model and the full vehicle model used for controller development in section 2-2 and 2-4 respectively. The non-linear tire model described in section 2-3 is included in the full vehicle formulation to increase the accuracy of the representation. The Pacejka magic formula model is chosen among the available tire models. Finally, the chapter culminates with a analysis of the models and the dynamics that can be reproduced for the simulations.

The vehicle model is highly complex and therefore traditionally, decomposed into different components that represent the dynamics of the system. Longitudinal, lateral and vertical direction can be chosen to analyse the behaviour of the system in that particular axis. Here, the interaction component to be included is based on a select few parameters that will be elaborated in the later section. For the purposes of decomposition, the co-ordinate system is introduced that can follow one of two approaches.

2-1 Coordinate system

The dynamic equations can be modelled with two approaches, in the inertial frame or the body centred frame. The body centered

The first approach using a Cartesian co-ordinate system as the reference frame. This is similar to describing the vehicle position and heading angle with the body centred axis system. The motion of the vehicle can be described using the X-Y co-ordinates. Here, the longitudinal motion of the vehicle is along the X-axis and the lateral motion of the car is along the Y-axis. This representation will be followed throughout the rest of the thesis and controller development.

Another approach is using a curved linear coordinates to show the relative position of the vehicle with respect to the curve. The difference between local vehicle axis and global axis has to be recognised when deducing the equations. This representation is especially useful in the spatial formulation of the problem and the situation when the road geometry is included.

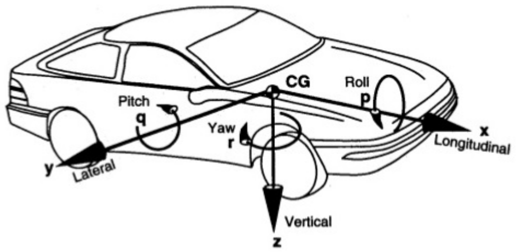

Figure 2-1: Co-ordinate system of the vehicle defining the longitudinal and lateral motion of the vehicle [3].

The representation of the the points and vectors that define the motion or, using the Cartesian coordinate system to define the position and movement with respect to a reference frame. The advantage with the Cartesian coordinate system is seen with the ability to capture the position and orientation essentially, denoting the all the available time derivatives. Any vector can be assigned a space using the three axes system of the coordinate system. Reference frames are helpful to understand the movements of vectors relative to each other.

Kinematic relations between vehicle components can be exploited to deduce key behaviours. The Cartesian co-ordinate system positioned in the modelling space. This method can be used to model orientation and position. The transformations from the wheels that can be steered and relevant wheel velocities are challenging to map. The steer angle is determined commonly as the Ackerman steering to obtain the degrees of freedom for steer.

Degrees of freedom

Vehicles behaviour can be represented with various degrees of freedom [56]. The simple dynamic model is a two **DOF!** (**DOF!**) bicycle model, representing the lateral and yaw motions. The third degree includes the longitudinal acceleration to the model and can essentially describe a full vehicle. The five degree of freedom model including the wheel rotations enables analysis of traction forces and useful for controller design of combined braking and steering. Following this full vehicle model, with eight or nine degree of freedom are used for accurate description of vehicle dynamics. The following subsections cover the vehicle dynamic model that are feasible to use in the control architecture. The instance a model with lower complexity is feasible, it can be used to save computational burden on the controller.

The choice of model for the model predictive controller is important as the approach uses a model of the system, formulates a prediction model for use in the optimization procedure. The main trade-off with using any model is between accuracy and complexity. This is crucial with our application due to the safety regulation of the automotive industry and real life implementation aspects. The difference in dynamics must be minimal for the controllers to be feasible on a the vehicle. On the other hand, due to the use of advanced controllers that rely on optimization approaches, model complexity greatly influences the speed of operation and again the feasibility of real time implementation.

Definitions

- **Tire slip** The small deviation between the the direction of heading and the direction of the rolling wheel. Due to the deformation in the tire, this difference exists and plays a huge role in the vehicle behaviour.

- **Wheel slip** The measure of the difference between the rotational speed of the wheel and the translational velocity of the wheel centre. The equation to define wheel slip is,

$$S = -\frac{(V - R_e\omega)}{V} \tag{2-1}$$

- **Braking torque** This is the torque required to bring the motor to a standstill. The engine torque is translated into the motion of the vehicle and can be a powerful control input when it comes to vehicle dynamics control.

$$T_b = F_b * r, \tag{2-2}$$

  Here, F_b is the clamping force at the brake pads and r is the effective radius of the wheel.

2-2 Point Mass model

The simplest of the vehicle model is in the form of point mass model, used to describe motion in the lateral and in the longitudinal directions. The dynamical system is given as follows:

$$\begin{aligned} m\ddot{x} &= F_x, \\ m\ddot{y} &= F_y, \\ \dot{X} = \dot{x} &= V_s cos\phi, \\ \dot{Y} = \dot{y} &= V_s sin\phi, \\ \dot{\phi} &= r, \end{aligned} \tag{2-3}$$

where x represents position of the point mass vehicle model in the longitudinal direction, y represents position of the point mass vehicle model in the lateral direction, and ϕ [rad] represents angle between the longitudinal and the lateral directions of the model or steering wheel angle and r denotes the commanded steering wheel turn rate $[rad/sec]$.

The speed is set to a constant velocity. In fact, if the angle is zero, travelling along the x direction only, we will end up with a first order differential equation in x. The tire forces for this model can be constrained by the friction circle as $F_x^2 + F_y^2 \leq (\mu mg)^2$.

Finally, the compact representation is given as:

$$\dot{\xi(t)} = f_p m(\xi(t), u(t)) \tag{2-4}$$

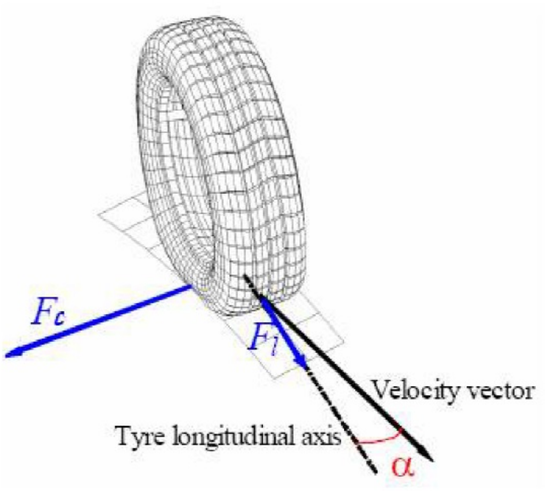

Figure 2-2: Forces acting on tire model [3].

With the state vector $\xi = [\dot{x}, \dot{y}, X, Y]$ and input vector $u = [F_x, F_y]'$.

The point mass model described fails to capture yaw dynamics, vehicle direction and tire dynamics. The relation between the behaviour of the states and the tire forces are not accounted. Although this model has several drawbacks to represent the vehicle, it can be used to represent vehicle motion and will be tested with the higher level controller. A higher priority is placed for the computation time of the trajectory generation controller.

2-3 Tire model

Tire behaviour modelling has been a challenging field due to the highly non-linear behaviour and transient characteristics. There are several different tyre models based on the modelling method [3]. Three main classifications can be noticed from empiric models, developed using data fitting regression techniques, to theoretical models, based on finite element analysis of tire, and semi-empiric models combining the two methods. Empirical models depend on direct data measurements which are hard to gather with transient vehicle dynamics. But they are compact, and suitable for fast online calculations. Theoretical models describe the steady state and transient dynamics of the tire, often rendering a computationally heavy model. Semi-empiric models represent sufficient dynamics suitable for vehicle simulation. Data and theoretical modelling approaches are combined thereby balancing accuracy and computational efficiency.

Tires are an essential vehicle component as the only medium of vehicle interaction with the road surface. The acceleration force to drive the vehicle forward (traction force) and the force to reduce the speed (braking force) is achievable through the force interaction between tire and road. The longitudinal and lateral forces are illustrated in 3-1 with respect to the direction of travel.

Introducing the nomenclature in tire modelling, the following equations are used to calculate the front and rear tire speeds along the x-axis and y-axis:

$$\begin{aligned} v_{x_i} &= v_i cos\alpha_i \\ v_{y_i} &= v_i sin\alpha_i \end{aligned} \tag{2-5}$$

where v_f and v_r are the front and rear tire speed, α_f and α_r are the front and rear tire slip angle. The correlation between tire speeds and controlled vehicle speed is given below:

$$\begin{aligned} v_{x_f} &= v_{h_x} \\ v_{x_r} &= v_{h_x} \\ v_{y_f} &= v_{h_y} + l_f\dot{\varepsilon} \\ v_{y_r} &= v_{h_y} - l_r\dot{\varepsilon} \end{aligned} \tag{2-6}$$

The slip angle represents the angle between the wheel velocity and the direction of the wheel itself [57]. Then tire side-slip angles are computed as

$$\begin{aligned} \alpha_f &= \delta_f - \arctan\frac{v_{y_f}}{v_{x_f}} \\ \alpha_r &= \delta_r - \arctan\frac{v_{y_r}}{v_{x_r}} \end{aligned} \tag{2-7}$$

Side-slip values and friction component between tire and road influence the tire forces. Wheel slip ratio in the longitudinal direction is the difference in angular speed between a purely rolling and sliding wheel, is explained in the following conditional equation,

$$s_i = \begin{cases} \frac{r\omega_i - v_{x_i}}{v_{x_i}}, & r\omega_i < v_{x_i} \ \text{Braking} \\ \frac{r\omega_i - v_{x_i}}{r\omega_i}, & r\omega_i > v_{x_i} \ \text{Driving} \end{cases}$$

where ω_f and ω_r and r are the angular velocities of front and rear tires, and the radius of wheels respectively.

The function can be used to represent acceleration and braking descriptions. A slip ratio value of zero indicates moving velocity and tire rolling speed are equal, thereby we can deduce zero engine torque. Positive slip shows tire has positive rolling velocity and a positive forward velocity. Negative slip shows there is equal or larger positive rolling velocity to that of forward velocity. Here actuator saturation can defined where -1 or $+1$ indicates 'locked' or 'spinning' wheel at zero speed. Slip goes to infinity when tire and vehicle velocity are equal to zero, and zero value assigned in simulations. Experiments to define friction characteristics between the tire and road under different conditions have been analysed in [57].

An accurate non-linear tire model is required to capture the dynamics of the tire. This is best achieved using a Pacejka model which is a complex semi-empirical one. The longitudinal forces are assumed to depend on the normal force, surface friction, and longitudinal slip ratio as shown [47] below;

$$F_{x_i} = f_x(s_i, \mu, F_{z_i}) \tag{2-8}$$

The Pacejka model calculates the friction coefficient as a function of slip ratio:

Table 2-1: Packeja model coefficients for different road surfaces.

Road Surface	B_P	C_P	D_P	E_P
Dry Tarmac	10	1.9	1	0.97
Wet Tarmac	12	2.3	0.82	1
Snow	5	2	0.3	1
Ice	4	2	0.1	1

$$\mu(s_i) = D_P sin(C_P \arctan(B_P s_i - E_P(B_P s_i - arctan(B_P s_i)))) \tag{2-9}$$

The value of B_P, C_P, D_P, E_P for different road types are shown in 2-1. The values of the model constants are determined using experimental data, and consequently the co-efficient values B_P, C_P, D_P, E_P are derived.

Note that it has been concentrated on dry tarmac (asphalt) coefficients $B_P = 10, C_P = 1.9, D_P = 1, E_P = 0.97$ in compliance with previously made assumptions. Having the friction coefficient $\mu(s_i)$ and the normal force F_{z_i} exerting on the wheel, the longitudinal traction force F_{x_i} is then calculated as follows:

$$F_{x_i} = \mu(s_i) F_{z_i} \tag{2-10}$$

The non-linear nature of tire forces vary with multiple parameters and the handling stability requirements vary as well. The lateral forces are important to handle lane change manoeuvres and depend on normal force, friction and slip angle.

$$F_{y_i} = f_y(\alpha_i, \mu, F_{z_i}) \tag{2-11}$$

The figure 2-3 shows the longitudinal force versus longitudinal slip. The next illustration shows the slip angle variation. The values are representing the condition for the front wheel with the one track model [47]. Figure 2-4 shows longitudinal motions of the vehicle are related to the tire forces and the variation in values. The non-linear properties become too complex hen the transient dynamics of longitudinal and lateral are coupled. Therefore, a decoupled relationship is assumed for the control problem. Since manoeuvring capability of the vehicle is limited at high speed in highways and slip angle commutes in linear region. Lateral tire forces are linear functions of slip angles, cornering stiffness of the tire for the sake of simplification:

$$F_{y_i} = \alpha_i C_{y_i} \tag{2-12}$$

where C_{y_i} are the cornering stiffness. The factor 2 in the one track model accounts for the fact that there are 2 tires per axle in the bicycle model adopted [5].

$$C_{y_i} = C_s \frac{F_{z_i,static}}{2} \tag{2-13}$$

where C_s is the cornering stiffness coefficient. Generally, typical values for the cornering stiffness coefficient C_s varies between 0.12/deg to 0.16/deg according to type of tires [58] . The longitudinal and lateral tire force F_{x_i} , F_{x_i} are limited physically by the adhesion limit

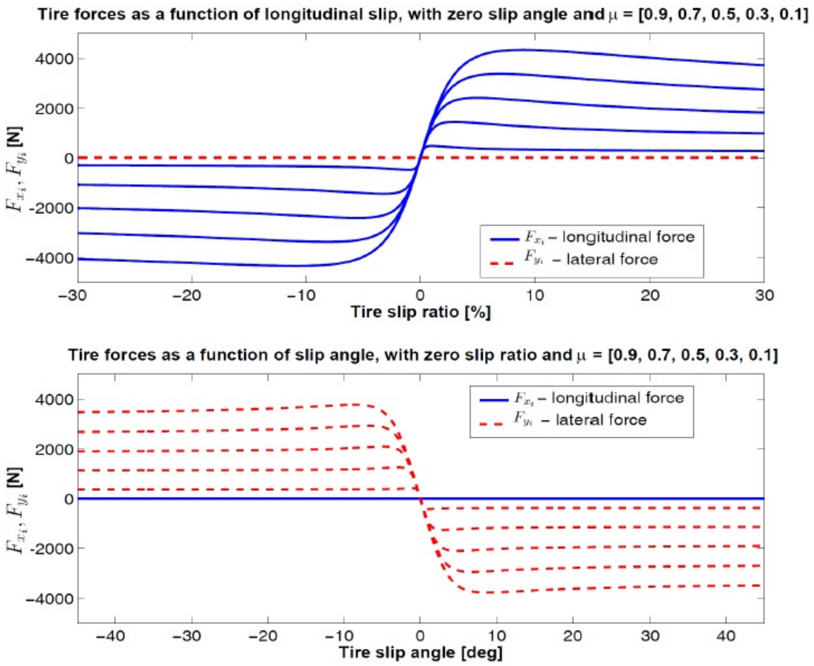

Figure 2-3: Lateral and longitudinal forces for different slip coefficient [3]

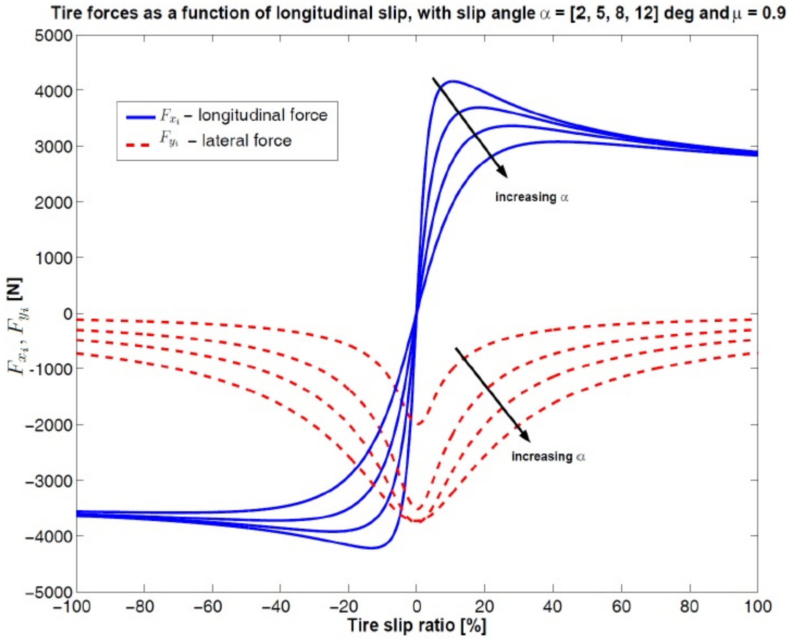

Figure 2-4: Lateral forces and longitudinal forces variation with braking and cornering [4]

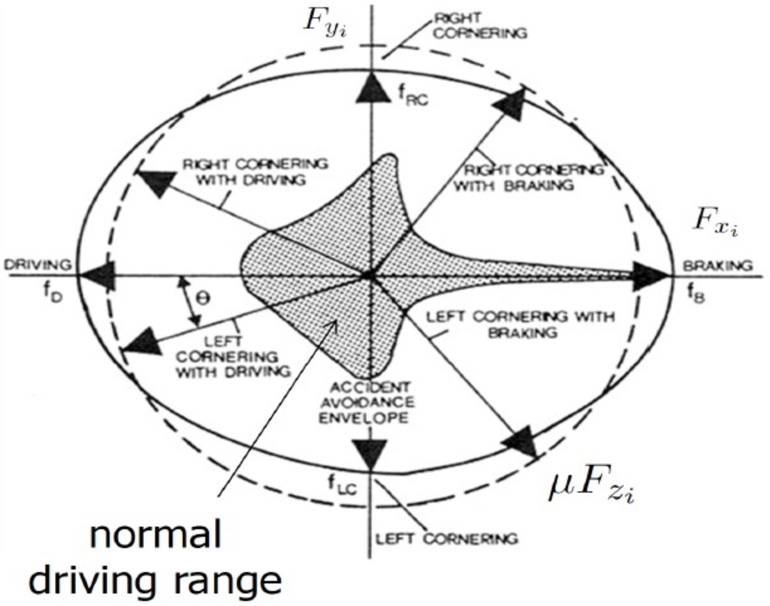
F_{y_i}
RIGHT CORNERING
f_{RC}
RIGHT CORNERING WITH DRIVING
RIGHT CORNERING WITH BRAKING
F_{x_i}
DRIVING
f_D
BRAKING
f_B
Θ
LEFT CORNERING WITH DRIVING
LEFT CORNERING WITH BRAKING
ACCIDENT AVOIDANCE ENVELOPE
f_{LC}
μF_{z_i}
normal driving range
LEFT CORNERING

Figure 2-5: Friction circle curves [3]

between tire and road. Because the forces are transmitted by friction as shown as in Figure 2-5. Therefore, during combined slip conditions the maximum transferred force is defined by Kamm circle as given

$$\sqrt{F_{y_i}^2 + F_{x_i}^2} \leq \mu F_{z_i} \tag{2-14}$$

2-4 Full Vehicle model

The full vehicle model is presented in this section where the dynamic equation is given by

$$\dot{\xi(t)} = f_f v(\xi(t), u(t)) \tag{2-15}$$

where $\xi(t) \in \mathfrak{Re}^n$ is the state of the system with 10 states and $u(t) \in \mathfrak{Re}^{m_r}$ is the system inputs with five available inputs. As discussed in [47], the ten states are lateral and longitudinal velocities in the body frame, the yaw angle, yaw rate, lateral and longitudinal vehicle coordinates in the inertial frame and the angular velocity on the four wheels. These are denoted,

$$\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X, \omega_{f,l}, \omega_{f,r}, \omega_{r,l}, \omega_{r,r}]^T \tag{2-16}$$

Here, $\omega_{f,l}, \omega_{f,r}, \omega_{r,l}, \omega_{r,r}$ are the angular velocity on the front left, front right, rear left and rear right sides, respectively. For the simplicity of notation, later we will use the subscript $\star \in f, r$ to denote the front or rear axles and $\bullet \in l, r$ to denote the left or right side of the vehicle. The system inputs are

$$u = [\delta_f, T_{f,l}, T_{f,r}, T_{r,l}, T_{r,r}]^T \tag{2-17}$$

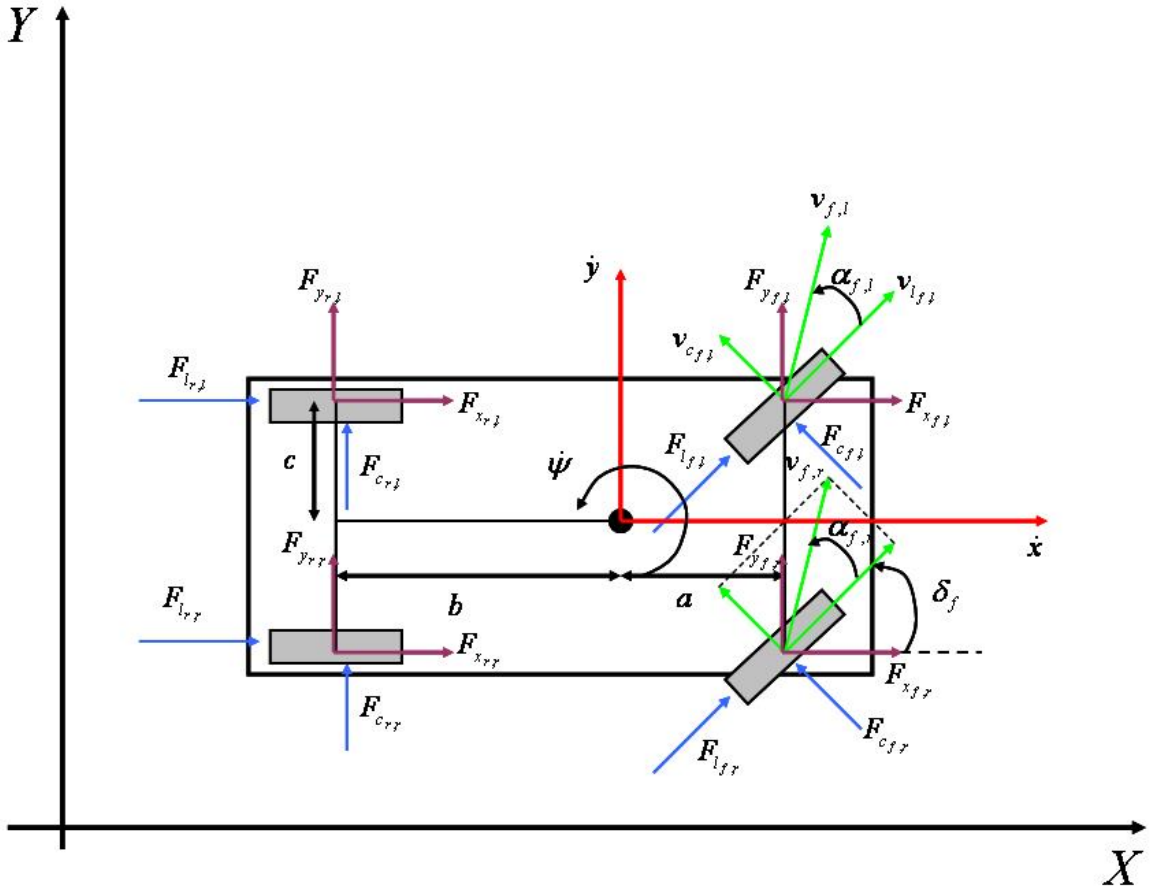
Y
$\boldsymbol{v}_{f,l}$
$\dot{y}$
$F_{y_{r,l}}$
$F_{y_{f,l}}$
$\alpha_{f,l}$
$\boldsymbol{v}_{l_{f,l}}$
$\boldsymbol{v}_{c_{f,l}}$
$F_{l_{r,l}}$
$F_{x_{r,l}}$
$F_{x_{f,l}}$
$F_{l_{f,l}}$
$F_{c_{f,l}}$
c
$F_{c_{r,l}}$
$\dot{\psi}$
$\boldsymbol{v}_{f,r}$
$\alpha_{f,r}$
$\dot{x}$
$F_{y_{r,r}}$
$F_{y_{f,r}}$
$F_{l_{r,r}}$
b
a
δ_f
$F_{x_{r,r}}$
$F_{x_{f,r}}$
$F_{c_{r,r}}$
$F_{c_{f,r}}$
$F_{l_{f,r}}$
X

Figure 2-6: Full vehicle model [5].

where δ_f is the front steering angle and $T_{\star,\bullet}$ gives the general braking and tractive torque on the wheels. Positive $T_{\star,\bullet}$ indicates driving torque while negative $T_{\star,\bullet}$ indicates braking torque.

Figure 2-6 illustrates the notation for the full vehicle model. Importantly, $F_{c_{\star,\bullet}}$ and $F_{l_{\star,\bullet}}$ are the lateral cornering and longitudinal tire forces in tire frame. $F_{y_{\star,\bullet}}$ and $F_{x_{\star,\bullet}}$ are the components of the tire forces along the lateral and longitudinal vehicle axes. $\alpha_{\star,\bullet}$ are the wheel slip angles which are defined later in this section. δf is the front wheel steering angle. a and b are the distances from the CoG to the front and rear axles and c is the distance from the CoG to the left/right side at the wheels. The dynamic equations can be derived using the vehicle motion about CoG and coordinate change between vehicle body and the inertial frame:

$$\begin{aligned}
m\ddot{y} &= -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}} \\
m\ddot{x} &= m\dot{x}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}} \\
I\ddot{\psi} &= a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}) \\
\dot{Y} &= \dot{x}sin\psi + \dot{y}cos\psi \\
\dot{X} &= \dot{x}cos\psi - \dot{y}sin\psi \\
I_w\dot{\omega}_{f,l} &= -F_{l_{f,l}}r_w + T_{f,l} - b_w\dot{\omega}_{f,l} \\
I_w\dot{\omega}_{f,r} &= -F_{l_{f,r}}r_w + T_{f,r} - b_w\dot{\omega}_{f,r} \\
I_w\dot{\omega}_{r,l} &= -F_{l_{r,l}}r_w + T_{r,l} - b_w\dot{\omega}_{r,l} \\
I_w\dot{\omega}_{r,r} &= -F_{l_{r,r}}r_w + T_{r,r} - b_w\dot{\omega}_{r,r}
\end{aligned} \tag{2-18}$$

where the constant m is the vehicle mass. I is the vehicle rotational inertia about the z axis. I_w includes the wheel and drive-line rotational inertias. r_w is the radius of the wheel. b_w is the damping coefficient. The x and y components of tire forces, $F_{x_{\star,\bullet}}$and $F_{y_{\star,\bullet}}$, are computed as follows:

$$\begin{aligned}
F_{y_{\star,\bullet}} &= F_{l_{\star,\bullet}}sin\delta_\star + F_{c_{\star,\bullet}}cos\delta_\star \\
F_{x_{\star,\bullet}} &= F_{l_{\star,\bullet}}cos\delta_\star - F_{c_{\star,\bullet}}sin\delta_\star
\end{aligned} \tag{2-19}$$

where steering angle is restricted to front steer only, thereby giving by $\delta_{f,l} = \delta_{f,r} = \delta_f$.

The longitudinal and lateral tire forces $F_{l_\star}$ and $F_{c_\star}$ are represented as in Pacejka tire model [3]. The non-linear functions of the tire slip angles $\alpha_\star$, slip ratios $\sigma_\star$, normal forces $F_{z,\star}$ and friction coefficient between the tire and road $\mu_\star$:

$$\begin{aligned}
F_{l_{\star,\bullet}} &= f_{l\alpha_{\star,\bullet}}, s_{\star,\bullet}, \mu, F_{z_{\star,\bullet}} \\
F_{c_{\star,\bullet}} &= f_{c\alpha_{\star,\bullet}}, s_{\star,\bullet}, \mu, F_{z_{\star,\bullet}}
\end{aligned} \tag{2-20}$$

where slip angles of tire, slip ratio, μ friction coefficient and F_z normal force are given by the equation:

$$\alpha_{\star,\bullet} = \arctan\frac{v_{c_{\star,\bullet}}}{v_{l_{\star,\bullet}}} \tag{2-21}$$

Table 2-2: Vehicle parameter values used to define the full vehicle model of the vehicle.

Parameter	**Symbol**	**Value**	**Unit**
Distance from CoG to front axles	a	1	$[m]$
Distance from CoG to rear axles	b	1.4	$[m]$
Distance from CoG to left/right side of the wheel	c	0.47	$[m]$
Mass of the vehicle	m	2050	$[Kg]$
Vehicle rotational inertia about the z-axis	$J1$	3344	$[Kgm^2/rad]$
Vehicle wheel and driveline interia	Jw	3951	$[Kgm^2/rad]$
Radius of the wheel	rw	0.41	$[m]$
Damping co-efficient	bw	3.8	
Cornering stiffness co-efficient	Cs	0.12	
Gravitational acceleration	g	9.81	$[m/s^2]$
Tire-road friction co-efficient	μ	0.3	

Here, tire lateral and longitudinal velocity components $v_{c_{\star,\bullet}}$ and $v_{l_{\star,\bullet}}$ are derived using,

$$\begin{aligned} v_{c_{\star,\bullet}} &= v_{y_{\star,\bullet}} \cos\delta_\star - v_{x_{\star,\bullet}} \sin\delta_\star, \\ v_{l_{\star,\bullet}} &= v_{y_{\star,\bullet}} \sin\delta_\star + v_{x_{\star,\bullet}} \cos\delta_\star, \\ v_{y_{f,\bullet}} &= \dot{y} + a\dot{\psi} v_{x_{\star,l}} && = \dot{x} + c\dot{\psi} \\ v_{y_{r,\bullet}} &= \dot{y} - b\dot{\psi} v_{x_{\star,r}} && = \dot{x} + c\dot{\psi} \end{aligned} \tag{2-22}$$

The slip ratios $s_\star$ are approximated as follows:

$$s_{\star,\bullet} = \begin{cases} \frac{r\omega_{\star,\bullet}}{v_{l_{\star,\bullet}}} - 1, & \text{if } v_{l_{\star,\bullet}} > r\omega_{\star,\bullet}, \quad v \neq 0 \ \text{ for braking,} \\ 1 - fracv_{l_{\star,\bullet}} r\omega_{\star,\bullet}, & \text{if } v_{l_{\star,\bullet}} < r\omega_{\star,\bullet}, \quad \omega \neq 0 \ \text{ for driving,} \end{cases} \tag{2-23}$$

The friction coefficient μ is taken from for the dry road surface from table . The normal force on the wheel are estimated with the non load transfer assumption, $F_{z_\star}$. Finally, considering g, g is the gravitational acceleration as well

$$F_{z_{f,\bullet}} = \frac{bmg}{2(a+b)}, F_{z_{r,\bullet}} = \frac{bmg}{2(a+b)} \tag{2-24}$$

The tire forces $F_{x_\star}$ and $F_{y_\star}$ are normal forces acting on one wheel and combined forces of the two wheels in the one track model.

2-5 Vehicle model parameters

The table 2-2 vehicle model parameters were chosen to represent a generic on road vehicle. The

The different models are examined to choose the best available fit for the controller development. Due to the computational burden of the higher level trajectory generation controller, a simple model would suffice. The point mass model that describes the orientation and motion of the vehicle is suitable for the higher level path planner. This stage of the controller will perform poorly with the complex dynamics of the vehicle model.

Another widely implemented choice for the vehicle model is the bicycle 'one track' model of the vehicle. This is a lumped parameter model where the left and right sides of the vehicle are combined. The accurate description of the vehicle motion and the non-linear behaviour is sufficiently captured with respect to our application. Nonetheless, the choice to use the full vehicle model was made due to the potential to include test the controller with evasive manoeuvres. The vehicle performs anomalously at the friction limits and thus more accurate description of the vehicle is necessary.

The use of full vehicle model would would potentially test the feasibility of the controller and provide scope to extend the controller operation to evasive manoeuvres. Full vehicle models are high fidelity models representing the dynamics of the four wheels and accounts for the asymmetric behaviour. Slip can vary at the different tires and thus tracking controllers which employ an accurate model can better represent the dynamics and controller performance of real life situations. The highly non-linear tire model is the most important consideration because all the forces transmitted with the vehicle and control inputs are directly influenced. The bicycle model is a one track model lumping the left and right wheels at the axles. This represents the yaw dynamics as well and sufficient for tracking control.

The assumptions made with the models are elaborated in the control methods section of the report. Due to the structured environment of the highway and limited scope of controller performance in the highly automated feature scheme, vehicle dynamic assumptions are made to allow for favourable controller performance.

Chapter 3

Hierarchical control for planning and control

The following chapter discusses the system architecture of autonomous systems and in particular on-road vehicles. Section 3-1 is used to highlight the planning and control modules. Following, this the need for control systems is emphasized and the theory of model predictive control is mentioned in section 3-2. The feasibility of the control architecture is highlighted with the advantages and the relevant drawbacks in section 3-3. The chapter ends with the concluding remarks provided in section 3-4.

3-1 System architecture and Control Hierarchy

Autonomy in systems is a vast research area comprising of different modules and several open problems in the topics. Initially, autonomy was first explored the field of robotics to automate motion tasks. The development of the fields has implications in other industrial process and transport domains. The field to achieve autonomy in vehicles soon followed with an emphasis on control architectures. Motion planning remains on of the significant open problem with vehicles due to the complex nature of the process.

Traditionally, the two approaches which were deliberative and reactive in nature were used to achieve autonomy. The deliberative paradigm primarily used a model of the world constructed by priori information or using sensor data. The reactive approach was more functionally oriented coupling the sensors and actuators of the systems. The different tasks were categorised into smaller elementary behaviours and reflex action was constructed to deal with the tasks. Each of these approaches were employed with different architectures focusing on a particular performance benchmark.

A brief exploration of these topics is necessary to understand the different architectures used and the relevant characteristics. The deliberative approach is synonymous with the sense-model-plan-act strategies commonly seen in [59], [60]. The perception module of the system using the sensors receive data regarding the surroundings and aid in constructing topological or grid based maps. The consequent planning stage is the decision module where motion planning takes precedence. Finally, the chosen decision is executed satisfying the operational tasks. To illustrate, two Stanford experiments using video camera sensor and 3D vision sensor supported by AI decision making capability were used to control a robot vehicle and robot cart respectively. The higher level decision making capacity was the attractive characteristic of this approach. Nevertheless, this approach had several drawbacks including the inability to deal with unpredicted events. The restricting factor being the time required in the modelling and planning stage. The sensor based models were inaccurate, and led to uncertainties due to model mismatch with the real world.

Secondly, the reactive approach was based on developing intelligent behaviours to handle complex tasks. This behaviour based approach did not require the model of the system and characteristic coupling between the sensors and actuators contributed to fast processing times. The appropriate behaviours for the system during operation were identified and consequently the competence layers were defined. Based on the conditions, the required behaviour would be evoked and executed. The commonly used subsumption architecture created a hierarchy of behaviours, each defined by a certain module. The activated module controlled the robot motion. Several notable methods were developed such as potential fields method [38], neural networks[] and Q-learning [61] in the pursuit of motion autonomy. The main drawback of this approach was the lack of higher level reasoning and limitation of defining all the required modules for the scope of operation.

The two approaches were combined in a hybrid scheme using the higher level reasoning of the deliberative approach and the robustness of the reactive approach. Hybrid schemes combined the two approaches but were predominantly deliberative based or reactive based hybrid architecture. This inherently brought some of the demerits previously mentioned with the methods. This led to the introduction of the three layer architecture where an intermediate layer interfaced the higher level planner and the lower level reactive layer. Here the behaviour of the reactive layer is instantiated by the intermediate sequencing layer subject to the constraints of the planning module. Deriving from the approach strategy used in Atlantis, 3T [62] and sharp schemes, it becomes clear that the two important modules are the motion planner and the motion controller.

This insight led to the implementation in this thesis, to develop a trajectory generator and trajectory tracking controller. The complex tasks of motion autonomy intrinsically requires hierarchical modules to achieve the desired results. Hence the following architecture was chosen to decompose the tasks in the planning and control modules.

Motion planner

The motion planner handles the task of generating and executing a motion plan for autonomous vehicles. The requirement to generate collision free trajectories must also satisfy the dynamic constraints of the vehicle and design constraints of the environment. The definition of the trajectory generation problem can be defined as,

"Given the present state of a single vehicle or team of vehicles and a map of the environment, compute a trajectory towards a desired goal state or configuration in real-time that optimizes

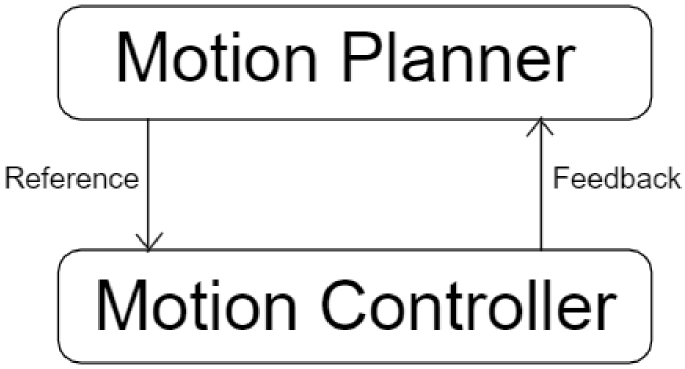

Figure 3-1: Modules involved in motion autonomy.

a certain objective function while respecting the kino-dynamic properties of the vehicle(s) and avoiding obstacles and collisions [63] ."

The methods used to generate these trajectories either use a model of the environment, construct one using sensor data or include behaviour logic for actions of the objects in the surrounding. The sensing and perception module provides information to the planning module and this information is assumed to be perfect for our system. From [], we can see that motion planning has been divided into path, manoeuvre and trajectory planning. The path planning task is to provide the geometric shape of the path to take whereas the manoeuvre planning is reactive in nature, choosing the right manoeuvre based on the conditions.

The motion planner defined for this thesis is designed to generate the best trajectories in a highway driving environment. The distinction is made with respect to the motion planner that the specific task is trajectory generation.

The planning module also includes the collision avoidance constraints which are essential to plan collision free trajectories. The methods available in literature dealing with static obstacle and moving obstacles vary in implementation,

Motion controller

The motion controller in this application handles the operational tasks of the vehicle. This involves tracking the reference trajectory by providing the appropriate control actions in the form of steering and brake forces. The comfort margins are defined with respect to the control gains in order to meet the comfortable driving standards.

The trajectory planning is taken care of in the higher level and the lower level controller using a nonlinear model is designed purely to test the feasibility of tracking a trajectory generated using a point mass vehicle model. The choice of the controller was evaluated between simple feedback-feedforward controllers and more complex predictive controller. The final implementation choice was MPC and the details are motivated in the tracking controller chapter.

3-1-1 Operational autonomous vehicles

In this subsection, the operational autonomous vehicles are discussed in detail with especially the planning stages. The system architecture discussed in the state-of-the-art methods, highlighted the development of different approaches with respect to deliberative, reactive and hybrid paradigms. Implementation of the these architectures with autonomous vehicles, uses a combination of features from the available approaches.

The DARPA challenge [64] benchmarked the technical competence of autonomous vehicle in practice. In particular, the vehicles Annieway from Karlsruhe, Boss from Carnegie Mellon and Junior from Stanford deserve special attention. All the vehicles exhibited the common architecture modules of perception, planning and control in different structures.

Team Annieway from Karlsruhe university, used a state machine in the higher level planner, and A* search algorithm. The higher level planner included a situational awareness module to enforce feasible manoeuvres. The cost function developed used a simple distance objective with respect to obstacles close to the controlled vehicle. The trajectory was estimated as a function of alignment with the reference path and the lateral offset from the lane centre, $f_t = f(a, b)$. The drawbacks with using motion primitives was compensated using frenet frames splitting the lateral and longitudinal controllers. The initial and terminal states were fixed using quintic and quartic polynomials respectively to enable a sampling based optimization as seen in 3-2 which achieved the final goal of generating collision free trajectories.

The vehicle from Carnegie Mellon used a different approach including a mission planning and manoeuvre executive blocks as seen in 3-3 to support the motion planning module. The global lane centre and local non-holonomic trajectory generator optimization goals were defined with the on-road environment. The advantages of the winning vehicle, Boss were the fast computation times for smooth operation and detailed global planning stages for enhanced performance. The need for accurate vehicle models was highlighted in this winning combination.

The last important example is the V-charge vehicle, that uses a hierarchical task decomposition framework to achieve the autonomous driving manoeuvres that are industry important in terms of product life cycle. The on-line motion planning stage uses a tree search algorithm. The common algorithms used in motion planning are the graph search methods, state lattices, and randomized approaches (RRT, RRT*). The collision detection terms need special attention and form an important segment of the planning modules. The methods used in these papers were analysed to make available all the methods during the implementation stage. The focus was shifted to model predictive control for planning and control stages with the relevant advantages. The choice of MPC is chosen for the robust and constraint handling capabilities. In accordance, the theory of MPC is explained in detail in the following section before illustrating the hierarchical structure for the controllers.

3-2 Model predictive control

The core theme of the thesis dealing with model predictive control deserves detailed attention with the advantages and theory for the implementation. This is additionally useful to implement the two level MPC which poses problems during implementations. The advantages of MPC are listed to motivate the choice of controllers.

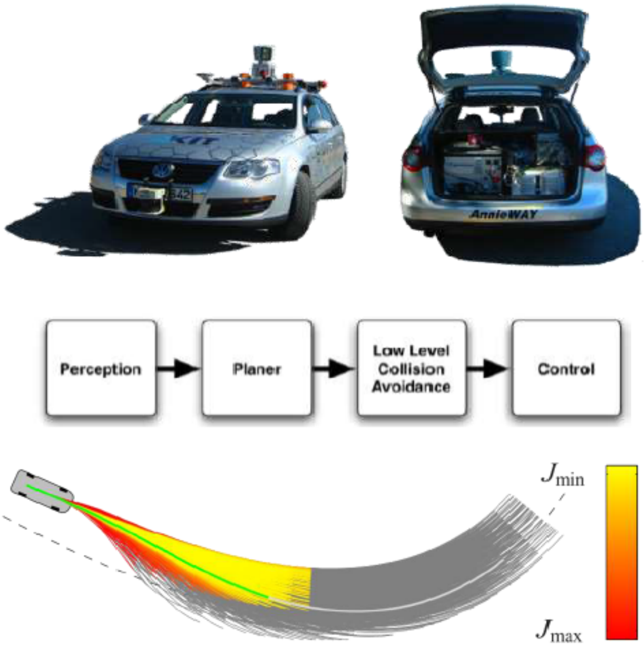
Perception
Planer
Low Level Collision Avoidance
Control
J_{min}
J_{max}

Figure 3-2: Annieway from Karlsruhe University system architecture and sampling optimization to decide optimal trajectories [6]

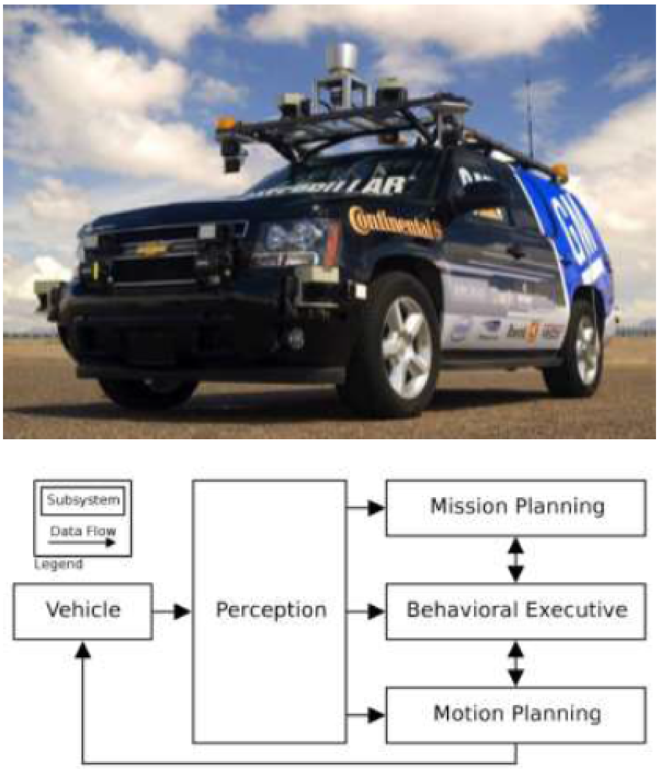
Subsystem
Data Flow
Legend
Vehicle
Perception
Mission Planning
Behavioral Executive
Motion Planning

Figure 3-3: Boss from Carnegie Mellon University system architecture in the vehicle.[6]

The advantages of using MPC:

- The ability to handle unstable, time variable, non-minimum phase systems,
- Built in feed-forward control to handle disturbances in the processes,
- Enhanced tuning features to achieve the best response including transient responses,
- Robustness feature with the uncertainties in the nonlinear systems
- Scope to include nonlinear equations for a nonlinear problem formulation
-

3-2-1 Generic Model Predictive Control problem

The theory of model predictive control is discussed in detail to highlight working principle. Model predictive control or receding horizon control is a method to generate the required control inputs are calculated at each sampling instance k, using the current state as initial conditions to solve a finite optimal control problem. The number of prediction steps for the control input are defined by the prediction horizon. The optimization process rendering the resulting optimal control solution is used on the system. This control input is implemented for the first step, and the sequence of control inputs is calculated by repeating the optimization process at the next sampling instance. This includes the important feedback mechanism that is built into the controllers. A brief introduction to the generic formulation is given and the resulting nonlinear model predictive control problem in formulated. For a given discrete system,

$$x_{k+1} = Ax_k + Bu_k \tag{3-1}$$

where $x_k \in \mathfrak{Re}^n$ and $u_k \in \mathfrak{Re}^m$ are the state and control inputs subject to constraints,

$$x_k \in \chi, \quad u_k \in U, \quad \forall k \geq 0$$

where $chi \in \mathfrak{Re}^n$ and $U \in \mathfrak{Re}^m$ represent states and control inputs. The normal regulation problem in control can be applied by deriving the solution to the optimal control problem at the time instance k.

$$\begin{aligned} & \min_{u_{k+i|k}} p(x_{k+N|k}) + \sum_{i=0}^{N-1} l(x_{k+i|k}, u_{k+i|k}) \\ s.t \quad & x_{k+i+1|k} = Ax_{x_{k+i|k}} + Bu_{k+i|k}, \quad i = 0, \ddots, N-1 \\ & x_{k+i|k} \in \chi, \quad u_{k+i|k} \in U, \quad i = 0, \ddots, N-1 \\ & x_{k+N|k} \in \chi_k \\ & x_{k|k} = x_k \end{aligned} \tag{3-2}$$

The MPC control problem accommodates, terminal cost $p(x_{k+N|k})$ and state cost $l(x_{k+i|k})$ terms. At N^{th} prediction step, the state constraint is a terminal cost. This feature is designed to ensure stable controller operation. The final constraint, on the initial condition is the current state of the system. The $k+1^{th}$ value of the state x, is obtained from the prediction model that is used in the MPC controller. Control inputs are calculated over the prediction horizon, from the current state to the $N-1^{th}$ term. The control input $u_{k|k}$ is applied to the plant at time k and the optimization problem of optimal control problem is solved again at time $k+1$ with initial state $xk+1|k+1 = x_{k+1}$.

The important points of the optimization problem is the formulation of the convex optimal cost function and the relevant constraints. Linear and quadratic formulations are feasible using the 1-norm and 2-norm respectively. In this thesis, quadratic formulations are used in structuring the optimal control problem. The property of the LTI system with respect to time invariance can be exploited to reformulate the optimal control problem in the 2-norm structure.

$$\begin{aligned} &\min_{u_i} x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \\ s.t \quad & x_{i+1} = Ax_i + Bu_i, \quad i = 0, \ddots, N-1 \\ & x_i \in \chi, \quad u_i \in U, \quad i = 0, \ddots, N-1 \\ & x_N \in \chi_k \\ & x_0 = x_k \end{aligned} \tag{3-3}$$

The tuning parameters in the MPC problem are weighting matrices Q, R and the prediction horizon. The matrix Q is used to tune the states of the system whereas the matrix R is used to tune the control gains. Hence they form the tuning parameters along with the prediction horizon N. Tuning of these parameters are described in section ..... Now that a brief explanation of MPC technique is provided, the constraints and MPC formulations for our application are elaborated in the next chapters.

Model Predictive Controller tuning

The development of control systems to satisfy the control objectives of the system requires tuning. Stable and conforming behaviour can be achieved by tuning the parameters. The importance is tuning is paramount as the even the correct implementation might not render feasible results. Discussing the tuning features available with MPC,

Tuning parameters Diagonal matrices Q and R can be used to weigh the system state matrix and the control inputs respectively. Each individual value on the diagonal corresponds to the individual states of the system such as position, or velocity, and control inputs such as steering, or voltage for actuators. The response of the system that is too slow can be influenced by adding high weighting values in the Q matrix, whereas the control gains are damped with high weighing values in the R matrix. The important aspect of finding the optimal trade-off is key to any controller behaviour with opposing objectives. **Stability of MPC controller**

The tuning parameters enforce a limited horizon on the MPC problem and this affects the stability of the controllers. The different techniques to tackle this problem is to set an infinite horizon, impose end point constraints, terminal cost function, or terminal constraints.

The terminal cost, prediction horizon and constraints can be tuned to obtain a stable controller. The prediction horizon is limited by the computational cost in the chosen system. The weights on the cost function can be set to obtain a stabilizing solution. Hard constraints on the reachable sets might backfire and render an infeasible controller. Thus, with the current system, the weights on the cost function and the prediction horizon are chosen parameters to render stability.

Automotive application The application to vehicle trajectory generation and vehicle dynamics control have the issues with finding the right trade-off. The trajectory generation must ensure the objectives of motion planning finding smooth and collision free trajectories. The tracking controller has to be able to follow the reference but with constraints placed to avoid aggressive gains. The tuning of the controller might seem appropriate with respect to the individual controllers but due to the nonlinear nature of the system, system response analysis is not straightforward. This is also reflected in the difference in behaviour with the individual controller and the hierarchical controller implemented in this thesis. The use of nonlinear model predictive control allows to incorporate the nonlinear dynamics model equations directly into the problem formulation.

Nonlinear Model predictive control

Consider a discrete dynamical system represented using nonlinear equations, where $x_k \in \mathfrak{Re}^{n_x}$ is the differential state, $z_k \in \mathfrak{Re}^{n_z}$is the algebraic state which is uniquely determined by the equality constraint once x_k and u_k are fixed, and $u_k \in \mathfrak{Re}^{n_u}$ is the control at time step k. The functions f_k and g_k are assumed to be twice differentiable.

$$x_{k+1} = f_k(x_k, z_k, u_k) g_k(x_k, z_k, u_k) = 0 \tag{3-4}$$

The NMPC problem can then be given by where N is the prediction horizon

$$\begin{aligned} \min_{(x,z,u)} \sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) + E(x_N) \\ \text{subject to} \\ x_0 - \bar{x_0} = 0 \\ x_{i+1} - f_i(x_i, z_i, u_i) = 0 \\ g_i(x_i, z_i, u_i) = 0 \\ h_i(x_i, z_i, u_i) \leq 0 \\ r(x_N) \leq 0 \end{aligned} \tag{3-5}$$

The differential state vector, algebraic state vector and the control vector are the decision variables or free variables considered at all discrete time points. The inclusion of the state vector, and control variations in the optimization problem, accommodation of equality and inequality constraints allow for the formulation of the controller problems in the subsequent chapters.

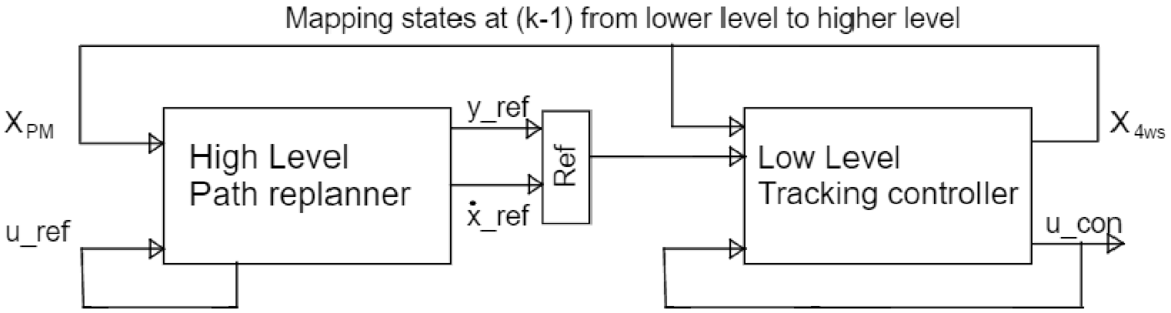

Figure 3-4: Controller architecture for trajectory generation and tracking.

3-3 Hierarchical control system

The introduction to system architecture for the problem of trajectory generation and tracking establishes a baseline to implement a hierarchical controller. The model predictive control approach and the theory provides motivation for the chosen approach for our application. The implementation scheme is discussed to elaborate the challenges with respect to the controller development.

The illustration in figure 3-4 illustrates the working of the implemented controller. The higher level controller is responsible for generating a feasible trajectory. The lower level controller is the tracking controller that receives the lateral position and velocity references from the higher level controller.

The interaction between the controllers occurs when the lower level controller passes the system states to the higher level controller. The two state vectors are x_4ws and x_PM for the four wheel vehicle and the point mass model respectively. The construction allows for mapping of the states as defined by the point mass model. This allows using the same value for the prediction horizon in the both the controllers.

$$\begin{bmatrix} \dot{y} \\ \dot{x} \\ \psi \\ \dot{\psi} \\ Y \\ X \end{bmatrix} \Rightarrow \begin{bmatrix} y \\ v_x \\ V_y \\ \Delta x_j \end{bmatrix} \tag{3-6}$$

Another important feedback value for the controllers are the control inputs generated. The higher level planner has control inputs for lateral acceleration (a_y) and longitudinal acceleration (a_x). The values a time instance $(k-1)$ in u_ref is sent as feedback in the higher level controller to produce smooth reference trajectories. The lower level controllers produce the control input values for steering angle and braking forces. The feedback of these values at time instance $k-1$ is crucial for the calculation of the Δu formulation.

The states passed from the lower level to the higher level is position X. This creates a dilemma between performance and theoretically proving the efficacy of the trajectory generation controller with the position feedback alone. The motivation behind passing the state X and not the velocity $\dot{x}$ or lateral position Y is to satisfy the objective of creating smooth reference trajectories. The velocity and position values passed back to the higher level controller would act as the initial conditions for the new optimization iteration. This would result in harsh manoeuvres to meet the original lane centre reference.

3-4 Concluding remarks

The system architecture of control systems in autonomous vehicles reveals the importance of different modules relating to perception, planning and control. These modules are arranged in different approaches and the hybrid architecture combining the deliberative and reactive paradigm occur in most autonomous systems. The motion planner and controller are explored in detail and defined with respect to the nonholonomic on-road vehicles. The different operational vehicles are discussed to support the general architecture used while allowing the scope to introduce MPC as a viable solution. The theory of model predictive control is explored in detail to aid in the problem formulation of future sections. Finally, the implementation scheme of the hierarchical control system is illustrated in section 3-3. The conceptual implementation of the controllers helps us to achieve the required hierarchical setup for the trajectory generation and tracking problem.

The following chapters are dedicated to explain the implementation of the individual controllers for the tasks of trajectory generation and trajectory tracking. The controller development with constraints, optimal control problem formulation and the approach to obtain the solution are detailed.

Chapter 4

Model predictive control methods for trajectory generation

In this chapter the methods for the controller for trajectory generation is discussed. The control objectives and practical constraints for the controller are defined in section 4-1, and 4-2 respectively, to guide the controller development. The optimal control problem is formulated in section 4-3 to form the core of the optimization based controller. Consequently, the important part of the trajectory generation operation is elaborated with the collision avoidance constraints. The approaches to formulate the collision avoidance constraints offer insight into the working of the MPC based trajectory generator. Section 4-5 further explains the optimization algorithm and choice of non-linear programming solver for the application.

4-1 Control Objectives

Trajectory generation with respect to automotive systems primarily deals with the safe trajectory generation while satisfying the vehicle and environment constraints. The previously described system architecture allows us to formulate the requirements of the controller. The following control objectives were defined to form a framework for the controller development. The controller needs to be able to,

- Generate safe trajectory reference in the motion planner level
- Maintain vehicle position with respect to the road boundary and surrounding vehicles
- Maintain the vehicle velocity as per the highway cruising speed.

The objectives broadly define the requirements of the higher level controller. The scenario representative of the highway environment would require cars travelling with the road lane boundaries. The condition for a two lane highway is considered for the thesis study. This is representative of the majority of the highway roads. The vehicle also needs to travel at a speed normally maintained on highway roads. Lastly, the important objective to generate safe collision free trajectory references for the vehicle to follow.

The automotive application brings along several physical constraints due to the nature of the system. The factors of safety, comfort, actuator limits, legal driving limits, and driving conditions have to be considered to have a realistic representation in the chosen scenario. As discussed in the previous section, the higher level planner uses a point mass model of the vehicle. The practical constraints are assigned appropriately.

4-1-1 Practical constraints

Position

The position of the vehicle on the road must comply with the road boundaries. Lane centre is chosen as the trivial choice to traverse the right lane of the highway. The keep right directive is followed in more two thirds of the countries in the world. The lateral position of the vehicle is monitored and constrained. The lane width is chosen as $3.2[m]$ averaging the lane width variation on highways. From this value the lane centre can be chosen accordingly. The CoG of the vehicle must align with the lane centre and has to be accounted in the constraint formulation.

Velocity

The speed objective is bounded by the legal speed limit taken at $120[kmph]$ limiting the longitudinal velocity to $33[m/s]$. A lower bound is set at 0 to avoid backward movement of the vehicle on the highway. This also further defines the highway scenario as a one way driving road with no oncoming traffic. The maximum velocity is kept slightly below the speed limit in a conservative formulation. Vehicles are capable of travelling faster and there are also many roads where the speed limit is higher at $130[kmph]$, but for the purposes of this study a hard constraint is set between 0 and $30[m/s]$.

Lateral velocity is an important consideration due to the implications on passenger comfort. The speed at which the vehicle position changes during a lane change scenario has effect on the values of lateral velocity. Vehicle safety is also considered in the constraint as high values of lateral velocity causes unstable vehicle behaviour. The conservative values $\pm 3[m/s]$ was used as a box constraint on the lateral velocity.

Acceleration

Acceleration limits mainly satisfy the consideration on passenger comfort and vehicle actuator limits in real life. The studies on comfort, conclusively point to acceleration or jerk metrics. Using bounds on the acceleration in the longitudinal direction can indicate the maximum comfortable acceleration and threshold braking limit. Braking limit is important here as $-4[m/s^2]$ indicates the threshold for comfortable braking in case a sudden stop is required. Maximum acceleration is set for $1[m/s^2]$.

The acceleration in the lateral direction is also constrained for comfortable movement during lane change manoeuvres. Here, $\pm 0.5[m/s^2]$ is set for the upper and lower bounds on lateral acceleration.

Non-holonomic constraints

The relevant constraints implemented reflect the motion planner stage where a point mass model is used. The model used does not satisfy the non-holonomic vehicle constraints. The case where longitudinal acceleration is zero in the presence of lateral acceleration produces a strict lateral movement. This is unrealistic for the vehicle. But there is a small angle assumption made with respect to the vehicle side slip, and accordingly constrained lateral and longitudinal velocity values using the equation $\beta = \arctan \frac{v_y}{v_x}$ where $\beta = 10\,\text{deg}$. The constraint $-0.17v_x \leq v_y \leq 0.17v_x$ is used to enforce the assumption and compensate for holonomic pure rolling property of the point mass model. This ensures a feasible path generated in the global co-ordinate system for the vehicle to follow.

4-2 Collision avoidance constraints

The problem of collision avoidance occurs concurrently with trajectory generation because the significant objective of safety. This is emphasized more in the automotive application due to stringent safety measures. Highway driving is also characteristic of higher speeds $100 - 130$*[kmph]* where directly correlates to an increase in fatality rates.

The traditional methods reviewed for the application such as potential fields methods, grid/-graph based methods are beneficial with the robotics applications but fail to provide a scalable solution. The force field characteristic defined in the potential fields method, defines a repulsive force for obstacles, but is severely limited with closely spaced obstacles. The grid/graph based approaches can provide feasible solutions but accuracy, grid resolution and computational time are linearly related. The choice of using optimization based trajectory generation carries the additional advantage of defining collision avoidance constraints in the control problem.

The formulation of the collision avoidance constraints is a crucial step to generate collision free trajectories. The behaviour of the trajectory generator is governed by the these methods. The performance index of the optimal control problem is formulated to achieve the best desired behaviour based on the objectives. The optimal control problem formulation to achieve the best performance with the control objectives and the following practical constraints respect the physical and design constraints of the vehicle. The additional constraints required to achieve safe collision free trajectories are added to the optimization problem formulation. In this section, the focus is placed on formulating the feasible region for the controlled vehicle to travel in the driving scenario. The regions critical for the safety of the vehicle, road boundaries and surrounding vehicles are avoided. This creates a non convex optimization problem with respect to the safe search space for trajectory generation. The following section is designed to simplify the optimization problem to a quadratic programming (QP) framework.

The simplest formulation of collision avoidance in the longitudinal direction is the time-to-collision (TTC) which uses the relative position and velocity between the controlled vehicle and surrounding vehicle in the same lane. This formulation is infeasible when road curvature exists and characteristic drawback of the method. Nonetheless, it allows for a simple calculation of the relative distance between two vehicles on the road, and studies [65], indicate the optimal time headway time. Time headway parameter determines the safety margins, and significantly important with collision avoidance. The main assumption of the collision avoidance constraints is the sensing system is perfectly capable of providing the position and velocity of all the vehicles involved.

4-2-1 Longitudinal distance from surrounding traffic

The relative distance between vehicles on the road is an important measurement to use in the safe operation of the controller. The relative distance is a benchmark safety feature that enables linear collision avoidance systems to enforce a hard constraint. The longitudinal control of cruise control systems employ the commonly used time-to-collision (TTC) calculation to assess the likelihood of the crash.

$$Time - to - collision = \frac{d_{measured}}{v_{relatative}} TTC = \frac{x_s - x_c}{v_s - v_c} \tag{4-1}$$

where, x_{star} is position of controlled and surrounding vehicle. $v_{x_{star}}$ is velocity of controlled and surrounding vehicle, for $\star = [c, s]$

The time measure in seconds gives the probability of crash if $TTC \leq 2$. The subjects of automotive crash safety heavily depend on this measure to assign time scale to the Haddon matrix [66]. The pre-crash and crash time frame activates the relevant safety measures for warning or injury prevention.

Minimum distance between vehicles

The basic longitudinal safety constraint with respect to the crash avoidance is minimum distance between vehicles which depends on vehicle velocity and safe minimum headway time. Setting a minimum longitudinal distance between vehicles ensures a safety margin, and in our application, the width of the surrounding vehicle is set at $5[m]$ to ensure an adequate safety margin. This is a crude approach to implement due to the limited scalability to all driving conditions but contributes to the comfort objective by reducing the chances of close passing manoeuvres.

$$d_{desired} = d_0 + v_{cx} t_{headway} \tag{4-2}$$

where d_0 is the minimum distance at standstill $(6m)$ $t_h w$ is the time headway which is between $[0.5, 5]$ can be set to 2 seconds for safety measure.

4-2-2 Forward collision avoidance constraints

The forward collision avoidance constraints are designed to avoid collision with the front side of the controlled vehicle. The optimization search space is designed such that the infeasible non-convex region does not affect the quadratic programming (QP) formulation of the optimal

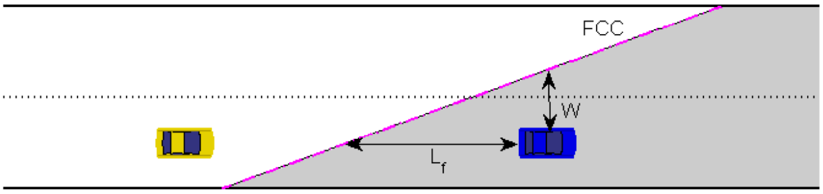

Figure 4-1: Forward collision avoidance constraint [7].

control problem. As shown in figure 4-1, the optimization space is maintained as a convex problem. This allows for the introduction of slack variables as seen in [7], which are

$$\frac{\Delta x_j}{L_f} \pm \frac{\Delta y_j}{W} + v\epsilon_{x_{jf}} + \frac{\epsilon_{y_{jf}}}{\phi} + \epsilon_{jf} \tag{4-3}$$

where,

$$\begin{aligned}\Delta y_j &= y_{s_i} - y;\\ \epsilon_{y_j} &= -\Delta y_i - \sigma;\\ \epsilon_{x_{jf}} &\geq 0;\\ \epsilon_{j_f} &\geq 0;\end{aligned} \tag{4-4}$$

The terms used in the collision avoidance constraints entail terms for the longitudinal, lateral directions along with slack variables to improve the performance. The relative longitudinal distance Δx_j and distance to the vehicle in front L_f, where $L_f = v_x\theta_f + L_c$ is defined using vehicle velocity, time headway and surrounding vehicle length. From the previous section, the time headway is chosen to be $3[s]$ whereas margin is included for the length of the vehicle at $5m$. The relative lateral distance Δy_j is intricately related to the road boundary condition using $W = 0.5W_l + W_c$ where lane width W_l, surrounding vehicle width W_c are considered. The last three terms include slack variables with respect to the behaviour in the longitudinal direction, lateral direction and the overall equation.

In depth analysis of the defined slack variables was carried out to understand the boundary regions that defined the feasible search space for the optimization problem. The terms ϵ_{jf} and ϵ_{j_r} were penalty terms placed to obtain feasible trajectories. The fourth term, $\frac{\epsilon_{y_{jf}}}{\phi}$ was important to relax the constraint after the controlled vehicle has passed the surrounding vehicle. This term also relaxes the constraint when the relative longitudinal distance is too large, avoiding lane change manoeuvres too quickly.

4-2-3 Rear collision avoidance constraints

The collision avoidance terms for the rear side prevent collisions with the rear end of the vehicle. This terms are similar to the forward collision constraints. The main distinction is the $L_r = v_x\theta_r + L_c$ that defines the safe time headway at $2[s]$. This parameter is determined at the start of each optimization cycle.

$$\frac{\Delta x_j}{L_r} \pm \frac{\Delta y_j}{W} + v\epsilon_{x_{jr}} + \frac{\epsilon_{y_{jr}}}{\phi} + \epsilon_{jr} \tag{4-5}$$

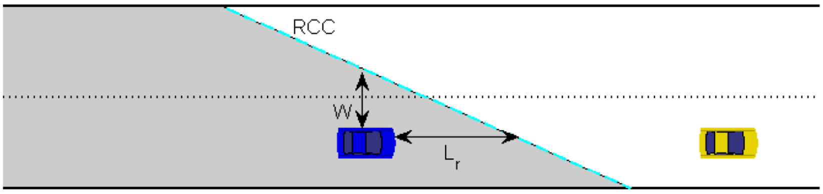

Figure 4-2: Rear collision avoidance constraint [7].

where,

$$\begin{aligned} \Delta y_j &= y_{s_i} - y; \\ \epsilon_{y_j} &= -\Delta y_i - \sigma; \\ \epsilon_{xrf} &\leq 0; \\ \epsilon_{j_r} &\leq 0; \end{aligned} \tag{4-6}$$

4-2-4 Limitations

The limitations of this approach extend to the inability to plan overtake manoeuvres irrespective of the horizon length. The collision avoidance in the forward direction is works well wit this formulation but limits the tuning scope of the of the prediction horizon.

A general problem with slack variables âĂŞ they are a mathematical way of turning inequalities into equalities. A common problem in many algorithms that have a stochastic working (genetic algorithm) or deterministic methods that are not allowed to converge. The current application is interrupted with maximum iteration limit and accept the currently best solution. This causes the slack variables to add further complexity. Here, the penalized variable ϵ_{j_f} is heavily penalized in the cost function in order to only affect the forward collision condition when no feasible solution exists. The overall case does not allowed to converge, which often creates suboptimal solutions.

4-3 Optimal control problem formulation for trajectory generation

The control objectives and practical constraints defined give an guideline for the objective function and design constraints. The collision avoidance terms, define the optimization search space for planning safe collision free trajectories. The cost function of the optimal control problem reflects the main control objectives with respect to desired velocity and lateral position. The following equation is the OCP problem formulation,

$$\min \sum_{k=0}^{H-1} \|V_x - V_{x,des}\|_2^{\alpha} + \|Y - Y_{ref}\|_2^{\kappa} + \|V_y\|_2^{\gamma} + \|a_x\|_2^{\vartheta} + \|a_y\|_2^{\vartheta_1}$$
$$\|\epsilon_{j_f}\|_2^{A} + \|\epsilon_{j_r}\|_2^{B} + \|\epsilon_{j2_f}\|_2^{C} + \|\epsilon_{j2_r}\|_2^{D}$$

$$\begin{aligned} \text{s.t} \quad -0.17v_x &\leq v_y \leq 0.17v_x \\ V_{x_{min}} &\leq V_x(t) \leq V_{x_{max}} \\ V_{y_{min}} &\leq V_y(t) \leq V_{y_{max}} \\ a_{x_{min}} &\leq a_x(t) \leq a_{x_{max}} \\ a_{y_{min}} &\leq a_y(t) \leq a_{y_{max}} \\ Y_{min} &\leq Y(t) \leq Y_{max} \\ \epsilon y_{j_f} &\geq 0 \\ \epsilon y_{j_r} &\leq 0 \\ \epsilon x_{j_f} &\geq 0 \\ \epsilon x_{j_r} &\leq 0 \\ \epsilon j_f &\geq 0 \\ \epsilon j_r &\leq 0 \end{aligned} \tag{4-7}$$

$$\begin{aligned} \xi &= [\Delta X_1 \quad Y \quad \dot{x} \quad \dot{y} \quad X \quad \Delta Y_1 \quad \Delta X_2 \quad \Delta Y_2 \quad X_2]^T \\ u &= [a_x \quad a_y] \end{aligned}$$

4-3-1 Prediction horizon

The prediction horizon determines the span of the system evolution for implementing the control gains. It is an important tuning parameter in the model predictive controller. The prediction horizon for path planning, would benefit from a long horizon to be able to cope with the overtake manoeuvres. However, the tradeoff between computational load and controller performance that can be scaled to real systems. The frequency of the system, sampling time and average time required to execute driving manoeuvres are factors that can be used to choose an appropriate prediction horizon. The sampling time of $0.1[s]$ is sufficient to simulated for a prediction horizon of $N = 30$

Different simulation length and sampling times were employed during the simulations to render the best possible trajectories.

4-3-2 Tuning weights

Here the weighting factors can be used to affect the performance of the different objective terms in the cost function. The weights play an important role in controller tuning with respect to the objectives and design constraints. The weights α, κ are associated with the most important objectives of the controller. The following terms, $\gamma, \vartheta, \vartheta_1$ are included to represent the passenger comfort. Finally, A, B, C, D are the weighting factors for the collision avoidance terms detailed in the following section.

4-3-3 Cost function

The cost function was analysed in the optimal control problem during the debugging phase of the controller. The contributing costs of each of the objective terms were analysed to locate the source of errors with respect to an infeasible trajectory.

The main objectives to reach desired velocity and keep lane centre were heavily influenced by the non-linearities in the full vehicle model. The controllers developed are intrinsically linked and the interaction between the controllers passing the states was a significant factor in deciding the cost function.

The objective term for lateral velocity was considered to increase the comfort of the trajectory generated. The choice to include lateral velocity was made as opposed to include the acceleration control inputs generated by the higher level controller.

Finally, the presence of several slack variables in the collision avoidance constraints, demanded for an optimal tradeoff between the slack variables themselves. The approach followed with the current implementation, is inherently prone to fine tuning the parameters. An auto tuning formulation was made with the need to find the optimal slack variable values and to find out the tradeoff between them.

4-3-4 Equality and box constraints

The equality constraints for the optimal control problem entail the vehicle point mass model relations. The relative longitudinal distance between the controlled vehicle and the surrounding vehicles, lateral velocity, longitudinal velocity, relative longitudinal distance between the controlled vehicle and the surrounding vehicle, and velocities of the surrounding vehicles are considered.

$$\begin{aligned}
\Delta\dot{X}_1 &= vs1 - vx \\
\Delta\dot{X}_2 &= vs2 - vx \\
\dot{y} &= vy \\
\dot{v_x} &= ax \\
\dot{v_y} &= ay \\
\dot{X} &= vx \\
\Delta\dot{Y}_1 &= -vy \\
\Delta\dot{Y}_2 &= -vy; \\
\dot{X}_1 &= vs1; \\
\dot{X}_2 &= vs2;
\end{aligned} \tag{4-8}$$

The information about the surrounding vehicles is included in the system states due to the linkage between the trajectory generator and the surrounding vehicles for collision avoidance objectives.

Box constraints set on the velocity, acceleration, and jerk satisfy the design and comfortable operational constraints of the vehicle.

4-4 Constrained quadratic optimization problem

The optimization problem that renders the solution to the MPC controller is a significant topic. The efficiency of optimal control inputs required to achieve the desired control objectives are influenced by the solution approach. Traditionally, model predictive control developed in the process industry with characteristic slow dynamics. The computational capacity currently available allows for convergence even with systems with faster dynamics. Another contributing factor are intelligent strategies developed to solve the optimization problems. The efficiency of the solution directly correlates to increased MPC feasibility in automotive systems.

The general form of the quadratic constrained non-linear programming problem is defined as,

$$\begin{aligned} \min_x f(x) &= \frac{1}{2}x^T F x + d^T x \\ s.t \quad & x_L \le x \le x_U \\ & b_L \le Ax \le b_U \\ & c_L \le c(x) \le c_U \end{aligned} \tag{4-9}$$

where, $x, x_L, x_U, d \in \Re^n, F \in \Re^{n*n}, A \in \Re^{m_1*n}, b_L, b_U \in \Re^{m_1}, c_L, c(x), c_U \in \Re^{m_2}$ and represent the nonliner constraint vectors, gradient matrix or Hessian to Langrangian function depending on the problem type.

The structure of the problem formulation was referred during the controller development and coding part to adhere to the correct syntax of the optimization suite. Here, the formulation example for a constrained QP problem renders a convex optimization problem. These formulations provide the advantage of a converging solution. The methods available to solve the QP problem involve active set and interior point methods. The optimization problem is not trivial with our nonlinear system. In that line of research, geometric optimization methods are used, which are potential future recommendations for implementation.

Sequential quadratic programming

The SQP (sequential quadratic programming) **SQP!** (**SQP!**) methods split the problem into sequence of subsets to find the optimal solution thereby reducing the augmented Lagrangians. Here, the major iteration representing the number of subsets and the minor iterations that determine active constraints in the subsets.

The method to solve the NLP problem by making a quadratic approximation of the Lagrangian function and linear approximation of the constraints. The resulting subproblem is

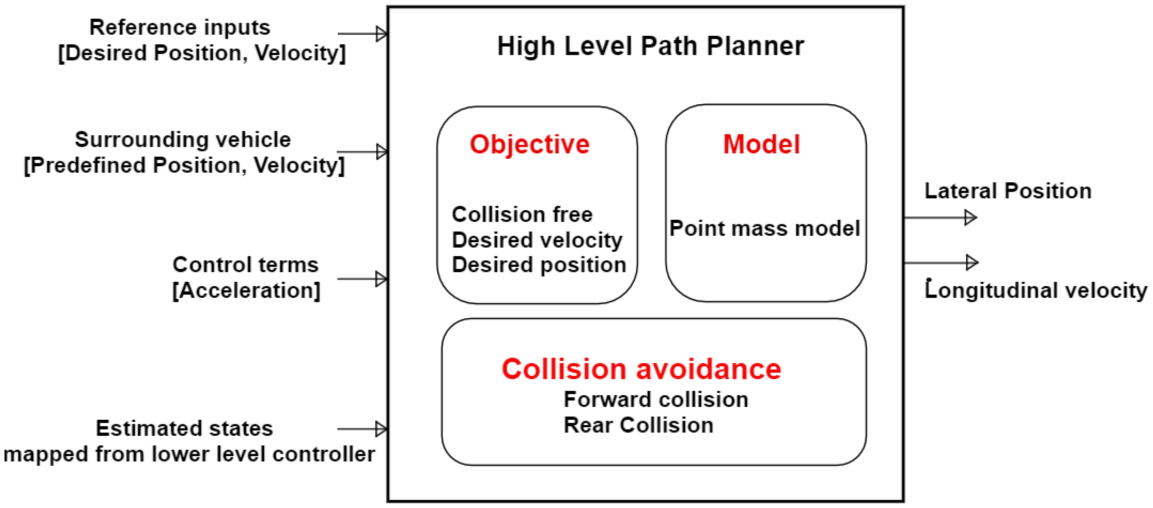
Reference inputs
[Desired Position, Velocity]
Surrounding vehicle
[Predefined Position, Velocity]
Control terms
[Acceleration]
Estimated states
mapped from lower level controller
High Level Path Planner
Objective
Collision free
Desired velocity
Desired position
Model
Point mass model
Collision avoidance
Forward collision
Rear Collision
Lateral Position
Longitudinal velocity

Figure 4-3: Higher level control block.

a Quadratic Programming (QP) problem given as

$$\begin{aligned} &\min_{y} F_k^{QP}(y) \\ \text{subject to} &G(y_k) + \nabla G(y_k)^T (y - y_k) = 0 \\ &H(y_k) + \nabla H(y_k)^T (y - y_k) = 0 \end{aligned} \tag{4-10}$$

If the hessian matrix is positive semidefinite, then the resulting QP problem is convex and solutions can be found reliably. Other variants of SQP are available, which approximate the Hessian in the objective function giving rise to the quasi-newton methods. Quasi - Newton methods are shown to converge superlinearly [67]. The most widely used approximation of the Hessian is the Broyden-Fletcher-Goldfarb-Shanno method and the Davidon-Fletcher-Powell method. The solution of this QP subproblem, is then used to find a new search direction and the process is repeated until the solution converges. A good survey of SQP variants is given in [32].

4-4-1 Implementation and Optimization solvers

The block illustration in figure 4-3 of the conceptual controller required an optimization environment using the MATLAB interface.

The implementation scheme in the previous chapter motivated the choice to use TOMLAB optimization suite for our application. The quadratic programming problem for the higher level trajectory generation was solved using SNOPT solver where the problem in the form of a quadratic equation with non-linear constraints. SNOPT solving a qpcon problem was suitable with the optimization suite supplying the objective derivatives in the form of gradient or Hessian matrices.

The main consideration for the solvers is the ability to scale systems to real time implementation. The state of the art solver, seDumi, ACADO, NPSOL, SNOPT were considered for the best approach. The solvers have an added advantage to use the cost function, constraints, and initial conditions of the QP problem to generate a *.cpp* file. This can be a bridging tool for hardware implementation.

The issues with optimal solver were experienced during the implementation phase. The seDumi solver operating over semi-definite cones was unable to provide converging solutions. In parallel, the ACADOToolkit as used for the integration routines. This approach yielded suboptimal results and a bottleneck was identified with the compatibility when integrated into MATLAB interface. Consequently, the seDumi solver was removed to favour a standalone MPC controller defined in the ACADOToolkit.

ACADOToolkit followed a three layered approach to solving the system differential equation, defining the optimal control problem with constraints and finally using the optimization algorithm with initialization and integrator settings. In this approach, the MPC controller comprised of a on-line control law to obtain the control inputs. The choice to setup up a dynamic systems was forgone despite the ability to setup different system equations for the simulation and optimization. The attempt to exploit this setup structure to formulate a hierarchical controller was unsuccessful.

The implementation of TOMLAB was not chosen as a last resort due to the commercial licensing limitation. This optimization offers multitude of solver for the nonlinear problems but due to the limited time constraints of a DEMO license, this superior choice was excluded in the initial approaches. The choice of optimization solver was introduced as SNOPT. However, for the purposes of nonlinear programming, two solvers namely NPSOL and SNOPT were evaluated.

SNOPT [68] for large scale constrained optimization is well suited for general nonlinear optimization problems. The structure of the toolbox allows for special maintenance and building tools while using the toolbox to test the performance of the controllers. Exit Flags were monitored to find out the status of the iteration steps ie the quadratic programming problems that were solved at every iteration. Exit Flag with value 10 shows up in the case of an infeasible solution. The general constraints for the general case $A_x + s = 0$, were not satisfied with respect to A and s. Minor constraint violations are allowed as long as one of the components in A and s are bounded. This makes it a problem with nonlinear systems and constraints as source of infeasibility is hard to determine. The last resort is to enter an elastic programming mode but this case will relax the nonlinear hard constraints and may not be a feasible optimization problem.

In the case, the nonlinear problem could not find a converging results, the toolbox finds the closest match to solve the problem linearizing the nonlinear constraints in the local region. If possible, nonlinear constraint formulation must satisfy feasible solutions in locally linear regions.

4-5 Concluding remarks

This chapter covered the important aspect of the higher level trajectory generating controller. The prerequisites for controller development were defined in the form of control objectives, practical environment design and operational constraints. Consequently, the essential working component of the higher level planner were introduced in section 4-2 with forward and rear collision avoidance constraints. The slack variables that significantly influence the behaviour of the controllers are introduced. The optimal control problem with all the contributing cost terms, vehicle and boundary constraints, state vector and input vectors are stated. The parameters that influence tuning the relevant terms is discussed. The chapter ends with the details of the QP problem and the optimization suite and nonlinear solver SNOPT used to obtain the optimal solution.

Chapter 5

Model Predictive control for trajectory tracking

This chapter details the working of the lower level MPC controller for trajectory tracking. The control objectives are defined in section 5-1. The following section details the practical constraints that have to be considered with respect to the vehicle actuators for steering and braking. Section 5-3 describes the optimal control problem for the lower level controller. The controller design is discusses in detail with the choice of acceleration and braking logic in section 5-4. The final section 5-5 is used to illustrate simulation results for the lower level controller.

5-1 Tracking control

The tracking control problem is one of the significant types in system control along with regulation problems. Tracking control poses additional challenges when applied to the constrained non-linear systems which represent most of the industrial processes. The real life applications have characteristic non-linear properties, dynamic references, and constrained actuators. The automotive application where the trajectory tracking feature is relevant, is challenging due to the non-linear vehicle dynamics and complex environment.

In general, the tracking deals with steering the system state to a stable point by solving an integral quadratic cost function. The supplied reference signal has to be followed as closely as possible by the actual system. Asymptotic tracking is ideally desired where $\lim_{t\to\inf}[r-z(t)] = 0$ drives the system under stability and detectability conditions of the system. Now, given the scenario with disturbances, the problem increases in complexity to reach a stable point, while rejecting disturbances. In the problem formulation for a disturbance rejection problem, process noise, measurement noise and system load can contribute to the disturbances acting on the system. Manipulating the control formulation, and equating to the system equation allows for closed loop control of the system and the error trajectories can be analysed.

Distubance rejection as introduced is an augmented trajectory tracking problem, th

Since the controller of choice is built to inherently handle disturbances, MPC is a fitting choice for the given problem of trajectory tracking.

The terminology used in literature with path planning, trajectory planning, tracking and motion planning are used interchangeably. Path following and trajectory tracking are two distinct tasks where path following is a geometric link between start and terminal points which avoids all possible collisions whereas trajectory planning is the on-line calculation of the vehicle states from one instance while meeting all the constraints. For the rest of this study, the tracking controller deals with following the reference trajectory with the set tolerances.

5-2 Control Objectives

The lower controller for trajectory tracking has to stabilize the system states with respect to the reference value. The reference trajectory generated by the higher level controller represents a collision free trajectory along the highway. The trajectory can represent a number of scenarios, such as highway cruising, static obstacle avoidance, car following, overtake manoeuvre and evasive manoeuvres. The lower level controller has to track the supplied trajectory with minimal error.

- Track reference generated by the motion planner.
- Generate comfortable control (u) and deviation of control (Δu) inputs.

The safety and comfort aspects of driving are considered with respect to the formulation of the cost function of the optimal control problem. The cost term minimizing the error in tracking the reference trajectory bounded by physical and environment constraints, is designed to ensure safety. The second objective is used to feedback output information and incremental output information to the model to increase the performance of the controllers.

Input-output (IO) models and the incremental input-output (IIO) models are used to build receding horizon controllers with improved performance.

5-2-1 Practical constraints

5-2-2 Steering angle for lower level controller

The steering angle output obtained from the controller is the gain obtained to travel the path of the reference trajectory. This actuator is the common controlled choice in lane change controllers. The maximum limit of the steering wheel turn depends the vehicle but block constraints can be placed on he steering angle and steering rate to ensure smooth performance and comfortable ride. The steering control action must be designed similar to a human controlled vehicle in the highway environment. The constraint chosen at $\pm 24[deg]$ on the steering angle is lower with respect to the maximum actuator limit around 60 deg. Aggressive response within the bounds of the steering angle is possible and constrained steering rate provides better comfort during vehicle control. The steering rate was constrained at $\pm 9[deg]$, to avoid jerky steering actions.

5-2-3 Lateral tire forces

The lateral tire forces are generated to control the braking via the brake torque. Yaw moment stabilization is the common approach with vehicle dynamics control, and performance can be evaluated. Tire forces are generated in the tire especially the contact patch. This is caused friction and slipping of the tire. The lateral and longitudinal tire forces with relation to slip are calculated in the modelling section. The brake torque distribution for the vehicle control at the individual wheels are estimated using the algorithm in controller design section. The complex model of the vehicle with the wheel render non integrable velocity constraints. The constraints arise from the condition where pure rolling occurs without slip. The constraints set on the front and rear lateral tire forces F_l, F_r were $\pm 15000[N]$. The constraints on the rate of change of lateral forces were also placed as a box constraint at $\pm 10000[N]$. This is expecting very capable actuators as the forces generated are much lower in real life. The constraints had to be relaxed to achieve better vehicle response.

5-3 Optimal control problem formulation

The lower level predictive controller must satisfy the previously defined control objectives and the constraints. The task of trajectory tracking is nothing but trivial due to the non-holonomoic nature of the on-road vehicle. Thus, the simple case of the traversing in a straight path without any obstacles is challenging. The ζ tracking cost term minimizes the error in tracking whereas the cost term for control inputs ensures smooth vehicle response.

$$
\begin{aligned}
\min \sum_{k=0}^{H-1} (\|\zeta_{t+k,t} - \zeta_{ref_{t+k,t}}\|_2^Q + \|\Delta u\|_2^S) \\
s.t \quad F_{l_{min}} &\leq F_l \leq F_{l_{max}} \\
F_{r_{min}} &\leq F_r \leq F_{r_{max}} \\
0 &\leq \dot{x} \\
Y_{min} &\leq Y \leq Y_{max} \\
\delta_{max} &\leq \delta \leq \delta_{max} \\
\dot{\delta} &\leq \dot{\delta} \leq \dot{\delta}
\end{aligned}
\tag{5-1}
$$

$$\xi = [\dot{y} \quad \dot{x} \quad \psi \quad \dot{\psi} \quad Y \quad X]$$

$$u = [\delta \quad F_l \quad F_r]$$

5-3-1 Prediction Horizon

The prediction horizon defined for both the controllers, the length of the system predicted by the controller. The following control horizon, implemented for a single step maintains the a constant control input over this step. The prediction horizon for this controller was chosen based on the length of the reference trajectory. This ensured the correct response with the high prediction horizon. The influence of the prediction horizon on the computation was lower with respect to the iteration limit factors. Therefore the sacrifice was made to include a high prediction horizon. In the case of the manoeuvre similar to double lane change, $H_p = 170$ was chosen.

5-3-2 Cost function and tuning parameters

The contributing objective terms minimizing the deviation from the reference state, control input and deviations in control input. The main tracking term $\zeta_{t+k,t} - \zeta_{ref_{t+k,t}}$, is the deviation of the state $\dot{x}, Y$ from the reference state vector $\dot{x}_{ref}, Y_{ref}$.

The second and third terms in the objective function u and Δu allows weighting large control gains and large deviations in control gains. This will satisfy the comfort factor for vehicles where abrupt control inputs, with respect to steering angle and braking are not desirable. The box constraints set on the control inputs, also add to the comfort factor. These limitations align with the physical and design constraints of the system. The important tuning variations considered here were to include only Δu term as opposed to using both control and control deviation cost terms in the objective function. This allowed for sharper tuning and lower tuning parameter values that can be influenced. The advantages of using only Δu values stood out as explained in the later subsection.

The weighting terms for the objectives Q, R, and S are essential in tuning an appropriate response from the system. Here, diagonal matrix values of $Q_{2,2}$, acting on the tracking objective weights the velocity and position terms respectively. The tuning of diagonal matrices $R_{3,3}$ and $S_{3,3}$ is challenging, as the control vectors $u = [\delta, F_l, F_r]$ and $\Delta u = [\dot{\delta}, \dot{F_l}, \dot{F_r}]$ that are included in the objective plays a significant role in smoothing the trajectory tracking feature. The parameter space for the tuning factors were influenced to obtain the optimal values for the lower level controller.

5-3-3 Δu formulation

The inclusion of the Δu formulation is to exploit the use of change in control input $\Delta u(k)$, as opposed to the control input $u(k)$. It is beneficial to minimize the steady state error mimicking an integral action. As explained in the feedback of control inputs in the controllers, the variation in control inputs is included in the state vector.

An additional constraint can be imposed on the Δu values called slew rate constraints. Since the current controller formulation uses the Δu in the objective function, the values are optimized. Thus the additional constraints do not add any functional advantages. The control input values at $u(k-1)$ can be used to estimate the Δu given the plant model is sufficiently accurate. Since the estimated values are based on the model of the plant, the issues with model mismatch are carried over rendering a non-zero steady state value.

The efficiency of the MPC during a transient input at the start of the simulations is hindered by the physical constraints. A possible solution would be to relax the constraints on the control input the first implementation step to avoid infeasible solutions from the MPC controller. Since the constraints on control inputs u and deviations of control input Δu have been defined, a strategic relaxation was employed during the controller tuning phase.

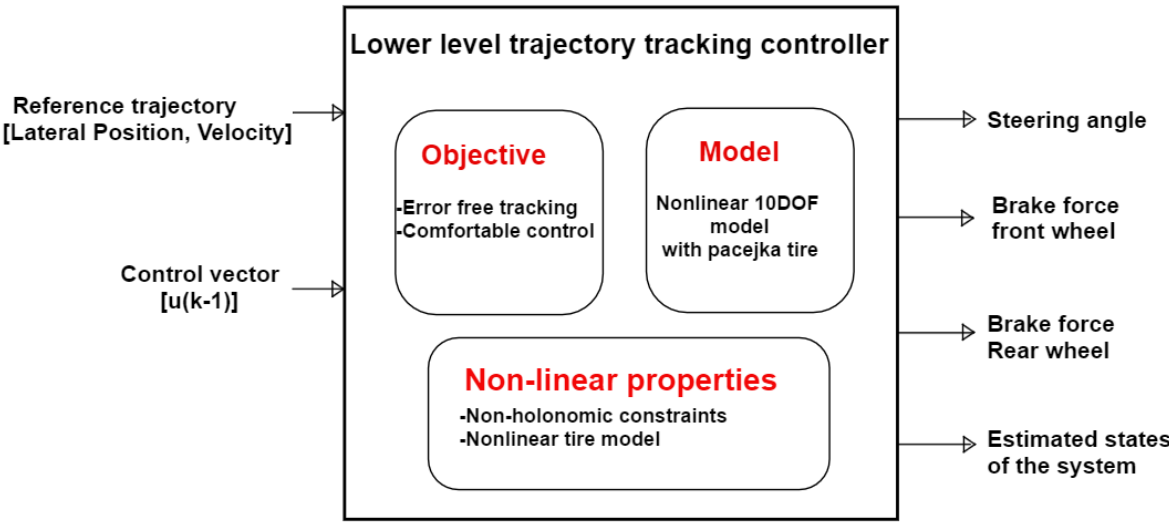
Lower level trajectory tracking controller
Reference trajectory
[Lateral Position, Velocity]
Control vector
[u(k-1)]
Objective
-Error free tracking
-Comfortable control
Model
Nonlinear 10DOF model with pacejka tire
Non-linear properties
-Non-holonomic constraints
-Nonlinear tire model
Steering angle
Brake force front wheel
Brake force Rear wheel
Estimated states of the system

Figure 5-1: Lower level control block

5-4 Controller design

The lower level controller gives the control inputs in the form of steering angle, brake forces for the front and the rear wheels. The control block in 5-1, illustrates the components in the operation of the lower level controller. The reference trajectory from the higher level controller is fed for tracking purpose along with the predicted control input vector. This forms the feedback loop for lower level controller and Δu formulation.

The controller generates the tire lateral forces at the front and the back wheels whereas the brake torque distribution logic is required The method behind the torque distribution can be computed from the optimal values of the front and rear brake forces. The tire slip angles determine the braking action on the wheels.

The commonly used single wheel braking logic is incorporated with respect to the dynamics control. The braking yaw moment to control the vehicle is induced without drastic effect on longitudinal dynamics. This can be explained by the concept of oversteer and understeer in vehicle dynamics. It is preferred to have the vehicle in the oversteer region as there is an understeer limit beyond which vehicle is uncontrollable. When the vehicle 5-3 turns more than the desired amount on the curve, the vehicle is said to be oversteered. In the condition when the amount of steer is insufficient to execute the turning manoeuvre, the vehicle is understeered. This phenomena is plotted in the linear region as shown in 5-2. The desired response can be determined based while keeping the vehicle stable and comfortable.

The pseudo algorithm 1 [69] explains the instances when the brake torque is distributed [8]. In relation to 5-3, the outer wheels receive brake torque when understeer is required while inner wheels receive brake torque when oversteer is required. This value depends on the slip angles at the front and rear of the vehicle. The dynamics behind the front and rear distribution which assumes an equal traction force ration between the rear and front side. The important thing to notice here is the saturation of lateral tire forces required to control the vehicle.

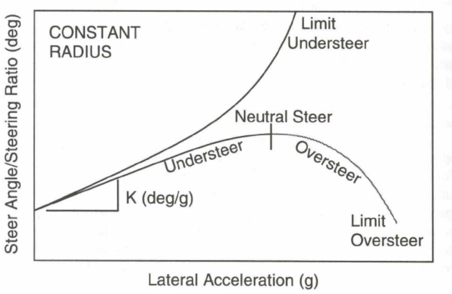

Figure 5-2: Vehicle oversteer and understeer limits with respect to steering angle and lateral acceleration [8].

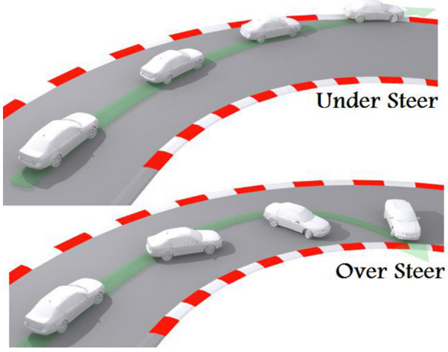

Figure 5-3: Vehicle oversteer and understeer representation. [9]

During aggressive manoeuvres, and fast control inputs that the vehicle has to adhere to, it is possible the vehicle cannot be stabilized. Alternative formulations are possible for the brake torque distribution but not considered in this thesis. The points highlighted in [69], show the performance is better with the left/right brake distribution as opposed to the application in our study. The preferred method to induce oversteer yaw moment, and understeer yaw moment, would be to brake the rear inside and front outside wheels respectively.

Algorithm 1: Braking torque calculation

```
Input: $F_{l_l}, F_{l_r}, \alpha_f, \alpha_r$
Output: $T_{b_{f,l}}, T_{b_{f,r}}, T_{b_{r,l}}, T_{b_{r,r}}$
begin
if      $F_{l_l} = 0$ and $F_{l_r} = 0$ then
            $T_{b_{f,l}} = 0, T_{b_{f,r}} = 0, T_{b_{r,l}} = 0, T_{b_{r,r}} = 0;$
else
        if $\alpha_f - \alpha_r > 0,$ then
            $T_{b_{f,l}} = 0, T_{b_{f,r}} = 0, T_{b_{r,l}} = -rF_{l_l}, T_{b_{r,r}} = -rF_{l_r};$
else
        else
            $T_{b_{f,l}} = -rF_{l_l}, T_{b_{f,r}} = -rF_{l_r}, T_{b_{r,l}} = 0, T_{b_{r,r}} = 0;$
        end
end
end
```

5-5 Tracking controller simulation

The lower level tracking controller handles several non-linear design and physical constraints of the vehicle model and tire. This produces less than optimal behaviour for simple manoeuvres. The controller was tested for standard manoeuvres, double lane change. This test was used to tune the lower level controller for the best performance when incorporated into the hierarchical scheme.

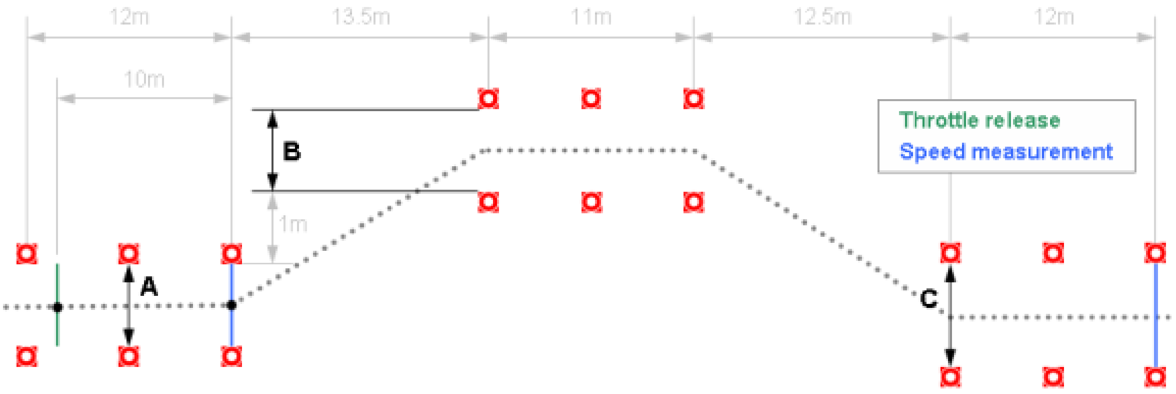

Figure 5-4: Double lane change manoeuvre specifications [8].

Primarily, different velocities were used to analyse the tracking controller performance with a double lane manoeuvre. The performance of the controller was measured against standard time domain performance metrics to avoid overshoot and reasonable setting and rise time to converge to the reference.

5-5-1 Double lane change manoeuvre

The following tests were conducted to evaluate the individual performance of the lower level tracking controller. A trajectory was constructed according to the rules stated per ISO 3888-1 for a double lane change manoeuvre.

The common driving manoeuvres used in testing vehicle dynamic control systems include the double lane change manoeuvre. It approximates the vehicle behaviour changing lanes from right lane to left and back again. The vehicle handling with understeer, oversteer conditions that are caused by saturated tire forces and high lateral acceleration forces during the execution. The stability in the lateral manoeuvres are especially tested with this case.

The test is defined by the International Standard ISO 3888-2 for double lane change manoeuvre in the normal case. The entry lane, exit lane, lane width, lateral and longitudinal lane offset are defined as shown in figure 5-4. The advantages of the test for autonomous controllers is the objective results obtained, and the reproducibility of favourable results.

The vehicle must achieve the a smooth manoeuvre while avoiding the cones along the path. The lateral distance between the cones are dependent on the width of the vehicle. This is included in the ISO standard test which allows the width to be modified based on the vehicle. The expression $W_{largest} = 2.1W_c + 2.25$ where, W_c is the width of the vehicle.

5-5-2 Implementation and Optimization

Here, the parameters to tune the cost terms Q and S were intuitively tuned but to improve the performance, an optimization using $'fminseach'$ was carried out to search the parameter space for the optimal values. This yielded a systematic approach to tuning the parameter values for the desired response.

The optimization solver SNOPT was used for the lower level tracking controller as well. The ability to solve general nonlinear optimization problem with the scope to handle quadratically constrained problems was a supporting factor in the choice. The lower level controller was tuned with respect to the required objectives and the the contributing objectives were the error tracking and the Δu formulations. Another technique to evaluate the performance of the setup dealt with the major iteration limit with respect to the SQP problem. The lower limit allowed for faster optimization solutions and ability to test more parameter values. Once acceptable results without response blow out, the iteration limit was increased to find optimal tracking responses. The function cost and the parameter values were monitor during each iteration to evaluate if the cost term was indeed converging to a minimum value.

The effect of the initial condition on velocity was noticed to affect the performance. The lower velocity value, intuitively yields better response when the full optimization procedure is completed. With the lower iteration limits, the parameter values were not robust posing more search directions for the optimizer. Thus the higher velocity value had a more desirable result with the current formulation. It is expected to change once the full optimization cycle with higher iteration limit is set.

5-5-3 Results

The following results were obtained for the tracking controller subject to the standard double lane change manoeuvre. The initial results obtained for the lower level controller shown in 5-5 rendered a smooth manoeuvre but violated the safety margin by colliding with the cones. This was improved by optimizing the tuning parameters in the parameter search space. The figure 5-6 indicates the path following of the vehicle along the cones following a safe manoeuvre. The initial lag can be attributed to the large entry speed $20[m/s]$of the vehicle. The tests in [47] test the vehicle upto $12[m/s]$ with constraints. This indicates the degradation in performance with respect to the increase in vehicle velocity. But the tuning efficacy of the controllers rendered a good result for the high speed entry. The ability of the vehicle to handle the double lane change manoeuvre at high speeds qualifies as an evasive manoeuvre when tested with the ISO 3888-1 manoeuvre.

The figure 5-7 shows the lateral position of the vehicle. This graph is indicative of the smooth curves generated with respect to the lane change manoeuvre.The plot 5-8 indicates the steering angle and steering rate while following devoid of any aggressive control inputs. This plot upholds the feasible performance criteria with respect to smooth following with comfortable control inputs. Finally the yaw rate and the brake torques are depicted in figures 5-10 and 5-9. The stability of the vehicle is maintained throughout the curve despite the high entry velocity. The braking logic applied can be seen to induce the required turning action as the vehicle traverses the curve.

5-6 Concluding remarks

The chapter details the tracking control problem and added complexity with disturbances. Equipped with disturbance handling features, the control objectives are defined for the trajectory tracking MPC controller. Here, the practical constraints are important due to the

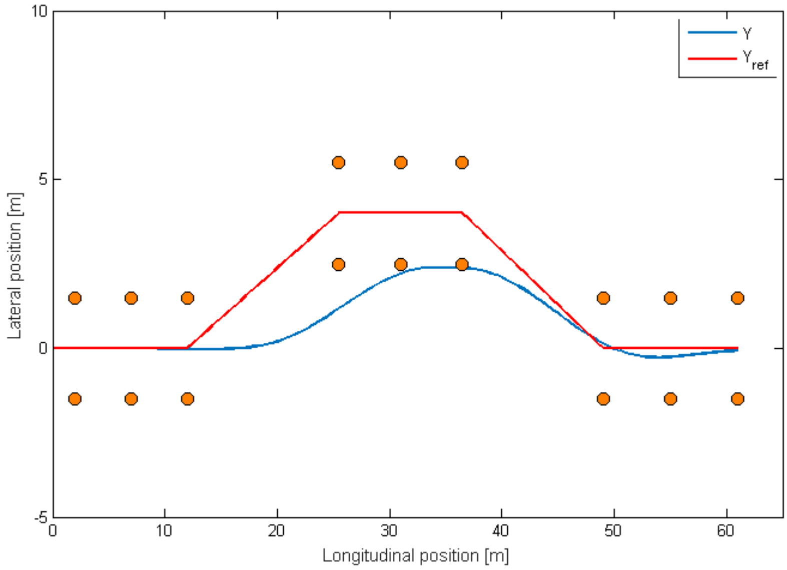

Figure 5-5: Path following performance of vehicle entering at $10[m/s]$

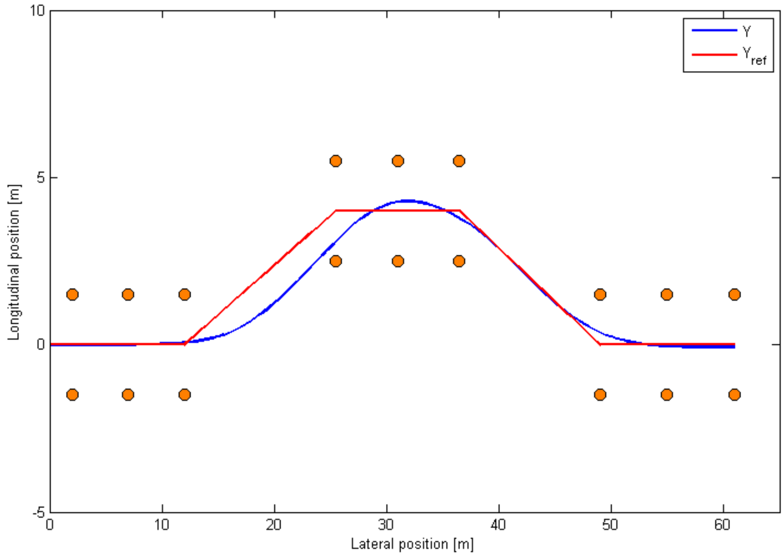

Figure 5-6: Path following performance of vehicle entering at $20[m/s]$

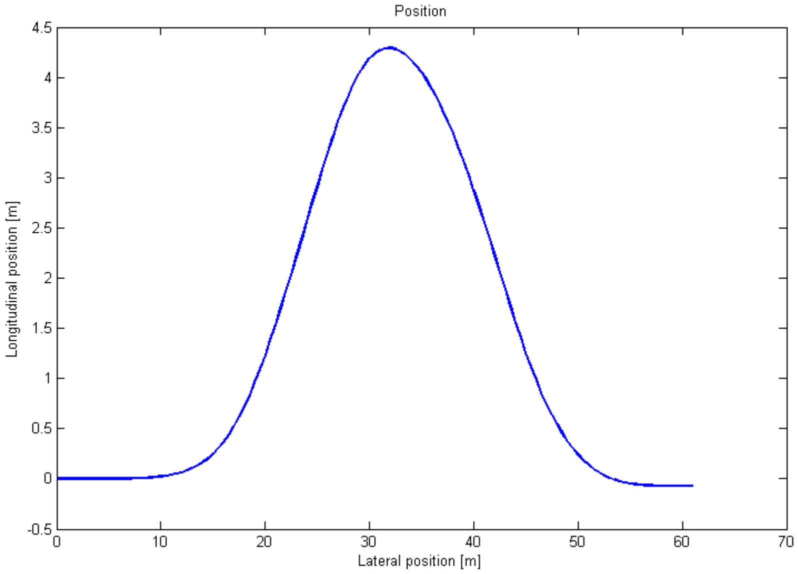

Figure 5-7: Lateral position Y of the controller vehicle.

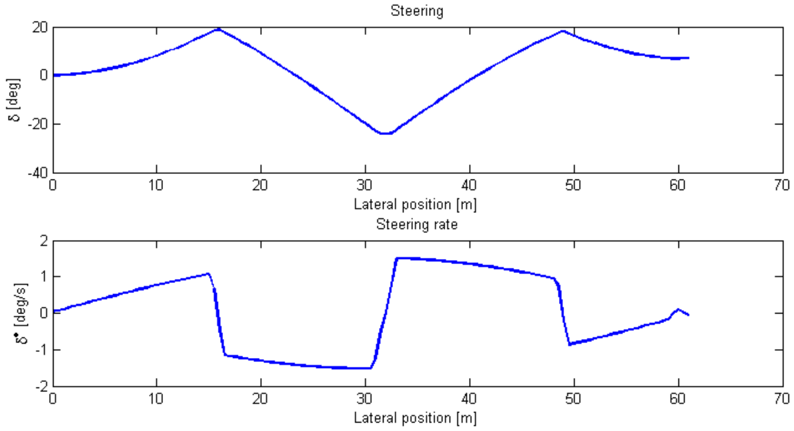

Figure 5-8: Steering and steering rate measures for the double lane change manoeuvre.

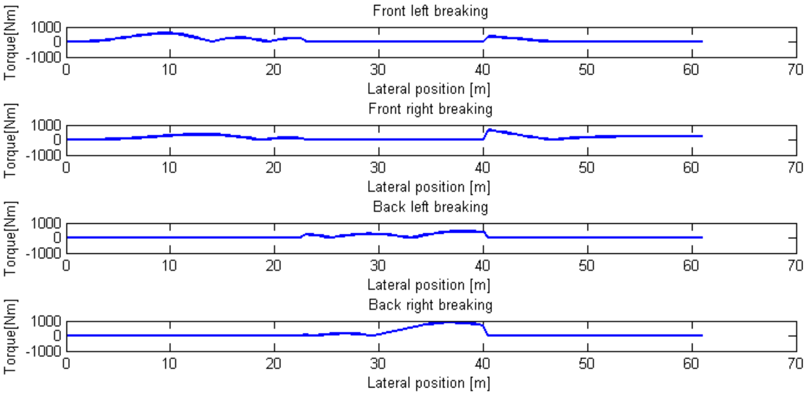

Figure 5-9: Brake torques at the four wheels.

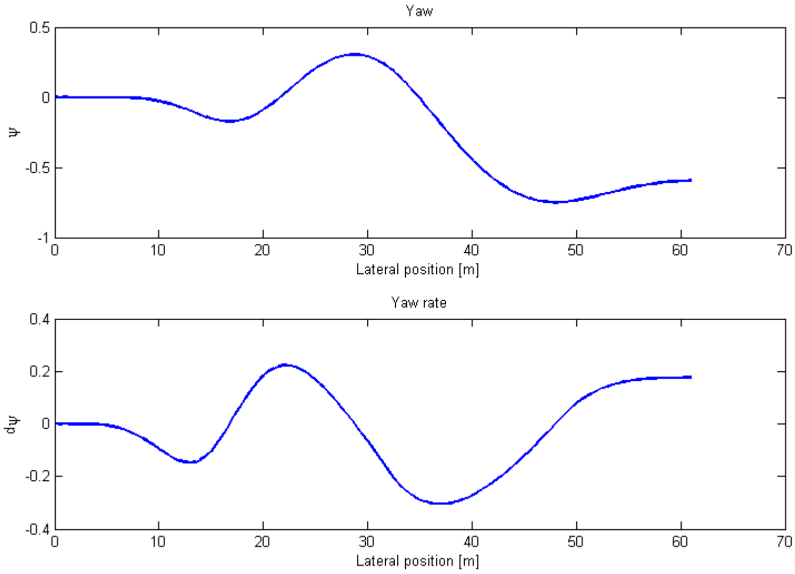

Figure 5-10: Yaw angle and yaw rate of the vehicle during the double lane change manoeuvre.

immediate connection to the vehicle actuator from the lower level control inputs. The optimal control problem is formulated with the relevant constraints and the tuning parameters are discussed for the MPC controller. The braking logic is discussed with respect to vehicle dynamics control to deduce the brake torque at all the wheels. Finally, the tracking controller is tested with a double lane change manoeuvre to evaluate the performance of the controller. The results obtained for the entry speed of $20[m/s]$ satisfied both the comfort and safety criteria. The results showed reasonable tracking and potential scalability to the hierarchical controller.

Chapter 6

Simulation Results

Testing the hierarchical controller developed using the methods discussed in the previous chapters evaluates the performance of the controller. This chapter presents the results for the simulations carried out in certain scenarios. The critical analysis of the results, reveals the conditions where the controller can satisfy the control objectives in the highway environment. The limitations of the controllers can be observed here for further development. Controller tuning remains one of the important tasks to calibrate for the best performance. General trial and error methods are employed at the start, and the results analysed to improve them accordingly.

The following chapter illustrates the performance of the controller with the chosen scenarios. Section 7-1 is used to define the tuning values for the higher level and lower level controller. These values were modified with respect to certain scenarios but the changes were kept to a minimum to display controller robustness. In section 7-2, the scenario for free highway driving is tested with respect to the higher level controller and the hierarchical controller. The following sections were used to show simple driving manoeuvres of car following and lane change. Section7-4 marked the complex problem of moving obstacles and the performance of the controller under those conditions. Finally the initial results to show the limitations of the problem formulation with overtake of moving obstacles. The problem added an evasive manoeuvre requirement to the controller where the vehicle encounters a slow moving vehicle in the desired lane.

6-1 Simulation environment and controller tuning

All simulations were carried on a HP ENVY Sleekbook with the system capacity listed in table 6-1. The limited computational capacity of the computer increased the computation time with respect to the optimization iterations.

The simulations were primarily executed in MATLAB 2014a and SIMULINK. The controller implemented was overhauled several times as explained in section 5-3 with the methods chosen. At final implementation, the ACADOToolkit was used to link the controller to the model.

Table 6-1: Computer hardware details.

Component	Specification
Processor	Intel(R) Core(TM) i5-3317U, 1.70GHz, 1701 Mhz
RAM	4 GB
Processing cores	2 Core(s), 4 Logical Processor(s)

Table 6-2: Controller tuning parameters

Parameters	Symbol	value
Prediction horizon	$H_{p_{HL}}$	$10 - 30$
Prediction horizon	$H_{p_{HL}}$	$10 - 30$
Weight	$Q_{1,1}$	$30 - 60$
Weight	$Q_{2,2}$	$90 - 120$
Weight	$R_{1,1}$	$0.1 - 0.3$
Weight	$R_{2,2}$	$0.01 - 0.03$
Weight	$R_{3,3}$	$0.01 - 0.03$
Weight	$S_{1,1}$	$0.01 - 0.03$
Weight	$R_{2,2}$	$0.01 - 0.03$
Weight	$R_{3,3}$	$0.01 - 0.03$
Weight	α	$10 - 30$
Weight	κ	$2 - 6$
Weight	γ	$5 - 15$
Longitudinal velocity	$\dot{x}_{ref}$	$30[m/s]$
Sampling time	T_s	0.1

TOMLAB was used to interface with MATLAB to gain access to the non-linear solvers available. This greatly improved the performance of the controller given the complex non-linear behaviour of the system and problem of trajectory generation.

Another important factor with running simulation on the laptop is the windows priority assessment for the tasks. The optimization routine running in MATLAB will take less priority over the movement of the cursor. This is a minute detail compared to the host of other tasks that are running which are assigned more priority by the windows 10 operating system.

The following controller tuning parameters were selected with the formulation of the MPC problem. The optimal range of tuning parameters was found and adjusted according to the required scenario.

Influence of tuning parameters are elaborated with respect to the particular scenarios. The control goal is relatively modified with the addition of specific requirement of the scenario and the tuning parameters are conservatively modified.

6-2 Scenario 1: Free driving

The first testing scenario was chosen to evaluate the operation of the controller. The simple case was chosen to evaluate the performance of the controller and aid in the fine tuning situation with the other scenarios. The higher level controller and the combined controller were tested with for the cruising scenario.

6-2-1 Trajectory generation controller

The trajectory generation for the simple case of highway cruising is illustrated in figures 6-1 and 6-2. The objectives for lane keeping and reaching desired velocity are satisfied for this scenario. The lane keeping objective to maintain the vehicle CoG at the centre of the road defined by $y_{ref} = 0$ is followed perfectly. The second objective to meet the desired velocity can be seen to follow a smooth curve in the case where desired velocity is set at $15[m/s]$ in figure 6-2. The initial conditions for the cases in 6-1 were set at 0 for the starting position and $15[m/s]$. In figure, 6-2, the performance was tested for a perturbed initial condition for position from $-1.1[m]$ and velocity $0[m/s]$ to simulate a vehicle starting from position away from the lane centre from a standstill. The results indicate feasible performance with both the objectives.

The speed objective is highlighted more in this scenario analysing the effect of desired velocity on the performance of the vehicle. Figure 6-3 depicts the time taken for the vehicle to reach desired velocities of $15[m/s]$ and $36[m/s]$ from an initial condition of $0[m/s]$. The constraint on longitudinal velocity $(0 \leq V_x \leq 30)$ was relaxed for this test case. This simulation is used to show the time taken to reach the average cruising speeds as per the NEDC [] driving cycle and the maximum legal speed limit on highways. The results show that the vehicle takes 38% more time to reach the legal speed limit of $36[m/s]$ in $11.2[s]$. This can be attributed to the constraints placed on velocity and acceleration for the longitudinal direction.

The performance of the trajectory generation is optimal with this case. As seen in the previous section, the performance of the lower level tracking control was optimal for the given trajectory. The next section is used to display the results for the hierarchical controller.

6-2-2 Hierarchical controller

Performance of the hierarchical controller was tested for similar scenarios. The figure 6-4 shows the performance of the combined controller for the simple case. The efficacy of the lower level controller following the position reference of higher level trajectory Y_{ref} is not ideal. The deviation of the lateral position does not exceed $0.7[m]$ in any instance. This is tolerable deviation but unacceptable condition for smooth ride. The velocity profile $\dot{x}$ indicates the erratic change of speed due to the deviation of the lateral position.

The second case tested in figure 6-5 is chosen with the non zero initial condition for lateral position $(Y0)$ at $1.1[m]$ and a higher starting velocity at $25[m/s]$. The deviation from the lane centre is minimal in this case, and the the velocity variation is smoother compared to the first case in 6-4.

6-3 Scenario 2: Car following

The car following case is included to test the ability of the controller to maintain the lane in case the desired velocity is met without need for a lane change. The non-holonomic nature

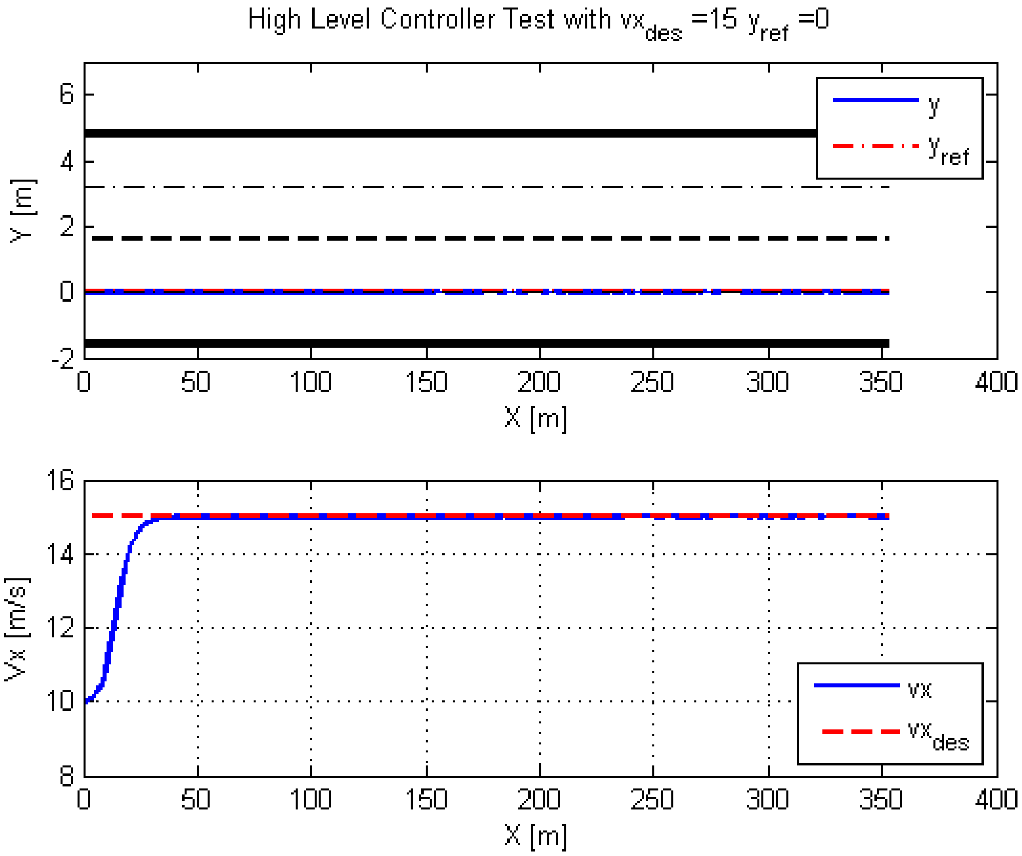

Figure 6-1: Trajectory generation in free highway cruising.

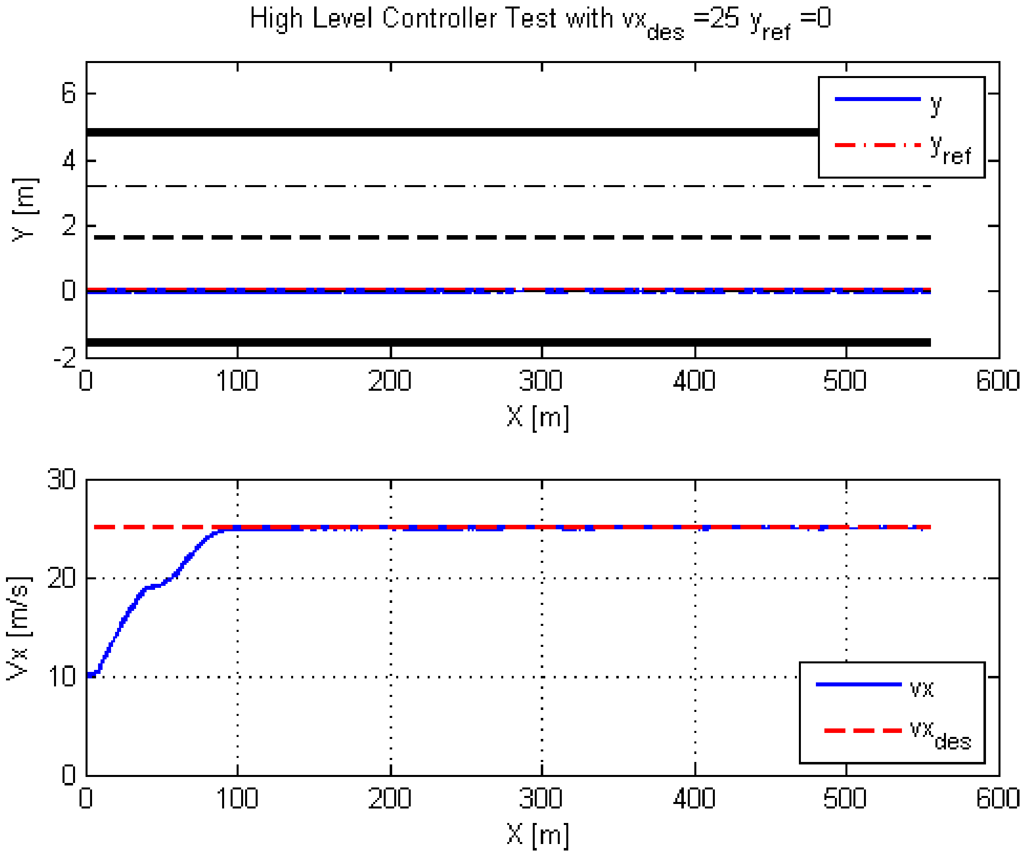

Figure 6-2: Trajectory generation in free driving at higher speed.

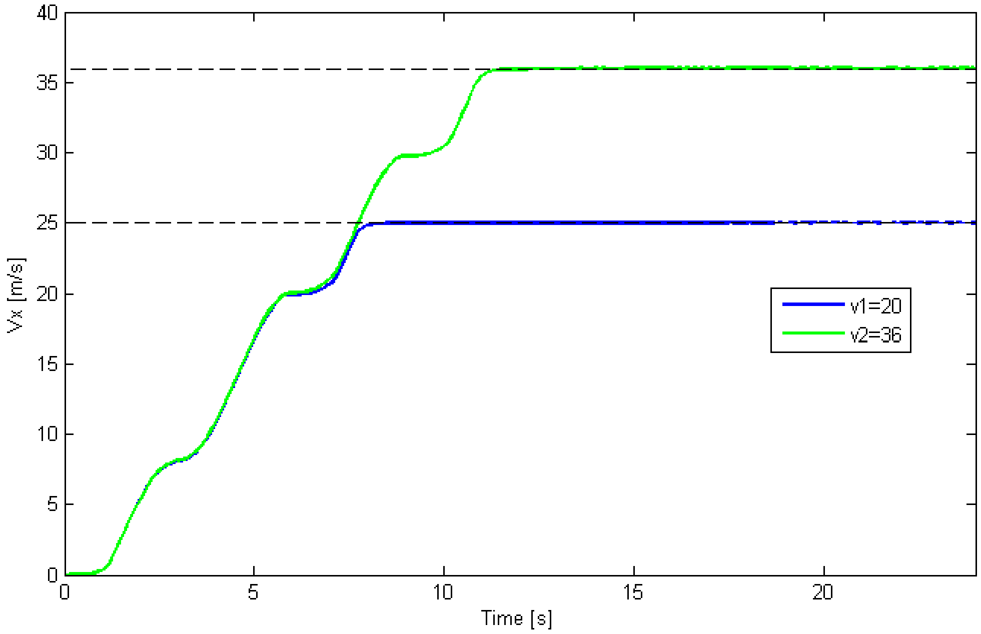

Figure 6-3: Time taken for controlled vehicle to reach different desired speeds.

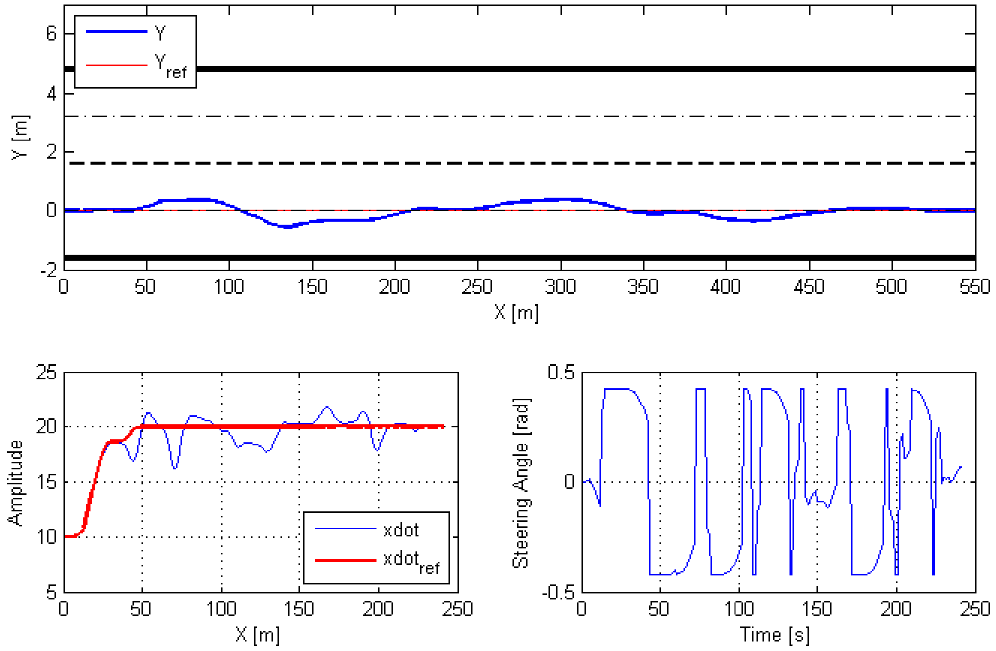

Figure 6-4: Performance of the hierarchical controller.

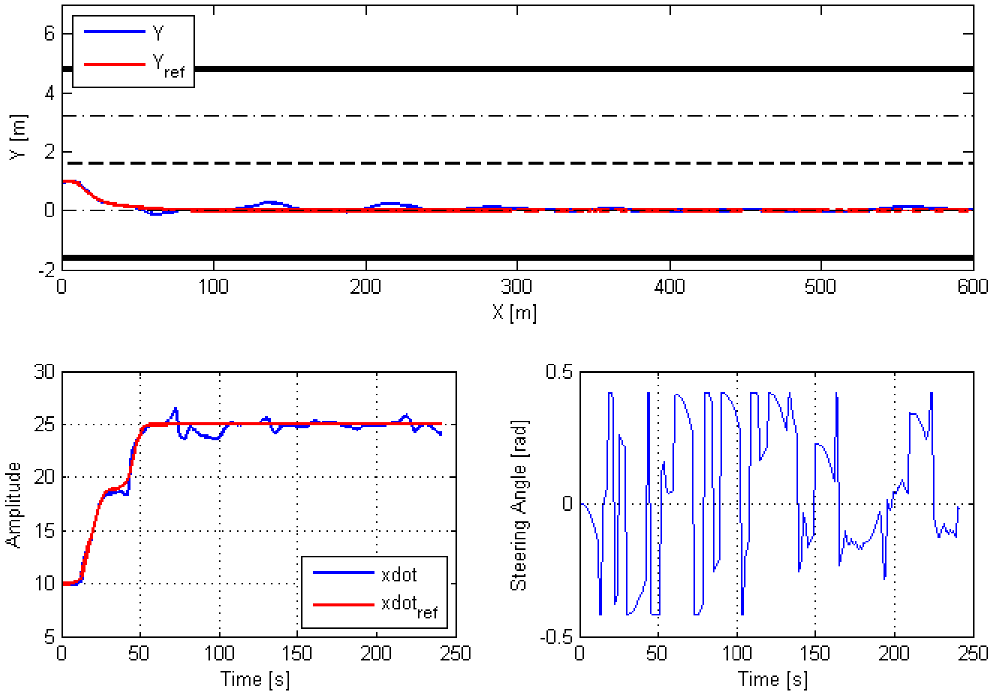

Figure 6-5: Performance of the hierarchical controller for non zero initial condition for lateral position Y.

of the vehicle does not allow for a trivial trajectory generation and tracking solution even in the straight path. The behaviour of the system for the car following provides insight into the effect of the prediction horizon.

6-3-1 Trajectory generation controller

In figure 6-6, the car following function is simulated with a surrounding vehicle in the same lane. The predefined motion of the surrounding vehicle is set with a starting position $80[m]$ ahead of the controlled vehicle. The initial velocity of the controlled vehicle is set at $10[m/s]$. The first observation with the results is the reduction in velocity as the along the length of the road. Here, the main issue with respect to the prediction horizon where $Hp = 30$. The controller takes into account the vehicle ahead, and initiates a slower trajectory following. This result indicates the importance the prediction horizon tuning parameter in the current approach.

The different parameters were tested while tuning, and the some results indicated that the presence of the obstacle avoidance constraints is the single most influential factor in the behaviour and performance of the controller. Thus the focus shifted to tuning the slack variables. In the figure 6-7, the lane keeping objective and the speed objective are satisfied. The changes made with respect to initial condition on lateral position $(Y0)$ with a value of 0.001 rendered the required result. The slight perturbation to the optimization problem resolved the issue. Due to the functioning of solver, this is a frequent occurrence in the optimization processes. Usually, when the limiting value is taken as initial, due to algorithm setup the optimizer fails to find a feasible initial solution, and the suboptimal out frequently violates many constraints. Introducing perturbations in the right direction can be very beneficial in obtaining the optimization results.

6-3-2 Hierarchical controller

The observation of suboptimal tracking can be observed with the hierarchical controller. The build of control gain in the lower level controller, was evident in the result but the cause of the anomaly was hard to pinpoint. The constant reduction in speed is followed by the vehicle whereas the position tracking is satisfied until $250[m]$ on the longitudinal direction. The larger deviation from the lane centre is observed which also reflects in the speed deviation. The change in the direction of the vehicle causes the reduction on vehicle speed.

The observation with the steering angle is seen to change aggressively with the maximum and minimum bounds set at $0.41[rad]$. The steering rate constraint set at $\pm 0.17[rad]$to restrict the aggressive changes is violated.

6-4 Scenario 3: Lane change manoeuvre

A lane change manoeuvre is one of the fundamental driving tasks along with the speed and lane keeping tasks. The choice to obtain a lane change manoeuvre but not a complete overtake manoeuvre can be motivated to test the ability of the controller to perform a lane change operation. The smoothness of the trajectory generated to obtain the results was evaluated

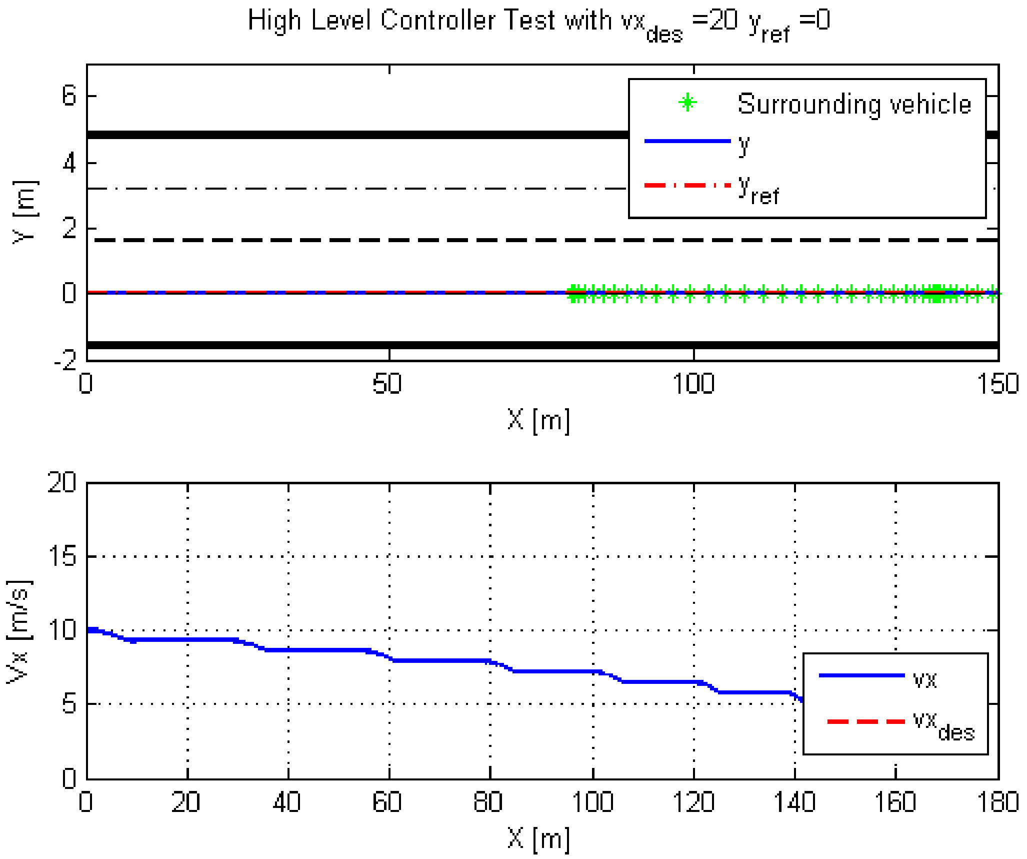

Figure 6-6: Car following ability when surrounding vehicle is travelling at desired highway velocity.

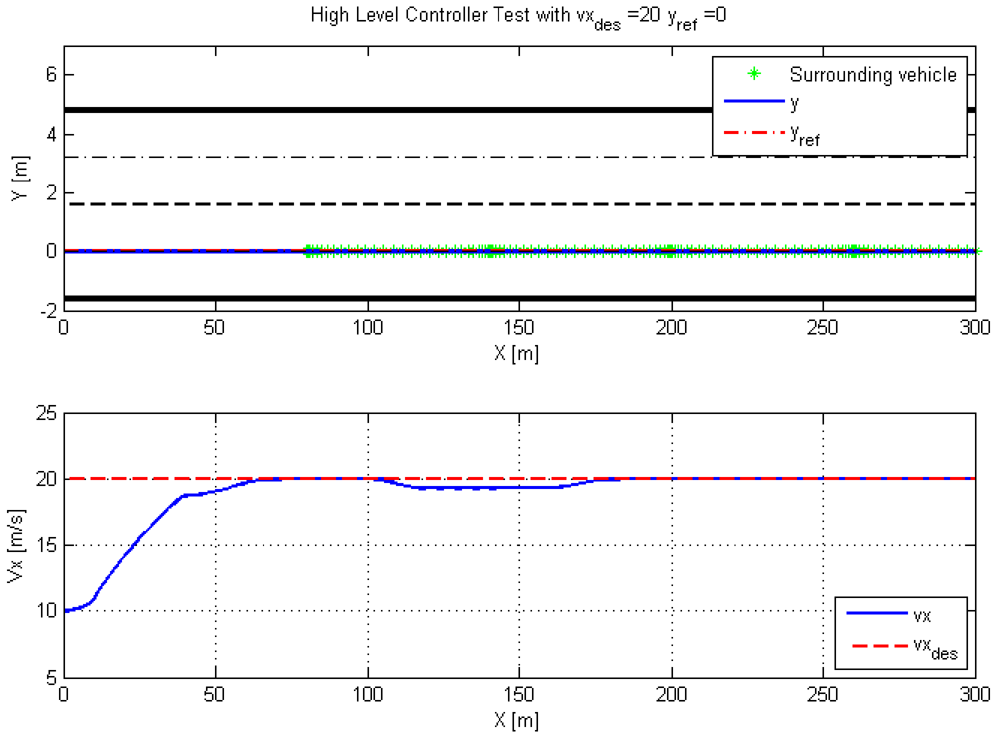

Figure 6-7: Car following ability when surrounding vehicle is travelling at desired highway velocity.

in the trajectory generation controller. Consequently, the hierarchical controller was tested with the lane change scenario.

6-4-1 Trajectory generation controller

The figure 6-9 is the simple case for the lane change manoeuvre with the lateral position reference y_{ref} set to 3.2 to indicate the lane centre in the outside lane. The vehicle CoG has to converge with the lateral reference position to meet the position objective and the velocity response is desirable for the vehicle. The vehicle performs the manoeuvre to reach the objective in $70[m]$. This value is acceptable due to the average lane change distance required lies between $60 - 100[m]$ for a single lane change manoeuvre.

The two cases seen in 6-9 and 6-9 indicate the difference in performance with different velocities. Comparing the figures, it is evident the vehicle requires more than twice the distance to reach the higher desired speed at $25[m/s]$ compared to $15[m/s]$.

6-4-2 Hierarchical controller

The hierarchical controller was tested for the lane change operation after obtaining the desired result with the lane change trajectory generator. The vehicle performance test on the double lane change as seen in the lower level controller was capable of following the trajectory. The behaviour of the hierarchical controller combining the the two individual controllers shows a

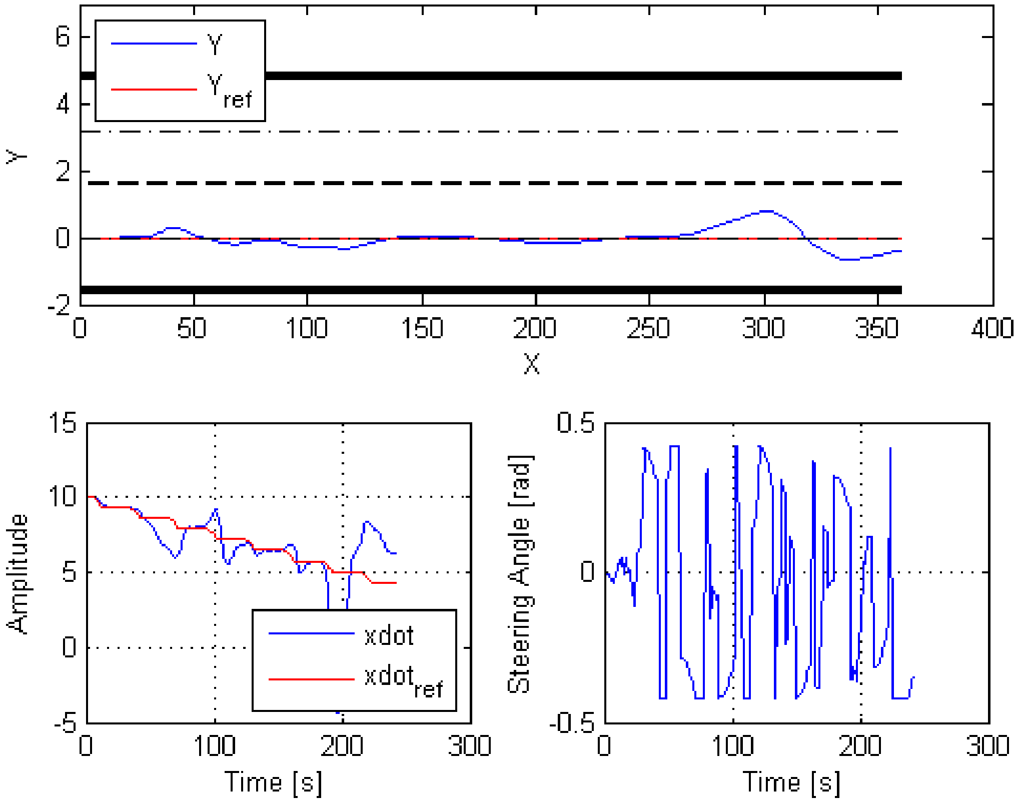

Figure 6-8: Hierarchical controller car following feature

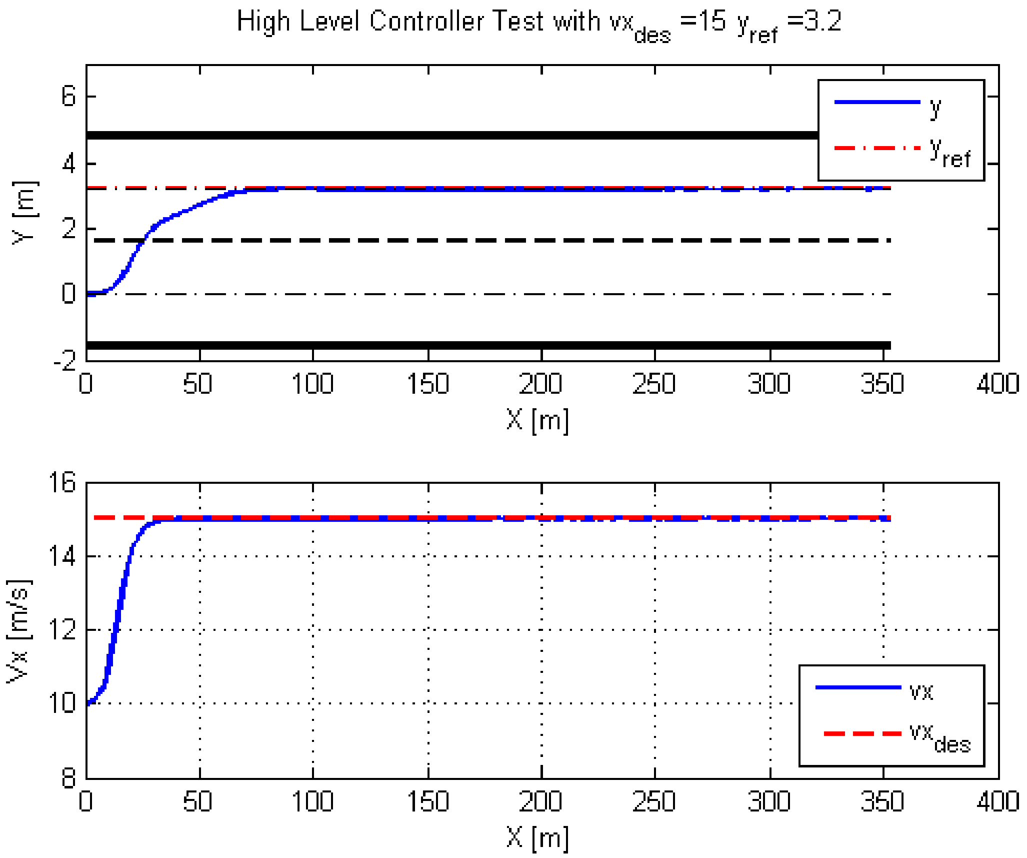

Figure 6-9: Lane change manoeuvre for vehicle.

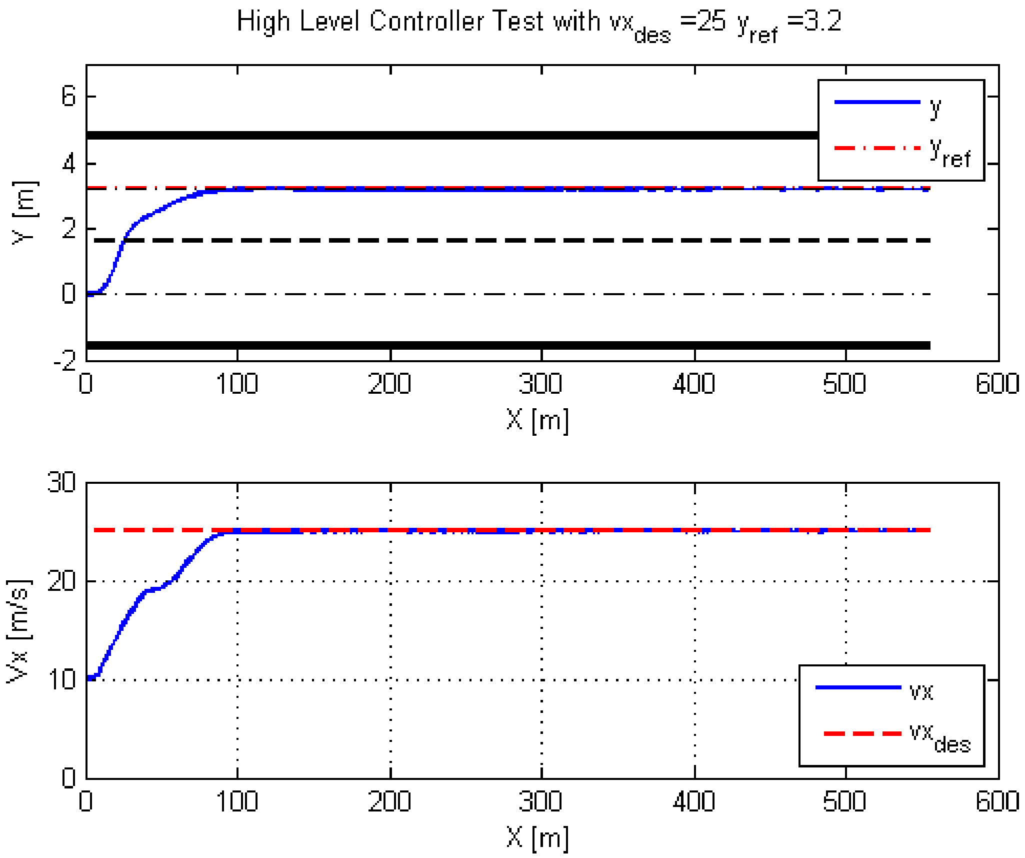

Figure 6-10: Lane change manoeuvre for vehicle with higher desired speed.

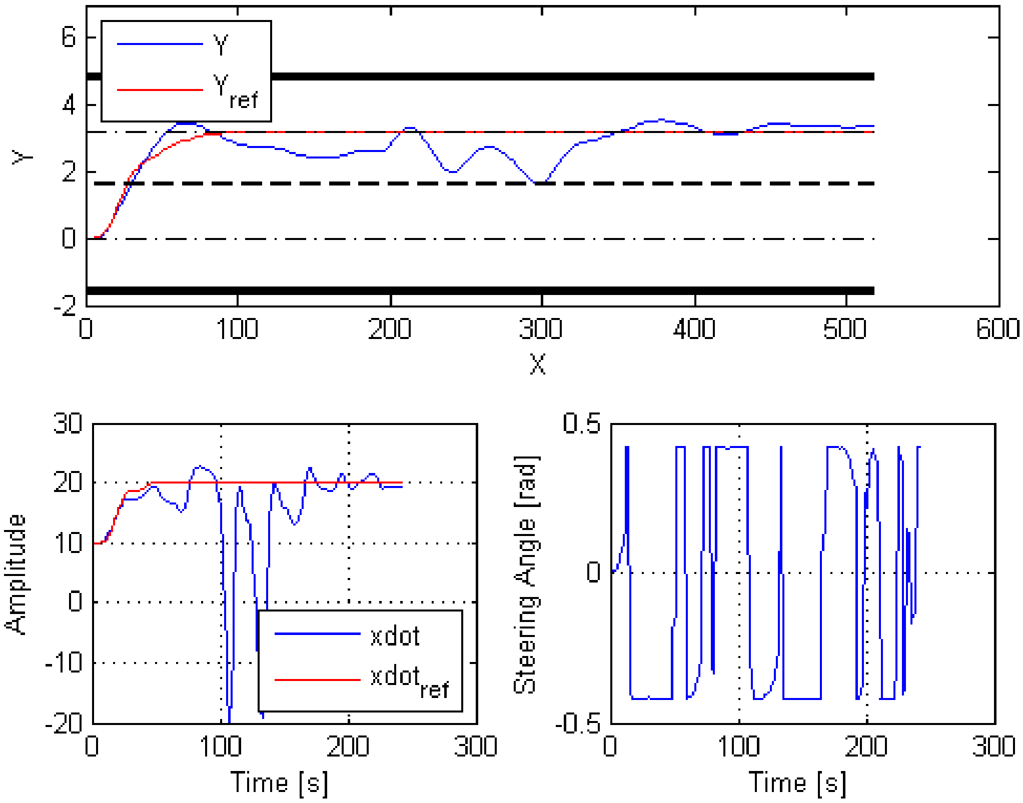

Figure 6-11: Lane change manoeuvre with trajectory generation and tracking.

desirable results at the beginning on the manoeuvre with the lower level sufficiently tracking the higher level controller. The slight overshoot of the lower level during the lane change manoeuvre quickly falls below satisfactory following condition. The undesirable behaviour at $x = 200[m]$ and $x = 300[m]$ are compensated to return the vehicle to the original position reference at $y_{ref} = 3.2[m]$. The unacceptable behaviour is seen in the drop in velocity due to the drop in change in direction however the minimum velocity constraint set at $0 \leq V_X$ is violated indicating a reverse direction. Physically, this is not acceptable. The steering angle on the other hand reaches the saturation limits of the constraint, and is restricted to use a higher control gain for the required result.

6-5 Scenario 4: Static obstacle avoidance

The static obstacle case depicts the capability of the controller to avoid static obstacles. The case chosen here, represents a scenario where a car has stalled or come to an abrupt stop, and the controlled vehicle has to successfully plan a path around the obstacle. The controlled vehicle plans an overtake manoeuvre around the surrounding vehicle to return to the original lane centre reference. The margins of safety on the obstacle, and the interaction of the two collision avoidance constraints acting varies the ability to cope with the obstacle. The scenario is tested with the higher level trajectory generator first and the feasibility of the trajectory generator is evaluated.

6-5-1 Trajectory generation controller

In figure 6-12 the case for the static obstacle positioned at a distance of $80[m]$ from the starting position of the vehicle is detected and appropriate trajectory is planned. The lateral position y curves around the static obstacle, and the first behaviour noticed is the path that overshoots the lane centre of the outside lane. This behaviour can be attributed to the small feasible region formed due to the collision avoidance constraints. This is also reflected in the increase in velocity above the desired velocity. The tuning of the path back to the lateral reference($y_{ref} = 0$) follows a less than smooth path. This behaviour has been achieved by extensive tuning of the slack variables $(\epsilon_{j_r}, \epsilon_{x_j r})$ despite the second spike at $130[m]$ which is reflected in the velocity curve as well. The overtake manoeuvre completed in $200[m]$ is above the average distance required for overtake manoeuvres on highways but the set desired speed is set at $20[m/s]$ which is also lower than the limit for highway driving speed.

Another scenario with the obstacle avoidance ending in a lane change operation reflects a situation where the controlled car would need to stay in the outside lane longer to anticipate another slow moving car or other obstacles. The convergence to the lane centre on the outside lane ($y_{ref} = 0$) is faster as opposed to returning to the left lane centre. The velocity profile on the other and does not change with the new reference value. The overshoot of $0.3[m]$ from the outside lane centre at $125[m]$, causes the spike in velocity to speed the reach to the lateral reference.

6-5-2 Hierarchical controller

The hierarchical controller tested for the static obstacle case, achieves the convergence to the lane centre after oscillating and violating the boundary constraint. The travel to the edge of the road beyond the lane centre from $35 - 110[m]$ on the outside lane and at $160[m]$ and $230[m]$ on the right lane is the lane centre violation. Whereas the velocity profile is infeasible with several undesirable behaviours seen with respect to the velocity reference. The nonlinear characteristics of the vehicle and the inability to track the higher level trajectory formed using the point mass model contribute to the unwanted behaviour. Steering angle variation reaches the saturation limit during the first operation and then varies with respect to the change in position.

6-6 Scenario 5: Overtake manoeuvre with slow moving vehicle

The scenario representing the overtake manoeuvre similar to vehicles passing in the presence of a slower moving vehicle in the travelling lane. The controlled vehicle behaviour in the presence of a slower moving vehicle, must overtake the slower moving vehicle and return to original lateral position reference.

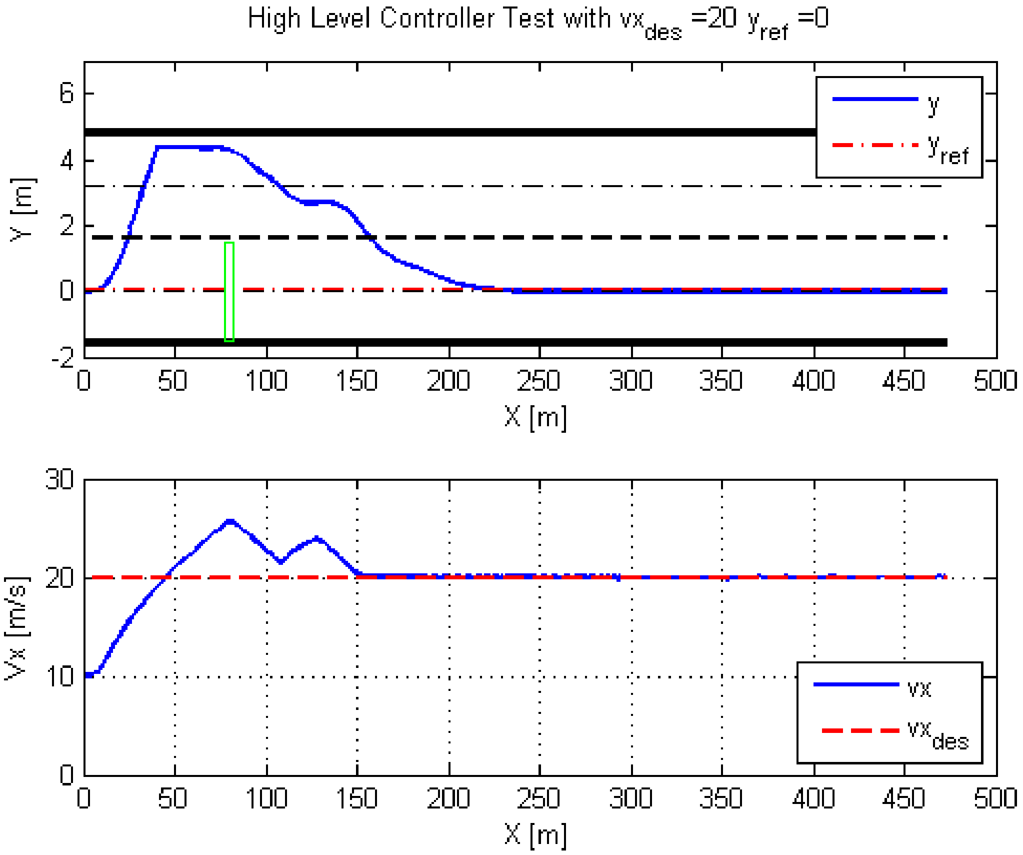

Figure 6-12: Overtake manoeuvre with static obstacle in desired lane.

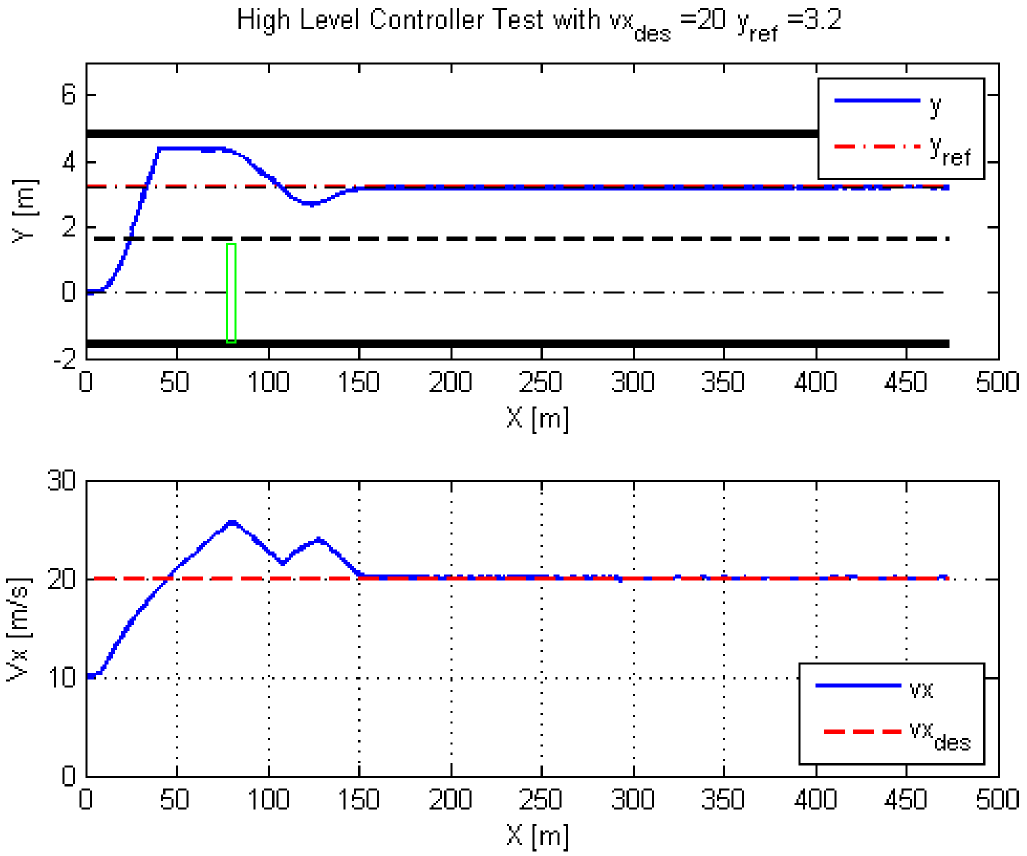

Figure 6-13: Overtake manoeuvre with static obstacle in desired lane with outside lane centre reference.

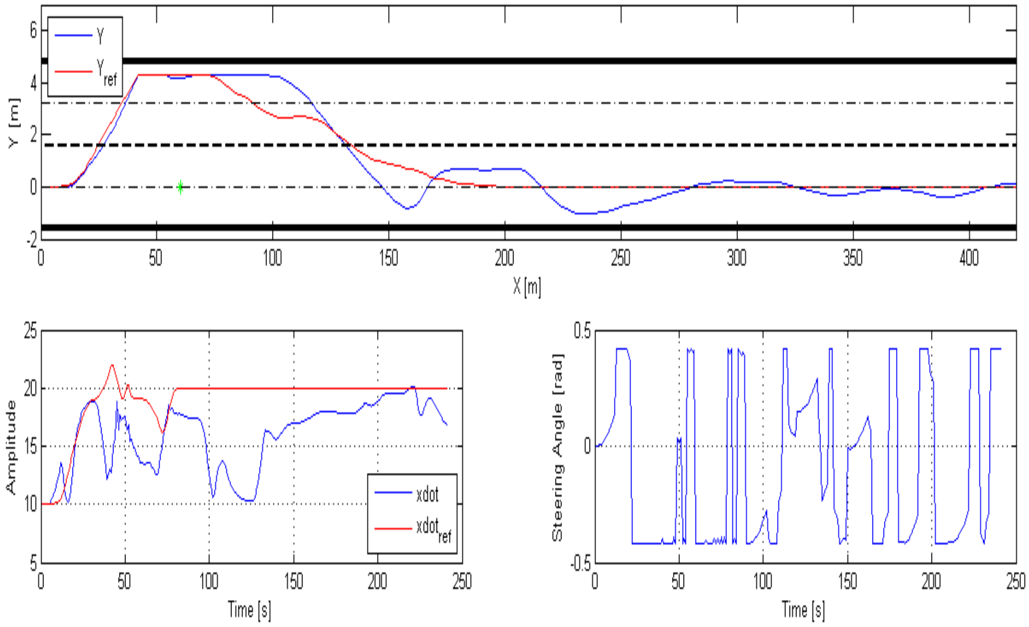

Figure 6-14: Hierarchical controller performing overtake manoeuvre with static obstacle in desired lane.

6-6-1 Trajectory generation controller

Figure 6-15 is the regular case where, the vehicle has to pass the slow moving surrounding vehicle starting to move from initial position $80[m]$ and travelling at a slower speed of $10[m/s]$. The behaviour noticed in this scenario highlights the issues with the trajectory generation in the overtake case. The lane change manoeuvre initiated reaches the road boundary due to the small feasible region in the optimization problem, and initiates a convergence to the lane centre, at $120[m]$, this is changed due to the presence of the slower moving forward collision avoidance boundary. The passing of the vehicle, at $230[m]$ avoids the slower moving vehicle at a small margin. The surrounding vehicle starting at $80[m]$ reaches, $190[m]$ in $11[s]$ of the simulation and trails behind the controlled vehicle. The rear collision avoidance constraint, is violated due to the small relative distance between the two vehicles.

The influence of the rear collision avoidance constraints is not seen in figure 6-16 where the controlled vehicle is the lateral position reference ($y_{ref} = 0$).

6-6-2 Hierarchical controller

The hierarchical controller for the original case degrades in performance with the infeasible trajectory.

The condition where the vehicle as to stay in the left lane after passing the obstacle is show in picture 6-17.

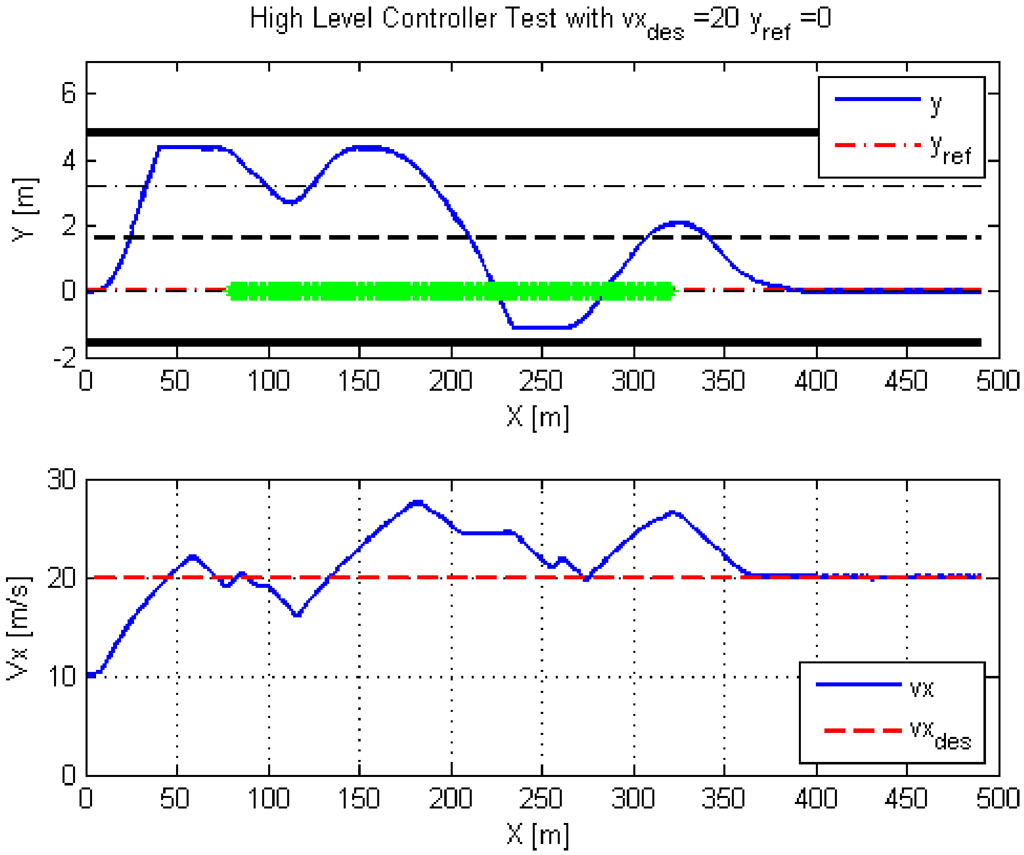

Figure 6-15: Overtake manoeuvre with slow moving obstacle in desired lane.

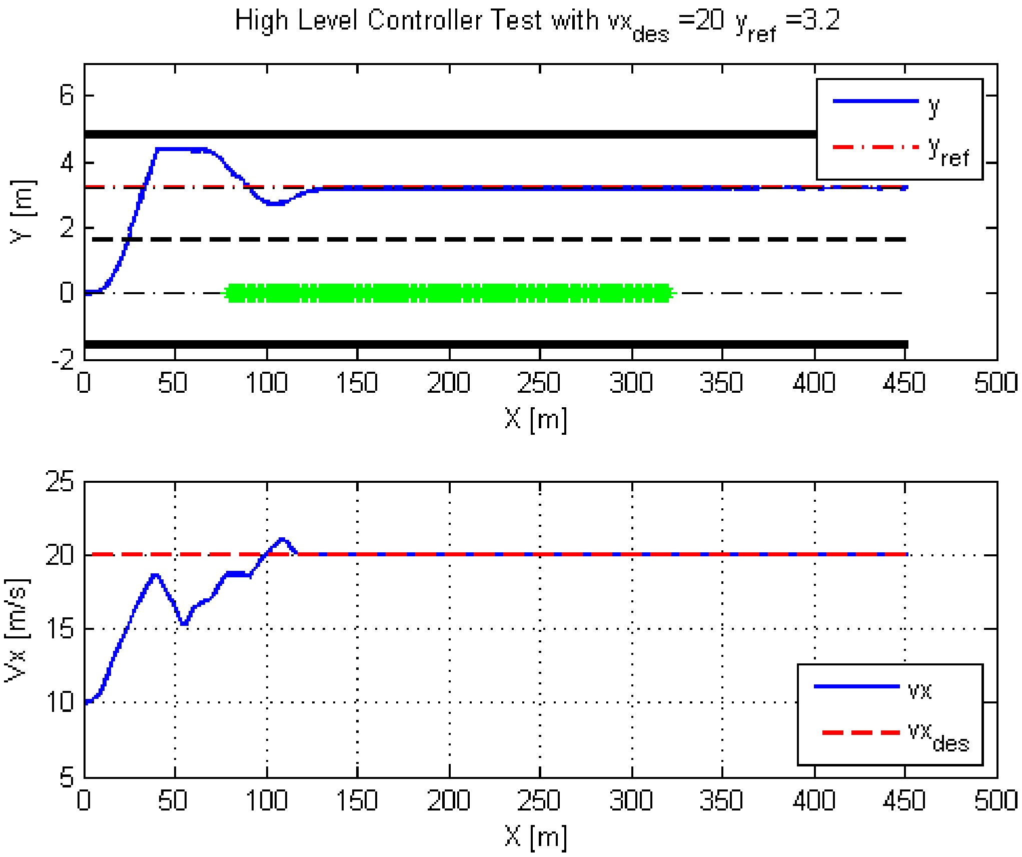

Figure 6-16: Lane change manoeuvre with slow moving obstacle in desired lane.

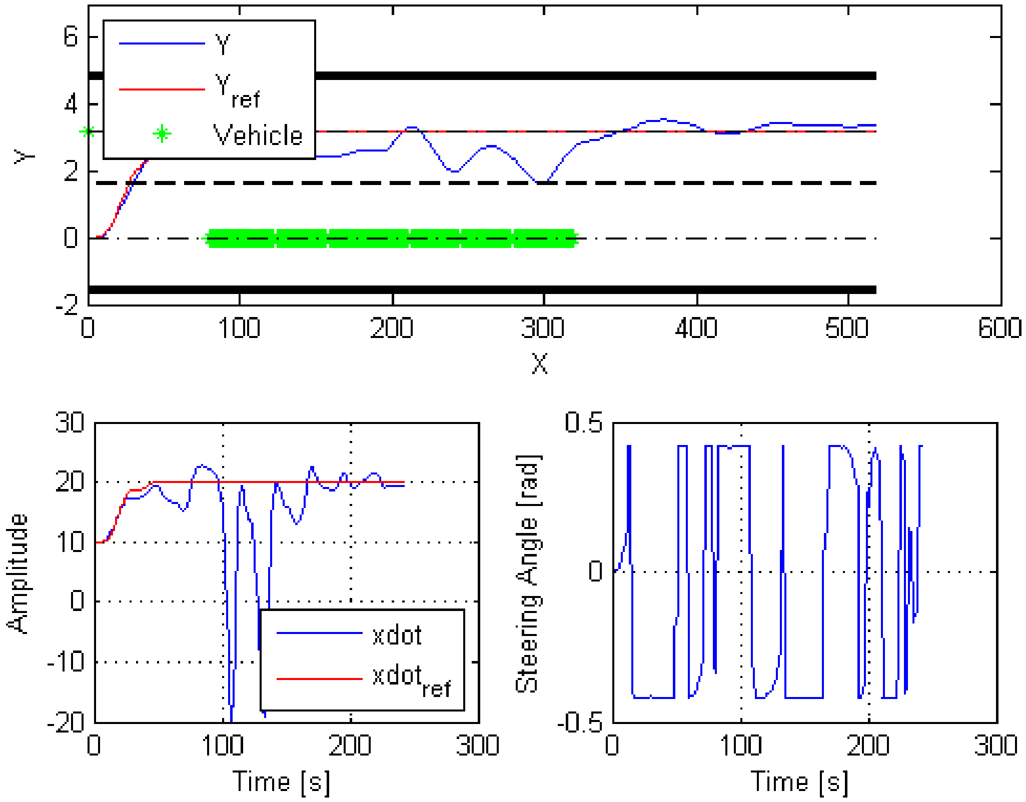

Figure 6-17: Lane change manoeuvre with slow moving obstacle in desired lane.

6-7 Concluding remarks

The problems that arose during the implementation phase of the hierarchical controller rendered suboptimal results in the complex scenarios. The systematic problem decomposition and different strategies of implementation were unable to resolve the issue. It would seem that there is a build-up of control gain in the lower controller with an effective instability in the upper controller for the proposed lower controller range. The attempt to search the hyper parameter space was not a viable method, due to the slow function execution.

The simple scenarios displayed feasible results while the errors in the higher level controllers can be accounted to the tuning parameters. The effectiveness of the approach with the collision avoidance constraints are subject to tedious tuning process despite the inclusion of the auto tuning feature in the cost function formulation of the higher level controller. The development and testing process must be subject to alternative formulations and further tuning procedures.

Chapter 7

Conclusion

The main goal of the thesis was to implement an hierarchical model predictive controller for trajectory generation and tracking in the highway driving environment. The motivation to pursue the topic and choice of approach were elaborated in the chapter 1 supported by the literature survey. The emphasis was placed on the performance of the controllers and test the feasibility of the hierarchical scheme with the various manoeuvre scenarios as seen in the previous chapter.

7-1 Thesis contribution

- Evaluating the issues with hierarchical controller setup for trajectory generation and tracking.
- Testing the hypothesis of generating the reference trajectory with the position state alone. The lower level controller passing a single state to the higher level controller generates a feasible trajectory. The trajectory is subject to tuning and improvement with adding one or more states in the feedback.
- Successfully developed a model predictive tracking controller capable of executing the lane change manoeuvre at an entry speed of $20[m/s]$

7-2 Future work

7-2-1 Model mismatch uncertainties

The model to describe the full vehicle was complex and represented the behaviour of the system. This model yet does not capture, longitudinal dynamics during weight shift, coupled longitudinal and lateral dynamics and suspension dynamics. The non-linear behaviour can be better modelled and kinematic relations can be exploited to obtain a robust model with tolerances for any mismatch with the real world.

7-2-2 Alternative interfacing for Higher level and Lower level controllers

The controllers work well individually but linking the controllers adds additional complexity. The system response is hard to analyse with the presence of nonlinear constraints. Thus an interfacing layer to accommodate the smooth passing of instructions from higher level to lower level and passing of feedback information from the lower to higher level must be incorporated.

7-2-3 Non-convex search spaces

The current formulation simplifies the feasible region for the vehicle trajectory to satisfy the quadratic programming (QP) problem formulation, and maintain convexity. The lane keeping objective was often violated due to the collision avoidance constraints and the cost function including a non-convex objective term penalizing the vehicle travelling at the road edges might render better lane keeping results.

7-2-4 Computational time

The initial goals of the controller development was real time implementation. Due to the complexity of the non-linear vehicle model and interfacing limitations between the higher level and lower controllers, the on-line optimization procedures required time. This optimization time was infeasible for real time implementation, and further strategies need to be developed to improve the speed of computation.

7-2-5 Prediction model for system and surroundings

The ability to detect the surrounding vehicles and environment are crucial to the operation of autonomous vehicles. The topic was explored in depth with respect to driver models, traffic models, and environment models to choose the best method to represent the dynamics of the surrounding. The augmented model which needs to be validated by itself before implementing with the controllers proved to be a vast research topic of the thesis. These methods could be further analysed to choose the best possible gain.

Appendix A

Appendix

A-1 Appendix A

A-2 Bicycle model

Bicycle model or Single track model is a simplified version of the full vehicle model by lumping the wheels at the front and rear axles.

The dynamic equations of the bicycle model is compactly written as:

$$\dot{\xi} = f_by(\xi(t), u(t)) \tag{A-1}$$

Compact function f_by is used for the bicycle model. $\xi \in \mathfrak{Re}^n$ is the state of the system and $u(t) \in \mathfrak{Re}^m$ is the input. There are six states and three inputs in this model. As seen in [47] The six states are lateral and longitudinal velocities in the body frame, the yaw angle, yaw rate, lateral and longitudinal vehicle coordinates in the inertial frame are by state vector is given by

$$\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X]'$$

. The input vector is given by

$$u = [\delta_f, T_f, T_r]'$$

where δ_f is the front steering angle and $T_\star$ is the braking and tractive torque on the wheel. Positive $T_\star$ denotes tractive torque while negative $T_\star$ denotes braking torque.

Figure A-1 shows the modelling terms of the one track bicycle model. In particular, $F_{c_\star}$ and $F_{l_\star}$ are the cornering and longitudinal tire forces in the tire frame. $F_{y_\star}$ and $F_{x_\star}$ are the components of the tire forces along the lateral and longitudinal vehicle axes. Front steering angle and distance between centre of gravity (CoG) to front and rear axles are given by δ_f , a and b respectively.

The detailed dynamic equations can be described by the following equations:

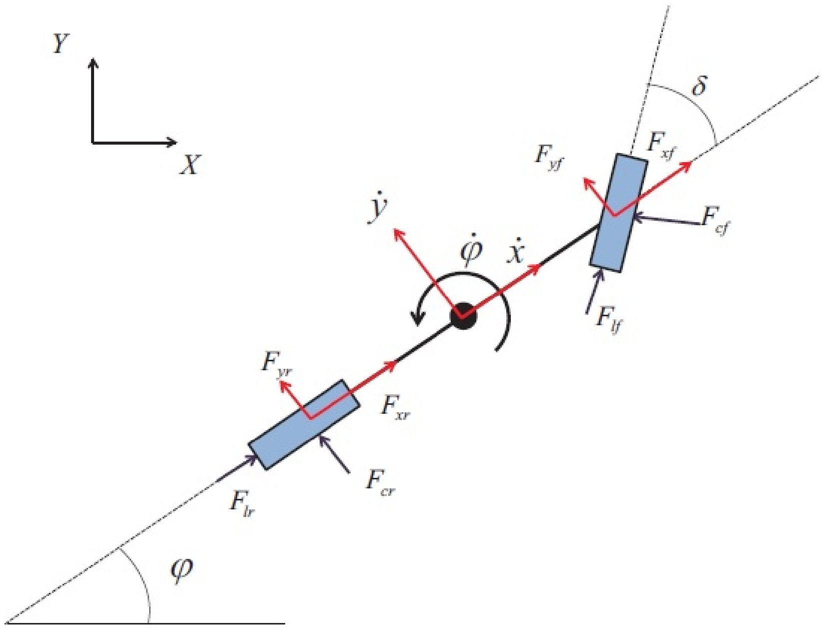

Figure A-1: Bicycle (one track) vehicle model for forces acting on a vehicle body fixed frame [5]

$$\begin{aligned} m\ddot{y} &= -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr} \\ m\ddot{x} &= m\dot{x}\dot{\psi} + 2F_{xf} + 2F_{xr} \\ I\ddot{\psi} &= 2a(F_{yf} + 2F_{yr}) \\ \dot{Y} &= \dot{x}sin\psi + \dot{y}cos\psi \\ \dot{X} &= \dot{x}cos\psi - \dot{y}sin\psi \end{aligned} \tag{A-2}$$

where m is vehicle mass and I denotes yaw inertia. $\dot{x}$ and $\dot{y}$ denote the vehicle longitudinal and lateral velocities, respectively, and $\dot{\psi}$is the yaw rate around a vertical axis at the vehicle CoG. Longitudinal position of the vehicle along the path is given by s. F_{yf} and F_{yr} are front and rear tire forces acting along the vehicle lateral axis, F_{xf} and F_{xr} forces acting along the vehicle longitudinal axis. The longitudinal and lateral tire force components $F_{x_\star}$ and $F_{y_\star}$ in the vehicle body frame are modelled as follows:

$$F_{y_{\star,\bullet}} = F_{l_{\star,\bullet}} sin\delta_\star + F_{c_{\star,\bullet}} cos\delta_\star F_{x_{\star,\bullet}} = F_{l_{\star,\bullet}} cos\delta_\star - F_{c_{\star,\bullet}} sin\delta_\star, \text{where} \star \in f, r \tag{A-3}$$

where steering angle is restricted to front steer only, thereby giving $\delta_f = \delta$.

The longitudinal and lateral tire forces $F_{l_\star}$ and $F_{c_\star}$ are represented as in Pacejka model. The non-linear functions of the tire slip angles $\alpha_\star$, slip ratios $\sigma_\star$, normal forces $F_{z,\star}$ and friction coefficient between the tire and road $\mu_\star$:

$$F_{l_\star} = f_l(\alpha_\star, \sigma_\star, F_{z,\star}, \mu_\star) \tag{A-4}$$

where slip angles are given by the equation:

$$\alpha_\star = \arctan \frac{v_{c_\star}}{v_{l_\star}} \tag{A-5}$$

Here, tire lateral and longitudinal velocity components $v_{c_\star}$ and $v_{l_\star}$ are derived using,

$$\begin{aligned} v_{c_\star} &= v_{y_\star} \cos \delta_\star - v_{x_\star} \sin \delta_\star \\ v_{l_\star} &= v_{y_\star} \sin \delta_\star + v_{x_\star} \cos \delta_\star \end{aligned} \tag{A-6}$$

where the tire lateral and longitudinal velocity components $v_{y_\star}$ and $v_{x_\star}$ can be derived from,

$$\begin{aligned} v_{y_f} &= \dot{y} + a\dot{\psi}, v_{x_f} = \dot{x} \\ v_{y_r} &= \dot{y} - b\dot{\psi}, v_{x_r} = \dot{x} \end{aligned} \tag{A-7}$$

The slip ratios $s_\star$ are approximated as follows:

$$s_\star = \begin{cases} \frac{r\omega_\star}{v_{l_\star}}, & \text{if } v_{l_\star} > r\omega_\star, \quad v \neq 0 \quad \text{for braking,} \\ 1 - \frac{v_{l_\star}}{r\omega_\star}, & \text{if } v_{l_\star} < r\omega_\star, \quad \omega \neq 0 \quad \text{for driving,} \end{cases}$$

The friction coefficient μ is taken from for the dry road surface from table . The normal force on the wheel are estimated with the non load transfer assumption, $F_{z_\star}$. Finally, considering g, g is the gravitational acceleration as well

$$F_{z_{f,\bullet}} = \frac{bmg}{2(a+b)}, F_{z_{r,\bullet}} = \frac{bmg}{2(a+b)} \tag{A-8}$$

The tire forces $F_{x_\star}$ and $F_{y_\star}$ are normal forces acting on one wheel and combined forces of the two wheels in the one track model.

Models can be rendered less complex by making assumptions, such as constant longitudinal velocity and small angle of deviation.

Appendix B

Appendix -2

B-1 Controller implementation in MATLAB

B-1-1 Main code

```
clear,clc,close all
warning off
%%run(’C:\tomlab\startup.m’)
%% Parameters
a = 1;                          % Distance from CoG to the front axles
b = 1.4;                        % Distance from CoG to the rear axles
c = 0.47;                       % Distance from CoG to the left/right
    side at the wheels
m = 2050;                       % Mass
J1 = 3344;                       % VehicleŠs rotational inertia about the
     z axis
Jw = (m*(1.75^2+4.48^2))/12;    % Wheel and driveline rotational inertias
rw = 0.41;                      % Radius of the wheel
bw = 3.8;                       % Damping coefficient
Cs = 0.12;                      % Cornering stiffness coefficient
    (0.12~0.16)
g = 9.81;
mu = 0.3;
%% Packeja model coefficients for different road surfaces
% 1 : Dry Tarmac
% 2 : Wet Tarmac
% 3 : Snow
% 4 : Ice
Road = 1;
switch Road
case 1
Bp = 10; Cp = 1.9; Dp = 1; Ep = 0.97;
case 2
```

```
Bp = 12; Cp = 2.3; Dp = 0.82; Ep = 1;
case 3
Bp = 5; Cp = 2; Dp = 0.3; Ep = 1;
case 4
Bp = 4; Cp = 2; Dp = 0.1; Ep = 1;
end
Param = [a,b,c,m,J1,Jw,rw,bw,rw,Cs,g,Bp,Cp,Dp,Ep,mu]';
%% Initial Conditions
xD0 = 20;
yD0 = 0;
PsiD0 = 0;
Psi0 = 0;
X0 = 0;
Y0 = 0.0001;
%% General Parameters
Hp = 30;
vs1 = 10; vsy1 = 0;
ys1 = 0; vsy2 = 0;
x1 = 80;
vs2 = 0;
ys2 = 3.2;
x2 = -100;
yref = 0;
vx_des = 20;
Ts = 0.1;
x_pm = [x1-X0,Y0,xD0,yD0,X0,ys1-Y0,0,0,ys2-Y0,x2-X0];
x_4w = [yD0,xD0,Psi0,PsiD0,Y0,X0];
Q = diag([30,120]);
R = diag([0.01,0.01,0.01]);
S = diag([0.01,0.001,0.001]);
T = 18;
%% High Level MPC Parameters
alpha = 20;
k = 15;
gamma = 5;
eps_x1f = 1;
eps_1f = 5;
eps_x1r = 0.4;
eps_1r = 1;
eps_x2f = 1;
eps_2f = 5;
eps_x2r = 0.4;
eps_2r = 1;
Par = [vs1,ys1,x1,yref,vx_des,Hp,alpha,k,gamma,eps_x1f,eps_1f,eps_x1r,eps_1r,vs2,ys2,x2,eps_x2f,eps_2f,eps_x2r,eps_2r,vsy1,vsy2];
u_k = zeros(1,3);
% FromXLS = xlsread('Scenarios.xlsx'); XLSSize = size(FromXLS); Count = XLSSize(1)+1;
%% Simulation
% 1 : Fast Path Replanning Simulation
% 2 : Entire Simulation
SimType = 2;
% Scenarios
% 1. No Obstacle,
% 2. Car following
% 3. Lane change
% 4. Static Obstacle
% 5. Moving Obstacle
% 6. Two Obstacles(optional)
Scenario = 5;
refmat = [];
tic
for i=1:T
    switch SimType
        case 1
            [w,w_x] = MPC_HL(x_pm(end,:),Par,[x1(end),x2(end),ys1(end),
                ys2(end)],Scenario);
            x_pm = [x_pm;w];
            x1 = [x1;w_x(:,1)];
            ys1 = [ys1;w_x(:,3)];
        case 2
            [w,w_x] = MPC_HL(x_pm(end,:),Par,[x1(end),x2(end),ys1(end),
                ys2(end)],Scenario);
            x_pm = [x_pm;w];
            x1 = [x1;w_x(:,1)];
            x2 = [x2;w_x(:,2)];
            ys1 = [ys1;w_x(:,3)];
            ys2 = [ys2;w_x(:,4)];
            ref = [w(:,3),w(:,2)];
            refmat = [refmat;ref];
            [w2,u_con] = MO_4w(x_4w(end,:),u_k(end,:),ref,Param,Hp,Ts,Q,R
                ,S);
            u_k = [u_k;u_con];
            x_4w = [x_4w;w2];
%             x_pm = [x1(end)-x_4w(end,6),x_4w(end,5),x_4w(end,2),x_4w(
    end,1),x_4w(end,6),ys1-x_4w(end,2),x_pm(end,7),x_pm(end,8)];
            x_pm(end,5) = x_4w(end,6);
    end
end
tim = toc;
%%
if SimType == 1
    Wid = 3;
    Len = 5;
    X = x_pm(:,5);
    figure
    subplot(211)
    plot(x1,ys1,'g*')
    hold on
    plot(X,x_pm(:,2),'LineWidth',2)
    hold on
    plot(X,yref*ones(length(X)),'-.r','LineWidth',2)
    hold on
    plot(X,-1.6*ones(length(X),1),'-black','LineWidth',3)
    hold on
    plot(X,4.8*ones(length(X),1),'-black','LineWidth',3)
    hold on
    plot(X,1.6*ones(length(X),1),'--black','LineWidth',2)
    hold on
    plot(X,0*ones(length(X),1),'-.black','LineWidth',1)
    hold on
    plot(X,3.2*ones(length(X),1),'-.black','LineWidth',1)
    hold on
    ylim([-2 7])
    xlim([0 150])
    legend('Surrounding vehicle','y','y_{ref}','location','northeast')
    xlabel('X [m]')
    ylabel('Y [m]')
    title(strcat('High Level Controller Test with vx_{des} = ',num2str(
        vx_des),' y_{ref} = ',num2str(yref)))
    subplot(212)
    plot(X,x_pm(:,3),'LineWidth',2)
    hold on
    plot(X,vx_des*ones(length(X)),'--r','LineWidth',2)
    grid on
    legend('vx','vx_{des}','location','southeast')
    xlabel('X [m]')
    ylabel('Vx [m/s]')
    break
end
%% Plotting
Yref = x_pm(:,2);
xDotref = x_pm(:,3);
X = x_4w(:,6);
Y = x_4w(:,5);
XDot = x_4w(:,2);
f1=figure;
subplot(2,2,[1,2])
plot(X,Y)
hold on
plot(X,Yref,'r')
hold on
plot(x1,ys1,'g*')
hold on
plot(x2,ys2,'g*')
hold on
plot(X,-1.6*ones(length(X),1),'-black','LineWidth',3)
hold on
plot(X,4.8*ones(length(X),1),'-black','LineWidth',3)
hold on
plot(X,1.6*ones(length(X),1),'--black','LineWidth',2)
hold on
plot(X,0*ones(length(X),1),'-.black','LineWidth',1)
hold on
plot(X,3.2*ones(length(X),1),'-.black','LineWidth',1)
ylim([-2 7])
xlabel('X [m]')
ylabel('Y [m]')
% grid on
```

```
legend('Y','Y_{ref}','location','northwest')
subplot(224)
plot(u_k(:,1))
xlabel('Time [s]')
ylabel('Steering Angle [rad]')
grid on
subplot(223)
plot(XDot)
hold on
plot(xDotref,'r')
legend('xdot','xdot_{ref}','location','southeast')
xlabel('Time [s]')
ylabel('Amplitude')
grid on
%%
f2=figure ;

Wid = 3;
Len = 5;
for i = 1:T*Hp
    if i ~=T*Hp
        hold off
    end
    h1=plot(X,-1.6*ones(length(X),1),'-black','LineWidth',3);
    hold on
    h2=plot(X,4.8*ones(length(X),1),'-black','LineWidth',3);
    hold on
    h3=plot(X,1.6*ones(length(X),1),'--black','LineWidth',2);
    hold on
    h4=plot(X,0*ones(length(X),1),'-.black','LineWidth',1);
    hold on
    h5=plot(X,3.2*ones(length(X),1),'-.black','LineWidth',1);
    xlim([0 max(X)+120])
    ylim([-4 10])
    h6=plot(X(1:i),Y(1:i),'*');
    hold on
    h7=plot(X(1:i),Yref(1:i),'r*');
    hold on
    h8 = rectangle('Position',[x1(i)-Len/2,ys1(i)-Wid/2,Len,Wid],'EdgeColor','green');
    hold on
    h9 = rectangle('Position',[x2(i)-Len/2,ys2(i)-Wid/2,Len,Wid],'EdgeColor','green');
    legend([h6,h8,h9],{strcat(['v_x=',num2str(X(i))]),strcat(['x_1=',num2str(x1(i))]),strcat(['x_2=',num2str(x2(i))])})
    pause(0.05)
    F = getframe(f2);

end
%% Scenarios XLS
%
```

```
ToXLS = [xD0,vx_des,x1(1),vs1,x2(1),vs2,Hp,Q(1,1),Q(2,2),R(1,1),R(2,2),R(3,3),S(1,1),S(2,2),S(3,3),alpha,k,gamma,eps_x1f,eps_1f,eps_x1r,eps_1r,eps_x2f,eps_2f,eps_x2r,eps_2r,T*Hp,tim];
print(f1,strcat(num2str(Count),'_1'),'-dpng')
print(f2,strcat(num2str(Count),'_2'),'-dpng')
xlswrite('Scenarios.xlsx',[Count,ToXLS],strcat('A',num2str(Count+1),':AC',num2str(Count+1)));
```

B-1-2 Higher level controller function

```
function [w,Obs] = MPC_HL(x,Par,xobs,Scenario)
vs1 = Par(1);vs2 = Par(14);vsy1 = Par(21);
ys1 = Par(2);ys2 = Par(15);vsy2 = Par(22);
yref = Par(4);
vx_des = Par(5);
x1 = Par(3);x2 = Par(16);
alpha = Par(7);   %Weight on desired velocity
k = Par(8);         %Weight on lane centre
gamma = Par(9);
Wl = 3.2;       %Length of the lane
Wc = 3;         %Width of surrounding vehicle
Lc = 5;         %Lenght of surrounding vehicle
sigma = 3.2;    %Lane centre adjacent to surrounding vehicle current lane
psi = 30;        %Yaw angle
v_sym = 1; g = 1; x_sym = 5e4; o_sym = 5e4;
theta_r = 1;    %Time headway rear
theta_f = 2;    %Time headway forward
c = 0.47; % Distance from CoG to the left/right side at the wheels
toms t
Hp = Par(6);
Ts = 0.1;
p1 = tomPhase('p1', t, 0, Hp*Ts, Hp);
setPhase(p1);
tomStates DX1 y vx vy X1 X DY1 DX2 DY2 X2 Y1 Y2
tomControls ax ay
W = 0.5*Wl+Wc;
% ++
eps_x1f = Par(10);eps_x2f = Par(17);
eps_1f = Par(11);eps_2f = Par(18);
% --
eps_x1r = Par(12);eps_x2r = Par(19);
eps_1r = Par(13);eps_2r = Par(20);
ceq = collocate({dot(DX1) == vs1 - vx;
    dot(DX2) == vs2 - vx;
    dot(y) == vy;
    dot(vx) == ax;
    dot(vy) == ay;
    dot(X) == vx;
    dot(DY1) == vsy1 - vy;
    dot(DY2) == vsy2 - vy;
    dot(X1) == vs1;
    dot(X2) == vs2;
    dot(Y1) == vsy1;
    dot(Y2) == vsy2;
    });
% DY1 = ys1-collocate(y);
Lf = collocate(vx)*theta_f+Lc;
Lr = collocate(vx)*theta_r+Lc;
x0 = {icollocate({vx == x(3); vy == x(4); y == x(2); DX1 == x(1); X == x(5); ax == x(7); ay == x(8); DY1 == x(6); DY2 == x(9); X1 == xobs(1); X2 == xobs(2); Y1 == xobs(3); Y2 == xobs(4); DX2 == x(10)})};
cbnd = initial({vx == x(3); vy == x(4); y == x(2); DX1 == x(1); X == x(5); ax == x(7); ay == x(8); DY1 == x(6); DY2 == x(9); X1 == xobs(1); X2 == xobs(2); Y1 == xobs(3); Y2 == xobs(4); DX2 == x(10)});
cbox = {
    -0.17*collocate(vx)<=collocate(vy)<=0.17*collocate(vx);

    -5 <= collocate(vy) <= 5;
    -1.6+c <= collocate(y) <= 4.8-c;
    -4 <= collocate(ax) <= 4;
    -2 <= collocate(ay) <= 2;
    -3 <= collocate(dot(ax)) <= 1.5;
    -0.5 <= collocate(dot(ay)) <= 0.5;
};
if Scenario ~=1 && Scenario ~=3
     if Scenario == 2
         cbox = {cbox;
         0 <= collocate(vx) <= 30};
     else
         cbox = {cbox;
         0 <= collocate(vx) <= 30;
         1 <= collocate(DX1)./Lf-collocate(DY1)./W-collocate(DX1).*eps_x1f+((-collocate(DY1)-sigma)./max(psi,abs(collocate(DX1))))+eps_1f};
         if yref == 0
         cbox = {cbox;
         collocate(DX1)./Lr+collocate(DY1)./W+collocate(DX1).*eps_x1r-((-collocate(DY1)-sigma)./max(psi,abs(collocate(DX1))))+eps_1r <= -1};
         end
     end
     if Scenario == 6
         cbox = {cbox;
         1 <= collocate(DX2)./Lf+collocate(DY2)./W-collocate(DX2).*eps_x2f+((-collocate(DY2)-sigma)./max(psi,abs(collocate(DX2))))+eps_2f;
         collocate(DX2)./Lr-collocate(DY2)./W+collocate(DX2).*eps_x2r-((-collocate(DY2)-sigma)./max(psi,abs(collocate(DX2))))+eps_2r <= -1};
     end
end
% objective = sum(norm(k*(collocate(vx)-vx_des))+norm(k*(collocate(y)-yref))+norm(gamma*collocate(vy))+norm(v_sym*collocate(ax))+norm(g*collocate(ay))+norm(x_sym*eps_1f)+norm(o_sym*eps_1r))
```

```
objective = norm(alpha*(collocate(vx)-vx_des))+norm(k*(collocate(y)-yref)
    )+norm(gamma*collocate(vy))+norm(v_sym*collocate(ax))+norm(g*collocate
    (ay));
options = struct;
options.derivatives = 'numerical';
options.use_H = 1;
options.use_d2c = 1;
Prob = sym2prob(objective,{cbox,cbnd,ceq},x0,options);
Prob.ADObj = 1;
Prob.ConsDiff = 1;
Result = tomRun ('SNOPT', Prob , 1);
solution = getSolution(Result);
w = [solution.DX1_p1(2:end),solution.y_p1(2:end),solution.vx_p1(2:end),
    solution.vy_p1(2:end),solution.X_p1(2:end),solution.DY1_p1(2:end),
    solution.ax_p1,solution.ay_p1,solution.DY2_p1(2:end),solution.DX2_p1
    (2:end)];
Obs = [solution.X1_p1(2:end),solution.X2_p1(2:end),solution.Y1_p1(2:end),
    solution.Y2_p1(2:end)];
tomCleanup(Prob);
```

B-1-3 Lower level controller function

```
function [x,u_con] = MO_4w(States,u_k,ref,Param,Hp,Ts,Q,R,S)
tf = Hp*Ts;
toms t
p = tomPhase('p', t, 0, tf, Hp);
setPhase(p);
tomStates yDot xDot Psi PsiDot Y X
tomControls u Fl Fr
a = Param(1);
b = Param(2);
c = Param(3);
m = Param(4);
J = Param(5);
Jw = Param(6);
r = Param(7);
bw = Param(8);
rw = Param(9);
Cs = Param(10);
g = Param(11);
Bp = Param(12);
Cp = Param(13);
Dp = Param(14);
Ep = Param(15);
mu = Param(16);
vy1 = yDot+a*PsiDot;
vy2 = yDot+a*PsiDot;
vy3 = yDot-b*PsiDot;
vy4 = yDot-b*PsiDot;
vx1 = xDot-c*PsiDot;
vx2 = xDot+c*PsiDot;
vx3 = xDot-c*PsiDot;
vx4 = xDot+c*PsiDot;
```

```
vc1 = vy1*cos(u)-vx1*sin(u);
vc2 = vy2*cos(u)-vx2*sin(u);
vc3 = vy3*cos(0)-vx3*sin(0);
vc4 = vy4*cos(0)-vx4*sin(0);
vl1 = vx1*cos(u)+vy1*sin(u);
vl2 = vx2*cos(u)+vy2*sin(u);
vl3 = vx3*cos(0)+vy3*sin(0);
vl4 = vx4*cos(0)+vy4*sin(0);
Fz1 = (b*m*g)/(2*a+2*b);
Fz2 = (b*m*g)/(2*a+2*b);
Fz3 = (a*m*g)/(2*a+2*b);
Fz4 = (a*m*g)/(2*a+2*b);
Cy1 = Cs*Fz1/2;
Cy2 = Cs*Fz2/2;
Cy3 = Cs*Fz3/2;
Cy4 = Cs*Fz4/2;
alpha1 = atan(vc1/(vl1));
alpha2 = atan(vc2/(vl2));
alpha3 = atan(vc3/(vl3));
alpha4 = atan(vc4/(vl4));
% Tfl = ifThenElse(u>0,0,-r*Fl);
% Tfr = ifThenElse(u>0,0,-r*Fr);
% Trl = ifThenElse(u>0,-r*Fl,0);
% Trr = ifThenElse(u>0,-r*Fr,0);
% omega_fl = (1/(Jw+bw))*(-mu*Fz1*rw+Tfl);
% omega_fr = (1/(Jw+bw))*(-mu*Fz2*rw+Tfr);
% omega_rl = (1/(Jw+bw))*(-mu*Fz3*rw+Trl);
% omega_rr = (1/(Jw+bw))*(-mu*Fz4*rw+Trr);
% s1 = ifThenElse(r*omega_fl<vl1,(r*omega_fl/vl1)-1,1-(vl1/(r*omega_fl)));
% s2 = ifThenElse(r*omega_fr<vl2,(r*omega_fr/vl2)-1,1-(vl2/(r*omega_rl)));
% s3 = ifThenElse(r*omega_rl<vl3,(r*omega_rl/vl3)-1,1-(vl3/(r*omega_fr)));
% s4 = ifThenElse(r*omega_rr<vl4,(r*omega_rr/vl4)-1,1-(vl4/(r*omega_rr)));
% mu1 = Dp.*sin(Cp.*atan(Bp.*s1-Ep*(Bp.*s1-atan(Bp.*s1))));
% mu2 = Dp.*sin(Cp.*atan(Bp.*s2-Ep*(Bp.*s2-atan(Bp.*s2))));
% mu3 = Dp.*sin(Cp.*atan(Bp.*s3-Ep*(Bp.*s3-atan(Bp.*s3))));
% mu4 = Dp.*sin(Cp.*atan(Bp.*s4-Ep*(Bp.*s4-atan(Bp.*s4))));
mu1 = 0.3; mu2 = 0.3; mu3 = 0.3; mu4 = 0.3;
Fl1 = mu1*Fz1;
Fl2 = mu2*Fz2;
Fl3 = mu3*Fz3;
Fl4 = mu4*Fz4;
Fc1 = alpha1*Cy1;
Fc2 = alpha2*Cy2;
Fc3 = alpha3*Cy3;
Fc4 = alpha4*Cy4;
Fy1 = Fl1*sin(u)+Fc1*cos(u);
Fy2 = Fl2*sin(u)+Fc2*cos(u);
Fy3 = Fl3*sin(0)+Fc3*cos(0);
Fy4 = Fl4*sin(0)+Fc4*cos(0);
```

```
% Fx1 = Fl1*cos(u)-Fc1*sin(u);
% Fx2 = Fl2*cos(u)-Fc2*sin(u);
% Fx3 = Fl3*cos(0)-Fc3*sin(0);
% Fx4 = Fl4*cos(0)-Fc4*sin(0);
ceq = collocate({dot(yDot) == (1/m)*(-m*xDot*PsiDot+Fy1+Fy2+Fy3+Fy4);
dot(xDot) == (1/m)*(m*yDot*PsiDot+Fl+Fr);
dot(Psi) == PsiDot;
dot(PsiDot) == (1/J)*(a*(Fy1+Fy2)-b*(Fy3+Fy4)+c*(-Fl+Fr));
dot(Y) == xDot*sin(Psi)+yDot*cos(Psi);
dot(X) == xDot*cos(Psi)-yDot*sin(Psi)});
%%
cbox = {
    -15e3 <= collocate(Fl) <= 15e3;
    -15e3 <= collocate(Fr) <= 15e3;
%     -10e3 <= collocate(dot(Fl)) <= 10e3;
%     -10e3 <= collocate(dot(Fr)) <= 10e3;
    -1.6+c <= collocate(Y) <= 4.8-c;
    deg2rad(-24) <= collocate(u) <= deg2rad(24);
    -0.17 <= collocate(dot(u)) <= 0.17;
    };
x0 = {icollocate({yDot == States(1); xDot == States(2); Psi == States(3);
     PsiDot == States(4); Y == States(5); X == States(6); u == u_k(1); Fl
    == u_k(2); Fr == u_k(3)})};
cbnd = initial({yDot == States(1); xDot == States(2); Psi == States(3);
    PsiDot == States(4); Y == States(5); X == States(6); u == u_k(1); Fl
    == u_k(2); Fr == u_k(3)});
objective = sum(norm(([collocate(xDot),collocate(Y)]-ref)*Q)+norm(([
    collocate(u),collocate(Fl),collocate(Fr)]*R))+norm(([collocate(dot(u))
    ,collocate(dot(Fl)),collocate(dot(Fr))]*S)));
%objective = sum(norm(([collocate(xDot),collocate(Y)]-[xd',tr'])*Q)+norm
    (([collocate(dot(u)),collocate(dot(Fl)),collocate(dot(Fr))]*S))); %+
    norm(([collocate(u),collocate(Fl),collocate(Fr)]*R)));\
options = struct;
Prob = sym2prob(objective,{cbox,cbnd,ceq},x0,options);
Prob.SOL.optPar(30) = 20000;
Prob.SOL.optPar(35) = 8000;
Prob.SOL.optPar(36) = 3500;
Result = tomRun ('SNOPT',Prob,1);
solution = getSolution (Result);
u_con = [solution.u_p,solution.Fl_p,solution.Fr_p];
x = [solution.yDot_p(2:end),solution.xDot_p(2:end),solution.Psi_p(2:end),
    solution.PsiDot_p(2:end),solution.Y_p(2:end),solution.X_p(2:end)];
```

B-1-4 Standalone lower level controller

```
clear,clc,close all

% Vehicle parameters
a = 1;                          % Distance from CoG to the front axles
b = 1.4;                        % Distance from CoG to the rear axles
c = 0.47;                       % Distance from CoG to the left/right
    side at the wheels
m = 2050;                       % Mass
```

```
J1 = 3344;                        % VehicleŠs rotational inertia about the z axis
Jw = (m*(1.75^2+4.48^2))/12;      % Wheel and driveline rotational inertias
rw = 0.41;                        % Radius of the wheel
bw = 3.8;                         % Damping coefficient
Cs = 0.12;                        % Cornering stiffness coefficient (0.12~0.16)
g = 9.81;
mu = 0.3;
Ts = 0.1;
%tr = [zeros(1,10),5*tanh(0:0.01:0.49),2.3*ones(1,50),5*tanh(0.49:-0.01:0),zeros(1,10)]; %double lane change
% tr = [zeros(1,10),5*tanh(0:0.01:0.49),2.3*ones(1,110)];                    %single lane change
%tr = [zeros(1,170)];  %straight line
%tr = sin
dl=0.5; % sampling of trajectory
tr=[zeros(1,12/dl+1),(dl:dl:13.5)*4/13.5,4*ones(1,11/dl),4-(dl:dl:12.5)*4/12.5,zeros(1,12/dl)];

%% Packeja model coefficients for different road surfaces
% 1 : Dry Tarmac
% 2 : Wet Tarmac
% 3 : Snow
% 4 : Ice
Road = 1;
switch Road
case 1
Bp = 10; Cp = 1.9; Dp = 1; Ep = 0.97;
case 2
Bp = 12; Cp = 2.3; Dp = 0.82; Ep = 1;
case 3
Bp = 5; Cp = 2; Dp = 0.3; Ep = 1;
case 4
Bp = 4; Cp = 2; Dp = 0.1; Ep = 1;
end
Param = [a,b,c,m,J1,Jw,rw,bw,rw,Cs,g,Bp,Cp,Dp,Ep,mu]';
xD0 = 20;
yD0 = 0;
PsiD0 = 0;
Psi0 = 0;
X0 = 0;
Y0 = 0;

%% General Parameters

Hp = length(tr);
States = [yD0,xD0,Psi0,PsiD0,Y0,X0];
u_k = zeros(3,1);
Q = diag([47.9,71.96]);
R = diag([0.3,0.1,0.1]);
S = diag([0.1028,0.0098,0.0154]);
xd = 20*ones(1,Hp);
```

```

%%
tf = Hp*Ts;
toms t
p = tomPhase('p', t, 0, tf, Hp);
setPhase(p);
tomStates yDot xDot Psi PsiDot Y X
tomControls u Fl Fr
a = Param(1);
b = Param(2);
c = Param(3);
m = Param(4);
J = Param(5);
Jw = Param(6);
r = Param(7);
bw = Param(8);
rw = Param(9);
Cs = Param(10);
g = Param(11);
Bp = Param(12);
Cp = Param(13);
Dp = Param(14);
Ep = Param(15);
mu = Param(16);
vy1 = yDot+a*PsiDot;
vy2 = yDot+a*PsiDot;
vy3 = yDot-b*PsiDot;
vy4 = yDot-b*PsiDot;
vx1 = xDot-c*PsiDot;
vx2 = xDot+c*PsiDot;
vx3 = xDot-c*PsiDot;
vx4 = xDot+c*PsiDot;
vc1 = vy1*cos(u)-vx1*sin(u);
vc2 = vy2*cos(u)-vx2*sin(u);
vc3 = vy3*cos(0)-vx3*sin(0);
vc4 = vy4*cos(0)-vx4*sin(0);
vl1 = vx1*cos(u)+vy1*sin(u);
vl2 = vx2*cos(u)+vy2*sin(u);
vl3 = vx3*cos(0)+vy3*sin(0);
vl4 = vx4*cos(0)+vy4*sin(0);
Fz1 = (b*m*g)/(2*a+2*b);
Fz2 = (b*m*g)/(2*a+2*b);
Fz3 = (a*m*g)/(2*a+2*b);
Fz4 = (a*m*g)/(2*a+2*b);
Cy1 = Cs*Fz1/2;
Cy2 = Cs*Fz2/2;
Cy3 = Cs*Fz3/2;
Cy4 = Cs*Fz4/2;
alpha1 = atan(vc1/(vl1));
alpha2 = atan(vc2/(vl2));
alpha3 = atan(vc3/(vl3));
alpha4 = atan(vc4/(vl4));
% Tfl = ifThenElse(u>0,0,-r*Fl);
% Tfr = ifThenElse(u>0,0,-r*Fr);
% Trl = ifThenElse(u>0,-r*Fl,0);
% Trr = ifThenElse(u>0,-r*Fr,0);
% omega_fl = (1/(Jw+bw))*(-mu*Fz1*rw+Tfl);
% omega_fr = (1/(Jw+bw))*(-mu*Fz2*rw+Tfr);
% omega_rl = (1/(Jw+bw))*(-mu*Fz3*rw+Trl);
% omega_rr = (1/(Jw+bw))*(-mu*Fz4*rw+Trr);
% s1 = ifThenElse(r*omega_fl<vl1,(r*omega_fl/vl1)-1,1-(vl1/(r*omega_fl)))
    ;
% s2 = ifThenElse(r*omega_fr<vl2,(r*omega_fr/vl2)-1,1-(vl2/(r*omega_rl)))
    ;
% s3 = ifThenElse(r*omega_rl<vl3,(r*omega_rl/vl3)-1,1-(vl3/(r*omega_fr)))
    ;
% s4 = ifThenElse(r*omega_rr<vl4,(r*omega_rr/vl4)-1,1-(vl4/(r*omega_rr)))
    ;
% mu1 = Dp.*sin(Cp.*atan(Bp.*s1-Ep*(Bp.*s1-atan(Bp.*s1))));
% mu2 = Dp.*sin(Cp.*atan(Bp.*s2-Ep*(Bp.*s2-atan(Bp.*s2))));
% mu3 = Dp.*sin(Cp.*atan(Bp.*s3-Ep*(Bp.*s3-atan(Bp.*s3))));
% mu4 = Dp.*sin(Cp.*atan(Bp.*s4-Ep*(Bp.*s4-atan(Bp.*s4))));
mu1 = 0.3; mu2 = 0.3; mu3 = 0.3; mu4 = 0.3;
Fl1 = mu1*Fz1;
Fl2 = mu2*Fz2;
Fl3 = mu3*Fz3;
Fl4 = mu4*Fz4;
Fc1 = alpha1*Cy1;
Fc2 = alpha2*Cy2;
Fc3 = alpha3*Cy3;
Fc4 = alpha4*Cy4;
Fy1 = Fl1*sin(u)+Fc1*cos(u);
Fy2 = Fl2*sin(u)+Fc2*cos(u);
Fy3 = Fl3*sin(0)+Fc3*cos(0);
Fy4 = Fl4*sin(0)+Fc4*cos(0);
% Fx1 = Fl1*cos(u)-Fc1*sin(u);
% Fx2 = Fl2*cos(u)-Fc2*sin(u);
% Fx3 = Fl3*cos(0)-Fc3*sin(0);
% Fx4 = Fl4*cos(0)-Fc4*sin(0);
ceq = collocate({dot(yDot) == (1/m)*(-m*xDot*PsiDot+Fy1+Fy2+Fy3+Fy4);
dot(xDot) == (1/m)*(m*yDot*PsiDot+Fl+Fr);
dot(Psi) == PsiDot;
dot(PsiDot) == (1/J)*(a*(Fy1+Fy2)-b*(Fy3+Fy4)+c*(-Fl+Fr));
dot(Y) == xDot*sin(Psi)+yDot*cos(Psi);
dot(X) == xDot*cos(Psi)-yDot*sin(Psi)});
%%
cbox = {
    -15e3 <= collocate(Fl) <= 15e3;
    -15e3 <= collocate(Fr) <= 15e3;
%      -10e3 <= collocate(dot(Fl)) <= 10e3;
%      -10e3 <= collocate(dot(Fr)) <= 10e3;
    -1.6+c <= collocate(Y) <= 4.8-c;
    deg2rad(-24) <= collocate(u) <= deg2rad(24);
    -0.17 <= collocate(dot(u)) <= 0.17;
    };
```

```
x0 = {icollocate({yDot == States(1); xDot == States(2); Psi == States(3); PsiDot == States(4); Y == States(5); X == States(6); u == u_k(1); Fl == u_k(2); Fr == u_k(3)})};
cbnd = initial({yDot == States(1); xDot == States(2); Psi == States(3); PsiDot == States(4); Y == States(5); X == States(6); u == u_k(1); Fl == u_k(2); Fr == u_k(3)});
objective = sum(norm(([collocate(xDot),collocate(Y)]-[xd',tr'])*Q)+norm(([collocate(dot(u)),collocate(dot(Fl)),collocate(dot(Fr))]*S))); %+norm(([collocate(u),collocate(Fl),collocate(Fr)]*R)));
options = struct;
Prob = sym2prob(objective,{cbox,cbnd,ceq},x0,options);
Prob.SOL.optPar(30) = 80000;
Prob.SOL.optPar(35) = 8000;
Prob.SOL.optPar(36) = 3500;
Result = tomRun ('SNOPT',Prob,1);
solution = getSolution (Result);
u_con = [solution.u_p,solution.Fl_p,solution.Fr_p];
x = [solution.yDot_p(2:end),solution.xDot_p(2:end),solution.Psi_p(2:end),solution.PsiDot_p(2:end),solution.Y_p(2:end),solution.X_p(2:end)];

%% plotting

% Y
figure
plot((0:length(tr)-1)*dl,x(:,5))
xlabel('Lateral position [m]')
ylabel('Longitudinal position [m]')
title('Position')

% Psi
figure
subplot(211)
plot((0:length(tr)-1)*dl,x(:,3))
xlabel('Lateral position [m]')
ylabel('\psi')
title('Yaw')
subplot(212)
plot((0:length(tr)-1)*dl,x(:,4))
xlabel('Lateral position [m]')
ylabel('d\psi')
title('Yaw rate')

% Steering
figure
subplot(211)
plot((0:length(tr)-1)*dl,solution.u_p*180/pi)
xlabel('Lateral position [m]')
ylabel('deg')
title('Steering')
subplot(212)
plot((0:length(tr)-1)*dl,gradient(solution.u_p)*180/pi)
xlabel('Lateral position [m]')
ylabel('deg/s')
```

```
title('Steering rate')

% Breaking
figure
Tfl = zeros(size(solution.Fl_p));
Tfr = zeros(size(solution.Fl_p));
Trl = zeros(size(solution.Fl_p));
Trr = zeros(size(solution.Fl_p));
[a1,a2,a3,a4]=generate_breaks(Param,x,solution.u_p);
af=(a1+a2)/2;
ar=(a3+a4)/2;
for i=1:length(Tfl)
    if solution.Fl_p(i)==0 && solution.Fr_p(i)==0
        Tfl(i)=0;Tfr(i)=0;Trl(i)=0;Trr(i)=0;
    else
        if af(i)-ar(i)>0
            %understeering
            Tfl(i)=0;Tfr(i)=0;Trl(i)=-rw*solution.Fl_p(i);Trr(i)=-rw*solution.Fr_p(i);
        else
            %oversteering
            Tfl(i)=-rw*solution.Fl_p(i);Tfr(i)=-rw*solution.Fr_p(i);Trl(i)=0;Trr(i)=0;
        end
    end
end
subplot(411)
plot((0:length(tr)-1)*dl,abs(Tfl))
xlabel('Lateral position [m]')
ylabel('Torque[Nm]')
title('Front left breaking')
subplot(412)
plot((0:length(tr)-1)*dl,abs(Tfr))
xlabel('Lateral position [m]')
ylabel('Torque[Nm]')
title('Front right breaking')
subplot(413)
plot((0:length(tr)-1)*dl,abs(Trl))
xlabel('Lateral position [m]')
ylabel('Torque[Nm]')
title('Back left breaking')
subplot(414)
plot((0:length(tr)-1)*dl,abs(Trr))
xlabel('Lateral position [m]')
ylabel('Torque[Nm]')
title('Back right breaking')

figure
plot((0:length(tr)-1)*dl,x(:,5))
hold on
plot((0:length(tr)-1)*dl,tr,'r')
legend('Y', 'Y_{ref}')
xlabel('Lateral position [m]')
```

```
ylabel('Longitudinal position [m]')
ylim([-5,10])
% hold on
% plot(170,-1.6*ones(170,1),'-black','LineWidth',3)
% hold on
% plot(170,4.8*ones(170,1),'-black','LineWidth',3)
% hold on
% plot(170,1.6*ones(170,1),'--black','LineWidth',2)
% hold on
% plot(170,0*ones(170,1),'-.black','LineWidth',1)
% hold on
% plot(170,3.2*ones(170,1),'-.black','LineWidth',1)
title({['No.iter: ',num2str(Prob.SOL.optPar(30)),', Parameters: ']; num2str([diag(Q)',diag(S)'])})

%Cone additions

% Cones in segment 1
mw=5; %marker width
yl1=-1.5; %y-location of markers in first segment
plot([2,2,7,7,12,12],yl1*[-1,1,-1,1,-1,1],'ko','LineWidth',1,'MarkerEdgeColor','k',...
                'MarkerFaceColor',[1 .5 0],'MarkerSize',8)

% Cones in segment 2
yl2=-1.5; %y-location of markers in second segment
plot([25.5,25.5,31,31,36.5,36.5],max(tr)+yl2*[-1,1,-1,1,-1,1],'ko','LineWidth',1,'MarkerEdgeColor','k',...
                'MarkerFaceColor',[1 .5 0],'MarkerSize',8)

% Cones in segment 3third segment
yl3=-1.5; %y-location of markers in second segment
plot([49,49,55,55,61,61],yl3*[-1,1,-1,1,-1,1],'ko','LineWidth',1,'MarkerEdgeColor','k',...
                'MarkerFaceColor',[1 .5 0],'MarkerSize',8)
```

Bibliography

[1] J. Akerman, D. Banister, K. Dreborg, P. Nijkamp, R. Schleicher-Tappeser, D. Stead, and P. Steen, *European transport policy and sustainable mobility.* Routledge, 2000.

[2] J. M. Anderson, K. Nidhi, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola, *Autonomous vehicle technology: A guide for policymakers.* Rand Corporation, 2014.

[3] H. B. Pacejka, "Tire and vehicle dynamics, society of automotive engineers," *Inc, Warrendale, USA*, 2002.

[4] T. Litman, "Autonomous vehicle implementation predictions," *Victoria Transport Policy Institute*, vol. 28, 2014.

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

[6] R. Okuda, Y. Kajiwara, and K. Terashima, "A survey of technical trend of adas and autonomous driving," in *VLSI Technology, Systems and Application (VLSI-TSA), Proceedings of Technical Program-2014 International Symposium on*, pp. 1–4, IEEE, 2014.

[7] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *ASME 2010 Dynamic Systems and Control Conference*, pp. 265–272, American Society of Mechanical Engineers, 2010.

[8] S. Motoyama, H. Uki, K. ISODA Manager, and H. YUASA Manager, "Effect of traction force distribution control on vehicle dynamics," *Vehicle System Dynamics*, vol. 22, no. 5-6, pp. 455–464, 1993.

[9] D. D. Hrovat and M. N. Tran, "Method for controlling yaw of a wheeled vehicle based on under-steer and over-steer containment routines," 1996. US Patent 5,576,959.

[10] U. Ozguner, T. Acarman, and K. Redmill, *Autonomous ground vehicles.* Artech House, 2011.

[11] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, “Autonomous driving in urban environments: approaches, lessons and challenges,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[12] M. L. Sichitiu and M. Kihl, “Inter-vehicle communication systems: a survey,” *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 2, pp. 88–105, 2008.

[13] E. M. Rogers and D. L. Kincaid, “Communication networks: toward a new paradigm for research.,” 1981.

[14] S. Prosser, “Automotive sensors: past, present and future,” in *Journal of Physics: Conference Series*, vol. 76, p. 012001, IOP Publishing, 2007.

[15] N. H. T. S. Administration *et al.*, “Preliminary statement of policy concerning automated vehicles,” *Washington, DC*, 2013.

[16] E. F. Camacho and C. Bordons, *Model predictive control in the process industry.* Springer Science & Business Media, 2012.

[17] M. L. Darby and M. Nikolaou, “Mpc: Current practice and challenges,” *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, 2012.

[18] M. Morari and J. H. Lee, “Model predictive control: past, present and future,” *Computers & Chemical Engineering*, vol. 23, no. 4, pp. 667–682, 1999.

[19] H. Peng, “Vehicle lateral control for highway automation,” 1992.

[20] D.-I. M. Keller, I. C. Haß, I. A. Seewald, *et al.*, “A vehicle lateral control approach for collision avoidance by emergency steering maneuvers,” in *6th International Munich Chassis Symposium 2015*, pp. 175–197, Springer, 2015.

[21] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, “Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons,” *Control Systems Technology, IEEE Transactions on*, vol. 8, no. 4, pp. 695–708, 2000.

[22] G. Burgio and P. Zegelaar, “Integrated vehicle control using steering and brakes,” *International Journal of Control*, vol. 79, no. 05, pp. 534–541, 2006.

[23] R. Bishop, *Intelligent vehicle technology and trends.* 2005.

[24] R. Behringer, B. Gregory, V. Sundareswaran, R. Addison, R. Elsley, W. Guthmiller, and J. Demarche, “Development of an autonomous vehicle for the darpa grand challenge,” in *IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, 2004.

[25] P. Kirwan, “This car drives itself,” *Wired UK*, 2012.

[26] “Bmw autonomous car, or how we drifted into love with a robot.,” *http://www.engadget.com/2014/01/08/highly-automateddriving- bmw/.*, 2014.

[27] R. Sheppard, “Tesla motors.,” *https://ryansheppard.me/static/files/tesla.pdf*, 2014.

[28] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[29] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, “An mpc/hybrid system approach to traction control,” *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 3, pp. 541–552, 2006.

[30] B. H. Krogh and C. E. Thorpe, “Integrated path planning and dynamic steering control for autonomous vehicles,” in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 1664–1669, IEEE, 1986.

[31] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*, vol. 26. Birkhäuser, 2012.

[32] M. Diehl, “Real-time optimization for large scale nonlinear processes,” 2001.

[33] D. Mayne, “Nonlinear model predictive control: Challenges and opportunities,” in *Nonlinear model predictive control*, pp. 23–44, Springer, 2000.

[34] E. Frazzoli, M. A. Dahleh, and E. Feron, “Real-time motion planning for agile autonomous vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[35] B. Cao, G. Doods, and G. W. Irwin, “Time-optimal and smooth constrained path planning for robot manipulators,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 1853–1858, IEEE, 1994.

[36] R. Fletcher and M. J. Powell, “A rapidly convergent descent method for minimization,” *The Computer Journal*, vol. 6, no. 2, pp. 163–168, 1963.

[37] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[38] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 1398–1404, IEEE, 1991.

[39] V. T. Minh, “Trajectory generation for autonomous vehicles,” in *Mechatronics 2013*, pp. 615–626, Springer, 2014.

[40] C. Dever, B. Mettler, E. Feron, J. Popovic, and M. McConley, “Nonlinear trajectory generation for autonomous vehicles via parametrized maneuver classes,” *Journal of guidance, control, and dynamics*, vol. 29, no. 2, pp. 289–302, 2006.

[41] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, “Constraint-based planning and control for safe, semi-autonomous operation of vehicles,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 383–388, IEEE, 2012.

[42] O. Von Stryk and R. Bulirsch, “Direct and indirect methods for trajectory optimization,” *Annals of operations research*, vol. 37, no. 1, pp. 357–373, 1992.

[43] B. d’Andrea Novel, G. Bastin, and G. Campion, “Dynamic feedback linearization of nonholonomic wheeled mobile robots,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp. 2527–2532, IEEE, 1992.

[44] H. Chen and F. Allgower, “A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability,” in *Control Conference (ECC), 1997 European*, pp. 1421–1426, IEEE, 1997.

[45] A. Bemporad, A. Casavola, and E. Mosca, “Nonlinear control of constrained linear systems via predictive reference management,” *Automatic Control, IEEE Transactions on*, vol. 42, no. 3, pp. 340–349, 1997.

[46] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, “State and output feedback nonlinear model predictive control: An overview,” *European journal of control*, vol. 9, no. 2, pp. 190–206, 2003.

[47] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, “Mpc-based approach to active steering for autonomous vehicle systems,” *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265–291, 2005.

[48] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 3, pp. 566–580, 2007.

[49] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat, “Predictive control approach to autonomous vehicle steering,” in *American Control Conference, 2006*, pp. 6–pp, IEEE, 2006.

[50] R. C. Arkin, *Behavior-based robotics.* MIT press, 1998.

[51] L. Vlacic, M. Parent, and F. Harashima, *Intelligent vehicle technologies: theory and applications.* Butterworth-Heinemann, 2001.

[52] T. Toledo, H. N. Koutsopoulos, and M. Ben-Akiva, “Integrated driving behavior modeling,” *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 2, pp. 96–112, 2007.

[53] S. Adams and L. Yu, “An evaluation of traffic simulation models for supporting its development,” *Center for Transportation Training and Research, Texas Southern University, Tech. Rep*, 2000.

[54] J. N. Darroch and R. W. Rothery, “Car following and spectral analysis,” *Publication of: Traffic Flow and Transportation*, 1997.

[55] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *Robomech Journal*, vol. 1, no. 1, p. 1, 2014.

[56] P. P. Ramanata, *Optimal vehicle path generator using optimization methods.* PhD thesis, Virginia Polytechnic Institute and State University, 1998.

[57] M. Short, M. J. Pont, and Q. Huang, “Simulation of vehicle longitudinal dynamics,” *Embedded System Laboratory University of Leicester, Safety and Reliability of Distributed Embedded Systems. Leicester*, 2004.

[58] N. D. Smith, “Understanding parameters influencing tire modeling,” *Department of Mechanical Engineering, Colorado State University*, 2004.

[59] N. J. Nilsson, “Shakey the robot,” tech. rep., DTIC Document, 1984.

[60] H. P. Moravec, *The Stanford cart and the CMU rover.* Springer, 1990.

[61] L.-J. Lin, “Reinforcement learning for robots using neural networks,” tech. rep., DTIC Document, 1993.

[62] E. Gat *et al.*, “On three-layer architectures,” *Artificial intelligence and mobile robots*, vol. 195, p. 210, 1998.

[63] T. Schouwenaars, *Safe trajectory planning of autonomous vehicles.* PhD thesis, Massachusetts Institute of Technology, 2005.

[64] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*, vol. 56. springer, 2009.

[65] T. Ayres, L. Li, D. Schleuning, and D. Young, “Preferred time-headway of highway drivers,” in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 826–829, IEEE, 2001.

[66] C. W. Runyan, “Using the haddon matrix: introducing the third dimension,” *Injury prevention*, vol. 21, no. 2, pp. 126–130, 2015.

[67] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19. Siam, 2010.

[68] K. Holmström, A. O. Göran, and M. M. Edvall, “UserâĂŹs guide for tomlab/snopt1,” 2008.

[69] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat, “Low complexity mpc schemes for integrated vehicle dynamics control problems,” in *9th international symposium on advanced vehicle control*, 2008.