Model Predictive Control as a Function for Trajectory Control during High Dynamic Vehicle Maneuvers considering Actuator Constraints

Master Thesis

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc.

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Tarun Bollineni
Student ID: 421968
Date:

Academic Supervisor 1: Prof. Dr. Wolfram. Hardt
Academic Supervisor 2: Mr. Owes Khan
External Supervisor: Dipl.-Ing.(Univ.) Florian Dauth

Confidentiality

Sperrvermerk Die vorliegende Abschlussarbeit mit dem *Titel "**Model Predictive Control as a function for trajectory control during high dynamic vehicle maneuvers considering actuator constraints**"* enthält schützenswerte Informationen der ZF Friedrichshafen AG.

Die Veröffentlichung, Vervielfältigung -auch in digitaler Form-, Weitergabe -auch auszugsweise- sowie das Einstellen dieser Abschlussarbeit in eine Hochschulbibliothek oder öffentlich zugängliche Bibliothek sowie jede weitere öffentliche Zugänglichmachung sind vor Ablauf der unten angegebenen Sperrfrist ohne schriftliche Zustimmung der ZF Friedrichshafen AG nicht gestattet. Ausgenommen von diesem Zustimmungserfordernis ist eine Verwendung der Abschlussarbeit, die aufgrund der jeweils anwendbaren Prüfungs- oder Studienordnung zwingend erforderlich ist.

Die Sperrfrist gilt mit Abgabe der Abschlussarbeit an der Hochschule 3 Jahre.

Der Abgabetermin an der Hochschule für diese Arbeit ist am

Ort, Datum Unterschrift Student/-in

Acknowledgements

The Master thesis at ZF Friedrichshafen AG in the department of DAC (Vehicle Dynamics Control) was successfully completed according to the rules and regulations of Technical University Chemnitz.

Firstly, I would express my gratitude towards the Informatik department of University for their relentless support during the course of the thesis. I would like to thank Prof. Dr. W. Hardt for accepting my thesis under this department. Mr. Owes Khan has been an incredible support and mentor in terms of definition of the research goals and orientation of the thesis work. His inputs have been very helpful in getting the right direction during research.

Also, my sincere thanks to my supervisor Mr. Florian Dauth who provided me this wonderful opportunity of doing my thesis at ZF Friedrichshafen AG and has helped me gain industrial exposure. I would like to express deep gratitude for his guidance, encouragement and gracious support throughout the thesis, for his expertise in the field that motivated me to work and his faith in me at every stage of thesis. His support and supervision helped me gain a lot of knowledge leading to its successful completion. Further I would thank all the department colleagues who have helped me in difficult situations.

Last but not least, I would like to express my warm regards to my family and friends for their support and motivation that has helped me to complete the research goals in stipulated time.

Abstract

Autonomous driving is a rapidly growing field and can bring significant transition in mobility and transportation. In order to cater a safe and reliable autonomous driving operation, all the systems concerning with perception, planning and control has to be highly efficient. MPC is a control technique used to control vehicle motion by controlling actuators based on vehicle model and its constraints. The uniqueness of MPC compared to other controllers is its ability to predict future states of the vehicle using the derived vehicle model. Due to the technological development & increase in computational capacity of processors and optimization algorithms MPC is adopted for real-time application in dynamic environments. This research focuses on using Model predictive Control (MPC) to control the trajectory of an autonomous vehicle controlling the vehicle actuators for high dynamic maneuvers. Vehicle Models considering kinematics and vehicle dynamics is developed. These models are used for MPC as prediction models and the performance of MPC is evaluated. MPC trajectory control is performed with the minimization of cost function and limiting constraints. MATLAB/Simulink is used for designing trajectory control system and interfaced with CarMaker for evaluating controller performance in a realistic simulation environment. Performance of MPC with kinematic and dynamic vehicle models for high dynamic maneuvers is evaluated with different speed profiles.

Keywords: Model Predictive control, Trajectory Control, Vehicle Model, Constraints, High Dynamic Maneuver

Contents

List of Figures

List of Abbreviations

MPC	Model Predictive Control
LQR	Linear Quadratic Regulator
GPS	Global Positioning System
CNN	Convolutional Neural Network
AI	Artificial Intelligence
PIP	Proportional Integral Derivative
LIDAR	Light Detection and Ranging
MIMO	Multiple Input Multiple Output
LQG	Linear Quadratic Gaussian
QR	Quadratic Programming
LR	Linear Programming
CoG	Center of Gravity
SiL	Software in Loop
HiL	Hardware in Loop
MiL	Model in Loop
ACC	Adaptive Cruise Control
ABS	Anti-lock Braking System
ARS	Active Roll Stabilization
API	Application Program Interface
GUI	Graphical User Interface
CM4SL	CarMaker for Simulink
ISO	International Organization for Standardization
QP	Quadratic Programming
LP	Linear Programming

1 About ZF

ZF is a global leading automotive supplier for chassis, driveline, active safety and passive safety technology. In an eventful history, ZF has seized its entrepreneurial opportunities and developed from its roots as a supplier specialized in the aviation industry to a global mobility technology company. ZF is controlled by a non-profit organization from the city of Friedrichshafen that dates back to the beginning of 20th century. ZF was founded in 1915 to develop precision cog-wheels for Zeppelin airships. The company was almost destroyed during World War II for its involvement in the production of transmissions and steering systems for war machines. The success story of ZF took its initiation after the dissolution by Allies to end the war. In 1959 at São Paulo, Brazil the first international production plant was established. ZF gained leadership in technology and several business areas by expanding their production network globally. Cooperation and merging with other reputed companies enabled a stable and continuous growth. 'Sachs' was purchased in 2001 and in 2011 it is incorporated with ZF Friedrichshafen AG. In 2013 acquisition of 'TRW Automotive', an American auto parts manufacturer made ZF to be ranked as the second largest automotive supplier in 2017. Currently, ZF is established across 40 countries with 136,820 employees at 230 plants. ZF is listed among the top 10 applicants for patents within Germany for filing 1200 patent applications in 2016.

ZF empowers vehicles to see, think and act. The company invests more than six percent of its annual sales in research and development. The main focus of research is in the field of autonomous driving and electrification of vehicles. The advancement in mobility and services related to passenger cars, commercial vehicles and industrial technology applications are also prime focus areas for ZF.

The division DAC (Vehicle Dynamics Control) is focused on advanced engineering of innovative products, identification and development of new technologies, concepts and functional principles in the areas of chassis, steering and full vehicle control.

2 Introduction

2.1 Motivation

The emergence of autonomous vehicles is evident since the last few years. Autonomous vehicles have significant impact on society as well as automotive companies. The demand for efficient, reliable and safe mobility has aroused interest in research and development of autonomous vehicles. In order to achieve fully autonomous driving capability of a vehicle, technological developments in terms of perception, planning and control approaches is required. However recent developments in the fields of sensor technology and artificial intelligence have systems capable of sensing the environment and use the perception information for motion planning. The research is mainly on analysing and understanding the perception of sensors, planning algorithms and control hardware. The perception information from improved sensor hardware along with mapping and localization approaches are used make accurate decisions for selection, planning and control of vehicle motion to have appropriate trajectories for certain maneuvers [1]. The control systems responsible for handling vehicle motion along with the stabilization level of autonomous vehicle architecture is capable of making the vehicle to have motion towards the desired path considering the state of vehicle [2]. The control systems handling vehicle motion must be able to generate motion as close as possible to the planned motion. There are several control approaches for controlling vehicle motion out of which MPC (Model Predictive Control) appears to be a unique and reliable control approach because of its ability to predict future states based on the current state of the vehicle. The inbuilt capability of handling constraints enables MPC to be used for vehicle motion control. Especially for trajectory control the state of the vehicle and constraints have an influence on vehicle motion. A motion controller should consider the dynamics of vehicle and its limitations for controlling motion. MPC control technique has the capability to consider the vehicle dynamics for generating control actuations. Generally vehicle dynamics are non-linear, hence the non-linear vehicle behavior is considered while predicting the future states by MPC making model predictive controller a robust approach than traditional control approaches. Especially for trajectory control of an autonomous car a detailed representation of the vehicle environment along with the internal states and capabilities of the vehicle is considered [2]. The autonomous car trajectory control is one of the most arduous automation challenges due to the non-linearity of vehicle dynamics and motion constraints. Several control techniques are developed for motion control in autonomous vehicles. For autonomous driving the lateral and longitudinal dynamics of the autonomous vehicle have to be controlled to have a safe, comfort and reliable motion. The change in state of the vehicle will have significant influence on the vehicle. To handle this decoupled control technique is adopted. The main advantage of decoupled control is to isolate the changes in states that influence both inputs and outputs of a system responsible for trajectory control [3]. Decoupled control technique enables to use separate controllers for handling lateral and longitudinal dynamics of the vehicle. Most widely used control systems for motion control of an autonomous vehicle are PID (Proportional-Integral-Derivative), LQR (Linear Quadratic Regulator) and MPC (Model Predictive Control). PID and LQR controllers perform better for only linear systems. Since the vehicle dynamics are non-linear, In recent times MPC is robust than other control techniques for autonomous vehicle motion control due to its ability of predicting the future states of the vehicle, handing nonlinearities and constraints.

The motivation of master thesis is to design and implementation of MPC control algorithm for motion control of autonomous car to achieve best possible trajectories for high dynamic maneuvers considering actuator constraints

2.2 Thesis Objectives

The thesis work focuses on implementing MPC for controlling the motion of autonomous car to have desired trajectory for high dynamic maneuvers. The control of vehicle is dependent on Actuator Constraints, Vehicle State and Target Trajectory. In order to achieve the global objective of the thesis some sub-goals have been proposed:

- Modelling of vehicle model for simulation
- Modelling and implementation of kinematic vehicle model
- Modelling and implementation of dynamic vehicle model
- Definition of a test framework for high dynamic vehicle maneuvers
- Implementation of the overall system for trajectory control in MATLAB/Simulink
- Develop a trajectory planner to provide necessary target variables to control algorithm
- Implementation of MPC control algorithm in MATLAB/Simulink
- Testing the MPC control algorithm for kinematic and dynamic vehicle models
- Creating an interface with a realistic driving simulator implemented in CarMaker

3 State of Art

3.1 Autonomous Driving Background

It is evident that we are on the edge of revolution for mobility. Autonomous cars play a vital role in enabling cost effective, safe and secured transportation in the near future. The emerging research areas show that the shift from conventional to autonomous driving is no longer a dream. It is evident that autonomous cars will bring unanticipated change in the realm of transportation. Considering the diversity in perspectives regarding mobility the main concern is acceptability of the individuals and society for autonomous cars. For users to accept mobility using autonomous cars, it raises new challenges concerned to comfort, safety and reliability. A safe and comfortable driving behavior is a basic necessity for the universal acceptance of autonomous driving. In order to cater a comfortable and reliable driving behavior of an autonomous car that mimics the human driving behavior, several systems and functions have to be developed using the latest technologies available. Technological advancements in autonomous systems, increase in the computation power and decrease in cost of the sensing, control, computing technologies along with the drastic developments of software systems of autonomous driving clearly shows that autonomous driving is the future of mobility [4]. Autonomous driving is divided into three modules as shown in Figure 1:

Figure 1: Autonomous driving major modules

- **Sensing:** The sensing module uses various perception and sensing devices such as sensors to retrieve the information of the environment and surroundings that can be used for planning.
- **Planning:** The planning module is responsible for generation of behavior and decision making. It generates feasible, safe and comfortable paths, driving scenarios using the information from sensing module and configures the controllers accordingly.

- **Control:** The control module is responsible for controlling the autonomous vehicle to follow the desired route, path or trajectory that is planned by the planning module. It guides the vehicle to have motion satisfying the dynamics of vehicle and constraints.

3.2 Autonomous Vehicle Architecture

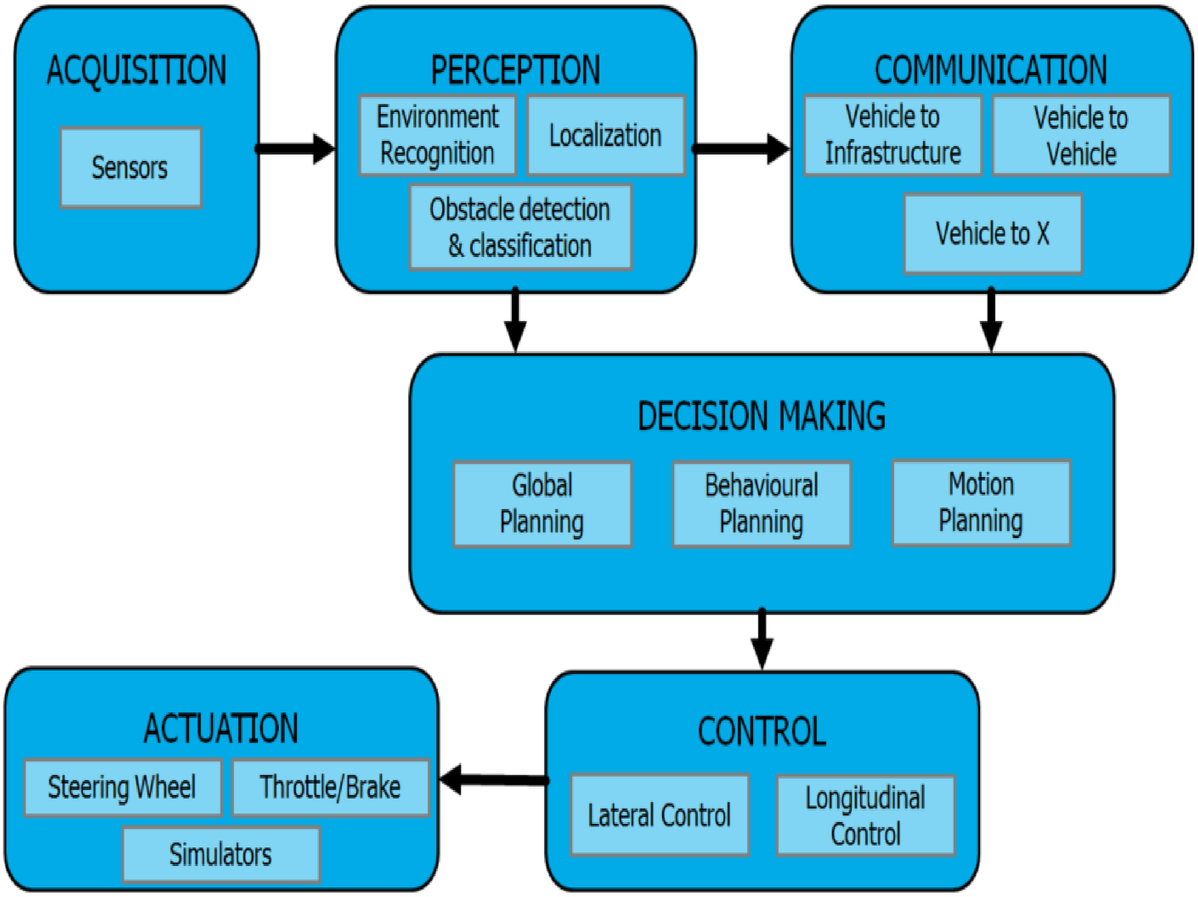


Figure 2: Architecture for Autnomous Vehicles

Autonomous driving vehicles have several architectures. Figure 2 depicts a general architecture for autonomous vehicles [5] that demonstrate the system architecture of an autonomous vehicle. The vehicle receives the information of the environment from sensors such as radars, GPS sensor, cameras, LIDAR etc. or by the fusion of multiple sensors [6]. Sensor fusion is the most significant technique of sensing technologies. Combination of various sensors results in determining more accurate information of environment since vehicles are subjected to dynamic environments.

Perception refers to the ability to acquire and extract the information of environment of an autonomous system. Perception stage deals with using data from sensors to detect road signs, objects, vehicle state, pedestrians, road detection and localization. Environmental recognition attributes to the contextual understanding of environment. Ego-vehicle localization is responsible for determining the position and orientation of the vehicle. The accuracy in determining the position of vehicle is important due to the fact that inaccuracy jeopardize decision making affecting safe and comfortable motion [6]. The data about the infrastructure and other vehicles is obtained from the communication module via communication mechanisms and protocols.

The data from the perception stage is used for decision making to plan vehicle behavior and motion. The planning stage is responsible for making precise decisions in order to bring the vehicle from one location to another. It consists of global planning, local planning and behavioral planning. The global planning module determines the most efficient route to the destination considering the current position of vehicle depending on several factors like distance, speed, time and traffic. Behavioral planning module performs the planning of appropriate driving behavior along the prescribed route ensuring stipulated traffic rules and interaction with other agents such as lane change, overtaking, parking and off-road driving [6]. The local planning module also called as motion planning performs the planning of vehicle trajectory along the specified route by the global planning module.

The generated trajectory is used by the control stage to develop actuator commands to follow the reference trajectory planned by the local planning module. The motion control module executes commands such as steering, brake and throttle to control the vehicle motion. The vehicle dynamics are considered for trajectory tracking and control to have efficient trajectory control and to maintain vehicle stability. Since the vehicle dynamics have significant effect on the motion of vehicle it is important to consider the lateral and longitudinal dynamics of the vehicle during the planning and control stages. In the control module the vehicle dynamics is controlled separately using lateral and longitudinal control systems. The lateral control provides actuation commands to change the lateral position and yaw of the vehicle. The longitudinal controller is responsible in controlling the vehicle’s longitudinal speed.

3.3 Control Problem for Autonomous Vehicle

The primitive control problem of autonomous vehicle is to determine the control signals for actuators based on mapping from the sensed data to ensure safe, reliable and comfortable drive in an autonomous vehicle. Therefore there must be an efficient approach to be followed in determining the decision making and control techniques. There are two main approaches addressing the autonomous vehicle control problem [7]. They are:

1. Learning Based Control
2. Planning Based Control

3.3.1 Learning Based Control

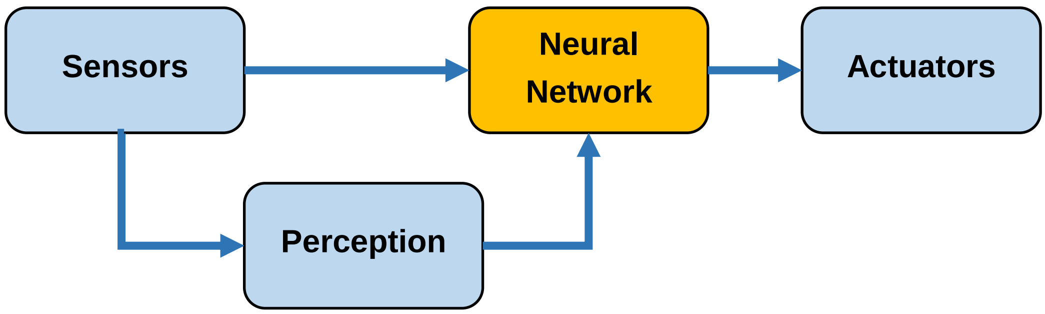


Figure 3: Illustration of Learning Based Control

Learning based control approach uses neural network for decision making based on input data from the sensors and perception module to generate outputs to control the driving behavior. The neural networks are trained with machine learning techniques to mimic the human driving behavior and hence it is also called as imitation learning [7]. Learning based control is divided into two categories based on the level of pre-processing of sensed data.

- End to end learning
- Perception based learning

3.3.1.1 End to end learning

This approach uses direct mapping of sensor data to generate the actuations. For example the data from the front camera and range sensor is used for the neural network to generate commands for steering wheel angle to control the orientation of the vehicle. The drawback of this technique is it's difficulty to analyse the factors influencing actuations [7].

3.3.1.2 Perception based learning

In this approach the sensor data is processed by a perception module for mapping and identifying the key indicators and relation between the environment and vehicle. The processed data from the perception module is used by a simple CNN network to map the indications into actuations. The processing steps are increased by the perception module but reduce the effort of CNN. The accuracy of perception is important for better performance

3.3.2 Planning Based Control

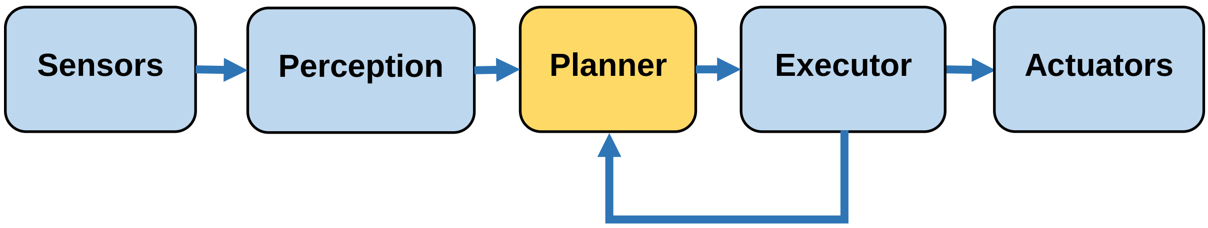


Figure 4: Illustration of Planning Based Control

In Planning based control the decision making to achieve a desired goal relies on more reasonable and feasible control approach. Similar to the learning based approach the sensor module is used to extract the data of environment. The perception module performs transformation of sensed data into cognitive world model that is used for planning decision. It performs environment detection such as objects, lanes, vehicles, obstacles, pedestrians as well as localization. The decision making in order to achieve a desired goal is performed by planner module. Depending upon the level of decision making the planner performs route planning, motion planning and behavioral planning. The executer is responsible for generating the desired control signals considering the dynamics of the vehicle. These control signals are used to produce actuations.

Planning based control performs planning based on vehicle motion, perception along with pre-stored map data and pre-programmed planning algorithms [7]. The performance of planning based control is mainly dependant on the accuracy of perception and planning algorithms. The uncertainties of the environment and the vehicle dynamics will also impact the performance of planning based control.

3.4 Planning for Autonomous Vehicles

As mentioned in the previous section there are two control approaches for autonomous vehicles. In this thesis the planning based control approach is considered. Planning is an increment approach that finds feasible sequence of state transitions for the vehicle to follow. This section focuses on the planning techniques for autonomous driving.

3.4.1 Architecture for Planning in Autonomous Vehicles

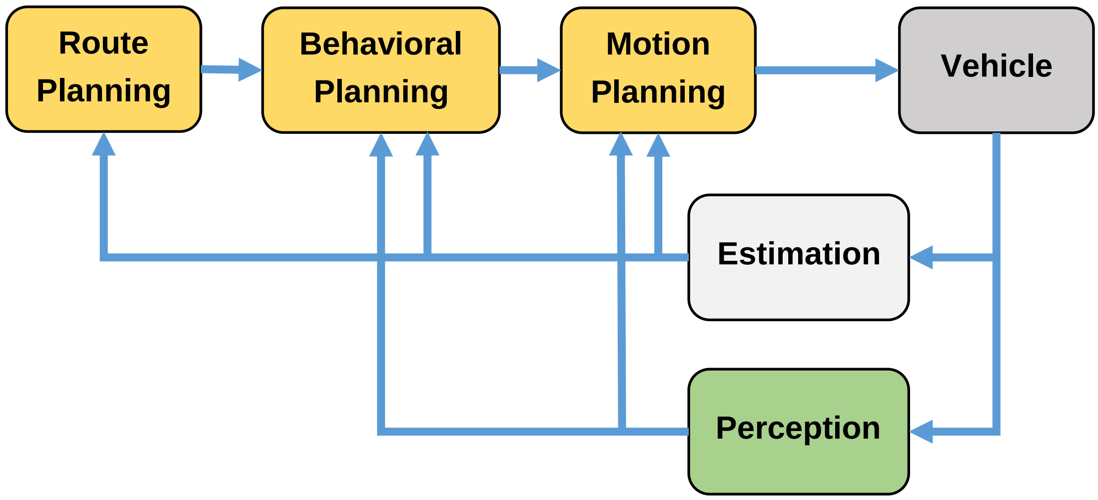


Figure 5: Architecture for Planning in Autonomous Vehicles

This section presents the architecture for planning in autonomous vehicles (Figure 5). The set of attributes specifying the condition of autonomous vehicle at a particular place and particular time instance is called 'state' of the vehicle [6]. This state of the vehicle is estimated by the estimator module. The state of a vehicle is represented as a vector. The state of the vehicle depicts position, orientation and velocities. A state space represents all possible states of the vehicle. Along with state of the vehicle, data from perception module is used for decision making. Route planning module makes planning depending on global objectives. The route planning module determines the most efficient route to the destination considering the current position of vehicle depending on several factors like distance, speed, time and traffic. Behavioral planning module performs the planning of appropriate driving behavior along the prescribed route ensuring stipulated traffic rules and interaction with other agents such as lane change, overtaking, parking and off-road driving. The motion planning module also called as local planning performs the planning of vehicle path, maneuver and trajectory along the specified route by the route planning module. The flowchart of planning modules is shown in Figure 6.

3.4.2 *Flowchart for Planning Modules for Autonomous Vehicles*

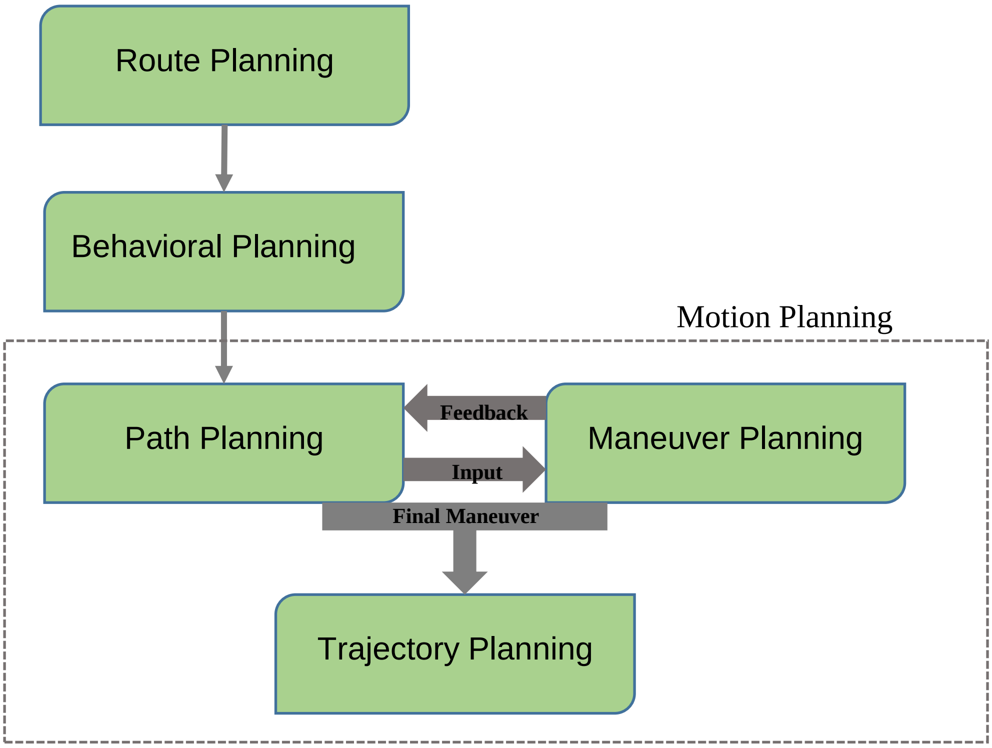


Figure 6: Flowchart of Planning Modules

Given the best route provided by the route planner the behavioral planner determines the type of behavior the vehicle must possess in order to reach the destination. Path is a geometric trace the vehicle should follow without any obstacles. Therefore path planning is finding feasible and safe geometrical paths along the route that adheres to motion constraints such as boundary conditions, traffic, lanes and road. The high level characterization of vehicle motion with respect to speed and position of the vehicle is called maneuver. Considering the path vehicle has to follow, maneuver planning performs high-level decision making to have correct and safe vehicle behavior. Trajectory is a sequence of states possessed by the vehicle with respect to time, position and velocity. Trajectory planning performs the real-time planning of vehicle transition from one state to another based on vehicle dynamics and constraints [8]. The steps associated with the planning modules are:

1. Finding the best possible route from one location to another
2. Finding the behavior of the vehicle to navigate the selected route according to traffic rules and driving conventions.
3. Finding geometric path the vehicle must follow considering the vehicle motion model
4. Finding feasible maneuver following desired path based on vehicle speed & position
5. Finding best trajectory to follow the geometric traces according to vehicle constraints

3.4.2.1 Route Planning

At the higher hierarchy of decision making systems of autonomous vehicles route planning module must select most efficient route through the road network to the requested destination based on the current position of the vehicle. The route planning module generally solves the problem of graph search by constructing a graph from the road map data to find the fastest and safest route from the current location to the destination location. Several algorithms are available out of which A* and Djikstra algorithms are the most widely used algorithms for autonomous driving applications [9]. These algorithms are mainly used with AI (Artificial intelligence) due to its accuracy and fast re-routing of the planned route when there are abnormalities in the route such as obstacles, road construction, tree branch on the road etc.

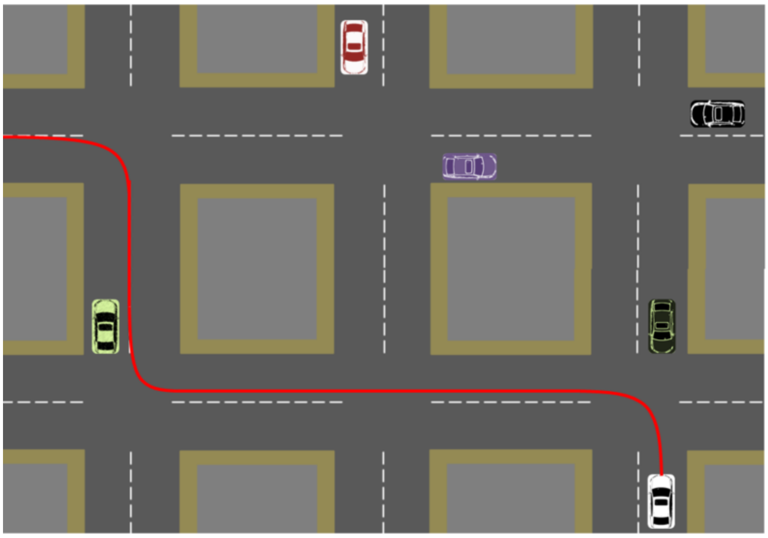


Figure 7: Route Planning

3.4.2.2 Behavioral Planning

After the route planning the autonomous vehicle must be able to navigate along the prescribed route by making interactions with other vehicles and the environment following the driving conventions and road regulations. The behavioral module is responsible for creating a feasible driving behavior at any time instance considering road conditions, traffic, signals etc. Autonomous driving with humanlike driving behavior is performed with interactions and interdependencies with the surroundings. This enables the vehicle to handle the uncertain environmental conditions. The driving behaviors are modeled as finite state machine with transitions subjected to perceived driving context and the planned route [3] as shown in Figure 8. However in real world urban driving environments the uncertainty with respect to the intentions of traffic participants and environment has to be considered for a safe and reliable behavioral planning.

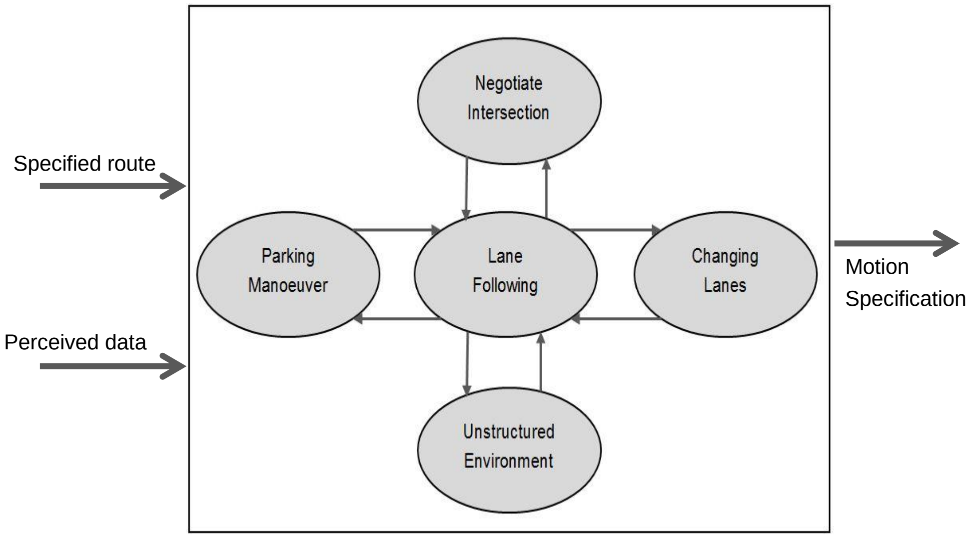


Figure 8: Behavioral Planning State diagram

3.4.2.3 Motion Planning

After the decision making of the driving behavior by the behavior module, motion planning module has to translate the planned behavior into a path, maneuver or trajectory than can be tracked by the controller to generate actuation commands. The resulting motion of the vehicle must avoid collisions, obstacles and provide safety and comfort to the occupants. The task of planning such a path or trajectory is performed by the motion planning system. The motion planning techniques are

- Path Planning: Path is a geometric trace the vehicle should follow without any obstacles. Path planning is responsible for determining safe and comfortable geometrical paths along the route dealing with motion constraints such as traffic, lanes, pedestrians, road etc.

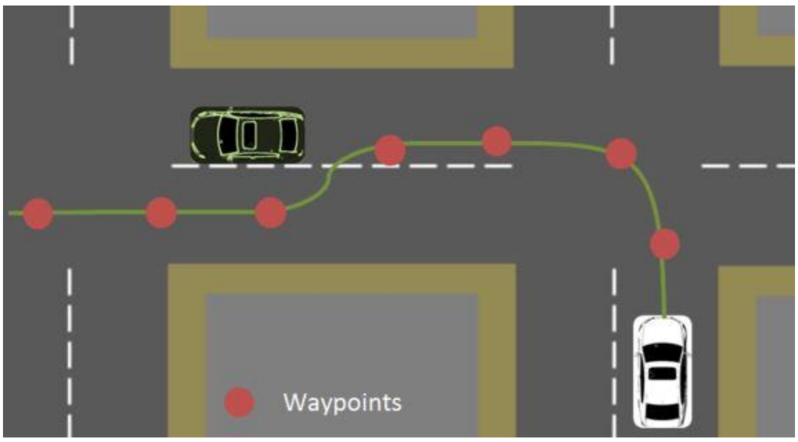


Figure 9: Path Planning

- Maneuver Planning: The high level characterization of vehicle motion with respect to speed, position and orientation of the vehicle is called maneuver. Considering the planned path the vehicle has to follow, maneuver planning performs high-level decision making to have correct and safe vehicle behavior.

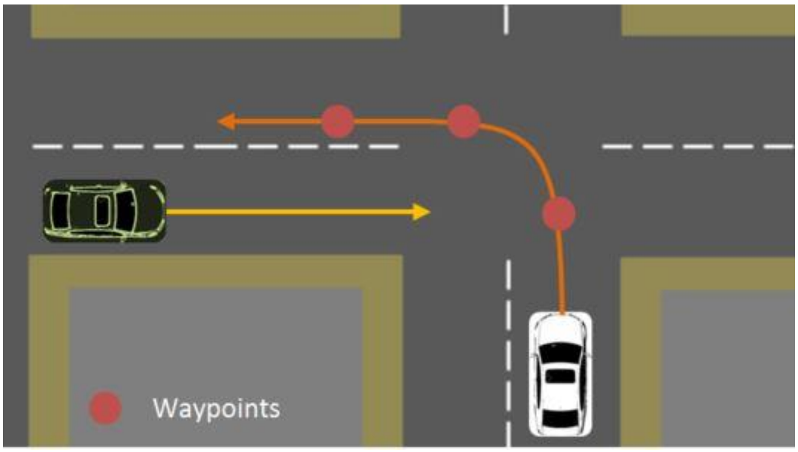


Figure 10: Maneuver Planning

- Trajectory Planning: Trajectory is a sequence of states possessed by the vehicle with respect to time, position, orientation and velocity. Trajectory planning performs the real-time planning of vehicle transition from one state to another based on vehicle dynamics and constraints.

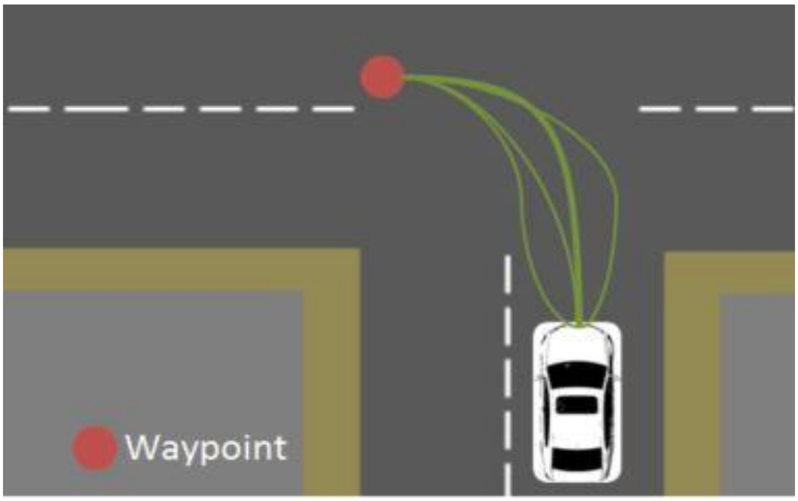


Figure 11 :Trajectory planning

3.5 Motion Control in Autonomous Driving

The execution capability of autonomous systems is generally referred as motion control. Controlling is the process of converting decisions into actions. The main purpose of controlling is to execute the planned decisions into desired vehicle motion by providing the required inputs to hardware systems. Controlling maps the real world interactions in terms of torques, forces and brakes, whereas the planning algorithms are mostly concerned with position, orientation and velocities of the autonomous vehicle with respect to environment. System behavior can be determined by the measurements and computations of the control system, which enables the control system to reject disturbances by altering the dynamics of the system.

3.6 Control Methodologies

3.6.1 Feedback Control

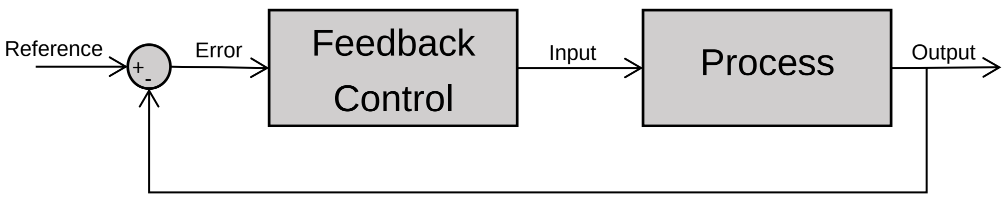


Figure 12: Feedback Control

Feedback control is the most commonly used control methodology for many applications. Feedback control measures the disturbances between the output and reference signals to compensate the deviations from desired system behavior. Feedback control reduces the effect of unwanted disturbances, parameter changes and modelling errors [7]. It is capable of modifying the system behavior and the effects of noise. The drawback of feedback controller is the delayed response to errors. Feedback control responds to errors only when they occur. The most common feedback controllers used in trajectory control applications is PID (Proportional-Integral-Derivative) controller. Also other variants like PI (Proportional-Integral) and PIP (Proportional-Integral-Plus) controllers are used for trajectory control for autonomous vehicles. PIP Controller has higher performance than PID because of more feedbacks. The major drawback of feedback control is its inability to handle the non-linear systems. Tuning of PID controllers is complex for MIMO systems.

3.6.2 *Optimal Control*

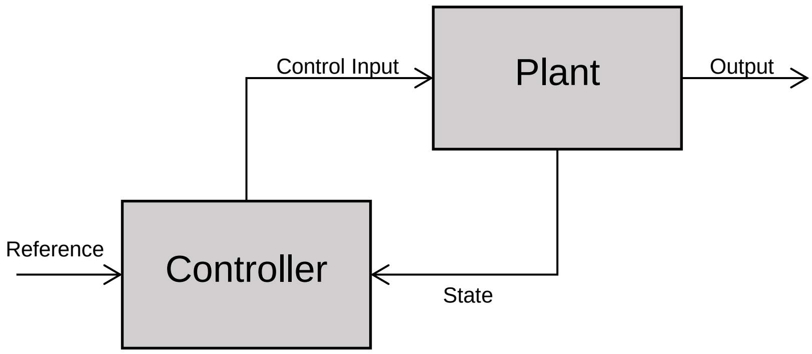


Figure 13: Optimal Control

The main objective of Optimal Control is to determine the optimal control input that drives the system from initial state to reference state. Optimal Control systems are MIMO control systems. The control problem includes the state and the control variables. The optimal control systems provides control inputs considering the model of the plant and calculates performance index based on the states of the plant. A plant is described as linear or non-linear mathematical equations. The current state of the plant is estimated using a state estimator and optimal control inputs are generated to make the plant operation to reach the desired state. The optimal control techniques used for trajectory control are LQR (Linear Quadratic Regulator) and LQG (Linear Quadratic Gaussian). For trajectory control applications, optimal control is not successful because of the following drawbacks.

- Inability to handle plant nonlinearities
- Lack of robustness
- Inability to handle non-linear cost function
- Constraint handling

Trajectory control of autonomous vehicle depends on the dynamics of the vehicle. Since the dynamics of the vehicle is non-linear, optimal control systems are not robust for autonomous vehicle trajectory control.

3.6.3 Predictive Control

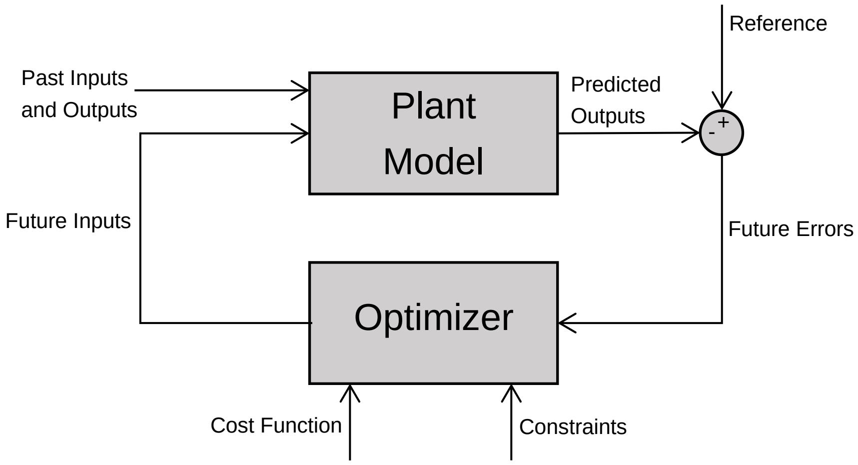


Figure 14: Basic Structure of Predictive Control

Predictive control is a control approach which uses the model of the plant for control execution. It predicts the change in model dependant variables. Based on the current state of the plant, it predicts the future states of the plant relying on the model behavior of the plant. Depending on predicted and curret state it generates control commands to have robust plant operation. The prediciton of states is performed by the optimizer depending on the constraint of the system and cost function. Predictive control simulates inputs at each time instant and selects the best resulting control commands in order to have robust system operation with respect to the reference [7]. It constantly reevaluaties the future states and predicts inputs over future horizon. Predicitve control is the most feasible control technique for trajectory control over other control techniques because of the following advantages

- Better performance for nonlinear MIMO systems
- Can handle constraints
- Robust
- Can handle non-linear cost functions

3.7 Control Framework for Predictive Control in Autonomous Vehicles

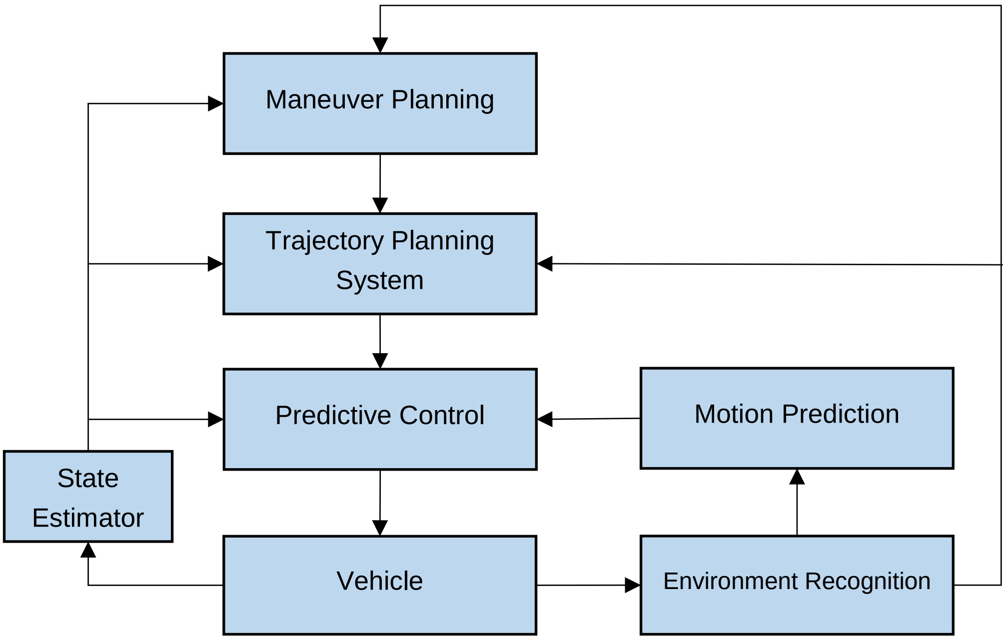


Figure 15: Control Framework

Maneuvers are assigned considering the environment and vehicle state. The trajectory planning system is responsible for processing the information from the maneuver planning module and environment recognition to plan appropriate trajectory to execute the defined maneuvers. The references, constraints and parameters needed for the controller formulation are taken into account for planning trajectories. The planned trajectory is used as a reference for the controller. The predictive controller produces actuations depending on the current state of the car and the predicted motion. The predictive controller is responsible for controlling the lateral and longitudinal motion of the vehicle. Based on the predicted motion of the autonomous vehicle by the motion prediction module the controller generates actuation commands at each time step. The prediction is dependent on the motion model of the vehicle. The motion models are mathematical differential equations representing the dynamics of the vehicle. More detailed modelling of the vehicle is necessary for accurate motion predictions. Predictive controllers are capable of handling both linear and non-linear models. The current position and orientation of the vehicle is used to depict the state of the vehicle to provide the state variables for the controller. At each prediction interval the predicted motion is selected according to reference trajectory.

4 Model Predictive Control

Model predictive control was developed in the late 1960s and was mainly used in chemical and oil refinery industries [10]. Initially MPC was used for control applications for SISO (Single Input Single Output) systems that require low computations. The development of hardware that can handle complex computations enabled MPC to spread across other sectors [10]. One of the major aspects of adopting MPC was its design formulation for MIMO systems capable of handling multiple variables by controlling the performance parameters of the system. MIMO systems require fast optimizations to make the systems perform optimally. MPC performs optimization using specific optimization algorithms at each control interval to produce optimal control. With the rapid developments of computers and embedded devices with high computational power enabled the usage of MPC for systems with multiples states and variables [11]. MPC has the ability to handle constraints to limit the physical behavior of the systems. This is the main advantage of using MPC for dynamic systems. MPC tries to predict the future states of the plant using the current state and model of the plant. The optimizations are based on performance index or objective function. The MPC algorithm tries to generate control moves to minimize the objective function. The main advantages of MPC are

- Explicit Constraint handling capability
- Easy tuning of MIMO systems
- Future predictions of system behavior
- Ability to handle linear and non-linear dynamic models
- Robustness
- Online and offline optimization capability

4.1 MPC Structure

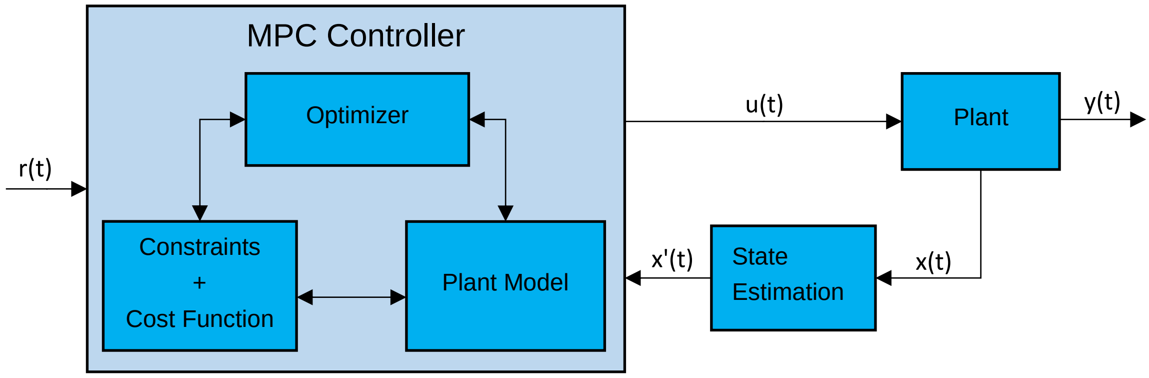


Figure 16: MPC Structure

The structure of MPC is given in Figure 16. Model predictive controller consists of three functional blocks. The optimizer uses optimization algorithms to find optimal control input u(t) trying to minimize the cost J without exceeding the constraints. The cost function J or control objective is a scalar criterion calculating the difference between the reference r(t) for controller and predicted future outputs of the plant [10]. Cost function is a measure of the plant behavior for next time steps which is called as prediction horizon (Refer Figure 17). The cost function is minimized for future control outputs estimating the future states of the plant. The prediction of future states x'(t) of the plant is done by state estimator using the current state x(t) of the plant. The plant model is a set of mathematical equations describing the input and output behavior of the plant. The plant models can be linear or non-linear. For non-linear plant models the model has to be linearized and has to be represented in state space form to use it for MPC. Constraints are the limits for the controller to limit the controller outputs without exceeding the boundary conditions of the plant. MPC is a control technique that uses model of the plant to predict the future behavior until prediction horizon by generating control inputs satisfying the plant constraints and minimizing objective function. The computations that is necessary for predictions is performed by the optimizer. The optimization is carried out for each sample time. Each computation by the optimizer results with optimal control outputs until the prediction horizon. The prediction horizon shifts for every time step and hence it is also termed as receding horizon [11]. The main objective of the optimizer is to compute new control outputs u(t) at every time step, that will be used by the system to perform the necessary control operation.

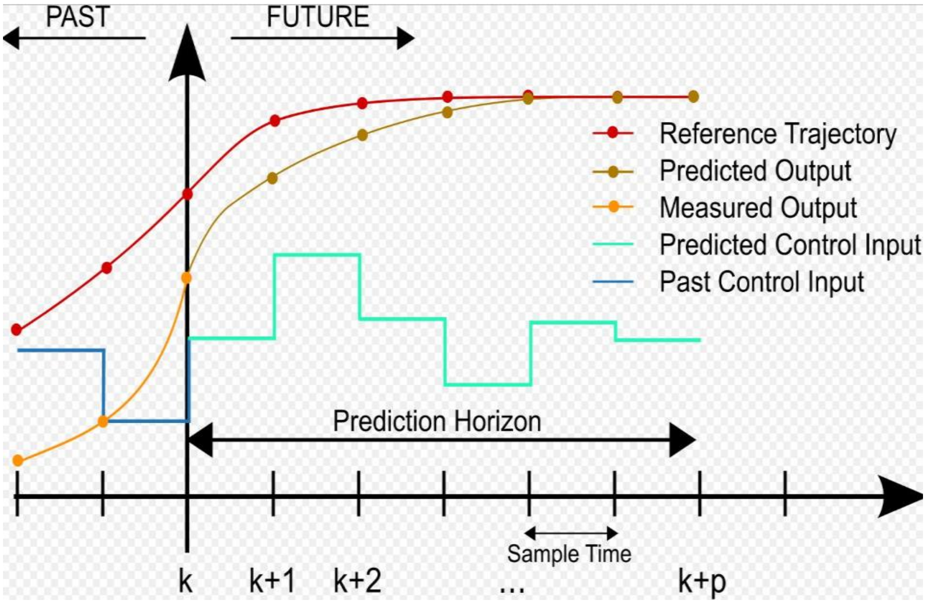


Figure 17: MPC Illustration [11]

The steps followed by MPC control algorithm are:

1. At time k, use the plant model to predict N future states of the plant y(k+p), p=1…N
2. Predict the future inputs u(k+j), j=0…N in order to reach the predicted states.
3. Define a cost function based on output y and input u
4. Optimize it with respect to constraints and future inputs u(k+j), j=0…N
5. Apply the first step of previously predicted future inputs
6. Repeat steps 1-5 for next sampling instance

4.1.1 Plant Model

The plant model is used to capture the dynamics of the plant in order to predict the future. More accurate representation of the dynamics of physical systems gives good future predictions. For MPC the plant model has to be a linear representation in state space form:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du,$$

Where x is state variable, u is control signal, y is plant output and A, B, C, D are state space matrices. Discretization of the state space model gives

$$x(k+1|k) = Ax(k) + Bu(k|k)$$

where |k indicates the estimation at kth time instance

4.1.1.1 Prediction Using State Space Model

Considering the current state values of state space model the predictions of future states are performed for time step H (Prediction horizon). From linearization of state space model we know that

$$x(k+1|k) = Ax(k) + Bu(k|k)$$

For state prediction of one time step ahead with time step H can be written as

$$x(k+1) = Ax(k) + Bu(k)$$
$$x(k+2) = Ax(k+1) + Bu(k+1)$$

.

.

$$x(k+H) = Ax(k+H-1) + Bu(k+H-1)$$

For state prediction of n time steps ahead with time step H, the expression becomes

$$x(k+1) = Ax(k) + Bu(k)$$

$$x(k+2) = A^2x(k) + ABu(k) + Bu(k+1)$$

.

.

$$x(k+H) = A^Hx(k) + A^{H-1}Bu(k) + \cdots + ABu(k+H-2) + Bu(k+H-1)$$

Expressed in matrix form the equation becomes,

$$\begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+H) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^H \end{bmatrix} x(k) + \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H-1}B & A^{H-2}B & \dots & B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+H-1) \end{bmatrix}$$

4.1.2 Cost Function

The cost function J or control objective is a mathematical optimization method. During the optimization of a control problem the optimizer tries to minimize the cost function to find the best solution for the control problem. It is dependent on several variables and can be subjected to the constraints of the plant. The objective of a controller is to minimize or eliminate the errors between output y and reference by performing appropriate selection of inputs u. For MPC the most generally used cost function is in quadratic equation form where P and Q are the weights for penalizing the cost function.

$$J = \sum_{i=1}^{H} \{P(x_i - x_{ref})^2 + Q(u_i - u_{ref})^2\}$$

4.1.3 Constraints

Constraints are the limits for controller output and state variables to have feasible controller behavior avoiding abnormality. Constraints are physical limitations of the plant to operate in a desirable manner. Constraints are classified as

1. Soft Constraints: These are the constraints where exceeding the limits have no significant impact on the controller performance and plant behavior.
2. Hard Constraints: There are the constraints where exceeding the limits have strong impact on the controller creating unfeasible plant behavior.

The types of constraints are

- Input amplitude constraints: $u_{min} \leq u_k \leq u_{max}$
- Input rate constraints: $\Delta u_{min} \leq \Delta u_k \leq \Delta u_{max}$
- Output constraints: $y_{min} \leq u_k \leq y_{max}$

4.1.4 Optimizer

Using the cost function formulation, constraints and the state space model of the plant the optimizer performs computations to predict the future control signal that are responsible for the future states. At each time step the computations are performed and it is important to ensure that the predictions are made as accurate as possible. Optimization is of two standard forms:

- Quadratic Programming (QP): QP optimization is used when plant model dynamics are linear with linear constraints and quadratic cost function.
- Linear Programming (LP): LP optimization is used for linear and non-linear plant models with linear cost function and linear constraints.

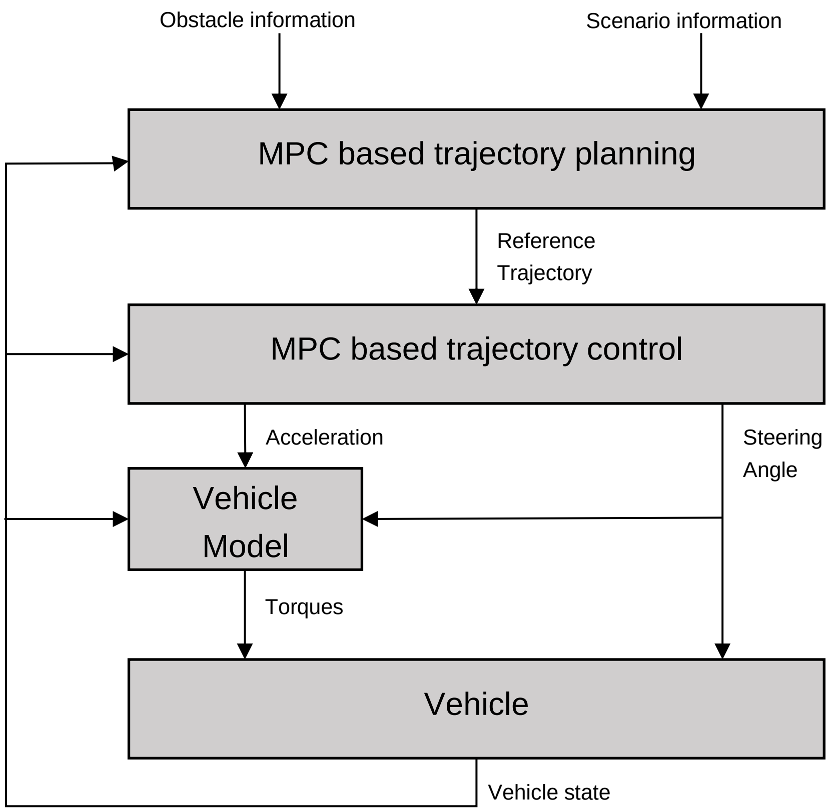
4.1.5 State Estimator

The MPC controls the plant model by optimizing the plant input to avoid the deviation from the current state of the plant and the predicted state. A robust control of the plant can be achieved if the predicted future states are accurate. Modelling errors and unknown disturbances can create inaccuracy in predictions. It is necessary to update the plant model at each time step for good predictions of the future plant behavior. The updating of the plant model is executed by state estimator. Kalman filters are typically used for state estimations for MPC [12].

4.2 MPC for Autonomous Vehicles

Autonomous vehicles depend on control systems for reliable motion control. MPC is used in both motion planning and motion control applications for autonomous driving. MPC is used for motion planning to compute collision and obstacle avoiding trajectories [13]. For motion control applications MPC is used to follow the reference trajectories by predicting future trajectories. This section presents the applications of MPC for trajectory Planning and trajectory control.

4.2.1 MPC Control framework for Trajectory planning and control



Hierarchical framework design approach is used for the control framework for autonomous driving to separate the planning and control sections. MPC uses a simplified vehicle models to generate the feasible trajectories that can be used as a reference for controller. Another MPC is used for trajectory control at the control level. This approach is used where MPC predictions are desired to use for both planning and control of the vehicle. For systems with separate planning algorithms, MPC can only be used for controlling the vehicle trajectory [14]. The current vehicle state is used for both planning and control. The MPC based motion planning considers the reference signal, for example constant speed profile for trajectory generation. The MPC based motion planner uses the predictive approach for future trajectory predictions based on the objective function to generate the reference trajectory for the controller. This reference trajectory is used by the controller to compute optimal control inputs such as acceleration and braking to control the motion of vehicle. The acceleration inputs are used to create braking torques that has to be exerted on the vehicle wheels. The steering angle input is used by the vehicle directly. The advantage of this approach is it enables to formulate unique constraints and objective functions separately for planning and control. Less complex vehicle models can be used for trajectory planning reducing the computational effort irrespective of the vehicle model for the controlling part.

5 Vehicle Models

The most important aspect of MPC is the model of the vehicle that is used to predict the future states of the vehicle. The first step is to represent the model in mathematical form and then use it for simulation to simulate the behavior of vehicle and evaluate the performance of the controller for the developed vehicle model. The designing of the controller is mainly dependent on the quality of vehicle models. Vehicle models must describe the real behavior of the vehicle as close as possible. Higher the complexity in modelling of vehicle models, higher the accuracy in predictions. A good vehicle model represents the dynamics of the vehicle closer to reality and reduces the complexity in using it for simulation. It is important to consider the lateral and longitudinal dynamics of the vehicle for modelling to represent the vehicle closer to reality. During high dynamic maneuvers the lateral and longitudinal dynamics have significant effect related to vehicle motion. It becomes more apparent to consider the dynamics of maneuvers with high speed profiles [15]. For trajectory control it is very crucial to include the lateral and longitudinal dynamics satisfying the constraints and other limitations of the vehicle. There are several vehicle models depicting different driving behavior of vehicle. The vehicle model complexity varies depending on the purpose of the vehicle model and availability of information related to vehicle state. There are several modelling approaches considering motor dynamics, brake dynamics and motion dynamics. Since trajectory control is mainly concerned with the motion of vehicle, in this thesis the modelling is done with only the vehicle motion dynamics. The most popular model to describe the vehicle dynamics of the vehicle is a single track model, also termed as bicycle model [16]. A bicycle model is a simplified model of a vehicle four wheel model, where the front and rear tires are assumed to have equal behavior. The front and rear tires are merged to represent a single front and single rear tire as shown in Figure 18. The vehicle is assumed to be a rigid body with vehicle mass acting on the Centre of Gravity (CoG).

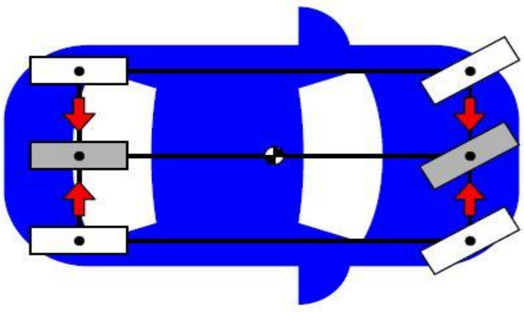


Figure 18: Four wheel model to Bicycle model

5.1 Co-ordinate Systems

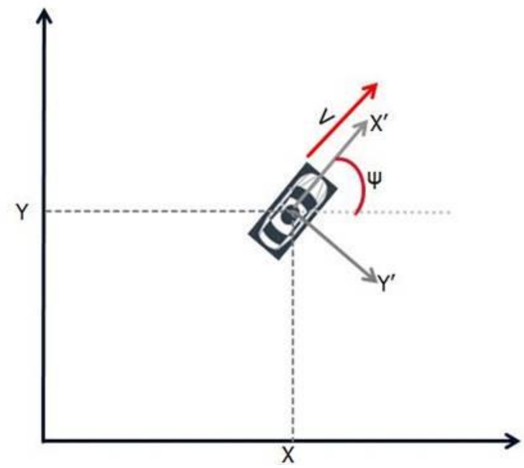


Figure 19: Illustration of Coordinate System

Figure 19 shows the illustration of coordinate systems. X and Y are the position of the car in global frame. X' and Y' represents the position with respect the vehicle frame. Ψ is the orientation of the vehicle and v is the velocity of the vehicle in a fixed vehicle coordinate systems.

State of the vehicle in global coordinate can be represented as

$$\xi=[X,Y,\psi,v]$$

State of the vehicle in vehicle coordinate can be represented as

$$\xi=[x',y',\psi,v]$$

5.2 Linear Models

Linear models are used for linear systems where the relationship between the inputs and outputs of the system are linear. Let us assume a system with input $x_1(t)$ produces output $y_1(t)$, similarly $x_2(t)$ gives output $y_2(t)$. The summed response of inputs $x_1(t) + x_2(t)$ produces the output $y_1(t) + y_2(t)$. Applying the input $x_1(t)$ for a system with time t gives the output of $y_1(t)$, similarly the input applied for k time steps x(t-k) produces identical output with time delay of k seconds y(t-k). The general form of representing linear systems is

$$x' = Ax + Bu$$

Where x represents the states of the model and u is the control input. Liner models are not preferred for vehicle modelling because motion control systems are nonlinear and linear models do not describe the behavior accurately.

5.2.1 Point Mass Model

Point mass model is a linear model used to represent the vehicle behavior. The vehicle is a single point mass ignoring all the dynamics of the vehicle. For point mass models the vehicle is always considered to be moving straight. The global coordinates is used for modeling vehicle motion.

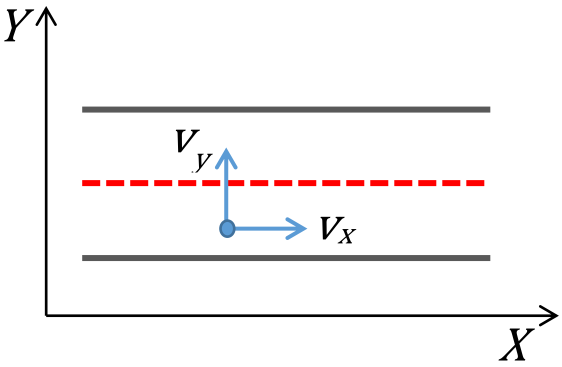


Figure 20: Point Mass Model

The motion equations for point mass model is represented as

$$x' = v_x$$
$$y' = v_y$$
$${v_x}' = a_x$$
$${v_y}' = a_y$$

Where,

x : longitudinal position of the vehicle in x direction

y : lateral position of the vehicle in y direction

v_x & v_y : lateral and longitudinal velocity

a_x & a_y : lateral and longitudinal acceleration

5.3 Nonlinear Models

Nonlinear models are used for dynamic systems where the system states are nonlinear. The state variables of nonlinear models are not linear independent components. Nonlinear models are usually represented by differential or partial differential equations. The vehicle motion dynamics are nonlinear and hence nonlinear models represent the vehicle motion behavior more realistically. In general nonlinear systems are described by the differential equation,

$$x' = f(x, u, t)$$

The dynamics of nonlinear systems are represented in state space form for linearization. State space representation is the depiction of model dynamics as a set of variables. These set of variables are a set of differential equations called as state variables that completely describe the response of the model for any set of inputs.

5.3.1 State equations

For nonlinear systems with multiples states and multiple inputs the nonlinear differentials equations are represented as state equations. Consider a nonlinear system with state variables $x_1(t), x_2(t),....., x_n(t)$ and inputs of $u_1(t), u_2(t),....., u_n(t)$. The state equations are defined as:

$$x'_1 = f_1(x, u, t)$$
$$x'_2 = f_2(x, u, t)$$
$$\vdots$$
$$x'_n = f_n(x, u, t)$$

Where $f_i(x,u,t)$ and $x'_i=dx_i/dt$, $i=1,2....n$ is a nonlinear function representing states, inputs and time. $\dot{x}_i$ The state equation in vector form is

$$x' = f_1(x, u, t)$$
$$y = h(x, u, t)$$

The linearized state space representation is of the form

$$x' = Ax + Bu$$
$$y = Cx$$

Where, states $x=\{x_1(t), x_2(t),....., x_n(t)\}$, inputs $u=\{u_1(t), u_2(t),....., u_q(t)\}$, output vector $y=\{y_1(t), y_2(t),....., y_m(t)\}$ and A,B *and* C are filter matrices of size $m * n$.

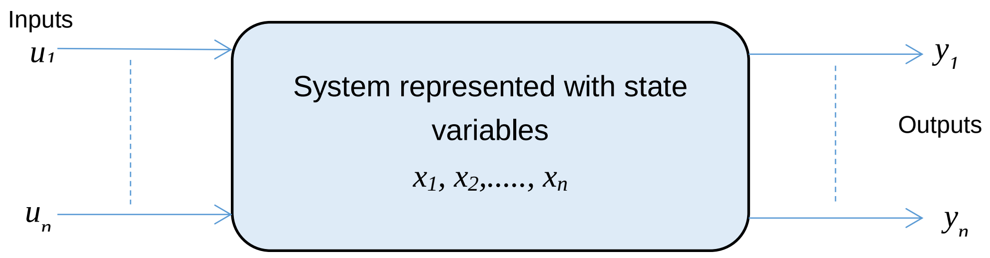


Figure 21: State Space Model

5.3.2 *Kinematic Bicycle Model*

The kinematic model is based on kinematic relationships of the vehicle. The kinematic model do not consider vehicle dynamics like mass, inertia, forces, roll, pitch and yaw of a vehicle. It assumes that the vehicle comply perfectly to the road. It performs well for low speed profiles. Kinematic models represent better realistic behavior than point mass models because of the consideration of side slip angle during modelling. Using bicycle model for modeling kinematic vehicle behavior has the advantage of reducing the states of the vehicle than a four wheel vehicle model. Kinematic models are modelled based on the geometry of the vehicle. Kinematic bicycle models are generally used for trajectory planning and control of the vehicle [1].

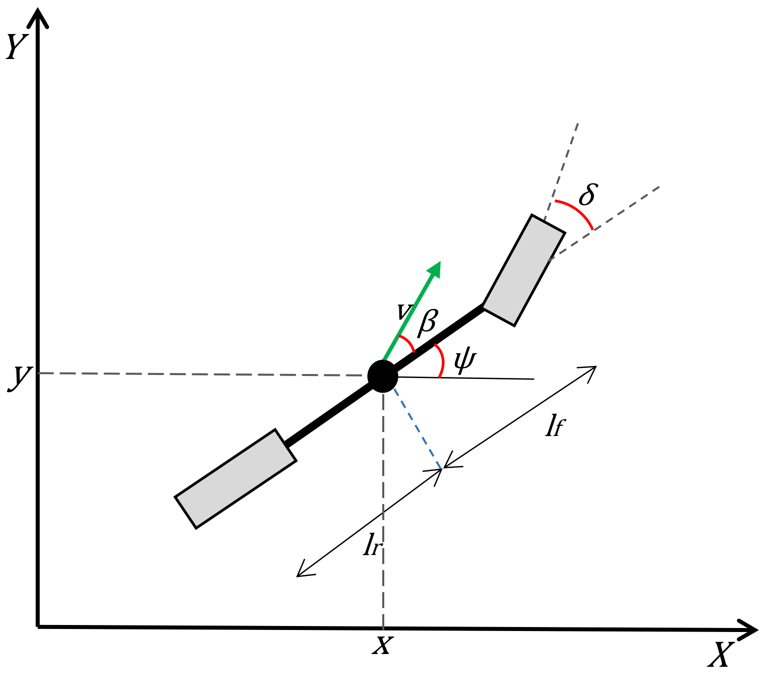


Figure 22: Kinematic Bicycle Model

The nonlinear equations describing the kinematics of the vehicle in global frame is given by

$$x' = vcos(\psi + \beta)$$

$$y' = vsin(\psi + \beta)$$

$$\psi' = \frac{v}{l_r}\sin(\beta)$$

$$v' = a$$

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

For constant speed profiles the side slip angle of the vehicle is neglected and the simplified model equations are given by

$$x' = vcos(\psi)$$
$$y' = vsin(\psi)$$
$$\psi' = \frac{v}{l_r + l_f}\tan(\delta)$$
$$v' = a$$

The state of the model are ξ= [$x\ y\ \theta\ v$] and control inputs u =[$\delta\ a$]

Where,

x and y: coordinates of center of mass in global frame (m)

ψ : Vehicle heading angle (rad)

v : Vehicle Speed (m/s)

a : Acceleration (m/s^2)

β : Vehicle Side Slip angle

δ : Steering angle (rad)

l_f & l_f : Distance between center of mass to front and rear axle (m)

5.3.3 Dynamic Bicycle Model

The dynamic vehicle model is based on vehicle dynamics including yaw, pitch, roll, mass, forces etc. It also includes tire model to take account of the tire forces. The dynamic models are modelled based on Newton's second low of motion including vehicle forces. Dynamic models replicate the vehicle behavior close to reality than kinematic and point mass models [17]. Dynamic models are highly nonlinear due to the fact of including the tire forces. Tire models like nonlinear Pacejka model is used for the modelling of dynamic model. However linear dynamic models are used to reduce the modelling complexity using the linear tire models where the tire behavior is in the linear region. It constraints the tire slip angles to keep the tire behavior in linear region making the vehicle modelling valid and reducing the complexity for linearization [16]. The tire dynamics increases the vehicle states and has less effect on the model accuracy. For trajectory control the vehicle dynamics with longitudinal, lateral motions at vehicle center of mass with lateral and longitudinal tire forces are considered for modelling to reduce the linearization complexity for MPC controllers.

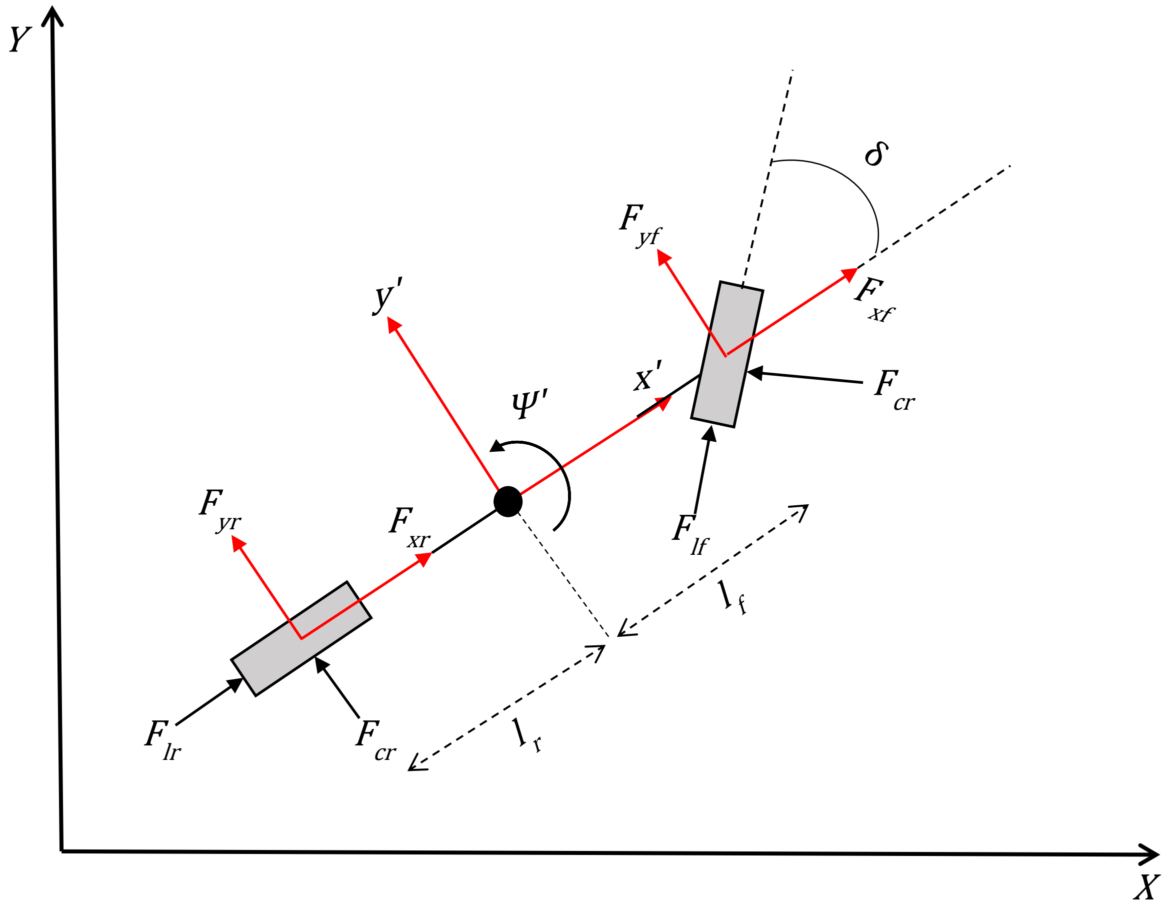


Figure 23: Dynamic Bicycle Model

The nonlinear equations describing the dynamics of the vehicle is derived by using Newton‘s second law of motion along Y-axis.

$$\Sigma(forces\ acting\ along\ y - axis) = ma_y$$

$$2F_{yf} + 2F_{yr} = ma_y \tag{1}$$

Where, m is the mass of the vehicle, a_y is the vehicle acceleration along Y-axis. The lateral acceleration is due to centripetal acceleration and acceleration due to vehicle motion. Hence,

$$a_y = y'' + v_x\psi' \tag{2}$$

Substitute (2) in (1)

The sideslip angle of the vehicle is given by $\beta = \frac{y\prime}{v_x}$ and therefore (3) becomes

$$2F_{yf} + 2F_{yr} = mv_x(\beta' + \psi') \tag{4}$$

Considering the moment M and inertia I acting on the vehicle, the total moment at the center of mass is given by

$$\Sigma M = I\psi'' - 2F_{yf}l_f + 2F_{yr}l_r = 0$$
$$I\psi'' = 2F_{yf}l_f - 2F_{yr}l_r$$

Similarly applying Newton's law of motion on X-axis we get

$$mx'' = my'\psi' + 2F_{xf} + 2F_{xr}$$
$$my'' = -mx'\psi' + 2F_{yf} + 2F_{yr}$$

Where,

x & y : Position of vehicle in vehicle coordinates (m)

ψ: Vehicle yaw angle in global frame (rad)

m: total mass of vehicle (kg)

I: Vehicle inertia (kg m^2)

F_{xf} & F_{xr} : longitudinal tire forces of front and rear tires along x-axis (N)

F_{yf} & F_{yr} : lateral tire forces of front and rear tire along y-axis (N)

l_f & l_f : Distance between center of mass to front and rear axle (m)

From kinematic modelling (5.3.2) we know that the equations for vehicle motion in global frame is given by

$$X' = vcos(\psi + \beta)$$
$$Y' = vsin(\psi + \beta)$$

Expanding the equations we get

$$X' = vcos(\beta) * cos(\psi) - vsin(\beta) * \sin(\psi)$$
$$Y' = vcos(\beta) * sin(\psi) + vsin(\beta) * \cos(\psi)$$

Writing vehicle's velocity in terms of longitudinal & lateral velocity i.e $x' = vcos(\beta)$ and $y' = vsin(\beta)$ the equations for motion becomes

$$X' = x' \cos(\psi) - y'\sin(\psi)$$
$$Y' = x' \cos(\psi) - y'\sin(\psi)$$

The forces acting on the center of mass is given by

$$F_x = F_{xf} + F_{xr}$$
$$F_y = F_{yf} + F_{yr}$$

The x and y components of tires forces are given by

$$F_{xf} = F_{lf}\cos(\delta) - F_{cf} sin(\delta)$$
$$F_{yf} = F_{lf}\sin(\delta) - F_{cf} cos(\delta)$$

According to nonlinear pacejka tire model [18]

$$F_l = f(\alpha, \mu, s, F_z)$$
$$F_c = f(\alpha, \mu, s, F_z)$$

Where, α is tire slip angle, it is the angle between the vehicle wheel and direction of velocity as shown in Figure 24, μ is road friction coefficient, s is the slip ratio and F_z is the vertical load on the wheels. Assuming the slip ratio, friction coefficient and load to be zero then longitudinal tire forces can be calculated as

$$F_{lf} = \frac{T_{bf}}{r}$$

$$F_{lr} = \frac{T_{br}}{r}$$

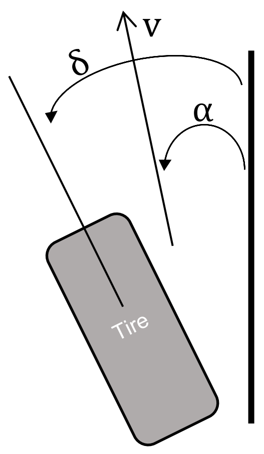


Figure 24: Tire Model

Where, T_{bf} and T_{br} are brake torques and r is wheel radius. According to linear pacejka model for small slip angles the lateral tires forces is proportional to and cornering stiffness of the wheel [18]. Hence the lateral tire forces are

$$F_{cf} = \alpha_f C_f$$
$$F_{cr} = \alpha_r C_r$$

Where, α_f and α_r are front and rear tire slip angle, C_f and C_r are front and rear tire cornering stiffness

The state of the vehicle for dynamic vehicle model is

$$\xi = [X, Y, \psi, v]$$

Where, X and Y is the position of vehicle in global coordinates, ψ is the yaw angle and v is velocity of the vehicle

The control inputs for the dynamic model are

$$u = [\delta, T_{bf}, T_{br}]$$

Where, δ is steering angle of vehicle and T_{bf} and T_{br} are front and rear brake torques.

6 MPC for Trajectory Control during High Dynamic Maneuvers

6.1 High Dynamic Maneuvers

Maneuver is the characterization of the vehicle motion with respect to speed, position and orientation. Dynamic maneuvers are the maneuvers where the motion of vehicle has impact on the vehicle dynamics creating discomfort and also affecting the occupant safety. Autonomous cars are subjected to dynamic environments where the car has to navigate during dynamic scenarios without disrupting safety and traffic regulations. High dynamic maneuvers are such maneuvers where the lateral and longitudinal vehicle dynamics such as speed, yaw, pitch and roll have significant effect on vehicle motion [19]. The maneuvers with high vehicle speed and extreme cornering scenarios producing rapidly varying steering control of the car are high dynamic maneuvers. During these maneuvers the resulting forces acting on the car produces jerk affecting passenger comfort and leads to motion sickness [20]. The acceleration, steering and braking for such maneuvers generates lateral and longitudinal forces. Advanced suspension and chassis functions are used to mitigate these forces. In this thesis the focus is mainly on creating high dynamic maneuvers for simulation and test the performance of MPC for high dynamic maneuvers. The high dynamic maneuvers considered are Slalom maneuver, Lane change maneuver and Steady circle maneuver.

- Slalom Maneuver: The slalom maneuver consists of traffic cones line up, seperated by equal distance. The vehicle is driven between the cones with a constant speed.
- Lane Change Maneuver: The lane change maneuver consists of an entry lane, side lane and exit lane, the vehicle is driven through the cones of entry lane, then changing the lane into the side lane of specific distance and then going into the exit lane.
- Steady Circle Maneuver: The vehicle is driven with a constant speed in a steady circle of specific circular diameter.

6.2 System Architecture

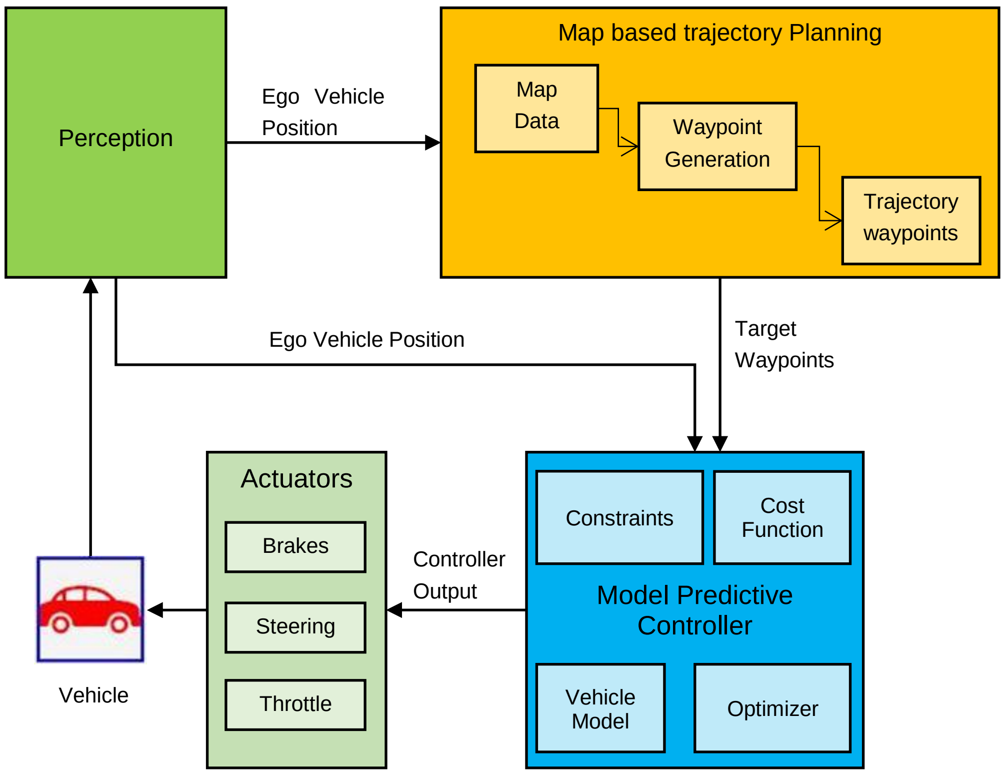


Figure 25: System Architecture

The system architecture for MPC based trajectory control for high dynamic maneuvers is as shown in Figure 25. It consists of Perception, planning and control modules. The perception module gives the information regarding the current state of the car. The state variables include position of the car in global coordinates, speed of the vehicle and orientation. The ego vehicle position is used by the planning module for generation of target waypoints for the controller. The planning module uses the predefined map data of maneuver profiles. Using the map data and locating the current position of the vehicle available from the perception module, the waypoints are generated. Before starting the simulation the maneuver for which the trajectory control has to be selected. Different maneuver specific maps are preloaded when the system model is initialized. The map data consists of the data related to position in x and y direction in global frame, heading angle and velocity to have perfect maneuvers. This map data is used to generate waypoints of equal gap. Knowing the ego position of the vehicle in global frame, the corresponding nearest waypoint in the map is detected. The eucledian distance between the current state of the car and all the waypoints is calculated. The minimum of all the distances provides the nearest waypoint in the map. From this detected waypoint in the map the next available waypoints are used as reference waypoints for the controller. MPC controller uses the mathematical vehicle model replicating the realistic behavior of the vehicle. The actual vehicle initial condition and the mathematical model initial conditions must be same before the simulation. The MPC controller also uses the current ego position of the vehicle and reference waypoints from the planning module to predict the future states of the car. The mathematical model is linearized to state space model for MPC to make future predictions. Also the constraints for the input and state variables must be specified before the simulation. The errors related to the position, orientation and velocities between the ego vehicle state and the target waypoints are used to formulate the cost function. Using the discretized vehicle model, constraints and cost function the optimizer performs computations to find the optimal control output to reach the target waypoints. Based on the prediction horizon of the controller, i.e the number of time steps in future the MPC predictions must be done for possible control values responsible for future vehicle states. Out of these predictions the control output with minimum cost is used for actuations. These actuations make the vehicle move toward the target waypoint. MPC makes predictions at each time step to produce optimal control signals to have best trajectory motion of the vehicle for the defined maneuver. The controller performance is tested for kinematic and dynamic vehicle models. The controller outputs differ with respect to the vehicle model used.

6.2.1 Perception

In reality sensors, cameras and other sensing devices are used for perception. In this thesis the simulation is performed in a simulation environment and hence state of the vehicle is extracted from the simulation environment creating an interface between the system model and the simulation model.

6.2.2 Map Based Trajectory Planning

6.2.2.1 Map data

The main focus of the thesis is to develop MPC for trajectory control and hence no route and maneuver planning algorithms are used for planning. The high dynamic maneuvers with accurate maneuver behaviors are simulated in CarMaker simulation environment without the controller. This maneuver data is extracted to use for trajectory planning. The path and maneuver is not generated, instead the predefined maneuvers available in CarMaker simulation environment is used. The high dynamic maneuvers are simulated in CarMaker using the inbuilt vehicle model in CarMaker capable of making accurate maneuvers for the given road profiles. The vehicle position, heading angle and velocities are recorded and stored in a form of matrix in a matfile to use it for waypoint generation. The map data consists of the vehicle coordinates namely global vehicle position (X, Y), heading angle (ψ) and velocity (v) that is recorded at each sampling time of the simulation in CarMaker.

6.2.2.2 Waypoint Generation

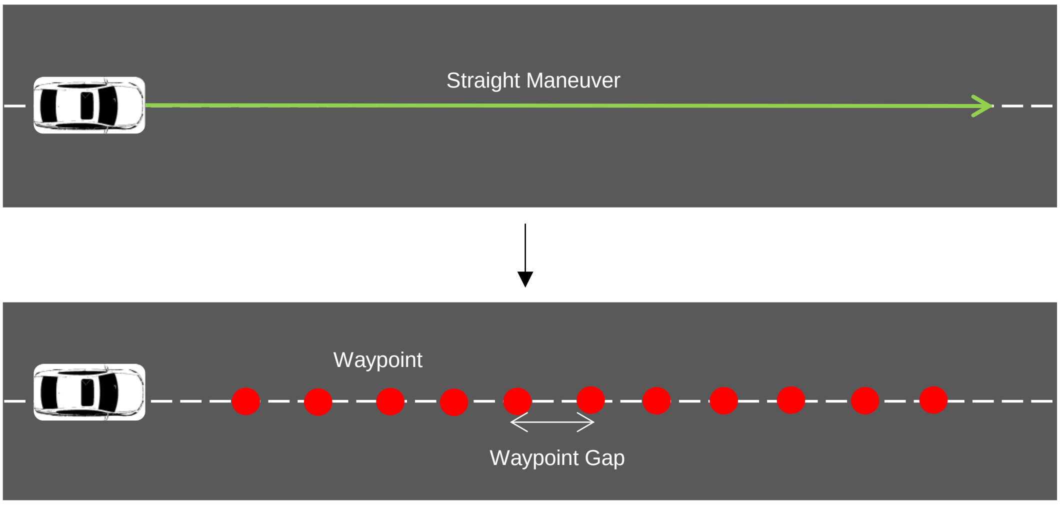


Figure 26: Maneuver to Waypoints

The maneuver data from the map is used to generate waypoints of equal gap to be used for trajectory planning. The vehicle coordinates from the map is used to generate waypoints, each waypoint is a state vector [X, Y, ψ, v]. The waypoint generation is done in two steps, first the coordinates of the vehicle for the maneuver to follow is detected depending on the selected maneuver for the simulation. Secondly the waypoint calculation is performed and equally spaced according to the predefined maneuver data that is available from the map data. The distance vector of X and Y co-ordinates of the map data is calculated and interpolation is performed for equal gap size. The generated waypoints are stored in the form of a matrix and later used for the calculation of target trajectory waypoints. The flowchart for waypoint generation is shown in Figure 27.

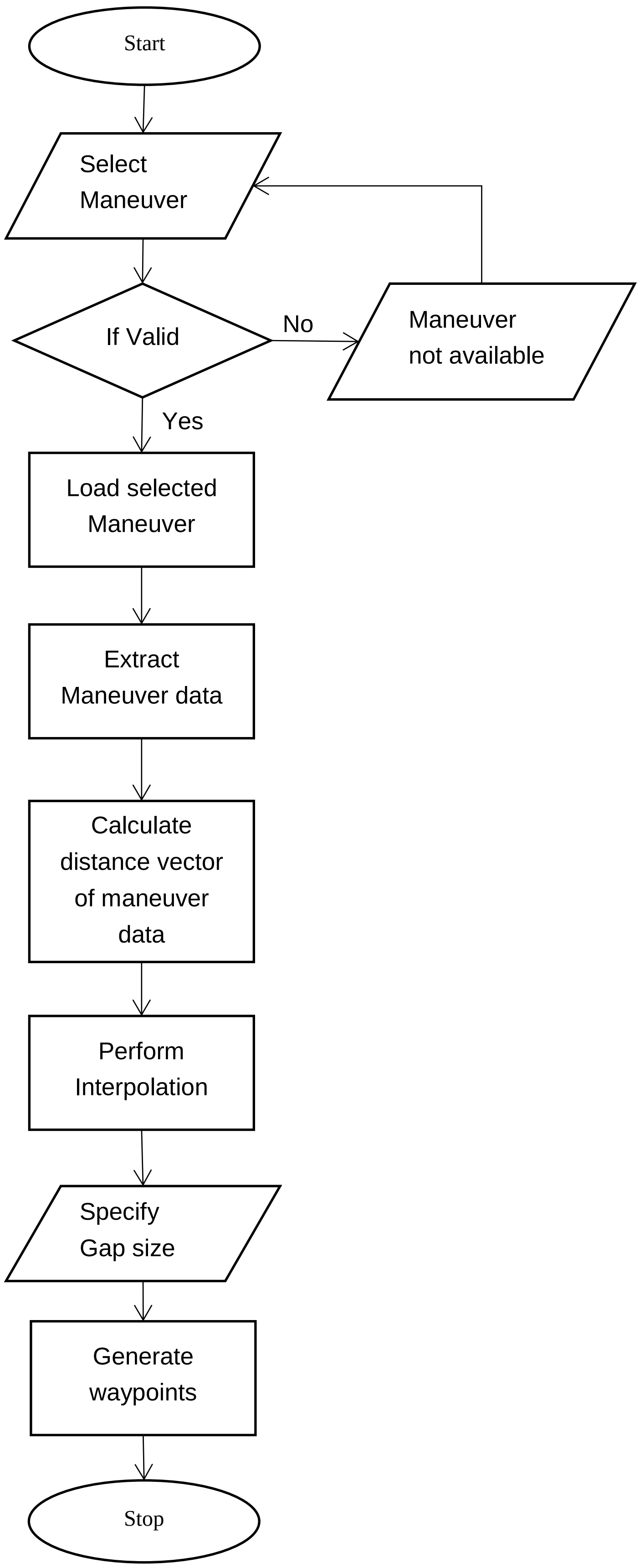


Figure 27: Waypoint Generation Flowchart

6.2.2.3 Trajectory Waypoints

The ego position of the vehicle along with the generated waypoints from the map data is used for target trajectory waypoints. Based on the current state of the car from perception is used to calculate the euclidean distance between the current ego position and the generated waypoints and stored in a matrix. The minimum of calculated distances is selected and the responsible corresponding waypoint in the generated waypoints matrix is used. From the selected waypoint a specific number of target waypoints in the generated waypoint matrix is used as target waypoints required for the controller. In this thesis we use 5 target waypoints that will be used as reference by the controller. The focus of thesis is trajectory control using MPC and hence basic approach is followed for calculating target trajectory waypoints. The flowchart for Trajectory waypoint calculation is in fig

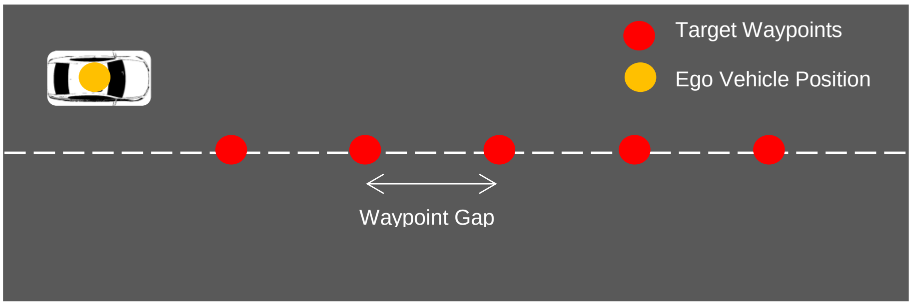


Figure 28: Target Trajectory Waypoints

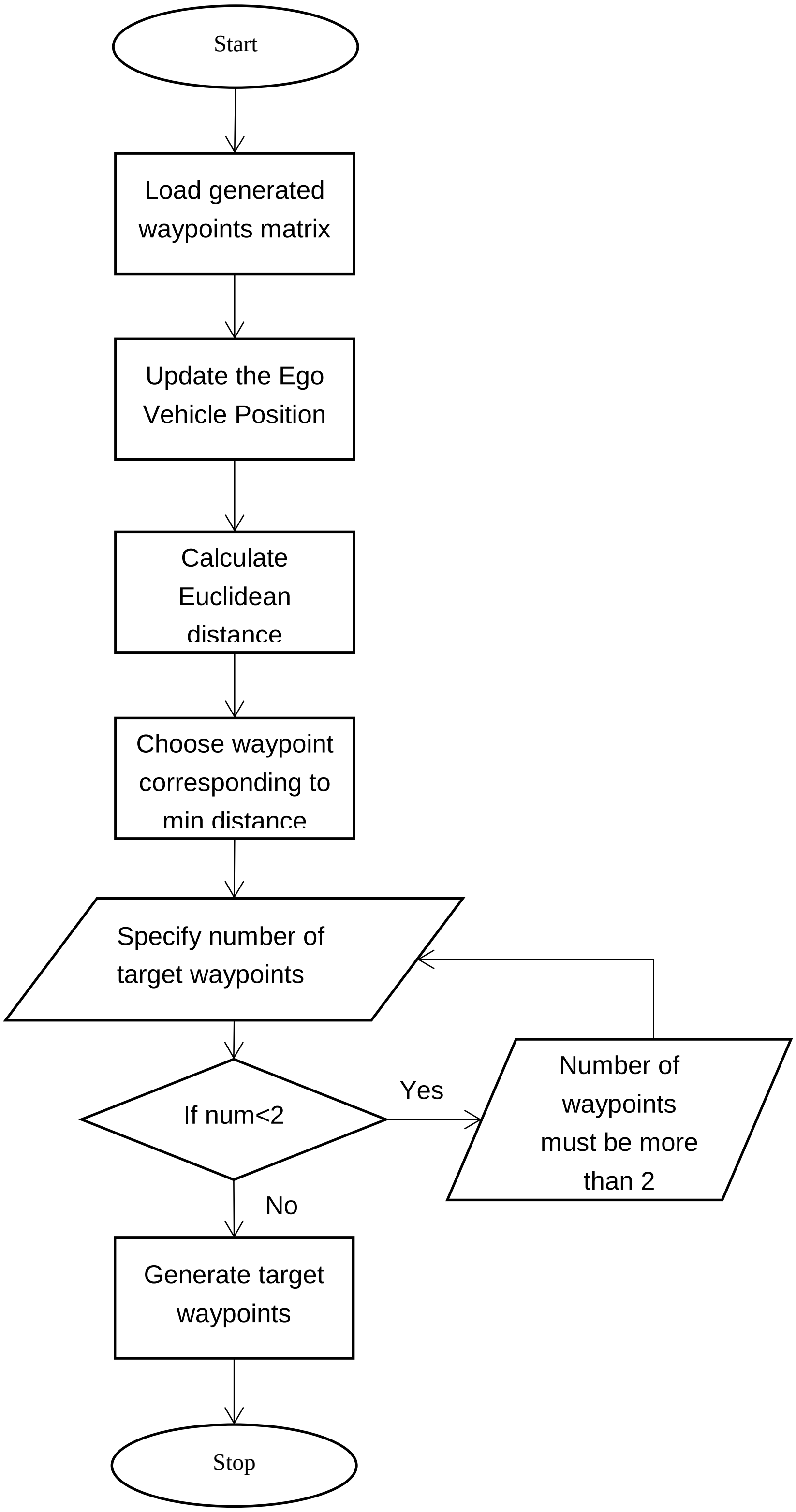


Figure 29: Flowchart of Target Waypoints Generation

6.2.3 *Model Predictive Controller for Trajectory Control*

6.2.3.1 *Errors used to formulate Cost function*

- Cross Track Error (Cte): The difference between the ego vehicle position and the reference track is called cross track error.

$$Cte = [(x_v, y_v) - (x_{ref}, y_{ref})]$$

Where,

x_v & y_v : Ego vehicle position in x and y directions

x_{ref} & y_{ref} : Reference position in x and y directions

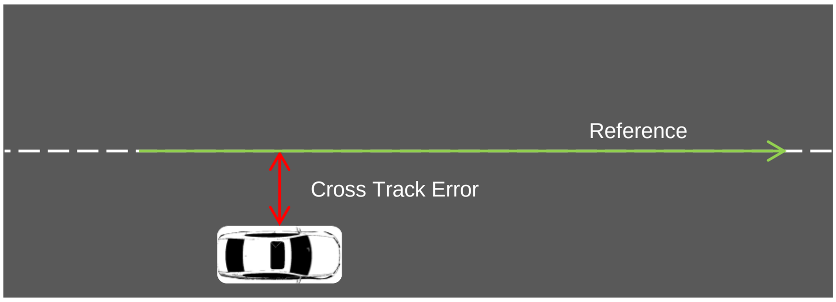


Figure 30: Cross Track Error

- Heading Angle Error (eψ): The error between the vehicle heading angle and reference heading angle is called heading angle error.

$$e\psi = \psi_v - \psi_{ref}$$

Where,

ψ_v : Vehicle heading angle

ψ_{ref} : Reference heading angle

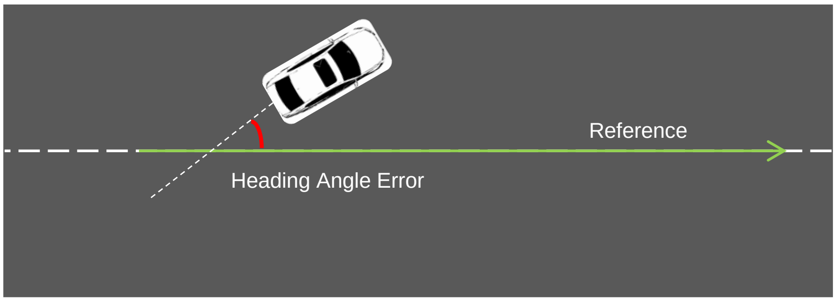


Figure 31: Heading Angle Error

- Velocity Error (ev): Difference between vehicle velocity and reference velocity

$$ev = v - v_{ref}$$

6.2.3.2 MPC for Kinematic Vehicle Model

Recall the kinematic vehicle model equations from section *5.3.2*

$$x' = vcos(\psi + \beta)$$

$$y' = vsin(\psi + \beta)$$

$$\psi' = \frac{v}{l_r}\sin(\beta)$$

$$v' = a$$

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

Written as $\xi' = f(\xi, u,)$ where $\xi = [x, y, \psi, v]$ is the state vector. x and y are position coordinates of the vehicle in x and y direction in global frame. Ψ is the heading angle of the car and v is velocity. The control inputs $u = [a, \delta]$, a is vehicle acceleration and δ is steering angle. Including the errors in the state vector, $\xi = [x, y, \psi, v, Cte, e\psi, ev]$ where *Cte* is cross track error, eψ is heading angle error and *ev* is the velocity error.

The cost function is formulated as

$$J = \sum_{i=1}^{H} w_{Cte}(Cte)^2 + w_{e\psi}(e\psi)^2 + w_{ev}(ev)^2$$

Where, w is the weights used for penalizing the cost function and H is the prediction horizon

Constrains for MPC with kinematic model are the inputs for the vehicle model, i.e steering angle (δ) and acceleration (a) and rate of change of these inputs.

$$\delta_{min} \leq \delta \leq \delta_{max}$$

$$\Delta\delta_{min} \leq \Delta\delta \leq \Delta\delta_{max}$$

$$a_{min} \leq a \leq a_{max}$$

$$\Delta a_{min} \leq \Delta a \leq \Delta a_{max}$$

Representing vehicle state in state space form

$$\xi_i^{k+1} = A\xi_i^k + Bu_i^k$$

Where k is the total number of time steps, A and B are state space matrices

MPC controller is formulated as

$$\min J_i(\xi_i, u_i) \;\; \forall i \in [0, \ldots., H]$$
$$\xi_i = [x_i, y_i, \psi_{i,} v_i, Cte_i, e\psi_i, ev_i]$$
$$u_i = [a_i, \delta_i]$$

Where, ξ is state vector, u is input vector and H is prediction horizon

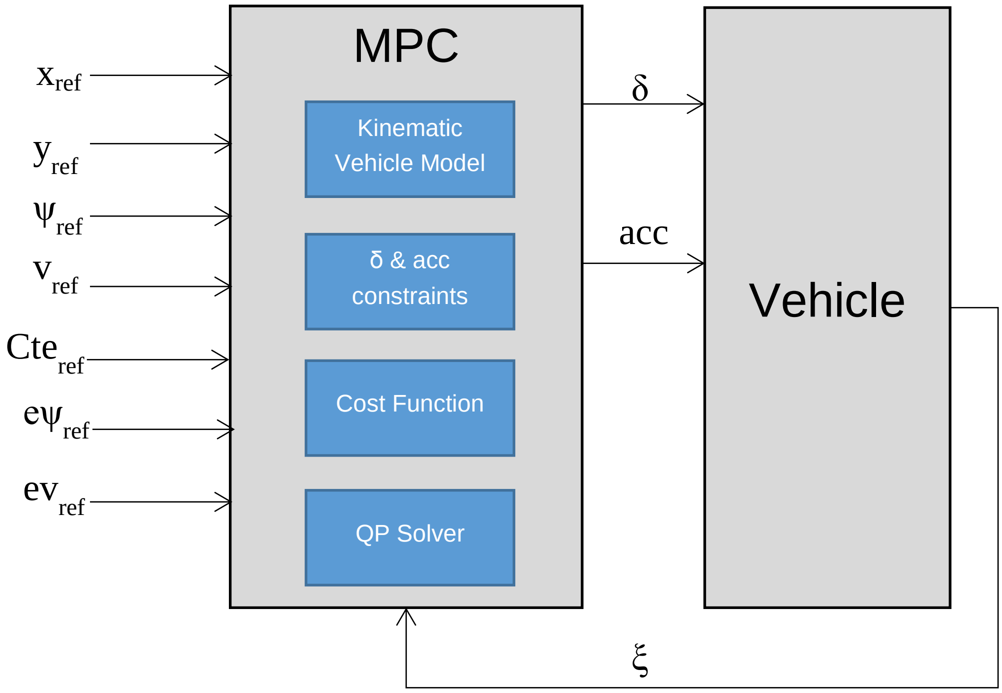


Figure 32:Structure MPC with Kinematic Model

6.2.3.3 MPC for Dynamic Model

Recalling the equations of dynamic vehicle model from section 5.3.3

$$X' = x' \cos(\psi) - y' \sin(\psi)$$
$$Y' = x' \cos(\psi) - y' \sin(\psi)$$
$$mx'' = my'\psi' + 2F_{xf} + 2F_{xr}$$
$$my'' = -mx'\psi' + 2F_{yf} + 2F_{yr}$$
$$I\psi'' = 2F_{yf} l_f - 2F_{yr} l_r$$

Where, X and Y are position coordinates of the vehicle in X and Y direction in global frame. Ψ is the heading angle of the car, x' is longitudinal velocity and y' is lateral velocity of the vehicle. Written as $\xi' = f(\xi, u,)$ The control inputs u = [T_{bf}, T_{br}, δ], T_{bf} *is* brake torque of front wheel, T_{br} is brake torque of rear wheel and δ is steering angle. Including the errors in the state vector, ξ= [*X, Y, ψ, v, Cte, eψ, ev*] where *Cte* is cross track error, eψ is heading angle error and *ev* is the velocity error.

The cost function is formulation is same as the cost function used for kinematic model since we are using errors for optimization, which is

$$J = \sum_{i=1}^{H} w_{Cte}(Cte)^2 + w_{e\psi}(e\psi)^2 + w_{ev}(ev)^2$$

Where, w is the weights used for penalizing the cost function and H is the prediction horizon.

Constrains for MPC with dynamic model are the inputs for vehicle model, i.e. steering angle (δ) and Brake torque of front wheel (T_{bf}), brake torque of rear wheel (T_{br}) and rate of change of these inputs.

$$\delta_{min} \le \delta \le \delta_{max}$$
$$\Delta\delta_{min} \le \Delta\delta \le \Delta\delta_{max}$$
$$T_{bf_{min}} \le T_{bf} \le T_{bf_{max}}$$
$$\Delta T_{bf_{min}} \le \Delta T_{bf} \le \Delta T_{bf_{max}}$$
$$T_{br_{min}} \le T_{br} \le T_{br_{max}}$$
$$\Delta T_{br_{min}} \le \Delta T_{br} \le \Delta T_{br_{max}}$$

Representing vehicle state in state space form

$$\xi_i^{k+1} = A\xi_i^k + Bu_i^k$$

Where k is the total number of time steps, A and B are state space matrices

MPC controller is formulated as

$$\min J_i(\xi_i, u_i) \ \forall i \in [0, \dots., H]$$
$$\xi_i = [x_i, y_i, \psi_{i,} v_i, Cte_i, e\psi_i, ev_i]$$
$$u_i = [T_{br_i}, T_{br_i}, \delta_i]$$

Where, ξ is state vector, u is input vector and H is prediction horizon.

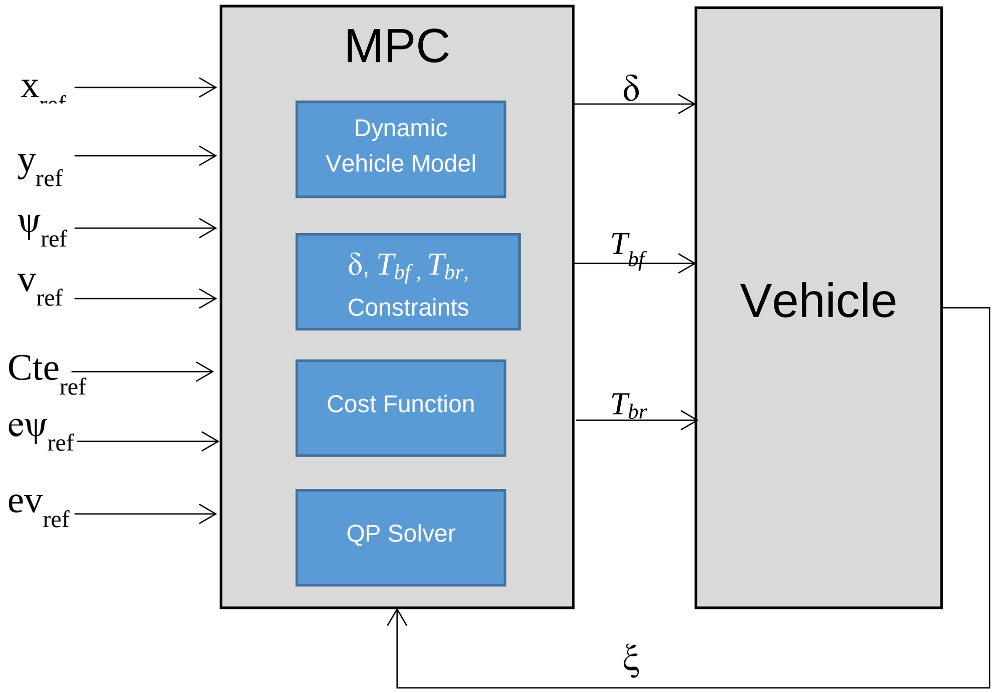


Figure 33: Structure of MPC with Dynamic Model

6.2.3.4 QP Solver

The optimizer used for MPC is QP solver which is the default optimizer for MPC in MATLAB Simulink. The optimization is carried out at each sampling time and it is computationally expensive. To reduce the complexity the vehicle model and constraints are linearized.

$$\min_{u} \sum_{i=1}^{H-1} \xi_i^T Q \xi_i + u_i^T R u_i$$
$$subject\ to\ \xi_{i+1} = A\xi_i + Bu_i$$

Where, Q and R are state weight and control weight matrices. The vehicle models used are non-linear and hence it must be linearized. The linearization is done by the MPC controller itself. The linearized model is given by

$$\xi_{i+1} = A_i \xi_i + B_i u_i$$

$$A_i = \frac{\partial f(\xi, u)}{\partial \xi} |_{\xi=\xi_{ri}, u=u_{ri}}$$
$$B_i = \frac{\partial f(\xi, u)}{\partial u} |_{\xi=\xi_{ri}, u=u_{ri}}$$

Where, ξ_{ri} and u_{ri} are the reference for state and control signals. Until now the minimization of ξ and u was carried out. The minimization of the errors between actual state, control signals and their references is formulated as

$$\min_{\tilde{u}} \sum_{i=1}^{H-1} \tilde{\xi}_i^T Q \tilde{\xi}_i + \tilde{u}_i^T R \tilde{u}_i$$
$$subject\ to\ \tilde{\xi}_{i+1} = A\tilde{\xi}_i + B\tilde{u}_i$$
$$\tilde{\xi}_i = \xi_i - \xi_{ref} i$$
$$\tilde{u}_i = u_i - u_{ref}$$

The optimization of the above formulation is solved as a QP (Quadratic problem) since the cost function for our MPC is a quadratic equation. And hence the default solver available with MPC in Simulink was considered. The general QP problem is of the form [21],

$$\min_{u} \frac{1}{2} u^T H u + f^T u$$

Since we want to minimize the errors between the actual state and the reference, the QP formulation becomes

$$\min_{u} \frac{1}{2} \tilde{u}^T H \tilde{u} + f^T \tilde{u}$$

In order to consider the cost function formulation in the above form let us consider two vectors as below,

$$\bar{\xi}(i+1) = \begin{bmatrix} \tilde{\xi}_{i+1|i} \\ \tilde{\xi}_{i+2|i} \\ \vdots \\ \tilde{\xi}_{i+H|i} \end{bmatrix}$$

$$\bar{u}(i) = \begin{bmatrix} \tilde{u}_{i|i} \\ \tilde{u}_{i+1|i} \\ \vdots \\ \tilde{u}_{i+H-1|i} \end{bmatrix}$$

Where, $i+H|i$ indicates the estimated value at sampling instance i+1 that is predicted at instance i. Using the vectors cost function can be formulated as,

$$J = \bar{\xi}^T \bar{Q} \bar{\xi} + \bar{u}^T \bar{R} \bar{u}$$

Where, $\bar{Q} = diagonal(Q)$ and $\bar{R} = diagonal(R)$. The evaluation of state and control signals over the prediction horizon is described using the state space matrices

$$\bar{A}(i) = \begin{bmatrix} A_{i|i} \\ A_{i|i}A_{i+1|i} \\ \vdots \\ a(i,1,0) \end{bmatrix}$$

$$\bar{B}(i) = \begin{bmatrix} B_{i|i} & 0 & \dots & 0 \\ B_{i|i}A_{i+1|i} & B_{i+1|i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a(i,1,1)B_{i|i} & a(i,1,2)B_{i+1|i} & \dots & B_{i+H-1|i} \end{bmatrix}$$

Where,

$$a(i,j,l) = \prod_{k=H-j}^{l} A_{i+k|i}$$

Using the vectors $\bar{\xi}(i+1)$, $\bar{u}(i)$ along with the matrices $\bar{A}(i)$ and $\bar{B}(i)$ in the state space equation form the prediction formulation is

$$\bar{\xi}_{i+1} = \bar{A}\bar{\xi}_i + \bar{B}\bar{u}_i$$

The minimization of the cost function formulation as per the general QP problem becomes

$$J = \frac{1}{2}\bar{u}^T H_i \bar{u} + \xi_i^T \bar{u}$$

Where,

$$H_i = 2(\bar{B}_i^T \bar{Q} \bar{B}_i + \bar{R})$$
$$\xi_i = 2\bar{B}_i^T \bar{Q} \bar{A}_i$$

7 Simulation Environment

This chapter presents the simulation environment used to evaluate the performance of MPC for both kinematic and dynamic models for high dynamic maneuvers. The implementation of derived MPC controller is developed in MATLAB/Simulink and tested in a virtual test driving simulator tool by IPG CarMaker. The MPC controller from the MPC tool box in MATLAB/Simulink is used. The optimizer used for MPC optimizations is QP solver which is the default solver implicitly available for the controller with the MPC toolbox. The system and the controller design is described in the previous chapter. The vehicle dynamics are simulated using the vehicle models described in Chapter 5. The Constraints and cost functions are implemented using MPC designer that is available with the Simulink MPC toolbox. The vehicle model parameters and MPC parameters is in the Appendix.

7.1 CarMaker

CarMaker is a virtual test driving simulation tool used to develop and test systems and functions of a vehicle in realistic scenarios. The real world scenarios are accurately modelled describing the surrounding environment of a vehicle in the virtual world. CarMaker is test platform that can be integrated throughout the development process from SiL, HiL and MiL.

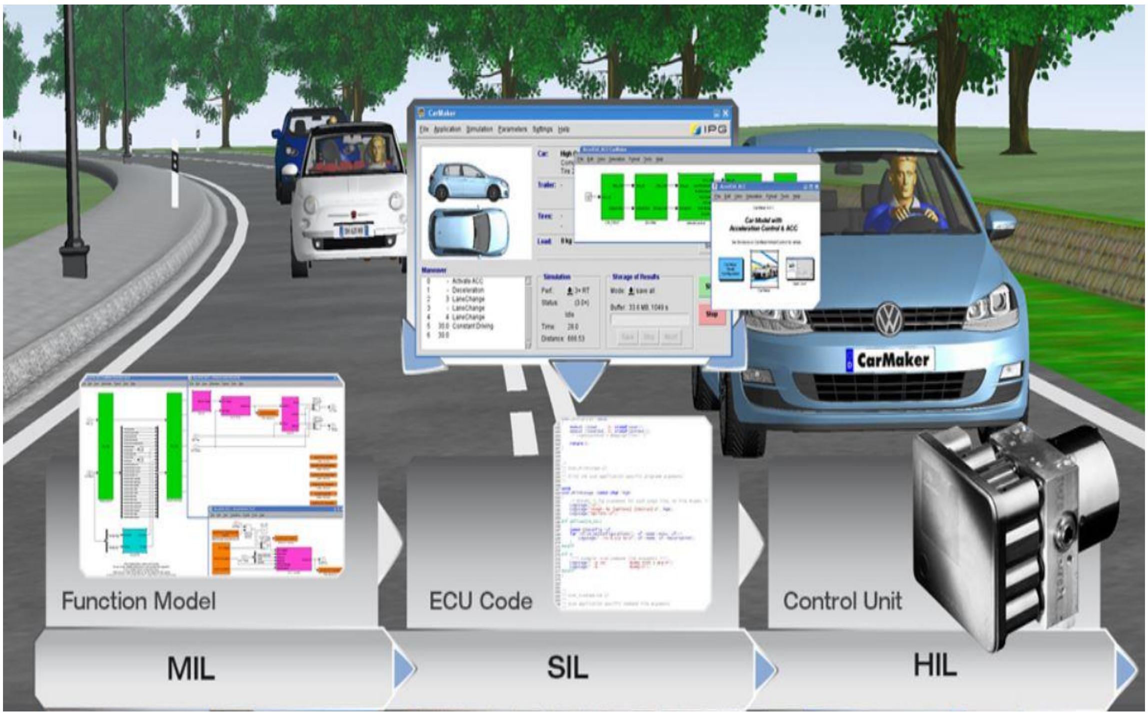


Figure 34: Integration of CarMaker with MATLAB and Simulink [22]

CarMaker for Simulink is a complete integration of IPG's test and simulation platform in to modeling and simulation environment of MATLAB and Simulink. CarMaker is a software tool used for development, testing and model design in the field of vehicle dynamics. Automotive control systems such as ACC, ABS, ARS, engine control systems and many other control systems can be developed and tested using CarMaker for Simulink. Using MATLAB S-functions and Simulink API functions the features of CarMaker are integrated to Simulink environment providing high performance and stability. The CM4SL blocks are similar to the built-in Simulink blocks with the same type of connecting blocks enabling the addition of existing Simulink blocks to the CarMaker vehicle model. The CarMaker GUI is used for simulation control, vehicle parameters adjustments, create road configurations and to define maneuvers. It is also used for data analysis and creation of realistic animation and graphics that brings the vehicle model to life by rendering the vehicle model into three dimensional space. CarMaker in Simulink environment is similar to Simulink S-function blocks. It can be connected in the similar way the blocks in Simulink are interconnected. CarMaker caters realistic animation in three dimensional space for visualization of the vehicle simulation. Access to CarMaker simulation results is granted using the *cmread* utility that can be called within MATLAB. This utility loads data from any CarMaker simulation result file into the MATLAB workspace. Furthermore, the data can be manipulated and viewed, for post processing purposes, using any of the available MATLAB tools.

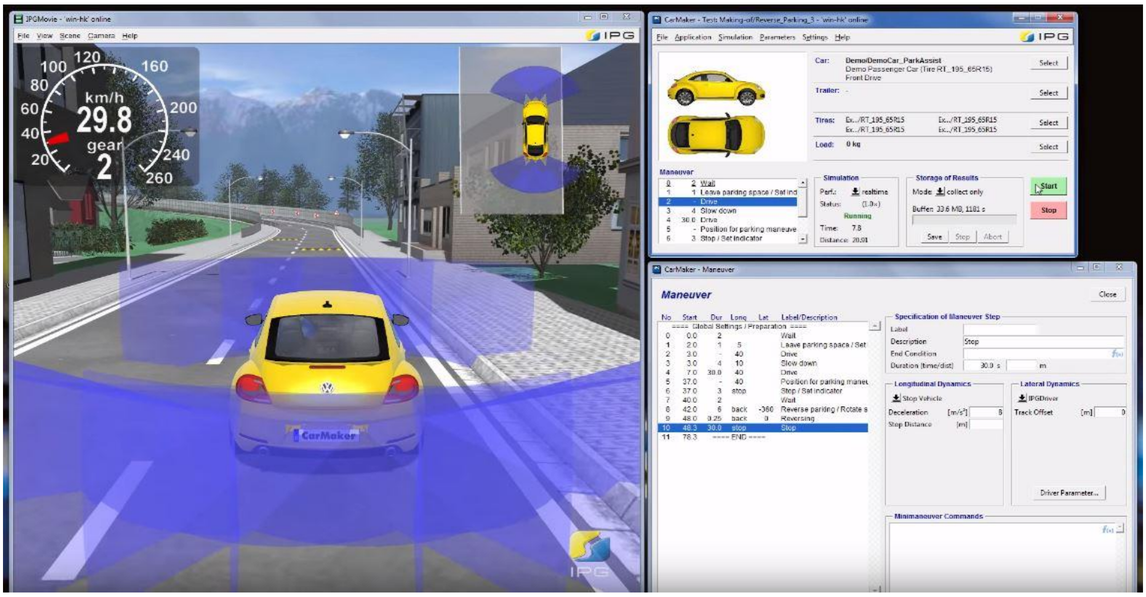


Figure 35: CarMaker GUI

7.2 Simulation Model

7.2.1 Overall Simulation Model

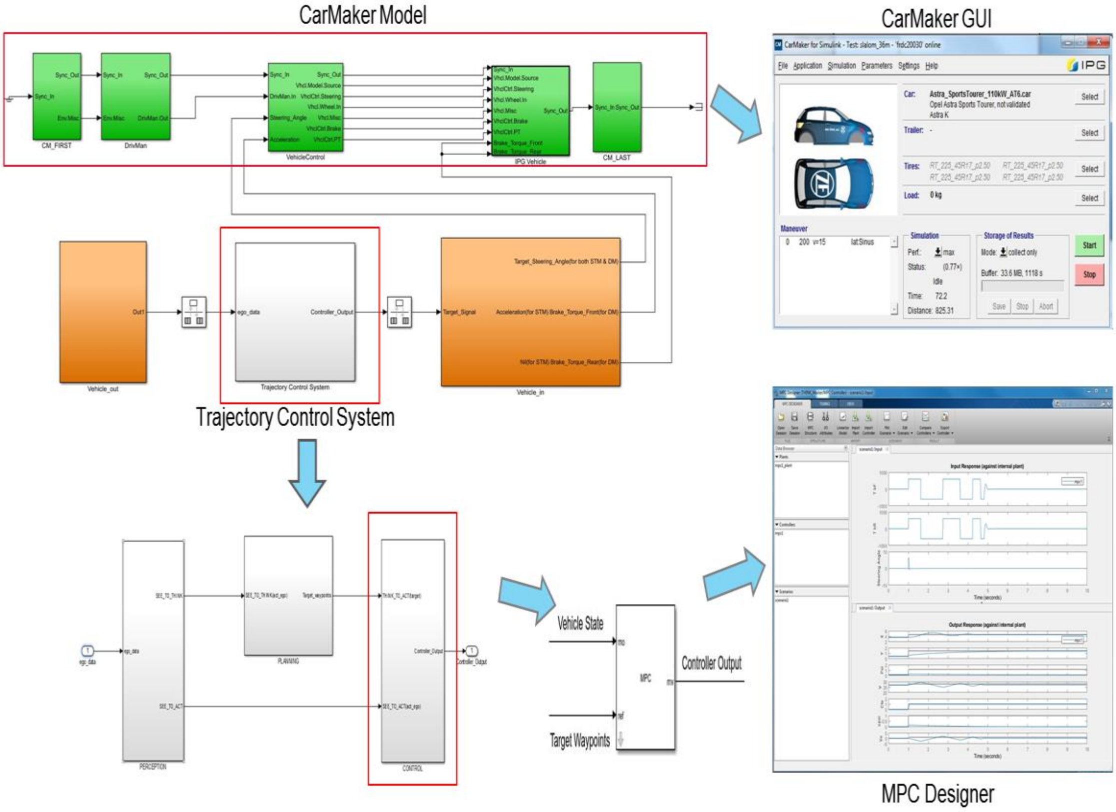


Figure 36: Simulation Environment

7.2.2 Trajectory Control System

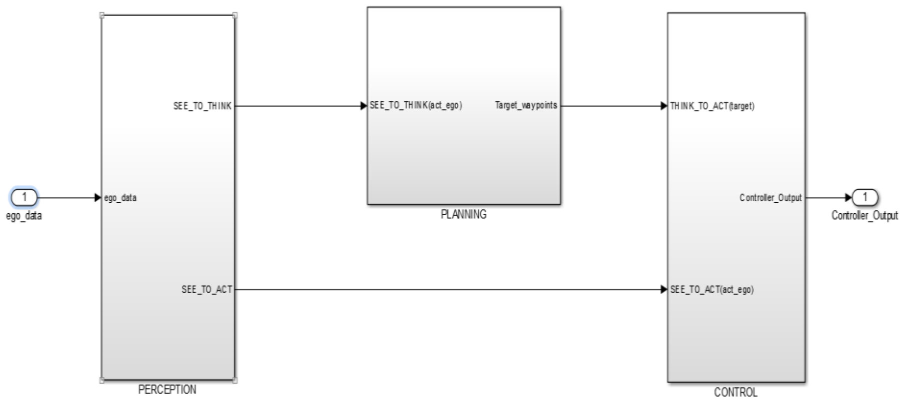


Figure 37: Trajectory Control System

7.2.3 Perception

For perception the vehicle state is extracted from CarMaker model using Read function.

7.2.4 Planning Model

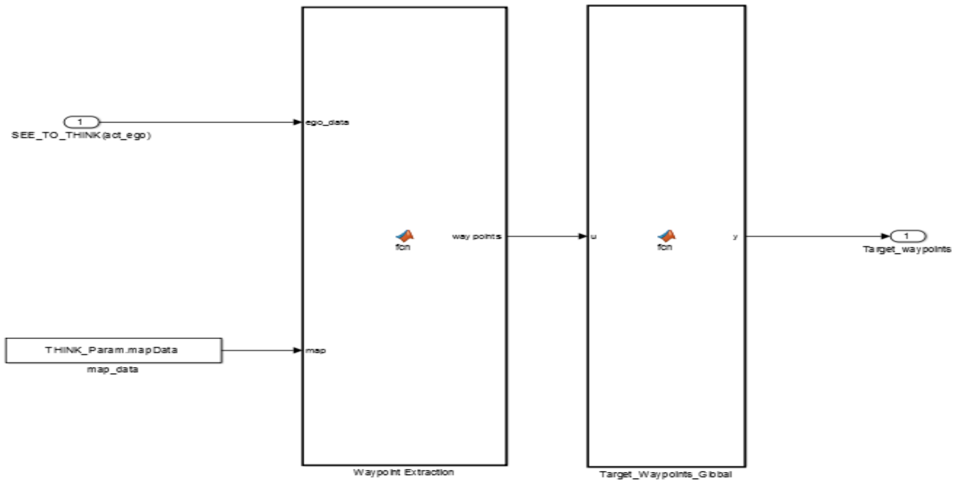


Figure 38: Planning Model

7.2.5 MPC Control Model

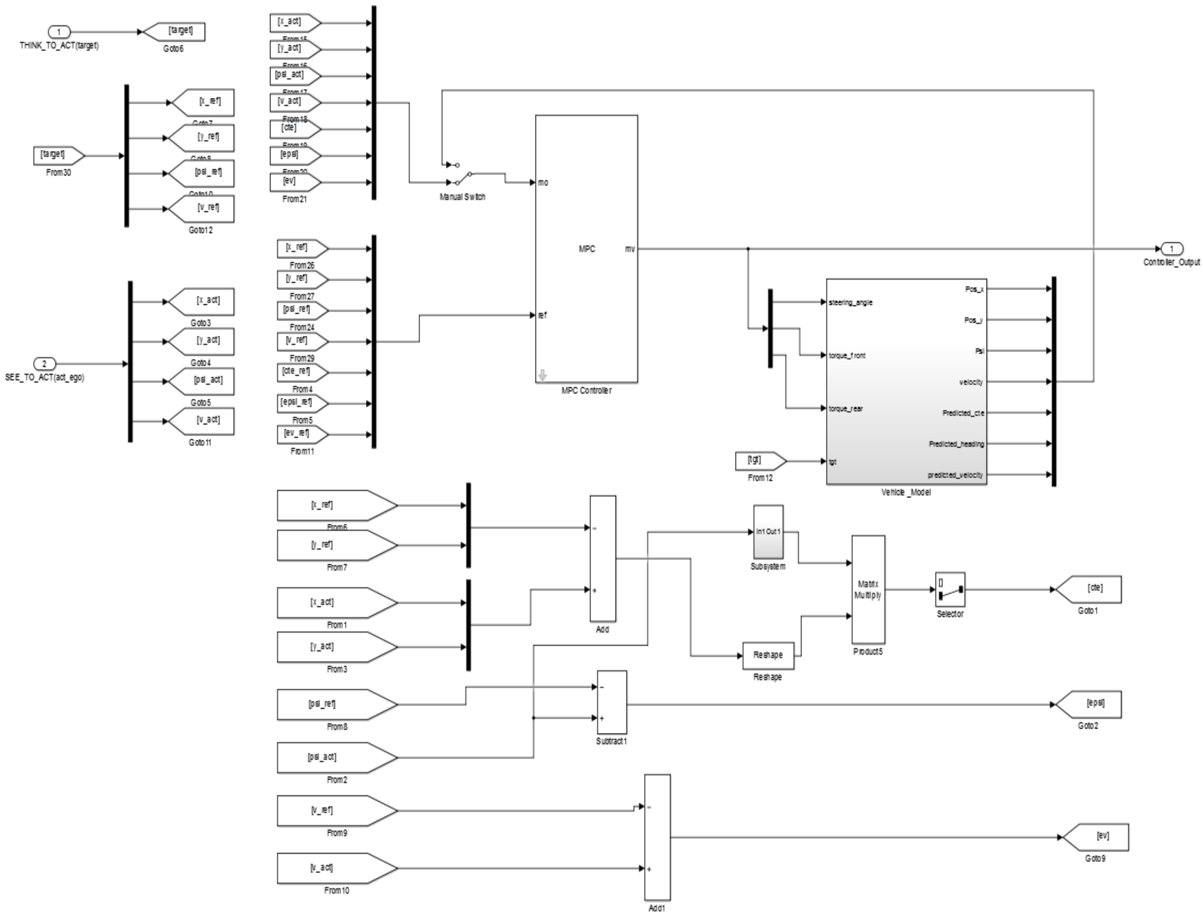


Figure 39: Control Model

8 Results

This chapter presents the results from simulations for different scenarios. The performance of model predictive controller with both kinematic and dynamic model is evaluated for different maneuvers with different speed profiles. The prediction horizon of MPC is 10 steps and the controller sampling time is 0.01s. The vehicle and controller parameters is in the appendix.

8.1 Scenario 1: Slalom Maneuver 36m

The distance between the cones is 36m as shown in Figure 40 and the waypoint gap between the trajectory waypoints is 5m. The simulation is tested with different speed profiles of 15, 50, 80 and 100kmph.

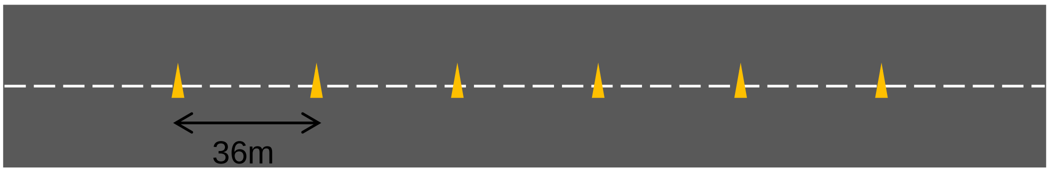


Figure 40: Slalom-36m

8.1.1 MPC performance for Slalom-36m maneuver with dynamic vehicle model

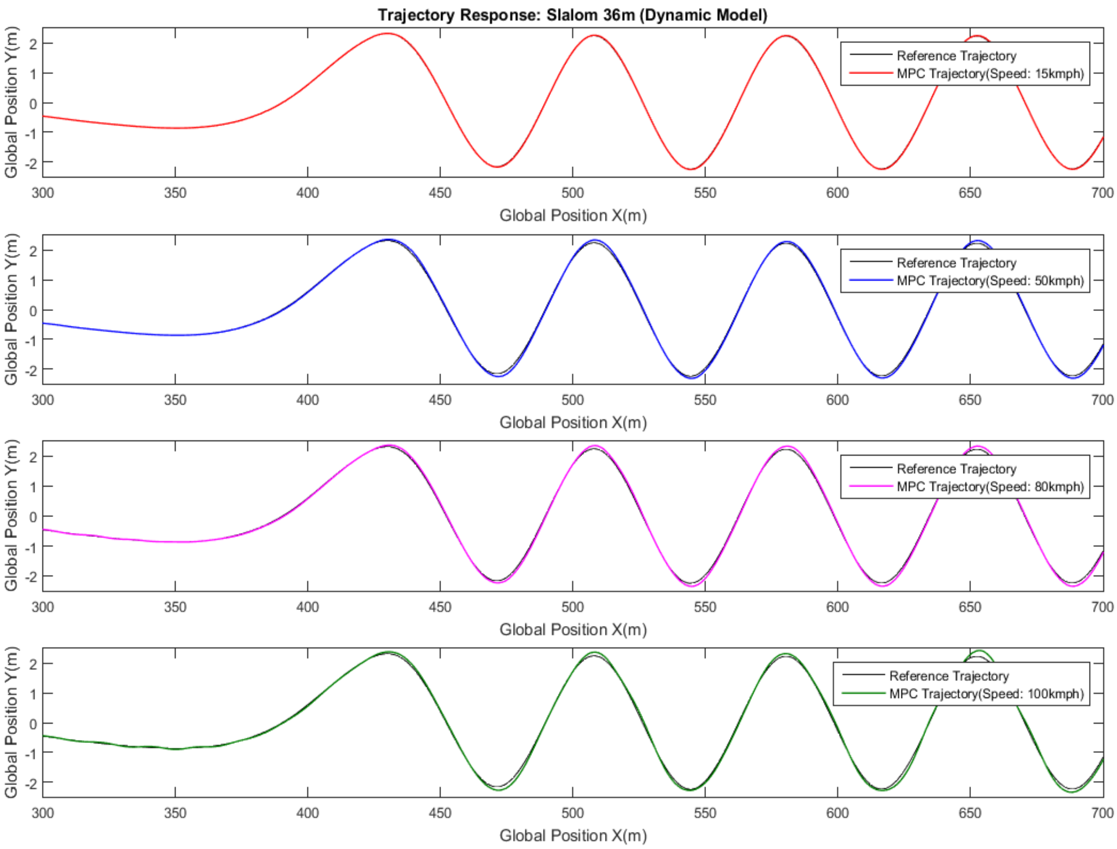


Figure 41: Trajectory response of MPC with dynamic vehicle model for Slalom-36 maneuver

The vehicle model used for evaluation of MPC is dynamic vehicle model (**Error! Reference source not found.**). The trajectory response of MPC with dynamic model is as shown in Figure 41. The steering angle limits used for the simulation is –60 degrees to 60 degrees. The rate limits for steering angle is set to -15 deg to 15 deg. The constraints for brake torques is -600Nm to 600Nm. The brake torques and steering angle are the controller ouputs used to control the vehicle motion. At any point of time the brake torques are either 600Nm or -600Nm. The brake torques are applied to the vehicle wheels inorder to reduce the speed of vehicle to maintain the trajectory as close as possible to the reference trajectory. First simulation was performed at slower speed of 15km/h. The overall simulation time is set to 80 seconds so that the vehicle will cover the entire road length of the slalom maneuver. It is observed that at low speed profile the performance of MPC for the given set of constraints is accurate. For speed profile of 50km/h there is a slight deviation from the reference and the deviation at the cornering tend to increase as speed increases.

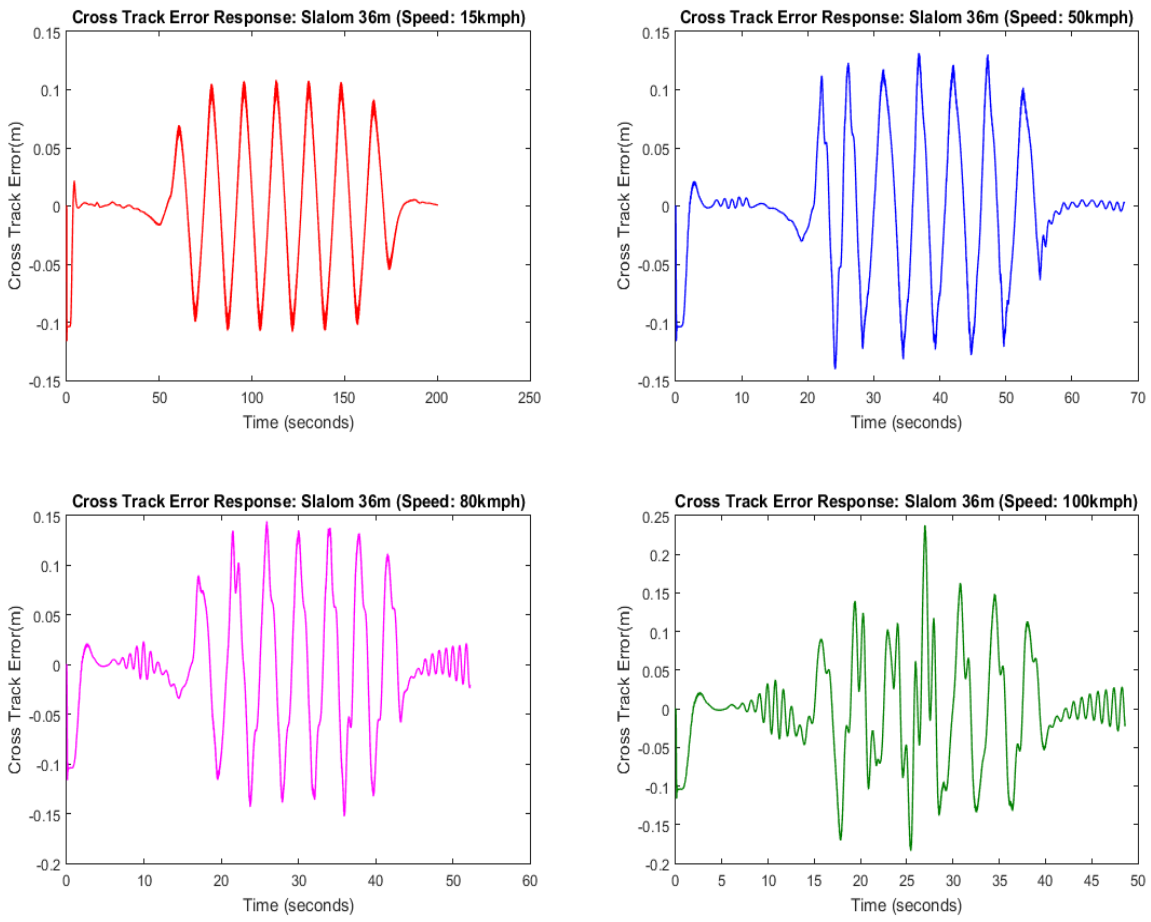


Figure 42: Cross Track Error Response of MPC with dynamic vehicle model for Slalom-36m maneuver

From Figure 42**,** The maximum deviation of the vehicle from the reference is observed to be 0.24m for vehicle speed of 100kmph. This is due to the influence of vehicle speed and waypoint gap of reference trajectory. For controller operating at 0.01s of sampling time, if the vehicle speed is faster, then the target waypoints are appearing very fast due to which the controller has to make more computations for reaching the reference. The controller has to make faster predictions and generate steering angle signals. As we can see the deviations are only observed when the vehicle is supposed to make cornering. This is because the controller makes better computations where the constraints of the vehicle have less influence on motion. During cornering since the vehicle is limited with a steering angle rate of -15deg to 15deg, the controller makes the best possible computations for these rate limits. Increase in constraints and its limits gives better performance but there is more possiblities for the vehicle to go out of the lane when travelling at high speed. Eventhough the reference speed is set to 50kmph the torques generated by the controller reduces the speed of vehicle. The velocity error response from Figure 43 depicts the difference in velocities of the reference speed and the vehicle speed. During cornering the speed of the vehicle is reduced and increased inorder to reach the reference trajectory position for higher speed profiles. This is mainly because the cost function weights are given more for the position of the vehicle rather than velocity of the vehicle. The controller produces steering angle signals within the constraint limits, more the vehicle speed more possibility of reaching maximum steering angle limits (Figure 44).

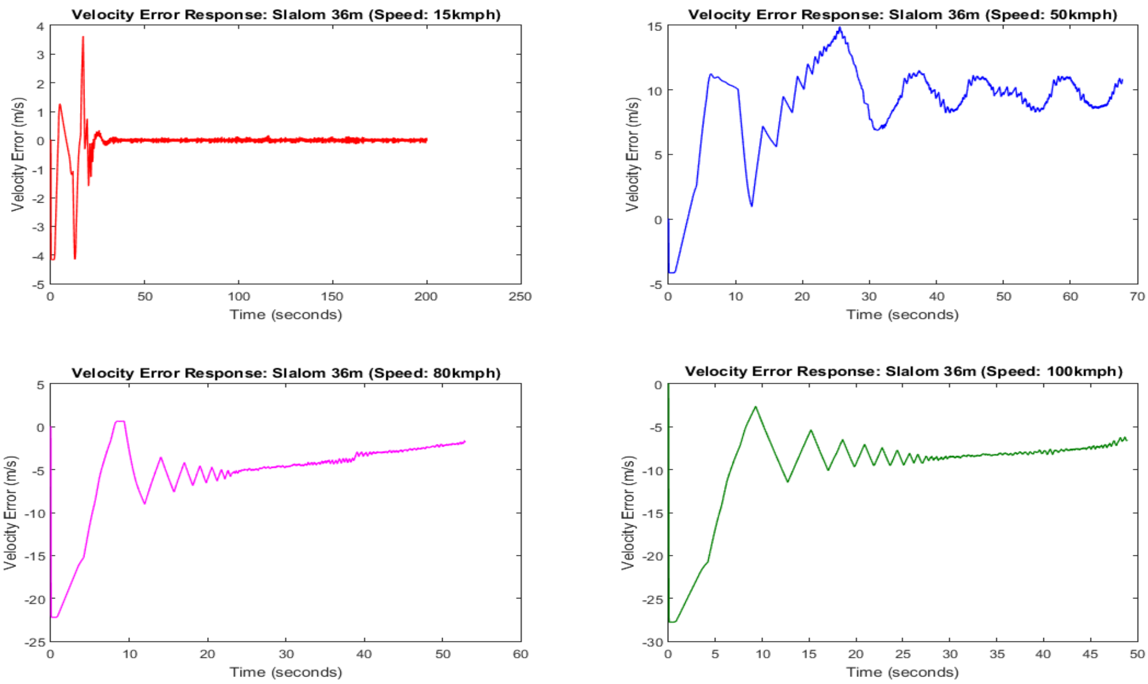


Figure 43: Velocity Error Response of MPC with dynamic vehicle model for Slalom-36m maneuver

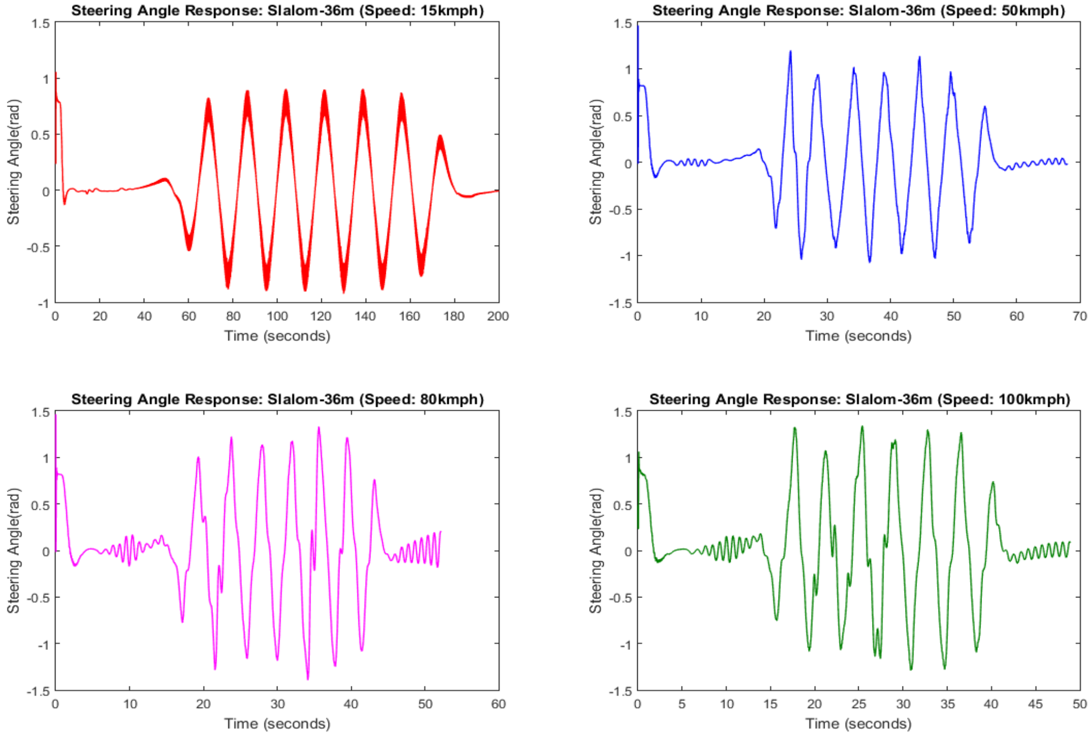


Figure 44: Steering Angle Response of MPC with dynamic vehicle model for Slalom-36m maneuver

8.1.2 MPC with Kinematic model for Slalom-36m maneuver

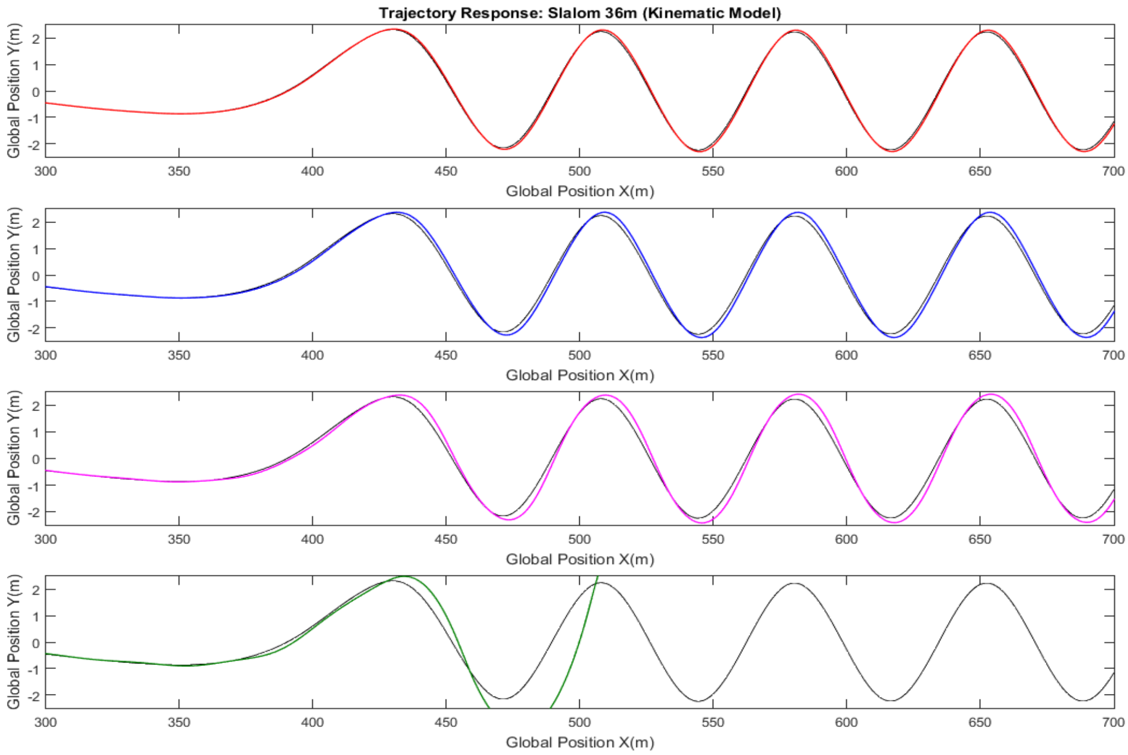


Figure 45:Trajectory response of MPC with kinematic vehicle model for Slalom-36 maneuver

The vehicle model used for evaluation of MPC is kinematic vehicle model (**Error! Reference source not found.**). The trajectory response of MPC with kinematic model is as shown in Figure 45. The steering angle limits used for the simulation is same as its used for MPC with dynamic vehicle model which is –60 degrees to 60 degrees. The rate limits for steering angle is set to -15 deg to 15 deg. For kinematic model the controller outputs for controlling vehicle motion are steering angle and acceleration. The speed of the vehicle is mainly controlled by acceleration and deceleration. The acceleration limits are -5m/s^2 to -5m/s^2. From (Figure 45) it is observed that the performance of MPC for lower vehicle speed is better with some deviations. As speed increases the deviations are more and for speed profile of 100kmph the vehicle is observed going out of the lane. This behaviour is mainly because kinematic vehicle models are based on the geometry of vehicle. The vehicle dynamics are not considered in modelling. The vehicle speed is mainly controlled by acceleration and deceleration. For higher speed profiles with a waypoint gap of 5m the controller is incapable of making faster predictions and produce control signals. Kinematic models are less detailed models than dynamic models, hence it is evident that kinematic models perform lesser than dynamic models. From (Figure 46) We can understand that for higher speed profiles the lateral acceleration of the vehicle also increases. For MPC controller with kinematic model the motion of the vehicle can be controlled in a better way implementing a braking controller. By which the acceleration signals are converted to braking signals.

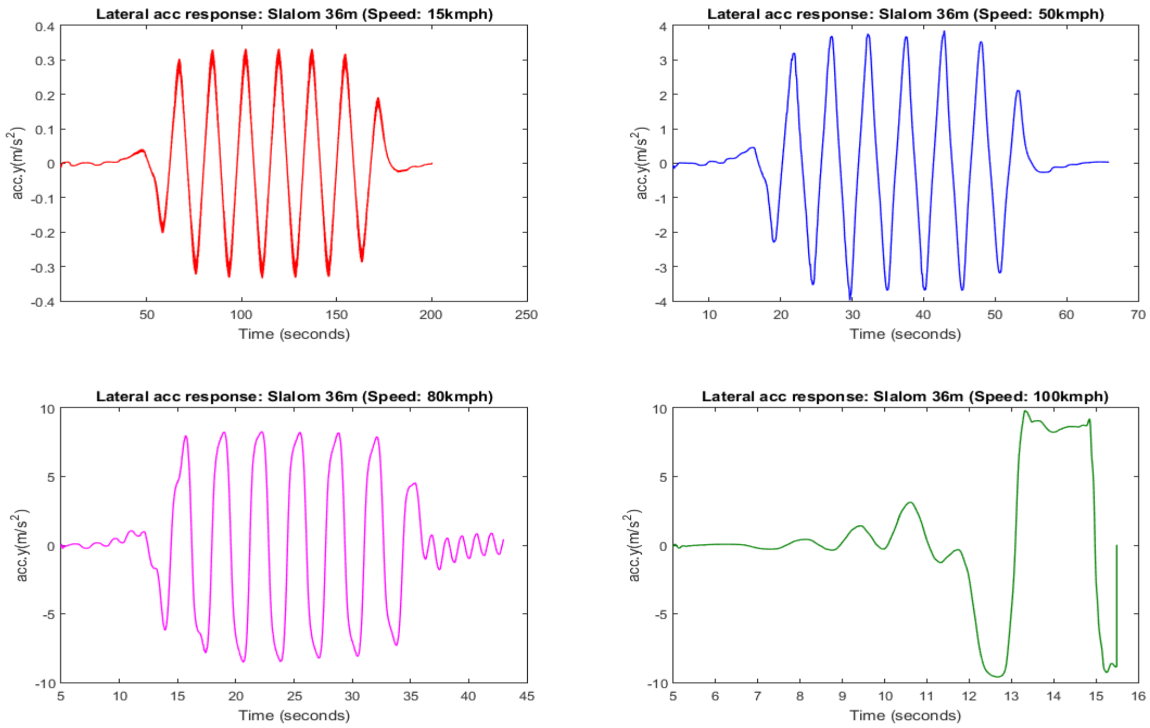


Figure 46: Lateral acceleration response of MPC with kinematic vehicle model for Slalom-36m maneuver

The fluctuations observed in the accleration response is because kinematic model used for MPC is a simplified vehicle model and there exists inaccuracy with the vehicle model used in simulation environment (CarMaker). The maximum deviation of vehicle position with respect to reference is found to be maximum of 0.29m for 15kmph, 3.9m for 50kmph and 8.1m for 80kmph. It is clear that for higher speed the vehicle tends to deviate more from the reference trajectory and hence the cross track error increases with respect to speed(Figure 47). The vehicle tend to reach maximum steering angle limits when travelling with high speed. While making turns between the cones the vehicle is limited with a steering angle rate of -15deg to 15deg and steering angle limits of 60 degrees. The controller computations based on these rate limits and hence for higher speed profiles the vehicle tend to have maximum steering angle. For vehicle speed of 100kmph the control tends to execute more actuations than the constraint limits, but because of the rate limits of steering angle with high speed the vehicle travels out of the lane (Figure 48). From the performance of MPC with both kinematic and dynamic models it is evident that the performance of MPC with dynamic model is better than kinematic model. The reason is because dynamic models are more detailed models and the vehicle is controlled by torques and forces where as with kinematic model there is no braking logic applied. Also there is no seperate longitudinal controller for controlling vehicle longitudinal motion.

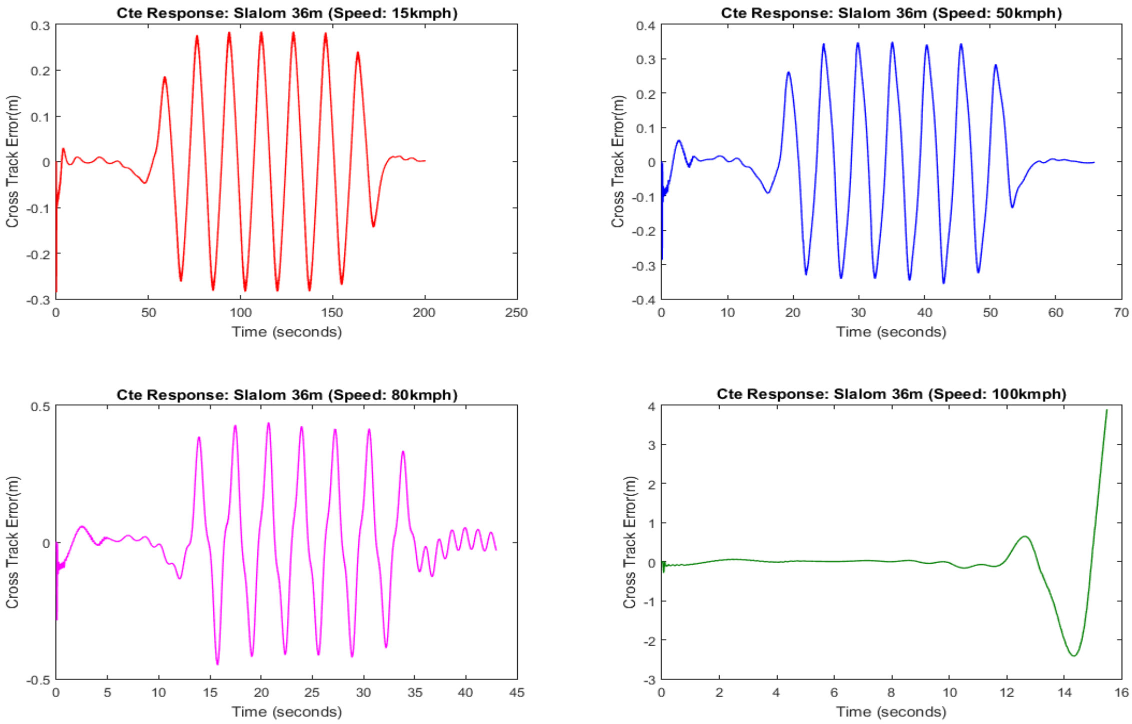


Figure 47: Cross Track Error Response of MPC with kinematic model for Slalom-36m maneuver

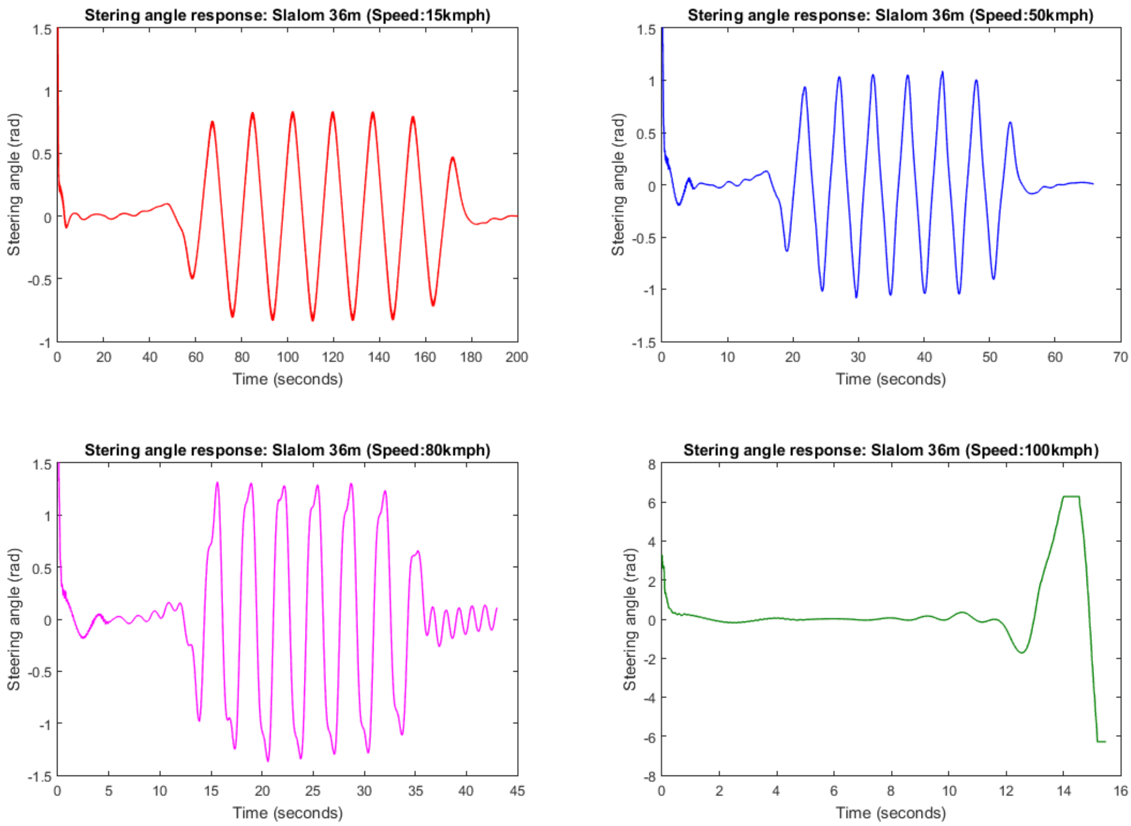


Figure 48: Steering angle response of MPC with kinematic vehicle model for Slalom-36m maneuver

8.2 Scenario 2: Lane Change ISO Maneuver

The lane change maneuver consists of an entry lane, side lane and exit lane, the vehicle is driven through the cones of entry lane, then changing the lane into the side lane of specific distance and then going into the exit lane. The width and length of the lanes are based on ISO 3888-2 standard used to evaluate vehicle handling performance [23].

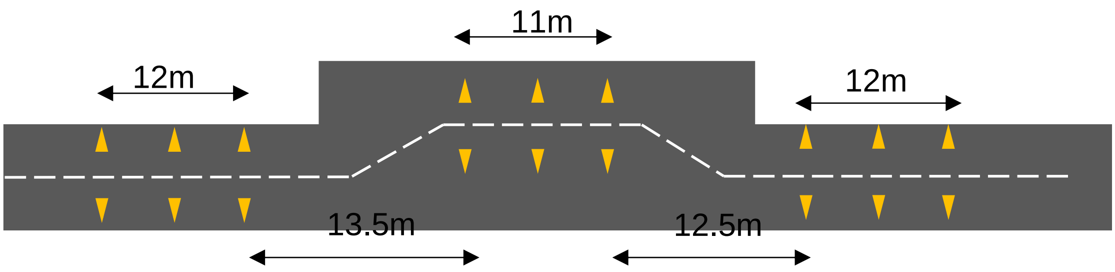


Figure 49: Lane Change ISO Maneuver

8.2.1 MPC for Lane Change ISO maneuver with dynamic vehicle model

The trajectory response of MPC with dynamic model for lane change maneuver is as shown in Figure 41. The steering angle limits used for the simulation is –45 degrees to 45 degrees. The rate limits for steering angle is set to -15 deg to 15 deg. The constraints for brake torques is -600Nm to 600Nm. The brake torques and steering angle are the controller ouputs used to controlling the vehicle trajectory. The brake torques are applied to the vehicle wheels control the vehicle speed order to maintain the trajectory as close as possible to the reference trajectory. First simulation was performed at slower speed of 15km/h. The overall simulation time is set to 85 seconds so that the vehicle will cover the entire road length of the lane change maneuver.

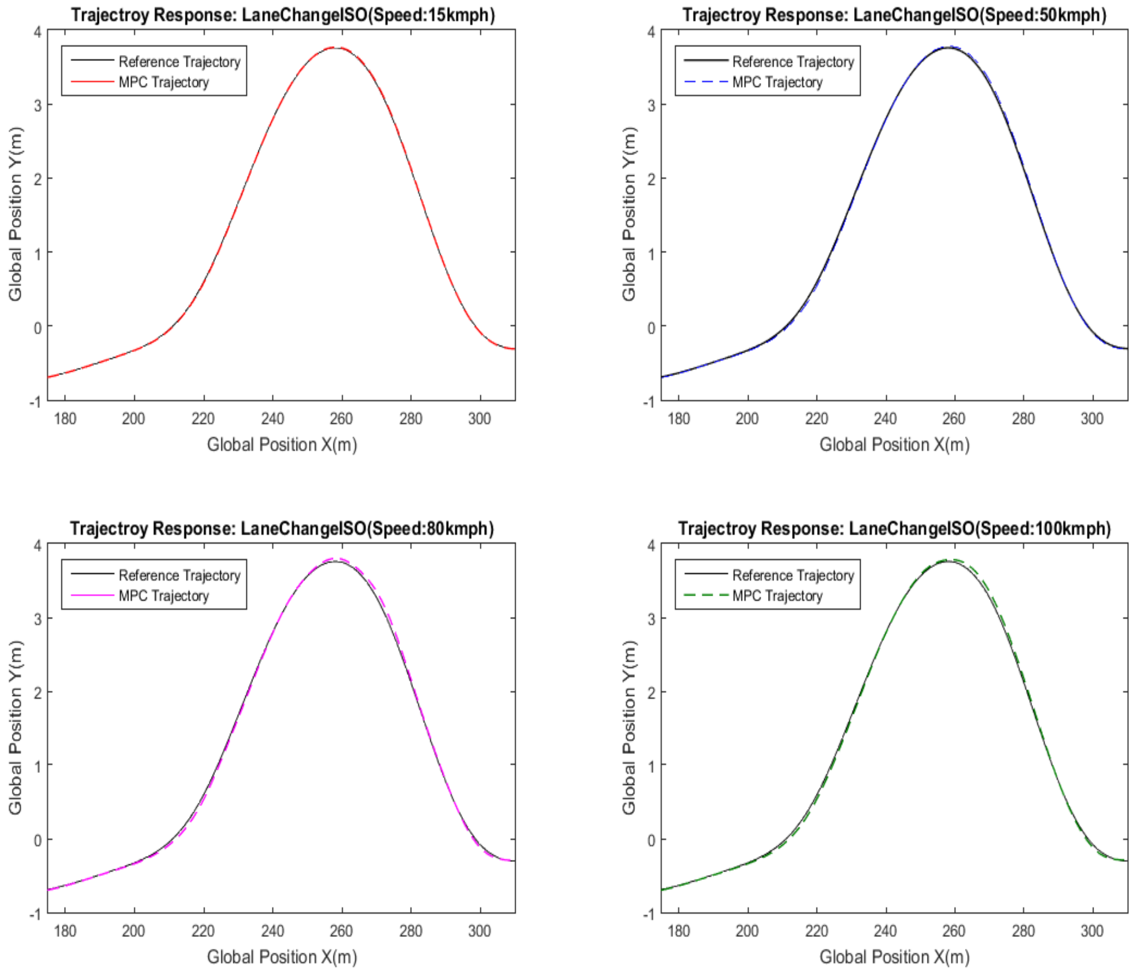


Figure 50: Trajectory Response of MPC with dynamic vehicle model for Lane change ISO Maneuver

For speed profile of 50km/h there is minimal deviation from the reference. For simulation with higher speed profiles the vehicle reaches the maximum velocity within 5 seconds from the starting of the simulation. As vehicle approaches the maneuver where it has to take a turn to make the lane change maneuver the controller provides brake torques to reduce the speed of the vehicle. Gradually the speed of the vehicle is reduced and again increased as soon as the lane change maneuver is completed. As we can see in Figure 51 the velocity error is gradually becoming zero, which depicts that the controller is not allowing the vehicle to attain the high reference speed which may make the vehicle to go out of lane. Maximum speed is attained when the vehicle is completed with lane changing maneuver.

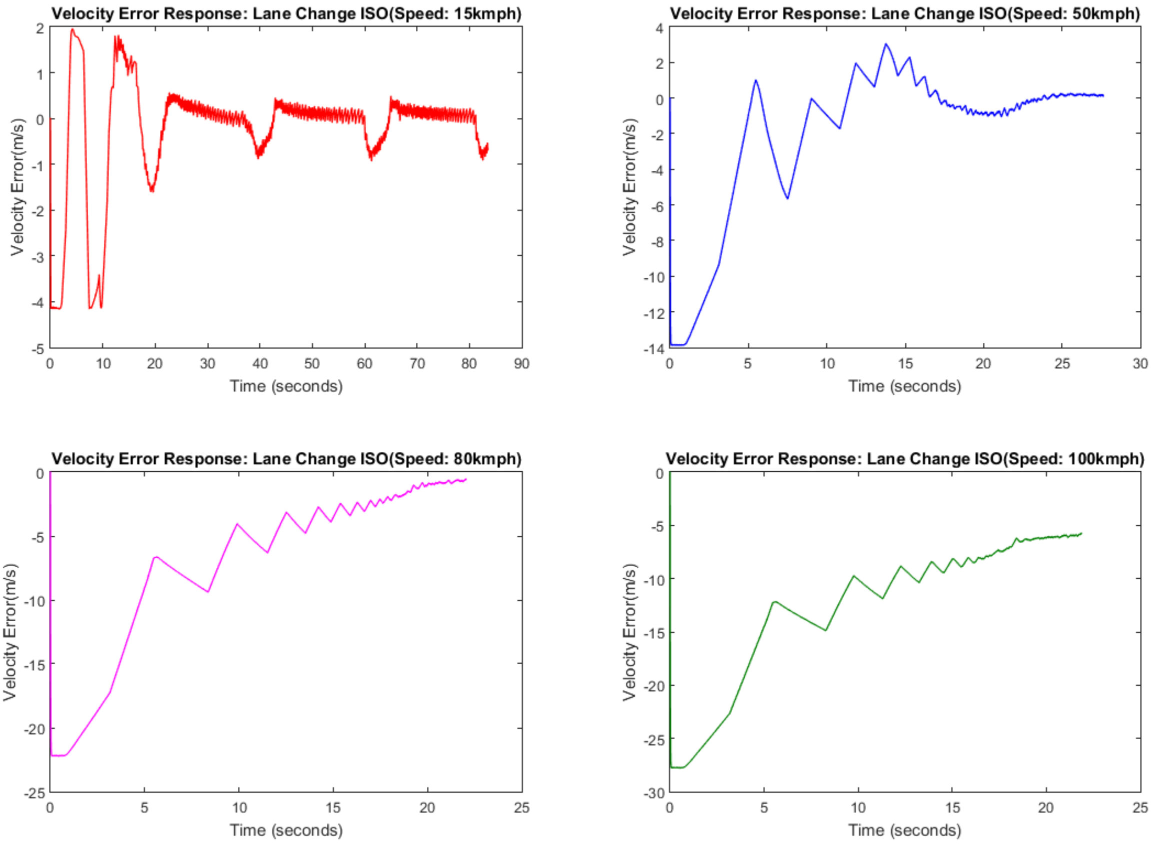


Figure 51: Velocity Error Response of MPC with dynamic vehicle model for Lane change ISO maneuver

Although the vehicle reference speed is high, controller tries to reduce the speed generating brake torques. This is due to the effect of the cost function weights that are used for the MPC design formulation. For this scenario more weights are apllied for cross track error than velocity error. Hence the controller makes predictions for feasilible control outputs to make the vehicle to have a trajectory resembling the reference trajectory. The maximum devaition in the position of the vehicle with respect to reference trajectory is observed to be 0.1m for speed profiles of 80kmph and 100kmph (Figure 52). The cross track error of 0.25m from 0-4s is because the simulation is not performed from the initial position of the maneuver map. The vehicle initial position is kept random to evlaute the performce of MPC if there is mismatch between the initial vehicle state and reference state. MPC make optimization depending on dynamic vehicle model and provides control signals of steering and brakes tries to reach target trajectory. After 5s the vehicle reaches closer to the target making the cross track error closer to zero. In Figure 53 we can observe that for vehicle speed of 80 and 100kmph the steering angle values are maximum during the lane changing behavior, which means the controller is trying to reach the maximum of constraints for controlling the vehicle trajectory during high speed maneuvers.

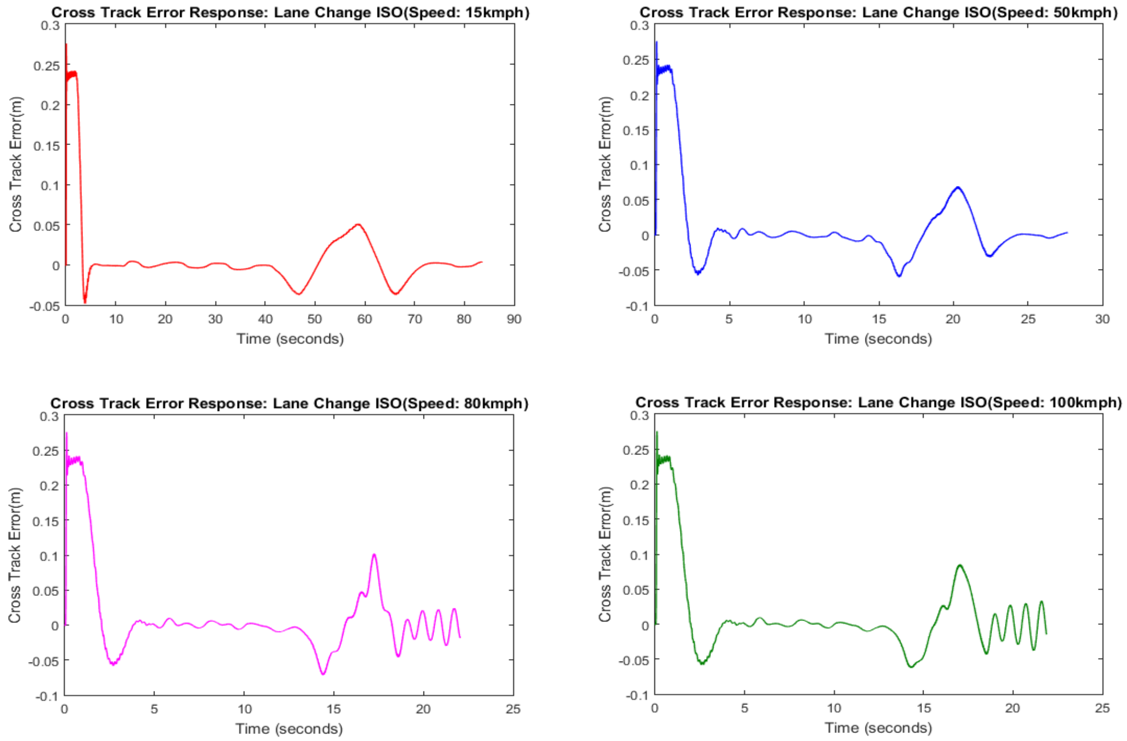


Figure 52: Cross Track Error response of MPC with dynamic vehicle model for Lane change ISO maneuver

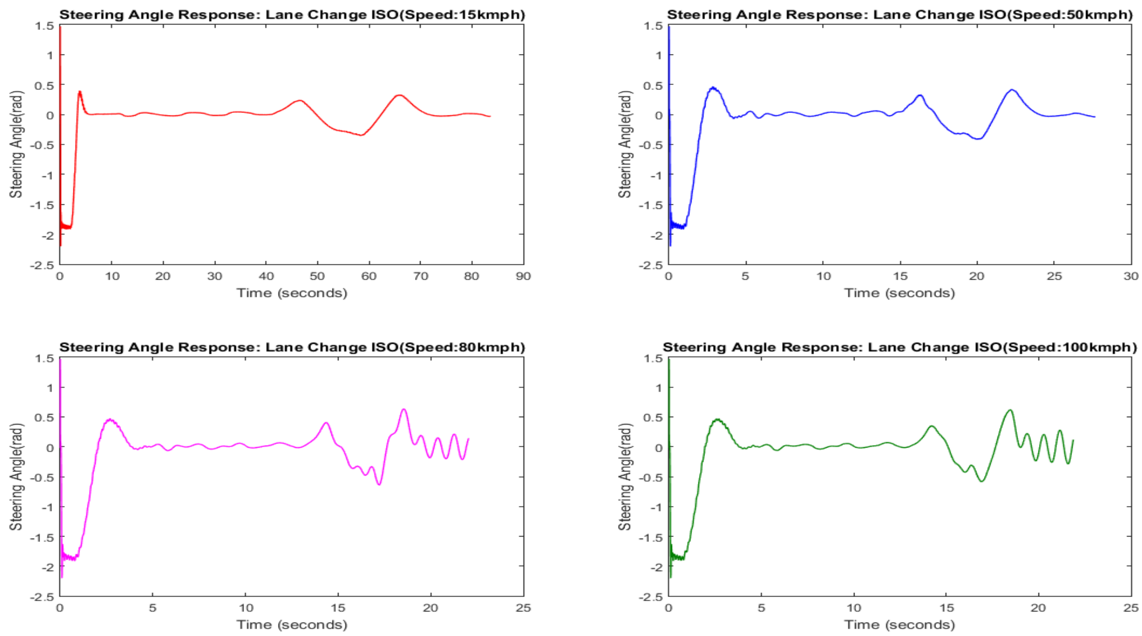


Figure 53: Steering Angle Response of MPC with dynamic vehicle model for Lane change ISO Maneuver

8.2.2 *MPC with kinematic model for Lane Change ISO maneuver*

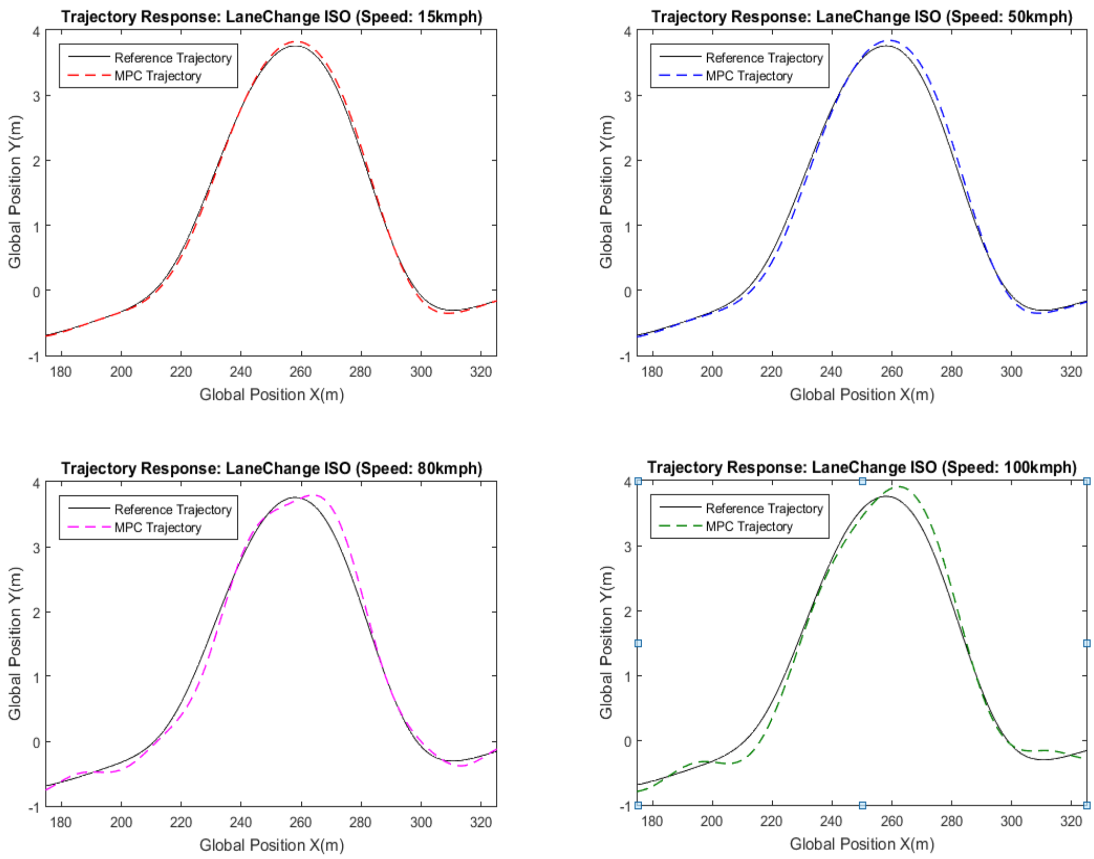


Figure 54: Trajectory Response of MPC with kinematic vehicle model for Lane Change ISO maneuver

The trajectory response of MPC with dynamic model is as shown in Figure 45. The steering angle limits used for the simulation is same as its used for MPC with dynamic vehicle model. The rate limits for steering angle is set to -15 deg to 15 deg. For kinematic model the controller outputs for controlling vehicle motion are steering angle and acceleration. The speed of the vehicle is mainly controlled by acceleration and deceleration. The acceleration limits are -5m/s^2 to -5m/s^2. From (Figure 54) it is observed that the performance of MPC for lower vehicle speed is better with some deviations in position compared to reference. As speed increases the deviations are more and for speed profile of 100kmph the vehicle is observed having maximum deviation. The acceleration signals by the MPC is used for speed control. For higher speed profiles with a waypoint gap of 5m the controller is incapable of making faster predictions and produces control signals.. From (Figure 54) We can understand that for higher speed profiles the longitudinal acceleration of the vehicle is changing in order to make the vehicle reach the target. The deviations are because at high speeds a little change in actuations of steering angle makes the vehicle deviate from the reference trajectory. Hence for high speed maneuvers the controller must perform more faster computations than for low speed maneuver. Also kinematic models do no include forces and torques for speed control and is only relying on acceleration signals from controller.

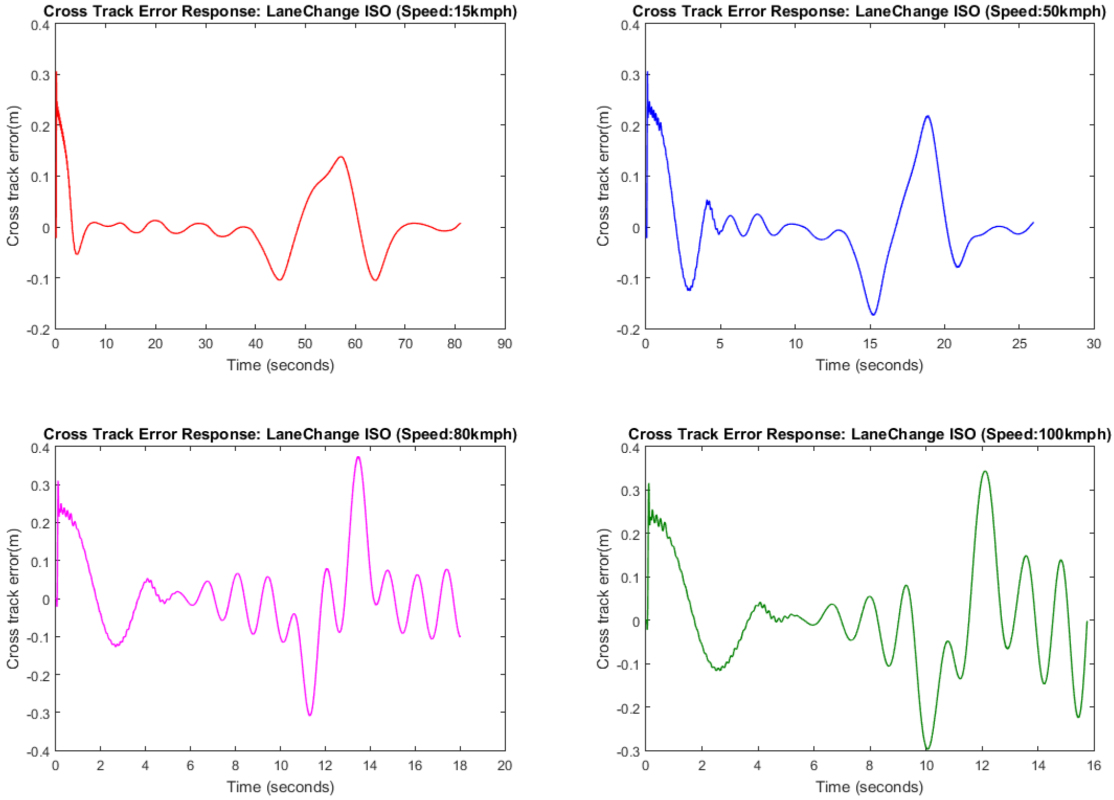


Figure 55: Cross Track Error Response for MPC with kinematic model for Lane Change ISO maneuver

The vehicle tends to have maximum cross track error of 0.4m for vehicle speed of 80kmph and 100kmph. For high speeds slight change in steering angle actuations makes the vehicle deviate from the reference trajectoy and hence the cross track error is more during cornering for high speed maneuvers. In Figure 56 we can observe that at 11s and 14s the steering angle reaches maximum during the lane changing behaviour of the vehicle. At this interval the controller tries to generate steering angle actuations reaching to the maximum limit of the constraints. Since the distance between the two lanes is small for low speed maneuver the waypoint gap of 5m is feasible. For high speed vehicle maneuvers the way point gap must be adjusted depending upon the vehicle speed. From Figure 57 we can observe that the lateral acceleration increases as speed of the vehicle increases. The fluctuations in the first few seconds of simulation is because of modelling inaccuracy between kinematic model and the vehicle model used in simulation environment. Kinematic model cannot represent the entire dynamics of the vehicle and hence there will be differences due to which the roll movements are generated which inturn is responsible for the lateral acceleration in the initial phase.

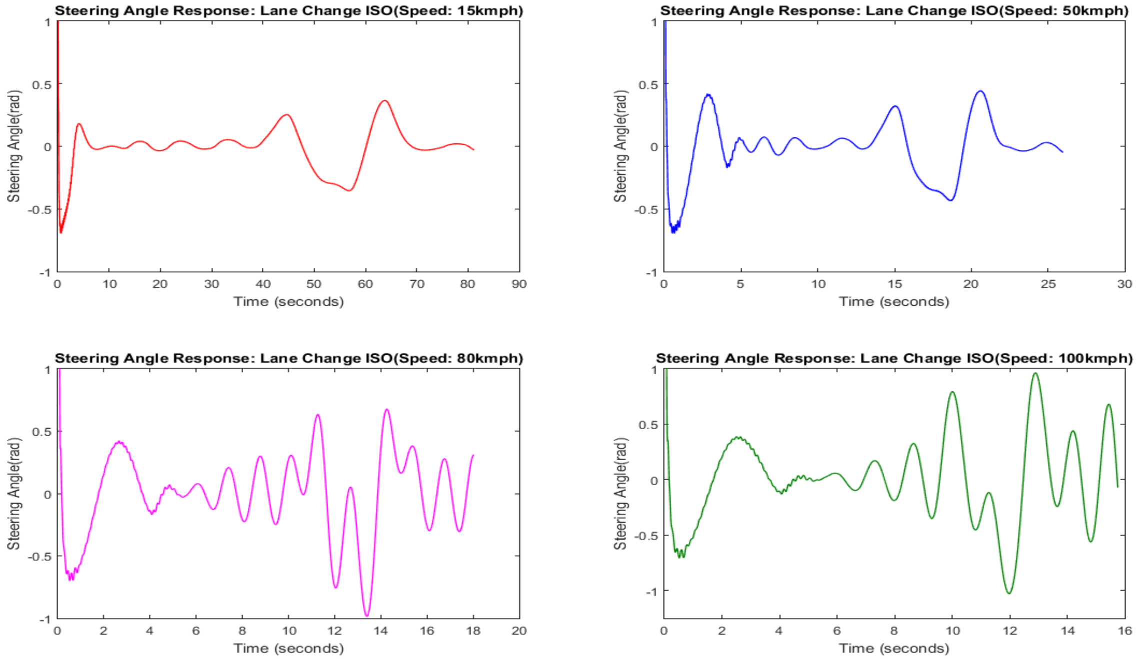


Figure 56: Steering Angle Response of MPC with kinematic vehicle model for Lane Change ISO maneuver

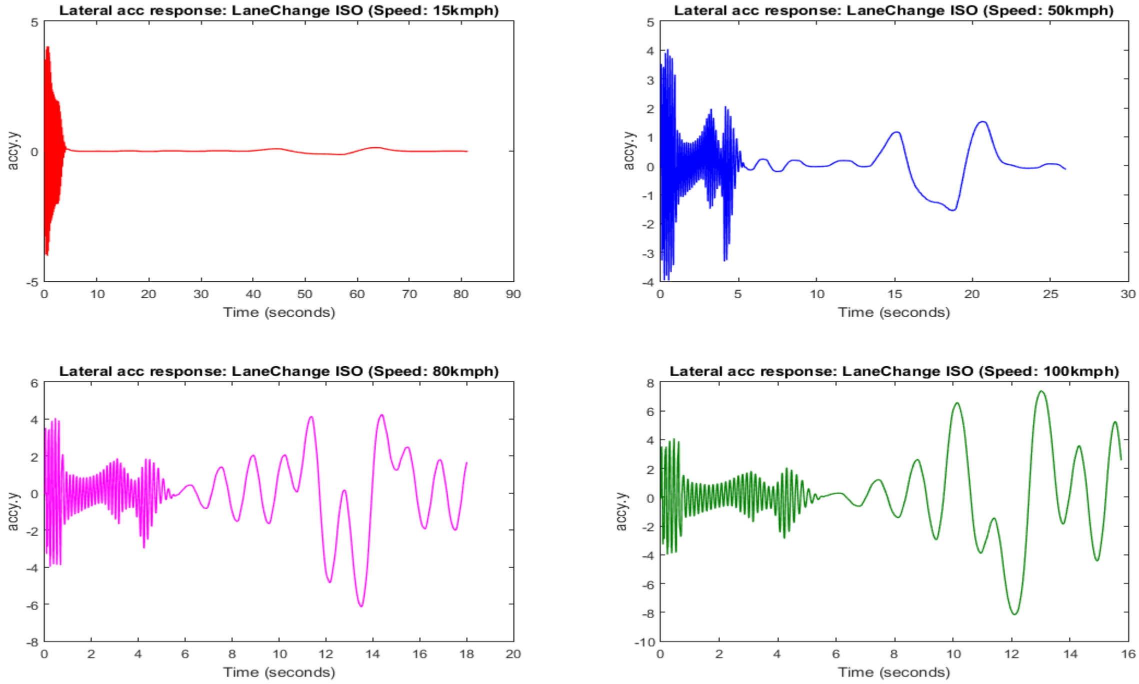


Figure 57: Lateral Acceleration Response of MPC with kinematic model for Lane Change ISO maneuver

8.3 Scenario 3: Circular Maneuver

The vehicle is driven with a constant speed in a steady circle of radius 50m.

8.3.1 *MPC performance for Circular maneuver with dynamic vehicle model*

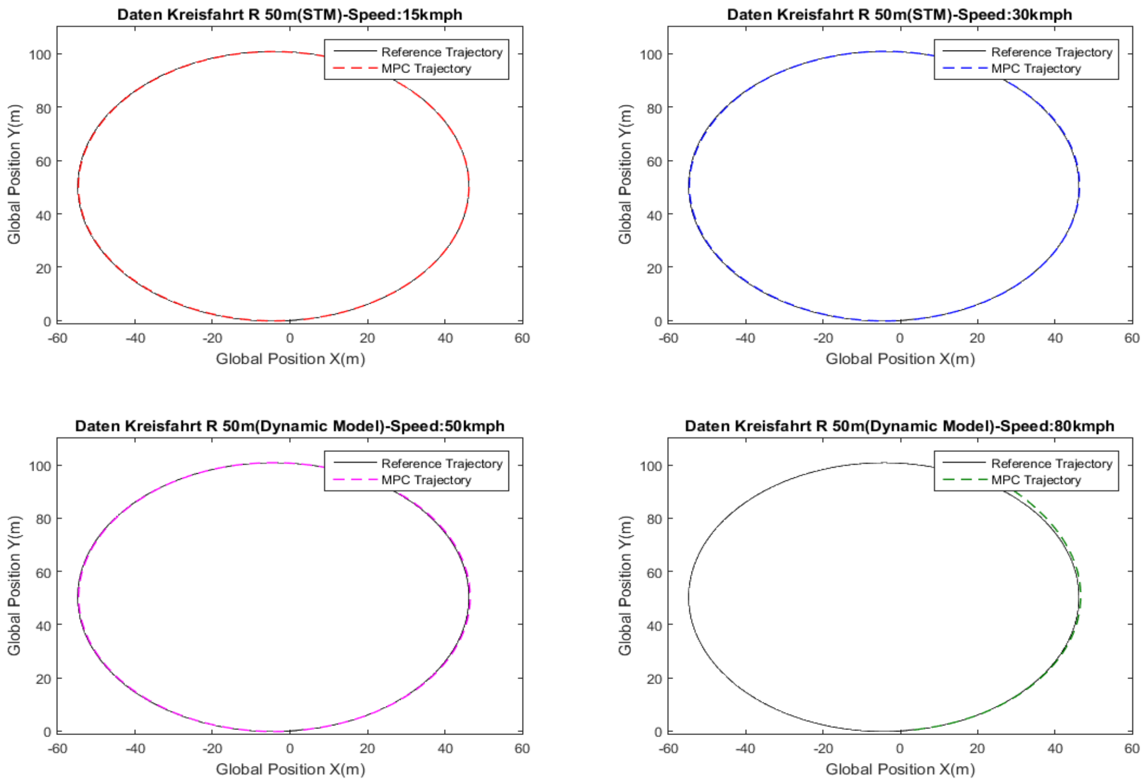


Figure 58: Trajectory response of MPC with dynamic vehicle model for Circe-50m maneuver

The trajectory response of MPC with dynamic model for circular maneuver is as shown in Figure 41. The steering angle limits used for the simulation is –60 degrees to 60 degrees. The rate limits for steering angle is set to -15 deg to 15 deg. The constraints for brake torques is -600Nm to 600Nm. The brake torques are applied to the vehicle wheels inorder to reduce the speed of vehicle to maintain the trajectory as close as possible to the reference trajectory.The simulations was performed at speeds of 15kmph , 30kmph, 50kmph and 80kmph. The overall simulation time is set to 80 seconds so that the vehicle will cover the entire circle. It is observed that at low speed profile the performance of MPC for the given set of constraints is accurate. For speed profile of 50kmph there is some deviation and for 80kmph the vehicle is observed to be going out of the track. This is because for this maneuver the maximum speed a vehicle could reach is 50kmph. Also for MPC with dynamic model the speed of vehicle is controller by brake torques. During circular behaviour at high speeds the torques applied makes the vehicle to change its orientation. The main reason for the vehicle going out of track is the way point gap of reference trajectory. For simulation we have considered the waypoint gap of 5m for all speed profiles. Reference waypoints with changing waypoint gap based on vehicle speed might give better results. In Figure 59 we can understand that the cross track error is more at the starting of simulation and is gradually decreased by the end of the simulation. This is because the initial conditions of vehicle and maneuver map and not kept same. The vehicle initial position is set with random values and hence the controller tries to optimize the contol signals to reach the reference trajectory. MPC controller makes better predictions once the vehicle is closer to reference.

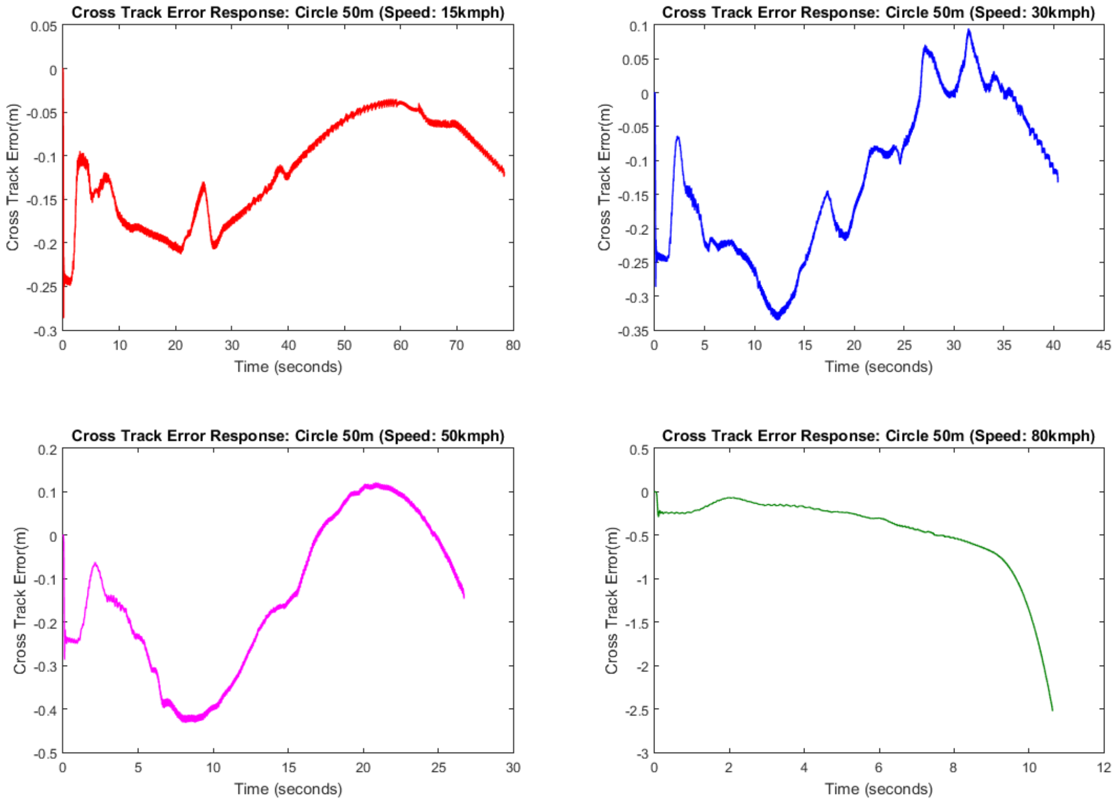


Figure 59: Cross track error response of MPC with dynamic model for Circle-50m maneuver

From the velocity repsonse (Figure 60) it is observed that for vehicle speed of 15kmph the velocity error, i.e the difference between the reference velocity and vehicle velocity is mostly around 1m/s to -2m/s. It means that the velocity of the vehicle is slightly increased and decreased than the reference velocity during simulation so that the trajectory response is accurate. Similarly for vehicle speed of 50kmph the vehicle tends to reach reference velocity in 6s and gradually decreases until 14s inorder to reduce the vehicle deviation. The steeirng angle values are observed to be within the constraint limits for speed profile of 50kmph. For 80kmph the brakes torques applied on wheels make the vehicle to have change in orientation. Also the due to the steering rate limits of 15degrees the controller could not exceed the constrainsts making the vehicle move out of the track. From Figure 60 it is observed that for speed profile of 15kmph the vehicle tries to reach reference velocity at 5s and tend to decrease the velocity as it approaches the lane changing maneuver. It increases the velocity at 10s when the vehicle is going away from the reference trajectory and it reduces the speed at 20s to make the vehicle follow the reference trajectory. For speed profile of 80kmph the velocity error is gradually becoming zero depicting that the veclocity of the vehicle is not reduced to follow the reference.

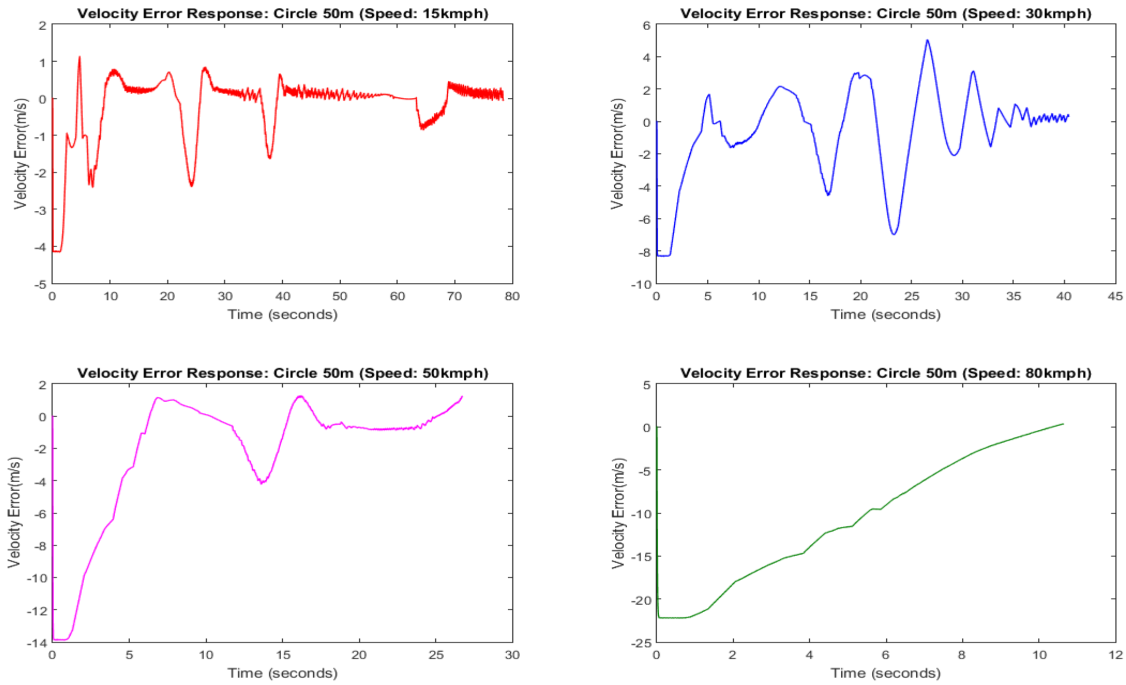


Figure 60: Velocity Error Response of MPC with dynamic vehicle model for Circle-50m maneuver

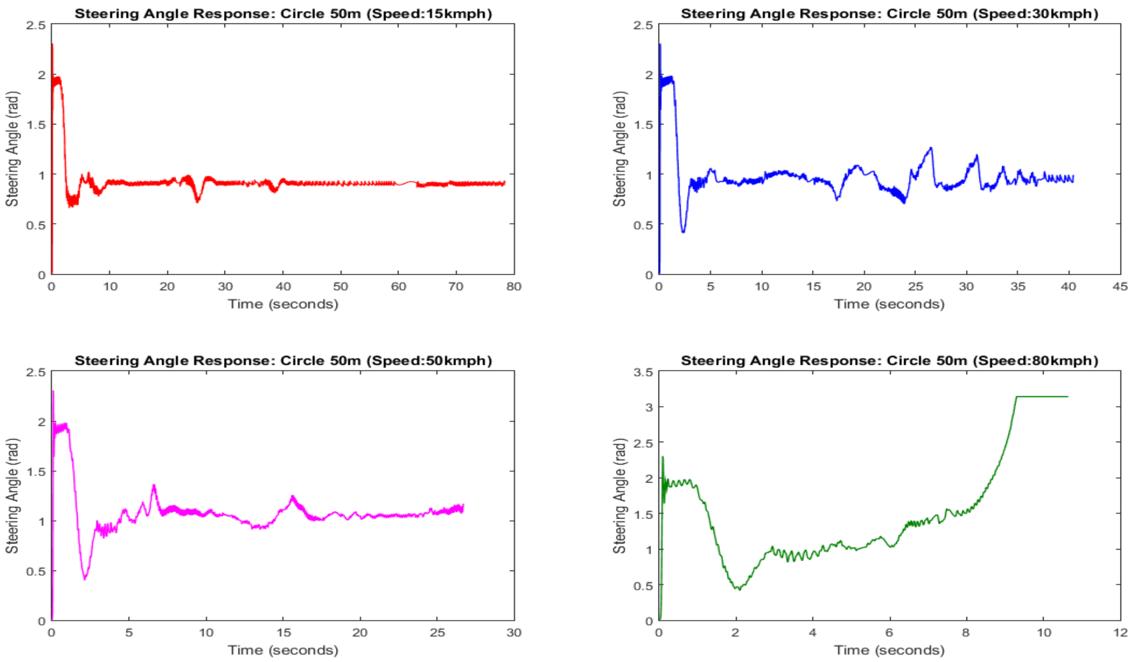


Figure 61: Steering Angle Response of MPC with Kinematic model for Circle-50m maneuver

8.3.2 *MPC performance for Circular maneuver with kinematic vehicle model*

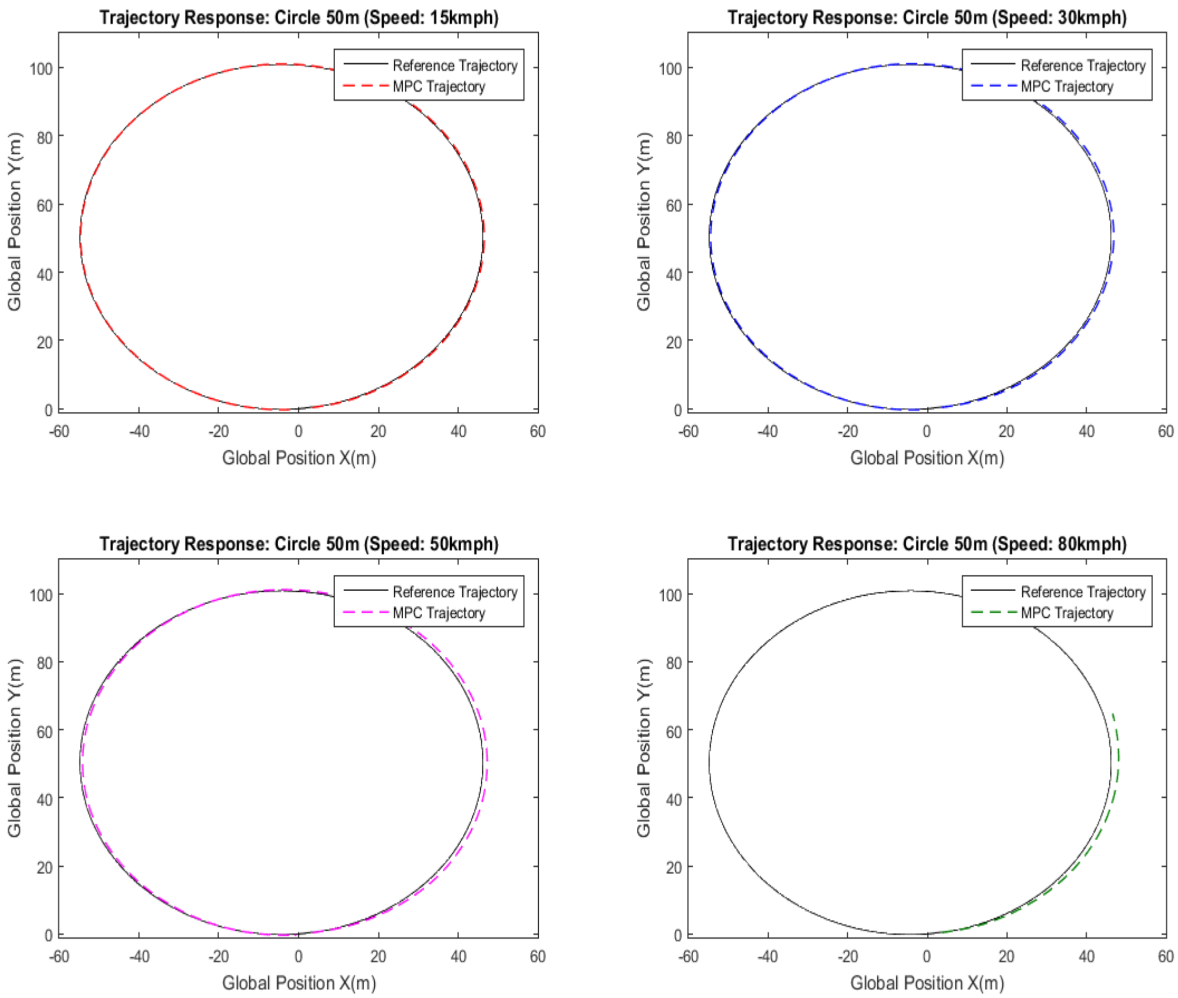


Figure 62: Trajectory Response of MPC with kinematic vehicle model for Circle-50m maneuver

In Figure 62 the results for the performance of MPC with kinematic vehicle model for circular maneuver is presented. It is observed that the vehicle tends to go out of track for vehicle speed of 80kmph.. As we increase the vehicle speed more deviations are observed. The steering angle limits used for simulation is -60 degrees to 60 degrees. The rate limits for steering angle is set to -15 deg to 15 deg. The acceleration limits are -5m/s^2 to -5m/s^2. The maximum cross track error for speed of 15kmph is 0.55m and as vehicle speed is increased i.e for 30kmph maximum value of Cte is 0.8m and 1.25m for 50kmph (Figure 63). The deviation until 15s is due to vehicle initial position at the start of simulation. Due to steering angle constraints and target waypoints gap the controller fails to optimize control signal faster for speed of 80kmph and hence the vehicle goes out of track. Comparing Figure 62 and Figure 58 it is observed that MPC with dynamic vehicle model have better performance than kinematic model. But for both models the steering angle constraints has to be increased to keep vehicle on track for high speed maneuvers.

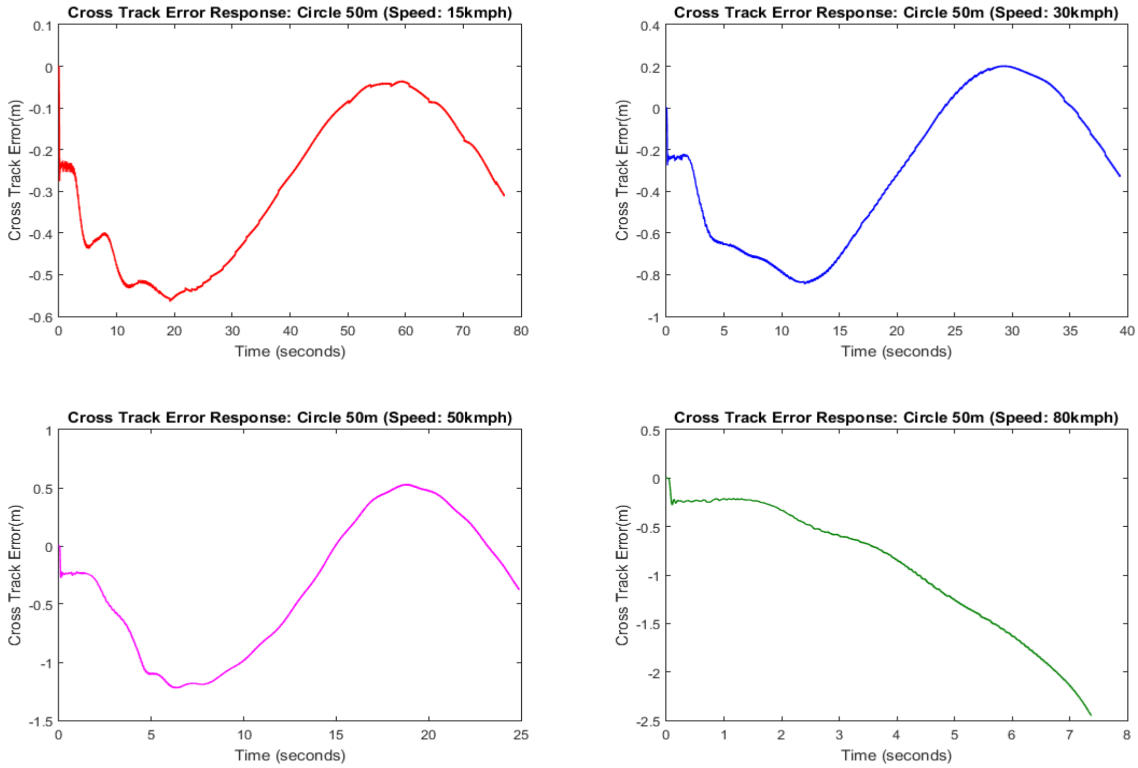


Figure 63: Cross Track Error Response of MPC with kinematic model for Circle-50m maneuver

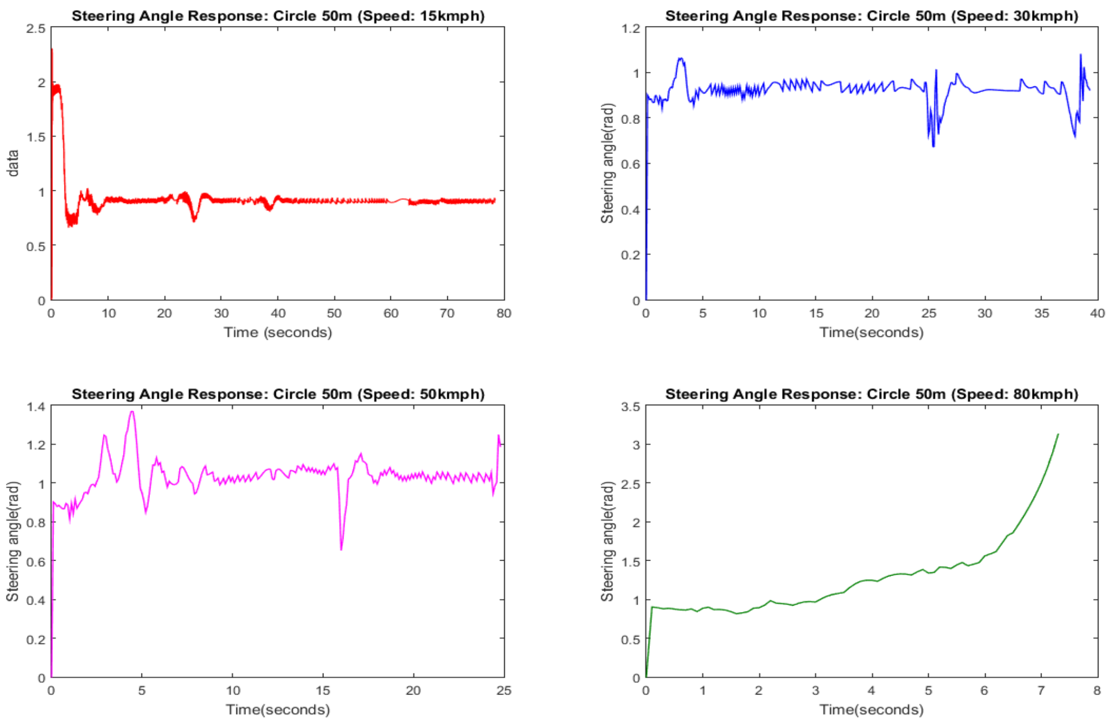


Figure 64: Steering Angle Response of MPC with kinematic model for Circle-50m maneuver

8.4 Comparison of MPC with PIP Controller

The performance of MPC is evaluated comparing with PIP (Proportional Integral Plus). The simulations are tested for Slalom-36m and Lane Change maneuvers

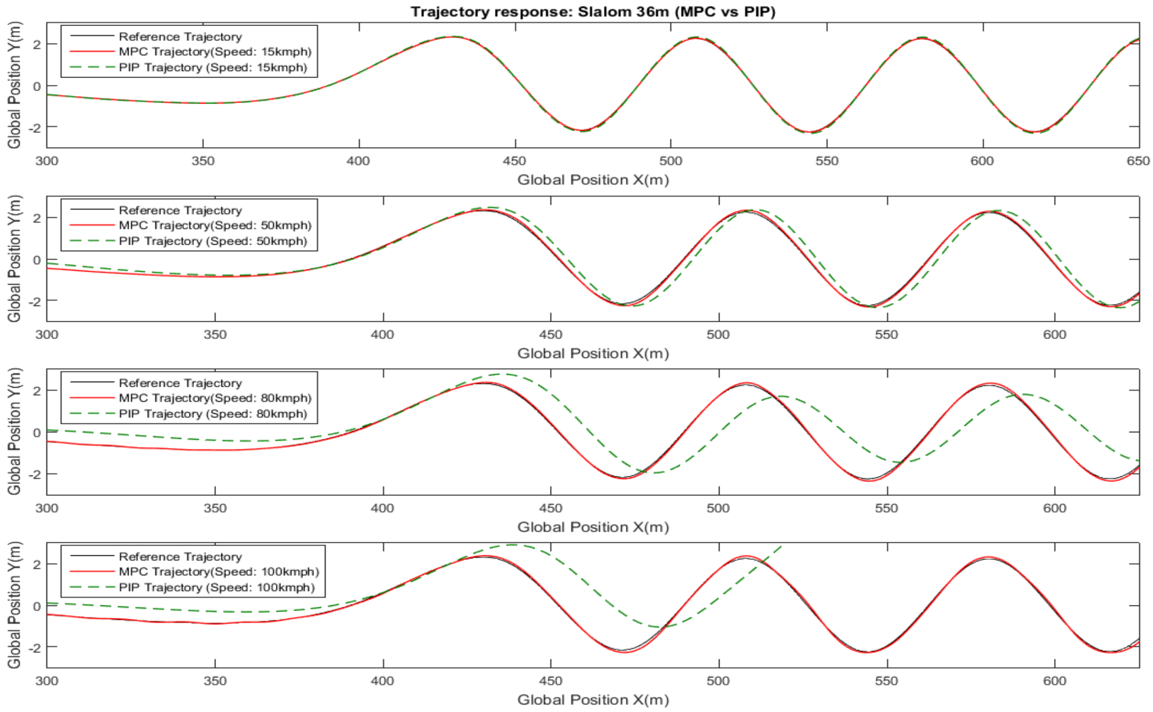


Figure 65: MPC vs PIP for Slalom-36m maneuver

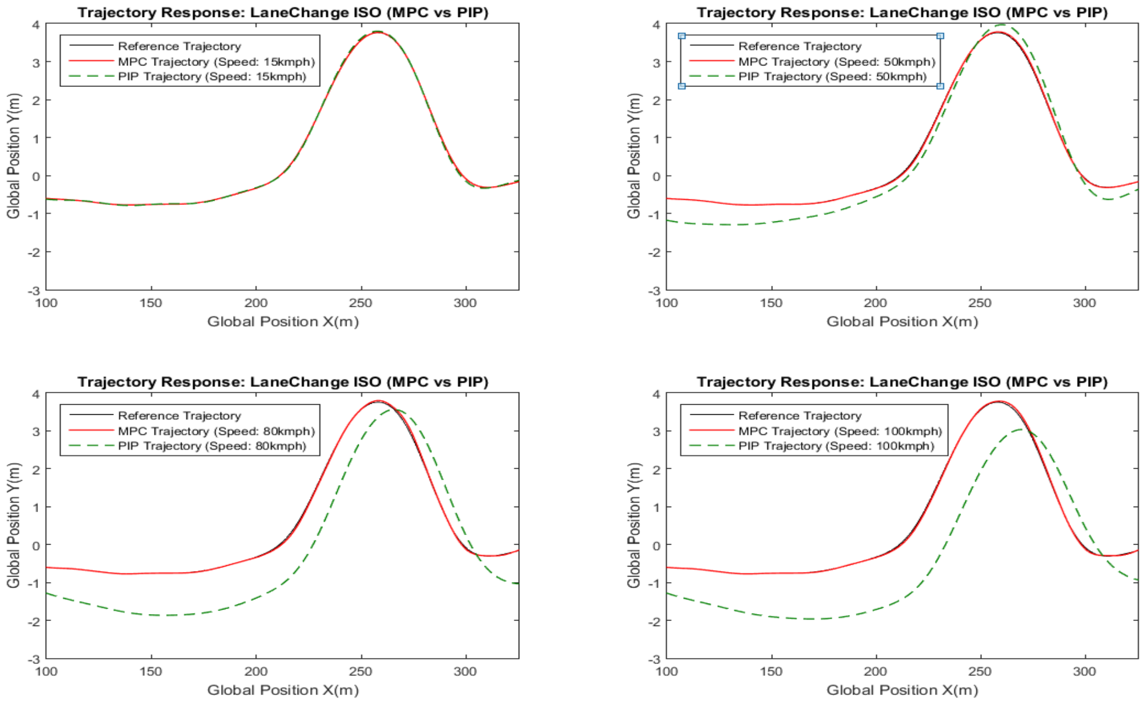


Figure 66: MPC vs PIP for Lane Change ISO maneuver

From Figure 65 and Figure 66 we can notice that for maneuver with 15kmph speed both the controllers are performing better. For high speed maneuvers performance of MPC is better than PIP controller. This is because MPC performs optimization based on constraints and vehicle models. Where as PIP controller is a traditional control approach where the controller tries to reach the refrence trajectory by computing the error between the current and previous vehicle states. For the simulation performed MPC is replaced with PIP controller for evluation. The longitudinal motion is controlled by brake torques in MPC. This is due to the advantage of using vehicle model implicitly with MPC. The vehicle models include the lateral and longitudinal dynamics of the vehicle by which the optimizations are performed making computations at each sampling instance. Also the implicit handing ability of contraints for MPC gives an advantage of making predicitons and then choose optimal control signals. With PIP controller the contraints must be handled explicitly which makes the controller performance inefficient. Seperate PIP controllers must be used for controlling lateral and longitudinal motion of the vehicle. Tuning the controller parameters is difficult if seperate controllers are used for controlling vehicle motion. PIP controller consider the state variables and tries to reach reference state using an explicit model to calculate the errors. A seperate model is developed for PIP controller for calculating cross track error, heading error and velocity error. The predicting ability of MPC for future vehicle states and control outputs optimizing the costfuntion including the actuator constraints makes MPC perform better than PIP controller. The dotted line in Figure 65 and Figure 66 represents the vehicle trajectories generated from the PIP controller. The deviation from the reference is huge when compared to MPC. The optimization of control signals is carried out at each control interval predicting optimal control values by MPC. The error deviation from PIP is comparitively bigger for high speed maneuvers than low speed profiles. This is because the longitudinal controller handling longitudinal motion of vehicle is not capable of altering the speed of vehicle to reach the target trajectory. The accuracy in prediction of errors in MPC is better than PIP controller. For better performance of PIP controller it is required to include contraints during the planning phase since there is no inbuit capability to handle constraints unlike MPC. From the result plots it is evident that MPC controller outperforms PIP controller for the selected maneuvers for simulation.

9 Conclusion

This thesis highlights Model Predictive Control for autonomous vehicle applications. We have studied the autonomous vehicle architecture for perception, planning and control. The control approaches for vehicle motion control and motion planning techniques are briefly discussed in the report. We described about the planning architecture and approaches for motion planning. In this thesis, MPC for autonomous vehicle trajectory control is designed and developed that predicts the future trajectories of the vehicle depending on target trajectory and actuator constraints.

First, we presented the modelling of vehicle models describing lateral and longitudinal dynamics of the vehicle that has to be used for control design. The vehicle model used for simuation is a bicycle model, where four wheels of a vehicle are lumped into two wheel model. The modelling is done using MATLAB/Simulink and tested in CarMaker simulation environment. The performance of the controller is analysed and evaluated for Slalom, lane change and circle maneuvers. The road profiles are created using CarMaker for Simulink simulation tool. The controller developed is a linear MPC since the costfuntion and constraints are linear. The nonlinear vehicle models are linearized into state space form using MPC designer tool available in Simulink. The implemented controller has constraints for steering angle, acceleration and torque control signals that is used for controlling the vehicle motion depending on the vehicle model used for MPC. For cost function formulation the errors considered are cross track error, heading angle error and velocity error. After linearizing the model equations, constraints and cost function the optimization is done by quadratic programming method. The optimizer used for controller is QP solver which is the default solver available with the MPC toolbox for Simulink. Map based trajectory planning is adopted for generating waypoints as a reference for controller. The errors related to cost function have significant role in controlling vehicle trajectory. More weights are used for cross track error and velocity error since we are interested in controlling the trajectory motion of vehicle.

The trajectory control system is developed in Simulink and the performance of MPC was evaluated for both kinematic and dynamic models. From the results its is observed that MPC with dynamic models perform better reducing the error deviation because dynamic model is more detailed model including both lateral anf longitudinal dynamics. Whereas kinematic models consider only the geometry of vehicle for motion control. The simulation is tested for different speed profiles for high dynamic maneuvers such as Slalom, Lane Change and Circle maneuvers. It is observed that for high speed maneuvers the trajectory response by MPC has slight deviations. The reason for this is the waypoint gap of the reference trajectories for the controller. For low speed maneuvers less waypoint gap is sufficient, as speed is increased the way point gap has to be adopted depending on vehicle speed. Additionally constraint limits has significant influence on controller performance. Higher the constraint limits better the performance but there is also a possibility of vehicle going out of the track for higher speed profiles. Hence determining the constraint and its rate limits is important for designing the controller. The performance of MPC was evalauted with PIP controller. It is observed that MPC outperforms PIP controller for trajectory control due to its ability of predicting future states and implicit handling of constraints.

9.1 Future Scope

The results can be improved by developing a waypoint generating algorithm capable of adopting to different speed profiles. The MPC designed is a linear MPC using Simulink toolbox. The state space modeling is done by the designer whereas linearizing the model equations manually could improve the performance of controller. Using third party solvers instead of QP solver can have better computational performance. The extention of the thesis could be to test the controller performance in a real vehicle. The vehicle models developed are bicycle model ignoring the tire dynamics. Developing a more detailed four wheel vehicle model including non linear tire dynamics gives more accuracy. It would be interesting to include more state variables and other error factors to cost function for evaluation. A better way to develop a trajectory control system using MPC could be to use the predictive capability of MPC for planning trajectories so that the vehicle model is considered during the planning phase. In order to implement such a system it requires two MPC controllers for planning and control of trajectory, which increases the computations and its is necesarry to use better solver to handle the increase in computations. One disadvantage of using MPC toolbox in simulink is the constraints has to be initialized before starting the simulation. The constraints cannot be modified during simulation. Implementing a constraint handling interface will give an option to evaluate controller performance for varying constraints. The performance of different MPC control architectures with other optimizers can be used for trajectory control of an autonomous vehicle. The tuning parameters and weights for costfunction was done by trail and error method. Implementing a systematic approach for tuning would improve the controller performance.

Bibliography

[1] Y. Gao, „Model Predictive Control for Autonomous and Semiautonomous Vehicles,“ University of California, Berkely, 2014.

[2] M. Nolte, M. Rose, T. Stolte und M. Maurer, „Model Predictive Control Based Trajectory Generation for Autonomous Vehicles – An Architectural Approach,“ *IEEE,* 2017.

[3] B. Paden, M. Cáp, S. Z. Yong, D. Yershov und E. Frazzoli, „A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles,“ 2016.

[4] J. I. Guzman, C. Laugier, J. D. Yoder und S. Thrun, „Autonomous Driving: Context and State-of-the-Art,“ *Research Gate,* 2012.

[5] D. González, J. Pérez , V. Milanés und F. Nashashibi, „A review of motion planning techniques for automated vehicles,“ *IEEE Transactions on Intelligent Transportation Systems,* 2016.

[6] S. D. Pendleton, H. Andersen, X. Du, X. Shen und M. Meghjani, „Perception, Planning, Control, and Coordination for Autonomous Vehicles,“ 2017.

[7] Q. Wang, „Predictive Maneuver Planning and Control of an Autonomous Vehicle in Multi-Vehicle Traffic with Observation Uncertainty,“ TigerPrints, 2017.

[8] C. Katrakazas, M. Quddus, W.-H. Chen und L. Deka, „Real-time motion planning methods for autonomous on-road driving state-of-the-art and future research directions,“ *ScienceDirect,* 2015.

[9] W. Schwarting, J. A. Mora und D. Rus, „Planning and Decision-Making for Autonomous Vehicles,“ 2018.

[10] . C. E. B. Alba und C. , Model Predictive Control, Springer, 2007.

[11] M. Behrendt, „Model predictive control,“ Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Model_predictive_control. [Zugriff am 1 10 2018].

[12] M. Hovd, A brief introduction to Model Predictive Control, 2004.

[13] F. P, B. F, T. H, A. J und H. D, „A Hierarchical Model Predictive Control Framework for Autonomous Ground Vehicles,“ *IEEE,* 2008.

[14] Y. S. Kang, „Development of Predictive Vehicle Control System using Driving Environment Data for Autonomous Vehicles and Advanced Driver Assistance Systems,“ Virginia Polytechnic Institute, Blacksburg, 2018.

[15] Y. Rasekhipour, „Prioritized Obstacle Avoidance in Motion Planning of Autonomous Vehicles,“ University of Waterloo, 2017.

[16] K. Zhang, J. Sprinkle und R. G. Sanfelice, „Computationally aware control of autonomous vehicles : a hybrid model predictive approach,“ *Autonomous Robots.*

[17] J. Kong, M. Pfeiffer, G. Schildbach und F. Borelli, „Kinematic and Dynamic Vehicle Models for Autonomous Driving Control design,“ *IEEE,* 2015.

[18] R. Rajamani, Vehicle Dynamics and Control, Springer, 2012.

[19] B. J. C. G. M. W. und A. R. V. P. , „Driver Behavior Comparison Between Static and Dynamic Simulation for Advanced Driving Maneuvers,“ *IEEE,* 2011.

[20] M. Elbanhawi, M. Simic und R. Jazar, „Investigating Ride Comfort Measures in Autonomous Cars,“ *IEEE ,* 2015.

[21] D. D. Ruscio, „MODEL PREDICTIVE CONTROL and optimization,“ Telemark University, 2001.

[22] „ipg-automotive,“ [Online]. Available: https://ipg-automotive.com/products-services/simulation-software/carmaker/..

[23] „Vehi.co,“ Vehico, [Online]. Available: http://www.vehico.de/index.php/en/applications/iso-lane-change-test. [Zugriff am 30 11 2018].

[24] G. C. PEREIRA, „Model Predictive Control for Autonomous Driving of Over-Actuated Research Vehicle,“ KTH ROYAL INSTITUTE OF TECHNOLOGY, Stockholm, 2016.

[25] J. Fredlund, „Autonomous driving using Model Predictive Control methods,“ Lund niversity, Sweden, 2017.

[26] D. Gonzalez, „Functional architecture for automated vehicles trajectory planning in complex environments,“ HAL, 2017.

[27] Y. Kang, „Development of Predictive Vehicle Control System using Driving Environment Data for Autonomous Vehicles and Advanced Driver Assistance Systems,“ Virginia Polytechnic Institute, Blacksburg, 2018.

[28] F. K. und B. , „MPC-Based Approach to Active Steering for Autonomous Vehicle Systems,“ *Research Gate,* 2005.

[29] L. Aliaga, „Reliable Autonomous Vehicle Control – a Chance Constrained Stochastic MPC Approach,“ Technische Universität Ilmenau, 2017.

[30] F. Curinga, „Autonomous racing using model predictive control,“ KTH ROYAL INSTITUTE OF TECHNOLOGY, Stockholm, 2018.

[31] M. Abbas, „Non-Linear MPC for Autonomous Vehicles,“ University of Ontario Institute Of Technology, 2011.

[32] X. Qian, „Model predictive control for autonomous and cooperative driving," HAL, 2017.

[33] Y. Goa und T. Lin, „PREDICTIVE CONTROL OF AUTONOMOUS GROUND VEHICLES WITH OBSTACLE AVOIDANCE ON SLIPPERY ROADS," *SemanticScholar,* 2010.

[34] A. Best, N. Sahil und D. Manocha, „Autonomous Vehicle Planning with Dynamic Maneuvers and Traffic Constraints," *IEEE,* 2017.

[35] E. Alcala, L. Sellart, V. Puig, S. Jordi und A. Lopez, „Comparison of two non-linear model-based control strategies for autonomous vehicles," *IEEE,* 2017.

[36] D. Calzolari, B. Schurmann und M. Althoff, „Comparison of Trajectory Tracking Controllers for Autonomous Vehicles," *IEEE,* 2017.

[37] K. Isa und A. jantan, „An Autonomous Vehicle Driving Control System," *ResearchGate,* 2005.

[38] L. G. und F. , „Motion Control of an Autonomous Vehicle Through Sensor Guided Manoeuvers," *ResearchGate,* 2015.

[39] W. Schwarting, D. Rus und J. A. Mora, „Planning and Decision-Making for Autonomous Vehicles," *Annual Reviews,* 2018.

[40] M. Hovd, „A brief introduction to Model Predictive Control," NTNU, 2004.

[41] F. Yakub, „Model Predictive Control for Car Vehicle Dynamics System – Comparative Study," ResearchGate, 2013.

[42] V. Turri, A. Carlvalho, H. Tseng und K. JOhannsson, „Linear Model Predictive Control for Lane Keeping and Obstacle Avoidance on Low Curvature Roads," *IEEE,* 2013.

[43] M. Maximo, „MODEL PREDICTIVE CONTROLLER FOR TRAJETORY TRACKING BY DIFFERENTIAL DRIVE ROBOT WITH ACTUATION CONSTRAINTS," *Semantic Scholar,* 2015.

[44] V. Putta, G. Zhu, D. Kim und J. Hu, „Comparative Evaluation of Model Predictive Control Strategies for Building HVAC System," *IEEE,* 2013.

[45] X. Zhang, „Fast MPC Solvers for Systems with Hard Real-Time Constraints," Delft University of Technology, 2016.

[46] T. M, T. H, F. P und B. F, „A Linear Time Varying Model Predictive Control Approach to the Integrated Vehicle Dynamics Control Problem in Autonomous Systems," IEEE, 2007.

[47] Y. Goa, „Model Predictive Control for Autonomous and Semiautonomous Vehicles," University of California, Berkeley, 2014.

[48] P. Lima, „Predictive control for autonomous driving with experimental evaluation on a heavy-duty construction truck,“ KTH Royal Insitute of Technology, Stockholm, 2016.

[49] S. Y. E, „Model-based Path Planning and Control for Autonomous Vehicles using Artificial Potential Fields,“ Delft University of Technology, 2018.

[50] L. Pori, „COMPARISON OF TWO INTERIOR POINT SOLVERS IN MODEL PREDICTIVE CONTROL OPTIMIZATION,“ Aalto University of Technology, 2016.

[51] D. Bautista, „Functional architecture for automated vehicles trajectory planning in complex environments,“ HAL, 2017.

[52] M. Palmqvist, „Model Predictive Control for Autonomous Driving of a Truck,“ KTH Royal Institute of Technology, 2016.

Appendix

1. Vehicle Model Parameters

Parameter	**Value**
Mass (m)	1484 kg
Inertia (I)	2223 kgm^2
Track front (lf)	1.548 m
Track Rear (lr)	1.566 m
Tire Radius (r)	0.31 m
Cornering Stiffness front	190000
Cornering Stiffness Rear	150000

2. Controller Parameters for Slalom Maneuver

Parameter	**Value**
Prediction Horizon	10steps
Control Horizon	2 Steps
Sampling Time	0.01s
Steering Angle Limit	-60deg to 60deg
Steering Angle Rate Limits	-15deg -15deg
Acceleration Limit	$-3m/s^2$ to $3m/s^2$
Torque Limits	-600 to 600
Cross Track Error Weights	10000
Course angle Error Weights	100
Velocity Error	3600

3. Controller parameters for Lane Change ISO Maneuver

Parameter	Value
Prediction Horizon	10steps
Control Horizon	2 Steps
Sampling Time	0.01s
Steering Angle Limit	-45deg to 45deg
Steering Angle Rate Limits	-15deg to 15deg
Acceleration Limit	$-3m/s^2$ to $3m/s^2$
Torque Limits	-600 to 600
Cross Track Error Weights	1000
Course angle Error Weights	100
Velocity Error Weights	700

4. Controller Parameters for Circle-50m Maneuver

Parameter	Value
Prediction Horizon	10steps
Control Horizon	2 Steps
Sampling Time	0.01s
Steering Angle Limit	-60deg to 60deg
Steering Angle Rate Limits	-15deg -15deg
Acceleration Limit	$-3m/s^2$ to $3m/s^2$
Torque Limits	-600 to 600
Cross Track Error Weights	1000
Course angle Error Weights	100
Velocity Error Weights	400

Selbstständigkeitserklärung

TECHNISCHE UNIVERSITÄT CHEMNITZ

Studentenservice – Zentrales Prüfungsamt
Selbstständigkeitserklärung

Name: BOLLINENI	**Bitte beachten:**
Vorname: TARUN	1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.
geb. am: 30.09.1992	
Matr.-Nr.: 421968	

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum:

Unterschrift: ....

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.