POLITECNICO DI TORINO

DIMEAS - DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

Master of science degree
in
Automotive Engineering

Master's degree thesis

Path planning with Rapidly-exploring Random Tree for autonomous race car

Supervisors:
Prof. Andrea Tonoli
Prof. Nicola Amati
Dr. Angelo Bonfitto

Candidate:
Karthikeyan Manohar
240393

Academic Year 2019-2020

Abstract

Autonomous vehicle research and development is emerging in recent years globally aimed to provide a highly reliable, secure, mobile, and intelligent transportation system. The procedure involves integration of multiple systems together to ensure safe driving. The functional reference standard architecture of the autonomous driving system is classified into three main categories: perception, decision & control, and actuation. Mainly the challenging part is decision making that requires high-level planning and vehicle control to accomplish the driving mission. This thesis work is a small contribution to design and develop an autonomous formula student racing vehicle, and it mainly focuses on the study of decision making strategies and vehicle control system. The experimentation aimed at the performance evaluation of a model-based control system and defining motion planning strategies for better vehicle navigation. In the first part of the thesis, a previously designed Model predictive based control system for lane following is validated on three different race track scenarios to evaluate the performance of model. In the second part of the thesis, development of vehicle motion planning and tracking control system using a model based algorithm to provide a highly safe navigation path to follow. For experimentation purpose a simplified vehicle model is used to validate the tracking control system, and suitable improvements are made based on Simulation results to reduce high-level tracking errors considering handling and safety limits. MATLAB® and Simulink® are the modern simulation tools that are used in this project work to perform all the analysis.

Acknowledgements

I wish to extend my sincere gratitude to Prof. Nicola Amati for creating an opportunity to pursue a high level of learning, to overcome problems, and to experience the research work culture.

I would like to thank Prof. Angelo Bonfitto for his kind support, and valuable suggestion.

My sincere thanks to Mr Sailesh for directing me in the right direction to finishing the works efficiently and correctly.

I am thankful to have kind, generous, and helpful research partner Ms Sara during this course of project work

I am so grateful to have Mr Kiran Kone and Mr Sharath babu Nagarajan to support me morally and technically.

Finally, I would like to express my special thanks to God, family and friends for having hope and faith in me as always.

Contents

List of Figures

Chapter 1

Introduction

1.1 Thesis motivation

The production of cars has became a leading industry in almost every area of the world. The world's car stock exceeded 80 million after the Second World War, then more than 90 million in 1960. Five years later this number was 130 million, 291 million in 1980, 419 million in 1990, and 731 million in 2011. According to the forecasts, it will reach two billion by 2020[1]. From the beginning, humans used vehicles for various purposes and it has reduced the effort and time in transportation. The innovations done by vehicle industries were helpful to have a better performance of it, but unfortunately, there were also majorly more fatalities of occupants due to poor safety features. Over the year there is numerous amount of development in the vehicle industry. Lately, the companies started to invest their valuable profits into the development of safety features of the vehicle to ensure the protection of the occupants inside and the surroundings.
The Global status report on road safety 2015, reflecting information from 180 countries, indicates that worldwide the total number of road traffic deaths has plateaued at 1.25 million per year, with the highest road traffic fatality rates in low-income countries. In the last three years, 17 countries have aligned at least one of their laws with best practice on seat-belts, drink–driving, speed, motorcycle helmets. While there has been progress towards improving road safety legislation and in making vehicles safer.Agenda for Sustainable Development: halving the global number of deaths and injuries from road traffic crashes by 2020 [2].

The statistics show 57% of road accidents are happening by human driving error[2]. Car making industries believed that automation technology would be the solution to reduce these errors and started researching on the relevant field. Automated Robots plays a vital role in each industry to reduce the risk, difficulties involved in the work and in the same motive the automated actuation of vehicle maneuvers are coming to exist nowadays. The autonomous vehicle is the future road transportation which will ensure high-level security. Self-driving vehicles are developed for various operational design domains (ODD). Operational design domains are the different scenario (Urban, Suburban and Highways) infrastructures.

The autonomous vehicle is capable enough to sense the environment and make the decision and move with little input or without input from humans. It is mainly equipped with multiple sensors for sensing and the vehicle's advance control system will receive inputs from sensors to follow the path without an obstacle. Challenging and competitive reach experiments are currently going on to propel it without any human interventions. Autonomous means having the power of self-governing and autonomous controllers has the power and ability to self-governance in the power of control function. The Control system of the vehicle is a very important part of the AD system, it should be capable enough to withstand and give the proper command for any vulnerable encounter in the environment. Intelligence autonomous control systems use AI (artificial intelligence) competent to take action on itself without any human interactions[4]. Research experiments of this intelligence system became highly progressive through the continuous effort of department engineers and graduates working for various competitions and company projects which has been funded by Government authorities.

The motivation of the project is to built an efficient driver-less vehicle with strategical control approach for vehicle's decision making and effective performance in the track to top the competition. In the process of building an autonomous system, there is three major division and they are Perception, Decision control, and Action. This thesis mainly focuses on the second part i.e, Decision and control. The objective of the analysis is to find the effectiveness of proposed strategy and define additional technical improvements to refine and make the system more robust for our application.

1.2 Evolution of Driver-less Vehicles

The first idea of creating an autonomous or self-driving vehicle has arisen to human brains in late 1920s. Norman Bel Geddes created the first self-driving car, which was an electric vehicle guided by radio-controlled electromagnetic fields generated with magnetized metal spikes embedded in the roadway. By 1958, General Motors had made this concept a reality. The car's front end was embedded with sensors called pickup coils that could detect the current flowing through a wire embedded in the road. The current could be manipulated to tell the vehicle to move the steering wheel left or right. In 1977, the Japanese improved upon this idea, using a camera system that relayed data to a computer to process images of the road. However, this vehicle could only travel at speeds below 20 mph. Improvement came from the Germans a decade later in the form of the VaMoRs, a vehicle outfitted with cameras that could drive itself safely at 56 mph. As technology improved, so did self-driving vehicles' ability to detect and react to their environment[5]. Later the concept de-

Figure 1.1: Evolution of Driver-less cars

velopment and innovation in the field became challenging. Integration of multiple

systems where progressing through constant research in the area of robotics. Companies started to explore this division to associate the vehicle to investigate the behavior of big movement in the road when the motive of creating was getting fewer interests over few decades due to its complexity a competition challenge initiated in 2003,2004 and 2007 by DARPA (Defense Advanced research project agency) triggered university students to overcome the impossible portions[6].
Most of the big names like Mercedes Benz, Audi, BMW, Tesla, Hyundai, etc. they have begun developing or forming partnerships around autonomous technology. They invested sizable resources into this, and by making this step they wanted to be leaders at the market of self-driving cars. Up to this point, numerous aids, software, and sensors have been put into these cars, but we are still far from full autonomy[1]. To have full autonomy there should be a lot of safety measure strategy that must be integrated with the global system, for instance, collision detection and avoidance, etc. Shortly after this competition, Google's Driver-less Car initiative has brought autonomous cars from the university laboratory into commercial research. In 2012 Google has developed and tested a fleet of cars and initiated campaigns to demonstrate the applications of the technology through, for example, videos highlighting mobility offered to the blind. Google is not alone. In 2013, Audi and Toyota both unveiled their AV visions and research programs at the International Consumer Electronics Show, an annual event held every January in Las Vegas (Hsu, 2013)[7].

1.3 SAE Autonomous Vehicle Standard levels

SAE (Society of Automotive Engineers) established the new international standard(J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems) is to have common and unified classifications and supporting definitions that are, Identify six levels of driving automation from "no automation" to "full automation", Base definitions and levels on functional aspects of technology,Base definitions and levels on functional aspects of technology, Describe categorical distinctions for a step-wise progression through the levels, Are consistent with current industry practice, Eliminate confusion and are useful across numerous disciplines (engineering, legal, media, and public discourse) and Educate a wider community by clarifying for each level what role (if any) drivers

have in performing the dynamic driving task while a driving automation system is engaged[8].

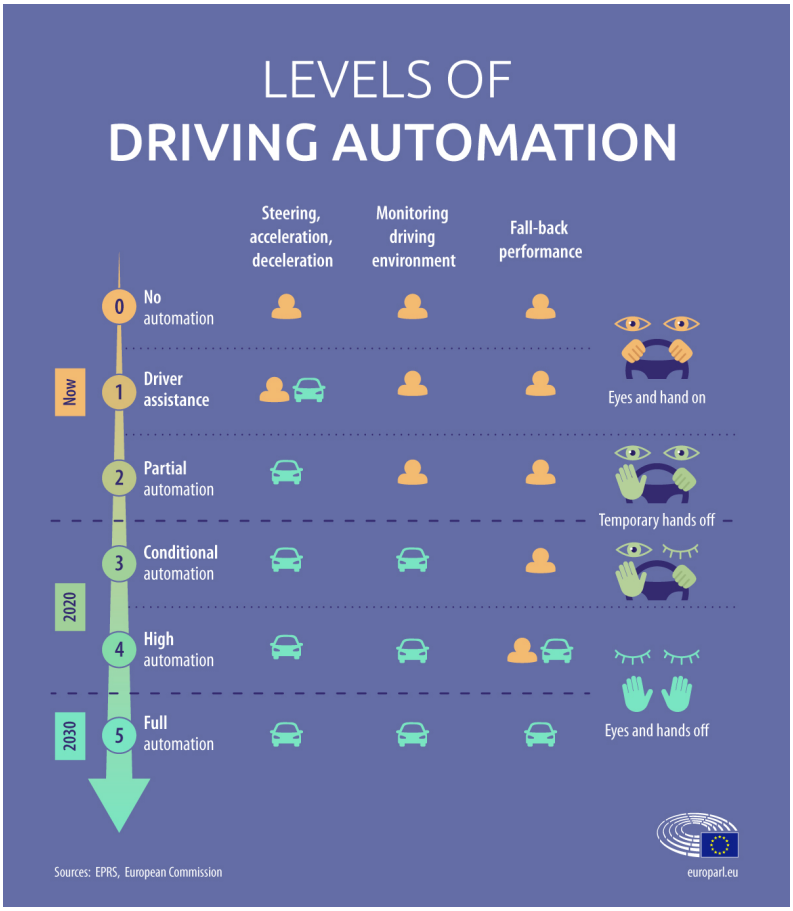


Figure 1.2: Levels of Driving Automation

According to the Figure1.2 ADS(Automated Driving System)categorized based on certain Driving Objectives, for instance, a. Whether the driving automation system performs either the longitudinal or the lateral vehicle motion control sub-task of the DDT(Dynamic Driving Task). b. Whether the driving automation system performs both the longitudinal and the lateral vehicle motion control sub-tasks of the DDT (Dynamic Driving Task)simultaneously. c. Whether the driving automation system also performs the OEDR (Object and Event Detection and Response) sub-task of the DDT(Dynamic Driving Task). d. Whether the driving automation system also performs DDT(Dynamic Driving Task) fallback. e. Whether the driving automation system is limited by an ODD(Operational Design Domains)[8].

Summarizing above mentioned five categorizations into 6 discrete levels of automation standards
Level 0: No Driving automation, Both longitudinal and lateral maneuvers are completely performed by Driver even though the vehicle is equipped with active safety systems.

Level 1: Driver assistance, The Vehicle's System is capable enough to perform either one of the maneuvers(lateral or longitudinal), but not both simultaneously. The driver will have manual monitoring control during the system action and perform other tasks. (eg. ACC(Automated Cruise control)

Level 2: Partial Driving Automation, The vehicle's AD system is capable enough to perform both lateral and longitudinal maneuvers under the complete supervision of the driver. The driver takes action in cases of any vulnerability in doing the driving tasks.

Level 3: Conditional Automation, The vehicle's AD system is capable enough to take full control over certain scenarios and in case of any malfunction fall back option warns the driver to take full control manually. The driver's full attention is required during the action.

Level4: High Driving Automation, The vehicle's AD system is capable enough to perform the full driving task over limited scenarios and it is capable enough to handle high constrains and there won't be any need in driver's intervention.

Level5: Full Driving Automation, The vehicle's AD system will have capabilities to function in all scenarios and there won't be any contingency safety system is needed in case of critical situations.

1.4 Driver-less Vehicle competition

Formula Student Germany(FSG) is an international student level design competition funded by top OEM (original equipment manufacturer) companies, for instance, Audi, BMW, Daimler, Tesla..ect. Students design and develop a one seated

formula race car and compete with other teams coming from various universities across the world. The motivation for this competition is to bring the creativity of innovative students, experience the value time and effort of teamwork. The competition is not won solely by the team with the fastest car, but rather by the team with the best overall package of construction, performance, and financial and sales planning and challenges students to build the vehicle considering the manufacturing feasibility in automotive sectors. As we all know DARPA challenge has created big confidence in creating the self-driving vehicle likewise In 2016 FSG announced a new competition competition to build an autonomous race car and it created a big motivation for all students in Europe to participate. The new, future-oriented competition fronts students with a completely new challenge. They are to develop a race car that can run without a driver in autonomous mode, or with a driver in a manual mode. The vehicles must meet the technical requirements of the two existing classes of the competition, Formula Student Combustion (engine) or Formula Student Electric (electric motor). Which driver-less car will win the new competition will not only be decided based on pure autonomy. Race condition is the vehicle

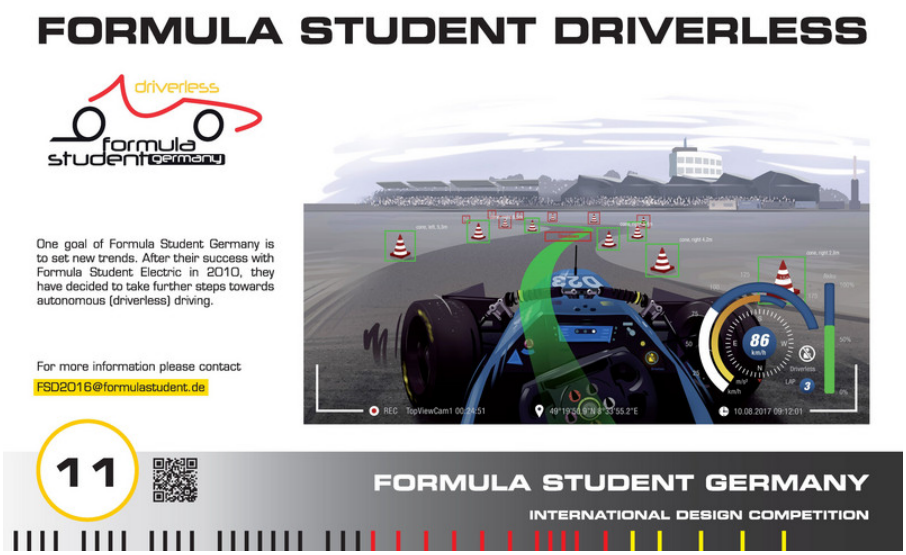


Figure 1.3: Formula Student Driver-less

should cover 5km over 10 laps as fast as it can with the inner engineering capabilities. Challenging part is racetrack is unknown prior to the competition and for the first lap the vehicle has to propel on its own with the help of proper vision sensors,

control and decision.Planning strategy includes integration of models such as the boundary detection, center-line Estimation and Local trajectory planning.Dynamic behavior of the vehicle should comply with the planning strategy in order to avoid the failure of systems. Data accumulation and filtration points from environment should be processed parallel to achieve optimal path to follow from the next consecutive lap. It involves minimization of time required to reach the destination.

1.5 State of art

1.5.1 Project Work Process

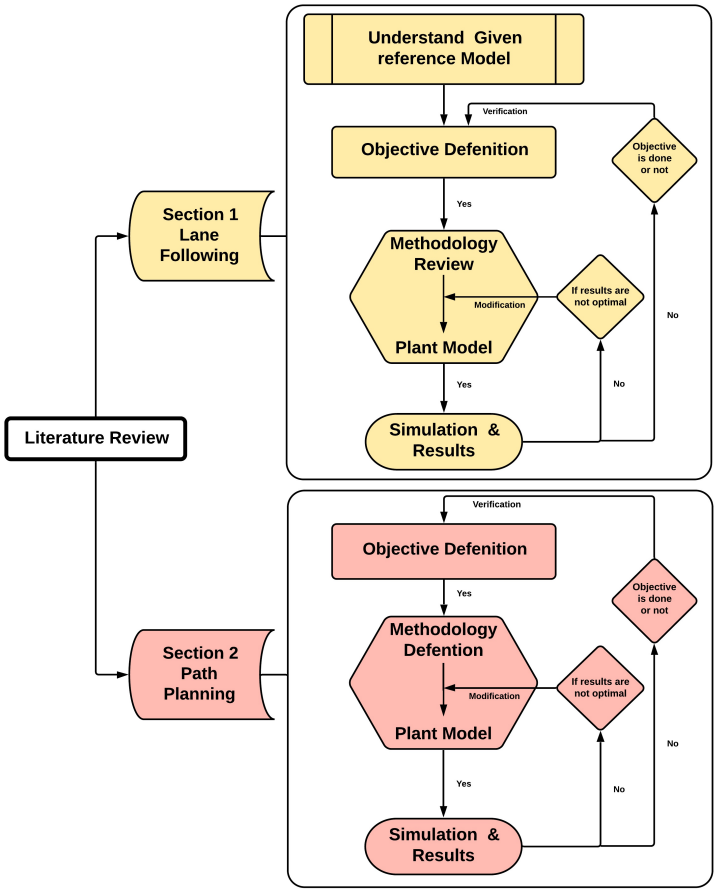


Figure 1.4: Project Process Chart

This project is an ongoing research project at LIM (Laboratory of Mechatronics) for a competition. The progress of the works has been illustrated as a flow chart in Figure1.4. Firstly complete review of relevant Literature papers made to have a basic idea about the nature of the project. With proper guidance from the mentors, the first section goal has been determined, which is basically to analyze the performance of the model and expand based on the project requirements.
The next process is defining a methodology to have a clear image of the project and its hierarchical process model, Once it is defined developed model is reconstructed to the required changes. Validation of the model is the last step in this section to control and verifying the results. If in case the determined objective is not achieved, then modification of the constraints in the model is made to achieve optimal results. Section 2 is also the same process as the first section, but the model is not verified with the previous model, because in this section new model is created for the further development of this project.

1.5.2 Overview of Lane following

Automation of driving maneuvers idea is being developed by researchers over decades throughout the world to make vehicles on roads are to be highly secure and safe without any driving task errors. In every Automation system, the controller plays a vital role in command generation for guidance, error rectification for safety and comfort of the vehicle, and this topic state of the art is to represent an overview of the global control architecture of the autonomous vehicle.
Relevant resources in the field of the control system and engineering helped researchers and engineers to make analysis using different control architectures. For this project, a Model predictive based (MPC) control system is adopted and in recent times it has been used by a lot of researches to develop their autonomous vehicle systems. The major reason for choosing MPC is that it has the predictive ability, whereas the PID controller does not have such capabilities. MPC has a wide range of functionalities for various applications, it utilizes the model of a system to predict its future behavior, and it solves an online optimization algorithm to select the best control action that drives the predicted output to the reference. It can control multi-input multi-output systems that might have interactions between inputs and outputs. MPC can handle constraints, and it has preview capability

[10].

Lane keep Assist system(LKA) and ACC(Automated cruise control) system are the two vital parts of and advance driver assistance system(ADAS). LKA is developed for better handling purpose and it deals completely with the lateral motions, Whereas ACC is developed for speed regulation purposes and it deals completely with longitudinal motions. Many researchers have presented the control architecture using these systems to develop ADS (Automated driving system). The idea presented in the paper [11] is for a Lane keep assist. The main goal proposed in this paper is to minimize the lateral position error and heading angle error with satisfying respective safety contains The Paper deals with the Robust model predictive control strategy for lateral control and longitudinal velocity is assumed to be constant. They have introduced recursive adaption of unknown steering offset in an MPC framework. The control strategy guarantees robust satisfaction of imposed operating constraints in the closed-loop along with model adaptation. It also assures the recursive feasibility of the proposed MPC(Model prdictive control) controller on a road path of fixed curvature. On a variable curvature, The proposed algorithm is accordingly modified to a switching strategy. It is well understood from this research that problem formulation is a critical part to get the desired output and it also proved that the adaptive MPC is more robust and efficient in satisfying the recursive constraints when compared to a normal MPC(Model Predictive Control).

Similarly, the Lane-keep assist system for a passenger car is presented in the paper[12]. The paper is about an active system in the car and it is termed as a lane-keep aid system (LKAS). This was developed with proper warning/ intervention strategies to decrease unwanted lane departures. Module formulation is simple and executed with the Electronic power-assisted steering actuator(EPAS). Intervention module is considered as a control model which will ensure when and how it should happen, its actuation is based on steering angle and its torque calculation from perceived input road information data.

The control architecture presented in the paper [13]. is based on the linear model predictive based lane keep and obstacle avoidance. The optimization problems are formulated as a quadratic program(QP) to determine optimal braking, throttle, and steering command. control model is separated into two levels as high and low, therefore the high level generates an obstacle-free trajectory, while low level tracks

this planned trajectory. Since the MPc formulation is online solution-based, so they have planned to spatial vehicle model to reduce its computational complexity, this controller is capable enough to handle constraints for low curvature road such as highways. Paper is presented with two main contributions, the first one is under the assumption of a large radius of curvature they derive a linear time-varying model (LTV) of the vehicle lateral dynamics as a function of longitudinal speed profile and the second is a model to formulate MPC problem for lane-keeping and obstacle avoidance, considering both lateral and longitudinal dynamics.

With inspiration from the reference papers and valuable research work, the control

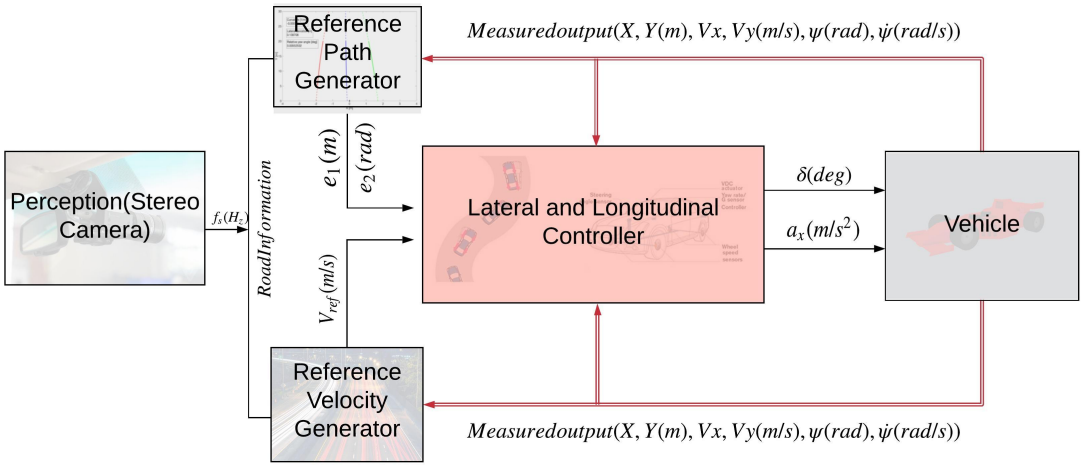


Figure 1.5: Global Control Architecture of Lane Following method

strategy presented in this thesis is completely built using Matlab and Simulink. A general overview of the complete architecture is presented in Figure1.5, it is based on the generic structure of the ADS model i.e, Perception,Extraction, Decision-control, and Action. In this model, the virtual stereo camera is used in simulation to perceive real-time data of the surrounding environment, and the control strategy is formulated based on an online optimization inside adaptive MPC. The adopted technique works in association with real-time data coming from the vision sensor and proper feedback states from the vehicle model to regulate longitudinal speed profile and tries to minimize path position errors in terms of lateral deviation and relative yaw angle abiding the stated conditions.

The performance of this technique is evaluated with two different vehicle models on the same simulated scenario and this type of analysis helps us to better

understand the behavior of control system in handling states from the higher vehicle models. It is too complicated to build a vehicle model equivalent to real, so the experimentation done by co-simulating with the another simulation software. CARSIM(Mechanical simulation software) is used to co-simulate with Simulink model.Visual simulation solver uses S function to solve the math model given from the Simulink environment and integration of Simulink and CARSIM is done by API (Application program interface)[14].

1.5.3 Overview of Motion planning

Path planning is very important system for AD vehicle as it is responsible for providing collision free path from start to destination. This section shows an overview of the global control architecture of the model. The responsibility of this system is to make proper decision, and it includes collision avoidance strategy,shortest path finding method, and Dynamic trajectory optimization.There are two processes of path planning and control, and they are online optimization and offline optimization. The online refers to real-time optimization, where the module is completely active during the process and tries to define a trajectory associating with the behavior of and generates sufficient velocity profile based on handling limits, therefore the implementation requires an efficient processing unit because the module increase computational cost. The offline process works in the ideal phase where the trajectory generation is created by strategical algorithms using the extracted data from the perceived environment.

A review of relevant papers helps us to increase the idea spectrum in defining the proper strategy for our problem. The idea proposed in the paper [15] is to achieve the minimum curvature path using a quadratic optimization problem(QP) formulation. The technique assures in terms of curvature approximation and based on curvature constraints it mainly focuses on reducing linearization errors on corners, and for longitudinal motion, they adopted forward and backward solver which provides velocity profile considering the lateral and longitudinal acceleration limits of the car. For an offline process, they used three manually driven data to generate the occupancy grid of the track by fusing GPS, odometer, and LIDAR measurements. The map is further post-processed because the run-off areas and curbs are

impossible to detect using LIDAR. It also includes the generation of the center-line required for the subsequent optimization algorithm. The actual driving cycle is obtained by finding a minimum curvature path and then generating a suitable velocity profile considering handling limits. In the online process, the module is active during the driving session. The Planning Unit (PU) performs two major tasks, one of them is monitoring the system mission, and this includes launching a car, monitoring the subprocesses, counting the number of laps, and adjusting the velocity profile based on given scale factor. The other one is for the extraction of local trajectory from the offline global trajectory and that is used for control.

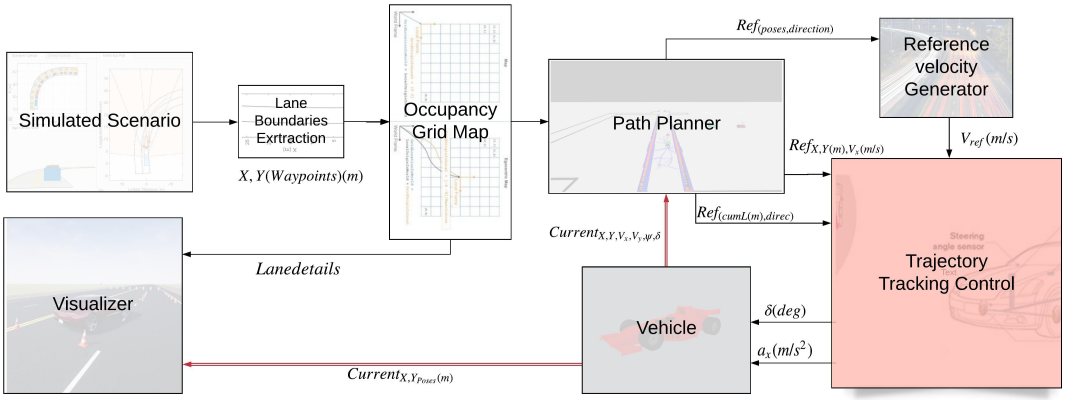


Figure 1.6: Global Control Architecture of Path Planning Method

The methodology presented in this thesis is offline process-based considering some assumptions. In the real case, the data is extracted from the vision sensors, whereas in this case, we used reference data to design simulated track scenarios. The occupancy grid map is created using the lane details of the simulated scenario and it is the surrounding environment for the AD vehicle. The incrementally based path planner generates a local trajectory for every discrete route points by analyzing the behavior of the path. It is completely a closed-loop model presented in figure (5), the velocity profile generates speed profile from the local trajectory geometrical data and the path analyzer provides the suitable reference velocities to the controller analyzing the current status of the vehicle. Discrete Stanley lateral and longitudinal controller is adapted to provide the required steering and acceleration command to the vehicle model. The trajectory tracking is based on the lateral deviation and relative heading angle. This type of control system is adopted for

simplicity and to reduce computational cost.

1.6 Thesis outline

The thesis is organized as follows:

- *Chapter 2*: It presents the detailed view of vehicle modeling used for validation in control system, in particular 3 DOF rigid vehicle model,Tire model, drive-line dynamics, Carsim vehicle model, Linearized vehicle model for MPC.
- *Chapter 3*: It presents the detailed view of Lane following methodology, in particular Perception, Reference trajectory and Speed Profile Generation,MPC control design and problem formulations.
- *Chapter 4*: It presents the detailed view of Path planning and tracking methodology, in particular Mapping, Planning method, Route planner, Behavior planner, Path analyzer, Stanley Lateral and Longitudinal control design for tracking.
- *Chapter 5*: It presents the performance evaluation of both lane following method and path planning and tracking methods by means of simulation. The analyzed results are discussed in detail
- *Chapter 6*: It presents the conclusions and future works of this project.

Chapter 2

Modelling

This chapter shows a detailed description of the vehicle model that is used to validate the Model predictive based control system.The Mathematical models and simulation tools increases the availability to investigate the dynamic behavior and the safety of the vehicle without a mandatory need to built or test a very costly vehicle (14 DOF paper). There are different types of models built for specific purposes, for example, The Quarter car model deals with 2 DOF allow to study the vertical dynamic behavior of single suspension system , and the 14DOF Model deals with six degrees of freedom that are classified mainly based on translation and rotational motions remaining two degrees of freedom deals with each wheel's vertical and rotational motions it allow us to study the vehicle's roll dynamics, and 7 DOF vehicle model deals without the contribution of vertical motions it allow us to study better yaw stability, and finally 3 DOF vehicle is a simple model deals with the planar dynamics of the vehicle based on translation and rotational motions. Higher models with more degrees of freedom and better interaction of motions will have good quality in the prediction of vehicle behavior, but it also increases the computational cost. In this thesis, we use a pre-assembled vehicle model from the Simulink blocks and it deals with only planar motions.The physics of this mathematical model are briefly detailed in section 2.1, and the Linearized tire model is discussed in section 2.2. The internal plant model for model-based control is the linearized vehicle model that is modeled to reduce the computational complexity of the system, and the equations are derived in section 3.To increase the accuracy of the vehicle model, we adopted a vehicle model from the commercial

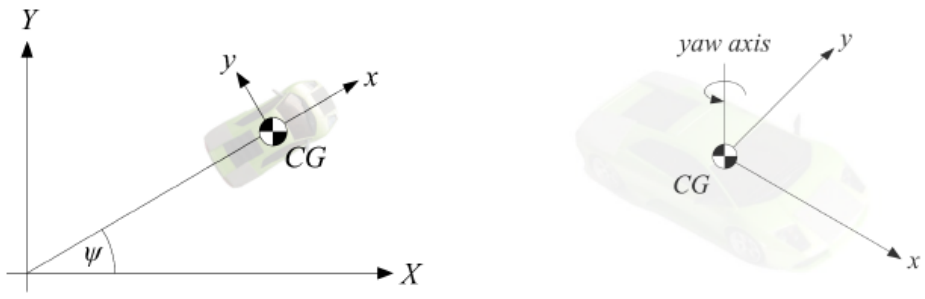
software that is equivalent to a real vehicle, and it is presented in section 4.

2.1 Vehicle model for validation

The mathematical model of the vehicle is a multi-body system well established to dynamic analysis. The vehicle model is a complex system with multiple degrees of freedom, and it is composed of the various subsystems (vehicle body, wheels and tires, Power train, suspension, steering, brakes, and control system). As stated before we use two vehicle models in this thesis for the analysis purpose, and they are rigid vehicle models (single track) and carsim complete vehicle model. The kinematic and dynamic equations of motion of these two are briefly derived in subtopics.

2.1.1 Vehicle coordinate system

The Cartesian co-ordinate system used in this thesis for modeling the vehicle dynamics is based on SAE J670 [16] This Figure2.1 represents two sub-figures a and



(a) Global coordinate system (X, Y, Ψ) **(b)** Local coordinate system (x, y)

Figure 2.1: Model Coordinate Systems

b, where a refer to the global coordinate system and b refer to the Local coordinate system. (X, Y) axis is the longitudinal and lateral directions of the global reference frame, and ψ is the heading angle of the vehicle concerning the global reference frame. The local coordinate system is denoted as the vehicle fixed system, where (x, y)-axis is the longitudinal and lateral directions with respect to the vehicle

reference frame, and the yaw axis refers to the vehicle's orientation concerning its center of gravity.

2.1.2 Kinematic model

The geometrical description of the mathematical model is known as kinematic model and it describes the vehicle motion without taking into account of the forces that are affecting its motion.The equations of motions are purely based on Geometric relations with certain assumptions, that is the velocity vector vector of both the axles that make steering angle δ_f and δ_r corresponds to longitudinal axis of the vehicle and due to the fact the the former assumption is true in this case then slip angle is also assumed to be zero. This is true only at low velocity, because the lateral force generated by tire is very small. The image presented in Figure2.1

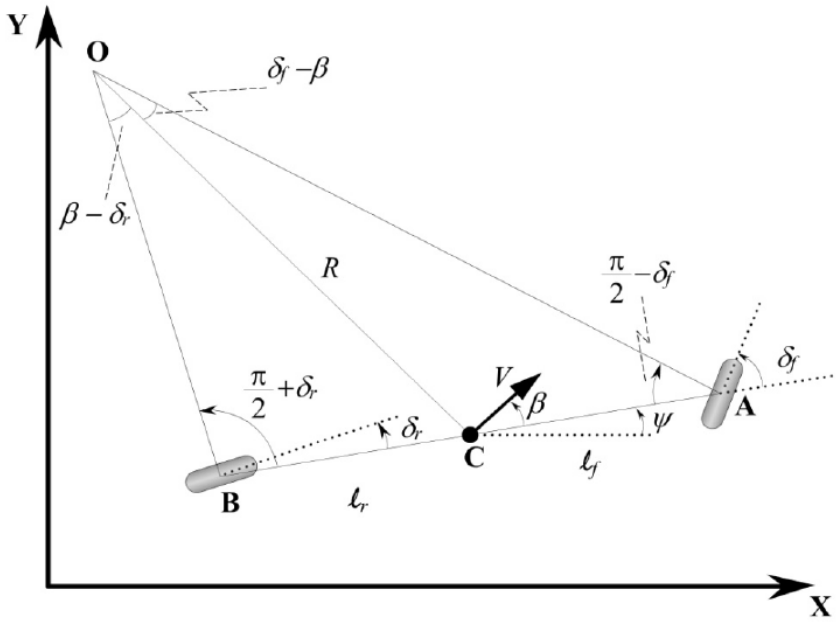


Figure 2.2: Vehicle kinematic model

depicts a bicycle model. It is understood from the diagram the two wheels of left and right are considered as single wheel in the front A and similarly in the rear B. δ_f and δ_r represents the steering angle of the front and rear wheel, whereas in this model the steering angle of rear wheel is set to be zero. It is said to be a front

wheel steered vehicle model.
The point C shows the vehicle's center of gravity(c.g) and l_f,l_r are the respective longitudinal distance from the center of gravity. The sum of both distances is know as wheel base L.

$$L = l_f + l_r \tag{2.1}$$

Since the vehicle deals with planar motions, the three coordinates X,Y, ψ helps to describes vehicle motion, where (X,Y) are the inertial co-ordinate of the vehicle that represents the location with respect to vehicle's center of gravity, and ψ is the orientation axis of the vehicle with respect to center of the gravity.
The point O refers to the instantaneous center of rotation of the vehicle and it is defined by the intersection of lines AO and BO. These two lines are drawn perpendicular to the orientation of the two wheels. The length of the line OC corresponds to the radius of the vehicle trajectory R, and it is perpendicular to the velocity vector V.
Applying the sine rule to triangles OCA and OCB, remembering that δ_r is equal to zero, it is possible to define the following equations [17]:

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin(\frac{\pi}{2} - \delta_f)}{R} \tag{2.2}$$

$$\frac{\sin(\beta)}{l_r} = \frac{1}{R} \tag{2.3}$$

After some manipulation and multiplying by $\frac{l_f}{\cos(\delta_f)}$, equation 2.2 becomes:

$$\tan(\delta_f)\cos(\beta) - \sin(\beta) = \frac{l_f}{R} \tag{2.4}$$

Likewise, multiplying by l_r, equation 2.3 can be re-written as:

$$\sin(\beta) = \frac{l_r}{R} \tag{2.5}$$

Adding equations 2.4 and 2.5, the following relation has been obtained:

$$\tan(\delta_f)\cos(\beta) = \frac{l_f + l_r}{R} \tag{2.6}$$

This formula allows to write the radius R of the vehicle trajectory as a function of the front steering angle δ_f, the slip angle β, and l_f.
If the value of radius R changes slowly due to low velocity, the yaw rate $\dot{\Psi}$ of the vehicle can be assumed equal to the angular velocity ω that is defined as:

$$\omega = \frac{V}{R} \tag{2.7}$$

Therefore, the yaw rate $\dot{\Psi}$ can be described as follows:

$$\dot{\Psi} = \frac{V}{R} \tag{2.8}$$

Using formula 2.6, the equation 2.8 can be re-written as:

$$\dot{\Psi} = \frac{V\cos(\beta)}{l_f + l_r}\tan(\delta_f) \tag{2.9}$$

After all these assumptions, the overall equations of the kinematic model can be defined as:

$$\dot{X} = V\cos(\Psi + \beta) \tag{2.10}$$

$$\dot{Y} = V\sin(\Psi + \beta) \tag{2.11}$$

$$\dot{\Psi} = \frac{V\cos(\beta)}{l_f + l_r}\tan(\delta_f) \tag{2.12}$$

2.1.3 Dynamic model

The kinematic model is quite simple with two degrees of freedom and the assumptions are valid only in the case of low speeds that are lower than 5m/s. To define the proper control system for the vehicle we need to consider a more complex and dynamic model because it exhibits interactions in multiple directions. In this thesis, initially, we used a pre-assembled 3DOF rigid vehicle model from the vehicle dynamics block set in Simulink. This model describes motion in three planes. The lateral and longitudinal are the two translation motions corresponding to the X,Y axis, and the yaw is the rotational motion corresponds to the orientation axis of the vehicle. The front and rear wheels of the model are considered as a single-center wheel and it is commonly denoted as a single track rigid vehicle model. The steering of both axles are possible in this model, but we set the rear steering to zero to have a model similar to our original front steered vehicle.

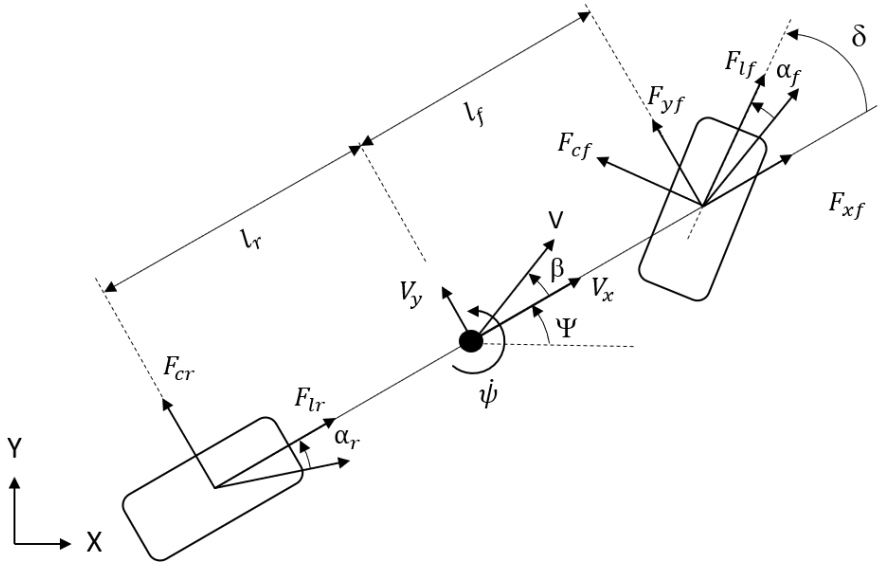


Figure 2.3: 3DOF rigid vehicle model

The Figure2.3 shows the schematic diagram of the 3DOF rigid vehicle model. The nomenclatures referred in the diagram is given F_l,F_c refers the tire forces acting along the longitudinal and lateral directions, it is also termed as traction and cornering forces of tires. F_x,F_y refers the longitudinal and lateral forces action on the vehicle's C.G(center of gravity). F_z refers the normal force that are acting on the wheel center. l_f and l_r refers the longitudinal distance between the front and rear

axles from vehicle center of gravity. δ refers to the steering angle, and a_f, a_r refer the tire slip angles.Newtons law of motions are given by [17]

$$m\dot{V_x} = mV_y\dot{\psi} + F_{xf} + F_{xr} - F_{aero} \tag{2.13}$$

$$m\dot{V_y} = mV_{xr}\dot{\psi} + F_{yf} + F_{yr} \tag{2.14}$$

$$I_{zz}\ddot{\Psi} = l_f F_{yf} - l_r F_{yr} \tag{2.15}$$

(2.13) , (2.14) equations denote the dynamic motions of longitudinal and lateral directions with respect to vehicle center of gravity,and yaw dynamics is denoted in (2.15). The forces acting on the vehicle with respect to vehicle's C.G(center of gravity) are given by (2.16), (2.17) both contributions are related to tire forces (Front and rear) and steering angle δ.

$$F_{xf} = F_{lf}cos\delta - F_{cf}sin\delta \tag{2.16}$$

$$F_{yf} = F_{lf}sin\delta - F_{cf}cos\delta \tag{2.17}$$

$$F_{xr} = F_{lr} \tag{2.18}$$

$$F_{yr} = F_{cr} \tag{2.19}$$

2.2 Tire Model

The tire is an elastic material that creates contact with the road to ensure the movement of the vehicle in the respective direction. The contact patch between the road and tire is the critical part of the tire that generates tire forces during driving maneuvers, for instance, Acceleration, steering maneuver, and braking. It servers mainly three main functions,

- Withstands the ground forces that are on the wheel and also provides a cushioning effect during high road irregularities.
- It generates longitudinal force during accelerating and decelerating.
- It generates lateral force during cornering.

Let us see the basic construction for tire and the factors that define its performance. The tire is a complex composite structure made up of many layers of plies and reinforcement chords, and it resembles two characteristics rigid and elastic. Cross-ply tires, Radial ply tires, and Belted ply tires are the three different schemes of tires used in the vehicle industries. The classifications are made by the engineers mainly based on their performance because each of its physical construction varies. Secondly, The factors that define the tire performance are 1. Effective Rolling Radius, 2. Rolling resistance, 3. Vehicle speed, 4. Structure and material of the tire, 5. Wear of the tire, 6. Working temperature, 7. Pressure and load, 8. Tire size, 9. Road condition, 10. Side slip angle, 11. Camber angle, 12. Traction and braking force, 13. Cornering force [18].To develop a tire model it is important to determine the wheel reference system to get an idea about the forces and moments that are experienced by the tire.
Figure2.4 depicts the SAE tire coordinate system. Nomenclatures and detentions are reported. (X(longitudinal), Y(Lateral), Z (vertical)) are directional axis that represents the interactions concerning wheel planes respectively. Wheel plane is the central plane of the tire normal to the axis of rotation, Wheel center is the interaction of the spin axis and the wheel plane, Center of tire is the contact interaction of the wheel plane and projection of the spin axis on to the road plane, Rolling radius is the distance between the contact point and wheel center in the z-direction, Longitudinal force F_x is the component of forces acting by the road in the x-direction,

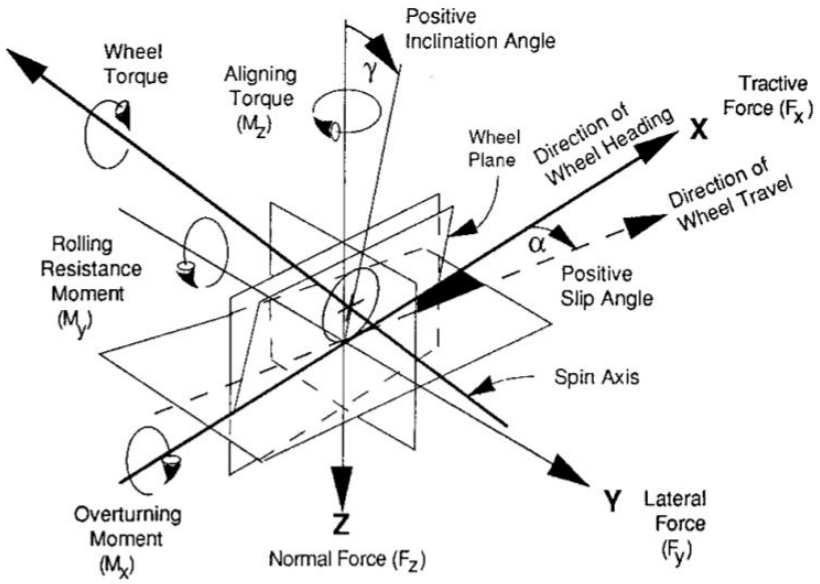


Figure 2.4: SAE tire coordinate system

Lateral force F_y is the component of forces acting along the y-axis direction, Normal Force F_z is the component of forces acting along the z-direction, Overturning Moment M_x, Rolling resistance Moment M_y, Aligning Moment M_z (are the moments created due to the interaction of tires force with respective orthogonal planes), Slip angle α is the angle between the wheel and the velocity vector, and camber Angle γ is the angle between wheel plane and vertical axis. The forces that are generated on the contact patch has high influence in the case of high slip ratios and slip angles because it no longer stays in linear region and exhibits highly nonlinear motion. We need computationally challenging model to define the vehicle dynamics or else the system would be completely nonlinear[19].

Many research has been made on this topic over the years to increase the dynamic performance of the vehicle. Researchers have developed some equivalent tire models to replicate the original behavior of the tire, for example linear tire model, semi empirical tire model, Dugoff's tire model, dynamic tire model, and they are used for different applications and test purposes. In this thesis, a simplified linear tire model is used in 3DOF rigid vehicle model and the semi empirical tire model (Pacejka tire model) used in the Carsim vehicle model. The first subsection presents the force characteristics of linear tire model, and the second subsection presents the Pacejka

tire model.

2.2.1 Linear tire model

Linear tire model is defined to generate longitudinal and lateral forces completely in the linear range. It is also said to be an ideal tire model. The forces are maintained in the linear graph with some assumptions, that is assuming low slip angle and slip ratio.
First the longitudinal tire force at lower slip angles is presented. The positive contribution of longitudinal forces are generated when the wheel is accelerating and it is opposite during braking. The longitudinal slip is developed when the wheel exchanges its longitudinal force with road. In other words it is the difference between the wheel velocity and the equivalent rational velocity[17]. For better understanding the longitudinal force vs longitudinal slip is presented in Figure2.5.It is evident from the graph that the force lie in linear region only at low slip σ if it exceeds the linear range wheel tends to slide. Equation (2.20) defines slip ratio

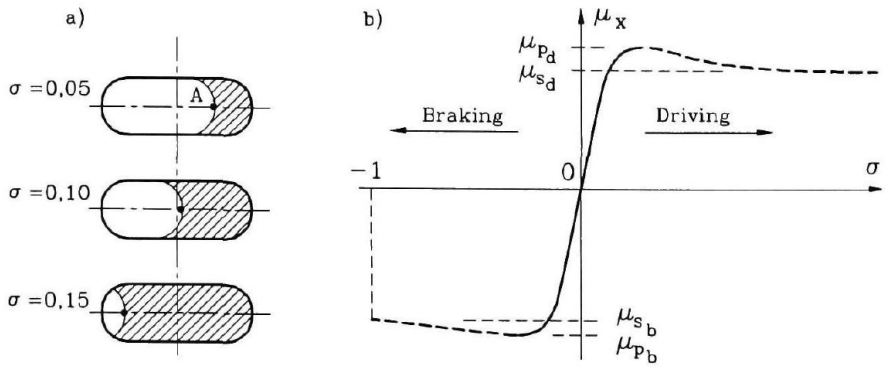


Figure 2.5: Longitudinal tire force as a function of slip
[18]

during the braking operation, and (2.21) defines it during the acceleration.

$$\sigma_x = \frac{r_{eff}\omega_w - V_x}{V_x} \tag{2.20}$$

$$\sigma_x = \frac{r_{eff}\omega_w - V_x}{r_{eff}\omega_w} \tag{2.21}$$

The major assumption considered in the model is the longitudinal force (F_x) roughly proportional to normal load (F_z) at equal value of slip [18] and the friction coefficient is assumed to be 1. It is clear from the Figure2.5 the force is proportional to slip for small slip values, therefore the tire force is given by (2.22), (2.23), where $C_{\sigma f}$, $C_{\sigma r}$ are the longitudinal tire stiffness corresponds to front and rear tire.

$$F_{xf} = C_{\sigma f}\sigma_{xf} \tag{2.22}$$

$$F_{xr} = C_{\sigma r}\sigma_{xr} \tag{2.23}$$

Secondly, the Lateral tire forces at low slip angle is presented. Cornering force is highly related to the lateral deformation of the tire at contact patch area. This tire force is really important during the turning maneuvers, where it ensures the vehicle to stay on the expected trajectory without much deviation, and it has also high influence on handling behavior of the vehicle.

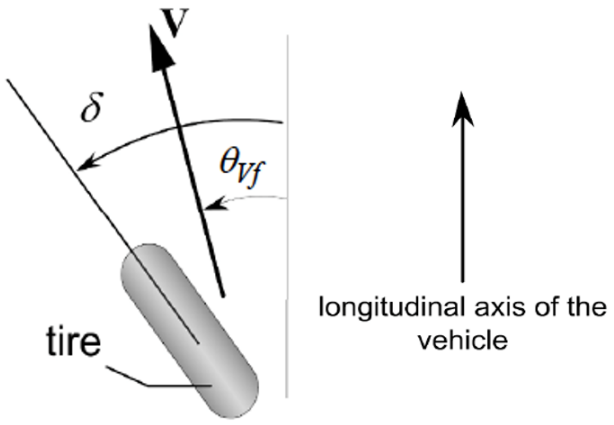


Figure 2.6: Slip angle and lateral force
[17]

This model is developed based on the two consideration, first the slip angle are

limited to low values so that the expected lateral forces lie on the linear range, in other for small slip angle the lateral tire force generated at the tire is proportional to the slip angles at the tire. Second, when the vehicle is not steered the slip angle is assumed to be zero. It is clear from the Figure2.6 the angle between the velocity vector and the wheel axis is known as slip angle and the lateral Force F_{yf} is acting perpendicular to the wheel orientation. Therefore slip angle can be given by (2.24), (2.25).

$$\alpha_f = \delta - \theta_{vf} \tag{2.24}$$

$$\alpha_r = -\theta_{vr} \tag{2.25}$$

θ_{vf} is the angle betweeng the velocity vector and the longitudinal axis, similarly θ_{vr} is the angle between the rear velocity vector and the longitudinal axis. δ represents the front wheel steering angle. According to the stated consideration we can thereby define lateral force acting on the front and rear axle.

$$F_{yf} = 2C_{\alpha f}(\delta - \theta_{Vf}) \tag{2.26}$$

$$F_{yr} = 2C_{\alpha r}(-\theta_{Vr}) \tag{2.27}$$

where C_a is the cornering stiffness that is proportional to the lateral tire force based on the respective assumptions.

$$\tan(\theta_{Vf}) = \frac{V_y + l_f\dot{\Psi}}{V_x} \tag{2.28}$$

$$\tan(\theta_{Vr}) = \frac{V_y - l_r\dot{\Psi}}{V_x} \tag{2.29}$$

θ_{Vf}, θ_{Vr} are calculated with the ratio between lateral and longitudinal velocity. It is given in (2.27) and (2.28). Where V_x, V_y are the lateral and velocity at the vehicle center of gravity, $\dot{Psi}$ is the yaw rate of the vehicle and l_f, l_r are the longitudinal

distance from center of gravity.

$$\theta_{Vf} = \frac{V_y + l_f\dot{\Psi}}{V_x} \tag{2.30}$$

$$\theta_{Vr} = \frac{V_y - l_r\dot{\Psi}}{V_x} \tag{2.31}$$

$$F_{yf} = 2C_{\alpha f}(\delta - \frac{V_y + l_f\dot{\Psi}}{V_x}) \tag{2.32}$$

$$F_{yr} = 2C_{\alpha r}(-\frac{V_y - l_r\dot{\Psi}}{V_x}) \tag{2.33}$$

2.2.2 Pacejka tire model

In order to analysis the the dynamic behavior of the vehicle we require more complex model, and the complexity of the tire model should be defined to approximate the tire behavior at different operating conditions,for example at higher speeds. The linear tire model are considerably good choice with the linear vehicle model where it deals motion equations based on a limit conditions. To move towards more realistic situation this tire model(Pacejka tire model) has developed a 'magic formula' to determine tire force F_x,F_y and aligning moment M_z for wide range of operating considering large slip angle, slip ratio and when both the forces interact at the tire contact point. The magic formula for the longitudinal, lateral force and aligning moment is given by(2.26),(2.27),(2.28).

$$F_x = D\sin(C\arctan B(1-E)(\sigma + S_h) + E\arctan[B(\sigma + S_h)]) + S_v \tag{2.34}$$

$$F_y = D\sin(C\arctan B(1-E)(\alpha + S_h) + E\arctan[B(\alpha + S_h)]) + S_v \tag{2.35}$$

$$M_z = D\sin(C\arctan B(1-E)(\alpha + S_h) + E\arctan[B(\alpha + S_h)]) + S_v \tag{2.36}$$

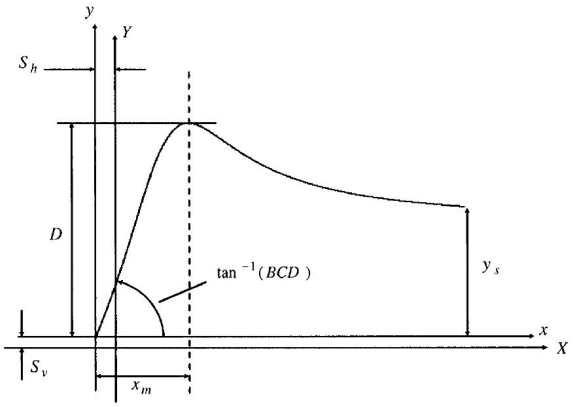


Figure 2.7: Magic formula parameters
[18]

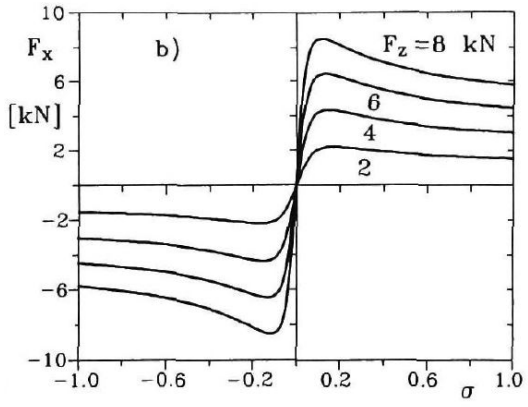


Figure 2.8: Longitudinal tire force as a function of slip(Pacejka)
[18]

Figure2.8,2.9 depicts the longitudinal tire force is proptional to its slip for different normal force F_z, similarly the lateral tire force and aligning moment propotional to slip angle for different normal force F_z.
The parameter that defines the magic formula B,C,D,E,S_h, S_v,The nomenclatures of Figure2.7 are reported here,

- B - Stiffness factor
- C - Shape factor
- D - Peak factor

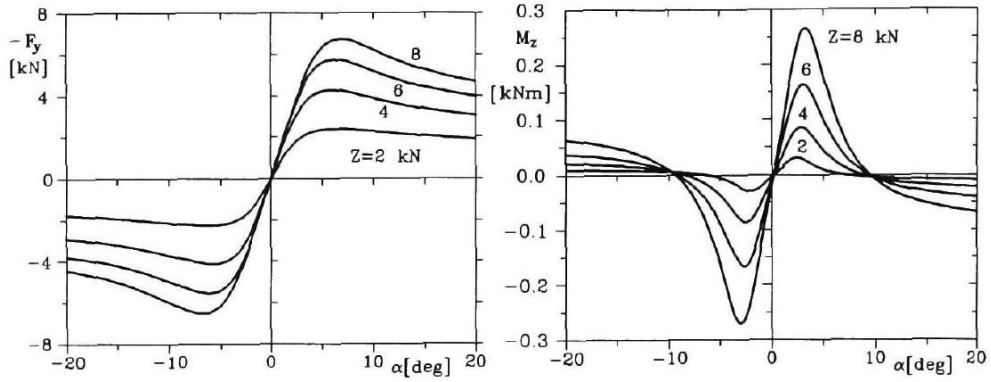


Figure 2.9: Lateral tire force and aligning moment as a function of slip ratio [18]

- E - Curvature
- S_h - Horizontal shift
- S_v - Vertical shift

2.3 Drive line dynamics

The drive line dynamics deals with the longitudinal motion, that is the engine model determines adequate torque, delivers to the transmission model and the torque converter divides sufficient torque to respective wheel model. The equation of motion is given by sum of all model contributions.It is a difficult process to design such a complex drive line model, so we have used a simplified longitudinal dynamic model and a first order drive line dynamics with time constant $\tau = 0.5$s to determine engine torque and to track the desired acceleration generated by the MPC(upper level controller).

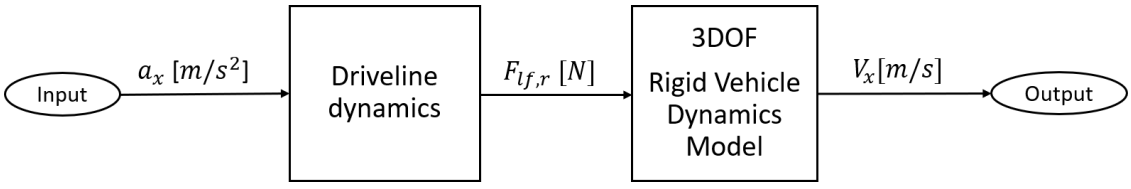


Figure 2.10: Driveline dynamics architecture

The model is developed based on certain assumptions that the torque converter in

the vehicle is locked and there is zero-slip between tire and road.Using the above assumptions, the engine torque required to track the desired acceleration command is first calculated, then with the engine torque values, engine maps and nonlinear control techniques are used to calculate to the throttle input command that will provide the required torque[17].

$$T_{engine} = r_w(ma_x + 0.5\rho AC_x{V_x}^2) \tag{2.37}$$

$$T_{wheel} = T_{engine}.i \tag{2.38}$$

$$F_{wheel} = \frac{T_{wheel}}{r_w} \tag{2.39}$$

Figure2.8 depicts the driveline dynamic architecture, where The input a_x is from the MPC model, and the lower level drive line dynamics determine the required engine torque that is given by (2.37) and (2.38). The calculated torque used to determine the tire force that is given as a input to vehicle model to follow the give velocity profile.For more information regarding the drive line dynamics refer [17]

2.4 Vehicle model for MPC

In this thesis work, the goal is to implement a combined lateral and longitudinal control system based on MPC for autonomous racing. For this purpose, a 2 degree of freedom vehicle model is used to define the lateral dynamics of the vehicle for controller internal plant model in terms of error with respect to the reference trajectory. The two errors are lateral displacement error e_1 , which is defined as the lateral distance between center of gravity of vehicle and the center line of the reference trajectory. Yaw angle error e_2 is defined as the difference between the yaw angle of the vehicle and desired yaw angle as dictated by the reference trajectory, as represented in Figure 2.11. The rate of change of lateral displacement error and yaw angle error are given by the equations.

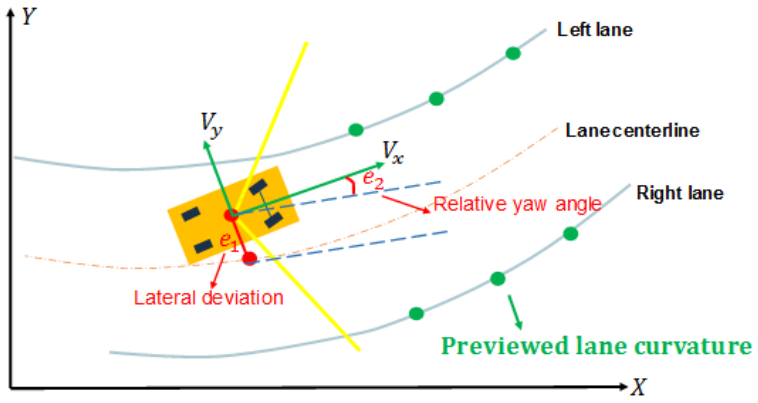


Figure 2.11: Bicycle model in terms of lateral deviation and relative yaw angle with respect to the center line of the lane

$$\dot{e_1} = V_x e_2 + V_y \tag{2.40}$$

$$e_2 = \Psi - \Psi_{des} \tag{2.41}$$

The desired yaw angle rate is given by:

$$\dot{\Psi}_{des} = V_x \kappa \tag{2.42}$$

Where, κ denotes the the road curvature.

The state-space model for lateral dynamics can be obtained by linearizing the bicycle model described in section 2.1.2. $\dot{x} = Ax + Bu$ is represented as:

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\Psi} \\ \ddot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}L_f - 2C_{\alpha r}L_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2L_f C_{\alpha f} - 2L_r C_{\alpha r}}{I_z V_x} & 0 & -\frac{2{L_f}^2 C_{\alpha f} + 2{L_r}^2 C_{\alpha r}}{I_z V_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \Psi \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2L_f C_{\alpha f}}{I_z} \end{bmatrix} \delta \tag{2.43}$$

For the longitudinal dynamics, the plant model used for control design is the transfer function between desired acceleration and actual vehicle speed and is given by[17]:

$$P(s) = \frac{1}{s(\tau s + 1)} \tag{2.44}$$

Where, τ is the time constant. Due to the finite bandwidth associated with the lower controller, the vehicle is expected to track its desired acceleration imperfectly. Thus there is a first order lag in the lower level controller performance and hence the use for the upper controller which incorporates a lag in tracking desired acceleration.

The lag in the performance of the lower controller comes from several sources, accumulating brake or engine actuation lags and sensor signal processing lags.

- The pure time delay in the engine response (60 milliseconds at 2000rpm),
- The bandwidth of the lower level multiple-sliding-surface controller that tracks acceleration
- The bandwidth of low pass filters used for other sensors such as engine manifold pressure sensor, wheel speed sensor, etc
- The bandwidth of the throttle actuator
- The lag due to discrete sampling at 50 Hz (20 ms sampling)
- The 200 ms lag due to the radar filter
- When braking, the brake actuator lag instead of engine time delay.

overall time constant of the lower level controller could be as much as 500 milliseconds.

A traditional MPC controller includes a nominal operating point at which the plant model applies, such as the condition at which you linearize a nonlinear model to obtain the LTI approximation. If the plant is strongly nonlinear or its characteristics vary dramatically with time, LTI prediction accuracy might degrade so much that MPC performance becomes unacceptable. Adaptive MPC can address this degradation by adapting the prediction model for changing operating conditions. As described in the Model Predictive Control Toolbox™, adaptive MPC uses a fixed model structure, but allows the models parameters to evolve with time. Ideally, whenever the controller requires a prediction (at the beginning of each control interval) it uses a model appropriate for the current conditions. So, in an adaptive MPC, the plant model is updated at each time step as the operating point keeps changing. i.e. Vehicle longitudinal speed. The plant model used as the basis for adaptive MPC is an LTI discrete-time, state-space model with a sampling time T_s

= 100 ms. The combined state space model for lateral and longitudinal dynamics which is used as the internal plant model for MPC is represented below:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_d v(k) \\ z(k) &= Cx(k) \end{aligned} \tag{2.45}$$

Where:

- k is time index (current control interval).
- x are plant model states.
- u are manipulated inputs. These are the one or more inputs that are adjusted by the MPC controller.
- v are measured disturbance inputs.
- A is the state matrix.
- B_u and B_d are the input matrices corresponding to inputs u and v respectively
- C is the output matrix.

$$\begin{bmatrix} \ddot{V_x} \\ \dot{V_x} \\ \dot{V_y} \\ \ddot{\Psi} \\ \dot{e_1} \\ \dot{e_2} \end{bmatrix} = \begin{bmatrix} \frac{-1}{\tau} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & -V_x - \frac{2C_{\alpha f}L_f - 2C_{\alpha r}L_r}{mV_x} & 0 & 0 \\ 0 & 0 & -\frac{2L_f C_{\alpha f} - 2L_r C_{\alpha r}}{I_z V_x} & -\frac{2{L_f}^2 C_{\alpha f} + 2{L_r}^2 C_{\alpha r}}{I_z V_x} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & V_x \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{V_x} \\ V_x \\ V_y \\ \dot{\Psi} \\ e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} & 0 \\ 0 & 0 \\ 0 & \frac{2C_{\alpha f}}{m} \\ 0 & \frac{2L_f C_{\alpha f}}{I_z} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -V_x \end{bmatrix} [\rho] \tag{2.46}$$

The inputs for the plant are separated to indicate that u correspond the fornt wheel steering angle and acceleration/deceleration command of the vehicle (controlled output of MPC), while v indicates the longitudinal velocity multiplied by the curvature κ (it is the disturbance). The inputs to the MPC y corresponds to the lateral deviation e_1 , relative yaw angle e_2 and velocity of the vehicle V_x. In the state vector, V_y denotes the lateral velocity, V_x denotes the longitudinal velocity and ϕ denotes the yaw angle. The vehicle model refers to a high-performance autonomous car characterized by the parameters listed below.

- $m = 1575$ kg, the total vehicle mass;
- $Iz = 2875$ Nms^2, the yaw moment of inertia of the vehicle;
- $l_f = 1.2$ m, the longitudinal distance from the center of gravity to the front wheels;
- $l_r = 1.6$ m, the longitudinal distance from the center of gravity to the rear wheels;
- $C_{\alpha f} = 19000$ N/rad, the cornering stiffness of the front tires;
- $C_{\alpha r} = 33000$ N/rad, the cornering stiffness of the rear tires.

Chapter 3

Lane Following Method

The chapter presents complete control methodology adapted to satisfy the objective of the autonomous racing. The proposed control strategy is for lateral and longitudinal guidance of the vehicle with respect the corresponding lane boundaries of the road and it is implemented by using model predictive based controller. According to Figure1.5 the global control architecture of the plant model is subdivided into three major division,

- "Perception" is a primary system of the ADV(Autonomous driving vehicle), where it is responsible to provide sufficient information of the surrounding environment viewing from vehicle axis. It is achieved by placing vision senors (Camera, Lidar, Radar, etc.) in appropriate position on the vehicle.
- "Extraction" is the process that accumulated the sensor data to create a point cloud map. From the accumulated data reference trajectory and speed profile is generated by implementing the geometry function.
- "Control" is the integral system responsible for providing required acceleration and steering command to follow the path, to have better guidance and stability of the vehicle.

A detailed architecture is presented in Figure3.2.It is the closed loop control model, where the blocks are decomposed according to stated subdivisions and both lateral and longitudinal controller is combined together to achieve requires optimal guidance. The control law for both the motion is formulated in Model predictive controller, where it gets the reference input from the perception to estimate the

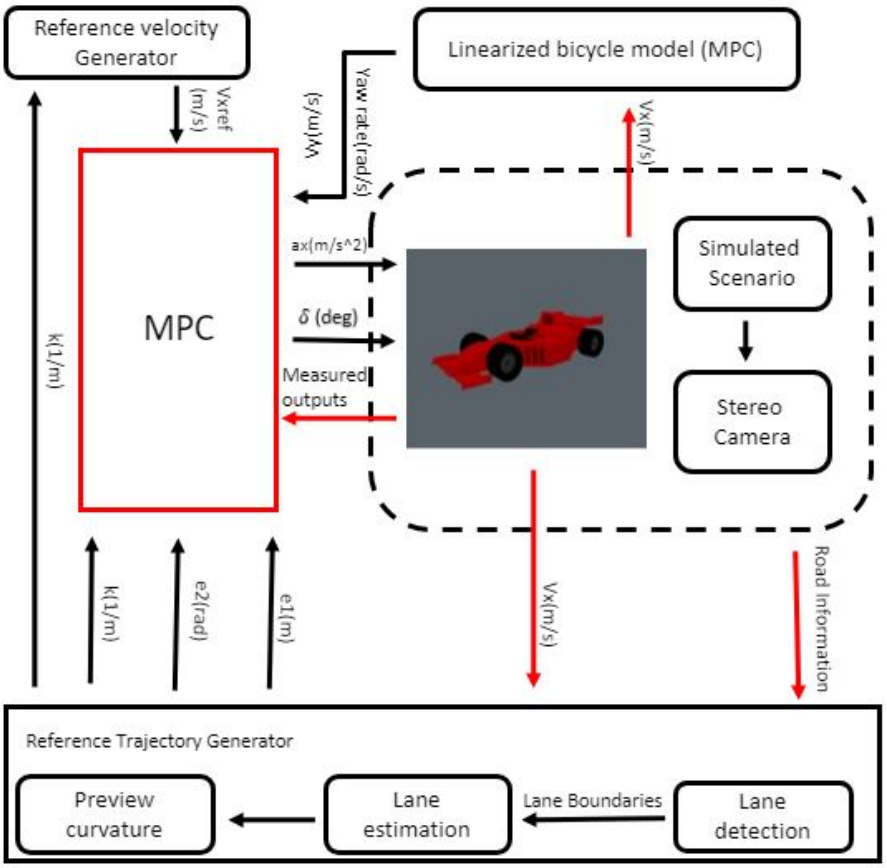


Figure 3.1: Detailed control architecture of lane following method

reference path details considering current position of the vehicle and tries to correct dynamic errors with respect to future states. To perform this simulation test we have used Matlab driving scenario creator that allow us to generate synthetic detection and test controller efficiency. Three race track scenario is build using this function and they are,

- Round track
- Handling track
- Berlin race track

Prediction model is defined in MPC to constrain the bound value concerning predictive states to initialize acceleration, deceleration command to move the vehicle

along the predicted path, whereas in this project the vehicle model used for control validation is 3dof rigid vehicle model.

3.1 Visual Perception

The visual perception is known as vehicle visionary to perceive the environment around the vehicle. Detection of static and dynamics objects is the major task of this system, and it is performed by using various vision sensors, for instance, camera is used to record the images of the surrounding along the vehicle axis to a certain focal distance, and Lidar sensor are used to percieve 360deg surrounding environment and represent as 3-D point cloud. There are multiple sensors are associated with the visual perception of the autonomous vehicle to increase the reliability and safety of the vehicle, therefore the vision detection is performed by fusing the vision sensors data to replicate the environment more realistic and also it is the primary information for vehicle control system to decision. In this thesis we have adapted computer vision detection in order to perceive the lane details. This is performed using *Driving scenario designer* from the *Automated driving toolbox* in Matlab, this allow us create synthetic driving scenario for testing the control strategy of the ADS(autonomous driving system).
The *drivingScenario* function enable us to create a road with corresponding lane numbers, place actor(ego vehicle) in the road at a specific location, configure the vision sensor (camera) at fron to vehicle, and create a manual trajectory for the vehicle to follow. Once this is designed it is exported as a function to the closed loop algorithm associated with the other block that is going to be explained in further sections. To perform the closed loop procedure for real time detection and generate information for tracking several set of blocks are used in Simulink environment. First,

- *'Scenario reader'* the current information of the actor(ego vehicle) is given as and input to the block to verify the state of the vehicle concerning global co-ordinate system. The current information includes the vehicle local position, velocities, yaw angle and yaw rate.

second,

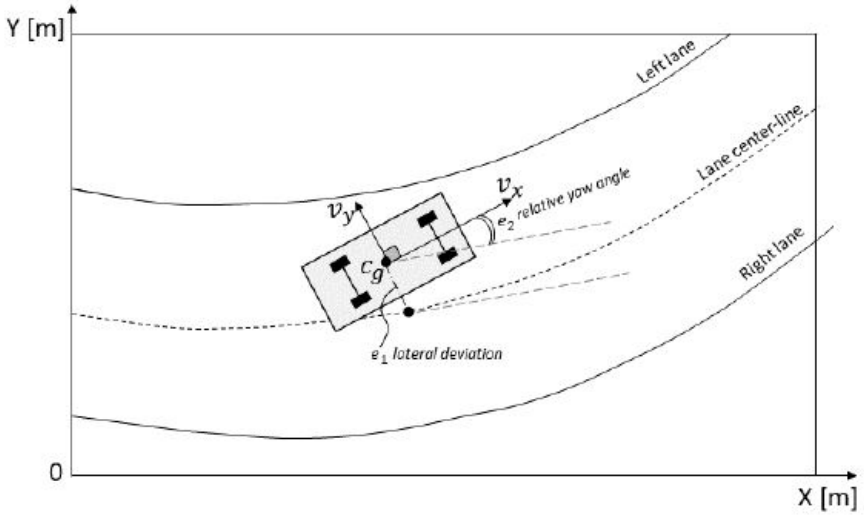


Figure 3.2: Scenario information

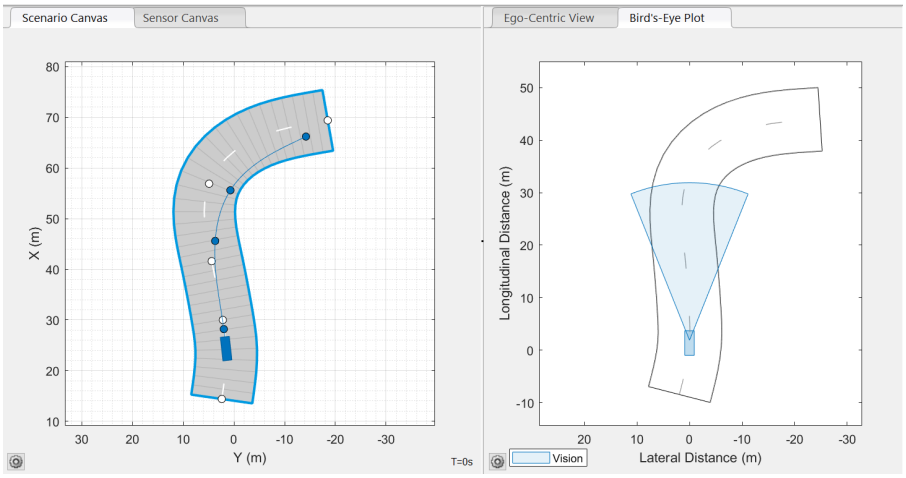


Figure 3.3: Driving scenario designer

- *'Vision Detection Generator'* sensor blocks consist of a monocular camera sensor. Camera configuration information includes the intrinsic (Focal length and optical center of the camera)and extrinsic parameters (Orientation (pitch, yaw, and roll) and the camera location within the vehicle to define the camera orientation with respect to the vehicle's chassis) in the *Vision Detection Generator* block. The camera is mounted on top of the vehicle at a height of 1.5 meters above the ground and a pitch of 1 degree toward the ground. This

information is later used to establish camera extrinsic that define the position of the camera coordinate system with respect to the vehicle coordinate system. Focal length = [800, 800]; Optical center of the camera = [320, 240];

In this thesis, monocular camera sensor uses the built-in *findParabolicLaneBoundaries* function has been used to fit the lane line model. This function uses RANSAC algorithm to find the lane line boundaries. As the function name suggests, the model created is a parabolic model that fits a set of boundary points and an approximate width. The selected boundary points correspond to inliers only if they fall into the boundary width. The final parabolic model has been obtained using a least-squares fit on the inlier points.
The function receives in input the candidate points in vehicle coordinate from the features extraction phase and it provides array of *parabolicLaneBoundary* objects for each model. The returned array includes the three coefficients [a b c] of the parabola, like a second-degree polynomial equation ax^2+bx+c, and in addition the strength, the type, and the minimum and maximum x positions of the computed boundary. The last three parameters are used to reject some curves that could be invalid using heuristics. For example, in order to reject short boundaries, the difference between the minimum and maximum x positions has been compared with a specific threshold, if the minimum threshold is not reached, the found boundaries are rejected; or, to reject weak lines, the value of the strength has to be higher than another threshold set ad hoc.

3.2 Reference trajectory generation

The controller of the lane keeping needs to receive the curvature of the trajectory like input to perform the control action on the steering angle.
"*The curvature of a curve parametrized by its arc length is the rate of change of direction of the tangent vector* [**?**]".

Considering a curve α(s), where s is the arc length and the tangential angle ϕ, computed counterclockwise from the x-axis to the tangent T = α'(s), as shown in

Figure 3.4, the curvature κ of α is defined, following the definition, as:

$$\kappa = \frac{d\phi}{ds} \tag{3.1}$$

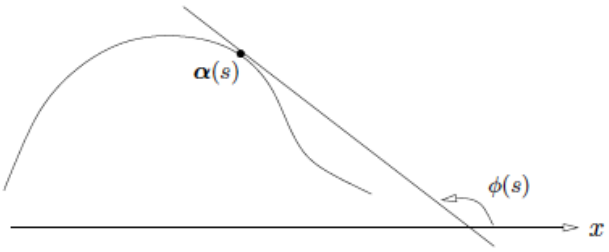


Figure 3.4: Curve α and tangential angle ϕ

The curvature can be also defined as the value of the turning of the tangent T(s) along the direction of the normal N(s), that is:

$$\kappa = T' \cdot N \tag{3.2}$$

It is easily to derive the first definition 5.41 from the second 3.2 (Figure 3.5), as follows:

$$\kappa = T' \cdot N = \frac{dT}{ds} \cdot N = \lim_{\Delta s \to 0} \frac{T(s + \Delta s) - T(s)}{\Delta s} \cdot N = \lim_{\Delta s \to 0} \frac{\Delta \phi \cdot \|T\|}{\Delta s} = \frac{d\phi}{ds} \tag{3.3}$$

To perform the measure of how sharply the curve bends, the absolute curvature

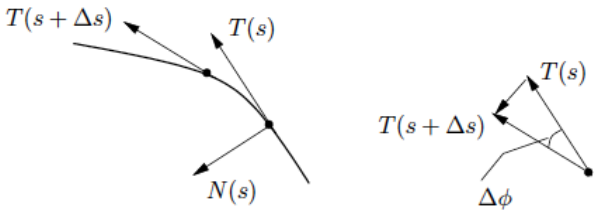


Figure 3.5: Demonstration that the definition 5.41 can be derived from the definition 3.2

of the curve at a point has been computed and it consists of the absolute value of

the curvature $|\kappa|$.
A small absolute curvature corresponds to curves with a slight bend or almost straight lines. Curves with left bend have positive curvature, while a negative curvature refers to curves with right bend.

With the second definition 3.2 it is possible defined that the curvature of a circle is the inverse of its radius everywhere. For this reason, the radius of curvature R has been identified as the inverse of the absolute value of the curvature κ of the curve at a point.

$$R = \frac{1}{|\kappa|} \tag{3.4}$$

The circle with radius equal to the curvature radius R, when $\kappa \neq 0$, and positioning at the center of curvature is called *osculating circle*, as shown in Figure 3.6. It allows to approximate the curve locally up to the second order.

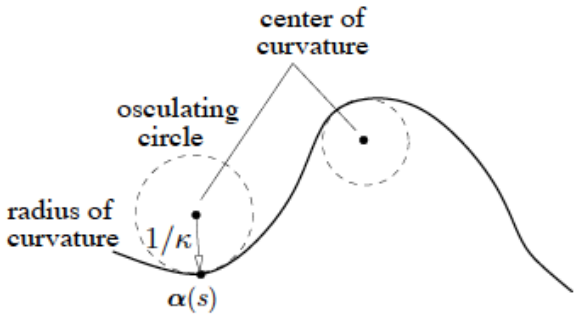


Figure 3.6: Osculating circle and radius of curvature

The curvature can be expressed in terms of the first and second derivatives of the curve α for simplicity in the computation, by the following formula:

$$\kappa = \frac{|\alpha''|}{[1 + (\alpha')^2]^{\frac{3}{2}}} \tag{3.5}$$

In order to compute the curvature in this thesis work, the *Geom2d* toolbox in MATLAB has been used. This toolbox provides the *polynomialCurveCurvature* function that allows to compute the local curvature at specific point of a polynomial curve. It receives in input the curve in parametric form x = x(t) and y = y(t) and the

point in which the curvature has to be evaluate.
The function *polynomialCurveCurvature* computes the curvature following the formula 3.5 that becomes:

$$\kappa = \frac{|x'y'' - x''y'|}{[(x')^2 + (y')^2]^{\frac{3}{2}}} \tag{3.6}$$

3.3 Reference velocity generation

The following subsections are devoted to determine the reference speed profile, two different criteria are considered here, which are available in literature. First one is based on the geometry of the road and the second one is based on the lateral comfort of the vehicle. So, maximum admissible longitudinal speed is estimated based on the road information and the speed for lateral comfort is calculated based on the information about the desired lateral acceleration. Both of them are exploited to calculate the reference speed profile by the speed profile generator.
Road information criteria: the performance of the path-following depends on the speed with which this following is done. The cruise speed is also important for the stability of the vehicle on the road. In fact, no controller can ensure the path-following if the cruise speed is excessive. Thus, the speed of the vehicle should be reduced when approaching a bend. This adaptation of the cruise speed depends on the difficulty to cross the bend. There are several systems designed by automakers for assisting driver when approaching a bend, like those developed by Daimler-Chrysler defining the maximum admissible speed based on the curvature of the road:

$$V_{max} = \sqrt{\frac{g\mu}{\kappa}} \tag{3.7}$$

where g, μ and κ are respectively the gravity, the friction coefficient and the road curvature. The description given by the model (3.7) is incomplete and may be inappropriate to determine the maximum admissible speed in some situations. Indeed, the only parameter considered in this model is the road curvature. However, other characteristics of the road can be considered. For this reason, more sophisticated models are proposed. The National Highway Traffic Safety Administration

(NHTSA) recommends for the calculation of the maximum entry speed in bends the following model:

$$V_{max} = \sqrt{\frac{g}{\kappa}(\frac{\phi_r + \mu}{1 - \phi_r \mu})} \tag{3.8}$$

where ϕ_r is the road camber angle.
Then, the acceleration a that should be applied to bring the speed of the vehicle to the maximum admissible speed given by (3.8) should be less then:

$$a_{max} = \sqrt{\frac{V^2 - V_{max}^2}{2(d - t_r V)}} \tag{3.9}$$

where V is the current vehicle speed, d distance to the summit of the bend and t_r the time-delay due to driver reaction. The purely geometric models (3.7) and (3.8) can be evaluated in real-time and can be used in a predictive way as the road data are already employed in the MPC strategy. Notice that these criteria do not handle the vehicle lateral dynamics. Thus, in our work these criteria are combined with other indicators on the lateral stability presented in the following section.
For lateral stability of the vehicle an additional condition is applied to improve the lateral motion. So, a desired longitudinal acceleration is calculated from physical limitation in braking with cornering.

$$a_x = \frac{\sqrt{(\mu g m)^2 - \sum (F_y)^2}}{m} \tag{3.10}$$

$$\sqrt{F_x^2 + F_y^2} < \mu F_z \tag{3.11}$$

In this way, a constrain on the longitudinal acceleration is imposed using the Kamm inequality, which keeps the forces developed in the tires within the physical limitations of the tire-road friction. Where, F_y can be either estimated or it can be measured using recently developed technology like smart tires or load sensing bearings to compute the a_x in real time.

The information on lateral dynamics is of capital importance as it helps to determine loss of control and help to preserve the lateral stability[**?**]. In this work, following criteria is used, which gives the β_{limit}:

$$\beta_{limit} < 10^{\circ} - 7^{\circ}\frac{(V_x)^2}{(40m/s)^2} \tag{3.12}$$

where, β is the sideslip angle of the vehicle and V_x is the vehicle speed.
The Reference Generation provides the lateral deviation and relative yaw angle to be minimized by the vehicle and a speed profile taking legal speed limits and vehicle comfort into account.

3.4 Control design

The MPC controller implemented in this thesis is based on the method of multiple-step optimization and feedback correction. Thanks to this method, the controller has good performances of control. Lateral control deals with the actuation of the steering of the vehicle to keep it in the center of the lane and follow the curved road. It is modeled as a reference path tracking problem for the MPC with the objective of minimizing the lateral deviation e_1 and relative yaw angle e_2. While, the longitudinal control deals with the actuation of the throttle/brake to control the longitudinal speed of the vehicle. It is modelled as a reference speed tracking problem, which is generated using the reference speed profile calculated using (3.12). Based on the reference velocity MPC computes the desired acceleration command to attain it. In other words, the objective of the MPC is to converge the speed of the vehicle to the desired reference speed. The inputs for the MPC are actual longitudinal velocity V_x, lateral deviation e_1 and relative yaw angle e_2, which are the outputs of the actual plant model. i.e. 3DoF rigid vehicle model. Based on these three inputs the MPC solves the optimization problem as reference tracking. The reference variables are given by reference velocity V_{ref}, while e_1 and e_2 are set equal to zero. So, The goal of the MPC controller is to compute the optimal steering angle and throttle/brake command to perform the autonomous driving. In order to achieve this goal, the controller calculates the steering angle and throttle/brake by minimizing its cost function. The description of the Adaptive MPC has been divided two parts:

- *Problem formulation* in which is explained how the MPC problem has been formulated;
- *Output prediction* in which is defined how the predicted output has been computed.

Problem formulation

The formulation of the MPC problem developed in this thesis starts defining a linear state-space model derived in section 2.4, which is represented as:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_d v(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3.13}$$

Where:

- A is the state matrix;
- B_u and B_d are the input matrices corresponding to inputs u and v respectively;
- C is the output matrix.

Given the linear model defined in equation 3.13, the Model Predictive Control algorithm is implemented as solving the following optimization problem at each time step:

$$\begin{aligned} \min_u J = &\sum_{j=1}^{N} ||y_p(k+j|k) - y_{ref}(k+j|k)||_{Q_y} + \sum_{j=0}^{M-1} ||u(k+j|k)||_{R_u} \\ s.t. \quad x(k+j+1|k) &= Ax(k+j|k) + B_u u(k+j|k) + B_d v(k+j|k) \\ x(k|k) &= x(k) \\ y(k+j|k) &= Cx(k+j|k) \\ |u(k+j|k)| &\leq u_{limit} \end{aligned} \tag{3.14}$$

Where u is the manipulated variable. Q_y and R_u are weights for outputs and manipulated variables respectively. This optimization problem refers to find the value

of input u that minimizes the sum of the weighted norms of the error between the predicted output vector y_p and the reference vector for those states y_{ref} and the input vector u for a defined prediction horizon N and control horizon M. The predicted output y has to satisfy the linear model, while the value of u should not exceed a specified limit u_{limit}.
The state vector y is given by:

$$\begin{bmatrix} V_x & e_1 & e_2 \end{bmatrix}^T$$

While, the state vector y_{ref} is given by:

$$\begin{bmatrix} V_{ref} & 0 & 0 \end{bmatrix}^T$$

V_x is directly taken from the vehicle dynamics block as an output while e_1 and e_2 are taken from the reference trajectory block. These three states are sent as feedback to the MPC controller in order to correct the control variables in the future step time with respect to the reference states.
The weighted norm of the vector $y = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^T$ corresponds to:

$$||y(k+j|k)||_{Q_y} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{3.15}$$

where the weights q_{11}, q_{22} and q_{33} are tuned to provide the needed damping on the corresponding output. The same definition is applied to the weighted norm of u given by:

$$||u(k+j|k)||_{R_u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.16}$$

Output prediction

The values of the predicted output $y(k+j|k), j = 1, 2, ..., N$, where N is the prediction horizon, have been computed using the linear state-space model described by the formula 3.13.
In particular, in order to make the computation, the following values have to be known:

- Present output measurement $y(k|k) = y(k)$;
- Applied input $u(k|k) = u(k)$;
- Entire set of predicted input values $v(k+j|k), j = 0, 1, 2, ..., N$.

If the prediction state is defined as follows:

$$\begin{aligned}
x(k+1|k) &= Ax(k) + B_u u(k|k) + B_d v(k|k) \\
x(k+2|k) &= Ax(k+1|k) + B_u u(k+1|k) + B_d v(k+1|k) = \\
&= A^2 x(k) + AB_u u(k|k) + AB_d v(k|k) + B_u u(k+1|k) + B_d v(k+1|k) \\
&\vdots \\
x(k+N|k) &= Ax(k+N-1|k) + B_u u(k+N-1|k) + B_d v(k+N-1|k) = \\
&= A^N x(k) + A^{N-1} B_u u(k|k) + A^{N-1} B_d v(k|k) + A^{N-2} B_u u(k+1|k) + \\
&A^{N-2} B_d v(k+1|k) + ... + B_u u(k+N-1|k) + B_d v(k+N-1|k)
\end{aligned} \tag{3.17}$$

The prediction output can be identified by the following equations:

$$\begin{aligned}
y(k|k) &= Cx(k) \\
y(k+1|k) &= Cx(k+1|k) \\
y(k+2|k) &= Cx(k+2|k) \\
&\vdots \\
y(k+N|k) &= Cx(k+N|k)
\end{aligned} \tag{3.18}$$

Using the equations 3.17 and 3.18, it is possible to express the predicted outputs $y(k+1|k), ..., y(k+N|k)$ as a function of the predicted inputs $u(k|k), ..., u(k+N-1|k)$, noted that the other signals are assumed to be known as stated above.

In order to make the relation between the equations 3.17 and 3.18 clearer, the prediction output of the future can be defined as follows:

$$Z(k) = Gx(k) + HU(k) + EV(k) \tag{3.19}$$

Where:

- *Z(k)* is the augmented vector of the predicted outputs;
- *U(k)* is the augmented vector of the computed future inputs;
- *V(k)* is the augmented vector of the predicted disturbances.

These vectors are obtained by the chaining of the input and the output vectors in the present time until the future N vectors (N - 1 vectors for the input u and v), and they are defined as follows:

$$Z(k) \equiv \begin{bmatrix} z(k|k) \\ z(k+1|k) \\ \vdots \\ z(k+N|k) \end{bmatrix}; U(k) \equiv \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \text{ and } V(k) \equiv \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N|k) \end{bmatrix}$$

The matrices G, H and E are determined in the following way:

$$G = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}; H = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB_1 & 0 & 0 & \dots & 0 \\ CAB_1 & CB_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{N-1}B_1 & CA^{N-2}B_1 & CA^{N-3}B_1 & \dots & CB_1 \end{bmatrix} \text{ and}$$

$$E = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB_2 & 0 & 0 & \dots & 0 \\ CAB_2 & CB_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{N-1}B_2 & CA^{N-2}B_2 & CA^{N-3}B_2 & \dots & CB_2 \end{bmatrix}$$

As mentioned before, the proposed control strategy maximizes the longitudinal speed while remaining in constrained speed range and without exceeding the adherence condition. At the same time, it eliminates the path error between the actual location and the desired path in terms of lateral deviation and desired yaw angle, assuring the handling stability during the motion.

This first section of the thesis is the extension work of the previous research, so the common terminology, equations and some section described in this chapter is used from the team members thesis literature [21],[22] and Matlab Documentation.

Chapter 4

Motion planning

In this chapter, we have discussed the technique adapted for Path planning and tracking control system. The method refers to the generation of a feasible trajectory to the future seconds with the information from the vision sensors. This system acts as a mind in an autonomous vehicle to make decisions to act visibly based on the road situation. Path planning mainly integrated with the control unit that initializes the reference data to tract the path which is safe and reliable. This system was mainly developed for mobile robots that are used for various applications, for instance, warehouse, space, humanoid, and industrial robots. To strengthen, the navigation system of the robot the department of mechatronics engineering gave us multiple research articles and papers to experiment in different forms. The idea behind all the ADS system is from the innovations by robotic engineers. The reference works are important information to breach the technicality of the engineering part. First, we look into the objective of the autonomous vehicle to further describe the procedure adopted in this part. The mission of this system is to navigate the vehicle from start to destination. To support this mission the system should capably define the collision-free path for a secure driving task, so to achieve it we need a high-level path planning strategy. In our case, we require certainly strong and robust system to define a highly efficient trajectory to cover up the lap distance and save time.

The path planner process is to define the path from the start to destination with a given map to navigate, likely the motion planning processes to execute the action defined in planned path.Figure4.1 presents the general hierarchy of the motion

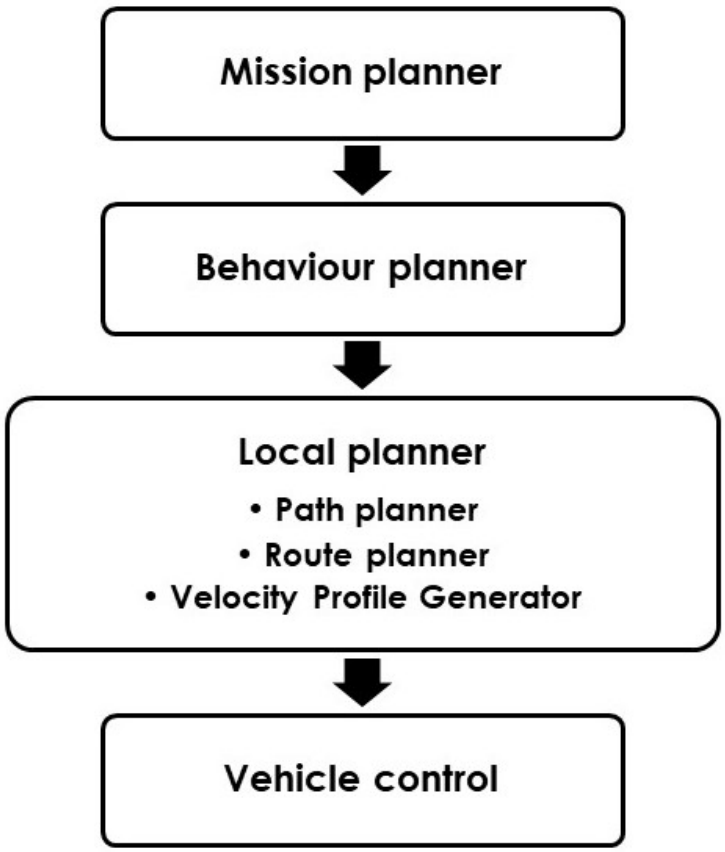


Figure 4.1: Hierarchical Path planning

planning of the system. First, the mission planner is the highest level planner that focuses on defining the right path and trajectory to navigate the AD vehicle. Second, the Behavioral planner decides whether the planned path is safe enough to follow, and this can be implemented in three different ways, they are a. Finite state machines (composed of states and transitions), Rule-based system(by using the hierarchy of rules, by evaluating logical predicate), Reinforcement learning. Third, the Local planner is responsible to generate feasible path adhering to all conditions. It is decomposed into path planner, route planner, velocity profile generation. Finally control model to initiate the command for respective maneuvers and to track, correct the deviations errors.

Engineers have developed multiple path planner algorithms based on the problem definition, feasibility, complications, model-based, computational complexity. They

are classified into three major classes,

- Variational Method
- Graph searched Methods

  Cell Decomposition

  Visibility Graph

  sampling based road map construction

  Tree of motion primitives

- Incremental-search Methods

  RRT, Rapidly exploring Random Trees

  RRT*, Optimal Rapidly exploring random Tree

The Variational method is a nonlinear optimization technique, where its parameters are optimized in terms of a cost function, and the path is usually represented as a spline. The method allows optimizing the position control points based on geometry and optimal control. To avoid collision locally the algorithm tries to find the gradient in the generated map. It is an efficient and widely applicable method to adapt, but it is incomplete in local convergence.The graph search methods discretize the possible configuration space of the car and use heuristic information to search for a cost-optimal path through the graph. the best-known example of this type is Dijkstra. It has been improved by the A* star family. The Incremental search methods are similar to the graph search method but are based on a random sampling of the configuration space. the sample is connected by an algorithm. the cost-optimal path to the target point, therefore, continues to improve the longer algorithm runs. The widely known approach is RRT and RRT* [23].
In this thesis, we used an incremental search method (RRT algorithm) to plan the path and define motion control. RRT (Randomly exploring the random tree) is a sampling-based method. The idea of this approach is the incremental growth of the tree rooted at the initial configuration to explore the reachable region of the configuration space, and it is briefly explained in section 4.2. The environment model (mapping) is designed by using the reference data that are extracted from the simulated scenario. We used the scenario creator from the Matlab toolbox to

create the path(track) and extract its way points, that is left and right lane, so the execution of planning strategy is done with the predetermined reference data, in other words, it is an offline process with accumulated information from vision sensors. The velocity profile generator and path planner gives the input to the control system to initiate the driving command for the vehicle model. we used a simplified vehicle model for validating the control architecture. The programming script of the automated parking valet example model in Matlab and Simulink used helped to formulate the control model for our application, and the procedure adopted for developing plant mode is described briefly in section 4.1.

4.1 Procedure Overview

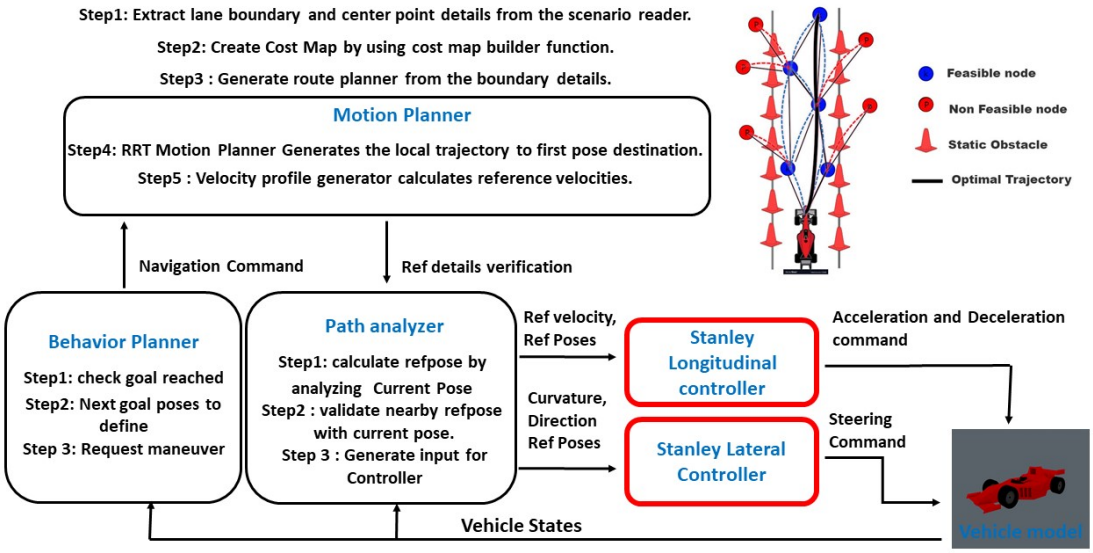


Figure 4.2: Overall Procedure of Motion planning

4.2 Mapping

The map is the base data for any ADS(Autonomous Driving system), it enables us to have the information about the surrounding environment from sensors in the format two dimensional or three-dimensional. In our experimentation analysis, we

have used synthetic data generated by the driving scenario simulator in Matlab to create the occupancy grids, and they represent the environment as a discrete grid. In robotic algorithms such as path planning, are used in mapping applications for integrating the sensor data, in path planning for finding obstacles, and for localizing robots in a known environment. This occupancy grid map can be created in two ways by

- Binary occupancy grid
- Probability occupancy grid

A binary occupancy grid uses true values and false values to represent the occupies space and free space. It also verifies whether our vehicle would collide with an obstacle. This type is widely used to reduce storage capacity because it uses less memory space. Next, the probability occupancy grid uses cost values to provide a more detailed map representation and the value close to 1 represents the probable occurrence of an obstacle, similarly, the value close to 0 represents the probable occurrence of no obstacle, that is free space.

Figure 4.2 depicts the coordinate system of the map, where the absolute reference

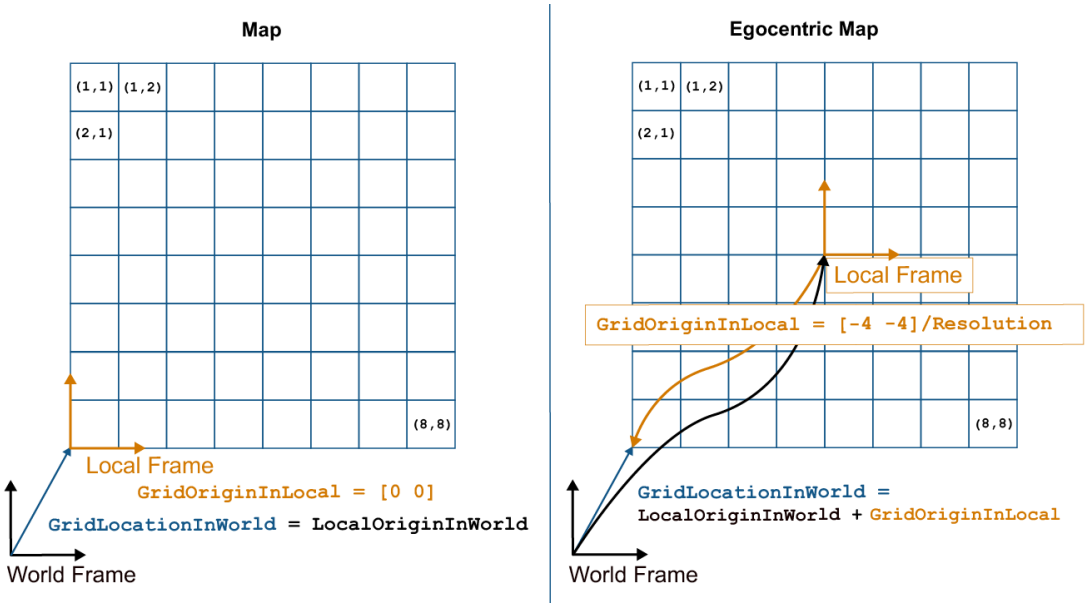


Figure 4.3: Map coordinate system
[24]

frame is the vehicle operating frame(world frame), and it is the coordinate system

with a fixed origin. Secondly, the local frame refers to the egocentric frame for vehicle navigation. The two function GridLocationInWorld, LocalOriginInWorld defines the initial, and location of the local frame relative to the world coordinates [24]. Occupancy grid map also uses the inflation function to inflate the grids to develop a buffer zone for the safety of the vehicle and obstacle, and it is briefed fully in section 4.3. For this experimentation, we have created three different tracks using the occupancy grid map functions. They are,

- Round track
- Handling track
- Berlin race track

The functions that define the map are Predetermined (x,y) coordinates of the track boundaries, Map width, Map length, Cost val, Cell size, Free value, and occupied value. Figure 4.3, 4.4, 4.6 represents the above-stated three different map.

4.3 Motion Planner

Motion planning is one of the challenging parts of ADS (Autonomous driving system) as it is responsible for providing a highly safe trajectory to follow in terms of Dynamic driving condition, Environment conditions (static and dynamic obstacles), and scenario transport regulations or rules. RRT (Rapidly exploring random tree) is an incremental sampling-based algorithm developed by Steven M.Lavalle and James J.Kuffner Jr, and it is an effective approach to computationally hard motion planning, and also grabbed a lot of attention through research papers since its introduction because of its model integrity, complexity and feasibility to provide the optimal solution. The core idea of sampling-based planning is mainly it randomly samples the states defined in state-space and try to connect those states considering the driving limits and free space or without an obstacle interference. It ensures, the complete connectivity between the start to the goal destination and it tries to keep finding feasible path until and unless it reaches the destination point, and so it is said to be probabilistically complete. In this section, we present the problem definition of the planner and the approach for collision check and avoidance[25].

4.3.1 Planner description

The algorithm is based on optimal rapidly exploring Random Tree, and it explores the environment around the vehicle by placing the states in the collision-free areas under stated constrains to fetch the trajectory from given initial state to the final state(Goal). Planner properties that define the entire algorithm,

- 'GoalTolerance' is set to 0.5(x_{tol}, y_{tol},θ_{tol}) to approximate the final goal position concerning world frame.
- 'Goal Bias' is the probability value of selecting the goal pose at each iteration as opposed to random pose. Larger value accelerates reaching the goal at the risk of failing to circumnavigate obstacle.
- 'Connectionmethod' is used to calculate geometrical values between two connecting states. Dublin's or reeds-sheep is the method that computes the path between two random states as a sequence of primitive motions.
- 'Connectiondistance' is to calculate the distance between two states, and for larger distance result in long path segments
- 'MinTurningRadius' is the radius of curvature at maximum steer concerning world coordinates, and for larger values, it limits the maximum steering angle used by the planner (35degree)
- 'MinIterations' is the minimum number of steps initiated to connect the probable sates in the free space
- 'MaxIterations' is the maximum number of steps initiated to connect the probable sates in the free space
- 'ApproximateSearch' it uses the near states to compute the path faster(when it is set to be true), In case of false, it will compute slowly that could cost us computational cost[26].

4.3.2 Problem definition

The problem is defined to the system with certain conditions,firstly the path should be planned within the track boundary, Secondly the probable poses should not lie on

the obstacle region, and thirdly it should take less computational time. To satisfy this condition formulation is done by the employing th optimal kinodynamic motion planning problem[25].It is considered to be a dynamical system,

$$\dot{x}(t) = f(x(t), u(t)), x(0) = x_0 \tag{4.1}$$

where $x(t) \in X, u(t) \in U$, the function X and U are called trajectories and controls for all$t \in [0, T]$. Given the domain X, obstacle region X_{obs}, goal region X_{goal} and a smooth function defines the system dynamics, with domain [0,T] for some $T \in R_{\geq 0}$.

- $x(t) \in X_{free}$ for all $t \in [0, T]$ - avoids obstacle
- $x(t) \in X_{goal}$- reaches the goal region
- $J_x = \int_0^t g(x(t))dt$, where this cost function is assumed to be line integral of Lipschitz continous function $g : X \to R_{\geq 0}$, and g is bounded away from X^n - minimizes the cost function

4.3.3 Algorithm description

This algorithm aims to satisfy the about mentioned problems by fining probable optimal path to feed to controller for tracking. This is done in real time process so it define the path to stated $X_{ninitial}$ to X_{ngoal}. n refers to the number of route points given to the planner. The n_{goal} are determined by the route planner that uses the estimated center reference poses. Every n_{goal} is calculated to be in the distance of $i_{30,50,70,90,110....n}$ points respectively to each destination points. The algorithm first extends the nearest vertex towards the sample.The trajectory that extends the nearest vertex z_{near} towards the sample is denoted as x_{new}.The final state on the trajectory x_{new} is denoted as z_{new}. If x_{new} is collision free, z_{new} is added to the tree and its parent is decided as follows. First, the Near Vertices procedure is invoked to determine the set z_{nearby} of near-by vertices around z_{new}. Then, among the vertices in z_{nearby}, the vertex that can be steered to z_{new} exactly incurring minimum cost to get to z_{new} is chosen as the parent. Once the new vertex z_{new} is inserted into the tree together with the edge connecting it to its parent, the extend operation also attempts to connect z_{new} to vertices that are already in the tree. That is, for any vertex z_{near} in z_{nearby}, the algorithm attempts to steer z_{new} towards z_{near}, if the

steering procedure can exactly connect z_{new} and z_{near} with a collision-free trajectory that incurs cost less than the current cost of z_{near}, then z_{new} is made the new parent of z_{near} ,the vertex z_{near} is "rewired"[25],[27],[28]. Once the the kinematic path is found out to X_{ngoal} the path smooth spline function smooth down created path with discretize poses and distance each initial and goal poses. This function help in rectify the discontinuities in the curvature where it joins with next reference point. The algorithm interpolates a parametric cubic spline that passes through passes through all the input reference pose points.It also makes sure the tangent direction of the output path matches with the vehicle orientation respect to each initial and goal poses.

4.3.4 Collision check method

The collision check is the verification of the vehicle pose lies on inflated area of the static obstacle. The algorithm works in following steps,

- Calculate the inflation radius, in world units, from the vehicle dimensions. The default inflation radius is equal to the radius of the smallest set of overlapping circles required to completely enclose the vehicle. The center points of the circles lie along the longitudinal axis of the vehicle. Increasing the number of circles decreases the inflation radius, which enables more precise collision checking.

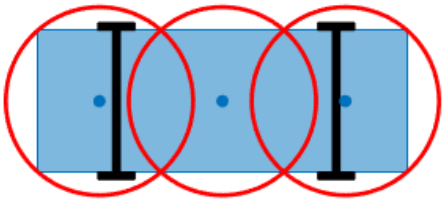


Figure 4.4: Overlapping circles for collision check

- Convert the inflation radius to a number of grid cells, R. Round up non-integer values of R to the next largest integer.

- Inflate the size of obstacles using R. Label all cells in the inflated area as occupied.

- Check whether the center points of the vehicle lie on inflated grid cells. If any center point lies on an inflated grid cell, then the vehicle pose is occupied. The check Occupied function returns true. An occupied pose does not necessarily mean a collision. For example, the vehicle might lie on an inflated grid cell but not on the grid cell that is actually occupied.If no center points lie on inflated grid cells, and the cost value of each cell containing a center point is less than Free Threshold, then the vehicle pose is free. The checkFree function returns true.If no center points lie on inflated grid cells, and the cost value of any cell containing a center point is greater than Free Threshold, then the vehicle pose is unknown. Both check Free and check Occupied return false[29].

4.4 Velocity Profile Generation

The Velocity profile is generated from the reference trajectory give by the motion planer. The distance and curvature profile of the reference path helps us to generate the reference velocity. The profile concerns about the dynamic properties of longitudinal motion and lateral motion, that is for the straight path it define the reference speed based on Maximum longitudinal acceleration,speed, jerk and for the curvature segment it limits maximum velocity concerning the lateral acceleration threshold.

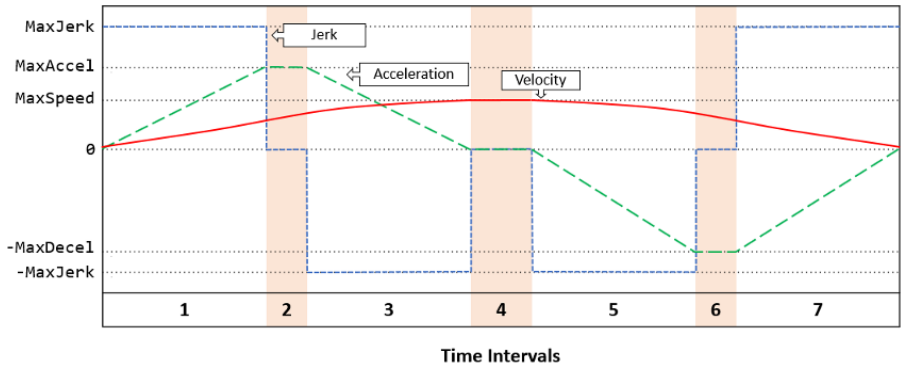


Figure 4.5: Internal curves of velocity Profiler

Overall speed profile is generated considering three family of curvature.

- When path Curvature profile is circular arc or pure clothoid, the minimum speed V_{min} is fixed by the curvature bound k_{max} and the maximum lateral

accelerationa_y. Therefore kinematic model equation is given by,

$$V_{min} = \sqrt{\frac{a_y}{K_{max}}} \tag{4.2}$$

- smooth transition between the minimum value to maximum value and return back to its minimum allowed speed satisfies the acceleration and jerk constrains.

- A set of two transition path are considered to be single in the straight line when the it goes zero to maximum speed or vice versa
  The block used in the Simulink corresponds to calculate the speed profile of the generated path adhering to stated condition.The inputs variables from the smooth spline profiler to velocity profiler is cumulative length of path segments (l), direction , curvature (k), start velocity V_{start}, and end velocity V_{end}. Finally output take out of the block is reference velocities of each respective reference poses[30].

4.5 Behaviour planner

Behaviour planner is the higher order system which initiate the propagate confirmation to vehicle considering the condition of the surrounding environment. The main responsibility of this planner is to define safe driving mission under different driving environments, that is it should act as subset behaviour of real driver action in case of various difficult situations. Firstly the planner main considerations are Rules and regulation of the road(i.e,traffic signals, left or right hand driving, lane markings,etc.), Static obstacle around the vehicle, and dynamic objects around the vehicle. During the driving maneuver it keeps track of the current speed, follow the front vehicle with safe headway, ready for acting in case of emergency situations, analyze the surrounding to initiate the lane change, and to remain stopped when it is parked.
Behaviour planner block is specified in the planning model to perform certain task. The input requirement for the planner is information of the scenario. In our case,the scenario features are the lane boundary that is inflated to certain radius in vehicle cost map, direction to lead the vehicle, length of the path, and curvature details.The

inflated area are considered to be static obstacle. The main aim in this model is to complete the planning work for entire track that means global planning, that is each local planner will be updated as soon as it reaches to first destination. The goal checker algorithm is the part that provides information to the behaviour planner to give initialize navigation command to motion planner to define next local trajectory. help behaviour planner function in Matlab is used to achieve the stated requirement and they are defined based on achieving three major task,

- Request next maneuver along the path
- Check if destination has been reached
- Command to request re-plan if the path is inconsistent to comply with the constrain.

4.6 Control design

The controller of ADS is mainly responsible for providing three input commands (Acceleration, deceleration and steering command) to perform the maneuver autonomously,to adjust values depending on the given conditions, to propel the vehicle with high safety, to make sure it follows the reference trajectory, and finally to provide smooth transition during the maneuver. For the purpose of analysis we deployed a decoupled control strategy to handle motions related to lateral and longitudinal. Stanley control law is used to govern motions concerning lateral, longitudinal dynamics. First, the strategy to control the vehicle speed is presented in section (6.1), second, the strategy to control the lateral and heading errors of the vehicle is presented in section (6.2).

4.6.1 Longitudinal control

The term "longitudinal controller" is typically used in referring to any control system that controls the longitudinal motion of the vehicle, for example, its longitudinal velocity, acceleration or its longitudinal distance from another preceding vehicle in the same lane on the highway. The throttle and brakes are the actuators used to implement longitudinal control [32]. The well known vehicle speed control model is cruise controller, and it developed to regulate the speed respect to the heading

distance. The strategy is subdivided into two controller that is termed as upper and lower controller, where upper is responsible to given desired acceleration concerning the reference velocity, and lower deals with the calculation of desired throttle input. Similarly in this analysis we used and simplified model to determine the desired acceleration for the vehicle from given speed reference, that is the reference speed calculated from the reference path is given as a input to profiler to generated desired command to follow the trajectory, and regulate it during turning maneuver. The longitudinal control law is given as a discrete proportional integral(PI) controller with integral anti-windup. The block used in the plant model from Simulink is PID controller described by the Anti-Windup method. The block equation is given by

$$u(k) = (K_p + K_i \frac{T_s z}{z-1}) e(k) \tag{4.3}$$

where,

- u(k) - Control signal of the k_{th} time step

  It determines the value of acceleration and deceleration command, and the block saturates the commands to respective ranges of [0,M_A] and [0,M_B] where,

  M_A - Maximum longitudinal acceleration $(\frac{m}{s})^2$

  M_B - Maximum longitudinal deceleration $(\frac{m}{s})^2$

- K_p - Proportional gain
- K_i - Integral gain
- T_s - Sample time (s)
- e(k) - Velocity error

$$(V_{k_{ref}} - V_{k_{current}}) \tag{4.4}$$

$$e_v(k+1) = K_{p,v}(v(k+1) - v_c(k+1)) + K_{i,v} e_{int}(i+1) \tag{4.5}$$

$$e_{int}(k+1) = e_{int}(k+1) + (v(k+1) - v_c(k+1)) \tag{4.6}$$

where k - time step of each iteration, $K_{p,v}$,$K_{i,v}$ values determine best trade off between disturbance rejection and e_{int} integral term is saturated to higher value to prevent the windup[32].

4.6.2 Lateral control

Lateral refers to the motion in Y direction, and the control refers to better handling the vehicle in that direction. This controller is responsible to provide proper steering command to track the reference planned path. For the purpose we adapted the lateral Stanley control law is implemented to determine required steering and reduce the tracking error considering lateral dynamics of the vehicle. The control law is defined using kinematic relation of the vehicle, that is it uses the center point of the front axles as the vehicle reference pose to track the error in heading direction and the position error relative to reference path(cross track error). The steering law is defined to rectify three major errors,

- to correct heading error
- to correct position error
- to limit the maximum steering angle according to bound conditions.

Steering law

Considering the kinematic equation of motion the steering law is given by

$$\delta(t) = \psi(t) \tag{4.7}$$

Desired steering angle is calculated to align the heading angle relative to required heading angle, in other words it is generated proportional to heading error.

$$\delta(t) = \arctan(\frac{K_e(t)}{V_f(t)}) \tag{4.8}$$

Desired steering angle is generated to reduce the cross track error or eliminate it. Consideration for calculating the angle is based on the proportional value of error

and that is inversely proportional to speed,the limit is effective for larger errors with arctan, and gain K_e is determine experimentally(trade of between rise time and overshoot).

$$\delta(t) \in [\delta_{min}, \delta_{max}] \tag{4.9}$$

Desired value is chosen between the maximum and minimum steering limit. Combing three relation the Stanley control law is given by

$$\delta(t) = \psi(t) + \arctan(\frac{K_e(t)}{V_f(t)}), \delta(t) \in [\delta_{min}, \delta_{max}] \tag{4.10}$$

two main problem addressed by the law is when large heading error is occurring the larger steering correction is required, and when the value is beyond the fixed limit, then it is assumed there is no cross track error. Next is when there is large cross track error, steering angle is generated sufficiently to meet the reference path, but the heading angle also changes with steering angle, so the heading correction counteracts the cross track correction and drives the steering angle back to zero. As soon as it nears the path the cross track error drops and steering command tries to correct heading alignment.For small steering angle the dynamics is given in (4.13), and for small cross track errors it leads to exponential decay characteristics (4.14) [32].

$$\arctan(\frac{K_e(t)}{V_f(t)}) \approx \frac{\pi}{2} \tag{4.11}$$

$$\delta(t) \approx \psi(t) + \frac{\pi}{2} \tag{4.12}$$

$$\dot{e}(t) = -v_f \sin(\psi(t) - \delta(t)) \tag{4.13}$$

$$\dot{e}(t) \approx -K_e(t) \tag{4.14}$$

Chapter 5

Results and Discussion

In this chapter the simulation procedure of both control model are presented with simulation results and discussed in details about the performance of the system.

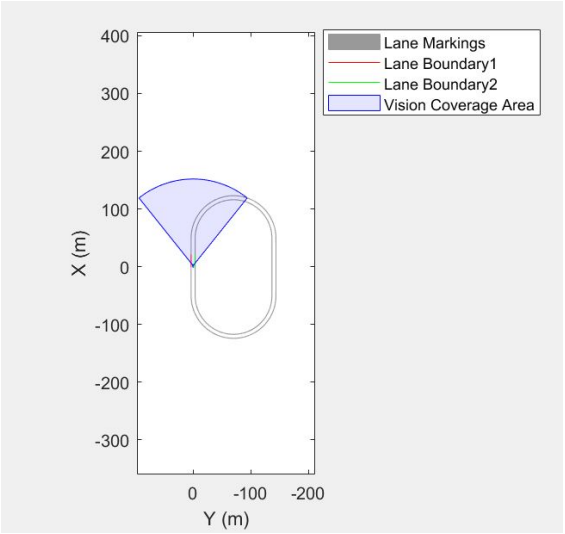
5.1 Lane following

5.1.1 Simulation Setup

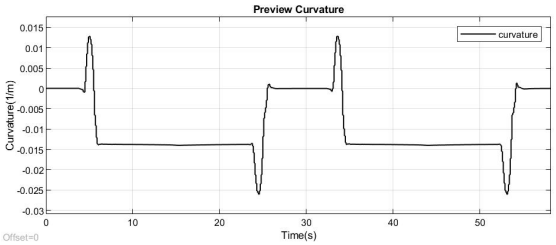
The simulation environment is build using MATLAB and Simulink. The race track scenarios are designed in *Drivingscenariodesigner* using the reference data and the lane and sensor features are added to ego vehicle (actor) in the interface to commence the experimentation. The track details are,

- *Roundtrack* It is the simple closed loop track, total length of track is calculated considering center line that is around 639.5(m). From the center line the lane is established on the two sides with the width of 7(m). The ego vehicle is positioned at the start point of the center coordinate.Figure5.1 shows the initial state of the vehicle, where the vehicle positioned at $[x_{initial}\ y_{initial}\ yaw_{initial}]$, The blue are depicts the vision coverage area and the boundaries left and right are identified by the system with the color code red and green.

- *Handlingtrack* It is complex closed loop track with multiple curves designed to test the handling features of the vehicle, total length of the track is around 623.2(m), and the track with is kept the same as round track. Figure5.1 shows



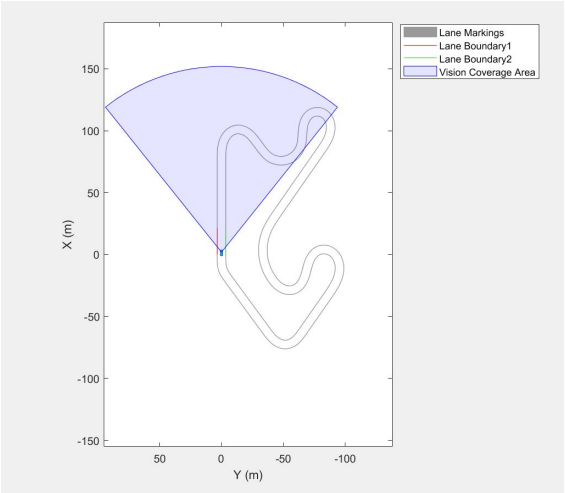
(a) Bird eye plot



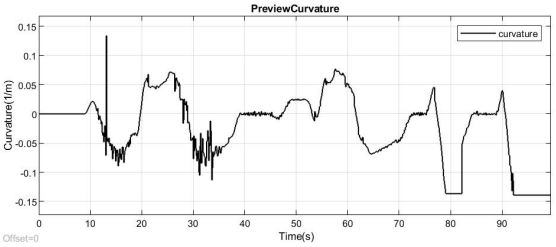
(b) Curvature

Figure 5.1: a.Designed closed loop round track scenario,b. detected curvature k

the initial state of the vehicle, where the vehicle positioned at $[x_{initial}\ y_{initial}\ yaw_{initial}]$, The blue are depicts the vision coverage area, and the boundaries left and right are identified by the system with the color code red and green.



(a) Bird eye plot

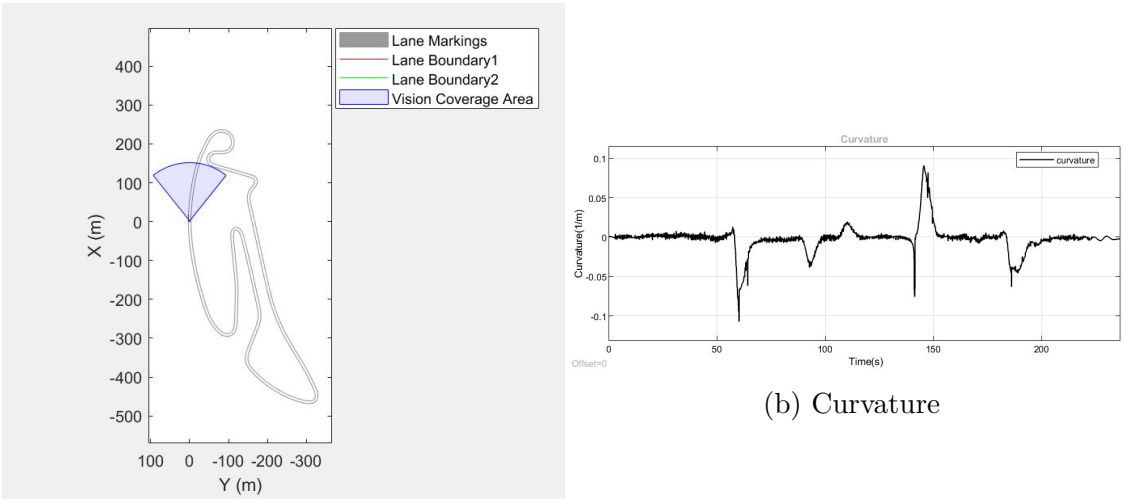


(b) Curvature

Figure 5.2: a.Designed closed loop Handling track scenario,b. detected curvature k

- *Handlingtrack* It is the benchmark of the real racing track and the track

length is around 2386(m). The track width is 7m(m) constructed from the reference center line.Figure5.1 shows the initial state of the vehicle, where the vehicle positioned at $[x_initial\ y_initial\ yaw_initial]$, The blue are depicts the vision coverage area, and the boundaries left and right are identified by the system with the color code red and green.



(a) Bird eye plot

(b) Curvature

Figure 5.3: a.Designed closed loop Berlin race track scenario,b. detected curvature k

the configuration of the ego vehicle (actor) are,

- $m = 1575$ kg, the total vehicle mass;
- $Iz = 2875$ Nms^2, the yaw moment of inertia of the vehicle;
- $l_f = 1.2$ m, the longitudinal distance from the center of gravity to the front wheels;
- $l_r = 1.6$ m, the longitudinal distance from the center of gravity to the rear wheels;
- $C_{\alpha f} = 19000$ N/rad, the cornering stiffness of the front tires;
- $C_{\alpha r} = 33000$ N/rad, the cornering stiffness of the rear tires.

5.1.2 Scenario 1 - Roundtrack

The simulation results of round track are presented in the below figures, where the first is the velocity profile graph that depicts the the reference velocityV_{ref} is followed by the measured velocityV_x. According to the track scenario and experimentation the maximum saturation speed is fixed to 12 (m/s) that is 43(km/hr) above this value the vehicle tends to enter the stable region.The construction of the track is with two large radius of curvature segment where the velocity is maintained almost constant. It is evident from the result that during the curvature detection there is sudden drop in velocity to produce antiquated force to generate proper cornering forces to turn the vehicle. In the second figure steering command generated by the MPC is reported. Scale factor mentioned in the constrain is from [max($\frac{\pi}{6}$),min(-$\frac{\pi}{6}$)].we can see that there is sudden change in steering angle at the curvature detection entry and exit point. e_1 & e_2 are Lateral deviation error and relative yaw angle error reported in the third and fourth figure. The range of deviation from the reference is at the admissible range of ±0.25, whereas for yaw deflection are seems to be at the minimum level during the straight and constant curvature road profile, while at the curvature detection point it defects from the range of ± 0.25 (rad). The optimal constrains and weights are added to MPC controller to achieve complete tracking of track. To complete the track at maximum velocity of 40(km/h) it took the simulation time of 62.2(s), where the constrains are,

- *S*ample Time= 0.1
- *P*rediction horizon = 20
- *C*ontrol horizon = 10
- *A*cceleration(MV1) = 8
- *S*teering angle(MV2)= 30
- *A*cceleration(Weights.MV1Rate) = 5
- *S*teering angle(Weights.MV2Rate)= 0.5
- *L*ogitudinal Velocity(Weights.OV1) = 2

- *L*ateral Velocity(Weights.O2) = 1
- *R*elative Yaw angle(Weights.OV3) = 0

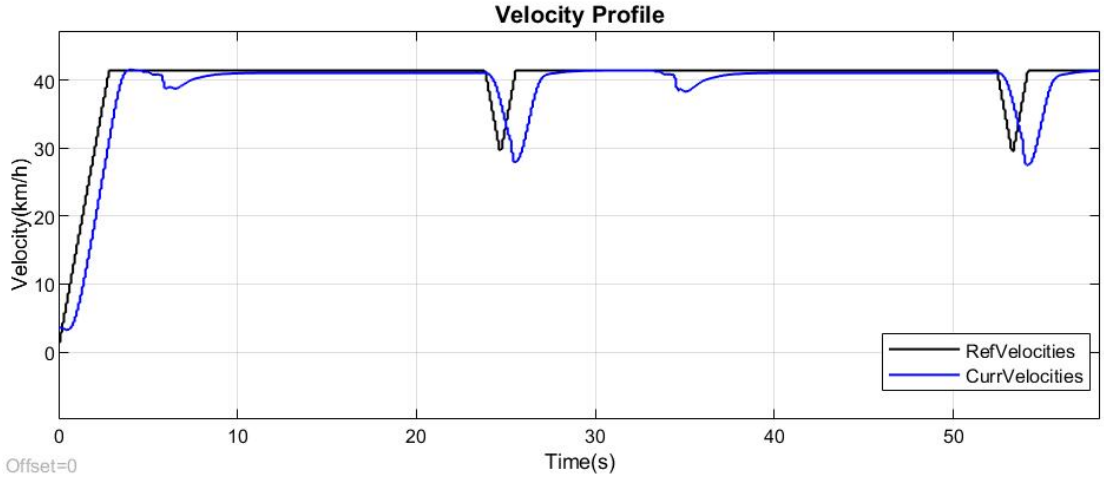


Figure 5.4: Meaured longitudinal velocity V_x vs Reference velocityV_{ref}

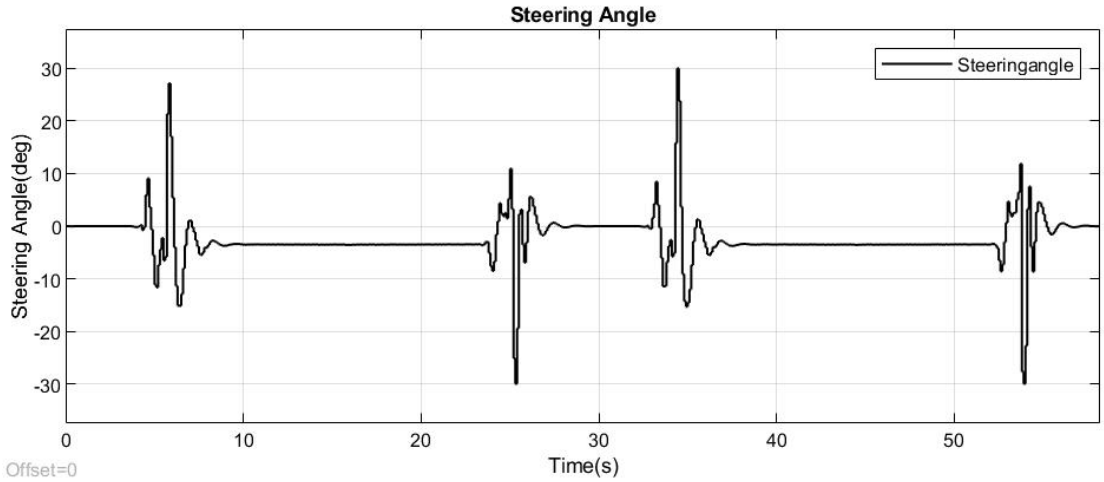


Figure 5.5: Steering angle δ command generated by MPC

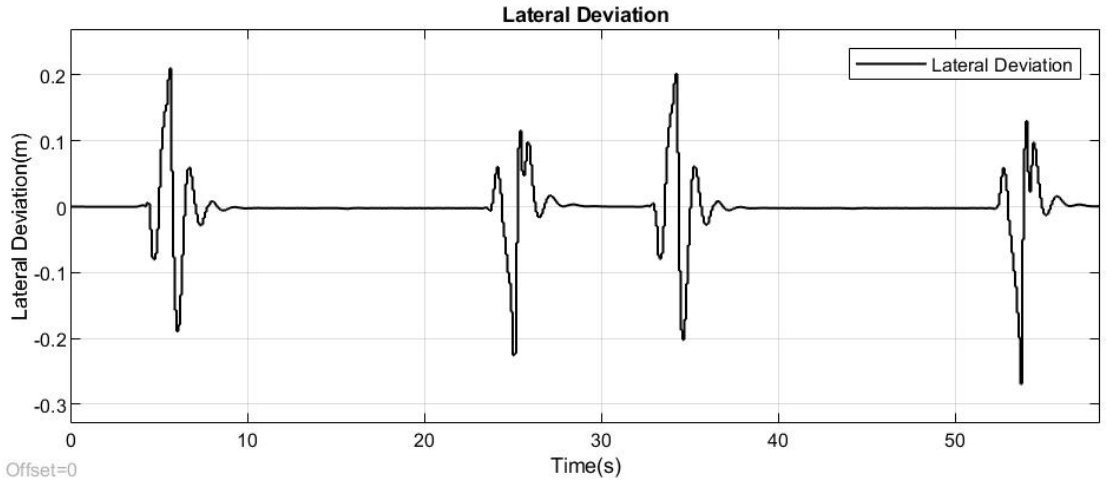


Figure 5.6: e_1-Lateral deviation

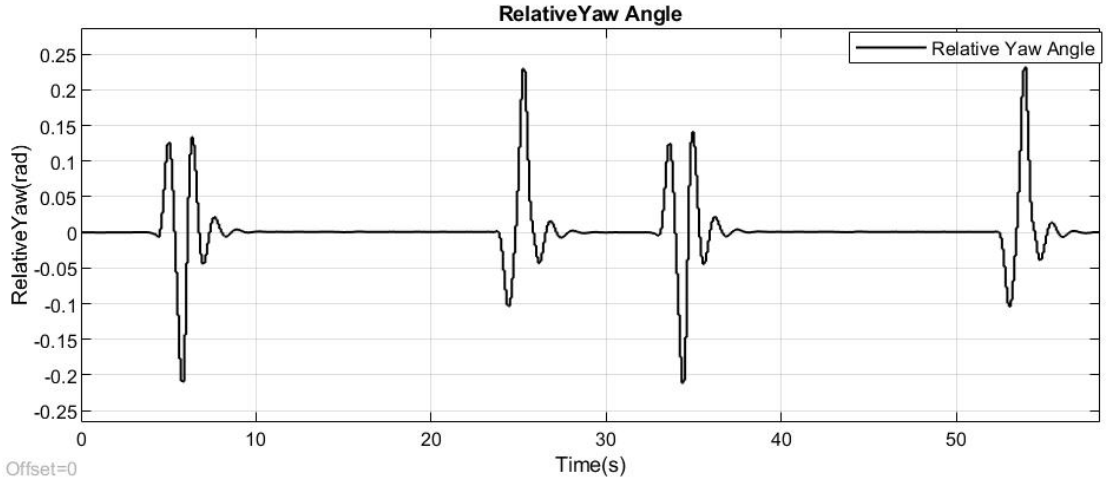


Figure 5.7: e_2-Relative yaw angle

5.1.3 Scenario 2 -Handling track

The simulation results of Handling track is presented in this section. The first graph follows with the velocity profile, where the measured velocity V_x with respect to reference velocity V_{xref}. The maximum speed is scaled to 8(m/s) that is 28.6 (km/h), this value is finalized after performing sensitivity analysis.If we scale above this value at certain point of curvature it fails to hold up the tracking and becomes unstable and leaves out of the track. As this track is designed to check the controller performance in providing acceleration, deceleration and steering command to follow the track even in sharp turns.It is evident from the result it reduce its velocity while cornering to keep the vehicle in adherence limit and it maintains constant scaled velocity at straight road path. Second is the steering angle reported in the figure, it is clear from the graph the steering angle generated to keep the vehicle under certain limit that is in range between[max($\frac{\pi}{6}$),min(-$\frac{\pi}{6}$)], and it generates maximum angle at peak deviation from respected reference profile. e_1 & e_2 are Lateral deviation error and relative yaw angle error reported in the third and fourth figure, where the lateral deviation reaches it scale parameter when the actor tries to cover steep curvature profile of the road the value ranges from $\pm$0.5(m), sudden shift is when adequate steering command to estimate the center line to follow.Relative yaw angle changes constantly to match the current vehicle state orientation with respect to the reference predicted state orientation. The value range in the limit of $\pm$0.6 (rad).The optimal constrains and weights are added to MPC controller to achieve complete tracking of track. To complete the track at maximum velocity of 28(km/h) it took the simulation time of 101.8(s), where the constrains are,

- *S*ample Time= 0.1
- *P*rediction horizon = 20
- *C*ontrol horizon = 15
- *A*cceleration(MV1) = 8
- *S*teering angle(MV2)= 30
- *A*cceleration(Weights.MV1Rate) = 1.5
- *S*teering angle(Weights.MV2Rate)= 0.3

- *L*ogitudinal Velocity(Weights.OV1) = 7.5
- *L*ateral Velocity(Weights.O2) = 4
- *R*elative Yaw angle(Weights.OV3) = 0.4

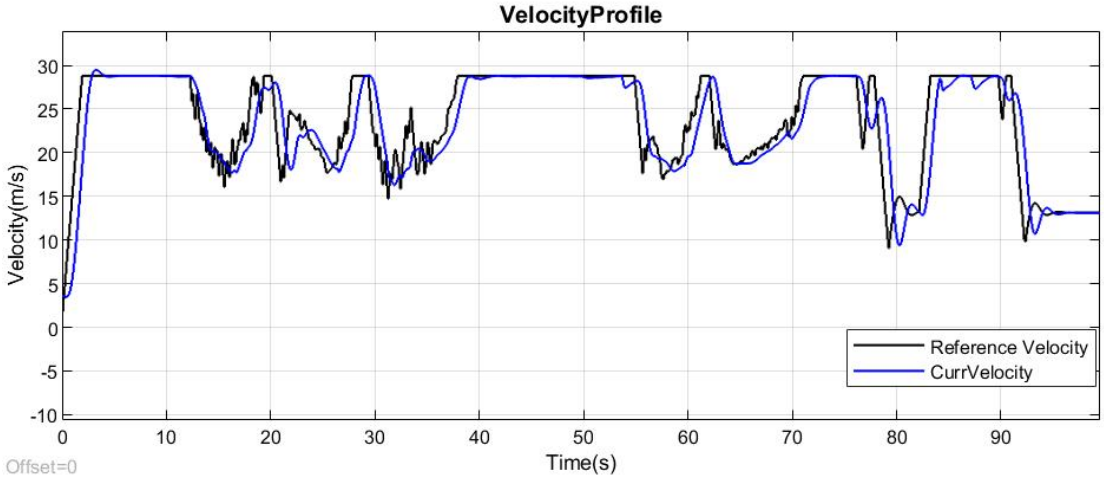


Figure 5.8: Meaured longitudinal velocity V_x (blue) vs Reference velocityV_{ref}

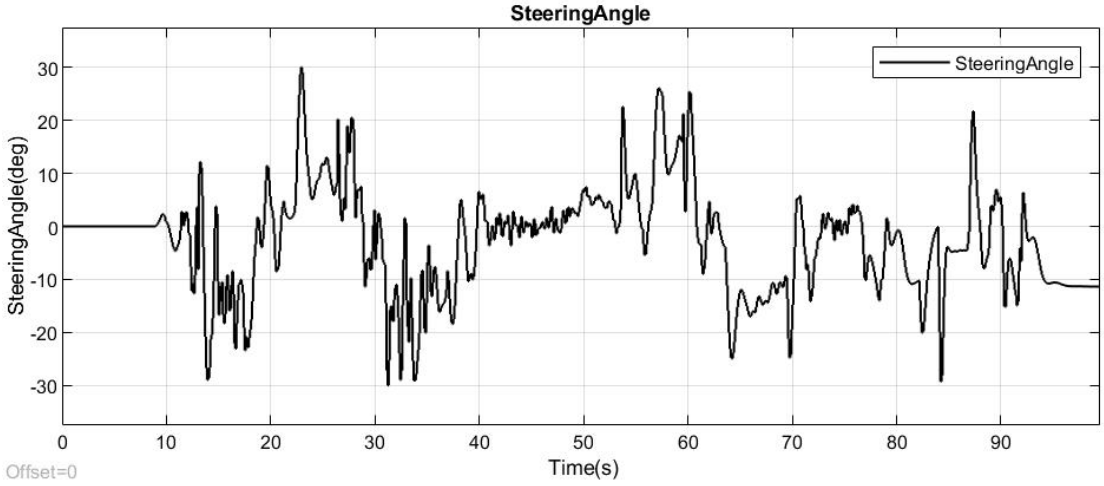


Figure 5.9: Steering angle δ command generated by MPC

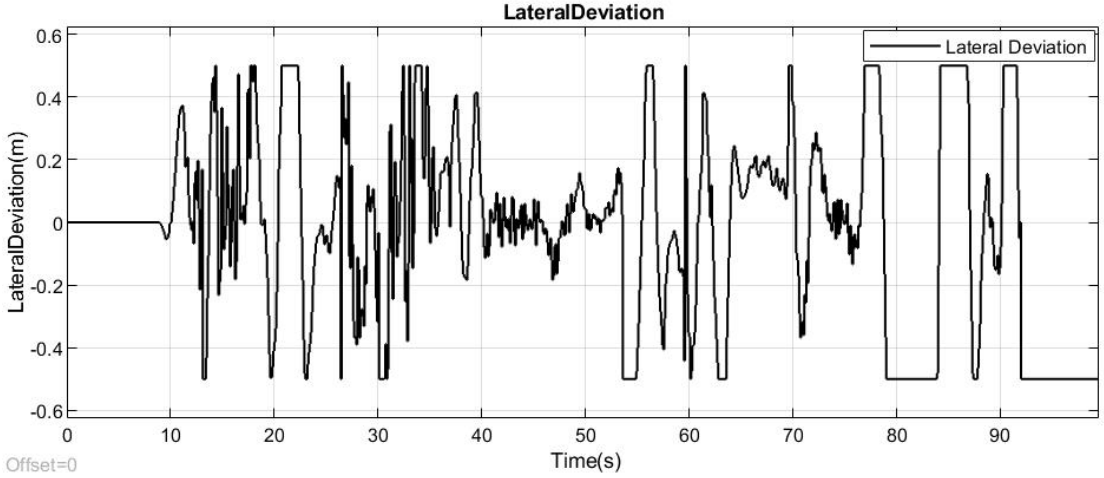


Figure 5.10: e_1-Lateral deviation

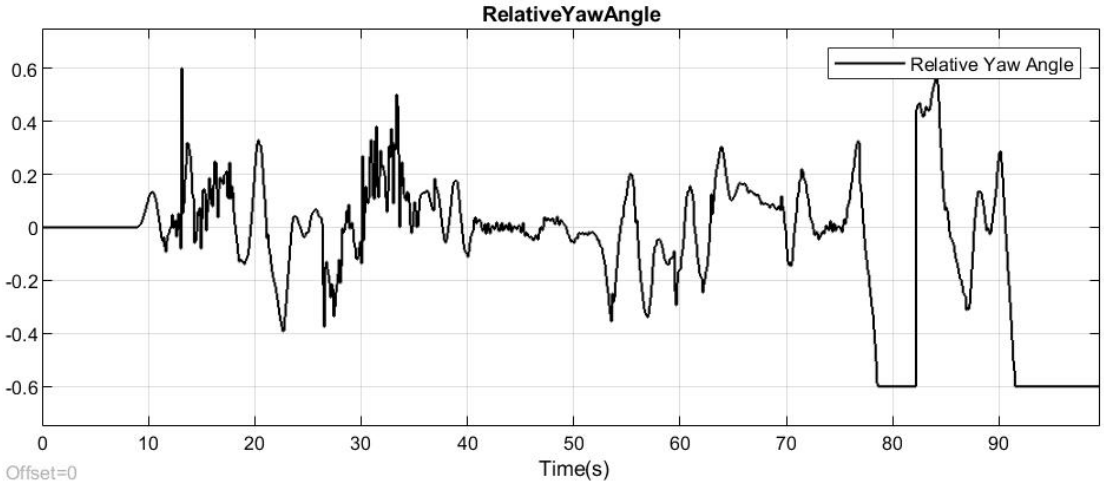


Figure 5.11: e_2-Relative yaw angle

5.1.4 Scenario 3 -Berlin Race track

The simulation results of Berlin race track is reported in below graphs. This track is the longest compared to other and it is used to check the robustness of the control for higher order of values. The first graph follows with the velocity profile, where the measured velocity V_x with respect to reference velocity V_{xref}. The maximum speed is scaled to 25 (km/h), that is very low compared to other scenario scale factor, reason for the limit is that the prediction model does not reduce its velocity when it detects suddenly the curvature profile as consequence it exits the bound condition and becomes completely unstable. The graph depicts that the steep reduction of velocity during the cornering and maintains it constant stated value. Second is the steering angle reported in the figure, it is clear from the graph the steering angle generated to keep the vehicle under certain limit that is in range between[max($\frac{\pi}{6}$),min(-$\frac{\pi}{6}$)], and it generates maximum angle at peak deviation from respected reference profile.The constant steering correction is evident from the graph at an angled profile path. e_1 & e_2 are Lateral deviation error and relative yaw angle error reported in the third and fourth figure, where the deviation are the extend value because of steep curvature that this system could not reduce the velocity prior to the curvature and the aggressive control action tries to limit its maneuver and pushes towards set point value that ranges from ±0.5(m) for the lateral deviation and ±0.6(m)for the yaw angle deflections.The optimal constrains and weights are added to MPC controller to achieve complete tracking of track. To complete the track at maximum velocity of 25(km/h) it took the simulation time of 230.6(s), where the constrains are,

- *S*ample Time= 0.1
- *P*rediction horizon = 20
- *C*ontrol horizon = 15
- *A*cceleration(MV1) = 8
- *S*teering angle(MV2)= 30
- *A*cceleration(Weights.MV1Rate) = 3
- *S*teering angle(Weights.MV2Rate)= 0.5

- *L*ogitudinal Velocity(Weights.OV1) = 8
- *L*ateral Velocity(Weights.O2) = 2.5
- *R*elative Yaw angle(Weights.OV3) = 0.3

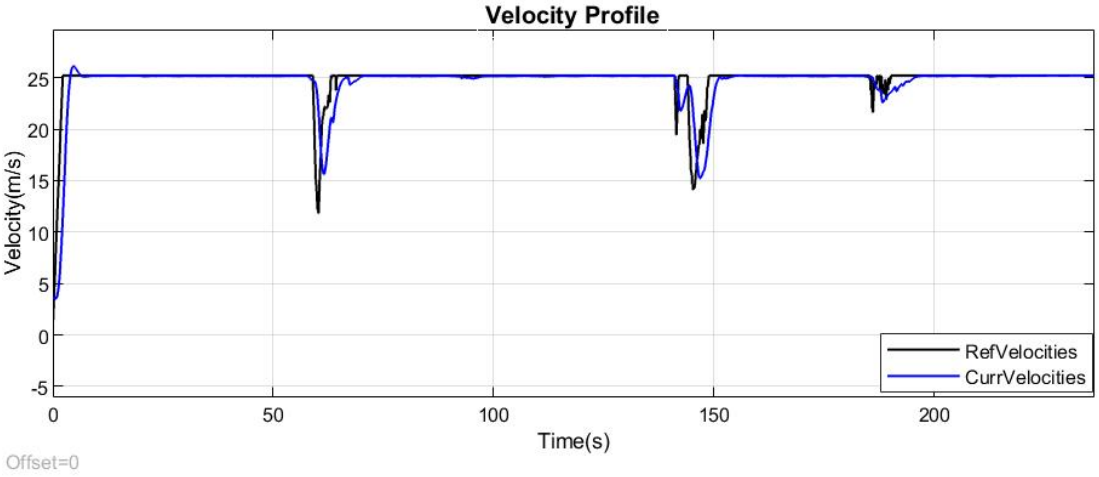


Figure 5.12: Measured longitudinal velocity V_x (blue) vs Reference velocityV_{ref}

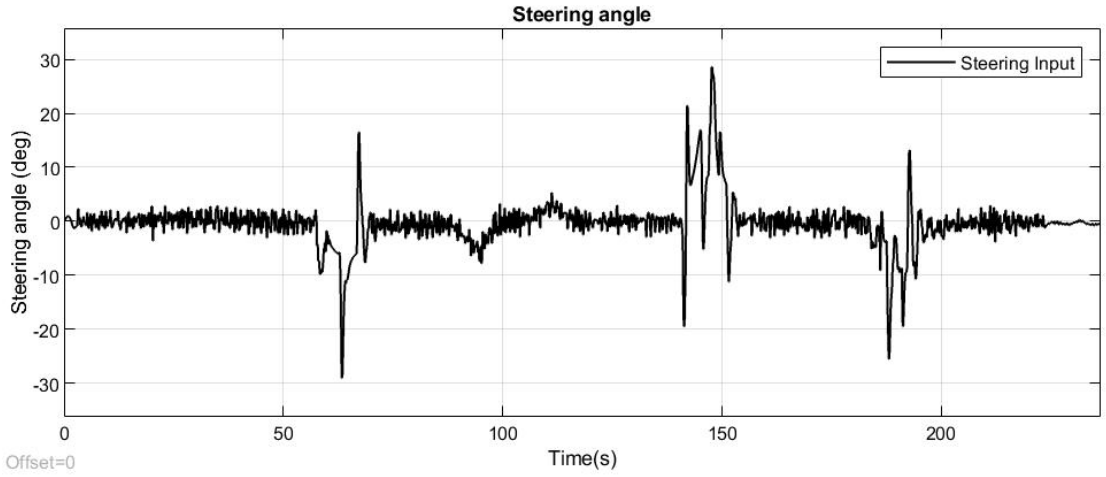


Figure 5.13: Steering angle δ command generated by MPC

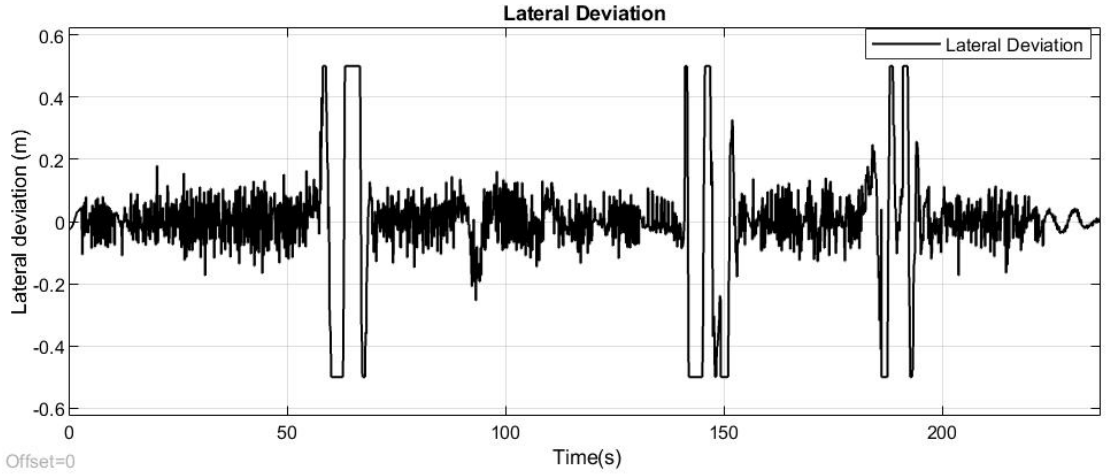


Figure 5.14: e_1-Lateral deviation

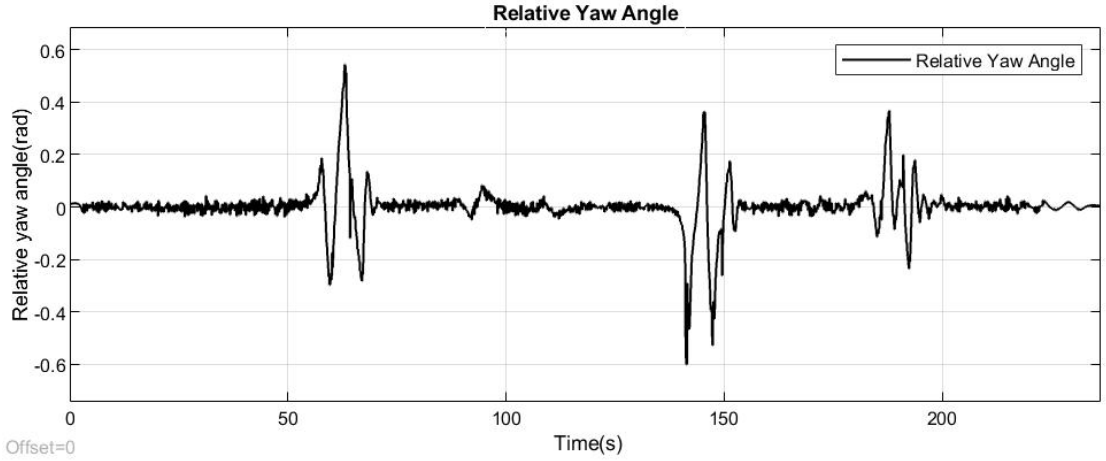


Figure 5.15: e_2-Relative yaw angle

5.1.5 Performance report

Overall the performance of this controller validated with the simplified vehicle model on three different race track scenario, it is quiet efficient only to the extent where there is less curvature profile and for sharp or narrow path the errors are high and it requires aggressive control characteristics to withhold the path to keep following. The model is designed with the reference from the adaptive cruise control and lane keep assist system that has been created and exiting in the industry for high driver assistance.The cost function defined in this model predictive control system is based on the integration of previously stated reference model,where it is formulated for the scenario with less deviation of the path, for instance highway scenarios. Figure5.16 depicts schematic representation of the optimization proce-

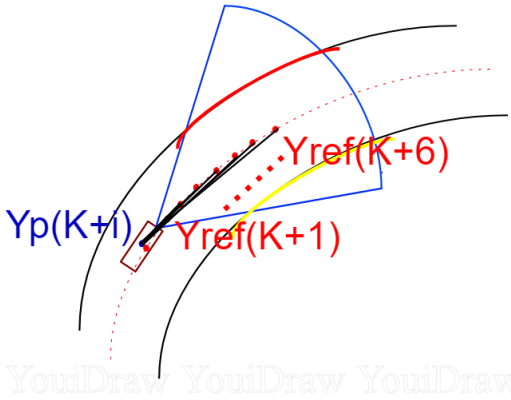


Figure 5.16: Schematic representation of cost optimization

dure. The cost optimization is happening between the current state of the vehicle Y_pk+i and the first predicted reference State Y_{ref}, that is the reason behind sudden drop in velocity or in some situation where it reduces later when it exceed limit conditions that's when the continuous change in steering command is generated by controller to make the vehicle to come and align with the predicted point. At higher velocities system is more unstable and exits the lane. Adding more weights to the controller could keep the vehicle inside the lane but its behavior becomes more aggressive apparently system would not be robust Results obtained by analyzing this model is by an optimal trade off in controller weights.

5.2 Motion Planning

5.2.1 Simulation setup

The motion planning algorithm and tracking control model is implemented in Matlab and Simulink. Initially to implement the planning methodology environment model is important, and as for as this thesis is concern the reference data is extracted from the *Drivingscenariodesigner* assuming that this is the data from vision sensors. In real condition the surrounding environment is perceived using visionary sensors and further processed using filters to determine the boundary condition around the vehicle and the process keeps updating every time step. In our case lane specification is extracted as the synthetic data, that is coordinated of left and right boundary of the lane. From that information the occupancy grid map is modeled. To build this map we use the function from the Matlab called *helpSLcreatecostmap*, where the main properties used in this function are,

- MapWidth =250
- MapLength = 350
- costVal = 0
- CellSize = 0.7

(a) Occupancy grid Round track map

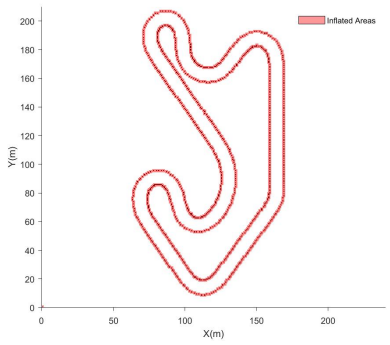
(b) Estimated center coordinates for route planner

Figure 5.17: a.Occupancy grid Round track map generated with boundary coordinate points, b.Estimated center points to plan the path

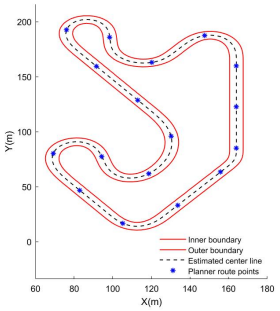
- FreeThreshold = 0.8
- OccupiedThreshold = 1
- vehicleDims = vehicleDimensions(4,1.7)
- costmap.CollisionChecker.NumCircles = 4
- costmap.CollisionChecker.InflationRadius = 0.5

Figure5.17(b) refers to the center point generation from the boundary data. The point generation is the simple procedure in which the algorithm determines the set of center points and its initial orientation to connecting all these destination points. This is defined with the distance approximation, that is three values 30,50, and 80 are different set of distance between start point and respective end points. Experimentation are made with this values to define effective planned path. To build this map we use the function from the Matlab called *helpSLcreatecostmap*, where the main properties used in this function are,

- MapWidth =240
- MapLength = 210
- costVal = 0
- CellSize = 0.7



(a) Occupancy grid Handling track map



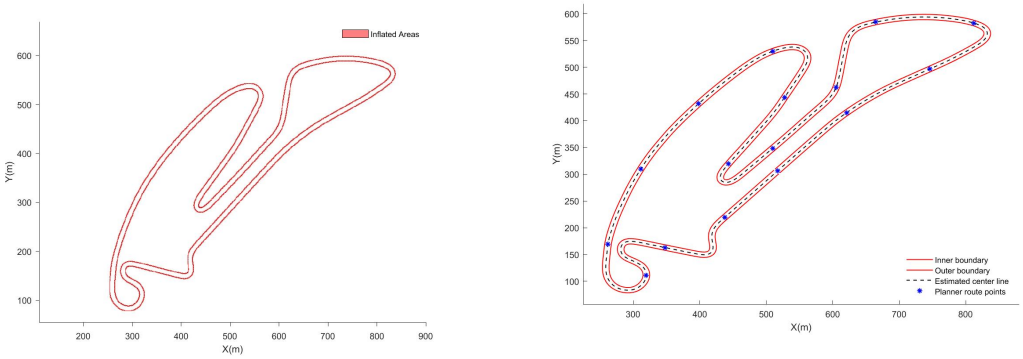
(b) Estimated center coordinates for route planner

Figure 5.18: a.Occupancy grid Handling track map generated with boundary coordinate points, b.Estimated center points to plan the path

- FreeThreshold = 1
- OccupiedThreshold = 1
- vehicleDims = vehicleDimensions(4,1.7)
- costmap.CollisionChecker.NumCircles = 4
- costmap.CollisionChecker.InflationRadius = 0.5

Figure5.18(b) refers to the center point generation from the boundary data. The point generation is the simple procedure in which the algorithm determines the set of center points and its initial orientation to connecting all these destination points. This is defined with the distance approximation, that is three values 30,50, and 80 are different set of distance between start point and respective end points. Experimentation are made with this values to define effective planned path. To build this map we use the function from the Matlab called *helpSLcreatecostmap*, where the main properties used in this function are,

- MapWidth = 900
- MapLength = 700
- costVal = 0



(a) Occupancy grid Berlin race track map

(b) Estimated center coordinates for local planner

Figure 5.19: a.Occupancy grid Berlin race track map generated with boundary coordinate points, b.Estimated center points to plan the path

- CellSize = 0.85
- FreeThreshold = 0.8
- OccupiedThreshold = 1
- vehicleDims = vehicleDimensions(4,1.7)
- costmap.CollisionChecker.NumCircles = 4
- costmap.CollisionChecker.InflationRadius = 0.5

Figure5.19(b) refers to the center point generation from the boundary data. The point generation is the simple procedure in which the algorithm determines the set of center points and its initial orientation to connecting all these destination points. This is defined with the distance approximation, that is three values 100,120 and 150 are different set of distance between start point and respective end points. Experimentation are made with this values to define effective planned path.

5.2.2 Scenario 1- Round track

In this section the motion panning simulation results of the round track is presented. Figure5.20 shows the two different configuration of route plan considered to determine the feasible path, and the route plan is the integral part of the closed loop tacking algorithm, where (a) shows the 6 goal destination points defined by selecting every respective 50th center points from the each local initial points (start point), similarly (b) shows the 9 goal destination points defined by selecting every 30th center point from each local start points respectively. The attribute properties that are defined to those routes are,

- Start pose $[x_n y_n \theta_n]$, n = 1....6 or n_i
- End pose $[x_n y_n \theta_n]$, n = 1....6 or n_i
- Route attributes

  Stop point= 0 or 1, O - false, 1 - true

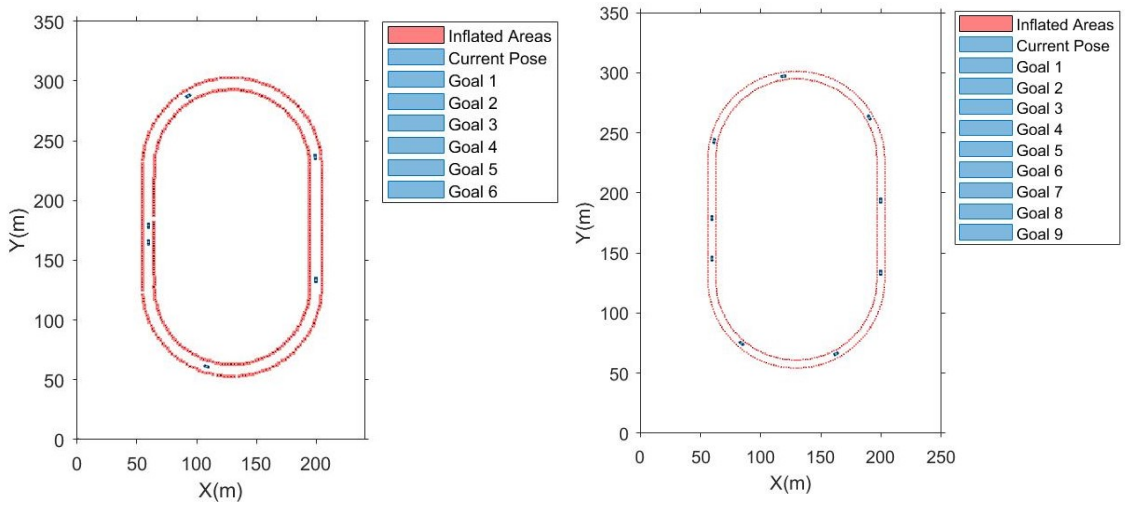
  Turn maneuver = 0 or 1, O - false, 1 - true

  Max speed = 10 (m/s), Max longitudinal lateral acceleration - 3 m/s$\hat{2}$

  End speed = 3.5 (m/s)

Figure5.21 shows the global planned path with two different route plan. the motion planner function used the RRT algorithm that is search based to method to define local kinematic trajectory to each local destination defined value, since it is loop track the planner tries to fetch trajectory until back initial position, and during this planning it tries various probable sample nodes to avoid the collision and if it fails the replan function intervene to provide new feasible trajectory, where (a) shows the smooth center trajectory as an optimal result compared to the (b) because shows second has much deviation in the straight profile and curvature profile. The planner properties that are defined to achieve optimal solution are,

- Minimum iteration = 1000
- Connecting distance = 30 (m)
- Minimum turning radius = 5 (m)



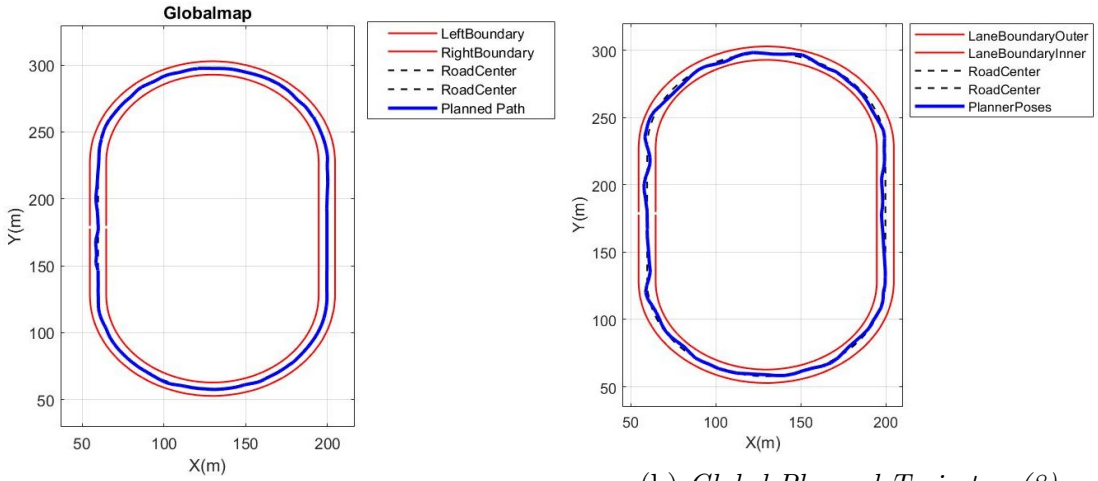
(a) *Route plan with 6 goal pose*

(b) *Route plan with 8 goal pose*

Figure 5.20: a. Route destination distance 50 sample points, b. Route destination distance 30 sample points

- Maximum steering angle = 35 (deg)

Figure5.22 shows the complete tracked path profile of configuration (a) and (b). The controller tries to run the vehicle sequentially for every local planned path, that is first planer fetches the an optimal path to follow to the first local destination, once

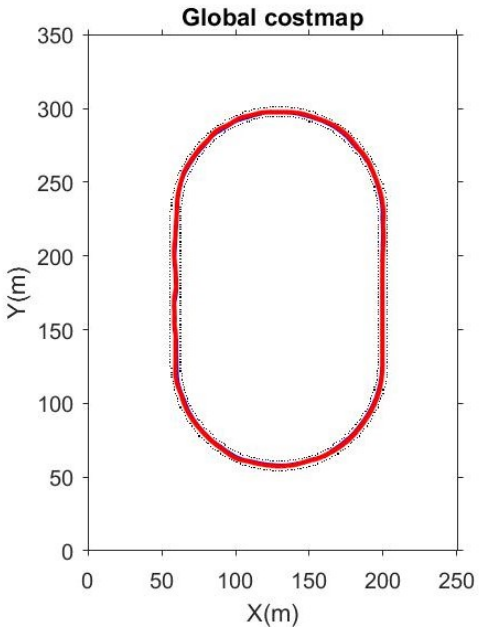


(a) *Global Planned Trajectory(6)*

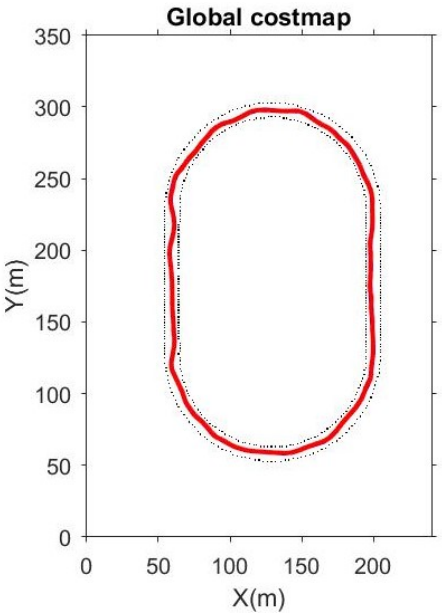
(b) *Global Planned Trajectory(8)*

Figure 5.21: a. Reference planned path considering 5 destination points, b. Reference planned path considering 8 destination points

it reaches the first goal point the goal checker initiates command to create next local path.Optimal tracking results (a) configuration are reported in the below figures, where first one Figure2.3 is the the measured velocity with respect to reference velocity. Maximum speed achieved during the tracking is around 8(m/s) and it reaches to the minimum speed of around 3.5(m/s). Figure5.24 shows the steering command generated by tracking controller to perform the turn maneuvers according to the planned trajectory, where the value ranges in between ±20 (deg) that is below the given constrain limit. Figure5.26 depicts the rate of change of vehicle orientation with respect to the reference path, where the deflection ranges in between ±0.5. Finally the lap time to complete the entire track is 119.2 (s).



(a) Global tracked path vs planned reference path(6 goal pose)



(b) Global tracked path vs planned reference path(8 goal pose)

Figure 5.22: Global tracked path

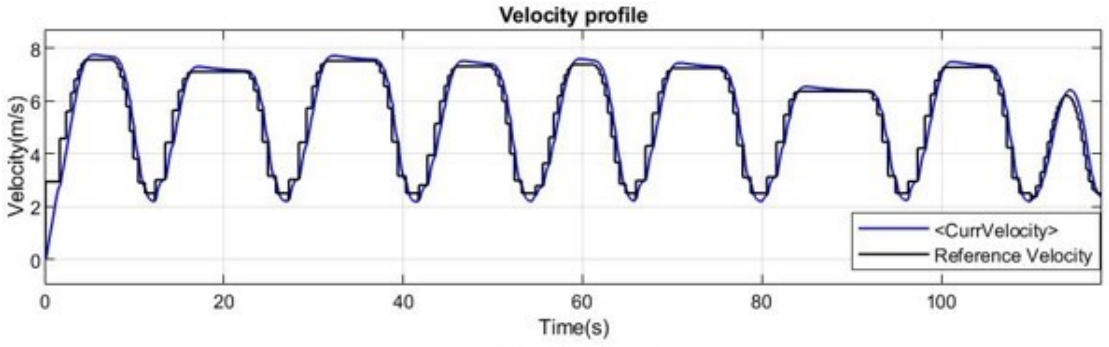


Figure 5.23: Reference generated Velocity V_{ref} vs Measured velocity V_x

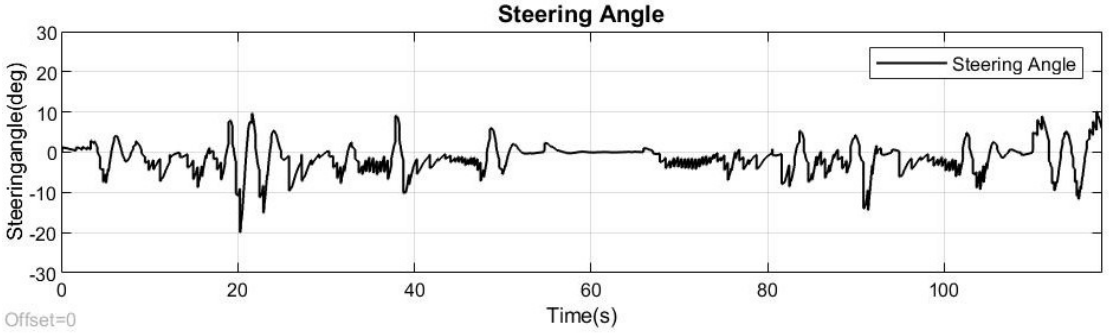


Figure 5.24: Steering angle δ command

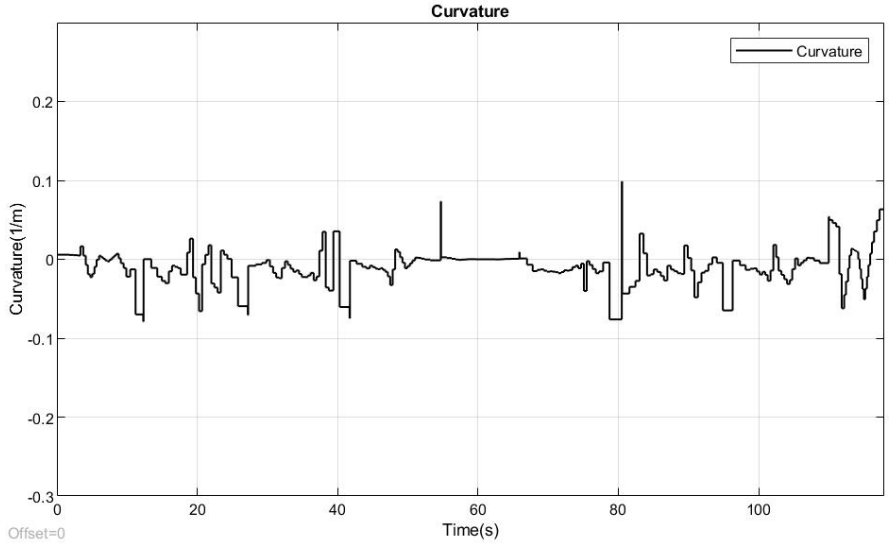


Figure 5.25: Path curvature profile

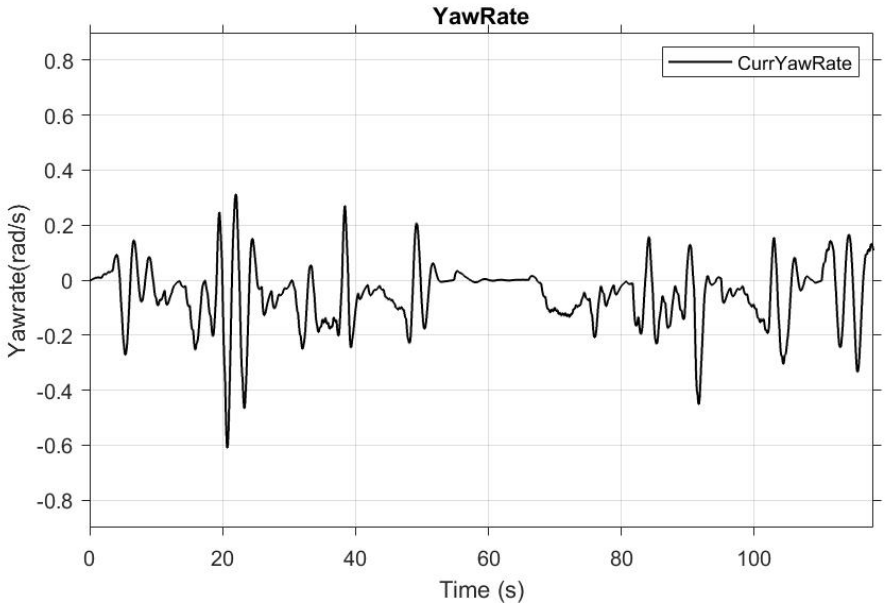


Figure 5.26: Heading angle error

5.2.3 Scenario 1 - Handling track

In this section the motion panning simulation results of the Handling track is presented.Figure5.27 shows the two different configuration of route plan considered to determine the feasible path, and the route plan is the integral part of the closed loop tacking algorithm, where (a) shows the 7 goal destination points defined by selecting every respective 50th center points from the each local initial points (start point), similarly (b) shows the 5 goal destination points defined by selecting every 80th center point from each local start points respectively. The attribute properties that are defined to those routes are,

- Start pose $[x_n y_n \theta_n]$, n = 1....7 or n_i
- End pose $[x_n y_n \theta_n]$, n = 1....7 or n_i
- Route attributes

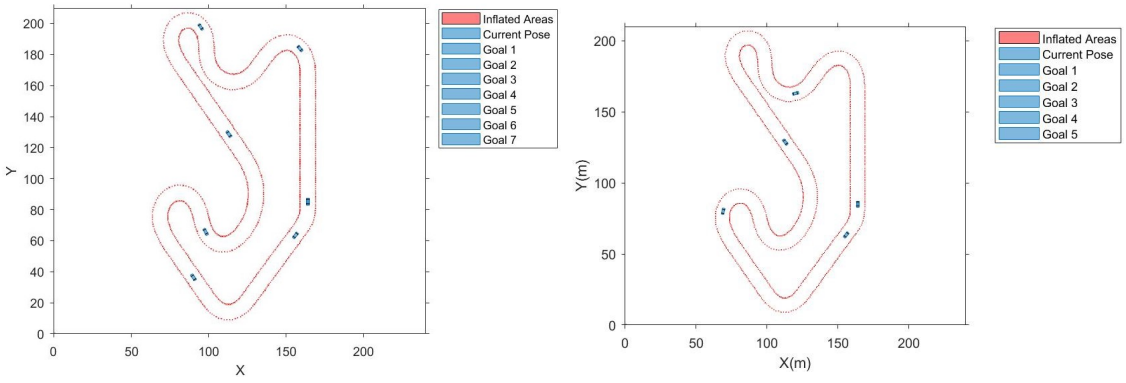
  Stop point= 0 or 1, O - false, 1 - true

  Turn maneuver = 0 or 1,O - false, 1 - true

  Max speed = 10 (m/s), Max longitudinal lateral acceleration - 3 m/s$\hat{2}$

  End speed = 3.5 (m/s)

Figure5.28 shows the global planned path of optimal route plan configuration. the motion planner function used the RRT algorithm that is search based method to



(a) *Route plan with 7 goal pose* (b) *Route plan with 5 goal pose*

Figure 5.27: a. Route destination distance 50 sample center points, b. Route destination distance 80 sample center points

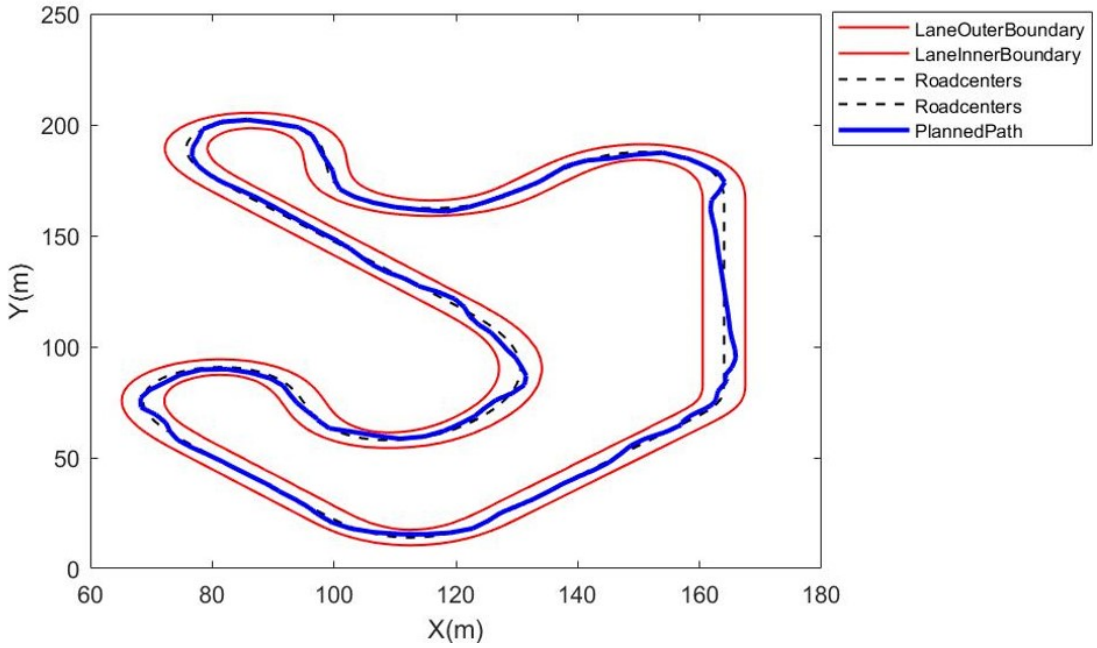
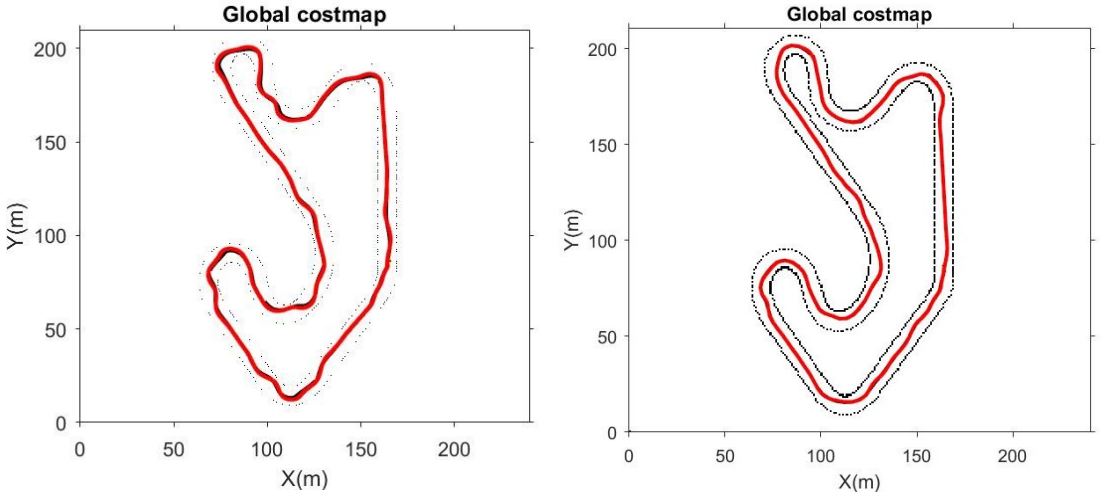


Figure 5.28: Global Planned path for Handling track

define local kinematic trajectory to each local destination defined value, since it is loop track the planner tries to fetch trajectory until back to initial position, during this planning it tries to connect various probable sample nodes to avoid the collision and if in-case it fails the replan function intervene to provide new feasible trajectory, where (b) shows the smooth center trajectory as an optimal result compared to the (a) because it has much deviation in the straight profile and curvature profile. The planner properties that are defined to achieve optimal solution are,

- Minimum iteration = 2000
- Connecting distance = 25 (m)
- Minimum turning radius = 7.5 (m)
- Maximum steering angle = 30 (deg)

Figure5.29 shows the complete tracked path profile of handling track . The controller tries to run the vehicle sequentially for every local planned path, that is first planer fetches the an optimal path to follow till the first local destination, once it reaches the first goal point the goal checker initiates command to create next

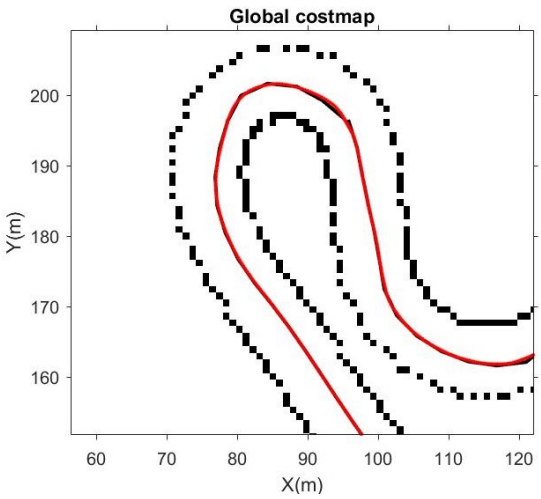


(a) Global tracked path vs planned reference path(5 goal pose)

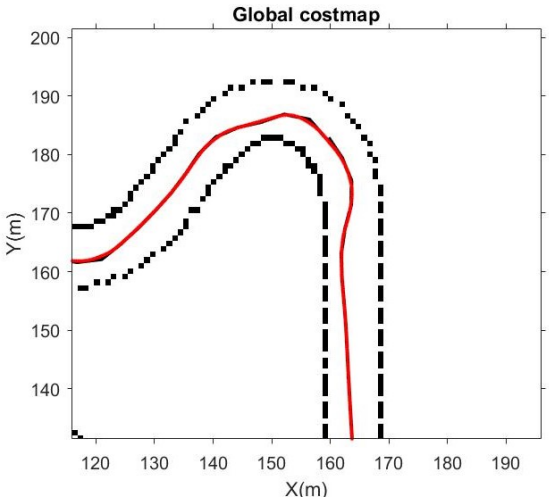
(b) Global tracked path vs planned reference path(7 goal pose)

Figure 5.29: Global Tracked path with respect to reference path

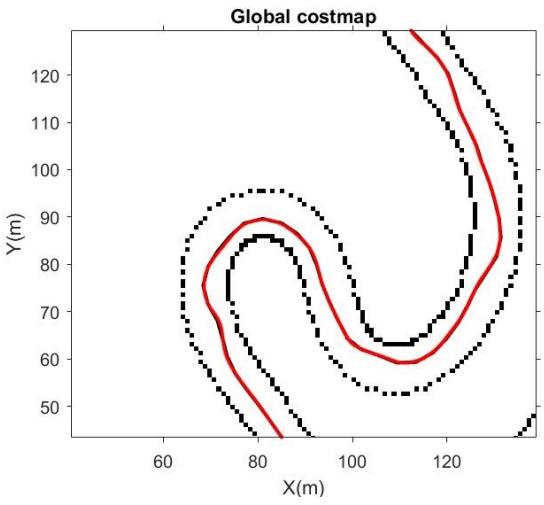
local path. Optimal Tracking results of (b) configuration are reported in the below figures, where first one Figure5.33 is the the measured velocity with respect to reference velocity. Maximum speed achieved during the tracking is around 7(m/s) and it reaches to the minimum speed of around 3.5(m/s). Figure5.34 shows the steering command generated by tracking controller to perform the turn maneuvers according to the planned trajectory, where the value ranges in between ±28 (deg) that is below the given constrain limit. Figure5.36 depicts the rate of change of vehicle orientation with respect to the reference path, where the deflection ranges in between ±0.8. Finally the lap time to complete the entire track is 162.5(s).



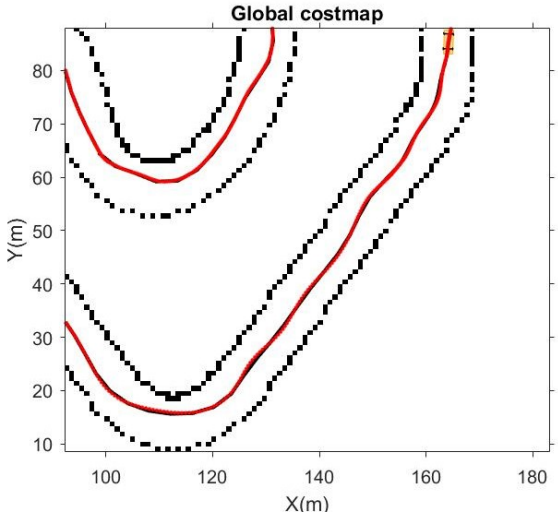
(a) *Curvature section 1*



(b) *Curvature section 1*



(a) *Curvature section 1*



(b) *Curvature section 1*

Figure 5.31: Handling track curvature sections with tracked pose

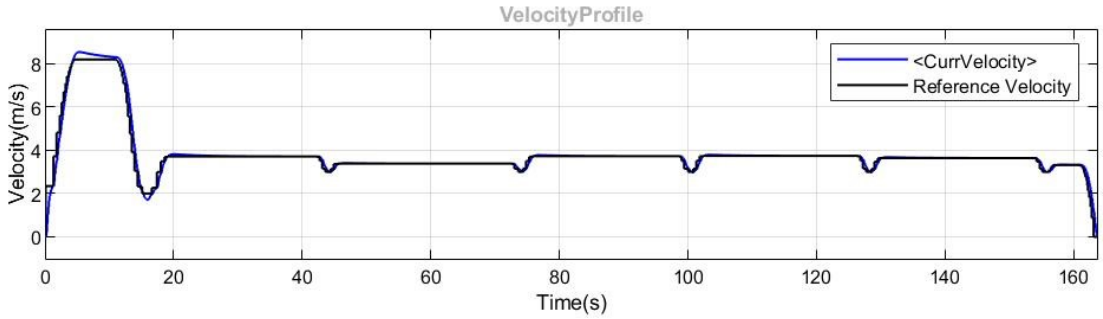


Figure 5.32: Reference generated Velocity V_{ref} vs Measured velocity V_x

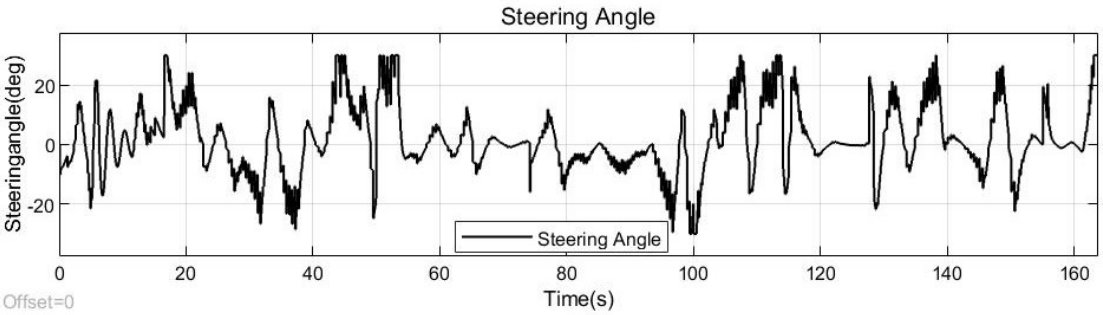


Figure 5.33: Steering angle δ command

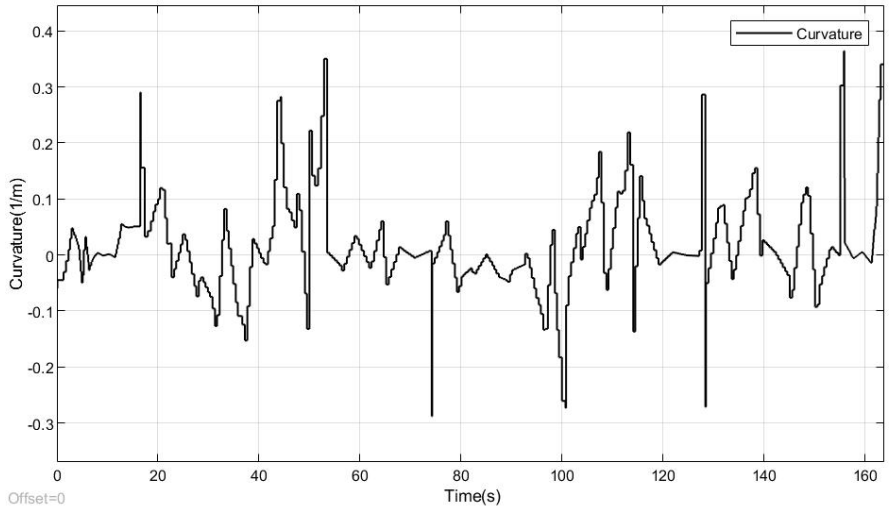


Figure 5.34: Path curvature profile

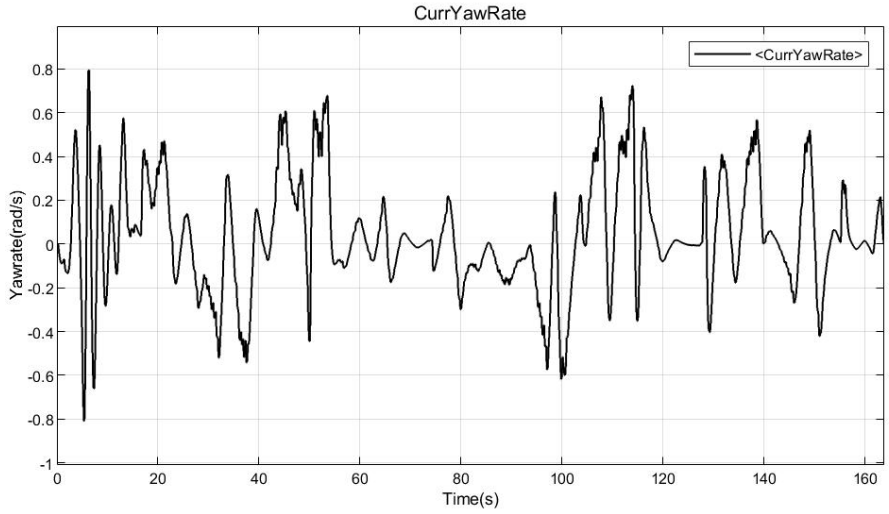


Figure 5.35: Heading angle error

5.2.4 Scenario 3 - Berlin race track

In this section the motion panning simulation results of the Berlin race track is presented.Figure5.36 shows the configuration of route plan considered to determine the feasible path, and the route plan is the integral part of the closed loop tacking algorithm, where it shows the 12 goal destination points defined by selecting every respective 80th center points from the each local initial points (start point). The attribute properties that are defined to those routes are,

- Start pose $[x_n y_n \theta_n]$, n = 1....12 or n_i
- End pose $[x_n y_n \theta_n]$, n = 1....12 or n_i
- Route attributes

  Stop point= 0 or 1, O - false, 1 - true

  Turn maneuver = 0 or 1,O - false, 1 - true

  Max speed = 10 (m/s), Max longitudinal lateral acceleration - 3 m/s$\hat{2}$

  End speed = 3.5 (m/s)

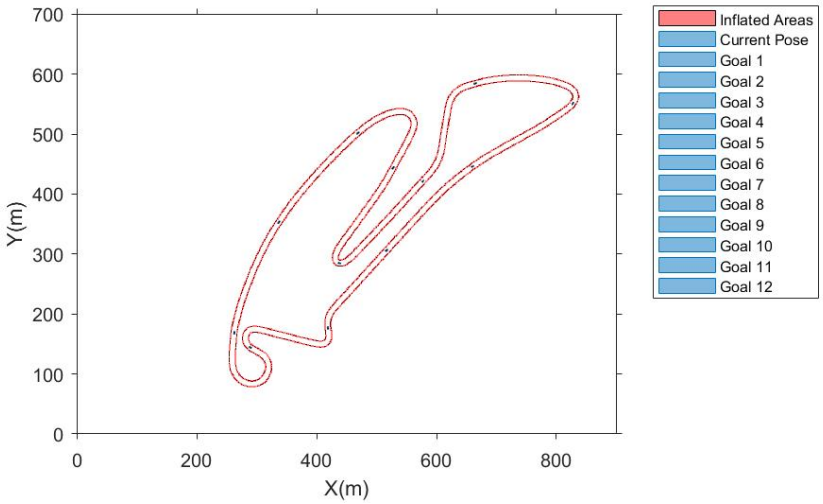


Figure 5.36: Route plan with 12 Goal pose

The motion planner function used the RRT algorithm that is search based to method to define local kinematic trajectory to each local destination points defined,

since it is loop track the planner tries to fetch trajectory until the initial position, during this planning it tries various probable sample nodes to avoid the collision and if it fails the replan function intervene to provide new feasible trajectory. The planner properties that are defined to achieve optimal solution are,

- Minimum iteration = 2000
- Connecting distance = 15 (m)
- Minimum turning radius = 5 (m)
- Maximum steering angle = 35 (deg)

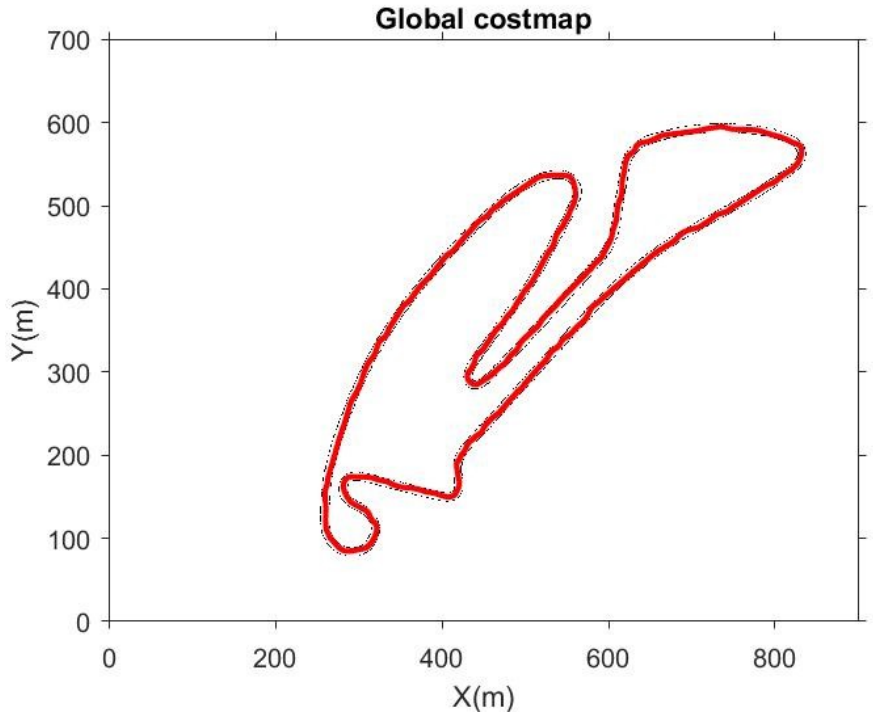
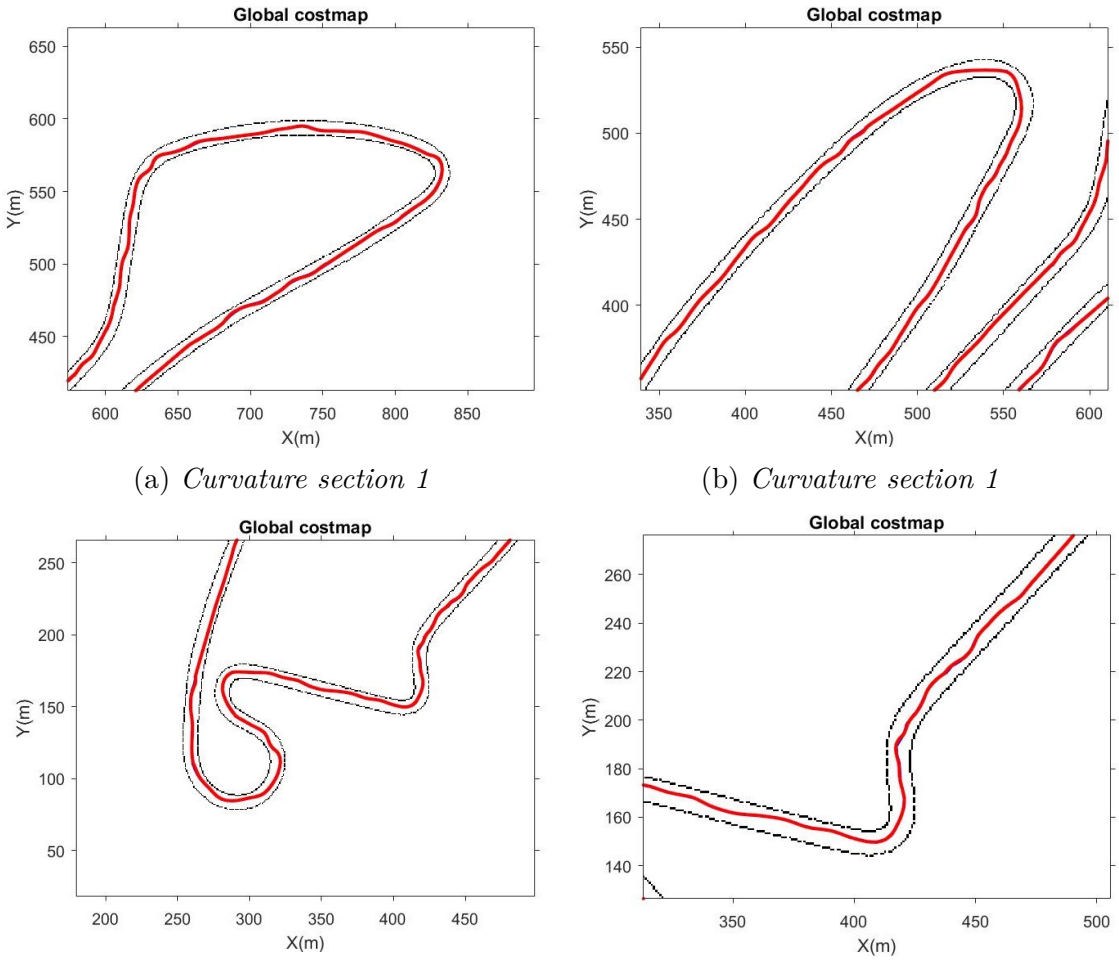


Figure 5.37: Global tracked path with respect to reference planned path

Figure5.37 shows the complete tracked path profile of configuration. The controller tries to run the vehicle sequentially for every local planned path, that is first planer fetches the an optimal path to follow to the first local destination, once it reaches the first goal point the goal checker initiates command to create next local path. Tracking results of the configuration are reported in the below figures, where

first one Figure5.39 is the the measured velocity with respect to reference velocity. Maximum speed achieved during the tracking is around 7.5(m/s) and it reaches to the minimum speed of around 3.5(m/s). Figure5.40 shows the steering command generated by tracking controller to perform the turn maneuvers according to the planned trajectory, where the value ranges in between ±30 (deg) that is below the given constrain limit. Figure5.42 depicts the rate of change of vehicle orientation with respect to the reference path, where the deflection ranges in between ±0.3. Finally the lap time to complete the entire track is 680.5(s).



(a) *Curvature section 1*

(b) *Curvature section 1*

(c) *Curvature section 3*

(d) *Curvature section 4*

Figure 5.38: Berlin race track curvature sections with tracked pose

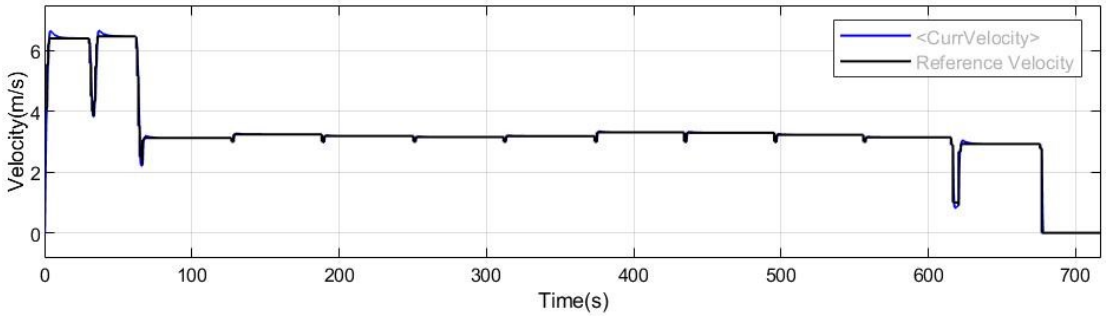


Figure 5.39: Reference generated Velocity V_{ref} vs Measured velocity V_x

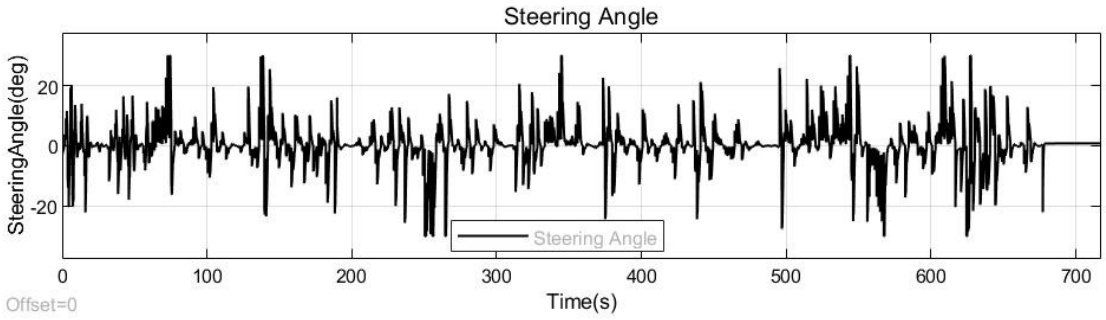


Figure 5.40: Steering angle δ command

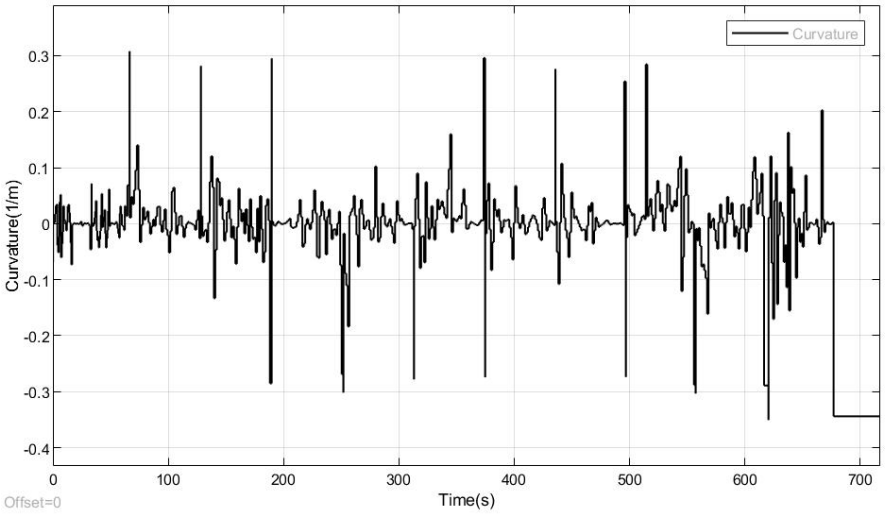


Figure 5.41: Path curvature profile

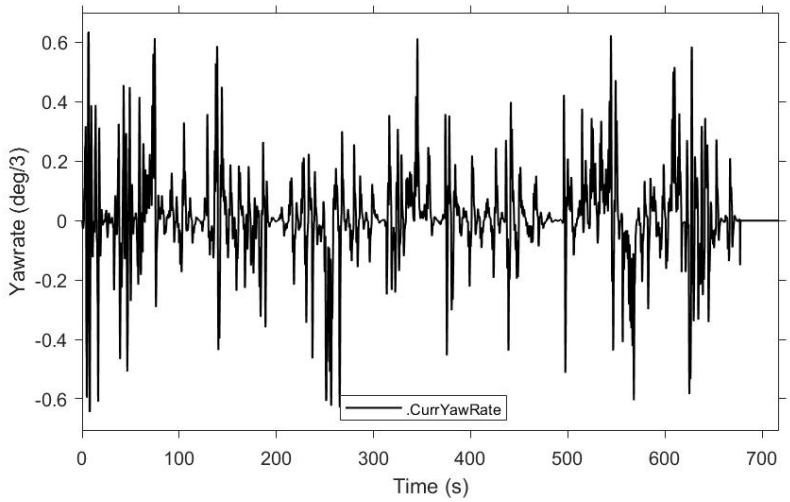


Figure 5.42: Heading angle error

Chapter 6

Conclusion and Future works

In this experimental thesis work, the performance evaluation of model predictive based combined lateral and longitudinal control system for lane tracking, and the incremental search based planned trajectory tracking control system is made for developing a proper autonomous vehicle control system.
In the first phase of the work, the adaptive model predictive control system is validated with a simplified vehicle model on three different race track scenarios, they are (a) Round track, (b) Handling track, (C) Berlin race track. The proper generation of acceleration, deceleration and steering command is the primary task of the controller and minimizing tracking errors such as lateral deviation error, relative heading angle error and longitudinal velocity is the secondary task concerning lateral stability of the vehicle. The analysis report shows that the behaviour of the controller is smooth with minimum errors while tracking straight and largely curved road profile, whereas for a sharply curved road profile the tracking errors are high and there is a loss in comfort of the vehicle. In the second phase of the thesis work, motion planning and trajectory tracking control system are studied. The incrementally search-based planning strategy is used to define the navigation path for three different race track scenario. This type of planner always tries to generate a path within a given start and destination point, therefore it is a local planner exploited in this work to obtain global track. Multiple configurations of the route plan have experimented to the generated smooth trajectory to follow, where the strategy with more route destinations results are not optimal when compared to the less one because it generates much more deviation along the route. Decoupled

lateral and longitudinal closed-loop tracking control system implemented to track the navigation path with minimum errors, where the errors are comparably less because the velocity generated from the reference trajectory is very low, and so it is seen as one of the draws back for lap time. Out of three scenarios, the trajectory tracking performance was smooth with less deviation and heading angle error for round track, whereas for handling and berlin race track the trajectory definition is sub optimal.

Certain works can be carried out in future to improve the control model. Formulate a proper optimization problem by integrating the lane detection control model with incremental search-based path planning strategy. Analyze different real-time robot navigation algorithm for the required race application. Develop a High-level offline path planning algorithm considering lap time improvement and curvature minimization. Better perception of the surrounding environment around the vehicle can be improved by fusing multiple sensors data, for instance, camera, lidar, etc. A vehicle configuration with more precise drive-line dynamics and tire dynamics should be modeled and used as the prediction model in the controller. Validate the tracking control system with complex vehicle model (14 DOF) in the simulation environment before implementation on the real vehicle.

In the end, this thesis work has contributed to the autonomous vehicle research team at LIM (Laboratorio Interdisciplinary di Meccatronica), and future thesis students can use the submitted project to progress related work.

Bibliography

[1] Péter Szikora Ph.D and Nikolett Madarász, *Self-driving cars – the human side Working paper.*
Weblink:https://www.researchgate.net/publication/324095773_Self-driving_cars_-_The_human_side

[2] World health Organization(WHO), *Road safety.*
Weblink:https://www.who.int/data/gho/data/themes/road-safety/GHO/road-safety.

[3] The European Commission, *Safety in the automotive sector,Motor vehicle safety.*
Weblink:https://ec.europa.eu/growth/sectors/automotive/safety_en

[4] PanosJ.Antsaklis, KevinM.Passino and S.J.Wang, *An Introduction to autonomous control system*
Weblink:https://www.researchgate.net/publication/324095773_Self-driving_cars_-_The_human_side, June 1991.

[5] Bonnie Gringer, *History of Autonomous car* Technical report, NHTSA, 2007.
Weblink:https://www.titlemax.com/resources/history-of-the-autonomous-car/

[6] Kambria , *The History and Evolution of Self-Driving Cars*,Jun 23, 2019.
Weblink:https://medium.com/kambria-network/the-history-and-evolution-of-self-driving-cars-c0c157feaba5.

[7] James M. Anderson, Nidhi Kalra, Karlyn D. Stanley, Paul Sorensen,Constantine Samaras and Oluwatobi A. Oluwatola, *Brief History and Current State of Autonomous Vehicles*, Published by, RAND Corporation, 2014
Weblink:https://www.jstor.org/stable/10.7249/j.ctt5hhwgz.11.

[8] SAE(Society of Automotive Engineers), *Taxonomy and Definitions for Terms*

Related to Driving Automation Systems for On-Road Motor Vehicles, J3016™, Revised in June 2018.

[9] FSG(Formula student Germany), *Autonomous Driving at Formula Student Germany 2017*, August 2016.
Weblink:https://www.formulastudent.de/nc/pr/news/details/article/868 System Dynamics, 52(2):261–279, 2014.

[10] Matlab documentation, *Understanding Model Predictive Control*
Weblink:https://it.mathworks.com/videos/understanding-\model-predictive-control-part-2-what-is-mpc--1528106359076.html.

[11] Monimoy Bujarbaruah, Xiaojing Zhang, Francesco Borrelli, *Adaptive MPC for Autonomous Lane Keeping*, Department of Mechanical Engineering University of California Berkeley, Berkeley, CA, USA, May 21, 2018.

[12] Jochen Pohl,Jonas Ekmark,*a lane keeping assist system for passenger cars*, Volvo Car Corporation,Paper No. 529.

[13] Valerio Turri, Ashwin Carvalho t, Hongtei Eric Tseng, Karl Henrik lohansson, Francesco Borrellit *Linear Model Predictive Control for Lane Keeping and Obstacle Avoidance on Low Curvature Roads*,IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013.

[14] Mechanical Simulation documentation,*Running a VS Vehicle Model in Simulink*
Weblink:https://www.carsim.com/downloads/pdf/Simulink_ABS_Example.pdf.

[15] Alexander Heilmeier, Alexander Wischnewski, Leonhard Hermansdorfer, Johannes Betz, Markus Lienkamp and Boris Lohmann,*Minimum curvature trajectory planning and control for an autonomous race car*
Weblink:https://doi.org/10.1080/00423114.2019.1631455, Received 09 Oct 2018, Accepted 27 May 2019, Published online: 18 Jun 2019.

[16] Matlab documentation, *Coordinate Systems in Vehicle Dynamics Block set*
Weblink:https://www.mathworks.com/help/vdynblks/ug/coordinate-systems-in-vehicle-dynamics-blockset.html

[17] Rajesh Rajamani,*Vehicle Dynamics and Control*, Mechanical Engineering Series,

[18] Giancarlo Genta Dipartimento di Meccanica Politecnico di Torino,

Italia,*MOTOR VEHICLE DYNAMICS,Modeling and Simulation*, First published 1997 Reprinted 1999, 2003 (in bigger trim size), 2006

[19] Thomas D. Gillespie,*Fundamentals of Vehicle Dynamics*, Published by SAE International with a Product Code of R-114, ISBN of 978-1-56091-199-9

[20] Mechanical Simulation documentation,*VS Solvers and Math Models* Weblink:`https://www.carsim.com/downloads/pdf/CarSim_Math_Models.pdf`.

[21] Kiran Kone,*Lateral and longitudinal control of an autonomous racing vehicle*

[22] Irfan Khan,*Combined lateral and longitudinal control for autonomous driving based on Model Predictive Control*

[23] Alexander Heilmeier, Alexander Wischnewski, Leonhard Hermansdorfer, Johannes Betz, Markus Lienkamp and Boris Lohmann,*Minimum curvature trajectory planning and control for an autonomous race car* Weblink:`https://doi.org/10.1080/00423114.2019.1631455`, Received 09 Oct 2018, Accepted 27 May 2019, Published online: 18 Jun 2019.

[24] Matlab documentation, *Occupancy Grids* Weblink:`https://www.mathworks.com/help/nav/ug/occupancy-grids.html`

[25] Karaman, Sertac, and Emilio Frazzoli, *"Optimal Kinodynamic Motion Planning Using Incremental Sampling-Based Methods."* , 49th IEEE Conference on Decision and Control (CDC). 2010.

[26] Matlab documentation, *pathPlannerRRT* Weblink:`https://www.mathworks.com/help/driving/ref/pathplannerrrt.html`

[27] Shkel, Andrei M., and Vladimir Lumelsky, *"Classification of the Dubins Set."* , Robotics and Autonomous Systems. Vol. 34, Number 4, 2001, pp. 179–202.

[28] Reeds, J. A., and L. A. Shepp, *"Optimal paths for a car that goes both forwards and backwards."*,Pacific Journal of Mathematics. Vol. 145, Number 2, 1990, pp. 367–393.

[29] Ziegler, J., and C. Stiller,*"Fast Collision Checking for Intelligent Vehicle Motion Planning."*,IEEE Intelligent Vehicle Symposium. June 21–24, 2010.

[30] Villagra, Jorge, Vicente Milanés, Joshué Pérez, and Jorge Godoy,*"Smooth path and speed planning for an automated public transport vehicle."*,Robotics and Autonomous Systems. Vol. 60, Number 2, February 2012, pp. 252–265.

[31] Rajesh Rajamani, *Vehicle Dynamics and Control*, Mechanical Engineering Series, 2012.

[32] Hoffmann, Gabriel M., Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun, *"Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing."*,American Control Conference. 2007, pp. 2296–2301. doi:10.1109/ACC.2007.4282788