Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Control

Trajectory Determination and Control for Autonomous Racing

Burak AYDIN

Supervisor: doc. Ing. Martin Hromcik, Ph.D.
May 2020

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Aydin Burak**	Personal ID number:	**480848**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Control Engineering**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Cybernetics and Robotics**		

II. Master’s thesis details

Master’s thesis title in English:

Treajectory determination and control for autonomous racing

Master’s thesis title in Czech:

Algoritmy k určení trajektorie a vedení po trati pro autonomní soutěžní vozy

Guidelines:

The goal of the thesis is to develop algorithms suitable for determination of feasible or optimal trajectories for fast dynamic autonomous driving. The motivation comes from the recent development in the student formula competition and the CVUT FEL eForce team's efforts towards autonomous racing involvment.
1. Implement the simulation framework, based on either available tools (e.g. the new Automotive Toolbox and Autonomous Driving Toolbox for
MATLAB) or your own custom-built MATLAB/Simulink solutions.
2. Get familiar with existing solutions. Implement simulation models implementing selected functionalities.
3. Propose and implement algorithms for estimation of the vehicle from the track centerline, based on information from an on-board stereoscopic camera.
4. Validate the developed functionalities on realistic models inspired by the past student formula autonomous races / tracks.
5. Implement an MPC (Model Predictive Control) solution to track the reference trajectory based on suitable simplified vehicle dynamics models (the single-track model).

Bibliography / sources:

[1] Kiencke, Uwe, Nielsen, Lars, Automotive Control Systems, Springer-Verlag Berlin Heidelberg, 2005, ISBN 978-3-540-23139-4
[2] Franklin, Gene F.; Powell, J. David; Emami-Naeini, Abbas, Feedback Control of Dynamic Systems, Global Edition, Pearson Education Limited, 2019, ISBN: 9781292274522

Name and workplace of master’s thesis supervisor:

doc. Ing. Martin Hromčík, Ph.D., Department of Control Engineering, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **13.02.2020** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until:
by the end of summer semester 2020/2021

doc. Ing. Martin Hromčík, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

I would like to thank my supervisor doc. Ing. Martin Hromcík, Ph.D. for his valuable advice and guidance during the creation of this thesis.

My sincere thanks also goes to Tomáš Haniš and Martin Gurtner for their very helpful consultations throughout my study.

I also thank my parents, Recep Nuri Aydin and Sonay Aydin, and my friends, Yagmur Ekmekci and ATA Keskekler for their support, without which this work would not be completed.

Declaration

I hereby declare that I wrote the presented thesis on my own and that I cited all the used information sources in compliance with the methodical instructions about the ethical principles for writing an academic thesis.

Prague, May 22, 2020

Author's Signature

Abstract

Autonomous vehicles are expected to shape the future of not only the daily life traffic or transportation but also the racing world as well. In recent years, the various autonomous racing competitions became widespread among university's research faculties. Therefore, the goal of the thesis was to develop algorithms suitable for determination of feasible trajectories and tracking those trajectories for fast dynamic autonomous driving. The motivation came from the recent development in the student formula competition and CVUT FEL eForce Formula team's efforts towards autonomous racing involvement.

The main contribution of the thesis was to estimate the deviation from track centerline based on stereo-optic camera. In that sense, in the first part of the thesis, a novel algorithm was developed to detect if vehicle deviates from track centerline. Then, the algorithm was enhanced and extended such that it is able to estimate the lateral deviation from track centerline.

In the second part, the control problem was introduced to track the centerline of track with minimum error. The task was decomposed into two sub-problems which were vehicle dynamical modelling and Model Predictive Control design. In particular, the controller was formulated as Lane Keeping Assist System such that it would utilize the inputs from trajectory determination.

In the last part, the simulations experiments were presented and results were discussed. The simulations were designed using Driving Scenario Designer Apps in MATLAB to create various driving scenarios which remind the real racing scenarios as in the competitions.

Keywords: centerline estimation,
crosstrack error,
vehicle modelling,
model predictive control,
MATLAB

Abstrakt

Autonomní vozidla by měla utvářet budoucnost nejen každodenního provozu nebo dopravy, ale také závodního světa. V posledních letech se mezi univerzitními výzkumnými fakultami rozšířily různé autonomní závodní soutěže. Cílem práce bylo proto vyvinout algoritmy vhodné pro stanovení proveditelných trajektorií a sledování těchto trajektorií pro rychlé dynamické autonomní řízení. Motivace vyplynula z nedávného vývoje v soutěži studentských formulí a úsilí týmu CVUT FEL eForce Formula o autonomní závodní zapojení.

Hlavním přínosem práce bylo odhadnout odchylku od osy tratě na základě stereooptické kamery. V tomto smyslu byl v první části práce vyvinut nový algoritmus pro detekci toho, zda se vozidlo odchyluje od osy tratě. Poté byl algoritmus vylepšen a rozšířen tak, že je schopen odhadnout boční odchylku od osy koleje.

Ve druhé části byl představen řídicí algoritmus ke sledování středové linie stopy s minimální chybou. Úkol byl rozložen na dva dílčí problémy, kterými byly dynamické modelování vozidla a návrh prediktivního řízení modelu. Zejména byl regulátor formulován jako asistenční systém jízdního pruhu tak, že by využíval vstupy z určování trajektorie.

V poslední části byly prezentovány simulační experimenty za účelem validace navržených metod a byly diskutovány výsledky. Simulace byly navrženy s využitím aplikací Driver Scenario Designer Apps v MATLABu k vytvoření různých jízdních scénářů, které připomínají skutečné závodní scénáře jako v soutěžích.

Klíčová slova: odhad středové linie,
chyba přesměrování,
modelování vozidel,
prediktivní kontrola modelu,
MATLAB

Contents

Figures

Chapter 1

Introduction

1.1 Introduction to Autonomous Vehicles

In recent years, there has been enormous improvements regarding autonomous vehicles(AVs). AVs have a wide range of benefits compared to human drivers. That's why, it has great emphasize for research laboratories and university faculties as well as vehicle manufacturers, such as Tesla.

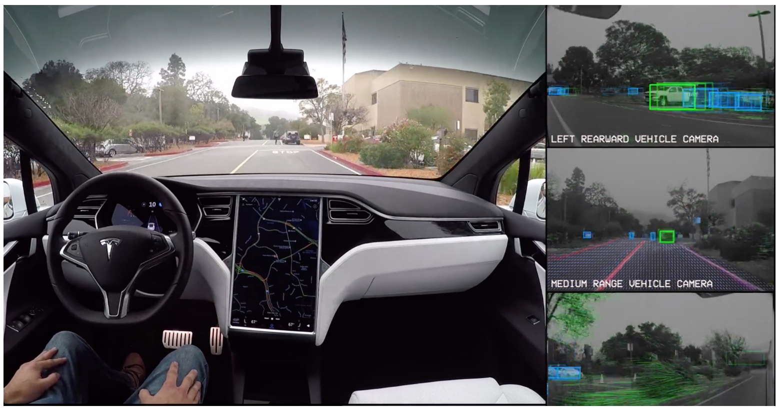


Figure 1.1: Interior view of Tesla's Autonomous Vehicle [1]

Major benefit of AVs in daily life is that they can reduce traffic accidents and prevent injuries and fatalities happened in those accidents. World Health Organization(WHO) reported that every year, approximately 1.35 million people die as a result of a road traffic crash. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury [9]. Same report also indicates that the main factor of those crashes is the failure of human driver. Therefore, safety is the biggest design concern for researchers, AVs can be great solutions to traffic accidents. Another advantage of using AVs is fuel consumption. Having more AVs in daily traffic would result in reduction of carbon dioxide (CO2) emissions as AVs cut emissions by 60 percent [10]. On the other hand, AVs have some positive effects on social life. AAA Foundation for Traffic Safety stated that each drivers in U.S. spending 50.6 minutes on the the road everyday [11]. By assigning the driving task to an autonomous vehicle, the time spent for driving would be freed and can be spend for leisure activities. More comprehensive information about AVs technology can be also found in [12] and [2].

On the other hand, the vehicle which performs fully autonomous in daily traffic has not been realized yet. AVs have 5 different autonomy levels. Stepping into higher level number means more advance features. Nowadays, Vehicles with having partial autonomous features, like self-parking and lane keeping assist, can be seen in daily life. This means to Level 2 autonomy. Experts in the industry say that fully autonomous vehicle would be part of daily life after 2030 [12]. Below figure shows the levels in AVs and description of each level.

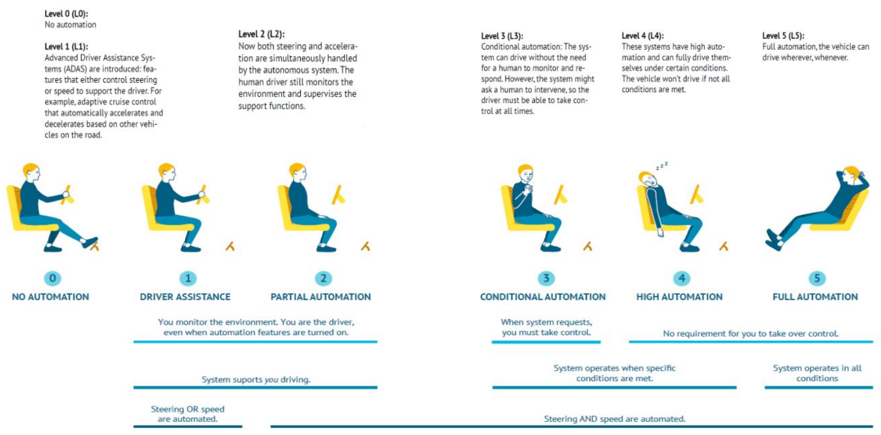

Level 0 (L0):
No automation
Level 1 (L1):
Advanced Driver Assistance Systems (ADAS) are introduced: features that either control steering or speed to support the driver. For example, adaptive cruise control that automatically accelerates and decelerates based on other vehicles on the road.
Level 2 (L2):
Now both steering and acceleration are simultaneously handled by the autonomous system. The human driver still monitors the environment and supervises the support functions.
Level 3 (L3):
Conditional automation: The system can drive without the need for a human to monitor and respond. However, the system might ask a human to intervene, so the driver must be able to take control at all times.
Level 4 (L4):
These systems have high automation and can fully drive themselves under certain conditions. The vehicle won't drive if not all conditions are met.
Level 5 (L5):
Full automation, the vehicle can drive wherever, whenever.
0
NO AUTOMATION
1
DRIVER ASSISTANCE
2
PARTIAL AUTOMATION
3
CONDITIONAL AUTOMATION
4
HIGH AUTOMATION
5
FULL AUTOMATION
You monitor the environment. You are the driver, even when automation features are turned on.
When system requests, you must take control.
No requirement for you to take over control.
System suports *you* driving.
System operates when specific conditions are met.
System operates in all conditions
Steering OR speed are automated.
Steering AND speed are automated.

Figure 1.2: Level of Autonomous Vehicles [2]

1.2 Thesis Motivation

According to information given in previous section, AVs would shape the future of world. However, it won't only change the daily life traffic or the transportation but also would change many other sectors. One of these sectors in which autonomous vehicle development has major impact, is motorsport industry. Recently, various new motorsport racing competitions began to emerge at which no human sits behind the steering well. This is a clear indication that autonomous racing would become an inevitable part of motorsport racing world in near future.

The first autonomous racing challenge was introduced in 2016, called as Roborace. Roborace is a competition with autonomously driven and electrically powered high performance racing cars. Fédération Internationale de l'Automobile (FIA) announced that Roborace would be part of FIA Formula E series soon [13]. The detailed information about the competition can be found in [14] and [15]. Fig. 1.3. represent first vehicle designed to race in Roborace.

The other autonomous racing challenge is Formula 1/10 [4]. In this one,

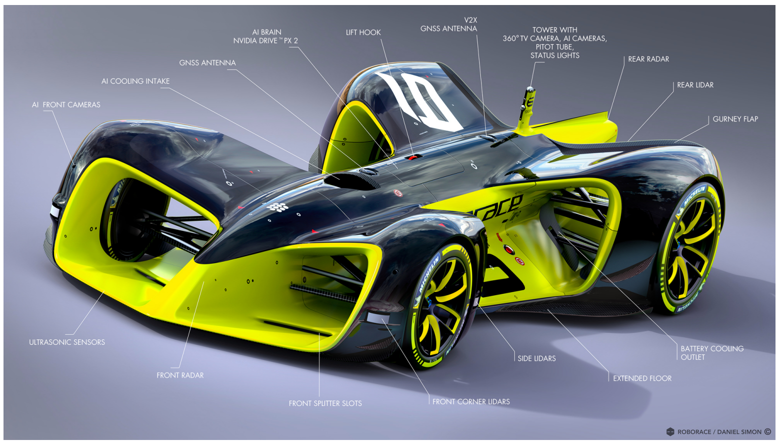


Figure 1.3: ROBOCAR, first autonomous racing car in Roborace [3]

teams try to implement best solution and get best results using an open source, affordable, and high-performance 1/10 scaled autonomous vehicle. Down-scaled vehicle carries a full suite of sensors, perception, planning and controller which make it similar to full scale solutions. Each team uses same hardware and software, such that competition becomes pure technical challenge. As a result, it is quite attractive among research institutions and faculties to compete with each other.

Figure 1.4: F1/10 Vehicle [4]

Apart from these two, another major competition for autonomous racing is Formula Student Driverless(FSD) [16]. FSD is one of the three parts of Formula Student design challenges, in which students aim to design and develop best autonomous formula vehicle solution to finish all laps in minimum time and defeat the others. Therefore, it is by far the most popular autonomous racing competition among universities. Czech Republic will be represented in this competition first time with CVUT FEL eForce Formula Team [5]. Therefore, the major motivation of this thesis is to provide practical algorithms for eForce Formula Team for trajectory determination and tracking control.

Figure 1.5: eForce Autonomous Formula Vehicle [5]

1.3 Problem Formulation

As described in previous section, the main aim of this work is to develop trajectory determination and control algorithms for CVUT FEL eForce Formula Team to be used in FSD. Therefore, the thesis consists of two major tasks. The primary task is to propose and develop algorithms to check if vehicle is moving on track centerline and estimate how much the vehicle deviates from track centerline using stereo-optics camera mounted on vehicle body. The estimation is heavily based on cones' positions, road curvatures and vehicle pose. The cones' positions can be calculated with stereo-camera using stereo triangulation. However, it is out of scope of this thesis. It is assumed that they have been already calculated and known in advance.

After accurately estimating deviation from track centerline, the next step is to design Model Predictive Control algorithms to track the reference path. That is the secondary task of thesis. In this regard, control structure is configured in such way that it would resemble Lane Keeping Assists in order to travel track centerline with minimum error. In this work, only lateral controller is introduced, the longitudinal speed is assumed as constant.

1.4 Related Work

In literature, there are a wide range of research and applications regarding autonomous racing, ranging from path planning and motion control design to complete system architecture solutions.

AMZ Driverless Team in ETH Zurich, who are winner of FSD 2017 and 2018, introduces complete real-time system design solution for FSD [17]. They uses both Lidar and Camera to estimate the cones position and extract the track map. Also, they propose sensor fusion algorithm to estimate velocity of the vehicle precisely. Regarding motion control, they employs almost same method present by Liniger.

Liniger in [18] and [19] presents several optimal control strategies for autonomous racing car. The first one is hierarchical control. In this one, trajectory is planned first in such a way that it returns maximum progress on the track at each time step. Then, planned trajectory is tracked using MPC controller. The second method, on the other hand, tries to combine planning and control problem in same convex optimization problem. This method called as contouring problem.

Braghin [20] presents another approach for trajectory determination and control for autonomous race car. It first calculates the shortest and minimum curvature path for the given track and defines velocity profile using forward-backward pass. Then, compare both method in terms of minimum lap time. Finally, it describes a control method, which is based on feed-forward compensation, to follow the determined trajectory. The paper in [21] employs the same methodology as well.

Heilmeier [22] introduces also similar approach with Braghin for planning of autonomous race car for Roborace Competition. That work generates minimum curvature path using quadratic optimization problem formulation. The main difference is the improvement in accuracy of curvature approximation by defining constraints for curvature and iterative invocation of quadratic problem.

The autonomous racing problem is taken into consideration by Kapania [23] as well. In contrast to previous two works, at first a velocity profile is generated. Then, a convex path optimisation problem is solved which minimises the resulting path curvature while taking the vehicle's handling limits into account. Similar work can be found also in [24].Additionally, Kapania servers novel control strategy. It is called as iterative learning control which enable autonomous vehicles to drive more effectively by learning from previous driving maneuvers.

1.5 Thesis Goals

The ultimate goal of this thesis is to determine trajectory for autonomous racing car based on stereo-optic camera. In particular, the trajectory of interest in this work is track centerline. In the first attempt, a novel algorithm is presented to detect if vehicle deviates from centerline based cone positions on the track. After that, the algorithms is enhanced and extended such that it would estimate the amount of deviation from track centerline. The essential input for the new algorithm is the road curvature along with vehicle orientation and speed. The output of this algorithm would also main input while designing centerline tracking controller. On the other hand, the secondary aim of the thesis to design model predictive controller to track the centerline with minimum amount of error. To achieve this, linearized vehicle dynamical model is introduced first. Then, model predictive control is formulated as lane keeping assist system such that it can utilize the cross=track error as internal plant model in the controller.

To overcome overall task, it is divided into sub-problems, so the rest of the thesis is organized as follows:

- *Chapter 2*: Trajectory determination for centerline estimation and estimating cross-track and heading error based on road curvature calculated using stereo-optics camera.
- *Chapter 3*: Kinematic and dynamical modelling of the vehicle to be used in model predictive control design.
- *Chapter 4*: General information and formulation of Model Predictive Control
- *Chapter 5*: Lateral MPC Design to track the path centerline using estimated cross-track and heading error.
- *Chapter 6*: Experiments with using different scenarios to validate proposed methods.
- *Chapter 7*: Conclusion of the work and suggested future works

Chapter 2
Trajectory Determination

As a primary objective of this thesis, this chapter introduces trajectory determination algorithms to be used in FSD competition, specifically for estimating the deviation from track centerline based on stereo-optic camera. Section 2.1 gives brief information about track specifications in the competition. Then, MATLAB's Driving Scenario Designer Apps is presented in Section 2.2. This toolbox is used to create sample tracks with respect to given specifications. Section 2.3, on the other hand, discusses how to ensure that the vehicle is going on centerline of track. Finally, an algorithm developed for estimation of deviation from centerline is proposed in Section 2.4.

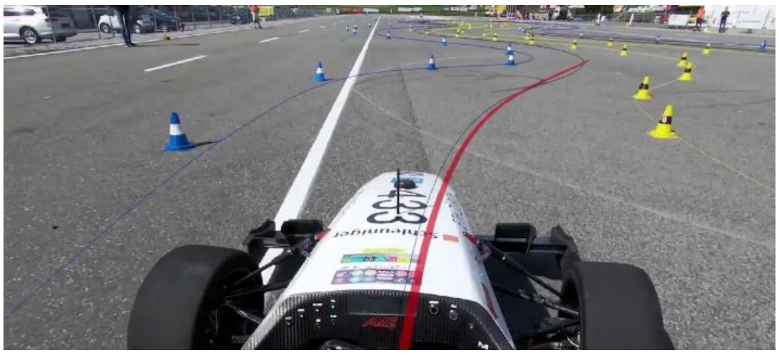


Figure 2.1: Centerline Driving in FSD Competition [6]

2.1 Track Specifications

In this section, the track specifications determined by FSD committee is introduced. In order to develop sufficient algorithms for trajectory planning, it crucial to have a track which has same properties with actual track used in FSD competition. Otherwise, designed algorithms won't be beneficial in real race situations.

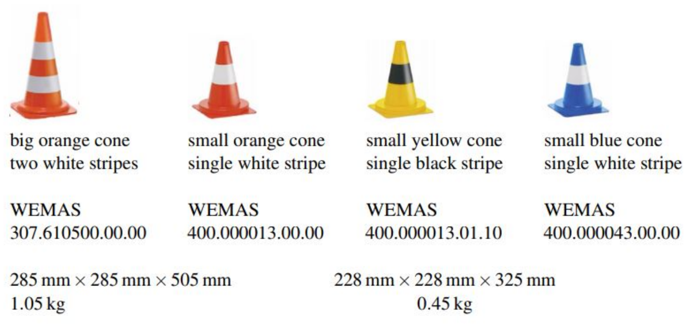


Figure 2.2: Cone Specs [7]

Official FSD Handbook[7] states that the tracks in the competition are supposed to have following characteristics:

- The track layout is realized with cones. Size and properties of cones are shown in Fig. 2.2.
- The left lane of the track are marked with small blue cones.
- The right lane of the track are marked with small yellow cones.
- Exit and entry lanes are marked with small orange cones.
- The maximum distance between two cones pairs in driving direction is 5 meters. However, in order to have better indication, this distance is reduced in corners.

According to given track specification, the below figure visualize the track layout description.

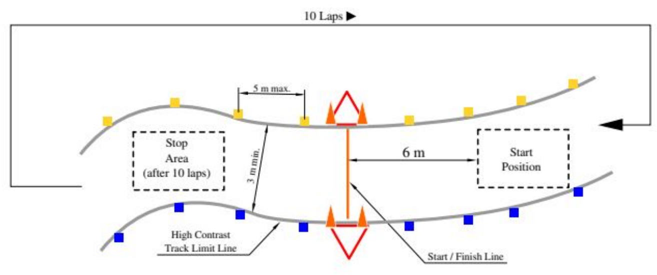


Figure 2.3: Example Track Visual [7]

2.2 Simulation Framework Overview

As mentioned beginning of this chapter, it is essential to have a track configured according to specifications in previous section. Therefore, one of the sub-objectives of thesis is to design simulation framework for racing track design. In that sense, this thesis employs Driving Scenario Designer Apps (DSDA) [25] to create required simulation environment, and this section is dedicated to basic explanation over how to use DSDA for the purpose of track configuration. This application includes sufficient tools and features to generate desired tracks in order to be used when developing algorithms.

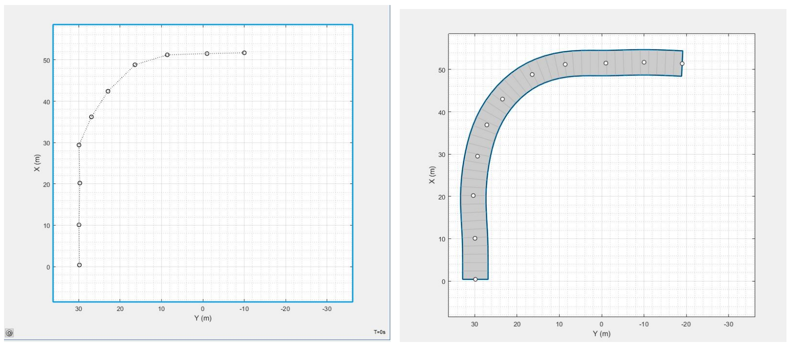


Figure 2.4: Adding Road

To create desired by using Driving Scenario Apps, first thing is to determine the track layout. It is done by clicking on the canvas throughout the desired path as shown in Fig. 2.4. Followed by this, cones are added to left and right lanes of the road. Since there is no specific **Add Cone** option in the application, **Add Barrier** feature is used and the shape and size are adjusted as defined in Section 2.1. Since there is no possibility to create all the barriers in yellow and blue colors, the left and right cones are marked as odd index and even index respectively.

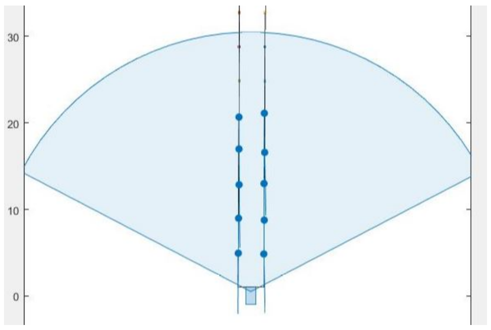


Figure 2.5: Detection of Cones with Camera

After obtaining the desired sample track with cones, the vehicle is presented into the simulation environment, and camera is placed on top the vehicle with preferred orientation and camera settings. The camera is the crucial feature of DSDA for this thesis. The major reason is that it returns the relative distances of the detected objects with respect to camera coordinate frame. Since the determining of cones position is not included in this work, that is quite beneficial feature when developing algorithms. Camera also provides feedback about vehicle position and velocity.

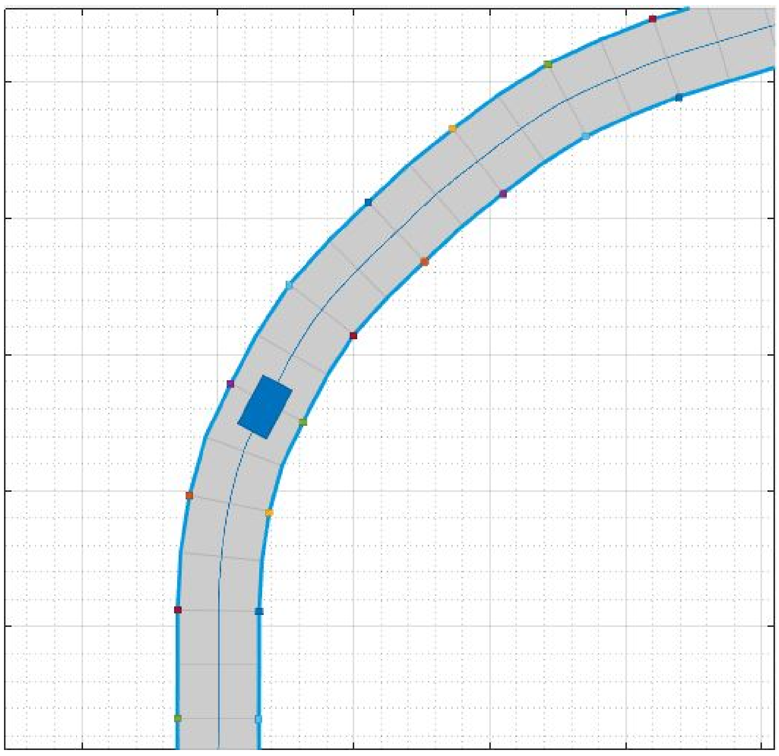


Figure 2.6: Running Simulation

As a final step, the reference path for the vehicle is defined. The path can be drawn on the road through desired direction. After all, the simulation is run and vehicle starts to move over reference path. Once finished, all the data can be exported to work space to be used for developing centerline and its deviations estimation algorithms.

2.3 Centerline Estimation

This section focuses on developing algorithms to indicate whether or not vehicle is going on centerline. In FSD competition, track map is not provided, so there is pre-information for path planning. In this regard, in order to obtain actual track map, going through track centerline in first or first two laps has great importance. Also, as stated earlier, the estimation is mainly depend on camera measurement.

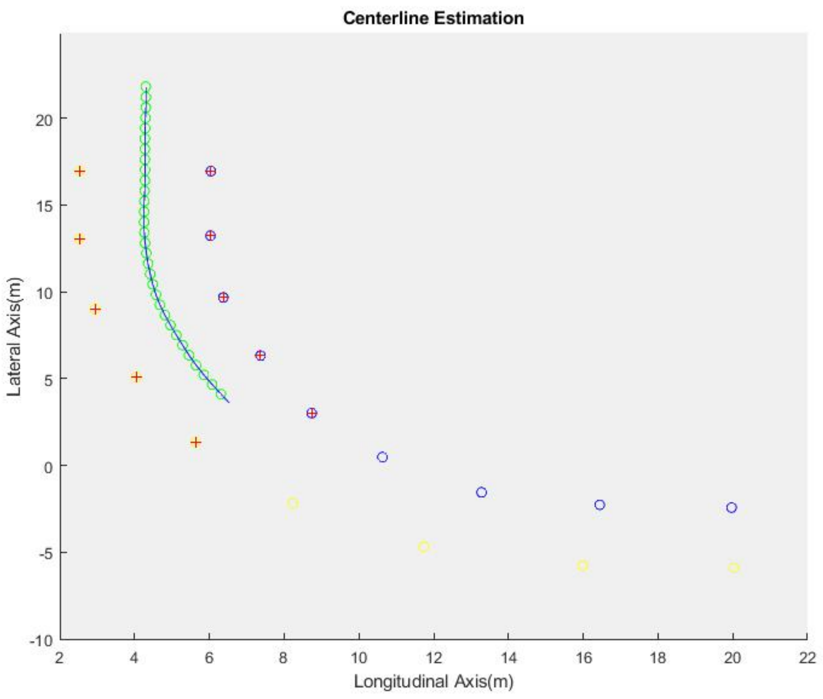


Figure 2.7: Centerline Driving Scenario

Using DSDA, a simple algorithm is proposed, first, to check the centrality of vehicle. The idea is to compare the distance to left and right target cones at each time instant. If the difference between is small enough, then it can be said that the vehicle is travelling in centerline. Also, when distance between vehicle and mid-points of target cone pairs is below certain threshold, it shifts to next one. Fig. 2.7 represents the results of first test, where green circle indicates that vehicle is in centerline at that current step. Blue and yellow circles, also, symbolize cones. From the figure, it can be pointed out that the simple algorithm works well when vehicle is moving on track centerline.

After that, another simulation test is performed in which vehicle follows wild motion path rather than centerline. Then, the resulting plot is obtained as in shown Fig. 2.8. In the plot, the black line going through mid-points of the cone pairs represents the exact track centerline, and red circle refers that vehicle deviates from centerline of track. Other symbols have same meanings.

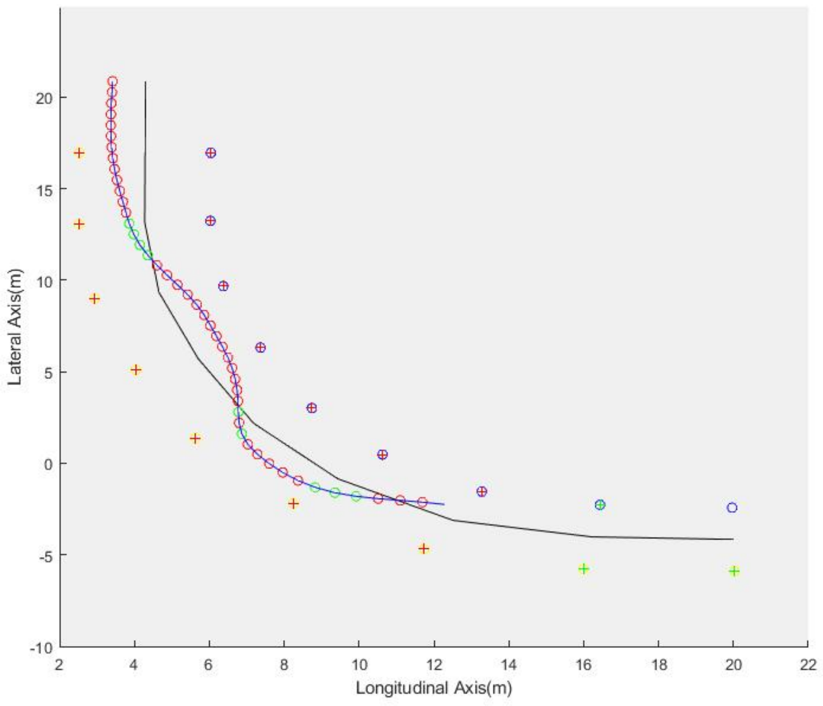


Figure 2.8: Wild Driving Scenario

From the plot, it is obvious that when vehicle is not on the black line, the algorithm returns green circle as if vehicle is on centerline. This proves that algorithms is not work well, so some adjustments have to be made to correctly estimate centerline check. In that sense, the method was completely changed and new method was developed. The main idea behind the new method is to create a circular corridor through centerline of track and check whether or not the vehicle crosses or inside that corridor. If the condition is satisfied, this would imply that vehicle is at the centerline of track. In order to implement the algorithm, first, the distance between mid-points of past and target cone pairs are linearly interpolated, and circles are generated centered at each of those points with same radius. Then, at each time step, algorithm find the nearest circle with respect to vehicle position. However, for crossing condition, not only the nearest circle is considered, the neighbour circles at each sides are taken into account as well.By doing so, algorithm becomes more robust. Fig. 2.9 represents the new method.

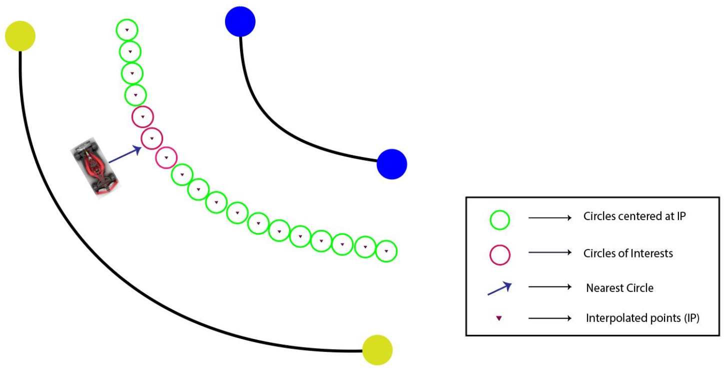


Figure 2.9: New Method

Apart from these, the new method has one controversial aspect. In this method, it is essential that circles are generated such that there would be no empty space between each other. In order to achieve this, there are two options, increasing either radius or number of interpolating points. However, increasing radius increase the threshold distance to check the centerline condition, the second option was employed in this work. After careful tests, it was realized that interpolating 10-15 points would return quite satisfied results.

On the other hand, when analyzing the results, it was also realized that condition for shifting to next cone pair shows undesired results at some points and needs some improvements as well. To overcome this problem, the condition was updated. The new idea is that instead of looking the distance between vehicle position and middle of target cone pair, the rate of change of that distance is considered. In this case, when vehicle is approaching to the mid-point regardless of which direction, the rate would be negative sign. Then, once the sign become positive, it would means that vehicle just crossed the lane between target cone pair, and algorithms shifts to next cone pair.

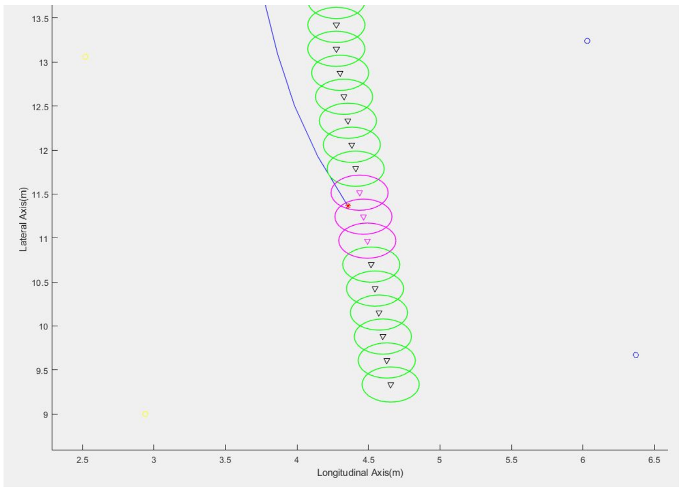


Figure 2.10: Implementation of New Method

After having necessary updates in the algorithm, it was tested by running same simulation again. Fig. 2.10 visualize the plot which was obtained during the test. As can be seen form the figure, algorithm seems working as desired. It is capable of find circles of interest in each time step and check whether vehicle cross or inside any of them. This is a clear confirmation that, on contrast to the previous version, the updated algorithm is able to successfully detect whether or not vehicle travels on centerline in more robust way. The pseudo-code of the final version of the algorithm is shown below.

Algorithm 1: Centerline Estimation

```
Data: Vehicle Position and Cone Positions
Result: True if vehicle is on Centerline, False Otherwise
Initialization;
ConeNo ⟵ Target Cone Pair Number;
r ⟵ Radius of Circles;
n ⟵ Interpolation Size;
interpolatedpoints ⟵ Interpolate points from start position to
  middle of target cone pair;
middlepoint ⟵ Calculate Middle Point of Target Cone Pair;
Then, at each step $k$;
while vehicle moves do
    VehiclePosition ⟵ Get Vehicle Position at time $k$ ;
    distance2mid ⟵ Calculate Rate of Change of Distance to
      middlepoint using VehiclePosition;
    if Sign(distance) == Positive then
        ConeNo ⟵ ConeNo + 1;
        interpolatedpoints ⟵ Interpolate New Points from middle
          of past cone pair to middle of new cones pair ;
        middlepoint ⟵ Calculate New Middle Point ;
    else
        Do Nothing
    end
    Center of Circles ⟵ interpolatedpoints ;
    C1 ⟵ Find Closest Circle to Vehicle Position ;
    C2 and C3 ⟵ Find Neighbours Circles at both sides ;
    CircleOfInterest ⟵ C1, C2 and C3 ;
    CrossCheck ⟵ $CircleOfInterest, VehiclePosition, r$ ;
    if CrossCheck == True then
        return True;
    else
        return False;
    end
end
```

2.4 Estimation of Deviation from Centerline

As mentioned before, it is crucial to track centerline at the beginning of the race in order to obtain the track map accurately. In the same sense, estimation of how far the vehicle is deviating from centerline is also one of the main considerations for this thesis. Previous section describes the method to detect when vehicle deviates from centerline, but it does not calculate the amount. Therefore, this section presents how to estimate deviation from track centerline, known as cross-track error along with heading error. The main idea behind estimating cross-track error is to formulate the problem as lane keeping assist system. That would be also basis for lateral control design. In this case, desired lane refers to centerline of the track. Fig. 2.11 depicts the cross-track error. The proposed method depends on [26] and [27].

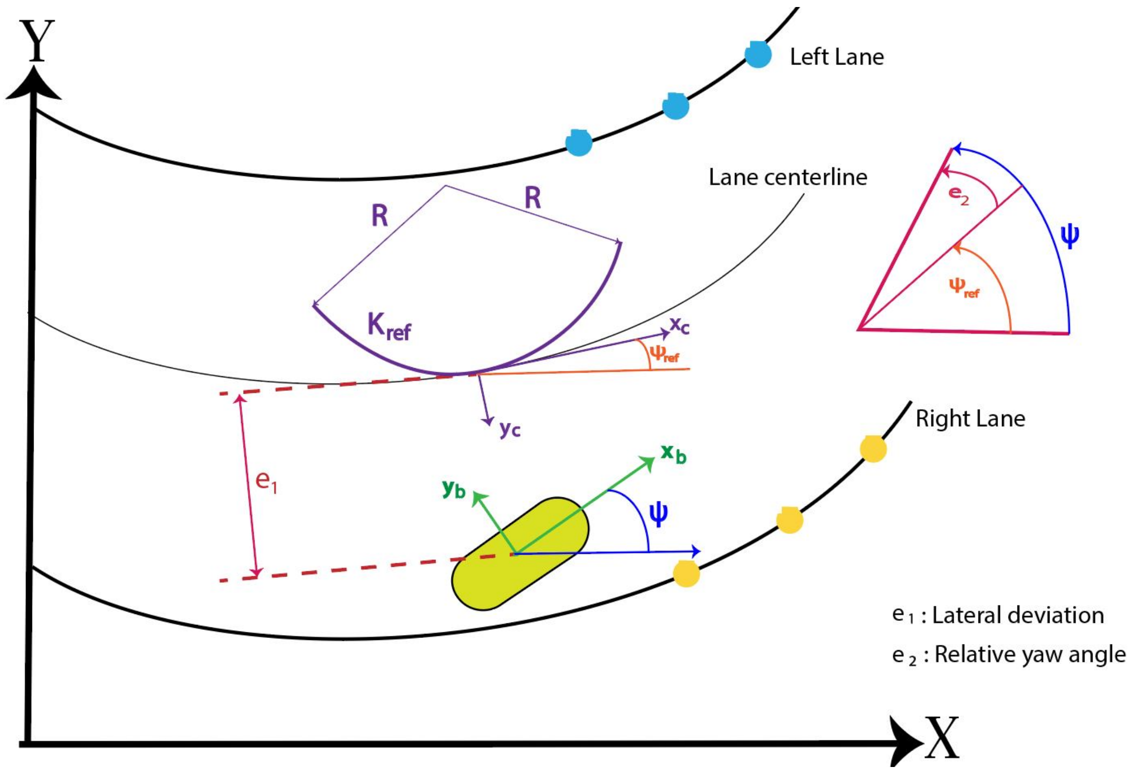


Figure 2.11: Deviation from Centerline

The cross-track is formulated as the projection of vehicle velocity in from body coordinate frame into the radial direction y_c which is perpendicular to reference heading pf the curve ψ_{ref}. Then, the equation which describe rate of cross-track error is obtained as;

$$\dot{e}_{dev} = \dot{x}_b \sin(\psi - \psi_{ref}) + \dot{y}_b \cos(\psi - \psi_{ref}) \tag{2.1}$$

where e_{dev} is crosstrack error, $\dot{x}_b$ and $\dot{y}_b$ represent vehicle velocities in body frame, and ψ and ψ_{ref} are vehicle and reference curve heading respectively. Since vehicle is assumed to travel along the reference curve closely, it can be say that the difference between the heading of vehicle and reference would be small enough such the differential formula in Eq. 2.1 can be linearized using small angle theorem. Then, formula is linearized as follows

$$\begin{aligned} \dot{e}_{dev} &= \dot{x}_b \sin(\psi - \psi_{ref}) + \dot{y}_b \cos(\psi - \psi_{ref}) && (2.2) \\ &= \underbrace{\dot{x}_b}_{V_x} \underbrace{(\psi - \psi_{ref})}_{e_{yaw}} + \underbrace{\dot{y}_b}_{V_y} && (2.3) \\ &= V_x e_{yaw} + V_y && (2.4) \end{aligned}$$

where e_{yaw} refers to heading error between vehicle heading and reference curve heading. As can be seen from Eq.2.4, the cross-track error is function of heading error, e_{yaw}. Therefore, heading error needs to be formalized as well. From [27], the rate of reference heading ψ_{ref} is defined as

$$\dot{\psi}_{ref} = V_x \rho \tag{2.5}$$

where ρ is curvature of the desired curve. Then, differential equation for heading error can be expressed as

$$\dot{e}_{yaw} = \dot{\psi} - \underbrace{V_x \rho}_{\dot{\psi}_{ref}} \tag{2.6}$$

On the other hand, this thesis claims to estimate the deviation with using stereo-optic camera. The inputs, vehicle velocities V_x and V_y in body frame, and heading of vehicle ψ can be provided by camera using visual odometry , as well as by other sensors like GPS or IMU. In that sense, particular role of the camera to estimate the deviation is coming from estimation the curvature of track centerline . As can be seen from Fig. 2.11, the reference curve is approximated as circle which is tangent to and has same radius with the curve. Then, the curvature of corresponding point can be found as

$$\kappa = \frac{1}{R_{circle}} \tag{2.7}$$

where κ is curvature and R_{circle} is radius of approximated circle. In that sense, next section would introduce how to calculate curvature of desired curve in order to estimate cross-track error accurately.

Algorithm 2: Estimation of Deviation from Centerline

```
Data: Yaw Rate, Vehicle Speeds at Body Frame, and Road Curvature
Result: Heading Error and Cross-track Error
1  initialization;
2  Same as in Algorithm 1 ;
3  Then, at each step $k$;
4  while vehicle moves do
5      CircleOfInterest $\longleftarrow$ Run Algorithm 1;
6      Estimate Road Curvature using $CircleOfInterest$ points ;
7      Get Yaw Rate and Vehicle Speeds at Body Frame ;
8      Heading Error $\longleftarrow$ Solution of Eq. 2.6 ;
9      Cross-track Error $\longleftarrow$ Solution of Eq. 2.4 ;
10 end
```

2.5 Curvature Estimation

Curvature estimation of curvy road is one of the main challenges in lane keeping assist systems. As described in previous section, it is the key input to calculate cross-track error. Therefore, this sections introduces two methods regarding how to find curvature of the desired curve.In this case, instead of approximating lane, the position of the cones are used to calculate the curvature of the track centerline.

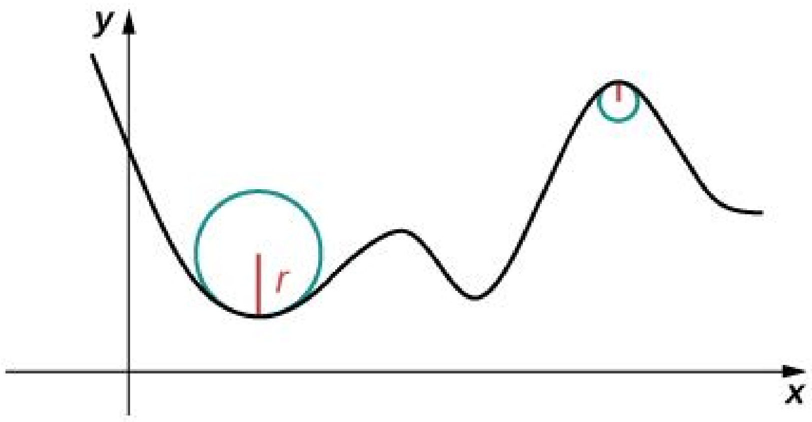


Figure 2.12: Curvature of Curve

2.5.1 Definition of Curvature of Curve

The definition of curvature is that it is a measure of how sharply a smooth curve turn. More sharper turn yields to bigger curvature. If the curve is circle, it has constant. However, in order to find the curvature of curve in general case, the curve is approximated as a circle at given point such that it is tangent to and has same radius with the curve. As depicted in Fig. 2.12, the approximated circle, known as also osculating circle, hugs the curve as closely as possible since the 2 curves have the same tangent and radius at the point where they meet. As a result, the curvature of curve at given point is equal to curvature of approximated circle. In this regard, next two subsection are dedicated to different methods of finding curvature.

2.5.2 Parabolic Approximation

This subsection focuses on first method to calculate curvature of curve. Since vehicle is moving 2D plane, the curve is represented as $y = f(x)$. If curve is the graph of a function and both $\dot{y}$ and $\ddot{y}$ exists, then the curvature at given point can be calculated as [28][29]

$$\kappa = \frac{\left[1 + (\frac{dy}{dx})^2\right]^{\frac{3}{2}}}{\mid \frac{d^2y}{dx^2} \mid} \tag{2.8}$$

However, the function which describe the curve is not known. A sufficient way to find the curve formula is to curve fitting with parabola using 3 consecutive data point. These data points can be found by interpolating the distance between mid-point of two cones pairs as described in Section 2.3. In general, a parabola is described as

$$y = ax^2 + bx + c \tag{2.9}$$

At each point $(x_p$,$y_p)$, the point itself and the points on either side of it are substituting into Eq. 2.9 gives a linear system consisting of 3 equations with 3 unknowns

$$\begin{bmatrix} x1^2 & x1 & 1 \\ x2^2 & x2 & 1 \\ x3^2 & x3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix} \tag{2.10}$$

Finally, solving this linear system, taking first and second derivative of parabolic equation, curvature at the point $(x_p$,$y_p)$ is computed as

$$R(x_p) = \frac{\left[1 + (2ax_p + b)^2\right]^{\frac{3}{2}}}{\mid 2a \mid} \tag{2.11}$$

$$\kappa(x_p) = \frac{1}{R(x_p)} \tag{2.12}$$

2.5.3 Circle Approximation

Previous method finds the curvature with parabola approximation. As different from that, this sections represents another method to calculate curvature at each point on the path. This method is called as Circle Approximation which is an exact method for finding the required radius of curvature. As in previous method, the method needs 3 points to approximate circle which is tangent to and has same radius with the path at given point. In literature, there are various way to how to approximate the circle with given 3 points. This work employs the method in [30]. Some other methods can be also found in [31] and [32].

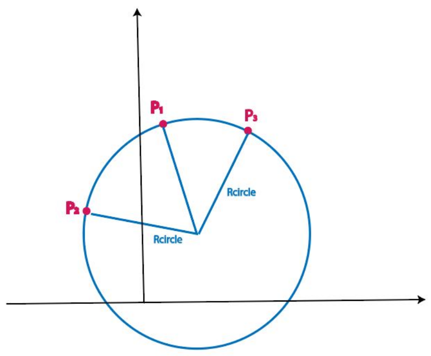


Figure 2.13: Approximated Circle through which the points pass

General form of circle equation is

$$x^2 + y^2 + 2ax + 2by + c = 0 \tag{2.13}$$

Plugging the three points into the equation

$$x_1^2 + y_1^2 + 2ax_1 + 2by_1 + c = 0 \tag{2.14}$$
$$x_2^2 + y_2^2 + 2ax_2 + 2by_2 + c = 0 \tag{2.15}$$
$$x_3^2 + y_3^2 + 2ax_3 + 2by_3 + c = 0 \tag{2.16}$$

Then, these three equations are formulated as in Eq. 2.10 and solved for unknowns, a, b and c. After obtaining unknown parameters, the radius of circle and curvature are calculated as

$$x_c = -a \tag{2.17}$$
$$y_c = -b \tag{2.18}$$
$$R_c = \sqrt{x_c^2 + y_c^2 - c} \tag{2.19}$$
$$\kappa = \frac{1}{R_c} \tag{2.20}$$

2.5.4 Comparison of Methods

Subsections 2.5.2 and 2.5.3 covers the description of two methods over curvature estimation of path. To find which method is superior to the other one, two test were performed. In the first one, elliptical circle was generated. Corresponding result is shown in Fig. 2.14.

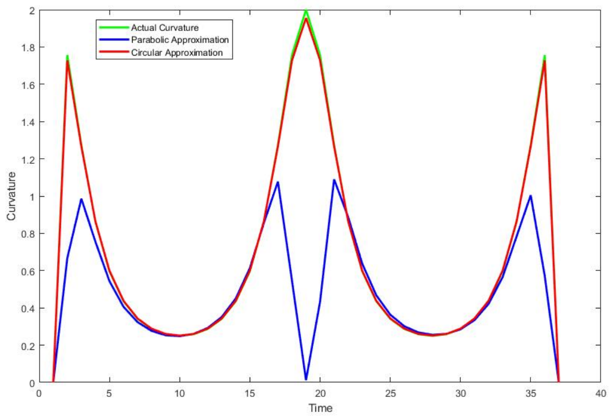


Figure 2.14: Curvature Estimation from Elliptical Circle

Above figure indicates that circular approximation work dramatically better than parabolic approximation. It returns almost same as actual values. However, one test is not enough to analyze the performance. Therefore, another test were applied in which a random curve was created.

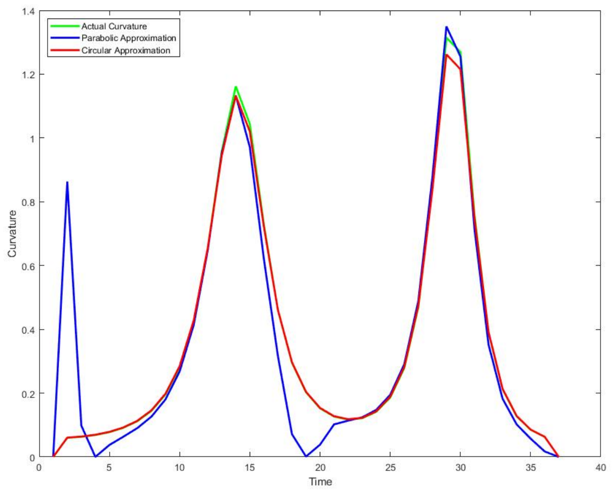


Figure 2.15: Curvature Estimation from Random Curve

Fig. 2.15 implies that circular method is still superior to parabolic method. As a result, circular approximation was selected to be used in experiments for curvature estimation.

Chapter 3
Vehicle Modelling

This chapter focuses on mathematical model of vehicle motion which is one of the core elements for model-based controller design using model predictive control. Section 3.1 gives brief discussion about kinematic bicycle model which purely refer to geometric motion of a vehicle. In Section 3.2, a simplified single-track dynamic model is introduced. It consists of a rigid body of mass m and inertia I_z, along with a front and read wheels. The dynamics is assumed to be planar thus all lifting, rolling and pitching motions are neglected. The essential part of dynamic model is the interaction between tires and road. Therefore, the brief overview of tire modelling and its effect on vehicle dynamic motion can be found in Section 3.3

3.1 Kinematic Model

This section gives detailed explanations of kinematic model of the vehicle which is commonly used at low speeds . Kinematic model focuses on geometric properties of vehicle and does not consider dynamics of the vehicle, such as tire forces, mass and inertia. The fundamental modelling assumption is that the vehicle has perfect road handling and moves without slipping sideways. This assumptions is valid as long as lateral acceleration is low enough and lateral force exerted by each tire are negligible. Also, during steady-state turning, the wheels are moving along concentric circular paths, aligned with the circle tangents and with no lateral slip. Therefore, it is possible to represent the two left and right wheels by a single wheel located in the center of the axle [33].

For kinematic model, the equations of motion are based on [34], which also compares kinematic and dynamic models in terms of control design, [35] and [36]. The nonlinear equations that describe kinematic bicycle model in an inertial frame are

$$\dot{x} = v\cos(\psi + \beta) \tag{2.1}$$
$$\dot{y} = v\sin(\psi + \beta) \tag{2.2}$$
$$\dot{\psi} = \frac{v}{l_r}\sin(\beta) \tag{2.3}$$
$$\dot{v} = a \tag{2.4}$$
$$\dot{\beta} = \tan^{-1}\left(\frac{l_r}{(l_f + l_r)}\tan(\delta)\right) \tag{2.5}$$

where x and y are the coordinates of the center of mass in an inertial frame. ψ is the inertial heading and v is the speed of the vehicle. l_f and l_r represent the distance from the center of the mass of the vehicle to the front and rear axles, respectively. β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the car. a is the acceleration of the center of mass in the same direction as the velocity. The control inputs are the front and rear steering angles δ_f and acceleration a. Since in most vehicles the rear wheels cannot be steered, δ_r is assumed to be zero.

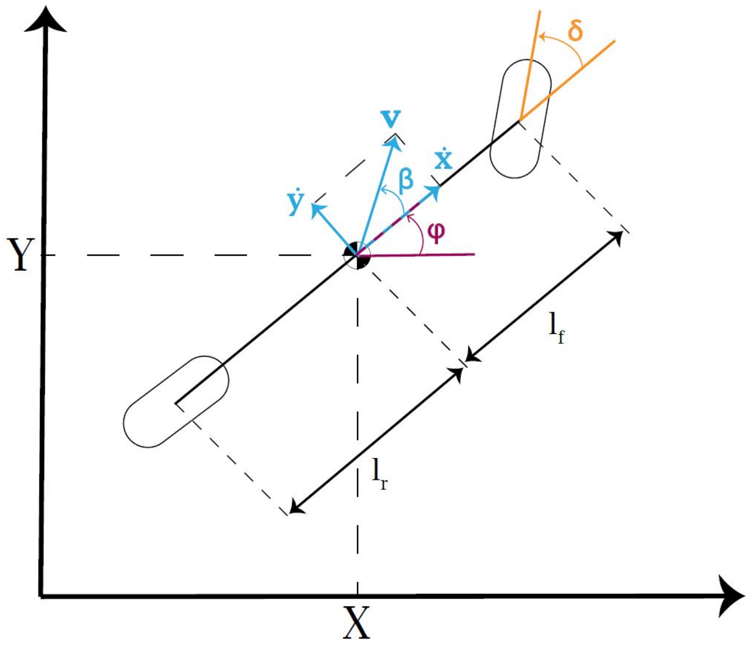


Figure 3.1: Kinematic Model

The main advantage of kinematic model is that compared to higher fidelity vehicle models, the system identification on the kinematic bicycle model is easier because there are only two parameters to identify, l_f and l_r.This makes it simpler to port the same controller or path planner to other vehicles with differently sized wheelbases[34].

3.2 Dynamical Model

Kinematic model in Section 3.1 is beneficial only if vehicle speed is low enough and there is no slipping to sideways. However, it would be inefficient when vehicle start to lose road handling between tires and road. Therefore, this section introduces dynamical model of vehicle motion which is mainly depend on forces exerted by each tire. The presented single-track model is based on [18],[37], and [38].

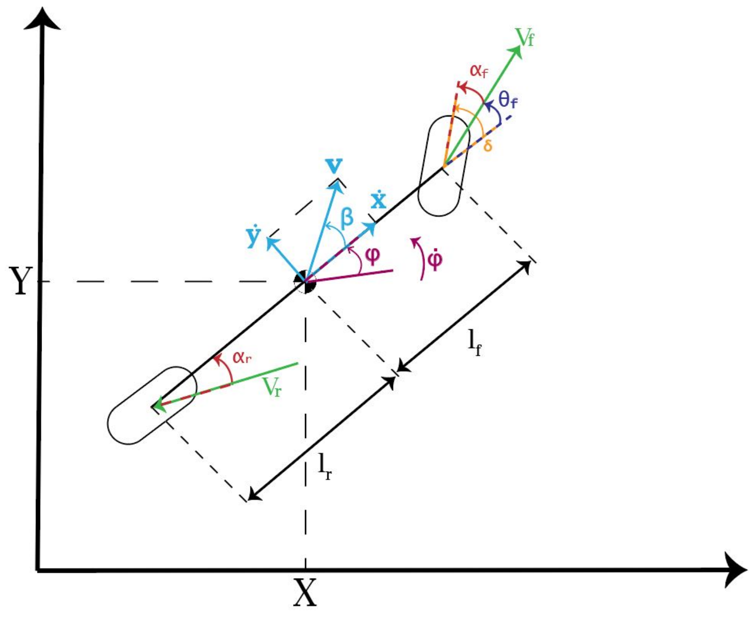


Figure 3.2: Dynamic Model

As similar in kinematic model, the single-track dynamical model is valid under following assumptions[39],[18];

- All lifting, rolling, and pitching motion is neglected.
- Vehicle mass is assumed to be concentrated at the center of gravity.
- Front and rear tires are represented as one single tire on each axle. Imaginary contact points of tires and surface are assumed to lie along the center of axles.
- Pneumatic trail and aligning torque resulting from a side-slip angle of a tire are neglected.
- Mass distribution on the axles is assumed to be constant.
- As the used cars are rear wheel driven and do not have active brakes, the longitudinal force on the front wheel is neglected

The governing differential equations for dynamical model are

$$\ddot{x} = \frac{1}{m}(F_{r,x} - F_{f,y}\sin\delta) + \dot{y}\dot{\psi} \tag{2.6}$$

$$\ddot{y} = \frac{1}{m}(F_{r,y} + F_{f,y}\cos\delta) - \dot{x}\dot{\psi} \tag{2.7}$$

$$\ddot{\psi} = \frac{1}{I_z}(F_{f,y}l_f\cos\delta - F_{r,y}l_r) \tag{2.8}$$

$$\dot{X} = \dot{x}\cos(\psi) - \dot{y}\sin(\psi) \tag{2.9}$$

$$\dot{Y} = \dot{x}\sin(\psi) + \dot{y}\cos(\psi) \tag{2.10}$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral speeds in the body frame, and $\dot{\psi}$ denotes the yaw rate in inertial frame. While m and I_z denote the vehicle's mass and yaw inertia respectively, l_f and l_r represent the distance from the center of the mass of the vehicle to the front and rear axles. F terms represents tire force exerted by each tire in tire coordinate frame. Therefore, they have to be transformed to body-fixed coordinate frame by rotation of δ. The tire forces will be handled more detailed in Section 3.3.

The main advantage of dynamical is that since it includes more sophisticated dynamic and cover larger envelope in terms of speed and acceleration, it reflects a realistic behaviour of vehicle better than kinematic model. This would increase the accuracy of model predictive control design which will be discussed later.

3.3 Tire Model

As can be seen from Section 3.2, the essential part of dynamical model of vehicle depends on tire forces. In this regard, this section gives briefly overview for tire modelling approach. Tire modelling , which represents interaction between the tires and road surface, is the biggest challenge in modelling and control design for vehicle. There are a lot of such model in nowadays, which estimate longitudinal and lateral tire forces based on vehicle states. In this work, only the most famous one which known as Pacejka's Magic Tire model is introduced in Section 3.3.1. The detailed information regarding the model can be found in [40] and [41]. The fine summary of the model is also in [42]. On the other hand, due to fact the model proposed by Pacejka is highly non-linear, especially at cornering situations, the model has to be linearized in order to be used in model-based control design. In that sense, Linear Tire model is discussed in Section 3.3.2.

3.3.1 Pacejka's Tire Model

This subsection gives description for Pacjecka's tire model. It is a complex semi-empirical model being able to describe the nonlinear behaviour of tire forces under wide operation range. The original formula consists of more than 20 coeffients. But, the model then is simplified and reduced to 4 main parameters. The approximate value of these main parameters is estimated by fitting the formula to empirical measurements of the tire behavior. On the other hand, the main inputs of model are of the tire normal force, slip ratio, slip angle and surface friction coefficient. As a result, both longitudinal and lateral tire forces can be computed in straightforward way. However, since the longitudinal tire force is neglected with the assumption in Section 3.2, only the lateral tire forces are calculated.

The analytical equations for the lateral tire forces are

$$a_f = \delta - \arctan\left(\frac{\dot{\psi} l_f + \dot{y}}{\dot{x}}\right) \tag{2.11}$$

$$a_r = -\arctan\left(\frac{\dot{\psi} l_r - \dot{y}}{\dot{x}}\right) \tag{2.12}$$

$$F_{f,y} = DF_{z,f}\sin(C\arctan(Ba_f - E(Ba_f - \arctan(Ba_f)))) \tag{2.13}$$

$$F_{r,y} = DF_{z,r}\sin(C\arctan(Ba_r - E(Ba_r - \arctan(Ba_r)))) \tag{2.14}$$

where B, C, D and E are set of shaping coefficients, $F_{z,f}$ and $F_{z,r}$ are the wheel loads for both tires, and a_f and a_r are side-slip angles of front and rear tire respectively.

The parameter D represents the peak value that a tire force can be. The parameter C determines the shape around the peak value. The B is the stiffness factor of tire. Finally, the parameter E describes curve shape. Fig. 3.3 shows the magnitude of lateral tire force for front tire with respect to side slip angle a, with varying wheel load F_z.

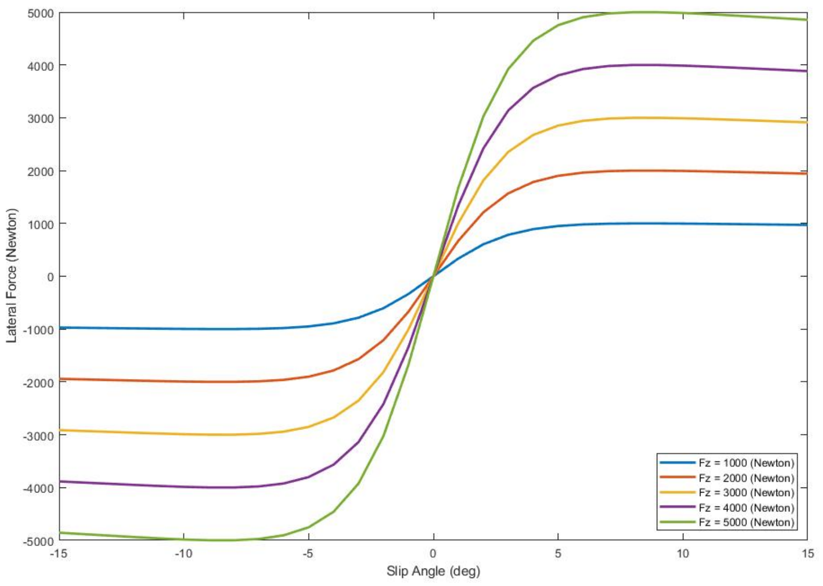


Figure 3.3: Lateral Tire Forces

3.3.2 Linear Tire Model

From Section 3.3.1, it is obvious that considering slip-angle phenomenon with having realistic tire model is a key factor when the vehicle reached to its limit at cornering situations. However, creating a realistic tire behavior for model-based control design using nonlinear tire model is quite challenging and complex. That's why, in this section, the linearized tire model of Pacejka's Magic Formula is introduced to be used in control design later. The major assumption behind the linearized model is that it is only valid if slip angle has small value. Figure X also proves the assumption, From the graph, it is seen that nonlinear model behaves linearly at small angles.

Using small angle approximation, the nonlinear equations regarding slip angles become

$$a_f = \delta - \arctan\left(\frac{\dot{\psi} l_f + \dot{y}}{\dot{x}}\right) \approx -\frac{\dot{y} + l_f\dot{\psi}}{\dot{x}} + \delta \tag{2.15}$$

$$a_r = -\arctan\left(\frac{\dot{\psi} l_r - \dot{y}}{\dot{x}}\right) \approx -\frac{\dot{y} - l_r\dot{\psi}}{\dot{x}} \tag{2.16}$$

Then, the lateral tire forces are linearly approximated by following formula;

$$F_{f,y} \approx C_f a_f \tag{2.17}$$

$$F_{r,y} \approx C_r a_r \tag{2.18}$$

where a_r and a_f are linearized slip angles, and C_f and C_r refers to cornering stiffness factor of each tire which corresponds to coefficients in original Pacejka model. The cornering stiffness factors can be found by taking partial derivative of 2.13 and 2.14 with respect to corresponding slip angles, and evaluated at zero angle. Figure 3.4 compares the linearized model with original and can be seen that assumption is valid under small angles.

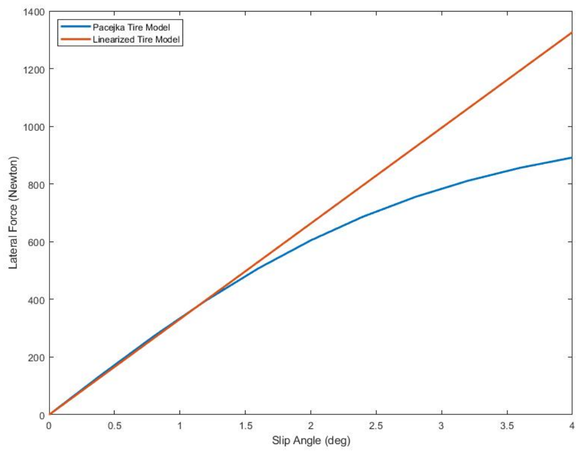
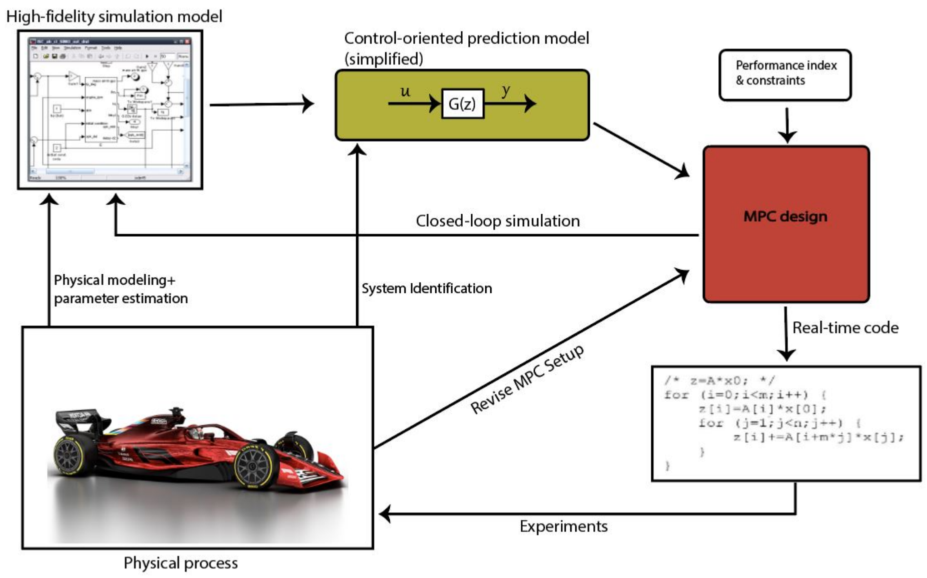


Figure 3.4: Linearized Tire Model

Chapter 4

Model Predictive Control

Model Predictive Control(MPC) is one of the widely used control design method in process industry. MPC 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. In this regard, this section is dedicated to comprehensive guide over MPC. More detailed information can be found in [43] and [44]. [45], [44] and [33] provides sufficient summaries and study materials. Fig. ?? represent basic workflow of general MPC design.


High-fidelity simulation model
Control-oriented prediction model (simplified)
u
G(z)
y
Performance index & constraints
MPC design
Closed-loop simulation
Physical modeling+ parameter estimation
System Identification
Real-time code
Revise MPC Setup
Experiments
Physical process

```
/* z=A*x0; */
for (i=0;i<m;i++) {
     z[i]=A[i]*x[0];
     for (j=1;j<n;j++) {
          z[i]+=A[i+m*j]*x[j];
     }
}
```

Figure 4.1: MPC Design Flow

Section 4.1 gives overview of working principles of MPC.Then, In Section 4.2, basic principles of the controller is presented. Finally, the detailed formulation of MPC problem can be found in Section 4.3.

4.1 Working Principle of MPC

This section basically discuss the working principles of MPC. It is an optimal control problem which tries to solve optimization problem shown in Figure below at each time step. As each controller, the goal of MPC is to calculate the input to the plant such that the plant output follows a desired reference. To achieve this, the main strategy that MPC employs is to compute this input which depends the prediction of the future system outputs.

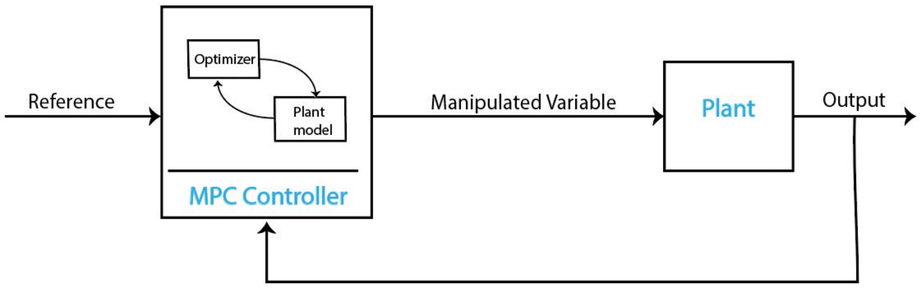


Figure 4.2: MPC Structure

At the current time, the MPC controller uses plant model to simulate the plant behaviour over the next N step. N is called as prediction horizon which is a measure of how far ahead MPC looks into the future. The MPC needs to find the best predicted output which has the lowest residual value over N. It simulates multiple future scenarios like this in an systematic way. This is the place where the optimizer comes into the problem. The key task of optimizer is to minimize the cost function in such a way that it minimize the error between output and reference and change in input increment while satisfying limit conditions.

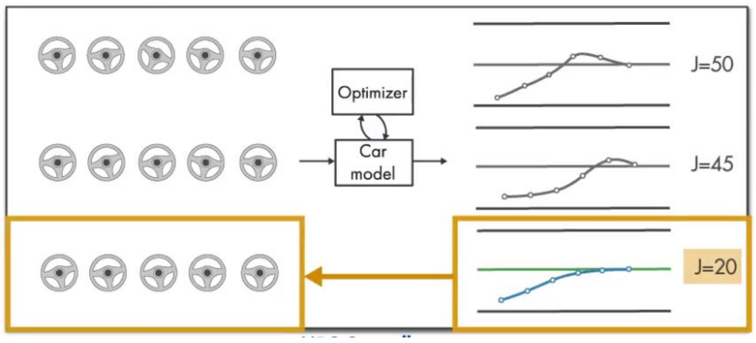


Figure 4.3: MPC Cost Function

The predicted output with the smallest J value gives the optimal solution and therefore determines the optimal input sequence through prediction horizon which will make the plant output as close as possible to desired value. However, applying all sequence to the system would results in open-loop control which makes system fragile to disturbance and model uncertainties.

In order to overcome this, after finding the optimal sequence, MPC applies only first step in the sequence and disregards the rest. By doing so, it turns the control problem into closed loop feedback system. This would be covered in more detail in next section. After that, in the next time step, the finite horizon window is shifted and the optimization is repeated.

$$
\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \|W^y(y_k - r(t))\|_2^2 + \|W^u(u_k - u_r(t))\|_2^2 \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \qquad \text{prediction model} \\
& y_k = g(x_k, u_k) \\
& u_{\min} \leq u_k \leq u_{\max} \qquad \text{constraints} \\
& y_{\min} \leq y_k \leq y_{\max} \\
& x_0 = x(t) \qquad \text{state feedback}
\end{aligned}
$$

Figure 4.4: MPC Formulation [8]

On the other hand, as can be seen from figure above, the cost function J involves weights which penalize both states and inputs. As a primary control design objective, it is aimed to both track desired reference with a smallest error possible with having smooth control moves. In order to balance between these competing objectives, one can adjust the weights corresponding to input rate with respect to output weights. If the ratio of input rate weights over output weights is higher than 1, it results in more aggressive system response. Otherwise, it means to put more emphasize on robustness of a system. Additionally, weights of these two groups are not only adjusted relative to each other, but adjust relative within the same groups as well, For example, if a system has two states and it is more critical to perform reference tracking of the first output than the second output, we assign a larger weight to the first output and the ratio between the outputs is greater than 1.

4.2 Properties of MPC

This section briefly introduces the the main properties of model predictive control. Here are some reasons which clearly expresses why it is demanding control system in the industry, especially in autonomous vehicle field.

First one is that MPC can handle multi-input multi-output(MIMO) systems that have interactions between their inputs and outputs. Suppose a MIMO system in which a change in first output also affect the second one. If PID controller was chosen, it would quite challenging to tune the gains because the two control loops would operate independent of each other as if there is no relation between them. The difficulty would increase for bigger systems as they would require too much controller gains to be tuned. In same sense, the advantage of MPC is that the controller is structured in such a way that it can control the outputs simultaneously by taking into account of all the interactions between system states.

Another feature of MPC is that it can handle constraints. Constraint issue is one of the important design consideration in control system design, because violating them can result in undesired outcomes. MPC takes the user-defined constraints into consideration and returns controller output accordingly.

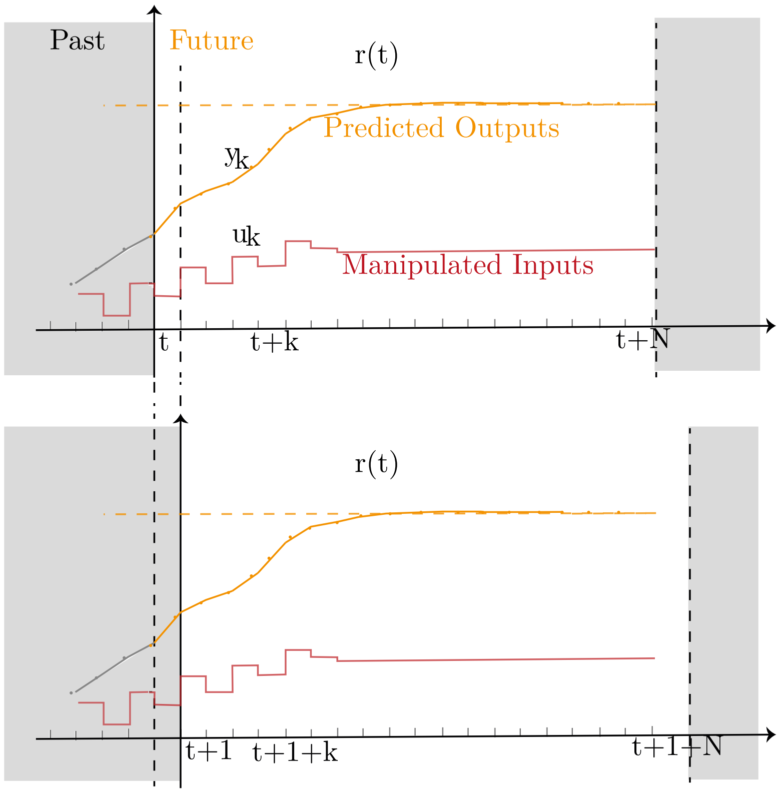


Figure 4.5: Preview Capability

The other unique strenth of MPC is its preview capability. Thanks to this feature, the controller can observe what will come in future and adjust its output based on the previewed references. This will considerably enhance both controller and system performance. To give an example, imagine that an autonomous racing car travel on curvy track. If the controller doesn't know that corner is coming ahead, it would be only able to apply brakes while its taking the corner. That would results in too much brake forces that is not desired in racing. However, with utilizing preview capability, the car controller would be informed about the corner. Then, it can brake early and exit from the corner with higher speed.

MPC has all these benefits, but as a cost of all of them, it has one main disadvantage.This is the one that it requires a powerful, fast processor with large memory. The reason is that MPC solves an online optimization problem at each step. However, thanks to increasing development in embedded design field, this problem stands no longer as a huge barrier for implementation of MPC as it was before.

4.3 General MPC Formulation

This section gives a detailed guide over formulation of MPC problem. The presented methods are based on [46],[8] and [47]. The main aim is to formulate the problem as convex optimization problem as in the below equations such that it has only one global optimal point and does not generate any infeasible solution. Also, since this thesis considers tracking, not regulation, only the methodology regarding tracking is discussed.

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{H}\mathbf{x} + \mathbf{F}^{\mathbf{T}}\mathbf{x} && (4.1) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} && (4.2) \end{aligned}$$

4.3.1 Simultaneous(Sparse)

First method for MPC problem formulation is Simultaneous approach, known as also Sparse. As may be understood from the name itself, it tries to optimize both system states and control input at the same time. Formulation begins with defining a linear state-space model which is represented as

$$\begin{aligned} \mathbf{x_{k+1}} &= \mathbf{A}\mathbf{x_k} + \mathbf{B}\mathbf{u_k} && (4.3) \\ \mathbf{y_k} &= \mathbf{C}\mathbf{x_k} && (4.4) \end{aligned}$$

where **A** is state matrix, **B** is input matrix and **C** is output matrix. Since it is assumed that there is no direct through, **D** term in Eq. 4.4 is neglected.

Given linear model, general MPC formulation over prediction horizon, **N**, at discrete time is

$$\begin{aligned} \underset{\mathbf{u_t,\dots,u_{t+N-1},x_{t+1},\dots,x_{t+N}}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{x_N^T S x_N} + \frac{1}{2}\sum_{\mathbf{k=1}}^{\mathbf{N-1}}\left(\mathbf{x_k^T Q x_k} + \mathbf{u_k^T R u_k}\right) && (4.5) \\ \text{subject to} \quad & \mathbf{x_{k+1}} = \mathbf{A}\mathbf{x_k} + \mathbf{B}\mathbf{u_k}, && (4.6) \\ & \mathbf{x_t} = \mathbf{given} && (4.7) \\ & \mathbf{x_{min}} \leq \mathbf{x_k} \leq \mathbf{x_{max}}. && (4.8) \\ & \mathbf{u_{min}} \leq \mathbf{u_k} \leq \mathbf{u_{max}}. && (4.9) \end{aligned}$$

For tracking, it is aimed to minimize error between reference and output rather than states. That's why, states variables in the formula are replaced with error terms. Then, the cost function turns into following form,

$$\mathbf{J} = \frac{1}{2}\mathbf{e}_{t+N}^{T}\mathbf{S}\mathbf{e}_{t+N} + \frac{1}{2}\sum_{k=1}^{N-1}\left(\mathbf{e}_{t+k}^{T}\mathbf{Q}\mathbf{e}_{t+k} + \mathbf{u}_{t+k}^{T}\mathbf{R}\mathbf{u}_{t+k}\right) \tag{4.10}$$

However, this leads to another problem. In steady state, error terms would go to zeros, but in general, input term wont be zero. If input term is minimized, that would make error term to increase. In order to solve this problem, it is aimed to minimize the change in the input increment instead. In this sense, input **u** first is defined in such way that it would sum of input increment and input at previous time. By doing so, it becomes possible to employ input at previous time, $\mathbf{u_{k\text{-}1}}$, as new state variable.

$$\triangle\mathbf{u_k} = \mathbf{u_k} - \mathbf{u_{k-1}} \tag{4.11}$$

$$\mathbf{u_k} = \underbrace{\mathbf{u_{k-1}}}_{\text{new extra state}} + \underbrace{\triangle\mathbf{u_k}}_{\text{new control input}} \tag{4.12}$$

Then, original system is augmented with the new control state as follows

$$\begin{bmatrix}\mathbf{x_{k+1}}\\ \mathbf{x^u_{k+1}}\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}}_{\tilde{\mathbf{A}}}\underbrace{\begin{bmatrix}\mathbf{x_k}\\ \mathbf{x^u_k}\end{bmatrix}}_{\tilde{\mathbf{x}}} + \underbrace{\begin{bmatrix}\mathbf{B}\\ \mathbf{I}\end{bmatrix}}_{\tilde{\mathbf{B}}}\triangle\mathbf{u_k} \tag{4.13}$$

$$\mathbf{y_k} = \underbrace{\begin{bmatrix}\mathbf{C} & \mathbf{0}\end{bmatrix}}_{\tilde{\mathbf{C}}}\begin{bmatrix}\mathbf{x_k}\\ \mathbf{x^u_k}\end{bmatrix} \tag{4.14}$$

Having augmented system and defining e = ref - output, the cost function in Eq. 4.10 is rearranged

$$\begin{aligned}\mathbf{J} &= \frac{1}{2}(\mathbf{r}_{t+N} - \tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+N})^{T}\mathbf{S}(\mathbf{r}_{t+N} - \tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+N})\\ &+ \frac{1}{2}\sum_{k=0}^{N-1}(\mathbf{r}_{t+k} - \tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+k})^{T}\mathbf{Q}(\mathbf{r}_{t+k} - \tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+k}) + \triangle\mathbf{u}_{t+k}^{T}\mathbf{R}\triangle\mathbf{u}_{t+k}\end{aligned} \tag{4.15}$$

Expanding the equation, eliminating constant terms and collecting the same terms into matrices as in Eq. 4.17,

$$\begin{aligned}\mathbf{J} &= \frac{1}{2}\mathbf{r}_{t+N}^{T}\mathbf{S}\mathbf{r}_{t+N} - \mathbf{r}_{t+N}^{T}\mathbf{S}\tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+N} + \frac{1}{2}\tilde{\mathbf{x}}_{t+N}^{T}\tilde{\mathbf{C}}^{T}\mathbf{S}\tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+N}\\ &+ \frac{1}{2}\sum_{k=0}^{N-1}\Bigg(\frac{1}{2}\mathbf{r}_{t+k}^{T}\mathbf{Q}\mathbf{r}_{t+k} - \mathbf{r}_{t+k}^{T}\mathbf{Q}\tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+k} + \frac{1}{2}\tilde{\mathbf{x}}_{t+k}^{T}\tilde{\mathbf{C}}^{T}\mathbf{Q}\tilde{\mathbf{C}}\tilde{\mathbf{x}}_{t+k}\\ &+ \triangle\mathbf{u}_{t+k}^{T}\mathbf{R}\triangle\mathbf{u}_{t+k}\Bigg)\end{aligned} \tag{4.16}$$

$$\mathbf{r} = \begin{bmatrix} \mathbf{r_t} \\ \mathbf{r_{t+1}} \\ \vdots \\ \mathbf{r_N} \end{bmatrix} \qquad \tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\mathbf{x}}_{\mathbf{t+1}} \\ \tilde{\mathbf{x}}_{\mathbf{t+2}} \\ \vdots \\ \tilde{\mathbf{x}}_{\mathbf{t+N}} \end{bmatrix} \qquad \triangle\mathbf{u} = \begin{bmatrix} \triangle\mathbf{u_t} \\ \triangle\mathbf{u_{t+1}} \\ \vdots \\ \triangle\mathbf{u_{t+N-1}} \end{bmatrix} \tag{4.17}$$

Overall optimization problem would look like,

$$\underset{\tilde{\mathbf{x}},\triangle\mathbf{u}}{\text{minimize}} \quad \frac{\mathbf{1}}{\mathbf{2}}\tilde{\mathbf{x}}^{\mathbf{T}} \underbrace{\begin{bmatrix} \tilde{\mathbf{C}}^{\mathbf{T}}\mathbf{Q}\tilde{\mathbf{C}} & & \\ & \ddots & \\ & & \tilde{\mathbf{C}}^{\mathbf{T}}\mathbf{Q}\tilde{\mathbf{C}} \end{bmatrix}}_{\bar{\bar{\mathbf{Q}}}} \tilde{\mathbf{x}} - \mathbf{r}^{\mathbf{T}} \underbrace{\begin{bmatrix} \mathbf{Q}\tilde{\mathbf{C}} & & \\ & \ddots & \\ & & \mathbf{Q}\tilde{\mathbf{C}} \end{bmatrix}}_{\bar{\bar{\mathbf{T}}}} \tilde{\mathbf{x}} + \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}} \underbrace{\begin{bmatrix} \mathbf{R} & & \\ & \ddots & \\ & & \mathbf{R} \end{bmatrix}}_{\bar{\bar{\mathbf{R}}}} \triangle\mathbf{u} \tag{4.18}$$

$$\text{subject to} \quad \tilde{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0} & & & \\ \tilde{\mathbf{A}} & \mathbf{0} & & \\ & \ddots & \ddots & \\ & & \tilde{\mathbf{A}} & \mathbf{0} \end{bmatrix}}_{\bar{\bar{\mathbf{A}}}} \tilde{\mathbf{x}} + \underbrace{\begin{bmatrix} \tilde{\mathbf{B}} & & \\ & \ddots & \\ & & \tilde{\mathbf{B}} \end{bmatrix}}_{\bar{\bar{\mathbf{B}}}} \triangle\mathbf{u} + \underbrace{\begin{bmatrix} \tilde{\mathbf{A}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\bar{\bar{\mathbf{A}}}_{\mathbf{0}}} \tilde{\mathbf{x}}_{\mathbf{0}} \tag{4.19}$$

Additionally, the inequality condition in Eq. 4.19 is re-ordered again and formed as

$$\tilde{\mathbf{x}} = \bar{\bar{\mathbf{A}}}\tilde{\mathbf{x}} + \bar{\bar{\mathbf{B}}}\triangle\mathbf{u} + \bar{\bar{\mathbf{A}}}_{\mathbf{0}}\tilde{\mathbf{x}}_{\mathbf{0}} \tag{4.20}$$

$$\mathbf{0} = \underbrace{\begin{bmatrix} (\bar{\bar{\mathbf{A}}} - \mathbf{I}) & \bar{\bar{\mathbf{B}}} \end{bmatrix}}_{\bar{\bar{\mathbf{a}}}} \underbrace{\begin{bmatrix} \tilde{\mathbf{x}} \\ \triangle\mathbf{u} \end{bmatrix}}_{\bar{\tilde{\mathbf{x}}}} + \underbrace{\bar{\bar{\mathbf{A}}}_{\mathbf{0}}\tilde{\mathbf{x}}_{\mathbf{0}}}_{\bar{\bar{\mathbf{b}}}} \tag{4.21}$$

Finally, MPC problem for Simultaneous form gets convex form to be solved as optimization problem.

$$\underset{\bar{\tilde{\mathbf{x}}}}{\text{minimize}} \quad \underbrace{\begin{bmatrix} \tilde{\mathbf{x}} \\ \triangle\mathbf{u} \end{bmatrix}^{\mathbf{T}}}_{\bar{\tilde{\mathbf{x}}}} \underbrace{\begin{bmatrix} \bar{\bar{\mathbf{Q}}} & \mathbf{0} \\ \mathbf{0} & \bar{\bar{\mathbf{R}}} \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} \tilde{\mathbf{x}} \\ \triangle\mathbf{u} \end{bmatrix} + \underbrace{\begin{bmatrix} -\mathbf{r}^{\mathbf{T}}\bar{\bar{\mathbf{T}}} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{F^T}} \begin{bmatrix} \tilde{\mathbf{x}} \\ \triangle\mathbf{u} \end{bmatrix} \tag{4.22}$$

$$\text{subject to} \quad \mathbf{0} = \bar{\bar{\mathbf{a}}}\bar{\tilde{\mathbf{x}}} + \bar{\bar{\mathbf{b}}} \tag{4.23}$$

4.3.2 Sequential(Dense)

As can be seen, Simultaneous MPC formulation tries to minimize both states and control increment. However, this method is less efficient in terms of complexity. In this regard, this section present another method, which is called as Sequential or Dense. This method focuses only on minimizing input increment. By doing so, the complexity of formulation would be reduced. Therefore, this thesis considers only sequential method.
Sequential formulation starts with eliminating the state term by expressing it as function of $\triangle\mathbf{u}$ and $\tilde{\mathbf{x}}$. In that sense, Eq. 4.19 is redefined as

$$\tilde{\mathbf{x}} = \underbrace{\begin{bmatrix} \tilde{\mathbf{B}} & & & & \\ \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{B}} & & & \\ \tilde{\mathbf{A}}^2\tilde{\mathbf{B}} & \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{B}} & & \\ \vdots & & & \ddots & \\ \tilde{\mathbf{A}}^{\mathbf{N}-1}\tilde{\mathbf{B}} & \dots & \dots & \dots & \tilde{\mathbf{B}} \end{bmatrix}}_{\bar{\bar{\mathbf{C}}}} \triangle\mathbf{u} + \underbrace{\begin{bmatrix} \tilde{\mathbf{A}} \\ \tilde{\mathbf{A}}^2 \\ \tilde{\mathbf{A}}^3 \\ \vdots \\ \tilde{\mathbf{A}}^{\mathbf{N}} \end{bmatrix}}_{\hat{\hat{\mathbf{A}}}} \tilde{\mathbf{x}}_0 \tag{4.24}$$

Then, putting Eq. 4.24 into equation Eq. 4.18, the cost function becomes

$$\begin{aligned} \mathbf{J} = \frac{\mathbf{1}}{\mathbf{2}}(\bar{\bar{\mathbf{C}}}\triangle\mathbf{u} + \hat{\hat{\mathbf{A}}}\tilde{\mathbf{x}}_0)^{\mathbf{T}}\bar{\bar{\mathbf{Q}}}(\bar{\bar{\mathbf{C}}}\triangle\mathbf{u} + \hat{\hat{\mathbf{A}}}\tilde{\mathbf{x}}_0) + \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}}\bar{\bar{\mathbf{R}}}\triangle\mathbf{u} \\ - \mathbf{r}^{\mathbf{T}}\bar{\bar{\mathbf{T}}}(\bar{\bar{\mathbf{C}}}\triangle\mathbf{u} + \hat{\hat{\mathbf{A}}}\tilde{\mathbf{x}}_0) \end{aligned} \tag{4.25}$$

Expanding and ignoring constant term as done previously, final form of sequential, without constraints, would be;

$$\underset{\triangle\mathbf{u}}{\text{minimize}} \quad \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}}\underbrace{(\bar{\bar{\mathbf{C}}}^{\mathbf{T}}\bar{\bar{\mathbf{Q}}}\bar{\bar{\mathbf{C}}} + \bar{\bar{\mathbf{R}}})}_{\bar{\bar{\mathbf{H}}}}\triangle\mathbf{u} + \begin{bmatrix} \tilde{\mathbf{x}}_0^{\mathbf{T}} & \mathbf{r}^{\mathbf{T}} \end{bmatrix} \underbrace{\begin{bmatrix} \hat{\hat{\mathbf{A}}}^{\mathbf{T}}\bar{\bar{\mathbf{Q}}}\bar{\bar{\mathbf{C}}} \\ -\bar{\bar{\mathbf{T}}}\bar{\bar{\mathbf{C}}} \end{bmatrix}}_{\bar{\bar{\mathbf{F}}}^{\mathbf{T}}}\triangle\mathbf{u} \tag{4.26}$$

On the other hand, solution to unconstrained formulation is as follows;

$$\mathbf{J} = \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}}\bar{\bar{\mathbf{H}}}\triangle\mathbf{u} + \begin{bmatrix} \tilde{\mathbf{x}}_0^{\mathbf{T}} & \mathbf{r}^{\mathbf{T}} \end{bmatrix}\bar{\bar{\mathbf{F}}}^{\mathbf{T}}\triangle\mathbf{u} \tag{4.27}$$

$$\nabla\mathbf{J}_{\triangle\mathbf{u}} = \bar{\bar{\mathbf{H}}}\triangle\mathbf{u} + \bar{\bar{\mathbf{F}}}\begin{bmatrix} \tilde{\mathbf{x}}_0 \\ \mathbf{r} \end{bmatrix} = \mathbf{0} \tag{4.28}$$

$$\triangle\mathbf{u} = -\bar{\bar{\mathbf{H}}}^{-1}\bar{\bar{\mathbf{F}}}\begin{bmatrix} \tilde{\mathbf{x}}_0 \\ \mathbf{r} \end{bmatrix} \tag{4.29}$$

4.3.3 Adding Constraints

Previous two subsections discusses the methods to formulate the general MPC problem without considering constraints. However, as expressed in Section 4.2, constraints are key element of MPC. Therefore, this subsection introduces how to define the constraints conditions into MPC problem. The ultimate goal is to define the constraints, such that overall convex optimization problem is formed as;

$$\underset{\triangle\mathbf{u}}{\text{minimize}} \qquad \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}}\bar{\bar{\mathbf{H}}}\triangle\mathbf{u}+\begin{bmatrix}\tilde{\mathbf{x}}_{\mathbf{0}}^{\mathbf{T}} & \mathbf{r}^{\mathbf{T}}\end{bmatrix}\bar{\bar{\mathbf{F}}}^{\mathbf{T}}\triangle\mathbf{u} \tag{4.30}$$

$$\text{subject to} \qquad \mathbf{G}\triangle\mathbf{u}\leq\mathbf{W}+\mathbf{S}\tilde{\mathbf{x}}_{\mathbf{0}} \tag{4.31}$$

In MPC problem, there are three terms that can be constrained, which are input increment, input and output. Adding constraints to input increment is pretty straightforward and shown in below equation

$$\underbrace{\begin{bmatrix}\mathbf{I} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \dots & \mathbf{0}\\ \vdots & & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}\\ -\mathbf{I} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & -\mathbf{I} & \dots & \mathbf{0}\\ \vdots & & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & -\mathbf{I}\end{bmatrix}}_{\mathbf{G}_{\triangle\mathbf{u}}}\triangle\mathbf{u} \leq \underbrace{\begin{bmatrix}\triangle\mathbf{u}_{\mathbf{max}}\\ \triangle\mathbf{u}_{\mathbf{max}}\\ \vdots\\ \triangle\mathbf{u}_{\mathbf{max}}\\ -\triangle\mathbf{u}_{\mathbf{min}}\\ -\triangle\mathbf{u}_{\mathbf{min}}\\ \vdots\\ -\triangle\mathbf{u}_{\mathbf{min}}\end{bmatrix}}_{\mathbf{W}_{\triangle\mathbf{u}}}+\underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\end{bmatrix}}_{\mathbf{S}_{\triangle\mathbf{u}}}\underbrace{\begin{bmatrix}\mathbf{x}_{\mathbf{k}}\\ \mathbf{u}_{\mathbf{k-1}}\end{bmatrix}}_{\tilde{\mathbf{x}}_{\mathbf{0}}} \tag{4.32}$$

As similar to input increment, constraints for input is also quite painless. The input $\mathbf{u}_{\mathbf{k}}$ can be expressed trough horizon $\mathbf{N}$ in such a way

$$\begin{aligned}\mathbf{u}_{\mathbf{k}} &= \mathbf{u}_{\mathbf{k-1+\triangle u_k}}\\ \mathbf{u}_{\mathbf{k+1}} &= \mathbf{u}_{\mathbf{k}}+\triangle\mathbf{u}_{\mathbf{k+1}}=\mathbf{u}_{\mathbf{k-1}}+\triangle\mathbf{u}_{\mathbf{k}}+\triangle\mathbf{u}_{\mathbf{k+1}}\\ \mathbf{u}_{\mathbf{k+2}} &= \mathbf{u}_{\mathbf{k+1}}+\triangle\mathbf{u}_{\mathbf{k+2}}=\mathbf{u}_{\mathbf{k-1}}+\triangle\mathbf{u}_{\mathbf{k}}+\triangle\mathbf{u}_{\mathbf{k+1}}+\triangle\mathbf{u}_{\mathbf{k+2}}\\ \vdots &= \vdots\\ \mathbf{u}_{\mathbf{k+N-1}} &= \mathbf{u}_{\mathbf{k-1}}+\triangle\mathbf{u}_{\mathbf{k}}+\dots+\triangle\mathbf{u}_{\mathbf{k+N-1}}\end{aligned} \tag{4.33}$$

Then, using the using this form, constraints equations for input term is configured as

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & & \ddots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ -\mathbf{I} & -\mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & & \ddots & \mathbf{0} \\ -\mathbf{I} & -\mathbf{I} & -\mathbf{I} & \dots & -\mathbf{I} \end{bmatrix}}_{\mathbf{G_u}} \triangle\mathbf{u} \leq \underbrace{\begin{bmatrix} \mathbf{u_{max}} \\ \mathbf{u_{max}} \\ \vdots \\ \mathbf{u_{max}} \\ -\mathbf{u_{min}} \\ -\mathbf{u_{min}} \\ \vdots \\ -\mathbf{u_{min}} \end{bmatrix}}_{\mathbf{W_u}} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & -\mathbf{I} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \dots & -\mathbf{I} & -\mathbf{I} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & -\mathbf{I} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{I} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{I} \end{bmatrix}}_{\mathbf{S_u}} \underbrace{\begin{bmatrix} \mathbf{x_k} \\ \mathbf{u_{k-1}} \end{bmatrix}}_{\tilde{\mathbf{x}}_0} \tag{4.34}$$

In contrast to input increment and input terms, adding constraints for output is relatively more complex and tricky. In this sense, first define the inequality conditions for general case at which optimization variable is $\mathbf{x}$. Then, convert it to desired case so that optimization variable would be $\triangle\mathbf{u}$.

General case is,

$$\underbrace{\begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{CA^{N-1}B} & \dots & \mathbf{CAB} & \mathbf{CB} \end{bmatrix}}_{\bar{\mathbf{g}}_{\mathbf{ymax}}} \mathbf{u} \leq \underbrace{\begin{bmatrix} \mathbf{y_{max}} \\ \mathbf{y_{max}} \\ \vdots \\ \mathbf{y_{max}} \end{bmatrix}}_{\mathbf{W_{ymax}}} - \underbrace{\begin{bmatrix} \mathbf{CA} \\ \mathbf{CA^2} \\ \vdots \\ \mathbf{CA^N} \end{bmatrix}}_{\bar{\mathbf{s}}_{\mathbf{ymax}}} \tag{4.35}$$

Then, using the same manner in Eq. 4.33, the Eq. in Eq. 4.35 is rewritten

$$\bar{\mathbf{g}}_{\mathbf{ymax}} \begin{bmatrix} \triangle\mathbf{u_k} + \mathbf{u_{k-1}} \\ \triangle\mathbf{u_{k+1}} + \triangle\mathbf{u_k} + \mathbf{u_{k-1}} \\ \vdots \\ \triangle\mathbf{u_{k+N-1}} + \dots + \triangle\mathbf{u_k} + \mathbf{u_{k-1}} \end{bmatrix} \leq \mathbf{W_{ymax}} - \bar{\mathbf{s}}_{\mathbf{ymax}} \tag{4.36}$$

After that, in order to form it as in Eq. 4.31, $\mathbf{u}$ and $\mathbf{u_{k\text{-}1}}$ are splinted

$$\bar{\mathbf{g}}_{\mathbf{ymax}} \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \end{bmatrix} \triangle\mathbf{u} + \underbrace{\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix}}_{\mathbf{I_u}} \mathbf{u_{k-1}} \right) \leq \mathbf{W_{ymax}} - \bar{\mathbf{s}}_{\mathbf{ymax}} \tag{4.37}$$

Finally, overall form of upper bound inequality conditions for output constraints becomes

$$\underbrace{\left(\mathbf{\bar{g}_{ymax}}\begin{bmatrix}\mathbf{I} & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{0}\\ \vdots & & \ddots & \vdots\\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I}\end{bmatrix}\right)}_{\mathbf{G_{ymax}}}\triangle\mathbf{u} \;\leq\; \mathbf{W_{ymax}} + \underbrace{\begin{bmatrix}-\mathbf{\bar{s}_{ymax}} & (\mathbf{\bar{g}_{ymax}}\mathbf{I_u})\end{bmatrix}}_{\mathbf{S_{ymax}}}\underbrace{\begin{bmatrix}\mathbf{x_k}\\ \mathbf{u_{k-1}}\end{bmatrix}}_{\mathbf{\tilde{x}_0}} \tag{4.38}$$

The conditions regarding lower bound of output constraints can be found by following same way. After defining constraints in each case, complete form of MPC problem with having constraints is defined as

$$\underset{\triangle\mathbf{u}}{\text{minimize}} \qquad \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}}\bar{\bar{\mathbf{H}}}\triangle\mathbf{u} + \begin{bmatrix}\mathbf{\tilde{x}_0^T} & \mathbf{r^T}\end{bmatrix}\bar{\bar{\mathbf{F}}}^{\mathbf{T}}\triangle\mathbf{u} \tag{4.39}$$

$$\text{subject to} \qquad \underbrace{\begin{bmatrix}\mathbf{G_u}\\ \mathbf{G_{\triangle u}}\\ \mathbf{G_{ymax}}\\ \mathbf{G_{ymin}}\end{bmatrix}}_{\mathbf{G}}\triangle\mathbf{u} \;\leq\; \underbrace{\begin{bmatrix}\mathbf{W_u}\\ \mathbf{W_{\triangle u}}\\ \mathbf{W_{ymax}}\\ \mathbf{W_{ymin}}\end{bmatrix}}_{\mathbf{W}} + \underbrace{\begin{bmatrix}\mathbf{S_u}\\ \mathbf{S_{\triangle u}}\\ \mathbf{S_{ymax}}\\ \mathbf{S_{ymin}}\end{bmatrix}}_{\mathbf{S}} \tag{4.40}$$

Chapter 5

Lateral Control Design

This section introduces design of MPC for lateral motion controller of vehicle. It is based on lane keeping assist system as described in Chapter 2. The main purpose of lateral controller is to follow the centerline of the track with minimum error. Fig. 5.1 shows the model structure of lateral controller. From dynamical motion of vehicle described in Section 3.2 at Chapter 3, it is seen that longitudinal and lateral motions are coupled with each other, both of them are highly non-linear. Therefore, in order to design lateral controller with using MPC, first lateral motion dynamics has to be decoupled from longitudinal one and linearized. The linearized dynamics is then used as internal model in the controller. This is covered in Section 5.1. Then, Section 5.2 introduces how to formulate lateral control problem as lane keeping assist controller. Finally, 5.3 presents the design of controller with new system model. The overall structure of lateral controller is shown in below figure.

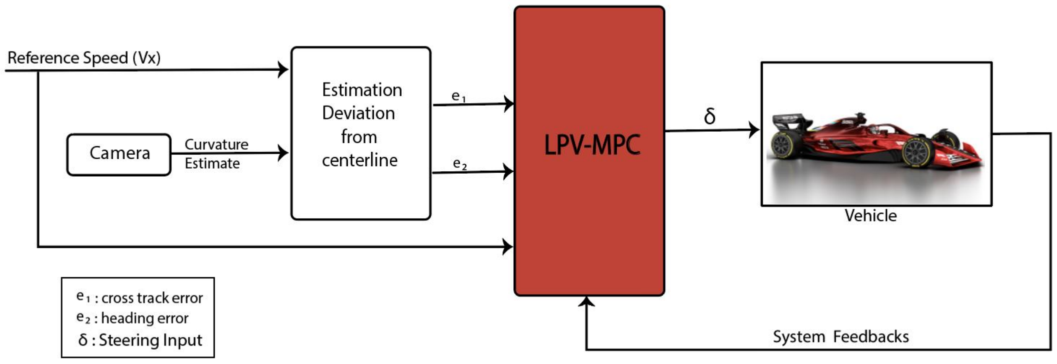


Figure 5.1: Lateral Controller Block Diagram

5.1 Linear Model for Lateral Motion

As described at the beginning of this chapter, lateral motion dynamics has to be decoupled and linearized. It starts with linearization of lateral terms of derived single-track dynamical model in Chapter 3. Also, longitudinal speed, $\dot{x}$, is considered as another input to the system and represented as V_x in order to decouple the longitudinal dynamics from lateral one.

Having linear tire model in chapter, rewriting Equations 2.7 and 2.8 with the ones 2.15, 2.16, 2.17 and 2.18 ,

$$\ddot{y} = \frac{1}{m}(C_f a_f \cos(\delta) + C_r a_r) - V_x\dot{\psi} \tag{5.1}$$

$$= \frac{1}{m}\left[C_f\left(\delta - \frac{\dot{y} + l_f\dot{\psi}}{V_x}\right)\cos(\delta) + C_r\left(-\frac{\dot{y} - l_r\dot{\psi}}{V_x}\right)\right] - V_x\dot{\psi} \tag{5.2}$$

$$\ddot{\psi} = \frac{1}{I_z}(C_f a_f l_f \cos(\delta) - C_r a_r l_r) \tag{5.3}$$

$$= \frac{1}{I_z}\left[C_f\left(\delta - \frac{\dot{y} + l_f\dot{\psi}}{V_x}\right)l_f\cos(\delta) - C_r\left(-\frac{\dot{y} - l_r\dot{\psi}}{V_x}\right)l_r\right] \tag{5.4}$$

Then, using small angle approximation and re-arranging the term in such a way that system states and inputs are lateral velocity and rate of change in heading,

$$\ddot{y} = \frac{1}{m}\left[C_f\delta - \frac{C_f\dot{y}}{V_x} - \frac{C_f l_f\dot{\psi}}{V_x} - \frac{C_r\dot{y}}{V_x} + \frac{C_r l_r\dot{\psi}}{V_x}\right] - V_x\dot{\psi} \tag{5.5}$$

$$= \left[-\frac{Cr + Cf}{mV_x}\right]\dot{y} + \left[\frac{C_r l_r - C_f l_f}{mV_x} - V_x\right]\dot{\psi} + \left[\frac{C_f}{m}\right]\delta \tag{5.6}$$

$$\ddot{\psi} = \frac{1}{I_z}\left[C_f l_f\delta - \frac{C_f l_f\dot{y}}{V_x} - \frac{C_f l_f^2\dot{\psi}}{V_x} + \frac{C_r l_r\dot{y}}{V_x} - \frac{C_r l_r^2\dot{\psi}}{V_x}\right] \tag{5.7}$$

$$= \left[\frac{C_r l_r - C_f l_f}{I_z V_x}\right]\dot{y} - \left[\frac{l_f^2 C_f + l_r^2 C_r}{I_z V_x}\right]\dot{\psi} + \left[\frac{C_f l_f}{I_z}\right]\delta \tag{5.8}$$

Finally, State Space form of general lateral dynamics extended with states, namely lateral position and heading angle in inertial frame;

$$\begin{bmatrix}\ddot{y}\\ \dot{y}\\ \ddot{\psi}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} -\frac{C_r+C_f}{mV_x} & 0 & \frac{C_r l_r - C_f l_f}{mV_x} - V_x & 0\\ 1 & 0 & 0 & 0\\ \frac{C_r l_r - C_f l_f}{I_z V_x} & 0 & -\frac{l_f^2 C_f + l_r^2 C_r}{I_z V_x} & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\dot{y}\\ y\\ \dot{\psi}\\ \psi\end{bmatrix} + \begin{bmatrix}\frac{C_f}{m}\\ 0\\ \frac{C_f l_r}{I_z}\\ 0\end{bmatrix}\delta \tag{5.9}$$

5.2 Augmented Model for Lateral Motion

In Section 5.1, the general linearized lateral model is presented. Taking previously derived model as basis, this sections introduces the more enhanced version of linearized lateral dynamics by formulating it as lane keeping problem as described in Section 2.4 at Chapter 2. Lane keeping assist system is one of famous ways to model the lateral motion dynamics of the vehicle to be used for model-based lateral control strategy. Nowadays, it is commonly implemented on-board in the autonomous vehicles to ensure that it is travelling in desired path. In this sense, LKA would be formulated as lateral model for path tracking. As stated before, the aim in this work is to follow the centerline of track. The presented method is based on [48]. The more comprehensive derivation can be also found at [49]. The key idea behind LKA is that it would augment the model in equation 5.9 with error dynamics composed of lateral deviation from desired path, known also cross-track error, and heading error.

From Section 2.4, the error dynamics is defined as follows,

$$\dot{e_y} = V_x e_\psi + \dot{y} \tag{5.10}$$

$$\dot{e_\psi} = \dot{\psi} - V_x \rho \tag{5.11}$$

where e_y and e_ψ are lateral deviation and heading error respectively, V_x is longitudinal velocity, $\dot{y}$ and ρ is road curvature. ρ comes to the error dynamics as measured disturbance(md) and will be handled in more detailed in next section.

After having error dynamics, it is augmented with previously linearized lateral dynamics. The final state space model is then formed as in equation below

$$\begin{bmatrix} \ddot{y} \\ \dot{y} \\ \ddot{\psi} \\ \dot{\psi} \\ \mathbf{\dot{e}_y} \\ \mathbf{\dot{e}_\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{C_r+C_f}{mV_x} & 0 & \frac{C_r l_r - C_f l_f}{mV_x} - V_x & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{C_r l_r - C_f l_f}{I_z V_x} & 0 & -\frac{l_f^2 C_f + l_r^2 C_r}{I_z V_x} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{V_x} \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}}_{\mathbf{A_{LKA}}} \begin{bmatrix} \dot{y} \\ y \\ \dot{\psi} \\ \psi \\ \mathbf{e_y} \\ \mathbf{e_\psi} \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{C_f}{m} \\ 0 \\ \frac{C_f l_r}{I_z} \\ 0 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B_{LKA}}} \delta + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{0} \\ \mathbf{-V_x} \end{bmatrix}}_{\mathbf{B_{md}}} \underbrace{\rho}_{\mathbf{md}} \tag{5.12}$$

5.3 Control Design

This section gives the overview of control design method for linear and augmented lateral model in equation 5.12. The design method is purely based on what is developed in Section 4.3 in Chapter 4. As can be seen from Eq. 5.12, the new system model employs the longitudinal velocity as internal parameter in matrix B_{md}. It means that the system dynamics would change as the the longitudinal velocity of vehicle varies. Therefore, it is necessary to update internal plant model in MPC in each time step. In this sense, Linear Parameter Varying(LPV) method is employed in lateral controller design. LPV model is a special case of a Linear Time Varying (LTV) model in which the internal model parameters changes over time. [50], [51] and [51] cover it in detail. Good summary can be also found in [47] and [52].

In traditional model predictive control, one linearizes the model at only one operating point and use it all the time. However, outside of this region, the approximated model won't work as desired. To overcome this problem, the nonlinear model is linearized at each operating point and internal plant in controller is updated with newly computed linear model. Number of states, prediction horizon and constraints, on the other hand, remain unchanged. In same sense, LPV-MPC computes new internal plant model based on the parameter value at each step, instead of linearizing the model. And, controller uses this new plant as to predict best control input sequence. Also, it is important to note that the new plant model stays same over prediction horizon. But, if one can estimate how the internal parameter vary over time, the controller uses the model that changes over the prediction horizon.

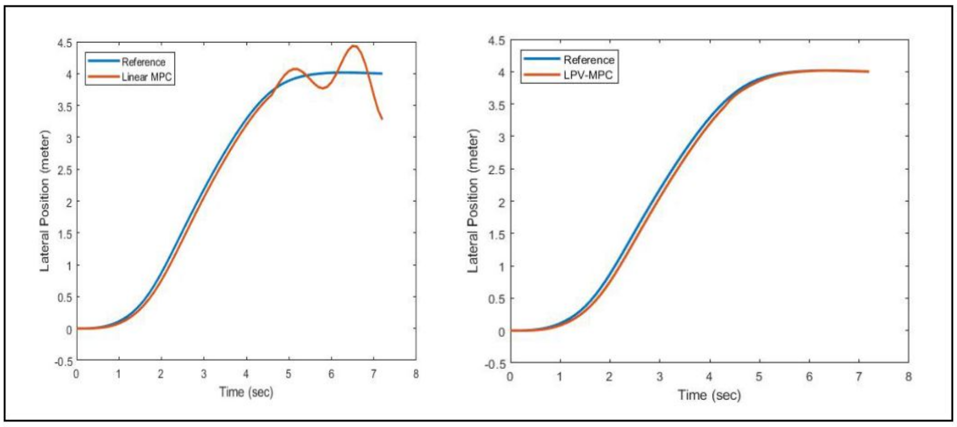


Figure 5.2: Linear and LPV MPC Comparison for varying longitudinal speed

On the other hand, apart from steering input. δ, the augmented system includes another input which is road curvature, ρ. However, it is not obtained through output feedback, so it is assumed as measured disturbance coming into the system. Consequently, the controller form designed in Section 4.3 is slightly changed in order to compensate new dynamics in augmented model and the new structure of MPC problem for the model in Eq. 5.12 can be designed as

$$\underset{\triangle\mathbf{u}}{\text{minimize}} \quad \frac{\mathbf{1}}{\mathbf{2}}\triangle\mathbf{u}^{\mathbf{T}}\mathbf{H}(\mathbf{p}(\mathbf{k}))\triangle\mathbf{u} + \begin{bmatrix} \tilde{\mathbf{x}}_{\mathbf{0}} \\ \mathbf{r} \\ \mathbf{md} \end{bmatrix}^{\mathbf{T}} \mathbf{F}(\mathbf{p}(\mathbf{k}))^{\mathbf{T}}\triangle\mathbf{u} \tag{5.13}$$

$$\text{subject to} \quad \mathbf{G}(\mathbf{p}(\mathbf{k}))\triangle\mathbf{u} \leq \mathbf{W}(\mathbf{p}(\mathbf{k})) + \mathbf{S}(\mathbf{p}(\mathbf{k}))\begin{bmatrix} \tilde{\mathbf{x}}_{\mathbf{0}} \\ \mathbf{md} \end{bmatrix} \tag{5.14}$$

where $p(k)$ is the parameter varying and md is the measured disturbances over prediction horizon.

As a final comment, in this thesis, it is assumed that the longitudinal velocity would be constant. In spite of this, LPV-MPC method is designed and employed during the experiments in order to be used in further cases at which the speed is not constant.

Chapter 6
Simulations

This chapter presents the experiments and validation of proposed methods. All experiments are designed using DSDA described in Chapter 2. Section 6.1 shows the experiments regarding Centerline Estimation. Validation of estimation of deviation from centerline can be found in Section 6.2. Section 6.3 covers the experiments regarding lateral control for centerline tracking. Finally, the overall review of results is discussed in Section 6.4.

6.1 Centerline Estimation

6.1.1 Centerline Driving

In this simulation, the algorithm proposed in Section 2.3 was tested. Vehicle path was set such that it would perfectly travel along track centerline. That's why, it is expected that developed algorithm would confirm centerline driving in each step.

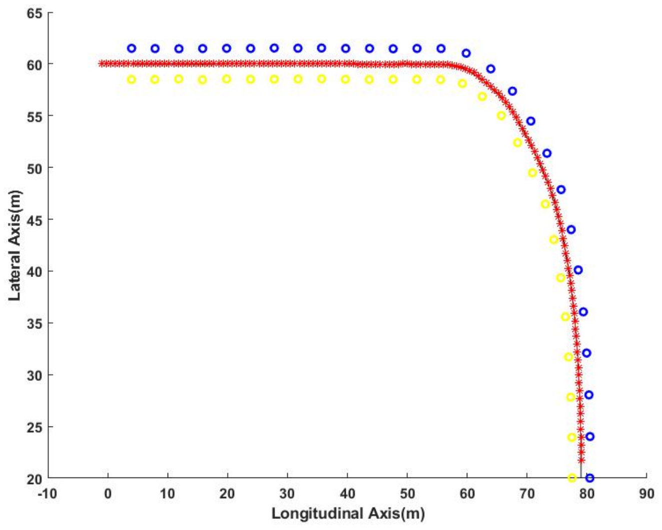


Figure 6.1: Result of Centerline Driving

Fig. 6.1 show the results of centerline driving simulation, where red stars indicates that the vehicle is in centerline, and black line is exact centerline of the track. From figure, it can be seen that the algorithm is sufficiently working as desired.

6.1.2 Free Driving

After centerline driving, the next simulation experiment was designed in which vehicle performs free driving, but it crosses the centerline a couple of times. Therefore, the algorithm should detect when crossing happens.

Below figure presents the results obtained after free driving. Blue stars means that the vehicle is not moving on track centerline. It clearly indicates that the proposed algorithm is prefctly able to detect when vehicle cross centerline.

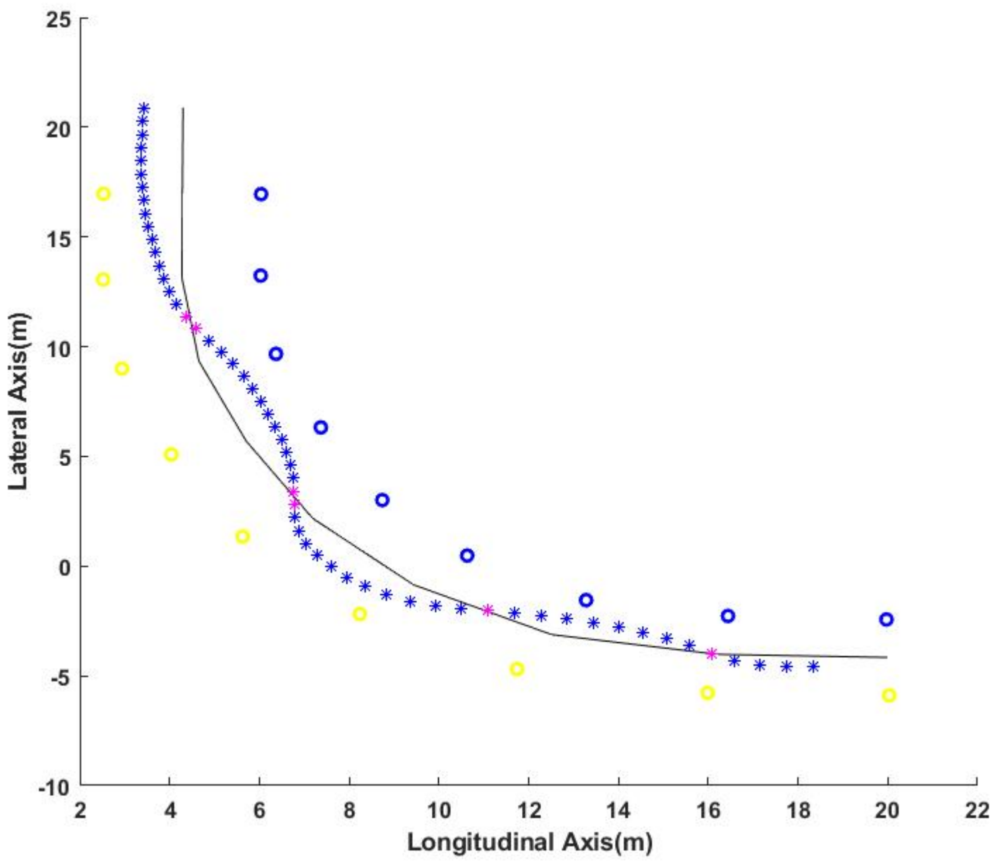


Figure 6.2: Result of Free Driving

6.2 Estimation of Deviation from Centerline

6.2.1 Free Driving

This simulation was performed to validate the method described in 2.4 in order to estimate the deviation from track centerline and heading error. The prepared test case is shown in Fig. 6.3.

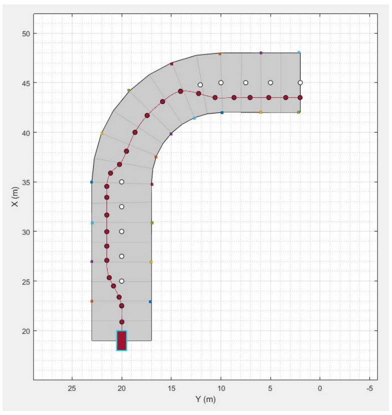


Figure 6.3: Test Case for Estimation of Deviation from Centerline

It was design such that the vehicle is first deviates 1.5 meters in positive direction and then, go back to centerline. Finally, it finishes its motion by deviating 1.5 meters in negative direction. Also, as can be seen from the plot, the road consists of three segments, straight-curve-straight. The curved segment was created in way that its centerline is the quarter of circle with radius 10 meter. This means that the algorithm should find the curvature 0 for straight segments and 0.1 through curved segment.

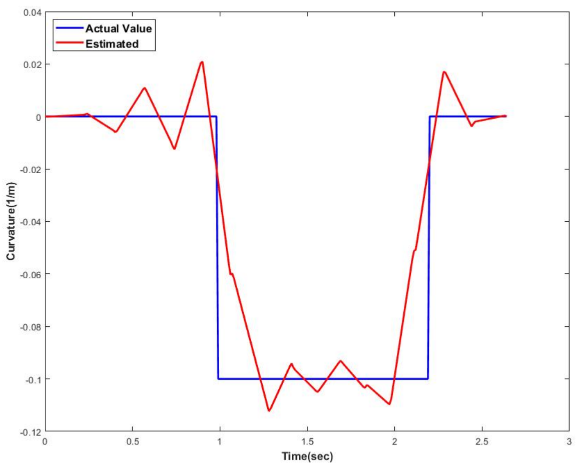


Figure 6.4: Centerline Curvature

Fig. 6.4 refers to curvature of centerline that the algorithm found. It indicates that the values are more or less same with the reference. On the other hand, below figures represents the lateral deviation and heading error. Fig.6.5 shows that the proposed method accurately estimates the deviation from track centerline. As result, it can be said that the algorithm is capable of estimating deviation from centerline correctly.

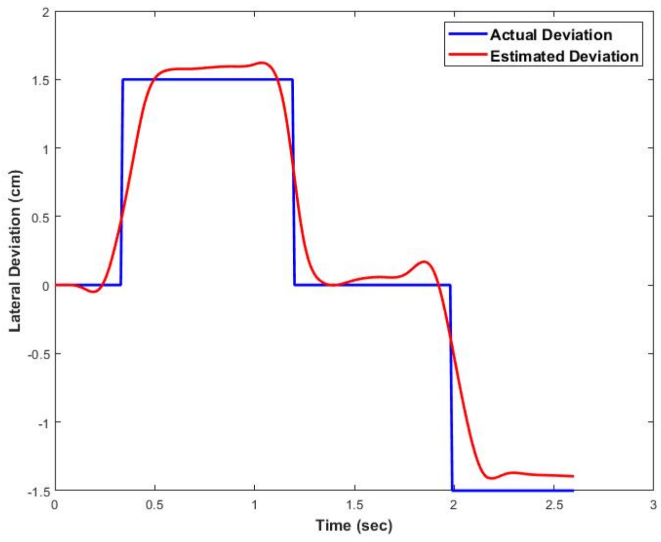


Figure 6.5: Lateral Deviation in Wild Driving

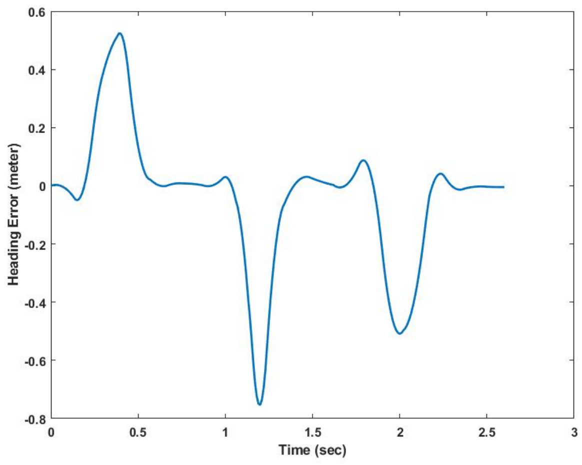


Figure 6.6: Heading Error in Wild Driving

6.2.2 Centerline Driving

After estimating the cross-track error in free driving scenario, this simulation was created to estimate same values when vehicle is moving on track centerline. The track was as in the previous simulation, just the vehicle path shown in Fig. 6.3 was changed to centerline path. That's why, it is expected that there should be zero lateral deviation and heading angle. The result regarding the lateral deviation can be found in below figure. It clearly expresses that the

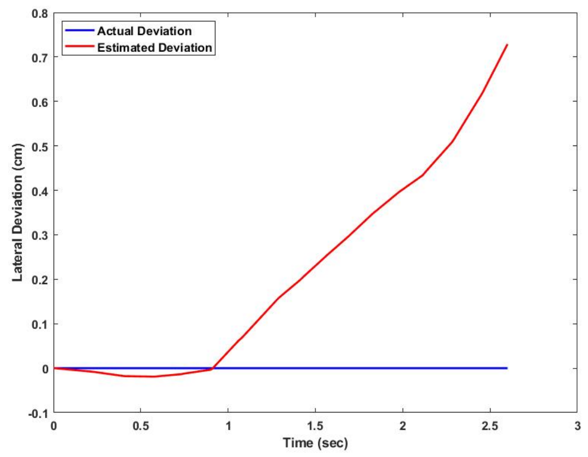


Figure 6.7: Lateral Deviation in Centerline Driving

the algorithm returned the values very close to zero as it is supposed to be. Also, the heading errors presented in Fig. 6.8 are small enough. Therefore, it can be pointed out that the proposed method for estimating the deviation from track centerline works efficiently as in the previous simulation.

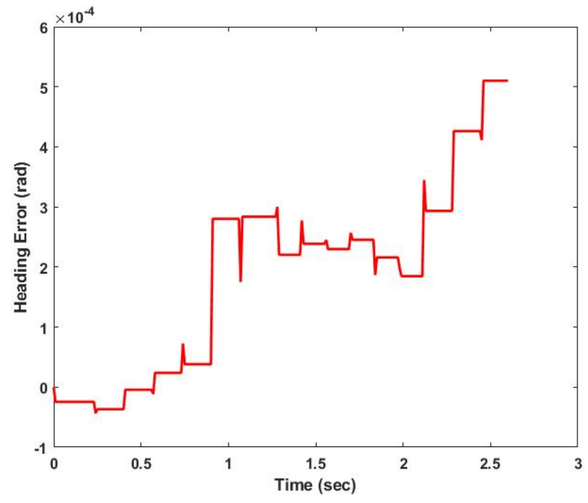


Figure 6.8: Heading Error in Centerline Driving

6.3 Centerline Tracking

After validating that the cross-track error can be estimated accurately using proposed method, this section covers the simulation experiment for lateral controller. As described in Chapter 6.8, lateral control problem is formulated as Lane Keeping Assist System such that it utilize the lateral deviation and heading error as internal plant model in the LPV-MPC controller. By doing so, the controller aims to minimize those error in order to track reference path as accurate as possible.

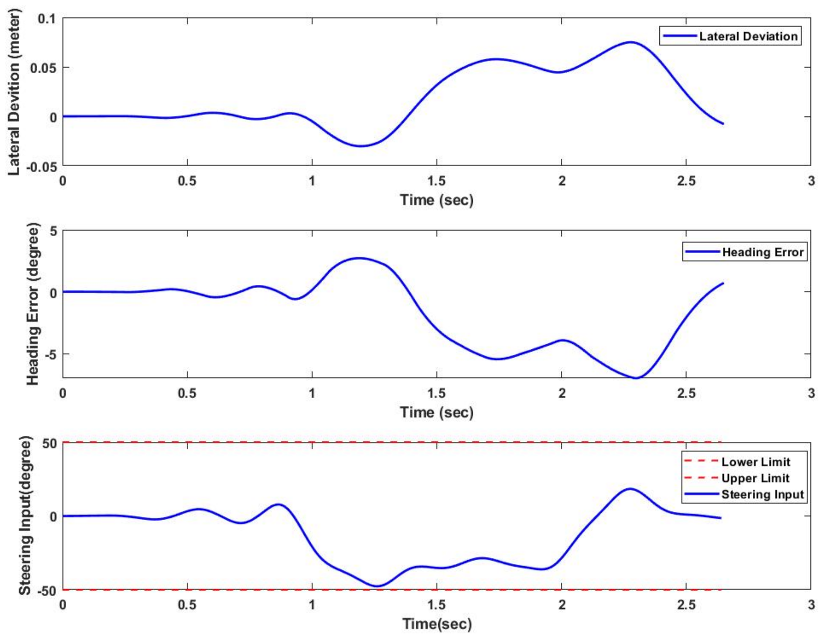


Figure 6.9: Results of Lateral Control - 1

For simulation, the scenario case used in Section 6.2 was performed. Fig. 6.9 and Fig. 6.10 represents the results obtained after simulation. Fig. 6.9 indicates that errors for both lateral deviation and heading error are quite small and Steering Input to system stayed inside the determined constraints, which means that the lateral controller was able to make vehicle to follow the desired path. It is also confirmed by Fig. 6.10 which shows that vehicle perfectly traveled on reference line.

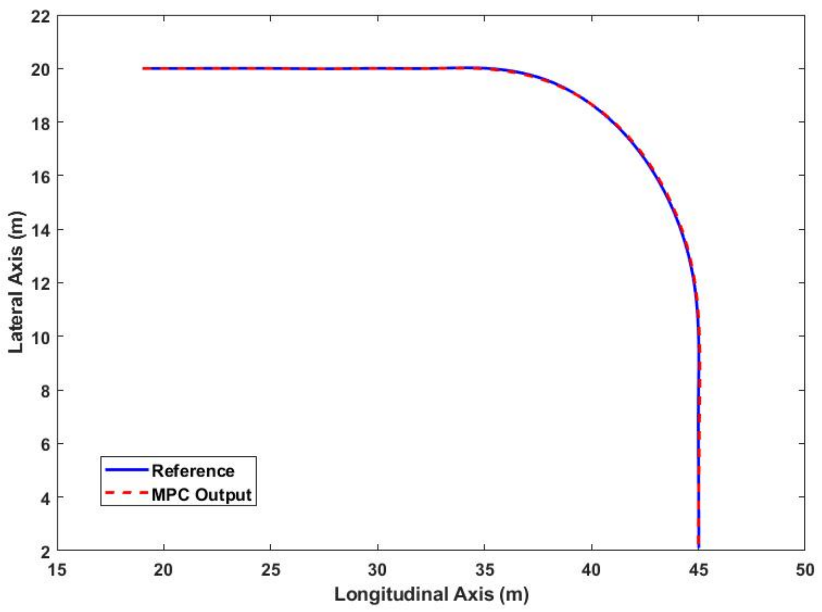


Figure 6.10: Results of Lateral Control - 2

6.4 Results

This section discuss overall conclusion of performed simulations in previous three sections. Fig. 6.1 and Fig. 6.2 proves that the method for detecting if vehicle is on track centerline, described in Algorithm 1, sufficiently works when vehicle performs free driving or moves on centerline of track. But, it is important to note that the accuracy of the algorithms heavily depends on the position of the cones, so it is essential to provide the accurate input as much as possible in order to get sufficient results.

Simulation results in Section 2.4 validates the method to estimate how much vehicle deviates from track centerline. From the corresponding figures, it can be confirmed that the proposed method in 2, which is extended version of 1, returns results as expected. The crucial part of the algorithm is to estimating of road curvature at each time instant. The algorithm is quite sensitive to curvature input, so accuracy in estimating the curvature is directly related to algorithm output.

On the other hand, the results in Section 6.3 indicates that lateral control design given in Chapter 5 meets the expectations and works as desired. Since it is formulated as lane keeping assist system, it needs lateral deviation and heading error in each time instant in order to calculate input to the system accurately.

Chapter 7

Conclusion And Future Works

In this thesis, the two main goals have been explored. The ultimate goals were to develop algorithm to detect if vehicle deviates from centerline of track and estimate the amount of deviation based on stereo-optic camera. After that, the secondary goal is to design model predictive control for lateral vehicle dynamics to make the vehicle to follow track centerline with as close as possible. All the two goals were fulfilled and the accomplished tasks uncovered possible future research directions for further improvements.

7.1 Accomplished Tasks

In chapter 2, simulation framework, Driving Scenario Designer Apps in MATLAB, was introduced first. It was necessary to create appropriate case studies according to FSD competition rules. Then, a novel algorithm was proposed to detect whether or not vehicle moves on track centerline. Then, the algorithm is extended and enhanced such that it would estimate the amount of deviation from centerline.

Chapter 3 presented vehicle modelling to be used in model predictive control. In that sense, kinematic vehicle model was given first. After that, dynamical modelling of vehicle with tire forces was developed. since it is not useful and valid at higher speed, dynamical modelling of vehicle was employed in this work.

Chapter 4 covered the comprehensive information and formulation of model predictive control. In the first part, the basic working principles and properties of the controller were presented. It was then followed by the detailed formulation. Since the thesis only takes into account of tracking, only formulation regarding tracking was discussed.

In Chapter 5, the lateral control design was introduced. It was heavily based on given formulation in Chapter 4. The difference was that it was formulated as Lane Keeping Assist System. In this regard, the dynamical model derived in Chapter 3 was linearized and augmented with lateral deviation and heading error estimated in 2. The augmented model was employed as internal plant model in the controller. After that, linear-time varying MPC structure was described in order to handle the conditions at which the longitudinal speed is not constant.

Chapter 6 presented the simulation experiments in order to validate proposed methods. The chapter was divided into three sections and each method was tested and validated separately. The simulations were created using DSDA explained in corresponding Section of Chapter 2. The results obtained after simulation experiments showed that all proposed method worked sufficiently as desired.

7.2 Suggested Future Works

This thesis takes into account of only track centerline. However, major aim in the FSD competition is to finish the described amount of lap in minimum time possible. In that sense, the one of future works can be finding optimal path for track using minimum curvature method. Then, optimal velocity profile is found using backward-forward pass to have complete trajectory. That would also result in the need of longitudinal control design, but its design is relatively simple compared to lateral control design.

State Estimation using Kalman Filter can be thought as another future work. In this thesis, there were no process or measurement noise in the performed simulations. However, the noise is inevitable phenomenon in real-time applications. That is why, the noise has to be add to the systems and feedback coming from the sensors has to be regulated in Kalman Filter to get noise-free feedback as much as possible.

On the other hand, the thesis includes only theoretical works. However, the competition would be real-time challenge, so it is necessary to ensure that the proposed algorithms sufficiently work in real-time scenarios. In that sense, it would be highly beneficial to implement the algorithms on-board to see if they are suitable for real racing conditions. This would realize the theoretical work made in this thesis in reality.

Bibliography

[1] "Tesla official website. https://www.tesla.com/."

[2] WEVOLVER, "2020 autonomous vehicle technology report."

[3] "Roborace details the tech behind its wild autonomous racecar. https://www.theverge.com/2016/8/4/12377262/roborace-car-images-technology-self-driving."

[4] "F1tenth, official website. https://f1tenth.org/."

[5] "eforce prague formula. https://eforce.cvut.cz/en/eforcedriverless."

[6] "Amz racing, official website. http://driverless.amzracing.ch/."

[7] "Fsg competition handbook 2019. https://www.formulastudent.de."

[8] A. Bemporad, "Linear mpc," 04 2019.

[9] "Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries."

[10] "Automated vehicles can reduce co2 emissions by 60 percent and two regional startups are taking note. https://www.wamda.com/memakersge/2018/02/automated-vehicles-reduce-co2-emissions-60-startups-taking-note."

[11] A. F. for Traffic Safety, "American driving survey," 2015-2016.

[12] "Ten ways autonomous driving could redefine the automotive world. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world."

[13] "Formula e announce driverless support series. https://www.fiaformulae.com/en/news/2015/november/formula-e-kinetik-announce-roborace-a-global-driverless-championship.aspx."

[14] "Roborace, official website. https://roborace.com,"

[15] “Racing algorithms. https://www.audi.com/en/experience-audi/mobility-and-trends/autonomous-driving/roborace.html.”

[16] “Formula student driverless. https://www.formulastudent.de/pr/news/details/article/autonomous-driving-at-formula-student-germany-2017/.”

[17] J. Kabzan, M. Valls, V. Reijgwart, H. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan, A. Dhall, E. Chisari, N. Karnchanachari, S. Brits, M. Dangel, I. Sa, R. Dube, A. Gawel, M. Pfeiffer, and R. Siegwart, “Amz driverless: The full autonomous racing system,” 05 2019.

[18] A. Liniger, A. Domahidi, and M. Morari, “Optimization-based autonomous racing of 1:43 scale rc cars,” *Optimal Control Applications and Methods*, vol. 36, p. 628–647, Jul 2014.

[19] A. Liniger, “Path planning and control for autonomous racing,” 2018.

[20] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, “Race driver model,” *Computers and Structures*, vol. 86, pp. 1503–1516, 07 2008.

[21] D. Caporale, A. Fagiolini, L. Pallottino, A. Settimi, A. Biondo, F. Amerotti, F. Massa, S. De Caro, A. Corti, and L. Venturini, “A planning and control system for self-driving racing vehicles,” in *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, pp. 1–6, 2018.

[22] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, “Minimum curvature trajectory planning and control for an autonomous race car,” *Vehicle System Dynamics*, pp. 1–31, 06 2019.

[23] N. R. Kapania, *Trajectory planning and control for an autonomous race vehicle.* PhD thesis, Stanford University, 2016.

[24] M. Mühlmeier and N. Müller, “Optimization of the driving line on a race track,” in *SAE Technical Paper*, SAE International, 12 2002.

[25] MathWorks, “Build a driving scenario and generate synthetic detections.”

[26] O. Törő, T. Bécsi, and S. Aradi, “Design of lane keeping algorithm of autonomous vehicle,” *Periodica Polytechnica Transportation Engineering*, vol. 44, pp. 60–68, 01 2016.

[27] B. Gutjahr, L. Gröll, and M. Werling, “Lateral vehicle trajectory optimization using constrained linear time-varying mpc,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2017.

[28] “Arc length and curvature. https://opentextbc.ca.”

[29] "Radius of curvature. https://www.intmath.com/applications-differentiation/8-radius-curvature.php."

[30] "Khan academy, cruvature formula. https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/curvature/v/curvature-formula-part-1."

[31] "Equation of a circle passing through 3 given points. https://planetcalc.com/8116/."

[32] "computing curvature equation. https://www.mathworks.com/matlabcentral/answers/284245-matlab-code-for-computing-curvature-equation."

[33] J. Filip, *Trajectory Tracking for Autonomous Vehicles.* PhD thesis, 05 2018.

[34] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," pp. 1094–1099, 2015.

[35] F. Curinga, "Autonomous racing using model predictive control," 01 2018.

[36] M. Mondek, "Active torque vectoring systems for electric drive vehicles," 01 2018.

[37] R. Verschueren, S. De Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, "Towards time-optimal race car driving using nonlinear mpc in real-time," pp. 2505–2510, 2014.

[38] D. Efremov, T. Haniš, and M. Hromčík, "Introduction of driving envelope and full-time-full-authority control for vehicle stabilization systems," pp. 173–178, 2019.

[39] D. Efremov, "Single-track model derivation."

[40] H. B. Pacejka, "Chapter 2 - basic tire modeling considerations," in *Tire and Vehicle Dynamics (Third Edition)* (H. B. Pacejka, ed.), pp. 59 – 85, Oxford: Butterworth-Heinemann, third edition ed., 2012.

[41] H. B. Pacejka, "Chapter 4 - semi-empirical tire models," in *Tire and Vehicle Dynamics (Third Edition)* (H. B. Pacejka, ed.), pp. 149 – 209, Oxford: Butterworth-Heinemann, third edition ed., 2012.

[42] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle System Dynamics*, vol. 21, no. sup001, pp. 1–18, 1992.

[43] S. V. Raković and W. Levine, *Handbook of Model Predictive Control.* 09 2018.

[44] F. Borrelli, A. Bemporad, and M. Morari, "Predictive control for linear and hybrid systems," 2017.

[45] M. Ulusoy, "Understanding model predictive control. https://www.mathworks.com/videos/series/understanding-model-predictive-control.html."

[46] Z. Hurak, "Discrete-time optimal control— direct approach," 03 2019.

[47] A. Bemporad, "Linear time-varying and nonlinear mpc," 04 2019.

[48] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," pp. 378–383, 2013.

[49] Y. Xu, B. Y. Chen, X. Shan, W. H. Jia, Z. F. Lu, and G. Xu, "Model predictive control for lane keeping system in autonomous vehicle," pp. 1–5, 2017.

[50] Z. Wang, Y. Bai, J. Wang, and X. Wang, "Vehicle path tracking ltv-mpc controller parameter selection considering cpu computational load," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, 12 2018.

[51] P. F. Lima, J. Mårtensson, and B. Wahlberg, "Stability conditions for linear time-varying model predictive control in autonomous driving," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2775–2782, 2017.

[52] M. Ulusoy, "Understanding model predictive control, part 4: Adaptive, gain-scheduled, and nonlinear mpc. https://www.mathworks.com/videos/understanding-model-predictive-control-part-4-adaptive-gain-scheduled-and-nonlinear-mpc-1530606851674.html."