Czech Technical University in Prague

Faculty of Electrical Engineering

MASTER THESIS

Bc. Bohdan Kashel

Protection of Driving Corridor with Respect of Various Road Conditions

Department of Control Engineering

Supervisor of the master thesis: Ing. Denis Efremov
Supervisor of the master thesis: Doc. Ing. Tomaš Haniš, Ph.D.

Study program: Cybernetics and Robotics

Prague 2022

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kashel Bohdan** Personal ID number: **452850**

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:

Protection of driving corridor with respect of various road conditions

Master’s thesis title in Czech:

Ochrana jízdního koridoru s ohledem na různé jízdní vlastnosti vozovky

Guidelines:

The thesis aims to define and implement a model predictive control for environmental envelope protection. The environmental envelope will be described as a driving corridor for a vehicle. It will include a track and a path, which respects various surface driving characteristics (reduced friction property of the drivable surface).
1) Familiarize yourself with vehicle dynamics simulator IPG CarMaker, optimizing solver Gurobi, and YALMIP optimizing framework for Matlab.
2) Study existing solutions for lane-keeping and obstacle avoidance.
3) Augment existing single-track vehicle model to be suited for position control in the predictive control framework.
4) Suggest a suitable optimization problem that will follow the driver's commands when the vehicle is in the predefined lane and protect its boundaries violation.
5) Augment the controller from the previous point to avoid fixed drivable obstacles, such as ice pads on the road inside the predefined lane.
6) Provide verification using IPG CarMaker and Matlab/Simulink testing environment.

Bibliography / sources:

[1] Schramm, D., Hiller, M., & Bardini, R. (2014). Vehicle dynamics. Modeling and Simulation. Berlin, Heidelberg,151.
[2] Erlien, S. M. (2015). Shared vehicle control using safe driving envelopes for obstacle avoidance and stability. Stanford University.
[3] Turri, V., Carvalho, A., Tseng, H. E., Johansson, K. H., & Borrelli, F. (2013, October). Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In 16th international IEEE conference on intelligent transportation systems (ITSC 2013) (pp. 378-383). IEEE.
[4] Chen, B. C., Luan, B. C., & Lee, K. (2014, August). Design of lane keeping system using adaptive model predictive control. In 2014 IEEE International Conference on Automation Science and Engineering (CASE) (pp. 922-926). IEEE.

Name and workplace of master’s thesis supervisor:

Ing. Denis Efremov Department of Control Engineering FEE

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

Date of master’s thesis assignment: **28.01.2022** Deadline for master's thesis submission: **15.08.2022**

Assignment valid until:
by the end of summer semester 2022/2023

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

Acknowledgements

I would like to say a special thank you to my supervisor, Ing. Denis Efremov. His support, guidance, and overall insights in this field have made this an inspiring experience for me. And my biggest thanks to my family for all the support you have shown me through this research, the culmination of my learning. I am also grateful to the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague. I am incredibly thankful and indebted for sharing expertise, and sincere and valuable guidance and encouragement extended to me.

I hereby declare that I have completed this thesis with the topic "Protection of driving corridor with respect of various road conditions" independently and that I have included a full list of used references. I have no objection to the usage of this work in compliance with the act §60 Zákon č.121/2000 Sb. (copyright law).

In ..... date ..... signature of the author

Abstrakt: Primárním cílem této diplomové práce je vytvořit pokročilý asistenční systém řidiče (ADAS) pro ochranu obálky prostředí, který pomůže řidiči udržet auto na silnici a vyhnout se pojízdným překážkám. Prezentovaný ADAS využívá potenciál moderních senzorů instalovaných dnes v autech a zvýšuje bezpečnost vozidel. Řídicí systém používá data s kamer, GPS a senzorů rychlosti a na rozdíl od většiny systémů, které jsou uvedeny v jiných publikacích diskutovaných v teto práci, nepoužívá předem definovanou cestu nebo jízdní koridor. Diky své schopnosti analyzovat data v reálném čase ADAS umožňuje pohodlnou bezpečnou jízdu prostřednictvím udržovaní řidiče na silnici s ohledem na předpověď trajektorie vozidla a na různé provozní vlastnosti povrchu. K zajištění stability vozidla a bezpečnosti jízdy systém napomaha vyhnout se pojízdným překážkám nebo nerovnostem vozovky. Funkčnost prezentovaného ADAS je ověřena v softwaru IPG CarMaker pro simulaci dynamiky vozidla na několika automatických jízdních testech.

Klíčová slova: Asistenční Systém Řidiče, Automatizace Jízdy, Hlídání Jízdních Pruhů, Model-Prediktivní Řízení, Vyhýbání se Překážkám

Abstract: The primary goal of this master thesis is to create an advanced driver assistance system (ADAS) for environmental envelope protection to help the driver keep a car on the road and avoid drivable obstacles. Presented ADAS realizes the potential of modern sensors installed in vehicles today and works towards vehicle safety. The control system cooperates with cameras, GPS, and velocities sensors and, unlike most of those presented in other papers discussed in work, does not use a predefined path or driving corridor. It analyzes real-time data, allowing comfortable driving with increasing safety by keeping the driver on the road with respect to the predicted from the steering wheel vehicle's path and various surface operating characteristics. The controller tries to avoid drivable obstacles, such as ice pads or road irregularities, by each wheel to succeed in vehicle stability protection and safety. The functionality of the presented ADAS is validated in IPG CarMaker vehicle dynamics simulation software on several automated ride tests.

Keywords: Advanced Driver Assistance System, Driving Automation, Lane-Keeping, Model Predictive Control, Obstacle Avoidance

Contents

1. Introduction

The general goal of this thesis is to create an advanced driver assistance system (ADAS) for environmental envelope protection. This work defines the environmental envelope as road borders with excluded drivable obstacles.

ADAS is a technology that assists drivers in different ways while driving. Famous examples of existing ADASes are anti-lock braking systems, cruise control, adaptive cruise control, alcohol ignition interlock devices, and automatic parking. The purpose of the proposed control system is to keep a vehicle on the road with respect to the path dictated by the driver through the steering wheel and various surface driving characteristics. In other words, the ADAS will help a driver stay on the driven road and avoid drivable obstacles (such as potholes, ice pads, or other road irregularities).

A modern vehicle is more than just seats with a motor and wheels. It is a sophisticated, computerized, mobile sensor cluster with high computational capacity onboard. The proposed ADAS consumes processed images from cameras and measurements from the GPS and velocities sensors to help the driver steer safer. The presented control structure obtains the road boundaries online and precomputes trajectory according to the steering angle commanded by the driver. Unlike today's lane-keeping systems, it does not use a predefined path or driving corridor, but it is a camera-ready control system. It relentlessly checks if each wheel will be driven on the "safe" road.

The chosen approach lets a vehicle be crafted as a sophisticated being that understands its position on the road and the place of each wheel. The designed control architecture prevents, for example, such unpleasant travel encounters as pothole hitting. Such events can cause a wheel deformation with dangerous consequences, and their avoidance will increase safety and driver's and passenger's comfort during a long journey, relieving driver's attention.

Presented driving improvement is inspired by horse riding. During riding, the rider chooses a direction only and does not bother with keeping attention to the road quality and its borders. A horse manages foot placement to avoid dangerous holes, tree roots, or other road irregularities. The proposed ADAS is targetting to do the same functionality in a car.

1.1 Outline

This work is divided into eight parts.

In the first two parts, which are [**Introduction**] and [**Objectives**], work description and goals are stated.

The part [**State of the Art**] are reviewed modern techniques and standard nowadays approach in the field of Autonomous driving.

Chapter [**Single-Track Models**] introduces of used dynamics models and modification made to better suit our purpose.

In part, which is named [**Control Architecture**], are presented designed control architecture and optimization problem formulation.

Chapter [**Experiments**] describes test configurations and shows the simulation results.

The next part, [**Results**], lists reached goals.

The last part, [**Conclusion**] summarises this thesis and proposes possible future works in the field of laterally unstable ground vehicles.

2. Objectives

The primary objectives of this thesis are:

- Familiarizing with vehicle dynamics simulator IPG CarMaker, optimizing solver Gurobi, and YALMIP optimizing framework for Matlab.
- Study existing solutions for lane-keeping and obstacle avoidance.
- Augment existing single-track vehicle model to be suited for position control in the predictive control framework.
- Suggest a suitable optimization problem that will follow the driver's commands when the vehicle is in the predefined lane and protect its boundaries violation.
- Augment the controller from the previous point to avoid fixed drivable obstacles, such as ice pads on the road inside the predefined lane.
- Provide verification using IPG CarMaker and Matlab/Simulink testing environment.

3. State of the Art

3.1 Driving Automation

A car is not just a wheeled motor used for transportation in the modern world. Today it is a big, computerized sensor cluster that can be used for driver's assistance and workload reduction. Increasing computational power and numbers and the complexity of sensors also expand the capabilities that engineers can realize. Such features are autonomous driving, automated maneuvering in various driving scenarios, and emergency assistance.

According to the Society of Automotive Engineers (SAE) [1] and the National Highway Traffic Safety Administration (NHTSA) [2], [3], there are defined in total six levels of driving automation, from zero to five, listed below. All levels of automation are schematically depicted in Fig.3.1

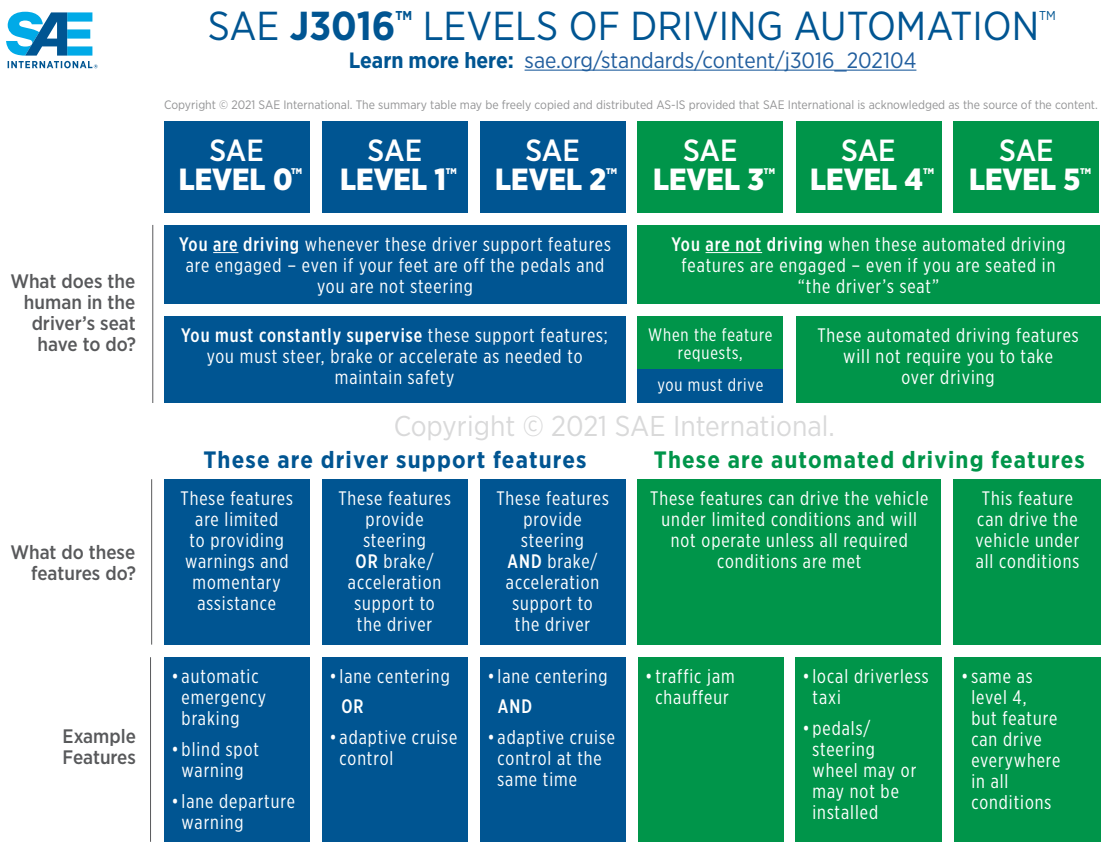


Figure 3.1: Levels of driving automation according to SAE [4].

Level 0: No-Automation

A driver has complete control over the vehicle's inputs (steering, brake, and throttle) and is solely responsible for monitoring road situations, reaction to hazards, and safe vehicle operation. If a vehicle contains any driver's assistance, it is represented by warning signals sent to the driver (such as collision warning, lane departing warning,

etc.)

Level 1: Function-Specific Automation

This level presents partial automation of primary functions. A vehicle could have more assistance systems such as an anti-locking braking system (ABS) or electronic stability program (ESP). However, if a vehicle has more than one assistance system, these systems must be interconnected (to fit the Level 1 definition.) A driver has complete control over the vehicle's inputs (steering, brake, and throttle) but can partially delegate authority over input to an assistance system. The driver is still solely responsible for monitoring road situations, reaction to hazards, and safe vehicle operation. An example of such a system are line centering *or* adaptive cruise control (ACC).

Level 2: Combined Function Automation

Level 2 is similar to Level 1, but multiple assistance systems are interconnected with the operating inputs. For example, the anti-locking braking system and electronic stability program operate with braking. However, in Level 2, the control engineers solve a hierarchy/priority/logic when both assistance systems are switched on. A driver can pass a control authority over a vehicle's input to the assistance systems in limited situations. The driver is still solely responsible for monitoring road situations, reaction to hazards, safe vehicle operation, and taking control over the vehicle on short notice. For example, line centering *with* adaptive cruise control simultaneously is a Level 2 assistance system.

Level 3: Conditional Automation

This level is characterized by heavily relying on automation under defined circumstances. The control is fully transferred to assistance systems in specific conditions. However, it is expected that a driver will take control of the vehicle if some predefined conditions would not met. An example is the latest models of Audi A8 equipped with a Traffic Jam Pilot [5]. According to the manufacturer, it is a Level 3 automation system that can work at speeds up to $60\,\mathrm{km\,h^{-1}}$. This system should be capable of vehicle navigation in traffic jams by taking the start, accelerate, and stop actions and keeping a vehicle in a lane.

Level 4: High Automation

The automation system entirely operates the vehicle in limited-service areas and does not need a driver. The Level 4 automation system example could be found outside of the automotive sector. For instance, Amazon's automatized warehouses are excellent examples of Level 4 of automation. In this case, a warehouse is a limited zone where robots can navigate and complete their tasks fully automatedly. In automotive, such an example is a driverless taxi in Singapore, presented by NuTonomy [6]. This taxi was adapted to work in Singapore's small, campus-like business district.

Level 5: Full Automation

A vehicle is entirely operated by an automation system anywhere and in any environmental conditions. This level of automation is typical for a vehicle from sci-fi works that does not have a steering wheel or other controllers. It entirely operates automatically, and a human is just a passenger.

3.2 Related Advanced Driver Assistance Systems

Advanced Driver Assistance Systems are today a common feature of the modern vehicle. They are an essential part of a modern car and the key element on which autonomous vehicles are based. Most current ADASes are vision-based [7] but achieved technological level permits using LIDAR and RADAR technologies in automotive.

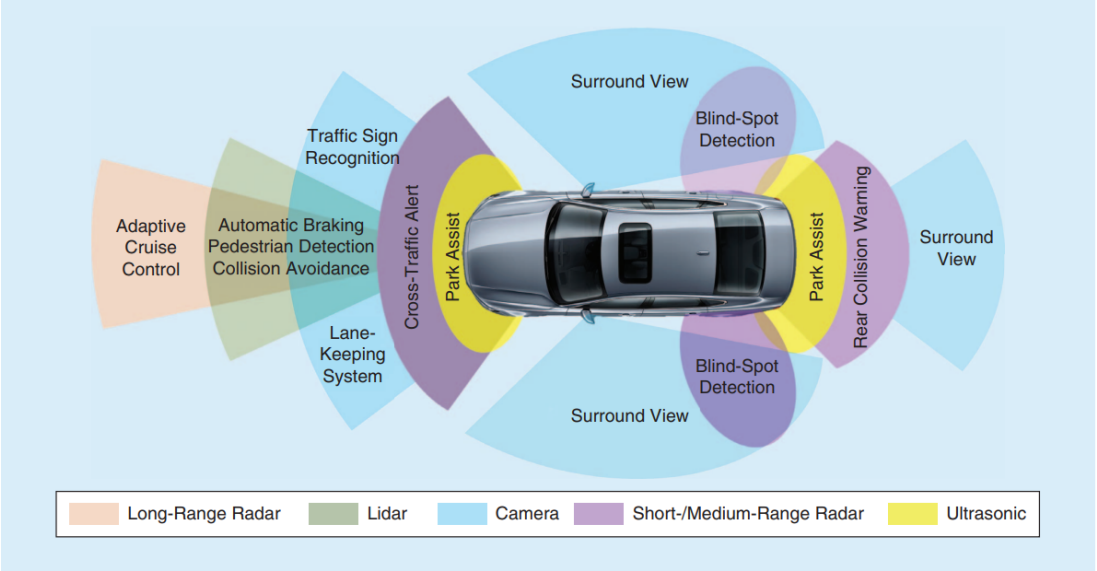


Figure 3.2: Common modern-day ADASes and sensors utilized by them [7].

Figure 3.2 shows popular ADASes and sensors utilized by them. Vision-based ADASes are very popular today among vehicle manufacturers because modern cameras are cheap, easy to install, capture a large amount of information (color, texture, contrast), and onboard computers have enough image-processing capacities. Cameras can be installed inside and outside of the vehicle. Inside cameras can be used to monitor a driver's status by face-detection, eyes-detection, and head-pose recognition. Cameras are a popular solution for such vehicle features as surround view, line-keeping, and traffic sign recognition. Cameras can be used as monocular or as stereo bases. Monocular is cheaper but lacks the ability for depth perception when stereo base can be used to receive three-dimensional information by matching two-dimensional images from several cameras. LIDAR is used to receive a three-dimensional overview of surroundings. Their cons are price, limited range, and exposure to external conditions. A typical application of LIDAR in ADASes is automatic braking, pedestrian detection, and collision avoidance. RADAR has the same purpose as LIDAR but can work on

longer distances and is unaffected by fog, rain, and other changes in environmental conditions. RADAR is a popular solution for adaptive cruise control, blind-spot detection, rear collision warning, and cross-traffic alert. Photonic mixer devices (PMD) are used for fast pixel-wise distance measurement with high resolution. Inertial measurement units (IMU) help improve the distance measurements with LIDAR and RADAR. This is just a part of the sensors used in ADAS today. More can be found in Figure 3.3.

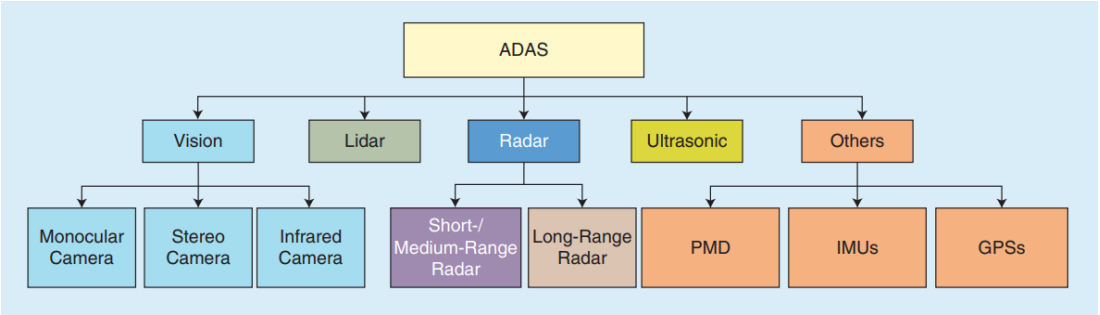


Figure 3.3: Schematic representation of the utilized sensors used in modern-day ADASes [7]. IMU stands for inertial measurement unit, PMD is photonic mixer devices, GPS is a global positioning system.

The following subsections discuss today's advanced driver assistance systems related to the proposed control strategy.

3.2.1 Lane-Keeping

Lane-keeping is a baseline technology for this work. Lane-keeping is an advanced driver assistance system that helps a driver stay on a chosen lane with steering modifying or overwhelming. It can be reactive (react on a crossing of road marking) or proactive (detect lanes borders and generate based-on input). Lane-keeping is a technology that consists of:

- road detection,
- position determination,
- heading determination,
- control input generation.

Road detection

Road detection is a vital part of lane-keeping. In this part, the world around a vehicle is being modeled. The primary function of this stage is to localize a road and determine boundaries where a vehicle can move.

Position determination

The position determination function determines the relative position between a vehicle and the environment.

Heading determination

This part of lane-keeping provides determination of a vehicle azimuth for a possibility to calculate and predict vehicle dynamics.

Control input generation

The final part of lane-keeping is generating of steering angle to keep a vehicle in the defined lane.

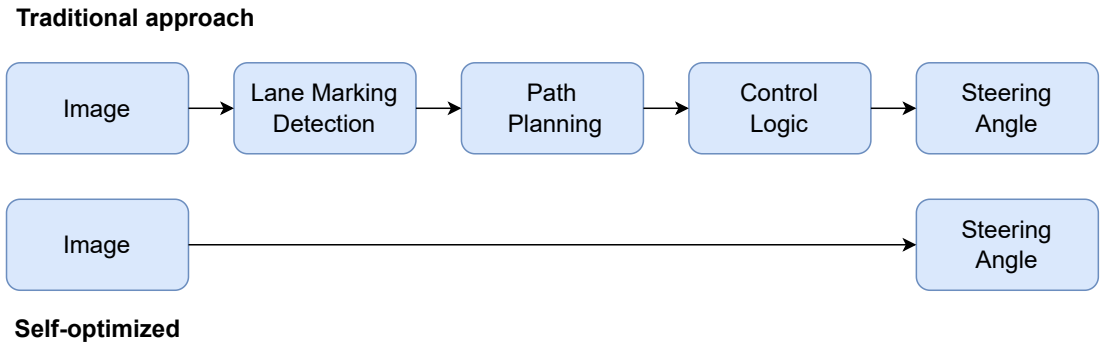


Figure 3.4: Schematic representation of lane-keeping workflow methods adopted from [8].

There are at least two approaches to lane-keeping (Fig. 3.4): traditional and end-to-end learning. The first one can be found in such works as [9] and [10]. The second one is usually represented by a neural network and is described in such articles as [8] and [11]. More about state of the art in lane-keeping can be found in [12].

3.2.2 Obstacle Avoidance

The second technology used in this work is obstacle avoidance. Typical obstacle avoidance performs lane changing if needed to avoid an obstacle. Obstacles are represented mainly by vehicles moving with constant velocity. This type of obstacle avoidance can be found in [13]. Important subsequent parts of this technology are:

- obstacle detection,
- position determination,
- heading determination,
- control input generation.

Components of obstacle avoidance are very similar to parts in lane-keeping. This fact allows us to talk about a combination of lane-keeping and obstacle avoidance in one control algorithm.

3.2.3 Combination of Lane-Keeping and Obstacle Avoidance

This work focuses on a combination of lane-keeping and obstacle avoidance. This type of ADASes is mostly hierarchical and typically contains the following modules:

- path planning,
- trajectory generation,
- trajectory tracking,

Even though modern vehicles have many components for fully autonomous driving, such as perception, navigation, and control tools, they may be insufficient for a path planner because of lack or inaccuracy of information. This is especially true for changing road conditions or nonlinear traction limits of the vehicle. Even with access to all this data, safely encoding all necessary information to navigate the environment as a single path will be challenging. If the predefined path cannot be perfectly tracked, a path tracking controller should know where it is safe to deviate from the path, such as within a lane, and where it is not, such as next to another vehicle.

Trajectory planning approaches can be divided into model-based algorithms, geometry-based, and heuristic-based methods [14].

The model-based approach uses optimal control techniques such as Nonlinear Programming (NLP) or Model Predictive Control (MPC) scheme presented for the first time by Kelly and Nagy in [15] to evaluate generated trajectory. A model-based algorithm allows direct trajectory planning instead of path planning by finding the appropriate control input functions that drive a vehicle model to the desired endpoint.

Trajectory planning with geometric-based methods relies on some parametric geometrical curves to generate trajectories. The working principle of these algorithms is calculating the parameters of the curves considering start and end points, initial and final values of the curve's derivatives, the vehicle's limited steering angle, and the maximal allowed lateral acceleration of a mass point moving along the curve. Geometric-based methods can be suitable mainly for low-speed applications such as automated parking, but they become unfeasible as the velocity of the vehicle increases. Geometric-based methods are represented in works such as [16] and [17].

Heuristic-based approaches usually apply artificial intelligence techniques such as search-based methods, random sampling methods, and machine learning methods.

Most of the geometric-based and heuristic-based methods generate rather paths than trajectories. This situation results in an additional need to convert the computed path into a set of trajectories which usually happens with the assignment of a speed profile.

Trajectory tracking is presented by regulators who are tracking the optimal path. The main pros of the classical scheme are that the problem is linear and convex. The main cons lay in the fact that all paths must be known, which is hard to realize in the real world. The classical scheme was discovered in works such as [18], [19], [20], and partially in [21].

This work suggests a model-based algorithm because it will allow completing the thesis's assignment. This thesis aims to move away from standard methods because they have problems working in real-time under realistic conditions. Presented work tries to implement a control concept for a drive-by-wire vehicle, which tracks the driver's

commands as closely as possible while preventing leaving road borders and running over the surface with different friction coefficients or road irregularities. Task formulation excludes foreknowing of all paths and all potential obstacles and focuses on creating a solution by assuming camera-based real-time detection.

4. Single-Track Models

The proposed control architecture uses the model predictive control (MPC) technique. It requires a dynamic model to predict the system's behavior and vehicle's reaction to the control input. The linear single-track model [22] was chosen as a reference model because it allows describing the lateral physical behavior of a moving vehicle without unnecessary complexity.

4.1 Used Assumptions and Simplifications

The used linear single-track model describes planar vehicle motion. Its derivation is provided under the following simplifications:

- Vehicle motion includes only lateral displacement and changes in the vehicle's heading.
- Vehicle mass is concentrated at the center of gravity.
- Both wheels on each axle are lumped together. Each axle is modeled by one tire model with imaginary contact points between tires and the surface on the center of axles.
- Pneumatic trail and self-aligning torque resulting from a side-slip angle on each tire are neglected.
- Mass distribution on the axles is assumed to be time-variant and measured.
- Longitudinal forces on tires, resulting from a normalized tire slip angle, are neglected. All longitudinal forces acting on each axle are assumed to be strictly from the engine.
- The velocity of the vehicle is assumed to be constant.

4.2 Linear Single-Track

The original model is taken from [23]. The vehicle coordinate system used in this thesis is the conventional right-hand Cartesian coordinate system with an x-axis following from the center of gravity (CG) to the front of the vehicle. The y-axis goes towards the left side of the car. The z-axis lies from CG to the vehicle's top, as shown in Fig. 4.1. The vehicle's yaw has a positive angle increment while turning to the left. The single-track model (Fig. 4.2) has two degrees of freedom and represents the planar translation and yaw motion of a vehicle.

The described model consists of two states listed in Table 4.1 and one control input, the steering angle. All parameters used in the original model are presented in Table 4.2.

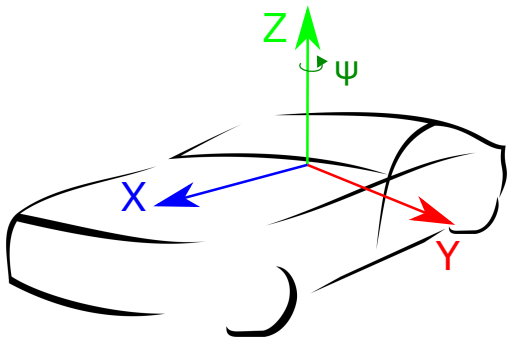


Figure 4.1: The vehicle coordinate system [24].

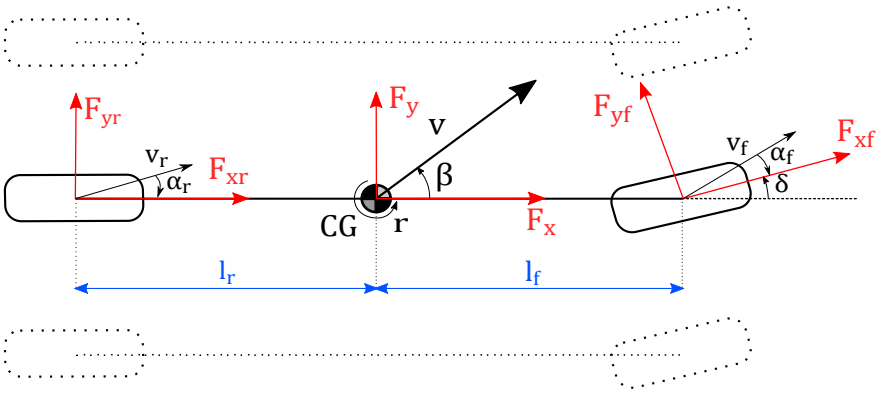


Figure 4.2: The single-track model [25].

Table 4.1: States and inputs of the original model

State/Input	Symbol	Units
Side-slip angle	β	rad
Yaw rate	r	$\mathrm{rad\,s^{-1}}$
Steering angle of the front axle	δ	rad

Table 4.2: Parameters of the single-track

Parameter	Symbol	Units
Distance from CG to axle	l	m
Yaw moment of inertia	I	$\mathrm{kg\,m^2}$
Nominal cornering stiffness	c	$\mathrm{rad^{-1}}$
Load force of wheels	F_z	N
Description	**Subindex**	
Lateral or longitudinal direction	x or y	
Front or rear axle	f or r	
Left or right side	l or r	

4.2.1 Mathematical Description of the Linear Single-Track Model

The velocity of the vehicle in the vehicle fixed coordinate system along axes (Fig. 4.2) can be expressed in the following way:

$$\vec{\mathbf{v}} = \begin{bmatrix} v\cos\beta \\ v\sin\beta \\ 0 \end{bmatrix}, \tag{4.1}$$

where v stands for velocity vector's length, e.g. $v = |\vec{\mathbf{v}}|$. From the assumption that velocity is constant, it follows that only normal acceleration affects planar motion of the vehicle ($\vec{\mathbf{a}}^T \cdot \vec{\mathbf{v}} = 0$):

$$\vec{\mathbf{a}} = \frac{d\vec{\mathbf{v}}}{dt} + \vec{\omega} \times \vec{\mathbf{v}} = \begin{bmatrix} -v\sin\beta\dot{\beta} \\ v\cos\beta\dot{\beta} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} \times \begin{bmatrix} v\cos\beta \\ v\sin\beta \\ 0 \end{bmatrix} = \begin{bmatrix} -v(r+\dot{\beta})\sin\beta \\ v(r+\dot{\beta})\cos\beta \\ 0 \end{bmatrix}, \tag{4.2}$$

where $\vec{\omega}$ is a vector of Euler's angles rate (roll rate, pitch rate, yaw rate). The lateral acceleration is derived as

$$a_{\mathrm{y}} = v(r+\dot{\beta})\cos\beta. \tag{4.3}$$

A vehicle operates with a slight deviation of the side-slip angle from zero. Assuming small side-slip angle β, the following approximation is valid: $\cos\beta \approx 1$. Therefore,

$$a_{\mathrm{y}} = v(r+\dot{\beta}). \tag{4.4}$$

The velocities of the tire contact point of front and rear axles can be calculated through the translation velocity of the CG and Euler's angles rate as:

$$\vec{\mathbf{v}}_{\mathrm{f}} = \vec{\mathbf{v}} + \vec{\omega} \times \begin{bmatrix} l_{\mathrm{f}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v\cos\beta \\ v\sin\beta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} \times \begin{bmatrix} l_{\mathrm{f}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v\cos\beta \\ v\sin\beta + l_{\mathrm{f}}r \\ 0 \end{bmatrix}, \tag{4.5}$$

$$\vec{\mathbf{v}}_{\mathrm{r}} = \vec{\mathbf{v}} + \vec{\omega} \times \begin{bmatrix} -l_{\mathrm{r}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v\cos\beta \\ v\sin\beta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} \times \begin{bmatrix} -l_{\mathrm{r}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v\cos\beta \\ v\sin\beta - l_{\mathrm{r}}r \\ 0 \end{bmatrix}. \tag{4.6}$$

From Fig. 4.2 can be seen that the vector of the velocity of the tire contact point for front $\vec{\mathbf{v}}_{\mathrm{f}}$ and rear $\vec{\mathbf{v}}_{\mathrm{r}}$ wheels also can be written as:

$$\vec{\mathbf{v}}_{\mathrm{f}} = \begin{bmatrix} v\cos\beta \\ v\sin\beta + l_{\mathrm{f}} r \\ 0 \end{bmatrix} = \begin{bmatrix} v_{\mathrm{f}}\cos\left(\delta - \alpha_{\mathrm{f}}\right) \\ v_{\mathrm{f}}\sin\left(\delta - \alpha_{\mathrm{f}}\right) \\ 0 \end{bmatrix}, \tag{4.7}$$

$$\vec{\mathbf{v}}_{\mathrm{r}} = \begin{bmatrix} v\cos\beta \\ v\sin\beta - l_{\mathrm{r}} r \\ 0 \end{bmatrix} = \begin{bmatrix} v_{\mathrm{r}}\cos\alpha_{\mathrm{r}} \\ -v_{\mathrm{r}}\sin\alpha_{\mathrm{r}} \\ 0 \end{bmatrix}, \tag{4.8}$$

where v_{f} and v_{r} are absolute values of velocities of the tire contact point for front and rear wheels, α_{f} and α_{r} are side-slip angles for front and rear wheels. From (4.7) and (4.8), relationships between side-slip angles α_{f} and α_{r} and the steering angle δ can be derived through states β, r, input δ, and known parameters of the system. The angle between v_{f} and front wheel's x-axis is $\delta - \alpha_{\mathrm{f}}$. The angle between v_{r} and x-axis of the rear wheel is α_{r}. Therefore, the following statements are valid:

$$\frac{v_{\mathrm{f}}\sin\left(\delta - \alpha_{\mathrm{f}}\right)}{v_{\mathrm{f}}\cos\left(\delta - \alpha_{\mathrm{f}}\right)} = \tan\left(\delta - \alpha_{\mathrm{f}}\right) = \frac{v\sin\beta + l_{\mathrm{f}} r}{v\cos\beta}, \tag{4.9}$$

$$-\frac{v_{\mathrm{r}}\sin\alpha_{\mathrm{r}}}{v_{\mathrm{r}}\cos\alpha_{\mathrm{r}}} = -\tan\alpha_{\mathrm{r}} = \frac{v\sin\beta - l_{\mathrm{r}} r}{v\cos\beta}. \tag{4.10}$$

Again, for small deviations of sides-slip angle β from zero, $\cos\beta \approx 1$ and $\sin\beta \approx \beta$. Tangents can be threatened as a linear function with a unit slope. Thus, the following linear approximation of side-slip angles of wheels can be derived:

$$\tan\left(\delta - \alpha_{\mathrm{f}}\right) \approx \beta + \frac{l_{\mathrm{f}} r}{v} \Rightarrow \alpha_{\mathrm{f}} \approx \delta - \beta - \frac{l_{\mathrm{f}} r}{v}, \tag{4.11}$$

$$\tan\alpha_{\mathrm{r}} \approx -\beta + \frac{l_{\mathrm{r}} r}{v} \Rightarrow \alpha_{\mathrm{r}} \approx -\beta + \frac{l_{\mathrm{r}} r}{v}. \tag{4.12}$$

Equations of motion also require forces affecting the vehicle. This work assumes that normal forces $F_{\mathrm{z,f}}$ and $F_{\mathrm{z,r}}$ acting on axles vary and can be measured directly. However, when the body of the vehicle remains at the rest, normal forces can be calculated in the following manner:

$$F_{\mathrm{z,f}} = mg\frac{l_{\mathrm{f}}}{l_{\mathrm{f}} + l_{\mathrm{r}}}, \tag{4.13}$$

$$F_{\mathrm{z,r}} = mg\frac{l_{\mathrm{r}}}{l_{\mathrm{f}} + l_{\mathrm{r}}}. \tag{4.14}$$

Lateral traction forces can be described linearly through nominal cornering stiffness c and normal force, which is acting on a particular axle, as:

$$F_{\mathrm{y,r}} = c_{\mathrm{r}} F_{\mathrm{z,r}} \alpha_{\mathrm{r}}, \tag{4.15}$$

$$F_{\mathrm{y,f}} = c_{\mathrm{f}} F_{\mathrm{z,f}} \alpha_{\mathrm{f}}. \tag{4.16}$$

The described system has 2 degree of freedom, motions equations represent planar

motion (which is derived earlier) and rotational motion:

$$mv(\dot{\beta} + r) = F_{\mathrm{y,f}} + F_{\mathrm{y,r}}, \tag{4.17}$$

$$I\dot{r} = l_{\mathrm{f}} F_{\mathrm{y,f}} - l_{\mathrm{r}} F_{\mathrm{y,r}}. \tag{4.18}$$

The final look of the motion equations is the following:

$$mv(\dot{\beta} + r) = c_{\mathrm{f}} F_{\mathrm{z,f}}(\delta - \beta - \frac{l_{\mathrm{f}}}{v} r) + c_{\mathrm{r}} F_{\mathrm{z,r}}(-\beta + \frac{l_{\mathrm{r}}}{v} r), \tag{4.19}$$

$$I\dot{r} = l_{\mathrm{f}} c_{\mathrm{f}} F_{\mathrm{z,f}}(\delta - \beta - \frac{l_{\mathrm{f}}}{v} r) - l_{\mathrm{r}} c_{\mathrm{r}} F_{\mathrm{z,r}}(-\beta + \frac{l_{\mathrm{r}}}{v} r). \tag{4.20}$$

Equations (4.19) and (4.20) are written with assumptions and simplification, which were mentioned earlier. The first one is an assumption that the vehicle is driven with constant velocity and that both axles' traction forces acting in the longitudinal direction are neglected. The second one is the usage of linear approximations of side-slip angles and linear approximations of lateral traction forces using nominal cornering stiffness of a tire and load force acting on an axle.

Equation (4.19) describes how the lateral force acts on CG. Equation (4.20) describes how rotational torque is applied around CG.

The overall linear lateral vehicle dynamics can be presented as a state-space model as

$$\dot{x} = Ax + Bu, \tag{4.21a}$$

where system dynamics matrices are defined as

$$A = \begin{bmatrix} \frac{c_{\mathrm{f}} F_{\mathrm{zf}} + c_{\mathrm{r}} F_{\mathrm{zr}}}{mv} & \frac{l_{\mathrm{r}} c_{\mathrm{r}} F_{\mathrm{zr}} - l_{\mathrm{f}} c_{\mathrm{f}} F_{\mathrm{zf}}}{mv^2} - 1 \\ \frac{l_{\mathrm{r}} c_{\mathrm{r}} F_{\mathrm{zr}} - l_{\mathrm{f}} c_{\mathrm{f}} F_{\mathrm{zf}}}{I} & -\frac{l_{\mathrm{r}}^2 c_{\mathrm{r}} F_{\mathrm{zr}} + l_{\mathrm{f}}^2 c_{\mathrm{f}} F_{\mathrm{zf}}}{vI} \end{bmatrix}, \tag{4.21b}$$

$$B = \begin{bmatrix} \frac{c_{\mathrm{f}} F_{\mathrm{zf}}}{mv} \\ \frac{l_{\mathrm{f}} c_{\mathrm{f}} F_z f}{I} \end{bmatrix}, \tag{4.21c}$$

$$x = \begin{bmatrix} \beta \\ r \end{bmatrix}, \tag{4.21d}$$

$$u = \delta. \tag{4.21e}$$

This is a basic single-track which will be augmented and expanded in the next section.

4.3 Single-Track Model Augmentation for Environmental Constrains

The linear single-track serves to model planar vehicle motion described by a constant velocity v, sideslip angle β, and yaw rate r. However, to present environmental constraints, the linear single-track must be augmented by position planar coordinates p_{x} and p_{y} and heading (yaw angle) ψ, increasing the number of states up to 5. The work

uses an inertial frame of reference depicted in Fig. 4.3 with the origin defined in CG for each step in $k = 0$ (for the first frame). The augmented model now consists of 5 states and 2 inputs, which are listed in Table 4.3. All parameters used in the model are presented in Table 4.4.

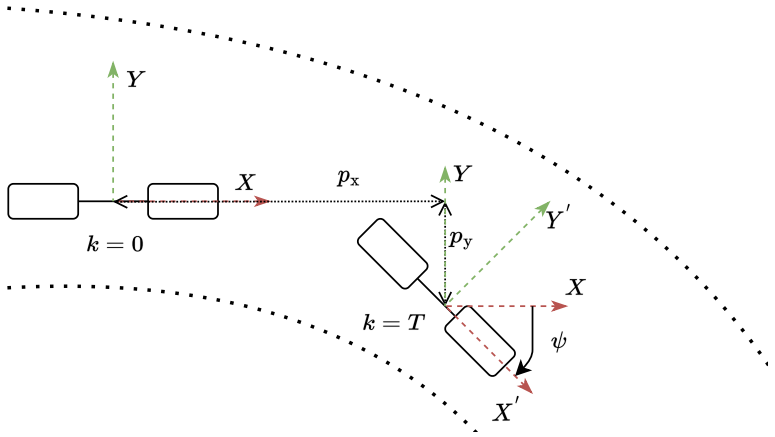


Figure 4.3: Sketch of inertial frame of reference. Visualization of changing parameters p_x, p_y, and ψ during future trajectory prediction

Table 4.3: States and inputs of the augmented model

State/Input	Symbol	Units
Side-slip angle	β	rad
Yaw rate	r	$\mathrm{rad\,s^{-1}}$
Heading angle	ψ	rad
Position of CG on x-axis	p_x	m
Position of CG on y-axis	p_y	m
Steering angle of the front axle	δ	rad

Table 4.4: Parameters of the augmented model

Parameter	Symbol	Units
Distance from CG	l	m
Vehicle mass	m	kg
Yaw moment of inertia	I	$\mathrm{kg\,m^2}$
Nominal cornering stiffness	c	$\mathrm{rad^{-1}}$
Load force of wheels	F_z	N
Description	**Subindex**	
Lateral or longitudinal direction	x or y	
Front or rear axle	f or r	
Left or right side	l or r	

The heading angle ψ is calculated as a simple integration of the yaw rate r. CG's position p_x, p_y is integral of the first two components of the velocity vector (4.1). Vector $\vec{\mathbf{v}}$ contains nonlinear trigonometrical functions, which are essential for position calculation. Those nonlinearities make the final augmented single track model, which is used for the controller implementation, nonlinear:

$$\dot{x} = f(x) + Bu, \tag{4.22a}$$

$$f(x) = \begin{bmatrix} -\frac{c_f F_{zf} + c_r F_{zr}}{mv}\beta + (\frac{l_r c_r F_{zr} - l_f c_f F_{zf}}{mv^2} - 1)r \\ \frac{l_r c_r F_{zr} - l_f c_f F_{zf}}{I}\beta - \frac{l_r^2 c_r F_{zr} + l_f^2 c_f F_{zf}}{vI} r \\ r \\ v\cos\psi \\ v\sin\psi \end{bmatrix}, \tag{4.22b}$$

$$B = \begin{bmatrix} \frac{c_f F_{zf}}{mv} \\ \frac{l_f c_f F_{zf}}{I} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.22c}$$

$$x = \begin{bmatrix} \beta \\ r \\ \psi \\ p_x \\ p_y \end{bmatrix}, \tag{4.22d}$$

$$u = \delta. \tag{4.22e}$$

The described system is continuous, but the model predictive controller works with discrete systems. Therefore, the system (4.22) must be discretized. Discretization of the system completed by zero-order hold method:

$$\dot{x}_{k+1} = f_d(x_k) + B_k u_k, \tag{4.23a}$$

$$f_d(x_k) = \mathrm{ZOH}(f(x)), \tag{4.23b}$$

$$B_d = \mathrm{ZOH}(B). \tag{4.23c}$$

Discrete model 4.23 is used in the next chapter Control Architecture.

5. Control Architecture

5.1 General Concept of the Proposed Advanced Driver Assistance System

This work presents an active safety system, which is always online. It relentlessly checks the vehicle's future position and detected road boundaries and corrects drivers' control actions applied from the steering wheel when it detects possible future violations of the environmental envelope constraints shown in Fig. 5.1 (which are derived below in this chapter). An active safety system format was chosen because it allows the realization of the potential of modern vehicles, which today are equipped with a variety of sensors and onboard computers with enough computational capabilities. The designed control architecture uses the model predictive control technique and is a typical model-based approach (from the classification described in Section 3.2.3).

Figure 5.1: Road with a pothole adopted from [26]. In the figure, the detected limitations (environmental envelope) are presented. The red curve stands for the left road border. The green curve shows the right edge. Yellow color encircles boundary defined by a detected pothole. It is a circle from the top-view perspective and an ellipse from a driver's view.

The proposed ADAS uses cameras to detect road boundaries and drivable obstacles. The algorithm for that detection is not a part of this work. However, there is much camera-based research about road borders and road abnormalities detection [26], [27], [28]. Mentioned research outcomes reduce the difficulty of future im-

plementation of the presented ADAS on a real vehicle.

The last reason to choose an active safety system (instead of autopilot) is human tolerance toward autonomous systems. According to a study provided by the American Automobile Association [29], 73 percent of U.S. drivers consider themselves above-average drivers, and 63 percent of U.S. drivers are afraid to ride in a fully self-driving vehicle. Mentioned research shows drivers' attitudes toward self-driving cars and driving automation. The less activity provided by the control architecture is felt by the driver, the more chance to be used by the real driver is has in the future.

According to the classification of levels of automation provided in section 3, the presented concept is Level 1 of automation because it operates only with the steering wheel and changes only the lateral vehicle dynamics. It will make ADAS easy to adapt to a real car and allow increasing acceptance of the proposed control system by the drivers' community. The proposed controller could be modified to achieve a higher level of automation.

The ADAS can be upgraded by adding the angular velocity of the front wheel operation in the controller (for FWD car) as the second input and velocity of the vehicle as a state to the single-track model. In this case, the ADAS will regulate steering and throttle/braking as related functions, which is the condition of Level 2 of automation.

Level 3 of automation can also be achieved. An example of the third level is conditional automation on a forest (or village) road (Fig. 5.2). In this case, ruts could define borders for each side of the vehicle separately as MPC constraints. The discussion controller could be a part of an autonomous driving concept, which will keep the vehicle on (or away from) ruts. These changes will allow the control concept to meet requirements for the third automation level, conditional automation, under defined circumstances.

(a) normal view

(b) possible constrains definition based on edges of the ruts

Figure 5.2: An example of a forest road.

5.1.1 Model Predictive Control Technique

Model predictive control is an optimal control method in which the resulting control input is calculated by minimizing a cost function by satisfying a set of constraints, part of which is a system dynamic (model description), over a finite horizon. The optimal control problem is solved reputedly, in each timestamp taking only the first step from a solution. The main advantage is that input in the current time is optimized by keeping future states in an account. Figure 5.3 presents a schematic representation of an MPC.

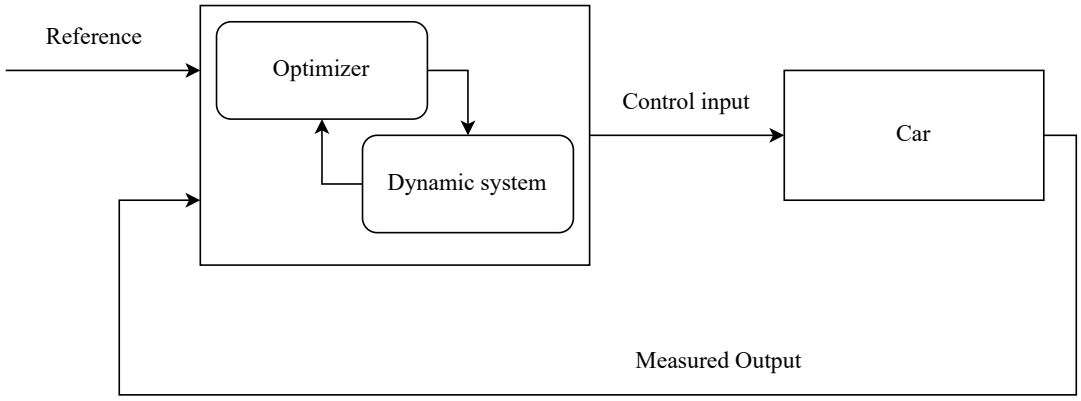


Figure 5.3: Diagram of an MPC applied to control field (a car).

To summarize: an MPC is a multivariable control algorithm that uses:

- an internal dynamic model of the process;
- a cost function J over the receding horizon;
- an optimization algorithm minimizing the cost function J using the control input u.

The primary task of this work is to create an MPC that ensures vehicle operation within the driving tube, which is defined by road lines and excludes drivable obstacles (road regularities). The controller must be minimally invasive to the drivers' operational commands while avoiding harsh interventions and ensuring compliance with conditions.

5.2 Optimal Problem Formulation

Generally, the environmental envelope contains limitations for a vehicle defined by the environment. This work considers a road environment with limitations presented by static elements such as road borders and drivable obstacles (road defects). The controller designed under assumption that the vehicle send to MPC such dynamical parameters of the car as: load force of the front wheels $F_{\mathrm{z,f}}$, load force of rear wheels $F_{\mathrm{z,r}}$; states of the car: side-slip angle β, yaw rate r, heading angle yaw ψ; input generated by human driver in form of steering, that before sending to MPC is recalculated to the steering angle of the front axle δ.

The general scheme of chosen control solution can be found in Fig. 5.4. The control scheme works in the following way: the commanded angle of the steering wheel is discretized and recalculated on a rotation of the wheels. After that, modified driver's input with detected lines, measured car's parameters, and states are sent to MPC. The MPC generates control input for the vehicle according to the predefined environmental envelope boundaries. The control input (output of the MPC) is recalculated back to the steering wheel angle and sent to the car.

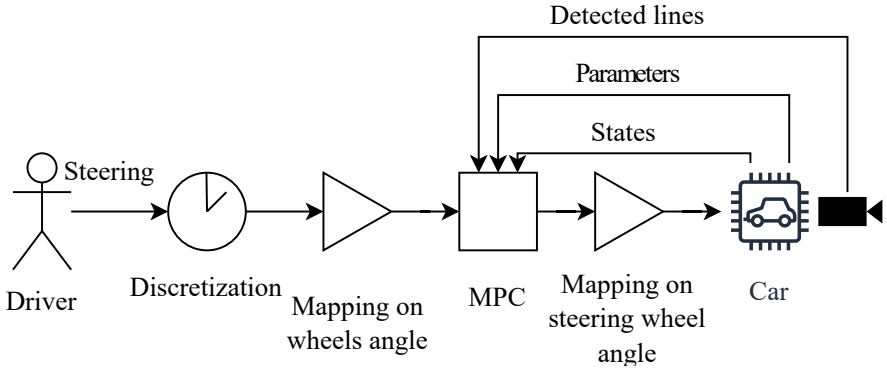


Figure 5.4: Proposed ADAS architecture. The commanded steering angle is scaled to fit the actual rotation of steered wheels. MPC calculates steering angle closely as possible to the commanded value subject to the presented constraints. The output value is scaled back to the angle of the steering wheel.

Table 5.1 lists symbols used below for the mathematical description of the MPC.

Description	Symbol
Distance from CG to a wheel (see Fig. 5.5)	d
Angle to CG to a wheel's anchor point (see Fig. 5.5)	γ
X and Y coordinates of a wheel (position of a wheel is filled in $\square$)	$p_{\mathrm{x},\square}$, $p_{\mathrm{y},\square}$
Weight for driver-input tracking	w_{d}
Weight for input change penalization	w
Weight for road borders slack penalization	q
Weight for delta slew slack penalization	q_{s}
Coefficient for borderline description	a
Slacks for soft-constraining of borderlines	s
Slacks for soft-constraining of Delta slew	s_δ
Safety distance from road boundaries	z
Input change	$\mathrm{d}u$
Driver's input	δ_{driver}
Rate limit for input change	δ_{slew}
Description	**Subindex**
Lateral or longitudinal direction	x or y
Front or rear axle	f or r
Left or right side	l or r

Table 5.1: Description of symbols used in MPC

In the next subsections optimal problem will be formulated for both functions of

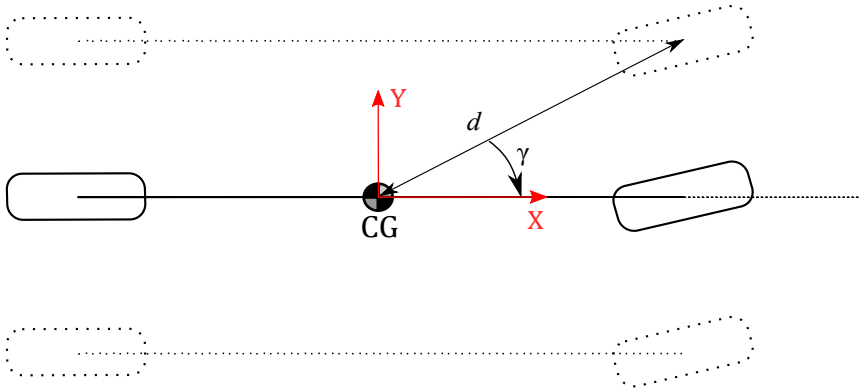


Figure 5.5: Positions of wheels are described using distance from its anchor point and CG and angle between the x-axis and the anchor point.

the chosen control concept: lane-keeping and obstacle avoidance.

5.2.1 Road Boarder Representation

Road borders are obtained by camera-based line detection. Line is detected as a set of points (Fig. 5.6). Then, those values are used to approximate road boundary as a parabolic function. Calculated parameters (a_0, a_1, a_2) are used to represent a border as follows:

$$y = a_2 x^2 + a_1 x + a_0. \tag{5.1}$$

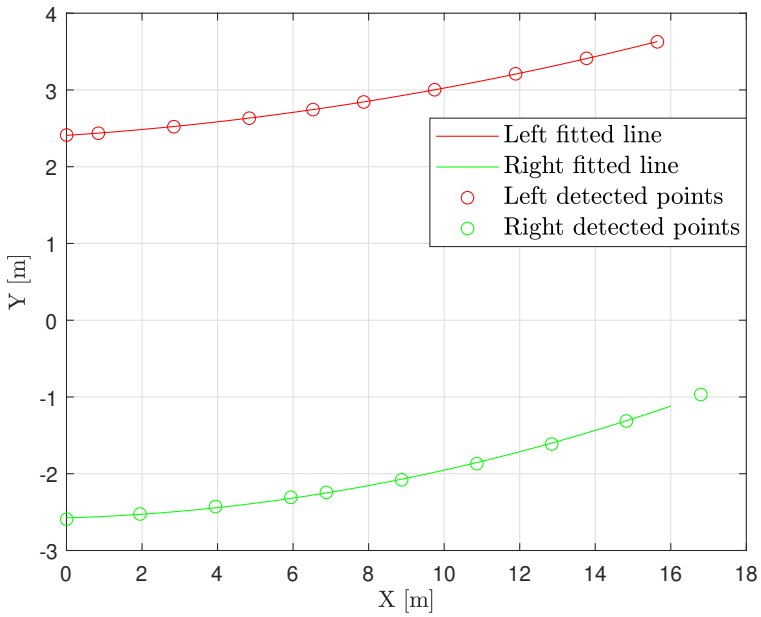


Figure 5.6: Visualization of sets of points returned from the line-detection sensor and approximations made by parabolic function.

Equation (5.1) is then used to calculate the road's upper (for the left side) or lower (for the right side) boundary.

5.2.2 Baseline Solution with Lane-Keeping Constraints

This section derives a baseline MPC with vehicle dynamics and road boundary constraints. The MPC is nonlinear in our case because the model used in the prediction is nonlinear (4.22a). Therefore, the optimization problem 5.6 is non-convex.

First, the objective function J, which will be minimized, has to be derived to define the MPC. This work assumes the following objective function:

$$J = \sum_k (w_\mathrm{d}|\delta_\mathrm{driver} - u_k|) + \sum_k (w_\mathrm{d}(\delta_\mathrm{driver} - u_k)^2) + \sum_k w \cdot \mathrm{d}u^2 + \sum_{k+1} kq \cdot s_{\mathrm{fl},k}^2 + \\ + \sum_{k+1} kq \cdot s_{\mathrm{fr},k}^2 + \sum_{k+1} kq \cdot s_{\mathrm{rl},k}^2 + \sum_{k+1} kq \cdot s_{\mathrm{rr},k}^2 + \sum_k q_\mathrm{s} \cdot s_{\delta,k}^2. \quad (5.2)$$

The MPC has to track commanded steering angle closely as possible. Therefore, the following two terms are presented in the objective function:

$$\sum_k (w_\mathrm{d}|\delta_\mathrm{driver} - u_k|), \quad (5.3)$$

and

$$\sum_k (w_\mathrm{d}(\delta_\mathrm{driver} - u_k)^2). \quad (5.4)$$

While term (5.4) penalizes significant deviations from the reference, term (5.3) reduces minor ones.

To ensure continuity of steering and prevent oscillation, the cost function should contain penalization for fast changes in the input

$$\sum_k w \cdot \mathrm{d}u^2, \quad (5.5)$$

where $\mathrm{d}u = u_k - u_{k-1}$.

Resting terms in (5.2) are defined to penalize used slack variables. Those slack vectors are presented for softening hard constraints to avoid numerical problems and possible infeasibilities in MPC formulation. Weight q for slack vectors for soft-constraining of borderlines is additionally multiplied by variable k to force the controller to take action earlier and avoid a rapid change of steering angle.

Vehicle dynamic is descritized (see Eq. 4.23) to represent dynamical constraints in MPC as

$$x_k = f_\mathrm{d}(x_{k-1}) + B_\mathrm{d} u_{k-1}. \quad (5.6)$$

The necessary physical constraint is the minimal and maximal angle of rotation of the

wheels:

$$u_{\min} \leq u_k \leq u_{\max}, \tag{5.7}$$

and limitation on available change in steering angle:

$$-\delta_{\text{slew}} + s_{\delta,k} \leq \mathrm{d}u \leq \delta_{\text{slew}} + s_{\delta,k}. \tag{5.8}$$

MPC's line-keeping part is represented in position limitations for each wheel. Each wheel must be under the upper (left road boundary) border and above the bottom (right road boundary) border. Each border is represented as an equation of a second-order curve (5.1). The final look of the constraints for MPC that represent line-keeping has the following look of four inequalities, one for each wheel:

$$a_{0,\text{l}} - z \geq p_{\text{y,fl},k} - a_{2,\text{l}} \cdot p^2_{\text{x,fl},k} - a_{1,\text{l}} \cdot p_{\text{x,fl},k} - s_{\text{fl},k}, \tag{5.9a}$$

$$a_{0,\text{r}} + z \leq p_{\text{y,fr},k} - a_{2,\text{r}} \cdot p^2_{\text{x,fr},k} - a_{1,\text{r}} \cdot p_{\text{x,fr},k} + s_{\text{fr},k}, \tag{5.9b}$$

$$a_{0,\text{l}} - z \geq p_{\text{y,rl},k} - a_{2,\text{l}} \cdot p^2_{\text{x,rl},k} - a_{1,\text{l}} \cdot p_{\text{x,rl},k} - s_{\text{fl},k}, \tag{5.9c}$$

$$a_{0,\text{r}} + z \leq p_{\text{y,rr},k} - a_{2,\text{r}} \cdot p^2_{\text{x,rr},k} - a_{1,\text{r}} \cdot p_{\text{x,rr},k} + s_{\text{fr},k}, \tag{5.9d}$$

where z (Fig 5.7) is used to create a safe distance from borders. Previous constraints are used with equalities, representing car geometry and calculating the position of a particular wheel with respect to the position of CG:

$$p_{\text{x},\square\square,k} = p_{\text{x},k} + d \cdot \cos(\gamma_{\square,\square}), \tag{5.10a}$$

$$p_{\text{y},\square\square,k} = p_{\text{y},k} + d \cdot \sin(\gamma_{\square,\square}), \tag{5.10b}$$

where $\square$ is used to replace sub-caption for all possible combination in wheel description (front/rear, left/right), and d is a distance from CG to wheel anchor point (can be found on Fig. 5.5).

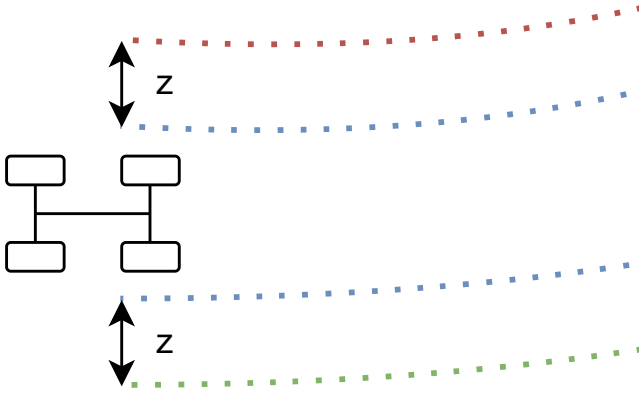


Figure 5.7: Sketch of line-keeping constrains representation. Variable z is used as a control parameter, representing a safety distance to road borders.

The final optimization problem has a following look:

$$\min \sum_k (w_\mathrm{d} |\delta_\mathrm{driver} - u_k|) + \sum_k (w_\mathrm{d} (\delta_\mathrm{driver} - u_k)^2) + \sum_k w \cdot \mathrm{d}u^2 + \sum_{k+1} kq \cdot s_{\mathrm{fl},k}^2 + \\ + \sum_{k+1} kq \cdot s_{\mathrm{fr},k}^2 + \sum_{k+1} kq \cdot s_{\mathrm{rl},k}^2 + \sum_{k+1} kq \cdot s_{\mathrm{rr},k}^2 + \sum_k q_\mathrm{s} \cdot s_{\delta,k}^2. \quad (5.11a)$$

subject to

$$x_k = f(x_{k-1}) + g(u_k), \quad (5.11b)$$

$$u_\mathrm{min} \leq u_k \leq u_\mathrm{max}, \quad (5.11c)$$

$$-\delta_\mathrm{slew} + s_{\delta,k} \leq \mathrm{d}u \leq \delta_\mathrm{slew} + s_{\delta,k}, \quad (5.11d)$$

$$p_{\mathrm{x,fl},k} = p_{\mathrm{x},k} + d \cdot \cos(\gamma_{\mathrm{l,f}}), \quad (5.11e)$$

$$p_{\mathrm{y,fl},k} = p_{\mathrm{y},k} + d \cdot \sin(\gamma_{\mathrm{l,f}}), \quad (5.11f)$$

$$p_{\mathrm{x,fr},k} = p_{\mathrm{x},k} + d \cdot \cos(\gamma_{\mathrm{r,f}}), \quad (5.11g)$$

$$p_{\mathrm{y,fr},k} = p_{\mathrm{y},k} + d \cdot \sin(\gamma_{\mathrm{r,f}}), \quad (5.11h)$$

$$p_{\mathrm{x,rl},k} = p_{\mathrm{x},k} + d \cdot \cos(\gamma_{\mathrm{l,r}}), \quad (5.11i)$$

$$p_{\mathrm{y,rl},k} = p_{\mathrm{y},k} + d \cdot \sin(\gamma_{\mathrm{l,r}}), \quad (5.11j)$$

$$p_{\mathrm{x,rr},k} = p_{\mathrm{x},k} + d \cdot \cos(\gamma_{\mathrm{r,r}}), \quad (5.11k)$$

$$p_{\mathrm{y,rr},k} = p_{\mathrm{y},k} + d \cdot \sin(\gamma_{\mathrm{r,r}}), \quad (5.11l)$$

$$a_{0,\mathrm{l}} - z \geq p_{\mathrm{y,fl},k} - a_{2,\mathrm{l}} \cdot p_{\mathrm{x,fl},k}^2 - a_{1,\mathrm{l}} \cdot p_{\mathrm{x,fl},k} - s_{\mathrm{fl},k}, \quad (5.11m)$$

$$a_{0,\mathrm{r}} + z \leq p_{\mathrm{y,fr},k} - a_{2,\mathrm{r}} \cdot p_{\mathrm{x,fr},k}^2 - a_{1,\mathrm{r}} \cdot p_{\mathrm{x,fr},k} + s_{\mathrm{fr},k}, \quad (5.11n)$$

$$a_{0,\mathrm{l}} - z \geq p_{\mathrm{y,rl},k} - a_{2,\mathrm{l}} \cdot p_{\mathrm{x,rl},k}^2 - a_{1,\mathrm{l}} \cdot p_{\mathrm{x,rl},k} - s_{\mathrm{fl},k}, \quad (5.11o)$$

$$a_{0,\mathrm{r}} + z \leq p_{\mathrm{y,rr},k} - a_{2,\mathrm{r}} \cdot p_{\mathrm{x,rr},k}^2 - a_{1,\mathrm{r}} \cdot p_{\mathrm{x,rr},k} + s_{\mathrm{fr},k}, \quad (5.11p)$$

$$s_{\mathrm{fl},k} \geq 0, \quad (5.11q)$$

$$s_{\mathrm{fr},k} \geq 0, \quad (5.11r)$$

$$s_{\mathrm{rl},k} \geq 0, \quad (5.11s)$$

$$s_{\mathrm{rr},k} \geq 0, \quad (5.11t)$$

$$x_0 = x(t), u_{-1} = u(t - T_\mathrm{s}) \quad (5.11u)$$

where variables to be optimized are the optimal input u_k. Tunable parameters in this optimization problem are weight w_d, which establishes the trade-off between a path chosen by MPC and matching the driver's present input, penalization of rapid input changing w and q_s, and penalization of border violation q. Instead of enforcing the driving envelope as hard constraints, the slack variables with high weights provide a possibility to violate the envelope constraints, increasing the feasibility of the optimization problem.

Also this formulation of optimization problem with some change can be used to keep the vehicle in a ruts (Fig 5.2). For this purpose, equations from 5.11m to 5.11p should be rewritten to describe limitations from the top and bottom lines for both left

and right wheels of the car.

5.2.3 Drivable Obstacle Avoidance Constraints

In this section, we presume that drivable obstacles have a round shape with a pre-knowing radius r. New symbols used to describe obstacle avoidance can be fond in Table 5.2.

Description	Symbol
Distance from an obstacle to a CG	λ
Slack variable	τ
Radius of drivable obstacle	r
Slack weight	w_τ
Description	**Subindex**
Lateral or longitudinal direction	x or y
Front or rear axle	f or r
Left or right side	l or r

Table 5.2: Description of symbols used in MPC for obstacle avoidance

Assumption that all drivable obstacle may be replaced by circle with known radius allow to add this obstacle to optimization problem as the next constraints:

$$(p_{\mathrm{y},\square\square,\mathrm{k}} - \lambda_{\mathrm{y},k})^2 + (p_{\mathrm{x},\square\square,\mathrm{k}} - \lambda_{\mathrm{x},k})^2 \geq r^2 - \tau_k, \tag{5.12a}$$

$$\tau_k > 0. \tag{5.12b}$$

Inequalities 5.12 mark zone with radius r and located relatively CG (Fig. 5.8) as that should be avoided by center of a wheel also described relatively CG ($p_{\mathrm{x},\square\square}$, $p_{\mathrm{y},\square\square}$).

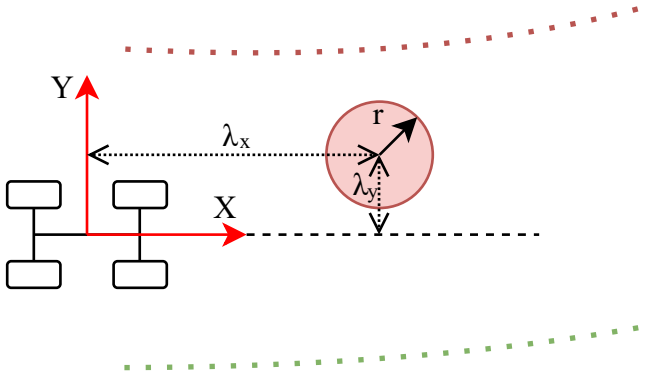


Figure 5.8: Sketch of line-keeping with obstacle avoidance constrains representation. Variable r is used as a control parameter, representing a radius of zone from center of a obstacle that may be avoided by center of a wheel.

This work use inequalities 5.12 just for front wheels because this allow increase computational speed with no loss in performance with assumption that rear wheels

track path of forward wheels. Cost function J should be expanded by adding

$$J' = J + \sum_{k+1} k w_\tau \cdot \tau_k^2, \tag{5.13}$$

with w_τ as tunable parameters multiplied by k to force the controller to take action earlier and avoid a rapid change of steering angle. .

Equation 5.12a can be used for drivable obstacle avoidance scaling and increasing number of obstacle for avoiding. In this situation, the equation 5.12a should be written for every obstacle and the variable λ will represent the distance to a specific obstacle.

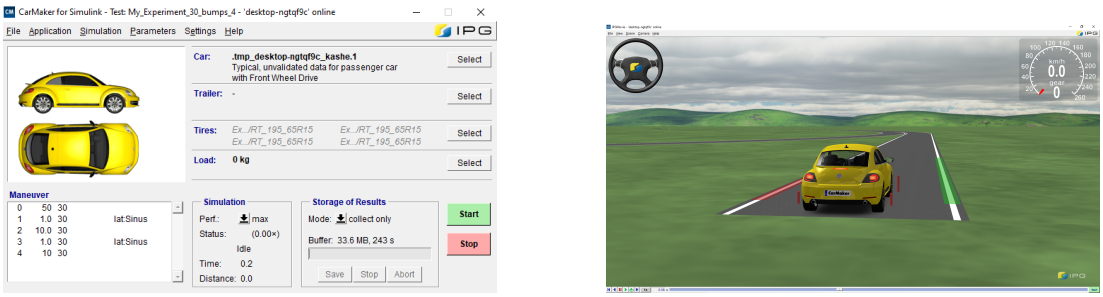
6. Experiments

This chapter describes the testing environment, test car configuration, and controller implementation. Experiments results and discussion are also presented in this chapter.

6.1 Environment

IPG CarMaker is a vehicle dynamics simulator that was explicitly designed for the development and seamless testing of cars and light-duty vehicles in all development stages. The open integration and test platform allow the implementation of virtual test scenarios for such application areas as autonomous vehicles, ADAS, power-train, and vehicle dynamics. It includes the ride visualization tool MovieNX. Various supported standards and interfaces also guarantee smooth integration into existing tool landscapes.

The CarMaker offers a variety of possibilities for data analysis and therefore increases transparency. On the one hand, significant amounts of data can be stored flexibly for subsequent analysis with third-party software. On the other hand, essential data can be monitored directly during the test run with the integrated IPGControl tool. It also enables the comparison of data from multiple test runs. Furthermore, the virtual dashboard instruments guarantee seamless access to all relevant instruments, scales, and information on the driver's behavior (e.g., steering wheel angle, gear selection) as well as on the vehicle's status (e.g., ignition, tachometer, turn signal lights).[30]



(a) IPG CarMaker interface

(b) IPG CarMaker visualization tool

Figure 6.1: IPG CarMaker screenshots.

IPG CarMaker driver (IPGDriver) is used for test automation. IPGDriver allow user to add the control actions of a human driver to complete vehicle simulation. These actions include steering, braking, gas pedal position, gear shifting and clutch operation. The environment contains different drivers to model various type of driving styles. Detailed description of IPGDriver can be found in User Manual IPGDriver (available in IPG CarMaker, not available online)

IPG CarMaker can work with Matlab/Simulink environment. Simulink can exchange data online with IPG CarMaker. This makes it possible to implement the MPC controller using Matlab/Simulink environment.

6.2 Used Software for the Implementation of the Controller

A solver is a necessary part of any control loop to implement and apply an MPC controller. This thesis provides a proof of concept solution regardless time consumption of the resulting solution needed for the optimal control problem calculation. A toolbox for optimization needs to be chosen to reduce a workflow over MPC implementation. Also, a suitable solver needs to be selected.

The software chosen initially and assigned for this thesis are YALMIP [31], an optimization toolbox for MATLAB, and Gurobi [32], a solver for quadratic optimization problems with linear constraints. However, when I provided a mathematical description of the optimal control problem, which needs to be formulated, I decided to use nonlinear constrains to define the road boundaries. Therefore, the resulting problem toop a form of a quadratic optimization problem with nonlinear constraints. Changing the solver also influences the used optimization toolbox. YALMIP traditionally is a more frequent choice for optimal problems with linear constraints. On the other hand, CasADi is more frequently used with nonlinear control problems.

This thesis is part of the research of Toyota Recearch on Automation Cars in Europe [33]. Because of that, choices are limited to open-source software even if commercial solvers usually show better performance and provide a time-limited free academic license. The decision was made, and the choice fell on the CasADi optimization toolbox [34] with the IPOPT solver [35].

6.3 Configuration of Experiments

6.3.1 Used Vehicle

As a test vehicle, I decided to use the most common vehicle in the IPG CarMaker software. It is "DemoCar," which has statically understeering characteristics [36]. It is front-wheel driven and represents a typical city car trying to "copy" Volkswagen Beetle's dynamics and characteristics. All needed modeling parameters used in the vehicle model in Chapter 4 of this work are listed in Table 6.1.

A standard driver is used to model a behavior of an average driver. Parameters of the driver can be seen in Fig. 6.2.

Parameter	Symbol	Value	Units
Mass of the vehicle	m	1463	kg
Distance from CG to front axis	l_f	0.971	m
Distance from CG to rear axis	l_r	1.566	m
Yaw moment of inertia	I	1968	$\mathrm{kg\,m^2}$
Cornering stiffness front	c_f	134	$\mathrm{rad^{-1}}$
Cornering stiffness rear	c_r	98	$\mathrm{rad^{-1}}$

Table 6.1: Parameters of the vehicle

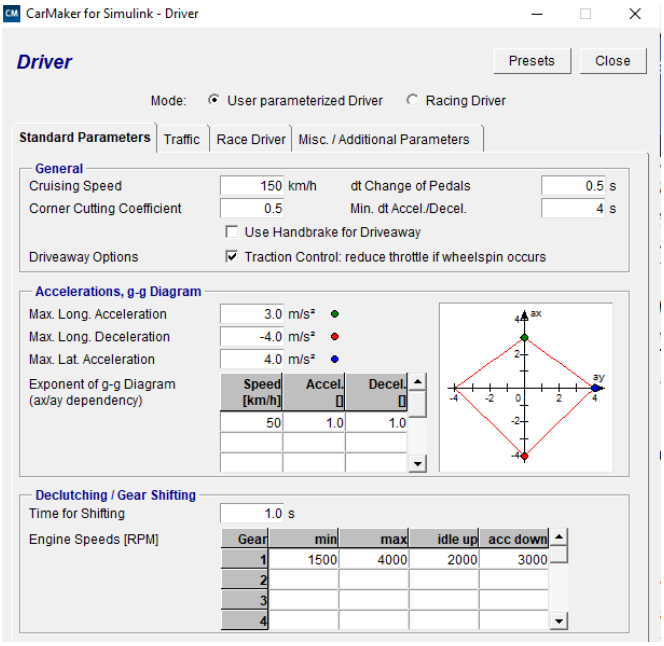


Figure 6.2: Configuration of the driver.

The following section will provide the parametrical configuration of the implemented ADAS.

6.3.2 Configuration of the Controller

The control logic is described in chapter Control Architecture. Table 6.2 contains tunable parameters that used in formulated optimal problem. The MPC receives both type of vehicle's parameters in Simulink: dynamical (nominal cornering stiffens c_f, c_r) and constant (mass m, yaw moment of inertia I, distance from CG to an axle l_f, l_r). The next signals to the MPC are states 4.22d and reference input 4.22e in current time t. Also the MPC receive detected lines 5.6 in format of points.

Description	Symbol	Value
Weights for driver-input tracking	R	$85 \cdot 10^{-4}$
Weights for input change penalization	w	0.2
Weights for slack penalization	q	10^2
Weights for input change rate penalization	q_s	10^2
Weights for radius slack	w_rad	10^3
Safety distance from road boundaries	z	0.5

Table 6.2: Tunable parameters value

The MPC is working on frequency 100 Hz and making predictions over one second (one hundred step horizon). The controller tracks the driver's input for five first steps. The optimization problem is defined for every third step to speed-up calculation. Performance time after optimization is less 0.01 in Real-Time.

6.3.3 Used Map

The map is a circle with a radius of 100 meters. The circle form of a road was chosen because this is a natural environment where a controller with a lane-keeping function can be used. Here, the driver must excite lateral vehicle dynamics in a constant turn. Also, the circle allows the simulation of an unexpected appearance of drivable obstacles that human drivers can miss a spot. Constant speed simulates movement within city limits (30 $\mathrm{km\,h^{-1}}$, 50 $\mathrm{km\,h^{-1}}$) and beyond dense city districts (72 $\mathrm{km\,h^{-1}}$). The map has divided into three virtual sections.

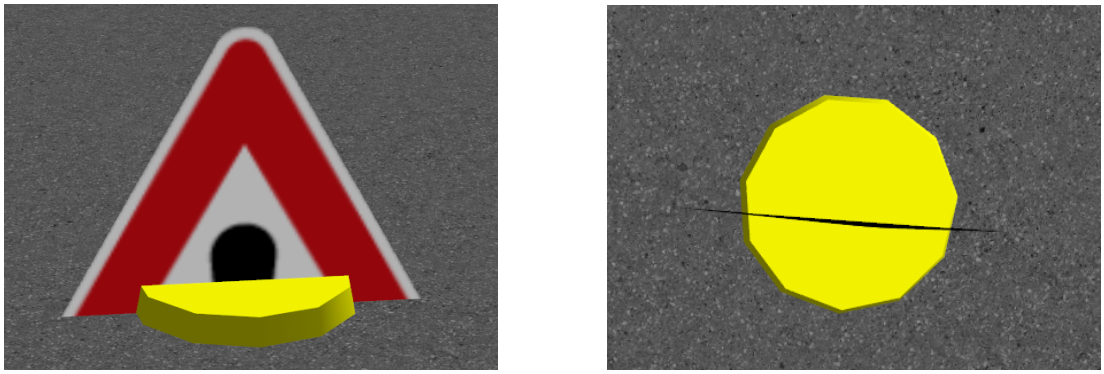
First section

The first section is the start section. Here, the MPC is not active. The vehicle accelerates and drives at a constant speed while getting to the next section.

Second section

The second road section contains bumps. Here, MPC is active, and the vehicle tries to avoid road bumps, which present drivable obstacles on the road. Driver does not react to road bumps.

A bump is a cone with 0.2 meters in radius and 5 millimeters in height. For detection purposes, a bump has a sign placed in the center. The look of a bump can be found in Fig 6.3a and Fig. 6.3b



(a) Bump from front view

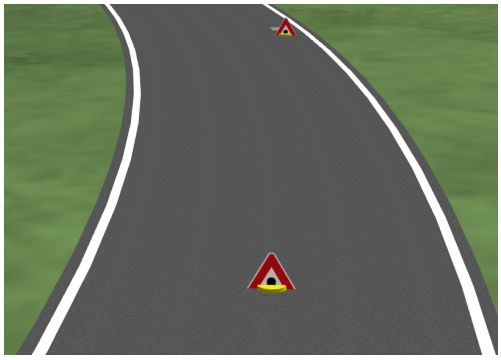
(b) Bump from top view

Figure 6.3: Front and top views on a bump, which represents a drivable obstacle.

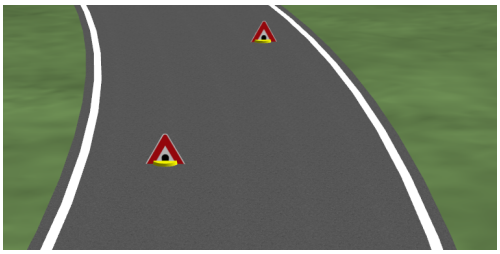
The bump section has two pairs of bumps placed on different parts of the road, shown in Fig. 6.4a and Fig. 6.4b.

Third section

The rest of the road is the section where a driver tries to leave the road, and the MPC should prevent this. The test represents momentary (1 second) loss of vehicle control. This loss in real life can be caused by phone calls or short-term sleep during driving. That can cause a dangerous situation, explicitly if a vehicle leaves a line and drive into



(a) First bump section. The distance between bumps is near 20 meters.



(b) Second bump section. The distance between bumps is near 10 meters.

Figure 6.4: Section with drivable obstacles on the road.

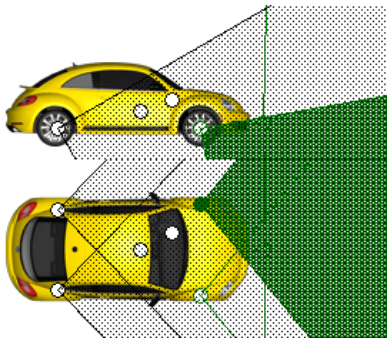
the opposite lane. In the simulation, a driver starts fast wheel rotation to the right and back. After ten seconds, a driver repeats the action for left turn.

6.3.4 Used Sensors

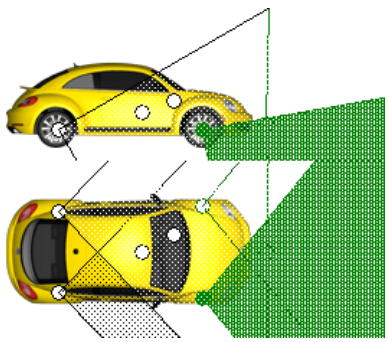
The vehicle is equipped with sensors for measuring yaw ψ, yaw rate r, velocity v, and slip angle β but this section tells about used sensors for drivable obstacles and lane detection.

Sign sensor

Using of sign sensor and sign for bump detection is a necessary measure. The IPG Carmaker has a large variety of sensors, including road and object sensors, but none of these sensors in the IPG Carmaker can detect a bump. In these circumstances, I decided to model bump detection with the help of a sign sensor and signs above obstacles (as was mentioned in the previous section). A sign represents the center of a bump and allows getting the required data for constraints in MPC. Also, because of that limit and surrogate solution of bump detection, I decided to assume that a drivable obstacle is a cone with the same predefined top and bottom radius.



(a) The first sign sensor is mounted above the front left wheel.



(b) The second sign sensor is mounted above the front right wheel.

Figure 6.5: Sign sensors location with its field of view marked by green color.

The car has two sign sensors with a 90-degree horizontal and vertical aperture. The first sensor is above the front left wheel, and the second one is placed above the front right wheel. The sensors' position and field of view are depicted in Fig. 6.5a and Fig. 6.5b.

Line sensor

The line sensor is used to detect lines, which represent road borders. It is located above the CG of the car. It has a horizontal and vertical aperture of 180 degrees in both directions. The line sensor's position and field of view can be seen in Fig. 6.6.

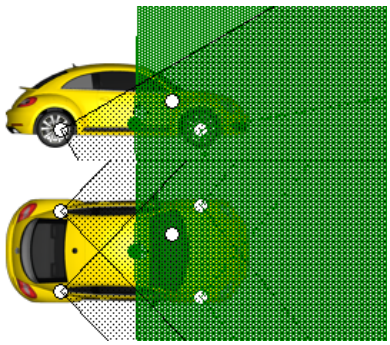


Figure 6.6: Line sensor and its field of view. It is mounted above the CG point.

The line sensor detects the right and left lines and returns a set of 10 coordinates in CG oriented vehicle frame for each line. Those points are approximated using a parabolic function, producing coefficients a_0, a_1, a_2 for MPC's constraints. Examples of data and line approximations are shown in Fig. 5.6.

6.4 Experiment Results

Tests were provided at three different speeds: 30 kilometers per hour, 50 kilometers per hour, and 72 kilometers per hour. Video recordings from this test can be found on the author's YouTube channel [37], [38], [39].

6.4.1 Test with Constant Speed at $30\ \mathrm{km\,h^{-1}}$

The first test is provided at $30\ \mathrm{km\,h^{-1}}$, simulating vehicle motion in a dense city center. The vehicle moved relatively slowly, and the designed control solution worked well, avoiding obstacles and keeping the vehicle within defined borders.

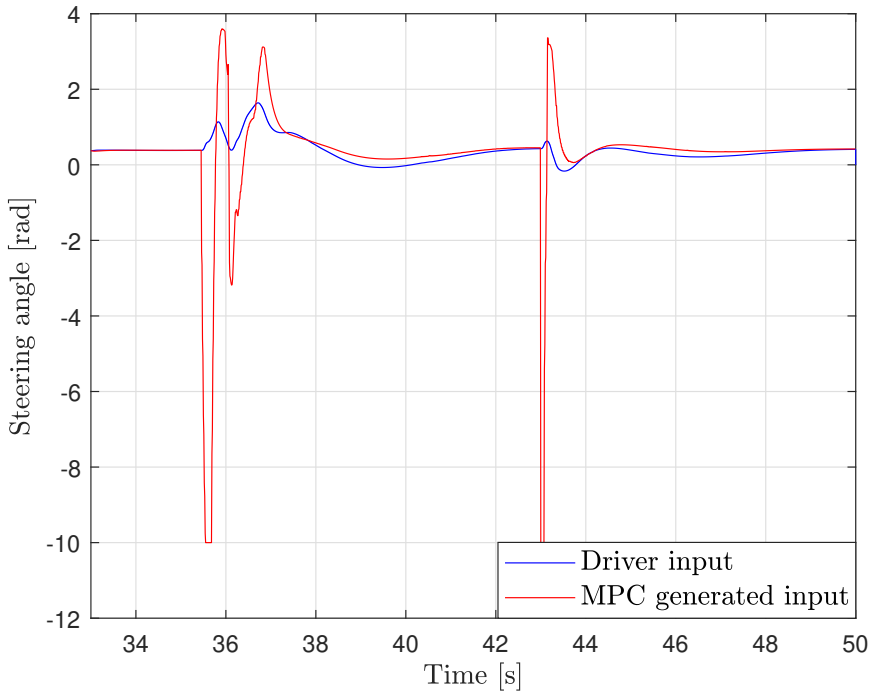


Figure 6.7: Steering wheel position during experiment provided at 30 km h^{-1}. Obstacle avoidance part.

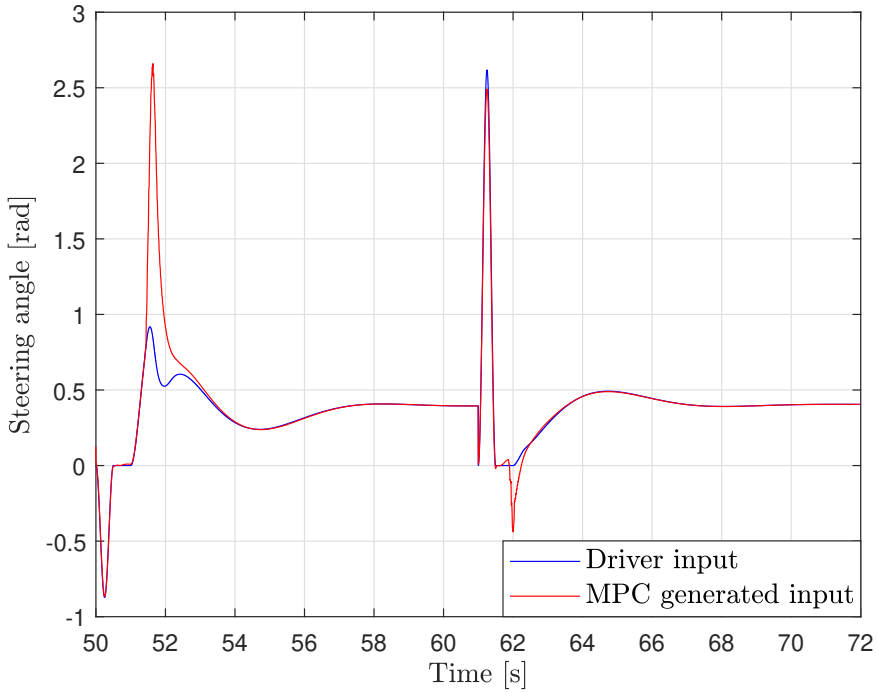


Figure 6.8: Steering wheel position during experiment provided at 30 km h^{-1}. Lane-keeping part.

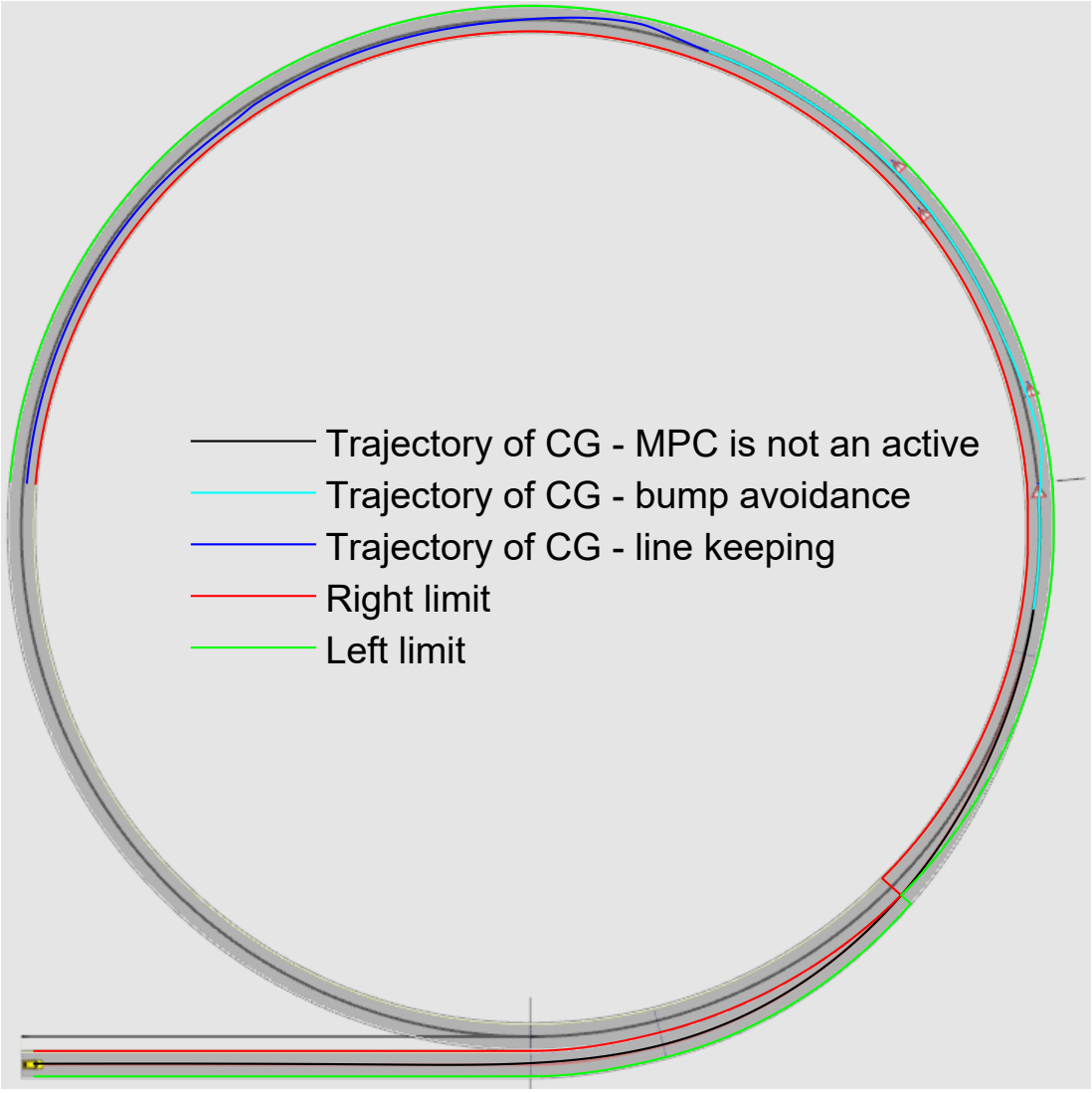


Figure 6.9: Vehicle's trajectory during experiment provided at 30 $\mathrm{km\,h^{-1}}$.

The MPC acted a bit aggressively (Fig. 6.7 and Fig. 6.8). However, it is a concept solution, and experiment results are acceptable because the objectives were to keep a vehicle within the predefined environmental envelope. The comfort of the driver and passenger must be an objective for future work. The trajectory of the test vehicle is presented in Fig. 6.9.

6.4.2 Test with Constant Speed at 50 $\mathrm{km\,h^{-1}}$

The second test [38] is provided at a speed of 50 $\mathrm{km\,h^{-1}}$. The designed control solution works pretty well, avoids obstacles, and keeps the vehicle within the defined borders despite the fact that steering wheel deviations are too aggressive (Fig. 6.10 and Fig. 6.11). In the first bump section can be seen that the controller chose to drive through a drivable obstacle to keep the vehicle in lane.

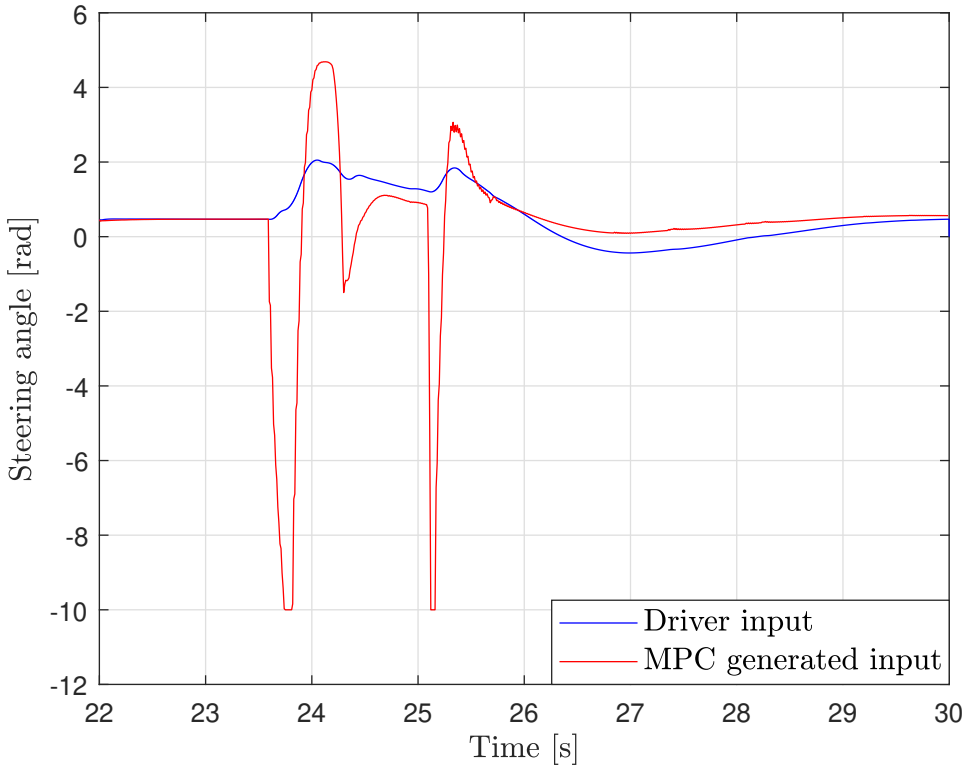


Figure 6.10: Steering wheel position during experiment provided at 50 km h^{-1}. Obstacle avoidance part.

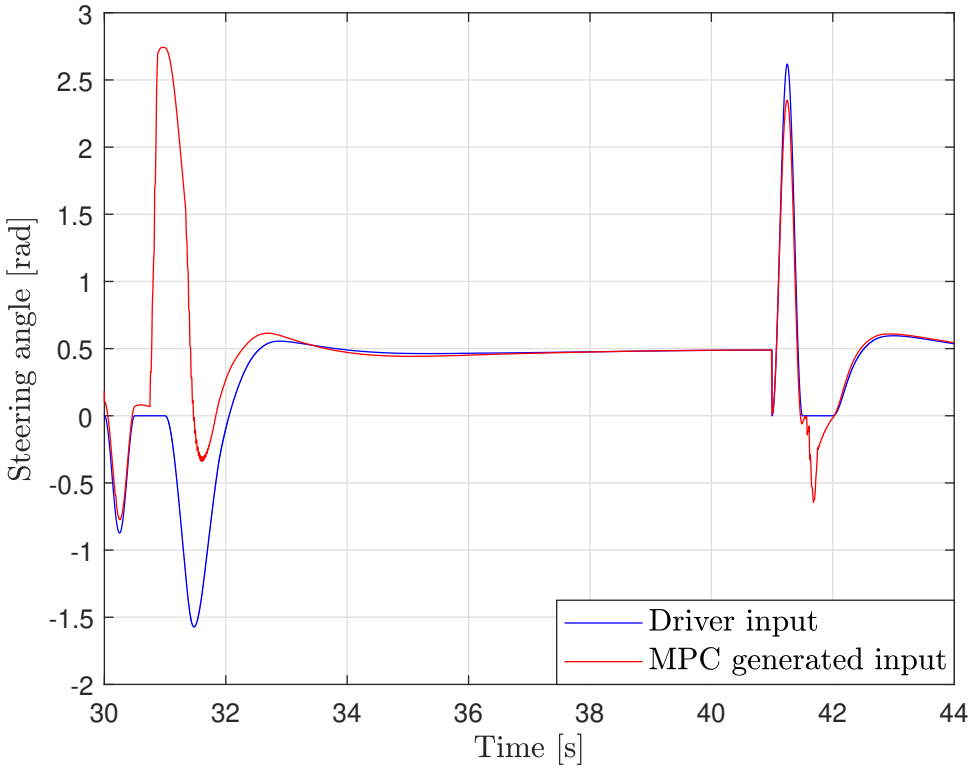


Figure 6.11: Steering wheel position during experiment provided at 50 km h^{-1}. Lane-keeping part.

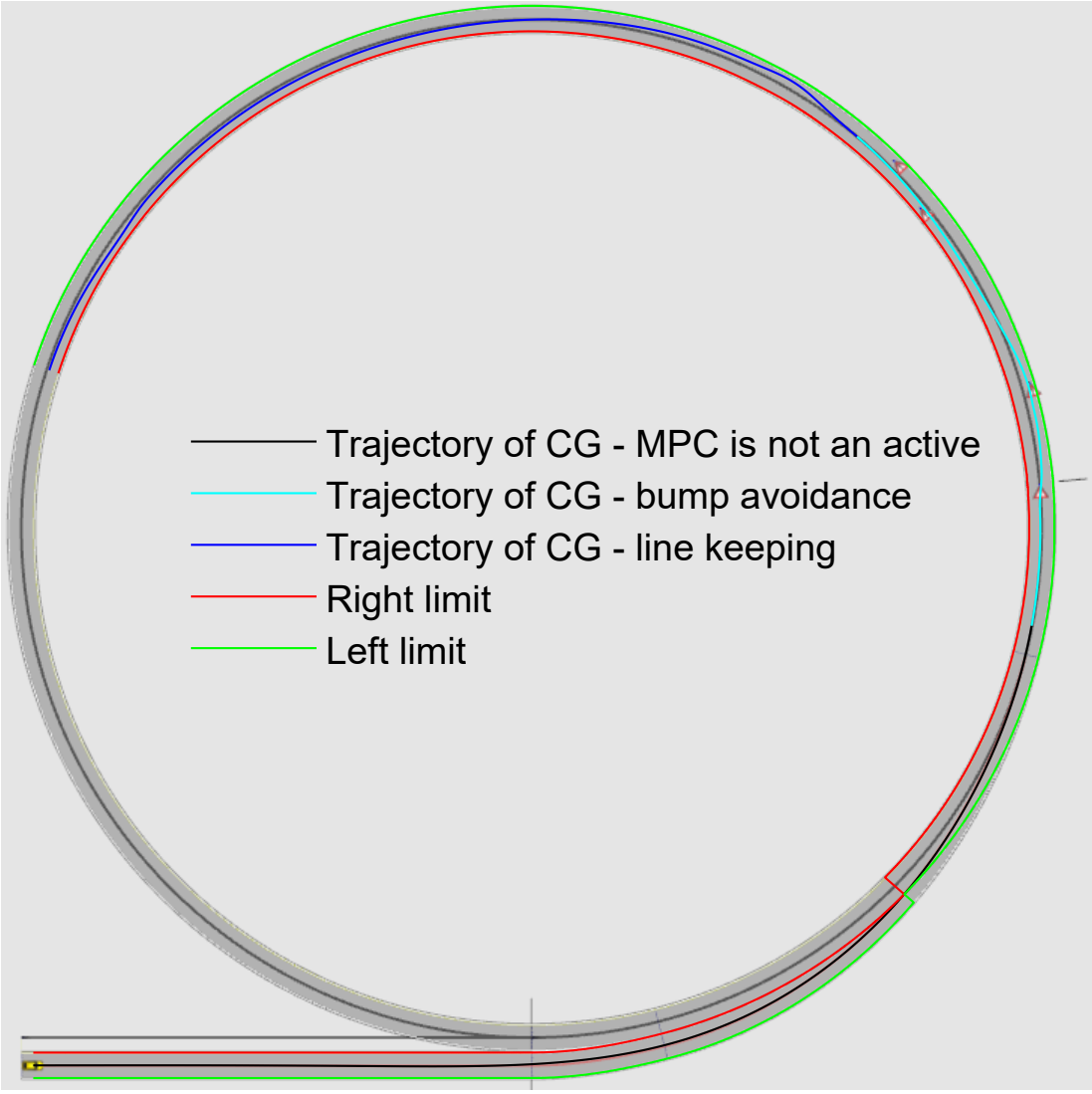


Figure 6.12: Vehicle's trajectory during experiment provided at 50 $\mathrm{km\,h^{-1}}$.

6.4.3 Test with Constant Speed at 72 $\mathrm{km\,h^{-1}}$

The third test [39] is provided on speed 72 $\mathrm{km\,h^{-1}}$. Controller shows excellent performance during test, avoided all obstacles, and keep a vehicle on lane.

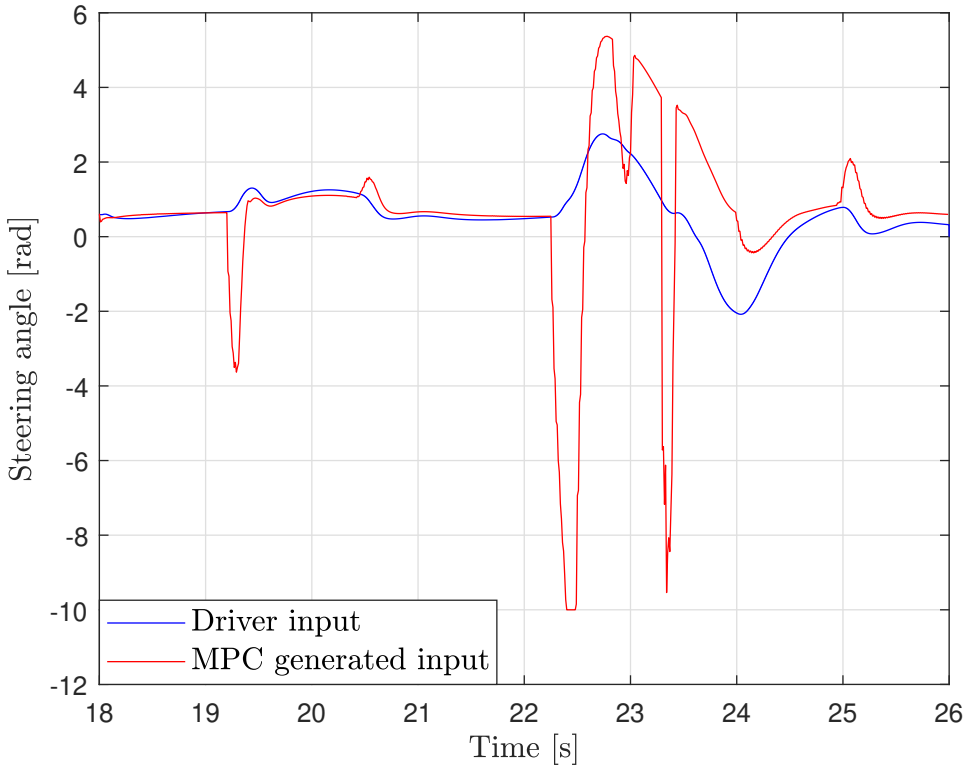


Figure 6.13: Steering wheel position during experiment provided at 72 km h^{-1}. Obstacle avoidance part.

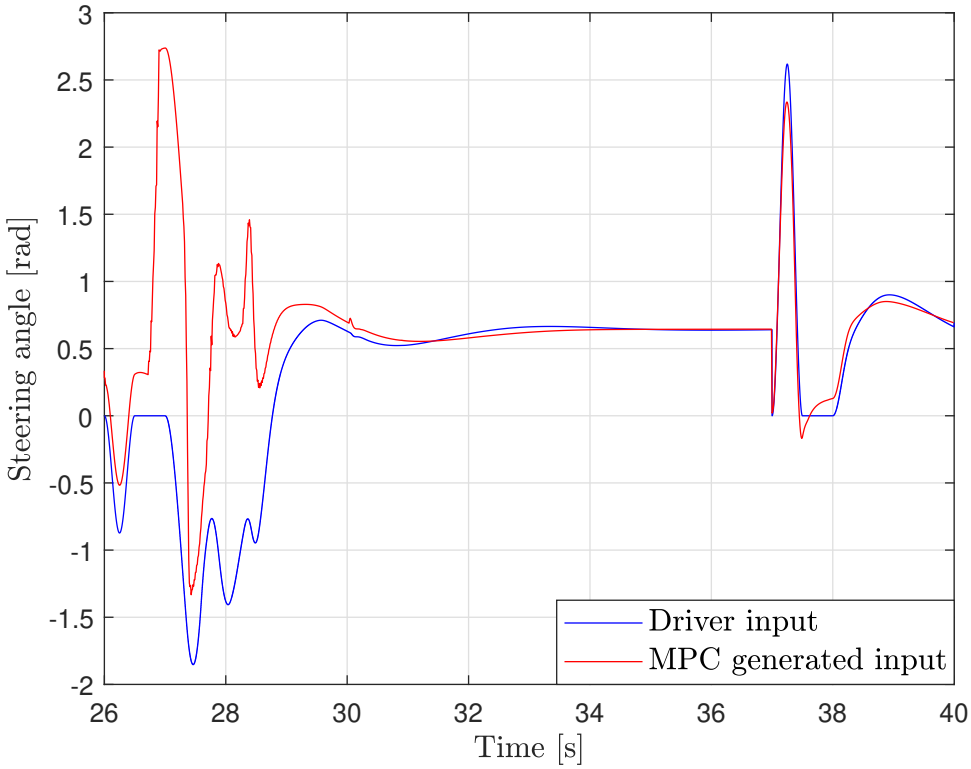


Figure 6.14: Steering wheel position during experiment provided at 72 km h^{-1}. Lane-keeping part.

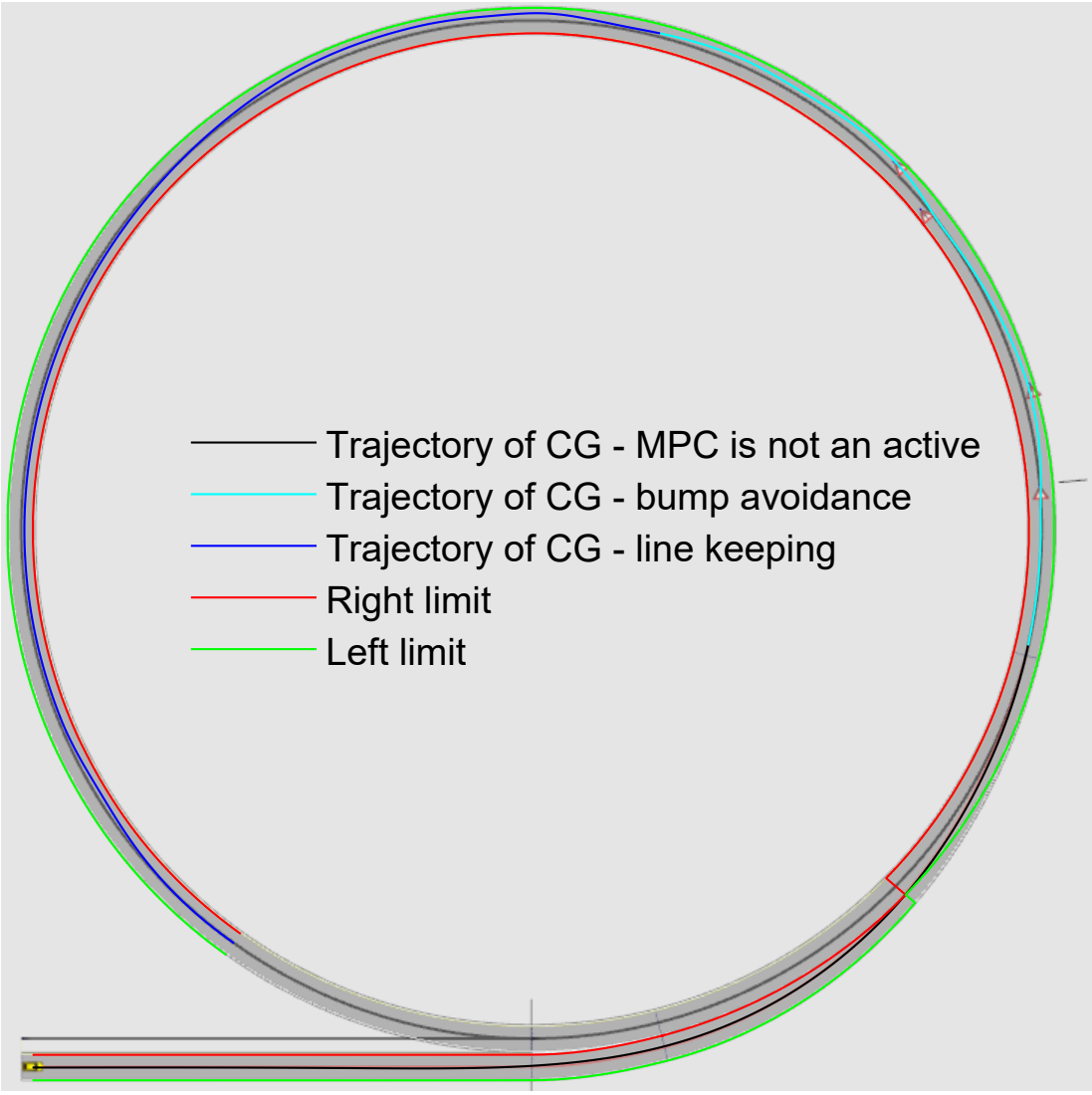


Figure 6.15: Vehicle's trajectory during experiment provided at 72 $\mathrm{km\,h^{-1}}$.

6.4.4 Test With Acceleration

This test [37] is very identical to the first test but on the final part of road the vehicle start accelerate (Fig. 6.19). This cause undefined reaction of MPC. Dynamic model (eq. 4.22a) has assumption that the vehicle move with constant velocity, which isn't true in this case. As result can be seen oscillation (Fig. 6.17) from 60 second to the end.

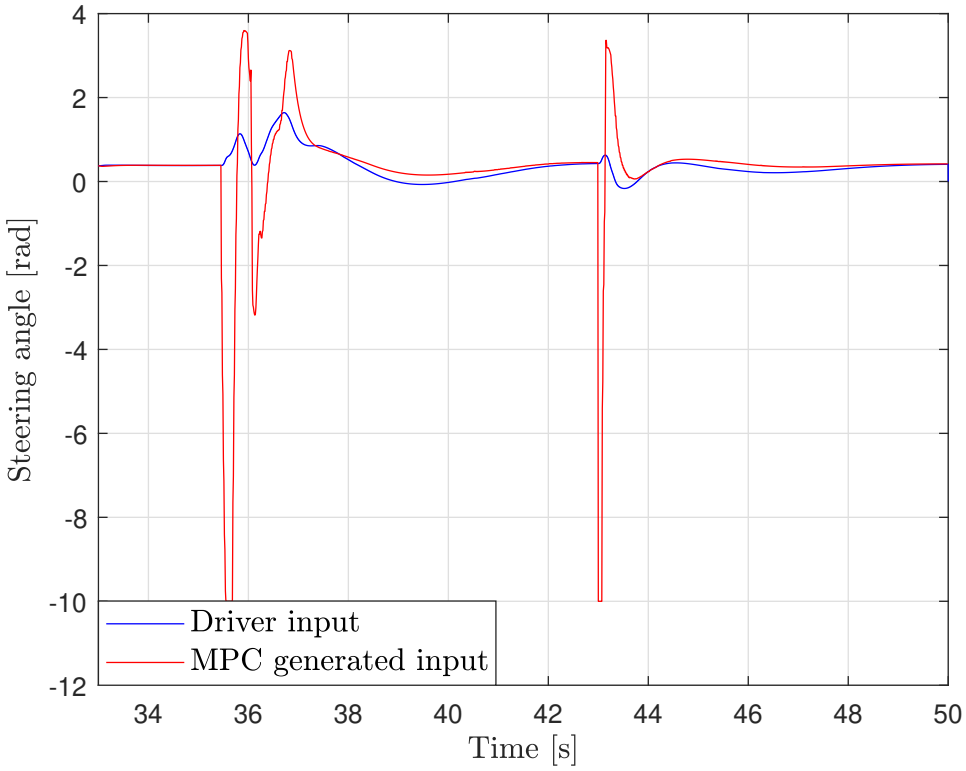


Figure 6.16: Steering wheel position during experiment provided at 30 $\mathrm{km\,h^{-1}}$ with acceleration on the end. Obstacle avoidance part.

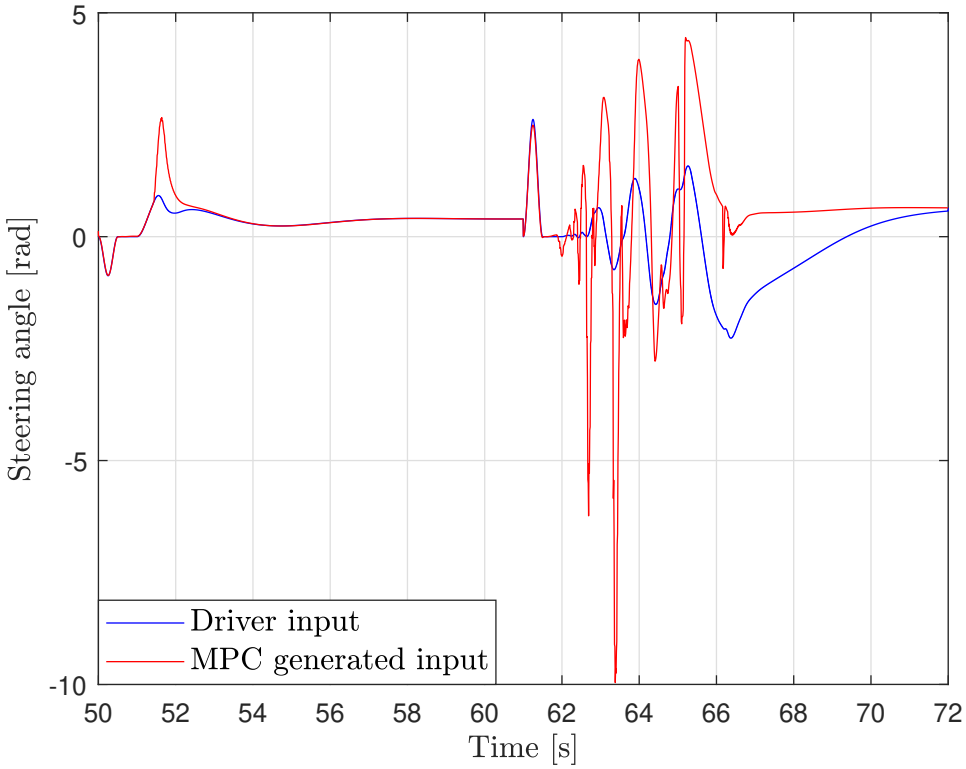


Figure 6.17: Steering wheel position during experiment provided at 30 $\mathrm{km\,h^{-1}}$ with acceleration on the end. Lane-keeping part.

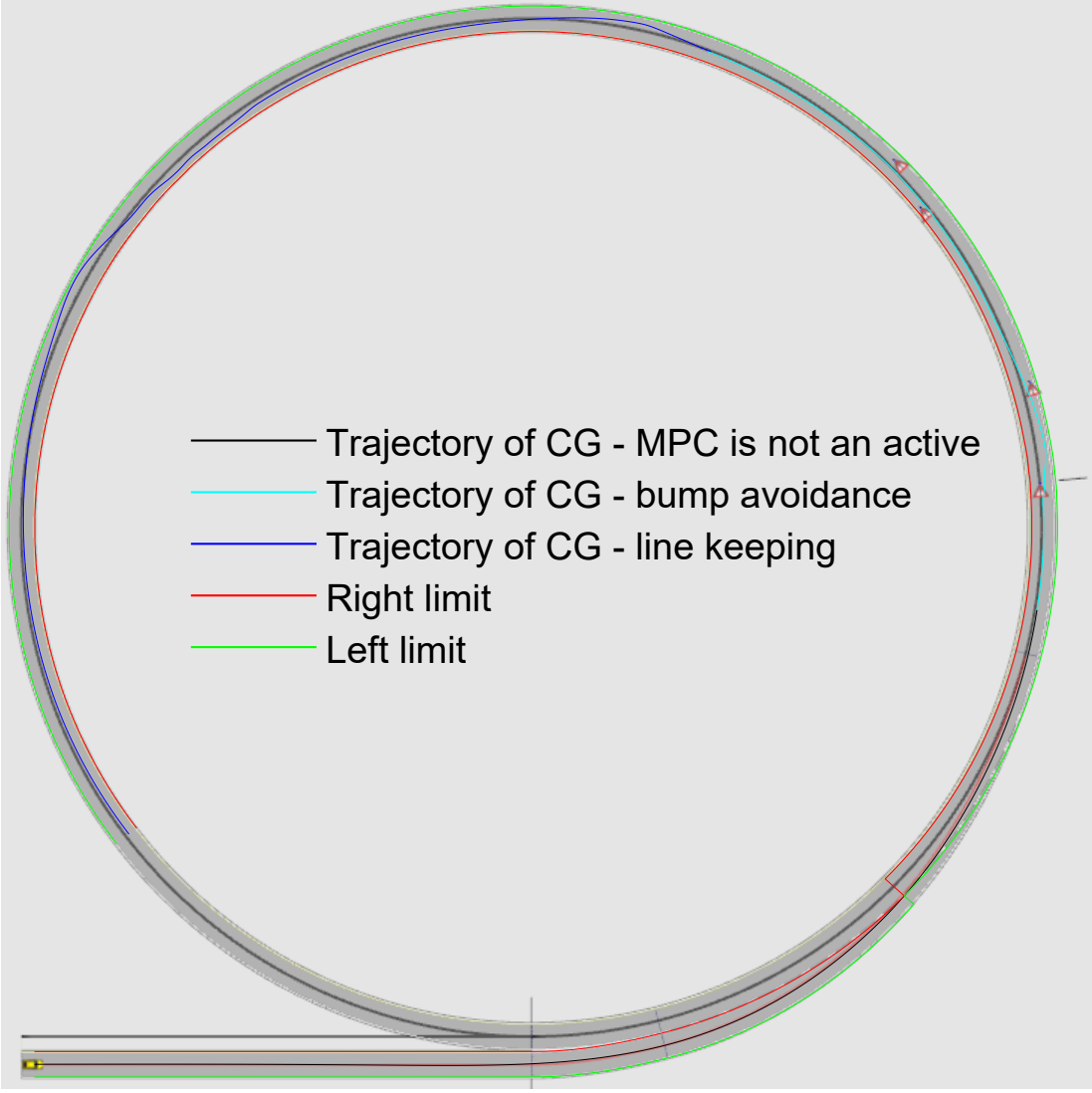


Figure 6.18: Vehicle's trajectory during experiment provided at 30 $\mathrm{km\,h^{-1}}$ with acceleration on the end.

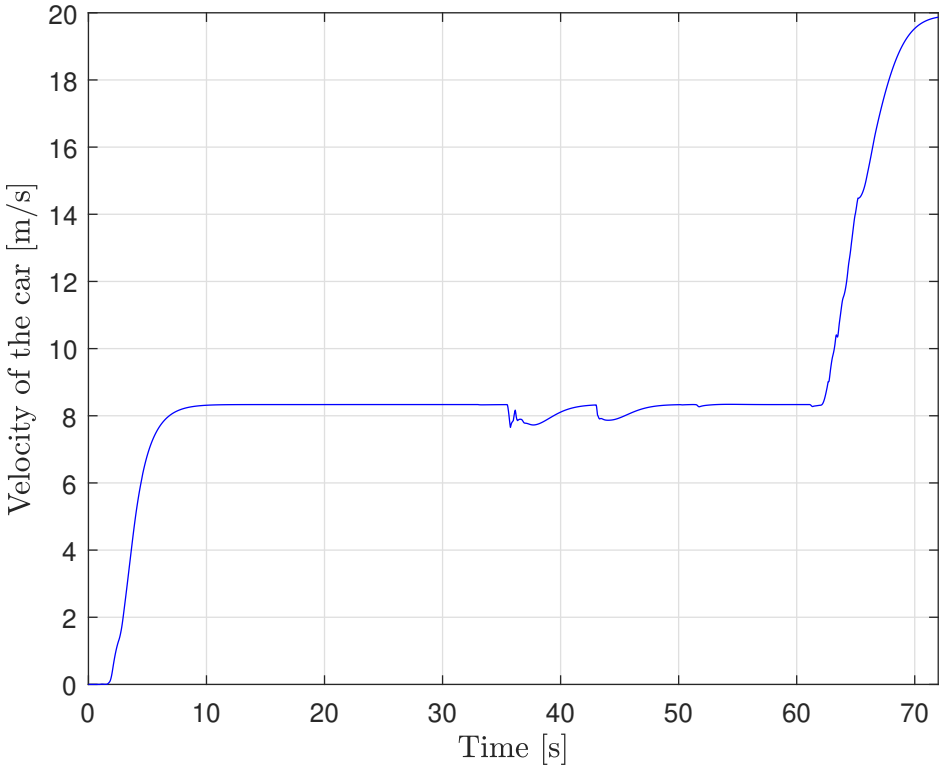


Figure 6.19: Velocity of the vehicle during forth experiment

6.4.5 Summary

Designed control shows good results on different speed. The vehicle avoids obstacle and stay on road without leaving. Some maneuvers performed by MPC seems a bit sharp and uncomfortable for human. This problem can be solved in futures work. The main problem of the designed control solution is poor time performance. The MPC requires a prediction horizon of near one or two seconds to show an adequate and satisfactory reaction to designed challenges. A lot of attempts were made to find a balance between control and time performance, but the result is the prolonged time of calculation compared to real-time (¡0.01 real-time).

7. Results

All of the goals, listed in chapter [**Objectives**], are reached. Namely:

- Familiarization with vehicle dynamics simulator IPG CarMaker, optimizing solver Gurobi, and YALMIP optimizing framework for Matlab was completed. In the result, Gurobi was replaced by Ipopt, and YALMIP was replaced by CasAdi.
- Existing solutions for lane-keeping and obstacle avoidance were studied and are described in [**State of the Art**] chapter.
- Existing single-track vehicle model was described and expanded to be suited for position control in the predictive control framework in chapter [**Single-Track Models**].
- Optimization problem that follows the driver's commands while keeping the vehicle is in the predefined lane and protect its boundaries violation was suggested in [**Control Architecture**].
- The controller from the previous point has been augmented to avoid fixed drivable obstacles. The resulting solution is presented in [**Control Architecture**].
- Simulation tests using IPG CarMaker and Matlab/Simulink testing environments are provided. Their results are discussed in chapter [**Experiments**].

8. Conclusion

This work describes the modern state of the art in driving automation in general and lane-keeping and obstacle avoidance in particular. In this thesis, the single-track model is derived and augmented for a novel lane-keeping and obstacle avoidance algorithm. The central thesis's outcome is the presentation of novel ADAS, which tracks the driver's commands closely as possible, and, at the same time, helps the driver to keep the vehicle in predefined environmental constraints, defined by lane borders and detected obstacles.

Provided automated simulation ride tests show promising results of the proposed solution. The resulting ADAS is a bit aggressive but could provide the functionality of a lane-keeping and obstacle avoidance system. It is possible to easily augment the proposed controller to semi-autonomously drive a vehicle in off-road conditions avoiding typical road irregularities.

Experiments showed that the implemented ADAS could work at a constant speed. However, it could fail when the test vehicle is accelerating. This problem could be addressed in future work.

The advanced Driver Assistance System presented in this work has Level 1 of automation. However, it can be augmented to Level 2 by adding angular velocity of the front axle as the second input and velocity of the vehicle as a next state. In this situation, ADAS could control steering and throttle/braking as interconnecting functions, which is the condition of the second level of automation. Also, this augmentation could solve the previously mentioned problem of failed ride tests with accelerating vehicles.

Bibliography

[1] On-Road Automated Driving (ORAD) committee, *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles*, 2021.

[2] "Automated vehicles for safety," 04.05.2022. [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[3] National Highway Traffic Safety Administration and others, "Preliminary statement of policy concerning automated vehicles," *Washington, DC*, vol. 1, p. 14, 2013.

[4] "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles - sae international," 04.05.2022. [Online]. Available: https://www.sae.org/standards/content/j3016_202104

[5] Italia, Volkswagen Group, "Tomorrow autonomous driving, today the traffic jam pilot," Jan 2019. [Online]. Available: https://modo.volkswagengroup.it/en/mobotics/tomorrow-autonomous-driving-today-the-traffic-jam-pilot

[6] J. M. Watts, "World's first self-driving taxis hit the road in singapore," *The Wall Street Journal*, 2016.

[7] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced driver-assistance systems: A path toward autonomous vehicles," *IEEE Consumer Electronics Magazine*, vol. 7, no. 5, pp. 18–25, 2018.

[8] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1856–1860.

[9] A. Mammeri, G. Lu, and A. Boukerche, "Design of lane keeping assist system for autonomous vehicles," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, 2015, pp. 1–5.

[10] C. Hu, Z. Wang, Y. Qin, Y. Huang, J. Wang, and R. Wang, "Lane keeping control of autonomous vehicles with prescribed performance considering the rollover prevention and input saturation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3091–3103, 2020.

[11] Z. Bing, C. Meschede, K. Huang, G. Chen, F. Rohrbein, M. Akl, and A. Knoll, "End to end learning of spiking neural network based on r-stdp for a lane keeping vehicle," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4725–4732.

[12] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous systems*, vol. 32, no. 1, pp. 1–16, 2000.

[13] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering using nonlinear model predictive control," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4772–4777.

[14] H. Ferenc, B. Tamás, A. Szilárd, and G. Péter, "Model based trajectory planning for highly automated road vehicles," in *IFAC World Congress: IFAC-PapersOnLine*, 2017.

[15] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 583–601, 2003.

[16] D. Ren, J. Zhang, J. Zhang, and S. Cui, "Trajectory planning and yaw rate tracking control for lane changing of intelligent vehicle on curved road," *Science China Technological Sciences*, vol. 54, no. 3, pp. 630–642, 2011.

[17] F. You, R. Zhang, G. Lie, H. Wang, H. Wen, and J. Xu, "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5932–5946, 2015.

[18] J. Wurts, J. L. Stein, and T. Ersal, "Minimum slip collision imminent steering in curved roads using nonlinear model predictive control," in *2019 American Control Conference (ACC)*, 2019, pp. 3975–3980.

[19] 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," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 378–383.

[20] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annual Reviews in Control*, vol. 45, pp. 76–86, 2018.

[21] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical systems and signal processing*, vol. 100, pp. 482–500, 2018.

[22] D. Schramm, M. Hiller, and R. Bardini, *Vehicle dynamics, Modeling and Simulation.* Springer, 2014.

[23] D. Efremov, Y. Zhyliaiev, B. Kashel, and T. Haniš, "Lateral driving envelope protection using cascade control," in *2021 21st International Conference on Control, Automation and Systems (ICCAS).* IEEE, 2021, pp. 1440–1446.

[24] D. Efremov, "Single-track vehicle model," Available online: https://github.com/SDS-RC-FEE-CTU-in-Prague/SingleTrack, 2020.

[25] D. Efremov, T. Haniš, and M. Klaučo, "Driving envelope: On vehicle stability through tire capacities." *Accepted and presented at 2022 IEEE Intelligent Vehicles Symposium (IV).*

[26] Y. Jo and S. Ryu, "Pothole detection system using a black-box camera," *Sensors (Basel, Switzerland)*, vol. 15, pp. 29 316 – 29 331, 2015.

[27] S.-S. Park, V.-T. Tran, and D.-E. Lee, "Application of various yolo models for computer vision-based real-time pothole detection," *Applied Sciences*, vol. 11, no. 23, p. 11229, 2021.

[28] A. Akagic, E. Buza, and S. Omanovic, "Pothole detection: An efficient vision based method using rgb color space image segmentation," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* IEEE, 2017, pp. 1104–1109.

[29] E. Edmonds, "More americans willing to ride in fully self-driving cars," Nov 2020. [Online]. Available: https://newsroom.aaa.com/2018/01/americans-willing-ride-fully-self-driving-cars/

[30] "IPG CarMaker." [Online]. Available: https://ipg-automotive.com/en/products-solutions/software/carmaker/

[31] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[32] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022. [Online]. Available: https://www.gurobi.com

[33] "Toyota research on automated cars in europe." [Online]. Available: https://www.trace-lab.com/

[34] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[35] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[36] D. Efremov, "Unstable ground vehicles and artificial stability systems," *Masterâ€™s thesis*, 2018.

[37] B. Kashel, "Protection of driving corridor with respect of various road conditions test on 30 km/h," May 2022. [Online]. Available: https://youtu.be/mE-3jVmSLMw

[38] ——, "Protection of driving corridor with respect of various road conditions test on 50 km/h," May 2022. [Online]. Available: https://youtu.be/rb0tQG8Z0j8

[39] ——, “Protection of driving corridor with respect of various road conditions” test on 72 km/h,” May 2022. [Online]. Available: https://youtu.be/NQESdp6Sq4E

List of Figures

List of Tables