UNIVERSITÀ DEGLI STUDI DI PADOVA

SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATION ENGINEERING
SECOND CYCLE DEGREE IN AUTOMATION ENGINEERING

MASTER THESIS

Local Path Planning with Moving Obstacle Avoidance based on Adaptive MPC in ATLASCAR2

Supervisor: Prof. ANGELO CENEDESE

Co-Supervisor: Prof. VITOR SANTOS

Master Candidate: ALBERTO FRANCO

Registration Number 1156523

15$^{\text{th}}$ April 2019

ACADEMIC YEAR 2018/2019

Povera mente,
io ti uccido ogni giorno con le mie idee.
Povero cuore,
io ti metto alla prova ma povero me.

Ringraziamenti/Acknowledgements

dedica a parenti e amici

The research work for this thesis was carried out at the Laboratory for Automation and Robotics (LAR) in the Department of Mechanical Engineering of Aveiro (Portugal), during a period of 5 months as an exchange student.

I would like to express my gratitude to Professor Vitor Santos, for welcoming me at the Laboratory for Automation and Robotics and making possible my experience of study at the University of Aveiro. His help has been invaluable both personally and academically. Moreover I would like to thank Professor Angelo Cenedese for his advices and his help during this work.

Abstract

Inserted in the ATLASCAR2 project this work aims to develop a short-term path planning algorithm for driver assistance in dynamic environments. In order to achieve this objective, it was made a preliminary study of the existing local path planning methods and the projects that have already been developed in this field, their advantages and disadvantages were weighed and the most successful approaches applied to local navigation in real autonomous driving projects were taken into account. This thesis presents two different strategies for a self-driving car short-term path planning among multiple moving obstacles. The main task is to study and implement a motion planning and execution framework in order to make ATLASCAR2 coexist with other moving obstacle vehicles by avoiding collision and overtake them when necessary and possible. The first method developted, is an obstacle avoidance system that moves the vehicle around different moving obstacles while the second algorithm is a lane following system that keeps the ATLASCAR2 traveling along the centerline of the lanes on the road. The proposed techniques, based on the adaptive Model Predictive Control paradigm, solve optimization problems formulated in terms of cost minimization under constraints. Simulation results, developted in a MATLAB/Simulink enviroment, demonstrate and verify the feasibility and the usefulness of methods considering different scenarios, opening space for real scenario implementation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In robotic research, the problem of navigation is among the most important. Basically, all autonomous mobile robots need some kind of navigation abilities to perform, localization, motion planning and guidance [6]. In the present context, we focus on navigation as a process of planning a path of a mobile robot from its current position to a desired goal location, following the planned path, and avoiding any discovered obstacles along the way. The desired paths have to fulfill several conditions to ensure safety and feasibility of the navigation. Moreover, the paths can be also defined in terms of specifications; for example, short or smooth paths are usually more desirable than long and curved ones in every dynamic environments.

Beyond the path planning, the navigation problem also involves reacting to changes of the environment model. Robots are required to move towards the target in a short time and avoid either static or dynamic obstacles observed by their sensors, which involves efficient path planning and valid obstacle avoidance.

Research and development of Unmanned Ground Vehicles has been dominated by DARPA (Defense Advanced Research Projects Agency) and NASA (National Aeronautics and Space Administration). The DARPA initiative started with the development of the first mobile robot, Shakey, and also includes the Autonomous Land Vehicle and the DARPA Demo I1 Program. NASA sponsors the development of unmanned vehicles for planetary surface exploration, from the Jet Propulsion Laboratory Mars Rover to the most recent Mars Pathfinder. Recent UGV design and development has been enhanced to build UGVs capable of operating in Intelligent Vehicle Highway Systems [7].

Over the last decades, the development of Advanced Driver Assistance Systems has become a critical endeavor to attain different objectives: safety enhancement, mobility improvement, energy optimization and comfort [8]. Much of the argument used in this discussion is based on the road mortality we see today. According to data from the World Health Organization, in 2013 there were about 1.25 million road deaths worldwide, and this number is expected to increase in the next decade [9]. Algorithms for autonomous navigation are increasingly robust and reliable and are starting to handle complex situations and decision problems.

According to Katrakazas [10], local navigation is responsible for guiding the vehicle, that is, for the planning of the direction and speed to be taken, in a space close to the current position based on information exclusively obtained by the sensors on board. The guiding must be planned in such a way as to guarantee the displacement, from the current state to the objective, without collisions.

The algorithms we have developed at the LAR, follow a new and different approach for an advanced control strategy for autonomous navigation. The idea is that these methods do not replace the algorithms developed previously but are a valid alternative, so that depending on the situation, the vehicle can choose the best strategy to overcome obstacles or solve problems that can occur on the road that need a decision in real time. Simulation results demonstrate and verify the feasibility and the usefulness of methods considering different scenarios.

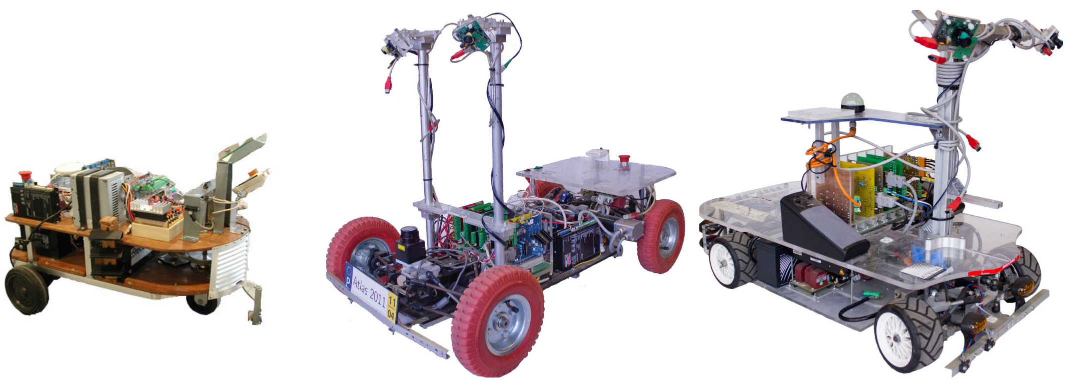
In this introductory chapter, the ATLAS project is presented in more detail in section 1.1, while examples of autonomous navigation projects are discussed later (section 1.2). The context of the problem and the proposed solution of this thesis are carried out in section 1.3 while the organization of the document is discussed in section 1.4.

1.1 ATLAS Project

The ATLAS project was created in 2003 by the Group for Automation and Robotics from the Department of Mechanical Engineering at the University of Aveiro [11]. The objective of this project is to study and develop advanced sensors and active systems to promote the autonomous control of cars and other platforms. The first projects in the autonomous driving area focused on small scale models in controlled environments for participation in the National Robotics Festival (FNR) and in many other competitions winning some awards for the best performance (subsection 1.1.1). The success and experience gained with these models allowed the evolution of the project for full-scale vehicles, where ATLASCAR (subsection 1.1.2) in 2010 and ATLASCAR2 (subsection 1.1.3) in 2016 have been developed.

1.1.1 ATLAS platforms

The first developed robot (Figure 1.1a) was based on an aluminum and wood structure. In this prototype only one camera was installed that pointed to a mirror to allow the complete visualization of the road in which the robot circulated. The traction movement was assured by a mechanical differential coupled to the rear wheels and the steering movement was given by a single front wheel. In order to create a model more similar to an ordinary car, the ATLAS group developed the ATLAS 2000 (Figure 1.1b) in scale (1:4), with which it managed to win the first autonomous driving competition of the FNR in 2006. After several improvements made in ATLAS 2000, in 2008 a new platform, ATLAS MV (Figure 1.1c) was created. This robot was designed on a smaller scale (1:5), with the intention of being lighter and faster. On board were installed a new steering system, hydraulic braking and an active perception unit. This robot allowed the conquest of new victories in the autonomus driving competitions.



(a) First ATLAS prototype. (b) ATLAS 2000. (c) ATLAS MV.

Figure 1.1: Some of the ATLAS project small scale platforms (adapted from [1]).

1.1.2 ATLASCAR

Driven by the positive results achieved with scale models and years of navigation experience in controlled environments, in 2010 the Group of Automation and Robotics decided to invest in a large-scale project, ATLASCAR (Figure 1.2). The vehicle used for this project was a Ford Escort Station Wagon of 1998 powered by a gasoline internal combustion engine, in which several sensors, processing units and actuators were installed. On-board sensors processed data collected from the vehicle and its surroundings, with different LIDARs for obstacle detection and environmental reconstruction, pedestrian detection cameras and a Global Navigation Satellite System (GNSS) for location and route planning. After passing through the processing units, these data were sent to the actuators that allowed the movement and execution of the maneuvers in a completely autonomous way on the part of the vehicle. To power all the equipment, a Uninterruptible Power Supply (UPS) was used, loaded from an auxiliary alternator. During this project, many works were developed in the Laboratory for Automation and Robotics, many of which produced master thesis. For example, in 2014 Cabral de Azevedo [12] developed a module to detect pedestrians using sensory fusion of LIDAR and vision data while in 2016 Vieira da Silva [13] created a multisensory calibration module that was exported to subsequent projects.

Figure 1.2: The car used in ATLASCAR, based on Ford Escort Station Wagon of 1998 (adapted from [1]).

1.1.3 ATLASCAR2

Given the different limits, to continue the project, in 2016, a new vehicle was acquired: ATLASCAR2 (Figure 1.3). This time it was chosen as a platform an electric car, a Mitsubishi iMiEV, with an autonomy range of 100 km. The fact that the vehicle is electric allows to use the energy stored in the batteries, making it easier to power the sensors installed. In fact, despite the short time of existence, 3 LIDARs, a camera, inclinometry sensors and a GNSS unit are already installed on the ATLASCAR2. Many of these sensors were transferred from ATLASCAR to this project during the work of Madureira Correia [14] in 2017, where a module for detecting free space around the car was also developed while in 2018 Ricardo Silva [2] created a local navigation module for driver assistance in the immediate decision making, identifing a solution based on a multiple hypothesis approach.

Figure 1.3: The vehicle used in ATLASCAR2, based on an electric car, a Mitsubishi iMiEV of 2015 (adapted from [2]).

1.2 Autonomous Cars

The legal definition of autonomous vehicle in the District of Columbia code is:

> *"Autonomous vehicle" means a vehicle capable of navigating District roadways and interpreting traffic-control devices without a driver actively operating any of the vehicle's control systems. The term "autonomous vehicle" excludes a motor vehicle enabled with active safety systems or driver-assistance systems, including systems to provide electronic blind-spot assistance, crash avoidance, emergency braking, parking assistance, adaptive cruise control (ACC), lane-keep assistance (LKA), lane-departure warning, or traffic-jam and queuing assistance, unless the system alone or in combination with other systems enables the vehicle on which the technology is installed to drive without active control or monitoring by a human operator.*

The modern automobile companies keep coming up with newer autonomous features in their recent models. Technological advancements seen every day in areas like information technology, communication, data analysis and storage etc. is not exclusive to these areas alone. The realm of autonomous cars is also progressing at a rapid rate these days [15]. Google's development of self-driving technology began in January 2009. The initial objective of the project was to develop a car able to navigate on highways with minimal human intervention. In December 2016, the unit was renamed Waymo; this name derived from its mission, "a new way forward in mobility". Waymo moved to further test its cars on public roads after becoming its own subsidiary.

(a) Google's Firefly self-driving prototype in 2015 (adapted from [16]).

(b) Waymo Chrysler Pacifica Hybrid self-driving prototype in 2017 (adapted from [16]).

Figure 1.4: Some of the Waymo/Google prototypes tested across multiple locations in the Unided States in recent years.

Waymo uses LIDAR which sends out millions of laser beams per second to build up a detailed picture of the world all 360 degrees around it. It also uses radar to detect how far away objects are and their speed and high-resolution cameras detect visual information like whether a traffic signal is red or green. It then combines all that data to understand the world around it and predict what those things might do next. It can do that for things up to three football fields away. Based on all this information, Waymo's software determines the exact trajectory, speed, lane and steering maneuvers needed to progress along this route safely [16] [17].

With the advances in autonomous technology, VIAC or VisLab Intercontinental Autonomous Challenge was one of the major competitions which led to improvements in the testing and analysis of autonomous vehicles and robotics. It was a 13,000 kilometers trip, nearly three months from Parma, Italy to Shanghai, China from July 20, 2010 to October 28, 2010. It involved four autonomous vehicles with negligible human intervention and high level of autonomy [18]. One of VisLab's advanced autonomous car, BRAiVE (Figure 1.5), drove in downtown Parma on July 12th, 2013. It successfully navigated narrow rural roads, crosswalks, traffic lights, pedestrian areas, roundabouts and artificial hazards. It was a pioneer in the field of vehicular robotics, since it was totally autonomous [3].

Figure 1.5: BRAiVE prototype developted by VisLab, based on a Hyundai Sonata (adapted from [3]).

Another example of autonomous system is Navya, a robotically driven electric shuttle which operates at a maximum speed of 25 kilometers per hour. Made by Induct Technology, France, it can accommodate 15 passengers. It uses four LIDAR units and stereoscopic optical cameras, and it does not require any road modifications. Its LIDAR unit and optical cameras help in generating a real-time three dimensional map of the surroundings. It is being successfully tested at various universities across Switzerland, England and Singapore [19].

Figure 1.6: Navya Shuttle developted in 2016 by Navya Group in France.

1.3 Context of the Problem and Proposed Approach

Dynamic environments pose several added difficulties to the motion planning problem. The dynamics of the ATLASCAR2 must be taken into account, and there are limitations due to the sensors range and uncertainty in measurements, that must be reflected on the motion plan. Besides it, a motion plan must incorporate time restrictions, meaning the vehicle will require a certain amount of time to accomplish a task. For example, when crossing a road, the ATLASCAR2 must do it fast enough to avoid incoming cars. The proposed algorithms were studied for the ATLASCAR2 project in which the group for Robotics and Automation at the University of Aveiro has setup and adapted a common commercial electric vehicle to provide a versatile framework to develop studies and research [11] [20]. The fact that the vehicle is electric allows to use the energy stored in the batteries, making it easier to power the sensors installed. In fact, the ATLACAR2 is equipped with sensors, such as lidar, that measure the distance to obstacles in front and around the vehicle. The obstacles can be static, such as a large pothole, or moving, such as a moving vehicle on the same or a nearby lane. The most common maneuver from the driver is to temporarily move to another lane, drive past the obstacle, and move back to the original lane afterward. In this case, we want to design an obstacle avoidance system that moves the ATLASCAR2 around a moving obstacle in the lane using throttle and steering angle. This system uses an adaptive Model Predictive Controller that updates both the predictive model and the mixed input/output constraints at each control interval. Moreover we want to develop a lane-keeping assist system for the vehicle: it has a sensor, such as camera or laser, that measures the lateral deviation and relative yaw angle between the centerline of a lane and the ATLASCAR2; it also measures the current lane curvature and its derivative. Depending on the curve length that the sensor can view, the curvature in front of the vehicle can be calculated from the current curvature and its derivative. This system keeps the autonomous car travelling along the centerline of the lanes on the road by adjusting the front steering angle. The goal for lane keeping control is to drive both lateral deviation and relative yaw angle close to zero.

1.4 Thesis Outline

In this section, we outline the thesis organization:

Chapter 1 is used to introduce the thesis focus areas of autonomous vehicle technology. In particular the ATLAS project and examples of autonomous navigation projects are presented. Moreover the context of the problem and the objectives to be achieved are carried out.

Chapter 2

Chapter 2

Literature Review

[21] [11] [22] [6]

Chapter 3

Model Predictive Control

In this chapter, the theory of Model Predictive Control is discussed in detail to highlight working principle. In particular for this work we used an advanced control strategy based on this paradigm called Adaptive MPC that uses a fixed model structure, but allows the model parameters to evolve with the time.

3.1 Generic Model Predictive Control problem

Model Predictive Control (MPC), also known as Moving Horizon Control (MHC) or Receding Horizon Control (RHC), is a popular method for the control of slow dynamical systems, to generate the required control inputs that are calculated at each sampling instance k, using the current state as initial conditions to solve a finite optimal control problem. Some of the advantages of using MPC are:

- the ability to handle unstable, time variable, non-minimum phase systems;
- robustness feature with the uncertainties in the nonlinear systems;
- built in feed-forward control to handle disturbances in the processes;
- enhanced tuning features to achieve the best response including transient responses;
- the possibility to introduce constraints in a natural form;
- if the references are known in advance, they can be used in order to optimize the reference tracking.

The methodology of all the controllers belonging to the MPC family is characterized by the following strategy, represented in Figure 3.1. The future outputs for a determined horizon, called the prediction horizon, are predicted at each instant k using the process model. These predicted outputs depend on the known values up to instant k (past inputs and outputs) and on the future control signals which are those to be sent to the system and calculated. The set of future control signals is calculated by optimizing a determined criterion to keep the process as close as possible to the reference trajectory. This criterion usually takes the form of a quadratic function of the errors between the predicted output signal and the predicted reference trajectory. The control effort is included in the objective function in most cases. An explicit solution can be obtained if the criterion is quadratic, the model is linear, and there are no constraints; otherwise an iterative optimization method has to be used. Some assumptions about the structure of the future control law are also made in some cases, such as that it will be constant from a given instant. Only the current control signal is send to the process. At the next sampling instant the measured output is evaluated and the sequence is repeated and all the steps brought

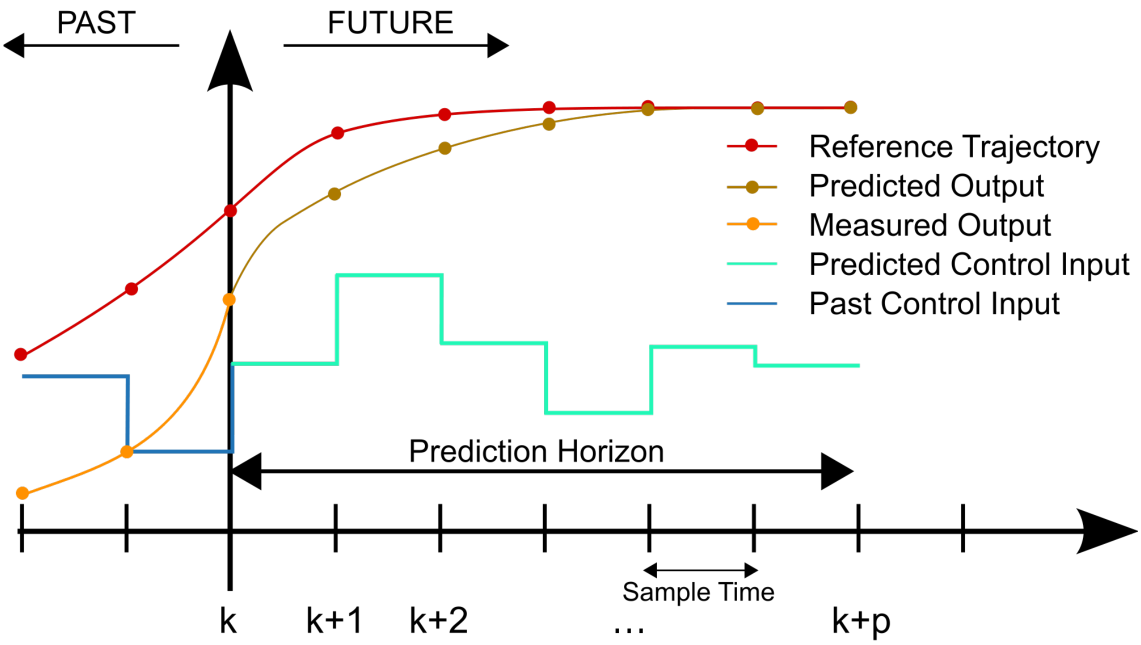


Figure 3.1: A discrete Model Predictive Control scheme adapted from [4].

up to date. Thus the predicted control input is then calculated using the receding horizon concept.

MPC is typically formulated in the state space. For a given discrete linear time-invariant (LTI) system:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \tag{3.1}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$, $\boldsymbol{u}(k) \in \mathbb{R}^m$ are the state and the input, respectively. The central idea in the Model Predictive Control is to minimize some cost function, while still ensuring that some constraints are fulfilled. The generic MPC problem can be written as follows:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & J(\boldsymbol{x}(k), \mathbf{u}) \\ \text{subject to} \quad & \boldsymbol{x}_{k+i+1} = \boldsymbol{A}\boldsymbol{x}_{k+i} + \boldsymbol{B}\boldsymbol{u}_{k+i} \quad \forall i = 0, \ldots N-1; \\ & \boldsymbol{x}_{k+i} \in \mathbb{X} \quad \forall i = 0, \ldots N-1; \\ & \boldsymbol{u}_{k+i} \in \mathbb{U} \quad \forall i = 0, \ldots N-1; \\ & \boldsymbol{x}_{k+N} \in \mathbb{X}_f; \quad \boldsymbol{x}_k = \boldsymbol{x}(k). \end{aligned} \tag{3.2}$$

where $\mathbf{u} = (\boldsymbol{u}_k, \ldots, \boldsymbol{u}_{k+N-1})$ is a sequence of control inputs, $\boldsymbol{x}_{k+i}$ is the state at time $k+i$ as predicted at time k, and N is the prediction horizon. The sets $\mathbb{X} \in \mathbb{R}^n$ and $\mathbb{U} \in \mathbb{R}^m$ define the constraints on the state and the input, respectively. Finally, the set $\mathbb{X}_f \subseteq \mathbb{X}$ defines the terminal constraint on the state. If we consider a regulation problem, the system 3.1 should be steered to the origin and the cost function $J(\boldsymbol{x}(k), \mathbf{u})$ could be in a quadratic form as follows:

$$J(\boldsymbol{x}(k), \mathbf{u}) = \boldsymbol{x}_{k+N}^{\mathsf{T}} \boldsymbol{P}_f \boldsymbol{x}_{k+N} + \sum_{i=1}^{N} \boldsymbol{x}_{k+i}^{\mathsf{T}} \boldsymbol{Q} \boldsymbol{x}_{k+i} + \boldsymbol{u}_{k+i}^{\mathsf{T}} \boldsymbol{R} \boldsymbol{u}_{k+i} \tag{3.3}$$

where $\boldsymbol{P}_f$, $\boldsymbol{Q} \geq 0$ (positive semi-definite) and $\boldsymbol{R} > 0$ (positive definite) are weighting matrices.

Instead if we consider a servo problem, like tracking of a reference signal, the cost function is changed as follows:

$$
\begin{aligned}
J(\boldsymbol{x}(k), \mathbf{u}) = & (\boldsymbol{x}_{k+N} - \boldsymbol{x}^{\text{ref}}_{k+N})^\intercal \boldsymbol{P}_f (\boldsymbol{x}_{k+N} - \boldsymbol{x}^{\text{ref}}_{k+N}) \\
& + \sum_{i=1}^{N} (\boldsymbol{x}_{k+i} - \boldsymbol{x}^{\text{ref}}_{k+i})^\intercal \boldsymbol{Q} (\boldsymbol{x}_{k+i} - \boldsymbol{x}^{\text{ref}}_{k+i}) + \boldsymbol{u}^\intercal_{k+i} \boldsymbol{R} \boldsymbol{u}_{k+i}
\end{aligned}
\tag{3.4}
$$

where $\boldsymbol{x}^{\text{ref}}_{k+i}$, $\boldsymbol{x}^{\text{ref}}_{k+N}$ describe the reference trajectory. The standard MPC algorithm can be summarized by the following steps:

Algorithm 1 Basic Model Predictive Control loop

1: Measure the current state $\boldsymbol{x}(k)$;
2: Solve the optimization problem 3.2 with $\boldsymbol{x}(k)$ as initial state, where $\boldsymbol{u}(k)$ is calculated;
3: Apply the first control of the optimal control sequence;
4: Wait one sampling time and repeat steps 1-3;

An MPC has many strengths. Given that the model is discrete and linear it handles multivariable problems very well. Also mathematical convexity is an important part of the resulting problem formulation of an MPC. In fact there exists efficient solvers for convex optimization problems but it is therefore desirable that the MPC problem 3.2 is convex which is ensured if:

1. the cost function is convex;
2. the prediction model is linear;
3. the constraint sets $\mathbb{X}, \mathbb{U}$ are convex.

The optimization handles actuator constraints and state constraints naturally in the optimization which allows for the process to be operated much closer to the hard constraints, which improves control performance and efficiency. Because of its predictive nature it is able to solve a variety of problems and handle disturbances smoothly.

3.1.1 Tuning Parameters

The two most important parameters to tune in order to satisfy the control objectives are the diagonal matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ that can be used to weight the system state matrix and the control inputs respectively. The response of the system that is too slow can be influenced by adding high weighting values in the $\boldsymbol{Q}$ matrix, whereas the control gains are damped with high weighing values in the $\boldsymbol{R}$ matrix. Find an optimal trade-off is a fundamental aspect for the constroller behaviour.

3.1.2 Stability of MPC controller

A limited horizon on the MPC problem affects the stability of the controllers; in order to avoid this problem it is possible to set an infinite horizon, impose end point constraints, terminal cost function or use other techniques. To obtain a stable controller, the parameters to tune are: the terminal cost, prediction horizon and constraints. Also the weights on the cost function can be tuned to ensure a stabilizing solution.

3.1.3 Robustness

If the stability can be guaranteed and the performance specifications are met with respect to a certain set of uncertainties, the system is said to be robust; in particular a controller with this property has to ensure that the constraints are never violated for any admissible disturbance realization. The uncertainties in a system are due to external disturbances, measurement noise, inaccurate values of the model parameters, non-linearities etc... The most common type of uncertainties considered in the literature is additive disturbance because usually the current state of the system can be measured hence there is no noise in the measurements.

3.2 Adaptive Model Predictive Control

We understood that Model Predictive Control is an advanced method that predicts future behavior using a linear-time-invariant (LTI) dynamic model. These predictions are not exact and a good strategy is to make MPC insensitive to prediction errors. If the plant is strongly nonlinear or its characteristics vary dramatically with time, MPC performance might become unacceptable because LTI prediction accuracy degrade [4]. A method that can address this degradation by adapting the prediction model for changing operating conditions is called Adaptive MPC: this control strategy uses a fixed model structure, but allows the model parameters to evolve with time. Ideally, whenever the controller requires a prediction, it uses a model appropriate for the current conditions. At each control interval, the adaptive MPC controller updates the plant model and nominal conditions. Once updated, the model and conditions remain constant over the prediction horizon. The plant model used as the basis for the adaptive MPC must be an LTI discrete-time, state-space model with a structure as follows:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}_u\boldsymbol{u}(k) + \boldsymbol{B}_v\boldsymbol{v}(k) + \boldsymbol{B}_d\boldsymbol{d}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}_v\boldsymbol{v}(k) + \boldsymbol{D}_d\boldsymbol{d}(k) \end{aligned} \tag{3.5}$$

where the matrices $\boldsymbol{A}$, $\boldsymbol{B}_u$, $\boldsymbol{B}_v$, $\boldsymbol{B}_d$, $\boldsymbol{C}$, $\boldsymbol{D}_v$ and $\boldsymbol{D}_d$ can vary with time. The other parameters in the previous expression (3.5) are:

- k is the time index/current control interval;
- $\boldsymbol{x}$ are the plant model states;
- $\boldsymbol{u}$ are the manipulated inputs that can be adjusted by the MPC controller;
- $\boldsymbol{v}$ are the measured disturbance inputs;
- $\boldsymbol{d}$ are the unmeasured disturbance inputs;
- $\boldsymbol{y}$ are the plant outputs, including both measured (necessary at least one) and unmeasured.

In the adaptive MPC control, there are additional requirements for the plant model, like the sample time T_s that has to be constant and identical to the MPC control interval. This control strategy prohibits direct feed-through from any manipulated variable to any plant output. Thus, $\boldsymbol{D_v} = \boldsymbol{0}$ in the above model. A traditional

MPC controller includes a nominal operating point at which the plant model applies, such as the condition at which you linearize a nonlinear model to obtain the LTI approximation (equilibrium, reference trajectory and the most updated value) [4]. In adaptive MPC, as time evolves it should update the nominal operating point to be consistent with the updated plant model. It is possible to rewrite the plant model in terms of deviations from the nominal conditions as follows:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \overline{\boldsymbol{x}} + \boldsymbol{A}(\boldsymbol{x}(k) - \overline{\boldsymbol{x}}) + \boldsymbol{B}(\boldsymbol{u}_t(k) - \overline{\boldsymbol{u}}_t) + \overline{\Delta \boldsymbol{x}} \\ \boldsymbol{y}(k) &= \overline{\boldsymbol{y}} + \boldsymbol{C}(\boldsymbol{x}(k) - \overline{\boldsymbol{x}}) + \boldsymbol{D}(\boldsymbol{u}_t(k) - \overline{\boldsymbol{u}}_t) \end{aligned} \tag{3.6}$$

where the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are updated with respect to time. The other parameters in the previous structure (3.6) are:

- $\boldsymbol{u}_t$ is the combined plant input variable, comprising $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{d}$ variables defined earlier;
- $\overline{\boldsymbol{x}}$ are the nominal states;
- $\overline{\Delta \boldsymbol{x}}$ are the nominal state increments;
- $\overline{\boldsymbol{u}_t}$ and $\overline{\boldsymbol{y}}$ are the nominal inputs and outputs.

The adaptive MPC uses a Kalman filter to update its controller states which include the plant, the disturbance and measurement noise model states. In particular this filter is linear-time-varying (LTV) because adjusts the gains at each control interval to maintain consistency with the updated plant model.

Chapter 4

Moving Obstacle Avoidance

4.1 Problem Formulation

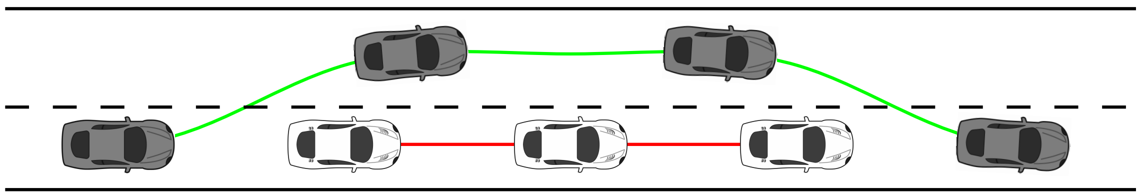


Figure 4.1: Problem description of collision avoidance on a road with only two lanes.

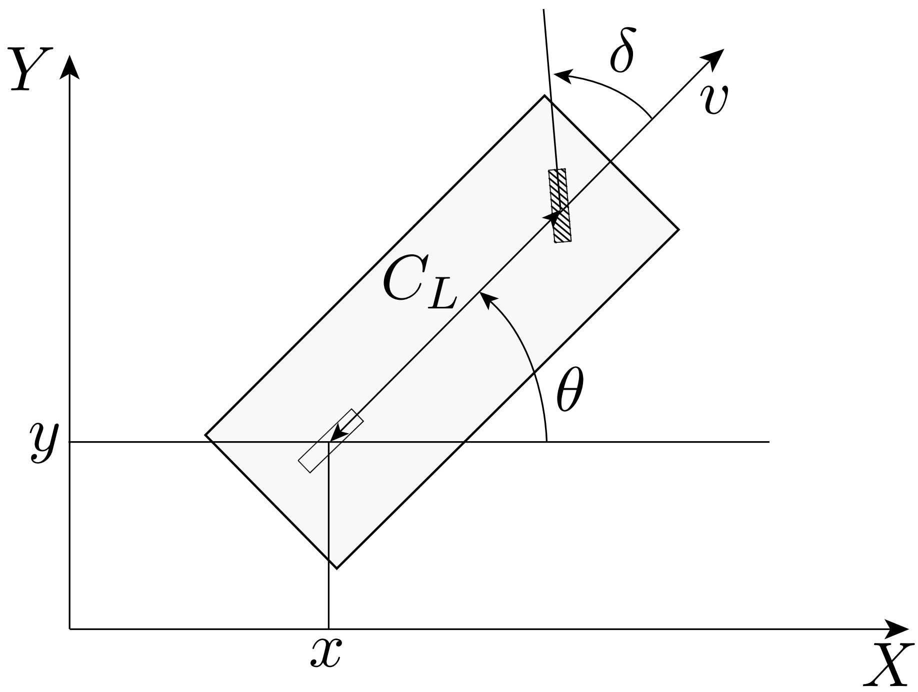


Figure 4.2: Bicycle model of a car, (adapted from [5]).

4.2 Design of Adaptive MPC

4.3 Simulation Results

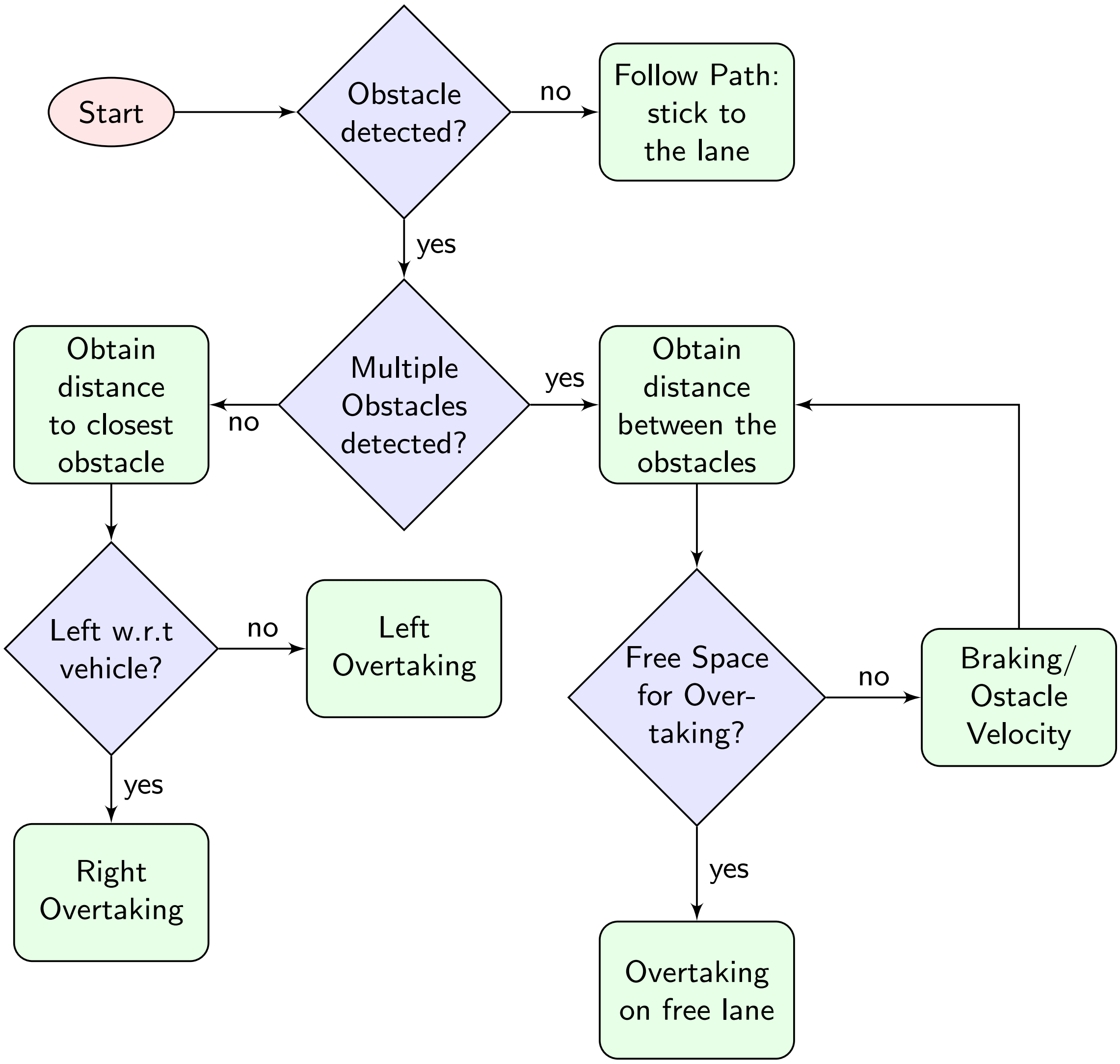


Figure 4.3: Behaviour planning conditional flowchart.

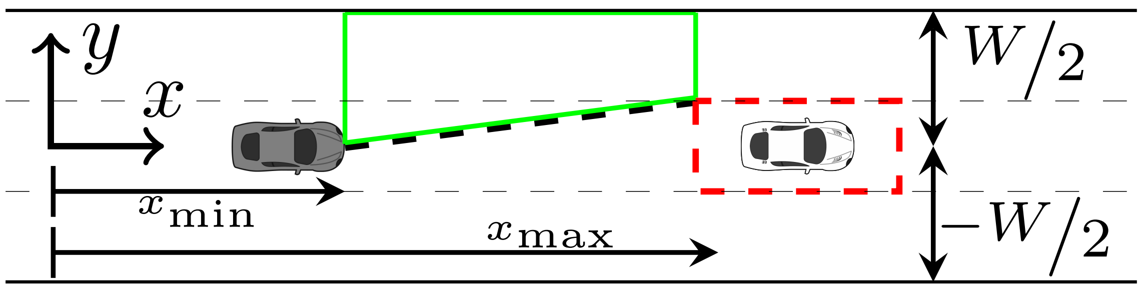


Figure 4.4: Constraints in the case of left overtaking.

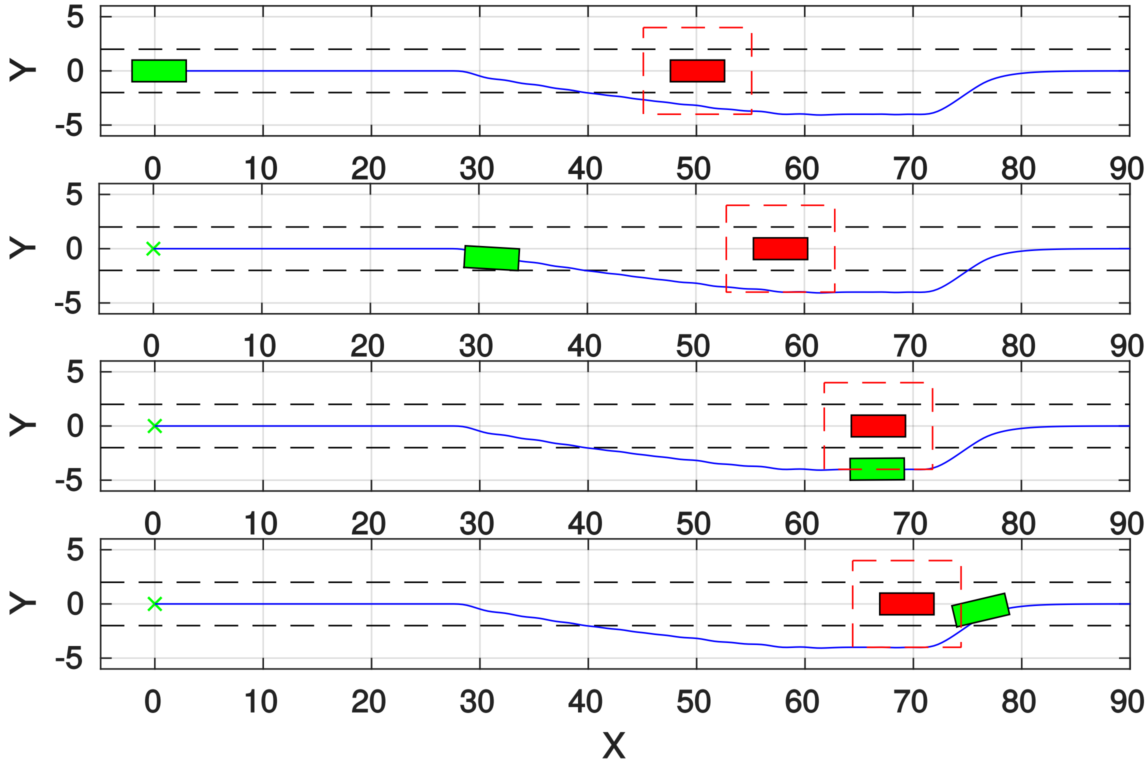


Figure 4.5: Simulation of right overtaking with one moving obstacle that moves in the same direction as the vehicle.

Algorithm 2 Right Overtaking if an obstacle is detected

function RightOvertaking(car, obstacle, road)
    $x_{\text{min}} \leftarrow \text{car}X$, $x_{\text{max}} \leftarrow +\infty$;
    obsYrr = obstacle.RearRightSafeY;
    obsXrr = obstacle.RearRightSafeX;
    if ATLASCAR2 is behind the obstacle **then**
        if ATLASCAR2 is in the adjacent lane **then**
            $cS \leftarrow 0$; $cI \leftarrow \text{obsYrr}$;
        else
            $cS \leftarrow \tan(\text{atan2}(\frac{\text{obsYrr} - \text{carY}}{\text{obsXrr} - \text{carX}}, 1))$;
            $cI \leftarrow \text{obsYrr} - cS * \text{obsXrr}$;
        end if
    else
        if ATLASCAR2 is parallel to the obstacle **then**
            $cS \leftarrow 0$; $cI \leftarrow \text{obsYrr}$;
        else
            $cS \leftarrow 0$; $cI \leftarrow W/2$;
        end if
    end if
    return x_{min}, x_{max}, cI, cS
end function

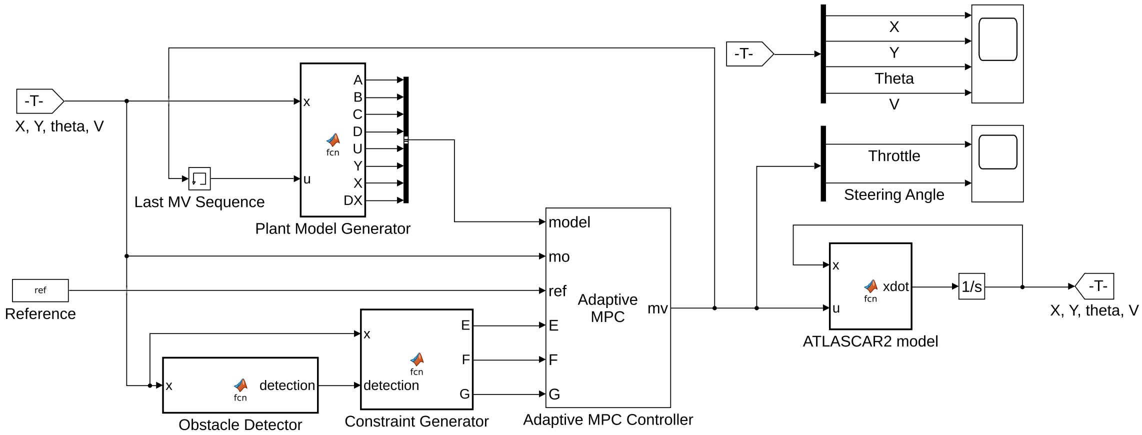


Figure 4.6: Overall procedure scheme moving obstacle avoidance.

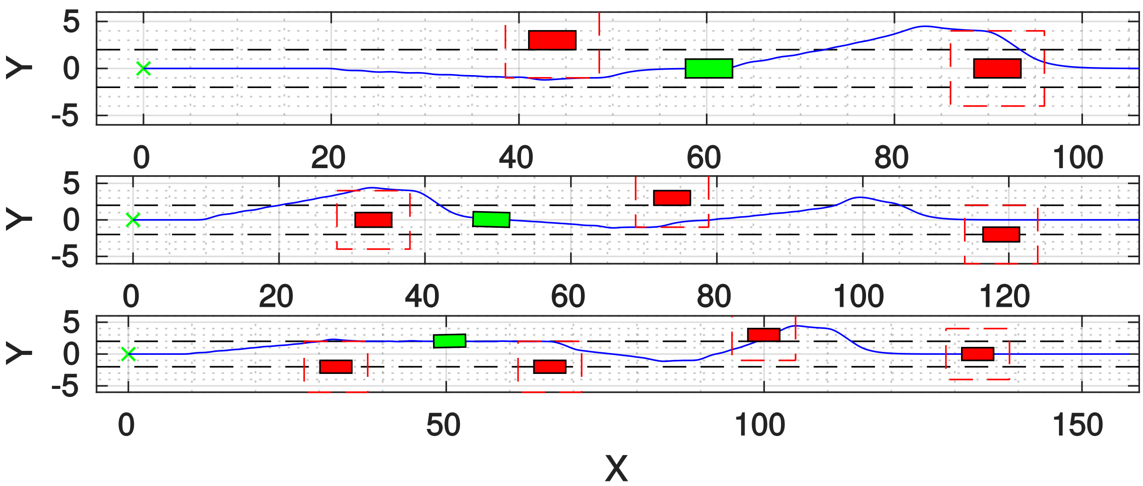


Figure 4.7: Simulations of overtaking with $N = 2, 3, 4$ moving obstacles that drive in the opposite direction with respect to the ATLASCAR2.

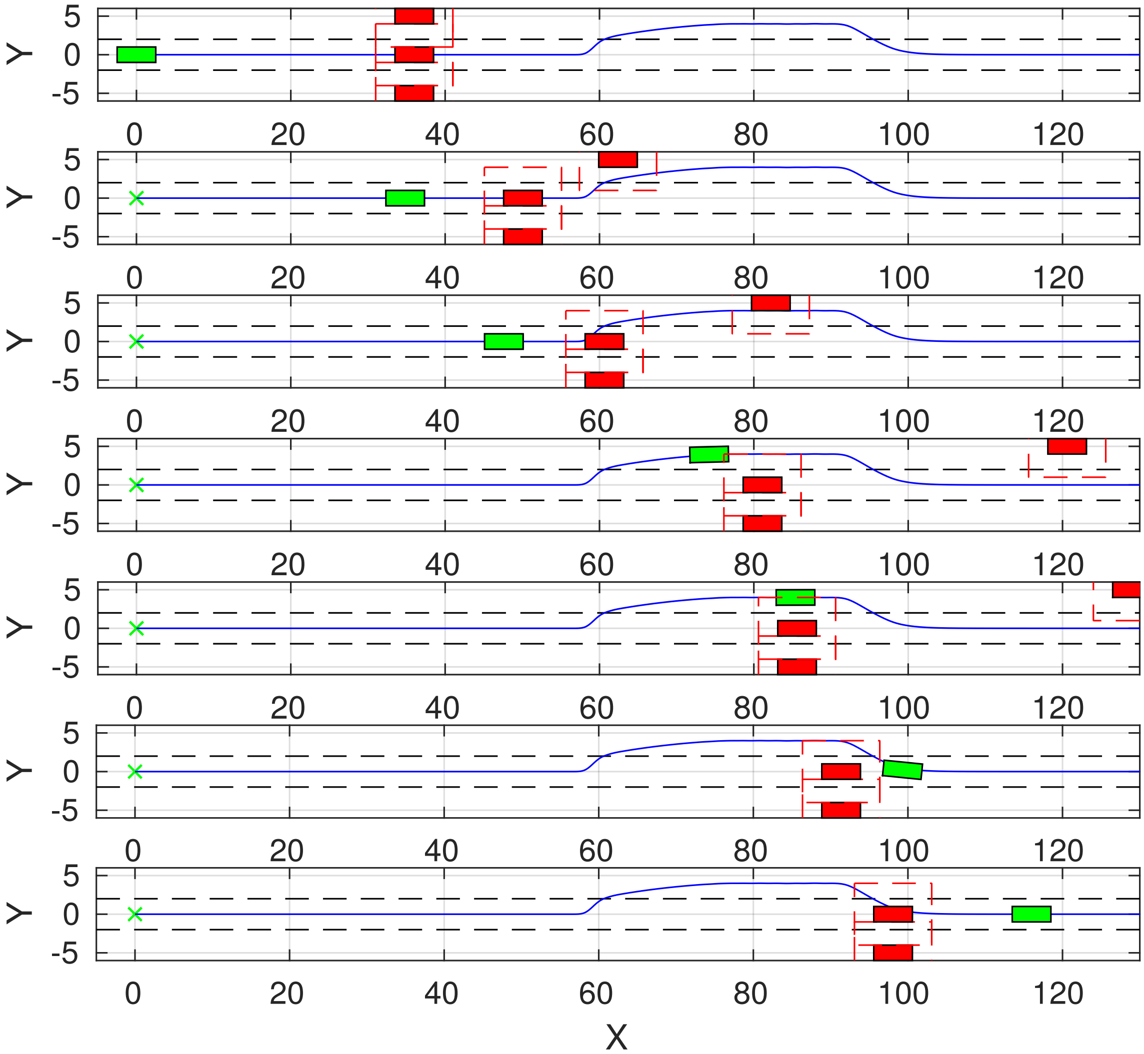


Figure 4.8: Simulation of braking and overtaking obstacles.

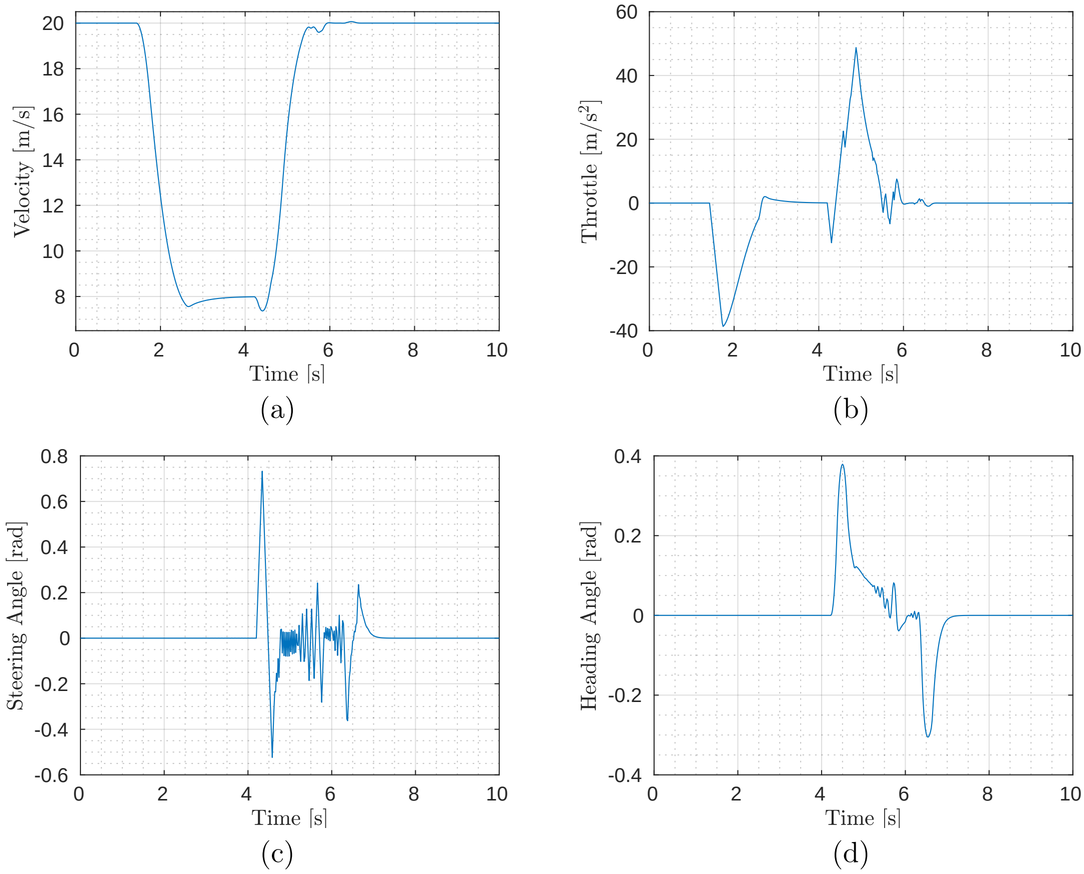


Figure 4.9: Time signals of the ATLASCAR2 in the simulation of braking and overtaking in the situation illustrated in Fig. 4.8.

Chapter 5

Lane Following

5.1 Problem Formulation

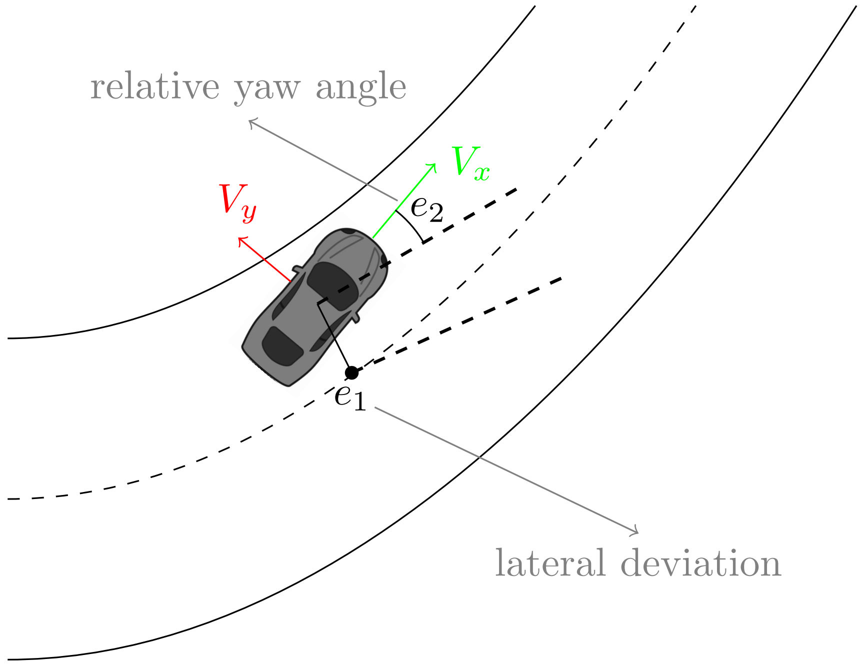
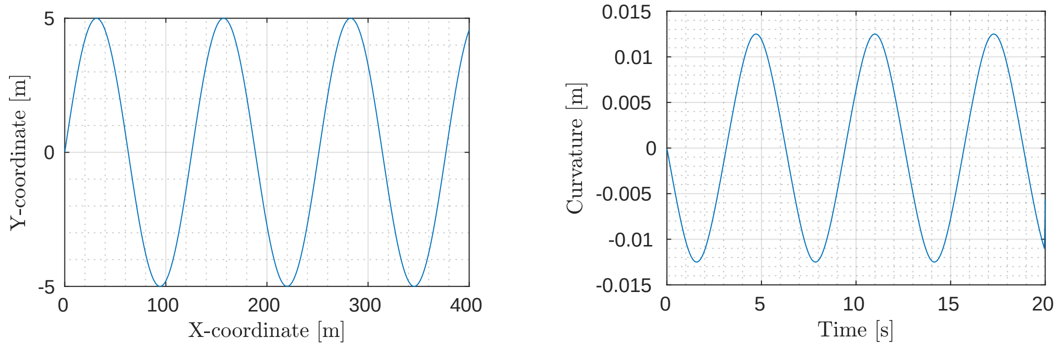


Figure 5.1: Problem description of a lane following system.

5.2 Design of Adaptive MPC

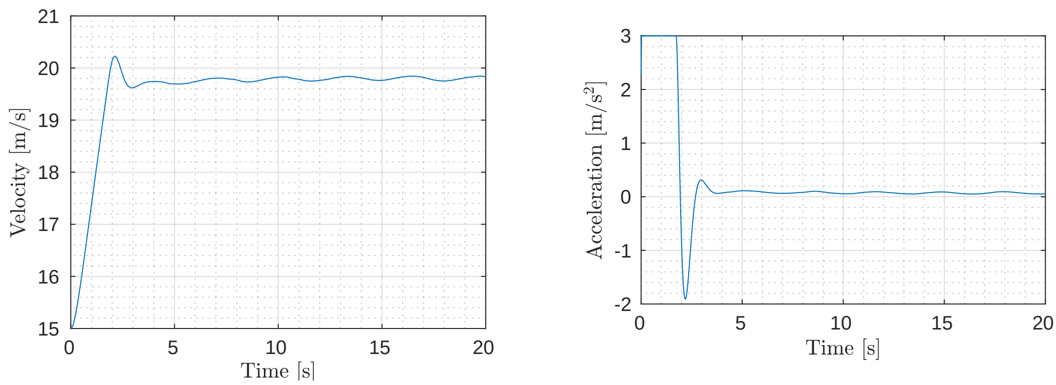
5.3 Simulation Results

Parameters	Values
m	1575 kg
I_z	2875 kgm^2
l_F	1.2 m
l_R	1.6 m
C_F	19 000 N/rad
C_R	33 000 N/rad
τ	0.2
V_0	15 m/s
V_{set}	20 m/s
T_s	0.02 s



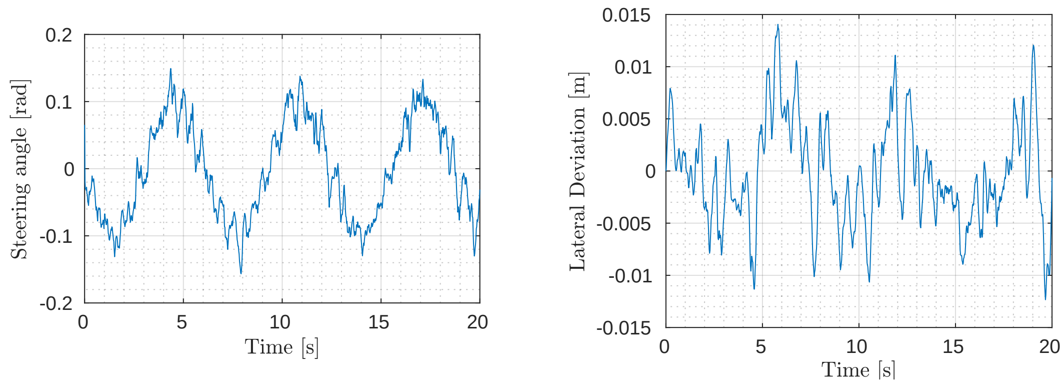
(a) (b)

Figure 5.2: Desired path and curvature of the ATLASCAR2 in a simulation of 20 s



(a) Longitudinal velocity V_x with respect to time.

(b) Acceleration u_1 with respect to time.



(c) Steering angle u_2 with respect to time.

(d) Lateral deviation e_1 with respect to time.

Figure 5.3: Time signals of the ATLASCAR2 in the simulation with a sinusoidal path.

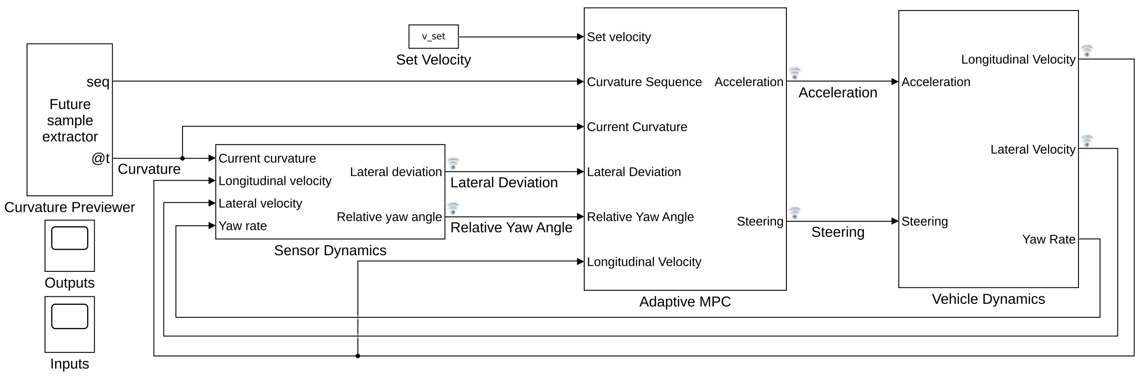


Figure 5.4: Overall procedure scheme lane following.

Conclusions and Future Work

Bibliography

[1] J. F. Pereira, "Estacionamento autónomo usando perceção 3D Autonomous parking using 3D perception," dissertação de mestrado, Departamento de Engenharia Mecânica, Universidade de Aveiro, 2012.

[2] R. Silva, "ATLASCAR2 Local Navigation for Autonomous Driving and Driver Assistance," Master's thesis, Universidade de Aveiro, Aveiro, 2018.

[3] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici, and P. Versari, "Extensive tests of autonomous driving technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1403–1415, 2013.

[4] A. Bemporad, M. Morari, N. L. Ricker, and H. T. C. Mathworks, "Model predictive control toolbox: Getting started guide, the math works."

[5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control.* Springer Publishing Company, Incorporated, 1st ed., 2008.

[6] J. Skoda, "3D Navigation for Mobile Robots," Master's thesis, Charles University, Prague, 2016.

[7] J. Bai, "Robot Navigation Using Velocity Potential Fields and Particle Filters for Obstacle Avoidance," Master's thesis, University of Ottawa, Ottawa, 2015.

[8] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila, and A. Rakotonirainy, "Perception, information processing and modeling: Critical stages for autonomous driving applications," *Annual Reviews in Control*, vol. 44, pp. 323–341, 2017.

[9] W. H. O. WHO, *Global Status Report on Road Safety 2015.* Nonserial Publication, World Health Organization, 2015.

[10] C. Katrakazas, M. Quddus, W. H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.

[11] V. Santos, J. Almeida, E. Avila, D. Gameiro, M. Oliveira, R. Pascoal, R. Sabino, and P. Stein, "ATLASCAR - technologies for a computer assisted driving system on board a common automobile," in *13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Madeira, Portugal, 19-22 September 2010*, pp. 1421–1427, 2010.

[12] R. F. Cabral de Azevedo, "Sensor Fusion of LASER and Vision in Active Pedestrian Detection," dissertação de mestrado, Departamento de Engenharia Mecânica, Universidade de Aveiro, 2014.

[13] D. T. Vieira da Silva, "Calibração Multissensorial e Fusão de Dados Utilizando LIDAR e Visão," dissertação de mestrado, Departamento de Engenharia Mecânica, Universidade de Aveiro, 2016.

[14] J. D. Madureira Correia, "Unidade de Perceção Visual e de Profundidade Para o Atlascar2," dissertação de mestrado, Departamento de Engenharia Mecânica, Universidade de Aveiro, 2017.

[15] K. Bimbraw, "Autonomous Cars: Past, Present and Future - A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology," *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics*, no. August, pp. 191–198, 2015.

[16] Waymo, "Waymo - journey," 2018.

[17] Waymo, "Waymo - technology," 2018.

[18] A. Broggi, P. Medici, E. Cardarelli, P. Cerri, A. Giacomazzo, and N. Finardi, "Development of the control system for the vislab intercontinental autonomous challenge," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 635–640, Sep. 2010.

[19] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 186–203, 2016.

[20] V. Santos, "ATLASCAR: A sample of the quests and concerns for autonomous cars," in *Informatics in Control, Automation and Robotics: 14th International Conference, ICINCO 2017 Madrid, Spain, July 26-28, 2017 Revised Selected Papers* (O. Gusikhin and K. Madani, eds.), Lecture Notes in Electrical Engineering, Springer International Publishing, 2019.

[21] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using rrt," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1681–1686, Sep. 2008.

[22] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," 01 2010.