



DEGREE PROJECT IN MECHANICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Model Predictive Control for path tracking and obstacle avoidance of autonomous vehicle

AHMED HATEM

Sammanfattning

Konceptet autonoma fordon har på senare tid blivit brett utforskat av bland annat bilindustrin som ett sätt att exempelvis förbättra bränsleekonomi eller få tillgång till miljöer som utgör en fara för mänskliga operatörer. Modell-prediktiv reglering (MPC) har traditionellt använts för att styra system med långsam dynamik men i och med uppkomsten av kraftfullare datorer används det nu i system med avsevärt snabbare dynamik. En av de huvudsakliga styrkorna hos MPC är dess förmåga att hantera restriktioner som finns i alla fysikaliska system. Målet med den här uppsatsen var att utveckla en linjär regulator bestående av ett lager för vägföljning och undvikning av hinder för en autonom bil. Dess förmåga att minimera avvikelser till referensbanan och samtidigt undvika statiska hinder utvärderades. Fokus placerades på spårningen av referensbanan följaktligen implementeras inte något system för planering av banan. Istället användes en fördefinierad bana. Simuleringar utvecklades i matlab baserat på den kinematiska cykelmodellen. Regulatorns prestanda utvärderas vidare på Smart Mobility Lab (SML) på KTH där en modifierad radiostyrd bil styrdes via Robotics Operating System (ROS). Resultaten från experimenten visade på att bilen klarar av att undvika hindren samtidigt som den följde banan. Dock misslyckades bilen i experimenten att respektera kraven på maximal avvikelse från hindren och banan.

Model Predictive Control for path tracking and obstacle avoidance of autonomous vehicle

Ahmed Hatem

Abstract—The concept of autonomous vehicles has been widely explored lately by, among others, automotive companies as a way to for example improve fuel efficiency or to gain access to environments which pose a danger to human operators. Model Predictive Control (MPC) has traditionally been used to control systems with slower dynamics but with the emergence of more powerful computers it is now being used in systems with considerably faster dynamics as well. One of the main strengths of MPC is its ability to handle constraints which are present in all physical systems. The aim of this thesis was to develop a single layer linear controller for path tracking and obstacle avoidance of an autonomous car. Its ability to minimize the deviations to the reference path while clearing static obstacles was evaluated. Focus was placed on the tracking problem hence no trajectory planning system was implemented. Instead a predefined path was used. Simulations were developed in MATLAB based on the kinematic bicycle model. The performance of the controller was further tested at Smart Mobility Lab (SML) in KTH where a modified R/C car was controlled through Robotics Operating System (ROS). The results from the experiments showed that it was able to successfully evade the obstacles while tracking the path. However, in the experiments the vehicle failed to respect the requirements on maximum deviation from the obstacles and the path.

I. INTRODUCTION

THE concept of autonomous vehicles was for not so long ago an unrealistic technology. Today autonomous road, surface and aerial vehicles are reality. One of the challenges that engineers face is the ability of the vehicle to track a given path, i.e. the tracking problem. To this end Model Predictive Control (MPC) has been employed partly due to its ability to handle constraints which are necessary in the modeling of all physical systems. The main idea of MPC is to solve a finite horizon optimal control problem at certain time steps and compute an optimal control sequence of which only the first control action is implemented. The procedure is then repeated at the next sampling step as new information about the state has been obtained from a model.

Due to an increase in the number of road vehicles in today's modern society safety has become a major issue. Studies have shown that by 2006 road traffic accidents was the leading cause of death by injury as well as the tenth-leading source to all deaths globally. Estimations suggest that 1.2 million people are killed in road crashes each year. Furthermore, as many as 50 million are injured [1]. If current trends persist road traffic injuries are predicted to become the third-leading contributor to the global burden of disease and injuries by 2020 [2]. Although passive as well as active systems have been developed and implemented in vehicles over the years [3], there is a desire to further reduce the number of traffic accidents. Since a great deal of light-duty vehicle crashes occur

due to human error and distracted driving partially-automated crash avoidance systems can offer a potential reduction in both the frequency and the severity of accidents [4]. This suggests that the implementation of fully-automated road vehicles could possibly provide an even greater improvement.

Recently predictive control for autonomous vehicles has also been studied, [5] is an example where the vehicle was instructed to follow a given reference path. In this specific case the path is assumed to be collision free, a scenario which may not be useful in case of unknown environments. Nevertheless, it can be utilized in for instance mining networks which are well mapped.

Experiments with autonomous vehicles have been successfully conducted over the years, [6] and [7] are two such examples. Although they proved to be successful there are still some years before the concept of autonomous road vehicles will be deployed on a large scale. Both tests were performed on days which were sunny and clear, in the absence of any heavy rain or snow. Another issue is that some maneuvers were performed with safety in mind and could sometimes deviate from common human behaviour. This needs to be considered in the future to avoid that the behavior of the vehicle becomes too inefficient when compared to a piloted vehicle. A better understanding of human behavior still remains a major difficulty to tackle.

While the two above mentioned studies targeted urban areas off-road path following also receives some attention. Scania has shown interest in developing autonomous vehicles with specialized applications such as for deployment in mining sites. In [8] the authors propose an Economic Model Predictive Control (EMPC) for smooth path following. The objective being to track a path while at the same time minimizing the first as well as the second derivative of the road curvature. The proposed controller is implemented on a construction truck and its performance in terms of path tracking accuracy is compared to a Pure Pursuit Controller (PPC). Results show that the suggested EMPC outperforms the PPC with an average deviation from the road of 6 cm. Hence the area of autonomous road vehicles has been widely studied over the years, although major challenges still remain to be tackled before it can be widely used in society.

A. Thesis outline

The objective of this thesis is to develop a simple MPC for path tracking and obstacle avoidance for autonomous road vehicle. The controller will be developed and tested in simulations performed in MATLAB. To assess the performance of the controller it will be implemented on a modified R/C car

with a scale of measure of approximately 1:10. The tests are carried out at Smart Mobility Lab in KTH. The vehicle should be able to track a straight line with a velocity of 0.5 /s while avoiding obstacles placed along the path. More specifically the vehicle should fulfill the following requirements:

- Handle disturbances in the form of initial lateral deviations from the path on up to 0.4 m.
- In the absence of any obstacles the settling distance must not exceed double the initial lateral deviation. This is the distance projected along the path that the vehicle has to travel in order to recover to the path with an over/undershoot no larger than 0.1 m.
- Avoid 3 static obstacles by driving around them.
- The vehicle should clear the obstacles with a distance not larger than 0.07 m.
- Over/undershoots from the path must not exceed 0.07 m.

Moreover, following assumptions are made:

- The vehicle travels with a constant velocity.
- The obstacles are identical with the dimension 0.14×0.14 m.
- The shape and position of the obstacles are known to the controller.
- The estimations of the states are free from errors and noise. No filters are used to deal with uncertainties and noise.
- No trajectory planning system is available, instead the path is predefined.

II. BACKGROUND

A control system for an autonomous vehicle can be regarded as a system composed of two components. Firstly there is the path planning algorithm which generates a collision free trajectory. Work in this area dates back to the 1980s and focused mainly on calculating a time-optimal and collision-free path reaching from one point to another [9], [10]. Ever since then many different methods and implementations have been reported including the Rapidly-exploring Random Tree Algorithm (RRT), generalized Voronoi diagram and the visibility graph method. The second component is the path tracking algorithm which ensures that the vehicle follows the planned trajectory. In order to achieve this one has to take into consideration that vehicles are made out of mechanical and electrical systems which are subjected to physical constraints. Furthermore, the dynamics of the vehicles are nonlinear. With its ability to handle both linear as well as nonlinear constraints in a systematic manner MPC is an attractive method for tackling this problem. However, the usage of MPC for systems with fast dynamics is challenging especially for nonlinear systems. Since an optimization problem has to be solved the process is computationally heavy and therefore requires powerful computers. Even though there are examples of successful implementations of MPC controllers on test vehicles [11], [12], they are not nonlinear but linear. Hence, implementations of nonlinear MPC on autonomous vehicles still pose some challenges [13]. A possible solution to this issue could be to divide the control structure into two layers where one layer deals with the nonlinear vehicle model along with constraints

and trajectory generation while the design of the second layer is based on a linearization around a trajectory provided by the first layer. This layer is used to stabilize the vehicle. An additional design technique that can be used to further decrease the computational demand necessary to obtain fast feedback required by vehicle stabilization is to use a lower sampling rate for the former control layer and a higher for the latter. These measures along with others were investigated in [14] with the aim to develop an MPC for obstacle avoidance. For the obstacle avoidance ability the authors suggest an approach where the obstacles are enclosed in circles and a term is added to the objective function which when penalized appropriately makes sure that the vehicle does not enter the region in which the obstacle lays. The results suggest that the two-level control framework overcomes the issue related to computational burden and the vehicle is not only able to track the reference path but also able to evade the obstacles successfully.

Studies have also been conducted regarding single layer controllers for nonlinear vehicle models with the aim of tracking a reference. In [15] the performance of a nonlinear vehicle model is compared to another formulation where the nonlinear model is successively linearized around the current operating point. The findings in this study show that the computational burden poses a problem for the former formulation while the latter is able to stabilize the vehicle for moderate speeds. This highlights the challenges present for implementation of an MPC with nonlinear dynamics for obstacle avoidance system. In this thesis it will be investigated, on the basis of the requirements, how well a simple linear single layer MPC performs.

A. Model Predictive Control

MPC is an optimal control strategy which traditionally has been applied to systems with rather slow dynamics, such as process control plants. However, as more powerful computing hardware have emerged over time it is today used in systems with considerably faster dynamics such as on flight control computers in airplanes as well as software for combustion engines. As mentioned above the main strength of the method lies in its ability to control linear and nonlinear systems while taking into account state as well as input constraints. The method uses a predictive control strategy where the future response of the controlled plant is predicted using a discrete linear time invariant dynamic model. The system is represented in state space form

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + u_t \end{aligned} \quad (1)$$

where $x_t \in \mathbb{R}^n$ is the state vector, $u_t \in \mathbb{R}^m$ is the control input and $y_t \in \mathbb{R}^p$ is the output vector. The time index $t \in \mathbb{Z}$ is integer valued. The principle is to find an open loop control sequence which minimizes a certain cost function over a finite

time horizon. The problem can be formulated as

$$\begin{aligned}
& \min_{\mathbf{u}} && J(x(t), \mathbf{u}) \\
\text{subject to} &&& x_{t+k+1} = Ax_{t+k} + Bu_{t+k}, \quad \forall k = 0, \dots, N_p - 1 \\
&&& x_{t+k} \in \mathbb{X}, \quad \forall k = 0, \dots, N_p - 1 \\
&&& u_{t+k} \in \mathbb{U}, \quad \forall k = 0, \dots, N_p - 1 \\
&&& x_{t+N_p} \in \mathbb{X}_f \\
&&& x_t = x(t),
\end{aligned} \tag{2}$$

where $\mathbf{u} = (u_t, \dots, u_{t+N_p-1})$ is a sequence of control inputs, x_{t+k} denotes the state at time $t+k$ as predicted at time t and N_p is the prediction horizon. The sets $\mathbb{X} \in \mathbb{R}^n$ and $\mathbb{U} \in \mathbb{R}^p$ are the constraints on the state and input respectively. The terminal constraint set in the state is $\mathbb{X}_f \subseteq \mathbb{X}$. Now if the goal is to steer system (2) to the origin the cost function $J(x(t), \mathbf{u})$ can be chosen as a quadratic function

$$J(x(t), \mathbf{u}) = x_{t+N_p}^T P_f x_{t+N_p} + \sum_{k=1}^{N_p-1} x_{t+k}^T Q x_{t+k} + u_{t+k}^T R u_{t+k}, \tag{3}$$

where P_f, Q are positive semidefinite ($P_f, Q \geq 0$) and R is positive definite ($R > 0$). The algorithm starts with a measurement of the state $x(t)$. Then optimization problem (2) is solved. In the optimal control sequence found in the previous step only the first control is applied. The system then evolves one time step after which the procedure is repeated.

Since there exist standard solvers for convex optimization problems it is desirable to make (2) convex. This is achieved if the cost function is convex, the system dynamics are linear and the constraint sets \mathbb{X}, \mathbb{U} are convex as well.

III. VEHICLE MODEL

A kinematic model representing the lateral motion of a vehicle can under some specific assumptions described below be developed. In contrast to a dynamic model where the forces acting on the system are taken into account the kinematic model is derived from mathematical relationships governing the system. Furthermore, compared to higher fidelity vehicle models the system identification on the kinematic model is easier because of the low number of system parameters needed to describe the system. In addition to that studies have shown that the kinematic model is under some conditions able to actually perform similarly well compared to a more complex and computationally heavy dynamic model [16]. For this application where the velocity of the vehicle as well as the lateral acceleration are low the kinematic model performs well [17]. This motivates the design of the controller based on a kinematic model.

A. Kinematic model

Consider a bicycle model of the vehicle where the two wheels in each axle are represented by a single wheel aligned with the center of gravity of the vehicle, see Figure 1. The

nonlinear continuous time equations that describe this model in an inertial frame rate are [18]

$$\dot{x} = v \cos(\psi) \tag{4a}$$

$$\dot{y} = v \sin(\psi) \tag{4b}$$

$$\dot{\psi} = \frac{v}{l} \tan(\delta) \tag{4c}$$

where x and y are the coordinates of the vehicle given in the global coordinate system. The yaw angle is denoted ψ and the wheelbase is given by l . The longitudinal component of the velocity is v . In the kinematic bicycle model it is assumed that the slip angles at both wheels are zero. Consequently the lateral component of the velocity is zero. This is a reasonable assumption to make for low velocities [19]. The steering angle of the front wheels with respect to the longitudinal axis of the car is denoted δ .

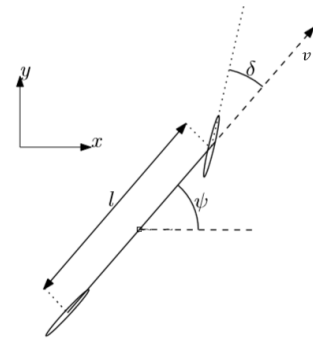


Fig. 1. The nonlinear kinematic bicycle model.

Note that in this model it is assumed that both left and right front wheel have the same steering angle. In reality the inner wheel turns with a greater angle due to the fact that it covers a shorter distance than that of the outer wheel. However, this difference is in this case small enough to be neglected.

B. Road-aligned coordinates

In order to formulate the tracking problem a model transformation from time-dependent vehicle dynamics to a track-dependent (also known as Frenet-Serret frame or curvilinear frame) dynamics is proposed. Similar ideas have previously been developed in the area of robotics [20]. To this end a new variable s , which represents the the distance traveled along the reference path, is introduced. The system dynamics presented in the previous section are modeled in terms of the independent variable s . In this frame of reference the states of interest are the lateral deviation to the reference path and the error in the heading angle here denoted e_y and e_ψ respectively. Figure 2 shows the road-aligned coordinate system where ρ_s is the road radius and ψ_s is the heading angle of the reference path. The two components of the velocity \dot{x} and \dot{y} are in the body frame of reference. The velocity projected along the direction of the center line, represented in Figure 2 by a solid arrow, is denoted v_s .

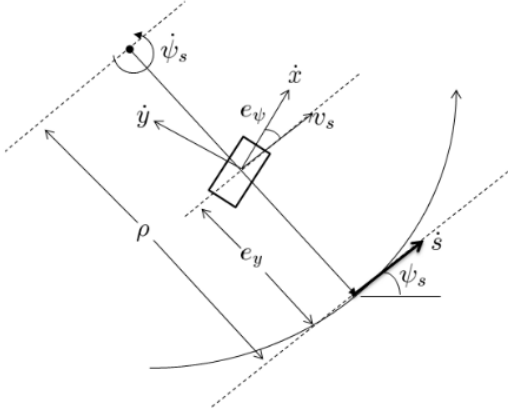


Fig. 2. The road-aligned coordinate system. The dynamics are derived about a curve defining the centerline of a path.

With the assumption that the lateral component of the vehicle velocity is zero, i.e $\dot{y}=0$ the following equations can be derived from Figure 2

$$\dot{e}_y = v_s \sin(\psi_s), \quad (5a)$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_s, \quad (5b)$$

$$\dot{s} = \frac{\rho_s v_s \cos(e_\psi)}{\rho_s - e_y}. \quad (5c)$$

In order to derive the spatial dynamics the following relation is used

$$\frac{d(\cdot)}{ds} = \frac{d(\cdot)}{dt} \frac{dt}{ds}. \quad (6)$$

If $\dot{s} \neq 0$ is assumed at any time, by the inverse function theorem it holds that

$$\frac{dt}{ds} = \frac{1}{\dot{s}}. \quad (7)$$

The derivatives of (5a) - (5b) with respect to s , represented by $(\cdot)'$, are

$$e_y' = \frac{de_y}{ds} = \frac{\dot{e}_y}{\dot{s}} = \frac{\rho_s - e_y}{\rho_s} \tan(e_\psi), \quad (8a)$$

$$e_\psi' = \frac{de_\psi}{de} = \frac{\dot{e}_\psi}{\dot{s}} = \frac{(\rho_s - e_y) \tan(\delta)}{\rho_s \cos(e_\psi)} - \psi_s', \quad (8b)$$

which is a spatial-based representation of the vehicle dynamics. In order to switch from the spatial coordinates to the global coordinates X , Y and ψ the following transformation is necessary:

$$X = X_s - e_y \sin(\psi_s), \quad (9a)$$

$$Y = Y_s + e_y \cos(\psi_s), \quad (9b)$$

$$\psi = \psi_s + e_\psi, \quad (9c)$$

where X_s , Y_s and ψ_s denote the the global coordinates as well as the heading angle of the desired path [21],[22].

C. Linearization

The proposed models are both nonlinear which results in a non-convex optimal open-loop control problem. One can expect conventional solvers for nonlinear problems to find local rather than global solutions. On the other hand for linear systems, quadratic cost functions and constraint sets which are polyhedral the problem reduces to a quadratic one for which efficient solvers exist. Furthermore, these solvers yield a global optimal solution [23]. In order to obtain good performance for the tests it is necessary to keep the solving time for the optimization down. This will be easier achieved with the use of a linear model. Consequently, it is highly preferable to work with linear systems. It is therefore necessary to linearize the systems around an appropriate point. For simplicity this point was chosen as $e_y = 0$, $e_\psi = 0$ and $\delta = 0$. The system was discretized using forward Euler's method with sampling distance $\Delta s = vT_s$ where T_s is the sampling time.

IV. OBSTACLE AVOIDANCE SYSTEM

The size and position of the obstacles are as previously mentioned assumed to be known. The system works in such a way that as soon as a collision is predicted a soft constraint will be set on the lateral deviation e_y . This allows the vehicle to pass the obstacle either by making a left or right hand turn. How the vehicle chooses to act is predefined in the algorithm. If the center of the obstacle is placed above the path the vehicle will make a right hand turn to pass it. Likewise, if the obstacle is centered below the path the vehicle is instructed to make a left hand turn. If on the other hand the obstacle is placed along the path the vehicle will make a turn based on the placement of the upcoming obstacle. In other words the controller will choose the way which minimizes the lateral distance to the reference path. In the simulations the center of gravity of the vehicle was chosen as the point which must avoid the obstacles. In reality the vehicle cannot be represented by a point so to take the width of the vehicle into consideration the size of the obstacles were simply increased.

V. CONTROL PROBLEM FORMULATION

Based on the spatial representation of the vehicle model and the objective to follow a desired path while evading obstacles the following MPC formulation was formulated:

$$\min \sum_{t=1}^{N_p} Q_1 e_{y_t}^2 + Q_2 e_{\psi_t}^2 + Q_3 \epsilon_t^2 + Q_4 u_{t-1}^2 + R_1 \Delta u_t^2 \quad (10a)$$

$$\text{s.t. } x_{t+1} = \mathbf{A}x_t + \mathbf{B}(u_{t-1} + \Delta u_t) + \mathbf{C} \quad \forall t = 0, \dots, N_p \quad (10b)$$

$$|u_t| - \epsilon_t \leq u_{max} \quad \forall t = 0, \dots, N_p \quad (10c)$$

$$|\Delta u_{t-1}| \leq \Delta u_{max} \quad \forall t = 0, \dots, N_p \quad (10d)$$

$$\epsilon_t \geq 0 \quad \forall t = 0, \dots, N_p \quad (10e)$$

To achieve the goal of following the desired path the objective function is defined such that the square of e_y and e_ψ are minimized. As mentioned in the previous section a bound is set on e_y in order for the vehicle to avoid the obstacles. To prevent

the optimization problem from becoming infeasible those constraints along with the hard constraints on the steering angle u_t had to be softened with a slack variable ϵ_t . As in any physical system the control input u_t at step t and the control correction variable Δu_{t-1} at step $t-1$ are limited. In this case the maximum steering angle is upper bounded by u_{max} while the change of rate is limited by Δu_{max} . In this formulation the control correction variable is treated as the input to the system while the steering angle is considered as a state. With this formulation the control input at time t can be computed recursively with

$$u_t = u_{t-1} + \Delta u_t. \quad (11)$$

The state vector $[e_y \ e_\psi \ s]^T$ is denoted x_t and the matrix \mathbf{A} , the vectors B and C are obtained by linearizing and discretizing (5c), (8a) and (8b). Moreover, since the intention is to drive in a straight line the limit of the above mentioned equations as $\rho \rightarrow \infty$ had to be calculated. The final results are

$$\mathbf{A} = \begin{bmatrix} 1 & vT_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{vT_s}{l} \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 \\ 0 \\ vT_s \end{bmatrix}. \quad (12)$$

The system dynamics can now as described in [24] be written as

$$\begin{bmatrix} x_{t+1} \\ u_t \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tilde{\mathbf{A}}} \begin{bmatrix} x_t \\ u_{t-1} \end{bmatrix} + \underbrace{\begin{bmatrix} B \\ I \end{bmatrix}}_{\tilde{B}} \Delta u_t + \underbrace{\begin{bmatrix} C \\ I \end{bmatrix}}_{\tilde{C}}. \quad (13)$$

VI. EXPERIMENTAL SETUP

The simulations were initially performed in MATLAB using YALMIP, a toolbox for modeling and optimization. The toolbox uses quadprog as standard for solving quadratic programming problems. At a later stage this simulation was replaced with a more efficient one where the problem was reformulated as a quadratic programming problem. The results are presented in the following sections. To assess the performance of the controller it was implemented and tested on a modified R/C car, see Figure 3. It is equipped with an Nvidia Jetson TX2 embedded computer. The position, orientation as well as velocity of the vehicle were tracked using a set of motion capture cameras (MOCAP) from Qualisys.

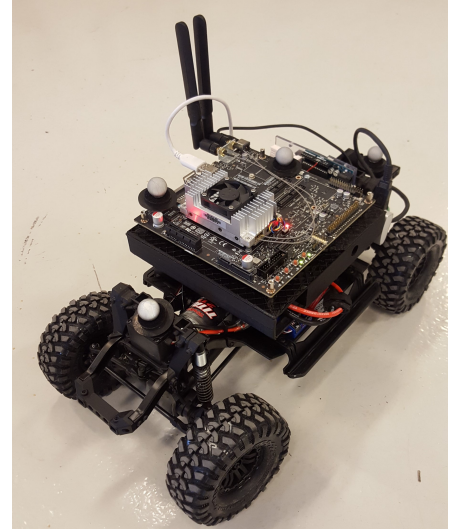


Fig. 3. Test vehicle.

A. Robotics Operating System

The optimization problem was solved on a separate computer using MATLAB. The communication between the computer, the vehicle and the sensors was carried out through the Robotics Operating System (ROS). Each of the components constitute a node in the system. A ROS master is set up making sure that the individual nodes are able to locate each other. The nodes communicate with each other through messages. As soon as the sensors publish new data to the topic `/qualisys_odom` the computer, being a subscriber to that topic, receives the data. The computer then solves the optimization problem and publishes the control input in the topic `/cmd` to which the vehicle is a subscriber. See Figure 4 for a schematic of the setup.

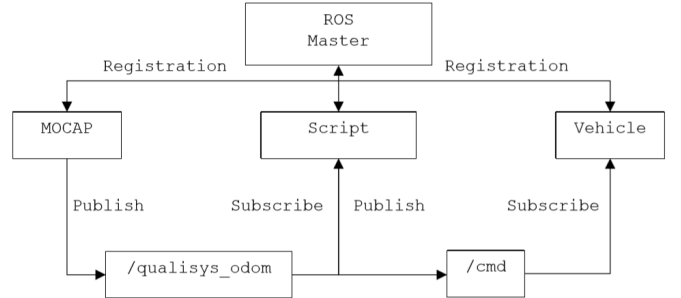


Fig. 4. Schematics of ROS setup.

B. MPC formulated as Quadratic Programming

In order to obtain good performance in the experiments the sampling time for the controller had to be minimized. The sensors fed the controller with new data with a rate of 20 Hz meaning that the optimization could not take longer than 50 ms to complete. Otherwise no feedback will be provided to the vehicle which could lead to slow responses and collision with obstacles. The simulations developed in YALMIP proved not to fulfill this requirement. To improve the efficiency of the controller the problem was rewritten

as a quadratic programming problem [25]. To start with the optimization problem was rewritten on the form

$$\begin{aligned} \min \quad & \frac{1}{2} z^T \mathbf{H} z \\ \text{subject to} \quad & \mathbf{A}_{eq} z = b_{eq} \\ & \mathbf{A} z \leq b \end{aligned} \quad (14)$$

where $z = (x_{t+1|t}^T, \dots, x_{N_p|t}^T, u_{t-1|t}, \dots, u_{t+N_p-1})^T$. The matrix \mathbf{H} associated with the objective function is a diagonal square matrix of the size $4N_p + N_p + 1 \times 4N_p + N_p + 1$. The structure of the matrix is

$$\mathbf{H} = \begin{bmatrix} \mathbf{Q} & 0 & \dots & \dots & 0 \\ 0 & \mathbf{Q} & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \dots & \mathbf{R} & 0 \\ 0 & 0 & \dots & \dots & w_\epsilon \end{bmatrix}$$

where the submatrices are defined as

$$\mathbf{Q} = \text{diag}(w_{e_y}, w_{e_\psi}, w_s, w_u), \quad \mathbf{R} = \text{diag}(r, \dots, r). \quad (15)$$

The size of \mathbf{Q} is 4×4 while \mathbf{R} is $N_p \times N_p$. Note also that w_ϵ is the weight for the slack variable and is a scalar. The matrix \mathbf{A}_{eq} along with the vector b_{eq} are attained by considering the system dynamics as

$$Z = \mathcal{A}z_{t|t} + \mathcal{B}U + \mathcal{C} \quad (16)$$

where $Z = (z_{t+1|t}^T, \dots, z_{N_p|t}^T)^T$ and $U = (u_{t|t}, \dots, u_{t+N_p-1|t})$. The matrices \mathcal{A} and \mathcal{B} and the vector \mathcal{C} are given as

$$\mathcal{A} = \begin{bmatrix} \tilde{\mathbf{A}} \\ \tilde{\mathbf{A}}^2 \\ \vdots \\ \tilde{\mathbf{A}}^{N_p} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} \tilde{\mathbf{B}} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{B}} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \tilde{\mathbf{A}}^{N_p-1}\tilde{\mathbf{B}} & \tilde{\mathbf{A}}^{N_p-2}\tilde{\mathbf{B}} & \dots & \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{B}} \end{bmatrix},$$

$$\mathcal{C} = \begin{bmatrix} \tilde{\mathbf{C}} \\ \tilde{\mathbf{C}} + \tilde{\mathbf{A}}\tilde{\mathbf{C}} \\ \vdots \\ \tilde{\mathbf{C}} + \tilde{\mathbf{A}}\tilde{\mathbf{C}}, \dots, + \tilde{\mathbf{A}}^{N_p-1}\tilde{\mathbf{C}} \end{bmatrix}$$

equation (16) can be rewritten as

$$Z - \mathcal{B}U = \mathcal{A}z_{t|t} + \mathcal{C}. \quad (17)$$

Note that $z = (Z^T U^T)^T$ hence the matrix \mathbf{A}_{eq} and the vector b_{eq} can now be identified as

$$\mathbf{A}_{eq} = [\mathbf{I} - \mathcal{B}], \quad b_{eq} = \mathcal{A}z_{t|t} + \mathcal{C}. \quad (18)$$

As for the limitation on the input and steering angle the matrix \mathbf{A} and the vector b are introduced. These are defined as

$$\mathbf{A} = \begin{bmatrix} 0 & \dots & \mathbf{I} \\ 0 & \dots & -\mathbf{I} \\ \dots & \tilde{\mathbf{A}} & \dots \\ \dots & \tilde{\mathbf{A}} & \dots \\ 0 & \dots & -1 \end{bmatrix}, \quad b = \begin{bmatrix} u_{max} \\ \vdots \\ \bar{b} \\ \Delta u_{max} \\ \vdots \\ -1 \end{bmatrix}. \quad (19)$$

The identity matrices are of size $N_p \times N_p$. The last element in both the matrix \mathbf{A} and the vector b is for the non-negativity constraint on the slack variable. The submatrix $\tilde{\mathbf{A}}$ of size $N_p \times N_p N_s + N_p + 1$ along with the subvector \bar{b} of size $N_p \times 1$ define the upper or lower constraints on the lateral deviation necessary for for the vehicle to evade an obstacle. If for instance the center of the obstacle is aligned with the path and the goal is to pass it from above there will be a positive bound on the lateral deviation equal to half of the width of the obstacle. As for the constraint on the control correction variable they are represented by the matrix $\hat{\mathbf{A}}$ which is of size $N_p \times N_p N_s + N_p + 1$.

It is evident from Figure 5 and Table I that the QP formulation of the MPC problem is significantly faster than the initial one developed in YALMIP. It shows the time to complete an optimization step for sampling time 0.05 s and with a prediction horizon of 20. Note that the first two iterations have been excluded since they take significantly longer to run because of the initialization in MATLAB. Because of their short duration they do not affect the performance of the controller.

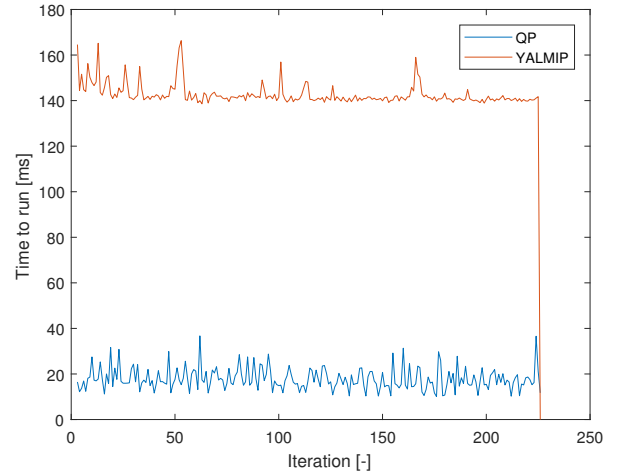


Fig. 5. Time to complete optimization for each sampling step.

	YALMIP	QP
Max [ms]	166	37
Mean [ms]	142	18

TABLE I

COMPARISON OF PERFORMANCE OF SCRIPTS.

C. Evaluation of performance

In this section the results from both the simulation as well as the experiments are presented. The performance of the controller was evaluated on basis of the requirements given in the introduction. To start with the ability of the controller to track a reference path was investigated. In these tests no obstacles were present.

1) *Reference tracking*: Figure 6 shows the path of the vehicle starting with a lateral offset of approximately 0.4 m from the reference line being the x axis. The experiments were

conducted for a set of different weights on the control input. To begin with the control correction variable was not limited, the prediction horizon was set to 25 and the mean speed, excluding the short time where the vehicle accelerates, was about 0.55 m/s. Only the lateral deviation was penalized with the weight $w_{e_y}=0.8$. The objective of this test was to investigate the effect different weights on the steering angle had on the performance.

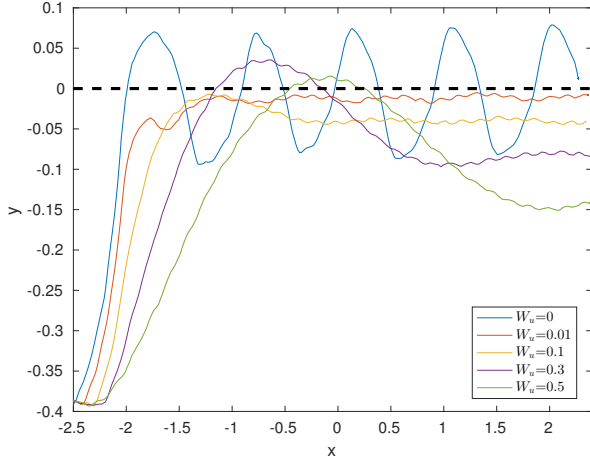


Fig. 6. Experiment showcasing the effect an increasing weight on the control input has on the ability of the vehicle to track the reference.

In the first test represented by the blue line the control input was not penalized but limited by the steering servo to a maximum of 30° . The result suggests that the vehicle travels with an oscillatory movement along the path. This is caused by the fact that since the input is not penalized even a small deviation from the desired path will result in the controller asking for maximum steering angle. This in combination with the fact that the steering servo in reality has a limitation on the change of rate cause the oscillations. If on the other hand the weight on the control input is increased the oscillations will be dampened. However, it is evident that even a too large value will result in oscillations although with longer periods. Note that the size of the oscillations visible in curves for $w_u = 0.01$ to $w_u = 0.3$ can be deceiving. In reality they are small and hardly visibly to the eye because of the small steering angles typically $1-2^\circ$. The results indicate that by choosing the weights carefully it might be possible to bring the vehicle to the reference path and stabilize it without taking the limitation in change of control input into account. Surely as can be seen in Figure 7 this is in fact possible. In this case $w_u=0.0015$, and the weight on the error in heading angle had to be increased to 0.03. From these results it is evident that the controller is indeed able to cope with a disturbance in the form of an initial lateral deviation from the path. Furthermore, the settling distance is just about the required 0.8 m. Hence, the vehicle fulfills the requirements although with no margin of safety.

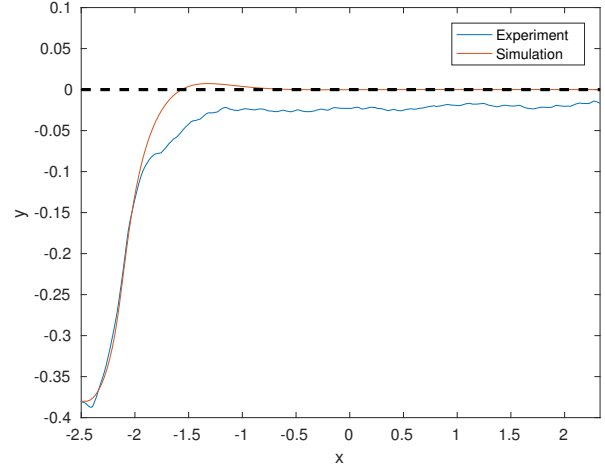


Fig. 7. Comparison of experiment and simulation, vehicle starting with a lateral offset of roughly 0.4 m.

In Figure 8 the control input and the error in heading angle as well as the velocity of the vehicle from both the experiment and simulation are presented. Since the vehicle does not accelerate in the simulation but starts instantly to travel with a constant velocity it reached the path faster. This could also explain why it changes steering earlier than in the experiment.

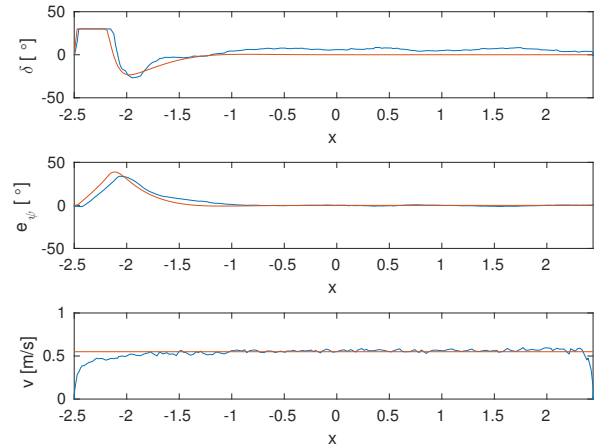


Fig. 8. Comparison of steering angle, error in heading angle and velocity of vehicle in both experiment (blue) and simulation (orange).

2) *Obstacle avoidance:* Having shown that the vehicle is able to successfully track a reference the next step is to introduce obstacles in its path. Three boxes in the shape of rectangles were used for this purpose.

Although the results so far suggest that it is possible to track a reference and stabilize the vehicle without taking the dynamics of the steering, namely limitation in change of rate of steering, into account this will as shown here pose a problem. In Figure 9 the path of the vehicle is illustrated for 3 different cases. While the weights on the control input as well as the states remain the same as in the previous tests i.e

$w_{e_y} = 0.8$, $w_{e_\psi} = 0$ and $w_u = 0.1$ the parameter that was changed was the limitation on the control correction as well as the prediction horizon. It is evident that an increase in Δu_{max} will cause a slower change of rate in the steering resulting in a larger lateral deviation when evading the obstacles. As the orange curve suggests the vehicle will at some point not be able to recover fast enough to avoid a collision with the next obstacle. An increase in prediction horizon is on the other hand enough to solve the problem.

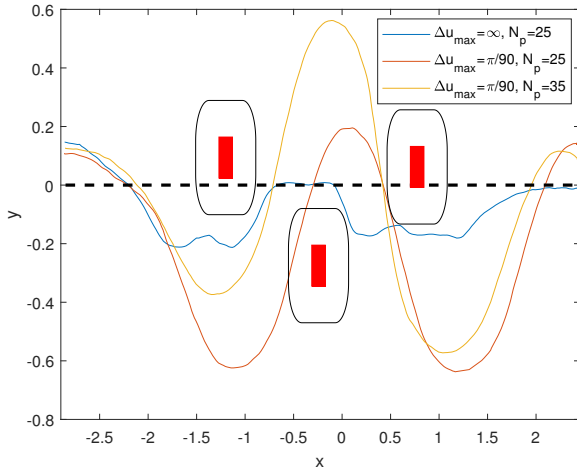


Fig. 9. Experiment with obstacles for showing the effect the limitation on the rate of change of the steering has on the vehicle. Red rectangles represent the obstacles while the line encircling them show the region which the center of gravity of the vehicle must not enter to avoid collision.

However, the increase in prediction horizon with 10 units comes with the cost of a rise in the time to solve the optimization problem. The average time increases from 28 ms to 37 ms. While the mean value is below the limit of 50 ms there were still peaks which surpassed it. In this case they were not many enough to affect the controller. However, if the prediction horizon is further increased the controller might not be able to solve the optimization problem in time which will result in no feedback and possibly a collision. With the same sets of parameters but an increase in the maximum allowed control correction to $\pi/60$ the vehicle was able to evade the obstacles. The results from the experiments along with the simulation are presented in Figure 10.

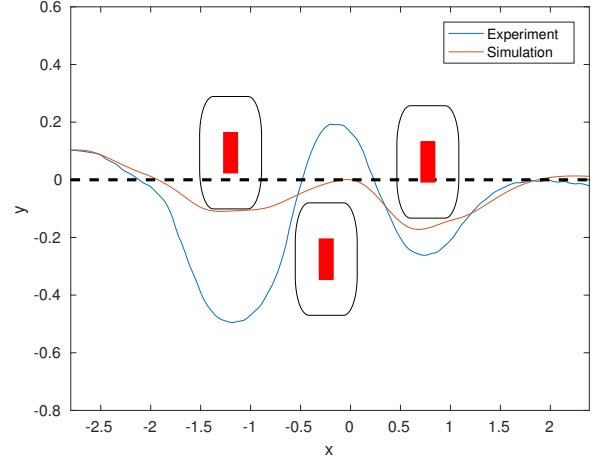


Fig. 10. Comparison between results obtained from experiment and simulation.

The obtained results suggest that although the vehicle was able to clear the obstacle there is a significant difference between what the simulation predicts and the experiment. While the simulation determines that the vehicle should be able to meet the requirements on maximum clearing distance from the obstacles and the over/undershoot this is obviously not the case in the experiment. The differences in the trajectories between the simulation and the experiments could be caused by modelling errors or inaccuracy in the linearized model that arise when the vehicle moves further away from the linearization point. While the wheelbase could be determined with a fairly high accuracy the maximum rate of change in the steering angle could not be decided with high accuracy. Therefore, it had to be estimated. Because of the uncertainty it is likely subjected to error and could therefore be a possible explanation to the differences in the different trajectories. To investigate the effect it has on the behaviour of the vehicle a series of tests were run in the simulation. To further ensure that the linearization is not to blame the tests were run with two different methods for linearization. The first being the linearization performed around a fixed point as described earlier while the second was performed around the last calculated state. As Figure 11 shows an increase in Δu_{max} results in larger oscillations when turning. If increased too much the vehicle started departing further away from the linearization point which resulted in the simulation crashing. On the other hand, if the the linearization was performed for every sampling instance the vehicle proved, as expected, to be able to cope with larger values on Δu_{max} . The difference in trajectories obtained with the two methods of linearization turned out not to be significant for $\Delta u_{max} = \pi/60$, the value used for the test shown in Figure 9. Moreover, the differences between the trajectories are most evident only for the last obstacle.

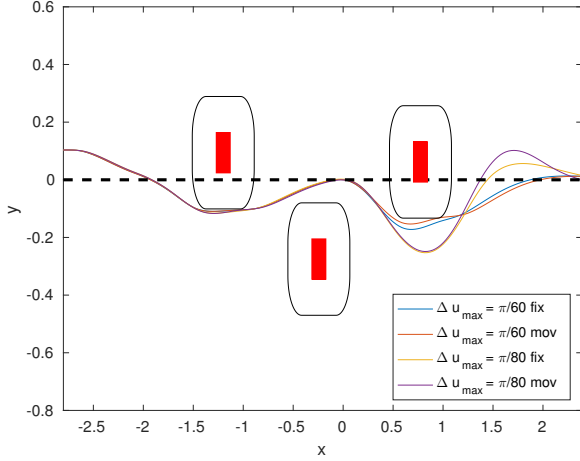


Fig. 11. Comparison of trajectories for linearization around fixed point and linearization around last calculated states for different limits on Δu_{max} .

VII. CONCLUSION

The objective of this thesis was to develop a controller to be used in an autonomous car for path tracking and obstacle avoidance. The vehicle should as put forward in this paper be able to deal with initial lateral deviation and recover to the desired path within a specified settling distance. All this while clearing the obstacles and at the same time keeping the distance to the reference path to a minimum. The results obtained from the simulations as well as the experiments show that the vehicle in general failed to do so. It did manage to fulfill the requirement on settling distance in absence of any obstacle. However, although it was able to clear the obstacles it did not respect the limit on the clearing distance from the obstacles. It was higher than necessary.

This should be regarded as a feasibility study or an investigation of the capabilities of a simple MPC. To determine whether or not it is possible to control a small scale vehicle using only a linear MPC, with no other components such as Kalman filter, through MATLAB. It can be concluded, on the basis of the obtained results, that even though it was able to track the path and avoid the obstacles it did not exhibit the level of accuracy required for a real life scenario. The main reasons why it is not suitable for controlling a vehicle in reality is partly because of the performance and the circumstances under which the tests were conducted. If scaled to the size of a real vehicle the results from the experiments suggest that a vehicle would clear obstacles with a distance several times larger than the width or length of the obstacle itself. Hence, the controller does not show the level of precision required for dealing with a traffic situation on a highway where the distance separating vehicles and/or obstacles is sometimes less than a meter. Moreover, in a real scenario when travelling at high speed one would like to include an additional control variable being the acceleration of the vehicle. Simply relying on lateral control may not be enough to avoid a collision without performing sudden turns at high speed which pose a danger not only to passengers but also to other vehicles. Therefore longitudinal control should also be

taken into account for this purpose. Another important point to be made here is the velocity with which the vehicle was travelling in the test, around 0.5 m/s. Although the kinematic vehicle model allows for significantly higher velocities before the model becomes too inaccurate it is not suitable to use for modelling high speed maneuvers. At that point lateral forces and interaction between the wheels and the ground would have to be modelled as well.

Nonetheless, the results should be regarded as relatively good for several reasons. First and foremost the vehicle that was used was an R/C car intended to be controlled manually by humans and not by a computer. Therefore one cannot expect the steering to perform with high accuracy. Secondly, the MPC is sensitive to modeling errors and uncertainties. Recall that the vehicle was controlled solely with a linear MPC neither a robust MPC nor a Kalman filter was used to deal with uncertainties and disturbances. Despite this the controller performed relatively well. If the accuracy of the controller, more specifically the large lateral deviations from the obstacles could be decreased it could possibly be used as an auto parking systems.

A. Future work

Although one can further develop the concept by for instance introducing a lidar device to detect and track the obstacles there is still room for improvements of the controller. First of all further investigation is needed to determine the underlying cause to the differences in the trajectories between simulation and experiment. For this setup the vehicle was controlled through MATLAB running on a separate computer. Even though the objective was met controlling the vehicle in this fashion proved not be optimal since MATLAB is relatively slow compared to other languages such as Python and C. Therefore with a more efficient solver running onboard the vehicle one can expect the solving time to drop and as a consequence of that the sampling time can be reduced while the prediction horizon can be increased for better performance. Using MATLAB for controlling a real vehicle is because of the computational burden not convenient.

In this case the velocity was considered to be constant while in reality it differed slightly at each time instance. Even if the controller handled this well it is worth considering adding the acceleration as an additional input to the system so that the velocity can be controlled. This would give the controller another control parameter to use in order to meet the objective.

B. Acknowledgment

First and foremost I would like to extend my sincere gratitude to professor Jonas Mårtensson for giving me the possibility to conduct this thesis and for supporting me. I would also like to thank Xiao Chen as well as Frank Jiang for helping me with the implementation. Their expertise and advice proved crucial for the success of this thesis. It has truly been a worthwhile experience. Last but not least thanks to my friends who have always been by my side throughout the duration of this thesis. In particular, I wish to thank my dear friend Federico Raiti who provided me with valuable advice and support through these hard times.

REFERENCES

- [1] H. Worley, "Road Traffic Accidents Increase Dramatically Worldwide," <http://www.prb.org/Publications/Articles/2006/RoadTrafficAccidentsIncreaseDramaticallyWorldwide.aspx>, 3 2006, [Online; accessed 22-January-2018].
- [2] A. D. L. Christopher J.L. Murray, *The Global Burden of Disease - A comprehensive assessment of mortality and disability from diseases, injuries, and risk factors in 1990 and projected to 2020*. Harvard School of Public Health, 1996.
- [3] S. Chang and T. J. Gordon, "A flexible hierarchical model-based control methodology for vehicle active safety systems," *Vehicle System Dynamics*, vol. 46, no. S1, pp. 63–75, 2008.
- [4] C. D. Harper, C. T. Hendrickson, and C. Samaras, "Cost and benefit estimates of partially-automated vehicle collision avoidance technologies," *Accident Analysis & Prevention*, vol. 95, pp. 104–115, 2016.
- [5] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "Mpc-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265–291, 2005.
- [6] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, M. Panciroli, and A. Prioletti, "Proud-public road urban driverless test: Architecture and results," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 648–654.
- [7] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "Stadtpilot: First fully autonomous test drives in urban traffic," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 919–924.
- [8] P. F. Lima, M. Trincavelli, M. Nilsson, J. Mårtensson, and B. Wahlberg, "Experimental evaluation of economic model predictive control for an autonomous truck," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 710–715.
- [9] Z. Shiller and Y.-R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, 1991.
- [10] Y. Kanayama and B. I. Hartman, "Smooth local path planning for autonomous vehicles," in *Autonomous robot vehicles*. Springer, 1990, pp. 62–67.
- [11] F. Kuhne, W. F. Lages, and J. G. da Silva Jr, "Model predictive control of a mobile robot using linearization," in *Proceedings of mechatronics and robotics*, 2004, pp. 525–530.
- [12] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and autonomous systems*, vol. 55, no. 6, pp. 460–469, 2007.
- [13] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3341–3346.
- [14] C. Liu, W.-H. Chen, and J. Andrews, "Optimisation based control framework for autonomous vehicles: Algorithm and experiment," in *Mechatronics and Automation (ICMA), 2010 International Conference on*. IEEE, 2010, pp. 1030–1035.
- [15] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.
- [16] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 1094–1099.
- [17] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 812–818.
- [18] P. F. Lima, "Optimization-based motion planning and model predictive control for autonomous driving: With experimental evaluation on a heavy-duty construction truck," Ph.D. dissertation, KTH Royal Institute of Technology, 2018.
- [19] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Boston, MA: Springer US, 2006.
- [20] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.
- [21] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 4136–4141.
- [22] Y. Gao, *Model predictive control for autonomous and semiautonomous vehicles*. University of California, Berkeley, 2014.
- [23] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [24] P. Li, X. Yue, G. Luo, and T. Ma, "Rigid spacecraft attitude reorientation control using receding horizon optimization," in *Control Conference (CCC), 2013 32nd Chinese*. IEEE, 2013, pp. 2221–2226.
- [25] B. P. Otta, "Numerical algorithms for quadratic programming for approximated predictive control," Ph.D. dissertation, Master Thesis, 2013.

TRITA EECS-EX-2018:654
ISSN 1653-5146