Advanced Methods for short-term Path Planning in Autonomous Driving based on Adaptive MPC

Alberto Franco

Department of Information Engineering
Università degli Studi di Padova
alberto.franco.3@studenti.unipd.it

Abstract—**In this paper we present two different strategies for a self-driving car short-term path planning. The aim of this project is to study and implement a path tracking framework in order to prevent ATLASCAR2 collision with moving obstacle vehicles. The proposed algorithms, based on the Model Predictive Control paradigm, solve optimal problems formulated in terms of cost minimization under constraints.**

I. Introduction

In robotic research, the problem of navigation is among the most important. Basically all autonomous mobile robots need some kind of navigation to fulfill the mobile term [1]. We understand 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 [2], [3]. The desired paths have to fulfill several conditions to ensure safety and feasibility of the navigation [4]. Moreover, the paths can be also compared in terms of desirability for example short or smooth paths are usually more desirable than long and curved ones. Such paths should therefore be preferred in the navigation process [5]. Beyond the path planning, the navigation problem also involves reacting to changes of the environment model. Robots are required to move towards 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 [6], [7], [8]. Though these two topics have been well researched, currently, there is no ideal solution to handling the navigation problem within cluttered dynamic environments. The proposed algorithms were studied for the ATLASCAR project in which the group for Robotics and Automation at the University of Aveiro, has setup and adapted a common commercial vehicle to provide a versatile framework to develop studies and research [9], [10], [11].

II. Context of the Problem and Proposed Solution

The ATLACAR2 has 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 slow-moving vehicle. 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, this vehicle is also equipped with a lane-keeping assist system: 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.

A. Theoretical background - Adaptive MPC

Model Predictive Control is an advanced method that predicts future behavior using a linear-time-invariant (LTI) dynamic model. These predictions are never 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. 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:

$$
\begin{aligned}
x(k+1) &= Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \\
y(k) &= Cx(k) + D_v v(k) + D_d d(k)
\end{aligned}
$$

where the matrices A, B_u, B_v, B_d, C, D_v and D_d can vary with time. The other parameters in the expression are:

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

In the adaptive MPC control there are additional requirements for the plant model as 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, $D_u = 0$ in the above model. Finally the input and output signal configuration remains constant. 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. In adaptive MPC, as time evolves you 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:

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

where the matrices A, B, C and D are updated with respect to time. The other parameters in the expression are:

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

The adaptive MPC uses a Kalman filter to update its controller states and adjusts the two required gain matrices, L and M, at each control interval to maintain consistency with the updated plant model. We are considering a linear-time-varying Kalman filter (LTVKF):

$$
\begin{aligned}
L_k &= \left(A_k P_{k|k-1} C_{m,k}^T + N\right)\left(C_{m,k} P_{k|k-1} C_{m,k}^T + R\right)^{-1} \\
M_k &= P_{k|k-1} C_{m,k}^T \left(C_{m,k} P_{k|k-1} C_{m,k}^T + R\right)^{-1} \\
P_{k+1|k} &= A_k P_{k|k-1} A_k^T - \left(A_k P_{k|k-1} C_{m,k}^T + N\right) L_k^T + Q.
\end{aligned}
$$

where

- Q, R and N are constant covariance matrices defined as in MPC state estimation;
- A_k and $C_{m,k}$ are state-space parameter matrices for the entire controller state;
- $P_{k|k-1}$ is the state estimate error covariance matrix at time k based on the information available at time $k-1$;
- L_k and M_k are the updated Kalmen filter gain matrices.

The gain and the state error covariance matrix depend upon the model parameters and the assumptions leading to the constant Q, R and N matrices.

III. Obstacle Avoidance

A. Problem Formulation

The collision avoidance problem is very dependent on the vehicle modeling since it is a requirement for adaptive MPC law design. The model used in this paper

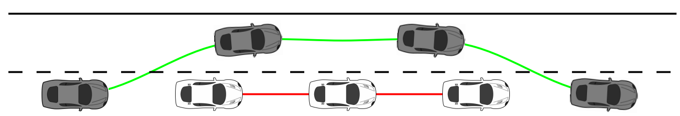

Fig. 1. Problem description of collision avoidance on road.

should take into account the kinematic and dynamic aspects of the vehicle. Here, we present a non linear mathematical model of a vehicle used for the development of a collision avoidance system. The model has four states and two inputs:

$$
\mathbf{x} = \begin{bmatrix} x & y & \theta & v \end{bmatrix}^T, \qquad \mathbf{u} = \begin{bmatrix} T & \delta \end{bmatrix}^T
$$

where (x, y) are the global coordinates of the center point of the car, θ is the heading angle of the car body with respect to the x-axis and v is the speed of the car (positive). The manipulated variables are T

the throttle (positive when accelerating/negative when braking) and δ the steering angle (0 when aligned with car, counterclockwise positive). The ATLASCAR2 can

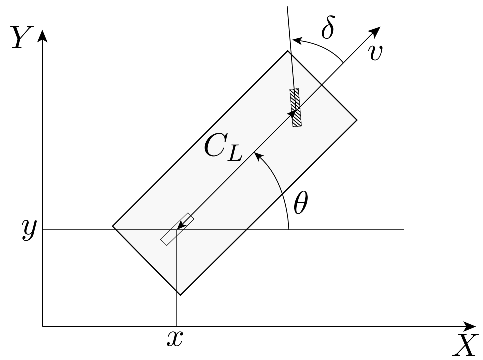

Fig. 2. Bicycle model of a car.

be modeled using the non-linear kinematic bicycle model described by the following equations of motion:

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \dfrac{v}{C_L}\tan(\delta) \\ \dot{v} = 0.5 \cdot T \end{cases} \implies \begin{array}{l} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = g(\mathbf{x}, \mathbf{u}) \end{array}$$

where C_L is the car length. In order to simplify the model, it is assumed that only the front wheel can be steered. Moreover, in this paper it is assumed that the ATLASCAR2 does not slip, any slippage is thus considered as an external disturbance. Under this assumption, the slip angle is zero, meaning that the velocity is directed along the heading of the vehicle. In order to use MPC, the state space model needs to be linearized and also re-written in a more compact form:

$$\begin{array}{l} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = g(\mathbf{x}, \mathbf{u}) \end{array} \implies \begin{array}{l} \dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u} \\ \mathbf{y} = C_c\mathbf{x} + D_c\mathbf{u} \end{array} \tag{1}$$

where the matrices A_c, B_c, C_c and D_c are obtained as follows:

$$A_c = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & -v\sin(\theta) & \cos(\theta) \\ 0 & 0 & v\cos(\theta) & \sin(\theta) \\ 0 & 0 & 0 & \tan(\delta)/C_L \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B_c = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \dfrac{v}{C_L}\big(\tan(\delta)^2 + 1\big) \\ 0.5 & 0 \end{bmatrix},$$

$$C_c = \frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = \mathbf{I}_4, \qquad D_c = \frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = \mathbf{0}_{4\times 2}.$$

The simple linearized approximation of the system to describe the dynamics of the ATLASCAR2 will be evaluated at the operating conditions; the extended form of the considered model is the following:

$$\begin{cases} \dot{x} = -v\sin(\theta)\cdot\theta + \cos(\theta)\cdot v \\ \dot{y} = v\cos(\theta)\cdot\theta + \sin(\theta)\cdot v \\ \dot{\theta} = (\tan(\delta)/C_L)\cdot v + \left(v\left(\tan(\delta)^2 + 1\right)/C_L\right)\cdot\delta \\ \dot{v} = 0.5\cdot T \end{cases}$$

The system to be controlled is usually modeled by a discrete state-space model in the MPC literature. Therefore, (1) is transformed into a discrete state-space model to be used by the Model Predictive Controller:

$$\begin{array}{l} \dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u} \\ \mathbf{y} = C_c\mathbf{x} + D_c\mathbf{u} \end{array} \Rightarrow \begin{array}{r} \mathbf{x}(k+1) = A_d\mathbf{x}(k) + B_d\mathbf{u}(k) \\ \mathbf{y}(k) = C_d\mathbf{x}(k) + D_d\mathbf{u}(k) \end{array}$$

where A_d and Bd are the state and control matrices for the discrete state-space equation, respectively, which can be calculated with the Euler method as

$$A_d = e^{A_c T_s}, \qquad B_d = \int_{kT_s}^{(k+1)T_s} e^{A_c[(k+1)T_s - \eta]} B_c d\eta$$

where T_s is the sampling interval for the discrete state-space model. The matrices C_d and D_d are equivalent to those in the continuous case. For simplicity, we assume that all the states are measurable and the ATLASCAR2 drives east with a constant speed at the nominal operating point. In the scenario that we are going to consider, the road is straight and our vehicle stays in the middle of the center lane when not passing. Without losing generality, the ATLASCAR2 passes an obstacle both to the right and to the left lane depending on where it is placed on the road. We create also a safe zone around the obstacles so that the vehicle does not get too close to the osbtacle when passing it.

B. Design of Adaptive Model Predictive Control

We designed a Model Predictive Controller that can make the ATLASCAR2 mantain a desired velocity of and stay in the middle of center lane. We used an Adaptive MPC controller because it handles the nonlinear vehicle dynamics more effectively than a traditional MPC controller; in fact, the latter uses a constant plant model but the former allows us to provide a new plant model at each control interval. Because the new model describes the plant dynamics more accurately at the new operating condition, an adaptive MPC controller performs better than a traditional MPC controller. In practice at each control interval, the adaptive MPC controller updates the plant model and the nominal conditions. Once updated, the model and the conditions remain constant over the prediction horizon. In path tracking using adaptive MPC, it is common to formulate the constrained control problem as a real-time optimization problem subject to hard constraints on plant variables and soft constraints on outputs; at the beginning, we specified the constraints for the manipulated variables: to prevent the ATLASCAR2 from accelerating or decelerating too quickly, we added an hard constraint on the throttle rate of change and another one on the steering angle rate of change. We used an approach that takes advantage of the ability of MPC to handle constraint explicitly. When an obstacle is detected, it defines an area on the road (in terms of constraints) that the ATLASCAR2 must not enter during the prediction horizon. At the next control interval the area is redefined based on the new positions of the vehicle and the obstacle until passing is completed. To define the area to avoid, we used the following mixed Input/Output constraints:

$$E\mathbf{u} + F\mathbf{y} \leq G$$

where **u** and **y** are respectively the manipulated variable vector and the output variable vector, while E, F, G are the constraint matrices that can be updated when the controller is running:

- the first constraint is an upper bound on the y-coordinate (right boundary of the road);
- the second constraint is a lower bound on the y-coordinate (left boundary of the road);
- the third constraint is for the obstacle avoidance; even though no obstacle is detected at the nominal condition, we must add this fake constraint here because we cannot change the dimensions of the constraint matrices at run time (it has the same form as the second one);
- the fourth constraint is an upper bound on the x-coordinate (the position of the closest obstacle);
- the fifth constraint is a lower bound on the x-coordinate (the position of the ATLASCAR2).

The matrices for the above inequality are the following:

$$E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \; F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ cS & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \; G = \begin{bmatrix} W/2 \\ W/2 \\ -cI \\ x_{\max} \\ x_{\min} \end{bmatrix}$$

where W is the width of the road, cI and cS are the required parameters such that the ATLASCAR2 must be above the line formed from the vehicle to safe zone corner for left/right passing and finally $x_{\max}$ represents the position of the closest obstacle while $x_{\min}$ depicts the location of our vehicle.

C. Simulation Results

The performances of the proposed adaptive MPC based vehicle control method are demonstrated in three simulation examples. We tried to choose parameters that were as close as possible to a real situation: the sampling time used in the discretization of the system is $T_s = 0.02$s while the values of the prediction and the control horizon are respectively $p_H = 25$ and $c_H = 5$. In all simulations, the distance between the front and rear axles is $C_L = 5$m and the witdh of the vehicle is $C_W = 2$m. The saturation ranges of the control inputs are: the steering angle lies in $\left[-\frac{\pi}{30}, +\frac{\pi}{30}\right]$rad/sec while in order to prevent the ATLASCAR2 from accelerating or decelerating too quickly, we impose an hard constraint of 2.5m^2/sec on the throttle rate of change. Moreover we are using a constant reference signal for the velocity of $v = 20$m/s ($\simeq 72$km/h). The vehicle is initially placed at $(0, 0)$ and moves in a straight line.

1) One Moving Obstacle - Right Overtaking: In this first simulation we assumed that the road is straight and has 3 lanes each one 4 meters wide. Moreover the ATLASCAR2 drives in the middle of the center lane while the road is completely free and when there is an obstacle the vehicle passes it only using the right lane (the same simulation can be launched so that the car goes over to the left fast lane). When the vehicle detects the obstacle, the constraints are computed: if the ATLASCAR2 is already in the adjacent lane, it uses the safety zone as the constraint, otherwise the vehicle must be above the line formed from the ATLASCAR2 to safe zone corner for right passing. If the vehicle is parallel to the obstacle, it uses the safety zone as the constraint and finally if it has passed the obstacle, it uses the inactive constraint to go back to the center lane. In this first example the obstacle moves in the same direction as the vehicle.

2) Multiple Moving Obstacles: Consequently, additional obstacles were added to make the scenario more complex. The vehicle is capable of overtaking the obstacles to the right or left depending on their positions with respect to the road. If the y-coordinate of the closest obstacle is greater than 0, then the vehicle overtakes to the right, while if the y-coordinate is less than 0 the overtaking takes place on the left lane. We also hypothesized that the obstacles move at a different speeds but that they are initially at a common distance. In case two obstacles are too close during the simulation

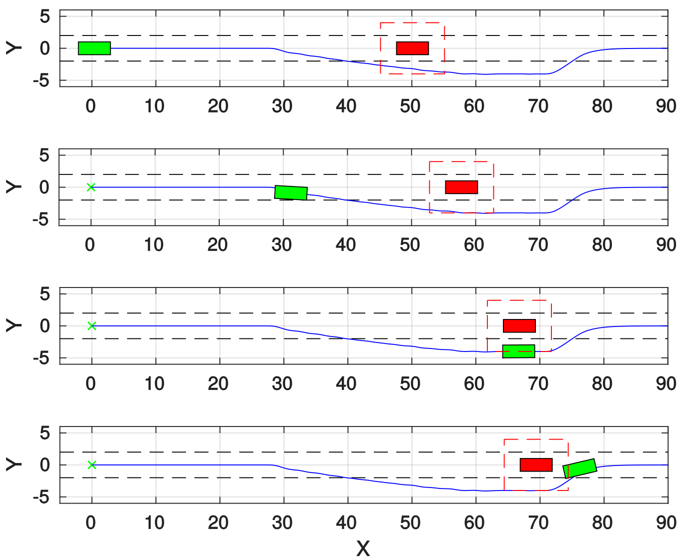

Fig. 3. Simulation of right overtaking with one moving obstacle

and their distance is less than the detection range which is 30m, the ATLASCAR2 perceives the objects as a single entity and adapts to the situation. In this case the obstacles drive in the wrong direction with respect to the ATLASCAR2 (but the same results can be verified in the same direction).

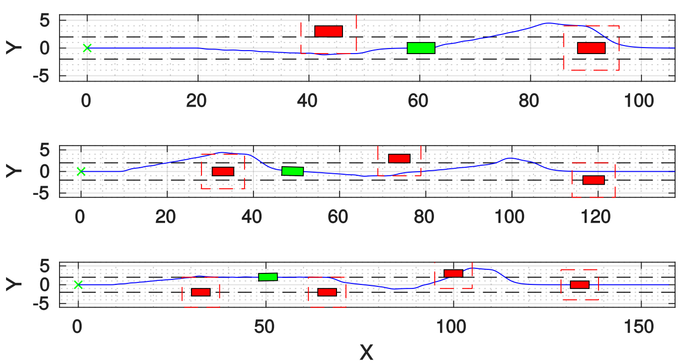

Fig. 4. Simulations of overtaking with $N = 2, 3, 4$ moving obstacles

3) Vehicle Braking and Obstacles Overtaking: Finally we have improved the code related to the mixed Input/Output constraints so that in the case there are 3 obstacles that block the road and drive at a lower speed than the ATLASCAR2, the velocity of the vehicle decreases in order to prevent the collision. Here it is depicted a simulation in which there are the obstacles at the same x-coordinate. Two obstacles have a constant speed of 8m/s while the one on the left lane has a velocity of 15m/s. At the beginning the vehicle detects all the other cars on the road. In the first part of the simulation, the vehicle brakes because there is no enough space for overtaking the cars as shown in the Fig.6. A collision will happen if the vehicle continues to follow the initially planned trajectory with the reference velocity. It is possible to notice that the speed decreases because the applied throttle is negative, so a consistent deceleration is set after $\simeq$ 1.5s as depicted in Fig.7. After few seconds the fastest obstacle moves and makes available the left lane. Dramatic changes of steering angle in early stage are observed in Fig.8 and consequently also on the heading angle in Fig.9. Then, the ATLASCAR2 returns to the reference velocity during the overtaking of the two obstacles. The path of the vehicle is is depicted in Fig. 10. It is seen that the ATLASCAR2 avoids the obstacles and returns to the road centerline with a low overshoot.

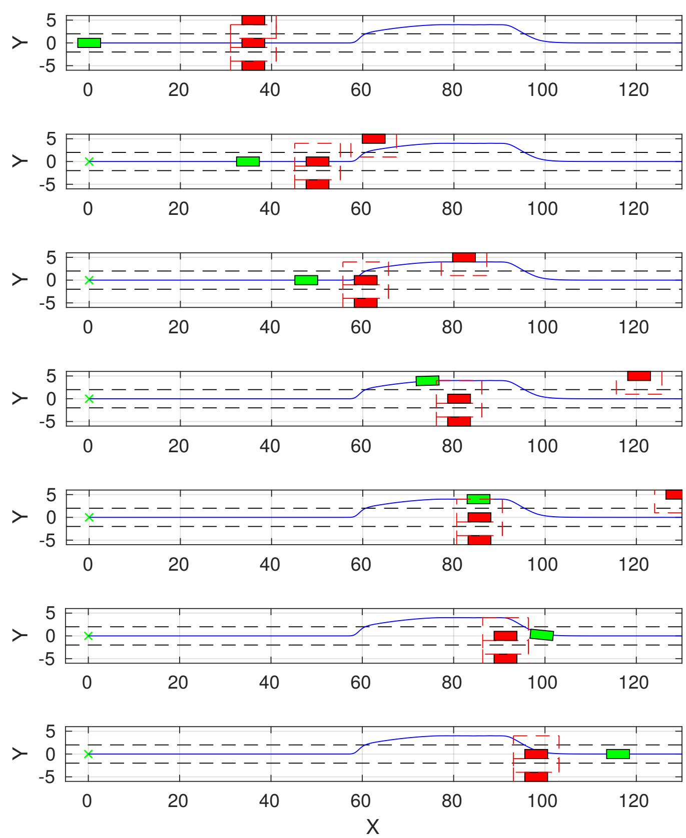

Fig. 5. Simulation of braking and overtaking obstacles.

IV. Lane Following

A. Problem Formulation

A lane-following system is a control system that keeps the vehicle traveling along the centerline of a highway

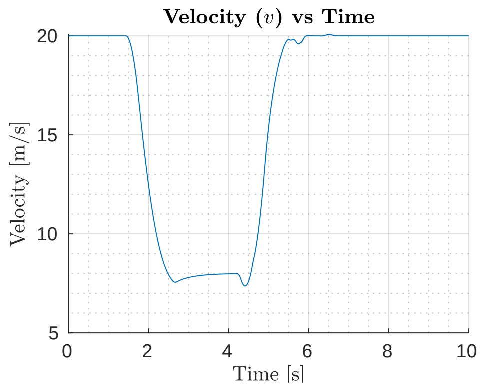

Fig. 6. Velocity v with respect to time.

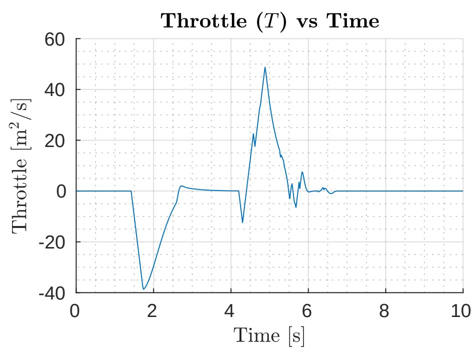

Fig. 7. Throttle T with respect to time.

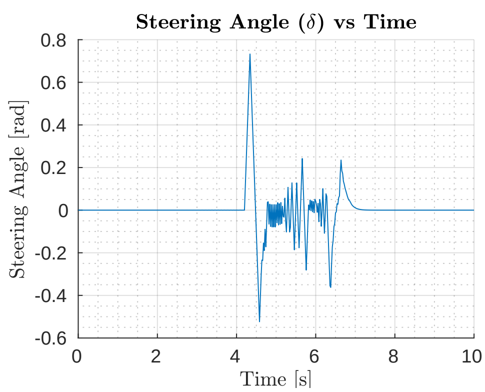

Fig. 8. Steering angle δ with respect to time.

lane, while maintaining a user-set velocity. The lane-following scenario is depicted in the following figure:

In a classic lane keeping assist, it is assumed that the longitudinal velocity is constant. This restriction is relaxed in this model because the longitudinal acceleration varies in this MIMO control system. This lane-following system manipulates both the longitudinal acceleration and the front steering angle of the vehicle

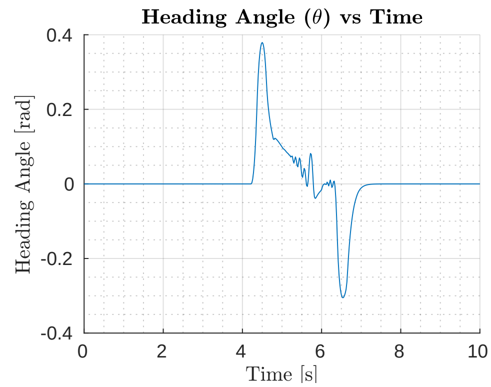

Fig. 9. Heading angle θ with respect to time.

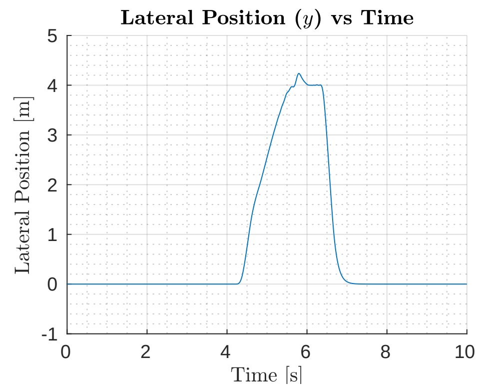

Fig. 10. Lateral position y with respect to time.

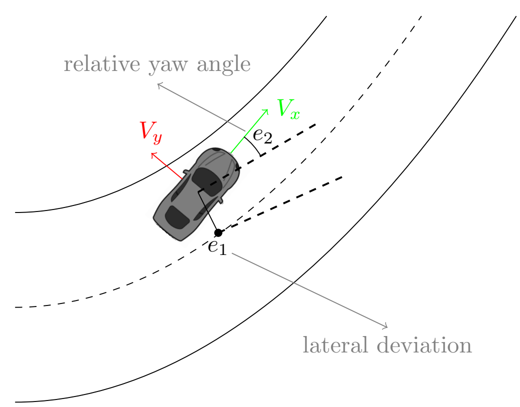

Fig. 11. Problem description of a lane following system.

to keep the lateral deviation and the relative yaw angle small and the longitudinal velocity close to a driver set velocity. If these two goals cannot be met at the same moment, the system tries to balance them. The model that we are considering, contains a lot of parameters. The first fundamental block describes the vehicle dynamics: we have applied the bicycle model of lateral vehicle dynamics and approximate the longitudinal dynamics using a time constant:

1) Longitudinal dynamics: We can use the following state space to describe the longitudinal model:

$$\begin{aligned} \dot{\mathbf{x}}_{\text{lon}} &= A_m \mathbf{x}_{\text{lon}} + B_m \mathbf{u}_{\text{lon}} \\ \mathbf{y}_{\text{lon}} &= C_m \mathbf{x}_{\text{lon}} + D_m \mathbf{u}_{\text{lon}} \end{aligned} \tag{2}$$

where the input is the acceleration and the states are the longitudinal velocity and the actual acceleration which is also the only output of this system.

$$\mathbf{x}_{\text{lon}} = \begin{bmatrix} \dot{V}_x & V_x \end{bmatrix}^T, \qquad \mathbf{u}_{\text{lon}} = a$$

and

$$A_g = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix}, \quad B_g = \begin{bmatrix} \frac{1}{\tau} \\ 0 \end{bmatrix},$$

$$C_g = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D_g = 0.$$

where τ is a time constant; in practice we are considering a second order transfer function like in [14].

2) Lateral dynamics: Local function: we have a continuous vehicle lateral model from parameters obtained by simplifying the one in [25]:

$$\begin{aligned} \dot{\mathbf{x}}_{\text{lat}} &= A_g \mathbf{x}_{\text{lat}} + B_g \mathbf{u}_{\text{lat}} \\ \mathbf{y}_{\text{lat}} &= C_g \mathbf{x}_{\text{lat}} + D_g \mathbf{u}_{\text{lat}} \end{aligned} \tag{3}$$

where the input is the steering angle in radians, and the outputs are the lateral velocity in meters per second and yaw angle rate in radians per second:

$$\mathbf{x}_{\text{lat}} = \begin{bmatrix} V_y & \dot{\psi} \end{bmatrix}^T \qquad \mathbf{u}_{\text{lat}} = \delta$$

and

$$A_g = \begin{bmatrix} -\frac{2C_F + 2C_R}{mV_x} & -\frac{2C_F l_F - 2C_R l_R}{mV_x} - V_x \\ -\frac{2C_F l_F - 2C_R l_R}{I_Z V_x} & -\frac{2C_F l_F^2 + 2C_R l_R^2}{I_Z V_x} \end{bmatrix},$$

$$B_g = \begin{bmatrix} 2C_F/m \\ 2C_F l_F / I_Z \end{bmatrix}, \quad C_g = \mathbf{I}_2, \quad D_g = \mathbf{0}_{2\times 1}.$$

The parameters in the previous matrices are:

- V_x is the longitudinal velocity of the car;
- m is the total mass parameter;
- I_Z is the yaw moment of inertia parameter;
- l_F and l_R are the longitudinal distances from center of gravity to front and rear tires parameters;
- C_F and C_R are the cornering stiffnesses of front and rear tires parameters.

The goal for the driver steering model is to keep the vehicle in its lane and follow the curved road by controlling the front steering angle . This goal is achieved by driving the yaw angle error $e_2 = \psi - \psi_{\text{des}}$ and lateral displacement error e_1 to zero ($\dot{e}_1 = V_x e_2 + V_y$). We can incorporate these two paramenters in the augmented model:

$$\begin{aligned} \dot{\mathbf{x}}_{\text{aug}} &= A_a \mathbf{x}_{\text{aug}} + B_a \mathbf{u}_{\text{aug}} \\ \mathbf{y}_{\text{aug}} &= C_a \mathbf{x}_{\text{aug}} + D_a \mathbf{u}_{\text{aug}} \end{aligned} \tag{4}$$

where

$$\mathbf{x}_{\text{aug}} = \begin{bmatrix} V_y & \dot{\psi} & e_1 & e_2 \end{bmatrix}^T, \qquad \mathbf{u}_{\text{aug}} = \begin{bmatrix} \delta & \dot{\psi}_{\text{des}} \end{bmatrix}^T$$

and

$$A_a = \begin{bmatrix} A_g & \mathbf{0}_{2\times 2} \\ \mathbf{I}_2 & \begin{matrix} 0 & V_x \\ 0 & 0 \end{matrix} \end{bmatrix}, \quad B_a = \begin{bmatrix} B_g & \mathbf{0}_{2\times 1} \\ 0 & 0 \\ 0 & -1 \end{bmatrix},$$

$$C_a = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{I}_2 \end{bmatrix}, \quad D_a = \mathbf{0}_{2\times 2}.$$

Combining (2) with (4) yields the state-space model that characterizes the Model Predictive Controller:

$$\begin{aligned} \dot{\mathbf{x}}_{\text{tot}} &= A_f \mathbf{x}_{\text{tot}} + B_f \mathbf{u}_{\text{tot}} \\ \mathbf{y}_{\text{tot}} &= C_f \mathbf{x}_{\text{tot}} + D_f \mathbf{u}_{\text{tot}} \end{aligned} \tag{5}$$

where

$$\mathbf{x}_{\text{tot}} = \begin{bmatrix} \dot{V}_x \\ V_x \\ V_y \\ \dot{\psi} \\ e_1 \\ e_2 \end{bmatrix}, \qquad \mathbf{u}_{\text{tot}} = \begin{bmatrix} a \\ \delta \\ \dot{\psi}_{\text{des}} \end{bmatrix}$$

and

$$A_f = \begin{bmatrix} A_m & \mathbf{0}_{2\times 4} \\ \mathbf{0}_{4\times 2} & A_a \end{bmatrix}, \quad B_f = \begin{bmatrix} B_m & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{4\times 1} & B_a \end{bmatrix},$$

$$C_f = \begin{bmatrix} C_m & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{2\times 2} & C_a \end{bmatrix}, \quad D_f = \mathbf{0}_{3\times 3}.$$

However the system to be controlled is usually modeled by a linear discrete state-space model:

$$\begin{aligned} \mathbf{x}_{\text{tot}}(k+1) &= A\mathbf{x}_{\text{tot}}(k) + B\mathbf{u}_{\text{tot}}(k) \\ \mathbf{y}_{\text{tot}}(k) &= C\mathbf{x}_{\text{tot}}(k) + D\mathbf{u}_{\text{tot}}(k) \end{aligned}$$

where A and B are the state and control matrices for the discrete state-space equation, respectively, which can be calculated, also in this case, with the Euler method as:

$$A = e^{A_f T_s}, \qquad B = \int_{kT_s}^{(k+1)T_s} e^{A_f[(k+1)T_s - \eta]} B_f d\eta$$

where T_s is the sampling interval for the discrete state-space model. The matrices C and D are equivalent to those in the continuous case.

B. Design of Adaptive Model Predictive Control

We created an Adaptive MPC controller with a prediction model that has six states, three outputs (longitudinal velocity, lateral deviation, relative yaw angle), and two inputs. The model has two manipulated signals: acceleration and steering. The product of the road curvature and the longitudinal velocity is modeled as a measured disturbance. We have set the constraints for manipulated variables and the scale factors. Moreover we have specified the weights in the standard MPC cost function. The third output, yaw angle, is allowed to float because there are only two manipulated variables to make it a square system. In this controller, there is no steady-state error in the yaw angle as long as the second output, lateral deviation, reaches 0 at steady state. Finally we have also penalized acceleration change more for smooth driving experience. This controller uses a linear model for the vehicle dynamics and updates the model online as the longitudinal velocity varies.

C. Simulation Results

The proposed adaptive MPC algorithm is designed in the MATLAB/Simulink and validated through simulation. The objective of this test is to evaluate the behavior of the proposed control strategy in critical situations. Table 3 shows the parameters used in the lane following simulation.

Parameters	Values
m	1575 kg
I_z	2875 kg·m^2
l_F	1.2 m
l_R	1.6 m
C_F	19000 N/rad
C_R	33000 N/rad
τ	0.2
V_0	15 m/s
V_{set}	20 m/s
T_s	0.02 s

Fig.12 and Fig.13 show the desired path that the car must follow and its curvature, where the former is described in terms of the lateral position Y_{ref} as function of the longitudinal position X_{ref}. The ATLASCAR2 is controlled to follow a sinusoidal trajectory which is given as follows:

$$X_{\text{ref}} = V_x \cdot t, \quad Y_{\text{ref}} = 5\sin(X_{\text{ref}}/20) \quad \text{with} \quad t \in [0, 20]\text{s}$$

Moreover the following figures show the trend of the main parameters confirming that the control strategy used allows the vehicle to follow the path. In particular

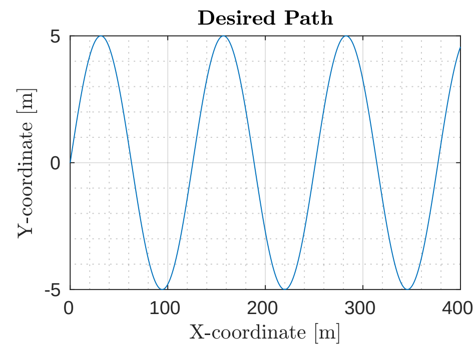

Fig. 12. Desired path of the ATLASCAR2.

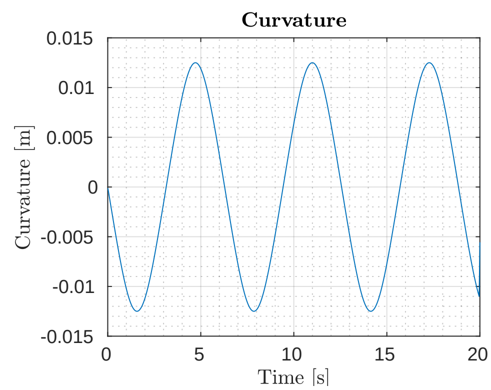

Fig. 13. Desired curvature.

we simulated also a small error in the sensor dynamics in order to make the simulation more realistic: we added a 3 percent error to the longitudinal velocity and this is evident from the small noise in the graphs of the steering angle and the lateral deviation.

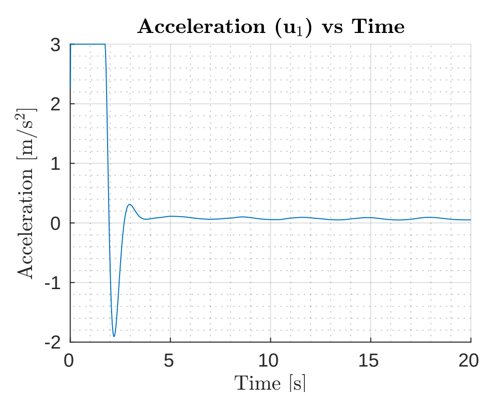

Fig. 14. Acceleration u_1 with respect to time.

Fig.18 shows the evolution of the vehicle longitudinal velocity. At the start of the simulation, this velocity is equal to the initial condition for longitudinal velocity

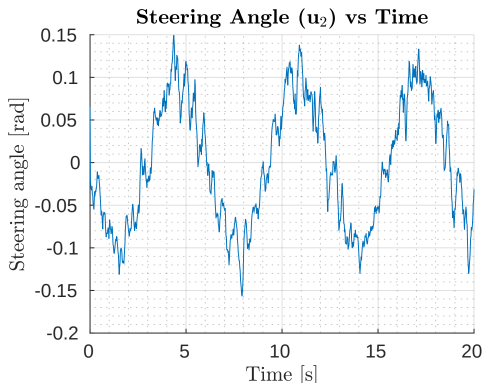

Fig. 15. Steering angle u_2 with respect to time.

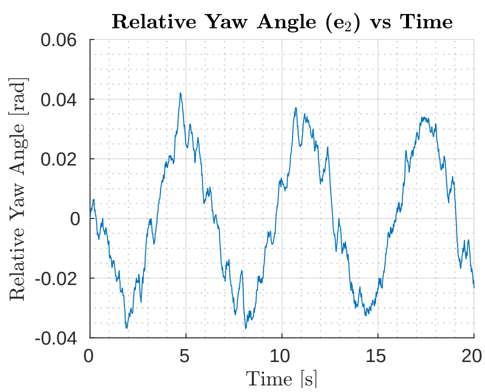

Fig. 16. Relative yaw angle e_2 with respect to time.

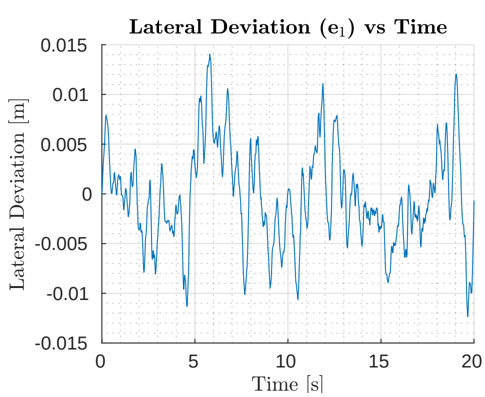

Fig. 17. Lateral deviation e_1 with respect to time.

parameter. At run time, this velocity is equal to the Longitudinal velocity input signal. We can note that V_x reaches the predefined value of 20 meters for second and then it stabilizes near the cruising speed because it continues to vary the steering angle to adapt to the path to be followed.

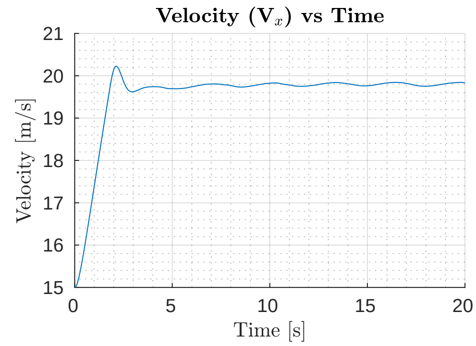

Fig. 18. Longitudinal velocity V_x with respect to time.

V. Conclusions and Future Work

In this work we proposed two advanced methods for the ATLASCAR2 based on adaptive Model Predictive Control. In the first case we analyzed an obstacle avoidance system that moves the vehicle around different moving obstacles in the lane using throttle and steering angle. This system updates both the predictive model and the mixed input/output constraints at each control interval. The vehicle is also able to brake in order to prevent collisions against closest obstacles. Instead, in the second scenario we have developed a lane following system that keeps the ATLASCAR2 traveling along the centerline of the lanes on the road by adjusting the front steering angle of the car. The flexibility of the concepts used in these methods allows a multitude of refinements and extensions to this project. For example the next possible work could be to combine the two previous control strategies in a way that they can operate simultaneously. Moreover the final method could be tested in a real autonomous driving scenario.

References

[1] J. Škoda, '3D Navigation for Mobile Robots', Dissertation, 2017.

[2] Y. Nishio, K. Nonaka and K. Sekiguchi, "Moving obstacle avoidance control by fuzzy potential method and model predictive control," 2017 11th Asian Control Conference (ASCC), Gold Coast, QLD, 2017, pp. 1298-1303.

[3] Yu, S., Li, X., Chen, H. and Allgöwer, F. (2015), "Nonlinear model predictive control for path following problems". Int. J. Robust Nonlinear Control, 25: 1168–1182.

[4] T. Xu and H. Yuan, "Autonomous vehicle active safety system based on path planning and predictive control," 2016 35th Chinese Control Conference (CCC), Chengdu, 2016, pp. 8889-8895.

[5] M. Nolte, M. Rose, T. Stolte and M. Maurer, "Model predictive control based trajectory generation for autonomous vehicles — An architectural approach," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 798-805.

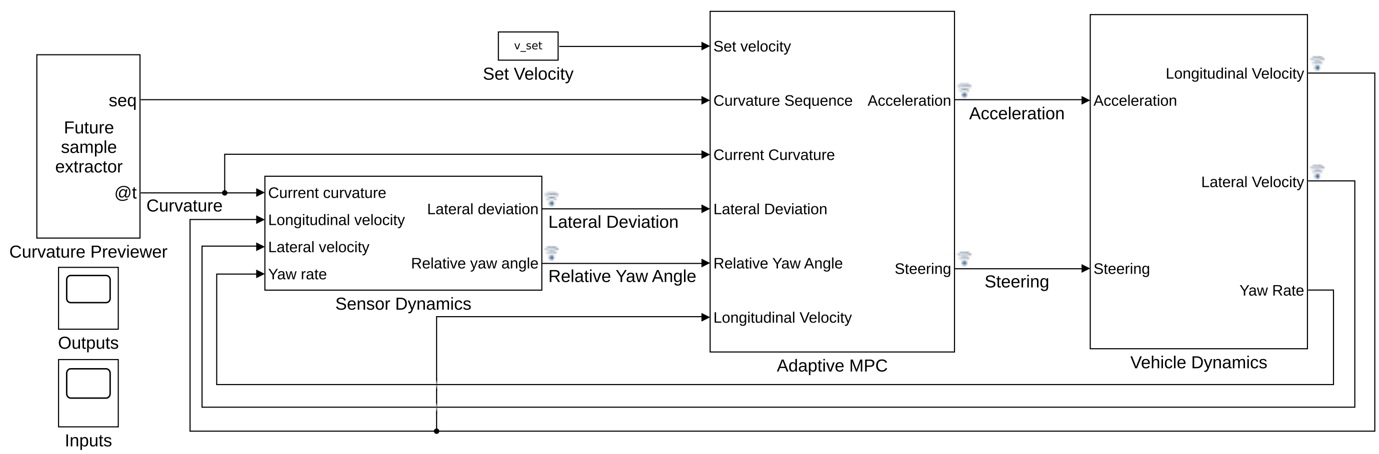

Fig. 19. Overall procedure scheme lane following.

[6] J. Ji, A. Khajepour, W. W. Melek and Y. Huang, ”Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints,” in IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 952-964, Feb. 2017.

[7] J. V. Frasch et al., ”An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles,” 2013 European Control Conference (ECC), Zurich, 2013, pp. 4136-4141.

[8] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson and F. Borrelli, ”Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads,” 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, 2013, pp. 378-383.

[9] ATLAS Project, Aveiro University, Portugal, http://atlas.web.ua.pt/, last visited July 2010.

[10] V. Santos et al., ”ATLASCAR - technologies for a computer assisted driving system on board a common automobile”, 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, 2010, pp. 1421-1427.

[11] V. Santos, ”ATLASCAR: A Sample of the Quests and Concerns for Autonomous Cars”, 14th International Conference, ICINCO 2017 Madrid, 2017.

[12] M. Werling and D. Liccardo, ”Automatic collision avoidance using model-predictive online optimization,” 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, 2012, pp. 6309-6314.

[13] Wei Xi and J. S. Baras, ”MPC based motion control of car-like vehicle swarms,” 2007 Mediterranean Conference on Control & Automation, Athens, 2007, pp. 1-6.

[14] Y. Wang, Y. Bin and K. Li, ”Longitudinal acceleration tracking control of low speed heavy-duty vehicles,” in Tsinghua Science and Technology, vol. 13, no. 5, pp. 636-643, Oct. 2008.

[15] A. Bemporad, M. Morafi, and N. Ricker, Model Predictive Control Toolbox User’s Guide, The MathWorks, Inc., https://www.mathworks.com/help/pdf_doc/mpc/mpc_ug.pdf

[16] J. Rawlings, D. Mayne (2010), ”Model Predictive Control: Theory and Design”, Nob Hill Publishing.

[17] C. Son and C. Ahn, ”Vehicle backward driving control with obstacle avoidance,” 2015 15th International Conference on Control, Automation and Systems (ICCAS), Busan, 2015, pp. 1771-1774.

[18] H. Zhou, L. Güvenç and Z. Liu, ”Design and evaluation of path following controller based on MPC for autonomous vehicle,” 2017 36th Chinese Control Conference (CCC), Dalian, 2017, pp. 9934-9939.

[19] Z. T. Yu, Obstacle Avoidance for Autonomous Vehicle Driving on Highways Based on MPC, JinLin University, 2012.

[20] M. Riccardo and S. Stefano, Integrated Control of Active Steering and Electronic Differentials in Four Wheel Drive Vehicles, SAE International Journal of Passenger Cars: Electronic and Electrical Systems, Vol.2, No.1, 141-149, 2009.

[21] R. Mikumo and H. Ichihara, ”Dynamic collision avoidance among multiple mobile robots: A model predictive control approach,” 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, 2017, pp. 1136-1137.

[22] X. Qian, A. de La Fortelle and F. Moutarde, ”A hierarchical Model Predictive Control framework for on-road formation control of autonomous vehicles,” 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, 2016, pp. 376-381.

[23] Berger, Niels. (2018). Lane Change Path Planning With state-dependent safety constraints using Nonlinear Model Predictive Control.

[24] E. Thilén, ‘Robust Model Predictive Control for Autonomous Driving’, Dissertation, 2017.

[25] Murali Madhavan Rathai, Karthik & Amirthalingam, Jegan & Jayaraman, Balaji. (2017). Robust tube-MPC based lane keeping system for autonomous driving vehicles. 1-6.

[26] B. Nordell, ‘Trajectory Planning for Autonomous Vehicles and Cooperative Driving’, Dissertation, 2016.

[27] C. Shen, H. Guo, F. Liu and H. Chen, ”MPC-based path tracking controller design for autonomous ground vehicles,” 2017 36th Chinese Control Conference (CCC), Dalian, 2017, pp. 9584-9589.

[28] N. Wada and T. Matsumoto, ”Driver assistance for collision avoidance by constrained MPC,” 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, 2017, pp. 90-93.