Short-term Path Planning with Collision Avoidance in Autonomous Driving based on Adaptive MPC

Alberto Franco
Department of Information Engineering,
Università degli Studi di Padova, Italy
alberto.franco.3@studenti.unipd.it
alberto.franco@ua.pt

Vitor Santos
Department of Mechanical Engineering,
IEETA, Universidade de Aveiro, Portugal
vitor@ua.pt

***Abstract*—In this paper we present a different strategy for a self-driving car short-term path planning in space-based approach. 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 algorithm, based on the Model Predictive Control paradigm, solves an optimal problem formulated in terms of cost minimization under constraints. Results demonstrate and verifies the feasibility and the usefulness of the method considering different scenarios.**

***Keywords*—autonomous driving, Model Predictive Control, obstacle avoidance.**

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 specifications 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 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 [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.

II. Context of the Problem and Proposed Solution

The proposed algorithm was 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]. In Fig.1 is reported the chosen platform which is an electric car, a Mitsubishii-MiEV of 2015 with a range of a 100km. The fact that the vehicle is electric allows to use the energy stored in the batteries, making it easier to power the sensors installed.

Fig. 1. ATLASCAR2, based on an electric car, a Mitsubishii-MiEV

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.

III. Theoretical background
Adaptive Model Predictive Control

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 [15]. 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:

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

where the matrices A, B_u, B_v, B_d, C, D_v and D_d can vary with time. The other parameters in (1) 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 (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, $D_u = 0$ 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) [15]. 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:

$$\begin{aligned} x(k+1) &= \overline{x} + A(x(k) - \overline{x}) + B(u_t(k) - \overline{u}_t) + \overline{\Delta x} \\ y(k) &= \overline{y} + C(x(k) - \overline{x}) + D(u_t(k) - \overline{u}_t) \end{aligned} \tag{2}$$

where the matrices A, B, C and D are updated with respect to time. The other parameters in (2) 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 which include the plant, the disturbance and measurement noise model states.

IV. Moving 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. In Fig.2 is depicted a typical scenario of overtaking of a moving obstacle.

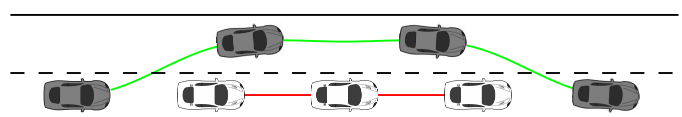

Fig. 2. Problem description of collision avoidance on road.

The model used in this paper 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 linear 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). Fig.3 shows a schematic sketch of the applied nonlinear bicycle model and the related parameters.

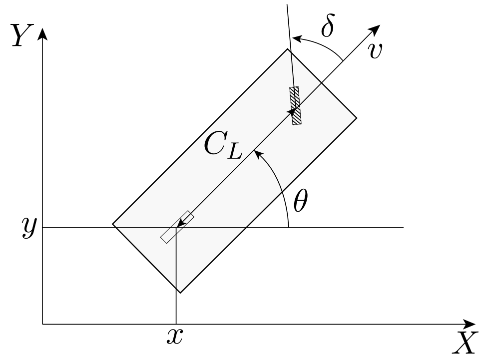

Fig. 3. Bicycle model of a car.

The ATLASCAR2 can be modeled using the non-linear kinematic bicycle model described by the following equations of motion [4] [13]:

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

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, so 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 with a first order approximation and also re-written in a more compact form:

$$\begin{array}{l} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = g(\mathbf{x}, \mathbf{u}) \end{array} \implies \begin{array}{l} \dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u} \\ \mathbf{y} = C_c\mathbf{x} + D_c\mathbf{u} \end{array} \tag{3}$$

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 system to be controlled is usually modeled by a discrete state-space model in the MPC literature. Therefore, (3) 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 B_d 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 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 motion planning 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. In Fig.4 is depicted a conditional state machine designed for higher-level behavior planning:

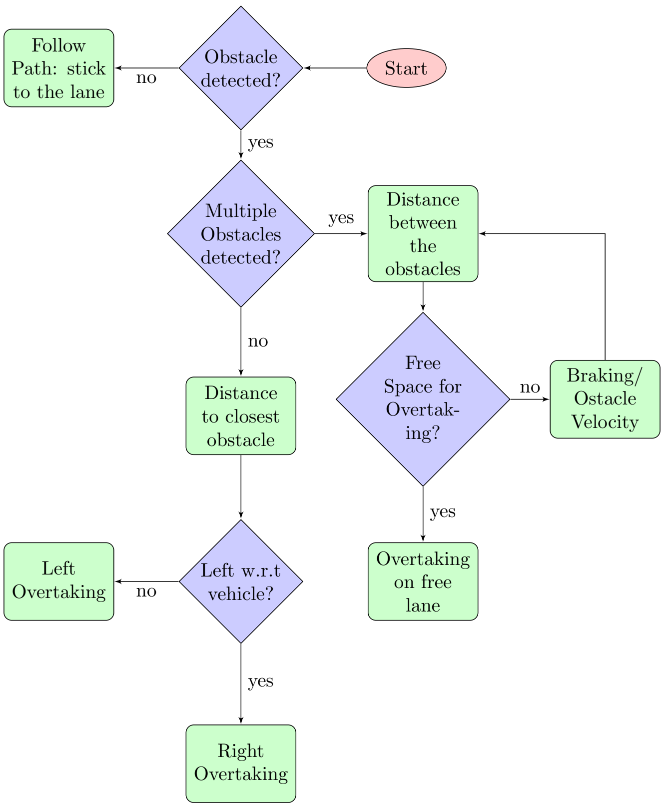

Fig. 4. Behaviour planning conditional flowchart.

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 (left boundary of the road);
- the second constraint is a lower bound on the y-coordinate (right 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;
- 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:

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

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. Fig.5 shows a schematic sketch of the constraints that are computed at each sampling time T_s in the case of a left overtaking.

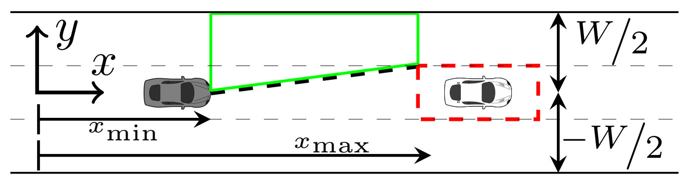

Fig. 5. Constraints in the case of left overtaking.

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/sec^2 on the throttle rate of change. Moreover we are using a constant reference signal for the velocity of $v = 20$m/s ($\simeq$ 72km/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 (Fig.6) we assumed that the road is straight and has 3 lanes of 4 meters wide each. 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 example the obstacle moves in the same direction as the vehicle.

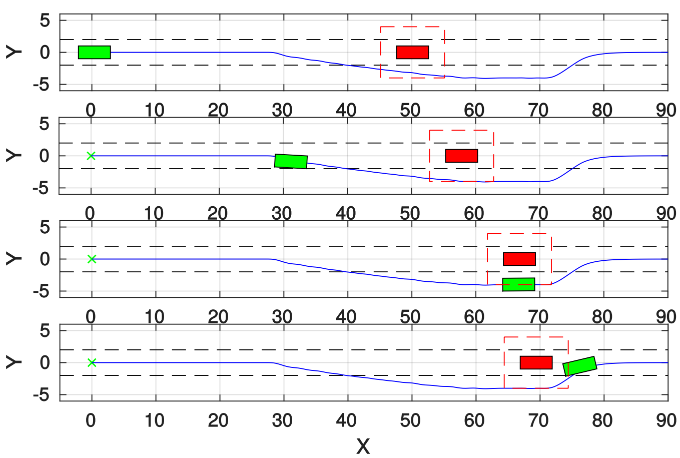

Fig. 6. Simulation of right overtaking with one moving obstacle

2) Multiple Moving Obstacles: Consequently, additional obstacles were added to make the scenario more complex (Fig.7). 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 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 opposite direction with respect to the ATLASCAR2 (but the same results can be verified in the same direction).

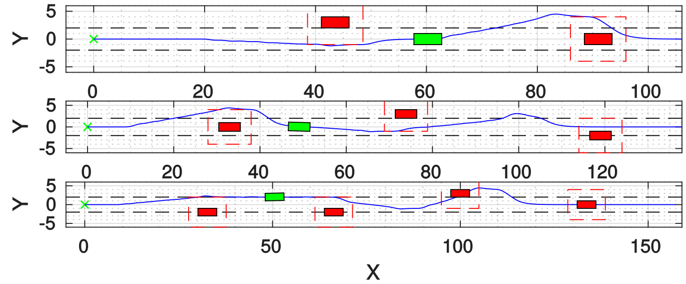

Fig. 7. Simulations of overtaking with $N = 2, 3, 4$ moving obstacles

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. In Fig.8 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.

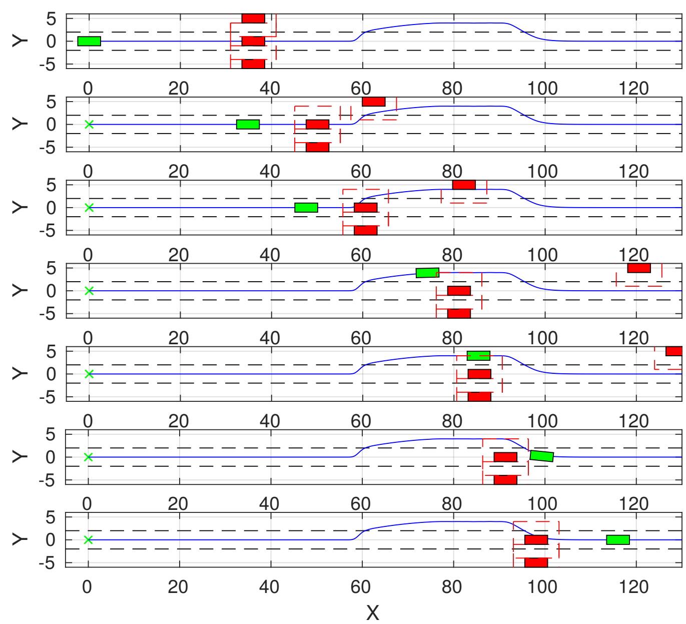

Fig. 8. Simulation of braking and overtaking obstacles.

At the beginning the vehicle detects all the other cars on the road. In the first part of the simulation, the ATLASCAR2 brakes because there is no enough space for overtaking the cars as shown in the Fig.9 (pag.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.10 (pag.6). 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.11 and consequently also on the heading angle in Fig.12. Then, the ATLASCAR2 returns to the reference velocity during the overtaking of the two obstacles. It is seen that the ATLASCAR2 avoids the obstacles and returns to the road centerline with a low overshoot.

V. Conclusions and Future Work

In this work we proposed an advanced method for the ATLASCAR2 based on adaptive Model Predictive Control. In this paper 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. The flexibility of the concepts used in the algorithm allows a multitude of refinements and extensions to this work. We can add to this algorithm 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. For example the next possible work could be to combine these two 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.

[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

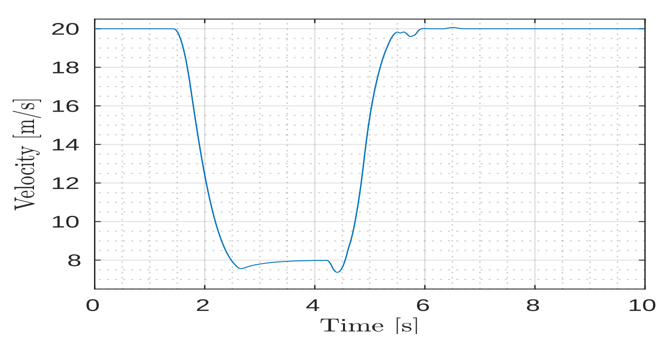

Fig. 9. Velocity v with respect to time.

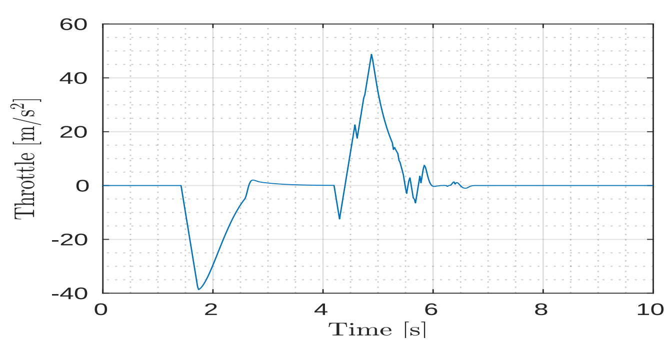

Fig. 10. Throttle T with respect to time.

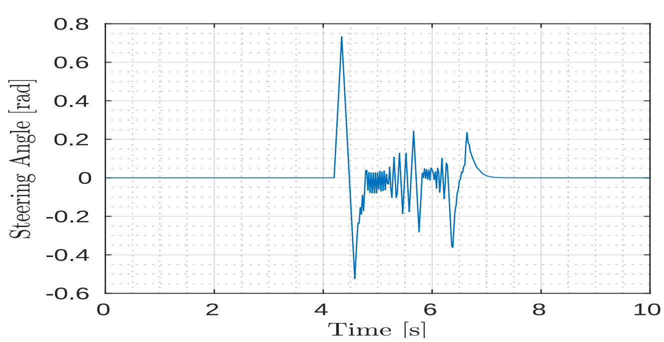

Fig. 11. Steering angle δ with respect to time.

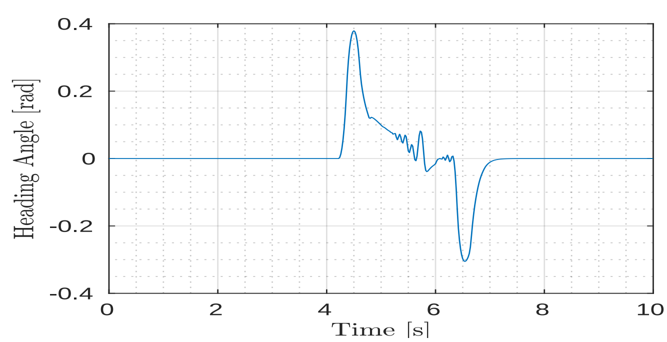

Fig. 12. Heading angle θ with respect to time.

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.