<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in . This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Lima, P F., Nilsson, M., Trincavelli, M., Mårtensson, J., Wahlberg, B. (2017)
Spatial Model Predictive Control for Smooth and Accurate Steering of an Autonomous Truck.

IEEE Transactions on Intelligent Vehicles, 2(4): 238-250

<https://doi.org/10.1109/TIV.2017.2767279>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-220573>

Spatial Model Predictive Control for Smooth and Accurate Steering of an Autonomous Truck

Pedro F. Lima¹, Mattias Nilsson², Marco Trincavelli²,
Jonas Mårtensson¹, and Bo Wahlberg¹

Abstract—In this paper, we present an algorithm for lateral control of a vehicle – a smooth and accurate model predictive controller. The fundamental difference compared to a standard MPC is that the driving smoothness is directly addressed in the cost function. The controller objective is based on the minimization of the first- and second-order spatial derivatives of the curvature. By doing so, jerky commands to the steering wheel, which could lead to permanent damage on the steering components and vehicle structure, are avoided. A good path tracking accuracy is ensured by adding constraints to avoid deviations from the reference path. Finally, the controller is experimentally tested and evaluated on a Scania construction truck. The evaluation is performed at Scania’s facilities near Södertälje, Sweden via two different paths: a precision track that resembles a mining scenario and a high-speed test track that resembles a highway situation. Even using a linearized kinematic vehicle to predict the vehicle motion, the performance of the proposed controller is encouraging, since the deviation from the path never exceeds 30 cm. It clearly outperforms an industrial pure-pursuit controller in terms of path accuracy and a standard MPC in terms of driving smoothness.

Fig. 1. Modified Scania G480 construction truck used as experimental and research platform.

I. INTRODUCTION

In the last decade, many major vehicle manufacturers, suppliers, and technology companies have started projects and collaborations with the objective of developing self-driving vehicles. This is a rapidly expanding technology that promises to play an important role in society.

The cruise controller, introduced in 1949, was the first step towards autonomous driving. In the meantime, other driving assistance systems have been developed providing an increasing level of support to the driver both in critical and tedious situations, reducing the number of traffic accidents and fatalities [1]. Driver assistance and autonomous driving technologies can be extremely beneficial not only for on-road driving applications, but also for off-road applications, such as gravel pits, mining areas, construction sites, and loading terminals. There, drivers are subject to dangerous environments and tedious tasks. Also, due to the increasing demand, the production sites are rapidly moving to remote locations and workers are experiencing more complicated working conditions. Additionally, many companies have recently

faced high administrative costs due to increased regulation over environmental compliance. These challenges make the development of autonomous vehicles an attractive idea, since it is a cost efficient option and it eliminates repetitive jobs that can lead to inattention and accidents, resulting in an efficiency and production increase. Furthermore, compared with an urban environment, a not so substantial investment and expansion in infrastructure and law adaption are required for autonomous driving [2]–[5]. By eliminating the human-in-the-loop factor, we expect to reduce the number of human fatalities and increase productivity and efficiency in these working sites [6].

A. Contributions

The work presented here is an extension of our previous work [7], where the concept of a smooth and accurate model predictive controller (SA-MPC) for lateral control of an autonomous vehicle was introduced together with some preliminary results.

The main contributions of this paper are:

- 1) the formulation of the SA-MPC algorithm by including smooth terms (minimization of the first- and second-order spatial derivatives of the curvature) directly in the cost function;
- 2) the development of a simulation environment that is accurate enough to be a reliable way of testing and verifying our control design;

¹Integrated Transport Research Lab and ACCESS Linnaeus Centre, Department of Automatic Control Department of Automatic Control, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. pfrdal@kth.se, jonas1@kth.se, bo@kth.se

²Research and Development, Scania CV AB, 151 87 Södertälje, Sweden, firstname.secondname@scania.com

This work has been funded by the Swedish government and automotive industry within the FFI program - Strategic Vehicle Research and Innovation under the project iQMatic 2012-04626.

- 3) the implementation of the SA-MPC in an industrial autonomous mining truck;
- 4) the benchmarking of the SA-MPC against a standard MPC controller (in simulation) and a pure-pursuit controller (PPC) (in practice);
- 5) simulation and experimental results using the SA-MPC showing the high degree of accuracy in path tracking, while performing smooth driving.

B. Background

Autonomous driving targets more efficient fuel consumption and harder safety requirements. Traffic congestion is reduced, since vehicles are able to drive closer to each other and probably at higher speeds. Hence, inefficient fuel consumption, wasted commuting time, and people frustration are translated into economic and financial benefits. The first boom in the autonomous vehicles field occurred in the 1980s and 1990s. Carnegie Mellon University (CMU, Pittsburgh, PA) presented its Navlab vehicles that operated in structured environments [8], UniBw Munich and Daimler-Benz's demonstrated autonomous driving at the European project PROMETHEUS [9], and the work done at the University of Parma with the ARGO project was promising. PROMETHEUS was indeed a great milestone, since autonomous driving was demonstrated in a three-lane French Autoroute with speeds up to 130 km/h, where the vehicles were able to change lane and perform lane and vehicle tracking. In parallel, the idea of automated highway systems was being developed within the California partners for advanced transportation technology (PATH) program [10]–[12]. At the time, the PATH program's main goal was to show technical feasibility rather than product development. In 1991, the PATH program was one of the first to demonstrate platooning at highway speed and using wireless communication [13]. Later, in 2004, the first DARPA grand challenge in 2004 was won by the Stanford's autonomous car, Stanley [14]. It autonomously completed an unrehearsed 230 km long course through the Mojave desert in less than 7 hours. After that, in 2007, DARPA organized a new competition, the urban challenge [15]. At the time, this event was spearheading, since the participants had to deal with other moving traffic and obstacles while respecting traffic rules. More recently, cooperative driving has captured interest in the grand cooperative driving challenge (GCDC) 2011 [16] and in GCDC 2016 [17]. In these, autonomous driving is more focused on cooperative driving and vehicle platooning. Countless other competitions have taken place around the world to test and demonstrate driverless vehicles' capabilities (e.g., [18]–[21]). In the last couple of years, autonomous vehicles have taken the next step by being released to the general public [22]–[24].

C. Outline

This paper is structured as follows. In Section II, the main modules of an autonomous vehicle system architecture are described with major focus on the vehicle control module and a brief literature review regarding each module of the system architecture is provided; in Section III, the problem of path following using a receding-horizon framework is addressed by

Fig. 2. Simplified system overview of an autonomous vehicle.

developing an SA-MPC controller for an autonomous vehicle. Also, the influence of the controller tuning parameters is analyzed; in Section IV, the performance of the SA-MPC is benchmarked against a pure-pursuit controller (PPC) and a standard MPC. To that end, we provide a statistical analysis of both the deviation from the path and the curvature change rate of the controllers. Also, the SA-MPC is compared with a PPC, where both are deployed in a Scania construction truck, and the validity of the simulation environment is discussed by comparing experimental and simulation results. Finally, in Section V, we provide some concluding remarks and outline future work.

II. SYSTEM ARCHITECTURE

In this section, we propose an autonomous vehicle system architecture and detail our contribution to its development. The architecture in Figure 2 provides an overview of an autonomous vehicle guidance system. We highlight three modules: the perception and localization, the motion planning, and the motion controller. Next, these modules are described in more detail.

A. Perception and localization

This module takes data generated from a set of sensors (e.g., RADARs, LIDARs, global positioning system (GPS) antennas, inertial measurement units (IMUs), and monocular and stereo cameras) and fuses them into a consistent representation of the world that is suitable for autonomous driving. Common approaches to overcome obstacle detection challenges, such as difficult visibility conditions, include the creation of an occupancy grid map [25], where each grid cell stores the probability for the existence of an obstacle [14], [26], [27]. In the case of dynamic objects, Bayesian filters, extended Kalman filters, and particle filters are used to perform tracking.

The main task of the localization module is to accurately estimate the autonomous vehicle state. An inaccurate localization may jeopardize the entire system and it is crucial for correct decision making and for a precise, safe, and comfortable motion. Typically, the localization of the vehicle is performed by fusing all available sensor information. Sensors gather measurements ranging from odometry to acceleration and can be combined with offline saved maps. Furthermore, including

knowledge about the vehicle motion in the measurement data provides smoother state estimations.

B. Motion planning

Motion planning is fundamental to cope with dynamic environments that might require computation of new paths. In the case of autonomous vehicles for mining applications, the need for a local planner is crucial since the landscape is constantly being altered. As a consequence of the mining works, mission end points might be dependent on external factors, and the maneuvering areas are typically small. The planning framework can be split in two planning layers. A global planner produces a high-level plan with long term goals, while a local planner ensures that the generated paths respect the vehicle limits and avoids possible obstacles. Early path planning work was not focused on path smoothness or planning that respected non-holonomic constraints. Nevertheless, according to [28], in the past two decades, autonomous driving has triggered research that targets those issues. The problem of path planning has been addressed using mainly interpolation-based methods, such as clothoids [29], [30] and polynomial curves [31]. Among other approaches, we can refer to sampling-based methods, such as rapidly-exploring random trees (RRT) [32], lattice-based motion planners [33], [34], optimal control theory [35], and hybrid approaches [27].

Furthermore, this module is responsible for selecting the autonomous vehicle behavior (e.g., lane keeping, overtaking, yielding, parking, off-road driving, and standing still). Hierarchical finite state machines and hierarchical concurrent finite state machines are popular when performing this task and they are successfully used in [22], [26], and [36].

C. Motion controller

The motion controller has been addressed in many different ways (e.g., fuzzy control [37], feedback control [38], and pure-pursuit algorithms [39]). Classical control approaches are computationally inexpensive and the resulting closed-loop properties, such as stability, are easy to prove. However, nonlinear time-varying models and constraints are not handled in a systematic way and the computed inputs are not optimal. To overcome these limitations, we choose to design the motion controller under the MPC framework. Recently, MPC has become prominent due to its ability to predict the vehicle behavior for a given set of inputs [40]–[42]. In MPC, a chosen cost function is minimized and the optimal sequence of inputs is computed in order to better follow a specified trajectory under known constraints on states and inputs. A subset of the optimal input sequence is applied to the vehicle and the process is then repeated.

MPC has been widely used to address the autonomous vehicle guidance problem (e.g., [43]–[51]). In this section, we perform a brief literature review of autonomous driving applications using MPC.

A particular area where MPC has been applied is active front steering (AFS). AFS modifies the vehicle steering angle without changing the steering wheel position, assisting the driver to drive around difficult corners and improving stability.

In [43], the authors combine differential braking and AFS to improve yaw stability. There, an MPC controller is designed to track a driver-requested yaw rate using offline computed optimal solutions by multiparametric programming [44]. The combination of AFS with braking is also explored in [45]. In this case, the intended contribution is to develop an autonomous vehicle capable of avoiding obstacles, rather than helping a human driver. In that work, the controller relies on a nonlinear MPC (NMPC), introduced in [46]. There, an NMPC controller is presented in order to stabilize a vehicle along a desired path while fulfilling its physical constraints. Since a nonlinear vehicle model is considered, the resulting MPC controller has a considerable computational complexity, which makes it difficult to be implemented in real-time. There are quite a few experimental validation results of MPC reported in the literature due to the use of complex models for predicting the vehicle behavior, which makes the optimization complex and slow. Also, adding constraints, such as imposing certain comfort measures, increases the problem complexity and the computational burden. To overcome these problems, a linear time-varying (LTV) MPC for AFS is proposed in [47] and a spatial vehicle model is proposed in [48]. In [47], the discrete time linear system is based on successive online linearizations of the nonlinear vehicle model, and the stability conditions for LTV-MPC controllers are also presented. Following the same reasoning, a control architecture based on LTV-MPC is presented in [49] to address the lane keeping and obstacle avoidance problems for a passenger car driving on low curvature and low friction roads. A study about the use of kinematic and dynamic vehicle models for MPC control design is detailed in [50]. The authors show that the kinematic model has better forecast errors when discretized at 200 ms, compared to 100 ms. Therefore, the authors motivate the MPC design based on a kinematic vehicle model since it is simpler than the dynamical model and still considerably accurate. In [51], the authors integrate an environment model with a stochastic predictive control approach to estimate and predict the state of surrounding vehicles. Thereafter, the authors demonstrate the ability of a constrained MPC to achieve certain objectives while avoiding collisions with other vehicles. Another approach to vehicle stabilization using MPC is presented in [52]. There, the vehicle motion is bounded within the region of the state space that does not contain unstable vehicle dynamics. In [53], an hierarchical framework based on MPC for autonomous vehicles is presented. At the high-level MPC, a desired trajectory is computed online using simplified models, which is then provided to a low-level MPC that computes the vehicle inputs using a detailed nonlinear vehicle model.

In the related literature, smoothness is not explicitly addressed in the control layer. Instead, smoothness is typically achieved by means of smooth trajectory generation. However, feedback is of utmost importance to be able to handle disturbances and model errors smoothly. This is the main reason why it is important to design a smooth controller.

III. SMOOTH AND ACCURATE

MODEL PREDICTIVE CONTROLLER

In autonomous driving, combining smoothness and comfort with accurate path tracking accuracy is a challenging task due to their conflicting objectives. To overcome this, we propose a smooth and accurate model predictive controller (SA-MPC) for autonomous driving. Standard MPC approaches aim at minimizing the deviation from a state trajectory reference. In our proposed formulation, we add a second objective to maximize the driving smoothness. In this way, we can trade-off accuracy versus smoothness. This is similar to an economic MPC, which includes an economic term in the objective function [54]. We explicitly include the vehicle curvature in the cost function to influence its shape and characteristics. We minimize the first- and second-order spatial derivatives of the curvature function. The underlying nonlinearities in the vehicle model lead to a LTV-MPC formulation.

A. Problem formulation

We intend to design a controller that steers the vehicle to a reference path under some constraints. Typically, the vehicle inputs are limited due to actuator limitations, and physical constraints are necessary to ensure safety, for example. Moreover, the vehicle dynamics can be described by a discrete nonlinear function. Also, we consider that the given path is feasible and obstacle free.

In short, the problem that we address in this paper is to design a controller that solves the following generic optimal control problem:

$$\max \quad \text{Tracking accuracy} + \text{driving smoothness} \quad (1a)$$

$$\text{s.t.} \quad \text{Vehicle dynamics} \quad (1b)$$

$$\text{Bounds on inputs} \quad (1c)$$

$$\text{Physical constraints} \quad (1d)$$

B. Control design

In this section, we address the problem of lateral control of an autonomous vehicle. We design an SA-MPC that combines path tracking accuracy and driving smoothness.

Let a reference path be constituted of N waypoints of the form $\mathbf{z}_i^{\text{ref}} = [x_i^{\text{ref}}, y_i^{\text{ref}}, \theta_i^{\text{ref}}, s_i]^T$, $i = 1, \dots, N$ representing the Cartesian coordinates, the orientation and the length of the path since the first point, respectively.

Let the cost function of the SA-MPC reflect the smoothness (or the comfort) of the vehicle motion and its predictions. To achieve that, the objective function is designed in order to penalize the first- and the second-order spatial derivative of the curvature function. The first-order spatial derivative of the curvature $\mathbf{D}_1\kappa$ is related to the vehicle lateral jerk, for which high values are perceived by a human as uncomfortable. The second-order spatial derivative of the curvature $\mathbf{D}_2\kappa$ is related to the angular acceleration of the steering wheel. In mining areas, vehicles are supposed to run uninterruptedly during long periods of time. Therefore, the vehicle motion control must target smooth driving and accurate path tracking. Aggressive steering results in a higher "wear and tear" in

any case. Hence, it is undesirable because it shortens the lifetime of the trucks and, as a consequence, increases the costs for the operators. Moreover, fast steering wheel angle changes can lead to aggressive and unpredictable driving, increased chance of vehicle roll-over, and damage to tires and steering components. Therefore, it is desirable to maintain both the first- and second-order spatial derivatives as low as possible while tracking the reference path in order to achieve smooth driving. With this formulation, the vehicle curvature is encouraged to be linear. Since the curvature in clothoids varies linearly with the path arc-length, we use the smoothness and comfort characteristics of clothoid-driving in the controller formulation. Clothoids are used for road design, since their curvature varies linearly with the path arc-length. This provides a linear change in the vehicle steering angle to perform the turn [55]. We have also used the concept of clothoids in the design of a clothoid-based model predictive controller (MPCC) [30]. The other goal of the SA-MPC is to accurately follow a given reference path. Therefore, soft constraints are added to the problem in order to avoid large deviations from the path. Moreover, the vehicle curvature and change rate are also limited.

Model-based control design is highly dependent on the quality of the models provided. On one hand, an accurate system model provides good predictions. However, these models are typically complex, computationally expensive, and hard to fit. On the other hand, simple models are less computationally demanding but are unable to provide good predictions. The movement of a car-like nonholonomic vehicle, such as a truck that performs curvy paths at low speeds (i.e., when the lateral dynamics have little influence) and drives straight at high-speeds can be described by its kinematics without significant loss of prediction accuracy [50].

The movement of a car-like nonholonomic vehicle can be described by its space-domain kinematic equations [38] given by

$$\frac{dx(s)}{ds} = \cos(\theta(s)), \quad (2a)$$

$$\frac{dy(s)}{ds} = \sin(\theta(s)), \quad (2b)$$

$$\frac{d\theta(s)}{ds} = \frac{1}{D} \tan(\delta(s)) := \kappa(s), \quad (2c)$$

where x and y are the coordinates of the vehicle in the global coordinate system, θ is the yaw angle, D is the distance between the front and rear axle, and δ is the steering angle of the front wheels. We explicitly define curvature $\kappa(s)$ in (2c) and we assume that $ds = v(t) \cdot dt$, $v(t) \neq 0$, where $v(t)$ is a continuous function and represents the longitudinal velocity in the vehicle coordinate system.

We discretize the vehicle model (2) by letting the vehicle state at each sampling point be defined as $\mathbf{z}(s_i) = \mathbf{z}_i = [x_i, y_i, \theta_i]^T$ and the vehicle input be $\kappa(s_i) = \kappa_i$. The vehicle state is sampled every $\Delta s_i = s_{i+1} - s_i$, which is the traveled distance between two consecutive points. The curvature of the vehicle is assumed to be constant between $\kappa(s_i)$ and $\kappa(s_{i+1})$. Thus, the position $[x(\kappa_n), y(\kappa_n)]^T$, with

$n = 1, \dots, H$, where $H \in \mathbb{Z}^+$ is the prediction horizon of the MPC, after applying $\kappa_n = [\kappa_1, \dots, \kappa_n]^\top$ is described as

$$x(\kappa_n) = \sum_{i=1}^n \cos \left(\sum_{j=1}^i \kappa(s_j) \Delta s_j \right) \Delta s_i, \quad (3a)$$

$$y(\kappa_n) = \sum_{i=1}^n \sin \left(\sum_{j=1}^i \kappa(s_j) \Delta s_j \right) \Delta s_i, \quad (3b)$$

with respect to the pose $(0, 0, 0)$. The equivalent vectorial version for $\kappa = [\kappa_1, \dots, \kappa_H]^\top$ is

$$\mathbf{x}(\kappa) = \mathbf{S} \cos(\mathbf{S}\kappa), \quad (4a)$$

$$\mathbf{y}(\kappa) = \mathbf{S} \sin(\mathbf{S}\kappa), \quad (4b)$$

where $\mathbf{S} \in \mathbb{R}^{H \times H}$ is a lower triangular matrix that performs the cumulative sum operation along a vector.

Note that the vehicle prediction is made as a function of curvature, rather than as a function of time. Also, the curvature is a function of the distance traveled by the vehicle, which causes the prediction model to be space-based.

The nonlinear SA-MPC is formulated as the following optimization problem:

$$\min_{\kappa} \quad \|\mathbf{D}_2 \kappa\|_2^2 + \alpha \|\mathbf{D}_1 \kappa\|_2^2 \quad (5a)$$

$$\text{s.t.} \quad |\mathbf{x}(\kappa) - \mathbf{x}_{\text{ref}}| \leq \varepsilon \quad (5b)$$

$$|\mathbf{y}(\kappa) - \mathbf{y}_{\text{ref}}| \leq \varepsilon \quad (5c)$$

$$|\mathbf{D}_1 \kappa| \leq \mathbf{1} c_{\text{max}} \quad (5d)$$

$$|\kappa| \leq \mathbf{1} \kappa_{\text{max}} \quad (5e)$$

$$\kappa_0 = \kappa_{\text{vehicle}}, \quad (5f)$$

where $\kappa \in \mathbb{R}^{H+1}$ is the curvature vector to be optimized, $\varepsilon \in \mathbb{R}^H$ is the maximum allowed deviations from the path waypoints in x and y directions, $\mathbf{D}_2 \in \mathbb{Z}^{(H-1) \times (H+1)}$ and $\mathbf{D}_1 \in \mathbb{Z}^{H \times (H+1)}$ are the matrix operators that calculate the first- and second-order differences of a vector, respectively, and $\alpha \in \mathbb{R}$ penalizes the curvature change rate. The operators $\mathbf{D}_2$ and $\mathbf{D}_1$ are inversely weighted with the distance between the waypoints (i.e., the shorter the distance, the bigger the weight is). The constants κ_{max} and c_{max} denote the maximum curvature and curvature change rate that the vehicle can perform, respectively, and $\mathbf{1}$ is a column vector filled with ones. Note that the coordinates of the vehicle motion $[\mathbf{x}(\kappa), \mathbf{y}(\kappa)]^\top$ are only dependent on the curvature κ , since we can accurately approximate the arc-lengths between points to be equal to the arc-lengths between the reference path waypoints. The inequalities (5b) to (5e) are performed element-wise, and the equality constraint sets the initial curvature value of the optimization problem. We assume constant speed over the prediction horizon as the vehicle longitudinal dynamics are slow. The speed profile is computed in a similar way to the one described in [56] in the path planning module.

Remark: Although not used, the cost function (5a) can include a l_1 -norm regularization term $\|\mathbf{D}_2 \kappa\|_1$ to induce sparsity in the obtained curvature commands, see [57], [58].

To calculate the second spatial derivative of the curvature, we use a second-order difference operator, $\mathbf{D}_2$, weighted with

the distance between the waypoints. The operator $\mathbf{D}_2$ used to approximate the second spatial derivatives of the curvature is based on finite differences. The first spatial derivative of the curvature function with respect to traveled distance is approximated as

$$\frac{d\kappa(s)}{ds} \Big|_{s_i} \approx \frac{\kappa(s_{i+1}) - \kappa(s_i)}{|s_{i+1} - s_i|} = \frac{\kappa(s_{i+1}) - \kappa(s_i)}{|\Delta s_i|} \quad (6)$$

and consequently the discrete second spatial derivative is

$$\frac{d^2\kappa(s)}{ds^2} \Big|_{s_i} \approx \frac{\frac{\kappa(s_{i+1}) - \kappa(s_i)}{\Delta s_i} - \frac{\kappa(s_i) - \kappa(s_{i-1})}{\Delta s_{i-1}}}{\Delta s_{i-1,i}} \quad (7a)$$

$$= \frac{(\kappa(s_{i+1}) - \kappa(s_i)) \Delta s_{i-1} - (\kappa(s_i) - \kappa(s_{i-1})) \Delta s_i}{\Delta s_{i-1,i} \Delta s_{i-1} \Delta s_i}, \quad (7b)$$

where $\Delta s_{i,j} = \frac{\Delta s_i + \Delta s_j}{2}$. Since $\mathbf{D}_2$ is a matrix operator over a vector, we stack (7b) from $i = 1, \dots, H$ where H is the prediction horizon and get

$$\mathbf{D}_2 = \begin{bmatrix} -\frac{\Delta s_1}{\Gamma_1} & \frac{\Delta s_0 + \Delta s_1}{\Gamma_1} & -\frac{\Delta s_0}{\Gamma_1} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{\Delta s_H}{\Gamma_H} & \frac{\Delta s_{H-1} + \Delta s_H}{\Gamma_H} & -\frac{\Delta s_{H-1}}{\Gamma_H} \end{bmatrix}, \quad (8)$$

where $\Gamma_i = \Delta s_{i-1,i} \Delta s_{i-1} \Delta s_i$.

Similarly to $\mathbf{D}_2$, $\mathbf{D}_1$ is obtained by stacking (6)

$$\mathbf{D}_1 = \begin{bmatrix} \frac{1}{\Delta s_0} & -\frac{1}{\Delta s_0} & & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{\Delta s_{H-1}} & -\frac{1}{\Delta s_{H-1}} \end{bmatrix}. \quad (9)$$

The reconstructed path $[\mathbf{x}(\kappa), \mathbf{y}(\kappa)]^\top$ is computed using (4), which are clearly nonlinear equations, leading to a non-convex problem formulation (5). A linear approximation is done using a first-order Taylor approximation of both cosine and sine around the path orientation θ^{ref} . The linearization around θ^{ref} takes the form

$$\mathbf{x}_l(\kappa) = \mathbf{S} (\cos(\theta^{\text{ref}}) - \sin(\theta^{\text{ref}}) \odot (\mathbf{S}\kappa - \theta^{\text{ref}})), \quad (10a)$$

$$\mathbf{y}_l(\kappa) = \mathbf{S} (\sin(\theta^{\text{ref}}) + \cos(\theta^{\text{ref}}) \odot (\mathbf{S}\kappa - \theta^{\text{ref}})), \quad (10b)$$

where $\odot$ is the Hadamard product. Also, the problem may be infeasible when there is no solution that respects the constraints (5b) and (5c). Therefore, we add a slack variable $\Delta = [\Delta_x^\top \Delta_y^\top]^\top$ and minimize the cost function taking the slack variable into account as well. Also, with the inclusion of the slack variable Δ , the deviations from the path are explicitly included in the cost function that enforces convergence to the reference path. The linear MPC formulation is

$$\min_{\kappa, \Delta} \quad \|\mathbf{D}_2 \kappa\|_2^2 + \alpha \|\mathbf{D}_1 \kappa\|_2^2 + \Delta^\top \Lambda \Delta \quad (11a)$$

$$\text{s.t.} \quad |\mathbf{x}_l(\kappa) - \mathbf{x}_{\text{ref}}| \leq \varepsilon + \Delta_x \quad (11b)$$

$$|\mathbf{y}_l(\kappa) - \mathbf{y}_{\text{ref}}| \leq \varepsilon + \Delta_y \quad (11c)$$

$$|\mathbf{D}_1 \kappa| \leq \mathbf{1} c_{\text{max}} \quad (11d)$$

$$|\kappa| \leq \mathbf{1} \kappa_{\text{max}} \quad (11e)$$

$$\Delta \geq 0 \quad (11f)$$

$$\kappa_0 = \kappa_{\text{vehicle}}, \quad (11g)$$

Fig. 3. Simulation of one run of the controller with $H = 10$, $\alpha = 0$, and $\lambda = 100$, where the size box constraint ε is varied.

where $\Lambda \in \mathbb{R}^{2H \times 2H}$ is a diagonal matrix of the form $\text{diag}(\lambda_1, \dots, \lambda_H, \lambda_1, \dots, \lambda_H)$ that penalizes constraint violations.

The final formulation of the controller is cast as a QP [59].

C. Controller parameters tuning

We constrain the predicted position of the vehicle to be inside of a box around the waypoint. By designing an off-road driving controller, we can interpret the size of such a constraint as the vehicle's desired maximum deviation from the path. In case of road driving, this is equivalent to the lane width minus half of the vehicle width. Hence, it is crucial to understand how the tuning parameters ε and λ influence the controller performance. The first one determines the size of the box constraints and, consequently, the freedom the controller has to choose a suitable curvature function, such that the predicted path is within these constraints. The latter parameter sets how sharp the quadratic penalization of the slack variables is when such a predicted path violates the box constraints.

Figures 3 and 4 depict the influence of the parameters ε and λ , respectively. Figure 3 shows the effect of the different sizes of the box constraints. The larger the box constraint, the smoother the curvature function is. In Figure 4, λ is constant over the horizon and we study how the curvature function changes for different values of λ . The smaller λ is, the smoother the curvature function is. However, the predicted vehicle path can lay outside of the constraint boxes with low penalization. In the extreme case $\lambda = 0$, the curvature function is equal to the current vehicle curvature throughout the prediction horizon.

Fig. 4. Simulation of one run of the controller with $H = 10$, $\varepsilon = 0$, and $\alpha = 0$, where the penalization to violate the constraint λ is varied. In this case, λ is constant along the horizon.

Another parameter to consider is α , which sets the importance of maintaining the current constant curvature throughout the horizon. Figure 5 depicts the influence of α on the curvature function. The larger α is, the more constant the curvature function is. We can see that if we penalize $\|\mathbf{D}_1 \kappa\|_2^2$ more, the variation in the curvature decreases. This is an intended property of the controller, since the term was included to regularize the amount of steering that we need in order to follow the path. This way, it is possible to favour changing the curvature as little as possible.

IV. RESULTS

To demonstrate the performance of the controller in a real vehicle, we deployed the SA-MPC described in Section III, on a Scania construction truck. We compare the performance of the SA-MPC with a pure-pursuit controller (PPC) that was previously implemented on the truck. Furthermore, we develop a simulation environment that resembles the real system. This simulation environment allows evaluating and validating the control design before deploying it in the experimental platform. There, the SA-MPC is compared with a PPC and with a standard MPC. The lateral dynamics of the simulation model are described by a 4-axes nonlinear bicycle model with 2 steering axles in the front and 2 traction axles in the back [59]. It is based on the modified Scania G480 construction truck shown in Figure 1. We analyze the performance of the controllers in terms of path tracking accuracy and driving smoothness and discuss the simulation environment reliability as a test environment.

The main idea behind the experimental controller benchmarking is to determine the quality of the designed controllers

Fig. 5. Simulation of one run of the controller with $H = 10$, $\varepsilon = 0.3$, and $\lambda = 1$, where the importance given to the first-order spatial derivative of curvature α is varied.

when compared with standard approaches. The SA-MPC formulation intends to provide driving smoothness. Therefore, the SA-MPC is compared with a standard MPC to understand the effect that the smoothing term has on path tracking accuracy. Finally, both MPC approaches are compared with a PPC. Although being a simple controller, the PPC is widely used in industry due to its ease of implementation and satisfactory results. However, it never achieves perfect tracking on curvy roads, which is a major drawback of the method.

A. Reference paths

The reference paths used are generated by recording a GPS trace using a RTK-GPS installed in the truck. The reference paths are recorded at Scania's test facilities near Södertälje, Sweden. The paths are then subsampled and consist of a waypoint sequence, in which the waypoints are two meters apart. We use a precision track, which is a challenging gravel road designed to perform tests with autonomous mining trucks, since it resembles a mining site. It is a narrow road with an approximate length of 1.5 km. It has sharp turns, where consequently the vehicle speed can not be higher than 50 km/h. The curvature in these turns typically exceeds 0.05 m^{-1} . Therefore, it is the perfect scenario to evaluate the performance of control algorithms designed to be extremely accurate at low speeds. In addition, to widen the scope of the designed controller applicability, a high-speed test track is also used. It consists of two long straights and two sharp U-turns (one with curvature 0.04 m^{-1} and the other with 0.05 m^{-1}). On one of the straights the speed limit is 70 km/h, while in the other straight the speed limit is 90 km/h.

Fig. 6. The pure-pursuit controller algorithm illustration (inspired from [39]).

B. Controllers

In this section, we introduce the pure-pursuit controller (PPC) and a standard MPC, and revisit the SA-MPC. All the controllers described in this section are evaluated in simulation. The SA-MPC and the PPC are also evaluated experimentally.

1) *Pure-pursuit controller*: Let a reference path be constituted of a finite number of points $N > 1$ of the form $\mathbf{z}_i^{\text{ref}} = [x_i^{\text{ref}}, y_i^{\text{ref}}]^T$, $i = 1, \dots, N$, which represent the Cartesian coordinates of the path $x_i^{\text{ref}}, y_i^{\text{ref}}$. The path coordinates are relative to the current vehicle position and orientation.

Let the desired curvature be the vehicle input. Figure 6 depicts an illustration of the pure-pursuit controller algorithm [39]. In fact, this method is inspired by the way humans drive. Humans tend to look some distance in front of the car and head towards that spot. The algorithm has only one user-defined parameter, the look-ahead distance $L_{\text{ahead}} > 0$, with which the goal position is chosen. With the goal position defined, simple geometric relations are used to determine the curvature required, represented by κ in Figure 6, to move the vehicle from its current position to the goal position, denoted by $(0, 0)$ and by (x, y) , respectively.

The pure-pursuit controller is often applied due to its versatility and ease of implementation. However, fine tuning is hard to accomplish and it is vehicle specific. Furthermore, the main parameter, the look-ahead distance, has a large influence on the path following accuracy, which can never obtain null deviation on curvy roads, resulting in a major drawback of the method.

2) *Standard MPC*: In this case, the reference path also includes the path orientation θ_i^{ref} , $i = 1, \dots, N$, such that it is given in the form $\mathbf{z}_i^{\text{ref}} = [x_i^{\text{ref}}, y_i^{\text{ref}}, \theta_i^{\text{ref}}]^T$.

Let $\mathbf{z}_i = [x_i, y_i, \theta_i]^T$ be the vehicle states where index i ranges from 1 to H , and H is the MPC prediction horizon. Again, the control input to the vehicle is the curvature, $\mathbf{u}_i = \kappa_i$. Typically, a standard MPC is formulated as

$$\begin{aligned} \min_{\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_H} \quad & \sum_{i=1}^H \tilde{\mathbf{z}}_i^T \mathbf{Q} \tilde{\mathbf{z}}_i + \tilde{\mathbf{u}}_i^T \mathbf{R} \tilde{\mathbf{u}}_i \quad (12a) \\ \text{s.t.} \quad & \mathbf{z}_{i+1} = f(\mathbf{z}_i, \mathbf{u}_i), \quad (12b) \\ & \mathbf{z}_0 = \mathbf{z}(s_0), \quad (12c) \\ & \mathbf{u} \in \mathbb{U}, \quad (12d) \end{aligned}$$

where $\tilde{\mathbf{z}}_i = \mathbf{z}_i - \mathbf{z}_i^{\text{ref}}$, $\tilde{\mathbf{u}}_i = \mathbf{u}_i - \mathbf{u}_i^{\text{ref}}$, $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{R} \in \mathbb{R}$ are the state penalization and the input penalization matrices, respectively, and they are generally chosen to be diagonal and positive definite and $\mathbb{U}$ is the set of feasible inputs. The cost function (12a) penalizes deviations from the reference path, (12b) represents the kinematic vehicle model (2) that is used to predict the vehicle's motion, (12c) is the initial condition, and (12d) describes the input constraints.

The vehicle model (2) is discretized and linearized around the reference path using the sample interval $\Delta s = vT_s$, where v is the current vehicle speed and $T_s = 200$ ms is the sampling rate of the prediction model (motivated by the results from [50]).

In the end, (12) is rewritten as an LTV-MPC and cast as a QP.

3) *Smooth and accurate MPC (SA-MPC)*: As before, let the vehicle be described solely by its position $\mathbf{z}_i = [x_i, y_i]^T$, $i = 1, \dots, H$ and $\mathbf{u}_i = \kappa_i$ be its desired curvature and input. Also, the path is only described by its Cartesian coordinates $\mathbf{z}_i^{\text{ref}} = [x_i^{\text{ref}}, y_i^{\text{ref}}]^T$.

As described in Section III, the objective function is defined as the driving smoothness and comfort. These concepts are assumed to be achieved if the cost function includes a term that encourages predictions using piecewise linear curvature function. Thus, let $\|\mathbf{D}_2 \kappa\|_2^2 + \alpha \|\mathbf{D}_1 \kappa\|_2^2$ be the cost associated with the vehicle motion, where $\mathbf{D}_2 \in \mathbb{Z}^{(H-1) \times (H+1)}$ and $\mathbf{D}_1 \in \mathbb{Z}^{H \times (H+1)}$ are the matrix operators that calculate the first- and second-order differences of a vector. We formulate the SA-MPC as

$$\min_{\mathbf{u}, \Delta} \quad \mathbf{u}^T \mathbf{R} \mathbf{u} + \Delta^T \mathbf{\Lambda} \Delta \quad (13a)$$

$$\text{s.t.} \quad |\mathbf{x} - \mathbf{x}_{\text{ref}}| \leq \varepsilon + \Delta_{\mathbf{x}}, \quad (13b)$$

$$|\mathbf{y} - \mathbf{y}_{\text{ref}}| \leq \varepsilon + \Delta_{\mathbf{y}}, \quad (13c)$$

$$\mathbf{u} \in \mathbb{U}, \quad (13d)$$

$$\Delta \geq 0, \quad (13e)$$

where $\mathbf{R} = \mathbf{D}_2^T \mathbf{D}_2 + \alpha \mathbf{D}_1^T \mathbf{D}_1$. Moreover, $\alpha \in \mathbb{R}$ penalizes the curvature change rate magnitude, $\mathbf{\Lambda} \in \mathbb{R}^{2H \times 2H}$ is a diagonal matrix of the form $\text{diag}(\lambda_1, \dots, \lambda_H, \lambda_1, \dots, \lambda_H)$ that penalizes how much the solution can violate the constraints and $\mathbb{U}$ is the set of feasible inputs. The cost function (13a) is minimized with respect to the input, which is the vehicle desired curvature, and to a slack variable $\Delta = [\Delta_{\mathbf{x}}^T \Delta_{\mathbf{y}}^T]^T$. This slack variable is included in the box constraints that limit the vehicle motion prediction (13b) and (13c) to avoid problem infeasibility.

Note that (13) is written in epigraph form. If $\varepsilon = 0$, the optimization problem can be rewritten as

$$\min_{\mathbf{u}} \quad \mathbf{u}^T \mathbf{R} \mathbf{u} + \tilde{\mathbf{z}}^T \mathbf{\Lambda} \tilde{\mathbf{z}} \quad (14a)$$

$$\text{s.t.} \quad \mathbf{z}_{i+1} = f(\mathbf{z}_i, \mathbf{u}_i), \quad (14b)$$

$$\mathbf{z}_0 = \mathbf{z}(s_0), \quad (14c)$$

$$\mathbf{u} \in \mathbb{U}, \quad (14d)$$

where $\tilde{\mathbf{z}} = \mathbf{z} - \mathbf{z}_{\text{ref}}$ describes the deviations from the reference path and (14b) represents the kinematic vehicle model (2) that is used to predict the vehicle's motion.

Comparing the SA-MPC (14) with the standard MPC (12) we can highlight several similarities and differences. In the SA-MPC, there is no penalization for deviations from the reference input $\mathbf{u}_{\text{ref}}$, while in the standard MPC it is present. However, in the SA-MPC, there is a cost associated with the vehicle curvature spatial derivatives, which is absent in the standard MPC. Moreover, in both the SA-MPC and in the standard MPC there is a cost associated with deviations from the reference path.

Again, (14) is rewritten as an LTV-MPC and cast as a QP.

C. Simulation results

Experimental evaluation requires some level of prior controller testing and validation. Therefore, we set up a simulation environment in MATLAB/Simulink resembling, as much as possible, the real system with the same modules and interfaces as in the vehicle. The system and the controller diagrams are described in detail in section II. Our intention is to demonstrate that the simulation environment is sufficiently accurate to allow, for example, controller tuning before deploying it on the vehicle. Also, in simulation we can extend our analysis further than experimentally, since we are not bound by implementation issues and platform availability. In this environment, we compare the PPC, the standard MPC, and the SA-MPC.

The simulation results are shown in Figures 7 and 8 for the three scenarios discussed in Section IV-A. In the boxes that are depicted in these figures, the central mark is the median, the edges of the box are the 25th and 75th percentiles and the whiskers extend to the most extreme data points not considered outliers. Table I summarizes the tuning of the controllers' parameters for each controller.

On the precision track, the vehicle speed is typically between 4 m/s and 10 m/s, while on the high-speed test track the speed can reach 25 m/s on one of the straights. In this case, both the standard MPC and SA-MPC approaches outperform the PPC. To be able to draw conclusions about the performance of the controllers, we also take into account the curvature change rate data shown in Figures 7 (b) and 8 (b). Both the SA-MPC and the standard MPC approaches have similar and effective performance, since they both have a low average deviation and a maximum deviation from the path below 20 cm (see Table II). However, the standard MPC approach is more aggressive if we compare curvature request change rate. First of all, we can see that a pure-pursuit strategy is, for the scenarios considered, the smoothest controller. In contrast, the standard MPC approach is the most aggressive, but also the most accurate. This is expected from the standard MPC parameter tuning, which can be seen in Table I. The

(a) Deviation from the path and controllers curvature request.

(b) Statistical analysis of the deviation from the path and the controller smoothness.

Fig. 7. Simulation run of the PPC, MPC, and SA-MPC on the precision track. The controllers parameters are presented in Table I.

controller can achieve a smoother response, but at the cost of a significant loss of accuracy. For understandable benchmarking, it is interesting to have the best path tracking accuracy possible, which we are interested in maximizing, and compare driving behaviors. Therefore, there is a clear tradeoff between accuracy and smoothness, which is challenging to balance. Nevertheless, the SA-MPC is able to effectively combine a smooth control action with an accurate path following.

D. Experimental results

The truck used in the experimental evaluation is a modified Scania G480 construction truck and is shown in Figure 1.

(a) Deviation from the path and controllers curvature request.

(b) Statistical analysis of the deviation from the path and the controller smoothness.

Fig. 8. Simulation run of the PPC, MPC, and SA-MPC on the high-speed test track. The controllers parameters are presented in Table I.

The vehicle is equipped with a sensor platform and a servo motor for automated control of the steering column. A detailed description of the system architecture is given in Section II. To perform these experimental tests, a GPS track is previously recorded containing the vehicle position and orientation. The GPS track is post-processed to include path curvature information and a speed profile. Then, the resultant waypoints are added to a database that is available to be queried at run time (i.e., an algorithm constantly determines if the vehicle is close to a GPS track previously recorded and feeds the waypoints to the controller). Another option is to use a local path planner. Two fixed reference paths are chosen to allow for a proper

TABLE I
TUNING OF THE CONTROLLERS PARAMETERS FOR SIMULATION
EVALUATION.

Parameters	PPC	MPC	SA-MPC
Look ahead, L_{ahead}	1.2 s		
Prediction horizon, H		10	10
Path sampling, T_s		200 ms	200 ms
Controller frequency	50 Hz	50 Hz	50 Hz
State pen. matrix, $\mathbf{Q}$		diag(50, 50, 0.1)	
Input pen. matrix, $\mathbf{R}$		500	
Slack penalization, λ_i			200
First derivative pen., α			200
Box constraints, ε_i			0

TABLE II
DEVIATION FROM THE PATH COMPARISON IN SIMULATION.

		Max. [m]	Avg. [m]	Std. dev. [m]
Precision track	PPC	0.59	0.10	0.12
	MPC	0.12	0.02	0.02
	SA-MPC	0.09	0.02	0.02
High-speed track	PPC	0.41	0.06	0.08
	MPC	0.13	0.02	0.02
	SA-MPC	0.13	0.03	0.02

comparison between the PPC and the SA-MPC. We choose to implement the SA-MPC on the truck, as it is the controller with the best performance in the simulation environment, while the PPC was already deployed and provided a good benchmark.

The experimental evaluation is performed on the precision track and the high-speed test track. The road is extremely narrow at some points, as the video frame from the experiment performed at the precision track in Figure 9 shows. The vehicle has just a little more space than its own width. In mining sites, the roads are expected to be similar to this one, which motivates this experiment and the crucial need for high accuracy. Additionally, we intend to demonstrate the capabilities of the controller outside the mining site application as the high-speed test track replicates a highway situation.

The experimental results are shown in Figures 10 and 11. On the top figures (a), we compare the curvature requests from the PPC and the SA-MPC and their lateral deviation from the path. The results are summarized in Table IV. Table III summarizes the tuning of the controllers' parameters for each controller. We also provide a statistical analysis of the distance to path in the bottom figures (b).

In both situations, it is clear that the SA-MPC outperforms

TABLE III
TUNING OF THE CONTROLLERS PARAMETERS FOR EXPERIMENTAL
EVALUATION.

Parameters	PPC	SA-MPC
Look ahead time gap, L_{ahead}	1.2 s	
Prediction horizon, H		10
Path sampling time, T_s		200 ms
Controller frequency	50 Hz	50 Hz
Slack penalization, λ_i		20
First-order derivative penalization, α		20
Box constraints sizes, ε_i		0

Fig. 9. From inside the truck cabin while performing the precision track (courtesy of Scania CV AB).

TABLE IV
DEVIATION FROM THE PATH COMPARISON IN REAL EXPERIMENTS.

		Max. [m]	Avg. [m]	Std. dev. [m]
Precision track	PPC	0.48	0.10	0.10
	SA-MPC	0.24	0.06	0.05
High-speed track	PPC	1.08	0.25	0.21
	SA-MPC	0.29	0.06	0.05

the PPC. Also, we can see that the performance of the SA-MPC on the high-speed test track (see Table IV) is similar to the one expected from simulation results (see Table II), which supports our claim of having a realistic simulation environment that can be used prior to experimental evaluation. This is not contradicted by the fact that the PPC performs quite differently at high speeds in reality when compared with the simulation. Namely, it means that there are high-speed dynamics that are not considered in the simulation vehicle model, which was expected. Also, it means that the PPC is not able to handle these dynamics in a real vehicle, while an MPC approach is intrinsically more capable. At low-speeds, the performance of both controllers is good with an average deviation from the path of less than 10 cm (see Table IV). Still, the SA-MPC outperforms the PPC especially in sharp turns. Note that the SA-MPC is able to run at both high and low speeds with the same parameter configuration.

Both in simulation and experimentally, the box constraints size of the SA-MPC is set to zero. Consequently, the original problem (13) is typically infeasible. However, the smaller box constraint the more accurate the controller is (and it is still smoother than the standard MPC (12)). Also, from the implementation point of view, zero-box constraints result in a simpler formulation (14). The main point of including box constraints on the original formulation (13) is to allow the MPC to predict the vehicle's motion in a corridor (e.g., between two lanes). In our opinion, this is a very useful feature and can be explored further in future work.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a smooth and accurate model predictive controller (SA-MPC) for smooth autonomous path following. In this case, the objective function is formulated to stress smooth and comfortable driving. With this purpose, we

(a) Deviation from the path and controllers' curvature request.

(b) Statistical analysis of the deviation from the path.

Fig. 10. Experimental run of the PPC and SA-MPC on the precision track. The controllers' parameters are presented in Table I.

(a) Deviation from the path and controllers' curvature request.

(b) Statistical analysis of the deviation from the path.

Fig. 11. Experimental run of the PPC and SA-MPC on the high-speed test track. The controllers' parameters are presented in Table III.

minimize the first- and second-order spatial derivatives of the vehicle curvature. Therefore, we encourage the vehicle to be guided using linear curvature in order to provide smooth driving. Nevertheless, we added soft constraints to the formulation in order to always drive in the reference path vicinity. In the end, we illustrated the influence of the parameter tuning in simulation. We provided simulation and experimental results using the SA-MPC and the MPCC. We compared them to existing approaches, such as the pure-pursuit controller and a standard MPC. The performance was measured in terms of deviation from the path and curvature change rate (i.e., driving smoothness). In our experiments and simulations, the pure-pursuit controller is always the controller that provides the smoothest driving. However, it was generally the least accurate controller. Finally, the SA-MPC and the standard MPC are the most accurate controllers, as expected. In order to point out which one is the best, we also analyzed the curvature change rate provided by the controllers, and concluded that the SA-MPC is smoother than the standard MPC with a similar path accuracy. The SA-MPC formulation plays a crucial role in the driving smoothness, since, unlike a standard MPC, it encourages smooth curvature predictions. Consequently, the

SA-MPC is the controller that provides a better trade-off in relation to smooth driving and accurate path following. Experimentally, we compared the SA-MPC with a PPC, both deployed in a Scania construction truck. As predicted from the simulations, the SA-MPC clearly outperformed the PPC. Even using a linearized kinematic vehicle to predict the vehicle motion, the performance of the controller is very promising, since the maximum deviation from the path never exceeds 30 cm and on average is 6 cm. Also, we concluded that the tuning of the controllers parameters using the simulation environment provided a good approximation of the actual controller performance.

In future work, we would like to analytically demonstrate stability properties of the designed controller [60] and we will consider the development of a procedure to systematically tune the parameters of the controller, possibly online while driving. One possible, interesting approach would be to combine machine learning and adaptive control techniques with model predictive control to adapt the controller or model parameters depending on the environment. Also, it would be engaging to extend the formulation in light of robust MPC to mitigate the influence of model uncertainties in the controller performance.

REFERENCES

- [1] P. Ross, "Robot, you can drive my car," *IEEE Spectrum*, vol. 51, no. 6, pp. 60–90, 2014.
- [2] J. Marshall, T. Barfoot, and J. Larsson, "Autonomous underground tramming for center-articulated vehicles," *J. Field Rob.*, vol. 25, no. 6-7, pp. 400–421, 2008.
- [3] S. Scheding, G. Dissanayake, E. Nebot, and H. Durrant-Whyte, "An experiment in autonomous navigation of an underground mining vehicle," *IEEE T. Robot. Autom.*, vol. 15, no. 1, pp. 85–95, 1999.
- [4] L. Thrybom, J. Neander, E. Hansen, and K. Landernas, "Future challenges of positioning in underground mines," in *Proceedings of the IFAC Conference on Embedded Systems, Computer Intelligence and Telematics*, June 2015, pp. 222–226.
- [5] H. Andreasson, A. Bouguerra, M. Cirillo, D. Dimitrov, D. Driankov, L. Karlsson, A. Lilienthal, F. Pecora, J. Saarinen, and A. Sherikov, "Autonomous transport vehicles: where we are and what is missing," *IEEE Rob. Autom. Mag.*, vol. 22, no. 1, pp. 64–75, 2015.
- [6] D. Bellamy and L. Pravica, "Assessing the impact of driverless haul trucks in Australian surface mining," *Resour. Policy*, vol. 36, no. 2, pp. 149–158, 2011.
- [7] P. Lima, M. Trincavelli, J. Mårtensson, M. Nilsson, and B. Wahlberg, "Experimental evaluation of predictive lateral control for an autonomous truck," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2016.
- [8] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the carnegie-mellon navlab," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 362–373, 1988.
- [9] E. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen, "The seeing passenger car VaMoRs-P," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, October 1994, pp. 68–73.
- [10] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 195–207, 1993.
- [11] J. Hedrick, M. Tomizuka, and P. Varaiya, "Control issues in automated highway systems," *IEEE Control Syst.*, vol. 14, no. 6, pp. 21–32, 1994.
- [12] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated vehicle control developments in the path program," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 114–130, 1991.
- [13] K. Chang, W. Li, P. Devlin, A. Shaikhbahai, P. Varaiya, J. Hedrick, D. McMahon, V. Narendran, D. Swaroop, and J. Olds, "Experimentation with a vehicle platoon control system," in *Proceedings of Vehicle Navigation and Information Systems Conference*, vol. 2, October 1991, pp. 1117–1124.
- [14] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA grand challenge," *J. Field Rob.*, vol. 23, no. 9, pp. 661–692, 2006.
- [15] C. Urmson, J. Bagnell, C. Baker, M. Hebert, A. Kelly, R. Rajkumar, P. Rybski, S. Scherer, R. Simmons, and S. Singh, "Tartan racing: A multi-modal approach to the DARPA urban challenge," Carnegie Mellon University, Tech. Rep., 2007.
- [16] J. Mårtensson, A. Alam, S. Behere, M. Khan, J. Kjellberg, K. Liang, H. Pettersson, and D. Sundman, "The development of a cooperative heavy-duty vehicle for the GCDC 2011: Team Scoop," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1033–1049, 2012.
- [17] TNO, I. TU/e, and V. S. ICT, "Grand cooperative driving challenge," Homepage: <http://www.gcdc.net/en/>, 2016.
- [18] P. Cerri, G. Sopranì, P. Zani, J. Choi, J. Lee, D. Kim, K. Yi, and A. Broggi, "Computer vision at the Hyundai autonomous challenge," in *Proceedings of the International IEEE Intelligent Transportation Systems Conference*, October 2011, pp. 777–783.
- [19] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, C. Gerdes, D. Langer, M. Hernandez, B. Muller-Bessler, and B. Huhnke, "Up to the limits: Autonomous Audi TTS," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2012, pp. 541–547.
- [20] M. Harris, "Documents confirm Apple is building self-driving car," *The Guardian*, 2015. [Online]. Available: <http://www.theguardian.com/technology/2015/aug/14/apple-self-driving-car-project-titan-sooner-than-expected>
- [21] J. Markoff, "Google cars drive themselves, in traffic," *New York Times*, 2010, October 9.
- [22] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, and C. Keller, "Making Bertha drive 2014 - An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014.
- [23] C. Ziegler, "Volvo will run a public test of self-driving cars with 100 real people in 2017," *The Verge*, 2015. [Online]. Available: <http://www.theverge.com/2015/2/23/8091455/volvo-drive-me-self-driving-car-test-2017>
- [24] M. Mchugh, "Tesla's cars now drive themselves, kinda," *Wired*, 2015. [Online]. Available: <http://www.wired.com/2015/10/tesla-self-driving-over-air-update-live/>
- [25] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [26] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebel, and F. Hundelshausen, "Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge," *J. Field Rob.*, vol. 25, no. 9, pp. 615–639, 2008.
- [27] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The Stanford entry in the urban challenge," *J. Field Rob.*, vol. 25, no. 9, pp. 569–597, 2008.
- [28] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE ITS*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [29] J. Funke and J. C. Gerdes, "Simple clothoid lane change trajectories for automated vehicles incorporating friction constraints," *Jrnl. of Dyn. Sys., Meas. and Ctrl.*, vol. 138, no. 2, pp. 021 002–021 002–9, 2016.
- [30] P. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," in *Proceedings of the European Control Conference*, July 2015, pp. 2983 – 2990.
- [31] K. Komoriya and K. Tanie, "Trajectory design and control of a wheel-less mobile robot using b-spline curve," in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, September 1989, pp. 398–405.
- [32] S. Lavalley, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Tech. Rep., 1998.
- [33] M. Pivtoraiko, R. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Rob.*, vol. 26, no. 3, pp. 308–333, 2009.
- [34] M. Cirillo, "From videogames to autonomous trucks: A new algorithm for lattice-based motion planning," in *Proceedings of the Conference on Automated Planning and Scheduling*, June 2016, pp. 6–12.
- [35] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, and V. Pratt, "Towards fully autonomous driving: Systems and algorithms," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2011, pp. 163–168.
- [36] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, and C. Geyer, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Rob.*, vol. 25, no. 8, pp. 425–466, 2008.
- [37] M. Sugeno and M. Nishida, "Fuzzy control of model car," *Fuzzy Sets Syst.*, vol. 16, no. 2, pp. 103–113, 1985.
- [38] A. De Luca, G. Oriolo, and C. Samson, *Feedback control of a nonholonomic car-like robot*. Springer Berlin Heidelberg, 1998, vol. 229, ch. 4, pp. 171–253.
- [39] R. Coulter, "Implementation of the pure-pursuit path tracking algorithm," Carnegie Mellon University, Tech. Rep., 1992.
- [40] A. Bemporad, "Model predictive control design: New trends and tools," in *Proceedings of IEEE Conference on Decision and Control*, December 2006, pp. 6678–6683.
- [41] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [42] C. Garcia, D. Prett, and M. Morari, "Model predictive control: theory and practice - a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [43] S. Di Cairano, H. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1236–1248, 2013.
- [44] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [45] P. Falcone, E. Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilisation via active front steering and braking," *Veh. Syst. Dyn.*, vol. 46, no. S1, pp. 611–628, 2008.

- [46] F. Borrelli, P. Falcone, and T. Keviczky, "MPC-based approach to active steering for autonomous vehicle systems," *Int. J. Veh. Auton. Syst.*, vol. 3(2), pp. 265–291, 2005.
- [47] P. Falcone, F. Borrelli, H. Tseng, J. Asgari, and D. Hrovat, "Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation," *Int J Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, 2008.
- [48] Y. Gao, A. Gray, J. Frasca, T. Lin, E. Tseng, J. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *Proceedings of the International Symposium on Advanced Vehicle Control*, September 2012.
- [49] V. Turri, A. Carvalho, E. Tseng, K. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *Proceedings of the International IEEE Intelligent Transportation Systems Conference*, October 2013, pp. 378–383.
- [50] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2015, pp. 1094–1099.
- [51] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *Proceedings of the International Symposium on Advanced Vehicle Control*, September 2014.
- [52] C. Beal and J. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans Control Syst Technol*, vol. 21, no. 4, pp. 1258–1269, 2013.
- [53] P. Falcone, F. Borrelli, E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *Proceedings of the American Control Conference*, June 2008, pp. 3719–3724.
- [54] J. Rawlings, D. Angeli, and C. Bates, "Fundamentals of economic model predictive control," in *Proceedings of IEEE Conference on Decision and Control*, December 2012, pp. 3851–3861.
- [55] H. Marzbani, R. Jazar, and M. Fard, "Better road design using clothoids," in *Sustainable Automotive Technologies 2014*. Springer International Publishing, 2015, pp. 25–40.
- [56] P. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Clothoid-based speed profiler and control for autonomous driving," in *Proceedings of the International IEEE Intelligent Transportation Systems Conference*, Sept. 2015, pp. 2194 – 2199.
- [57] M. Gallieri and J. Maciejowski, "Lasso MPC: Smart regulation of over-actuated systems," in *Proceedings of the American Control Conference*. IEEE, 2012, pp. 1217–1222.
- [58] M. Annergren, A. Hansson, and B. Wahlberg, "An ADMM algorithm for solving l_1 regularized MPC," in *Proceedings of IEEE Conference on Decision and Control*, 2012, pp. 4486–4491.
- [59] P. F. Lima, "Predictive control for autonomous driving, with experimental evaluation on a heavy-duty construction truck," Licentiate Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2016.
- [60] P. Lima, J. Mårtensson, and B. Wahlberg, "Stability conditions for linear time-varying model predictive control in autonomous driving," in *Proceedings of IEEE Conference on Decision and Control*, 2017, to appear.

Pedro F. Lima received the dual M.Sc degree in Electrical Engineering with a major in Systems, Control, and Robotics from Instituto Superior Técnico, University of Lisbon, Portugal and KTH Royal Institute of Technology, Stockholm, Sweden in 2013. He started his Ph.D. in the Department of Automatic Control of the School of Electrical Engineering at KTH in 2013. His research interests include heavy-duty vehicles control with main focus on model predictive control, optimal control, optimization, and path planning.

Mattias Nilsson received the M.Sc. degree in computer science and engineering from Linköping University in 2008. He has been working at Scania for several years with his main focus being on automatic gear changes for heavy trucks and busses. He currently holds a Senior Engineer position at Scania in the Autonomous Transportation System division. His main research focus is on automatic control for heavy vehicles, motion and task planning, and task scheduling.

Marco Trincavelli received both his BSc degree (2003) and his MSc degree (2006) in Computer Engineering from the Politecnico di Milano, Milan, Italy. He additionally received an MSc degree in Electrical Engineering and Computer Science from the Lund Tekniska Högskola, Lund, Sweden, in 2006. In 2010, he received his PhD in Computer Science from Örebro University, Örebro, Sweden. From 2010 to 2014, he was postdoctoral researcher at Örebro University, Örebro, Sweden. Since 2014, he has been a Development Engineer at Scania in the Autonomous Transport Solutions division. Since 2016, he is an IEEE Senior Member. His research interests include sensor fusion, robotics, and machine learning with particular focus on autonomous driving applications.

Jonas Mårtensson received the M.Sc. degree in vehicle engineering and the Ph.D. degree in automatic control from KTH Royal Institute of Technology, Stockholm, Sweden, in 2002 and 2007, respectively. He holds a tenure-track position as an Assistant Professor with the Department of Automatic Control, KTH Royal Institute of Technology. He is engaged as a Program Leader in the Integrated Transport Research Lab and he is a Thematic Leader in the area of Transport in the Information Age within the KTH Transport Platform. His main research interests are within cooperative and autonomous transport systems, in particular related to heavy-duty vehicle platooning. He is involved in several industrial collaboration projects, dealing with collaborative adaptive cruise control, look-ahead platooning, route optimization and coordination for platooning, path planning and predictive control of autonomous heavy vehicles, and related topics.

Bo Wahlberg received the M.Sc. degree in Electrical Engineering in 1983 and the Ph.D. degree in 1987 from Linköping University, Sweden. In December 1991, he became Professor of the Chair of Automatic Control at KTH Royal Institute of Technology, Stockholm, Sweden. He was a visiting professor at the Department of Electrical Engineering, Stanford University, USA, August 1997 - July 1998 and August 2009 - June 2010, and vice president of KTH 1999 – 2001. He is a Fellow of the IEEE for his contributions to system identification using orthonormal basis functions. He is a co-founder of Centre of Autonomous Systems and the Linnaeus Center ACCESS on networked systems at KTH. He is the KTH director and PI for the Wallenberg Autonomous Systems Program (WASP), that was recently granted over 200 million USD over ten years for research into autonomous systems and software development. His research interests include system identification, modeling and control of industrial processes, and statistical signal processing with applications in smart transportation systems.