

Trajectory Tracking for Autonomous Vehicles on Varying Road Surfaces by Friction-Adaptive Nonlinear Model Predictive Control

Berntorp, Karl; Quirynen, Rien; Uno, Tomoki; Di Cairano, Stefano

TR2020-005 January 08, 2020

Abstract

We propose an adaptive nonlinear model predictive control (NMPC) for vehicle tracking control. The controller learns in real time a tire force model to adapt to a varying road surface that is only indirectly observed from the effects of the tire forces determining the vehicle dynamics. Learning the entire tire model from data would require driving in the unstable region of the vehicle dynamics with a prediction model that has not yet converged. Instead, our approach combines NMPC with a noise-adaptive particle filter for vehicle state and tire stiffness estimation and a pre-determined library of tire models. The stiffness estimator determines the linear component of the tire model during normal vehicle driving, and the control strategy exploits a relation between the tire stiffness and the nonlinear part of the tire force to select the appropriate full tire model from the library, which is then used in the NMPC prediction model. We validate the approach in simulation using real vehicle parameters, demonstrate the real-time feasibility in automotive-grade processors using a rapid prototyping unit, and report preliminary results of experimental validation on a snow-covered test track.

Journal of Vehicle Systems Dynamics

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Trajectory Tracking for Autonomous Vehicles on Varying Road Surfaces by Friction-Adaptive Nonlinear Model Predictive Control

K. Berntorp^a, R. Quirynen^a, T. Uno^b and S. Di Cairano^a

^aMitsubishi Electric Research Laboratories, Cambridge, MA, 02139, USA.

^bMitsubishi Electric Corp., Adv. Technology R&D Center, Amagasaki, 661-8661, Japan.

ARTICLE HISTORY

Compiled November 18, 2019

ABSTRACT

We propose an adaptive nonlinear model predictive control (NMPC) for vehicle tracking control. The controller learns in real time a tire force model to adapt to a varying road surface that is only indirectly observed from the effects of the tire forces determining the vehicle dynamics. Learning the entire tire model from data would require driving in the unstable region of the vehicle dynamics with a prediction model that has not yet converged. Instead, our approach combines NMPC with a noise-adaptive particle filter for vehicle state and tire stiffness estimation and a pre-determined library of tire models. The stiffness estimator determines the linear component of the tire model during normal vehicle driving, and the control strategy exploits a relation between the tire stiffness and the nonlinear part of the tire force to select the appropriate full tire model from the library, which is then used in the NMPC prediction model. We validate the approach in simulation using real vehicle parameters, demonstrate the real-time feasibility in automotive-grade processors using a rapid prototyping unit, and report preliminary results of experimental validation on a snow-covered test track.

KEYWORDS

Vehicle dynamics, model predictive control, particle filtering, parameter learning.

1. Introduction

As the automotive industry progresses towards autonomous vehicles, significant attention is devoted to technologies for enabling automated driving (AD) which can also be possibly used in a more recent future in advanced driving assistance systems (ADAS). One example are technologies for trajectory tracking, that are reliable, i.e., consistent, and robust to changes in the environment, such as road and weather conditions.

Because predictive information is available in automated driving, due to sensors detecting the road ahead for several meters and the path planner generating desired vehicle motion for several seconds in the future, model predictive control (MPC) [1] is expected to hold significant promises for these applications [2,3]. Several MPC methods have been applied to vehicle steering, for both cornering and stability control and trajectory following, see e.g., [4–7], and references therein. As MPC exploits a vehicle model to perform predictions in its optimal control problem (OCP), for achieving

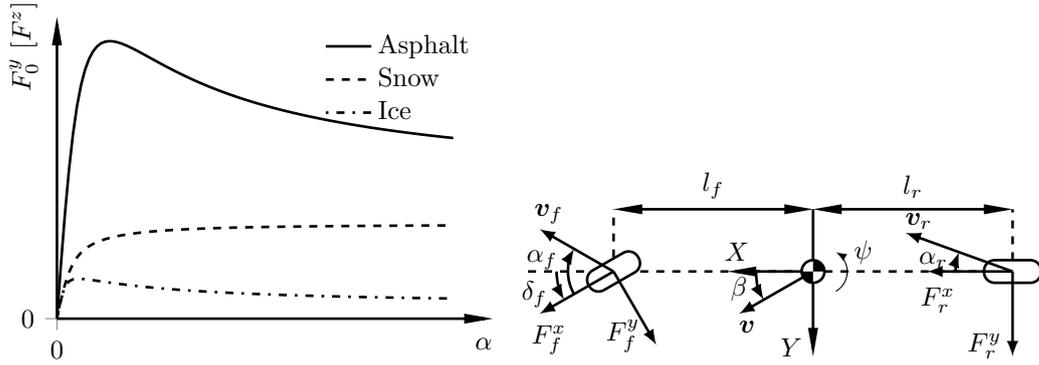


Figure 1. Left: Examples of lateral force as a function of slip angle α for asphalt, loose snow, and ice. Right: A schematic of the single-track vehicle model and related notation.

performance and robustness the model must adequately represent the current vehicle behavior. While the vehicle equation of motion in standard conditions can be described by first principles with few parameters measured on test benches, the vehicle dynamics are also affected by the surrounding environment that is continuously changing, in particular the road and the weather. Thus, a key issue in applying MPC to vehicle tracking control is its combination with estimation algorithms that can adjust the prediction model to the current environmental conditions, rapidly and using a reduced amount of data from noisy sensors. For instance, in [7] we showed that in challenging maneuvers it is imperative to have a well-informed guess about the road surface on which the car is driving, since this affects the the forces driving the vehicle motion. Knowing at least the approximate curve of the function describing the vehicle driving forces can be crucial for achieving an effective and safe vehicle operation.

In this work, we consider the specific case of adapting a Nonlinear MPC (NMPC) for vehicle control to different conditions of the road surface, i.e., to changes in the tire force functions. The driving forces due to the interaction between tire and road are described by functions that are highly nonlinear, see e.g., [8,9], and vary heavily between different surfaces. Fig. 1 shows examples of the function relating the lateral tire force to the tire side slip angle, for different surfaces. Such tire force function is approximately linear for small slip values, which are the ranges that are commonly excited when driving in normal conditions, such as highway driving on well paved and dry roads. However, when driving close to the adhesion limits, which may happen in emergency maneuvers, on unpaved road, or on wet and icy roads, the nonlinear part of the tire force function may be excited, and hence the full tire curve shape must be considered. Thus, when driving over different surfaces the time varying tire force curve needs to be identified in real time, rapidly and using noisy on-board sensors.

A complicating factor in identifying nonlinear functions is that the entire range of the function should be excited to generate data for identification. In case of the tire force function, obtaining data for the nonlinear part is challenging as it requires to drive the vehicle to the limits of its performance envelope and, usually, this is not done unless an emergency maneuver is needed or a particularly challenging surface is encountered. Furthermore, driving in the nonlinear region of the tire force function before a reliable model of the same is obtained is challenging and possibly dangerous. If the controller approaches the nonlinear region of the force curve without having a reliable model, this may cause control errors possibly leading to vehicle instability.

To avoid these issues, in this paper we leverage the dependence between the slope of

the tire force curve for small slip values, the so called tire stiffness, and the entire force function. While normally the tire stiffness can be used directly in ADAS [5,6] only for normal, i.e., linear, driving conditions, or to classify surface types for road-condition monitoring [8,10], here we combine the estimates with model knowledge to achieve reliable operation at the vehicle limits. In particular, we use a library of precomputed tire models for different road surfaces and switch between the tire models according to the current tire stiffness. The tire stiffness is estimated by a recently developed estimator [11] based on particle filtering. Our tire-stiffness estimator operates under normal driving conditions, i.e., when the slip values are small, and hence does not require the vehicle to operate at the limit of performance while the force curve is being identified. Using the estimated stiffness to discriminate the current model among the ones in the library allows us to obtain information on the entire tire force curve while using only data from the linear region.

The resulting method that combines data-based and model-based techniques is efficient in terms of both data requirements and computational requirements, which is appealing for automotive applications where the sensing and computational resources are limited. Since we use data only to discriminate between models and we operate an NMPC which is a feedback algorithm and hence naturally compensate for prediction model errors, few data points are enough to obtain an effective and safe behavior of the closed-loop system. Furthermore, we are identifying a limited number of parameters, i.e., the tire stiffnesses, with an efficient implementation of a particle filter [11], and we have developed an efficient block-sparse QP solver [12] for use within the RTI framework of nonlinear optimal control. Therefore, our adaptive NMPC algorithm is real-time feasible for current automotive micro-controllers, as demonstrated here by implementing it on a dSPACE MicroAutoBox-II rapid prototyping unit.

Outline: The rest of the paper is organized as follows. Sec. 2 describes the vehicle model, the considered sensor, actuator, and computational platform setup, and the problem definition. Sec. 3 introduces the NMPC formulation, and Sec. 4 describes the particle filter algorithm for estimating the tire stiffness. The method for adapting the NMPC model based on the estimated tire stiffness and the library of models is described in Sec. 5, which is followed by a simulation study, the assessment of real-time feasibility, and some preliminary results of in-vehicle experiments, in Sec. 6. Our conclusions are summarized in Sec. 7.

Notation: the notation is standard, with only few exceptions. $\mathcal{R}(a)$ denotes the 2D rotation matrix of angle a . Vectors are shown in bold, \mathbf{x} , we denote the stacking of two vectors \mathbf{a} , \mathbf{b} by (\mathbf{a}, \mathbf{b}) , and constraints between vectors are intended componentwise. We denote a family of functions parametrized by the parameter vector $\boldsymbol{\theta}$ as $\mathbf{f}_{\boldsymbol{\theta}}$. The symbol \approx reads as approximately equals, e.g., to a first order, while \propto denotes equality up to a constant scaling coefficient. We denote a Gaussian distribution with mean m covariance P by $\mathcal{N}(m, P)$, and a random variable \mathbf{y} distributed according to such distribution as $\mathbf{y} \sim \mathcal{N}(m, P)$, while $p(\mathbf{y})$, $p(\mathbf{y}|\mathbf{x})$ denote the (generic) probability density function of \mathbf{y} , and the (generic) probability density function of \mathbf{y} conditional to \mathbf{x} . For a continuous-time signal $\mathbf{x}(t)$ sampled with period T_s , x_k denotes the k^{th} sample, i.e., $\mathbf{x}_k = \mathbf{x}(kT_s)$ and $\mathbf{x}_{k+h|k}$ is the value of \mathbf{x} predicted h steps ahead from k , i.e., the predicted value of $\mathbf{x}((k+h)T_s)$ based on $\mathbf{x}(kT_s)$.

2. Modeling and Problem Description

First, we describe the model of the vehicle which is later used to derive the estimator model and the prediction model of the NMPC. Then, we discuss the problem definition and the physical platform considered, in term of sensors, actuators, and computational capabilities, and we describe the problem that our control system addresses.

2.1. Vehicle Model Dynamics

The vehicle model is composed of a chassis model describing the motion of the rigid body due to the forces generated at the tires, a tire model describing what forces the tires generate depending on the chassis and wheels velocities, and a wheel model describing how the wheel speed changes as function of the acceleration/braking torques.

For the chassis, we consider the standard single-track model, where the left and right track of the car are lumped into a single centered track, shown in Fig. 1. Hence, only a single front and a single rear tire are considered, and roll and pitch dynamics are ignored, resulting in two translational and one rotational degrees of freedom. While for performance driving it may be advantageous to use a double-track chassis model, which includes lateral and longitudinal load transfer [13], in [7,13] the single-track model was shown to be sufficiently accurate for regular driving conditions, including when tire forces are in the nonlinear region, because in such conditions the roll and pitch angles remain relatively small. Similarly, the single track model seems sufficient in most evasive maneuvers, because the focus of such maneuvers is on preserving safety rather than achieving optimality, and hence a high precision model is often unnecessary. On the other hand, the single-track model results in a reduced computing load, which is always desirable in automotive applications [3], particularly for evasive maneuvers.

Taking the longitudinal and lateral velocities in the vehicle frame, v^X , v^Y , and the yaw rate, $\dot{\psi}$, as states, the single-track model is described by

$$\dot{v}^X - v^Y \dot{\psi} = \frac{1}{m}(F_f^x \cos(\delta_f) + F_r^x - F_f^y \sin(\delta_f)), \quad (1a)$$

$$\dot{v}^Y + v^X \dot{\psi} = \frac{1}{m}(F_f^y \cos(\delta_f) + F_r^y + F_f^x \sin(\delta_f)), \quad (1b)$$

$$I_{zz} \ddot{\psi} = l_f F_f^y \cos(\delta_f) - l_r F_r^y + l_f F_f^x \sin(\delta_f), \quad (1c)$$

where F_i^x , F_i^y are the total longitudinal/lateral forces in the tire frame for the lumped left and right tires, and the subscripts $i = f, r$ indicate front and rear, respectively, m is the vehicle mass, I_{zz} is the vehicle inertia about the vertical axis, and δ_f is the front wheel (road) steering angle. The vehicle position in global coordinates $p = (p^{\mathbf{X}}, p^{\mathbf{Y}})$ is obtained from the kinematic equation

$$\begin{bmatrix} \dot{p}^{\mathbf{X}} \\ \dot{p}^{\mathbf{Y}} \end{bmatrix} = \mathcal{R}(\psi) \begin{bmatrix} v^X \\ v^Y \end{bmatrix}. \quad (2)$$

The response from wheel angle command δ_f^{cmd} to wheel angle actuated by the steering mechanism is modeled as a first order system with time constant τ_s

$$\dot{\delta}_f = -\frac{1}{\tau_s} (\delta_f - \delta_f^{\text{cmd}}). \quad (3)$$

The tire model describes how the tire forces F_i^x and F_i^y in (1) are generated. The

nominal tire forces $F_{0,i}^x$ and $F_{0,i}^y$, i.e., the forces under pure longitudinal or lateral slip conditions, can be described using the Magic Formula model [14],

$$\begin{aligned} F_{0,i}^x &= \mu_i^x F_i^z \sin(C_i^x \arctan(B_i^x(1 - E_i^x)\lambda_i + E_i^x \arctan(B_i^x \lambda_i))), \\ F_{0,i}^y &= \mu_i^y F_i^z \sin(C_i^y \arctan(B_i^y(1 - E_i^y)\alpha_i + E_i^y \arctan(B_i^y \alpha_i))), \end{aligned} \quad (4)$$

where α_i are the slip angles, λ_i are the slip ratios, F_i^z are the normal forces resting on the wheels, μ_i^x and μ_i^y are the friction coefficients, and B_i^h , C_i^h and E_i^h , $i \in \{f, r\}$, $h \in \{x, y\}$, are the stiffness, shape, and curvature factor, respectively. In what follows, we use the short-hand notation $\boldsymbol{\theta} = \{\mu_i^h, B_i^h, C_i^h, E_i^h\}_{i=f,r}^{h=x,y}$ to denote the set tire/road parameters, with values that vary with external conditions, such as, among others, road type, temperature, weather, tire pressure, so that $\boldsymbol{\theta}$ is not exactly known. In (4), the normal forces resting on lumped front/rear wheels F_i^z , $i \in \{f, r\}$, are $F_f^z = mgl_r/l$, $F_r^z = mgl_f/l$, where g is the gravity acceleration, l_f , l_r are the distances of front and rear axles from the center of gravity, and $l = l_f + l_r$ is the vehicle wheel base.

Under combined slip conditions, i.e., when both λ and α are nonzero, the coupling between longitudinal and lateral forces may be represented by the friction ellipse (FE)

$$F_i^y = F_{0,i}^y \sqrt{1 - \left(\frac{F_{0,i}^x}{\mu_i^x F_i^z}\right)^2}, \quad i \in \{f, r\}. \quad (5)$$

Even though in (5) the longitudinal force does not explicitly depend on the lateral slip so that more accurate could be used, see, e.g., [14,15], the FE model is desirable for its simplicity, and it has been proven satisfactory for the purposes pursued in this paper.

The slip angles α_i and slip ratios λ_i in (4) are defined as [14],

$$\dot{\alpha}_i \frac{\sigma}{v_i^x} + \alpha_i = -\arctan\left(\frac{v_i^y}{v_i^x}\right), \quad (6)$$

$$\lambda_i = \frac{R_w \omega_i - v_i^x}{v_i^x}, \quad i \in \{f, r\}, \quad (7)$$

where σ is the relaxation length, R_w is the wheel radius, ω_i is the wheel angular velocity for wheel i , and v_i^x and v_i^y are the longitudinal and lateral wheel velocities for wheel i in the coordinate system of the wheel. Given the velocity vector at the center of mass, $v = [v^X \ v^Y]^\top$, the velocity vectors at the wheels are

$$\begin{bmatrix} v_i^x \\ v_i^y \end{bmatrix} = \mathcal{R}(\delta_i)^\top \begin{bmatrix} v^X \\ v^Y + c_i \dot{\psi} \end{bmatrix}, \quad i \in \{f, r\}, \quad c_f = l_f, \quad c_r = -l_r. \quad (8)$$

Finally, the wheel dynamics are given by

$$T_i - I_w \dot{\omega}_i - F_i^x R_w = 0, \quad i \in \{f, r\}, \quad (9)$$

where I_w is the wheel inertia and $T_i = T_i^e - T_i^b$ is the torque on wheel $i \in \{f, r\}$, due to the engine, T_i^e , and brake T_i^b . We model the engine torque response from a torque command $T_i^{e,\text{cmd}}$ as a first order system [16] with time constant $\tau_e > 0$

$$\dot{T}_i^e = -\frac{1}{\tau_e}(T_i^e - T_i^{e,\text{cmd}}), \quad i \in \{f, r\}. \quad (10)$$

Since T_i^e is the net torque, it can take negative values due to friction, losses and auxiliary loads on the engine [16], thus providing engine braking. On the other hand, since the brake actuation tends to be significantly faster than the engine, we model the brake torque T_i^b as instantaneous, thus ignoring the brake circuit pressure dynamics. Different powertrain and braking system configurations may be modeled by adding constraints on the commands for engine and brake torques on each single wheels.

2.2. Measurements, Controls, and Problem Definition

In this paper, we consider a production-like sensor setup where we measure the vehicle position $(p^{\mathbf{X}}, p^{\mathbf{Y}})$ and the yaw (heading) ψ by GPS, the individual wheel speeds $\omega_{i,j}$, $i \in \{f, r\}$, where $j \in \{l, r\}$ denotes left and right tires, by wheel encoders, which are used to determine also the vehicle speed, approximated equal to v^X . We measure the vehicle longitudinal, $a^X = \dot{v}^X - v^Y \dot{\psi}$, and lateral, $a^Y = \dot{v}^Y + v^X \dot{\psi}$, acceleration and the yaw rate $\dot{\psi}$ by an automotive grade inertial measurement unit and the front road wheel steering angle δ_f by a relative encoder. The control inputs are the front road wheel steering angle command δ_f^{cmd} and the command for the front wheel torque from engine $T_f^{e,\text{cmd}}$, and possibly the brake torques T_i^b , $i \in \{f, r\}$.

We target for implementation automotive micro-controllers, which, due to harsh operating conditions, hard real-time requirements, and cost considerations, are significantly less capable of desktop computers [3] in terms of memory and speed.

Under the above conditions, the objective of this paper is to design a control strategy that makes the vehicle motion follow a time-dependent reference trajectory $(p_{\text{ref}}^{\mathbf{X}}(\cdot), p_{\text{ref}}^{\mathbf{Y}}(\cdot), \psi_{\text{ref}}(\cdot), v_{\text{ref}}^X(\cdot))$, possibly generated in real time with an adequate preview, while operating over different surfaces and environmental conditions. Since some of the vehicle states, e.g., v^Y , are not directly measured and the tire parameters θ are not exactly known and change over time, the control strategy needs to estimate the vehicle state and the tire parameters as the controller operates. The entire control strategy must be able to operate in real time on computational platforms with capabilities similar to today's automotive micro-controllers, and hence we aim at limiting the amount of computations and data storage needed by our approach.

The proposed control strategy is schematically depicted in Fig. 2, where the estimator (PF) uses vehicle measurements (y) to estimate the vehicle state (x) and the tire stiffness (C), and provides the former to the vehicle controller, and the latter to the library (LIB) of tire models for selecting the tire model parameters (θ). The controller (NMPC) uses the tire model together with the vehicle model and the vehicle state to determine the control actions (u) that are actuated at the vehicle (VS). The three blocks in the control strategy are described in the next three sections.

3. Nonlinear MPC for Real-Time Vehicle Control

Since our NMPC must track a time varying reference trajectory, for correctly formulating the optimal control problem we first define as decision variable the rate of change of steering command $\dot{\delta}_f^{\text{cmd}}$. Based on the chassis model, tire model, wheel model, and actuator response (1)–(10), we can write the complete model as a compact

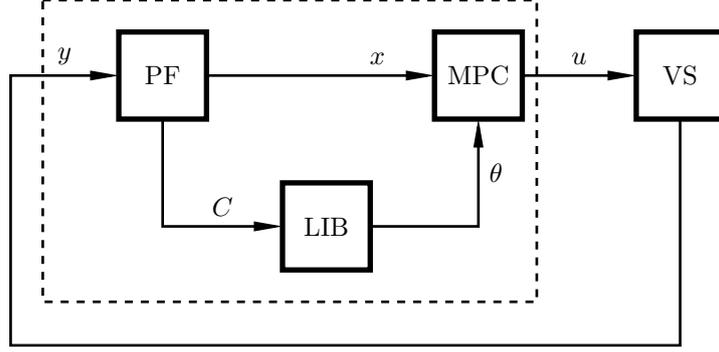


Figure 2. Architecture for the adaptive control system proposed in this paper.

set of ordinary differential equations as follows

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in [0, T], \quad (11)$$

where the vehicle dynamics are parameterized by the set of estimated tire parameters $\boldsymbol{\theta} = \{\mu_i^h, B_i^h, C_i^h, E_i^h\}_{i=f,r}^{h=x,y}$ according to Eq. (4), and for every parameter vector $\boldsymbol{\theta}$, the function $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ is twice continuously differentiable in all its arguments. For the purpose of the NMPC formulation, the state and control vector read as

$$\begin{aligned} \mathbf{x}(t) &:= \left[p^{\mathbf{X}}, p^{\mathbf{Y}}, \psi, v^X, v^Y, \dot{\psi}, \delta_f, \omega_f, \omega_r, \alpha_f, \alpha_r, \delta_f^{\text{cmd}}, T_f^e, T_r^e, \tau \right]^{\top}, \\ \mathbf{u}(t) &:= \left[\dot{\delta}_f^{\text{cmd}}, T_f^{e,\text{cmd}}, T_r^{e,\text{cmd}}, \dot{\tau}, s \right] \in \mathbb{R}^{n_u}, \end{aligned} \quad (12)$$

such that $n_x = 15$ and $n_u = 5$, and in which τ denotes a path variable to parameterize the reference trajectory [17] as will be described further. The dynamic equation for this path variable reads as $\frac{d\tau}{dt} = \dot{\tau}$, in which $\dot{\tau}$ is an additional control variable in the NMPC problem formulation. We then introduce the tracking-type optimal control problem formulation in continuous time,

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_0^T \|\mathbf{F}(\mathbf{x}(t), \mathbf{u}(t)) - \mathbf{y}_{\text{ref}}(t)\|^2 dt \quad (13a)$$

$$\text{s.t. } \mathbf{x}(0) = \hat{\mathbf{x}}_0, \quad (13b)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in [0, T], \quad (13c)$$

$$0 \geq \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in [0, T], \quad (13d)$$

$$0 \geq \mathbf{r}(\mathbf{x}(T)). \quad (13e)$$

The tracking objective in Eq. (13a) consists of a nonlinear least-squares type Lagrange term. For simplicity, T denotes both the control and prediction horizon length. The parametric optimization problem (13) depends on the current state estimate $\hat{\mathbf{x}}$ through Eq. (13b). Eqs. (13d) and (13e) denote the path and terminal inequality constraints, respectively. The tire parameter vector $\boldsymbol{\theta}$ in Eq. (13c) is estimated as described later, and the estimated value is kept constant for the entire control horizon $t \in [0, T]$. In order for the NMPC controller to achieve offset-free tracking under constant disturbances and model mismatch [1], and possibly compensating for a steering system

offset, Eq. (3) in the front wheel angle dynamics (13c) is adjusted to

$$\dot{\delta}_f = -\frac{1}{\tau_s} \left(\delta_f - (\delta_f^{\text{cmd}} + \delta_f^{\text{off}}) \right), \quad (14)$$

where δ_f^{off} is an input additive offset value that is estimated online.

3.1. Objective Function and Inequality Constraints

The cost function in (13a) allows us to formulate any standard tracking-type objective. In our optimal control problem formulation, the nonlinear least squares cost function consists of three terms

$$\int_0^T \left(\|\mathbf{F}_{\text{ref}}(\mathbf{x}(t)) - \mathbf{y}_{\text{ref}}(\tau, \mathbf{d})\|_{\mathbf{W}}^2 + \|\mathbf{x}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{u}(t)\|_{\mathbf{R}}^2 + r_s s(t) \right) dt, \quad (15)$$

including a term for tracking a reference motion plan and two regularization terms for penalizing state and control variables, where $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{R} \in \mathbb{R}^{n_u \times n_u}$ are the corresponding positive definite weighting matrices on the outputs, from which the state is observable. The reference motion plan is provided by a smooth function $\mathbf{y}_{\text{ref}}(\tau, \mathbf{d})$ that depends on the path variable τ as well as on additional parameters \mathbf{d} . By using this parametric formulation, a continuous time approximation can be constructed based on a sequence of discrete time sample points. We define the tracking function in the objective based on a polynomial approximation of the reference trajectories

$$\mathbf{F}_{\text{ref}}(\mathbf{x}(t)) - \mathbf{y}_{\text{ref}}(\tau, \mathbf{d}) = \begin{bmatrix} e^{\mathbf{Y}}(t, \tau, \mathbf{d}) \\ p^{\mathbf{X}}(t) - p_{\text{ref}}^{\mathbf{X}}(\tau, \mathbf{d}_X) \\ p^{\mathbf{Y}}(t) - p_{\text{ref}}^{\mathbf{Y}}(\tau, \mathbf{d}_Y) \\ \psi(t) - \psi_{\text{ref}}(\tau, \mathbf{d}_\psi) \\ v^{\mathbf{X}}(t) - v_{\text{ref}}^{\mathbf{X}}(\tau, \mathbf{d}_v) \end{bmatrix} = \begin{bmatrix} e^{\mathbf{Y}}(t, \tau, \mathbf{d}_X, \mathbf{d}_Y, \mathbf{d}_\psi) \\ p^{\mathbf{X}}(t) - \sum_{i=0}^{n_X} \mathbf{d}_X^i \tau^i \\ p^{\mathbf{Y}}(t) - \sum_{i=0}^{n_Y} \mathbf{d}_Y^i \tau^i \\ \psi(t) - \sum_{i=0}^{n_\psi} \mathbf{d}_\psi^i \tau^i \\ v^{\mathbf{X}}(t) - \sum_{i=0}^{n_v} \mathbf{d}_v^i \tau^i \end{bmatrix}, \quad (16)$$

where the path error $e^{\mathbf{Y}}(\cdot) = \cos(\psi_{\text{ref}}(\tau, \mathbf{d}_\psi)) (p^{\mathbf{Y}} - p_{\text{ref}}^{\mathbf{Y}}(\tau, \mathbf{d}_Y)) - \sin(\psi_{\text{ref}}(\tau, \mathbf{d}_\psi)) (p^{\mathbf{X}} - p_{\text{ref}}^{\mathbf{X}}(\tau, \mathbf{d}_X))$ is the orthogonal distance to the parameterized reference trajectory. The prediction model state is observable through the output function \mathbf{F}_{ref} , whose components are also weighted with a positive definite matrix $\mathbf{W} \in \mathbb{R}^{5 \times 5}$. The dependence of the reference trajectory on the path variable $\tau(t)$, instead of the time variable t , is quite standard in path following MPC [17]. This formulation results in an additional degree of freedom for the NMPC controller to result in an improved tracking performance. More specifically, tracking a time-dependent motion plan can sometimes lead to large tracking errors even when the resulting path closely corresponds to the planned path, e.g., caused by the vehicle slowing down or speeding up relative to the reference motion.

The path constraints in the NMPC problem formulation consist of geometric and physical limitations of the system, such as constraints on the distance of the vehicle position to the parameterized reference trajectory. In practice, it is important to reformulate these requirements as soft constraints, based on the slack variable s , since otherwise the problem may become infeasible due to unknown disturbances and modeling errors, and the controller will cease operating. Thus, the path constraints in (13d) include the soft bounds on distance to the reference, on the front wheel angle and on

the lateral acceleration $a^Y = (F_f^y \cos(\delta_f) + F_r^y + F_f^x \sin(\delta_f))/m$

$$-\bar{e}^Y \leq e^Y + s, \quad -\bar{\delta}_f \leq \delta_f + s, \quad -\bar{a}^Y \leq a^Y + s, \quad (17a)$$

$$e^Y - s \leq \bar{e}^Y, \quad \delta_f - s \leq \bar{\delta}_f, \quad a^Y - s \leq \bar{a}^Y. \quad (17b)$$

In addition, these path constraints (13d) include the following hard constraints on control input variables and on the path variable derivative to ensure that the vehicle does not deviate excessively from the reference path

$$\underline{T}_f^{e,\text{cmd}} \leq T_f^{e,\text{cmd}} \leq \bar{T}_f^{e,\text{cmd}}, \quad \underline{T}_r^{e,\text{cmd}} \leq T_r^{e,\text{cmd}} \leq \bar{T}_r^{e,\text{cmd}}, \quad (18a)$$

$$-\bar{\delta}_f^{\text{cmd}} \leq \dot{\delta}_f^{\text{cmd}} \leq \bar{\delta}_f^{\text{cmd}}, \quad 1 - \Delta\bar{\tau} \leq \dot{\tau} \leq 1 + \Delta\bar{\tau}, \quad 0 \leq s. \quad (18b)$$

Defining soft constraints as in (17) allows for penalizing in the cost function the L1 norm, i.e., $r_s |s(t)| = r_s s(t)$ in (15) given that $s(t) \geq 0$, which is an exact penalty function and does not add nonlinearities [18].

The terminal constraints and the corresponding terminal cost can be designed to ensure closed-loop stability properties as discussed in [1]. Instead, here we rely on selecting a horizon long enough with respect to the system dynamics, which, under reasonable assumptions on the NMPC value function and on the equilibrium being in the interior of the feasible region of the OCP, ensures asymptotic stability and some degrees of robustness to the closed-loop system [19].

3.2. Online NMPC Algorithm and Software Implementation

In order to solve (13) we transform the infinite dimensional OCP into a nonlinear program (NLP) by control and state parameterization using direct multiple shooting [20]. We discretize in time using an integration scheme to simulate the differential equations (11) on each of N shooting intervals that are defined by a grid of consecutive equidistant time points t_i for $i = 0, \dots, N$, $t_{i+1} - t_i = \frac{T}{N}$, and we enforce a piecewise constant control $\mathbf{u}(t) = \mathbf{u}_i$ for $t \in [t_i, t_{i+1})$. The discrete-time dynamics $\mathbf{x}_{i+1} = F_i(\mathbf{x}_i, \mathbf{u}_i)$ are defined for each shooting interval $[t_i, t_{i+1}]$, by an explicit or implicit integration formula [21]. Implicit integration schemes are typically preferred because of improved accuracy and numerical stability properties.

We solve the NLP at each control time step by a tailored implementation of sequential quadratic programming (SQP) known as the real-time iterations (RTI) scheme [22] using the open-source ACADO code generation tool [21]. RTI performs one SQP iteration per control time step, and uses a continuation-based warm starting of the state and control trajectories from one time step to the next. The stability of the resulting closed-loop system can be guaranteed also in presence of inaccuracies and external disturbances under reasonable assumptions [22]. RTI may not approximate well the NLP if the problem is linearized far from a local minimum. However, in our view of the autonomous driving system [23], the MPC tracks a trajectory generated by a motion planner to be a suitable reference for linearization, e.g., kinematically feasible and constraint aware. Thus, RTI seems a suitable approach for this application. We use the QP solver PRESAS [12,24], which applies block-structured factorization techniques with low-rank updates to preconditioning of an iterative solver within a primal active-set algorithm with tailored initialization methods. This results in a simple, efficient and reliable QP solver suitable for embedded control hardware.

To compensate [3] for a time delay T_d due to vehicle network communication and actuator interface at time t , we define $\hat{\mathbf{x}}_0$ as equal to the predicted state value $\hat{\mathbf{x}}(t+T_d)$, computed from the current state estimate $\hat{\mathbf{x}}(t)$ and past input signals $\mathbf{u}(\cdot)$, which are stored in a buffer. Such time-delay compensation has been proven important for maintaining both robustness and high control performance.

4. Particle Filter for Stiffness Estimation

In this section we present our method for real-time estimation of the tire stiffnesses, which are subsequently used in Sec. 5 for selecting the tire parameters in (13) using a library of tire models. From Fig. 2, (13), and hence its solution, depends on the tire parameters in the Pacejka model through (13c), which includes (4). The tire-stiffness estimator (i.e., the estimator of the initial slope of the tire-force curve in Fig. 1) is based on a recently developed adaptive particle-filter approach. Here, we briefly outline the formulation, see [11] for a complete discussion.

4.1. Estimation Model

The method employs the single-track vehicle model (1) and a linear approximation of the front and rear tire forces,

$$F^x \approx C_{s,i}^x \lambda_i, \quad F^y \approx C_{s,i}^y \alpha_i, \quad i \in \{f, r\}, \quad (19)$$

where $C_{s,i}^x$ and $C_{s,i}^y$ are the longitudinal and lateral stiffness for the front and rear wheel, respectively. The slip ratios are defined as in (7), but unlike (6) the slip angles are assumed to be small such that they can be approximated by

$$\alpha_f \approx \delta_f - \frac{v^Y + l_f \dot{\psi}}{v^X}, \quad \alpha_r \approx \frac{l_r \dot{\psi} - v^Y}{v^X}. \quad (20)$$

The small-angle approximations (20) are not necessary, but they simplify the estimation problem and are valid when the tire forces remain in the linear region.

In (19) the stiffnesses are decomposed into a nominal part and an unknown part,

$$C_s^x = C_{s,n}^x + \Delta C_s^x, \quad C_s^y = C_{s,n}^y + \Delta C_s^y, \quad (21)$$

where $C_{s,n}$ is the nominal, known, value of the stiffness, for example, a priori determined on a nominal surface, and ΔC_s is a time-varying, unknown part. We include the unknown stiffness components into the vector $\mathbf{w}_k \in \mathbb{R}^{n_w}$, which we model as random process noise acting on the otherwise deterministic system. The noise term \mathbf{w}_k is assumed Gaussian distributed according to $\mathbf{w}_k \sim \mathcal{N}(\mathbf{C}_k, \mathbf{\Sigma}_k)$, where \mathbf{C}_k and $\mathbf{\Sigma}_k$ are the unknown, usually time varying, mean and covariance. Inserting (19)–(21) into (1) and discretizing with sampling period T_s gives the discrete-time dynamics as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) \mathbf{w}_k, \quad (22)$$

where $\mathbf{x} = [v^X \ v^Y \ \dot{\psi}]^\top$ is the vehicle state and $\mathbf{u} = [\omega_{i,j} \ \delta_f]^\top$, $i \in \{f, r\}$, $j \in \{l, r\}$ are the inputs to the estimator.

The tire stiffness parameters affect the vehicle state, which is only partially (and indirectly) observed through the inertial sensors. Hence, to estimate the tire stiffness we must also estimate the vehicle state. To this end, we are interested in estimating both the state \mathbf{x}_k and the parameters $\mathbf{C}_k, \mathbf{\Sigma}_k$, being the mean and variance of the process noise \mathbf{w}_k . One interpretation of the parameters is that the mean models the stiffness variations based on the surface type, such as asphalt or snow, and the variance models the uncertainty due to either variations on a surface, such as road unevenness, patches of loose snow, road in mixed conditions, or other unmodeled effects. The estimator uses the acceleration, a^X, a^Y , and yaw-rate measurements $\dot{\psi}$ as measurements, forming the measurement vector $\mathbf{y}_k = [a^X \ a^Y \ \dot{\psi}]^\top \in \mathbb{R}^{n_y}$, with $n_y = 3$. Note that a^X and a^Y can be extracted from the right-hand sides of (1a) and (1b), and the last measurement is the yaw rate. Automotive-grade inertial sensors usually have a slowly time-varying bias, which must be modeled for any realistic implementation. We model the bias $\mathbf{b}_k \in \mathbb{R}^{n_b}$, $n_b = 3$, of the inertial measurements as a random walk, which results in

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{b}_k + \mathbf{d}(\mathbf{x}_k, \mathbf{u}_k)\mathbf{w}_k + \mathbf{e}_k, \quad (23)$$

where the measurement noise is modeled as Gaussian noise with zero mean and covariance \mathbf{R} , $\mathbf{e}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$.

Thus, the estimation problem consists of estimating the vehicle state \mathbf{x}_k and stiffness parameters $\mathbf{C}_k, \mathbf{\Sigma}_k$ using the estimation model (22) and (23). Note that because of the inertial sensor measurements, the stiffness components enter both in the vehicle model and the measurement model through \mathbf{w}_k , which implies that the estimation model has a dependence between the process and measurement noise [11].

Remark 1. Because of the approximation (19), the tire-stiffness estimator performs under the assumption of moderate steering angles and sufficiently small driving/braking torques. Thus, in the implementation the estimator is activated only when the estimated slip angles are such that (20) holds within some predefined threshold.

4.2. Formulation of Joint State and Tire-Stiffness Estimation

The considered estimation problem is in general hard to solve for multiple reasons. First, the estimation quality of the vehicle state affects the identification of the noise statistics, and vice versa. Second, the measurements from the inertial sensors are biased and significantly noisy. Third, the estimation model for our target application shows dependence between the process and measurement noises. The result is a highly non-Gaussian estimation problem.

Since both the dynamic model (22) and measurement model (23) include stochastic disturbances, we formulate the problem in a Bayesian framework as a joint state and parameter estimation problem, and use a previously developed computationally efficient particle filter-based approach that accounts for the aforementioned bias and noise dependence [11]. In a Bayesian framework, the joint state and tire-stiffness estimation can be formulated as approximating the joint density

$$p(\mathbf{b}_k, \mathbf{C}_k, \mathbf{\Sigma}_k, \mathbf{x}_{0:k} | \mathbf{y}_{0:k}), \quad (24)$$

i.e., the posterior density function of the variables of interest given the measurement

history $\mathbf{y}_{0:k} = \{\mathbf{y}_0, \dots, \mathbf{y}_k\}$. We decompose (24) as

$$p(\mathbf{b}_k, \mathbf{C}_k, \boldsymbol{\Sigma}_k, \mathbf{x}_{0:k} | \mathbf{y}_{0:k}) = p(\mathbf{b}_k | \mathbf{C}_k, \boldsymbol{\Sigma}_k, \mathbf{x}_{0:k}, \mathbf{y}_{0:k}) p(\mathbf{C}_k, \boldsymbol{\Sigma}_k | \mathbf{x}_{0:k}, \mathbf{y}_{0:k}) p(\mathbf{x}_{0:k} | \mathbf{y}_{0:k}). \quad (25)$$

The three densities at the right-hand side of (25) are estimated recursively. The key idea is that given the state trajectory, we can update the sufficient statistics of the unknown tire-stiffness parameters. Given the parameters and the state trajectory, the bias estimation consists of applying a Kalman filter conditioned on the state trajectory and tire-stiffness parameters (cf. [11]). From the particles, we extract estimates of both vehicle state \mathbf{x}_k , stiffness estimate \mathbf{C}_k , and associated covariance $\boldsymbol{\Sigma}_k$.

5. Adaptation of NMPC via Particle Filter and Model Library

Our approach exploits a pre-stored library of M sets $\{\boldsymbol{\theta}_j\}_{j=1}^M$ of predetermined tire parameters defining the tire model (4), which are used in the NMPC, see Fig. 2. Such tire parameters can be determined using a testbench or from field tests. Determining the tire parameters for every combination of tire and vehicle and then during runtime distinguishing between tire parameters for different setups would be ideal, since in general there are several different parameter sets for the same surface that lead to similar tire-force curves. However, this is intractable with the sensor setup in current production vehicles. A further complicating factor is that the correspondence between the tire stiffness and the peak friction is not one-to-one, as there are experimental data indicating that wet asphalt can have a larger initial slope but a smaller peak friction coefficient, due to that the peak is obtained at smaller slip values [8].

However, for our purposes where the final objective is to maintain good reference-tracking performance, the key element is to have a nominal parameter set that distinguishes between surfaces. The tire stiffness can be used for such a differentiation [8,9]. For instance, the optimal vehicle behavior can fundamentally differ between snow and asphalt, but typically it is less important whether the asphalt is dry or wet [13,15]. Thus, loosely speaking, we can use the stiffness to differentiate among the pre-stored models, such that we capture the tire behavior precisely enough. Then, we can rely on the, appropriately calibrated, feedback control system to provide robustness to the remaining parameter uncertainties caused by different vehicle and tire combinations.

We use the estimates $\hat{C}_s^{x,y}$ of the tire stiffness in the following way. From a linearization of the Pacejka tire model (4), we get for the lateral tire force

$$F^y \approx \mu_i^y F_i^z C_i^y B_i^y \alpha_i, \quad (26)$$

and similarly for the longitudinal direction. Ignoring higher order terms, i.e., setting (19) equal to (26), results in

$$\mu_i^y F_i^z C_i^y B_i^y = \hat{C}_{s,i}^y. \quad (27)$$

The vertical force in (27) is given by the vehicle parameters and the right-hand side is given by the estimated value from the stiffness estimator. To select the tire parameter set $\boldsymbol{\theta}_j$ to be used in the NMPC, we determine the set of parameters that fits best to the estimated stiffness value. A straightforward optimization criterion that minimize the difference between the estimated tire stiffness and stiffness computed from the

library of tire parameters is

$$\boldsymbol{\theta}^* = \arg \min_{j \in \{1, M\}} |\mu_{i,j}^y F_i^z C_{i,j}^y B_{i,j}^y - \hat{C}_{s,i}^y|. \quad (28)$$

However, (28) does not take into account the uncertainty of the estimate and would also lead to a symmetry when choosing between two surfaces. Instead, in terms of vehicle stability, it is typically worse to overestimate the available friction than to underestimate it [7]. In determining the parameter set $\boldsymbol{\theta}^*$, we therefore propose two alternative approaches that account for the desired asymmetry in determining the tire parameters and incorporate the uncertainty of the estimates.

In the first approach, we order the parameter sets with increasing peak friction coefficient, starting with the parameters corresponding to the lowest-friction surface, $\boldsymbol{\theta}_1$. We use the normalized residual,

$$\epsilon_k = \boldsymbol{\Sigma}_k^{-1/2} (\mu_{i,1}^y F_i^z C_{i,1}^y B_{i,1}^y - \hat{C}_{s,i}^y) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (29)$$

and the test statistic

$$T(\mu_{i,1}^y F_i^z C_{i,1}^y B_{i,1}^y) = \frac{(\mu_{i,1}^y F_i^z C_{i,1}^y B_{i,1}^y - \hat{C}_{s,i}^y)^2}{\Sigma_{i,k}}, \quad (30)$$

where $\Sigma_{i,k}$ is the i th diagonal element of $\boldsymbol{\Sigma}_k$ corresponding to the front or rear lateral stiffness. Then, approximately [25],

$$T(\mu_{i,1}^y F_i^z C_{i,1}^y B_{i,1}^y) \sim \chi_\eta^2(1) \quad (31)$$

where $\chi_\eta^2(1)$ is the Chi-squared distribution with one degree of freedom and η is the significance level. We choose the parameters $\boldsymbol{\theta}_1$ corresponding to the lowest-friction surface as the parameters if

$$T(\mu_{i,1}^y F_i^z C_{i,1}^y B_{i,1}^y) > \chi_\eta^2(1). \quad (32)$$

Otherwise, we proceed for increasing peak friction until a parameter set is found.

The selection (32) based on outlier detection will always choose the parameter set corresponding to the lower-friction surface. While this is good from a vehicle-stability control perspective, it may be conservative. An approach that is not so heavily biased but still accounts for the uncertainty in the stiffness estimation is to maximize the likelihood. This results in the selection criteria

$$\boldsymbol{\theta}^* = \arg \max_{j \in \{1, M\}} \mathcal{N}(\mu_{i,j}^y F_i^z C_{i,j}^y B_{i,j}^y | C_k, \boldsymbol{\Sigma}_k). \quad (33)$$

Algorithm 1 summarizes the proposed control strategy.

Remark 2. We have focused on the lateral forces for determining the parameter set. The case for the longitudinal forces is analogous. We have focused on the lateral vehicle dynamics, hence the parameters associated with the lateral forces, because usually these are the most critical and challenging for vehicle stability and ADAS.

Algorithm 1 Proposed NMPC with Friction Adaptation

- 1: **for** $k \leftarrow 0$ **to** T **do**
 - 2: Estimate state vector $\hat{\mathbf{x}}_k$, tire stiffness mean \hat{C}_k and covariance Σ_k using the approach in [11].
 - 3: Determine parameter set θ^* using (28), (32), or (33).
 - 4: Solve NMPC problem (13) using parameter set θ^* and state estimate $\hat{\mathbf{x}}_k$.
 - 5: **end for**
-

6. Friction-Adaptive NMPC Closed-Loop Simulation Results

Next, we validate the proposed method of friction-adaptive NMPC for tracking control in simulation, in a real-time computing environment, and in preliminary vehicle experiments. For our simulation studies, we consider a sequence of double lane-change maneuvers similar to the standardized ISO 3888 – 2 double lane-change maneuver, commonly used in vehicle stability assessment (also known as Moose Test). Unlike some previous work [26], the computational timing results are obtained in a dSPACE MicroAutoBox-II equipped IBM PPC 900 MHz processor and with 16 MB main memory, which closely resemble the capabilities of current and near future embedded microcontrollers for automotive applications.

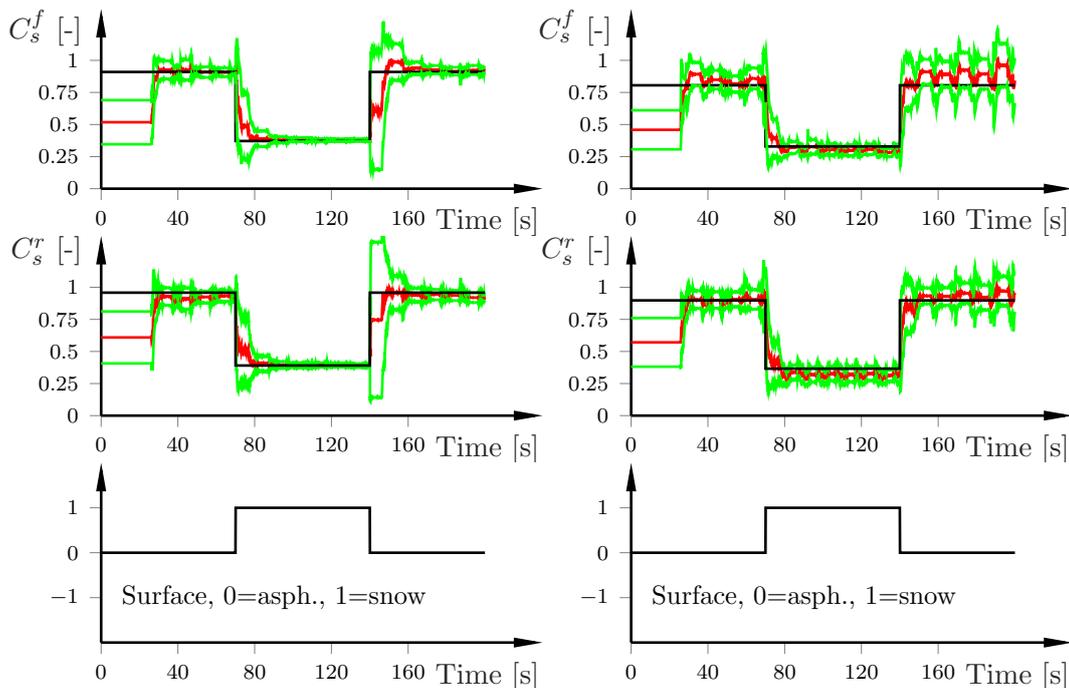


Figure 3. Stiffness estimates (red), standard deviation (green), and true stiffness (black) for multiple double lane-change maneuvers (upper two rows, front and rear, respectively) and true surface changes (lowest row). The estimates are normalized to the largest estimated value due to confidentiality. The surface detection time is less than 0.5 s. Left: steering inputs are small enough for the linear-slope assumption to hold. Right: The linear-slope assumption is violated at times (see Fig. 4), which results in slightly worse performance.

The vehicle parameters are from a mid-size SUV, and the tire parameters for the different surfaces are taken from [15]. The NMPC prediction model is the nonlinear single-track model (1) with the Pacejka tire model (4) and the FE (5) modeling the

combined slip, and the stiffness estimator uses a linear single-track model with the linear force approximation (19). The controller is simulated in closed-loop with a nonlinear double-track vehicle dynamics model [13,27] that accounts for roll and pitch dynamics, including load transfer across the four wheels. To emulate more realistic conditions, we add measurement noise to the sensors and a bias to the steering actuator angle, which is estimated by an extended Kalman filter (see [23]). For simplicity, the longitudinal velocity reference is set to 10 m/s in all simulations and only the adaptation of the lateral tire force function is considered. Due to the nature of the simulations, only the front wheel angle δ_f^{cmd} and the wheel engine torque $T_f^{e,\text{cmd}}$ are used as control inputs, while the brake torques on front and rear wheels, $T_i^b, i \in \{f, r\}$, are not used in these simulation results.

The tire-stiffness estimator uses $N = 100$ particles and the inertial sensor measurement noise values are taken from those of a low-cost inertial measurement unit typical for automotive applications. The initial estimates and the different tuning parameters in the estimator are fairly generic, and the same as in [11].

6.1. Simulation of Multiple Road Surface Changes

Fig. 3 shows the tire-stiffness estimates for a scenario of multiple double lane-change maneuvers at small steering amplitudes such that the slip angles are in the linear region (left column) and excite the nonlinear region (right column), respectively. At first, the vehicle drives on asphalt. At $t = 70$ s, the surface abruptly changes to snow, which is followed by a surface change back to asphalt at $t = 140$ s. When driving in the linear region of the tire-force curve (left figure), the stiffness estimator finds the correct stiffness values with high certainty, indicated by the decreasing standard deviations in green.

The stiffness estimates for more aggressive steering maneuvers such that the tire forces reach the nonlinear region, as shown in the right column of Fig. 3, the surface changes are accurately detected, even though the actual stiffness estimates are slightly biased because the forces enter the nonlinear region. The bias can be suppressed, or at least mitigated, by setting the deactivation threshold for the estimator tighter. In any case, the *detection* of the road surface conditions appears to be relatively insensitive to small errors in the stiffness estimates. The corresponding force-slip diagrams showing the resulting normalized tire forces are in Fig. 4.

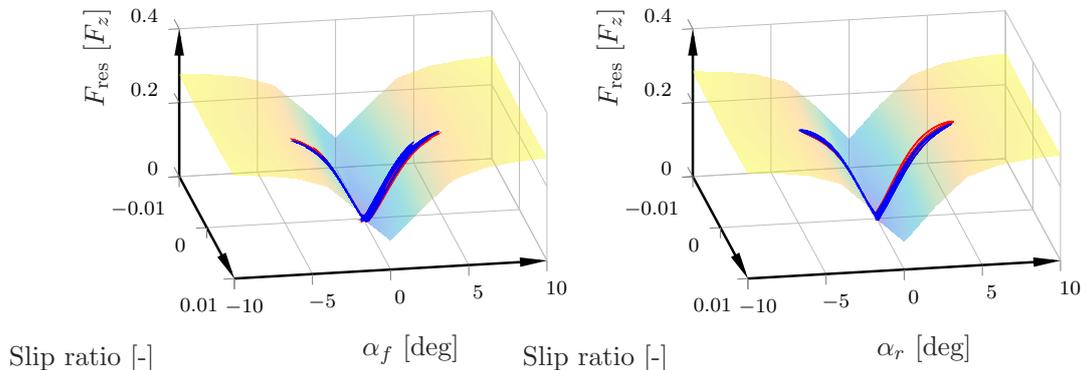


Figure 4. Resulting normalized tire forces for the friction-adaptive NMPC closed-loop simulation results with multiple double lane-change maneuvers corresponding to the right column in Fig. 3.

The closed-loop simulation results are shown in Fig. 5, which demonstrate that the friction-adaptive NMPC scheme handles the multiple double lane-change maneuvers with relative ease, and that the trajectory is tracked well.

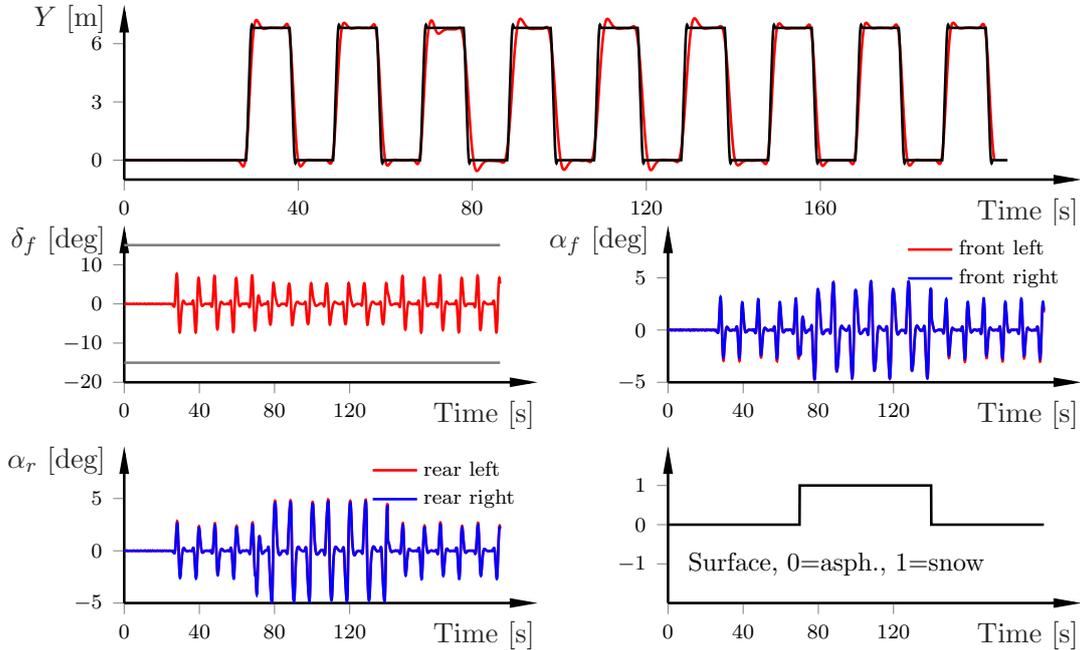


Figure 5. Friction-adaptive NMPC closed-loop simulation results at 10 m/s for multiple double lane-change maneuvers with surface changes, corresponding to the stiffness estimates in the right column in Fig. 3. The upper-most plot shows the lateral position. The steering constraints for the front wheel angle are shown as gray horizontal lines. The lowest-right plot shows the road surface through the simulation time.

6.2. Comparison with Non-adaptive NMPC

To illustrate the importance of knowing the road conditions, Fig. 6 displays the resulting normalized tire forces in the simulations with non-adaptive NMPC. Both the front and rear tire forces enter significantly inside the tire saturation region, which is sharply different from the friction-adaptive NMPC results in Fig. 4.

6.3. Robustness to Tire-Parameter Uncertainty

From Fig. 6, it is obvious that knowing the road conditions is crucial for the NMPC to work reliably. To demonstrate the robustness of the friction-adaptive NMPC to the predetermined parameter sets, we execute four closed-loop simulations where in each we perturb the underlying parameters in the Pacejka tire model (4). The involved parameters are perturbed by drawing from a uniform distribution with variation of $\pm 30\%$ around the correct parameter values. Fig. 7 shows the results for the lateral position and rear left slip angle.

Comparing with the results in Fig. 5 where the tire parameters are known, the tracking performance is slightly worse for some of the simulations. However, the vehicle maintains stability in all of the simulations, which shows that the key to achieve vehicle stability is not the exact knowledge of the tire parameters, but rather to have reasonable estimates of them. In Fig. 7, the parameters corresponding to the blue

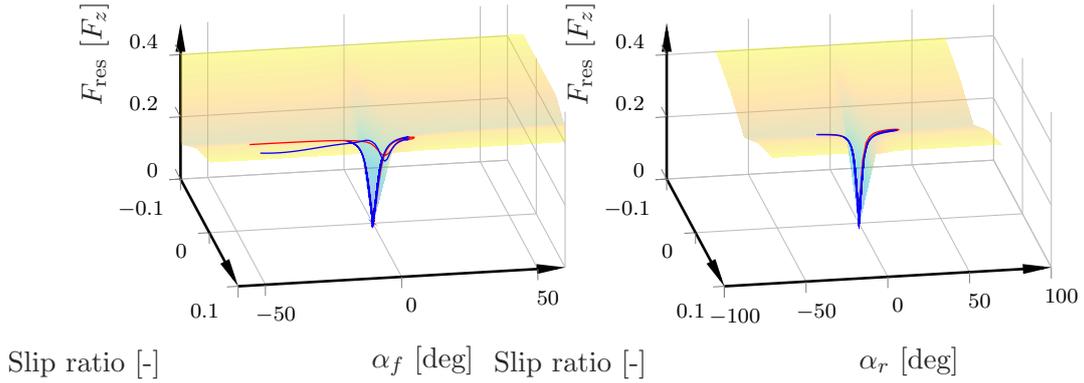


Figure 6. Resulting normalized tire forces for the non-adaptive NMPC that assumes asphalt parameter values in closed-loop simulation results with multiple double lane-change maneuvers.

curve are more than 20% from the true values on snow in the friction coefficient μ^y and B^y , for both the front and rear wheels, which is an overestimate of the errors one would expect from an informed estimation procedure [8,9].

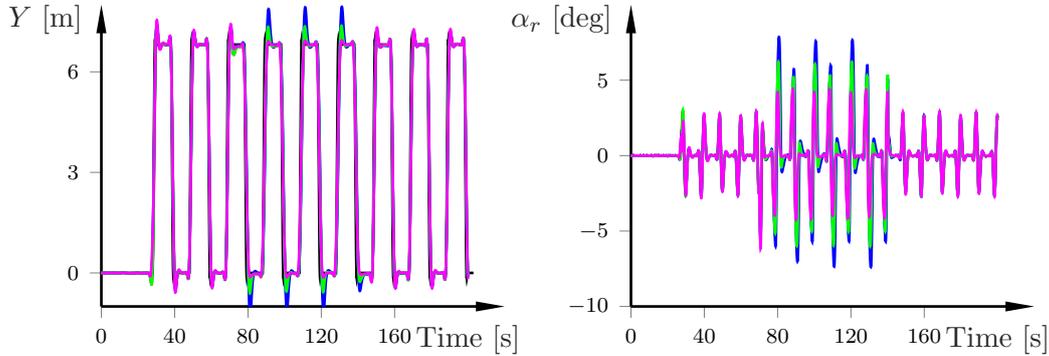


Figure 7. Friction-adaptive NMPC closed-loop simulation results at 10 m/s for multiple double lane-change maneuvers with surface changes, where we have perturbed the tire parameters in (4) used in the NMPC. The four different tire parameter sets are drawn from a uniform distribution around the correct values, with a variation of $\pm 30\%$.

6.4. Real-Time Feasibility for Embedded Implementation

Finally, we assess the real-time computational feasibility of the proposed method. In Table 1, we show the closed-loop computation times of the friction-adaptive NMPC on a dSPACE MicroAutoBox-II rapid prototyping unit. The table includes the worst-case computation time for different values of the control horizon length N_c and the number of particles N_{PF} in the estimator. From these results, the computation time for both the NMPC and the tire-stiffness estimator scales linearly and the total time remains well below the desired sampling time of 50 ms imposed by the considered vehicle interface. Each QP has been solved until convergence with a tolerance of 10^{-8} in the PRESAS solver [12].

6.5. Preliminary Results of Experimental Validation

While an exhaustive experimental assessment of the friction-adaptive NMPC controller, including surface transitions, requires special proving grounds that were not

	$N_c = 20, N_{PF} = 500$ $N_c = 30, N_{PF} = 750$		
NMPC – QP preparation	7.6 ms	11.3 ms	
NMPC – PRESAS solver	6.2 ms	9.3 ms	
Particle-filter estimator	4.7 ms	7.1 ms	
Total turnaround time	21.1 ms	30.8 ms	< 50 ms

Table 1. Worst-case timing results (ms) for closed-loop friction-adaptive NMPC simulations using different values for the control horizon N_c and number of particles N_{PF} on dSPACE MicroAutoBox-II.

yet available to us, we have been able to obtain some preliminary experimental results on a snow-covered test track in Hokkaido, Japan. We consider tracking with a target speed of 40km/h the centerlane of an approximately rectangular test track (450m \times 50m), which includes a sequence of corners emulating the ISO 3888 – 2 double lane change test. The vehicle used for testing is the same from which the simulation model for the closed-loop simulations of Sections 6.1–6.3 was derived, and the computing platform is the same rapid prototyping unit used in Section 6.4.

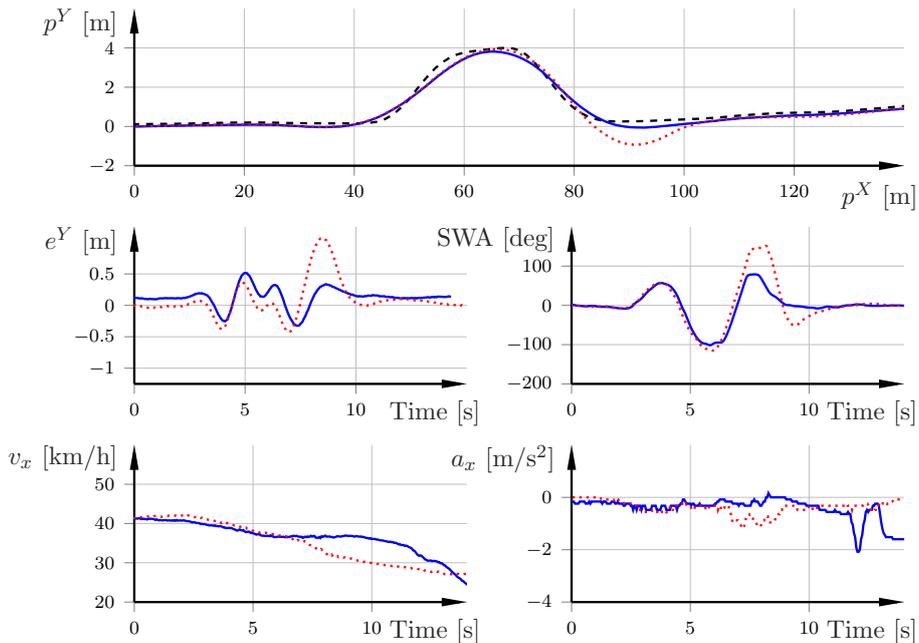


Figure 8. Experimental results for a double lane change on snow. First row: Trajectories for the center lane reference (black, dash), the NMPC controller with asphalt tire model (red, dot), and for the friction-adaptive NMPC controller that has learned the snow surface (blue, solid). Second row: Time histories of the tracking error (left) and the steering wheel angle (SWA, right), for the NMPC with asphalt tire model (red, dot), and the friction-adaptive NMPC controller that has learned the surface (blue, solid). Third row: Time histories of the vehicle velocity (left) and longitudinal acceleration (right), for the NMPC controller with asphalt tire model (red, dot), and the friction-adaptive NMPC controller that has learned the surface (blue, solid).

The experimental results of the test are shown in Fig. 8, focusing on the corners emulating a double lane change turn. As previously, we have compared the results of a

non-adaptive NMPC controller with an asphalt-type tire model, against the friction-adaptive NMPC control results. Due to running on the snow-covered test track, the tire model is learned rapidly. When the vehicle arrives at the corners emulating a double lane change turn, the model for snow is already in use.

The trajectories are shown in the first row in Fig. 8 for the standard NMPC based on an asphalt-type tire model, for the friction-adaptive NMPC, and the center lane reference trajectory that is generated from the track map for the target velocity of 40km/h. While the NMPC is generally robust even when the tire model is incorrect, due to its feedback action, it is evident that when the prediction is based on a tire model for asphalt, the controller tries to command large steering inputs that would be very effective on asphalt to provide high performance tracking. On the other hand, such steering inputs are excessively large for the snow surface, as they cause the tires to enter deeply into the saturation region. As a consequence, the tracking error (first column, second row in Fig. 8) grows large and eventually the controller needs to sensibly reduce the speed to recover the tracking performance (first column, third row in Fig. 8). Instead, when the tire model for snow has been learned, the controller is aware of the limitations imposed by the surface and hence produces more limited steering actions that avoid excessive tire saturation and as a consequence, this reduces the peak lateral tracking error $\max_t |e^Y(t)| = \max_t |p_{\text{ref}}^Y(t) - p^Y(t)|$ by more than 50%.

7. Conclusions and Future Outlook

This paper presented an NMPC for vehicle trajectory tracking that adapts to the road surface. To avoid requiring data from the unstable region of the vehicle dynamics to be collected when the prediction model may still be incorrect model, we combine data-based adaptation with pre-computed models. Estimation of the initial slope of the tire-force curve (i.e., the linear region) is used to select the full tire force curve parameters from a library of models, then used in NMPC.

While the method assumes a set of tire parameters for the surfaces of interest, it is not necessary to achieve the exact tire parameters for the particular vehicle setup currently employed. Rather, the tire model should captures the important characteristics, such as the peak friction coefficient, see, e.g., Fig. 7.

The simulation results demonstrated the validity of the approach, and also the potential vehicle instability if the tire parameters are not adapted to the changing road conditions. We discussed timing results using a dSPACE MicroAutoBox-II, which indicate that the presented method may be suitable for implementation on automotive embedded platforms. Finally, we presented preliminary results of an in-vehicle experimental validation on a snow-covered test track.

References

- [1] Rawlings J, Mayne D. Model predictive control: Theory and design. Madison, WI: Nob Hill Publishing, LLC; 2002.
- [2] Hrovat D, Di Cairano S, Tseng H, et al. The development of model predictive control in automotive industry: A survey. In: IEEE Conf. Control Applications; 2012. p. 295–302.
- [3] Di Cairano S, Kolmanovsky IV. Real-time optimization and model predictive control for aerospace and automotive applications. In: Amer. Control Conf.; 2018. p. 2392–2409.
- [4] Falcone P, Borrelli F, Asgari J, et al. Predictive active steering control for autonomous vehicle systems. IEEE Trans Control Sys Technology. 2007;15(3):566–580.

- [5] Di Cairano S, Tseng H, Bernardini D, et al. Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. *IEEE Trans Control Sys Technology*. 2013;21(4):1236–1248.
- [6] Di Cairano S, Kalabić U, Berntorp K. Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds. In: *IEEE Conf. Decision and Control*; 2016.
- [7] Quirynen R, Berntorp K, Di Cairano S. Embedded optimization algorithms for steering in autonomous vehicles based on nonlinear model predictive control. In: *Amer. Control Conf.*; 2018.
- [8] Gustafsson F. Slip-based tire-road friction estimation. *Automatica*. 1997;33:1087–1099.
- [9] Braghin F, Cheli F, Sabbioni E. Environmental effects on Pacejka’s scaling factors. *Veh Syst Dyn*. 2006 Jul;44(7):547–568.
- [10] Lundquist C, Schön TB. Recursive identification of cornering stiffness parameters for an enhanced single track model. In: *15th IFAC Symp. System Identification*; 2009.
- [11] Berntorp K, Di Cairano S. Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering. *IEEE Trans Control Syst Technol*. 2018;In press.
- [12] Quirynen R, Knyazev A, Di Cairano S. Block structured preconditioning within active-set based real-time optimal control. In: *Eur. Control Conf.*; Jun.; Limassol, Cyprus; 2018.
- [13] Berntorp K, Olofsson B, Lundahl K, et al. Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres. *Veh Syst Dyn*. 2014;52(10):1304–1332.
- [14] Pacejka HB. *Tire and vehicle dynamics*. 2nd ed. Oxford, United Kingdom: Butterworth-Heinemann; 2006.
- [15] Berntorp K. *Particle filtering and optimal control for vehicles and robots [dissertation]*. Department of Automatic Control, Lund University, Sweden; 2014.
- [16] Di Cairano S, Doering J, Kolmanovsky I, et al. Model predictive control of engine speed during vehicle deceleration. *IEEE Trans Control Sys Technology*. 2014;22(6):2205–2217.
- [17] Faulwasser T, Findeisen R. Nonlinear model predictive control for constrained output path following. *IEEE Trans Automatic Control*. 2015;61(4):1026–1039.
- [18] Fletcher R. *Practical methods of optimization*. John Wiley & Sons; 2013.
- [19] Grüne L. *NMPC Without Terminal Constraints*. In: *IFAC Conf. Nonlinear MPC*; 2012.
- [20] Bock HG, Plitt KJ. A multiple shooting algorithm for direct solution of optimal control problems. In: *IFAC World Congress*; 1984.
- [21] Quirynen R, Vukov M, Zanon M, et al. Autogenerating microsecond solvers for nonlinear MPC: a tutorial using ACADO integrators. *Optimal Control Applications and Methods*. 2014;36:685–704.
- [22] Diehl M, Findeisen R, Allgöwer F, et al. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Control Theory Appl*. 2005;152(3):296–308.
- [23] Berntorp K, Hoang T, Quirynen R, et al. Autonomous vehicle control design, implementation, and verification using scaled road vehicles. In: *IEEE Conf. Control Tech. and Applications*; 2018.
- [24] Quirynen R, Knyazev A, Di Cairano S. Projected preconditioning within a block-sparse active-set method for MPC. In: *IFAC Conf. Nonlinear MPC*; 2018.
- [25] Gustafsson F. *Statistical sensor fusion*. Lund, Sweden: Studentlitteratur; 2010.
- [26] Frasch JV, Gray A, Zanon M, et al. An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. In: *Proc Eur. Control Conf.*; 2013.
- [27] Berntorp K. *Derivation of a six degrees-of-freedom ground-vehicle model for automotive applications*. Dept. Automatic Control, Lund University; 2013. 7627.