Hierarchical Predictive Control for Ground-Vehicle Maneuvering

Berntorp, Karl; Magnusson, Fredrik

Published in:
Proceedings of the 2015 American Control Conference

2015

Link to publication

Citation for published version (APA):
Berntorp, K., & Magnusson, F. (2015). Hierarchical Predictive Control for Ground-Vehicle Maneuvering. In *Proceedings of the 2015 American Control Conference* (pp. 2771-2776)

Total number of authors:
2

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Download date: 05. Dec. 2025

Hierarchical Predictive Control for Ground-Vehicle Maneuvering

Karl Berntorp[1] and Fredrik Magnusson[2]

Abstract— **This paper presents a hierarchical approach to feedback-based trajectory generation for improved vehicle autonomy. Hierarchical vehicle-control structures have been used before—for example, in electronic stability control systems, where a low-level control loop tracks high-level references. Here, the control structure includes a nonlinear vehicle model already at the high level to generate optimization-based references. A nonlinear model-predictive control (MPC) formulation, combined with a linearized MPC acting as a backup controller, tracks these references by allocating torque and steer commands. With this structure the two control layers have a physical coupling, which makes it easier for the low-level loop to track the references. Simulation results show improved performance over an approach based on linearized MPC, as well as feasibility for online implementations.**

I. INTRODUCTION

Currently, one of the main trends in the automotive research is improving situation awareness, where, for example, (semi) autonomous lane-keeping systems are natural extensions to the Electronic Stability Program (ESP) [1]. The enabler for improved situation awareness is the increased sensing and computing capabilities in modern vehicles [2].

In this paper we propose a combined high- and low-level optimal-control approach to lane keeping and trajectory generation for road vehicles, which already at the high level accounts for nonlinear chassis and tire dynamics. The work presented here builds on the conclusions from [3]–[5], which showed that single-track models can replicate expert-driving behavior for high-level variables, such as yaw rate and velocity. It was also shown that the vehicle states are qualitatively similar for single- and double-track models when using an appropriate tire model, but the input torques and steer angles are significantly different. This conclusion is utilized here, where a high-level trajectory-generation problem is cast as a dynamic optimization problem over a road-curvature dependent horizon. The dynamics is already at the high level based on a nonlinear vehicle model, unlike most previous work on vehicle autonomy for lane keeping. We use the single-track model combined with the experimentally verified weighting-functions tire model, which incorporates combined-slip behavior. The low-level control-input allocator is formulated as a nonlinear model-predictive control (NMPC) problem [6] over a part of the high-level references. Nonlinear optimization problems sometimes fail to converge, or the convergence is slow. By combining the approach with linearized MPC (LMPC), which results in a convex optimization problem, a control solution is provided also when the NMPC fails to converge in a timely manner. A simulation example shows that this combined approach yields improved reference tracking compared with LMPC, despite higher sampling rates in the LMPC.

Application of optimal control to automotive systems is a popular research topic. In [7], a hierarchical approach for automated highway driving was introduced, where the high-level control uses a point-mass representation of the vehicle. This might work well for steady-state conditions. However, the reference trajectories that are generated may not be feasible in more aggressive maneuvering, because there is little physical coupling between the high- and low-level controllers. An approach based on robust invariant sets is explored in [8]. Other work on MPC in vehicle-dynamics control is [9]. MPC was also used in [10] for predictive prevention of roadway departure, with operation restricted to the linear region of the dynamics. Mitigation of collision impact has been explored in a series of papers, see [11] and references therein. Optimization of emergency maneuvers has also been treated—some examples are [12]–[14].

We assume that the vehicle's position, velocity, and parameters are known, and road-preview information is assumed available. See [2], [15]–[17] for some examples of how to achieve the desired information. Regarding actuation, the assumption is that individual wheel torques and steer angle can be controlled. If this is not the case, it is straightforward to reformulate the optimization problem [4], [18].

II. VEHICLE MODELING

The high-level trajectory generator uses a nonlinear single-track model, see Fig. 1, where the two wheels on each axle are lumped together. The model has three degrees of freedom, two translational and one rotational:

$$\begin{aligned} \dot{v}^X - v^Y\dot{\psi} &= \frac{1}{m}(F_f^x\cos(\delta) + F_r^x - F_f^y\sin(\delta)), \\ \dot{v}^Y + v^X\dot{\psi} &= \frac{1}{m}(F_f^y\cos(\delta) + F_r^y + F_f^x\sin(\delta)), \\ I_{ZZ}\ddot{\psi} &= l_fF_f^y\cos(\delta) - l_rF_r^y + l_fF_f^x\sin(\delta), \end{aligned} \tag{1}$$

where m is the vehicle mass, I_{ZZ} is the vehicle inertia about the Z-axis, $\dot{\psi}$ is the yaw rate, δ is the steer angle, $[v^X, v^Y]$ are the longitudinal and lateral velocities at the center of

---

[1] Karl Berntorp (corresponding author) was with the Department of Automatic Control, Lund University, Lund, Sweden at the time of this research. He is now with Mitsubishi Electric Research Laboratories, 02139 Cambridge, MA, USA. Email: `karl.o.berntorp@ieee.org`. This work was not supported by Mitsubishi Electric or any of its subsidiaries.

[2] Fredrik Magnusson is with the Department of Automatic Control, Lund University, Lund, Sweden. Email: `fredrik.magnusson@control.lth.se`

This work was supported by the Swedish Foundation for Strategic Research through the project ENGROSS, the LCCC Linnaeus Center, and the ELLIIT Excellence Center at Lund University.

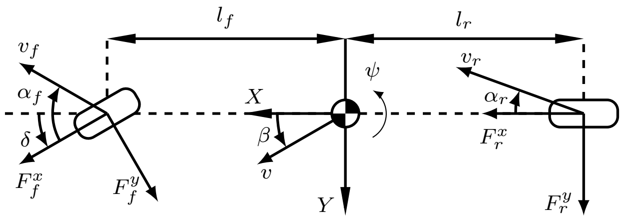

Fig. 1. The single-track model used in the high-level optimization problem.

gravity, $[l_f, l_r]$ are the distances from the mass center to the front and rear wheel base, and $[F^x, F^y]$ are the longitudinal and lateral tire forces acting at the front and rear wheels.

Fig. 2 provides a schematic of the double-track model that is used in the low-level formulation. It has five degrees of freedom: two translational (v^X and v^Y) and three rotational (the roll-pitch-yaw angles (ϕ, θ, ψ)). The suspension model is a rotational spring-damper system, and longitudinal and lateral load transfer is included. The derivation and details of both models are found in [5], [19].

The nominal tire forces F_0^x and F_0^y for the longitudinal and lateral directions under pure slip conditions are computed with the Magic formula [20], given by

$$\begin{aligned} F_0^x &= \mu_x F^z \sin\Big(C_x \arctan\big(B_x \lambda_i \\ &\quad - E_x(B_x\lambda - \arctan B_x\lambda)\big)\Big), \\ F_0^y &= \mu_y F^z \sin\Big(C_y \arctan\big(B_y \alpha \\ &\quad - E_y(B_y\alpha - \arctan B_y\alpha)\big)\Big), \end{aligned} \tag{2}$$

with lateral slip α_i and longitudinal slip λ_i defined as

$$\dot{\alpha}_i \frac{\sigma}{v_i^x} + \alpha_i := -\arctan\left(\frac{v_i^y}{v_i^x}\right), \tag{3a}$$

$$\lambda_i := \frac{R_w \omega_i - v_i^x}{v_i^x}, \tag{3b}$$

where σ is the relaxation length, R_w is the wheel radius, ω_i is the wheel angular velocity for wheel $i \in \{f, r\}$ or $\{1, 2, 3, 4\}$, and $[v_i^y, v_i^x]$ are the lateral and longitudinal wheel velocities for wheel i. In the following we suppress the index i for brevity. In (2), μ_x and μ_y are friction coefficients and B, C, and E are parameters. The nominal normal force acting on each wheel axle is given by

$$F_{0,f}^z = mg\frac{l_r}{l}, \quad F_{0,r}^z = mg\frac{l_f}{l},$$

where g is the gravitational acceleration and $l = l_f + l_r$. In the single-track model $F^z = F_0^z$ in (2). This is not true for the double-track model, because of load transfer. An experimentally verified approach to tire modeling under combined slip constraints is to scale the nominal forces (2) with a weighting function G for each direction, which depends on α and λ [20]. The relations are

$$\begin{aligned} F^{x,y} &= F_0^{x,y} G_m, \\ G_m &= \cos(C_m \arctan(H_m m)), \\ H_m &= B_{m1} \cos(\arctan(B_{m2} m)), \end{aligned} \tag{4}$$

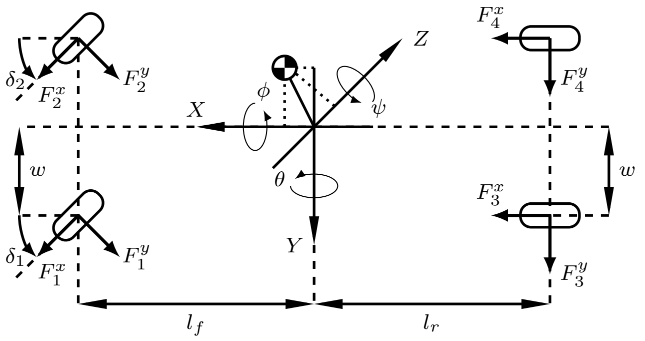

Fig. 2. The double-track model used in the low-level MPC formulations.

where m is either α or λ. Moreover, since it is the torques that can be controlled in a physical setup, we introduce a model for the wheel dynamics, namely

$$\tau = I_w \dot{\omega} - R_w F^x,$$

where I_w is the wheel inertia and τ is the input torque. To account for that commanded steer angle and brake/drive torques are not achieved instantenously, we incorporate first-order models from reference to achieved value according to

$$T\dot{\delta} = -\delta + \delta_{\text{ref}}, \tag{5}$$

and similarly for the torques, where T in (5) is the time constant of the control loop. The parameter values used here correspond to a medium-sized passenger car on dry asphalt.

III. Proposed Control Structure

Fig. 3 displays the control structure. It consists of a high-level optimizer that uses information about the road geometry and surrounding vehicles as inputs, in addition to estimates of the position p, velocity v, yaw angle ψ, and yaw rate $\dot{\psi}$. Based on this information, it computes reference trajectories for the position, velocity, yaw angle, and yaw rate. These references are then fed to an NMPC, which computes desired wheel torques τ and steer angle δ. If the NMPC fails to converge, or if the convergence is deemed too slow, the references are instead sent to an LMPC. The LMPC uses a linearization of the double-track model, and computes desired wheel torques and steer angle.

A. High-Level Trajectory Generation

The goal of the high-level optimizer is to find a path and corresponding state trajectories that minimize a suitable cost $\mathcal{J}$ while staying in lane. In a lane-keeping scenario, it is natural to include the deviation e from the middle of the lane in the cost. A common measure of vehicle stability is the vehicle sideslip angle β, defined as

$$\beta := \arctan\left(\frac{v^Y}{v^X}\right).$$

A large β indicates poor maneuverability for the average driver. It is traditionally used as a performance measure in electronic stability control systems [1]. In theory, an

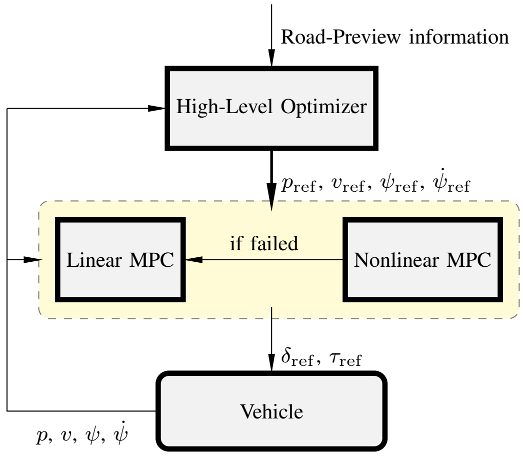
Road-Preview information
High-Level Optimizer
p_{ref}, v_{ref}, ψ_{ref}, $\dot{\psi}_{\text{ref}}$
Linear MPC
if failed
Nonlinear MPC
δ_{ref}, τ_{ref}
Vehicle
p, v, ψ, $\dot{\psi}$

Fig. 3. The proposed control structure. The high-level optimizer finds references over the considered horizon. The inputs are measurements and/or estimates of the relevant vehicle states. Based on the high-level trajectories, the NMPC aims to find wheel torques and steer angle. If the NMPC fails to converge in time, the references are instead sent to an LMPC. The LMPC uses a linearized version of the double-track model.

optimization-based safety system does not suffer from a large β, because it by definition finds the optimal solutions and can thus operate in the unstable regions of the tire-ground interaction. In practice, however, model errors will lead to deviations from the computed trajectories. Therefore, it is still desired to keep the vehicle in the small-slip region, if possible. Furthermore, a large β is a measure of driver and passenger discomfort. Thus, the cost depends on the mid-lane deviation e and the body-slip angle β according to

$$\mathcal{J} = \int_{t_0}^{t_f} (\kappa_1 e^2 + \kappa_2 \beta^2)\,\mathrm{d}t,$$

where $[\kappa_1, \kappa_2]$ are positive, scalar weights, t_0 is the predetermined start time, and t_f is the free final time.

The prediction horizon (look-ahead) is dependent on the road geometry and sampling periods. It must be chosen such that the reference trajectories span over the control horizon of the MPC, before a new high-level optimization is performed. However, the horizon cannot be made arbitrarily large. A larger horizon implies longer optimization times, and if the road curvature is steep, the available look-ahead information prevents a large prediction horizon. In practice the horizon is determined by a terminal constraint p_{t_f} on the mass center's position, given by a higher-level planner.

Constraints on input torques τ and steer angle δ are also introduced. The single-track dynamics (1) in combination with the tire dynamics (3)–(4) can be written as an index-one system of differental-algebraic equations (DAEs):

$$F(x, \dot{x}, w, \delta_{\text{ref}}, \tau_{f,\text{ref}}, \tau_{r,\text{ref}}) = 0, \tag{6}$$

where x contains the differential (state) variables, w contains the algebraic variables, and $\tau_{f,\text{ref}},\ \tau_{r,\text{ref}}$ are the desired wheel drive/brake torques on the front and rear axle, respectively. The dynamic optimization problem is then formulated over the time horizon $t \in [t_0, t_f]$, with free final time, as

$$\begin{aligned}
\underset{\delta_{\text{ref}},\tau_{f,\text{ref}},\tau_{r,\text{ref}}}{\text{minimize}} \quad & \int_{t_0}^{t_f} (\kappa_1 e^2 + \kappa_2 \beta^2)\,\mathrm{d}t \\
\text{subject to} \quad & |\tau_{i,\text{ref}}| \le \tau_{i,\max},\ \forall i \in \{f, r\}, \\
& |\delta_{\text{ref}}| \le \delta_{\max}, \\
& \|p(t_f) - p_{t_f}\| \le \epsilon, \\
& \Gamma(p) \le 0, \quad x(t_0) = x_0, \\
& F(\dot{x}, x, w, \delta_{\text{ref}}, \tau_{f,\text{ref}}, \tau_{r,\text{ref}}) = 0
\end{aligned} \tag{7}$$

where x_0 is the initial state and $\Gamma(p)$ is a mathematical description of the road constraint for the vehicle's mass center and its endpoints. Note that it is possible to express collision avoidance tasks in $\Gamma(p)$. We have introduced a slack ϵ in the terminal constraint for the position, since exact tracking is typically not crucial. Moreover, it improves convergence since exact terminal constraints are harder to fulfill. To generate an initial guess for the nonconvex problem (7), we simulate the system with a constant steer angle and zero input torques. When (7) has been solved, the optimal trajectories for p, v, ψ, $\dot{\psi}$, are sent to the low-level layer for allocation to the wheel and steer actuators. The optimization problem (7) is solved repeatedly, with a sampling period of $T_{s,h}$ s, and a new optimization is started directly after sending the trajectory references to the low-level layer.

B. Low-Level Control-Input Allocation

The aim of the low-level controller is to track the references. This is done by allocating appropriate wheel-torque and steer-angle references to the vehicle's internal controllers using an MPC. To this end, introduce the notation

$$r = \begin{bmatrix} p^{\mathrm{T}} & v^{\mathrm{T}} & \psi & \dot{\psi} \end{bmatrix}^{\mathrm{T}} \tag{8}$$

for the references to the MPC. Let

$$u = \begin{bmatrix} \delta_{\text{ref}} & \tau_{1,\text{ref}} & \tau_{2,\text{ref}} & \tau_{3,\text{ref}} & \tau_{4,\text{ref}} \end{bmatrix}^{\mathrm{T}} \tag{9}$$

denote the control input to the vehicle. Note that there are four torque references in (9), since we use a double-track model for torque allocation, but only one steer-angle reference. The individual steer angles (Fig. 2) are determined through the Ackermann geometry of the steering mechanism.

The chassis dynamics for the double-track model and the tire dynamics are formulated as the DAE system

$$F(x, \dot{x}, w, u) = 0, \tag{10}$$

where x is the state for the combined double-track and tire model and w are the corresponding algebraic variables. The function and variables in (10) are not the same as in (6).

The references are tracked by introducing a quadratic cost on the deviations from (8). For better tracking, it is typically advantageous to include a specific cost term on the terminal position. In addition, we include a terminal constraint on the position. Tracking of references is not the only objective, since driver comfort also needs attention. This is accommodated by introducing a penalty on the control signals as well. The low-level NMPC problem formulation is in each time step k stated as (with a slight abuse of notation)

$$\underset{u}{\text{minimize}} \quad \int_{\bar{t}_0}^{\bar{t}_f} (\|x - r\|_Q^2 + \|u\|_R^2)\,\mathrm{d}t + \|p(\bar{t}_f) - p_{\bar{t}_f}\|_{Q_f}^2 \tag{11a}$$

$$\begin{aligned} \text{subject to} \quad & F(x, \dot{x}, w, u) = 0, && \text{(11b)} \\ & \|p(\bar{t}_f) - p_{\bar{t}_f}\| \leq \epsilon, && \text{(11c)} \\ & -u_{\max} \leq u \leq u_{\max}, && \text{(11d)} \\ & \Gamma(p) \leq 0, && \text{(11e)} \\ & x(\bar{t}_0) = \bar{x}_0, && \text{(11f)} \end{aligned}$$

where $\bar{t}_0 = t_0 + kT_{s,l}$ and $\bar{t}_f = t_0 + (k + H_l)T_{s,l} \leq t_f$ are the initial and final time, respectively, $T_{s,l}$ is the sampling period of the MPC, and H_l is the prediction horizon of the MPC. Also, $\|x\|_Q^2 = x^{\mathrm{T}}Qx$. Moreover, Q, R, and Q_f in (11a) are the weight matrices, $p_{\bar{t}_f}$ in (11c) is the position reference at time $\bar{t}_f$, $u_{\min}$ and $u_{\max}$ contain the input-reference bounds, and $\bar{x}_0$ in (11f) is the initial state vector at time $\bar{t}_0$, given by estimates and/or measurements. Note that the path constraints are also included at the low level, in (11e). In each time step, (11) is solved with the constraint that the control-input vector is piecewise constant over the sampling periods. To generate an initial guess for (11), we use the optimal control inputs from the previous time step. When (11) has been solved, the control inputs from the first sampling period are sent to the internal vehicle controllers.

The highly nonlinear dynamics (11b) will sometimes cause the convergence of (11) to be too slow, or even fail. We therefore design an additional controller, an LMPC, which is based on repeated linearizations of the dynamics. The resulting LMPC can be written as a quadratic program, for which there exist very efficient and reliable solvers. To reduce the problem size, note that (10) is a DAE system that can be reformulated as an ordinary differential equation (ODE). The algebraic variables w arise from the slip definition (3b) and the tire-force equations (2)–(4). These can be solved for, see Sec. IV, and the result is an ODE of the form $\dot{x} = f(x, u)$. By introducing

$$A_k = \left.\frac{\partial f}{\partial x}\right|_{x_k, u_k}, \quad B_k = \left.\frac{\partial f}{\partial u}\right|_{x_k, u_k}, \tag{12}$$

where x_k and u_k are the measured and/or estimated quantities at time $t_0 + kT_{s,l}$, the LMPC formulation becomes

$$\underset{u}{\text{minimize}} \quad \int_{\bar{t}_0}^{\bar{t}_f} (\|x - r\|_Q^2 + \|u\|_R^2)\,\mathrm{d}t + \|p(\bar{t}_f) - p_{\bar{t}_f}\|_{Q_f}^2 \tag{13a}$$

$$\begin{aligned} \text{subject to} \quad & \dot{x} = f_k + A_k(x - x_k) + B_k(u - u_k) && \text{(13b)} \\ & \|p(\bar{t}_f) - p_{\bar{t}_f}\| \leq \epsilon && \text{(13c)} \\ & -u_{\max} \leq u \leq u_{\max} && \text{(13d)} \\ & \bar{\Gamma}(p) \leq 0 && \text{(13e)} \\ & x(\bar{t}_0) = \bar{x}_0, && \text{(13f)} \end{aligned}$$

where $f_k = f(x_k, u_k)$. Compared with (11), (13) involves a linearized version of the dynamics, (12) and (13b), which introduces approximation errors. Hence, (13) is only executed when (11) fails to converge or when the convergence rate is slow. The path constraint $\bar{\Gamma}(p)$ in (13e) is an approximate version of (11e) (e.g., ellipses or hyperplanes), to preserve convexity. The endpoint constraint (13c) can be removed if a quadratic program is wanted, otherwise the problem becomes a second-order cone program [21]. One cycle of the complete algorithm is summarized in Algorithm 1, where conv is an indicator of whether the NMPC has converged or not.

Algorithm 1.

```
1: Given state estimates $x(t_0)$ and road-preview information, solve (7) and form
       $r = \begin{bmatrix} p^{\mathrm{T}} & v^{\mathrm{T}} & \psi & \dot{\psi} \end{bmatrix}^{\mathrm{T}}$
   for the time period $t \in [t_0, t_f]$.
2: Set $k = 0$.
3: while $k \leq \lfloor T_{s,h}/T_{s,l} \rfloor$ do
4:     Acquire state estimates $x_k$ and solve (11).
5:     if conv $\neq$ True then
6:         Compute (12) and solve (13).
7:     end if
8:     Apply the first control to the plant.
9:     Set $k = k + 1$.
10: end while
```

Algorithm 1 executes with the sampling period $T_{s,h}$ s, and the while-loop executes with the sampling period $T_{s,l}$ s. The convergence condition conv on line 5 in Algorithm 1 is based on an analysis of mean convergence time of the LMPC: Assume that the mean solution time of the LMPC is h s. Then the NMPC is terminated and conv is set to false if the execution time is larger than $T_{s,l} - h + \Delta$, where Δ is a slack that is introduced to provide robustness with respect to variations in execution time.

IV. IMPLEMENTATION

The high-level trajectory generation and the MPC formulations are implemented using the open-source software platform JModelica.org [22]. The DAE-constrained optimization problems (7) and (11) are first transformed into ODE-constrained optimization problems and then discretized using the procedures in [23], [24]. The resulting nonlinear program (NLP) is solved using IPOPT [25] and the linear solver MA27 [26]. CasADi [27] is used to obtain the relevant first- and second-order derivatives of the NLP functions.

The symbolic transformations to ODE-constrained optimization problems lead to drastically reduced number of system variables and hence improved convergence speed, as the algebraic variables are eliminated from the equation system. Moreover, it provides solution times that enable online implementations. We have also noticed that the convergence is more robust, which is important for an online implementation. A way to reduce the computation time further is to generate C code for evaluation of the NLP functions and their derivatives. We expect this to reduce the solution time with approximately 30%.

V. SIMULATION STUDY

The simulation results are from a road segment with a curvature radius of 30 m, obtained by applying the proposed control structure to the double-track model in Sec. II. The initial velocity is $v_0 = 70$ km/h. The input constraints are

$$\delta_{\max} = 0.5,\ \tau_{i,\max} = F_i^z \mu_x R_w,\ i \in \{f, r\} \text{ or } \{1, 2, 3, 4\}.$$

The steer-angle constraint is based on the achievable wheel-steer angle for a standard passenger vehicle, and the torque constraints are based on the maximum attainable longitudinal forces. In reality the torque limits depend on several factors, such as transmission ratios and vehicle speed. Moreover, it is typically possible to generate much larger brake torques than acceleration torques. These factors are neglected here, but is not a restriction for the considered scenario (Fig. 4).

For the high-level optimization problem (7), $\kappa_1 = 1$ and $\kappa_2 = 15$. The sampling period is $T_{s,h} = 0.4$ s. The look-ahead in this example is between 10–20 m, resulting in 0.5–1 m between the dicretization points for the high-level control problem. For the low-level MPCs, the prediction horizons are $H_l = 5$ samples and $T_{s,l} = 0.04$ s. The choice for when to terminate the NMPC is decided based on estimations of how long execution time the LMPC needs to converge. With the settings used here, the LMPC typically converges within 10 ms (15–20 iterations). Thus, when the NMPC has been executing more than approximately 30 ms without converging, the LMPC is turned on. In the actual implementation, however, to facilitate reproducibility the number of iterations are used as the termination criterion.

For comparison, we also show results from a setup where an LMPC is responsible for allocating control inputs. The LMPC uses $T_{s,l} = 0.01$ s, corresponding to its average computation time. The same controller parameters are used in both LMPC and NMPC, but the tuning is custom-tailored to the LMPC to give a good tradeoff between tracking and control aggressiveness. The weights Q and R are

$$\begin{aligned} Q &= \operatorname{diag}([50, 50, 30, 10, 10, 50]), \\ R &= \operatorname{diag}([1, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}]), \end{aligned}$$

where $\operatorname{diag}(\cdot)$ is the diagonal matrix. The outer path constraint is modeled as a circle for the considered segment in the MPC. The inner path constraint is neglected, because it will never be active in the considered scenario. The LMPC can therefore be posed as a second-order cone program without imposing approximations on the path (13c).

A. Results

The control signals are shown in Fig. 4. The combined NMPC/LMPC gives more aggressive steer angle than LMPC, but the torques do not differ much. An interpretation is that the LMPC overestimates the available lateral tire force, and therefore does not turn as aggressively. Fig. 5 displays the position references and actual positions. The NMPC/LMPC results in much better position tracking. Fig. 6 contains some of the states that are often connected to vehicle stability, and these are followed closely for most of the maneuver. There are three convergence failures in the NMPC out of the 77 optimizations for this particular scenario, see Fig. 7.

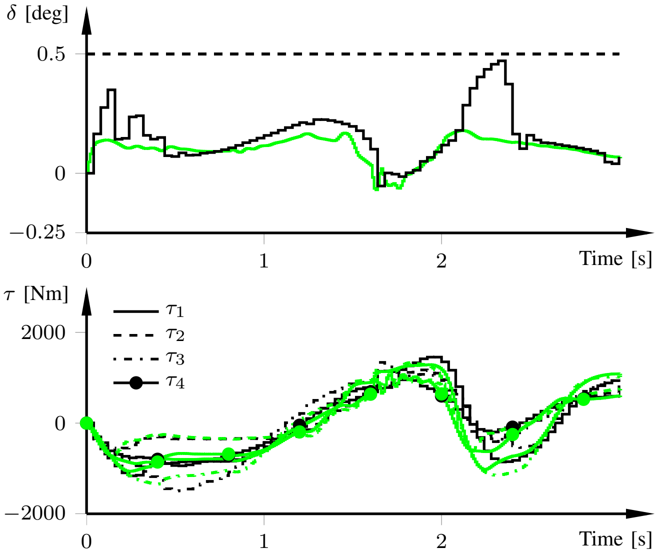

Fig. 4. Control inputs as computed from the low-level control architecture (black) and when using LMPC only (green). Using linearization seems to overestimate the available tire forces, especially the lateral forces (the steer angle is smaller for the LMPC). The same tuning has been used for both setups, and can be considered conservative.

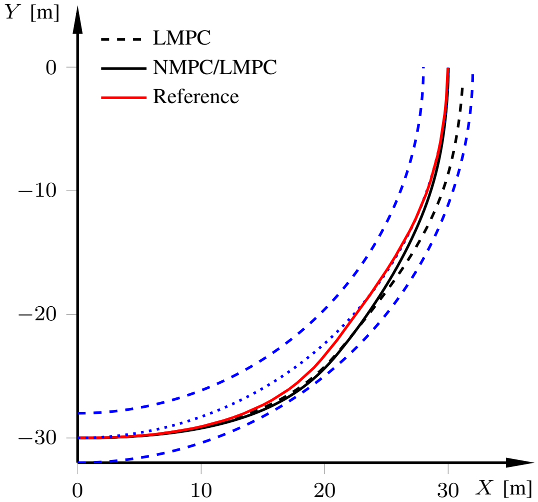

Fig. 5. Position references (red) from the high-level optimizer and actual positions (black). The results for LMPC only are black dashed. Clearly, using NMPC improves tracking performance. The mid-line segment is shown in blue dotted, and the road constraints are shown in blue dashed.

VI. CONCLUSION

This paper presented a hierarchical approach to optimal motion planning for vehicles. It uses a nonlinear vehicle model with tire modeling in the optimization problem at the high level, which provides for better coupling with the low-layer control-input allocator, especially for aggressive maneuvering. We designed a low-level control structure that uses an NMPC for allocating the torques to the wheels, incorporating a nonlinear double-track model with suspension dynamics as well as rotations in space. We combined this

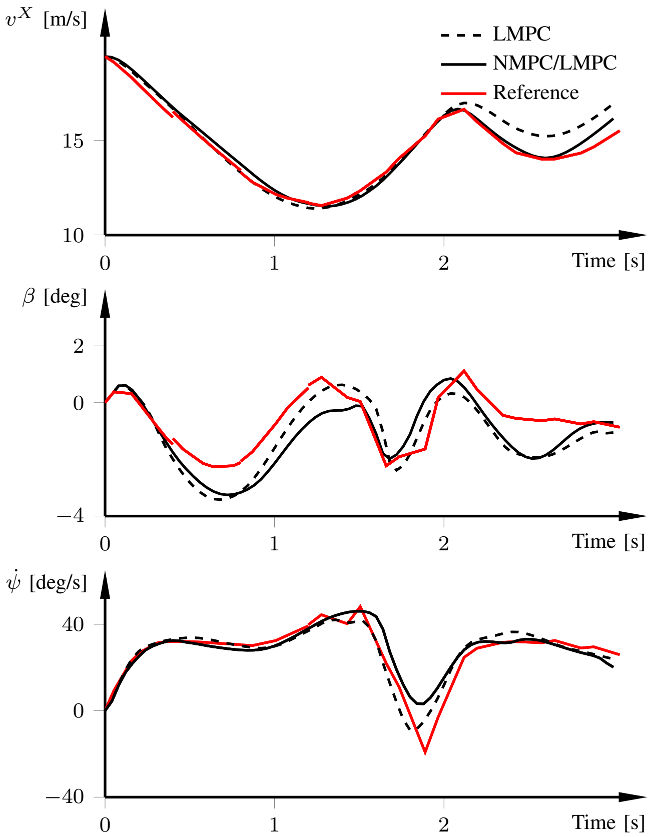

Fig. 6. State-trajectory references (red) and actual values from the different controllers.

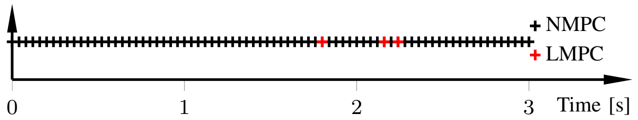

Fig. 7. The figure shows when the NMPC and LMPC are active. The NMPC converges before the computation-time limit in 74 out of the 77 optimizations.

with LMPC, to be used in those cases when the NMPC fails to find a solution within a prescribed time limit. Results showed that viable computation times are achieved, even when using a general framework for implementation. We compared with only using an LMPC for allocating controls. Despite that the tuning was custom-tailored to the LMPC and the higher sampling rate in the LMPC, the NMPC gave a clear performance increase.

REFERENCES

[1] R. Bosch, Ed., *Bosch Automotive Handbook*, 8th ed., ser. Bosch Invented for life. Wiley, 2011.

[2] C. Lundquist, "Sensor fusion for automotive applications," Ph.D. dissertation, Linköping University, Automatic Control, The Institute of Technology, 2011.

[3] K. Berntorp, B. Olofsson, K. Lundahl, B. Bernhardsson, and L. Nielsen, "Models and methodology for optimal vehicle maneuvers applied to a hairpin turn," in *Am. Control Conf.*, Washington, DC, Jun. 2013.

[4] K. Berntorp, B. Olofsson, K. Lundahl, and L. Nielsen, "Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres," *Vehicle Syst. Dyn.*, vol. 52, no. 10, pp. 1304–1332, 2014.

[5] K. Berntorp, "Particle filtering and optimal control for vehicles and robots," Ph.D. dissertation, Department of Automatic Control, Lund University, Sweden, May 2014.

[6] L. Del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. E. Kolmanovsky, *Automotive Model Predictive Control: Models, Methods and Applications*, ser. Lecture notes in Control and Information Sciences. Springer Verlag, 2010.

[7] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *ASME Dyn. Syst. Control Conf.*, Cambridge, MA, Sep. 2010.

[8] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Vehicle Syst. Dyn.*, vol. 52, no. 6, pp. 802–823, 2014.

[9] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilisation via active front steering and braking," *Vehicle Syst. Dyn.*, vol. 46, pp. 611–628, 2008.

[10] M. Ali, P. Falcone, C. Olsson, and J. Sjöberg, "Predictive prevention of loss of vehicle control for roadway departure avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 56–68, 2013.

[11] I. Chakraborty, P. Tsiotras, and R. S. Diaz, "Time-optimal vehicle posture control to mitigate unavoidable collisions using conventional control inputs," in *Am. Control Conf.*, Washington, DC, Jun. 2013.

[12] P. Dingle and L. Guzzella, "Optimal emergency maneuvers on highways for passenger vehicles with two- and four-wheel active steering," in *Am. Control Conf.*, Baltimore, MD, Jun. 2010.

[13] S. Anderson, S. Peters, T. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *Int. J. Veh. Auton. Syst.*, vol. 8, pp. 190–216, 2010.

[14] Z. Shiller and S. Sundar, "Emergency lane-change maneuvers of autonomous vehicles," *J. Dyn. Syst.*, vol. 120, no. 1, pp. 37–44, 1998.

[15] C. R. Carlson and J. C. Gerdes, "Consistent nonlinear estimation of longitudinal tire stiffness and effective radius," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 6, pp. 1010–1020, 2005.

[16] K. Berntorp, "Particle filter for combined wheel-slip and vehicle-motion estimation," in *Am. Control Conf.*, Chicago, IL, Jul. 2015, accepted for publication.

[17] J. Svendenius, "Tire modeling and friction estimation," Ph.D. dissertation, Department of Automatic Control, Lund University, Sweden, Apr. 2007.

[18] K. Lundahl, B. Olofsson, K. Berntorp, J. Åslund, and L. Nielsen, "Towards lane-keeping electronic stability control for road-vehicles," in *19th IFAC World Congress*, Cape Town, South Africa, Aug. 2014.

[19] K. Berntorp, "Derivation of a six degrees-of-freedom ground-vehicle model for automotive applications," Department of Automatic Control, Lund University, Sweden, Technical Report ISRN LUTFD2/TFRT--7627--SE, Feb. 2013.

[20] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Oxford, United Kingdom: Butterworth-Heinemann, 2006.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*, 6th ed. New York: Cambridge University Press, 2008.

[22] J. Åkesson, K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit, "Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problem," *Computers and Chemical Engineering*, vol. 34, no. 11, pp. 1737–1749, Nov. 2010.

[23] F. Magnusson, K. Berntorp, B. Olofsson, and J. Åkesson, "Symbolic transformations of dynamic optimization problems," in *10th Int. Modelica Conf.*, Lund, Sweden, Mar. 2014.

[24] F. Magnusson and J. Åkesson, "Collocation methods for optimization in a Modelica environment," in *9th Int. Modelica Conf.*, Munich, Germany, Sep. 2012.

[25] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[26] HSL, "A collection of Fortran codes for large scale scientific computation," http://www.hsl.rl.ac.uk, 2014.

[27] J. Andersson, J. Åkesson, and M. Diehl, "CasADi – A symbolic package for automatic differentiation and optimal control," in *Recent Advances in Algorithmic Differentiation*, ser. Lecture Notes in Computational Science and Engineering. Berlin, Germany: Springer, 2012.