IEEE Transactions on Control Systems Technology, Volume 26 , Issue 6 , November 2018, 1982-1994
DOI:10.1109/TCST.2017.2753169

A Real-Time Nonlinear Model Predictive Control Strategy for Stabilisation of an Electric Vehicle at the Limits of Handling

Efstathios Siampis, Efstathios Velenis, Salvatore Gariuolo, and Stefano Longo

***Abstract*—In this paper we propose a real-time nonlinear Model Predictive Control strategy for stabilisation of a vehicle near the limit of lateral acceleration using the rear axle electric torque vectoring configuration of an electric vehicle. A nonlinear four-wheel vehicle model coupled with a nonlinear tyre model are used to design three Model Predictive Control strategies of different levels of complexity that are implementable online: one that uses a linearized version of the vehicle model and then solves the resulting Quadratic Program problem, a second one that employs the Real Time Iteration scheme on the nonlinear Model Predictive Control problem and a third one that applies the Primal Dual Interior Point method on the nonlinear Model Predictive Control problem instead until convergence. After analysing the relative trade-offs in performance and computational cost between the three Model Predictive Control strategies by comparing them against the optimal solution in a series of simulation studies, we test the most promising solution in a high fidelity environment.**

***Index Terms*—vehicle dynamics; nonlinear control systems; predictive control; accident prevention.**

I. Introduction

The effectiveness of controlling a vehicle at the limit of lateral acceleration by regulating its velocity is well documented in the vehicle dynamics literature. For example, in [1] a simple proportional feedback controller is shown to be more effective than brake actuated yaw control in following the intended vehicle path, as defined by the driver. In [2] a strategy that reduces the torque request from the driver when the lateral acceleration exceeds a specific threshold is presented, while in [3] a high level controller providing decoupled longitudinal force and yaw moment requests is combined with a static control allocation scheme to calculate forces and actuator inputs. In [4] we presented a multivariable control architecture that uses combined velocity, sideslip and yaw rate regulation to stabilize the vehicle in terminal understeer conditions using the torque vectoring capabilities of the rear axle of an electric vehicle. A nonlinear vehicle model combined with a nonlinear tyre model was employed to design a scheduled Linear Quadratic Regulator (LQR). The actuator limits and system constraints have been taken into account in [5], where we presented a linear Model Predictive Control (MPC) strategy and compared it against the LQR from [4] in two limit-handling manoeuvres. Results showed that accounting for the input and state constraints has a noticeably positive impact on the stabilisation of the vehicle under such scenarios. In this paper we explore recent developments in the area of fast Nonlinear MPC (NMPC) solutions that can be implemented online and the relative trade-offs in performance and computational time when such strategies are used in the context of controlling an electric vehicle near the limit of lateral acceleration, as already established in [4], [5].

The huge leaps in computational power and memory storage in the past 20 years along with the introduction of new optimisation algorithms and the continuous improvement of existing ones have led to extensive research on potential application of MPC in large volume domains, such as the aerospace, automotive and robotics industries [6]. In the automotive sector, replacing the currently finely-tuned controllers with optimal controllers that require a reduced number of parameters to calibrate can dramatically reduce development times. For this reason, a variety of MPC solutions for automotive applications have been already proposed in the literature, ranging from steering [7] to active and semi-active suspension control [8], engine management [9], emission regulation [10] and control of vehicle platoons [11]. Looking more specifically in the area of vehicle dynamics control systems, we can distinguish two main MPC application areas: 1) on the control of autonomous and semi-autonomous vehicles and 2) on active safety control systems. However, it is interesting to note here that the distinction between autonomous vehicle control and active safety control is now becoming less clear, mainly due to the rapid development of sensor technologies and sensor fusion algorithms.

The series of papers from Borrelli, Falcone and Keviczky [7], [12]–[14] explore the application of MPC for trajectory tracking in an autonomous vehicle application using the Active Front Steering (AFS) system with or without differential braking and traction control. In [12], [13] an NMPC strategy for controlling the vehicle in a highly transient manoeuvre is constructed using a bicycle vehicle model that neglects load transfer effects. The authors report the necessary increase in both the prediction and control horizon with higher entry speeds in a double-lane change scenario to keep the vehicle stable, and the subsequent increase in computational time. Since the NMPC strategy proposed in [12], [13] is not implementable online, a Linear Time Varying MPC (LTV-MPC) controller is presented in [7]. Simulation and experimental results show that the LTV-MPC strategy shows no infeasibility

E. Siampis, E. Velenis and S. Longo are with the Centre for Automotive Engineering, School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, UK (e-mail: e.siampis@cranfield.ac.uk; e.velenis@cranfield.ac.uk; s.longo@cranfield.ac.uk). S. Gariuolo is with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy (email:s.gariuolo@studenti.unina.it)

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

problems with higher initial velocities but poorer tracking when compared to the NMPC. Finally in [14] the authors construct two NMPC strategies using internal vehicle models of different levels of fidelity, one that employs a four-wheel vehicle model with wheel dynamics and control inputs the front steering and individual wheel brake torques and one that uses a bicycle model instead with a direct yaw moment along with AFS as control inputs. While simulation tests on a double-lane change show promising results, the main problem for both controllers remains the high computational cost which makes it impossible to implement them online. A real-time NMPC strategy that employs the Real Time Iteration (RTI) scheme originally proposed in [15] on an autonomous vehicle application is presented in [16]. The authors use a four-wheel vehicle model and a nonlinear tyre model to derive the track-dependent (spatial) dynamics for the NMPC strategy. Results show that the proposed solution is implementable online and that is can successfully navigate around two consecutive obstacles in a simulation test. In [17] a collision avoidance method for an autonomous vehicle is presented, with the NMPC strategy constructed using a single-track vehicle model and a nonlinear tyre model and solved using the C/GMRES algorithm [18]. Simulation results show that a vehicle equipped with the proposed controller can successfully avoid an obstacle, however the computational time is about 6 times longer than the sampling time so the solution is not implementable online. In the context of semi-autonomous vehicles applications, Gray et. al. [19], [20] presented a NMPC strategy for obstacle avoidance: the best path for the vehicle to follow according to specific criteria on the tracking error, distance from the obstacle and aggressiveness of the manoeuvre is computed by the high level path planner and passed on to the low-level path follower, which uses an NMPC strategy to follow the desired path, subject to input constraints. Results show the effectiveness of the proposed strategy however the high computational complexity of the overall solution meant that the problem could be solved again only offline.

In the scope of active safety systems, most solutions have so far focused in the control of the lateral dynamics of the vehicle using a linear or an explicit MPC formulation. For example, in [21] a yaw stability controller based on a LTV-MPC formulation using independent braking of the four wheels is presented. Hard constraints are imposed on both state and input and the proposed strategy is successfully tested in the sine and dwell test but with a considerable decrease in speed. In [22], a linear MPC is used in a lateral stability control application using the steer-by-wire system of a Rear Wheel Drive prototype vehicle. The sampling time is chosen at the low rate of 10ms with prediction and control horizons at 15 steps, and delay compensation is also employed by solving the optimization problem for the next time step. Simulation and experimental results using a slalom manoeuvre at a speed of 10m/s on a loose surface, show that the controller can successfully restrict the steering command from the driver when the yaw rate and sideslip angle limits are violated. In [23] a hybrid MPC and a switched MPC formulation for a yaw stability controller using AFS and differential wheel braking are presented. Simulation results using a hybrid MPC formulation against a standard ESC strategy show that the hybrid MPC converges faster to the target yaw rate and slip angle targets. Since the complexity of the hybrid MPC makes it unsuitable for online application, an explicit switched MPC is presented next, with experimental results showing it can successfully stabilize the vehicle under various scenarios on a slippery road by constraining the tyre slip angles within their limits. Another example of an explicit MPC law can be found in [24], where a yaw control strategy using a rear active differential is presented. Here the NMPC strategy is constructed using a single track model and, since it cannot be solved fast online, it is solved offline using the Nearest Point approach. Simulation results using more than 10^5 points from a rather limited set of variables show a good agreement between the proposed approach and a nominal NMPC controller but with some chattering, a problem which could be potentially corrected with higher number of offline computed points but at the expense of higher memory and computational requirements.

In this paper we explore recent developments in the area of fast NMPC and its application on the problem of controlling an electric vehicle at the limits of handling using combined longitudinal and lateral dynamics control. To this end, we employ a four-wheel nonlinear vehicle model coupled with a nonlinear tyre model to construct three strategies of different complexity: 1) a linear MPC strategy [5], 2) an NMPC strategy that employs the Real Time Iteration (RTI) scheme [15] as available in the ACADO Toolkit [25], and 3) an NMPC strategy that employs the PDIP method as available in FORCES Pro [26]. After comparing the three strategies against each other and against the optimal solution in terms of closed-loop performance and computational cost in a series of case studies, we deploy the most promising solution on an automotive-grade processor board and finally validate it in a high fidelity simulation environment under two limit-handling manoeuvres.

The paper is organised as follows. Section II introduces the nonlinear tyre and vehicle model and the steady-state cornering analysis used to generate the reference for the controllers to follow. Section III presents the general nonlinear program problem and how the three NMPC strategies can be derived from it. Section IV details the comparison of the three strategies as already introduced above, along with the dSPACE deployment. Finally, Section V presents the complete control structure that is then validated in CarMaker environment in Section VI.

II. Vehicle Model and Reference Generation

In this section we present the vehicle model along with a short description of the steady-state analysis used to generate feasible targets for the controller to follow. The formulation is similar to [4], [27], where the interested reader is referred to for more details.

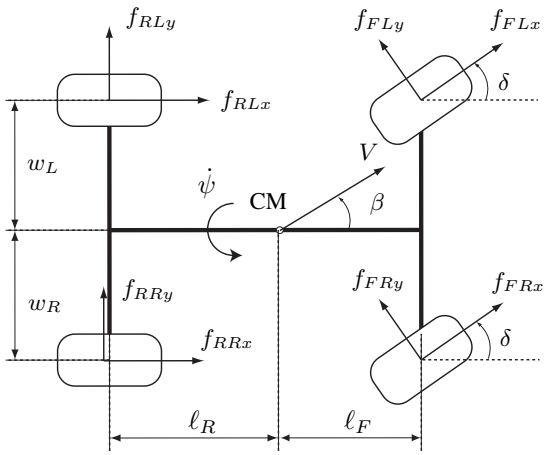


Fig. 1. The four-wheel vehicle model.

A. Vehicle Model

The Equations Of Motion (EOM) for the four-wheel vehicle model with front wheel steering (Fig. 1) are

$$\begin{aligned} m\dot{V} &= (f_{FLx}+f_{FRx})\cos(\delta-\beta) \\ &- (f_{FLy}+f_{FRy})\sin(\delta-\beta) \\ &+ (f_{RLx}+f_{RRx})\cos\beta \\ &+ (f_{RLy}+f_{RRy})\sin\beta, \qquad (1a) \\ \dot{\beta} &= \frac{1}{mV}\left[(f_{FLx}+f_{FRx})\sin(\delta-\beta)\right. \\ &+ (f_{FLy}+f_{FRy})\cos(\delta-\beta) \\ &- (f_{RLx}+f_{RRx})\sin\beta \\ &+ \left.(f_{RLy}+f_{RRy})\cos\beta\right] - \dot{\psi}, \qquad (1b) \\ I_z\ddot{\psi} &= \ell_F\left[(f_{FLy}+f_{FRy})\cos\delta\right. \\ &+ \left.(f_{FLx}+f_{FRx})\sin\delta\right] - \ell_R\left(f_{RLy}+f_{RRy}\right) \\ &+ w_L\left(f_{FLy}\sin\delta - f_{FLx}\cos\delta - f_{RLx}\right) \\ &+ w_R\left(f_{FRx}\cos\delta - f_{FRy}\sin\delta + f_{RRx}\right) \qquad (1c) \\ I_w\dot{\omega}_{ij} &= T_{ij} - f_{ijx}R_w, \qquad i=F,R,\ j=L,R. \qquad (1d) \end{aligned}$$

where the relevant variables and parameters are as defined in the *Notation* section at the beginning of the paper.

The tyre forces f_{ijx} and f_{ijy} in the above EOM are found as functions of the tyre slip using Pacejka's Magic Formula (MF) [28]. In particular, we first find the resultant tyre force coefficient μ_{ij} at each tyre using the MF:

$$\mu_{ij}(s_{ij}) = \mathrm{MF}(s_{ij}) = D\sin(C\mathrm{atan}(Bs_{ij})),$$

where $s_{ij} = \sqrt{s_{ijx}^2 + s_{ijy}^2}$ is the resultant tyre slip [28], and subsequently we calculate the longitudinal and lateral tyre force coefficients from

$$\mu_{ijk} = -\frac{s_{ijk}}{s_{ij}}\mu_{ij}(s_{ij}).$$

Then, the longitudinal and lateral tyre forces are given by

$$f_{ijx} = \mu_{ijx}f_{ijz}, \qquad f_{ijy} = \mu_{ijy}f_{ijz},$$

where the vertical force f_{ijz} on each of the four wheels is calculated as the sum of the static load on that wheel and the longitudinal/lateral weight transfers under longitudinal/lateral acceleration [27].

TABLE I
VEHICLE AND TYRE PARAMETERS

Parameter	Value	Parameter	Value
m (kg)	1137	ℓ_F (m)	1.187
I_z (kgm^2)	1174	ℓ_R (m)	1.313
I_w (kgm^2)	1.04	R_w (m)	0.298
w_L (m)	0.687	B	11.24
w_R (m)	0.687	C	1.45
h (m)	0.317	D	1

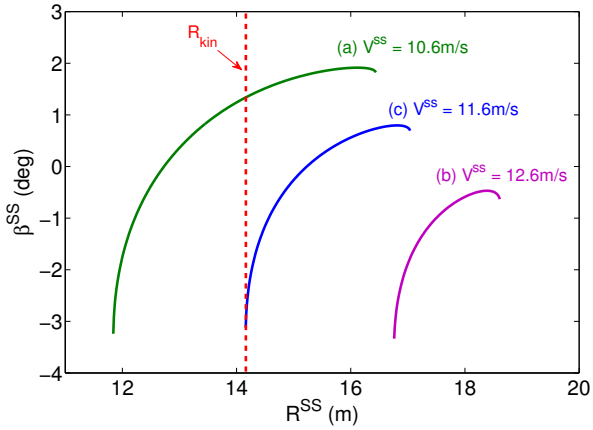


Fig. 2. Selection of target steady-state according to the driver's steering angle command of δ=10deg: (a) R_{kin} feasible at $V^{\mathrm{ss}} = 10.6$m/s; (b) R_{kin} not feasible at $V^{\mathrm{ss}} = 12.6$m/s; (c) R_{kin} coincides with the minimum calculated R^{ss} at $V^{\mathrm{ss}} = 11.6$m/s.

Table I shows the values for the above vehicle and tyre parameters, which correspond to a small sports car.

B. Reference Generation

Steady-state cornering analysis of the four-wheel vehicle model (1) is used to derive feasible targets for the controller to follow where, similar to common practice in vehicle stability control [29], we set the desired path radius from the driver as a function of the steering input by the kinematic relationship $R_{kin} = (\ell_F + \ell_R)/\delta$.

The desired path radius R_{kin} may or may not be feasible depending on the vehicle's velocity. Consider for example the steady-state conditions for a fixed δ^{ss} and a range of V^{ss} in Fig. 2. In all three cases the desired $R^{\mathrm{ss}} = R_{kin}$ is around 14m, according to the steering command of $\delta^{\mathrm{ss}} = 10$deg. Then, if the vehicle velocity is 10.6m/s the requested R_{kin} is feasible, whereas if the vehicle velocity is 12.6m/s the R_{kin} is smaller than the minimum achievable R^{ss} and not feasible anymore. In this case the controller will reduce the vehicle velocity so that the desired R_{kin} becomes feasible again by selecting a steady-state velocity such that R_{kin} coincides with the minimum R^{ss}, which in the above example corresponds to a maximum vehicle velocity of $V_{max} = 11.6$m/s (please refer to [5] for a detailed discussion in the reference generation used in this paper).

III. NonLinear Program Problem and MPC Strategies

In this section, we compare three MPC strategies of different levels of complexity in a series of simple simulation studies designed so that both the advantages and disadvantages of each strategy can be observed. To this end, we first obtain the optimal solution for each study and use it as a benchmark to compare the three solutions from two points of view: closed-loop performance and computational complexity. The section is therefore comprised by two parts, the first one presenting the optimal control problem under consideration and how this can be solved offline, and the second one showing how the problem can be simplified and solved online.

A. Optimal Solution

For the nonlinear continuous-time system with state and input x and u respectively

$$\dot{x} = f(x, u), \tag{2}$$

the discrete optimal control problem under consideration is

$$\begin{aligned}
\min_{x,u} \quad & \sum_{k=0}^{N-1} \Big[(x_k - x_{ref})^T Q (x_k - x_{ref}) \\
& \qquad + (u_k - u_{ref})^T R (u_k - u_{ref}) \Big], && \text{(3a)} \\
\text{s.t.} \quad & x_0 = x_{in}, && \text{(3b)} \\
& x_{k+1} = f(x_k, u_k), \qquad k = 0, ..., N-1, && \text{(3c)} \\
& h(x_k, u_k) \leq 0, \qquad k = 0, ..., N-1, && \text{(3d)}
\end{aligned}$$

The aim is to minimize the state and input error from a given reference (3a) along the simulation time $T_{sim} = NT_s$ where T_s is the sampling time, subject to the initial condition (3b), the discretised system dynamics (3c) and the state and input constraints (3d). The resulting NonLinear Program (NLP) problem can then be solved offline using one of the popular optimization methods: we employ the Sequential Quadratic Program (SQP) algorithm with an active set method to solve it, as available in the ACADO Toolkit [25]. In this way we obtain the benchmark against which the three online MPC strategies will be compared.

B. MPC Strategies

For the MPC strategies, the problem to solve is

$$\begin{aligned}
\min_{x,u} \quad & \sum_{k=0}^{M-1} \Big[(x_k - x_{ref})^T Q (x_k - x_{ref}) \\
& \qquad + (u_k - u_{ref})^T R (u_k - u_{ref}) \Big], && \text{(4a)} \\
\text{s.t.} \quad & x_0 = x_{in}, && \text{(4b)} \\
& x_{k+1} = f(x_k, u_k), \qquad k = 0, ..., M-1, && \text{(4c)} \\
& \underline{x} \leq x_k \leq \overline{x}, \qquad k = 0, ..., M-1, && \text{(4d)} \\
& \underline{u} \leq u_k \leq \overline{u}, \qquad k = 0, ..., M-1, && \text{(4e)}
\end{aligned}$$

where $M \leq N$ is the prediction horizon and the nonlinear constraints on state and input (3d) are replaced by simpler box constraints (4d)-(4e) for fairness of comparison between the simpler linear MPC strategy and the two NMPC strategies.

Then, the three real-time-implementable formulations investigated here are:

- a linear MPC strategy, where the nonlinear system dynamics (2) are linearized and discretised with the resulting Quadratic Program (QP) problem solved using the Primal Dual Interior Point method (PDIP) as available in FORCES Pro [26]
- an NMPC strategy that applies only the first SQP iteration on problem (4) according to the RTI scheme as available in the ACADO Toolkit [25]
- an NMPC strategy that applies the PDIP method as available in FORCES Pro [26] to (4) until convergence to the optimal solution

1) Linear MPC: From (4) and the short description of the MPC strategies above we can see that the main difference in the problem definition between the linear MPC and the rest of the strategies is how the discrete system dynamics are defined. Linearising the continuous system dynamics (2) about the equilibrium point $(x^{\text{ss}}, u^{\text{ss}})$ gives

$$\dot{x} = A_c x + B_c u - (A_c x^{\text{ss}} + B_c u^{\text{ss}}), \tag{5}$$

where $(A_c x^{\text{ss}} + B_c u^{\text{ss}})$ is a constant. Then discretising the above affine system we get

$$x_{k+1} = A_d x_k + B_d u_k - c, \tag{6}$$

with

$$\begin{aligned}
A_d &= e^{A_c T_s}, \\
B_d &= \int_0^{T_s} e^{A_c \eta} d\eta B_c, \\
c &= \int_0^{T_s} e^{A_c \eta} d\eta (A_c x^{\text{ss}} + B_c u^{\text{ss}}),
\end{aligned}$$

assuming that the input remains constant for the discretisation interval. The resulting QP can then be solved using the PDIP method as available in FORCES Pro [26].

2) NMPC - RTI scheme and PDIP method: For the two NMPC strategies we use one step of the explicit Runge-Kutta 4th order method to derive the nonlinear discrete dynamics (4c) from the continuous dynamics (2): the specific method was found to give a good approximation of the continuous dynamics for our system at the chosen sampling time of 50ms. The resulting NMPC can then be solved using the RTI scheme or the PDIP method:

- NMPC-RTI
  In the case of a real-time application like the one considered here, the RTI scheme can be used for fast solutions of problem (4): this scheme, in its simplest form, has the benefit of producing fast but suboptimal solutions by precomputing the necessary sensitivities and performing only one SQP iteration (see [15], [25] for more details). This approach can quickly lead to convergence if the

solution does not change much from one time step to the next but can also diverge.

- NMPC-PDIP
  We can also try to solve (4) using the PDIP method, as available in the Forces Pro NLP solver [26], until convergence. This approach attempts to solve the NMPC problem in a relatively short time by employing a Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm for the computation of the Hessian of the Lagrangian and can give solutions that are very close to the optimal.

IV. Comparison of the Three MPC Strategies

In this section we first compare the linear MPC, NMPC-RTI and NMPC-PDIP strategies as presented in section III-B against the optimal solution from section III-A for a range of simple simulation studies on a standard desktop machine (i7-2600k at 3.40GHz with 16GB of memory), and then deploy the most promising solution on a dSPACE DS1005 board (PowerPC 750GX at 1.00GHz with 128MB of memory).

We will neglect for now the fast wheel dynamics (1d), so we set for both the simulation model and the internal model for the MPC strategies $x = [V\ \beta\ \dot{\psi}]^T$ and $u = [s_{RLx}\ s_{RRx}]^T$ [5]. The input constraints are then set, according to the MF parameters of Table I, to

$$|s_{Rjx}| \leq 0.15, \tag{8}$$

while we also set a constraint on the product of the vehicle's yaw rate and velocity based on the lateral acceleration limit

$$-\mu_{max} g \leq \dot{\psi} V \leq \mu_{max} g, \tag{9}$$

which for the MPC strategies is simplified to a constraint on the yaw rate only as a function of the velocity at the beginning of the prediction horizon [5]:

$$|\dot{\psi}| \leq \mu_{max} g / V_{in}. \tag{10}$$

In the test scenarios considered here, the vehicle is initially moving on a straight line and at time $t = 0$s we apply a step steering input for the duration of $T = 10$s [1], with the initial speed chosen so that it is greater than the corresponding V_{max} for that steering input. Each controller will then aim to stabilize the vehicle to the steady-state reference $x_{ref} = [V^{\text{ss}}\ \beta^{\text{ss}}\ \dot{\psi}^{\text{ss}}]^T$, $u_{ref} = [s^{\text{ss}}_{RLx}\ s^{\text{ss}}_{RRx}]^T$ by minimising (4b) subject to (4c)-(4e). The sampling time and the prediction horizon for the MPC strategies are set to $T_s = 50$ms and $M = 20$steps respectively, while for the evaluation of the performance of the MPC strategies we use the closed-loop cost, defined as the summation of the running costs

$$J_{cl} = \sum_{k=0}^{\lceil \frac{T-T_s}{T_s} \rceil} \Big[(x_k - x_{ref})^T Q (x_k - x_{ref}) + (u_k - u_{ref})^T R (u_k - u_{ref}) \Big],$$

where $\lceil \cdot \rceil$ is the ceiling function which maps a real number to the smallest following integer.

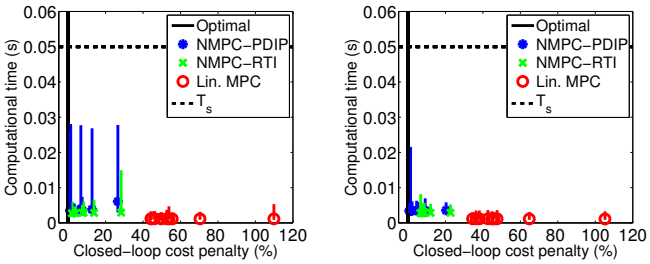
[1] the simulation time chosen long enough so that the states always converge to the steady-state reference before the end of each test.

TABLE II
Comp. times and performance results from the MPC strategies

	Avg comp. time (ms)	Max comp. time (ms)	Min per. penalty (%)	Max per. penalty (%)
Linear MPC	1.1	5.3	28.08	109.85
NMPC-RTI	3.0	14.9	2.01	$5.91 \cdot 10^5$
NMPC-PDIP	3.6	29.5	0.79	28.23

Table II shows the average and maximum computational times along with the minimum and maximum closed-loop costs (expressed as percentage difference from the optimal) for the three MPC strategies for a range of step steering inputs from 2 to 10deg and different initial velocities, ranging from 1m/s to 4m/s above the V_{max} for that steering input[2]. Looking at the computational times in Table II, we can see that they scale according to the problem complexity, with the linear MPC being the fastest and the NMPC-PDIP the slowest across all results. Another interesting point is the maximum observed time for the NMPC-PDIP which is much higher than the two other strategies: this happens when the NMPC-PDIP reaches the maximum number of iterations allowed (which in our tests is set to 200 iterations) without fully converging, at which point it gives the last computed sub-optimal solution. Looking at the performance penalty for the three strategies on the last two columns of Table II, we observe that the linear MPC is consistently above 28.08% difference from the optimal, but does not go above 110%, while the NMPC-PDIP only reaches a maximum of 28.23%. The NMPC-RTI strategy on the other hand reaches high maximum closed-loop cost values due to infeasibility problems, a result that shows the main disadvantage of performing only one SQP iteration at each time step. Fig. 3 shows the computational time versus performance penalty plots for the set of simulation tests from Table II. It can be confirmed that the linear MPC strategy (in red, with the red circle showing the average for each test) performs almost the same across all the tests and, apart from only a few occasions when more iterations of the PDIP method are used to find a solution, it returns a solution in less than 5ms. On the other hand, the NMPC-PDIP strategy (in blue, with the blue asterisk showing the average for each test) performs closer to the optimal across all tests and mostly drops in performance when the initial velocity is further away from the reference velocity V_{max}. However this is done at the expense of longer computational times since in quite a few tests the maximum number of iterations is reached at least once, hence the much larger maximum times observed in some of the results. Finally, the NMPC-RTI strategy (in green, with the green x showing the average for each test), shows excellent performance with low computational times when the initial state is close to the target, but quickly drifts to higher closed-loop penalty values for higher initial state errors, showing the main disadvantage of using this strategy as already observed in the analysis of Table II above.

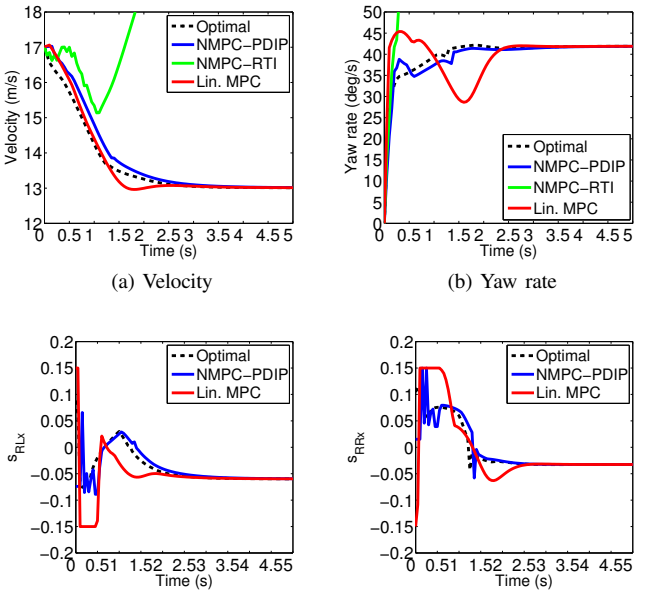
[2] the range of steering inputs and initial velocities chosen so that the original NLP problem subject to the hard yaw rate constraint is always feasible for the given vehicle topology and actuator limits.



(a) $V_0 = V_{max} + 1m/s$ (b) $V_0 = V_{max} + 2m/s$

(c) $V_0 = V_{max} + 3m/s$ (d) $V_0 = V_{max} + 4m/s$

Fig. 3. Computational times versus performance penalty from the optimal solution for a range of step steering inputs from 2 to 10deg and different initial velocities.



(a) Velocity (b) Yaw rate

(c) Rear-left long. slip (d) Rear-right long. slip

Fig. 4. Velocity, yaw rate and longitudinal slip histories for a step steering input of 8deg and an initial velocity difference from V_{max} of 4m/s for the three MPC strategies (note that for clarity reasons the longitudinal slip results for the NMPC-RTI have been omitted).

An example of the difference in state regulation from the optimal for the three MPC strategies in one of the test scenarios presented in Fig. 3 above can be seen in Fig. 4 where we find the velocity, sideslip angle, yaw rate and longitudinal slip time histories for a step steering input of 8deg and an initial velocity which is 4m/s higher than V_{max} for this steering input. While the velocity time histories for the linear MPC and the NMPC-PDIP strategies are similar and both close to the optimal trajectory (Fig. 4a), the yaw

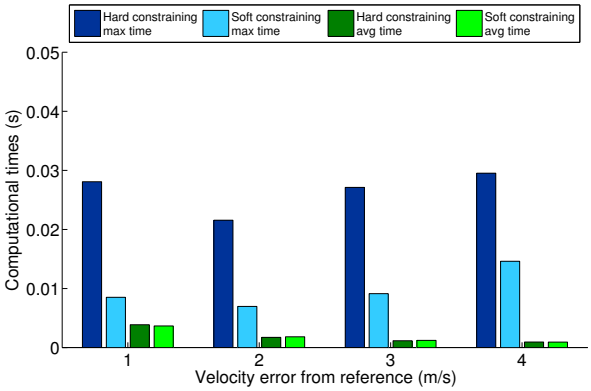


Fig. 5. Comparison of maximum (blue bars) and average (green bars) computational times for the NMPC-PDIP (in dark blue and green) and the NMPC-PDIP with soft constraints (in light blue and green) for the range of test scenarios considered in this section, starting from different initial velocities.

rate time histories are quite different. While the linear MPC strategy exhibits large oscillations, the NMPC-PDIP strategy remains close to the optimal solution (Fig. 4b), with only a small overshot at the yaw rate, which is directly connected to the oscillations observed from the NMPC-PDIP strategy in the longitudinal slip time histories (Figs. 4c-4d) and is the result of the NMPC-PDIP strategy finding it difficult to cope with the hard yaw rate constraint. Despite this, the NMPC-PDIP strategy shows excellent response with results very close to the optimal solution and demonstrates the importance of accounting for the nonlinear system dynamics in the form of the equality constraint (4c) rather than linearising the system dynamics as is the case with the linear MPC strategy. Finally, for this test scenario the vehicle with the NMPC-RTI strategy quickly becomes unstable due to the high initial state error.

While the NMPC-RTI convergence problems with higher initial state errors, as explained above, could be possibly addressed using a shorter sampling time and/or more SQP iterations, the fact remains that the NMPC-PDIP strategy shows more promising results, the main problem been the longer computational times. One way to help the PDIP solver achieve convergence faster while avoiding infeasibility problems is by soft constraining the state by introducing slack variables into the cost function (4a) and relaxing the state constraints (4d):

$$\min_{x,u} \quad \sum_{k=0}^{M-1} \Big[(x_k - x_{ref})^T Q (x_k - x_{ref}) + (u_k - u_{ref})^T R (u_k - u_{ref}) + \rho_\epsilon \epsilon_k \Big], \tag{12a}$$

$$\text{s.t.} \quad x_0 = x_{in}, \tag{12b}$$

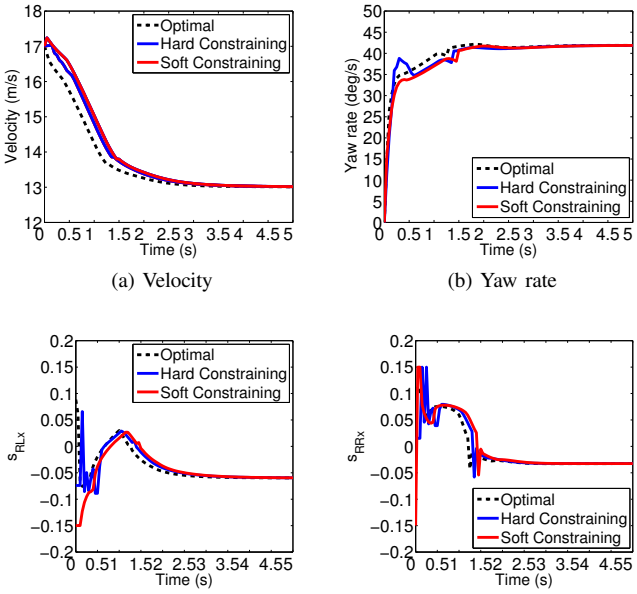
$$x_{k+1} = f(x_k, u_k), \qquad k = 0, ..., M-1, \tag{12c}$$

$$\underline{x} - \epsilon_k \leq x_k \leq \overline{x} + \epsilon_k, \; k = 0, ..., M-1, \tag{12d}$$

$$\underline{u} \leq u_k \leq \overline{u}, \qquad k = 0, ..., M-1, \tag{12e}$$

$$\epsilon_k \geq 0, \qquad k = 0, ..., M-1, \tag{12f}$$

where $\epsilon_k \in \mathbb{R}^+$ $(k = 0, ..., M-1)$ and ρ_ϵ are the slack variables and their weight respectively.



(a) Velocity (b) Yaw rate

(c) Rear-left long. slip (d) Rear-right long. slip

Fig. 6. Velocity, yaw rate and longitudinal slip histories for a step steering input of 8deg and an initial velocity error of 4m/s for the hard constrained and the soft constrained NMPC-PDIP strategy.

Fig. 5 shows the change in average and maximum computational times for the NMPC-PDIP strategy after softening the yaw rate constraint (10). The maximum time has decreased to less than half in all cases, while the average times show no difference from the hard constrained NMPC-PDIP strategy despite the fact that the inclusion of the slack variables has increased the number of optimisation variables. It is worth noting here also that no infeasibility problems have been observed after softening the yaw rate constraint and that the maximum number of 200 iterations was never reached across all cases. These results confirm that soft constraining not only removes infeasibility problems in the solution of the optimisation problem at hand but also helps in reaching a solution faster.

Returning to the example scenario examined in Fig. 4, in Fig. 6 we see the difference in response from the vehicle with the NMPC-PDIP strategy after softening the yaw rate constraint. While the velocity time histories are similar (Fig. 6a), the yaw rate overshot has disappeared in the soft constrained NMPC-PDIP case (Fig. 6b), a result also linked to the smoother longitudinal slip inputs from this strategy, as evidenced in Figs. 6c-6d.

dSPACE Deployment

The soft constrained NMPC-PDIP strategy has been also deployed on a dSPACE DS1005 board (PowerPC 750GX at 1.00GHz with 128MB global main memory). The limited processing power of such platform means that it was necessary to cap the maximum number of iterations that the solver can perform before returning a (sub-)optimal solution to 25. However, since each iteration takes a fixed time to run, this also means that we can guarantee that the solver will always return a solution within the given sampling time.

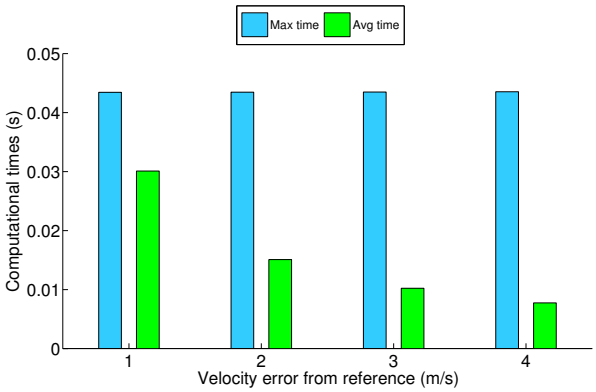
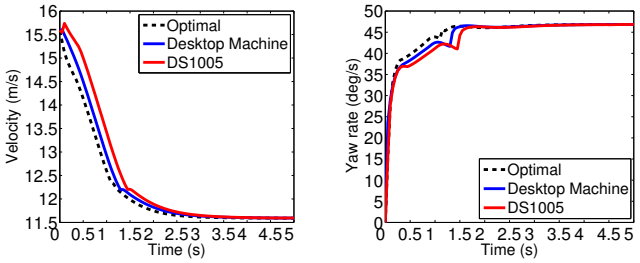


Fig. 7. Maximum (blue bars) and average (green bars) computational times for the soft constrained NMPC-PDIP after deployment on the DS1005.

In order to test the soft-constrained NMPC-PDIP strategy in real-time, we connected it again with a simulation model that neglects the fast wheel speed dynamics (1d) and deployed the complete closed-loop control system on the dSPACE DS1005 board. This involved deploying the source code for the soft constrained NMPC-PDIP solver and the simulation model as one closed-loop model, along with linking any additional files needed by the solver. Then, to record the computational times for the solver the dSPACE Profiler was used: this application runs on the host machine and, by receiving time-stamped events, can provide information on the timing of a defined task (such as the time to run the solver per call).

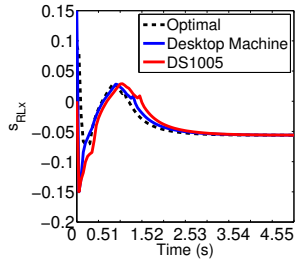
Fig. 7 shows the average and maximum computational times when the same series of case studies as before is performed on the DS1005. We notice that the maximum computational time across all case studies is around 43ms which corresponds to the set maximum number of 25 iterations per call of the solver, while the relative increase in computational effort can also be seen in the average times. However, the loss in performance due to the cap in the maximum number of iterations is less than expected. As we can see from Fig. 8 for a characteristic example of a scenario where the maximum number of iterations is reached multiple times, the velocity, yaw rate and longitudinal slip trajectories for the deployed controller remain close to the trajectories obtained from the desktop machine (where the maximum number of iterations is never reached).

From the above short analysis it is obvious that NMPC solutions are in general very demanding in terms of required computational power. However, using the PDIP method we can obtain maximum performance for the given hardware and therefore – after setting a cap on the maximum number of iterations – it is possible to deploy such solutions on real time hardware: as we have seen here, the proposed soft constrained NMPC-PDIP strategy can be successfully deployed with minimal performance loss, even for the extreme step steering input cases considered so far.

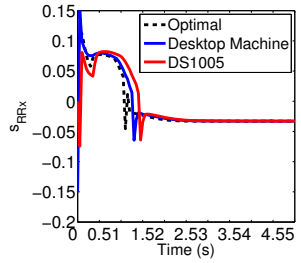


(a) Velocity

(b) Yaw rate



(c) Rear-left long. slip



(d) Rear-right long. slip

Fig. 8. Velocity, yaw rate and longitudinal slip histories for a step steering input of 10deg and an initial velocity error of 4m/s for the soft constrained NMPC-PDIP strategy on the desktop machine and the DS1005.

V. NMPC-PDIP with Sliding Mode Slip Controller

The soft constrained NMPC-PDIP strategy (12) is cascaded with a Sliding Mode slip controller that computes the necessary torques on the rear wheels based on the requested longitudinal slips [5], with the complete control structure seen in Fig. 9. We also set two extra inequality constraints due to implementation reasons, one that restricts the sideslip angle of the vehicle for subjective feel and another one that considers the electric motor limits in the form of its static torque map. Then the state and input constraints are:

A. State constraints

As in (10), a yaw rate constraint as a function of the current velocity V_{in} is set at the beginning of the optimization and fixed throughout the prediction horizon

$$|\dot{\psi}| \leq \mu_{max} g / V_{in}. \tag{13}$$

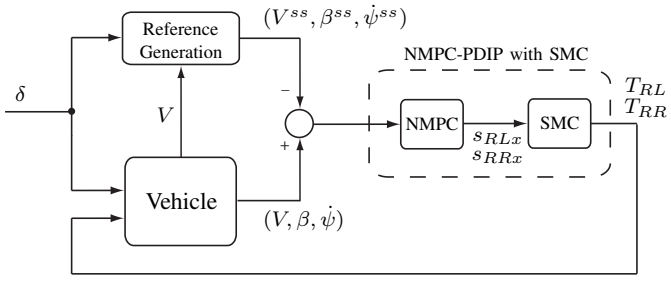


Fig. 9. Block diagram of the final control structure.

A constraint on the maximum sideslip angle is also set as a function of V_{in}:

$$|\beta| = \begin{cases} 2\dfrac{k_1}{V_{ch}^3} V_{in}^3 - 3\dfrac{k_1}{V_{ch}^2} V_{in}^2 + k_2, & V_{in} < V_{ch} \\ k_2 - k_1, & V_{in} \geq V_{ch} \end{cases} \tag{14}$$

where V_{ch} is the characteristic velocity of the vehicle and the positive constants k_1 and k_2 are tuning parameters.

B. Input constraints

The longitudinal slips on the rear wheels should never exceed the maximum allowable value for safe operation of the vehicle so we set, similarly to (8) the slip input constraints as

$$|s_{Rjx}| \leq 0.15. \tag{15}$$

Since the wheel dynamics are neglected in the internal model for the NMPC-PDIP strategy, we can not directly account for the motor limits. We therefore construct an additional constraint on the slip input in order avoid excessive torque requests to the two motors. If the maximum torque that can be provided by a motor is T_{Rj}^{max}, then the maximum longitudinal force on the wheel – assuming steady-state conditions – is

$$f_{Rjx}^{max} = T_{Rj}^{max} / R_w, \tag{16}$$

and using the reverse MF the torque based limit on the longitudinal slip on the tyre can be computed as

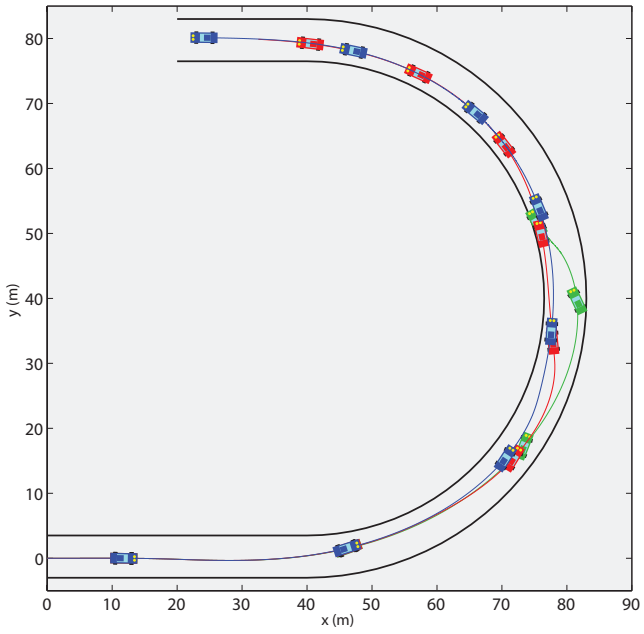
$$s_{Rjx}^{max} \leq \frac{1}{B} tan\left(\frac{1}{C} sin^{-1}\left(\frac{f_{Rjx}^{max}}{D f_{Rjz}}\right)\right). \tag{17}$$

Then, assuming that the motor provides equal maximum torque in the positive and negative direction, we can compare the two limits (15) and (17) and set the input constraints at the beginning of the prediction horizon as

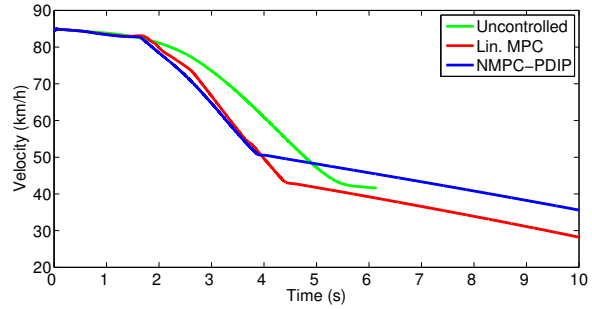
$$|s_{Rjx}| \leq min(0.15, s_{Rjx}^{max}). \tag{18}$$

VI. Simulation Results

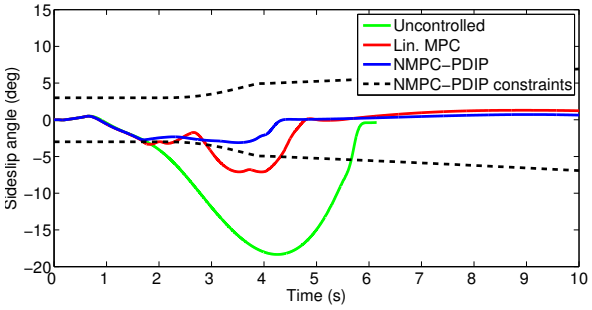
In this section we compare the NMPC-PDIP with Sliding Mode slip controller from section V in Carmaker environment against a vehicle without a controller and one that applies a linear MPC controller instead on problem (12) with the same constraints (13)-(18) in two simulation scenarios: 1) a U-turn, where the vehicle enters a corner with excessive speed and 2) an obstacle avoidance manoeuvre according to ISO 3888-2:2011 [30]. The purpose of the two tests is to show how the velocity regulation combined with the lateral dynamics control – while respecting the system constraints – from the two MPC strategies manage to keep the vehicle stable and what are the advantages of using a NMPC strategy against the faster but sub-optimal linear MPC strategy in real world critical situations. Note that for both simulation studies a standard desktop machine (i7-2600k at 3.40GHz with 16GB of memory) is used.



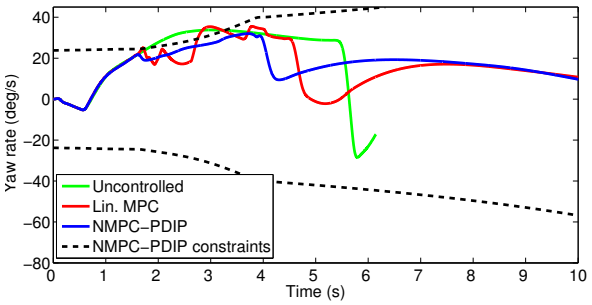
(a) Trajectory



(b) Velocity

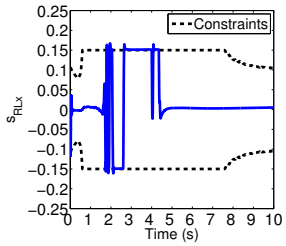


(c) Sideslip angle

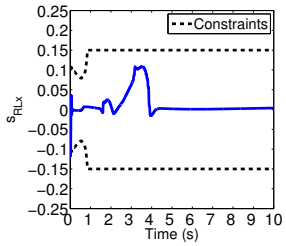


(d) Yaw rate

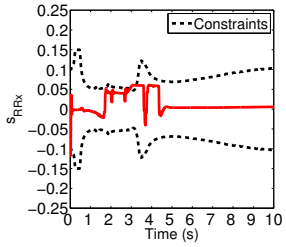
Fig. 10. Comparison of the uncontrolled vehicle (in green), the vehicle with the linear MPC (in red) and the vehicle with the NMPC-PDIP (in blue) in the U-turn avoidance scenario.



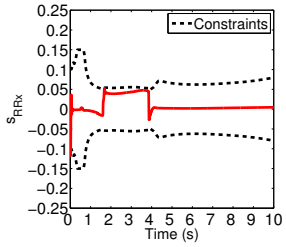
(a) Lin. MPC rear-left long. slip



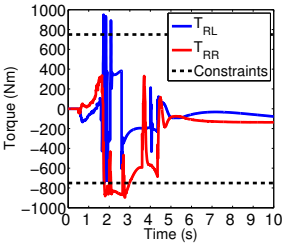
(b) NMPC-PDIP rear-left long. slip



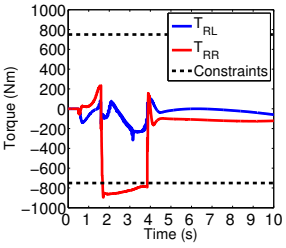
(c) Lin. MPC rear-right long. slip



(d) NMPC-PDIP rear-right long. slip



(e) Lin. MPC torques



(f) NMPC-PDIP torques

Fig. 11. Longitudinal slip (actual) and torque (requested) time histories for the linear MPC and the NMPC-PDIP strategies in the U-turn scenario.

A. *U-turn Scenario*

For the U-turn scenario, we use the driver model in CarMaker to steer the vehicle through a turn of 40m radius. The road is dry (μ_{max} =1) and 6.5m wide, the entry speed is set at 85km/h, and we assume that no acceleration or braking commands come from the driver.

As we can see from Fig. 10a, the uncontrolled vehicle looses control due to high entry speed and eventually leaves the road. The two MPC strategies on the other hand keep the vehicle on the road, but with a small difference: looking more closely especially to the first half of the turn, we can see that the NMPC-PDIP manages a much smoother trajectory compared to the linear MPC.

The above observation on the difference between the trajectories of the vehicle with the NMPC-PDIP strategy against the one with the linear MPC is directly connected to how the two strategies regulate the state as seen in Fig. 10. While the velocity regulation from the two strategies is, apart from the exit speed, mostly the same (Fig. 10b), the sideslip angle and yaw rate time histories (Figs. 10c-10d) show oscillations for the linear MPC strategy due to the simpler linear internal model used in this case which can not predict as effectively the state violations.

The difference in response between the two strategies is also apparent in the longitudinal slip and torque time histories as

found in Fig. 11, where we observe excessive oscillations in the longitudinal slip demands from the linear MPC (Fig. 11a and Fig. 11c), especially in the case of the less loaded rear left wheel, which also translate into violent torque commands (Fig. 11e). The NMPC-PDIP strategy on the other hand shows much smoother torque commands (Fig. 11f) and a more efficient longitudinal slip regulation (Fig. 11b and Fig. 11d). Note that the torque limit violations as seen in Figs. 11e-11f occur due to the fact that the two MPC strategies do not directly control the torque on the wheels. However, however it has been noticed in our studies that removing them from the MPC formulations result in much higher demanded torques.

Finally, the computational times for the linear MPC returned an average and a maximum time of 0.42ms and 0.98ms respectively, while for the NMPC-PDIP the corresponding times were 1.9ms and 3.4ms, which are much lower than the sampling time of 50ms for the two strategies.

B. Obstacle Avoidance Scenario

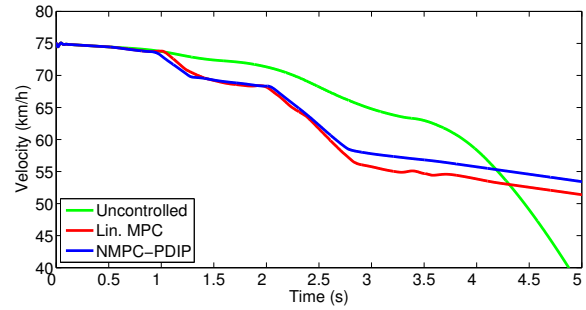
For the obstacle avoidance scenario we use again the driver model available in CarMaker, but this time to navigate through a double-lane change, as defined by three valleys of cones according to the specifications of ISO 3888-2:2011 [30]. The road is assumed again dry ($\mu_{max} = 1$), the entry speed is set to 75km/h, while no acceleration or braking commands come from the driver.

Fig. 12 shows the trajectories for the three vehicles. We can see that the uncontrolled vehicle spins out of control towards the end of the manoeuvre, while the two MPC strategies manage to keep the vehicle stable. However, only the vehicle with the NMPC-PDIP strategy manages to successfully complete the test since the linear MPC fails to pass through the last valley of cones without hitting them.

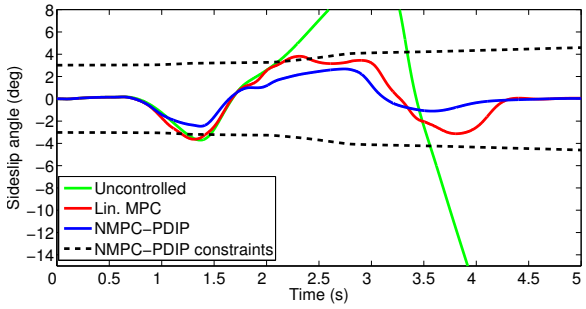
This slight difference between the trajectories of the two MPC strategies is again related, as in the U-turn scenario above, to the way they handle the system constraints. As observed in Fig. 13, while the velocity time histories between the linear MPC and the NMPC-PDIP are almost identical throughout the manoeuvre (Fig. 13a), the sideslip angle and yaw rate histories are quite different, with the linear MPC showing higher values and more oscillations in Figs. 13b-13c caused again by the simpler linear internal model used in this case.

Looking at Fig. 14, excessive oscillations are again observed in the longitudinal slip time histories from the linear MPC (Fig. 14a and Fig. 14c) and violent torque commands (Fig. 14e) which are in strong contrast to the subtle regulation from the NMPC-PDIP (Fig. 14b, Fig. 14d and Fig. 14f). Note that the torque limit violations (Figs.14e-14f) occur again due to the fact that the two MPC strategies do not directly control the torque on the wheels.

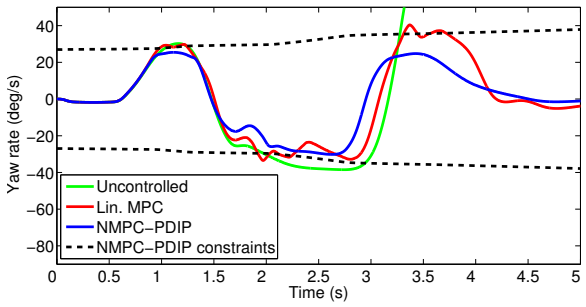
Finally, for the double-lane change scenario the average and maximum computational times for the linear MPC were 0.44ms and 0.75ms respectively, while for the NMPC-PDIP the corresponding times were 2.1ms and 3.3ms, times similar to the ones found for the U-turn scenario.



(a) Velocity



(b) Sideslip angle



(c) Yaw rate

Fig. 13. Velocity, sideslip angle and yaw rate time histories for the uncontrolled vehicle (in green), the vehicle with the linear MPC (in red) and the one with the NMPC-PDIP (in blue) in the obstacle avoidance scenario.

VII. Conclusion

In this paper we have presented a real-time NMPC for stabilisation of an electric vehicle near the limits of handling using combined longitudinal and lateral dynamics control. Using a nonlinear four-wheel vehicle model coupled with a nonlinear tyre model, three MPC strategies of different complexity that can be implemented online have been constructed and compared against each other and against the optimal solution in terms of closed-loop performance and computational cost. Results show that, while the linear MPC strategy is the fastest solution, the NMPC strategy using the PDIP method can achieve a much better performance close to the optimal solution while still been implementable online. The importance of soft constraining the state is investigated next, with results showing that it not only eliminates infeasibility problems in the solution of the optimisation problem, but it can also help reach a solution faster. The derived soft constrained NMPC-PDIP strategy is also deployed on an automotive grade dSPACE board: here, to avoid overrun problems, the maximum number

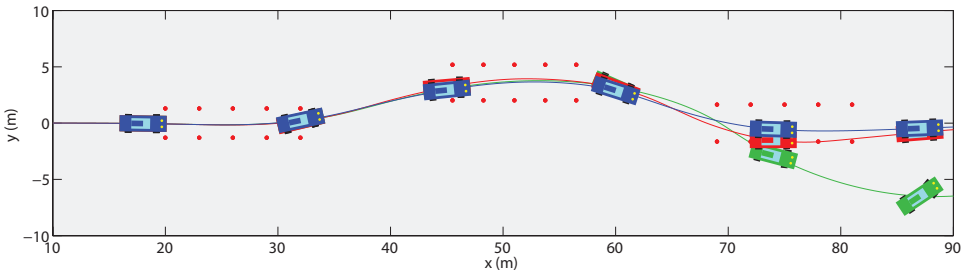
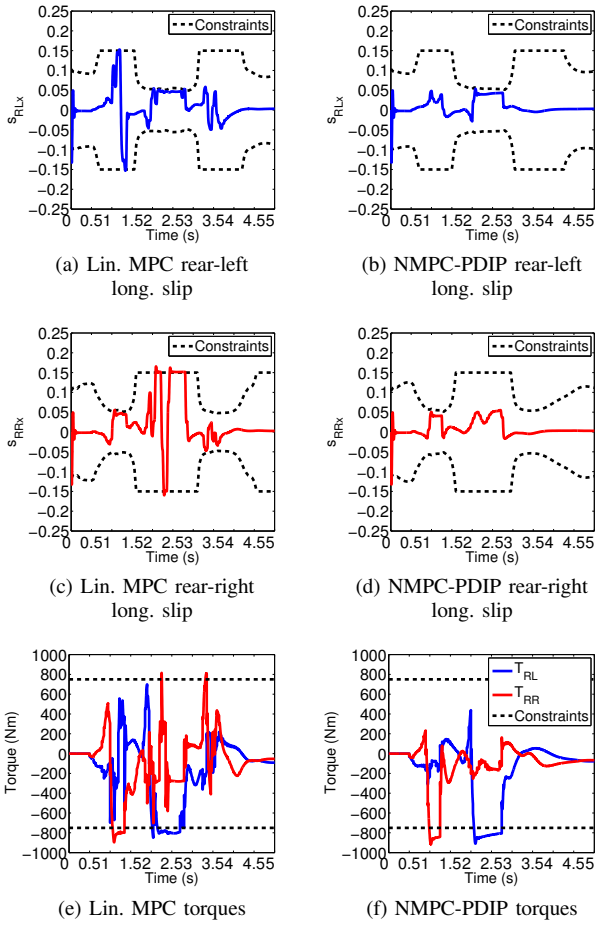


Fig. 12. Trajectory of the uncontrolled vehicle (in green), the vehicle with the linear MPC (in red) and the one with the NMPC-PDIP (in blue) in the obstacle avoidance scenario.



(a) Lin. MPC rear-left long. slip

(b) NMPC-PDIP rear-left long. slip

(c) Lin. MPC rear-right long. slip

(d) NMPC-PDIP rear-right long. slip

(e) Lin. MPC torques

(f) NMPC-PDIP torques

Fig. 14. Longitudinal slip (actual) and torque (requested) time histories for the linear MPC and the NMPC-PDIP strategies in the obstacle avoidance scenario.

of iterations had to be capped. This shows once again the relative trade-off between problem complexity and performance in order to stay real-time implementable, but also points to the fact that using the PDIP method we can obtain either maximum performance for a given hardware or conversely select the necessary hardware given a required minimum performance. Finally the soft constrained NMPC-PDIP stategy is tested in a high-fidelity simulation environment under two limit-handling scenarios: a U-turn, where the vehicle enters a corner with excessive speed and an obstacle avoidance manoeuvre. It is shown that the NMPC-PDIP strategy can achieve a better negotiation of both manoeuvres when compared to a linear MPC strategy, with lower sideslip angle and yaw rate values and smoother torque demands.

References

[1] T. J. Gordon, M. Klomp, and M. Lidberg, "Control mitigation for over-speeding in curves: Strategies to minimize off-tracking," in *11th International Symposium on Advanced Vehicle Control*, Seoul, Korea, September 9-12 2012.

[2] J. Kim and H. Kim, "Electric vehicle yaw rate control using independent in-wheel motor," in *Power Conversion Conference - Nagoya, 2007. PCC '07*, April 2007, pp. 705–710.

[3] J. Kang, Y. Kyongsu, and H. Heo, "Control allocation based optimal torque vectoring for 4WD electric vehicle," 2012, sAE Technical Paper 2012-01-0246.

[4] E. Siampis, M. Massaro, and E. Velenis, "Electric rear axle torque vectoring for combined yaw stability and velocity control near the limit of handling," in *Decision and Control (CDC), 2013 IEEE 52st Annual Conference on*, 2013.

[5] E. Siampis, E. Velenis, and S. Longo, "Rear wheel torque vectoring model predictive control with velocity regulation for electric vehicles," *Vehicle System Dynamics*, vol. 53, no. 11, pp. 1555–1579, 2015.

[6] S. Di Cairano, "An industry perspective on mpc in large volumes applications: Potential benefits and open challenges," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 4, no. PART 1, 2012, pp. 52–59.

[7] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 3, pp. 566–580, May 2007.

[8] M. Canale, M. Milanese, and C. Novara, "Semi-active suspension control using fast model-predictive techniques," *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 6, pp. 1034–1046, Nov 2006.

[9] N. Giorgetti, A. Bemporad, I. Kolmanovsky, and D. Hrovat, "Explicit hybrid optimal control of direct injection stratified charge engines," in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, vol. 1, June 2005, pp. 247–252.

[10] R. Schallock, K. Muske, and J. Peyton Jones, "Model predictive functional control for an automotive three-way catalyst," 2009, sAE Int. J. Fuels Lubr. 2(1):242-249.

[11] W. Qiu, Q. Ting, Y. Shuyou, G. Hongyan, and C. Hong, "Autonomous vehicle longitudinal following control based on model predictive control," in *Control Conference (CCC), 2015 34th Chinese*, 2015, pp. 8126–8131.

[12] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "Mpc-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265–291, 2005.

[13] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat, "Predictive control approach to autonomous vehicle steering," in *American Control Conference, 2006*, June 2006, pp. 6 pp.–.

[14] 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 System Dynamics*, vol. 46, no. sup1, pp. 611–628, 2008.

[15] M. Diehl, H. Bock, J. P. Schlder, R. Findeisen, Z. Nagy, and F. Allgwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577 – 585, 2002.

[16] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles," in *Control Conference (ECC), 2013 European*, July 2013, pp. 4136–4141.

[17] M. Nanao and T. Ohtsuka, "Nonlinear model predictive control for vehicle collision avoidance using c/gmres algorithm," in *Control Applications (CCA), 2010 IEEE International Conference on*, Sept 2010, pp. 1630–1635.

[18] "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563 – 574, 2004.

[19] A. Gray, M. Ali, Y. Gao, J. Hedrick, and F. Borrelli, "Integrated threat assessment and control design for roadway departure avoidance," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 1714–1719.

[20] Y. Gao, A. Gray, J. Frasch, T. Lin, E. Tseng, J. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *Proceedings of the 11th International Symposium on Advanced Vehicle Control*, Sept 2012.

[21] O. Barbarisi, G. Palmieri, S. Scala, and L. Glielmo, "Ltv-mpc for yaw rate control and side slip control with dynamically constrained differential braking," *European Journal of Control*, vol. 15, no. 34, pp. 468 – 479, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0947358009710015

[22] C. Beal and J. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 4, pp. 1258–1269, July 2013.

[23] S. Di Cairano, H. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 4, pp. 1236–1248, July 2013.

[24] M. Canale and L. Fagiano, "Vehicle yaw control using a fast NMPC approach," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec 2008, pp. 5360–5365.

[25] B. Houska, H. Ferreau, and M. Diehl, "An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

[26] A. Domahidi and J. Jerez, "FORCES Professional," embotech GmbH (http://embotech.com/FORCES-Pro), Jul. 2014.

[27] E. Velenis, D. Katzourakis, E. Frazzoli, P. Tsiotras, and R. Happee, "Steady-state drifting stabilization of RWD vehicles," *Control Engineering Practice*, vol. 19, no. 11, pp. 1363–1376, November 2011.

[28] E. Bakker, L. Nyborg, and H. Pacejka, "Tyre modelling for use in vehicle dynamics studies," 1987, sAE Technical Paper 870421.

[29] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Springer, 2012.

[30] "BS ISO 3888-2:2011. Passenger cars – Test track for a severe lane-change manoeuvre Part 2: Obstacle avoidance," British Standards Institution, London, GB, Standard, 2011.

https://dspace.org/

A real-time nonlinear model predictive control strategy for stabilisation of an electric vehicle at the limits of handling

Siampis, Efstathios; Velenis, Efstathios; Gariuolo, Salvatore; Longo, Stefano

2018-10-09
IEEE
https://doi.org/10.1109/TCST.2017.2753169
Efstathios Siampis, Efstathios Velenis, Salvatore Gariuolo and Stefano Longo. A real-time nonlinear model predictive control strategy for stabilisation of an electric vehicle at the limits of handling. IEEE Transactions on Control Systems Technology, Volume 26, Issue 6, November 2018, pp. 1982-1994

Downloaded from a DSpace repository. https://dspace.org/