

# Approximate NMPC for vehicle stability: design, implementation and SIL testing

Massimo Canale<sup>a,b</sup> Lorenzo Fagiano<sup>a</sup> Valentino Razza<sup>a</sup>

<sup>a</sup>*Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy.*

<sup>b</sup>*Corresponding author. e-mail: massimo.canale@polito.it*

---

## Abstract

In this paper, Nonlinear Model Predictive Control is used to improve vehicle stability and handling by means of a rear active differential. In order to allow on-line control computations within the required sampling time, a Set Membership approximation of the designed controller is employed. The real applicability and effectiveness of such a technique, as well as the improvement over an existing control approach based on Internal Model Control, is shown through the implementation on a commercial embedded device with limited computational capacity and the testing via software-in-the-loop simulations of demanding maneuvers, using an accurate non-linear vehicle model.

*Key words:* Nonlinear model predictive control, Vehicle yaw control, Software-in-the-loop test, Constrained control, Efficient model predictive control implementation

---

## 1 Introduction

Active vehicle control systems aim to enhance handling performance ensuring stability in critical situations. Several solutions to active chassis control have appeared in the last years. All of the proposed strategies modify the vehicle

dynamics by means of suitable yaw moments that can be generated in different ways (see e.g. the works of Ackermann and Sienel (1993); Van Zanten et al. (1995); Ackermann et al. (1995); Van Zanten (2000); Kohen and Ecrick (2004); Güvenç et al. (2004); Canale et al. (2007)). Common to all solutions is the fact that they are able to generate limited values of the yaw moment. The immediate consequence is that the input variable may saturate and this could deteriorate the control performance. Moreover, good damping properties and vehicle safety (i.e. stability) can be considered as well by imposing suitable constraints on the yaw rate  $\dot{\psi}(t)$  and on the sideslip angle  $\beta(t)$  values as described by Kiencke and Nielsen (2000). Therefore, the presence of such constraints motivates the employment of Nonlinear Model Predictive Control (NMPC) (see e.g. the survey of Mayne et al. (2000)) techniques. However, the realization of NMPC control laws requires the on-line solution of an optimization problem which may not be solvable within the sampling interval required for such kind of application. Nevertheless, predictive control has been successfully employed in vehicle lateral control and vehicle stability control by means of suitable solutions aimed at improving the computational times. In particular, Falcone et al. (2007) used predictive control techniques in active steering control for an autonomous vehicle, where on-line linearization of the vehicle model gave rise to an effective suboptimal solution which allowed the real time implementation. Moreover, Tøndel and Johansen (2005) proposed an interesting contribution to the problem of control allocation in yaw stabilization by means of approximate nonlinear multi-parametric programming, where an approximate solution obtained by means of a piecewise affine function is used for the implementation of the controller. In this paper, the problem of efficient NMPC implementation is solved using an approximated control function, with lower required computational time with respect to on-line optimization, derived using the Fast Model Predictive Control (FMPC) methodology introduced and described by Canale et al. (2009). In this context, the approximating function which realizes the predictive controller is based on the off-line computation of a finite number  $\nu$  of exact NMPC control solutions and guarantees stability as well as constraint satisfaction. In order to show in a realistic way the effectiveness of the proposed control approach,

the approximate NMPC law is implemented on an embedded device with low computational capacity and tested through extensive Software-In-the-Loop (SIL) simulations in demanding driving situations, using a detailed nonlinear 14 degrees of freedom (d.o.f.) vehicle model. Moreover, improvements over a well-assessed approach which employs an Internal Model Control (IMC) structure to handle input constraints are shown too.

The paper is organized as follows. Sections 2 and 3 introduce the control problem and the design procedure of the nominal NMPC law respectively. The adopted NMPC approximation technique and its implementation on the embedded device are described in Sections 4 and 5. Finally, Section 6 describes the SIL tests and conclusions are drawn in Section 7.

## 2 Vehicle modeling and control requirements

Vehicle dynamics can be described using the following nonlinear single track model (see e.g. the book from Rajamani (2005)):

$$\begin{aligned} mv(t)\dot{\beta}(t) + mv(t)\dot{\psi}(t) &= F_{yf}(t) + F_{yr}(t) \\ J_z\ddot{\psi}(t) &= aF_{yf}(t) - bF_{yr}(t) + M_z(t). \end{aligned} \tag{1}$$

In (1)  $m$  is the vehicle mass,  $J_z$  is the moment of inertia around the vertical axis,  $\beta(t)$  is the sideslip angle,  $\psi(t)$  is the yaw angle,  $v(t)$  is the vehicle speed,  $a$  and  $b$  are the distances between the center of gravity and the front and rear axles respectively.  $F_{yf}(t)$  and  $F_{yr}(t)$  are the front and rear tyre lateral forces, which can be expressed as nonlinear functions of  $\beta(t)$ ,  $\dot{\psi}(t)$ ,  $v(t)$  and of the front steering angle  $\delta(t)$  (see e.g. the work of Bakker et al. (1989) for more details):

$$\begin{aligned} F_{yf}(t) &= F_{yf}(\beta(t), \dot{\psi}(t), v(t), \delta(t)) \\ F_{yr}(t) &= F_{yr}(\beta(t), \dot{\psi}(t), v(t), \delta(t)). \end{aligned} \tag{2}$$

Finally,  $M_z(t)$  in (1) is the yaw moment that can be generated by an active device, through appropriate combinations of longitudinal and/or lateral tyre

forces. In this paper, the yaw moment  $M_z(t)$  is supposed to be generated by a Rear Active Differential (RAD). The considered device (see the paper from Canale et al. (2007) for a more detailed description), which is the subject of the patents of Ippolito et al. (1992); Frediani et al. (2002), is basically a traditional bevel gear differential that has been modified in order to transfer motion to two clutch housings, which rotate together with the input gear. Clutch friction discs are fixed on each differential output axle. The ratio between the input angular speed of the differential and the angular speeds of the clutch housings is such that the latter rotate faster than their respective discs in almost every vehicle motion condition (i.e. except for narrow cornering at very low vehicle speed), thus the sign of each clutch torque is always known and the torque magnitude only depends on the clutch actuation force, which is generated by an electro-hydraulic system whose input current  $i(t)$  is determined by the control algorithm. The main advantage of this system with respect to conventional active differentials is the capability of generating yaw moments of every value within the actuation system saturation limits, regardless of the input driving torque value and the speed values of the rear wheels. The yaw moment saturation value is  $\pm 2500$  Nm, due to the physical limits of its electro-hydraulic system. As a first approximation, the actuator behavior can be described by the model:

$$M_z(t) = K_A i(t - \vartheta), \quad (3)$$

where  $K_A$  and  $\vartheta$  are the actuator gain and delay respectively. As a matter of fact, the actuator model considered by Canale et al. (2007) includes also a first-order dynamic which is not considered here, since its bandwidth (about 11 Hz) is higher than the bandwidth of the vehicle considered in this work, which is of about 2.2 Hz. Moreover, the use of a first-order actuator dynamic would add one unmeasured state to the vehicle model: thus, a state observer should be employed in order to compute the NMPC control moves, with consequent possible problems related to observer accuracy and stability, due to the system nonlinearities and the model uncertainty. Furthermore, the use of a larger state dimension would also lead to an increase of the complexity of the approximated NMPC approach employed in this paper since, in general, it

grows exponentially with the state dimension. Anyway, it will be shown that good performance are obtained even by neglecting the first-order actuator dynamics in the control design.

Equations (1), (2) and (3) can be rearranged in the state equation form:

$$\begin{bmatrix} \ddot{\psi}(t) \\ \dot{\beta}(t) \end{bmatrix} = f(\dot{\psi}(t), \beta(t), \delta(t), i(t - \vartheta)). \quad (4)$$

The current  $i(t)$  is the control input, while the steering angle  $\delta(t)$  is assumed to be commanded by the driver via a conventional steering system. It is also assumed that  $\delta(t)$  is measured through a standard steering angle sensor. The improvements of the vehicle handling characteristics can be taken into account in the control design by a suitable choice of a reference signal  $\dot{\psi}_{\text{ref}}(t)$ , generated by means of a nonlinear static map

$$\dot{\psi}_{\text{ref}}(t) = \mathcal{M}(\delta(t), v(t)), \quad (5)$$

which uses as inputs the measured values of the steering angle and of the vehicle speed. Details on the computation of the map  $\mathcal{M}(\cdot)$  can be found in Canale et al. (2007). The tracking of  $\dot{\psi}_{\text{ref}}$  can be taken into account by minimizing the amount of the error  $e(t)$ :

$$e(t) = \dot{\psi}_{\text{ref}}(t) - \dot{\psi}(t).$$

Good damping properties and vehicle safety (i.e. stability) performance can be considered as well by imposing suitable constraints on the state variables, i.e. the yaw rate  $\dot{\psi}(t)$  and the sideslip angle  $\beta(t)$ , as described by Kiencke and Nielsen (2000). However, the value of the yaw moment generated by the employed active device is subject to its physical limits. In particular, the considered active differential has an input current limitation of  $\pm 1$  A which corresponds to the range of allowed yaw moment of  $\pm 2500$  Nm that can be mechanically generated (see the patents of Ippolito et al. (1992) and Frediani et al. (2002) for details). Thus, saturation of the control input (i.e. the actuator current  $i(t)$ ) has to be carefully taken into account in the control design. Therefore, given the presence of state and input constraints, the employment

of NMPC techniques (see e.g. the survey of Mayne et al. (2000)) appears to be an appropriate method to solve the problem. In the next Section, the details of the predictive approach to yaw control are introduced.

### 3 NMPC strategy for yaw control

This Section describes the design of a NMPC law for vehicle yaw control. Although a specific actuator is considered in this paper (i.e. a rear active differential), the described NMPC approach is quite general and can be adapted with little modifications to other kinds of actuation devices. However, as it will be highlighted at the end of this Section, the computational burden of NMPC limits its application with the actual automotive Electronic Control Units (ECU). In order to solve this issue, in this paper an efficient controller implementation is obtained through an approximation of the exact NMPC control law, as it will be described in Section 4. The approximated control law is then implemented on an embedded device with low computational capacity and tested through SIL simulations (as it will be described in Sections 5 and 6).

In the proposed NMPC approach for yaw control, the control move computation is performed at discrete time instants  $kT_s$ ,  $k \in \mathbb{N}$ , defined by the sampling period  $T_s$  and on the basis of the state equations (6) obtained by discretization of (4), e.g. by means of forward difference approximation (for simplicity, the notation  $k + j \triangleq (k + j)T_s$  will be used in the following):

$$\begin{bmatrix} \dot{\psi}_{k+1} \\ \beta_{k+1} \end{bmatrix} = \tilde{f}(\psi_k, \beta_k, \delta_k, i_{k-r}), \quad (6)$$

where  $r$  is defined as:

$$r = \text{int}(\vartheta/T_s) \quad (7)$$

and  $\text{int}(\cdot)$  denotes the nearest integer approximation. Note that if the delay  $\vartheta$  is not an integer multiple of the sampling time  $T_s$ , the use of the nearest integer approximation leads to an error in the delay considered by the NMPC

controller. Such an error is equal to  $T_s/2$  in the worst case and do not lead to significant performance degradation as long as the sampling frequency (i.e. 100 Hz in our case) is sufficiently larger than the bandwidth of the vehicle lateral dynamics (i.e. about 2.2 Hz with the commercial vehicle considered in this paper). In this work, both state variables  $\dot{\psi}$  and  $\beta$  are assumed measurable for control input computation. However, it is well true that, while a measure of  $\dot{\psi}$  can be easily obtained using a gyroscope, a measure of the sideslip angle  $\beta$  is much more difficult and expensive to obtain. On the other hand, quite good and accurate solutions have been proposed in the literature (see, e.g. the works of Van Zanten (2000), Ryu and Gerdes (2004) and Piyabongkarn et al. (2006)), ensuring the reliability of suitable control techniques involving a sideslip angle loop. According to the NMPC strategy, at each sampling time  $k$ , the values of  $\dot{\psi}_k$ ,  $\beta_k$ , of the past input variables  $i_{k-1}, \dots, i_{k-r}$  and of the value of the steering angle  $\delta_k$ , are used to compute the control move, through the solution of the following optimization problem:

$$\min_{[i_{k|k}, \dots, i_{k+N_c-1|k}]} \sum_{j=1}^{N_p} e_{k+j|k}^2 + \sum_{j=0}^{N_p-r} \rho i_{k+j|k}^2 \quad (8a)$$

subject to

$$|i_{k+j|k}| \leq \bar{i} > 0, \quad \forall j \in [0, N_c - 1] \quad (8b)$$

$$i_{k+j|k} = i_{k+N_c-1|k}, \quad \forall j \in [N_c, N_p - 1] \quad (8c)$$

$$|\beta_{k+j|k}| \leq \bar{\beta} > 0, \quad j \in [1, N_p - 1], \quad (8d)$$

where  $N_p \in \mathbb{N}$  is the prediction horizon,  $\rho \in \mathbb{R}^+$  is a weighting factor, that can be tuned to achieve a suitable tradeoff between tracking accuracy and control effort, and  $e_{k+j|k}$  is the  $j^{\text{th}}$  step ahead prediction of the tracking error, obtained as

$$e_{k+j|k} \triangleq \dot{\psi}_{\text{ref},k} - \dot{\psi}_{k+j|k}, \quad j = 1, \dots, N_p.$$

The value of  $\dot{\psi}_{\text{ref},k}$  is computed using the actual values of  $\delta_k$  and  $v_k$ , through the static map  $\mathcal{M}$  (5). The predicted yaw rate  $\dot{\psi}_{k+j|k}$  is obtained via the state equation (6), starting from the “initial conditions”  $\dot{\psi}_{k|k} = \dot{\psi}_k$  and  $\beta_{k|k} = \beta_k$

and using the following sequences of steering angle  $\delta$  and of the input  $i$ :

$$\begin{bmatrix} \delta_{k|k} = \delta_{k+1|k} = \dots = \delta_{k+N_p-1|k} = \delta_k \\ \dot{i}_{k-r}, \dots, \dot{i}_{k-1}, \dot{i}_{k|k}, \dots, \dot{i}_{k+N_c-1|k}, \dots, \dot{i}_{k+N_p-1|k} \end{bmatrix},$$

where  $N_c \leq N_p$  is the control horizon and the assumption  $i_{k+j|k} = i_{k+N_c-1|k}$ ,  $\forall j \in [N_c, N_p - 1]$  is made, as highlighted in (8c). Note that the optimization of the index (8a) is performed with respect to the variables  $[i_{k|k}, \dots, i_{k+N_c-1|k}]$ , while the value of the steering angle  $\delta$  is kept constant at the value  $\delta_{k|k} = \delta_k$ , measured at time  $k$ , during the whole prediction horizon. Moreover, the initial state  $\dot{\psi}_k$  can be expressed as:

$$\dot{\psi}_k = \dot{\psi}_{\text{ref},k} - e_k.$$

Therefore, since  $\psi_{\text{ref},k}$  is a function of  $\delta_k$  and  $v_k$ , the performance index (8a) depends on the vector  $w_k \in \mathbb{R}^{4+r}$  of the measured variables:

$$w_k \triangleq [e_k, \beta_k, \delta_k, v_k, i_{k-r}, \dots, i_{k-1}]^T. \quad (9)$$

Then, the predictive control law is computed using a receding horizon strategy:

1. At time instant  $k$ , get  $w_k$ .
2. Solve the optimization problem (8)
3. Apply the first element of the optimal solution sequence  $[i_{k|k}^*, \dots, i_{k+N_c-1|k}^*]$  as the actual control action  $i_k = i_{k|k}^*$ .
4. Repeat the whole procedure at the next sampling time  $k + 1$ .

Note that no constraints have been imposed on  $\dot{\psi}$ , as its limitation on the basis of comfort and directional stability criteria similar to the ones introduced by Kiencke and Nielsen (2000) have been implicitly taken into account in the computation of  $\dot{\psi}_{\text{ref}}$  (for details see the paper from Canale et al. (2007)). On the other hand, constraints (8d) on  $\beta$  and (8b) on  $i$  account for vehicle directional stability and actuator saturation respectively. Finally, it has to be noted that  $N_p$  has to be greater than the number of delay steps  $r$ , so that the first predicted input  $i_{k|k}$  can influence the cost function (8a).

The obtained predictive controller is a nonlinear static function  $\kappa$  of the vari-

able  $w_k$  defined in (9):

$$i_k = \kappa(w_k). \quad (10)$$

For a given value of  $w_k$ , the value of the function  $\kappa(w_k)$  is implicitly computed by solving at each sampling time  $k$  the constrained optimization problem (8). Control law (10) results to be quite effective, as shown in the example of Fig. 1, that shows the course of the controlled vehicle yaw rate during a step steer maneuver, simulated using a detailed nonlinear 14 degrees-of-freedom (d.o.f.) model, together with the one obtained by the uncontrolled vehicle. It can be

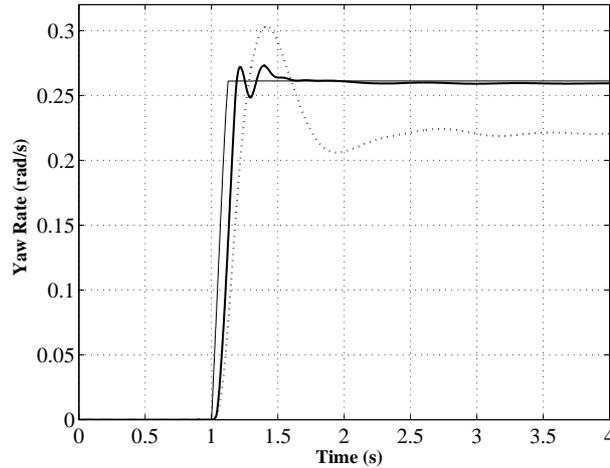


Fig. 1. Courses of the reference yaw rate (thin solid line) and of the yaw rate obtained by the controlled vehicle (solid) and by the uncontrolled one (dotted) during a  $50^\circ$  step-steer maneuver at 100 km/h.

noted that a faster response and better damping are achieved by the controlled vehicle. The test shown in Fig. 1 has not been performed in real-time, since no strict sampling time was imposed and the numerical solver employed for the simulation (i.e. Simulink<sup>®</sup> with ode45 algorithm) “awaited” for the completion of the numerical optimization (8) at each sampling time. As a matter of fact, the on-line solution of the optimization problem (8) can not be solved at the sampling period required for this application, which is equal to 10 ms, using the actual commercial ECUs. In fact, the on-line computational time required to solve (8), using a sequential quadratic programming algorithm (see e.g. the book of Nocedal and Wright (2006)), is about 30 ms on a standard PC (i.e. an Intel<sup>®</sup> Core<sup>™</sup>2 at 2.4 GHz with 2 GB RAM) and about 15 s on the embedded device considered in this paper (described in Section 5). In order to overcome

this problem and to allow the real-time implementation of the NMPC strategy, in this paper an approximated controller  $\hat{\kappa} \approx \kappa$  is derived, suitable for on-line implementation. Function  $\hat{\kappa}$  is obtained using a finite number  $\nu$  of values of  $\kappa(w_k)$ , computed off-line and stored. Such an approach is presented and discussed in the next Section.

## 4 Fast MPC (FMPC) implementation

The problem of efficient NMPC implementation is tackled in this paper using an approximated control function, with lower required computational time, derived by means of Set Membership (SM) techniques as described by Canale et al. (2009). In particular, under this context, the “Nearest Point” (NP) approach will be employed. It has to be remarked that the use of the NP technique for function approximation is well-assessed in different applications (see e.g. the book from Fukunaga (1990)). The novelty of the approach as it is presented by Canale et al. (2009) lies in the accuracy analysis, carried out in the framework of SM approximation theory, and in the consequent analysis of the closed loop properties.

### 4.1 Prior information

The approximating function  $\kappa^{\text{NP}}$  is computed over a compact subset  $\mathcal{W} \subset \mathbb{R}^{4+r}$  of the domain of the exact function  $\kappa$ . Note that, due to the input constraints (8b), the image set of function  $\kappa(w)$  is  $[-\bar{i}, \bar{i}]$ . Moreover, it is assumed that the function  $\kappa$  is continuous in  $\mathcal{W}$ . Such property depends on the characteristics of the optimization problem (8): results on this aspect can be found e.g. in the works of Mayne and Michalska (1990), Meadows (1994) and Spjøtvold et al. (2007) and the references therein. Continuity of the nominal control law can be also evaluated a posteriori via numerical analyses. Note that stronger regularity assumptions (e.g. differentiability) cannot be made, since even in the simple case of linear dynamics, linear constraints and quadratic cost function,  $\kappa$  is a piece-wise affine continuous function (see e.g. the papers from Bemporad

et al. (2002) and Seron et al. (2003)). Inside  $\mathcal{W}$ , a finite number  $\nu$  of points  $\tilde{w}^h, h = 1, \dots, \nu < \infty$  is suitably chosen, giving rise to the set:

$$\mathcal{W}_\nu = \{\tilde{w}^h \in \mathcal{W}, h = 1, \dots, \nu\}. \quad (11)$$

For each value of  $\tilde{w} \in \mathcal{W}_\nu$ , the corresponding value  $\tilde{i} = \kappa(\tilde{w})$  is computed by solving off-line the optimization problem (8), so that:

$$\tilde{i} = \kappa(\tilde{w}), \quad \forall \tilde{w} \in \mathcal{W}_\nu. \quad (12)$$

Such values of  $\tilde{w}, \tilde{i}$  are stored to be used for the on-line computation of  $\kappa^{\text{NP}}$ . The set  $\mathcal{W}_\nu$  is supposed to be chosen such that the following property holds:

$$\lim_{\nu \rightarrow \infty} d_H(\mathcal{W}, \mathcal{W}_\nu) = 0, \quad (13)$$

where  $d_H(\mathcal{W}, \mathcal{W}_\nu)$  is defined as:

$$d_H(\mathcal{W}, \mathcal{W}_\nu) = \sup_{w \in \mathcal{W}} \inf_{\tilde{w} \in \mathcal{W}_\nu} (\|w - \tilde{w}\|_2). \quad (14)$$

Condition (13) ensures that as  $\nu \rightarrow \infty$  the set  $\mathcal{W}$  is densely covered. Since both  $\mathcal{W}$  and the set  $[-\bar{i}, \bar{i}]$  are compact, it follows that function  $\kappa$  is Lipschitz continuous:

$$\|\kappa(w^1) - \kappa(w^2)\|_2 \leq \gamma \|w^1 - w^2\|_2, \quad \forall w^1, w^2 \in \mathcal{W}. \quad (15)$$

All this prior information can be summarized by concluding that  $\kappa$  belongs to the Feasible Function Set (*FFS*) defined as:

$$\kappa \in FFS \doteq \{\kappa \in \mathcal{A}_\gamma : \kappa(\tilde{w}) = \tilde{i}, \quad \forall \tilde{w} \in \mathcal{W}_\nu\}, \quad (16)$$

where  $\mathcal{A}_\gamma$  is the set of all continuous functions  $\kappa : \mathcal{W} \rightarrow \mathcal{I}$ , such that (15) holds.

## 4.2 Nearest Point approximation

The approximating function  $\kappa^{\text{NP}}$  is computed as follows. For any  $w \in \mathcal{W}$ , denote with  $\tilde{w}^{\text{NP}}$  a value such that:

$$\tilde{w}^{\text{NP}} = \arg \min_{\tilde{w} \in \mathcal{W}_\nu} \|\tilde{w} - w\|_2. \quad (17)$$

Then, the NP approximation  $\kappa^{\text{NP}}(x)$  is defined as:

$$\kappa^{\text{NP}}(w) = \kappa(\tilde{w}^{\text{NP}}). \quad (18)$$

As shown in the work of Canale et al. (2009), such approximation has the following properties:

**i)** the input constraints are always satisfied:

$$\kappa^{\text{NP}}(w) \in \mathcal{I}, \quad \forall w \in \mathcal{W}; \quad (19)$$

**ii)** for a given  $\nu$ , a bound  $\zeta^{\text{NP}}(\nu)$  on the pointwise approximation error can be computed:

$$\|\kappa(w) - \kappa^{\text{NP}}(w)\|_2 \leq \zeta^{\text{NP}} = \gamma d_H(\mathcal{W}, \mathcal{W}_\nu), \quad \forall w \in \mathcal{W}; \quad (20)$$

**iii)**  $\zeta^{\text{NP}}(\nu)$  is convergent to zero:

$$\lim_{\nu \rightarrow \infty} \zeta^{\text{NP}} = 0. \quad (21)$$

It can be proved that, if properties (19)-(21) hold, then there exists a finite value of  $\nu$  such that closed loop stability can be guaranteed also using the approximated controller. In particular, the considered stability properties are the boundedness of the system trajectories, their convergence to an arbitrarily small neighborhood of the origin (or, more in general, of a given set-point) and an arbitrarily small distance with respect to the closed loop trajectories obtained by the exact control law. Unfortunately, at present there are no techniques to find out a priori the number and the values of the vector  $\tilde{w}$  to be considered in the off-line computations in order to guarantee given closed-loop performance (e.g. an upper bound on the steady-state tracking error norm). Such a drawback is shared by all of the present approaches to approximate

NMPC. However, for a given value of  $\nu$  it is possible to estimate a posteriori the performance of the approximated control law. Thus, in the design of the control law  $\kappa^{\text{NP}}$  an iterative procedure can be employed, as described by Canale et al. (2009), where the value of  $\nu$  is gradually increased until the required performance is achieved. As to the Lipschitz constant  $\gamma$ , which is needed to compute the approximation error bound  $\zeta^{\text{NP}}$ , an estimate  $\hat{\gamma}$  can be derived as:

$$\hat{\gamma} = \inf \left( \tilde{\gamma} : \tilde{i}^h + \tilde{\gamma} \|\tilde{w}^h - \tilde{w}^k\|_2 \geq \tilde{i}^k, \forall k, h = 1, \dots, \nu \right). \quad (22)$$

Such estimate guarantees that  $FFS \neq \emptyset$  (since it follows as an extension of Theorem 1 in the paper from Milanese and Novara (2004)). In the work of Canale et al. (2009) it has been shown that:

$$\lim_{\nu \rightarrow \infty} \hat{\gamma} = \gamma. \quad (23)$$

Note that, for a given value of  $\nu$ , the estimate  $\hat{\gamma}$  can be far away from  $\gamma$ . However, usually in practice the estimate (22) converges in a fast and reliable way by iteratively increasing  $\nu$ .

### 4.3 Design procedure

The overall design procedure for the fast NMPC approach proposed in this paper can be resumed as follows:

1. Design the nominal NMPC control law according to (8), tuning the parameters  $\rho$ ,  $N_p$ ,  $N_c$  through simulation tests.
2. Choose the set  $\mathcal{W}$  where the FMPC control law is defined and collect the values  $\tilde{w}^j$ ,  $\tilde{i}^j$ ,  $j = 1, \dots, \nu$  (12) such that (13) holds, e.g. by performing simulations of suitably chosen maneuvers using the closed loop model with the nominal NMPC controller.
3. Implement on-line the NP approximated control law using (17) and (18).
4. If needed, tune the number  $\nu$  of off-line computed values in order to find a satisfactory tradeoff between computational time, memory usage and performance.

## 5 Controller implementation on a low performance embedded processor

### 5.1 Hardware description and SIL test setup

In order to show the effectiveness of the FMPC approach described in the previous Section, the control law is implemented on a commercial embedded device and it is tested through SIL simulations. The experiments have been performed at the laboratory of Politecnico di Torino, where a Rabbit BL2600<sup>®</sup> device is available (for detailed characteristics see the data-sheet Rabbit Semiconductor (2006)). Such an embedded platform is provided with a CISC 8-bit processor working at 44 MHz, with reduced computational capacity with respect to other processors used in automotive ECUs. In fact, differently from other micro-controllers employed in automotive applications, such as the Siemens 80C166<sup>®</sup> (see Siemens (2005)) and the Motorola 68336<sup>®</sup> (see Freescale Semiconductor (2000)), the Rabbit is not provided with a floating point execution unit and its processing time is four times higher on average. The BL2600<sup>®</sup> is equipped with a memory reader slot that supports up to 128 MB. Such a memory can be employed to save the off-line sampled values needed to implement the designed NP approximation.

The SIL test setup is reported in the picture of Fig. 2. A detailed 14 d.o.f.

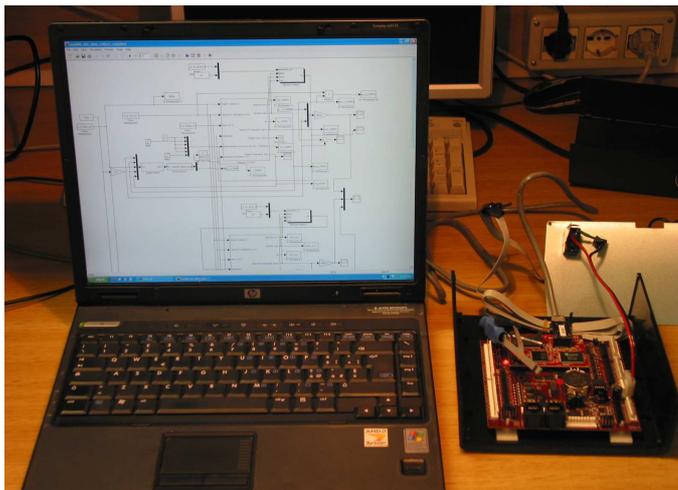


Fig. 2. Software-in-the-loop test setup.

vehicle model (described in Section 6), running under MatLab<sup>®</sup> Simulink<sup>®</sup> on a Personal Computer (PC), simulates the real vehicle. The Rabbit BL2600<sup>®</sup> hosts the controller. The communication between the embedded device and the PC is performed with a 100 Mb Ethernet interface and the TCP/IP protocol. Although the Ethernet transmission data rate is higher than the one of the Car Area Network (CAN), usually employed on vehicles, the amount of the data to be transmitted, given simply by the vector  $w_k$  and by the control input  $\kappa^{\text{NP}}(w_k)$ , is such that also on a CAN bus the transmission time would be negligible with respect to the control computation time.

The aim of such an experimentation is twofold. First of all, it allows to show that an advanced control technique like NMPC, usually limited to “slow” processes and/or processors with high computational capacity, can be practically employed with fast sampling rate, using a commercial device with poor computational performance. Moreover, with the chosen device it can be shown that with the SM approximation techniques a suitable compromise can be set up, between computational efficiency and memory usage, in order to adapt the control law implementation to the hardware characteristics.

## 5.2 Control law implementation

In order to implement the approximated NMPC law  $i = \kappa^{\text{NP}}(w)$  on the Rabbit BL2600<sup>®</sup> device, the values of  $\tilde{w}$ ,  $\tilde{i}$  in (12) have been computed using uniform gridding over the set  $\mathcal{W}$ . The latter has been chosen through an extensive number of simulations tests, using the nominal NMPC controller and a detailed vehicle model, aimed at finding the operational limits of each component of the vector  $w$ . Thus, the chosen compact set  $\mathcal{W}$  can be indicated as:

$$\mathcal{W} \doteq \{w : w^l \preceq w \preceq w^u\} \subset \mathbb{R}^{4+r}, \quad (24)$$

where the symbol  $\preceq$  indicates element-wise inequalities and  $w^l$  and  $w^u$  represent, respectively, the chosen lower and upper bound of  $w$ . Then, for each component  $w_\ell$ ,  $\ell = 1, \dots, 4 + r$  of  $w$ , the corresponding grid step value  $\Delta w_\ell$  has been chosen in order to achieve a suitable tradeoff between accuracy and memory usage. On the basis of the grid step  $\Delta w_\ell$ , the number of samples of

each component  $w_\ell$  within the interval  $[w_\ell^l, w_\ell^u]$  is given by:

$$n_\ell = \left( \text{int} \left( \frac{w_\ell^u - w_\ell^l}{\Delta w_\ell} \right) + 1 \right), \quad (25)$$

where  $\text{int}(\cdot)$  is the nearest integer approximation. Therefore, in the case of uniform gridding of  $\mathcal{W}$ , the size  $\nu_{\text{UG}}$  of the set (11) is:

$$\nu_{\text{UG}} = \prod_{\ell=1}^{4+r} n_\ell. \quad (26)$$

Once the elements  $\tilde{w}^h$ ,  $h = 1, \dots, \nu_{\text{UG}}$  of the set (11) have been defined, the corresponding values  $\tilde{i}^h$  (see (12)) are computed. The data obtained in such a way can be sorted in a unique matrix  $W_{\text{UG}}$ , as illustrated in the example reported below, referred to the case of  $i = \kappa(w)$ ,  $w = [w_1, w_2] \in \mathbb{R}^2$  with  $n_1 = 3$  and  $n_2 = 4$  (i.e.  $\nu_{\text{UG}} = 12$ ):

$$W_{\text{UG}} = \begin{bmatrix} \tilde{w}_1^1 & \tilde{w}_2^1 & \tilde{i}^1 = \kappa(\tilde{w}_1^1, \tilde{w}_2^1) \\ \tilde{w}_1^1 & \tilde{w}_2^2 & \tilde{i}^2 = \kappa(\tilde{w}_1^1, \tilde{w}_2^2) \\ \tilde{w}_1^1 & \tilde{w}_2^3 & \tilde{i}^3 = \kappa(\tilde{w}_1^1, \tilde{w}_2^3) \\ \tilde{w}_1^1 & \tilde{w}_2^4 & \tilde{i}^4 = \kappa(\tilde{w}_1^1, \tilde{w}_2^4) \\ \tilde{w}_1^2 & \tilde{w}_2^1 & \tilde{i}^5 = \kappa(\tilde{w}_1^2, \tilde{w}_2^1) \\ \tilde{w}_1^2 & \tilde{w}_2^2 & \tilde{i}^6 = \kappa(\tilde{w}_1^2, \tilde{w}_2^2) \\ \tilde{w}_1^2 & \tilde{w}_2^3 & \tilde{i}^7 = \kappa(\tilde{w}_1^2, \tilde{w}_2^3) \\ \tilde{w}_1^2 & \tilde{w}_2^4 & \tilde{i}^8 = \kappa(\tilde{w}_1^2, \tilde{w}_2^4) \\ \tilde{w}_1^3 & \tilde{w}_2^1 & \tilde{i}^9 = \kappa(\tilde{w}_1^3, \tilde{w}_2^1) \\ \tilde{w}_1^3 & \tilde{w}_2^2 & \tilde{i}^{10} = \kappa(\tilde{w}_1^3, \tilde{w}_2^2) \\ \tilde{w}_1^3 & \tilde{w}_2^3 & \tilde{i}^{11} = \kappa(\tilde{w}_1^3, \tilde{w}_2^3) \\ \tilde{w}_1^3 & \tilde{w}_2^4 & \tilde{i}^{12} = \kappa(\tilde{w}_1^3, \tilde{w}_2^4) \end{bmatrix}.$$

Such an ordering of the data is useful to efficiently find out, for a given value  $w \in \mathcal{W}$ , the corresponding nearest point  $\tilde{w}_{\text{UG}}^{\text{NP}} \in \mathcal{W}_\nu$ . In fact,  $\tilde{w}_{\text{UG}}^{\text{NP}}$  can be

computed as:

$$\tilde{w}_{UG}^{\text{NP}} = \begin{bmatrix} \Delta w_1 \text{int} \left( \frac{w_1 - w_1^l}{\Delta w_1} \right) + w_1^l \\ \vdots \\ \Delta w_{4+r} \text{int} \left( \frac{w_{4+r} - w_{4+r}^l}{\Delta w_{4+r}} \right) + w_{4+r}^l. \end{bmatrix}$$

Moreover, the index  $n^{\text{NP}}$  of the row containing the nearest point  $\tilde{w}_{UG}^{\text{NP}}$  inside the matrix  $W_{UG}$  can be computed using the following formula:

$$n^{\text{NP}} = \sum_{\ell=1}^{4+r} \left( \text{int} \left( \frac{w_\ell - w_\ell^l}{\Delta w_\ell} \right) M_\ell \right) + 1, \quad (27)$$

where

$$M_\ell = \prod_{q=\ell+1}^{4+r} n_q \quad (28)$$

$$M_{4+r} = 1.$$

Thus, the control input corresponding to a given value  $w$  is  $\kappa^{\text{NP}}(w) = \tilde{i}^{n^{\text{NP}}}$ . At the generic step  $k$ , the computation of the control input  $i_k = \kappa^{\text{NP}}(w_k)$  can be then performed using the following algorithm:

1. Acquire  $w_k$
2. Compute the index  $n^{\text{NP}}$  (29)
3. Apply  $i_k = \tilde{i}^{n^{\text{NP}}}$ .

Note that the values of  $\tilde{w}$  are not needed in the computation of  $n^{\text{NP}}$ . Thus, it is possible to reduce the memory usage by storing only the minimal needed information, i.e. the values  $\tilde{i}^j, j = 1, \dots, \nu_{UG}, w^l$  (24) and  $M_\ell, \ell = 1, \dots, 4+r$  (28).

## 6 SIL test results

### 6.1 Design of the nominal and approximated NMPC laws

The nominal predictive controller  $\kappa$  has been designed using model (1)-(3) with the nominal parameter values indicated in Table 1. In order to be used

Table 1  
Vehicle model parameters

| Variable name | Description  | Value                 |
|---------------|--|-----------------------|
| $m$           | Vehicle mass   | 1715 kg               |
| $J_z$         | Moment of inertia  | 2700 kgm <sup>2</sup> |
| $a$           | Distance from the center of the gravity and the front axle | 1.07 m                |
| $b$           | Distance from the center of the gravity and the rear axle  | 1.47 m                |
| $\vartheta$   | Actuator delay   | 20 ms                 |
| $K_A$         | Actuator gain  | 2500 Nm/A             |

for the predictions in the optimization algorithm, the vehicle model has been discretized using forward difference approximation, with sampling time  $T_s = 10$  ms. Therefore, since the nominal actuator delay value is  $\vartheta = 20\text{ms} = 2T_s$ , according to (7)  $r = 2$ , i.e. at the generic time step  $k$  the past input values  $i_{k-1}, i_{k-2}$  have to be used to compute the predicted vehicle behavior. The weight  $\rho$  in cost function (8a) has been chosen as  $\rho = 10^{-6}$  and the employed state and input constraints are  $\bar{\beta} = 5^\circ$  and  $\bar{i} = 1$  A. The chosen prediction and control horizons are  $N_p = 10$  and  $N_c = 5$  respectively. The nominal off-line control move computation has been performed using a sequential constrained Gauss-Newton quadratic programming algorithm (see e.g. the book from Nocedal and Wright (2006)), where the underlying quadratic programs have been solved using the MatLab<sup>®</sup> optimization function `quadprog`. Note that such a method is a local solver and it might not provide the global minimum in the case the optimization problem (8) is not convex. However, in this particular case, the numerical optimization has been solved in an efficient and reliable

way for all of the considered values of  $w$  and the change in the optimal solution due to a “small” perturbation of  $w$  was always small, leading us to infer that the nonlinear program (8) was convex and that the global solution has been therefore computed (up to the numerical precision of MatLab®) for every considered value of  $w$ . As it has been described in Section 3, the nominal control law at sampling time  $k$  results to be a static function of the variable  $w_k = [e_k, \beta_k, \delta_k, v_k, i_{k-1}, i_{k-2}]^T \in \mathbb{R}^6$ . The set  $\mathcal{W}$  (24) has been computed by performing simulations involving an extensive set of handwheel steps and sinusoids maneuvers. In order to improve the regulation performance and to “optimize” memory occupation, two different sets  $\mathcal{W}_\nu$  (11), namely  $\mathcal{W}_\nu^1$  and  $\mathcal{W}_\nu^2$ , have been considered inside the set  $\mathcal{W}$ . In particular, the set  $\mathcal{W}_\nu^1$ , characterized by larger gridding intervals, is used when the tracking error is sufficiently far from zero; on the other hand the set  $\mathcal{W}_\nu^2$ , with smaller gridding intervals, is employed when the tracking error approaches the zero value. The sets  $\mathcal{W}_\nu^1$  and  $\mathcal{W}_\nu^2$  are defined as:

$$\mathcal{W}_\nu^1 = \underbrace{\begin{bmatrix} -0.43, & -0.08, & -0.1, & 22, & -1, & -1 \end{bmatrix}^T}_{w^{l,1}} \preceq w \preceq \underbrace{\begin{bmatrix} 0.43, & 0.08, & 0.1, & 33, & 1, & 1 \end{bmatrix}^T}_{w^{u,1}},$$

$$\mathcal{W}_\nu^2 = \underbrace{\begin{bmatrix} -0.03, & -0.08, & -0.1, & 22, & -1, & -1 \end{bmatrix}^T}_{w^{l,2}} \preceq w \preceq \underbrace{\begin{bmatrix} 0.03, & 0.08, & 0.1, & 33, & 1, & 1 \end{bmatrix}^T}_{w^{u,2}},$$

where the superscript symbol  $T$  stands for transpose operation. The corresponding gridding intervals are:

$$\Delta w^1 = \begin{bmatrix} 0.08, & 0.04, & 0.01, & 5.55, & 0.5, & 0.5 \end{bmatrix}^T,$$

$$\Delta w^2 = \begin{bmatrix} 0.005, & 0.0175, & 0.001, & 2.77, & 0.5, & 0.5 \end{bmatrix}^T.$$

Note that the sets  $\mathcal{W}_\nu^1$  and  $\mathcal{W}_\nu^2$  differ by the considered range of values of the first component of  $w$  (i.e. the tracking error  $e_k$ ) and by the employed gridding intervals. In particular, the set  $\mathcal{W}_\nu^2$ , characterized by a smaller tracking error range and by a finer gridding of all of the components of  $w$ , is more suitable to be employed when the tracking error is close to zero. In order to allow the NP algorithm to use the most appropriate set between  $\mathcal{W}_\nu^1$  and  $\mathcal{W}_\nu^2$ , the following

variable is computed at each sampling time:

$$c_k = |e_k| - 0.03.$$

If  $c_k \geq 0$ , the NP algorithm (29) is applied to the set  $\mathcal{W}_\nu^1$ , otherwise the set  $\mathcal{W}_\nu^2$  is used.

It has to be noted that more sophisticated gridding choices could have been employed (e.g. adaptive gridding) in order to achieve higher accuracy (with respect to uniform gridding) for a given value of  $\nu$ . However, uniform gridding is much simpler to implement and it is a feasible approach on processors with low computational power, like the one employed in this paper. This aspect also motivates the choice of the NP approach over more complex methods (e.g. piece-wise affine approximations like the one considered by Johansen (2004), or the optimal SM approach described by Canale et al. (2009)), that are able to achieve higher accuracy with lower memory usage, at the cost of increased on-line computational time. More in general, a tradeoff between accuracy, memory usage and on-line computational time has to be achieved when the approximated NMPC law is derived, also taking into account the features of the employed hardware. In this work, the Rabbit<sup>®</sup> processor has very poor computational power and quite high memory capacity: thus, we chose a very simple but computationally efficient uniform gridding approximation, with two different sets of data in order to achieve a better accuracy when the tracking error is small.

With the above-described choice of the off-line computed control moves, a total value of  $\nu_{\text{UG}} = \nu_{\text{UG}}^1 + \nu_{\text{UG}}^2 \simeq 2.2 \cdot 10^6$  is obtained. The corresponding memory occupation is 2.2 MB: such a memory requirement is in agreement with the size of flash memory available on recent ECUs for automotive applications, like the Fujitsu MB91F467BA<sup>®</sup> micro-controller or the Freescale Power Architecture<sup>®</sup> Mobile Control Units. Moreover, on-board flash memory on vehicles grows by a factor of about 10 every 3-4 years (see e.g. the work of Damm (2006)), meaning that in the next future there will be sufficient memory to host the required data.

## 6.2 SIL simulation settings

SIL simulations according to the setup described in Section 5 have been performed using a detailed nonlinear 14 d.o.f. Simulink<sup>®</sup> model, which gives an accurate description of the vehicle dynamics as compared to actual measurements and includes nonlinear suspension, steer and tyre characteristics, obtained on the basis of measurements on the real vehicle. The model degrees of

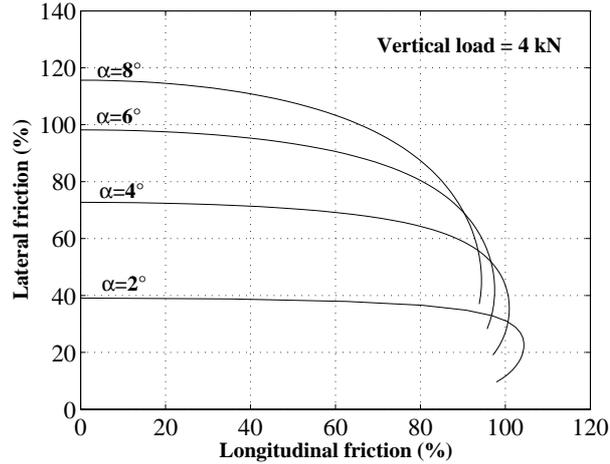


Fig. 3. Front tyre friction ellipses considered in the 14 degrees of freedom model, with different values of lateral slip angle  $\alpha$ , for a constant vertical load of 4 kN.

freedom correspond to the standard three chassis translations and yaw, pitch and roll angles, the four wheel angular speeds and the four wheel vertical movements with respect to the chassis. Nonlinear characteristics obtained on the basis of measurements on the real vehicle have been employed to model the tyre, steer and suspension behavior. The employed tyre model, described e.g. in the book from Genta (2003), takes into account the interaction between longitudinal and lateral slip, as well as vertical tyre load and suspension motion, to compute the tyre longitudinal and lateral forces and self-aligning moments. An example of the employed tyre friction ellipses is shown in Fig. 3, where the lateral friction coefficient is reported as a function of the exploited longitudinal friction (during traction) and of the tyre slip angle  $\alpha$ . Unsymmetrical friction ellipses for traction/braking longitudinal forces is also considered. The comparisons between yaw rate and lateral acceleration measured on the real vehicle, during a track test, and the ones obtained in simulation with the con-

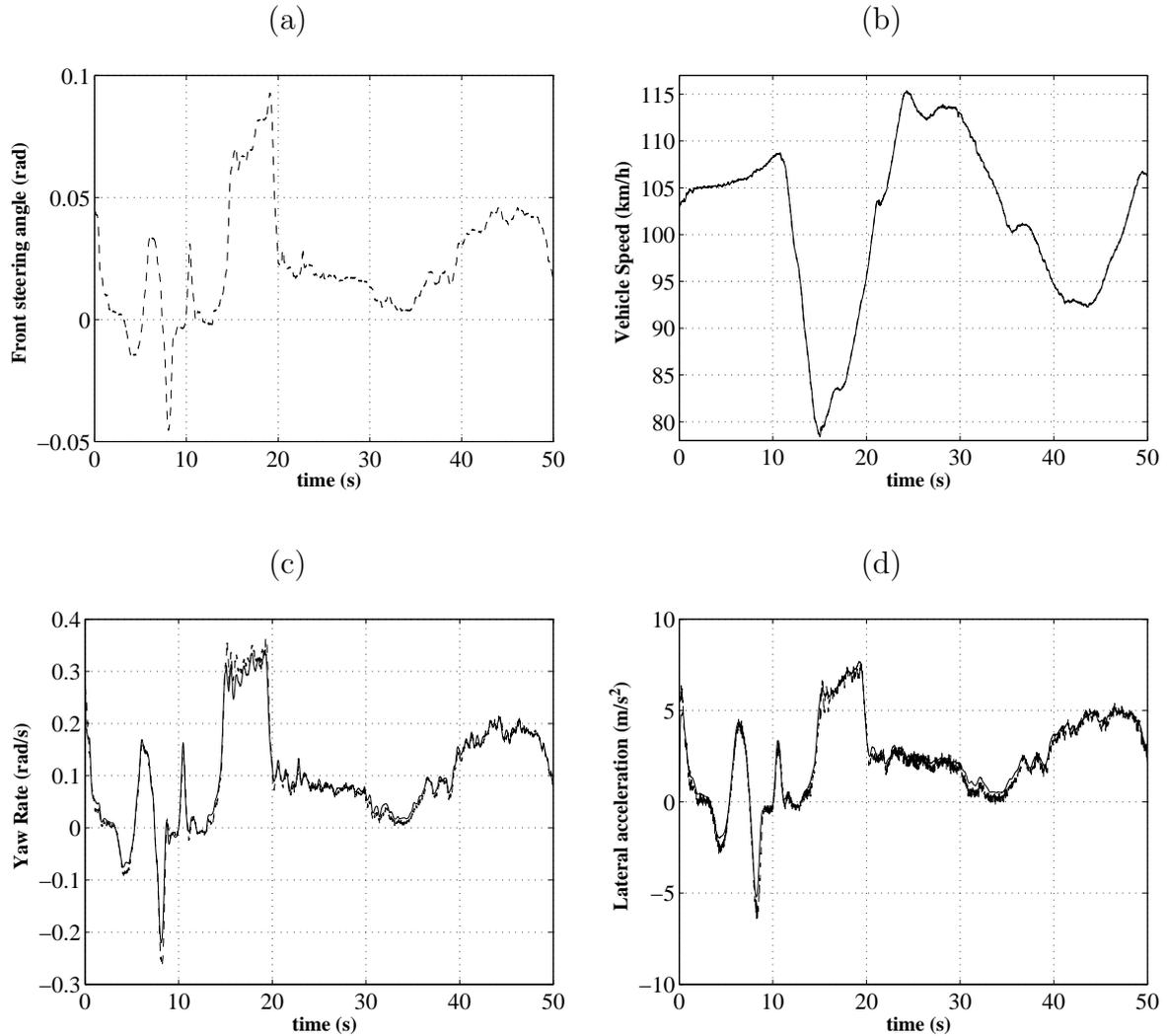


Fig. 4. Comparison between real data (dashed) and simulation results obtained with the 14 d.o.f. model (solid) during a track test. (a) Front steering angle  $\delta$  (used as input for the 14 d.o.f. model), (b) vehicle speed  $v$ , (c) yaw rate  $\dot{\psi}$ , (d) lateral acceleration  $a_y$ .

sidered model, reported in Fig. 4(a)-(d), show the good accuracy properties of the employed 14 d.o.f. model.

The following open loop maneuvers (i.e. without driver's feedback) have been chosen to test the control effectiveness:

- steer reversal test with handwheel angle of  $50^\circ$  performed at 100 km/h, with a steering wheel speed of  $400^\circ/\text{s}$ . This test aims to evaluate the controlled

car transient and steady state performances: the employed handwheel course is shown in Fig. 5;

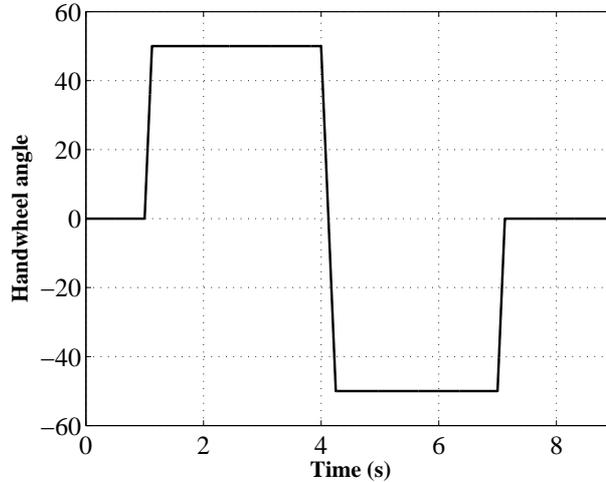


Fig. 5. Handwheel angle input (deg) for the steer reversal test.

- $\mu$ -split braking maneuver performed at 100 km/h with dry road on one side and icy road on the other, with braking pedal input corresponding to a deceleration value of 0.5 g on dry road. The objective of this test is to evaluate the system response in the presence of strong disturbances. Note that the  $\mu$ -split maneuver implies a differential left-right change in the tyre-road friction coefficients, which was not taken into account in the control design, since the maneuvers considered in the off-line computation of the control moves were performed with a single track model;
- steering wheel frequency sweep performed at 90 km/h in the frequency range 0-7 Hz with steering wheel angle amplitude of 30°;
- constant speed steering pad at 100 km/h: the handwheel angle is increased slowly (1°/s) to evaluate the steady state tracking behavior.

The performance obtained with the approximated predictive controller has been compared with those of the uncontrolled vehicle, of the vehicle controlled with the nominal NMPC law and of the vehicle controlled using an advanced IMC structure, which proved its good effectiveness in the same control problem (see the paper from Canale et al. (2007)). Only the approximated controller has been implemented on the Rabbit<sup>®</sup> processor and tested with strict sampling time, while the simulations with the exact NMPC law have been computed on

a standard PC without strict sampling time, in order to allow the completion of the numerical optimization procedure at each sampling interval.

### 6.3 SIL simulation results

The results of the  $50^\circ$  steer reversal test are reported in Fig. 6-8. In Fig. 6 it can be noted that the yaw rate courses obtained with the approximated NMPC controller (solid line) and the nominal one (dash-dotted) are practically superimposed, with only a slight difference due to the presence of moderate oscillations caused by the control law approximation. Such yaw rate oscillations are too small to be perceived by the driver, so the driving comfort is preserved. The IMC structure achieves slightly worse performance in the second part of the maneuver (at about 4.4s), with a higher overshoot. The

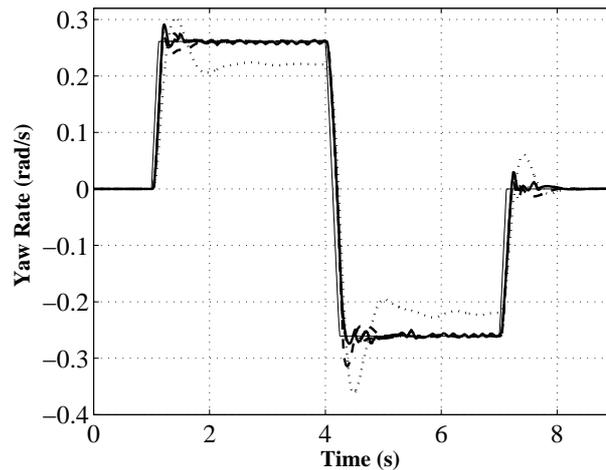


Fig. 6.  $50^\circ$  steer reversal test at 100 km/h. Comparison between the reference (thin solid line) vehicle yaw rate course and those obtained with the uncontrolled vehicle (dotted), the nominal NMPC (dash-dotted) and FMPC (solid) controlled vehicles and with the IMC structure (dashed).

steady state yaw rate reference is reached and, according to the reference map (see Canale et al. (2007)), it is higher than the uncontrolled vehicle yaw rate, thus improving the car maneuverability. The obtained sideslip angle  $\beta(t)$  (Fig. 7) is kept inside the considered constraint, with a maximum absolute value of  $2.8^\circ$ . Constraints on the input variable  $i$  are satisfied too (see Fig. 8, solid

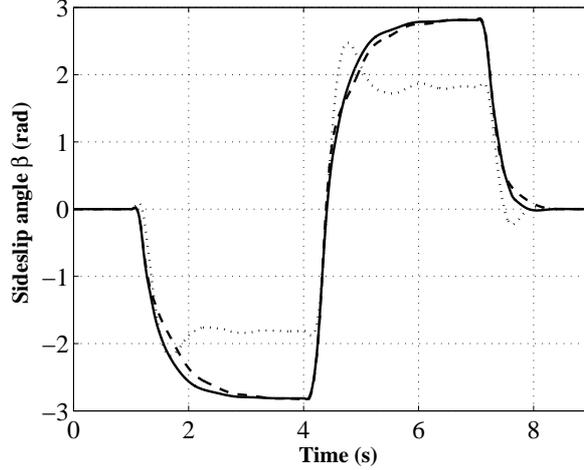


Fig. 7.  $50^\circ$  steer reversal test at 100 km/h. Comparison between the vehicle sideslip angle obtained with the uncontrolled vehicle (dotted) and those obtained the nominal NMPC (dash-dotted) and FMPC (solid) controlled vehicles and with the IMC structure (dashed).

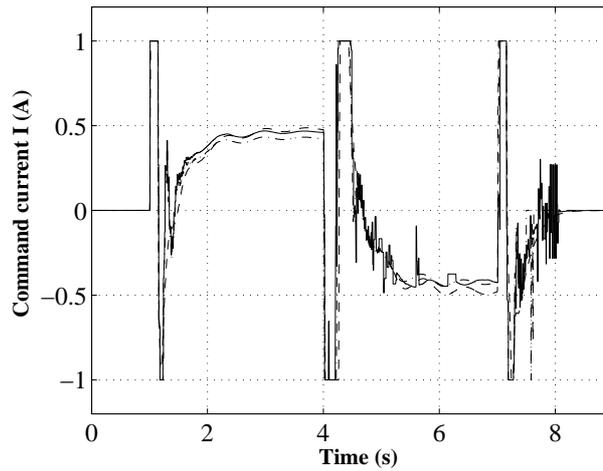


Fig. 8.  $50^\circ$  steer reversal test at 100 km/h. Comparison between the input variable  $u$  obtained with the nominal NMPC (dash-dotted), the FMPC (solid) and the IMC (dashed) controllers.

line). Note that some chattering of the input variable occurs with the FMPC control law: such phenomenon can be mitigated by increasing the number  $\nu$  of off-line computed control moves. Indeed, the choice of the value of  $\nu$  is a crucial point. A higher value of  $\nu$  leads to better accuracy, but also to higher memory requirements. With the uniform gridding approximation and the implementation presented in Section 5.2, the on-line computational time is independent

on  $\nu$ . The obtained average computational time for the FMPC control law on the Rabbit<sup>®</sup> processor, considering extensive series of SIL simulations, is 2.8 ms with a maximal value of 3 ms, including also the communication delay between the Rabbit<sup>®</sup> device and the PC. Such computational time is much lower than the chosen sampling interval (i.e. 10 ms). On the other hand, as anticipated in Section 3, the on-line computational time of the exact NMPC law is about 30 ms on a standard PC and about 15 s on the Rabbit<sup>®</sup> processor, thus hampering the possibility to compute the exact NMPC law in real-time. As to the considered  $\mu$ -split braking maneuver, Fig. 9 shows the vehicle trajectories obtained in the uncontrolled case (black), with the IMC controller (white) and with the FMPC controller (gray). It can be noted that the approximated predictive control law achieves the best performance, in terms of trajectory deviation. The steering wheel frequency sweep maneuver (Fig. 10)

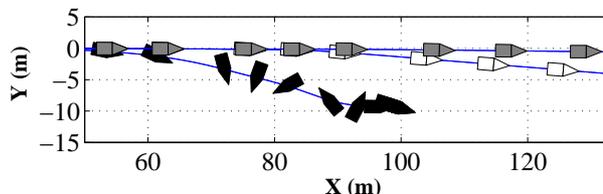


Fig. 9.  $\mu$ -split braking maneuver at 100 km/h. Comparison between the trajectories obtained with the uncontrolled vehicle (black) and the FMPC (gray) and IMC (white) controlled ones.

aims at evaluating the improvement achieved by the FMPC controlled vehicle in terms of resonance peak reduction and bandwidth increase. In particular, the frequency course of the transfer ratio:  $T_m(\omega) = (\bar{\psi}(\omega))/(\bar{\psi}(0))$  has been analyzed, where  $\bar{\psi}(\omega)$  is the steady state yaw rate amplitude obtained in the presence of the sinusoidal 30° handwheel input at frequency  $\omega$ , and  $\bar{\psi}(0)$  is the steady state yaw rate in presence of a constant handwheel input of 30°. The FMPC controlled vehicle achieves a lower resonance peak (1 dB) and higher bandwidth (3.4 Hz) with respect to the case of the uncontrolled vehicle (2.8-dB resonance peak and 2.2-Hz bandwidth). The vehicle controlled by the nominal NMPC law has the same behavior as the FMPC controlled one, while the vehicle controlled with the IMC structure achieves a higher resonance peak (about 1.9 dB) and slightly lower bandwidth (about 3 Hz).

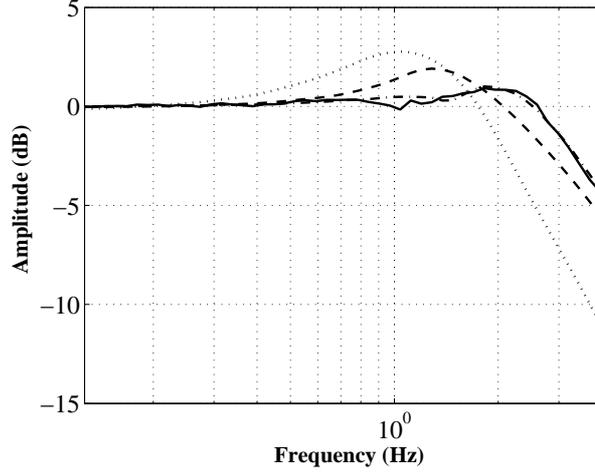


Fig. 10. Steering wheel frequency sweep test at 100 km/h. Comparison between the frequency response obtained with the uncontrolled vehicle (dotted) and those obtained the nominal NMPC (dash-dotted) and FMPC (solid) controlled vehicles and with the IMC structure (dashed).

All the presented results show that the designed FMPC law improves the closed loop performance with respect to the IMC structure. Finally, the re-

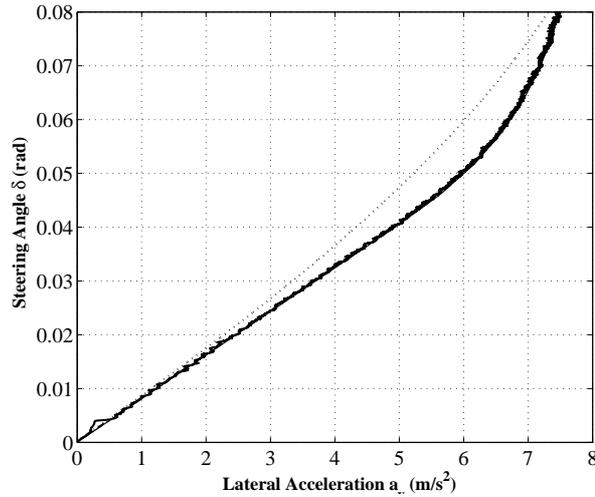


Fig. 11. Steering pad test at 100 km/h. Comparison between the reference curve (thin solid line) and those obtained with the uncontrolled vehicle (dotted), with the nominal NMPC (dash-dotted) and FMPC (solid) controlled vehicles and with the IMC structure (dashed).

sults of the steering pad test, shown in Fig. 11, indicate that the steady state behaviors obtained with all of the considered controllers are practically superimposed and correspond to the desired one.

## 7 Conclusions

A NMPC strategy for vehicle yaw control has been introduced, and the designed NMPC law has been approximated by means of SM techniques via a Nearest Point approach, using a finite number of exact off-line solutions. SIL results, performed with a commercial embedded processor with low computational performance and an accurate model of the considered vehicle, show the effectiveness of the NMPC approach and of the proposed approximation technique. In particular, it has been shown that a highly damped behavior in steer reversal maneuvers has been obtained; stability is guaranteed in presence of demanding driving conditions like  $\mu$ -split braking and resonance peak has been significantly reduced in the frequency response. Finally, improvements over a well assessed approach which employs an enhanced IMC structure have been shown too.

## References

- Ackermann, J., Guldner, J., Steinhausner, R., Utkin, V. I., 1995. Linear and nonlinear design for robust automatic steering. *IEEE Trans. on Control System Technology* 3 (1), 132–143.
- Ackermann, J., Sienel, W., 1993. Robust yaw damping of cars with front and rear wheel steering. *IEEE Trans. on Control Systems Technology* 1 (1), 15–20.
- Bakker, E., Lidner, L., Pacejka, H., 1989. A new tyre model with an application in vehicle dynamics studies. In: SAE Paper 890087.
- Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E., 2002. The explicit linear quadratic regulator for constrained systems. *Automatica* 38, 3–20.
- Canale, M., Fagiano, L., Milanese, M., 2009. Set membership approximation theory for fast implementation of model predictive control laws. *Automatica* 45 (1), 45–54.
- Canale, M., Fagiano, L., Milanese, M., Borodani, P., 2007. Robust vehicle yaw control using an active differential and imc techniques. *Control Engineering*

- Practice 15, 923–941.
- Damm, W., 2006. Embedded system development for automotive applications: trends and challenges. In: 6<sup>th</sup> ACM & IEEE International conference on Embedded software. Seoul, Korea.
- Falcone, M., Borrelli, F., Tseng, H. C., Hrovat, D., May 2007. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control System Technology* 15 (3), 566–580.
- Frediani, S., Gianoglio, R., Giuliano, F., 2002. System for the active control of a motor vehicle differential. Patent no. US 2002/0016661 A1, Applicant Centro Ricerche Fiat.
- Freescale Semiconductor, 2000. Motorola 68336 datasheet. [Http://www.freescale.com/](http://www.freescale.com/).
- Fukunaga, K., 1990. *Introduction to Statistical Pattern Recognition*, 2nd Edition. Academic Press, San Diego.
- Genta, G., 2003. *Motor Vehicle Dynamics*, II ed. World Scientific.
- Güvenç, B. A., Bünte, T., Güvenç, L., 2004. Robust two degree-of-freedom vehicle steering controller design. *IEEE Trans. on Control System Technology* 12 (4), 627–636.
- Ippolito, L., Lupo, G., Lorenzini, A., 1992. System for controlling torque distribution between the wheels of a common vehicle axle. EU Patent no. 92121621.4, Applicant Centro Ricerche Fiat.
- Johansen, T., 2004. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica* 40, 293–300.
- Kiencke, U., Nielsen, L., 2000. *Automotive control systems*. Springer Verlag.
- Kohen, P., Ecrick, M., 2004. Active steering - the BMW approach towards modern steering technology. In: SAE Technical Paper No. 2004-01-1105.
- Mayne, D. Q., Michalska, H., 1990. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control* 35 (7), 814–824.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., Sokaert, P., 2000. Constrained model predictive control: Stability and optimality. *Automatica* 36, 789–814.
- Meadows, E., December 1994. Stability and continuity of nonlinear model predictive control. Ph.D. thesis, University of Texas.
- Milanese, M., Novara, C., 2004. Set membership identification of nonlinear

- systems. *Automatica* 40, 957–975.
- Nocedal, J., Wright, S., 2006. *Numerical Optimization*. Springer.
- Piyabongkarn, D., Rajamani, R., Grogg, J., Lew, J., 2006. Development and experimental evaluation of a slip angle estimator for vehicle stability control. In: 25<sup>th</sup> American Control Conference. Minneapolis, MN.
- Rabbit Semiconductor, 2006. Rabbit bl2600 datasheet. <http://www.rabbit.com/products/bl2600/bl2600.pdf>.
- Rajamani, R., 2005. *Vehicle Dynamics and Control*. Springer Verlag.
- Ryu, J., Gerdes, J., 2004. Integrating inertial sensors with gps for vehicle dynamics control. *ASME Journal on Dynamic Systems, Measurement and Control* 126, 243–254.
- Seron, M., Goodwin, G., Doná, J. D., 2003. Characterization of receding horizon control for constrained linear systems. *Asian Journal of Control* 5 (2), 271–286.
- Siemens, 2005. Siemens 80c166 datasheet. <http://www.datasheetcatalog.org/datasheet/siemens/SAB80C166-M-T3.pdf>.
- Spjøtvold, J., Tøndel, P., Johansen, T. A., 2007. Continuous selection and unique polyhedral representation of solutions to convex parametric quadratic programs. *Journal of Optimization Theory and Applications* 134, 177–189.
- Tøndel, P., Johansen, T., June 2005. Control allocation for yaw stabilization in automotive vehicles using multiparametric nonlinear programming. In: *American Control Conference*.
- Van Zanten, A. T., 2000. Bosch ESP systems: 5 years of experience. In: *SAE Technical Paper No. 2000-01-1633*.
- Van Zanten, A. T., Erhart, R., Pfaff, G., 1995. VDC, the vehicle dynamics control system of bosch. In: *SAE Technical Paper No. 95759*.