Nonlinear Model Predictive Control: an Optimal Search Domain Reduction

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 6253–6258

Nonlinear Model Predictive Control: an Optimal Search Domain Reduction ⋆

Mattia Boggio* Carlo Novara* Michele Taragna*

* *Politecnico di Torino, Dept. of Electronics and Telecommunications Corso Duca degli Abruzzi 24, 10129 Turin, Italy (e-mail: mattia.boggio@polito.it; carlo.novara@polito.it; michele.taragna@polito.it)*

Abstract: Nonlinear Model Predictive Control (NMPC) is a powerful control method, used in many industrial contexts. NMPC is based on the online solution of a suitable Optimal Control Problem (OCP) but this operation may require high computational costs, which may compromise its implementation in "fast" real-time applications. In this paper, we propose a novel NMPC approach, aiming to improve the numerical efficiency of the underlying optimization process. In particular, a Set Membership approximation method is applied to derive from data tight bounds on the optimal NMPC control law. These bounds are used to restrict the search domain of the OCP, allowing a significant reduction of the computation time. The effectiveness of the proposed NMPC strategy is demonstrated in simulation, considering an overtaking maneuver in a realistic autonomous vehicle scenario.

Keywords: Nonlinear Predictive Control; Numerical methods for optimal control; Search Domain Reduction; Set Membership; Nonlinear system identification.

1. INTRODUCTION

Model Predictive Control (MPC) is widely recognized as a flexible and powerful control approach for a large number of industrial and technological applications, thanks to its capability to design control algorithms for multivariable systems under state, input, and output constraints (see, e.g., Richalet et al. (1978), Qin and Badgwell (2000), Mayne (2014) and Franzè et al. (2019)). To cope with nonlinear dynamics and constraints, as well as with nonconvex performance indexes, Nonlinear MPC (NMPC) techniques have been introduced (see, e.g., Allgöwer et al. (2004), Diehl et al. (2005) and references therein). These control techniques are based on suitable Optimal Control Problems (OCPs), whose solution must be obtained within a sufficiently short time interval, depending on the application of interest.

In general, MPC and its nonlinear version require the availability of efficient optimization algorithms, able to meet the time constraints of real-time closed-loop control applications. In the case of linear MPC, since it can often be formulated as a structured convex quadratic programming (QP) problem, a wide range of fast algorithms have been developed (see, e.g., Wright (1997) and Wang and Boyd (2010)). On the other hand, NMPC deals in general with nonconvex optimization and therefore mainly relies on sophisticated algorithms with higher computational costs than linear MPC. Nevertheless, thanks to progress in nonlinear optimization algorithms, efficient implementations of NMPC can be found also in "fast" applications (see, e.g., Houska et al. (2011), Gros et al. (2012), and Albin et al. (2015)).

In this paper, we propose a novel NMPC approach based on an optimal reduction of the search domain of the underlying optimization process. Thanks to this operation, we obtain a significant reduction of the required computation time, thus enabling the real-time NMPC implementation in many situations where a high sampling rate is necessary. It must be remarked that the domain reduction technique is not restricted to a specific optimization algorithm, but it can be used in combination with any algorithm to increase its numerical efficiency.

In general, a nonlinear optimization algorithm consists of two mechanisms for the search process, namely exploration and exploitation. Exploration refers to the process of investigating the search domain looking for the possible optimal solutions. Exploitation refers to the process of finding the feasible solution around the candidate solutions obtained during the exploration phase. The larger the search domain, the more computationally intensive the exploration phase is, since more evaluations of the cost function are required to find an optimal solution. In this perspective, the proposed approach aims to restrict, in an optimal way, the search domain, allowing a significant reduction of the computation time. In particular, a Set Membership (SM) approximation method is used to derive from data tight bounds on the optimal NMPC control law, reducing accordingly the search range of the optimization process. Unlike many classical estimation methods, the SM approach makes use of the so-called *interval bounds* to compute the estimate of an unknown function, also ensur-

⋆ This work was supported by the NewControl project, within the Electronic Components and Systems For European Leadership Joint Undertaking (ESCEL JU) in collaboration with the European Union's Horizon2020 Framework Programme and National Authorities, under grant agreement N° 826653-2.

10.1016/j.ifacol.2023.10.768

ing that the true value is contained inside the resulting uncertainty band (see Milanese and Novara (2004)).

Taking advantage of this property, the method proposed in this paper, called Reduced Domain Nonlinear Model Predictive Control (RD-NMPC), is based on two basic operations: (i) approximation of the NMPC control law, i.e., the nonlinear function that links the state of the system to the optimal command and using this approximation for the warm start of the nonlinear optimization algorithm; (ii) computation of tight bounds on the NMPC control law, allowing a reduction of the search domain. To our knowledge, this is a novelty with respect to previously developed methods, where the approximating function was used just to reproduce the MPC/NMPC law (see, e.g., Parisini and Zoppoli (1995) and Canale et al. (2006)) or directly to find an optimal open-loop solution (see, e.g., Canale et al. (2014)). In our case, the approximating function gives a "warm" start point of the NMPC optimization algorithm. This yields a further improvement of the controller performance and a better generalization capability in the case that the approximating function is not accurate in some regions of the controller domain. Moreover, no reductions of the search domain are performed in those previous methods and, in general, in the majority of NMPC techniques.

The proposed RD-NMPC strategy is tested in simulation, considering an overtaking maneuver in an autonomous vehicle scenario. The performance of the strategy is shown to be significantly better compared to a standard NMPC approach, in terms of computation time and optimality of the solutions found.

The paper is organized as follows. Section 2 introduces the NMPC mathematical formulation. In Section 3, the RD-NMPC approach is presented. The obtained results and comparison with a standard NMPC are shown in Section 4. Finally, the conclusions are drawn in Section 5.

2. NONLINEAR MODEL PREDICTIVE CONTROL

Consider a Multiple-Input-Multiple-Output (MIMO) nonlinear system described by the following state equations:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is the command input and $y \in \mathbb{R}^{n_y}$ is the output; $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are two functions characterizing the system dynamics and output variables, respectively. Assume that the state is measured in real-time, with a sampling time T_s, according to

$$x(t_k), \quad t_k = T_s k,\ k = 0, 1 \ldots.$$

If the state is not measured, an observer or a model of (1) in input-output form has to be employed.

Suppose that the system output $y(t)$ is required to track a desired reference signal $r(t)$. The state, output and input variables may be subject to constraints, and it may be of interest to have a suitable trade-off between performance and command effort. NMPC is a suitable approach to tackle such a control problem and is based on two key operations: prediction and optimization. At each time $t = t_k$, the system state and output are predicted over the time interval $[t, t + T_p]$, where $T_p \geq T_s$ is called the *prediction horizon.* The prediction is obtained by integration of (1). For any $\tau \in [t, t + T_p]$, the predicted output $\hat{y}(\tau)$ is a function of the "initial" state $x(t)$ and the input signal:

$$\hat{y}(\tau) \equiv \hat{y}(\tau, x(t), u(t : \tau)) \tag{2}$$

where $u(t : \tau)$ denotes the input signal in the interval $[t, \tau]$. The basic idea of NMPC (and of the most predictive approaches) is to look for an input signal $u^*(t : \tau)$ at each time $t = t_k$, such that the prediction $\hat{y}(\tau, x(t), u^*(t : \tau))$ has the desired behavior in the time interval $[t, t+T_p]$. The concept of desired behavior is formalized by defining the *objective function*

$$J(u(t : t + T_p)) \doteq \int_t^{t+T_p} \left(\|e_p(\tau)\|_Q^2 + \|u(\tau)\|_R^2\right) d\tau + \|e_p(t + T_p)\|_P^2 \tag{3}$$

where $e_p(\tau) \doteq r(\tau) - \hat{y}(\tau)$ is the predicted tracking error, $r(\tau) \in \mathbb{R}^{n_y}$ is the reference to track, and $\|\cdot\|_*$ is a weighted Euclidean norm. For example, if Q is a positive definite weight matrix, the norm of a vector w is defined as $\|w\|_Q^2 \doteq w^\top Q w$.

The input signal $u^*(t : t + T_p)$ is the minimizer of the objective function $J(u(t : t + T_p))$. In particular, at each time $t = t_k$, for $\tau \in [t, t + T_p]$, the following optimization problem is solved:

$$\begin{aligned} & u^*(t : t + T_p) = \arg\min_{u(\cdot)} J(u(t : t + T_p)) \\ & \text{subject to:} \\ & \quad \dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)), \quad \hat{x}(t) = x(t) \\ & \quad \hat{y}(\tau) = h(\hat{x}(\tau), u(\tau)) \\ & \quad \hat{x}(\tau) \in X_c,\ \hat{y}(\tau) \in Y_c,\ u(\tau) \in U_c. \end{aligned} \tag{4}$$

The first two constraints in this problem ensure that the predicted state and output are consistent with the system equations (1). The sets X_c and Y_c account for other constraints that may hold for the predicted state/output (e.g., obstacles or barriers). The set U_c accounts for input constraints (e.g., input saturation).

The NMPC closed-loop command is obtained according to a so-called *receding horizon strategy* (RHS). At time $t = t_k$, the input signal $u^*(t : t + T_p)$ is computed by solving (4). Then, only the first optimal input value $u(\tau) = u^*(t_k)$ is applied to the plant (1), keeping it constant $\forall \tau \in [t_k, t_{k+1}]$. The complete procedure is repeated at the subsequent time steps $t = t_{k+1}, t_{k+2} \ldots$.

The optimization problem (4) is in general non-convex. Moreover, the decision variable $u(\cdot)$ is a signal, and optimizing a function with respect to a signal is a difficult task. To overcome these issues, the command signal is parametrized as follows: The prediction interval $[t, t + T_p]$ is divided into n_s sub-intervals, and u and r are kept constant on each sub-interval. The command and reference values at time k in the ith sub-interval are denoted by u_{ki} and r_{ki}, respectively. The command and reference sequences in the prediction interval are $u_k \doteq (u_{k1}, \ldots, u_{kn_s})$ and $r_k \doteq (r_{k1}, \ldots, r_{kn_s})$, respectively. In this way, the optimization problem reduces to a finite-dimension problem, which can be solved using an efficient numerical optimization algorithm.

3. REDUCED DOMAIN NMPC

This section describes the method that we propose to improve the computational performance of an NMPC algorithm.

According to the formulation of Section 2, the optimal command sequence $u_k^* \doteq (u_{k1}^*, \ldots, u_{kn_s}^*)$ obtained solving (4) is a static nonlinear function $\phi(\cdot)$ of the current state $x_k \doteq x(t_k)$ and the reference sequence $r_k \doteq (r_{k1}, \ldots, r_{kn_s})$. Hence, the optimal sequence u_k^* at time t_k is given by

$$u_k^* = \phi(w_k) \tag{5}$$

where $w_k \doteq (x_k, r_k)$.

The proposed method consists in approximating $\phi(\cdot)$ and computing tight bounds of this function from a set of data computed offline. The approximation is used for warm starting the solution of the optimization problem, whereas the bounds are used to reduce the search domain of this problem. The phases of the method are as follows.

1) *Data Collection.* Several offline simulations are performed. A set of state data $\tilde{x}_k$ and reference sequences $\tilde{r}_k$, with $k = 1 \ldots, M$, are generated, giving a set of values of the regressor $\tilde{w}_k \doteq (\tilde{x}_k, \tilde{r}_k) \in \mathbb{W}$, where $\mathbb{W}$ is a bounded region where the regressor can evolve. For each $\tilde{w}_k$, the corresponding optimal command sequence is computed by solving (4), giving rise to a set of command data $\tilde{u}_k = \phi(\tilde{w}_k)$, $k = 1, \ldots, M$.

2) *Clustering.* A clustering procedure is carried out to reduce the number of data used to derive the approximation of the NMPC control law ϕ. The *K-Medoids* approach is used, which is based on medoids to represent the various clusters (see Kaufman and Rousseeuw (2009)). A medoid is a point of the data set whose sum of dissimilarities with respect to all the other points in the cluster is minimal. Among many algorithms for K-medoids clustering, the CLustering LARge Applications (CLARA) is used, able to deal with large data sets.

At the end of the clustering analysis, the size of the database can be reduced by at least 10 times. This means that:

$$K \leq \frac{M}{10} \tag{6}$$

where K is the number of clusters and then the number of data used to identify the function ϕ. These can be seen as the data that best characterize the controller.

3) *Set Membership Approximation.* After the clustering process, the resulting data set is composed of K regressor and command values $\tilde{w}_{mk}$ and $\tilde{u}_{mk}$. The subscript m is used to indicate that the data are the medoids of the clusters found in the previous phase.

On the basis of the data $\tilde{w}_{mk}$, $\tilde{u}_{mk}$, $k = 1, \ldots, K$, tight function bounds $\overline{\phi}$ and $\underline{\phi}$, and an approximated control law ϕ^c are computed by means of the SM approach of Milanese and Novara (2004). These functions are defined as follows:

$$\begin{aligned} \overline{\phi}(w) &\doteq \sup_{\varphi \in FFS} \varphi(w) \\ \underline{\phi}(w) &\doteq \inf_{\varphi \in FFS} \varphi(w) \\ \phi^c(w) &\doteq \frac{1}{2}(\overline{\phi}(w) + \underline{\phi}(w)) \end{aligned} \tag{7}$$

where the supremum and infimum are intended as element-wise operations. FFS is called the Feasible Function Set and is the set of all functions consistent with the prior assumptions and the data. This set is defined as

$$FFS \doteq \{\varphi \in \mathcal{F}(\gamma) : \tilde{u}_{mk} = \varphi(\tilde{u}_{mk}), k = 1, \ldots, K\} \tag{8}$$

where $\mathcal{F}(\gamma)$ is the set of Lipschitz continuous functions with local Lipschitz constant $\gamma(w)$, $w \in \mathbb{W}$. The Lipschitz constant can be estimated from the collected data, allowing the explicit evaluation of $\overline{\phi}(w)$, $\underline{\phi}(w)$ and $\phi^c(w)$, for any $w \in \mathbb{W}$ (see Milanese and Novara (2004)). Furthermore, the following properties are shown in Milanese and Novara (2004): (i) $\overline{\phi}$ and $\underline{\phi}$ are *optimal bounds* of ϕ: they are the tightest upper and lower bounds that can be derived from the available prior information on the function and the data. (ii) ϕ^c is an *optimal approximation* of ϕ: it minimizes the so-called *worst-case identification error*, defined as the maximum error given by all possible approximations that are compatible with the prior information and the data (i.e., that belong to FFS).

4) *Optimal Search Domain Reduction.* Once the approximate function and its bounds have been found, they can be used within the NMPC optimization process in order to improve the computational efficiency.

As described in detail in Section 2, the NMPC finds the optimal control by solving the Optimal Control Problem (OCP) formulated in (4) over a finite sequence of control actions at each sampling time. This operation can be prohibitive for "fast" real-time applications. Here, a solution for reducing the computational complexity is proposed. The novelty lies in the way the initial conditions of the algorithm are computed and the definition of optimal bounds within which the algorithm searches for the optimal solution.

In standard NMPC approaches, the bounds of the search domain typically are derived from the specifications of the considered application. However, wide bounds lead to large search domains and this can be a serious issue. Indeed, nonlinear optimization algorithms are characterized by a combination of the following operations: (i) exploration: trying to find which part of the feasibility region is most promising; (ii) exploitation: trying to reach the optimum as fast as possible. The exploration phase is very important, since the objective function may have several local minima, and the algorithm has to explore the search domain as much as possible, to find a satisfactory one. Hence, the larger the search domain the more time-consuming the exploration phase becomes.

The RD-NMPC algorithm exploits ϕ^c, $\overline{\phi}$ and $\underline{\phi}$ as follows: (i) finding an initial condition suitable for warm starting the optimization algorithm; (ii) finding optimal command bounds. The latter is the key point of the algorithm: The tighter these optimal bounds are, the smaller the search domain becomes, and the less cumbersome the exploration phase of the algorithm is. This leads to a reduction in the number of cost functions to be evaluated and, consequently, to less computational time needed to find the optimal command.

5) *RD-NMPC algorithm.* In summary, the RD-NMPC algorithm consists in solving the OCP (4) where the warm start $u_{start} = \phi^c(w_k)$ is used and the command domain is

given by

$$U_c = \{u_k \in \mathbb{R}^{n_s n_u} : \underline{\phi}(w_k) \le u \le \overline{\phi}(w_k)\} \tag{9}$$

where "≤" are element-wise inequalities.

4. EXAMPLE: AUTONOMOUS VEHICLE OVERTAKING MANEUVER

The RD-NMPC algorithm has been tested in simulation in a road scenario involving an overtaking maneuver. An example of overtaking trajectory is shown in Fig. 1. The parameters a and b determine the shape of this trajectory.

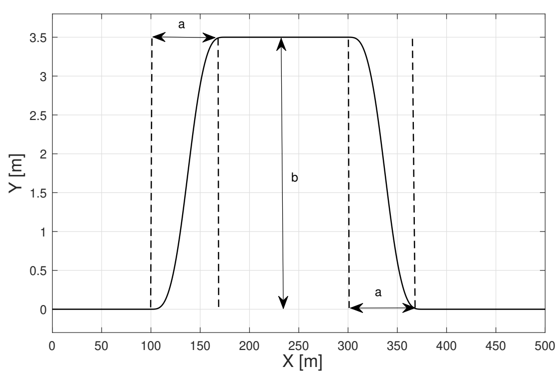

Fig. 1. Example of overtaking maneuver.

Firstly, the models used to describe the ego-vehicle are presented. When implementing the NMPC approach, it is important to distinguish between two models: a "high-fidelity" plant model that simulates the real vehicle and a prediction model that the NMPC optimization algorithm uses to predict the system future behavior. Generally, the prediction model is less complex than the high-fidelity plant model. To simulate the real vehicle, the Matlab Dual-Track Vehicle Body 3DOF block (see MATLAB (2018)) is used, which implements a rigid two-axle vehicle body model to calculate longitudinal, lateral, and yaw motion. This block accounts for body mass, aerodynamic drag, and weight distribution between the axles due to acceleration and steering. Regarding the NMPC prediction model, a standard model of the lateral and longitudinal dynamics of a vehicle is considered, called the Dynamic Single-Track (DST) Model. Although simple, this model captures the main aspects of the vehicle dynamics and, for this reason, it is suitable for the design and preliminary test of vehicle control systems. The state equations of the DST model are:

$$\begin{aligned}
\dot{X} &= v_x \cos\psi - v_y \sin\psi \\
\dot{Y} &= v_x \sin\psi + v_y \cos\psi \\
\dot{\psi} &= w \\
\dot{v}_x &= v_y \dot{\psi} + a_x \\
\dot{v}_y &= -v_x \dot{\psi} + \frac{2}{m}\left(F_{yf} + F_{yr}\right) \\
\dot{w} &= \frac{2}{I_z}\left(l_f F_{yf} - l_r F_{yr}\right).
\end{aligned} \tag{10}$$

where X and Y denote the position of the vehicle, ψ the yaw angle, w the yaw rate, v_x, v_y the longitudinal and lateral speeds. The parameters are as follows: $m = 1575\,\text{kg}$ and $I_z = 4000\,\text{kg*m}^2$ are the vehicle mass and yaw polar inertia, and $l_f = 1.2\,\text{m}$ and $l_r = 1.6\,\text{m}$ are the distances from the Center of Gravity (CoG) to the front and rear wheels, respectively. F_{yf} and F_{yr} are the lateral forces exchanged between the wheels and the road. The following simple linear model is used for these forces:

$$F_{yf} = -c_f \beta_f, \qquad F_{yr} = -c_r \beta_r \tag{11}$$

where $c_f = 2.7 \cdot 10^4\,\text{N/rad}$ and $c_r = 2 \cdot 10^4\,\text{N/rad}$ are the front/rear cornering stiffnesses. The tire slip angles β_f and β_r are defined as:

$$\beta_f = \text{atan}\left(\frac{v_y + l_f \dot{\psi}}{v_x}\right) - \delta_f, \ \ \beta_r = \text{atan}\left(\frac{v_y - l_r \dot{\psi}}{v_x}\right) \tag{12}$$

The longitudinal acceleration a_x and the steering angle δ_f are the control variables. The output of the system is (X, Y).

The operations outlined in Section 3 are now presented for this example.

A set of 1000 overtaking maneuvers were simulated, considering different values of the parameters a and b (see Fig. 1) by using the Latin Hypercube Sampling (LHS) technique (see McKay et al. (1979)). A constant speed of 60 km/h was assumed. Starting from this set, a Monte Carlo (MC) simulation campaign was carried out using the NMPC algorithm (4) without domain reduction (the design parameters are listed in Table 1). The simulations were performed in Simulink. The optimization problem was embedded in an *interpreted Matlab function* and solved using the Matlab command *fmincon*, with the Sequential Quadratic Programming (SQP) algorithm. In the following, this algorithm without domain reduction will be called "Standard NMPC". The NMPC command was parametrized considering 2 sub-intervals of the prediction horizon $[t_k, t_k + T_p]$, implying that there are a total of 4 commands: 2 for the longitudinal acceleration a_x and 2 for the steering angle δ_f. At the end of this campaign, a database of about $M = 5e5$ samples was obtained.

Table 1. NMPC design parameters

Parameter	Value
T_s	0.1 s
T_p	3 s
Q	diag(1, 1)
R	diag(0.01, 1)
Upper bounds	$[3\,\text{m/s}^2, \pi/8, 3\,\text{m/s}^2, \pi/8]$
Lower bounds	$[-3\,\text{m/s}^2, -\pi/8, -3\,\text{m/s}^2, -\pi/8]$

Once the database was created, a Clustering Analysis using the CLARA algorithm was performed in order to suitably reduce it. After several trials, a number of K medoids equal to $2e4$ was found as a compromise between quantity of data (and then memory occupation) and exploration of the control law domain.

Therefore, after the clustering process, the database was reduced from 5e5 to 2e4 samples $\{\tilde{w}_{mk}, \tilde{u}_{mk}\}$ with $k = 1, \ldots, 2e4$. On the basis of them, the approximated control law ϕ^c, and the corresponding bounds $\overline{\phi}$ and $\underline{\phi}$ were computed by means of the SM approach shown in Section 3. Fig. 2 shows the approximation and the relative bounds of one of the steering angle commands. As it can be seen, the bounds were reduced more than 10 times with respect to the original ones, given by the physical limitations of the steering actuator.

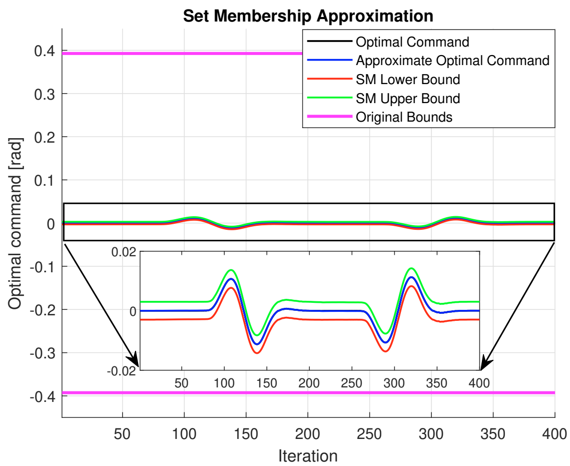

Fig. 2. Set Membership Approximation of δ_f.

Once the approximate Set Membership model was created, it was used in combination with the NMPC for reducing the computational time of the optimization algorithm. In order to test the effectiveness of this technique and the robustness of the obtained model, different values of a and b, from those considered previously, were taken into account. Then, a MC campaign of 100 simulations was carried out. The standard NMPC and the developed RD-NMPC were simulated in Simulink on a Dell Precision 5820 (Processor: Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz). The optimization problem was solved using *fmincon* and the SQP algorithm, embedded in an *interpreted Matlab function*. It must be noted that the proposed approach is not restricted to any specific optimization algorithm but it can be used in combination with any algorithm to increase its numerical efficiency.

The indexes used for comparing the performance of the two algorithms are:

(1) Number of evaluated cost functions (Eval. Cost. Funct.) for finding the minimum;
(2) Computational time (Comp. Time);
(3) Root-Mean-Square (RMS) Error of both the Lateral Error (Lat. E.) and the Orientation Error (Or. E.).

In Table 2, the mean and maximum values of the above performance indexes are shown for the two NMPC algorithms. The term Mean Value refers to the average number of evaluated cost functions, computational times, and tracking errors throughout the simulations, while the term Maximum Value refers to the highest absolute value. Regarding the number of evaluated cost functions, with the RD-NMPC a reduction of about 11 times, on average, is obtained. Regarding the computational time, the use of RD-NMPC leads to an improvement in the performance of about 4 times, on average. The same considerations also apply to the maximum values of both metrics. Finally, considering the RMS errors, we can see that the obtained results for the Mean and Maximum Values are quite similar for both the NMPC algorithms.

Table 2. Comparison between Standard NMPC and RD-NMPC

	Standard NMPC		**RD-NMPC**	
	Mean Value	Maximum Value	Mean Value	Maximum Value
Eval. Cost Funct.	98.3	102.4	8.4	9.1
Comp. Time [s]	0.0329	0.0359	0.009	0.0092
RMS Lat. E. [m]	0.0355	0.0418	0.0339	0.0402
RMS Or. E. [rad]	0.0024	0.0033	0.002	0.0031

Examples of overtaking maneuvers performed by the RD-NMPC algorithm are shown in Fig. 3.

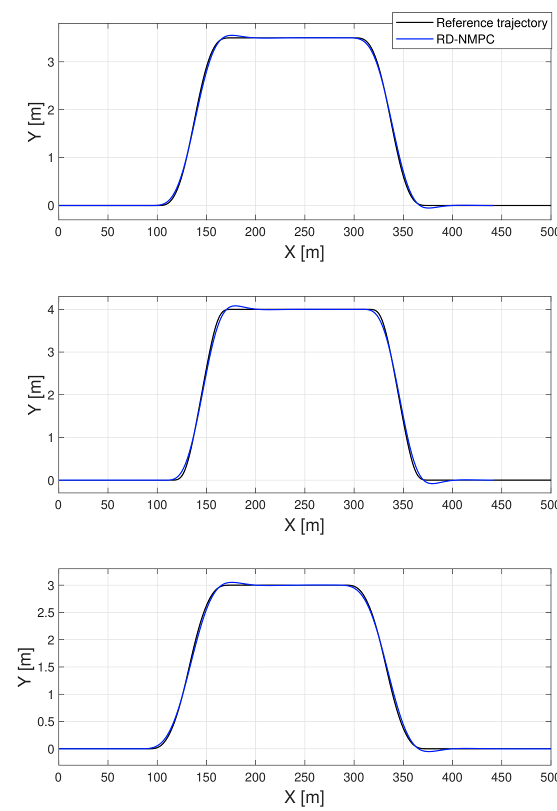

Fig. 3. Examples of RD-NMPC overtaking maneuvers: 50 km/h (top), 60 km/h (middle) and 70 km/h (bottom).

5. CONCLUSIONS

The paper proposes a novel approach for improving the numerical efficiency of NMPC algorithms, enabling their real-time implementation even in systems with high sampling rates. In particular, a Set Membership approximation method is applied to derive from data tight bounds on the optimal NMPC control law. These bounds are used to reduce the search domain of the underlying optimization process, allowing a significant decrease of the computation time. The proposed approach has been tested in simulation, considering an overtaking maneuver in an autonomous driving scenario. The obtained results demonstrate the effectiveness of the method, in terms of computation time and optimality of the solutions found, with respect to the NMPC implementation without domain reduction.

REFERENCES

Albin, T., Ritter, D., Abel, D., Liberda, N., Quirynen, R., and Diehl, M. (2015). Nonlinear MPC for a two-stage turbocharged gasoline engine airpath. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 849–856.

Allgöwer, F., Findeisen, R., and Nagy, Z. (2004). Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs*, 35, 299–315.

Canale, M., Fagiano, L., and Signorile, M. (2014). Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control*, 24(1), 123–139.

Canale, M., Milanese, M., and Novara, C. (2006). Semi-active suspension control using “fast” model-predictive techniques. *IEEE Transactions on Control Systems Technology*, 14(6), 1034–1046.

Diehl, M., Bock, H.G., Diedam, H., and Wieber, P.B. (2005). Fast Direct Multiple Shooting Algorithms for Optimal Robot Control. In *Fast Motions in Biomechanics and Robotics*. Heidelberg, Germany.

Franzè, G., Mattei, M., Ollio, L., and Scordamaglia, V. (2019). A robust constrained model predictive control scheme for norm-bounded uncertain systems with partial state measurements. *International Journal of Robust and Nonlinear Control*, 29(17), 6105–6125.

Gros, S., Quirynen, R., and Diehl, M. (2012). Aircraft control based on fast non-linear MPC & multiple-shooting. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 1142–1147.

Houska, B., Ferreau, H.J., and Diehl, M. (2011). An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47(10), 2279–2285.

Kaufman, L. and Rousseeuw, P.J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

MATLAB (2018). Vehicle body 3dof. https://it.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.

McKay, M.D., Beckman, R.J., and Conover, W.J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2), 239–245.

Milanese, M. and Novara, C. (2004). Set membership identification of nonlinear systems. *Automatica*, 40(6), 957–975.

Parisini, T. and Zoppoli, R. (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10), 1443–1451.

Qin, S.J. and Badgwell, T.A. (2000). An overview of nonlinear model predictive control applications. In *Nonlinear Model Predictive Control*, 369–392. Birkhäuser Basel, Basel.

Richalet, J., Rault, A., Testud, J., and Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5), 413–428.

Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.

Wright, S.J. (1997). Applying new optimization algorithms to model predictive control. *Chemical Process Control-V*, 93(316), 147–155.