Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-15 (2024) 217–222

Set Membership identification for NMPC complexity 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: A Set Membership (SM) approach is proposed to reduce the computational burden of Nonlinear Model Predictive Control (NMPC) algorithms. In particular, a SM identification method is applied to derive an approximation and tight bounds of the NMPC control law, using a set of its values computed offline. These quantities are used online to reduce the dimension and the volume of the search domain of the NMPC optimization algorithm, and to perform a warm start, allowing a significant shortening of the computational time. The developed NMPC methodology is tested in simulation, considering an obstacle avoidance application in a realistic autonomous vehicle scenario. The obtained results demonstrate the effectiveness of the proposed approach in terms of computation time, without affecting the solution quality.

Keywords: Set Membership identification; Nonlinear Model Predictive Control; complexity reduction; data-aided control; autonomous vehicles.

1. INTRODUCTION

Nonlinear Model Predictive Control (NMPC) has become a robust and flexible approach capable of imposing optimal trajectories over a finite time interval, handling input/state/output constraints, and systematically managing the trade-off between performance and command activity (see, e.g., Mayne (2014) and references therein). This control strategy has been successfully applied in various fields, including automotive engineering, aerospace engineering, chemical processes, robotics, energy, biomedicine, and more (see, e.g., Siampis et al. (2018) and Pagone et al. (2021)). The NMPC approach is based on an optimal control problem (OCP), which must be solved online within a short time. However, the OCP is generally nonconvex and its solution can be computationally expensive, making unfeasible the real-time NMPC implementation. To address this issue, different NMPC techniques have been proposed in the literature. These techniques can be broadly categorized into two groups: i) improving the numerical efficiency of the optimization algorithms, and ii) offline approximation of the control law. The former group includes specific online algorithms that reduce the computational burden of the underlying nonlinear program (NLP), such as multiple shooting method (Bock et al. (1999)), collocation methods (Biegler (2000)) and Real Time Iteration (RTI) scheme (Diehl et al. (2002)). These algorithms enable the development of NMPC algorithms that can be used in real-time applications with short sampling times, as demonstrated by Houska et al. (2011) and Gros et al. (2012). The latter group considers approximating functions, derived offline, to reproduce the MPC/NMPC law, as seen in Parisini and Zoppoli (1995) and Canale et al. (2006). However, these methods lack efficiency in presence of huge number of system states, complex constraints and time-varying references.

Relevant issues that strongly affect the computational time of nonlinear optimization algorithms are the following: (i) dimension of the search domain, i.e., the number of decision variables; (ii) volume of the search domain, i.e., the overall set where the decision variables are defined; (iii) starting point of the algorithm.

The main contribution of this paper is to propose a Set Membership (SM) identification approach allowing us to mitigate the aforementioned issues. In particular, the nonlinear Set Membership method presented in Milanese and Novara (2004), and applied in Canale et al. (2006, 2009) to derive offline an approximation and tight bounds of the optimal NMPC control law, is here used online to: (i) reduce the dimension of the search domain of the NMPC optimization algorithm; (ii) shrink the volume of the search domain; (iii) warm start the algorithm. These operations allow a significant shortening of the computation time, thus enabling the NMPC real-time implementation, also in situations where a high sampling rate is required. The resulting NMPC approach, enhanced by the SM identification method, is called Reduced Complexity NMPC (RC-NMPC). This approach can be used in combination with any optimization algorithm to enhance its numerical efficiency, since it is not tailored to a specific strategy. The idea of using approximating functions to obtain a volume reduction and a warm start was first introduced in Boggio et al. (2023a,b). However, these works did not include the dimension reduction technique proposed in the present paper, which represents a novel contribution to the existing NMPC literature.

Another contribution of the paper consists in a simulated case study, where an obstacle avoidance system based on the RC-NMPC strategy is designed and tested in a realistic autonomous vehicle scenario. The performance of this approach is compared both with a standard NMPC and with the NMPC developed in Boggio et al. (2023a,b), hereinafter called Reduced Volume NMPC (RV-NMPC).

⋆ 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.2024.08.531

Notably, the results demonstrate significant computational time improvements with respect to both the standard NMPC and the RV-NMPC. These outcomes validate the rationale behind implementing additional algorithmic modifications to the previously developed RV-NMPC. Moreover, concerning the quality of the solution found, all three strategies yield similar results.

The paper is organized as follows. Section 2 introduces the mathematical formulation of NMPC. In Section 3, the Nonlinear Set Membership Approximation is described. Section 4 presents the developed RC-NMPC approach. The obtained results and the comparison with respect to standard NMPC and RV-NMPC are shown in Section 5. Finally, the conclusions are drawn in Section 6.

2. NONLINEAR MODEL PREDICTIVE CONTROL

Consider a Multiple-Input-Multiple-Output (MIMO) nonlinear continuous-time dynamic system characterized by the following 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; $f : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_y}$ are two functions defining the system dynamics and output variables, respectively. It is assumed that the state is measured in real-time at discrete intervals with a sampling time T_s, given by:

$$x(t_k),\ t_k = T_s k,\ k = 0, 1 \ldots. \tag{2}$$

In cases where the state cannot be directly measured, an observer or a model of (1) in input-output form needs to be employed.

NMPC is built upon two fundamental 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 referred to as the *prediction horizon*. The prediction is obtained by integrating the equations (1). For any time instant τ within the time interval $[t, t + T_p]$, the predicted output $\hat{y}(\tau)$ is a function of the "initial" state $x(t)$ and the input signal, as expressed by the equation:

$$\hat{y}(\tau) \equiv \hat{y}(\tau, x(t), u(t : \tau)) \tag{3}$$

where $u(t : \tau)$ represents a generic input signal within the interval $[t, \tau]$. The fundamental concept behind NMPC (as well as most predictive approaches) involves the search for an input signal denoted as $u^*(t : \tau)$, at each time instant $t = t_k$, such that the prediction $\hat{y}(\tau, x(t), u^*(t : \tau))$ exhibits the desired behavior in the time interval $[t, t+T_p]$. The notion of desired behavior is formalized by introducing the *objective function*, defined as

$$J(u(t : t + T_p)) \doteq \int_t^{t+T_p} \left(\|e_p(\tau)\|_Q^2 + \|u(\tau)\|_R^2\right) d\tau + \|e_p(t + T_p)\|_P^2 \tag{4}$$

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. As an example, when Q is a positive definite weight matrix, the norm of a column vector w is formally defined as $\|w\|_Q^2 \doteq w^\top Q w$.

The input signal $u^*(t : t + T_p)$ is chosen by minimizing the objective function $J(u(t : t + T_p))$. Specifically, at each time instant $t = t_k$, for $\tau \in [t, t + T_p]$, the following nonlinear OCP is solved:

$$\begin{aligned}&u^*(t : t + T_p) = \arg\min_{u(\cdot)} J(u(t : t + T_p))\\ &\text{subject to:}\\ &\quad \dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)),\ \hat{x}(t) = x(t)\\ &\quad \hat{y}(\tau) = h(\hat{x}(\tau), u(\tau))\\ &\quad \hat{x}(\tau) \in X_c,\ \hat{y}(\tau) \in Y_c,\ u(\tau) \in U_c\end{aligned} \tag{5}$$

The first two constraints in this problem guarantee that the predicted state and output are consistent with the system equations (1). The sets X_c and Y_c encompass additional constraints that may hold for the predicted state/output, such as those associated with obstacles or barriers. The set U_c accounts for input constraints, including input saturation.

The optimization problem (5) is typically non-convex. Moreover, optimizing a function with respect to a signal, as is the case with $u(\cdot)$, can be inherently challenging. To overcome this issue, the prediction interval $[t_k, t_k + T_p]$ can be divided into sub-intervals $[t_k + \tau_i, t_k + \tau_{i+1}] \subseteq [t_k, t_k + T_p]$, $i \in \{1, 2, \ldots, n_s\}$, where the τ_i's are called the nodes, and u and r remain constant on each sub-interval. Hence, we use the notation u_{ki} and r_{ki} to represent the command and reference values at time k in the ith sub-interval, and the reference sequence for the prediction interval is denoted as $r_k \doteq (r_{k1}, \ldots, r_{kn_s})$. This approach transforms the optimization problem into a finite-dimensional problem, which can be efficiently solved using numerical optimization algorithms.

The NMPC closed-loop command is obtained according to a so-called *receding horizon strategy* (RHS). At each time instant $t = t_k$, the input signal $u^*(t : t + T_p)$ is computed by solving (5). Subsequently, only the first optimal input value, denoted as $u(\tau) = u^*(t_k)$, is applied to the plant and kept constant $\forall \tau \in [t_k, t_{k+1}]$. This complete procedure is repeated at the subsequent time steps $t = t_{k+1}, t_{k+2} \ldots$.

3. NONLINEAR SM IDENTIFICATION

Finding the optimal NMPC command in real-time can be computationally expensive, as it requires solving a non-trivial optimization problem. To address this issue, an approximation and tight bounds of the NMPC control law are derived using the Nonlinear Set Membership (SM) Identification method of Milanese and Novara (2004).

According to the formulation of Section 2, the NMPC control law is a static nonlinear function $\phi(\cdot)$ of the regressor $w_k \doteq (x_k, r_k)$, where $x_k \doteq x(t_k)$ is the current state and $r_k \doteq (r_{k1}, \ldots, r_{kn_s})$ is the reference sequence. The NMPC command u_{ki} at time t_k in the ith prediction sub-interval is thus given by

$$u_{ki} = \phi(w_k). \tag{6}$$

For simplicity, in this section we assume that $n_u = 1$. The generalization to the case $n_u > 1$ is trivial and can be accomplished by applying the SM method to each component of u_k, see also the paragraph "RC-NMPC online algorithm" in Section 4. In general, due to the complexity of the OCP (5), it is not possible to write the function ϕ in closed-form. To overcome this issue, we derive an approximation of ϕ, based on the offline computation of its values at a given number of points.

Let $\mathbb{W} \subset \mathbb{R}^n$, $n = n_x + n_s n_y$, be a bounded region where the regressor w_k can evolve, and assume that the function ϕ is Lipschitz continuous on $\mathbb{W}$. A number M of values of ϕ are generated by solving offline the OCP (5), considering different values $\tilde{w}_k \in \mathbb{W}$, $k = 1, \ldots, M$, so that

$$\tilde{u}_k = \phi(\tilde{w}_k), \qquad k = 1, \ldots, M, \tag{7}$$

where the tilde is used to indicate the collected data. From these values of $\tilde{u}_k$ and $\tilde{w}_k$, the known properties of ϕ, and the input limitations $\underline{u} \leq \tilde{u}_k \leq \overline{u}$, an *approximation* of ϕ and tight *function bounds* are derived using the nonlinear SM approach of Milanese and Novara (2004). These functions will be key elements of the NMPC method proposed in Section 4.

The nonlinear SM approach of Milanese and Novara (2004) is now briefly summarized (in particular, its "local" version is presented here). Let us define the following functions:

$$\begin{aligned}\overline{b}(H,\gamma,w) &\doteq \min[\overline{u}, \min_{k=1,\ldots,M}(h_k + \gamma\|(w-\tilde{w}_k)\|)]\\ \underline{b}(H,\gamma,w) &\doteq \max[\underline{u}, \max_{k=1,\ldots,M}(h_k - \gamma\|(w-\tilde{w}_k)\|)]\end{aligned} \tag{8}$$

where $H = \{h_k\}_{k=1}^M$, $h_k \in \mathbb{R}$, $\gamma \in \mathbb{R}$ and $w \in \mathbb{W}$. Define also the functions

$$\begin{aligned}\phi^g(w) &\doteq \left(\overline{b}(H_\phi,\gamma_\phi,w) + \underline{b}(H_\phi,\gamma_\phi,w)\right)/2\\ \overline{\phi}(w) &\doteq \phi^g(w) + \overline{b}(H_\Delta,\gamma_\Delta,w)\\ \underline{\phi}(w) &\doteq \phi^g(w) + \underline{b}(H_\Delta,\gamma_\Delta,w)\\ \phi^c(w) &\doteq \left(\overline{\phi}(w) + \underline{\phi}(w)\right)/2\end{aligned} \tag{9}$$

where $H_\phi \doteq \{\tilde{u}_k\}_{k=1}^M$, $H_\Delta \doteq \{\tilde{u}_k - \phi^g(\tilde{w}_k)\}_{k=1}^M$; γ_ϕ and γ_Δ are the Lipschitz constants of ϕ and $\phi - \phi^g$ on $\mathbb{W}$, respectively. These constants can be systematically estimated using the validation procedures in Milanese and Novara (2004) and Canale et al. (2006). The following theoretical properties are proven 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 guaranteed from the data and the available prior information on the function. (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.

4. REDUCED COMPLEXITY NMPC

This section outlines how the computational complexity of the NMPC algorithm is reduced by using the nonlinear SM identification method. The following steps summarize the procedure.

Data Collection. Offline simulations are performed, either in open-loop or closed-loop. Using Monte Carlo campaign simulations, a set of state data $\tilde{x}_k$ and reference values $\tilde{r}_k$, with $k = 1,\ldots,M$, are generated, and the regressor $\tilde{w}_k \doteq (\tilde{x}_k, \tilde{r}_k)$ is formed. For each $\tilde{w}_k$, the corresponding optimal control command is computed, on the basis of (5), giving rise to a set of command data $\tilde{u}_k$, $k = 1,\ldots,M$.

Clustering. A clustering procedure is carried out to reduce the number of data used to derive the SM approximation of the NMPC control law ϕ. The *K-Medoids* approach is used, applying the CLustering LARge Applications (CLARA) algorithm to deal with large data sets (see Kaufman and Rousseeuw (2009)). At the end of the clustering analysis, a reduced database is obtained, given by the set of medoids. The size of this database is reduced by at least 10 times with respect to the original one. This means that $K \leq \frac{M}{10}$, where K is the number of medoids (and corresponding clusters). The set of medoids is used to identify the function ϕ by means of the SM approach.

Set Membership Approximation. Following the clustering process, the resulting dataset comprises K regressors $\tilde{w}_{mk}$ and commands $\tilde{u}_{mk}$, where the subscript m denotes the medoids of the clusters identified in the previous step. Using the data $\tilde{w}_{mk}$, $\tilde{u}_{mk}$, $k = 1,\ldots,K$, the SM approach described in Section 3 is employed to compute the optimal bounds $\overline{\phi}$ and $\underline{\phi}$, as well as the approximated control law ϕ^c. If the command u is multi-dimensional, and u and r are non constant (over the prediction horizon), the SM approach is applied to each component of $\tilde{u}_{mk}$ and for each sub-interval of the entire prediction time interval.

RC-NMPC online algorithm. As discussed in Section 2, in order to make the optimization problem (5) numerically tractable, the prediction interval $[t_k, t_k + T_p]$ is divided into sub-intervals $[t_k + \tau_i, t_k + \tau_{i+1}] \subseteq [t_k, t_k + T_p]$, $i \in \{1, 2, \ldots, n_s\}$, where the τ_i's are called the nodes. Then, u and r are assumed constant on each sub-interval. In particular, u_{ki} and r_{ki} denote their values at time k in the ith sub-interval. Similarly, ϕ_i^c, $\overline{\phi}_i$ and $\underline{\phi}_i$ denote the SM optimal approximation and bounds of the NMPC command in the ith sub-interval. If the command is of dimension $n_u > 1$, then ϕ_i^c, $\overline{\phi}_i$ and $\underline{\phi}_i$ are vectors with components ϕ_{ji}^c, $\overline{\phi}_{ji}$ and $\underline{\phi}_{ji}$, $j = 1,\ldots,n_u$. Each of these components is obtained using the SM approximation method described in Section 3. The RC-NMPC online algorithm is formally presented below (Algorithm 1).

Algorithm 1 RC-NMPC online algorithm.

Input: x_k, $r_k \doteq (r_{k1},\ldots,r_{kn_s})$
Output: $u_k = u(\tau)$, $\tau \in [t_k, t_k + \tau_1]$

1: Define $U_1 \doteq \prod_j U_{j1}$

$$U_{j1} \doteq \{\mathfrak{u} \in \mathbb{R} : \underline{\phi}_{j1}(w_k) \leq \mathfrak{u} \leq \overline{\phi}_{j1}(w_k)\}$$

where $w_k \doteq (x_k, r_k)$ and $\prod_j$ is the Cartesian product.

2: Solve the following OCP:

$$u_k = \arg\min_{u_1 \in U_1} J(u)$$

subject to:

$$\begin{aligned}&\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)),\ \hat{x}(t) = x(t)\\ &\hat{y}(\tau) = h(\hat{x}(\tau), u(\tau))\\ &\hat{x}(\tau) \in X_c,\ \hat{y}(\tau) \in Y_c\\ &u = (u_1, \phi_2^c(\tilde{w}_k), \ldots, \phi_{n_s}^c(\tilde{w}_k))\end{aligned} \tag{10}$$

where the initial guess of u_1 is $\phi_1^c(\tilde{w}_k)$.

The key features of the algorithm are now presented.

Dimension Reduction. According to the RHS strategy, the first optimal input value is the most important decision variable, since it is the one actually applied to the plant. The idea is thus to reduce the number of variables to be optimized to those of the first node u_1, keeping the others fixed and equal to the optimal values computed through the approximated control law $\phi_2^c,\ldots,\phi_{n_s}^c$. This reasoning leads to the formulation of the new OCP (10). Ongoing research activity is dedicated to develop a method to optimally choose the nodes to consider for optimization. In standard NMPC and RV-NMPC algorithms, the number of decision variables is proportional to the number of nodes in which the prediction interval is divided (see Section 2). Since the computational complexity of Quasi-Newton methods based on single shooting algorithms is proportional to $O((n_s n_u)^2)$, increasing the number of nodes results in a quadratic worsening of the performance of both algorithms. Instead, for the RD-NMPC, where $n_s = 1$, this complexity remains constant with the number of nodes and only proportional to n_u. Future investigations will explore possible benefits for multiple shooting approaches.

Volume Reduction. The input constraint set U_1 is defined by the optimal bounds $\underline{\phi}_{j1}(w_k)$ and $\overline{\phi}_{j1}(w_k)$, which produce a shrinking of the volume of U_1. As shown in the example presented below, this shrinking is quite relevant. This results in fewer cost function evaluations, thus shortening the computation time required to find u_k. This type of volume reduction is not typically implemented in standard NMPC algorithms.

Warm Start. The optimization algorithm is initialized with the optimal initial condition $\phi_1^c(\tilde{w}_k)$ which is computed using the SM approximated control law. This can help accelerate convergence, since it provides initial values closer to the optimal solutions. In contrast, many standard NMPC algorithms employ the so-called *shift initialization* strategy. This technique sets the initial guess for the decision variables in the current optimization problem to match the solution obtained in the preceding time step. Shift initialization is particularly advantageous when the changes between consecutive time steps are small or when the system operates within similar regimes over successive iterations. However, if the system dynamics change abruptly or if the current optimal solution significantly deviates from the previous one, initializing the optimization problem with the previous solution may lead to poor suboptimal solution.

5. CASE STUDY: OBSTACLE AVOIDANCE FOR AUTONOMOUS VEHICLES

In the last decades, a remarkable amount of research and progress has been made towards creating intelligent technologies for autonomous vehicles. Significant developments include adaptive cruise control, lane-keeping assist, and decision-making algorithms. These technologies are designed with a fundamental objective: enabling autonomous vehicles to track given reference trajectories, efficiently avoiding obstacles and ensuring safety. Indeed, as vehicles become increasingly autonomous, the need for effective obstacle avoidance mechanisms becomes crucial, with these systems playing an essential role in enhancing the safety of passengers, pedestrians, and other road users.

In this context, we consider the scenario depicted in Fig. 2, featuring a rural road with two lanes, one for each direction, along with an obstacle at the exit of a curve due to roadworks and a truck traveling in the opposite direction on the other lane. The vehicle's objective is to avoid the construction site and return to its lane before the arrival of the truck. This is done by designing safety ellipses around the obstacles as follows:

$$(x - x_{v_1})^2/a_1^2 + (y - y_{v_1})^2/b_1^2 \geq 1, \tag{11}$$

$$(x - x_{v_2})^2/a_2^2 + (y - y_{v_2})^2/b_2^2 \geq 1, \tag{12}$$

where equation (11) describes the roadworks obstacle centered at x_{v_1} and y_{v_1}, with major and minor axes represented by a_1 and b_1, respectively, represented with the red ellipse in Fig. 2. On the other hand, equation (12) pertains to the obstacle posed by the truck, represented with the blue ellipse in Fig. 2. In order to create a critical scenario for the vehicle, forcing it to return promptly to its lane, the truck is considered to be in motion at a speed of 20 km/h. To account for this during optimization, a prediction of its position is performed, instant by instant, by using a rough estimate of the truck velocity. This implies that the center characterized by x_{v_2} and y_{v_2} is not fixed but changes at every time instant. Note that equations (11)-(12) will be included as constraints within the optimization problems (5) and (10).

5.1 Vehicle models

Two models were developed/used.

Plant model. For simulating the real vehicle, the Matlab Dual-Track Vehicle Body 3DOF block described in MATLAB (2018) is used. The block implements a rigid two-axle vehicle body model that computes longitudinal, lateral, and yaw motion. It accounts for body mass, aerodynamic drag, and weight distribution between the axles due to acceleration and steering. The main physical parameters of the model and their values are as follows: mass m = 1575 kg; moment of inertia $I_z = 4000\,\text{kg*m}^2$; distances CoG-front/rear wheels $l_f = 1.2$ m and $l_r = 1.6$ m, respectively. The aerodynamic parameters and other details can be found in MATLAB (2018). The main variables of the model are the following: X and Y are the coordinates of the vehicle in an inertial frame, ψ is the yaw angle, v_x and v_y are the longitudinal and lateral speeds, respectively, and ω is the yaw rate. The command inputs of the model are the vehicle longitudinal acceleration a_x and the steering angle δ_f.

NMPC internal prediction model. A classical Dynamic Single-Track (DST) Model is used in the NMPC optimization algorithm to predict the future behavior of the system. This model considers simplified equations of the lateral and longitudinal dynamics of a vehicle. The state equations of the DST model are as follows:

$$\begin{aligned}
\dot{X} &= v_x \cos\psi - v_y \sin\psi \\
\dot{Y} &= v_x \sin\psi + v_y \cos\psi \\
\dot{\psi} &= \omega \\
\dot{v}_x &= v_y\dot{\psi} + a_x \\
\dot{v}_y &= -v_x\dot{\psi} + 2\left(F_{yf} + F_{yr}\right)/m \\
\dot{w} &= 2\left(l_f F_{yf} - l_r F_{yr}\right)/I_z
\end{aligned} \tag{13}$$

where the variables and parameters are the same as those of the Plant. Note however that the prediction model is a simplified version of the plant, not accounting for aerodynamic forces and weight distribution between the axles due to acceleration and steering. F_{yf} and F_{yr} are the lateral forces between the wheels and the vehicle, given by $F_{yf} = -c_f\beta_f$, $F_{yr} = -c_r\beta_r$ where $c_f = 2.7\cdot10^4$ N/rad and $c_r = 2\cdot10^4$ N/rad are the front/rear cornering stiffnesses. The tire slip angles are defined as $\beta_f = \text{atan}\left(\frac{v_y + l_f\dot{\psi}}{v_x}\right) - \delta_f$ and $\beta_r = \text{atan}\left(\frac{v_y - l_r\dot{\psi}}{v_x}\right)$.

The model state is $x = (X, Y, \psi, v_x, v_y, \omega)$, while the longitudinal acceleration and the steering angle are the control variables: $u = (a_x, \delta_f)$. The model output is (X, Y). Note that, although simple, the DST model captures the main aspects of the vehicle dynamics and, for this reason, is suitable to be used inside NMPC algorithms.

5.2 Closed-loop system

A closed-loop system has been implemented in Simulink, consisting of the plant controlled in feedback by a NMPC block. The input of the plant is the command $u = (a_x, \delta_f)$ provided by the NMPC block, which computes this command on the basis of the plant state and the reference trajectory. The NMPC block can be either a standard NMPC, a RV-NMPC algorithm or a RC-NMPC algorithm. These three algorithms are described below.

5.3 Standard NMPC algorithm

A standard NMPC algorithm was designed, finalized at lateral and longitudinal control of the vehicle dynamics.

The algorithm is based on the optimization problem (5) and uses equations (13) as the internal prediction model. The inputs of the algorithm are the plant state and the reference trajectory. The output is the command $u = (a_x, \delta_f)$, used to track the reference trajectory, allowing the vehicle to accomplish the lane keeping task, and to avoid potential obstacles along the path, while maintaining a desired speed. This command was parametrized considering four nodes, i.e., $n_s = 4$, implying that, in the prediction interval $[t, t+T_p]$, there are a total of 8 command samples: 4 for the longitudinal acceleration a_x and 4 for the steering angle δ_f. The values of the NMPC parameters were chosen through a trial-and-error procedure and are listed in Table 1.

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	$[3\,\mathrm{m/s^2}, \pi/4, 3\,\mathrm{m/s^2}, \pi/4]$
Lower bounds	$[-3\,\mathrm{m/s^2}, -\pi/4, -3\,\mathrm{m/s^2}, -\pi/4]$

5.4 RC-NMPC and RV-NMPC algorithms

Data Collection. A campaign of 500 simulations of the closed-loop system with the standard NMPC algorithm was carried out, considering rural roads with slightly different curvatures and assuming a speed of 60 km/h. This simulation campaign provided a database of about $M = 2 \cdot 10^5$ samples. Note that here the database was constructed by means of closed-loop simulations but it is also possible to generate it in "open-loop", just by evaluating the output of the NMPC law for different values of the regressor.

Clustering. To reduce the size of the database obtained in the previous step a clustering procedure was used. The K-medoids clustering method with the CLARA algorithm was employed. A reduced set of 10^4 data was derived, giving an optimal compromise between amount of data, memory usage and exploration of the control law domain.

Set Membership Approximation. The clustering process reduced the database from $2 \cdot 10^5$ to 10^4 samples, which were collected in the set $\{\tilde{w}_{mk}, \tilde{u}_{mk}\}_{k=1}^{10^4}$. Using these data, the SM approach described in Section 3 was applied to derive the approximated control law ϕ^c, as well as the corresponding bounds $\overline{\phi}$ and $\underline{\phi}$. The optimal SM approximation and the bounds for one of the steering angle commands are shown in Fig. 1. It can be observed that the interval between the bounds is reduced more than 10 times compared to the original one.

RV-NMPC online algorithm. The RV-NMPC consists in solving the OCP (5) where the warm start $u_{start} = \phi^c(w_k)$ is used and the command domain is given by $U_c = \{u_k \in \mathbb{R}^{n_s n_u} : \underline{\phi}(w_k) \leq u \leq \overline{\phi}(w_k)\}$ where "$\leq$" are element-wise inequalities (see Boggio et al. (2023a,b) for more details).

RC-NMPC online algorithm. The RC-NMPC consists in applying online Algorithm 1.

In RV-NMPC and RC-NMPC, equations (13) are used as the internal prediction model. The design parameters are the same as those of the standard NMPC, see Table 1.

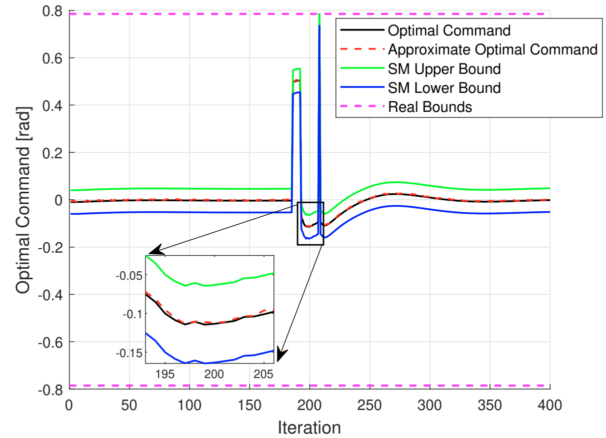

Fig. 1. Set Membership approximation of δ_f.

5.5 Comparison between the NMPC algorithms

A Monte Carlo (MC) campaign of 100 trials was carried out, considering rural roads with different curvatures compared to those used during the Data Collection phase. The reference velocity was 60 km/h.

The simulations were run on a Dell Precision 5820 (Processor: Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz). The optimization problems were solved using uniquely the Matlab function `fmincon` with the Sequential Quadratic Programming (SQP) algorithm. It should be noted that the proposed approach can be combined with any optimization algorithm to improve its numerical efficiency. However, the most significant advantages are observed when it is used with the single shooting method, where the optimization variables are only the command inputs u.

The performance of the three NMPC algorithms was compared using the following indexes:

(1) Number of evaluated cost functions (Eval. Cost. Funct.) for finding the minimum;
(2) NMPC total computational time (NMPC Time);
(3) Optimization computational time (Opt. Time);
(4) Root-Mean-Square (RMS) Error both for the Lateral Error (Lat.E.) and the Orientation Error (Orient.E.);
(5) Minimum distance from the obstacles (Min.Dist.).

Note that "NMPC total computational time" refers to the time elapsed between when the NMPC receives inputs and when it produces the commands. This means that this time takes into account not only the optimization part but also the entire pre-processing phase. Specifically, in the case of RC-NMPC and RV-NMPC, it includes also the use of the SM functions to obtain the warm start and the bounds. On the other hand, "Optimization computational time" only considers the time required by the solver (fmincon) to find the optimal command. Table 2 shows the mean value of the performance indexes for the three NMPC algorithms. The Mean Value represents the average number of evaluated cost functions, computational times, RMS errors and minimum distances throughout the Monte Carlo simulations. To compute the RMS error, only the reference tracking phase has been taken into consideration, discarding the overtaking part where the vehicle must move away from the reference. Indeed, to make the scenario challenging, the same reference, i.e., the center of the road, has always been considered, leaving the NMPC to autonomously avoid the obstacle. Note that this is more challenging than just imposing a reference change to obtain the overtaking. Clearly, the RMS, being referred to the center of the first road, becomes very large during the obstacle avoidance phase. Finally, the minimum distance from the obstacle highlights that throughout the entire overtaking maneuver, a safe distance was always maintained from all obstacles. Fig. 2 shows an example of an obstacle avoidance using the RC-NMPC algorithm.

Table 2. Comparison of the NMPC algorithms

	St-NMPC	**RV-NMPC**	**RC-NMPC**
	Mean Value	Mean Value	Mean Value
Eval. Cost Funct.	194.65	16.64	4.52
NMPC Time [s]	0.0926	0.0146	0.0091
Opt. Time [s]	0.0918	0.0102	0.0049
RMS Lat. E. [m]	0.2105	0.2179	0.2256
RMS Or. E. [rad]	0.0135	0.0137	0.0142
Min. Dist. [m]	2.814	2.806	2.792

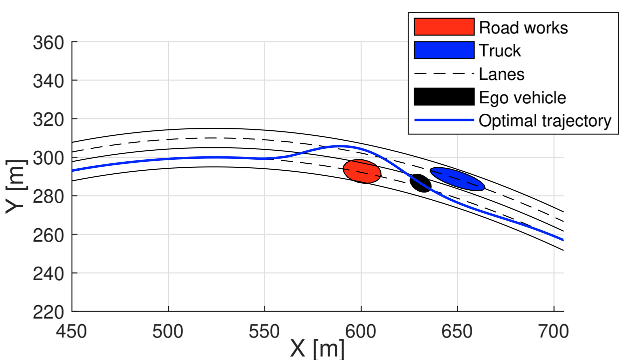

Fig. 2. Example of obstacle avoidance using RC-NMPC

From these results, we can observe that: i) The RC-NMPC reduces the number of function evaluations by a factor of approximately 50 compared to the standard NMPC and 4 with respect to the RV-NMPC. ii) The RC-NMPC improves the computational time of 5 ms compared to the RV-NMPC and about 10 times with respect to the Standard NMPC. The discrepancy between this result and the one obtained for the cost functions is due to the fact that the RC-NMPC algorithm requires the evaluation of the SM approximated control law before performing optimization, which is not present in the standard NMPC. This operation implies additional computation time. Research activities are currently being dedicated to reduce this additional time. iii) Regarding the times required to solve the OCP with fmincon, compared to the previous index, more pronounced improvements are obtained both with respect to the Standard NMPC (a factor of 20) and to the RV-NMPC (a factor of 2). In particular, a quadratic computational speedup is expected, see the paragraph “Dimension Reduction” in Section 4. iv) Finally, the RMS errors and the minimum distances are quite similar for all the NMPC algorithms.

6. CONCLUSIONS

The paper has proposed a data-aided approach to improve the numerical efficiency of NMPC algorithms, based on the Set Membership approximation method. The approach has been tested in a simulated case study, concerned with an obstacle-avoidance maneuver in an autonomous driving scenario. The obtained results demonstrate its effectiveness, in terms of computation time, with respect to both the standard NMPC and the RV-NMPC strategies.

REFERENCES

Biegler, L.T. (2000). Efficient solution of dynamic optimization and NMPC problems. In Allgöwer, F. and Zheng, A. (eds.), *Nonlinear Model Predictive Control*, 219–243. Birkhäuser Basel, Basel.

Bock, H.G., Diehl, M., Leineweber, D.B., and Schlöder, J.P. (1999). Efficient direct multiple shooting in nonlinear model predictive control. *Scientific Computing in Chemical Engineering II: Simulation, Image Processing, Optimization, and Control*, 2, 218–227.

Boggio, M., Novara, C., and Taragna, M. (2023a). Nonlinear model predictive control: an optimal search domain reduction. *IFAC-PapersOnLine*, 56(2), 6253–6258. 22nd IFAC World Congress.

Boggio, M., Novara, C., and Taragna, M. (2023b). Trajectory planning and control for autonomous vehicles: a “fast” data-aided NMPC approach. *European Journal of Control*, 100857.

Canale, M., Fagiano, L., and Milanese, M. (2009). Set membership approximation theory for fast implementation of model predictive control laws. *Automatica*, 45(1), 45–54.

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., Schlöder, J.P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4), 577–585.

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.

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

Pagone, M., Boggio, M., Novara, C., and Vidano, S. (2021). A pontryagin-based NMPC approach for autonomous rendez-vous proximity operations. In *2021 IEEE Aerospace Conference (50100)*, 1–9.

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

Siampis, E., Velenis, E., Gariuolo, S., and Longo, S. (2018). A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling. *IEEE Transactions on Control Systems Technology*, 26(6), 1982–1994.