Trajectory planning and control for autonomous vehicles: a “fast” data-aided NMPC approach

European Journal of Control 74 (2023) 100857

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

# Trajectory planning and control for autonomous vehicles: a "fast" data-aided NMPC approach☆

Mattia Boggio*, Carlo Novara, Michele Taragna

*Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino 10129, Italy*

A R T I C L E I N F O

*Article history:*
Received 15 May 2023
Accepted 8 June 2023
Available online 14 June 2023

Recommended by Prof. T Parisini

*Keywords:*
Autonomous systems
Predictive control for nonlinear systems
Computational methods

A B S T R A C T

A huge research effort is being spent worldwide by automotive companies and academic institutions for developing vehicles with high levels of autonomy, ranging from advanced driving-assisted systems to fully automated vehicles. Nonlinear Model Predictive Control (NMPC) has the potential to become a key technology in this context, thanks to its capability to deal with linear and nonlinear systems, manage physical constraints and satisfy multi-objective performance criteria. However, NMPC is based on the on-line solution of a nonconvex optimization problem and this operation may require a high computational cost, compromising its real-time implementation. In this paper, a "fast" data-aided NMPC approach is developed, aimed at trajectory planning and control for autonomous vehicles. In particular, a Set Membership approximation method is used to derive from data tight bounds on the optimal NMPC control law. These bounds are used to restrict the search domain of the underlying NMPC optimization process, allowing a significant reduction of the computation time. The proposed NMPC trajectory planning and control approach is tested in simulation and compared with other state-of-the-art methods, considering different road scenarios.



## 1. Introduction

Autonomous driving is considered one of the most ground-breaking technologies of the near future and is expected to completely reshape the sector of transportation systems (see, e.g., [7,9,13]). In this regard, a huge research effort is being spent worldwide by automotive companies and academic institutions for developing vehicles with high levels of autonomy, ranging from advanced driving-assisted systems to fully automated vehicles (see, e.g., [29,34]).

Modern control theory offers a multitude of approaches and design paradigms that can be exploited for these applications. Among them, Model Predictive Control (MPC) has the potential to become a key technology, thanks to its capability to design control algorithms for multivariable systems under state, input, and output constraints (see, e.g., [14,25,31,32]). To cope with nonlinear dynamics and constraints, as well as with nonconvex performance indexes, Nonlinear MPC (NMPC) has been introduced (see, e.g., [2,10] and references therein).

For both NMPC and MPC, fast and reliable optimization algorithms are needed, able to meet the hard time constraints of real-time closed-loop control applications. Serious issues may occur especially in the case of NMPC, where on-line solutions of nonconvex optimizations rely on the use of sophisticated algorithms with higher computational costs than linear MPC. During the past decades, significant progress has been carried out in reducing the computational complexity of NMPC approach. In [23], a method was developed for linearizing the nonlinear model around a nominal trajectory and then solving a unique Sequential Quadratic Programming (SQP) over the time horizon. In [11], a Real-Time Iteration (RTI) scheme was introduced that performs a single SQP iteration per sampling time. It uses the direct multiple shooting method of [5] for simultaneous Nonlinear Program (NLP) parametrization, with full derivatives and condensing. The implementation of the multi-level version of RTI is described in Bock et al. [4], allowing for further reduction of the computational load. Moreover, the continuation/GMRES of Ohtsuka [28] and the advanced-step controller of Zavala and Biegler [35] can be mentioned. These improvements have allowed the NMPC implementation also in real-time systems with high sampling rate requirements, see, e.g., [1,15,16].

☆ This work was supported by the NewControl project, within the Electronic Components and Systems For European Leadership Joint Undertaking (ECSEL JU) in collaboration with the European Union's Horizon 2020 Framework Programme and National Authorities, under grant agreement no. 826653-2.

* Corresponding author.
*E-mail address:* mattia.boggio@polito.it (M. Boggio).

https://doi.org/10.1016/j.ejcon.2023.100857
0947-3580/

a rigid two-axle vehicle body model to calculate longitudinal, lateral, and yaw motion. The block accounts for body mass, aerodynamic drag, and weight distribution between the axles due to acceleration and steering. The NMPC has been used to address the autonomous trajectory planning and control of the described scenarios. This approach is formulated in detail in the following section.

## 3. Nonlinear model predictive control

Consider a Multiple-Input-Multiple-Output (MIMO) nonlinear dynamic 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: $x(t_k),\ t_k = T_s k,\ k = 0, 1 \ldots$. If the state is not measured, an observer or a model of (1) in input-output form has to be employed.

NMPC 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 a 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, $r(\tau) \in \mathbb{Y} \subset \mathbb{R}^{n_y}$ is a reference to track, $\mathbb{Y}$ is a bounded set, and $\|\cdot\|_*$ is a weighted Euclidean norm. For example, letting $Q$ be a positive definite weight matrix, the norm of a column vector $w$ is defined as $\|w\|_Q^2 \doteq w^\top Q w$.

The input signal $u^*(t:t+T_p)$ is chosen as one minimizing the objective function $J\big(u(t:t+T_p)\big)$. In particular, at each time $t = t_k$, for $\tau \in [t, t+T_p]$, the following nonlinear Optimization Control Problem (OCP) is solved:

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

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

The optimization problem (4) is generally nonconvex. Moreover, the decision variable $u(\cdot)$ is a signal, and optimizing a function with respect to a signal is generally a difficult task. To overcome this problem, the prediction interval $[t_k, t_k + T_p]$ can be divided into $n_s$ sub-intervals $[t_k + \tau_i, t_k + \tau_{i+1}] \subset [t_k, t_k + T_p]$, $i \in \{1, 2, \ldots, n_s\}$, where the $\tau_i$'s are called the nodes, and $u$ and $r$ can be kept constant on each sub-interval. Hence, $u_{ki}$ and $r_{ki}$ denote the command and reference values at time $k$ in the $i$th sub-interval, respectively. The command and reference sequences in the prediction interval are indicated with $u_k \doteq (u_{k1}, \ldots, u_{kn_s})$ and $r_k \doteq (r_{k1}, \ldots, r_{kn_s})$, respectively. In this way, the optimization problem reduces to a finite-dimensional problem, which can be solved using an efficient numerical optimization algorithm.

The NMPC closed-loop command is obtained according to a so-called *receding horizon strategy*. 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, keeping it constant for $\forall \tau \in [t_k, t_{k+1}]$. The complete procedure is repeated at the next time steps $t = t_{k+1}, t_{k+2} \cdots$.

In the reminder of the paper, it is assumed that, for some choice of the parameters $T_s, T_p, Q, R, P$, the NMPC algorithm defined by (4), applied according to the receding horizon strategy to the plant (1), provides a bounded tracking error $e(t) \doteq r(t) - y(t)$, for all $t \geq 0$ and for every reference signal $r$ such that $r(t) \in \mathbb{Y}$, $\forall t \geq 0$.

## 4. Set Membership approximation and tight guaranteed bounds

The optimal NMPC command must be computed in real-time and this task may require a large computational time, since a non-trivial optimization problem has to be solved. In order to overcome this problem, an approximation of the NMPC control law is derived, based on the nonlinear Set Membership (SM) Identification method. This method is now summarized.

According to the formulation of Section 3, the optimal NMPC command $u_k \doteq u^*(t_k)$ is a static nonlinear function of the current state $x_k \doteq x(t_k)$ and the reference sequence $r_k$. The NMPC command $u_{ki}$ at time $t_k$ in the $i$th prediction sub-interval is thus given by

$$u_{ki} = \phi(w_k) \tag{5}$$

where $w_k \doteq (x_k, r_k)$ and $\phi$ is a static nonlinear function. For simplicity, in this section $n_u = 1$ is assumed. The generalization to the case $n_u > 1$ is trivial and can be accomplished by applying the SM method to each component of $u_{ki}$. See also Section 5.2. In general, due to the complexity of the OCP (4), writing the function $\phi$ in closed-form is not possible. To overcome this issue, an approximation of $\phi$ is derived, based on the off-line computation of its values at a given number of points, using the nonlinear SM approach of Milanese and Novara [27].

Let $\mathbb{W} \subset \mathbb{R}^{n_x+n_y}$ be a region where the regressor $w_k$ can evolve, and assume that the function $\phi$ is Lipschitz continuous on $\mathbb{W}$. Note that this region is bounded since the NMPC algorithm is assumed to guarantee a bounded tracking error and the reference is assumed bounded. A number $M$ of values of $\phi$ is generated by solving off-line the OCP (4), starting from different initial conditions $\tilde{w}_l \in \mathbb{W}$, so that

$$\tilde{u}_l = \phi(\tilde{w}_l), \qquad l = 1, \ldots, M, \tag{6}$$

where the tilde is used to indicate the collected data. From these values of $\tilde{u}_l$ and $\tilde{w}_l$, the known properties of $\phi$, and the input limitations $\underline{u} \leq \tilde{u}_l \leq \overline{u}$, an *approximation* of $\phi$ and tight *function bounds* are derived using the nonlinear SM approach. These functions will be key elements of the NMPC method proposed in Section 5.

The nonlinear SM approach of Milanese and Novara [27] is now briefly summarized (in particular, its “local” version is presented here). Let us define the following functions:

$$\begin{aligned} \overline{b}(H, \gamma, w) &\doteq \min[\overline{u}, \min_{l=1,\ldots,M} (h_l + \gamma \|(w - \tilde{w}_l)\|)] \\ \underline{b}(H, \gamma, w) &\doteq \max[\underline{u}, \max_{l=1,\ldots,M} (h_l - \gamma \|(w - \tilde{w}_l)\|)] \end{aligned} \tag{7}$$

where $H = \{h_l\}_{l=1}^{M}$, $h_l \in \mathbb{R}$, $\gamma \in \mathbb{R}$ and $w \in \mathbb{W}$ are the independent variables of the functions. Define now the functions

$$\begin{aligned} \phi^g(w) &\doteq \left(\overline{b}(H_\phi, \gamma_\phi, w) + \underline{b}(H_\phi, \gamma_\phi, w)\right)/2 \\ \overline{\phi}(w) &\doteq \phi^g(w) + \overline{b}(H_\Delta, \gamma_\Delta, w) \\ \underline{\phi}(w) &\doteq \phi^g(w) + \underline{b}(H_\Delta, \gamma_\Delta, w) \\ \phi^c(w) &\doteq \left(\overline{\phi}(w) + \underline{\phi}(w)\right)/2 \end{aligned} \quad (8)$$

where $H_\phi \doteq \{\tilde{u}_l\}_{l=1}^{M}$, $H_\Delta \doteq \{\tilde{u}_l - \phi^g(\tilde{w}_l)\}_{l=1}^{M}$, $\gamma_\phi$ and $\gamma_\Delta$ are the Lipschitz constants of $\phi$ and $\phi - \phi^g$ on $\mathbb{W}$, respectively. These constants can be systematically estimated using the validation procedure in Milanese and Novara [27].

The following theoretical properties are proven in Milanese and Novara [27]:

- The functions $\overline{\phi}$ and $\underline{\phi}$ are *optimal bounds* of $\phi$: they are the tightest upper and lower bounds that can be derived from the available prior information on the function and the data.
- The function $\phi^c$ is an *optimal approximation* of $\phi$: 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.

## 5. Set Membership nonlinear model predictive control

This section describes how the nonlinear SM identification method is exploited to improve the computational performance of an NMPC algorithm. In the first subsection, the off-line SM-NMPC design procedure is described. In the second one, the on-line algorithm is presented. A preliminary version of this approach, called Reduced Domain NMPC, can be found in Boggio et al. [6].

### 5.1. SM-NMPC off-line design procedure

*Data collection.* To describe the evolution of the system, several off-line simulations are performed. In particular, a set of state data $\tilde{x}_l$ and reference signals $r_l$, with $l = 1 \dots, M$, are generated, collecting the regressor $\tilde{w}_l \doteq (\tilde{x}_l, r_l)$. Starting from each $\tilde{w}_l$, the corresponding optimal control command is computed, on the basis of (4), giving rise to a set of control data $\tilde{u}_l$. The resulting dataset is thus given by $\{\tilde{w}_l, \tilde{u}_l\}_{l=1}^{M}$.

*Clustering.* Since, in general, the number $M$ of collected data can be very large, firstly a clustering process is performed using the *K-Medoids* approach [21]. This method uses the medoids to represent the clusters. A medoid is an element of the dataset whose sum of dissimilarities to all the elements in the cluster is minimal. Among many algorithms for $K$-medoids clustering, due to the large dataset, CLustering LARge Applications (CLARA) [20] is used.

At the end of the clustering process, the size of the dataset must be reduced by at least 10 times. This means that $K \le \frac{M}{10}$, where $K$ is the number of clusters and then the number of data used to identify the function $\phi$. The resulting dataset, that best characterizes the overall system, is $\{\tilde{w}_{ml}, \tilde{u}_{ml}\}_{l=1}^{K}$, composed of $K$ regressors $\tilde{w}_{ml}$ and commands $\tilde{u}_{ml}$. The subscript $m$ is used to indicate that the data are the medoids of the clusters found in this step.

*Set Membership approximation.* On the basis of the dataset $\{\tilde{w}_{ml}, \tilde{u}_{ml}\}_{l=1}^{K}$, the optimal bounds $\overline{\phi}$ and $\underline{\phi}$, and approximated control law $\phi^c$ are computed by means of the SM approach [27]. If the command $u$ is multi-dimensional, and $u$ and $r$ are not constant (over the prediction horizon), the SM approach is applied to each component of $\tilde{u}_{ml}$ and for each sub-interval of the entire prediction time interval.

*Summary of the off-line procedure.* The off-line steps of the SM-NMPC design procedure are summarized in Algorithm 1.

**Algorithm 1** SM-NMPC off-line algorithm.

**Input:** Model of the plant (1); NMPC parameters $T_p, Q, R, P$.
**Output:** $\overline{\phi}$, $\underline{\phi}$ and $\phi^c$.

1: Several off-line simulations are carried out to generate the dataset $\{\tilde{w}_l \doteq (\tilde{x}_l, r_l)\}_{l=1}^{M}$.
2: For each $\tilde{w}_l$, the optimal control command $\tilde{u}_l$ is computed by solving the NMPC optimization problem (4) off-line, thus obtaining the design dataset $\{\tilde{w}_l, \tilde{u}_l\}_{l=1}^{M}$.
3: *K-Medoids* clustering is applied to reduce the size of the design dataset from $M$ to $K \le \frac{M}{10}$.
4: On the basis of the reduced dataset, $\overline{\phi}$, $\underline{\phi}$ and $\phi^c$ are derived according to (8).

### 5.2. SM-NMPC on-line algorithm

As discussed in Section 3, in order to make the optimization problem numerically tractable, the prediction interval $[t_k, t_k + T_p]$ is divided into $n_s$ sub-intervals $[t_k + \tau_i, t_k + \tau_{i+1}] \subset [t_k, t_k + T_p]$, $i \in \{1, 2, \dots, n_s\}$, where the $\tau_i$'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 $i$th sub-interval. Similarly, $\phi_i^c$, $\overline{\phi}_i$ and $\underline{\phi}_i$ denote the SM optimal approximation and bounds of the optimal command in the $i$th sub-interval. If the command is of dimension $n_u > 1$, then $\phi_i^c$, $\overline{\phi}_i$ and $\underline{\phi}_i$ are vectors with components $\phi_{ji}^c$, $\overline{\phi}_{ji}$ and $\underline{\phi}_{ji}$, $j = 1, \dots, n_u$. Each of these components is obtained using the SM approximation method described in Section 4. The SM-NMPC on-line algorithm is formally presented below (Algorithm 2).

**Algorithm 2** SM-NMPC on-line algorithm, applied at each time $t_k$.

**Input:** $x_k$, $r_k \doteq (r_{k1}, \dots, r_{kn_s})$.
**Output:** $u(\tau)$, $\tau \in [t_k, t_{k+1}]$.

1: For $i = 1, \dots, n_s$ and $j = 1, \dots, n_u$, define the interval
$$U_{ji} \doteq [\underline{\phi}_{ji}(w_k), \overline{\phi}_{ji}(w_k)]$$
where $w_k \doteq (x_k, r_k) \in \mathbb{R}^{n_x + n_y n_s}$.
2: Solve the OCP (4) with:
   1. $u(\tau) = u_{ki}$, $\tau \in [t_k + \tau_i, t_k + \tau_{i+1}]$, $i = 1, \dots, n_s$.
   2. Warm start command sequence: $u_0 = \phi^c(w_k) = (\phi_1^c(w_k), \dots, \phi_{n_s}^c(w_k))$.
   3. Reduced search domain: $U_c = \prod_{ji} U_{ji}$, where $\prod_{ji}$ denotes the Cartesian product.

3: Set the optimal command as $u(\tau) = u_{k1}^*$, $\tau \in [t_k, t_{k+1}]$, where $u_{k1}^*$ is the first sample of the OCP solution $u_k^*$.

The main features of the on-line algorithm are now discussed.

#### 5.2.1. Search domain reduction

Each input constraint set $U_{ji}$ is defined by the optimal bounds $\underline{\phi}_{ji}(w_k)$ and $\overline{\phi}_{ji}(w_k)$, that shrink the initial search domain $U_c$. This leads to a reduction of the number of cost function evaluations and consequently to a shortening of the computation time needed to find $u_k$. Such a search domain reduction is not operated in standard NMPC algorithms.

#### 5.2.2. Warm start

The optimization algorithm is warm-started using the optimal initial condition $\phi^c(w_k)$, computed through the SM approximated control law. On the other hand, many standard NMPC algorithms use the so-called *shift initialization* strategy, where the starting values of the decision variables are taken equal to the solution obtained in the previous time step. This latter strategy works well if the new optimal solution is not far from the previous one, but may lead to non-satisfactory local solutions if this condition does not hold.

### 5.3. General considerations on the SM-NMPC approach

#### 5.3.1. Presence of disturbances/uncertainties

On one hand, the computation of the optimal SM approximation and bounds is not affected by any kind of disturbance or uncertainty. Indeed, they are computed from a set of data generated solving the NMPC optimization problem off-line. This data-generation mechanism is fully deterministic and not affected by disturbances/uncertainties. On the other hand, the effects of disturbances or uncertainties may show up when NMPC is used to control in closed-loop some plant, but this effect is exactly the same using an NMPC algorithm with the SM domain reduction or a pure NMPC algorithm without the reduction. The SM domain reduction can be used in combination with any NMPC algorithm to increase the performance in terms of computational speed but it leaves unchanged all the control performance of the original NMPC algorithm.

#### 5.3.2. Data used for SM-NMPC design

The SM bounds are defined on the whole domain of the NMPC algorithm and thus they hold true in all possible working conditions of the plant. However, the amplitude of the command range defined by the bounds depends on how the collected data are distributed in the NMPC domain: the amplitude is smaller in regions where the data are more "densely" distributed, larger in regions not "densely" explored by the data. Hence, in order to obtain a significant shrink of the range, and a subsequent reduction of the NMPC computation time for all the driving conditions of interest, it is necessary to collect data that explore the regions corresponding to these conditions. A "learning" version of the SM-NMPC algorithm could also be developed, where the data are collected online and the SM bounds and approximation are improved at each time step.

## 6. Autonomous driving simulation results

In this section, the presented approach is validated and compared with a standard NMPC implementation considering the road scenarios described in Section 2.

### 6.1. Parallel parking

To show the effectiveness of the proposed method, the SM-NMPC algorithm has been tested in simulation on a real road scenario regarding a Parallel Parking maneuver.

The models used to describe the ego vehicle are first introduced. When using the NMPC approach, it is necessary to distinguish between two models: a "high-fidelity" plant model, used to simulate the real vehicle, and a prediction model, used within the NMPC optimization algorithm to predict the future behavior of the system (this latter model is typically simpler than the former).

As mentioned in Section 2, to simulate the real vehicle, the Matlab Dual-Track Vehicle Body 3DOF block is used. Regarding the NMPC prediction model, the classical kinematic bicycle equations are considered. Since the vehicle travels at low speed, these equations provide a sufficiently accurate description of the vehicle motion. The kinematic bicycle model is the following:

$$
\begin{aligned}
\dot{\xi} &= v_\xi \cos\psi \\
\dot{\eta} &= v_\xi \sin\psi \\
\dot{\psi} &= \frac{v_\xi}{w_b}\tan(\delta_f)
\end{aligned}
\tag{9}
$$

**Table 1**
NMPC design parameters.

| Parameter | Value |
|---|---|
| $T_s$ | 0.1 s |
| $T_p$ | 15 s |
| $Q$ | diag(0.25, 0.25, 0.5) |
| $R$ | diag(0.5, 0.5) |
| $P$ | diag(2, 10, 20) |
| Upper bounds | $[2\,\text{m/s}, \pi/4, 2\,\text{m/s}, \pi/4]$ |
| Lower bounds | $[-2\,\text{m/s}, -\pi/4, -2\,\text{m/s}, -\pi/4]$ |

where $\xi$ and $\eta$ denote the position of the vehicle, $\psi$ its yaw angle, and the parameter $w_b = 2.8$ m represents the wheelbase of the vehicle. The longitudinal speed $v_\xi$ and the steering angle $\delta_f$ are the control variables. The output of the system is $(\xi, \eta, \psi)$. Concerning the state constraint, safety ellipses were designed around the parking vehicles in order to avoid possible collisions.

Below are reported the steps, described in Section 5, for this case study.

#### 6.1.1. Data collection

A starting set of initial state conditions $x_{0p}$, $p = 1, \ldots, 1000$ of the vehicle was obtained through the Latin Hypercube Sampling (LHS) technique (see McKay et al. [26]). Starting from these initial conditions, a Monte Carlo (MC) campaign was carried out using the NMPC algorithm (4) without domain reduction (the design parameters are listed in Table 1). The optimization problem was solved using the Matlab function `fmincon` with the Sequential Quadratic Programming (SQP) algorithm. In the following, this algorithm without domain reduction will be called "Standard NMPC". Note that the NMPC command is parametrized considering two nodes, i.e., $n_s = 2$. This means that there are a total of 4 commands: 2 for the speed $v_\xi$ and 2 for the steering angle $\delta_f$. At the end of this campaign, a dataset of about $M = 3e5$ samples $(\tilde{w}_l, \tilde{u}_l)$ was obtained.

#### 6.1.2. Clustering

A clustering procedure was carried out to reduce the size of the dataset generated in the previous step. In particular, the K-medoids clustering method with the CLARA algorithm was used. After several trials, a reduced set of $1e4$ data was found as an "optimal" compromise between quantity of data (and then memory occupation) and exploration of the control law domain. As mentioned in Section 5.1, the benefit of using $K$-medoids is that the center point of each cluster, i.e., the medoid, is an actual element of the dataset. This allows to always associate a sample of the regressor with the corresponding optimal command.

#### 6.1.3. Set Membership approximation

After the clustering process, the dataset was reduced from $3e5$ to $1e4$ samples $\{\tilde{w}_{ml}, \tilde{u}_{ml}\}_{l=1}^{1e4}$. On the basis of them, the approximated control law $\phi^c$, and the corresponding bounds $\overline{\phi}$ and $\underline{\phi}$ were computed by means of the SM approach shown in Section 4. Figure 2 shows the approximation and the relative bounds of the velocity control command. As it can be seen, the bounds were reduced by about 10 times with respect to the original ones.

#### 6.1.4. Comparison between SM-NMPC and standard NMPC

Once the approximate Set Membership model was created, it is 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 initial state conditions of the ego vehicle, from those considered previously, were taken into account. Then, a MC campaign of 100 simulations was carried out.

**Table 2**
Comparison between standard NMPC and SM-NMPC for a MC campaign of 100 simulations.

| | Standard NMPC | | SM-NMPC | |
|---|---|---|---|---|
| | Mean value | Maximum value | Mean value | Maximum value |
| Eval. Cost. Funct. | 94.4 | 106.1 | 5.46 | 12.16 |
| Comp. Time [s] | 0.0313 | 0.0365 | 0.0057 | 0.0059 |
| Pos. T.E. [m] | 0.1237 | 0.7 | 0.1106 | 0.1388 |
| Orient. T.E. [rad] | 0.0241 | 0.29 | 0.0213 | 0.029 |

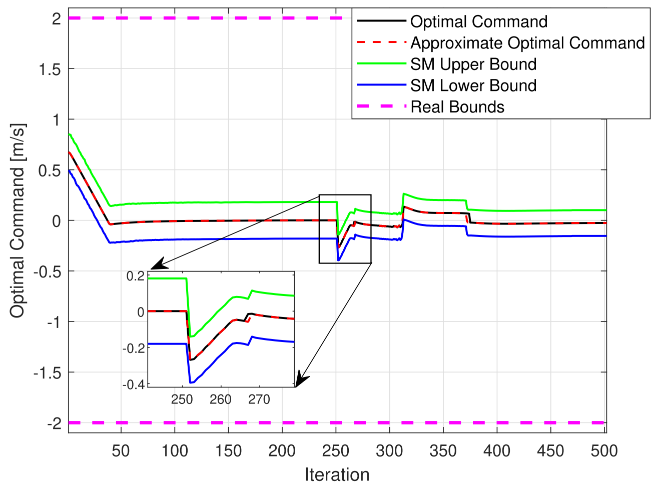


**Fig. 2.** Set Membership approximation of $v_\xi$.

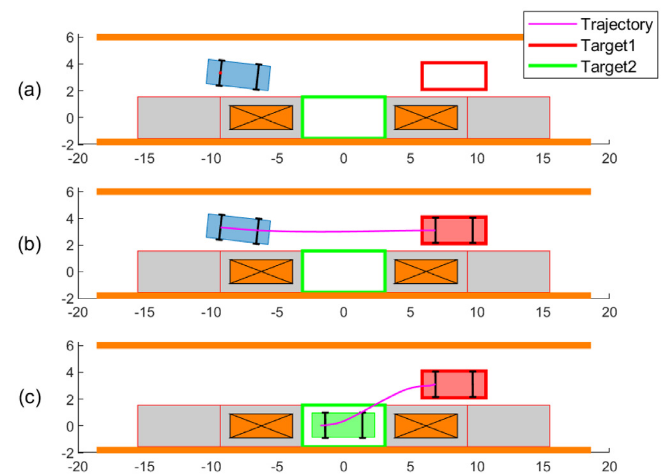


**Fig. 3.** Example of autonomous parallel parking performed by SM-NMPC.

The standard NMPC and the developed SM-NMPC were simulated on a Dell Precision 5820 (Processor: Intel(R) Xeon(R) W-2123 CPU @ 3.60 GHz). The optimization problem was solved using the Matlab function `fmincon` with the Sequential Quadratic Programming (SQP) algorithm. It is important to emphasize that the proposed approach is not limited to a particular optimization algorithm. Instead, it can be employed in combination with any algorithm to enhance its computational efficiency.

The metrics 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. Accuracy in reaching the final target.

In Table 2, the mean and maximum values of the above performance indexes are shown for the two NMPC algorithms. Note that the term Mean Value refers to the average number of evaluated cost functions, computational times, and tracking errors throughout the simulation. For example, mean number of evaluated cost functions ($M_{ecf}$) means:

$$M_{ecf} = \frac{\sum_{k=1}^{n_t} ecf_k}{n_t} \tag{10}$$

where $ecf_k$ is the number of evaluated functions for the $k$th iteration and $n_t$ is the number of iterations required to complete the simulation. The term Maximum Value refers to the highest value during the 100 simulations.

Regarding the number of evaluated cost functions, with the SM-NMPC a reduction of about 17 times, on average, is obtained. Note that, since in this nonlinear optimization problem there are 4 variables to be optimized, at least 5 evaluations of the cost function are required to obtain a numerical estimate of the gradient. Thus, in the case of SM-NMPC, results very close to this minimum number are obtained. For the computational time, the use of SM-NMPC leads to an improvement in the performance of about 6 times, on average. The discrepancy between the enhancement found for the cost functions and this one is due to the fact that the SM-NMPC algorithm requires, before the optimization, the evaluation of the SM approximated control law. This operation, not present in the standard NMPC, implies additional computation time. Note that the code for accomplishing this operation is at a preliminary stage and further improvements are expected. The same considerations also apply to the maximum values of both metrics. With regard to the tracking error, since the values of the position $(\xi, \eta)$ are generally larger than those of the orientation $\psi$, it is split into: Position Tracking Error (Pos. T.E.) and Orientation Tracking Error (Orient. T.E.). For the computation of the Position Tracking Error, the Euclidean distance between Target 2 in Fig. 3 and the final pose of the vehicle is considered. Regarding the Mean Value, the obtained results are quite similar. Instead for the Maximum Value, there is a considerable difference. Indeed, in the case of standard NMPC, the very high value reported in Table 2 is due to the fact that out of 100 simulations the parking maneuver fails four times. The SM-NMPC, instead, always succeeds in completing it. Thus, besides being more efficient from a computational point of view, this approach is also more robust. An example of a complete maneuver performed by the SM-NMPC is shown in Fig. 3.

## 6.2. Lane keeping

As road profile, the sinusoidal signal defined in Section 2 was used. For the plant, the same model of the previous example was considered. Regarding the one for the NMPC prediction, the kinematic model is no longer used. Indeed, it becomes inadapted when the velocity is high and then the vehicle is brought to its limit of adherence and tires start to lose grip on the road (this is referred as drifting). For this reason, 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 cap-

**Table 3**
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 | $[-3\,\mathrm{m/s^2}, -\pi/4, -3\,\mathrm{m/s^2}, -\pi/4]$ |

tures 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:

$$\begin{aligned}
\dot{\xi} &= v_\xi \cos\psi - v_\eta \sin\psi \\
\dot{\eta} &= v_\xi \sin\psi + v_\eta \cos\psi \\
\dot{\psi} &= \omega \\
\dot{v}_\xi &= v_\eta \omega + a_\xi \\
\dot{v}_\eta &= -v_\xi \omega + \frac{2}{m}\left(F_{\eta f} + F_{\eta r}\right) \\
\dot{\omega} &= \frac{2}{I_z}\left(l_f F_{\eta f} - l_r F_{\eta r}\right)
\end{aligned} \tag{11}$$

where $\xi$ and $\eta$ denote the position of the vehicle, $\psi$ the yaw angle, $\omega$ the yaw rate, $v_\xi, v_\eta$ the longitudinal and lateral speeds, $m = 1575\,\mathrm{kg}$ and $I_z = 4000\,\mathrm{kg\,m^2}$ the mass and yaw polar inertia, and $l_f = 1.2\,\mathrm{m}$ and $l_r = 1.6\,\mathrm{m}$ the distance CoG - front/rear wheel center. $F_{\eta f}$ and $F_{\eta r}$ are the lateral forces between the wheels and the vehicle:

$$F_{\eta f} = -c_f \beta_f, \qquad F_{\eta r} = -c_r \beta_r \tag{12}$$

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

$$\beta_f = \mathrm{atan}\left(\frac{v_\eta + l_f\omega}{v_\xi}\right) - \delta_f, \;\; \beta_r = \mathrm{atan}\left(\frac{v_\eta - l_r\omega}{v_\xi}\right). \tag{13}$$

The longitudinal acceleration $a_\xi$ and the steering angle $\delta_f$ are the control variables. The output of the system is $(\xi, \eta)$.

The same steps were performed as in the previous example. Note that for the data collection, 1000 different sinusoidal references were considered, with $5 < A_s\,[\mathrm{m}] < 10$ and $0.01 < \omega_s\,[\mathrm{rad/m}] < 0.04$. For the generation of the values in these intervals, the LHS technique was used. Then, a MC campaign was carried out using the standard NMPC (Table 3 lists the design parameters). Even in this case, the command $u$ was parametrized considering two nodes, and so there are a total of 4 commands. At the end of the MC campaign, a dataset of about $M = 5e5$ samples was obtained. Afterward, the clustering was performed using $K$-medoids, with $K = 2e4$. Then the approximating control law $\phi^c$, and the corresponding bounds $\overline{\phi}$ and $\underline{\phi}$ were computed by means of Set Membership approach. Figure 4 shows the approximation and the relative bounds of one of the steering angle command. Once the SM model was created, it is used in combination with

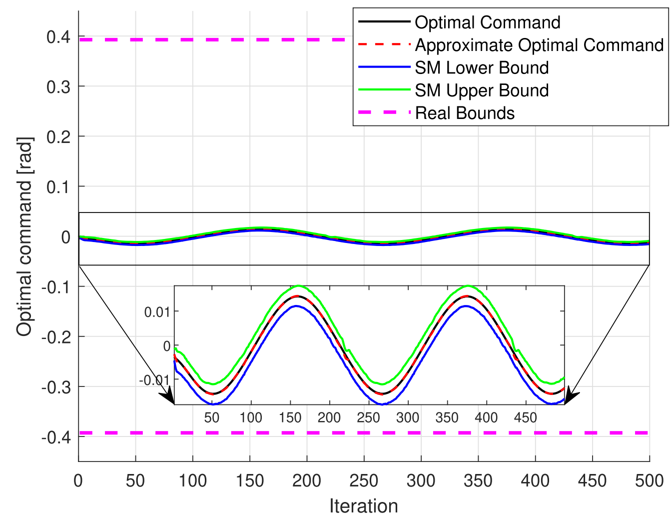


**Fig. 4.** Set Membership approximation of $\delta_f$.

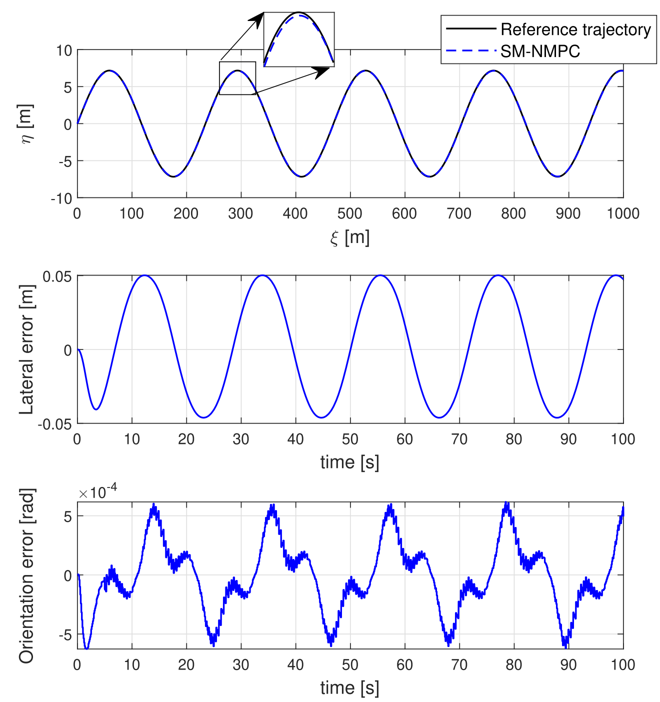


**Fig. 5.** Example of SM-NMPC lane keeping with lateral and orientation errors.

the NMPC. In order to test the SM-NMPC and compare it with the standard NMPC implementation, different values of $A_s$ and $\omega_s$, from those considered previously, were considered. Then, a MC campaign of 100 simulations was carried out on the Dell Precision 5820. Table 4 reports the results considering the same metrics as the previous example. Even in this case, the SM-NMPC outperforms

**Table 4**
Comparison between standard NMPC and SM-NMPC.

| | Standard NMPC | | SM-NMPC | |
|---|---|---|---|---|
| | Mean value | Maximum value | Mean value | Maximum value |
| Eval. Cost Funct. | 107.36 | 109.35 | 7 | 7.74 |
| Comp. Time [s] | 0.041 | 0.043 | 0.0088 | 0.0089 |
| RMS Lat. E. [m] | 0.0254 | 0.0785 | 0.0208 | 0.0454 |
| RMS Orient. E. [rad] | 0.00049 | 0.0011 | 0.00048 | 0.0007 |

the standard implementation. Note that for the accuracy, the Root-Mean-Square (RMS) error is used for evaluating the Lateral Error (Lat. E.) and the Orientation Error (Orient. E.). An example of a sinusoidal road profile and the corresponding trajectory, lateral and orientation errors obtained with the SM-NMPC is shown in Fig. 5.

## 7. Conclusions

The last decades have seen increasingly rapid progress in driverless vehicle technology. In this context, the paper proposes a "fast" data-aided NMPC approach, called Set Membership based Nonlinear Model Predictive Control (SM-NMPC), aimed at trajectory planning and control for autonomous vehicles. In particular, a Set Membership approximation method is applied to derive from data tight bounds on the optimal NMPC control law. This results in a significant reduction of the computation time, thus enabling the real-time NMPC implementation even in systems with high sampling rate. Realistic autonomous vehicle scenarios, concerned with parallel parking and lane keeping, are taken into account for testing in simulation the developed SM-NMPC approach. The obtained performances are compared with a standard NMPC implementation, demonstrating the effectiveness of the proposed method.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


[1] T. Albin, D. Ritter, D. Abel, N. Liberda, R. Quirynen, M. Diehl, Nonlinear MPC for a two-stage turbocharged gasoline engine airpath, in: 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 849–856.
[2] F. Allgöwer, R. Findeisen, Z.K. Nagy, Nonlinear model predictive control: from theory to application, J. Chin. Inst. Chem. Eng. 35 (2004) 299–315.
[3] N. Amer, H. Zamzuri, K. Hudha, Z. Kadir, Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges, J. Intell. Robot. Syst. 86 (2017) 1–30.
[4] H.G. Bock, M. Diehl, E. Kostina, J.P. Schlöder, Constrained optimal feedback control of systems governed by large differential algebraic equations, in: Real-Time PDE-Constrained Optimization, SIAM, 2007, pp. 3–24.
[5] H.G. Bock, K.J. Plitt, A multiple shooting algorithm for direct solution of optimal control problems, IFAC Proc. Vol. 17 (1984) 1603–1608.
[6] M. Boggio, C. Novara, M. Taragna, Nonlinear model predictive control: an optimal search domain reduction, IFAC-PapersOnLine (2023). 22nd IFAC World Congress
[7] M. Campbell, M. Egerstedt, J.P. How, R.M. Murray, Autonomous driving in urban environments: approaches, lessons and challenges, Philos. Trans. R. Soc. A 368 (1928) (2010) 4649–4672.
[8] M. Canale, M. Milanese, C. Novara, Semi-active suspension control using "fast" model-predictive techniques, IEEE Trans. Control Syst. Technol. 14 (6) (2006) 1034–1046.
[9] T.J. Crayton, B.M. Meier, Autonomous vehicles: developing a public health research agenda to frame the future of transportation policy, J. Transp. Health 6 (2017) 245–252.
[10] M. Diehl, H.G. Bock, H. Diedam, P.-B. Wieber, Fast direct multiple shooting algorithms for optimal robot control, in: Fast Motions in Biomechanics and Robotics, Heidelberg, Germany, 2005, pp. 65–93.
[11] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, F. Allgöwer, Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations, J. Process Control 12 (4) (2002) 577–585.
[12] Y. Ding, W. Zhuang, L. Wang, J. Liu, L. Guvenc, Z. Li, Safe and optimal lane-change path planning for automated driving, Proc. Inst. Mech. Eng., Part D 235 (4) (2021) 1070–1083.
[13] D.J. Fagnant, K. Kockelman, Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations, Transp. Res. Part A 77 (2015) 167–181.
[14] G. Franzè, M. Mattei, L. Ollio, V. Scordamaglia, A robust constrained model predictive control scheme for norm-bounded uncertain systems with partial state measurements, Int. J. Robust Nonlinear Control 29 (17) (2019) 6105–6125.
[15] J.V. Frasch, A. Gray, M. Zanon, H.J. Ferreau, S. Sager, F. Borrelli, M. Diehl, An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles, in: 2013 European Control Conference (ECC), 2013, pp. 4136–4141.
[16] N. Guo, B. Lenzo, X. Zhang, Y. Zou, R. Zhai, T. Zhang, A real-time nonlinear model predictive controller for yaw motion optimization of distributed drive electric vehicles, IEEE Trans. Veh. Technol. 69 (5) (2020) 4935–4946.
[17] G. Hongbo, X. Guotao, Z. Xinyu, C. Bo, Autonomous parking control for intelligent vehicles based on a novel algorithm, J. China Univ. Posts Telecommun. 24 (4) (2017) 51–56.
[18] M.F. Hsieh, U. Ozguner, A parking algorithm for an autonomous vehicle, in: 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, 2008, pp. 1155–1160, doi:10.1109/IVS.2008.4621317.
[19] J.-H. Jhang, F.-L. Lian, An autonomous parking system of optimally integrating bidirectional rapidly-exploring random trees and parking-oriented model predictive control, IEEE Access 8 (2020) 163502–163523.
[20] L. Kaufman, P.J. Rousseeuw, Clustering large data sets, in: E.S. Gelsema, L.N. Kanal (Eds.), Pattern Recognition in Practice, Elsevier, 1986, pp. 425–437.
[21] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 2009.
[22] K.D. Kusano, H.C. Gabler, Comprehensive target populations for current active safety systems using national crash databases, Traffic Injury Prev. 15 (7) (2014) 753–761.
[23] W.C. Li, L.T. Biegler, Multistep, newton-type control strategies for constrained, nonlinear processes, Chem. Eng. Res. Des. 67 (1989) 562–577.
[24] MATLAB, Vehicle body 3DOF, 2018, https://it.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html.
[25] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.
[26] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (2) (1979) 239–245.
[27] M. Milanese, C. Novara, Set membership identification of nonlinear systems, Automatica 40 (6) (2004) 957–975.
[28] T. Ohtsuka, A continuation/GMRES method for fast computation of nonlinear receding horizon control, Automatica 40 (4) (2004) 563–574.
[29] R. Okuda, Y. Kajiwara, K. Terashima, A survey of technical trend of ADAS and autonomous driving, in: Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test, Hsinchu, Taiwan, 2014, pp. 1–4.
[30] T. Parisini, R. Zoppoli, A receding-horizon regulator for nonlinear systems and a neural approximation, Automatica 31 (10) (1995) 1443–1451.
[31] S.J. Qin, T.A. Badgwell, An overview of nonlinear model predictive control applications, in: F. Allgöwer, A. Zheng (Eds.), Nonlinear Model Predictive Control, Birkhäuser Basel, Basel, 2000, pp. 369–392.
[32] J. Richalet, A. Rault, J.L. Testud, J. Papon, Model predictive heuristic control: applications to industrial processes, Automatica 14 (5) (1978) 413–428.
[33] S. Sternlund, J. Strandroth, M. Rizzi, A. Lie, C. Tingvall, The effectiveness of lane departure warning systems-a reduction in real-world passenger car injury crashes, Traffic Injury Prev. 18 (2) (2017) 225–229.
[34] E. Yurtsever, J. Lambert, A. Carballo, K. Takeda, A survey of autonomous driving: common practices and emerging technologies, IEEE Access 8 (2020) 58443–58469.
[35] V.M. Zavala, L.T. Biegler, The advanced-step NMPC controller: optimality, stability and robustness, Automatica 45 (1) (2009) 86–93.