Received: 9 October 2023 | Revised: 1 March 2024 | Accepted: 28 April 2024
DOI: 10.1002/rnc.7411

RESEARCH ARTICLE

WILEY

# Learning Lyapunov terminal costs from data for complexity reduction in nonlinear model predictive control

**Shokhjakhon Abdufattokhov** | **Mario Zanon** | **Alberto Bemporad**

IMT School for Advanced Studies in Lucca, Lucca, Italy

**Correspondence**
Mario Zanon, IMT School for Advanced Studies in Lucca, Piazza S.Francesco, 19, 55100, Lucca, Italy.
Email: mario.zanon@imtlucca.it

**Funding information**
Italian Ministry of University and Research, Grant/Award Number: 2017J89ARP

**Abstract**
A classic way to design a nonlinear model predictive control (NMPC) scheme with guaranteed stability is to incorporate a terminal cost and a terminal constraint into the problem formulation. While a long prediction horizon is often desirable to obtain a large domain of attraction and good closed-loop performance, the related computational burden can hinder its real-time deployment. In this article, we propose an NMPC scheme with prediction horizon $N = 1$ and no terminal constraint to drastically decrease the numerical complexity without significantly impacting closed-loop stability and performance. This is attained by constructing a suitable terminal cost from data that estimates the cost-to-go of a given NMPC scheme with long prediction horizon. We demonstrate the advantages of the proposed control scheme in two benchmark control problems.

**KEYWORDS**
constrained systems, data-driven control, neural networks, nonlinear model predictive control

## 1 | INTRODUCTION

Nonlinear model predictive control (NMPC) is an optimization-based control method that has gained increasing popularity thanks to its ability to achieve good closed-loop performance while handling nonlinear system dynamics and enforcing hard constraints on the system variables .[1] Stability is typically guaranteed in NMPC by introducing suitable terminal cost and constraints in the problem formulation .[2–4] In order to obtain a large domain of attraction and good closed-loop performance one is often required to choose a sufficiently long prediction horizon, which can result in a significant computational complexity.

Preserving these desirable properties while reducing the computational burden has motivated several research papers in recent years. Some approaches aim at finding approximate explicit NMPC laws through function regression [5–8] and fuzzy clustering [9] methods, although preserving feasibility with respect to system constraints can be challenging .[10] When implicit NMPC is the preferred choice, the computational burden of online optimization can be decreased by shortening the number of degrees of freedom using move blocking [11–13]; by parameterizing the decision variables with basis functions [14,15]; by decomposing the control input space to construct low-dimensional input subspaces [16,17]; or by developing approximate but efficient optimization algorithms .[18,19] Regardless of the used implicit NMPC approach, shortening the prediction horizon leads to lighter computational requirements. However, the terminal cost and constraints may negatively impact closed-loop performance and introduce conservativeness by restricting the domain of attraction. The authors of Reference 20 presented an approach to remove the terminal constraint while ensuring its satisfaction automatically and preserving asymptotic closed-loop stability in a characterized domain of attraction. However, achieving a good trade-off between closed-loop performance and computational complexity requires properly tuning both the prediction horizon and the terminal cost, which needs to be a Lyapunov function in the (implicit) terminal region. Unfortunately, designing a non-conservative Lyapunov function and the corresponding terminal region is, in general, difficult .[21] One possibility to tackle this difficulty is to resort to data-driven approaches, such as in, for example, References 22 and 23. Despite obtaining some form of stability and recursive feasibility guarantees, attaining good closed-loop performance with a short horizon remains an open issue.

This article proposes an implicit NMPC formulation with a prediction horizon of length one, which approximates a given long-horizon one. Based on a set of samples collected from optimal solutions of the latter, we obtain the horizon-one NMPC reformulation by learning a parametric terminal cost that satisfies Lyapunov conditions on the data samples and is a good approximation of the cost-to-go of the original problem. The terminal-cost model is parameterized to be quadratic with respect to the terminal state, although generically nonlinear with respect to the initial state of the prediction, reference signals, and possibly other exogenous signals. To ensure that the approximate cost-to-go is positive definite, we parameterize it as the product of a lower triangular matrix and its transpose plus a small regularization, where each entry of the triangular matrix is the output of a feedforward neural network.

We combine the approach of Reference 20 with our recent contribution ,[24] where the terminal cost for an MPC scheme with prediction horizon one is also learned from data. An approach similar to Reference 24 was recently proposed in Reference 25, where a different parameterization of the terminal cost is used.

We call our design approach *learned Lyapunov terminal cost NMPC* (LLTC-NMPC). Besides being attractive from a computational point of view, as opposed to approximate explicit NMPC methods ,[26,27] our approach has the advantage of keeping the constraints on the system variables at the first step in the optimization problem. Contrary to the approximate implicit MPC approaches in References 24 and 25, we do enforce stability conditions on the data samples. We perform numerical simulations on two benchmark control problems, showing that the proposed control scheme drastically reduces online computational burden with respect to the original NMPC formulation while maintaining good closed-loop performance.

The remainder of this article is organized as follows. After providing preliminary results and formulations leading to an NMPC optimization problem with a single prediction horizon in Section 2, we detail how to learn a quadratic Lyapunov approximation of the cost-to-go function and discuss the proposed LLTC-NMPC formulation in Section 3. The effectiveness of the proposed approach is demonstrated in simulations in Section 4. Finally, concluding remarks are drawn in Section 5.

## 2 | PRELIMINARIES AND PROBLEM FORMULATION

We consider the following discrete-time nonlinear system

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, and $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$ denote the state and input at a time instant $t$, respectively. The states and the commanded inputs are constrained in the respective compact sets $\mathcal{X}$ and $\mathcal{U}$.

### 2.1 | Classical NMPC

We introduce a classical NMPC problem $\mathbb{P}_N(p_t, \mathcal{X}_\mathrm{T}(p_t))$ formulated as

$$\begin{aligned} \mathcal{J}^N(p_t) = \min_{U_t, X_t} \ & \sum_{k=0}^{N-1} \ell(x_{k|t}, u_{k|t}, p_t) + F(x_{N|t}, p_t) \\ \text{s.t.} \ & x_{0|t} = \mathcal{M}_x p_t \\ & x_{k+1|t} = f(x_{k|t}, u_{k|t}) && k \in \mathbb{I}_0^{N-1} \\ & u_{k|t} \in \mathcal{U} && k \in \mathbb{I}_0^{N-1} \\ & x_{k|t} \in \mathcal{X} && k \in \mathbb{I}_1^{N-1} \\ & x_{N|t} \in \mathcal{X}_\mathrm{T}(p_t), \end{aligned} \tag{2}$$

where $p_t = [x_t' \ x_r' \ u_r']' \in \mathbb{R}^{n_p}$ is a vector of parameters belonging to a bounded set $\mathcal{P}$ that consists of the current state $x_t \in \mathcal{X}$ and reference signals $x_r \in \mathcal{X}_r \subseteq \mathcal{X}$ and $u_r \in \mathcal{U}_r \subseteq \mathcal{U}$ that are kept constant over the prediction horizon $N$, $\mathcal{M}_x = [I \ 0 \ 0]$, $U_t = (u_{0|t}, \ldots, u_{N-1|t})$ is the sequence of manipulated variables $u_{k|t} \in \mathbb{R}^{n_u}$, $X_t = (x_{0|t}, \ldots, x_{N|t})$ is the sequence of predicted states $x_{k|t} \in \mathbb{R}^{n_x}$ evaluated from the initial state $x_{0|t}$, and $\mathbb{I}_a^b$ is the set of all integers in the interval $[a, b]$.

In what follows, $\mathcal{F}_N(p_t)$ defines the set of initial states for which Problem $\mathbb{P}_N(p_t, \mathcal{X}_T(p_t))$ is feasible. Let $U_t^\star = (u_{0|t}^\star, \ldots, u_{N-1|t}^\star)$ denote the optimal control input and $X_t^\star = (x_{0|t}^\star, \ldots, x_{N|t}^\star)$ the corresponding state trajectory vectors obtained from (2). The NMPC law is defined by applying the first optimal control action $u_t = u_{0|t}^\star$ to the controlled system (1).

The stage cost function $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \in \mathbb{R}_{\geq 0}$ is defined as

$$\ell(x_{k|t}, u_{k|t}, p_t) = \|x_{k|t} - x_r\|_{Q_x}^2 + \|u_{k|t} - u_r\|_{Q_u}^2, \tag{3}$$

where $\|z\|_Q^2 := z'Qz$, $Q_x$ and $Q_u$ are positive definite matrices, $\mathcal{X}_T(p) \subseteq \mathcal{X}$ is a closed terminal constraint set containing $x_r$ and $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \in \mathbb{R}_{\geq 0}$ is a quadratic terminal cost

$$F(x_{N|t}, p_t) = \|x_{N|t} - x_r\|_P^2 \tag{4}$$

with a positive definite matrix $P$. Note that, while the terminal cost could take a generic form, selecting it as quadratic eases the enforcement of the necessary conditions for stability that will be discussed next.

We assume that the given NMPC formulation (2) is a setpoint tracking problem which satisfies the following standard conditions on smoothness, recursive feasibility, stage cost, and stability:

**Assumption 1.** Function $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is continuously differentiable over $\mathcal{X} \times \mathcal{U}$, and $f(x_r, u_r) = x_r$ for all $x_r \in \mathcal{X}_r$ and $u_r \in \mathcal{U}_r$.

**Assumption 2.** For each $p_t \in \mathcal{P}$ there exists optimal sequences $U_t^\star$ and $X_t^\star$ from Problem $\mathbb{P}_N(p_t, \mathcal{X}_T(p_t))$.

**Assumption 3.** For all $p_t \in \mathcal{P}$, there exists a $\mathcal{K}_\infty$-function (i.e., continuous, strictly increasing, unbounded, choice and zero at the origin) $\mu_1(\cdot)$ such that

$$\mu_1(\|x - x_r\|) \leq \ell(x, u, p_t) \quad \forall x \in \mathcal{F}_N(p_t), \ \forall u \in \mathcal{U}. \tag{5}$$

**Assumption 4.** Let $F : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$, $l_T : \mathcal{P} \to \mathbb{R}_{>0}$ and

$$\mathcal{X}_T(p_t) = \{x \in \mathcal{X} : F(x, p_t) \leq l_T(p_t)\}$$

be such that $F$ is a Lyapunov function associated to a local stabilizing controller $u_T(x, p_t) : \mathcal{X}_T(p_t) \times \mathcal{P} \to \mathcal{U}$, that is, $\forall x \in \mathcal{X}_T(p_t)$ and $f(x, u_T(x, p_t)) \in \mathcal{X}_T(p_t)$ it holds that

$$\mu_2(\|x - x_r\|) \leq F(x, p_t) \leq \mu_3(\|x - x_r\|), \tag{6a}$$

$$F(f(x, u_T(x, p_t)), p_t) - F(x, p_t) + \ell(x, u_T(x, p_t), p_t) \leq 0, \tag{6b}$$

where $\mu_2$ and $\mu_3$ are $\mathcal{K}_\infty$ functions.

The terminal cost and the terminal constraint defined in (6) guarantee closed-loop stability.

*Remark* 1. Note that in the functions $\ell(x, u, p_t)$ and $F(x, p_t)$, and other functions resulting from them, the variable $x$ is a $k$-step ahead predicted state at a time $t$ that needs to be optimized over the prediction horizon $N$, while the parameter vector $p_t$ is fixed with the current state $x_t$ and reference signals $x_r$ and $u_r$.

Under Assumptions 1 to 3, the authors of Reference 28 proposed an approach to construct the terminal set and the terminal cost using a locally stabilizing linear controller for discrete-time systems. Moreover, if Assumption 4 also holds, the optimal cost $\mathcal{J}^N$ in (2) is a Lyapunov function and the NMPC feedback law asymptotically stabilizes the system for all initial states in $\mathcal{F}_N(p_t)$. However, enforcing the terminal state to lie inside the terminal region $\mathcal{X}_T(p_t)$ may yield a small region of attraction. On the other hand, removing the terminal constraint might jeopardize the feasibility and stability guarantees of the problem $\mathbb{P}_N(p_t, \mathcal{X}_\mathrm{T}(p_t))$.

### 2.2 | Full-length NMPC without terminal constraint

We first start considering the full-length NMPC problem with horizon $N \geq 1$ that we will approximate in Section 2.3. The authors of Reference 20 proposed an approach ensuring implicit satisfaction of the terminal constraint of the problem $\mathbb{P}_N(\mathcal{X})$, where, with a slight abuse of notation, the dependence on $p_t$ has been dropped as their problem formulation is parameter-independent. In the following, we extend their stability results to cover the case of a parameter-dependent formulation, that is, an MPC problem $\mathbb{P}_N(p_t, \mathcal{X})$ defined as follows:

$$
\begin{aligned}
\mathcal{J}^N_{\mathrm{CDA}}(p_t) = \min_{U_t, X_t} \quad & \sum_{k=0}^{N-1} \ell(x_{k|t}, u_{k|t}, p_t) + F(x_{N|t}, p_t) \\
\text{s.t.} \quad & x_{0|t} = \mathcal{M}_x p_t \\
& x_{k+1|t} = f(x_{k|t}, u_{k|t}) && k \in \mathbb{I}_0^{N-1} \\
& u_{k|t} \in \mathcal{U} && k \in \mathbb{I}_0^{N-1} \\
& x_{k|t} \in \mathcal{X} && k \in \mathbb{I}_1^{N}.
\end{aligned}
\tag{7}
$$

We will next characterize a positively invariant subset $\Omega_N(p_t)$ of the region of attraction, for which we can prove asymptotic stability. Therefore, we will denote problem $\mathbb{P}_N(p_t, \mathcal{X})$ as *NMPC with Characterized Domain of Attraction* (CDA-NMPC). In order to prove that $\Omega_N$ is a positively invariant set in which asymptotic stability can be proven, we first introduce two useful intermediate results.

**Lemma 1** (Extension of Reference 29 (Section 3.4)). *Given optimization problem $\mathbb{P}_N(p_t, \mathcal{X})$, suppose that Assumptions 1–3 hold and $F(x_{0|t}, p_t)$, $\mathcal{X}_\mathrm{T}(p_t)$ satisfy Assumption 4 for all $x_{0|t} \in \mathcal{X}_\mathrm{T}(p_t)$ and for all $p_t \in \mathcal{P}$. Then $\mathcal{J}^N_{\mathrm{CDA}}(p_t) \leq F(x_{0|t}, p_t)$ holds.*

*Proof.* Let $U_t^\star$ be the optimal control input vector and $X_t^\star$ be the corresponding optimal predicted trajectories of the problem $\mathbb{P}_N(p_t, \mathcal{X})$, then Bellman's principle of optimality implies the following

$$
\begin{aligned}
\mathcal{J}^N_{\mathrm{CDA}}(p_t) &= \sum_{k=0}^{N-1} \ell(x^\star_{k|t}, u^\star_{k|t}, p_t) + F(x^\star_{N|t}, p_t) \\
&\leq \sum_{k=0}^{N-2} \ell(x_{k|t}, u_\mathrm{T}(x_{k|t}, p_t), p_t) + \ell(x_{N-1|t}, u_\mathrm{T}(x_{N-1|t}, p_t), p_t) + F(x_{N|t}, p_t) \\
&\overset{(6b)}{\leq} \sum_{k=0}^{N-2} \ell(x_{k|t}, u_\mathrm{T}(x_{k|t}, p_t), p_t) + F(x_{N-1|t}, p_t) \\
&= \sum_{k=0}^{N-3} \ell(x_{k|t}, u_\mathrm{T}(x_{k|t}, p_t), p_t) + \ell(x_{N-2|t}, u_\mathrm{T}(x_{N-2|t}, p_t), p_t) + F(x_{N-1|t}, p_t) \\
&\overset{(6b)}{\leq} \cdots \overset{(6b)}{\leq} \ell(x_{0|t}, u_\mathrm{T}(x_{0|t}, p_t), p_t) + F(x_{1|t}, p_t) \\
&\overset{(6b)}{\leq} F(x_{0|t}, p_t),
\end{aligned}
$$

where $x_{k|t}$ is the state trajectory obtained when applying the terminal control law $u_\mathrm{T}$ satisfying Assumption 4 to system (1) from the initial state $x_{0|t}$. ■

**Lemma 2** (Extension of Reference 20 (Lemma 1)). *Given optimization problem $\mathbb{P}_N(p_t, \mathcal{X})$, suppose that Assumptions 1–3 hold and $F(x_{0|t}, p_t)$, $\mathcal{X}_\mathrm{T}(p_t)$ satisfy Assumption 4 for all $x_{0|t} \in \mathcal{F}_N(p_t)$ and for all $p_t \in \mathcal{P}$. Let $X_t^\star$ be the optimal predicted sequence of states corresponding to the optimal solution $U_t^\star$ of the problem $\mathbb{P}_N(p_t, \mathcal{X})$. If $x^\star_{N|t} \notin \mathcal{X}_\mathrm{T}(p_t)$, then $x^\star_{k|t} \notin \mathcal{X}_\mathrm{T}(p_t)$ for any $k \in \mathbb{I}_0^{N-1}$.*

*Proof.* Assume $x^\star_{N|t} \notin \mathcal{X}_\text{T}(p_t)$ and there exists a $k \in \mathbb{I}_0^{N-1}$ such that $x^\star_{k|t} \in \mathcal{X}_\text{T}(p_t)$. Consider an independent CDA-NMPC problem $\mathbb{P}_{N-k}(p_{k|t}, \mathcal{X})$ with prediction horizon $N-k$ and initial state $x^\star_{k|t}$, matching the solution of the initial problem $\mathbb{P}_N(p_t, \mathcal{X})$ at time $k|t$, where $p_{k|t}$ is defined as

$$p_{k|t} = [x^{\star\,\prime}_{k|t}\ x_\text{r}'\ u_\text{r}']'. \tag{8}$$

Let $\mathcal{J}^{N-k}_\text{CDA}$ be the optimal cost for the problem $\mathbb{P}_{N-k}(p_{k|t}, \mathcal{X})$, and $\bar{U}^\star_{k|t} = (\bar{u}^\star_{k|t}, \dots, \bar{u}^\star_{N-1|t})$ and $\overline{X}^\star_{k|t} = (\overline{x}^\star_{k|t}, \dots, \overline{x}^\star_{N|t})$ be the respective optimal control input and optimal predicted state vectors. As a consequence of the optimality principle, we have $\bar{u}^\star_{i|t} = u^\star_{i|t}$ and $\overline{x}^\star_{i+1|t} = x^\star_{i+1|t}$ for $i \in \mathbb{I}_k^{N-1}$ which implies

$$\ell(\overline{x}^\star_{i|t}, \bar{u}^\star_{i|t}, p_{k|t}) = \ell(x^\star_{i|t}, u^\star_{i|t}, p_t), \quad i \in \mathbb{I}_k^{N-1}, \tag{9}$$

$$F(\overline{x}^\star_{N|t}, p_{k|t}) = F(x^\star_{N|t}, p_t). \tag{10}$$

Thus, the optimal cost $\mathcal{J}^{N-k}_\text{CDA}$ can be written as

$$\mathcal{J}^{N-k}_\text{CDA}(p_{k|t}) = \sum_{i=k}^{N-1} \ell(x^\star_{i|t}, u^\star_{i|t}, p_t) + F(x^\star_{N|t}, p_t).$$

By applying Lemma 1 to the problem $\mathbb{P}_{N-k}(p_{k|t}, \mathcal{X})$ and combining the result with the equation above, we obtain the following inequality

$$F(x^\star_{k|t}, p_t) \geq \mathcal{J}^{N-k}_\text{CDA}(p_{k|t}) \geq F(x^\star_{N|t}, p_t) > l_\text{T}(p_t),$$

where the last inequality follows from the assumption $x^\star_{N|t} \notin \mathcal{X}_\text{T}(p_t)$. However, this implies $x^\star_{k|t} \notin \mathcal{X}_\text{T}(p_t)$, which contradicts the initial assumption for $x^\star_{k|t} \in \mathcal{X}_\text{T}(p_t)$. ■

**Theorem 1.** *Suppose Assumptions 1 and 3 hold and let the terminal cost $F(x^\star_{N|t}, p_t)$ together with the terminal set $\mathcal{X}_\text{T}(p_t)$ and $l_\text{T}(p_t)$ satisfy Assumption 4. Then the* CDA-NMPC *controller associated with problem $\mathbb{P}_N(p_t, \mathcal{X})$ stabilizes system (1) asymptotically for each initial state $x_{0|t} \in \Omega_N(p_t) \subseteq \mathcal{F}_N(p_t)$, with:*

$$\Omega_N(p_t) := \{x_{0|t} \in \mathcal{X} : \mathcal{J}^N_\text{CDA}(p_t) \leq \ell(x_{0|t}, u^\star_{0|t}, p_t) + C_N(p_t)\}, \tag{11}$$

*where*

$$C_N(p_t) := (N-1)d(p_t) + l_\text{T}(p_t) \tag{12}$$

*and*

$$\begin{aligned} d(p_t) := \inf_{x,u}\ & \ell(x, u, p_t) \\ \text{s.t.}\ & x \in \mathcal{X} \setminus \mathcal{X}_\text{T}(p_t) \\ & u \in \mathcal{U}. \end{aligned} \tag{13}$$

*Remark* 2. Note that $d(p_t) > 0$ exists for all $p_t \in \mathcal{P}$ since $\ell(x, u, p_t)$ is positive definite in $x$ and $u$ as per Assumption 3, $\mathcal{X}$ and $\mathcal{U}$ are compact sets, and the target state is in the interior of the terminal constraint set $\mathcal{X}_\text{T}(p_t)$. Note further that $\Omega_N(p_t)$ does not depend on $x_t$, but only on $x_\text{r}$, $u_\text{r}$, and we write it as a function of $p_t$ for simplicity.

*Proof of Theorem* 1. We first prove that the optimal solution of problem $\mathbb{P}_N(p_t, \mathcal{X})$ satisfies the terminal constraint of problem $x^*_{N|t} \in \mathcal{X}_\text{T}(p_t)$. To this end, assume by contradiction that $x^\star_{N|t} \notin \mathcal{X}_\text{T}(p_t)$. Lemma 2 proves that $x^\star_{k|t} \notin \mathcal{X}_\text{T}(p_t),\ \forall\ k < N$. This implies that $\ell(x^\star_{k|t}, u^\star_{k|t}, p_t) > d(p_t)$ and $F(x^\star_{N|t}, p_t) > l_\text{T}(p_t)$, and thus

$$\begin{aligned}\mathcal{J}_{\text{CDA}}^{N}(p_t) &= \ell(x_{0|t}, u_{0|t}^{\star}, p_t) + \sum_{k=1}^{N-1} \ell(x_{k|t}^{\star}, u_{k|t}^{\star}, p_t) + F(x_{N|t}^{\star}, p_t) \\ &> \ell(x_{0|t}, u_{0|t}^{\star}, p_t) + (N-1)d(p_t) + l_{\text{T}}(p_t) \\ &= \ell(x_{0|t}, u_{0|t}^{\star}, p_t) + C_N(p_t),\end{aligned} \tag{14}$$

which contradicts $x_{0|t} \in \Omega_N(p_t)$ as per (11). Therefore, the terminal constraint is satisfied, that is, $x_{N|t}^{\star} \in \mathcal{X}_{\text{T}}(p_t)$.

We prove next that $\Omega_N(p_t)$ is a positively invariant set for the closed-loop system. To this end, let $(\bar{u}_{1|t}^{\star}, \dots, \bar{u}_{N-1|t}^{\star})$ and $(\bar{x}_{2|t}^{\star}, \dots, \bar{x}_{N|t}^{\star})$ be obtained from solving the problem $\mathbb{P}_{N-1}(p_{1|t}, \mathcal{X})$ for the initial condition $x_{0|1} = \mathcal{M}_x p_{1|t}$ constructed according to (8) and let $\mathcal{J}_{\text{CDA}}^{N-1}(p_{1|t})$ be the associated optimal cost. The following inequality can be inferred:

$$\begin{aligned}\mathcal{J}_{\text{CDA}}^{N-1}(p_{1|t}) &\overset{(9)-(10)}{=} \sum_{k=1}^{N-1} \ell(x_{k|t}^{\star}, u_{k|t}^{\star}, p_{1|t}) + F(x_{N|t}^{\star}, p_{1|t}) \\ &\overset{(6b)}{\geq} \sum_{k=1}^{N-1} \ell(x_{k|t}^{\star}, u_{k|t}^{\star}, p_{1|t}) + \ell(x_{N|t}^{\star}, u_{\text{T}}(x_{N|t}^{\star}, p_{1|t}), p_{1|t}) + F(f(x_{N|t}^{\star}, u_{\text{T}}(x_{N|t}^{\star}, p_{1|t}), p_{1|t}) \\ &\geq \sum_{k=1}^{N} \ell(x_{k|t}^{\star}, u_{k|t}^{\star}, p_{1|t}) + F(x_{N+1|t}^{\star}, p_{1|t}) \\ &= \mathcal{J}_{\text{CDA}}^{N}(p_{1|t}),\end{aligned} \tag{15}$$

from which we also get

$$\begin{aligned}\mathcal{J}_{\text{CDA}}^{N}(p_{1|t}) &\overset{(15)}{\leq} \mathcal{J}_{\text{CDA}}^{N-1}(p_{1|t}) \\ &\overset{(9)-(10)}{=} \sum_{k=1}^{N-1} \ell(x_{k|t}^{\star}, u_{k|t}^{\star}, p_{1|t}) + F(x_{N|t}^{\star}, p_{1|t}) \\ &\overset{(14)}{=} \mathcal{J}_{\text{CDA}}^{N}(p_t) - \ell(x_{0|t}, u_{0|t}^{\star}, p_t) \leq C_N(p_t).\end{aligned} \tag{16}$$

This implies $x_{1|t}^{\star} \in \Omega_N(p_t)$.

To prove that the optimal cost $\mathcal{J}_{\text{CDA}}^{N}$ is a Lyapunov function $\forall\, x_{0|t} \in \Omega_N(p_t)$ and $\forall\, p_t \in \mathcal{P}$, we first establish the following bounds. Using Assumption 3, the lower bound is obtained as

$$\mathcal{J}_{\text{CDA}}^{N}(p_t) \geq \ell(x_{0|t}, u_{0|t}^{\star}, p_t) \geq \mu_1(\|x_{0|t} - x_{\text{r}}\|),$$

while from Lemma 1 we have

$$0 \leq \mathcal{J}_{\text{CDA}}^{N}(p_t) \leq F(x_{0|t}, p_t) \leq \mu_3(\|x_{0|t} - x_{\text{r}}\|).$$

$\forall\, x_{0|t} \in \mathcal{X}_{\text{T}}(p_t) \subseteq \Omega_N(p_t)$, with $\mu_3$ from Assumption 4. Finally, in order to prove the decrease condition for $\mathcal{J}_{\text{CDA}}^{N}$, we use the inequality derived in (16) for $\mathcal{J}_{\text{CDA}}^{N}(p_{1|t})$ and $\mathcal{J}_{\text{CDA}}^{N}(p_t)$, and conclude that

$$\begin{aligned}\mathcal{J}_{\text{CDA}}^{N}(p_{1|t}) - \mathcal{J}_{\text{CDA}}^{N}(p_t) &\leq -\ell(x_{0|t}, u_{0|t}^{\star}, p_t) \\ &\leq -\mu_1(\|x_{0|t} - x_{\text{r}}\|).\end{aligned}$$

This proves asymptotic stability of system (1) in closed-loop with the control law yielded by CDA-NMPC for initial states $x_{0|t} \in \Omega_N(p_t)$. ∎

The result of Theorem 1 is important as it proves that the easy-to-characterize set $\Omega_N(p_t)$ is a subset of the region of attraction of the NMPC problem $\mathbb{P}(p_t, \mathcal{X})$. As we will explain later on, this will be useful while learning the terminal cost, see Algorithm 1. We focus next on the data-driven version of CDA-NMPC with low complexity.

### 2.3 | Complexity reduction

NMPC formulations with long prediction horizons may have an excessive computational burden for real-time implementation. To overcome this issue, one option consists in designing a proper cost-to-go function that allows a short prediction horizon $N$ without incurring excessive performance loss. In this article, we will take the extreme case $N = 1$.

In order to define an NMPC formulation with $N = 1$ which matches the long-horizon CDA-NMPC problem $\mathbb{P}_N(p_t, \mathcal{X})$ in (7), we exploit Bellman's principle of optimality to reformulate the NMPC problem as

$$\begin{aligned} \mathcal{J}^N_{\mathrm{CDA}}(p_t) = \min_{u_{0|t}, x_{0|t}, x_{1|t}} \quad & \ell(x_{0|t}, u_{0|t}, p_t) + \mathcal{V}(x_{1|t}, p_t) \\ \text{s.t.} \quad & x_{0|t} = \mathcal{M}_x p_t \\ & x_{1|t} = f(x_{0|t}, u_{0|t}) \\ & u_{0|t} \in \mathcal{U} \\ & x_{1|t} \in \mathcal{X}, \end{aligned} \tag{17}$$

where $\mathcal{V} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}$ is the *cost-to-go* defined as

$$\begin{aligned} \mathcal{V}(x^{\star}_{1|t}, p_t) = \min_{U_{1|t}, X_{1|t}} \quad & \sum_{k=1}^{N-1} \ell(x_{k|t}, u_{k|t}, p_t) + F(x_{N|t}, p_t) & \\ \text{s.t.} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}) & k \in \mathbb{I}_1^{N-1} \\ & u_{k|t} \in \mathcal{U} & k \in \mathbb{I}_1^{N-1} \\ & x_{k|t} \in \mathcal{X} & k \in \mathbb{I}_2^{N} \end{aligned} \tag{18}$$

with $U_{1|t} = (u_{1|t}, \dots, u_{N-1|t})$ and $X_{1|t} = (x_{1|t}, \dots, x_{N|t})$.

The formulation (17) and (18) highlights the well-known fact that an NMPC problem with prediction horizon $N = 1$ can yield the solution of a longer-horizon NMPC, provided that the cost-to-go function is used as terminal cost. Unfortunately, this does not immediately help reduce the complexity of NMPC, due to the possible complexity of $\mathcal{V}$. However, by accepting some possible performance loss, one can replace $\mathcal{V}$ with a different terminal cost $\hat{\mathcal{V}}$ which approximates $\mathcal{V}(x, p_t)$ but is functionally simpler and is a Lyapunov function, so that asymptotic stability is ensured. In the next section, we will discuss how to compute such a terminal cost using a learning-based approach supported by neural networks, similar to References 30–32.

## 3 | NMPC WITH LEARNED LYAPUNOV TERMINAL COST

In this section, we propose an approach to learn a Lyapunov terminal cost function $\hat{\mathcal{V}}^{\mathrm{LLTC}}(x, p_t)$ that approximates the cost-to-go function $\mathcal{V}(x, p_t)$ from data using neural networks. This will allow us to formulate problem $\mathbb{P}_N(p_t, \mathcal{X})$ with prediction horizon $N = 1$, therefore reducing the computational burden with respect to the original, long-horizon problem formulation. Because computing a Lyapunov function is in general very difficult, we resort to a data-driven approach, in which we enforce the Lyapunov conditions only on a finite amount of data points, as we detail next. Clearly, in this case, the stability guarantees only hold on training data. Extending stability guarantees beyond training data would require additional assumptions and possibly constraints imposed during the learning phase, a topic that we leave for future research.

### 3.1 | Learning a Lyapunov cost-to-go function

Asymptotic stability requires that the approximate terminal cost is a Lyapunov function in $\Omega_N(p_t)$, that is, $\hat{\mathcal{V}}^{\mathrm{LLTC}}(x, p_t)$ must be a decreasing, continuous, and positive definite function for all $x \in \Omega_N(p_t)$ and all $p_t \in \mathcal{P}$. While many functional forms are possible, we propose to select

$$\hat{\mathcal{V}}^{\mathrm{LLTC}}(x, p_t) = (x - x_{\mathrm{r}})' \hat{P}(p_t)(x - x_{\mathrm{r}}), \tag{19}$$

**Algorithm 1.** Data collection

**Input:** $M, f, \ell, \mathcal{X}, \mathcal{U}, \mathcal{X}_{\mathrm{r}}, \mathcal{U}_{\mathrm{r}}$.
**Output:** Training data $\mathbb{D}$

1: Initialization: $\mathbb{D} = \emptyset,\ \ i = 1$
2: **while** $i \leq M$ **do**
3: pick $p_t^i = \left[x_0^{i\prime}\ \ x_{\mathrm{r}}^{i\prime}\ \ u_{\mathrm{r}}^{i\prime}\right]'$ with random $x_0^i \in \mathcal{X}$ and steady-state pair $(x_{\mathrm{r}}^i, u_{\mathrm{r}}^i) \in \mathcal{X}_{\mathrm{r}} \times \mathcal{U}_{\mathrm{r}}$
4: design the terminal components $l_{\mathrm{T}}(p^i)$, $F(x, p_t^i)$ and $\mathcal{X}_{\mathrm{T}}(p_t^i)$ to construct the problem $\mathbb{P}_N(p_t^i, \mathcal{X})$
5: solve $\mathbb{P}_N(p_t^i, \mathcal{X})$ to obtain $U^i = (u_0^i, \dots, u_{N-1}^i)$, $X^i = (x_0^i, \dots, x_N^i)$
6: calculate $\ell_0^i := \ell(x_0^i, u_0^i, p_t^i)$ using (3), $\mathcal{V}_1^i := \mathcal{V}(x_1^i, p_t^i)$ using (18), $d(p_t^i)$ using (13), $C_N^i := C_N(p_t^i)$ using (12), $\mathcal{J}_{\mathrm{CDA}}^N(p_t^i) = \ell_0^i + \mathcal{V}_1^i$
7: **if** $x_0^i \notin \mathbb{D}$ and $x_0^i \in \Omega_N(p_t^i)$, that is, $\mathcal{J}_{\mathrm{CDA}}^N(p_t^i) \leq \ell_0^i + C_N^i$ **then**
8: $\mathbb{D} = \mathbb{D} \cup \{p_t^i,\ x_1^i,\ \ell_0^i,\ \mathcal{V}_1^i,\ C_N^i\}$
9: $i = i + 1$
10: **else**
11: go back to 3
12: **end if**
13: **end while**

which is positive definite by construction, provided that $\hat{P}(p_t) \succ 0$. Because $\hat{P}(p_t) : \mathbb{R}^{n_p} \to \mathbb{R}^{n_x \times n_x}$ is a function of the parameter to be learned, we enforce positive definiteness by learning the lower-triangular matrix $\hat{L} : \mathbb{R}^{n_p} \to \mathbb{R}^{n_x \times n_x}$ instead, and defining $\hat{P}$ as

$$\hat{P}(p_t) = \hat{L}(p_t)\hat{L}'(p_t) + \epsilon I, \tag{20}$$

where $\epsilon > 0$ is a small positive number and $I$ is the identity matrix. Note that, as $\hat{L}$ is lower triangular, the parameterization (19) and (20) consists of $\frac{n_x(n_x+1)}{2}$ functions of $p_t$.

*Remark* 3. Directly parameterizing the symmetric part of $\hat{P}(p_t)$, that also requires $\frac{n_x(n_x+1)}{2}$ predictors, is another valid choice, though it would require constraining the learning problem. A further valid parameterization is $\hat{P}(p_t) = \hat{L}(p_t) D \hat{L}'(p_t)$, where $\hat{L}(p_t)$ is a lower triangular matrix with entries equal to one along the main diagonal, and $D$ a proper diagonal positive definite matrix. Regardless of the chosen parameterization, complexity either appears in the form of constraints or in the parameterization itself.

In order to learn $\hat{P}$ from data, we select samples $p_t^i = \left[x_0^{i\prime}\ x_{\mathrm{r}}^{i\prime}\ u_{\mathrm{r}}^{i\prime}\right]'$, with $x_{\mathrm{r}}^i \in \mathcal{X}$, $u_{\mathrm{r}}^i \in \mathcal{U}$, and $x_0^i \in \Omega_N(p_t^i)$, for $i \in \mathbb{I}_1^M$. Then, we solve problem $\mathbb{P}_N(p_t^i, \mathcal{X})$ and store the optimal control input $u_0^i := u_{0|t}^\star$, the corresponding next optimal state $x_1^i := x_{1|t}^\star$, the optimal initial stage cost $\ell_0^i := \ell(x_0^i, u_0^i, p_t^i)$, the cost-to-go $\mathcal{V}_1^i := \mathcal{V}(x_1^i, p_t^i)$ given by (18), and the value $C_N^i := C_N(p_t^i)$ given by (12), which we collect in dataset $\mathbb{D}$ as summarized in Algorithm 1. Note that the number of iterations of such a semi-algorithmic data collection procedure depends on how stringent the condition $x_0^i \in \Omega_N(p_t^i)$ is.

In order to learn $\hat{\mathcal{V}}^{\mathrm{LLTC}}$ as in (19) and (20) using the collected dataset $\mathbb{D}$, we parameterize matrix $\hat{L}(p_t)$ using a finite-dimensional parameter $\theta$. In the following, we will denote this parameterized matrix as $\hat{L}_\theta(p_t)$. In this article, we adopt a feedforward neural network (FNN) as it is a universal approximator ,[33] which takes the following form

$$\Phi(p_t; \theta, \mathcal{H}) = [\mathcal{A}_{\mathcal{H}+1} \circ \mathcal{G}_{\theta_{\mathcal{H}+1}} \circ \mathcal{A}_{\mathcal{H}} \circ \mathcal{G}_{\theta_{\mathcal{H}}} \circ \cdots \circ \mathcal{A}_1 \circ \mathcal{G}_{\theta_1}](p_t), \tag{21}$$

where $\Phi(p_t; \theta, \mathcal{H})$ yields the vectorized form of $\hat{L}_\theta(p_t)$. The function is therefore obtained as a sequence of layers in which function $\mathcal{G}_{\theta_{\mathcal{H}}}(z_{h-1})$ is affine, takes as input the output $z_{h-1}$ of layer $h - 1$, and is composed with an activation function $\mathcal{A}_h$ to yield the output $z_h$ of layer $h$ as

$$\mathcal{G}_{\theta_h}(z_{h-1}) = w_h z_{h-1} + b_h,$$
$$z_h = \mathcal{A}_h(\mathcal{G}_{\theta_h}(z_{h-1})), \quad h \in \mathbb{I}_1^{\mathcal{H}+1}$$

with $z_0 = p_t$ and $\theta \in \mathbb{R}^{n_\theta}$

$$\theta = [\theta_1, \theta_2, \dots, \theta_{\mathcal{H}+1}],$$

with $\theta_h = [w_h, b_h]$ containing weights $w_h \in \mathbb{R}^{n_n^{h-1} \times n_n^h}$ and biases $b_h \in \mathbb{R}^{n_n^h}$, where $n_n^h$ stands for a number of neurons in the layer $h$. Additional details on neural networks are discussed in Reference 34. In this work, we consider a rectified linear unit (ReLU) activation function

$$\mathcal{A}_h(\mathcal{G}_{\theta_h}) = \text{ReLU}(\mathcal{G}_{\theta_h}) := \max\{\mathcal{G}_{\theta_h}, 0\}, \quad h \in \mathbb{I}_1^{\mathcal{H}}. \tag{22}$$

As schematically illustrated in Figure 1, we employ the network $\Phi(p; \theta, \mathcal{H})$ defined in (21) to parameterize the non-zero entries of matrix $\hat{L}_\theta(p_t)$. Hence, we rewrite (19) and (20) in the following form

$$\hat{\mathcal{V}}_\theta^{\text{LLTC}}(x, p_t) = (x - x_r)' \hat{P}_\theta(p_t)(x - x_r), \tag{23}$$

$$\hat{P}_\theta(p_t) = \hat{L}_\theta(p_t)\hat{L}_\theta'(p_t) + \epsilon I. \tag{24}$$

Finally, we obtain the following finite-dimensional learning problem

$$\min_\theta \gamma \|\theta\|_2^2 + \frac{1}{M}\sum_{i=1}^{M} \phi(\mathcal{V}_1^i, \hat{\mathcal{V}}_\theta^{\text{LLTC}}(x_1^i, p_t^i)), \tag{25a}$$

$$\text{s.t.} \quad \hat{\mathcal{V}}_\theta^{\text{LLTC}}(x_1^i, p_t^i) \le C_N^i \qquad i \in \mathbb{I}_1^M, \tag{25b}$$

$$\hat{\mathcal{V}}_\theta^{\text{LLTC}}(x_1^i, p_t^i) - \hat{\mathcal{V}}_\theta^{\text{LLTC}}(x_0^i, p_t^i) + \ell_0^i \le 0 \qquad i \in \mathbb{I}_1^M, \tag{25c}$$

where $\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a suitably defined loss function; constraint (25b) ensures the positive invariance property of the domain of attraction $\Omega_N(p_t^i)$; and constraint (25c) enforces the decrease of the Lyapunov terminal cost for all observed samples, and an $L_2$-regularization with parameter $\gamma$ is introduced to prevent overfitting.

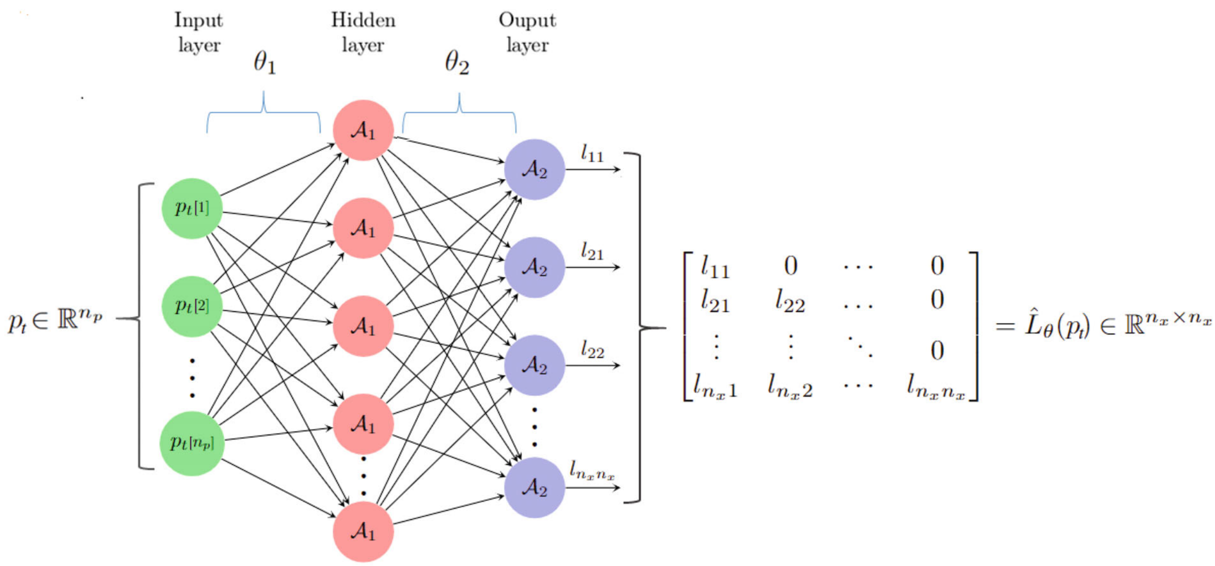


**FIGURE 1** A schematic illustration of FNN $\Phi(p_t; \theta, \mathcal{H} = 1)$ to predict the matrix $\hat{L}_\theta(p_t)$ for a given input $p_t$.

**Algorithm 2.** LLTC-NMPC scheme

**Input:** $x_{\mathrm{r}}, u_{\mathrm{r}}, \theta^\star \in \mathbb{R}^{n_\theta}, \Phi(p_t;\theta^\star,\mathcal{H}), \epsilon > 0$
**Output:** $u_t$

1: get $p_t = [x_t', x_{\mathrm{r}}', u_{\mathrm{r}}']' \in \mathcal{P}$
2: obtain $\hat{L}_{\theta^\star}(p_t)$ from $\Phi(p_t;\theta^\star,\mathcal{H})$
3: evaluate $\hat{P}_{\theta^\star}(p_t) = \hat{L}_{\theta^\star}(p_t)\hat{L}_{\theta^\star}'(p_t) + \epsilon I$
4: solve problem $\mathbb{P}_{\mathrm{LLTC}}(p_t, \mathcal{X})$ and obtain $u_{0|t}^\star$
5: apply $u_t = u_{0|t}^\star$ to system (1)
6: set $t = t+1$ and go back to 1.

For practical purposes, as this allows us to use standard learning algorithms, we reformulate the problem above as an unconstrained problem by using $\ell_1$-penalties on constraint violations, as proposed in Reference 35. This yields the following unconstrained learning problem

$$\theta^\star := \arg\min_{\theta} \ \gamma\|\theta\|_2^2 + \frac{1}{M}\sum_{i=1}^{M}\phi(\mathcal{V}_1^i, \hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p_t^i)) + \lambda_1\mathcal{E}_{\mathrm{dom}}^i(\theta) + \lambda_2\mathcal{E}_{\mathrm{dec}}^i(\theta), \tag{26}$$

where

$$\mathcal{E}_{\mathrm{dom}}^i(\theta) := \max\left\{\hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p_t^i) - C_N^i, 0\right\}$$
$$\mathcal{E}_{\mathrm{dec}}^i(\theta) := \max\left\{\hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p_t^i) - \hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_0^i, p_t^i) + \ell_0^i, 0\right\},$$

and $\lambda_1$, $\lambda_2$ are sufficiently large positive penalty parameters to ensure that the optimal solution matches the one of problem (25), if the latter exists.

### 3.2 | LLTC-NMPC

Once the Lyapunov terminal cost $\hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x, p_t)$ is learned, the data-driven NMPC optimization problem is formulated as:

$$\begin{aligned} \mathcal{J}_{\mathrm{LLTC}}(p_t) = \min_{u_{0|t}, x_{0|t}, x_{1|t}} \quad & \ell(x_{0|t}, u_{0|t}, p_t) + \hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x_{1|t}, p_t) \\ \text{s.t.} \quad & x_{0|t} = \mathcal{M}_x p_t \\ & x_{1|t} = f(x_{0|t}, u_{0|t}) \\ & u_{0|t} \in \mathcal{U} \\ & x_{1|t} \in \mathcal{X}, \end{aligned} \tag{27}$$

which we call NMPC with learned Lyapunov terminal cost (LLTC-NMPC) and denote problem (27) as $\mathbb{P}_{\mathrm{LLTC}}(p_t, \mathcal{X})$. The LLTC-NMPC control strategy at time $t$ is summarized in Algorithm 2. As opposed to standard NMPC approaches, once the initial state is known, we first evaluate the FNN (21) to define the terminal cost and then solve the 1-step-ahead NMPC problem (27).

## 4 | ILLUSTRATIVE EXAMPLES

We consider two benchmark optimal control problems, the first related to controlling a chemical process and the second to an autonomous driving application. To demonstrate the potential of the proposed method, we perform simulations that show the superiority of our LLTC-NMPC in terms of computational time as compared to classic NMPC and CDA-NMPC, in which the terminal conditions, that is, $l_{\mathrm{T}}(p_t)$, $\mathcal{X}_{\mathrm{T}}(p_t)$, and $F(x, p_t)$ satisfying Assumption 4 for all $p_t \in \mathcal{P}$ are chosen by following the procedures proposed in Reference 28.

The computational performance is compared based on the worst-case $\mathcal{T}_w$ and the average-case $\mathcal{T}_a$ execution time defined as

$$\mathcal{T}_a := \frac{1}{N_{\mathrm{init}}} \sum_{i=1}^{N_{\mathrm{init}}} \frac{1}{N_{\mathrm{sim}}} \sum_{t=0}^{N_{\mathrm{sim}}} \tau(p_t^i), \tag{28a}$$

$$\mathcal{T}_w := \max_{1,\ldots,N_{\mathrm{init}}} \max_{1,\ldots,N_{\mathrm{sim}}} \tau(p_t^i), \tag{28b}$$

where $N_{\mathrm{sim}}$ is the simulation length, $N_{\mathrm{init}}$ the number of simulations from different initial conditions $x_0^i$, and $\tau(p_t^i)$ the execution time measured at each time step $t$ to solve the NMPC problem for a given initial condition $x_0^i$, $i = 1, \ldots, N_{\mathrm{init}}$.

In order to demonstrate the necessity of incorporating constraints (25b) and (25c) while learning $\hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x, p_t)$, we make a comparison with the approach proposed in Reference 24, called learned terminal cost NMPC (LTC-NMPC), which also learns the terminal cost from data, but does not include (25b) and (25c).

We employ CasADi [36] via its MATLAB interface to formulate all NMPC problems and automatically generate the corresponding nonlinear programming (NLP) problems. These are solved using Ipopt ,[37] with computational time $\tau(p_t^i)$ measured on an Intel Core i5-5200U (2.7 GHz) processor. The continuous-time systems are discretized using a fourth-order explicit Runge–Kutta integrator. The computational results might be further improved by exploiting specialized libraries for real-time MPC such as, for example, Acados .[38]

To make a fair comparison without exaggerating the computational cost required to solve CDA-NMPC for each given initial condition, the prediction horizon is selected as follows: $N_{\mathrm{init}}$ random initial conditions $p_t^i$ are generated; for each $N \in [10,100]$, the closed-loop cost over $N_{\mathrm{sim}}$ time steps is computed by solving the problem $\mathbb{P}_N(p_t^i, \mathcal{X}_{\mathrm{T}}(p_t^i))$ for all $p_t^i$; the prediction horizon $N$ is selected such that longer prediction horizons do not yield significant performance improvements.

The $M$ data samples for learning the approximate Lyapunov terminal cost function $\hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x, p_t)$ are generated by executing Algorithm 1. We use $M_{\mathrm{train}} = 0.8M$ training samples for learning the FNN parameters using the quadratic loss function $\phi(z_1, z_2) = (z_1 - z_2)^2$. The remaining $M_{\mathrm{test}} = 0.2M$ test samples are used to assess the performance of the learned network model in terms of normalized root mean squared error (NRMSE) and $R^2_{\mathrm{score}}$ (coefficient of determination). Additionally, we count the number $C_{\mathrm{dom}}$ and $C_{\mathrm{dec}}$ of samples for which constraints (25b) and (25c) are, respectively, violated. Finally, we also compare the computational time $\mathcal{T}_w$ and $\mathcal{T}_a$.

For a given complexity, quantified as the total number of neurons in the network $\Phi(p_t; \theta, \mathcal{H})$, we analyze the influence of the layer structure on the approximation quality of $\hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x, p_t)$. In all considered networks, we adopt the ReLU$(\cdot)$ activation function (22). The neural networks are trained in Pytorch ,[39] using Adam [40] to solve problem (26) over 5000 epochs with penalty parameters $\lambda_1 = \lambda_2 = 10^4$, learning rate $\alpha = 10^{-3}$, first and second moment decay rates $\beta = (0.995, 0.999)$, and $L_2$-regularization parameter $\gamma = 10^{-6}$. We refer the reader to Reference 41 for a wide range of possible alternative loss functions, validation metrics, and numerical optimization algorithms.

## 4.1 | CSTR problem

For an exothermic and irreversible reaction $a \rightarrow b$ with constant liquid volume $V = 100$ l and flow rate $q = 100$ l/ min, a continuous time stirred tank reactor (CSTR) model based on a component balance for reactant $a$ is given as follows [42]

$$\dot{C}_a = \frac{q}{V}(C_a^n - C_a) - C_a k_0 \exp^{-\frac{E}{RT}}$$

$$\dot{T} = \frac{q}{V}(T_a^n - T) - C_a \frac{\Delta H}{\rho C_p} k_0 \exp^{-\frac{E}{RT}} + \frac{UA}{\rho V C_p}(T_c - T),$$

where $C_a$ is the concentration of the reactant $a$ in the reactor, $T$ is the reactor temperature, and $T_c$ is the temperature of the coolant stream. The unstable steady state $C_a^{\mathrm{r}} = 0.5$ mol/l, $T^{\mathrm{r}} = 350$ K, and $T_c^{\mathrm{r}} = 300$ K is chosen under the following nominal operating conditions: $C_a^n = 1$ mol/l and $T_a^n = 350$ K, $k_0 = 7.2 \times 10^{10}$ min$^{-1}$, $E/R = 8750$ K, $\Delta H = -5 \times 10^4$ J/mol, $\rho = 10^3$ g/l, $C_p = 0.239$ J/g K, and $UA = 5 \times 10^4$ J/ min K. A nonlinear discrete-time state-space model is obtained with sampling time $T_s = 0.03$ min. The state and the commanded input of the system, together with the corresponding reference signals, are defined as $x = [C_a \ \ T]'$, $x_r = [C_a^r \ \ T^r]'$, $u = T_c$ and $u_r = T_c^r$, respectively. The system is subject to the box constraints

$$\mathcal{U} = [280, \ 370] \qquad \mathcal{X} = [0,1] \times [280,370],$$

where the units are omitted as they are clear from context.

The MPC problems are formulated using the stage cost defined in (3) with weight matrices $Q_x = \text{diag}[50, 1]$ and $Q_u = 1$. Our prediction horizon selection procedure yields $N = 50$. Moreover, we obtain $C_N(p_t) = 2.01 \times 10^3$ from (12), with $l_T(p_t) = 1.77 \times 10^3$ and $d(p_t) = 4.53$. The classic NMPC and the CDA-NMPC schemes that solve problems $\mathbb{P}_N(p_t, \mathcal{X}_T(p_t))$ and $\mathbb{P}_N(p_t, \mathcal{X})$, respectively, are designed with the ellipsoidal terminal set $\mathcal{X}_T(p_t) = \{x \in \mathcal{X} : F(x, p_t) \leq 1.77 \times 10^3\}$ where the terminal cost $F(x, p_t)$ in (4) is quadratic with Hessian matrix

$$P = \begin{bmatrix} 83223 & 1735 \\ 1735 & 62 \end{bmatrix}.$$

We collect a dataset consisting of $M = 8000$ samples; the corresponding state trajectories are shown in Figure 2.

We learn matrix $\hat{P}_{\theta^\star}(p_t)$ defining $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x, p_t)$ as in (23) using $\epsilon = 0.02$. For this problem, the FNN has $n_p = 5$ inputs and $n_o = 3$ outputs. For a given budget of 120 neurons, we test different FNN structures with $\mathcal{H}$ hidden layers each having $n^h$ neurons, as reported in Table 1. The table shows that the structure with 3 hidden layers, each containing 40 neurons, yields the best model to represent $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x, p_t)$, since there is no stability constraint violation and small NRMSE and $R^2_{\text{score}}$ are obtained.

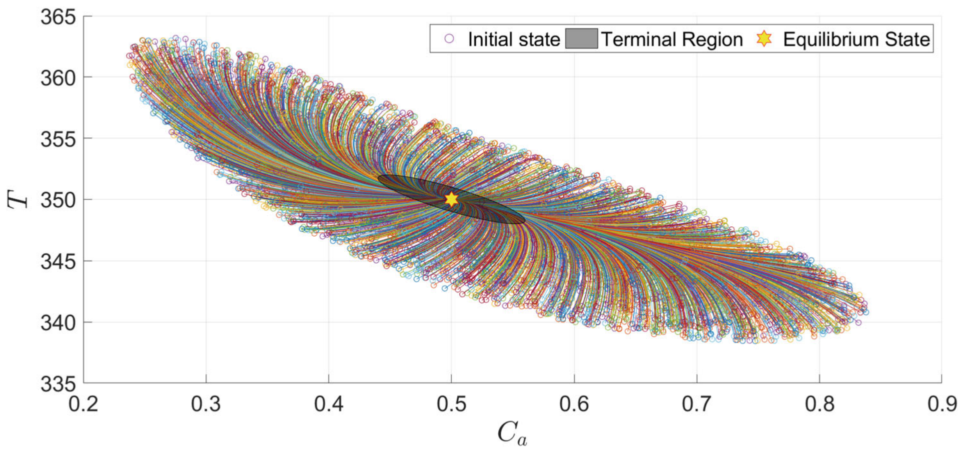


**FIGURE 2** State trajectory samples generated by CDA-NMPC with $N = 50$ for a given initial state $x_0^i$ based on the Algorithm 1.

**TABLE 1** Analysis of trained Lyapunov terminal-cost models based on different FNN structures.

| FNN model architecture | Fit quality | | Violation of stability constraints | |
|---|---|---|---|---|
| | NRMSE | $R^2_{\text{score}}$ | $C_{\text{dom}}$ | $C_{\text{dec}}$ |
| $\mathcal{H} = 1, n^h = 120$ | 0.034/0.046 | 0.906/0.882 | 450/75 | 900/132 |
| $\mathcal{H} = 2, n^h = 60$ | 0.023/0.027 | 0.918/0.912 | 400/50 | 600/68 |
| $\mathcal{H} = 3, n^h = 40$ | 0.015/0.017 | 0.956/0.951 | 0/0 | 0/0 |
| $\mathcal{H} = 4, n^h = 30$ | 0.014/0.015 | 0.964/0.960 | 0/0 | 175/25 |

*Note*: For each metric (NRMSE, $R^2_{\text{score}}$, $C_{\text{dom}}$, $C_{\text{dec}}$), the values obtained on train/test data are reported.

All controllers are simulated over $N_{\text{sim}} = 100$ steps from $N_{\text{init}} = 10$ different initial states. The corresponding state trajectories are displayed in Figure 3, where it can be seen that the classic NMPC and CDA-NMPC schemes yield indistinguishable trajectories, while the LLTC-NMPC trajectories only slightly differ due to the imperfect approximations of the cost-to-go defined in (18).

The computational time required by each controller is shown in Figure 4. As the evaluation time of the FNN yielding $\hat{P}_{\theta^\star}(p_t)$ is 0.023 ms, that is, essentially negligible, the proposed LLTC-NMPC scheme yields a significantly lower computational time compared to CDA-NMPC and Classic NMPC.

In order to discuss the importance of including Lyapunov conditions in the learning the cost-to-go approximation $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x, p_t)$, we compare our LLTC-NMPC with the LTC-NMPC approach proposed in Reference 24, which approximates $\mathcal{V}(x, p_t)$ by solving problem (25) without constraints (25b) and (25c). The terminal cost $\hat{\mathcal{V}}^{\text{LTC}}_{\theta^\star}(x, p_t)$ is modeled with the same FNN structure used for $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x, p_t)$. As shown in Figure 3, the LTC-NMPC yields trajectories that are closer to CDA-NMPC than LLTC-NMPC: because of the absence of the Lyapunov constraints, $\hat{\mathcal{V}}^{\text{LTC}}_{\theta^\star}(x, p_t)$ of $\mathcal{V}(x, p_t)$ yields a more accurate cost-to-go approximation than $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x, p_t)$. However, LTC-NMPC violates the stability constraints (25b) and (25c) 24 and 75 times respectively, while LLTC-NMPC never violates any constraint.

## 4.2 | Autonomous parking problem

In order to demonstrate the effectiveness of the proposed control scheme also for fast-sampling mechanical systems, we consider next a vehicle parking problem.

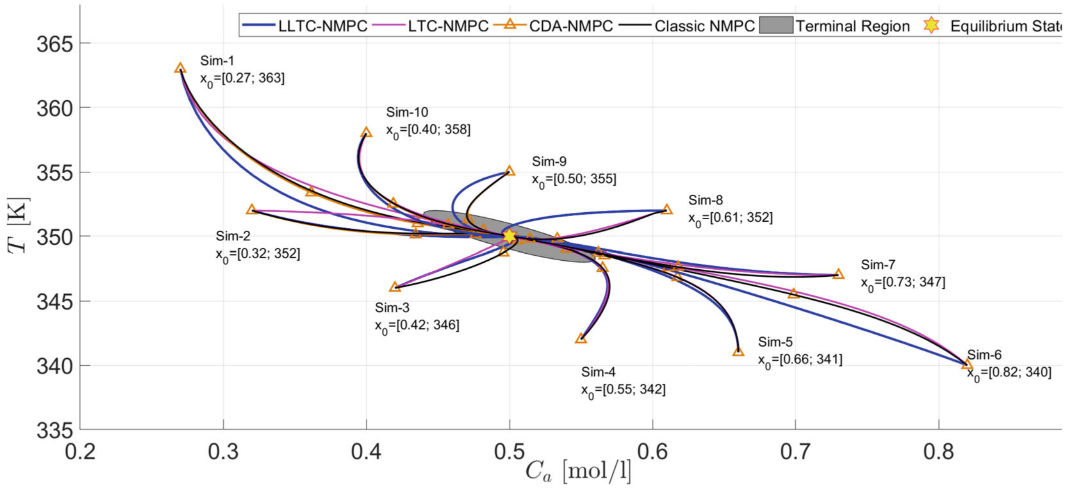


**FIGURE 3** State trajectories of different NMPC schemes starting from an initial state $x_0$.

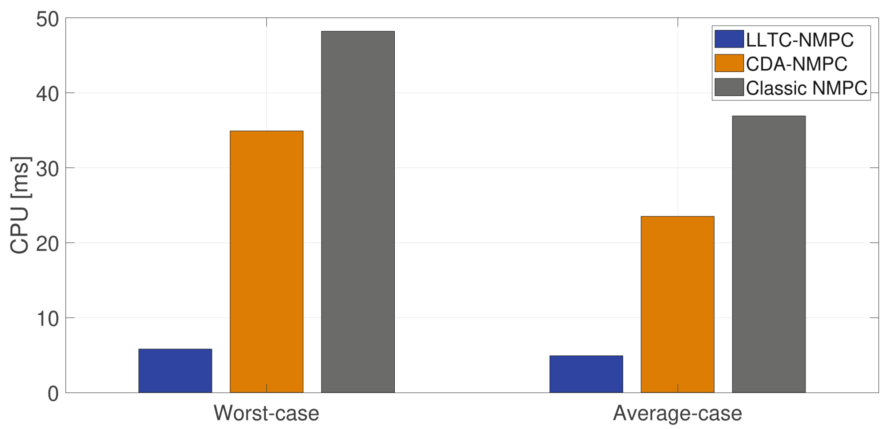


**FIGURE 4** Average and worst-case computational time comparison.

Consider a vehicle with nonlinear dynamics $\dot{x} = f(x, u)$ with $x = [s_x \; s_y \; \psi]'$; $u = [v \; \delta]'$ and $f$ given by the following continuous-time bicycle model $f : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$

$$\begin{cases} \dot{s}_x = v\cos(\psi) \\ \dot{s}_y = v\sin(\psi) \\ \dot{\psi} = \delta, \end{cases} \tag{29}$$

where $(s_x, s_y)$ denote the Cartesian position of the vehicle on a fixed reference frame, $\psi$ the orientation of the vehicle with respect to the $x$-axis, $v$ the longitudinal velocity, and $\delta$ the angular velocity. The system is discretized with sampling time $T_s = 0.1$ s. The control task consists in driving the vehicle towards the reference point $x_r = [0.3 \; 0.5 \; \pi/4]'$ and $u_r = [0 \; 0]'$ within the box constraints defined by

$$\mathcal{U} = [-0.2, 0.6] \times [-\pi/3, \pi/3]$$
$$\mathcal{X} = [-2, 2] \times [-2, 2] \times [-\pi, \pi],$$

(all units are in the SI).

The stage cost function is defined by matrices $Q_x = \text{diag}[3 \; 5 \; 0.01]$ and $Q_u = \text{diag}[0.5 \; 0.01]$. CDA-NMPC is designed with $N = 60$, $d(p_t) = 0.32$, and the terminal set $\mathcal{X}_T(p_t) = \{x \in \mathcal{X} : F(x, p_t) \leq 1.31\}$, which yields $C_N(p_t) = 20.19$. We generate $M = 5000$ data samples and learn $\hat{\mathcal{V}}_{\theta^\star}^{\text{LLTC}}(x, p_t)$ and $\hat{\mathcal{V}}_{\theta^\star}^{\text{LTC}}(x, p_t)$ with fit results NRMSE $= 0.11/0.13$ and $R^2_{\text{score}} = 0.88/0.86$, and NRMSE $= 0.06/0.09$ and $R^2_{\text{score}} = 0.961/0.949$ for the train/test datasets, respectively. The matrix $\hat{P}_{\theta^\star}(p_t)$ is defined using $\epsilon = 10^{-3}$, and the FNN has $n_p = 8$ inputs, $n_o = 6$ outputs and $\mathcal{H} = 3$ hidden layers with $n^h = 15$ neurons each.

We simulate the system in closed loop over $N_{\text{sim}} = 80$ steps, starting from $N_{\text{init}} = 15$ initial states picked outside the terminal region $\mathcal{X}_T(p_t)$. LLTC-NMPC steers the system to the reference by respecting the stability constraints, while LTC-NMPC violates them. As shown in Figure 5, the terminal cost $\hat{\mathcal{V}}_{\theta^*}^{\text{LTC}}$ is not a Lyapunov function as it does not always decrease.

The computational times of LLTC-NMPC and CDA-NMPC are reported in Table 2, where it can be seen that the proposed data-driven approach requires a significantly shorter execution time than CDA-NMPC: even though one needs

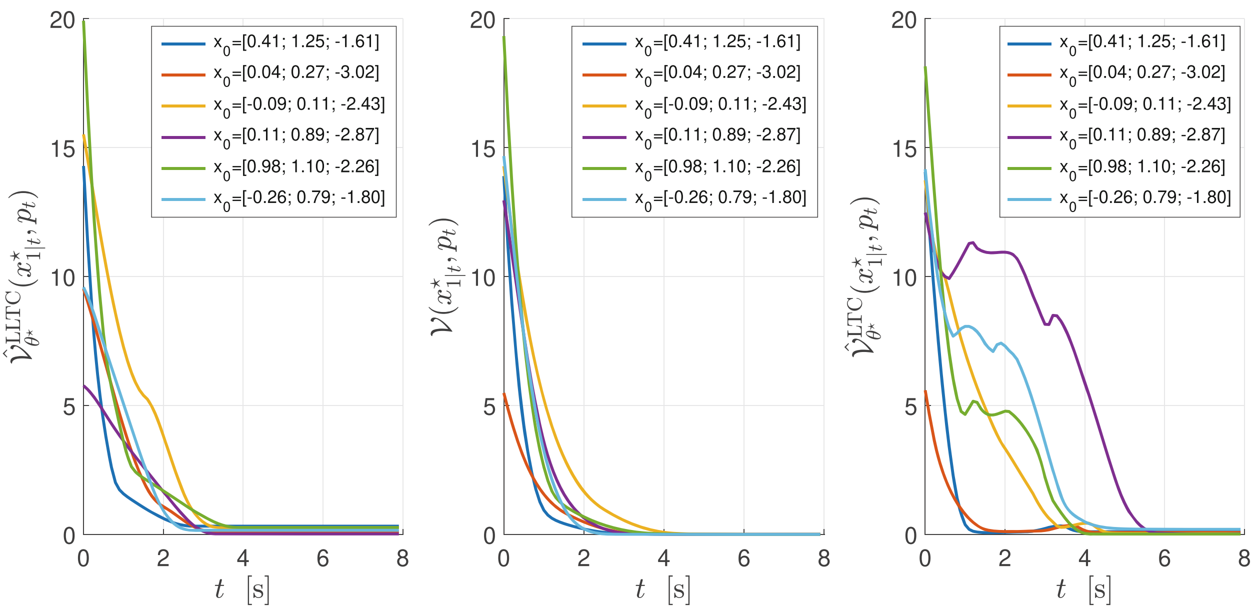


**FIGURE 5** Terminal cost of LLTC-NMPC, CDA-NMPC and LTC-NMPC for a set of given initial states $x_0^i$ over the simulation horizon $N_{\text{sim}} = 80$.

**TABLE 2** Computational performance of LLTC-NMPC and CDA-NMPC.

| | NLP dimension | | Convergence | Computational time [ms] | | |
|---|---|---|---|---|---|---|
| Controller | # decision variables | # constraints | # iterations | FNN | $\mathcal{T}_w$ | $\mathcal{T}_a$ |
| CDA-NMPC with $N = 60$ | 303 | 183 + 600 | 23 | - | 73.151 | 55.530 |
| LLTC-NMPC | 8 | 6 + 10 | 7 | 0.017 | 8.321 | 6.124 |

to also evaluate the FNN in LLTC-NMPC, the LLTC-NMPC computational time is approximately 9 times smaller than the one of CDA-NMPC.

## 5 | CONCLUSIONS

This work presented a computationally efficient data-driven NMPC technique that employs a one-step-ahead prediction horizon in combination with a LLTC. The proposed control scheme successfully approximates a given long-horizon NMPC problem and satisfies Lyapunov stability conditions on a given training dataset. For two benchmark problems, our proposed method yields good control performance while requiring significantly smaller computational time compared to long-horizon NMPC.

Future work will be devoted to extending the proposed approach to generalize the approach to control tasks with more general costs, for example, economic costs and time-varying references.

### ACKNOWLEDGMENTS

This article was partially supported by the Italian Ministry of University and Research under the PRIN'17 project "Data-driven learning of constrained control systems," contract No. 2017J89ARP.

### CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

### DATA AVAILABILITY STATEMENT

No external data used. All data is used in simulations that are reproducible by reading the paper.

### ORCID

*Mario Zanon* https://orcid.org/0000-0001-5925-0440
*Alberto Bemporad* https://orcid.org/0000-0001-6761-0856

### REFERENCES

1. Grüne L, Pannek J. *Nonlinear Model Predictive Control*. 2nd ed. Springer International Publishing; 2017.
2. Michalska H, Mayne D. Receding horizon control of nonlinear systems. *Proceedings of the 28th IEEE Conference on Decision and Control*. Vol 1. IEEE; 1989:107-108.
3. Rawlings J, Mayne D, Diehl M. *Model Predictive Control: Theory, Computation and Design*. Nob Hill Publishing; 2019.
4. Chen H, Allgöwer F. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*. 1998;34(10):1205-1217.
5. Chen S, Saulnier K, Atanasov N, et al. Approximating explicit model predictive control using constrained neural networks. *2018 Annual American Control Conference (ACC)*. IEEE; 2018:1520-1527.
6. Hertneck M, Köhler J, Trimpe S, Allgöwer F. Learning an approximate model predictive controller with guarantees. *IEEE Control Syst Lett*. 2018;2(3):543-548.
7. Lucia S, Karg B. A deep learning-based approach to robust nonlinear model predictive control. *IFAC-PapersOnLine*. 2018;51:511-516.
8. Gersnoviez A, Brox M, Baturone I. High-speed and low-cost implementation of explicit model predictive controllers. *IEEE Trans Control Syst Technol*. 2019;27(2):647-662.
9. Nematollah Changizi KS, Siahi M. Complexity reduction of explicit MPC based on fuzzy reshaped polyhedrons for use in industrial controllers. *Int J Syst Sci*. 2023;54(3):463-477.
10. Grancharova A, Johansen T. Nonlinear model predictive control. *Explicit Nonlinear Model Predictive Control: Theory and Applications*. Springer; 2012:43-66.
11. Cagienard R, Grieder P, Kerrigan E, Morari M. Move blocking strategies in receding horizon control. *J Process Control*. 2007;17(6):563-570.
12. Shekhar R, Manzie C. Optimal move blocking strategies for model predictive control. *Automatica*. 2015;61:27-34.

13. Makarow A, Rösmann C, Bertram T. Suboptimal nonlinear model predictive control with input move-blocking. *Int J Control*. 2022;97:450-459.
14. Muehlebach M, D'Andrea R. A method for reducing the complexity of model predictive control in robotics applications. *IEEE Robot Autom Lett*. 2019;4(3):2516-2523.
15. Lawrynczuk M. Nonlinear model predictive control for processes with complex dynamics: a parameterisation approach using Laguerre functions. *Int J Appl Math Comput Sci*. 2020;30(1):35-46.
16. Pan G, Faulwasser T. NMPC in active subspaces: dimensionality reduction with recursive feasibility guarantees. *Automatica*. 2023;147:110708.
17. Boggio M, Novara C, Taragna M. Nonlinear model predictive control: an optimal search domain reduction. *IFAC-PapersOnLine*. 2023;56(2):6253-6258.
18. Ohtsuka T. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*. 2004;40:563-574.
19. Diehl M, Ferreau H, Haverbeke N. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Magni L, Raimondo DM, Allgöwer F, eds. *Nonlinear Model Predictive Control*. Lecture Notes in Control and Information Sciences. Springer; 2009:391-417.
20. Limon D, Alamo T, Salas F, Camacho E. On the stability of constrained MPC without terminal constraint. *IEEE Trans Automat Contr*. 2006;51(5):832-836.
21. Giesl P, Hafstein S. Review on computational methods for Lyapunov functions. *Discrete Continuous Dyn Syst Ser B*. 2015;20:2291-2331.
22. Mittal M, Gallieri M, Quaglino A, Salehian S, Koutnik J. Neural Lyapunov model predictive control. arXiv preprint arXiv:abs/2002.10451, 2020.
23. Moreno-Mora F, Beckenbach L, Streif S. Predictive control with learning-based terminal costs using approximate value iteration. arXiv preprint arXiv:abs/2212.00361, 2022.
24. Abdufattokhov S, Zanon M, Bemporad A. Learning convex terminal costs for complexity reduction in MPC. *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE; 2021:2163-2168.
25. Turan E, Mdoe Z, Jäschke J. Learning convex objectives to reduce the complexity of model predictive control. arXiv preprint arXiv:abs/2312.02650, 2023.
26. Zieger T, Savchenko A, Oehlschlägel T, Findeisen R. Towards safe neural network supported model predictive control. *IFAC-PapersOnLine*. 2020;53:5246-5251.
27. Karg B, Lucia S. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Trans Cybern*. 2020;50(9):3866-3878.
28. Rajhans C, Patwardhan S, Pillai H. Discrete time formulation of quasi infinite horizon nonlinear model predictive control scheme with guaranteed stability. *IFAC-PapersOnLine*. 2017;50:7181-7186.
29. Mayne D, Rawlings J, Rao C, Scokaert P. Constrained model predictive control: stability and optimality. *Automatica*. 2000;36(6):789-814.
30. Prokhorov D, Feldkamp L. Application of support vector machines to Lyapunov function approximation. *International Joint Conference on Neural Networks Proceedings (Cat No. 99CH36339)*. Vol 1. IEEE; 1999:383-387.
31. Serpen G. Empirical approximation for Lyapunov functions with artificial neural nets. *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*. Vol 2. IEEE; 2005:735-740.
32. Gaby N, Zhang F, Ye X. Lyapunov-net: a deep neural network architecture for Lyapunov function approximation. arXiv preprint arXiv:abs/2109.13359, 2021.
33. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Netw*. 1989;2:359-366.
34. Aggarwal C. *Neural Networks and Deep Learning: A Textbook*. Springer; 2018.
35. Zangwill W. Nonlinear programming via penalty functions. *Manag Sci*. 1967;13(5):344-358.
36. Andersson J, Gillis J, Horn G, Rawlings J, Diehl M. CasADi—a software framework for nonlinear optimization and optimal control. *Math Program Comput*. 2019;11(1):1-36.
37. Wächter A, Biegler L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program*. 2006;106:25-57.
38. Verschueren R, Frison G, Kouzoupis D, et al. acados: a modular open-source framework for fast embedded optimal control. *Math Program Comput*. 2019;14:147-183.
39. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.; 2019:8024-8035.
40. Kingma D, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:abs/1412.6980, 2014.
41. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016 http://www.deeplearningbook.org
42. Magni L, de Nicolao G, Magnani L, Scattolini R. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*. 2001;37:1351-1362.


**How to cite this article:** Abdufattokhov S, Zanon M, Bemporad A. Learning Lyapunov terminal costs from data for complexity reduction in nonlinear model predictive control. *Int J Robust Nonlinear Control*. 2024;1-16. doi: 10.1002/rnc.7411