Learning Lyapunov Terminal Costs from Data for Complexity Reduction in Nonlinear Model Predictive Control

Shokhjakhon Abdufattokhov*, Mario Zanon, Alberto Bemporad

***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 paper, 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.**

***Index Terms*—Data-driven control, nonlinear model predictive control, constrained systems, neural networks.**

I. 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 methods [5]–[8], although preserving feasibility with respect to system constraints can be challenging [9]. 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 [10]–[12]; by parameterizing the decision variables with basis functions [13], [14]; by decomposing the control input space to construct low-dimensional input subspaces [15]; or by developing approximate but efficient optimization algorithms [16], [17]. 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 [18] 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 [19]. One possibility to tackle this difficulty is to resort to data-driven approaches, such as in, e.g., [20], [21]. Despite obtaining some form of stability and recursive feasibility guarantees, attaining good closed-loop performance with a short horizon remains an open issue.

This paper proposes an implicit NMPC formulation with a prediction horizon of length one, which retains good closed-loop performance and satisfies Lyapunov stability conditions on a given dataset. We combine the approach of [18] with our recent work [22], where the terminal cost for an MPC scheme with prediction horizon one is learned from data. A feedforward neural network is used to fit given samples of the cost-to-go associated with the original NMPC formulation. In addition to our previous work, starting from a long-horizon NMPC scheme, we obtain a horizon-one NMPC reformulation by constructing a terminal cost from data that satisfies the Lyapunov conditions on the data samples and is a good approximation of the cost-to-go of the original problem. 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 [23], [24], our approach has the advantage of keeping the constraints on the system variables at the first step in the optimization problem. 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 paper is organized as follows. After providing preliminary results and formulations leading to an NMPC optimization problem with a single prediction horizon in Section II, we detail how to learn a quadratic Lyapunov approximation of the cost-to-go function and discuss

All authors are from IMT School for Advanced Studies in Lucca, Italy. *Corresponding author: s.abdufattokhov@imtlucca.it. This paper 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.

the proposed LLTC-NMPC formulation in Section III. The effectiveness of the proposed approach is demonstrated in simulations in Section IV. Finally, concluding remarks are drawn in Section V.

II. 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 $\mathcal{U}$.

Assumption 1. *Function $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is continuously differentiable over $\mathcal{X} \times \mathcal{U}$ and $f(0, 0) = 0$.*

A. Classical NMPC

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

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

where $p_t \in \mathbb{R}^{n_p}$ is a vector of parameters, $p_t = [x_t' \; x_\mathrm{r}' \; u_\mathrm{r}']'$, belonging to a bounded set $\mathcal{P}$ that consists of the current state $x_t \in \mathcal{X}$ and constant reference signals $x_\mathrm{r} \in \mathcal{X}$ and $u_\mathrm{r} \in \mathcal{U}$, $\mathcal{M}_x = [I \; 0 \; 0]$, $U_t = (u_{0|t}, \ldots, u_{N-1|t})$ 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 $[a, b]$.

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

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

where $\|z\|_Q^2 = z'Qz$, Q_x and Q_u are positive definite matrices, $\mathcal{X}_\mathrm{T}(p) \subseteq \mathcal{X}$ 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

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

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.

In what follows, $\mathcal{F}_N(p_t)$ defines the set of initial states for which Problem $\mathbb{P}_N(p_t, \mathcal{X}_\mathrm{T}(p_t))$ is feasible. Let $U_t^\star = (u_{0|t}^\star, \ldots, u_{N-1|t}^\star)$ denote the optimal control inputs and $X_t^\star = (x_{0|t}^\star, \ldots, x_{N|t}^\star)$ the corresponding state trajectory 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).

Assumption 2. *For each $p_t \in \mathcal{P}$ there exists an optimal sequence $U_t^\star$ from Problem $\mathbb{P}_N(p_t, \mathcal{X}_\mathrm{T}(p_t))$.*

Definition 1. *A function $\mu : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is $\mathcal{K}_\infty$ if it is continuous, strictly increasing, unbounded, and zero at the origin.*

Assumption 3. *For all $p \in \mathcal{P}$, there exists a $\mathcal{K}_\infty$-function $\mu_1(\cdot)$ such that*

$$\mu_1(\|x - x_\mathrm{r}\|) \leq \ell(x, u, p) \quad \forall x \in \mathcal{F}_N(p), \; \forall u \in \mathcal{U} \tag{5}$$

The terminal cost and the terminal constraint are usually chosen such that the following assumption holds, in order to guarantee closed-loop stability.

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

$$\mathcal{X}_\mathrm{T}(p) = \{x \in \mathcal{X} : F(x, p) \leq l_\mathrm{T}(p)\}$$

be such that F is a Lyapunov function associated to a local stabilizing controller $u_\mathrm{T}(x, p) : \mathcal{X}_\mathrm{T}(p) \times \mathcal{P} \to \mathcal{U}$, i.e., $\forall x \in \mathcal{X}_\mathrm{T}(p)$ and $f(x, u_\mathrm{T}(x, p)) \in \mathcal{X}_\mathrm{T}(p)$ it holds that

$$\mu_2(\|x - x_\mathrm{r}\|) \leq F(x, p) \leq \mu_3(\|x - x_\mathrm{r}\|) \tag{6}$$

$$F(f(x, u_\mathrm{T}(x, p)), p) - F(x, p) + \ell(x, u_\mathrm{T}(x, p), p) \leq 0 \tag{7}$$

where μ_2 and μ_3 are $\mathcal{K}_\infty$ functions.

Under Assumptions 1-3, the authors of [25] 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 $\mathcal{X}_\mathrm{T}(p_t)$ 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 $\mathbb{P}_N(p_t, \mathcal{X}_\mathrm{T}(p_t))$.

B. NMPC without terminal constraint

The authors of [18] 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, i.e., an MPC problem $\mathbb{P}_N(p_t, \mathcal{X})$ defined as follows:

$$\mathcal{J}_\mathrm{CDA}^N(p_t) = \min_{U_t, X_t} \quad \sum_{k=0}^{N-1} \ell(x_{k|t}, u_{k|t}, p_t) + F(x_{N|t}, p_t) \tag{8}$$

$$\begin{aligned} \text{s.t.} \quad & x_{0|t} = \mathcal{M}_x p_t \\ & x_{k+1|t} = f(x_{k|t}, u_{k|t}) && k \in \mathbb{I}_0^{N-1} \\ & u_{k|t} \in \mathcal{U} && k \in \mathbb{I}_0^{N-1} \\ & x_{k|t} \in \mathcal{X} && k \in \mathbb{I}_1^{N} \end{aligned}$$

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 Ω_N is a positively invariant set in which asymptotic stability can be proven, we first introduce two useful intermediate results.

Lemma 1 (Extension of [26, 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

$$
\begin{aligned}
&\mathcal{J}^N_{\mathrm{CDA}}(p_t) = \\
&\quad \sum_{k=0}^{N-1} \ell(x^\star_{k|t}, u^\star_{k|t}, p_t) + F(x^\star_{N|t}, p_t) \\
&\leq \sum_{k=0}^{N-2} \ell(x_{k|t}, u_{\mathrm{T}}(x_{k|t}, p_t), p_t) \\
&\qquad + \ell(x_{N-1|t}, u_{\mathrm{T}}(x_{N-1|t}, p_t), p_t) + F(x_{N|t}, p_t) \\
&\overset{(7)}{\leq} \sum_{k=0}^{N-2} \ell(x_{k|t}, u_{\mathrm{T}}(x_{k|t}, p_t), p_t) + F(x_{N-1|t}, p_t) \\
&= \sum_{k=0}^{N-3} \ell(x_{k|t}, u_{\mathrm{T}}(x_{k|t}, p_t), p_t) \\
&\qquad + \ell(x_{N-2|t}, u_{\mathrm{T}}(x_{N-2|t}, p_t), p_t) + F(x_{N-1|t}, p_t) \\
&\overset{(7)}{\leq} \dots \overset{(7)}{\leq} \ell(x_{0|t}, u_{\mathrm{T}}(x_{0|t}, p_t), p_t) + F(x_{1|t}, p_t) \\
&\overset{(7)}{\leq} F(x_{0|t}, p_t)
\end{aligned}
$$

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

Lemma 2 (Extension of [18, 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 $x^\star_{N|t} \notin \mathcal{X}_{\mathrm{T}}(p_t)$ and there exists a $k \in \mathbb{I}_0^{N-1}$ such that $x^\star_{k|t} \in \mathcal{X}_{\mathrm{T}}(p_t)$. Consider vector $p_{k|t}$ defined as

$$
p_{k|t} = [{x^\star_{k|t}}' \; x_{\mathrm{r}}' \; u_{\mathrm{r}}']' \tag{9}
$$

Let $\mathcal{J}^{N-k}_{\mathrm{CDA}}$ be the optimal cost for the CDA-NMPC problem $\mathbb{P}_{N-k}(p_{k|t}, \mathcal{X})$ with prediction horizon $\mathrm{N-k}$, and $\bar{U}^\star_{k|t} = (\bar{u}^\star_{k|t}, \dots, \bar{u}^\star_{N-1|t})$, $\bar{X}^\star_{k|t} = (\bar{x}^\star_{k|t}, \dots, \bar{x}^\star_{N|t})$ 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 $\bar{x}^\star_{i+1|t} = x^\star_{i+1|t}$ for $i \in \mathbb{I}_k^{N-1}$ which implies

$$
\ell(\bar{x}^\star_{i|t}, \bar{u}^\star_{i|t}, p_{k|t}) = \ell(x^\star_{i|t}, u^\star_{i|t}, p_t), \quad i \in \mathbb{I}_k^{N-1} \tag{10}
$$

$$
F(\bar{x}^\star_{N|t}, p_{k|t}) = F(x^\star_{N|t}, p_t). \tag{11}
$$

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

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

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

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

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

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 $\mathcal{X}_{\mathrm{T}}(p_t)$ and $l_{\mathrm{T}}(p_t)$ 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:*

$$
\begin{aligned}
&\Omega_N(p_t) := \\
&\quad \{x_{0|t} \in \mathcal{X} : \mathcal{J}^N_{\mathrm{CDA}}(p_t) \leq \ell(x_{0|t}, u^\star_{0|t}, p_t) + C_N(p_t)\}
\end{aligned} \tag{12}
$$

where

$$
C_N(p_t) := (N-1)d(p_t) + l_{\mathrm{T}}(p_t) \tag{13}
$$

and

$$
\begin{aligned}
d(p_t) := \inf_{x,u} \;& \ell(x, u, p_t) \\
\text{s.t.} \;& x \in \mathcal{X} \setminus \mathcal{X}_{\mathrm{T}}(p_t) \\
& u \in \mathcal{U}.
\end{aligned} \tag{14}
$$

Remark 1. 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 $\mathcal{X}_{\mathrm{T}}(p_t)$. Note further that $\Omega_N(p_t)$ does not depend on x_t, but only on x_{r}, u_{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 $x^*_{N|t} \in \mathcal{X}_{\mathrm{T}}(p_t)$. To this end, assume by contradiction that $x^\star_{N|t} \notin \mathcal{X}_{\mathrm{T}}(p_t)$. Lemma 2 proves that $x^\star_{k|t} \notin \mathcal{X}_{\mathrm{T}}(p_t)$, $\forall\, k < N$. This implies that $\ell(x^\star_{k|t}, u^\star_{k|t}, p_t) \geq d(p_t)$ and $F(x^\star_{N|t}, p_t) > l_{\mathrm{T}}(p_t)$, and thus

$$
\begin{aligned}
&\mathcal{J}^N_{\mathrm{CDA}}(p_t) = \\
&\qquad \ell(x_{0|t}, u^\star_{0|t}, p_t) + \sum_{k=1}^{N-1} \ell(x^\star_{k|t}, u^\star_{k|t}, p_t) + F(x^\star_{N|t}, p_t) \\
&\qquad > \ell(x_{0|t}, u^\star_{0|t}, p_t) + (N-1)d(p_t) + l_{\mathrm{T}}(p_t) \\
&\qquad = \ell(x_{0|t}, u^\star_{0|t}, p_t) + C_N(p_t)
\end{aligned} \tag{15}
$$

which contradicts $x_{0|t} \in \Omega_N(p_t)$ as per (12). Therefore, the terminal constraint is satisfied, i.e., $x^\star_{N|t} \in \mathcal{X}_{\mathrm{T}}(p_t)$.

We prove next that $\Omega_N(p_t)$ is a positively invariant set for the closed-loop system. To this end, let $(\bar{u}^\star_{1|t}, \dots, \bar{u}^\star_{N-1|t})$ and $(\bar{x}^\star_{2|t}, \dots, \bar{x}^\star_{N|t})$ 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 (9) and let $\mathcal{J}^{N-1}_{\rm CDA}(p_{1|t})$ be the associated optimal cost. The following inequality can be inferred:

$$\begin{aligned}
&\mathcal{J}^{N-1}_{\rm CDA}(p_{1|t}) \\
&\overset{(10)-(11)}{=} \sum_{k=1}^{N-1} \ell(x^\star_{k|t}, u^\star_{k|t}, p_{1|t}) + F(x^\star_{N|t}, p_{1|t}) \\
&\overset{(7)}{\geq} \sum_{k=1}^{N-1} \ell(x^\star_{k|t}, u^\star_{k|t}, p_{1|t}) + \ell(x^\star_{N|t}, u_{\rm T}(x^\star_{N|t}, p_{1|t}), p_{1|t}) \\
&\qquad + F(f(x^\star_{N|t}, u_{\rm T}(x^\star_{N|t}, p_{1|t}), p_{1|t}) \\
&\geq \sum_{k=1}^{N} \ell(x^\star_{k|t}, u^\star_{k|t}, p_{1|t}) + F(x^\star_{N+1|t}, p_{1|t}) \\
&= \mathcal{J}^{N}_{\rm CDA}(p_{1|t}) \qquad (16)
\end{aligned}$$

from which we also get

$$\begin{aligned}
\mathcal{J}^{N}_{\rm CDA}(p_{1|t}) &\overset{(16)}{\leq} \mathcal{J}^{N-1}_{\rm CDA}(p_{1|t}) \\
&\overset{(10)-(11)}{=} \sum_{k=1}^{N-1} \ell(x^\star_{k|t}, u^\star_{k|t}, p_{1|t}) + F(x^\star_{N|t}, p_{1|t}) \\
&\overset{(15)}{=} \mathcal{J}^{N}_{\rm CDA}(p_t) - \ell(x_{0|t}, u^\star_{0|t}, p_t) \leq C_N(p_t). \quad (17)
\end{aligned}$$

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

To prove that the optimal cost $\mathcal{J}^N_{\rm CDA}$ 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

$$\mathcal{J}^N_{\rm CDA}(p_t) \geq \ell(x_{0|t}, u^\star_{0|t}, p_t) \geq \mu_1(\|x_{0|t} - x_{\rm r}\|),$$

while from Lemma 1 we have

$$0 \leq \mathcal{J}^N_{\rm CDA}(p_t) \leq F(x_{0|t}, p_t) \leq \mu_3(\|x_{0|t} - x_{\rm r}\|)$$

$\forall\, x_{0|t} \in \mathcal{X}_{\rm T}(p_t) \subseteq \Omega_N(p_t)$, with μ_3 from Assumption 4. Finally, in order to prove the decrease condition for $\mathcal{J}^N_{\rm CDA}$, we use the inequality derived in (17) for $\mathcal{J}^N_{\rm CDA}(p_{1|t})$ and $\mathcal{J}^N_{\rm CDA}(p_t)$, and conclude that

$$\begin{aligned}
\mathcal{J}^N_{\rm CDA}(p_{1|t}) - \mathcal{J}^N_{\rm CDA}(p_t) &\leq -\ell(x_{0|t}, u^\star_{0|t}, p_t) \\
&\leq -\mu_1(\|x_{0|t} - x_{\rm r}\|).
\end{aligned}$$

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 $\mathbb{P}(p_t, \mathcal{X})$. We focus next on the data-driven version of CDA-NMPC with low complexity.

C. 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 paper, we will take the extreme case $N = 1$.

Let us exploit Bellman's principle of optimality and reformulate the CDA-NMPC problem $\mathbb{P}_N(p_t, \mathcal{X})$ as

$$\begin{aligned}
\mathcal{J}^N_{\rm CDA}(p_t) = \min_{u_{0|t}, x_{0|t}, x_{1|t}} \quad & \ell(x_{0|t}, u_{0|t}, p_t) + \mathcal{V}(x_{1|t}, p_t) \\
\text{s.t.} \quad & x_{0|t} = \mathcal{M}_x p_t \\
& x_{1|t} = f(x_{0|t}, u_{0|t}) \\
& u_{0|t} \in \mathcal{U} \\
& x_{1|t} \in \mathcal{X} \qquad (18)
\end{aligned}$$

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

$$\begin{aligned}
\mathcal{V}(x^\star_{1|t}, p_t) = \min_{U_{1|t}, X_{1|t}} \quad & \sum_{k=1}^{N-1} \ell(x_{k|t}, u_{k|t}, p_t) + F(x_{N|t}, p_t) \\
\text{s.t.} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}) \qquad k \in \mathbb{I}_1^{N-1} \\
& u_{k|t} \in \mathcal{U} \qquad k \in \mathbb{I}_1^{N-1} \\
& x_{k|t} \in \mathcal{X} \qquad k \in \mathbb{I}_2^{N} \qquad (19)
\end{aligned}$$

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

The formulation (18)–(19) 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 [27]–[29].

III. NMPC WITH LEARNED LYAPUNOV TERMINAL COST

In this section, we propose an approach to learn a Lyapunov terminal cost function $\hat{\mathcal{V}}^{\rm LLTC}(x, p)$ that approximates the cost-to-go function $\mathcal{V}(x, p)$ from data using neural networks. This will allow us to formulate problem $\mathbb{P}_N(p, \mathcal{X})$ 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.

A. Learning a Lyapunov cost-to-go function

Asymptotic stability requires that the approximate terminal cost is a Lyapunov function in $\Omega_N(p)$, i.e., $\hat{\mathcal{V}}^{\rm LLTC}(x, p)$ must be a decreasing, continuous, and positive definite function for

all $x \in \Omega_N(p)$ and all $p \in \mathcal{P}$. While many functional forms are possible, we propose to select

$$\hat{\mathcal{V}}^{\mathrm{LLTC}}(x,p) = (x - x_{\mathrm{r}})' \hat{P}(p)(x - x_{\mathrm{r}}), \tag{20}$$

which is positive definite by construction, provided that $\hat{P}(p) \succ 0$. Because $\hat{P}(p) : \mathbb{R}^{n_p} \to \mathbb{R}^{n_x \times n_x}$ 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

$$\hat{P}(p) = \hat{L}(p)\hat{L}'(p) + \epsilon I \tag{21}$$

where $\epsilon > 0$ is a small positive number and I is the identity matrix. Note that, as $\hat{L}$ is lower triangular, the parameterization (20)- (21) consists of $\frac{n_x(n_x+1)}{2}$ functions of p.

Remark 2. Directly parameterizing the symmetric part of $\hat{P}(p)$, 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 $\hat{P}(p) = \hat{L}(p)D\hat{L}'(p)$, where $\hat{L}(p)$ 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 $p^i = \begin{bmatrix} x_0^{i\prime} & x_{\mathrm{r}}^{i\prime} & u_{\mathrm{r}}^{i\prime} \end{bmatrix}'$, with $x_{\mathrm{r}}^i \in \mathcal{X}$, $u_{\mathrm{r}}^i \in \mathcal{U}$, and $x_0^i \in \Omega_N(p^i)$, for $i \in \mathbb{I}_1^M$. Then, we solve problem $\mathbb{P}_N(p^i, \mathcal{X})$ 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 $\ell_0^i := \ell(x_0^i, u_0^i, p^i)$, the cost-to-go $\mathcal{V}_1^i := \mathcal{V}(x_1^i, p^i)$ given by (19), and the value $C_N^i := C_N(p^i)$ given by (13), 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 $x_0^i \in \Omega_N(p^i)$ is.

In order to learn $\hat{\mathcal{V}}^{\mathrm{LLTC}}$ as in (20)-(21) using the collected dataset $\mathbb{D}$, we parameterize matrix $\hat{L}(p)$ using a finite-dimensional parameter θ. In the following, we will denote this parameterized matrix as $\hat{L}_\theta(p)$. In this article, we adopt a feedforward neural network (FNN) as it is a universal approximator [30], which takes the following form

$$\Phi(p;\theta,\mathcal{H}) = [\mathcal{A}_{\mathcal{H}+1} \circ \mathcal{G}_{\theta_{\mathcal{H}+1}} \circ \mathcal{A}_{\mathcal{H}} \circ \mathcal{G}_{\theta_{\mathcal{H}}} \circ \cdots \circ \mathcal{A}_1 \circ \mathcal{G}_{\theta_1}](p) \tag{22}$$

where $\Phi(p;\theta,\mathcal{H})$ yields the vectorized form of $\hat{L}_\theta(p)$. 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

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

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

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

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 [31]. In this work, we consider a Rectified Linear Unit (ReLU) activation function

$$\mathcal{A}_h(\mathcal{G}_{\theta_h}) = \mathrm{ReLU}(\mathcal{G}_{\theta_h}) := \max\{\mathcal{G}_{\theta_h}, 0\}, \quad h \in \mathbb{I}_1^{\mathcal{H}} \tag{23}$$

As schematically illustrated in Figure 1, we employ the network $\Phi(p;\theta,\mathcal{H})$ defined in (22) to parameterize the non-zero entries of matrix $\hat{L}_\theta(p)$. Hence, we rewrite (20)-(21) in the following form

$$\hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x,p) = (x - x_{\mathrm{r}})' \hat{P}_\theta(p)(x - x_{\mathrm{r}}) \tag{24}$$

$$\hat{P}_\theta(p) = \hat{L}_\theta(p)\hat{L}'_\theta(p) + \epsilon I. \tag{25}$$

Finally, we obtain the following finite-dimensional learning problem

$$\min_\theta \ \gamma\|\theta\|_2^2 + \frac{1}{M}\sum_{i=1}^M \phi(\mathcal{V}_1^i, \hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p^i)) \tag{26a}$$

$$\text{s.t. } \hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p^i) \le C_N^i \qquad i \in \mathbb{I}_1^M \tag{26b}$$

$$\hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p^i) - \hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_0^i, p^i) + \ell_0^i \le 0 \quad i \in \mathbb{I}_1^M \tag{26c}$$

where $\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a suitably defined loss function; constraint (26b) ensures the positive invariance property of the domain of attraction $\Omega_N(p^i)$; and constraint (26c) enforces the decrease of the Lyapunov terminal cost for all observed samples, and an L_2-regularization with parameter γ is introduced to prevent overfitting.

For practical purposes, as this allows us to use standard learning algorithms, we reformulate the problem above as an unconstrained problem by using ℓ_1-penalties on constraint violations, as proposed in [32]. This yields the following

Algorithm 1: Data collection

Input: M, f, $\mathcal{X}$, $\mathcal{U}$, $\mathbb{P}_N(p,\mathcal{X})$.
Output: Training data $\mathbb{D}$

Initialization: $\mathbb{D} = \emptyset$, $i = 1$;
while $i \le M$ **do**
  pick $p^i = \begin{bmatrix} x_0^{i\prime} & x_{\mathrm{r}}^{i\prime} & u_{\mathrm{r}}^{i\prime} \end{bmatrix}'$ randomly, with $x_0^i, x_{\mathrm{r}}^i \in \mathcal{X}$, $u_{\mathrm{r}}^i \in \mathcal{U}$;
  solve $\mathbb{P}_N(p^i,\mathcal{X})$ to obtain $U^i = (u_0^i, \ldots, u_{N-1}^i)$, $X^i = (x_0^i, \ldots, x_N^i)$;
  calculate $\ell_0^i := \ell(x_0^i, u_0^i, p^i)$ using (3), $\mathcal{V}_1^i := \mathcal{V}(x_1^i, p^i)$ using (19), $d(p^i)$ using (14), $C_N^i := C_N(p^i)$ using (13), $\mathcal{J}_{\mathrm{CDA}}^N(p^i) = \ell_0^i + \mathcal{V}_1^i$;
  if $\mathcal{J}_{\mathrm{CDA}}^N(p^i) \le \ell_0^i + C_N^i$ **then** // $x_0^i \in \Omega_N(p_i)$
    $\mathbb{D} = \mathbb{D} \cup \{p^i,\ x_1^i,\ \ell_0^i,\ \mathcal{V}_1^i,\ C_N^i\}$;
    $i = i + 1$;
  else
    go back to 3 ;
  end
end

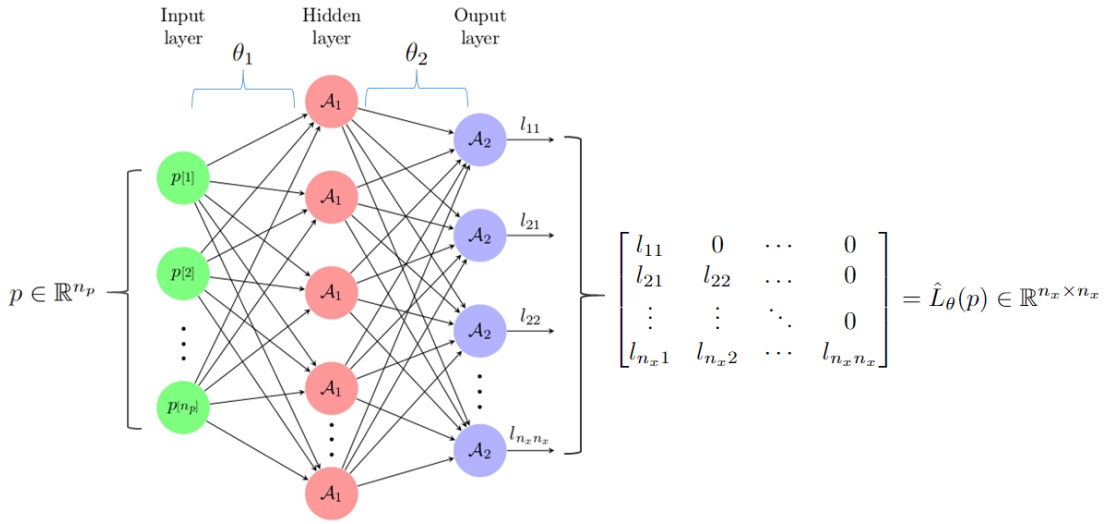

Fig. 1: A schematic illustration of FNN $\Phi(p;\theta,\mathcal{H}=1)$ to predict the matrix $\hat{L}_\theta(p)$ for a given input p.

unconstrained learning problem

$$\theta^\star := \arg\min_{\theta} \ \gamma\|\theta\|_2^2 + \frac{1}{M}\sum_{i=1}^{M} \phi(\mathcal{V}_1^i, \hat{\mathcal{V}}_\theta^{\mathrm{LLTC}}(x_1^i, p^i)) + \lambda_1 \mathcal{E}_{\mathrm{dom}}^i(\theta) + \lambda_2 \mathcal{E}_{\mathrm{dec}}^i(\theta) \quad (27)$$

where

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

and λ_1, λ_2 are sufficiently large positive penalty parameters to ensure that the optimal solution matches the one of problem (26), if the latter exists.

B. LLTC-NMPC

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

$$\begin{aligned} \mathcal{J}_{\mathrm{LLTC}}(p_t) = \min_{u_{0|t}, x_{0|t}, x_{1|t}} \quad & \ell(x_{0|t}, u_{0|t}, p_t) + \hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x_{1|t}, p_t) \\ \text{s.t.} \quad & x_{0|t} = \mathcal{M}_x p_t \\ & x_{1|t} = f(x_{0|t}, u_{0|t}) \\ & u_{0|t} \in \mathcal{U} \\ & x_{1|t} \in \mathcal{X} \end{aligned} \quad (28)$$

which we call NMPC with Learned Lyapunov Terminal Cost (LLTC-NMPC) and denote problem (28) 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 (22) to define the terminal cost and then solve the 1-step-ahead NMPC problem (28).

IV. 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, i.e., $l_{\mathrm{T}}(p)$, $\mathcal{X}_{\mathrm{T}}(p)$, and $F(x,p)$ satisfying Assumption 4 for all $p \in \mathcal{P}$ are chosen by following the procedures proposed in [25].

Algorithm 2: LLTC-NMPC scheme

Input: x_{r}, u_{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.

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

$$\mathcal{T}_a := \frac{1}{N_{\mathrm{init}}}\sum_{i=1}^{N_{\mathrm{init}}} \frac{1}{N_{\mathrm{sim}}}\sum_{t=0}^{N_{\mathrm{sim}}} \tau(p_t^i) \quad (29a)$$

$$\mathcal{T}_w := \max_{1,\ldots,N_{\mathrm{init}}} \max_{1,\ldots,N_{\mathrm{sim}}} \tau(p_t^i) \quad (29b)$$

where N_{sim} is the simulation length, N_{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 (26b)-(26c) while learning $\hat{\mathcal{V}}_{\theta^\star}^{\mathrm{LLTC}}(x,p)$, we make a comparison with the approach proposed in [22], called Learned Terminal Cost NMPC (LTC-NMPC), which also learns the terminal cost from data, but does not include (26b)-(26c).

We employ CasADi [33] via its MATLAB interface to formulate all NMPC problems and automatically generate the

corresponding nonlinear programming (NLP) problems. These are solved using Ipopt [34], with computational time $\tau(p_t^i)$ measured on an Intel Core i5-5200U (2.7GHz) 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, e.g., Acados [35].

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_{init} random initial conditions p^i are generated; for each $N \in [10, 100]$, the closed-loop cost over N_{sim} time steps is computed by solving the problem $\mathbb{P}_N(p^i, \mathcal{X}_{\text{T}}(p^i))$ for all p^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 $\hat{\mathcal{V}}_{\theta^\star}^{\text{LLTC}}(x, p)$ are generated by executing Algorithm 1. We use $M_{\text{train}} = 0.8M$ training samples for learning the FNN parameters using the quadratic loss function $\phi(z_1, z_2) = (z_1 - z_2)^2$. The remaining $M_{\text{test}} = 0.2M$ 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_{score} (coefficient of determination). Additionally, we count the number $\mathcal{C}_{\text{dom}}$ and $\mathcal{C}_{\text{dec}}$ of samples for which constraints (26b) and (26c) 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 $\Phi(p; \theta, \mathcal{H})$, we analyze the influence of the layer structure on the approximation quality of $\hat{\mathcal{V}}_{\theta^\star}^{\text{LLTC}}(x, p)$. In all considered networks, we adopt the $\text{ReLU}(\cdot)$ activation function (23). The neural networks are trained in Pytorch [36], using Adam [37] to solve problem (27) 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 [38] for a wide range of possible alternative loss functions, validation metrics, and numerical optimization algorithms.

A. CSTR problem

For an exothermic and irreversible reaction $a \longrightarrow b$ 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 [39]

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

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 $C_a^{\text{r}} = 0.5$ mol/l, $T^{\text{r}} = 350$ K, and $T_c^{\text{r}} = 300$ K is chosen under the following nominal operating conditions: $C_a^n = 1$ mol/l and $T_a^n = 350$ K, $k_0 = 7.2 \cdot 10^{10}$ min^{-1}, $E/R = 8750$ K, $\Delta H = -5 \cdot 10^4$ J/mol, $\rho = 10^3$ g/l, $C_p = 0.239$ J/g $\cdot$ K, and $UA = 5 \cdot 10^4$ J/min $\cdot$ K. A nonlinear discrete-time state-space model is obtained with sampling time $T_{\text{s}} = 0.03$ min. The state and the commanded input of the system, together with the corresponding reference signals, are defined as $x = [C_a \;\; T]'$, $x_{\text{r}} = [C_a^{\text{r}} \;\; T^{\text{r}}]'$, $u = T_c$ and $u_{\text{r}} = T_c^{\text{r}}$, 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 $C_N(p) = 2.01 \cdot 10^3$ from (13), with $l_{\text{T}}(p) = 1.77 \cdot 10^3$ and $d(p) = 4.53$. The Classic NMPC and the CDA-NMPC schemes that solve problems $\mathbb{P}_N(p, \mathcal{X}_{\text{T}}(p))$ and $\mathbb{P}_N(p, \mathcal{X})$, respectively, are designed with the ellipsoidal terminal set $\mathcal{X}_{\text{T}}(p) = \{x \in \mathcal{X} : F(x, p) \leq 1.77 \cdot 10^3\}$ where the terminal cost $F(x, p)$ 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.

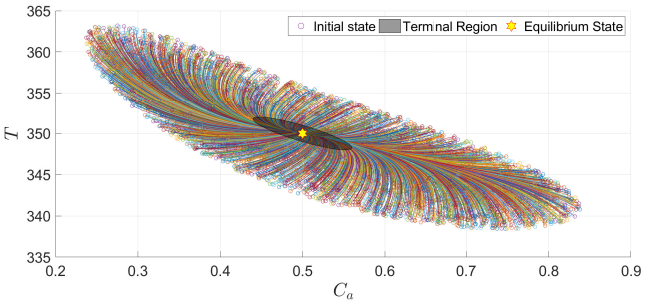

Fig. 2: State trajectory samples generated by CDA-NMPC with $N = 50$ for a given initial state x_0^i.

We learn matrix $\hat{P}_{\theta^\star}(p)$ defining $\hat{\mathcal{V}}_{\theta^\star}^{\text{LLTC}}(x, p)$ as in (24) 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 I. The table shows that the structure with 3 hidden layers, each containing 40 neurons, yields the best model to represent $\hat{\mathcal{V}}_{\theta^\star}^{\text{LLTC}}(x, p)$, since there is no stability constraint violation and small NRMSE and R^2_{score} are obtained.

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 (19).

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

TABLE I: Analysis of trained Lyapunov terminal-cost models based on different FNN structures. For each metric (NRMSE, R^2_{score}, $\mathcal{C}_{\text{dom}}$, $\mathcal{C}_{\text{dec}}$), the values obtained on train/test data are reported.

FNN model architecture	fit quality		violation of stability constraints	
	NRMSE	R^2_{score}	$\mathcal{C}_{\text{dom}}$	$\mathcal{C}_{\text{dec}}$
$\mathcal{H}=1$, $n^h=120$	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

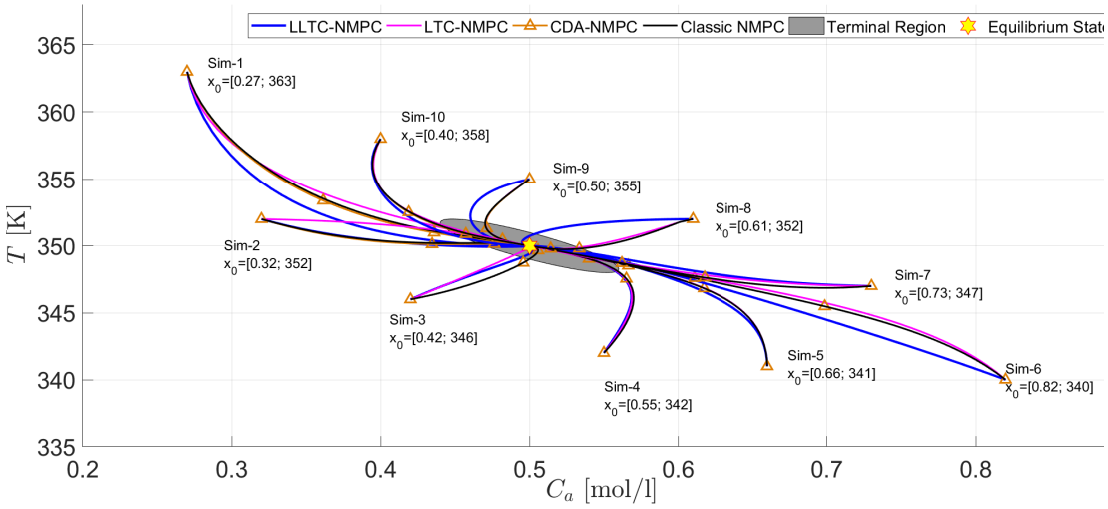

Fig. 3: State trajectories of different NMPC schemes starting from initial state $x_{0|t}$.

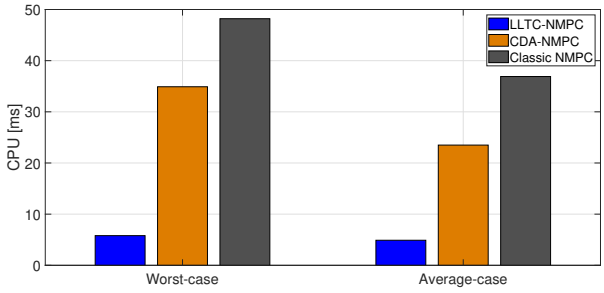

Fig. 4: Average and worst-case computational time comparison.

In order to discuss the importance of including Lyapunov conditions in the learning the cost-to-go approximation $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x,p)$, we compare our LLTC-NMPC with the LTC-NMPC approach proposed in [22], which approximates $\mathcal{V}(x,p)$ by solving problem (26) without constraints (26b) (26c). The terminal cost $\hat{\mathcal{V}}^{\text{LTC}}_{\theta^\star}(x,p)$ is modeled with the same FNN structure used for $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x,p)$. 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, $\hat{\mathcal{V}}^{\text{LTC}}_{\theta^\star}(x,p)$ of $\mathcal{V}(x,p)$ yields a more accurate cost-to-go approximation than $\hat{\mathcal{V}}^{\text{LLTC}}_{\theta^\star}(x,p)$. However, LTC-NMPC violates the stability constraints (26b) and (26c) 24 and 75 times respectively, while LLTC-NMPC never violates any constraint.

B. 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.

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

$$\begin{cases} \dot{s}_{\text{x}} &= v\cos(\psi) \\ \dot{s}_{\text{y}} &= v\sin(\psi) \\ \dot{\psi} &= \delta \end{cases} \tag{30}$$

where $(s_{\text{x}}, s_{\text{y}})$ denote the Cartesian position of the vehicle on a fixed reference frame, ψ the orientation of the vehicle with respect to the x-axis, v the longitudinal velocity, and δ the angular velocity. The system is discretized with sampling time $T_{\text{s}} = 0.1$ s. The control task consists in driving the vehicle towards the reference point $x_{\text{r}} = [0.3\ 0.5\ \pi/4]'$ and $u_{\text{r}} = [0\ 0]'$ 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 $N = 60$, $d(p) = 0.32$, and the terminal set $\mathcal{X}_{\mathrm{T}}(p) = \{x \in \mathcal{X} : F(x,p) \leq 1.31\}$, which yields $C_N(p) = 20.19$. We generate $M = 5000$ data samples and learn $\hat{\mathcal{V}}^{\mathrm{LLTC}}_{\theta^\star}(x,p)$ and $\hat{\mathcal{V}}^{\mathrm{LTC}}_{\theta^\star}(x,p)$ with fit results $\mathrm{NRMSE} = 0.11/0.13$ and $R^2_{\mathrm{score}} = 0.88/0.86$, and $\mathrm{NRMSE} = 0.06/0.09$ and $R^2_{\mathrm{score}} = 0.961/0.949$ for the $\mathrm{train}/\mathrm{test}$ datasets, respectively. The matrix $\hat{P}_{\theta^\star}(p)$ 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 $N_{\mathrm{sim}} = 80$ steps, starting from $N_{\mathrm{init}} = 15$ initial states $x_{0|t}$ picked outside the terminal region $\mathcal{X}_{\mathrm{T}}(p_t)$. 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 $\hat{\mathcal{V}}^{\mathrm{LTC}}_{\theta^*}$ is not a Lyapunov function as it does not always decrease.

The computational times of LLTC-NMPC and CDA-NMPC are reported in Table II, where it can be seen that the proposed data-driven approach requires a significantly shorter execution time than CDA-NMPC: even though one needs to also evaluate the FNN in LLTC-NMPC, the LLTC-NMPC computational time is approximately 9 times smaller than the one of CDA-NMPC.

V. Conclusions

This work presented a computationally efficient data-driven NMPC technique that employs a one-step-ahead prediction horizon in combination with a learned Lyapunov terminal cost. 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, e.g., economic costs and time-varying references.

References

[1] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Communications and Control Engineering. Springer International Publishing, 2 edition, 2017.

[2] H. Michalska and D. Q. Mayne. Receding Horizon Control of Nonlinear Systems. *Proceedings of the 28th IEEE Conference on Decision and Control,*, 1:107–108, 1989.

[3] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control : Theory, Computation and Design*. Nob Hill Publishing, 2019.

[4] H. Chen and F. Allgöwer. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica*, 34(10):1205–1217, 1998.

[5] S. Chen et al. Approximating Explicit Model Predictive Control using Constrained Neural Networks. In *American Control Conference*, pages 1520–1527, 2018.

[6] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer. Learning an Approximate Model Predictive Controller with Guarantees. *IEEE Control Systems Letters*, 2(3):543–548, 2018.

[7] S. Lucia and B. Karg. A Deep Learning-based Approach to Robust Nonlinear Model Predictive Control. *IFAC-PapersOnLine*, 51:511–516, 2018.

[8] A. Gersnoviez, M. Brox, and I. Baturone. High-Speed and Low-Cost Implementation of Explicit Model Predictive Controllers. *IEEE Transactions on Control Systems Technology*, 27(2):647–662, 2019.

[9] A. Grancharova and T. A. Johansen. *Nonlinear Model Predictive Control*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[10] R. Cagienard, P. Grieder, E.C. Kerrigan, and M. Morari. Move Blocking Strategies in Receding Horizon Control. *Journal of Process Control*, 17(6):563–570, 2007.

[11] R.C. Shekhar and C. Manzie. Optimal Move Blocking Strategies for Model Predictive Control. *Automatica*, 61:27–34, 2015.

[12] A. Makarow, C. Rösmann, and T. Bertram. Suboptimal Nonlinear Model Predictive Control with Input Move-Blocking. *International Journal of Control*, 2022.

[13] M. Michael and D. Raffaello. A Method for Reducing the Complexity of Model Predictive Control in Robotics Applications. *IEEE Robotics and Automation Letters*, 4(3):2516–2523, 2019.

[14] M. Lawrynczuk. Nonlinear Model Predictive Control for Processes with Complex Dynamics: A Parameterisation Approach using Laguerre Functions. *International Journal of Applied Mathematics and Computer Science*, 30(1):35–46, 2020.

[15] G. Pan and T. Faulwasser. NMPC in Active Subspaces: Dimensionality Reduction with Recursive Feasibility Guarantees. *Automatica*, 147:110708, 2023.

[16] T. Ohtsuka. A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon control. *Automatica*, 40:563–574, 2004.

[17] M. Diehl, H. J. Ferreau, and N. Haverbeke. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. *Lecture Notes in Control and Information Sciences*, pages 391–417, 2009.

[18] D. Limon, T. Alamo, F. Salas, and E.F. Camacho. On the Stability of Constrained MPC without Terminal Constraint. *IEEE Transactions on Automatic Control*, 51(5):832–836, 2006.

[19] P. Giesl and S. F. Hafstein. Review on Computational Methods for Lyapunov Functions. *Discrete and Continuous Dynamical Systems-series B*, 20:2291–2331, 2015.

[20] M. K. Mittal et al. Neural Lyapunov Model Predictive Control. *ArXiv*, abs/2002.10451, 2020.

[21] F. Moreno-Mora, L. Beckenbach, and S. Streif. Predictive Control with Learning-based Terminal Costs using Approximate Value Iteration. *ArXiv*, abs/2212.00361, 2022.

[22] S. Abdufattokhov, M. Zanon, and A. Bemporad. Learning Convex Terminal Costs for Complexity Reduction in MPC. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2163–2168, 2021.

[23] T. Zieger et al. Towards Safe Neural Network Supported Model Predictive Control. *IFAC-PapersOnLine*, 53:5246–5251, 2020.

[24] B. Karg and S. Lucia. Efficient Representation and Approximation of Model Predictive Control Laws via Deep Learning. *IEEE Transactions on Cybernetics*, 50(9):3866–3878, 2020.

[25] Ch. Rajhans, S. C. Patwardhan, and H. K. Pillai. Discrete Time Formulation of Quasi Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *IFAC-PapersOnLine*, 50:7181–7186, 2017.

[26] D. Q. Mayne, J. B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. *Automatica*, 36(6):789–814, 2000.

[27] D. V. Prokhorov and L. A. Feldkamp. Application of SVM to Lyapunov Function Approximation. *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, 1:383–387, 1999.

[28] Gürsel Serpen. Empirical Approximation for Lyapunov Functions with Artificial Neural Nets. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2:735–740, 2005.

[29] N. Gaby, F. Zhang, and X. Ye. Lyapunov-Net: A Deep Neural Network Architecture for Lyapunov Function Approximation. *ArXiv*, abs/2109.13359, 2021.

[30] K. Hornik, M. B. Stinchcombe, and H. L. White. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2:359–366, 1989.

[31] C. C. Aggarwal. *Neural Networks and Deep Learning: a textbook*. Springer, Cham, Switzerland, 2018.

[32] W. I. Zangwill. Nonlinear Programming via Penalty Functions. *Management Science*, 13(5):344–358, 1967.

[33] J.A.E. Andersson et al. CasADi - A Software Framework for Nonlinear Optimization and Optimal Control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[34] A. Wächter and L. T. Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106:25–57, 2006.

[35] R. Verschueren et al. Acados: A Modular Open-Source Framework for Fast Embedded Optimal Control. *Mathematical Programming Computation*, 14:147–183, 2019.

TABLE II: Computational performance of LLTC-NMPC and CDA-NMPC.

Controller	NLP dimension		Convergence	Computational time [ms]		
	# 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

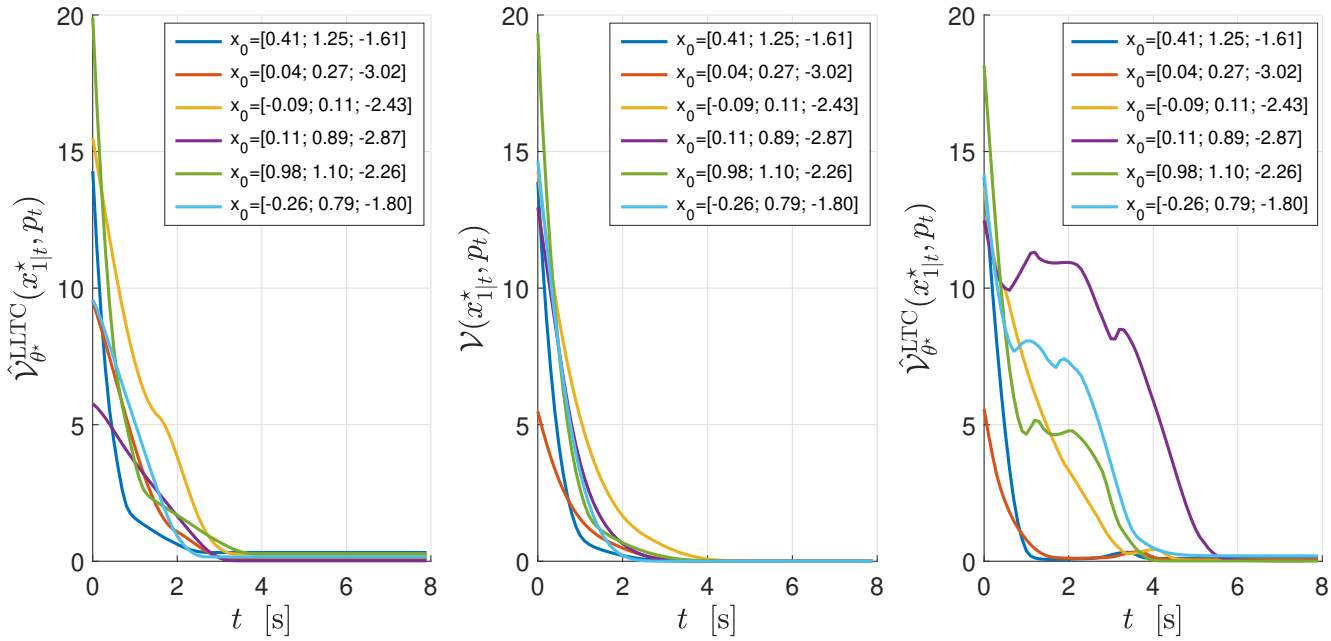

Fig. 5: Terminal cost of LLTC-NMPC, CDA-NMPC and LTC-NMPC for a set of given initial states x_0^i over the simulation horizon $N_{sim} = 80$.

[36] A. Paszke et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32:8024–8035, 2019.

[37] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ArXiv*, abs/1412.6980, 2014.

[38] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[39] P. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A Stabilizing Model-based Predictive Control Algorithm for Nonlinear Systems. *Automatica*, 37:1351–1362, 2001.