Data-Driven MPC from Non-Expert Demonstrations: Performance Guarantees and Sample Complexity

Shijie Pan*, Agustin Castellano and Enrique Mallada

***Abstract*— We study data-driven policy construction for dynamical systems using trajectories generated by surrogate finite-horizon Model Predictive Control (MPC), with the goal of approximating an infinite-horizon optimal control policy. We interpret these trajectories as non-expert demonstrations and propose a memory-based, nonparametric policy that constructs a Lipschitz-regularized upper envelope of the finite-horizon surrogate value function, which serves as an explicit upper bound on the value of the resulting policy, thereby providing a computable performance certificate while bypassing explicit optimization at runtime. We establish relative optimality guarantees with respect to the infinite-horizon optimal value function and derive sufficient MPC horizon conditions, together with sample complexity bounds, for achieving a prescribed relative error. Numerical experiments on a rocket landing task validate the theoretical predictions and demonstrate near-optimal performance.**

I. INTRODUCTION

Model Predictive Control (MPC) has become an important tool for sequential decision-making problems, with applications in robotics [1], autonomous racing [2], and rocket landing [3]. MPC operates in a receding-horizon manner, where an infinite-horizon (or long horizon) optimal control problem is approximated by repeatedly solving a finite-horizon problem online. Achieving high performance typically requires a long prediction horizon to accurately approximate the infinite-horizon objective. However, in many real-time scenarios, control inputs must be computed under strict latency constraints, often at millisecond or even sub-millisecond time scales. This significantly increases online computational burden, often exceeding the real-time capabilities of CPUs, GPUs, or embedded processors [4]. As a result, MPC faces a fundamental time–accuracy tradeoff: long horizons improve performance but are computationally prohibitive, while shorter horizons are tractable but may lead to large optimality gaps [5, Section 10.3]. This motivates the development of efficient methods for computing near-optimal control policies in real time.

A common data-driven approach is to learn a mapping $\pi(\mathbf{x})$ from the initial state $\mathbf{x}$ to the optimal control $\mathbf{u}^\star(\mathbf{x})$ via regression. Under specific structural assumptions, such as linear dynamics and quadratic stage costs, the MPC problem admits an analytical solution, leading to the framework of explicit MPC [6], where the optimal control policy is represented as a piecewise affine function of the state [7]. For more general MPC problems where such analytical solutions are unavailable, data-driven approaches have been widely explored to learn the policy mapping. Deep neural networks (DNNs) are commonly used for this purpose, a paradigm often referred to as deep MPC [8]–[10]. Beyond DNNs, other regression-based methods have also been investigated, including set-membership approximation [11], kernel regression [12], Gaussian processes [13], and decision trees [14]. Although these approaches enable fast online implementation, they generally lack a clear characterization of the sample complexity required to achieve a prescribed performance guarantee, nor provide guarantees on monotonic improvement of policy performance as more data is collected.

As such, theoretical guarantees for data-driven MPC remain limited. For example, Hertneck et al. [15] provide Hoeffding-based probabilistic guarantees on the policy accuracy gap solely applicable to sampled trajectories under stability assumptions, while Tokmak et al. [16] establish deterministic bounds on the policy gap and associated sample complexity guarantees under RKHS and geometric regularity assumptions on the policy mapping. However, these results rely on strong assumptions on the policy class.

In this work, we focus on two fundamental questions in (data-driven) MPC, related to the optimization horizon N of the receding horizon controller.

1) How does N influence the performance of the receding horizon controller?
2) If we use solutions from the receding horizon controller to build a data-driven policy, what can we say about its performance as a function of N?

Many works have addressed the first question, both in terms of performance [17], [18] and stability [19]–[21], but usually requiring strong assumptions such as a Bellman inequality [22], [23]. The second question is of great interest, and to the best of our knowledge hasn't been addressed before. In this paper, we establish lower bounds on N that enable a particular family of policies—built with data from the receding horizon controller—to achieve a desired performance. Our work builds upon the recent work of Castellano et al. [24], where the authors proposed a nonparametric data-driven policy that is greedy with respect to an upper bound of the optimal value function. Their method, however, relied on solutions to the full-length optimization problem—*expert demonstrations*—while on this work we use solutions from the receding horizon controller—i.e. *non-expert demonstrations*. In particular, we leverage data from the receding horizon controller (RHC) with horizon N to construct a similar nonparametric policy with

Shijie Pan, Agustin Castellano and Enrique Mallada are with both Department of Electrical and Computer Engineering and Data Science and AI Institute, Johns Hopkins University, Baltimore, MD, USA.
Emails: {span34, acaste11, mallada}@jhu.edu
*Corresponding author: Shijie Pan

performance guarantees. Specifically, we make the following contributions:

1) We derive lower bounds on the horizon length N of the RHC to achieve a desired level of performance.
2) We propose a nonparametric policy that uses trajectories from the RHC, and that is greedy with respect to a computable upper bound of its value function.
3) Under mild assumptions, we establish a lower bound on N ensuring this data-driven policy outperforms the RHC's upper bound.
4) We derive sample complexity results for our policy to achieve a desired performance.

The rest of the paper is organized as follows. Section II establishes preliminaries on MPC, our overall goal, and lower bounds on the horizon of the receding horizon controller to get a prescribed performance. Section III introduces our data-driven policy framework, and our two main results: conditions that make our policy *improving* with respect to the RHC's upper bound, and the sample complexity needed to achieve a desired performance. Experiments on Section IV empirically validate our framework.

II. Preliminaries

A. Infinite-Horizon optimal control problem

In this section, we introduce the system model and the optimal control formulation used throughout the paper. Consider a discrete-time invariant system,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t),\ \mathbf{x}_t \in \mathbb{X},\ \mathbf{u}_t \in \mathbb{U},\ t \geq 0. \tag{1}$$

where $\mathbb{X} \subset \mathbb{R}^{d_1}$ and $\mathbb{U} \subset \mathbb{R}^{d_2}$ are compact sets. The optimal control problem can be cast as the following optimization problem in $\mathbb{R}^{\infty}$.

$$\begin{aligned} J(\mathbf{x}_0) = \min_{\{\mathbf{u}_t,\, t\geq 0\}} \quad & \sum_{t=0}^{\infty} \gamma^t l(\mathbf{x}_t, \mathbf{u}_t) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \quad t \geq 0, \\ & \mathbf{u}_t \in \mathbb{U}, \qquad t \geq 0, \\ & \mathbf{x}_0 \in \mathbb{X} \text{ is given.} \end{aligned} \tag{2}$$

In (2), $\gamma \in (0, 1)$. Generally, infinite-horizon problems like (2) can be addressed in two ways:

- Directly finding the optimal value function J via fixed point iteration (e.g. dynamic programming or RL).
- Approximating (2) with a finite-horizon surrogate as in model predictive control (MPC).

In this work, we adopt the latter approach. The following assumptions are used throughout the paper.

Assumption 1 (Lipschitz continuity)**.** *System* (1) *is Lipschitz continuous with respect to states and controls, that is* $\forall \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{X}$ *and* $\forall \mathbf{u}, \hat{\mathbf{u}} \in \mathbb{U}$ *we have:*

$$\|f(\mathbf{x}, \mathbf{u}) - f(\hat{\mathbf{x}}, \hat{\mathbf{u}})\|_{\mathbb{X}} \leq L_f \|\mathbf{x} - \hat{\mathbf{x}}\|_{\mathbb{X}} + L_u \|\mathbf{u} - \hat{\mathbf{u}}\|_{\mathbb{U}},$$

where $\|\cdot\|_{\mathbb{X}}$ *and* $\|\cdot\|_{\mathbb{U}}$ *are norms defined in the state space* $\mathbb{X}$ *and action space* $\mathbb{U}$*, respectively.*

Assumption 2 (Non-negative cost)**.** *In* (2)*, stage costs satisfy* $l(\mathbf{x}, \mathbf{u}) \geq 0$, $\forall \mathbf{x} \in \mathbb{X}$, $\mathbf{u} \in \mathbb{U}$.

Assumption 3 (Value function decay)**.** *Optimal trajectories under* (2) *satisfy* $J(\mathbf{x}_0) \geq J(\mathbf{x}_t)$, $t \geq 0$.

Assumption 4 ($\mathbb{X}$ regularity)**.** $\mathbb{X}$ *is a compact robust positively invariant set.*

We make the following remarks regarding these assumptions.

Remark 1. *Assumption 3 ensures that the value function decays sufficiently fast along optimal trajectories, despite discounting. This is would be generally satisfied for tracking/stabilization problems where the convergence rate is faster than the discounting rate* γ.

Remark 2 (Restriction on $\mathbb{X}$)**.** *In this paper, we assume that* $\mathbb{X}$ *is a robust positively invariant set (Assumption 4) to focus on the optimality analysis within the limited page budget. By relaxing the requirement of expert demonstrations in [24, Proposition 2], this direction can be incorporated into our analysis as future work.*

B. Goal of the paper

The goal of data-driven MPC is to learn a control policy π, from recorded state data and their corresponding control inputs, over the state space $\mathbb{X}$. Given a policy $\pi : \mathbb{X} \to \mathbb{U}$, its associated value function is defined as:

$$J_\pi(\mathbf{x}) := \sum_{t=0}^{\infty} \gamma^t\, l(\mathbf{x}_t, \pi(\mathbf{x}_t)),$$

where $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \pi(\mathbf{x}_t))$ with $\mathbf{x}_0 = \mathbf{x}$. The Bellman operator $\mathcal{T}^\pi$ associated with policy π is given by

$$\mathcal{T}^\pi \hat{J}(\mathbf{x}) = l(\mathbf{x}, \pi(\mathbf{x})) + \gamma \hat{J}\big(f(\mathbf{x}, \pi(\mathbf{x}))\big),\ \ \forall \hat{J} \in \mathcal{J},$$

where $\mathcal{J} := \{\hat{J} : \mathbb{X} \to \mathbb{R} \mid \hat{J} \text{ is bounded}\}$. The operator $\mathcal{T}^\pi$ is monotone and admits a unique fixed point [25]. Further discussion is provided in the Appendix I.

Our aim in this paper is for J_π to satisfy the following relative error criterion with respect to the optimal value function $J(\mathbf{x})$:

$$\sup_{\mathbf{x} \in \mathbb{X}} \frac{J_\pi(\mathbf{x}) - J(\mathbf{x})}{J(\mathbf{x}) + \eta} \leq \mu, \tag{3}$$

where $J_\pi(\mathbf{x}) \geq J(\mathbf{x})$ always holds since $J(\mathbf{x})$ denotes the minimal value function of (4) while $J_\pi(\mathbf{x})$ is the cost induced by the policy π. Without loss of generality, we assume tolerance $\mu > 0$. The constant $\eta > 0$ is introduced to ensure the relative error is well-defined when $J(\mathbf{x}) \to 0$.

Relative error is widely adopted in industrial optimization solvers (e.g., Gurobi [26] and CPLEX [27]). It is also an important performance metric in various application domains, including power systems [28] and autonomous racing [29]. However, learning methods that explicitly optimize or provide guarantees in terms of relative error remain largely underexplored.

C. Non-expert demonstrations

To solve Problem (2), it is customary to transform or approximate it by a finite horizon problem. We distinguish between two settings:

- **Expert setting [30, Assumption 2.5]:** the terminal cost is given by the exact value function $J(\mathbf{x}_N)$:

$$\begin{aligned} J(\mathbf{x}_0) = \min_{\{\mathbf{u}_{0:N-1}\}} \quad & \sum_{t=0}^{N-1} \gamma^t l(\mathbf{x}_t, \mathbf{u}_t) \; + \; \gamma^N J(\mathbf{x}_N) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t),\, t = 0, \dots, N-1, \\ & \mathbf{u}_t \in \mathbb{U},\, t = 0, \dots, N-1, \\ & \mathbf{x}_0 \in \mathbb{X} \text{ is given.} \end{aligned} \tag{4}$$

- **Non-expert setting:** the terminal cost $J(\mathbf{x}_N)$ is approximated by a surrogate function $F(\mathbf{x}_N)$:

$$\begin{aligned} J_N(\mathbf{x}_0) = \min_{\{u_{0:N-1}\}} \quad & \sum_{t=0}^{N-1} \gamma^t l(\mathbf{x}_t, \mathbf{u}_t) \; + \; \gamma^N F(\mathbf{x}_N) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t),\, t = 0, \dots, N-1, \\ & \mathbf{u}_t \in \mathbb{U},\, t = 0, \dots, N-1, \\ & \mathbf{x}_0 \in \mathbb{X} \text{ is given.} \end{aligned} \tag{5}$$

The expert setting requires solving the infinite-horizon problem to evaluate $J(\mathbf{x}_N)$, which is generally impractical for data collection and control computation for a data-driven policy. Therefore, we focus on the non-expert setting. The challenge in this case is to find a suitable surrogate $F(\cdot)$.

Much attention has been given in the literature to the design of a terminal cost function $F(\cdot)$, particularly when stability is desired. For constrained LQR, it is customary to use a quadratic terminal cost $F(\mathbf{x}_N) = \mathbf{x}_N^\top P \mathbf{x}_N$ where P is the solution to the algebraic Ricatti equation [31]; this approach has been used for nonlinear systems too [32]. The key idea, in general, is to design an $F(\cdot)$ that satisfies a one-step Bellman inequality [33].

In (5), we design a non-negative terminal cost $F(\mathbf{x}_N) \geq 0$ that is always a lower approximate of $J(\mathbf{x}_N)$ such that $F(\mathbf{x}_N) \leq J(\mathbf{x}_N)$, $\forall \mathbf{x}_N \in \mathbb{X}$. For any initial state $\mathbf{x}_0 \in \mathbb{X}$, let $\mathbf{u}_0^F(\mathbf{x}_0)$ denote the first control input obtained by solving (5), $\mathbf{x}_1^F(\mathbf{x}_0)$ the corresponding predicted next state, and $\mathbf{x}_N^F(\mathbf{x}_0)$ the associated terminal state. Using this notation, we further make the following implicit assumptions on the terminal cost $F(\cdot)$.

Assumption 5 (Stage-cost bounds)**.** *Consider the approximate problem* (5)*. For all $\mathbf{x}_0 \in \mathbb{X}$, there exists a constant $v > 0$ such that*

$$l(\mathbf{x}_0, \mathbf{u}_0^F(\mathbf{x}_0)) \geq v J_N(\mathbf{x}_0). \tag{6}$$

Moreover, there exists a constant $C > 0$ such that the terminal state $\mathbf{x}_N^F(\mathbf{x}_0)$ under (5) *satisfies*

$$J(\mathbf{x}_N^F(\mathbf{x}_0)) \leq C\, J(\mathbf{x}_0). \tag{7}$$

Assumption 5 ensures that the stage cost at the initial state $l(\mathbf{x}_0, \mathbf{u}_0^F(\mathbf{x}_0))$ is sufficiently large relative to the non-expert value function $J_N(\mathbf{x}_0)$, while the true tail cost $J(\mathbf{x}_N^F(\mathbf{x}_0))$ in (5) is bounded in terms of the initial expert value function $J(\mathbf{x}_0)$. In the rocket landing experiment in Section IV (nonlinear MPC with quadratic cost), we empirically show that the terminal cost can be chosen as $F \equiv 0$ for quadratic cost MPCs. An additional discussion v is shown in Appendix II.

The following lemma provides a simple but important observation. Since the function $F(\cdot)$ is designed as a lower approximation of the unknown tail cost $J(\cdot)$, replacing $J(\mathbf{x}_N)$ with $F(\mathbf{x}_N)$ in the truncated MPC problem results in a smaller (optimization) cost.

Lemma 1 (Value lower approximation)**.** $J_N(\mathbf{x}) \leq J(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{X}$.

Lemma 1 shows that the non-expert optimization cost J_N always underestimates the optimal control cost J in (4). The cost J_N, however, should not be confused with the value $J_{\pi_{\text{MPC}}}$ associated with applying the MPC policy π_{MPC} via (5). Nevertheless, the closer J_N is to J, e.g., as N grows, the better is the quality of the resulting control $\mathbf{u}_0^F$. This property will be used in the next section.

The next result characterizes how large the MPC horizon N should be for the truncated problem to provide a good relative approximation.

Proposition 1 (Multiplicative suboptimality-based sufficient MPC horizon)**.** *Under Assumptions 2-5, consider the non-expert problem* (5)*, then*

$$(1 - \gamma^N(1 + C))J(\mathbf{x}) \leq J_N(\mathbf{x}),\, \forall \mathbf{x} \in \mathbb{X}. \tag{8}$$

Moreover, if $N \geq \frac{\log(1+C)}{\log\left(\frac{1}{\gamma}\right)}$, the following δ-Bellman inequality holds:

$$\delta\, l(\mathbf{x}_0, \mathbf{u}_0^F(\mathbf{x}_0)) \leq J_N(\mathbf{x}_0) - \gamma J_N\big(f(\mathbf{x}_0, \mathbf{u}_0^F(\mathbf{x}_0))\big),\, \forall \mathbf{x}_0 \in \mathbb{X}, \tag{9}$$

where,

$$\delta \leq 1 - \frac{C\gamma^N}{v(1 - \gamma^N(1 + C))}. \tag{10}$$

The proof of Proposition 1 is shown in the Appendix III.

III. Data-driven Policy

In this section we present our data-driven nonparametric policy, and provide explicit theoretical guarantees on the requirement of data points to achieve μ-relative optimality (3).

A. Non-expert nonparametric policy

Following the approach in [24, Section 4], we construct a nonparametric policy $\pi_{\mathcal{D}}$ by sampling initial states from the state space $\mathbb{X}$ and repeatedly solving (5) to generate trajectories. In contrast to [24], which assumes access to expert trajectories (4), our approach relies solely on non-expert trajectories generated by finite-horizon MPC (5), and thus does not require expert information.

Let k denote the trajectory index. For each sampled initial state, we generate a non-expert trajectory $\tau_k := \{\tilde{\mathbf{x}}_{t,k}, \tilde{\mathbf{u}}_{t,k}\}$

by repeatedly solving (5). The resulting tuples are stored in the dataset

$$\mathcal{D} \triangleq \bigcup_k \{(\tilde{\mathbf{x}}_{t,k}, \tilde{\mathbf{u}}_{t,k}, \tilde{\mathbf{J}}_{t,k})\}_{t\geq 0}, \tag{11}$$

where $\tilde{\mathbf{u}}_{t,k} = \mathbf{u}_0^F(\tilde{\mathbf{x}}_{t,k})$ and $\tilde{\mathbf{J}}_{t,k} = J_N(\tilde{\mathbf{x}}_{t,k})$, with $\mathbf{u}_0^F(\cdot)$ denoting the first optimal control input of (5).

Definition 1 (Non-expert nonparametric policy [24])**.** *Given a dataset $\mathcal{D}$ as in* (11) *and a parameter $\lambda > 0$, define,*

$$\pi_{\mathcal{D}} : \mathbb{X} \to \mathbb{U},\ \pi_{\mathcal{D}}(\mathbf{x}) = \tilde{\mathbf{u}}_\ell(\mathbf{x}),$$

where $\ell(\mathbf{x}) = \arg\min_{1\leq i\leq |\mathcal{D}|} \left\{\tilde{\mathbf{J}}_i + \lambda\|\mathbf{x} - \tilde{\mathbf{x}}_i\|_{\mathbb{X}}\right\}$. If multiple minimizers exist, one of them is selected arbitrarily.

A suitable choice of λ will be provided later in Theorem 2. The proposed nonparametric policy $\pi_{\mathcal{D}}$ can be interpreted as a value-aware nearest neighbor based on the regularized score $\tilde{\mathbf{J}}_i + \lambda\|\mathbf{x} - \tilde{\mathbf{x}}_i\|_{\mathbb{X}}$, where the first control associated with the selected sample is applied to the system. The sampling mechanism used to collect this data will be described later on in Algorithm 1.

B. Policy evaluation bound

We now derive an upper bound for $J_{\pi_{\mathcal{D}}}(\mathbf{x})$ based on sampled non-expert values $\tilde{\mathbf{J}}_i$ contained in the dataset $\mathcal{D}$. First, we introduce the following assumption on the Lipschitz continuity of $J_N(\mathbf{x})$ and the stage cost $l(\mathbf{x}, \mathbf{u})$ over state space $\mathbb{X}$.

Assumption 6 (Lipschitz Continuity)**.** *For system* (1)*, J_N is Lipschitz continuous, i.e.,*

$$|J_N(\mathbf{x}) - J_N(\hat{\mathbf{x}})| \leq L_J\|\mathbf{x} - \hat{\mathbf{x}}\|_{\mathbb{X}},\ \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{X}.$$

Similarly, the stage cost $l(\mathbf{x}, \mathbf{u})$ is Lipschitz continuous in $\mathbf{x}$, uniformly over $\mathbf{u} \in \mathbb{U}$, i.e.,

$$\sup_{\mathbf{u}\in\mathbb{U}} |l(\mathbf{x}, \mathbf{u}) - l(\hat{\mathbf{x}}, \mathbf{u})| = L_l\|\mathbf{x} - \hat{\mathbf{x}}\|_{\mathbb{X}},\ \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{X},$$

and we assume there exists $\kappa > 0$ such that $L_l \leq \kappa L_J$.

From Assumption 6, for any $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$, we know,

$$J_N(\mathbf{x}) \leq J_N(\mathbf{x}') + L_J\|\mathbf{x} - \mathbf{x}'\|_{\mathbb{X}}\ .$$

We then establish a global upper bound for $J_{\pi_{\mathcal{D}}}(\cdot)$, based on the data in $\mathcal{D}$.

Definition 2. *For given $\lambda > 0$ and $\delta \in (0, 1)$, define:*

$$J_{\text{ub}}^\lambda : \mathbb{X} \to \mathbb{R} : J_{\text{ub}}^\lambda(\mathbf{x}) \triangleq \delta^{-1} \min_{1\leq i\leq |\mathcal{D}|} \left\{\tilde{\mathbf{J}}_i + \lambda\|\mathbf{x} - \tilde{\mathbf{x}}_i\|_{\mathbb{X}}\right\},$$

where δ is the coefficient of Bellman inequality described in Proposition 1.

Assumption 7 (Dataset consistency)**.** *For any $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i, \tilde{\mathbf{J}}_i) \in \mathcal{D}$, there exists $(\tilde{\mathbf{x}}_j, \tilde{\mathbf{u}}_j, \tilde{\mathbf{J}}_j) \in \mathcal{D}$ such that $\tilde{\mathbf{x}}_j = f(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i)$.*

A preparatory lemma on functional monotonicity is presented below to facilitate the proof of Theorem 1.

Lemma 2 (Policy value bound [25])**.** *If $\hat{J} : \mathbb{X} \to \mathbb{R}$ satisfies $(T^\pi \hat{J})(\mathbf{x}) \leq \hat{J}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{X}$, then $J_\pi(\mathbf{x}) \leq \hat{J}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{X}$.*

Our main result shows that, with appropriate choices of λ and the MPC horizon N, $J_{\pi_{\mathcal{D}}}(\mathbf{x}) \leq J_{\text{ub}}^\lambda(\mathbf{x})$.

Theorem 1 (Policy evaluation inequality)**.** *Under Assumptions 1-7, suppose that $\gamma L_f < 1$. Given $1 > \delta > 0$, if N is greater or equal to the following:*

$$\max\left\{\frac{\log\left(-\frac{1}{4(1+C)} + \frac{1}{2}\sqrt{\frac{1}{4(1+C)^2} + \frac{2v(1-\delta)}{C(1+C)}}\right)}{\log(\gamma)}, \frac{\log(2(1+C))}{\log\left(\frac{1}{\gamma}\right)}\right\}, \tag{12}$$

where both C and v are constants from Assumption 5, then (9) *holds. Furthermore, for any λ satisfying $\lambda \geq \frac{\kappa}{\delta^{-1}(1-\gamma L_f)} L_J$, we have $J_{\pi_{\mathcal{D}}}(\mathbf{x}) \leq J_{\text{ub}}^\lambda(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{X}$.*

Proof. **Step 1. Sufficient MPC horizon to ensure δ-Bellman inequality of $J_N(\mathbf{x})$:** To ensure that the δ-Bellman inequality of $J_N(\mathbf{x})$ in (9) holds, we seek the minimum horizon length N such that (13) is satisfied, where (13) is obtained by rearranging (10).

$$\frac{C\gamma^N}{v(1 - \gamma^N(1 + C))} \leq 1 - \delta. \tag{13}$$

We next derive a sufficient condition on the horizon length N to satisfy (13). Since the denominator $1 - \gamma^N(1 + C)$ is positive when $\gamma^N(1 + C) < 1$, we restrict to the regime where this condition holds. To further simplify the bound, we upper bound the LHS in (13) via,

$$\begin{aligned}\frac{C\gamma^N}{1 - \gamma^N(1 + C)} &= C\gamma^N\left(1 + \frac{\gamma^N(1 + C)}{1 - \gamma^N(1 + C)}\right)\\ &\leq C\big(\gamma^N + 2(1 + C)\gamma^{2N}\big),\end{aligned}$$

provided that $\gamma^N(1 + C) \leq \frac{1}{2}$. Thus, a sufficient condition for (13) is to solve the following quadratic inequality:

$$\gamma^N + 2(1 + C)\gamma^{2N} \leq \frac{v(1 - \delta)}{C} \tag{14}$$

$$\gamma^N \leq -\frac{1}{4(1 + C)} + \frac{1}{2}\sqrt{\frac{1}{4(1 + C)^2} + \frac{2v(1 - \delta)}{C(1 + C)}}. \tag{15}$$

Because $\frac{v(1-\delta)}{C} > 0$, then the corresponding quadratic equation (14) always has 2 roots. Taking logarithms on both sides (recalling that $\gamma \in (0, 1)$) and combining with $\gamma^N(1 + C) \leq \frac{1}{2}$ yields the horizon bound (12). It is worth noting that for certain small values of δ, the right-hand side of (15) may exceed 1. In this case, the constraint $\gamma^N(1+C) \leq \frac{1}{2}$ becomes effective.

Step 2. Bellman inequality of J_{ub}^λ: In the 2nd part of the proof, we try to show the $J_{\text{ub}}^\lambda(\mathbf{x})$ satisfies the Bellman inequality described in Lemma 2, thus $J_{\pi_{\mathcal{D}}} \leq J_{\text{ub}}^\lambda$. From the Definition 2, we assign the upper bound as $J_{\text{ub}}^\lambda(\mathbf{x}_0) = \min_{i\in[|\mathcal{D}|]} \delta^{-1}(\tilde{\mathbf{J}}_i + \lambda\|\mathbf{x}_0 - \tilde{\mathbf{x}}_i\|_{\mathbb{X}})$. With a slight abuse of notation, we assume $i \in [|\mathcal{D}|]$ minimizes the right hand side of $J_{\text{ub}}^\lambda(\mathbf{x}_0)$ and denote $\tilde{\mathbf{x}}_j = f(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i)$. Then, $\forall \mathbf{x}_0 \in \mathbb{X}$,

$$\mathcal{T}^{\pi_{\mathcal{D}}} J_{\text{ub}}^\lambda(\mathbf{x}_0) - J_{\text{ub}}^\lambda(\mathbf{x}_0) = l(\mathbf{x}_0, \tilde{\mathbf{u}}_i) + \gamma J_{\text{ub}}^\lambda(\mathbf{x}_0') - J_{\text{ub}}^\lambda(\mathbf{x}_0)$$

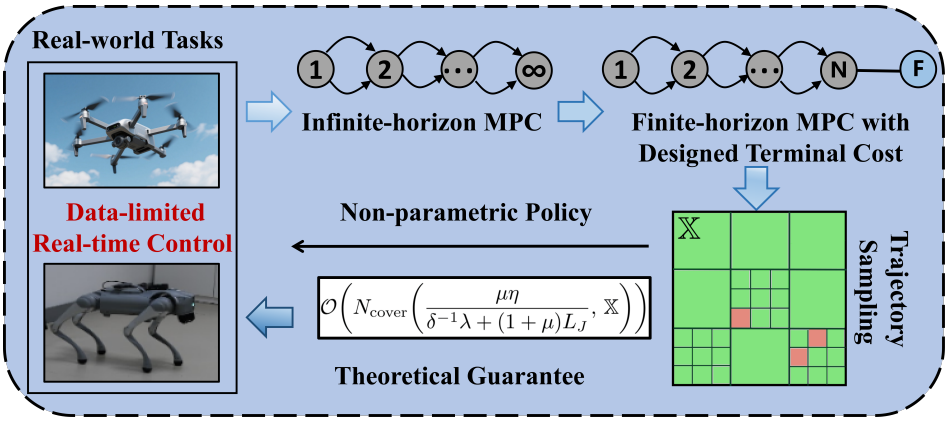
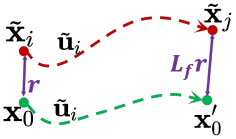


Fig. 1. Non-expert data-driven MPC flowchart. An infinite horizon MPC problem is approximated via a finite horizon problem with a terminal cost function $F(\cdot)$. Trajectory solutions from this approximate problem are used to define our data-driven policy, which enjoys—with sufficient coverage—guarantees on performance.

$$
\begin{aligned}
&\overset{(viii)}{\leq} l(\tilde{\mathbf{x}}_i,\tilde{\mathbf{u}}_i)+|l(\tilde{\mathbf{x}}_i,\tilde{\mathbf{u}}_i)-l(\mathbf{x}_0,\tilde{\mathbf{u}}_i)|+\gamma J^{\lambda}_{\mathrm{ub}}(\mathbf{x}'_0)-J^{\lambda}_{\mathrm{ub}}(\mathbf{x}_0)\\
&\overset{(ix)}{\leq} \delta^{-1}(J_N(\tilde{\mathbf{x}}_i)-\gamma J_N(\tilde{\mathbf{x}}_j))+\gamma J^{\lambda}_{\mathrm{ub}}(\mathbf{x}'_0)-J^{\lambda}_{\mathrm{ub}}(\mathbf{x}_0)\\
&\quad+|l(\tilde{\mathbf{x}}_i,\tilde{\mathbf{u}}_i)-l(\mathbf{x}_0,\tilde{\mathbf{u}}_i)|\\
&\overset{(x)}{\leq} \delta^{-1}(J_N(\tilde{\mathbf{x}}_i)-\gamma J_N(\tilde{\mathbf{x}}_j))+\gamma J^{\lambda}_{\mathrm{ub}}(\mathbf{x}'_0)-J^{\lambda}_{\mathrm{ub}}(\mathbf{x}_0)\\
&\quad+\kappa L_J\|\tilde{\mathbf{x}}_i-\mathbf{x}_0\|_{\mathbb{X}}\\
&\overset{(xi)}{\leq} \delta^{-1}\left[\gamma\lambda\|\mathbf{x}'_0-\tilde{\mathbf{x}}_j\|_{\mathbb{X}}-\lambda\|\mathbf{x}_0-\tilde{\mathbf{x}}_i\|_{\mathbb{X}}\right]+\kappa L_J\|\tilde{\mathbf{x}}_i-\mathbf{x}_0\|_{\mathbb{X}}\\
&\overset{(xii)}{\leq} (\delta^{-1}(\gamma\lambda L_f-\lambda)+\kappa L_J)\|\mathbf{x}_0-\tilde{\mathbf{x}}_i\|_{\mathbb{X}}\leq 0,
\end{aligned}
$$

where $\mathbf{x}'_0=f(\mathbf{x}_0,\tilde{\mathbf{u}}_i)$ is the next state obtained by applying $\tilde{\mathbf{u}}_i$ on $\mathbf{x}_0$, $\tilde{\mathbf{u}}_i$ is the first optimal control by solving (5) with initial condition $\tilde{\mathbf{x}}_i$, and $i\in\arg\min_{k\in[|\mathcal{D}|]}\delta^{-1}(\tilde{\mathbf{J}}_k+\lambda\|\mathbf{x}_0-\tilde{\mathbf{x}}_k\|_{\mathbb{X}})$. The goal of step $(viii)$ is to replace the first term $l(\mathbf{x}_0,\tilde{\mathbf{u}}_i)$ by $l(\tilde{\mathbf{x}}_i,\tilde{\mathbf{u}}_i)$. Then, the δ-Bellman inequality (9) is applied to obtain (ix). Besides, step (x) is using the Lipschitz condition (Assumptions 1 and 6) to bound the comparison term on l. Finally, (xi) is obtained from the following expansion:

$$
\begin{aligned}
&J_N(\tilde{\mathbf{x}}_i)=\tilde{\mathbf{J}}_i,\quad J_N(\tilde{\mathbf{x}}_j)=\tilde{\mathbf{J}}_j,\\
&J^{\lambda}_{\mathrm{ub}}(\mathbf{x}'_0)\leq\delta^{-1}(\tilde{\mathbf{J}}_j+\lambda\|\mathbf{x}'_0-\tilde{\mathbf{x}}_j\|_{\mathbb{X}}),\\
&J^{\lambda}_{\mathrm{ub}}(\mathbf{x}_0)=\delta^{-1}(\tilde{\mathbf{J}}_i+\lambda\|\mathbf{x}_0-\tilde{\mathbf{x}}_i\|_{\mathbb{X}}),
\end{aligned}
$$



where $\tilde{\mathbf{J}}_j=J_N(\tilde{\mathbf{x}}_j)$ since (11). Because of Assumption 7, $\tilde{\mathbf{x}}_j=f(\tilde{\mathbf{x}}_i,\tilde{\mathbf{u}}_i)$ is definitely in the data set $\mathcal{D}$. Substitute it into (x) to get (xi). Then, by further applying Assumption 1 (see Figure above, take $r=\|\mathbf{x}_0-\tilde{\mathbf{x}}_i\|_{\mathbb{X}}$), we have,

$$
\delta^{-1}(\tilde{\mathbf{J}}_j+\lambda\|\mathbf{x}'_0-\tilde{\mathbf{x}}_j\|_{\mathbb{X}})\leq\delta^{-1}(\tilde{\mathbf{J}}_j+\lambda L_f\|\mathbf{x}_0-\tilde{\mathbf{x}}_i\|_{\mathbb{X}}).
$$

By substituting it into (xi), we obtain (xii).

Now, in order to apply Lemma 2, we need,

$$
\delta^{-1}(\gamma\lambda L_f-\lambda)+\kappa L_J\leq 0\rightarrow\lambda\geq\frac{\kappa}{\delta^{-1}(1-\gamma L_f)}L_J,
$$

which also encodes the additional requirement $\gamma L_f<1$ to keep the denominator positive. Therefore, since $J_{\pi_{\mathcal{D}}}$ is the fixed point of $\mathcal{T}^{\pi_{\mathcal{D}}}$, we have $\mathcal{T}^{\pi_{\mathcal{D}}}J_{\pi_{\mathcal{D}}}=J_{\pi_{\mathcal{D}}}$. Moreover, $\mathcal{T}^{\pi_{\mathcal{D}}}J^{\lambda}_{\mathrm{ub}}\leq J^{\lambda}_{\mathrm{ub}}$. By monotonicity of the Bellman operator (Lemma 2), it follows that $J_{\pi_{\mathcal{D}}}\leq J^{\lambda}_{\mathrm{ub}}$. This completes the proof. □

Theorem 1 is related to Theorem 1 in [30] and Theorem 1 in [24]. The key difference is that we consider the non-expert setting (5), which avoids explicit use of the Bellman equation. To compensate, we require a sufficiently large MPC horizon N so that the δ-Bellman inequality (9) holds, enabling policy evaluation via data-driven upper bounds. The condition $\gamma L_f<1$ is a standard contraction requirement in the analysis of error propagation for discounted dynamic programming and Lipschitz systems; see, e.g., [34]–[36].

C. Adaptive Sampling with Performance Guarantees

This section presents an active sampling mechanism for nonparametric data-driven policies (Definition 1) along with the theoretical guarantees. We start by providing conditions coverage conditions on the data the ensure at relative optimality gap of at most μ. This will guide the design of our sampling algorithm.

Theorem 2 (Performance guarantees)**.** *Let $\mu>0$, $\delta>\frac{1}{1+\mu}$, and $\eta>0$. Suppose that the policy $\pi_{\mathcal{D}}$ satisfies the conditions of Theorem 1, and that the dataset $\mathcal{D}$ satisfies the following coverage condition: for every $\mathbf{x}\in\mathbb{X}$, there exists $\tilde{\mathbf{x}}_i\in\mathcal{D}$ such that*

$$
\|\mathbf{x}-\tilde{\mathbf{x}}_i\|_{\mathbb{X}}\leq\frac{(1-\delta^{-1})\tilde{\mathbf{J}}_i+\mu(\tilde{\mathbf{J}}_i+\eta)}{\delta^{-1}\lambda+(1+\mu)L_J}. \tag{16}
$$

Then the relative performance guarantee (3) *holds. Moreover, the sample complexity is*

$$
\mathcal{O}\left(N_{\mathrm{cover}}\left(\frac{\mu\eta}{\delta^{-1}\lambda+(1+\mu)L_J},\mathbb{X}\right)\right),
$$

where $N_{\mathrm{cover}}(r,\mathbb{X})$ denotes the covering number of $\mathbb{X}$ with radius r.

The proof is given in Appendix IV. Theorem 2 shows that the required coverage radius determines the achievable relative error: a covering of $\mathbb{X}$ at this resolution ensures (3), and the corresponding sample complexity follows from standard covering arguments. The sample complexity is defined as the

number of cells verified in Algorithm 1, which is equal to the number of memorized trajectories. Moreover, increasing δ (e.g., by increasing the MPC horizon used to generate the data) enlarges the admissible covering radius in (16), as illustrated in Table 1.

Algorithm 1 Relative Error MPC Adaptive Sampling (REMAS)

Initialize: $\mathcal{D} = \emptyset$; Unverified $= \mathbb{X}$; Verified $= \emptyset$.
repeat
  for $\mathbb{X}_k \in$ Unverified **do**
    Get trajectory $\phi_k = \{(\tilde{\mathbf{x}}_{t,k}, \tilde{\mathbf{u}}_{t,k}, \tilde{\mathbf{J}}_{t,k})\}_{t=0}$.
    Trajectory from solving (5) from $\tilde{\mathbf{x}}_{0,k}$, center of $\mathbb{X}_k$.
    if $(\tilde{\mathbf{x}}_{0,k}, \tilde{\mathbf{J}}_{0,k})$ perform over $\mathbb{X}_k$ ((16), Thm 2) **then**
      Verified.add($\mathbb{X}_k$); Unverified.remove($\mathbb{X}_k$).
    else
      Split $\mathbb{X}_k$ into 3^n cells; add children to Unverified
    end if
  end for
until Unverified $= \emptyset$

To obtain a dataset $\mathcal{D}$ that satisfies Theorem 2, Algorithm 1 adaptively partitions $\mathbb{X}$ to collect data for the nonparametric policy in Definition 1. Each cell is represented by its center $\mathbf{x}_i$, where a non-expert trajectory is generated by (11) and verified using Theorem 2. Cells that fail verification are recursively refined, while verified cells are retained. The resulting dataset ensures the policy satisfies (3).

IV. Rocket Landing Experiments

We consider a 2D planar rocket landing task modeled as a 6D system. The state is given by $\mathbf{x} = [p_x,\, p_z,\, v_x,\, v_z,\, \theta,\, \omega]^\top$, where p_x, p_z denote the horizontal and vertical positions, v_x, v_z the corresponding velocities, θ the vehicle attitude, and ω the angular velocity. The control inputs are $\mathbf{u} = [\tau_1, \tau_2]^\top$, where τ_1 is the thrust magnitude and τ_2 is the attitude torque. The control inputs are constrained as $0 \le \tau_1 \le 20$, $|\tau_2| \le 0.2$. The continuous–time dynamics are,

$$\dot{p}_x = 0.3v_x,\ \dot{p}_z = 0.3v_z,\ \dot{v}_x = \frac{\tau_1}{3m}\sin\theta,\ \dot{v}_z = \frac{\tau_1}{3m}\cos\theta - \frac{g}{3},$$
$$\dot{\theta} = \omega,\ \dot{\omega} = \tau_2/I,$$

where $m = 1$ denotes the mass, $I = 0.1$ the moment of inertia, and $g = 9.8$ the gravitational acceleration. The control objective is to land the rocket at the origin with zero velocity and upright attitude. The stage cost is,

$$l(\mathbf{x}, \mathbf{u}) = \mathbf{x}^\top S\mathbf{x} + \mathbf{u}^\top R\mathbf{u},\ \ \mathbf{x} \in \mathbb{X};\quad l = +\infty \text{ otherwise},$$

where $S, R \succ 0$, the discount factor is set to $\gamma = 0.8$, and the terminal cost is chosen as $F = 0$. This configuration empirically yields $C = 2.056$ and $v = 0.232$ (Assumption 5). State constraints are imposed to represent feasible flight conditions. In particular, the state space $\mathbb{X}$ is defined as

$$\mathbb{X} = \begin{Bmatrix} |p_x| \le 1, & 0 \le p_z \le 2, & |v_x| \le 1, \\ |v_z| \le 1, & |\theta| \le 0.35, & |\omega| \le 1 \end{Bmatrix}.$$

We consider four different values of N to study the relative error performance trend. Table I reports the values of the δ-Bellman inequality (computed from (10), Proposition 1) and the number of sampled trajectories (sample complexity) required by Algorithm 1 for different choices of the horizon N, with $\eta = 3$ and $\mu = 1.2$. As indicated by Theorem 2, increasing the MPC horizon N drives δ closer to 1, which in turn reduces the number of trajectories required by Algorithm 1 to satisfy (16).

TABLE I
δ AND TRAJECTORY NUMBER ON THE MPC HORIZON N.

N	15	18	20	27
δ	0.6504	0.8309	0.894	0.9784
Number of trajectories	531441	413505	354537	354537

We apply Algorithm 1 across different horizon lengths N on 50 rocket landing tasks that uniformly sampled from $\mathbb{X}$.

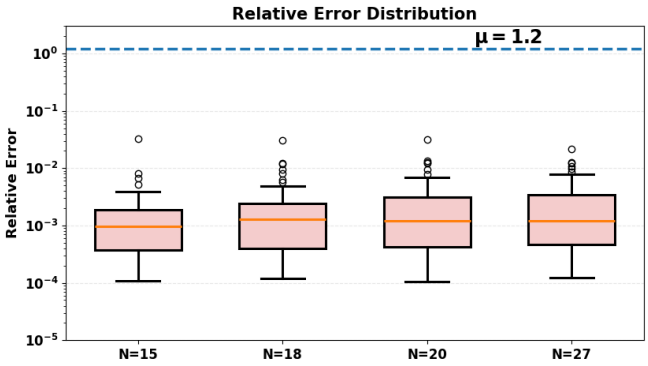


Fig. 2. Distribution of the relative error (orange boxplots) in (3) for different choices of N in 50 rocket landing tasks. Each distribution is much lower than the theoretical guarantee (blue dash line) with $\mu = 1.2$.

The relative error performance is shown in Fig. 2. The empirical relative error is lower than the theoretical guarantee claimed in (3) to a great extent. Although the mean and variance remains similar, the number of trajectories is significantly reduced as the MPC horizon N increases.

V. Conclusion

In this paper, we proposed a data-driven nonparametric policy constructed from finite-horizon MPC trajectories without requiring expert demonstrations. We established μ-relative performance guarantees with respect to the original infinite-horizon optimal value function and characterized sufficient conditions on the MPC horizon and data coverage to achieve this. Future work includes extensions to stochastic systems and improving scalability in high-dimensional settings.

References

[1] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[2] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control," in *2017 American control conference (ACC)*, pp. 5115–5120, IEEE, 2017.

[3] J. Jang, C. H. Lee, and S. He, "Convex programming approach of robust powered descent guidance through dynamic tube mpc," in *34th Congress of the International Council of the Aeronautical Sciences, ICAS*, pp. 1–12, 2024.

[4] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3238–3251, 2014.

[5] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[6] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear model predictive control: towards new challenging applications*, pp. 345–369, Springer, 2009.

[7] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[8] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American control conference (ACC)*, pp. 1520–1527, IEEE, 2018.

[9] Y. Cao and R. B. Gopaluni, "Deep neural network approximation of nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11319–11324, 2020.

[10] E. T. Maddalena, C. d. S. Moraes, G. Waltrich, and C. N. Jones, "A neural network architecture to learn explicit mpc controllers from data," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11362–11367, 2020.

[11] M. Canale, L. Fagiano, and M. Milanese, "Set membership approximation theory for fast implementation of model predictive control laws," *Automatica*, vol. 45, no. 1, pp. 45–54, 2009.

[12] L. Huang, J. Lygeros, and F. Dörfler, "Robust and kernelized data-enabled predictive control for nonlinear systems," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 2, pp. 611–624, 2023.

[13] Y. Sasaki and D. Tsubakino, "Explicit model predictive control with gaussian process regression for flows around a cylinder," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 38–43, 2018.

[14] J. Ren, Q. Mao, T. Zhao, and Y. Cao, "Exact learning of linear model predictive control laws using oblique decision trees with linear predictions," in *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 7018–7023, IEEE, 2025.

[15] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[16] A. Tokmak, C. Fiedler, M. N. Zeilinger, S. Trimpe, and J. Köhler, "Automatic nonlinear mpc approximation with closed-loop guarantees," *IEEE Transactions on Automatic Control*, 2025.

[17] L. Grüne, "Analysis and design of unconstrained nonlinear mpc schemes for finite and infinite dimensional systems," *SIAM Journal on Control and Optimization*, vol. 48, no. 2, pp. 1206–1228, 2009.

[18] L. Grune and A. Rantzer, "On the infinite horizon performance of receding horizon controllers," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2100–2111, 2008.

[19] P. Braun, J. Pannek, and K. Worthmann, "Predictive control algorithms: Stability despite shortened optimization horizons," *IFAC Proceedings Volumes*, vol. 45, no. 25, pp. 274–279, 2012.

[20] L. Grüne, "Nmpc without terminal constraints," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 1–13, 2012.

[21] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 674–678, 2005.

[22] E. D. Sontag, "Control-lyapunov functions," in *Open problems in mathematical systems and control theory*, pp. 211–216, Springer, 1999.

[23] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel, "Model predictive control: for want of a local control lyapunov function, all is not lost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 546–558, 2005.

[24] A. Castellano, S. Pan, and E. Mallada, "Data-driven acceleration of mpc with guarantees," *Learning for Dynamics and Control Conference*, 2026.

[25] D. P. Bertsekas *et al.*, "Dynamic programming and optimal control 3rd edition, volume ii," *Belmont, MA: Athena Scientific*, vol. 1, 2011.

[26] Gurobi Optimization, LLC, "What is the mipgap?." `https://support.gurobi.com/hc/en-us/articles/8265539575953-What-is-the-MIPGap`, 2023. Accessed: 2026-03-18.

[27] IBM ILOG CPLEX, "Cplex parameter reference manual: Epgap." `https://www-eio.upc.edu/lceio/manuals/cplex-11/html/refparameterscplex/refparameterscplex40.html`, 2011. Accessed: 2026-03-18.

[28] H. Sangrody and N. Zhou, "An initial study on load forecasting considering economic factors," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, IEEE, 2016.

[29] Z. Guo, H. Yu, and J. Xi, "Time-optimal learning-based ltv-mpc for autonomous racing," in *Advanced Vehicle Control Symposium*, pp. 228–234, Springer, 2024.

[30] A. Castellano, S. Rezaei, J. Markowitz, and E. Mallada, "Nonparametric policy improvement in continuous action spaces via expert demonstrations," in *Reinforcement Learning Conference*, 2025.

[31] P. O. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Transactions on automatic control*, vol. 43, no. 8, pp. 1163–1169, 2002.

[32] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[33] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding-horizon control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 5, pp. 776–783, 2002.

[34] L. Buşoniu, E. Páll, and R. Munos, "Continuous-action planning for discounted infinite-horizon nonlinear optimal control with lipschitz values," *Automatica*, vol. 92, pp. 100–108, 2018.

[35] H. Harder and S. Peitz, "On the continuity and smoothness of the value function in reinforcement learning and optimal control," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 1935–1940, IEEE, 2024.

[36] A. Mahajan, "Stochastic control and markov decision processes." `https://adityam.github.io/stochastic-control/`, 2024. Online lecture notes.

[37] J. B. Rawlings, "Nonlinear model predictive control: Closed-loop properties." `https://www.syscop.de/files/2025ss/rmpc25/slides/lecture01_nmpc_1_rawlings.pdf`, 2025. Lecture slides, Systems Control and Optimization Laboratory, University of Freiburg, Sept. 15–19, 2025.

APPENDIX I
ON THE BELLMAN OPERATOR $\mathcal{T}^\pi$

The monotonicity and the fixed point uniqueness of $\mathcal{T}^\pi$ are defined as follows.

1. For any policy π, $\mathcal{T}^\pi$ is monotone, i.e.

$$\hat{J}_1(\mathbf{x}) \leq \hat{J}_2(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{X},\ \forall \hat{J}_1, \hat{J}_2 \in \mathcal{J}$$
$$\implies \big(\mathcal{T}^\pi \hat{J}_1\big)(\mathbf{x}) \leq \big(\mathcal{T}^\pi \hat{J}_2\big)(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{X},\ \forall \hat{J}_1, \hat{J}_2 \in \mathcal{J}.$$

2. For any policy π, J_π is the unique fixed point of T^π:

$$\lim_{t\to\infty} \underbrace{\big(\mathcal{T}^\pi \circ \mathcal{T}^\pi \circ \cdots \circ \mathcal{T}^\pi\big)}_{t \text{ times}}\ \hat{J} = J_\pi,\ \forall \hat{J} \in \mathcal{J}.$$

where the uniqueness and existence of J_π is guaranteed by $\gamma \in (0, 1)$.

APPENDIX II
INTERPRETATION OF ASSUMPTION 5

For the stage cost dominance condition (6), we argue that it is not restrictive. Based on [37, Assumption 11, Remark 12], if $0 \in \mathbb{X}$ is the unique equilibrium point, standard MPC assumptions imply the existence of $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that

$$l(\mathbf{x}, u) \geq \alpha_1(\|\mathbf{x}\|_{\mathbb{X}}), \quad \forall \mathbf{x} \in \mathbb{X},\ u \in \mathbb{U},$$
$$J_N(\mathbf{x}) \leq J(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|_{\mathbb{X}}), \quad \forall \mathbf{x} \in \mathbb{X}.$$

If, in addition, α_1 and α_2 are of comparable growth, i.e., there exists $v > 0$ such that $\alpha_1(r) \geq v\,\alpha_2(r)$, $\forall r \geq 0$, then it follows that

$$l(\mathbf{x}, u) \geq \alpha_1(\|\mathbf{x}\|_{\mathbb{X}}) \geq v\,\alpha_2(\|\mathbf{x}\|_{\mathbb{X}}) \geq v\,J_N(\mathbf{x}), \forall \mathbf{x} \in \mathbb{X}, u \in \mathbb{U}$$

which implies (6).

Appendix III
Proof of Proposition 1

We prove Proposition 1 by two parts.

Part 1. Multiplicative bound between $J(\mathbf{x})$ and $J_N(\mathbf{x})$: Without loss of generality, let $\mathbf{x} = \mathbf{x}_0$, for a given input sequence $\mathbf{u}$, define the truncated cost $V_N(\mathbf{x}_0, \mathbf{u}) = \sum_{t=0}^{N-1} \gamma^t l(\mathbf{x}_t, \mathbf{u}_t)$, and the augmented cost $V(\mathbf{x}_0, \mathbf{u}) = V_N(\mathbf{x}_0, \mathbf{u}) + \gamma^N J(\mathbf{x}_N)$, which corresponds to the finite-horizon cost with the true tail. Let $\mathbf{u}^\star$ be the optimal control sequence associated with $J(\mathbf{x}_0)$ in (4). To simplify notation, we denote $\mathbf{x}_i^F(\mathbf{x}_0)$ and $\mathbf{u}_i^F(\mathbf{x}_0)$ by $\mathbf{x}_i^F$ and $\mathbf{u}_i^F$, respectively, for $i = 0, \dots, N-1$. Then,

$$\begin{aligned} &J(\mathbf{x}_0) - J_N(\mathbf{x}_0) \\ &= \underbrace{J(\mathbf{x}_0) - V_N(\mathbf{x}_0, \mathbf{u}^\star)}_{\text{(I)}} + \underbrace{V_N(\mathbf{x}_0, \mathbf{u}^\star) - J_N(\mathbf{x}_0)}_{\text{(II)}}. \end{aligned} \tag{17}$$

Using the definition of $J(\mathbf{x}_0)$ and Assumption 3, we bound (I) as,

$$J(\mathbf{x}_0) - V_N(\mathbf{x}_0, \mathbf{u}^\star) = \gamma^N J(\mathbf{x}_N) \le \gamma^N J(\mathbf{x}_0).$$

On the other hand, to bound (II), we let $\mathbf{u}^F$ denote the optimal control sequence of $J_N(\mathbf{x}_0)$. By optimality of $\mathbf{u}^\star$ in (4), we have,

$$V(\mathbf{x}_0, \mathbf{u}^F) \ge V(\mathbf{x}_0, \mathbf{u}^\star) = J(\mathbf{x}_0).$$

Then,

$$V_N(\mathbf{x}_0, \mathbf{u}^F) + \gamma^N J(\mathbf{x}_N^F) \overset{(i)}{\ge} V_N(\mathbf{x}_0, \mathbf{u}^\star) + \gamma^N J(\mathbf{x}_N),$$

(i) is obtained from expanding both sides from the definition of V_N. Next,

$$\begin{aligned} V_N(\mathbf{x}_0, \mathbf{u}^F) + C\gamma^N J(\mathbf{x}_0) &\overset{(ii)}{\ge} V_N(\mathbf{x}_0, \mathbf{u}^F) + \gamma^N J(\mathbf{x}_N^F) \\ &\overset{(i)}{\ge} V_N(\mathbf{x}_0, \mathbf{u}^\star) + \gamma^N J(\mathbf{x}_N), \end{aligned}$$

where applying (7) in Assumption 5 leads to (ii). At last, we have,

$$\begin{aligned} C\gamma^N J(\mathbf{x}_0) &\ge V_N(\mathbf{x}_0, \mathbf{u}^\star) - V_N(\mathbf{x}_0, \mathbf{u}^F) + \gamma^N J(\mathbf{x}_N) \\ &\overset{(iii)}{\ge} V_N(\mathbf{x}_0, \mathbf{u}^\star) - V_N(\mathbf{x}_0, \mathbf{u}^F) \\ &\ge V_N(\mathbf{x}_0, \mathbf{u}^\star) - J_N(\mathbf{x}_0), \end{aligned}$$

where (iii) is from rearranging (ii) and applying $J(\mathbf{x}_N) \ge 0$. Hence, (II) is bounded by $C\gamma^N J(\mathbf{x}_0)$. Last, combining the upper bound of (I), (II) and rearranging terms, we obtain (8).

Part 2. δ-Bellman inequality of $J_N(\mathbf{x})$: We consider the Bellman type of inequality on $J_N(\mathbf{x}_0)$, $\mathbf{x}_0 \in \mathbb{X}$ under first optimal control input u_0^F, then,

$$\begin{aligned} &\gamma J_N(\mathbf{x}_1^F) - J_N(\mathbf{x}_0) \\ &\overset{(iv)}{\le} \left[\sum_{t=1}^{N-1} \gamma^t l(\mathbf{x}_t^F, \mathbf{u}_t^F) + \gamma^N l(\mathbf{x}_N^F, \mathbf{u}_N^\star) + \gamma^{N+1} F(\mathbf{x}_{N+1}^\dagger)\right] \\ &\quad - \left[l(\mathbf{x}_0, \mathbf{u}_0^F) + \sum_{t=1}^{N-1} \gamma^t l(\mathbf{x}_t^F, \mathbf{u}_t^F) + \gamma^N F(\mathbf{x}_N^F)\right] \\ &= -l(\mathbf{x}_0, \mathbf{u}_0^F) + \gamma^N l(\mathbf{x}_N^F, \mathbf{u}_N^\star) + \gamma^{N+1} F(\mathbf{x}_{N+1}^\dagger) - \gamma^N F(\mathbf{x}_N^F) \\ &\overset{(v)}{\le} -l(\mathbf{x}_0, \mathbf{u}_0^F) + \gamma^N l(\mathbf{x}_N^F, \mathbf{u}_N^\star) + \gamma^{N+1} J(\mathbf{x}_{N+1}^\dagger) - \gamma^N F(\mathbf{x}_N^F) \\ &\overset{(vi)}{\le} -l(\mathbf{x}_0, \mathbf{u}_0^F) + \gamma^N J(\mathbf{x}_N^F) \\ &\overset{(vii)}{\le} -l(\mathbf{x}_0, \mathbf{u}_0^F) + C\gamma^N J(\mathbf{x}_0), \end{aligned}$$

where (iv) follows by applying the shifted control sequence $\{\mathbf{u}_t^F\}_{t=1}^{N-1}$ obtained from (5) at $\mathbf{x}_0$ to the state $\mathbf{x}_1^F$, and appending the infinite-horizon optimal input $\mathbf{u}_N^\star$ to get $\mathbf{x}_{N+1}^\dagger$. (v) comes from the fact that $F(\mathbf{x}_{N+1}^\dagger) \le J(\mathbf{x}_{N+1}^\dagger)$. (vi) is obtained from throw $F(\mathbf{x}_N^F) \ge 0$ away and have $l(\mathbf{x}_N^F, \mathbf{u}_N^\star) + \gamma J(\mathbf{x}_{N+1}^\dagger) = J(\mathbf{x}_N^F)$ by Bellman equation. Finally, by applying $J(\mathbf{x}_N^F) \le CJ(\mathbf{x}_0)$, we obtain (vii).

Considering the v-relative lower boundness (Assumption 5) and changing $J(\mathbf{x}_0)$ by its upper bound $\frac{J_N(\mathbf{x}_0)}{1-\gamma^N(1+C)}$, we have,

$$\begin{aligned} &\left(1 + \frac{C\gamma^N}{1-\gamma^N(1+C)}\right) J_N(\mathbf{x}_0) - l(\mathbf{x}_0, \mathbf{u}_0^F) \ge \gamma J_N(\mathbf{x}_1^F) \\ &\left(1 + \frac{C\gamma^N}{1-\gamma^N(1+C)}\right) J_N(\mathbf{x}_0) - \gamma J_N(\mathbf{x}_1^F) \ge l(\mathbf{x}_0, \mathbf{u}_0^F) \\ &\left(1 - \frac{C\gamma^N}{v(1-\gamma^N(1+C))}\right) l(\mathbf{x}_0, \mathbf{u}_0^F) \le J_N(\mathbf{x}_0) - \gamma J_N(\mathbf{x}_1^F). \end{aligned}$$

Thus by assigning $\delta \le 1 - \frac{C\gamma^N}{v(1-\gamma^N(1+C))}$, we finished the proof. It is remarkable that we need $N > \frac{\log(1+C)}{\log\left(\frac{1}{\gamma}\right)}$ to ensure $1 - \gamma^N(1+C) > 0$ to let the δ-bound effective.

Appendix IV
Proof of Theorem 2

From $J(\mathbf{x}) \ge J_N(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{X}$ (Lemma 1), we may bound the objective as

$$\sup_{\mathbf{x}\in\mathbb{X}} \frac{J_{\pi_\mathcal{D}}(\mathbf{x}) - J(\mathbf{x})}{J(\mathbf{x}) + \eta} \le \sup_{\mathbf{x}\in\mathbb{X}} \frac{J_{\pi_\mathcal{D}}(\mathbf{x}) - J_N(\mathbf{x})}{J_N(\mathbf{x}) + \eta}.$$

Hence, a sufficient condition on radius $r_\mathbf{x} = \|\mathbf{x} - \tilde{\mathbf{x}}_i\|$, $\exists i \in [|\mathcal{D}|]$ is described as follows.

$$\begin{aligned} J_{\pi_\mathcal{D}}(\mathbf{x}) - J_N(\mathbf{x}) &= \underbrace{J_{\pi_\mathcal{D}}(\mathbf{x}) - \tilde{\mathbf{J}}_i}_{\text{(III)}} + \underbrace{\tilde{\mathbf{J}}_i - J_N(\mathbf{x})}_{\text{(IV)}} \\ &\overset{(xiii)}{\le} (\delta^{-1} - 1)\tilde{\mathbf{J}}_i + \delta^{-1}\lambda r_\mathbf{x} + L_J r_\mathbf{x} \\ &\overset{(xiv)}{\le} \mu(\tilde{\mathbf{J}}_i - L_J r_\mathbf{x} + \eta) \overset{(xv)}{\le} \mu(J_N(\mathbf{x}) + \eta), \end{aligned}$$

The term (III) follows from Theorem 1. The term (IV) is upper bounded by the Lipschitz continuity of $J_N(\mathbf{x})$ (Assumption 6), which yields $(xiii)$. Inequality (xiv) is the condition we want to enforce, and (xv) again follows from the Lipschitz continuity in Assumption 6. To guarantee (xiv), given $1 - \delta^{-1} + \mu \ge 0$, the covering radius $r_\mathbf{x}$ should satisfy,

$$\begin{aligned} (\delta^{-1} - 1)\tilde{\mathbf{J}}_i + \delta^{-1}\lambda r_\mathbf{x} + L_J r_\mathbf{x} &\le \mu(\tilde{\mathbf{J}}_i - L_J r_\mathbf{x} + \eta) \\ \delta^{-1}\lambda r_\mathbf{x} + (1+\mu) L_J r_\mathbf{x} &\le (1 - \delta^{-1} + \mu)\tilde{\mathbf{J}}_i + \mu\eta \\ r_\mathbf{x} &\le \frac{(1-\delta^{-1})\tilde{\mathbf{J}}_i + \mu(\tilde{\mathbf{J}}_i + \eta)}{\delta^{-1}\lambda + (1+\mu)L_J}, \end{aligned}$$

Therefore, by assigning $\tilde{\mathbf{J}}_i = 0$ from $\inf_{\mathbf{x}\in\mathbb{X}} J_N(\mathbf{x}) \ge 0$ (Assumption 2), the sample complexity of Algorithm 1 is

$$\mathcal{O}\left(N_{\text{cover}}\left(\frac{\mu\eta}{\delta^{-1}\lambda + (1+\mu)L_J}, \mathbb{X}\right)\right).$$