

Stability-Preserving Online Adaptation of Neural Closed-loop Maps

Danilo Saccani, Luca Furieri, Giancarlo Ferrari-Trecate

Abstract—The growing complexity of modern control tasks calls for controllers that can react online as objectives and disturbances change, while preserving closed-loop stability. Recent approaches for improving the performance of nonlinear systems while preserving closed-loop stability rely on time-invariant recurrent neural-network controllers, but offer no principled way to update the controller during operation. Most importantly, switching from one stabilizing policy to another can itself destabilize the closed-loop. We address this problem by introducing a stability-preserving update mechanism for nonlinear, neural-network-based controllers. Each controller is modeled as a causal operator with bounded ℓ_p -gain, and we derive gain-based conditions under which the controller may be updated online. These conditions yield two practical update schemes, time-scheduled and state-triggered, that guarantee the closed-loop remains ℓ_p -stable after any number of updates. Our analysis further shows that stability is decoupled from controller optimality, allowing approximate or early-stopped controller synthesis. We demonstrate the approach on nonlinear systems with time-varying objectives and disturbances, and show consistent performance improvements over static and naive online baselines while guaranteeing stability.

I. INTRODUCTION

Modern control systems operate in increasingly dynamic settings, enabled by advances in sensing, onboard computation, and model-based design. While closed-loop stability is a core requirement, alone it is not enough, as applications also demand good tracking, efficiency, and constraint handling under time-varying operating conditions. Nonlinear Optimal Control (NOC) provides a natural framework to improve performance by minimizing a task-dependent cost [1]. However, enforcing stability while optimizing a general nonlinear cost is difficult, especially for complex and time-varying dynamics [2], [3]. Classical approaches such as dynamic programming and the maximum principle [1] face significant computational limitations, and existing stability guarantees typically require restrictive assumptions on the cost, which limits applicability in practice.

Nonlinear Model Predictive Control (NMPC) offers an approximate solution to the NOC problem by implementing an implicit policy obtained from a finite-horizon problem solved at each time step in receding-horizon fashion [2].

This work was supported as a part of NCCR Automation, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40_225155), the Swiss National Science Foundation Ambizione (grant number PZ00P2_208951) and the NECON project (grant number 200021-219431).

D. Saccani and G. Ferrari-Trecate are with the Institute of Mechanical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. (email: {danilo.saccani, giancarlo.ferraritrecate}@epfl.ch). L. Furieri is with the Department of Engineering Science, University of Oxford, United Kingdom (luca.furieri@eng.ox.ac.uk);

While NMPC provides a structured way to handle constraints and nonlinearities, it typically imposes strict conditions on the cost to guarantee stability, which makes it difficult to apply in scenarios with rapidly changing objectives [4].

Neural networks (NNs) can parametrize rich families of high-performance feedback policies [5], but certifying closed-loop stability remains a nontrivial task. Indeed, generic NN architectures do not lend themselves to the utilization of Lyapunov or small-gain conditions [6].

Recent works utilize System Level Synthesis (SLS) [7], [8], Internal Model Control (IMC) [9], [10], and Youla parametrizations [11]–[13] to improve nonlinear closed-loop performance while guaranteeing ℓ_p -stability. The key idea is to search over a space of stable \mathcal{L}_p operators used to parametrize the controller. In practice, these operators are implemented as time-invariant neural parametrizations whose stability is certified for any choice of weights. The main limitation is that the controller is fixed after training: the weights are not updated online, so significant offline training is required to cover changing operating conditions.

A key obstacle to online adaptation is that replacing one stabilizing policy with another may destabilize the transient closed loop, even when both policies are stabilizing in isolation, as known from switched-systems theory [14]–[16].

In this work, building on the neural IMC architecture of [10], we propose gain-budgeted triggering mechanisms, in both time-scheduled and state-triggered form, that certify when a newly optimized controller may safely replace the current one.

The main contributions of this work are: (i) a gain-budgeted update condition, derived from finite-gain and small-gain arguments, that guarantees the closed-loop remains ℓ_p -stable under repeated online controller updates; (ii) two implementable triggering mechanisms, one time-based and one state-based, that enforce this condition while trading off computational effort against adaptivity; and (iii) evidence on two nonlinear benchmarks of consistent improvements over offline baselines and receding-horizon open-loop controllers.

NOTATION

We denote by \mathbb{N} the set of non-negative integers and $\mathbb{N}_{\geq a}$ the integers $\{a, a + 1, \dots\}$. The set of all sequences $\mathbf{v} = (v_0, v_1, v_2, \dots)$, where $v_t \in \mathbb{R}^n$ for all $t \in \mathbb{N}$, is denoted as ℓ^n . Moreover, \mathbf{v} belongs to $\ell_p^n \subset \ell^n$ with $p \in [1, \infty]$ if $\|\mathbf{v}\|_p = (\sum_{t=0}^{\infty} |v_t|^p)^{\frac{1}{p}} < \infty$, where $|\cdot|$ denotes an arbitrarily chosen vector norm. We say that $\mathbf{v} \in \ell_\infty^n$ if $\sup_t \|v_t\| < \infty$. We refer to $v_{0:T}$ to denote the truncation of \mathbf{v} with t ranging from 0 to T . An operator $\mathbf{A} : \ell^n \rightarrow \ell^m$ is said to be

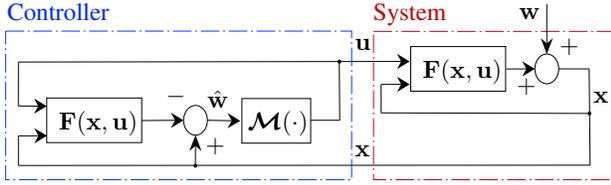


Fig. 1: IMC architecture parametrizing all stability-preserving controllers via a free operator $\mathcal{M} \in \mathcal{L}_p$.

causal if $\mathbf{A}(\mathbf{x}) = (A_0(x_0), A_1(x_{0:1}), \dots, A_t(x_{0:t}), \dots)$. If $A_t(x_{0:t}) = A_t(0, x_{0:t-1})$, then \mathbf{A} is said to be *strictly causal*. An operator $\mathbf{A} : \ell^n \rightarrow \ell^m$ is ℓ_p -stable if it is causal and $\mathbf{A}(\mathbf{a}) \in \ell_p^m$ for all $\mathbf{a} \in \ell_p^n$. Equivalently, we write $\mathbf{A} \in \mathcal{L}_p$. We say that an \mathcal{L}_p operator $\mathbf{A} : \mathbf{w} \mapsto \mathbf{u}$ has a finite \mathcal{L}_p -gain $\gamma_p(\mathbf{A}) > 0$ if $\|\mathbf{u}\|_p \leq \gamma_p(\mathbf{A})\|\mathbf{w}\|_p$, for all $\mathbf{w} \in \ell_p^n$. When p is clear from the context, we simply write the gain as $\gamma(\mathbf{A}) > 0$ and the norm as $\|\cdot\|$.

II. PRELIMINARIES

Let us consider the following discrete-time, time-varying nonlinear system:

$$x_t = f_{t-1}(x_{t-1}, u_{t-1}) + w_t, \quad t = 1, 2, \dots, \quad (1)$$

where $x_t \in \mathbb{R}^n$ is the state vector, $u_t \in \mathbb{R}^m$ is the control input, and $w_t \in \mathbb{R}^n$ represents an unknown process noise embedding the initial value of the system state by setting $w_0 = x_0$. Rewriting system (1) in operator form, we obtain:

$$\mathbf{x} = \mathbf{F}(\mathbf{x}, \mathbf{u}) + \mathbf{w}, \quad (2)$$

where $\mathbf{F} : \ell^n \times \ell^m \rightarrow \ell^n$ is a strictly causal operator defined as $\mathbf{F}(\mathbf{x}, \mathbf{u}) = (0, f_0(x_0, u_0), \dots, f_{t-1}(x_{t-1}, u_{t-1}), \dots)$. We model the system as the input-to-state operator

$$\mathcal{F} : (\mathbf{u}, \mathbf{w}) \mapsto \mathbf{x} \in \mathcal{L}_p, \quad (3)$$

and assume \mathcal{F} is causal and admits a finite \mathcal{L}_p -gain $\gamma(\mathcal{F}) < \infty$. This is satisfied when the plant is stable, or when f_t represents the dynamics of a system in closed-loop with a stabilizing controller K^{t1} . Additionally, we assume that $\mathbf{w} \in \ell_p$ and that the process noise w_t follows a distribution \mathcal{D} . To regulate system (1), we consider a nonlinear, time-varying state-feedback controller of the form:

$$\mathbf{u} = \mathbf{K}(\mathbf{x}) = (K_0(x_0), K_1(x_{0:1}), \dots, K_t(x_{0:t}), \dots). \quad (4)$$

where $\mathbf{K} : \ell^n \rightarrow \ell^m$ is a causal operator to be designed. Due to causality, each disturbance sequence $\mathbf{w} \in \ell^n$ induces unique trajectories in the closed-loop system (1)–(4) (see e.g. [10]). For a given system \mathbf{F} and controller \mathbf{K} , we denote by $\Phi^x[\mathbf{F}, \mathbf{K}]$ and $\Phi^u[\mathbf{F}, \mathbf{K}]$ the induced closed-loop operators mapping $\mathbf{w} \mapsto \mathbf{x}$ and $\mathbf{w} \mapsto \mathbf{u}$, respectively. Thus,

$$\mathbf{x} = \Phi^x[\mathbf{F}, \mathbf{K}](\mathbf{w}), \quad \mathbf{u} = \Phi^u[\mathbf{F}, \mathbf{K}](\mathbf{w}), \quad \forall \mathbf{w} \in \ell^n.$$

Our objective is to design a policy $\mathbf{K}(\mathbf{x})$ addressing the following problem:

¹The latter case is the one considered in the experiments in Section V.

Problem 1: Design \mathbf{K} by solving the infinite-horizon Nonlinear Optimal Control (NOC) problem:

$$\min_{\mathbf{K}(\cdot)} \mathbb{E}_{\mathbf{w}} [l(\mathbf{x}, \mathbf{u})] \quad (5a)$$

$$\text{s.t.} \quad x_t = f_{t-1}(x_{t-1}, u_{t-1}) + w_t, \quad w_0 = x_0, \quad (5b)$$

$$u_t = K_t(x_{0:t}), \quad \forall t = 0, 1, \dots, \quad (5c)$$

$$(\Phi^x[\mathbf{F}, \mathbf{K}], \Phi^u[\mathbf{F}, \mathbf{K}]) \in \mathcal{L}_p, \quad (5d)$$

where $l : \ell^n \times \ell^m \rightarrow \mathbb{R}$ is any piecewise differentiable loss satisfying $l(\mathbf{x}, \mathbf{u}) \geq 0$. This loss can be defined through a (possibly time-varying) stage cost, e.g., $l(\mathbf{x}, \mathbf{u}) = \sum_{t=0}^{\infty} c_t(x_t, u_t)$, and is meant to quantify infinite-horizon closed-loop performance. \square

The expectation $\mathbb{E}_{\mathbf{w}}[\cdot]$ captures the impact of disturbances \mathbf{w} on the loss, while (5d) imposes a hard ℓ_p -stability constraint on the closed-loop system.

It has been shown in [8], [10] that the NOC problem admits an equivalent formulation by exploiting an IMC control architecture based on an operator $\mathcal{M} : \ell^n \rightarrow \ell^m$ to be designed. Specifically, all and only stability-preserving controllers, i.e., controllers verifying (5d), are parametrized by $\mathcal{M} \in \mathcal{L}_p$. The control architecture proposed in [8], [10] is shown in Figure 1 and incorporates a model of the system dynamics to estimate the disturbance \mathbf{w} . The following theorem summarizes these results:

Theorem 1: (adapted from Theorem 1 in [10]) Assume that the operator \mathcal{F} is ℓ_p -stable, and consider the evolution of (2) where \mathbf{u} is defined as

$$\mathbf{u} = \mathcal{M}(\mathbf{x} - \mathbf{F}(\mathbf{x}, \mathbf{u})), \quad (6)$$

for a causal operator $\mathcal{M} : \ell^n \rightarrow \ell^m$. Let \mathbf{K} be the operator for which $\mathbf{u} = \mathbf{K}(\mathbf{x})$ is equivalent to (6). The following two statements hold true.

- (i) If $\mathcal{M} \in \mathcal{L}_p$, then the closed-loop system is ℓ_p -stable.
- (ii) If there is a causal policy \mathbf{C} such that $\Phi^x[\mathbf{F}, \mathbf{C}], \Phi^u[\mathbf{F}, \mathbf{C}] \in \mathcal{L}_p$, then

$$\mathcal{M} = \Phi^u[\mathbf{F}, \mathbf{C}],$$

gives $\mathbf{K} = \mathbf{C}$.

Theorem 1 implies that exploring the space of operators $\mathcal{M} \in \mathcal{L}_p$ is sufficient to identify *all and only stability-preserving* policies, i.e. all policies satisfying (5d). This suggests that solving Problem 1 is equivalent to replacing $\mathbf{u} = \mathbf{K}(\mathbf{x})$ in (5c) with (6) and $\min_{\mathbf{K}(\cdot)}$ in (5a) with $\min_{\mathcal{M} \in \mathcal{L}_p}$. However, solving Problem 1 through numerical methods requires some simplifications. First, searching within the infinite-dimensional space \mathcal{L}_p is infeasible. This suggests considering subsets $\mathcal{O} \subset \mathcal{L}_p$ of operators $\mathcal{M}(\theta)$ depending on finitely many parameters $\theta \in \mathbb{R}^d$. Various methods for defining different sets \mathcal{O} are provided in [17], [18]. With the structures proposed in these papers, the operators \mathcal{M} can be modelled using time-invariant nonlinear dynamical systems embedding NNs. The formulations in [17], [19], [20] also allow imposing an explicit upper bound on the operator's ℓ_2 -gain, a feature we will exploit in Section III for control design. We highlight that the resulting operators can be

larger controller gains amplify both state and input on each inter-update window. Note that (11) is an admissibility condition: an update at time t_i is only allowed if the condition is met. We do not require t_i to be the first time this condition becomes true; any t_i satisfying (11) is permissible. In the next section, we show how to select $\{r^{(i)}\}_{i \geq 1}$, $\{\epsilon^{(i)}\}_{i \geq 1}$, and the triggering mechanism (9) so as to manage this trade-off.

IV. IMPLEMENTATION

In this section, we derive practical update rules from Theorem 2. An update is admissible if the state at switching is sufficiently small and the gain of the new operator is compatible with that state through (10)–(11). The key design quantities are the thresholds $\epsilon^{(i)}$ and the sequence $\mathbf{r} = \{r^{(i)}\}$: larger thresholds make updates easier to trigger but force smaller gains, whereas smaller thresholds allow more aggressive controllers but delay the update.

We consider a time-scheduled scheme and a state-triggered scheme. Both require an upper bound $\hat{\gamma}(\mathcal{F})$ on the gain of the stable or pre-stabilized plant; conservatism only makes updates more cautious. In the ℓ_2 case, such a bound can be obtained from a storage inequality under the convention $w_0 = x_0$. Since stability is enforced by the gain condition rather than by optimality, problem (8) may be solved only approximately, for instance using shorter horizons, fewer samples, or budgeted warm-started gradient steps. A practical discussion on choosing \mathbf{r} can be found in Appendix B.

A. Algorithm 1: Predefined Update Times

We first consider a time-scheduled update mechanism, where μ in (9) is assumed to be constant, in which the designer selects (i) an update period $t_{\text{opt}} \in \mathbb{N}_{\geq 1}$ and (ii) a nonnegative budget sequence $\{r^{(i)}\}_{i \geq 1} \in \ell_p$. Updates are only attempted every t_{opt} steps:

$$t_0 = 0, \quad t_{i+1} = t_i + t_{\text{opt}}, \quad i \in \mathbb{N}_{\geq 1}.$$

At each scheduled time t_i , we (a) measure the current state and set $\epsilon^{(i)} := |x_{t_i}|$, and (b) check if an update is admissible under Theorem 2. Rearranging (10) yields an upper bound on the allowable gain of the new controller:

$$0 < \gamma_p(\mathcal{M}^{(i)}) \leq \frac{r^{(i)}}{\gamma_p(\mathcal{F}) \epsilon^{(i)}} - 1. \quad (12)$$

We also require $\gamma_p(\mathcal{M}^{(i)}) \leq \bar{\gamma}$, where $\bar{\gamma}$ is the uniform gain bound assumed in Theorem 2. If at t_i there is $\gamma_p(\mathcal{M}^{(i)})$ satisfying (12), we solve (8) (with that gain bound enforced) to obtain new parameters $\theta^{(i)}$ for $\mathcal{M}^{(i)}$, and we apply this controller on $[t_i, \dots, t_{i+1}-1]$. If it is infeasible (e.g., because $|x_{t_i}|$ is too large or $r^{(i)}$ is too small), we keep the previous controller for the next interval, $\mathcal{M}^{(i)} = \mathcal{M}^{(i-1)}$, which preserves ℓ_p -stability. Whenever there is $\gamma_p(\mathcal{M}^{(i)})$ satisfying (12), Theorem 2 guarantees that we can update the controller again and the resulting time-varying closed-loop remains ℓ_p -stable.

In summary, $t_{\text{opt}} \in \mathbb{N}_{\geq 1}$ is freely chosen (with no minimum dwell-time constraint) and determines how often we attempt an update and $r^{(i)}$ sets how aggressive the i -th

update is allowed to be: for a given state norm $|x_{t_i}|$, larger $r^{(i)}$ allows larger controller gain and thus potentially faster transients.

B. Algorithm 2: State-Dependent Triggering

The second scheme removes the fixed update period. Instead, the designer specifies (i) an admissible gain level $\gamma_p(\mathcal{M})$ that upper-bounds the \mathcal{L}_p -gain of any controller to be used, and (ii) a nonnegative sequence $\{r^{(i)}\}_{i \geq 1} \in \ell_p$.

Given $\gamma_p(\mathcal{M})$ and $r^{(i)}$, we define the admissible state threshold for the i -th update as

$$\epsilon^{(i)} = \frac{r^{(i)}}{\gamma_p(\mathcal{F})(\gamma_p(\mathcal{M}) + 1)},$$

which is exactly the feasibility condition of Theorem 2. We then set t_i to be the first time after t_{i-1} such that

$$|x_{t_i}| \leq \epsilon^{(i)}.$$

At that instant t_i , we solve (8) to obtain a new controller $\mathcal{M}^{(i)}$ with $\gamma_p(\mathcal{M}^{(i)}) \leq \gamma_p(\mathcal{M})$, and keep it active until the next trigger time t_{i+1} . This “self-triggered” rule inverts Algorithm 1. Rather than updating on a fixed schedule and then checking if the state is small enough, we wait until the state is small enough to admit an update of the operator \mathcal{M} . This can reduce unnecessary recomputations when the state is still large. The only design degrees of freedom are therefore the gain bound $\gamma_p(\mathcal{M})$ and the budget sequence $\{r^{(i)}\}$, which together determine when updates occur. Again, enforcing the gain cap $\gamma_p(\mathcal{M}^{(i)}) \leq \gamma_p(\mathcal{M})$ and trigger condition implies ℓ_p -stability by Theorem 2.

V. NUMERICAL EXAMPLE

In this section, we demonstrate the application of the proposed approach to robotic tasks. First, we apply our solution to the *mountains* problem from [10], [24] to highlight the advantages of our method. Then, we present a tracking problem in a time-varying environment with dynamic obstacles.

We consider a point-mass system of mass $m \in \mathbb{R}_+$ subject to nonlinear friction forces:

$$\begin{bmatrix} q_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} q_t + T_s v_t \\ v_t + T_s m^{-1} (F_t + G(v_t)) \end{bmatrix} + w_t, \quad (13)$$

where $q_t \in \mathbb{R}^2$ and $v_t \in \mathbb{R}^2$ represent the position and velocity of the system, respectively, while $F_t \in \mathbb{R}^2$ is the control input. The sampling time is denoted as $T_s = 0.05$ seconds. The term $w_t \in \mathbb{R}^4$ represents process noise. The nonlinear damping term is defined as: $G(v_t) = -b_1 v_t + b_2 \tanh(v_t)$, where $0 < b_2 < b_1$. To parameterize the operators $\mathcal{M}^{(i)} \in \mathcal{O}$, we employed a Recurrent Equilibrium Network (REN) [17] with prescribed gain. Problem (8) is solved using backpropagation through time to compute gradients (see [10]), and the Adam optimizer to update the parameters via stochastic gradient descent. The approach is implemented using PyTorch, and the code to reproduce the examples is available at: https://github.com/DecodEPFL/Online_neurSLS.git.

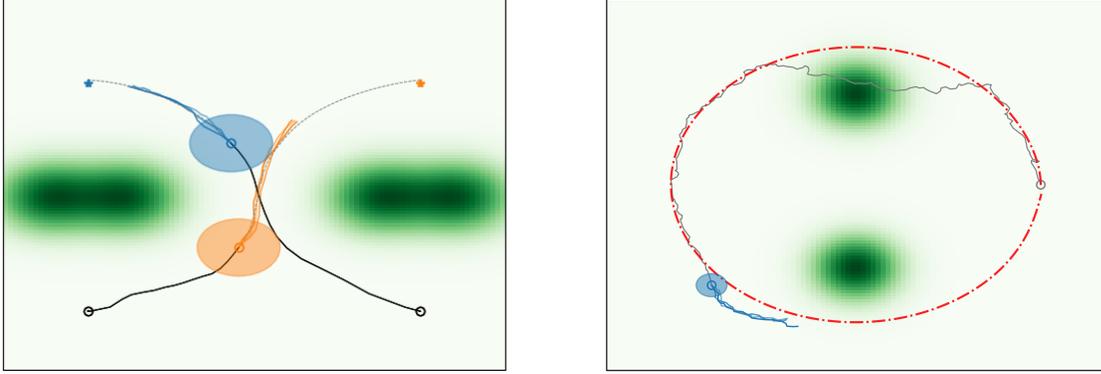


Fig. 2: Qualitative closed-loop behavior of Algorithm 1 in the two case studies. (Left) Mountains problem at $\tau = 0.8$ s. Green: obstacles. Colored lines: predicted trajectories over $[\tau, \tau + 1.25]$; black: executed trajectory over $[0, \tau)$; gray dashed: predicted continuation after τ . Colored disks show agent positions and radii. (Right) Dynamic-obstacles problem at $\tau_1 = 9.3$ s. Gray: executed trajectory; red dash-dot: reference; blue: predicted trajectory over $[\tau_1, \tau_1 + 0.5]$; green: obstacle positions at τ_1 .

A. Mountains problem

The *mountains* scenario depicted in Figure 2 features two agents aiming to collaboratively navigate through a narrow valley. Each agent $j \in \{1, 2\}$ evolves according to (13) and has a target position $\bar{q}_j \in \mathbb{R}^2$ that must be reached with zero velocity (i.e., $\bar{v}_j = 0_2$). We stack both agents' states, so the overall state is $x_t = [q_{t,1}, v_{t,1}, q_{t,2}, v_{t,2}]^\top \in \mathbb{R}^8$ and \bar{x} stacks the target positions and zero velocities. We study the regulation problem in error coordinates with respect to the desired equilibrium \bar{q}_j, \bar{v}_j . We apply a baseline linear feedback $K' = \text{diag}(k_1, k_2)$ with gains $k_1, k_2 > 0$, around the equilibrium, and we set $F_{t,j} = K'(\bar{q}_j - q_{t,j}) + u_{t,j}$, where $u_{t,j}$ is an auxiliary input. We treat $u_t = [u_{t,1}, u_{t,2}]^\top \in \mathbb{R}^4$, and disturbance $w_t \in \mathbb{R}^8$. The associated input-to-state operator $\mathcal{F}_e(u, w) \mapsto e$ has finite ℓ_2 -gain. An upper bound on this gain is obtained from a discrete-time bounded-real LMI [25]; see Appendix C.1. In this example, we set $p = 2$, and we model the disturbance sequence w as Gaussian noise with exponential decay. The loss function is defined as:

$$\tilde{l}(e_{0:T}, u_{0:T}) = \sum_{t=0}^T (l_{\text{traj}}(e_t, u_t) + l_{\text{ca}}(e_t) + l_{\text{obs}}(e_t)),$$

where $l_{\text{traj}}(e, u) = [e^\top \ u^\top]Q[e^\top \ u^\top]^\top$ with $Q \succeq 0$ penalizing deviation from the targets and control effort. The terms $l_{\text{ca}}(e)$ and $l_{\text{obs}}(e)$ are barrier functions that prevent collisions between agents and obstacles (see [10] for their definitions). We compare with the approach proposed in [10], where the controller is parametrized by an operator $\mathcal{M} \in \mathcal{L}_2$ obtained offline over a horizon $T = 100$, matching the total simulation time used for our online controller. Figure 2 (left) illustrates closed-loop trajectories obtained using Algorithm

1, which attempts an update to a new operator $\mathcal{M}^{(i)}$ at fixed intervals of $t_{\text{opt}} = 2$ time steps. Problem (8) is solved online to update the operator with a horizon of $H = 25$ and $S = 3$ disturbance realizations. Two scenarios are simulated: the nominal scenario replicating the example in [10], and a perturbed scenario involving abrupt impulse disturbances. Specifically, the local disturbance $w_{t,j}$ is replaced by:

$$w_{t,j} + 0.3\delta(t-1) + 0.3\delta(t-8), \quad j = 1, 2, \quad (14)$$

where δ is the Kronecker delta. Over 50 runs, our method reduces the average cost by 35.1% (nominal) and 40.3% (perturbed) versus [10], indicating improved adaptation to abrupt, previously unseen disturbance events; boxplots are reported in Appendix C.

B. Dynamic obstacles

In the second scenario (Figure 2), an agent tracks a circular path while avoiding dynamic obstacles. We track a reference trajectory $x_{\text{ref},t}$ by applying a baseline stabilizing feedback K' together with a feedforward term $u_{\text{ref},t}$. We define the tracking error $e_t = x_t - x_{\text{ref},t} \in \mathbb{R}^4$ and write the total input as $F_t = u_{\text{ref},t} - K'e_t + u_t$, where u_t is an auxiliary control signal that we optimize online. Under this pre-stabilization, the closed-loop error dynamics for e_t can be written as a nonlinear system $e_t = f_e(e_{t-1}, u_{t-1}) + w_t$, which matches the general form of (1) with state e_t , input u_t , and disturbance w_t . As in the previous example, the gain of the associated input-to-state operator $\mathcal{F}_e(u, w) \mapsto e$ is computed upper-bounding the worst-case realization of the nonlinearity. Two dynamic obstacles move on the y -axis following a perturbed sinusoidal trajectory:

$$y_{t,j}^{\text{obs}} = A \sin((2\pi\psi_j + \eta_j)t + \phi_j) + y_{0,j}^{\text{obs}}, \quad j = 1, 2. \quad (15)$$

where ψ_j is the nominal frequency, $\eta_j \sim \mathcal{N}(0, 0.01)$ is a random frequency perturbation, and ϕ_j is a random phase.

The finite-horizon objective \tilde{l} in (8) is now time-varying because the obstacles move. We define

$$\tilde{l}(e_{0:T}, u_{0:T}) = \sum_{t=0}^T (l_{\text{traj}}(e_t, u_t) + l_{\text{obs}}(e_t, t)).$$

where l_{traj} is a positive semidefinite quadratic penalty on tracking error and control effort, and $l_{\text{obs}}(e_t, t)$ penalizes collisions with the moving obstacle. The system is simulated for 300 time steps under a disturbance sequence generated at each step with bounded amplitude and interpreted as an element of ℓ_2 by zero extension beyond the simulation horizon. In other words, the disturbance is persistent over the considered horizon, but has finite energy on the full infinite sequence. Accordingly, in this experiment we set $p = 2$, consistently with the certified ℓ_2 -gain of the REN controller class. We compare against (i) the offline controller of [10] and (ii) a receding-horizon open-loop (RHO) planner that solves at time t the problem

$$\begin{aligned} \min_{u_{0:H-1}} \quad & \sum_{s=0}^S \tilde{l}(e_{0:H}, u_{0:H}) \\ \text{s.t.} \quad & e_{k+1}^s = f_e(e_k^s, u_k) + w_k^s, \quad e_0^s = e_t, \quad \forall s, \quad \mathbf{w}^s \sim \mathcal{D}, \end{aligned}$$

and applies the first input $u_t = u_0^*$, advances the state, and re-solves at $t+1$. For a fair comparison, all approaches use the same horizon ($H = 10$) and cost weights. Algorithm 1 attempts updates \mathcal{M} every $t_{\text{opt}} = 1$ time step, using a budgeted, inexact solution (100 epochs of gradient descent warm-started from the previous parameters). This is admissible under Theorem 2 because stability is guaranteed even without reaching local optimality. The right panel in Figure 2 shows the closed-loop trajectory (in grey) and the predicted trajectories (in blue). Persistent disturbances make perfect tracking unachievable, and apparent overlaps between moving obstacles and past trajectory segments are only visualization artifacts. Over 50 runs, the proposed approach achieves lower cost than both RHO and the offline controller from [10], while also exhibiting smaller variability; the corresponding boxplots are reported in Appendix C. Overall, the results indicate that online adaptation with closed-loop prediction improves both performance and reliability relative to the two baselines.

VI. CONCLUSIONS

We introduced a framework for iteratively updating neural-network-based controllers while maintaining closed-loop stability via gain-based update conditions. This opens several research directions. First, robustness margins against model-plant mismatch could be embedded directly into the update logic, preserving stability under imperfect models. Second, online optimization could be used to update both the control operator and the system model used by the controller. Finally, connecting these ideas to reinforcement learning by enforcing our gain-based admissibility condition during policy improvement may offer stability guarantees during learning, not only after convergence.

REFERENCES

- [1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.
- [2] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill Publishing, 2017.
- [3] D. Saccani, L. Fagiano, M. N. Zeilinger, and A. Carron, “Model predictive control for multi-agent systems under limited communication and time-varying network topology,” in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 3764–3769.
- [4] L. Grüne and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.
- [5] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, 1989.
- [6] E. D. Sontag, “Neural networks for control,” in *Essays on Control: Perspectives in the Theory and its Applications*. Springer, 1993.
- [7] D. Ho, “A system level approach to discrete-time nonlinear systems,” in *2020 American Control Conference (ACC)*. IEEE, 2020.
- [8] L. Furieri, C. L. Galimberti, and G. Ferrari-Trecate, “Neural system level synthesis: Learning over all stabilizing policies for nonlinear systems,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 2765–2770.
- [9] D. Saccani, L. Massai, L. Furieri, and G. Ferrari-Trecate, “Optimal distributed control with stability guarantees by training a network of neural closed-loop maps,” in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 3776–3781.
- [10] L. Furieri, C. L. Galimberti, and G. Ferrari-Trecate, “Learning to boost the performance of stable nonlinear systems,” *IEEE Open Journal of Control Systems*, 2024.
- [11] R. Wang and I. R. Manchester, “Youla-ren: Learning nonlinear feedback policies with robust stability guarantees,” in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 2116–2123.
- [12] N. H. Barbara, R. Wang, and I. R. Manchester, “Learning over contracting and lipschitz closed-loops for partially-observed nonlinear systems,” in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 1028–1033.
- [13] C. L. Galimberti, L. Furieri, and G. Ferrari-Trecate, “Parametrizations of all stable closed-loop responses: From theory to neural network control design,” *Annual Reviews in Control*, vol. 60, p. 101012, 2025.
- [14] D. Liberzon, *Switching in systems and control*. Springer, vol. 190.
- [15] A. S. Morse, “Control using logic-based switching,” *Trends in control: A European perspective*, pp. 69–113, 1997.
- [16] C. De Persis, R. De Santis, and A. S. Morse, “Switched nonlinear systems with state-dependent dwell-time,” *Systems & Control Letters*, vol. 50, no. 4, pp. 291–302, 2003.
- [17] M. Revay, R. Wang, and I. R. Manchester, “Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness,” *IEEE Transactions on Automatic Control*, 2023.
- [18] F. Bonassi, C. Andersson, P. Mattsson, and T. B. Schön, “Structured state-space models are deep wiener models,” *IFAC-PapersOnLine*, vol. 58, no. 15, pp. 247–252, 2024.
- [19] L. Massai and G. Ferrari-Trecate, “Free parametrization of 12-bounded state space models,” *arXiv preprint arXiv:2503.23818*, 2025.
- [20] M. Zakwan and G. Ferrari-Trecate, “Neural port-hamiltonian models for nonlinear distributed control: An unconstrained parametrization approach,” *arXiv preprint arXiv:2411.10096*, 2024.
- [21] D. Liberzon and A. S. Morse, “Basic problems in stability and design of switched systems,” *IEEE control systems magazine*, 1999.
- [22] H. Lin and P. J. Antsaklis, “Stability and stabilizability of switched linear systems: a survey of recent results,” *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009.
- [23] W. P. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *2012 IEEE 51st Conference on Decision and Control (CDC)*. IEEE, 2012.
- [24] D. Onken, L. Nurbekyan, X. Li, S. W. Fung, S. Osher, and L. Ruthotto, “A neural network approach applied to multi-agent optimal control,” in *2021 European Control Conference (ECC)*. IEEE, 2021.
- [25] C. E. de Souza and L. Xie, “On the discrete-time bounded real lemma with application in the characterization of static state feedback h_∞ controllers,” *Systems & Control Letters*, vol. 18, no. 1, 1992.
- [26] Z.-P. Jiang and Y. Wang, “Input-to-state stability for discrete-time nonlinear systems,” *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.

A. Proof of Theorem 2 and Input to State stability relation

We first establish an upper bound for the state and input norms when applying $\mathcal{M}^{(0)}$ on the first interval $[t_0, t_1 - 1]$, where $t_0 = 0$. From the definition of \mathcal{F} in (2) and the controller \mathcal{M} , the controller input is $\mathbf{z}^{(0)} = (x_0, w_1, \dots)$. From (1), one has $w_0 = x_0$, so $\mathbf{z}^{(0)} = \mathbf{w}$. Let $\bar{x}_0 = x_{0:t_1-1}$ and $\bar{w}_0 = w_{0:t_1-1}$.

$$\|\mathbf{u}_{0:t_1-1}\| \leq \gamma(\mathcal{M}^{(0)})\|\bar{w}_0\| \quad (16)$$

$$\|\bar{x}_0\| \leq \gamma(\mathcal{F})(\|\mathbf{u}_{0:t_1-1}\| + \|\bar{w}_0\|) \quad (17)$$

Substituting (16) into (17) yields:

$$\|\bar{x}_0\| \leq \underbrace{\gamma(\mathcal{F})(\gamma(\mathcal{M}^{(0)}) + 1)}_{C_0} \|\bar{w}_0\|. \quad (18)$$

Now consider the evolution on a subsequent window $i \geq 1$ on the interval $[t_i, t_{i+1} - 1]$. Let $\bar{x}_i = x_{t_i:t_{i+1}-1}$, $\bar{u}_i = u_{t_i:t_{i+1}-1}$, $\bar{w}_i = w_{t_i:t_{i+1}-1}$, and $\bar{z}_i = (x_{t_i}, w_{t_i+1}, \dots, w_{t_{i+1}-1})$. We obtain

$$\|\bar{x}_i\|_p \leq 2\gamma_p(\mathcal{F})|x_{t_i}| + \gamma_p(\mathcal{F})(\|\bar{u}_i\|_p + \|\bar{w}_i\|_p) \quad (19)$$

The controller's output \bar{u}_i is bounded by its gain with zero initial condition:

$$\|\bar{u}_i\|_p \leq \gamma_p(\mathcal{M}^{(i)})\|\bar{z}_i\|_p. \quad (20)$$

The input \bar{z}_i is bounded using the triangle inequality:

$$\|\bar{z}_i\|_p \leq |x_{t_i}| + \|w_{t_i+1:t_{i+1}-1}\|_p \leq |x_{t_i}| + \|\bar{w}_i\|_p \quad (21)$$

Substitute (20) and (21) into (19) and grouping terms by $|x_{t_i}|$ and $\|\bar{w}_i\|_p$ yields:

$$\begin{aligned} \|\bar{x}_i\|_p &\leq \gamma_p(\mathcal{F})(\gamma_p(\mathcal{M}^{(i)}) + 1)|x_{t_i}| \\ &\quad + \gamma_p(\mathcal{F})(\gamma_p(\mathcal{M}^{(i)}) + 1)\|\bar{w}_i\|_p \end{aligned}$$

Now, apply the triggering condition (11), $|x_{t_i}| \leq \epsilon^{(i)}$, and the gain condition (10), $\gamma_p(\mathcal{F})(\gamma_p(\mathcal{M}^{(i)}) + 1)\epsilon^{(i)} \leq r^{(i)}$:

$$\|\bar{x}_i\|_p \leq r^{(i)} + C\|\bar{w}_i\|_p, \quad (22)$$

where $C = \gamma_p(\mathcal{F})(\bar{\gamma} + 1)$ (using $\bar{\gamma}$ as an upper bound). We now analyze the total norm of $\mathbf{x} = (\bar{x}_0, \bar{x}_1, \dots)$.

Case: $1 \leq p < \infty$. Summing the p -th power of (22) over all windows i :

$$\begin{aligned} \|\mathbf{x}\|_p^p &= \|\bar{x}_0\|_p^p + \sum_{i=1}^{\infty} \|\bar{x}_i\|_p^p \\ &\leq (C_0\|\bar{w}_0\|_p)^p + \sum_{i=1}^{\infty} \left(r^{(i)} + C\|\bar{w}_i\|_p \right)^p \end{aligned} \quad (23)$$

Let $\tilde{\mathbf{w}} = (\|\bar{w}_1\|_p, \|\bar{w}_2\|_p, \dots)$. The sum in (23) is the p -norm of a sum of sequences in ℓ_p :

$$\begin{aligned} \|\mathbf{x}\|_p^p &\leq (C_0\|\bar{w}_0\|_p)^p + (\|\mathbf{r} + C\tilde{\mathbf{w}}\|_p)^p \\ &\leq (C_0\|\bar{w}_0\|_p)^p + (\|\mathbf{r}\|_p + C\|\tilde{\mathbf{w}}\|_p)^p \end{aligned}$$

²The system \mathcal{F} on this window has a non-zero initial state x_{t_i} and external inputs \bar{u}_i and \bar{w}_i , and since as established by the $w_0 = x_0$ convention, the unforced response to an initial state x_{t_i} is equivalent to the response to a disturbance impulse $\mathbf{w}' = (x_{t_i}, 0, \dots)$.

Using $(a + b)^p \leq 2^{p-1}(a^p + b^p)$, which holds for $p \in \mathbb{N}_{\geq 1}$ on the second term: $\|\mathbf{x}\|_p^p \leq (C_0\|\bar{w}_0\|_p)^p + 2^{p-1}(\|\mathbf{r}\|_p^p + C^p\|\tilde{\mathbf{w}}\|_p^p)$. The windows \bar{w}_i are disjoint. Let $\mathbf{w}' = (\|\bar{w}_0\|, \tilde{\mathbf{w}})$. Then $\|\mathbf{w}'\|_p^p = \sum_{i=0}^{\infty} \|\bar{w}_i\|_p^p = \|\mathbf{w}\|_p^p$. The total norm is bounded by:

$$\begin{aligned} \|\mathbf{x}\|_p^p &\leq 2^{p-1} \left((C_0\|\bar{w}_0\|_p)^p + \|\mathbf{r}\|_p^p + C^p\|\tilde{\mathbf{w}}\|_p^p \right) \\ &\leq 2^{p-1} \left(\max(C_0, C)^p \|\mathbf{w}\|_p^p + \|\mathbf{r}\|_p^p \right) \end{aligned}$$

Since $\mathbf{r}, \mathbf{w} \in \ell_p$, the right-hand side is finite, so we conclude that $\|\mathbf{x}\|_p < \infty$. The same arguments applied to (20) show $\|\mathbf{u}\|_p < \infty$. Thus, the system is ℓ_p -stable.

Case: $p = \infty$. The bound (22) (which also holds for $p = \infty$) becomes: $\|\bar{x}_i\|_{\infty} \leq r^{(i)} + C\|\bar{w}_i\|_{\infty}$. Taking the supremum over all $i \in \mathbb{N}_{\geq 1}$:

$$\sup_{i \in \mathbb{N}_{\geq 1}} \|\bar{x}_i\|_{\infty} \leq \sup_{i \in \mathbb{N}_{\geq 1}} r^{(i)} + \sup_{i \in \mathbb{N}_{\geq 1}} (C\|\bar{w}_i\|_{\infty})$$

The total state norm is $\|\mathbf{x}\|_{\infty} = \max(\|\bar{x}_0\|_{\infty}, \sup_{i \in \mathbb{N}_{\geq 1}} \|\bar{x}_i\|_{\infty})$. From (18), $\|\bar{x}_0\|_{\infty} \leq C_0\|\bar{w}_0\|_{\infty}$, and hence

$$\|\mathbf{x}\|_{\infty} \leq \max(C_0\|\bar{w}_0\|_{\infty}, \|\mathbf{r}\|_{\infty} + C \sup_{i \in \mathbb{N}_{\geq 1}} \|\bar{w}_i\|_{\infty}) \quad (24)$$

Since $\sup_{i \in \mathbb{N}_{\geq 0}} \|\bar{w}_i\|_{\infty} \leq \|\mathbf{w}\|_{\infty}$, and $\mathbf{r}, \mathbf{w} \in \ell_{\infty}$, the right hand side of (24) is finite. Thus, the system is ℓ_{∞} -stable.

Therefore, for any $p \in [1, \infty]$ and any disturbance sequence $\mathbf{w} \in \ell_p$, the corresponding closed-loop trajectories satisfy $\mathbf{x} \in \ell_p$ and $\mathbf{u} \in \ell_p$. Hence, the closed loop induced by the concatenated controller \mathcal{M} is ℓ_p -stable. ■

Next, we discuss how imposing additional conditions on \mathbf{r} can yield an input-to-state type bound for the resulting closed loop, on top of ℓ_p -stability.

Corollary 1 (ISS under time-scheduled persistent updates):

Consider Algorithm 1 with $p = \infty$ and fixed update times $t_i = i t_{\text{opt}}$, $i \in \mathbb{N}$, and assume that the conditions of Theorem 2 hold at every scheduled update. Suppose that

$$r^{(i)} = d^{(i)} \gamma(\mathcal{F})(\gamma(\mathcal{M}^{(0)}) + 1)|x_0| + g\|\mathbf{w}\|_{\infty}, \quad i \geq 1,$$

for some positive, strictly decreasing scalar sequence $\{d^{(i)}\}_{i \geq 0}$ such that $d^{(0)} \geq 1$ and $\lim_{i \rightarrow \infty} d^{(i)} = 0$, and for some constant $g > 0$. Then the resulting closed loop is Input-to-State Stable (ISS) [26]; that is, there exist functions $\beta \in \mathcal{KL}$ and $\alpha \in \mathcal{K}_{\infty}$ such that, for any initial condition x_0 and any bounded disturbance sequence $\mathbf{w} \in \ell_{\infty}$, the state trajectory satisfies

$$|x_t| \leq \beta(|x_0|, t) + \alpha(\|\mathbf{w}\|_{\infty}), \quad \forall t \in \mathbb{N}.$$

Proof: Let

$$D := \gamma(\mathcal{F})(\gamma(\mathcal{M}^{(0)}) + 1), \quad G := \gamma(\mathcal{F})(\bar{\gamma} + 1).$$

From the proof of Theorem 2, if $t \in [t_i, t_{i+1})$ with $i \geq 1$, then

$$|x_t| \leq r^{(i)} + G\|\mathbf{w}\|_{\infty} = D d^{(i)}|x_0| + (g + G)\|\mathbf{w}\|_{\infty}.$$

For $t \in [t_0, t_1)$, (18) gives

$$|x_t| \leq D|x_0| + G\|\mathbf{w}\|_{\infty} \leq D d^{(0)}|x_0| + (g + G)\|\mathbf{w}\|_{\infty},$$

since $d^{(0)} \geq 1$ and $g > 0$.

Now define a continuous, nonincreasing function $\bar{d} : [0, \infty) \rightarrow (0, \infty)$ as follows: set $\bar{d}(t) = d^{(0)}$ for $t \in [0, t_1]$, and for each $i \geq 1$ let \bar{d} be linear on $[t_i, t_{i+1}]$ with

$$\bar{d}(t_i) = d^{(i-1)}, \quad \bar{d}(t_{i+1}) = d^{(i)}.$$

Since $\{d^{(i)}\}$ is strictly decreasing and $t_i \rightarrow \infty$, the function \bar{d} is continuous, nonincreasing, and satisfies $\bar{d}(t) \rightarrow 0$ as $t \rightarrow \infty$. Moreover, for every $t \in [t_i, t_{i+1})$ one has $\bar{d}(t) \geq d^{(i)}$.

Therefore, for all $t \geq 0$,

$$\|x_t\| \leq D \bar{d}(t) \|x_0\| + (g + G) \|\mathbf{w}\|_\infty.$$

Define

$$\beta(s, t) := D \bar{d}(t) s, \quad \alpha(s) := (g + G)s.$$

Then $\alpha \in \mathcal{K}_\infty$. Also, for each fixed $t \geq 0$, $\beta(\cdot, t) \in \mathcal{K}$, and for each fixed $s \geq 0$, $\beta(s, \cdot)$ is continuous, nonincreasing, and converges to 0 as $t \rightarrow \infty$. Hence $\beta \in \mathcal{KL}$, and the claimed ISS estimate follows. ■

B. Implementation details

1) *Design of the sequence \mathbf{r}* : The sequence $\mathbf{r} = \{r^{(i)}\}_{i \geq 1}$ regulates how aggressive updates are allowed to be over time. The design of \mathbf{r} can leverage *a priori* information obtained through offline closed-loop simulations. Let $\tilde{x}_{0:T}$ be a nominal state trajectory generated under a representative disturbance and a baseline controller $\mathcal{M}^{(0)}$. We define a time-indexed budget profile

$$0 \leq r_t = \rho \|\tilde{x}_t\|, \quad t = 0, \dots, T,$$

with $\rho > 0$ a scaling factor. To obtain an infinite-horizon sequence compatible with Theorem 2, the profile is extended for $t > T$ by a nonnegative tail $\{r_t\}_{t > T}$ chosen so that $\{r_t\}_{t \geq 0} \in \ell_p$. For instance, when $1 \leq p < \infty$, one may append any summable tail, e.g. $r_t = r_T \eta^{t-T}$, $t > T$, for some $\eta \in (0, 1)$, while for $p = \infty$ any bounded continuation is admissible (and a decaying continuation can be used if one also wants the update budget to vanish asymptotically). Online, at each actual update time t_i , we set $r^{(i)} := r_{t_i}$.

In this way, large nominal deviations $\|\tilde{x}_t\|$ early in the system evolution yield larger $r^{(i)}$, allowing updates even for relatively large states (or higher controller gains), while smaller $\|\tilde{x}_t\|$ later in the system evolution yield smaller $r^{(i)}$, enforcing more conservative updates near the origin. This provides a simple, simulation-driven procedure to generate a sequence \mathbf{r} consistent with Theorem 2.

2) *Controller Updates through optimization*: The method for updating the controller depends significantly on the chosen update strategy and the available computational resources. Indeed, solving problem (8) can be computationally demanding. A practical way to reduce computational burden is to solve a simplified instance of (8) (shorter horizon H , fewer disturbance samples). Equivalently, one may warm-start from the previously found parameters θ and take a budgeted number of gradient steps during the sampling interval; the resulting inexact iterate can be deployed without risking destabilization. Indeed, *the proposed method ensures*

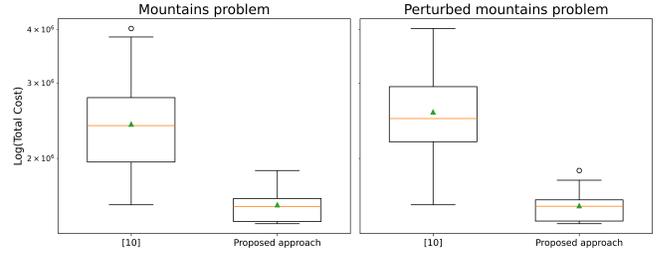


Fig. 3: Mountains problem - Total cost (log scale) over 50 runs: nominal scenario (center) and perturbed scenario with impulse disturbances (14) (right).

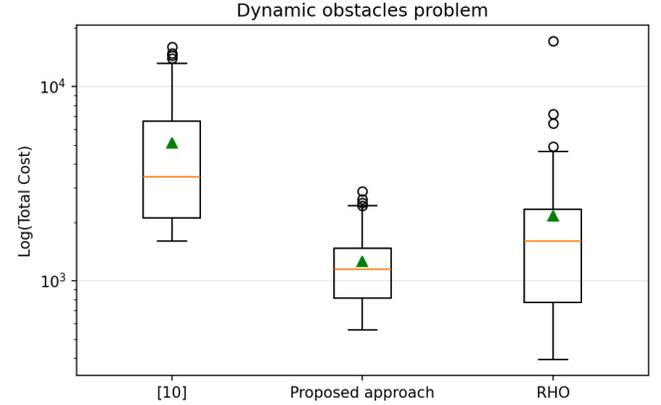


Fig. 4: Dynamic obstacles problem - Total cost (log scale) over 50 runs for the offline controller in [10], a receding-horizon open-loop (RHO) planner, and the proposed approach.

stability without requiring optimality and without requiring specific assumptions on the cost function, except that it must be differentiable (see Problem 1).

C. Numerical Example

Figures 3 and 4 report the boxplots of the total closed-loop cost for the mountains and dynamic-obstacles scenarios, respectively.

1) *Computation of the upper bound of the ℓ_2 -gain*: For the pre-stabilized error dynamics, we consider the least favorable slope of the friction term

$$G(v) = -b_1 v + b_2 \tanh(v), \quad \sigma_{\text{wc}} := -(b_1 - b_2),$$

which corresponds to the minimum-damping case since

$$-b_1 \leq \frac{\partial G}{\partial v} \leq -(b_1 - b_2).$$

Using this worst-case slope, we obtain a linear error model of the form

$$e_{t+1} = A_{\text{wc}} e_t + B_u u_t + w_{t+1}.$$

Let $\bar{B} := [B_u \ I]$ and $\xi_t := [u_t^\top \ w_{t+1}^\top]^\top$. We compute an upper bound on the ℓ_2 -gain of $\mathcal{F}_e : (\mathbf{u}, \mathbf{w}) \mapsto \mathbf{e}$ by finding

$P \succ 0$, $\eta > 0$, and $\rho > 0$ such that

$$\begin{bmatrix} A_{\text{wc}}^\top P A_{\text{wc}} - P + \eta I & A_{\text{wc}}^\top P \bar{B} \\ \bar{B}^\top P A_{\text{wc}} & \bar{B}^\top P \bar{B} - \rho I \end{bmatrix} \preceq 0. \quad (25)$$

Then, with $V(e) = e^\top P e$, (25) implies

$$V(e_{t+1}) - V(e_t) \leq -\eta \|e_t\|^2 + \rho (\|u_t\|^2 + \|w_{t+1}\|^2).$$

Summing over time and using the convention $w_0 = e_0$ yields

$$\eta \|e\|_2^2 \leq \lambda_{\max}(P) \|w_0\|^2 + \rho \|\mathbf{u}\|_2^2 + \rho \|w_{1:\infty}\|_2^2 \leq c (\|\mathbf{w}\|_2^2 + \|\mathbf{u}\|_2^2),$$

where $c := \max\{\lambda_{\max}(P), \rho\}$. Therefore,

$$\|e\|_2 \leq \hat{\gamma}(\mathcal{F}_e) (\|\mathbf{w}\|_2 + \|\mathbf{u}\|_2), \quad \hat{\gamma}(\mathcal{F}_e) := \max\left\{1, \sqrt{\frac{c}{\eta}}\right\}.$$