Digital Chemical Engineering 18 (2026) 100291

Contents lists available at ScienceDirect

Digital Chemical Engineering

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

Original article

Robust reinforcement learning for nonlinear process control with stability guarantees

Xiaodong Cui [a], Arthur Khodaverdian [a], Panagiotis D. Christofides [a,b,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA
[b] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

ARTICLE INFO

Keywords:
Reinforcement learning
Model predictive control
Lyapunov-based control
Offset-free model predictive control
Nonlinear systems
Chemical processes

ABSTRACT

Reinforcement learning (RL) offers a promising route to fast, nonlinear feedback control for complex process systems; however, its deployment is hindered by the lack of formal stability guarantees and sensitivity to model-plant mismatch under constraints. This paper proposes a stability and robustness-oriented RL-based control framework for nonlinear constrained processes by explicitly integrating Lyapunov-based decision rules into the RL closed loop and importing an offset-free approach from model predictive control (MPC). The RL policy is treated as a performance-seeking candidate controller that is supervised by a Lyapunov-certified fallback controller: at each sampling instant, the learned candidate input is evaluated against a Lyapunov condition, and any violating proposal is rejected in favor of a stabilizing backup input, yielding Lyapunov-certified practical stability under sample-and-hold implementation. To mitigate steady-state offsets and enhance robustness to disturbances and mismatches, the state available to the learning agent is augmented with online-estimated uncertainty/disturbance variables in the spirit of offset-free MPC, enabling the policy/value function to condition its decisions on the magnitude of uncertainty rather than overfitting nominal dynamics. The proposed architecture is demonstrated using two representative RL methods—an HJB-based value-critic approach and a TD3-based actor-critic approach—both deployed under the same Lyapunov-supervisory switching logic. Simulation studies on nonlinear chemical process control problems show that the proposed RL-based control framework preserves the low online computational cost while enforcing Lyapunov stability and improving robustness under disturbances, thereby advancing RL toward reliable process control deployment.

1. Introduction

Advanced control has played a central role in improving safety, efficiency, and product quality in modern process systems. Among existing approaches, model predictive control (MPC) has become a mature and widely deployed paradigm due to its explicit handling of multivariable interactions, input and output constraints, and economic objectives (Qin and Badgwell, 2003). In particular, nonlinear MPC and its variants have demonstrated strong performance in complex chemical and energy processes where nonlinearities and constraints are unavoidable (Rawlings et al., 2020). Despite this success, MPC requires solving a constrained finite-horizon optimal control problem online at each sampling instant, which can be computationally demanding for nonlinear models, long horizons, and large-scale architectures, potentially limiting achievable sampling rates in fast or safety-critical applications (Rawlings et al., 2020). A significant body of work in the process control community has therefore focused on scalable and stability-oriented MPC formulations, including Lyapunov-based MPC designs that retain formal stability properties under network imperfections such as sensor data losses (Muñoz de la Peña and Christofides, 2008), and distributed MPC frameworks for nonlinear process systems subject to asynchronous and delayed measurements (Liu et al., 2010; Christofides et al., 2013). Economic MPC further expands the scope by directly optimizing economic performance, but may exacerbate online computational requirements due to nonquadratic objectives and nonlinear constraints (Ellis et al., 2014).

These considerations motivate growing interest in computationally efficient advanced control strategies, where most of the computational burden is shifted offline and online implementation reduces to evaluating an explicit feedback law. Reinforcement learning (RL) naturally fits this paradigm: an RL controller can be trained offline (from data and/or simulators) to produce a policy, often parameterized by function approximators such as neural networks, that can be evaluated online with minimal overhead (Sutton and Barto, 2018; Lillicrap et al., 2015; Faria et al., 2022; Nian et al., 2020). It is also noted that, for complex

* Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.dche.2026.100291
Received 16 January 2026; Received in revised form 30 January 2026; Accepted 30 January 2026
Available online 3 February 2026

industrial processes, RL typically requires a large number of interaction samples and therefore relies heavily on representative simulators for offline training. Moreover, when MPC can be solved to (near-)global optimality using the same underlying model, an RL-based controller trained on that model is not expected to outperform such an MPC benchmark (Nian et al., 2020). Beyond computational efficiency, RL offers additional advantages for nonlinear process control, including reduced reliance on high-fidelity first-principles models, the ability to optimize long-horizon and nonstandard objectives directly from interaction data, and the capability to leverage historical plant or simulation data via offline RL to improve performance without repeated online trial-and-error (Sutton and Barto, 2018; Levine et al., 2020; Faria et al., 2022).

Despite these appealing features, deploying RL controllers in safety-critical process systems remains challenging. A central issue is robustness: learned policies can be sensitive to distribution shift caused by model-plant mismatch, unmodeled disturbances, measurement noise, and operating-condition changes, and performance may degrade when the closed-loop system visits regions insufficiently represented in training data (Dulac-Arnold et al., 2019; Morimoto and Doya, 2005). While robust RL and robust Markov decision process formulations provide principled ways to reason about uncertainty, they often introduce conservatism and can be difficult to calibrate for nonlinear continuous-time plants with hard operating constraints (Nilim and El Ghaoui, 2005; Iyengar, 2005). Equally important, standard RL objectives do not explicitly enforce closed-loop stability or constraint satisfaction, and safe exploration is inherently difficult when constraint violations are unacceptable; consequently, providing verifiable stability/safety guarantees for neural-network policies remains an active research area and is a key reason why real-plant RL deployments are still relatively limited in the process industries (Garcıa and Fernández, 2015; Berkenkamp et al., 2017; Chow et al., 2018; Dulac-Arnold et al., 2019).

A promising route to address these limitations is to leverage established advanced control theory as a backbone for RL design and deployment. In particular, Lyapunov-based MPC provides constructive mechanisms to ensure stability and constraint satisfaction for nonlinear constrained systems (Mhaskar et al., 2006), including extensions that retain stability properties under communication network imperfections such as measurement/data losses (Muñoz de la Peña and Christofides, 2008). Likewise, offset-free MPC formulations incorporate disturbance/mismatch estimation into the prediction model to mitigate steady-state offsets and improve robustness in tracking problems (Pannocchia, 2015; Pannocchia et al., 2015). These MPC concepts can be embedded into RL in multiple ways (e.g., as safety filters/shields, training regularizers, or offset-free state augmentation), and recent work has begun to demonstrate practically implementable RL controllers by explicitly leveraging offset-free MPC structures and stability-oriented MPC ideas (Hassanpour et al., 2024a,b; Khodaverdian et al., 2025a). Nevertheless, robustness-oriented stability analysis for RL closed loops under general uncertainty/disturbances remains comparatively underdeveloped, leaving an important gap between emerging stability-guaranteed RL designs and broad deployment in realistic process settings (Dulac-Arnold et al., 2019; Garcıa and Fernández, 2015).

Therefore, in this paper, we propose a stability and robustness-oriented RL framework for nonlinear constrained process systems by explicitly integrating Lyapunov-based decision rules into the RL closed loop and importing the offset-free method from MPC. The key idea is to treat the RL policy as a performance-seeking candidate controller that is supervised by a Lyapunov-certified fallback controller. Specifically, at each sampling instant, we first compute the RL action and check the stability condition; if the RL decision fails this condition, we switch to the fallback controller that is designed to satisfy the Lyapunov requirement so that the implemented control action always preserves the desired stability property. In parallel, to mitigate steady-state offsets and sensitivity to model-plant mismatch, we augment the learning state with disturbance/mismatch variables (estimated online) in the spirit of offset-free MPC, enabling the RL policy to recognize the magnitude of uncertainty and compensate for it rather than overfitting to nominal dynamics. We demonstrate the application of this architecture using two representative RL methods, namely an HJB-based value-critic RL and a TD3-based actor-critic RL, by conditioning the learned value/policy on the augmented variables and deploying them under the same Lyapunov-supervisory switching logic. The resulting approach retains the low online computational cost of neural-network feedback while providing a systematic mechanism to enforce Lyapunov stability and improve robustness under disturbances, thereby moving RL closer to reliable deployment in process control applications.

2. Preliminaries

2.1. Notation

The transpose of a vector x, the set of real numbers, set difference, functions, and piecewise-constant functions with period Δ are denoted by $x^\top$, $\mathbb{R}$, $\Omega_1 \backslash \Omega_2$, $f(\cdot)$, and $S(\Delta)$ respectively, where both f and S are arbitrary notations. The initial instance of time (i.e., where $t = 0$) is denoted t_0, whereas arbitrary reference instances of time are denoted t_k. A function $\alpha : [0, a) \to [0, \infty)$ is said to be of class-$\mathcal{K}$ if it is continuous, strictly increasing, and satisfies $\alpha(0) = 0$.

2.2. Class of systems

This paper considers nonlinear first-order ordinary differential equation (ODE) systems of the form:

$$\dot{x} = F(x, u) \tag{1}$$

The state vector $x = \left[x_1, x_2, \ldots, x_n\right]^\top \in \mathbb{R}^n$ collects all relevant process state variables and is assumed to be measured at fixed sampling instants t_k, as is standard for sampled-data state feedback control. The control input vector $u = \left[u_1, u_2, \ldots, u_m\right]^\top \in \mathbb{R}^m$ represents all control actions applied to the process. In practice, actuator limitations impose hard bounds on the inputs. The set of admissible control actions is defined as follows:

$$U := \left\{ u \in \mathbb{R}^m \middle| \begin{array}{c} u = \left[u_1, u_2, \ldots, u_m\right]^\top \\ u_{i,\min} \le u_i \le u_{i,\max} \\ \forall\ i = 1, 2, \ldots, m \end{array} \right\} \subset \mathbb{R}^m \tag{2}$$

We employ the deviation-variable form of the system so that the origin is an equilibrium of the open-loop nominal model, i.e., $F(0, 0) = 0$ without loss of generality. We further assume that the state evolves in a domain $D \subset \mathbb{R}^n$ containing the origin (e.g., the operating region of interest) and that $F(\cdot, \cdot)$ is sufficiently smooth nonlinear vector function on $D \times U$.

In practice, the true process may be affected by unmodeled dynamics, parametric mismatch, and unknown disturbances. We represent these effects by the perturbed system

$$\dot{x} = F(x, u) + W(x, t) \tag{3}$$

where $W : D \times \mathbb{R} \to \mathbb{R}^n$ denotes an unknown disturbance/model-plant mismatch term that may be time-varying. We assume that W belongs to a known bounded disturbance set and is uniformly bounded in magnitude by a constant $W_{\max} > 0$ over the region of interest:

$$\mathcal{W} := \left\{ W(\cdot, \cdot) \,||\, |W(x, t)| \le W_{\max} \quad \forall (x, t) \in D \times \mathbb{R} \right\} \tag{4}$$

Unless otherwise stated, all controller design and stability analysis in this paper are carried out with respect to the nominal model Eq. (1); the perturbed dynamics Eq. (3) are introduced only to represent the actual process behavior and the associated model-plant mismatch.

2.3. Stabilizability assumption

The core assumptions that ensure stabilizability are collectively referred to as the stabilizability assumption. The stabilizability assumption consists of two main assumptions, stated with respect to the nominal model Eq. (1) (i.e., without explicitly accounting for the unknown disturbance term $W(x,t)$ in Eq. (3)). The first is that we assume the existence of a sufficiently smooth explicit feedback control law that renders the origin of the nominal system Eq. (1) exponentially stable. This controller is referred to as the reference stabilizing controller, or reference controller, as a shorthand, and is of the form:

$$\boldsymbol{\Phi} : \mathbb{R}^n \to U \tag{5}$$

$$u(x) = \boldsymbol{\Phi}(x) \tag{6}$$

The second is the assumption that there exists a sufficiently smooth Lyapunov function $V(x)$ defined on a region where the associated stabilizing controller produces admissible inputs, i.e., $u = \boldsymbol{\Phi}(x) \in U$, which, when applied to the closed-loop nominal system under the reference controller, satisfies the following inequalities:

$$c_1 |x|^2 \leq V(x) \leq c_2 |x|^2 \tag{7a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \boldsymbol{\Phi}(x)) \leq -c_3 |x|^2 \tag{7b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4 |x| \tag{7c}$$

$$c_i > 0 \ \forall \ i \in \{1,2,3,4\} \tag{7d}$$

for all x in an open neighborhood of the origin, denoted D.

The remaining parts of the stabilizability assumption are derived from the sufficiently smooth assumption for the system dynamics mentioned earlier. This implies Lipschitz continuity for $V(x)$, $\boldsymbol{\Phi}(x)$, and $F(x, \boldsymbol{\Phi}(x))$. Additionally, because $\boldsymbol{\Phi}(x)$ is bounded (due to the presence of constraints on the control action since $u \in U$), when x is bounded within a level set $\Omega_\rho := \{x \mid V(x) \leq \rho\} \subset D$, we have that $F(x, \boldsymbol{\Phi}(x))$ is bounded for all $x \in \Omega_\rho$ and $u \in U$. Also, we note that the product of two continuously differentiable functions yields a function that is at least continuously differentiable. Thus, the stabilizability assumption implies the existence of positive constants M_F, L_x, L'_x that ensure, for all $x, x' \in \Omega_\rho$ and $u \in U$, that the following inequalities are satisfied:

$$\left| F(x', u) - F(x, u) \right| \leq L_x |x - x'| \tag{8a}$$

$$|F(x, u)| \leq M_F \tag{8b}$$

$$\left| \frac{\partial V(x)}{\partial x} F(x, u) - \frac{\partial V(x')}{\partial x} F(x', u) \right| \leq L'_x |x - x'| \tag{8c}$$

2.4. Lyapunov-based model predictive control

The stability assumption can be applied to MPC to yield the Lyapunov-based MPC (LMPC) that solves for optimal control while ensuring closed-loop stability (Mhaskar et al., 2006).

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_k + N\Delta} L(\tilde{x}(t), u(t)) \, \mathrm{d}t \tag{9a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), u(t)) \tag{9b}$$

$$u \in U, \ \forall \ t \in [t_k, t_k + N\Delta) \tag{9c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{9d}$$

$$\dot{V}(\tilde{x}(t_k), u(t_k)) \leq \dot{V}(\tilde{x}(t_k), \boldsymbol{\Phi}(\tilde{x}(t_k))) \tag{9e}$$

In this formulation, the optimization takes place over a horizon of length $N\Delta$, with Δ denoting the controller's sampling period and N the horizon's sampling step count. This formulation uses sample-and-hold control, as continuous-time control is infeasible for real-world processes. For simplicity, the sampling interval for both the controllers and state measurements is treated as equivalent. Eq. (9a) denotes an arbitrary cost as a function of the control actions and estimated states over the horizon. Eq. (9b) represents the process dynamics, which are used for numerical integration during optimization to predict how the states of the closed-loop system evolve over the horizon. Eq. (9d) enforces an initialization step where the sensor readings are used as a ground truth initial state, and Eq. (9c) enforces the control bounds. Eq. (9e) denotes the stability constraint; the implementation of the stabilizability assumption. This constraint ensures that the system is at least as stabilizing as the reference controller. An alternative form is provided:

$$\dot{V}(\tilde{x}(t_k), u(t_k)) \leq -\alpha V(\tilde{x}(t_k)) \tag{10}$$

This formulation uses the properties of the Lyapunov function from Section 2.3 as opposed to using the reference controller. Using a positive constant α to control the strength of this constraint, this formulation allows for stability guarantees without needing to directly apply the reference controller.

Remark 1. LMPC does not pose constraints on the form of the cost function. Although this paper will use a quadratic cost function, other formulations are supported. Economic MPC is one such modification that can enable enhanced cost-efficiency of processes in a manner that supports time-varying economics (Ellis et al., 2014; Khodaverdian et al., 2025b).

Remark 2. Eq. (9e) is only applied at t_k because this formulation is a receding horizon LMPC, where only the first control input from the solution is applied. After applying the first solution for one sampling interval, the LMPC problem is re-solved. This approach relaxes the constraints of the optimization problem, allowing for faster solutions, but comes at the cost of marginally reduced accuracy of the cost-optimal trajectory.

Remark 3. Consider an LMPC formulation that satisfies the design above. We consider two cases of this LMPC: one with a long horizon and one with a short horizon. The long-horizon case is used purely for reference of what the truly optimal behavior would be (MPC optimal control action calculation improves with increased horizon length), as this case would take longer to calculate than the sampling interval, thereby making it infeasible for real-time control. The short-horizon case is a suboptimal solution relative to the long-horizon LMPC that is, however, faster to solve in real-time. This short-horizon LMPC can thus be used as a fallback controller as a means to enforce the stability guarantees for the closed-loop system at the expense of poor cost optimality.

2.5. Offset-free Lyapunov-based model predictive control

The LMPC formulation in Eq. (9) is based on the nominal prediction model Eq. (1), while the true process may evolve according to Eq. (3) due to unknown disturbances/model-plant mismatch. Such mismatch can lead to steady-state offsets when using a purely nominal MPC. To mitigate this issue, we employ an offset-free MPC structure that augments the nominal model with additional disturbance states, estimates them online via an extended Luenberger observer, and then uses these estimates in the prediction model within the MPC optimization. The augmented offset-free prediction model is as follows:

$$\dot{\tilde{x}}(t) = F(\tilde{x}(t), u(t)) + G_\theta \tilde{\theta}(t) \tag{11a}$$

$$\dot{\tilde{\theta}}(t) = 0 \tag{11b}$$

where G_θ represents the gain matrix of the augmented term and $\tilde{\theta}(t)$ is treated as constant over the prediction horizon. By defining $\chi := \left[\tilde{x}^\top, \ \tilde{\theta}^\top\right]^\top$, Eq. (11) can be compactly written as:

$$\dot{\chi}(t) = \bar{F}(\chi(t), u(t)) := \begin{bmatrix} F(\tilde{x}(t), u(t)) + G_\theta \tilde{\theta}(t) \\ 0 \end{bmatrix} \tag{12}$$

We assume that the state $x(t)$ is available as a continuous-time measurement. Over the same interval, we employ the extended Luenberger observer:

$$\dot{\hat{x}}(t) = F(\hat{x}(t), u(t)) + G_\theta \hat{\theta}(t) + K_x [x(t) - \hat{x}(t)] \tag{13a}$$

$$\dot{\hat{\theta}}(t) = K_\theta [x(t) - \hat{x}(t)] \tag{13b}$$

where K_x and K_θ are constant observer gain matrices. The controller uses the real $x(t_k)$ and $\hat{\theta}(t_k)$ at time t_k. In deviation-variable coordinates, we take the tracking setpoint as $x_{sp} = 0$; under mismatch/disturbance, the corresponding steady-state input generally shifts and must be recomputed online using the disturbance estimate. Specifically, the offset-free steady-state input $u_{sp}(t_k)$ is defined as any admissible solution of the equilibrium condition for the offset-free model:

$$0 = F\left(0, u_{sp}\left(t_k\right)\right) + G_\theta \hat{\theta}\left(t_k\right), \qquad u_{sp}(t_k) \in U \tag{14}$$

which is a set of nonlinear algebraic equations in $u_{sp}(t_k)$. In implementation, Eq. (14) can be solved by a standard root-finding method (e.g., Newton method), warm-started from the previously computed $u_{sp}(t_{k-1})$ and followed by projection onto U if needed. To ensure well-posedness, we assume that for each admissible $\hat{\theta}(t_k)$ in a neighborhood of interest there exists a solution $u_{sp}(t_k) \in U$ to Eq. (14) and that the input Jacobian of the equilibrium map is nonsingular at the solution, i.e.,

$$\det\left(\frac{\partial}{\partial u}\left[F(0,u) + G_\theta \hat{\theta}\left(t_k\right)\right]\right)\Big|_{u=u_{sp}(t_k)} \neq 0 \tag{15}$$

so that a locally unique and smoothly varying mapping $u_{sp} = \Psi(\hat{\theta})$ exists (by the implicit function theorem). Using the estimates, the offset-free MPC solves the following optimization at each sampling instant:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_k + N\Delta} L(\tilde{x}(t), u(t))\, \mathrm{d}t \tag{16a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), u(t)) + G_\theta \tilde{\theta}\left(t_k\right), \qquad \forall t \in [t_k, t_k + N\Delta) \tag{16b}$$

$$\dot{\tilde{\theta}}(t) = 0, \qquad \forall t \in [t_k, t_k + N\Delta) \tag{16c}$$

$$u \in U, \qquad \forall t \in [t_k, t_k + N\Delta) \tag{16d}$$

$$\tilde{x}(t_k) = x(t_k), \ \tilde{\theta}(t_k) = \hat{\theta}(t_k) \tag{16e}$$

In Eq. (16), $\tilde{x}(t)$ denotes the nominal predicted state used within the optimizer, whereas the measured state $x(t)$ evolves according to the true process Eq. (3). The disturbance estimate $\hat{\theta}(t_k)$ provides an offset-free correction to the prediction model, enabling improved tracking and reduced steady-state offsets under bounded disturbances and model-plant mismatch. Although Eq. (16) addresses steady-state offsets, it does not explicitly enforce the Lyapunov decrease condition used in Eq. (9). We therefore define an offset-free Lyapunov-based MPC (OF-LMPC) by augmenting Eq. (16) with a Lyapunov constraint that is activated only in an outer region. Let $\Omega_\rho := \{x \mid V(x) \leq \rho\} \subset D$ and fix $0 < \rho_{sw} < \rho$, defining $\Omega_{\rho_{sw}} := \{x \mid V(x) \leq \rho_{sw}\}$. Define the gate threshold

$$S(x) \in \left\{ \dot{V}\left(x, \Phi(x)\right), \ -\alpha V(x) \right\} \tag{17}$$

consistent with whether the reference-controller form in Eq. (9e) or the αV form in Eq. (10) is adopted. The offset-free LMPC optimization is of the form:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_k + N\Delta} L(\tilde{x}(t), u(t))\, \mathrm{d}t \tag{18a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), u(t)) + G_\theta \tilde{\theta}\left(t_k\right), \qquad \forall t \in [t_k, t_k + N\Delta) \tag{18b}$$

$$\dot{\tilde{\theta}}(t) = 0, \qquad \forall t \in [t_k, t_k + N\Delta) \tag{18c}$$

$$u \in U, \qquad \forall t \in [t_k, t_k + N\Delta) \tag{18d}$$

$$\tilde{x}(t_k) = x(t_k), \ \tilde{\theta}(t_k) = \hat{\theta}(t_k) \tag{18e}$$

$$\dot{V}\left(x\left(t_k\right), u\left(t_k\right)\right) \leq S\left(x\left(t_k\right)\right), \qquad \forall\, x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{sw}} \tag{18f}$$

Because $\tilde{x}(t_k) = x(t_k)$, the Lyapunov constraint Eq. (18f) is evaluated at the current estimated state. Enforcing Eq. (18f) only for $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{sw}}$ reduces conservativeness inside $\Omega_{\rho_{sw}}$ while retaining the Lyapunov-based stabilizing mechanism in the outer region; in practice, the steady-state target $u_{sp}(t_k)$ from Eq. (14) is updated online using $\hat{\theta}(t_k)$ and can be used wherever a reference input is needed (e.g., for defining tracking objectives or for constructing the reference controller $\Phi(\cdot)$ around the current offset-free equilibrium).

Remark 4. If Eq. (15) does not hold globally, then the setpoint-tracking costs in Eq. (16a) and (18a) can be modified to remove any dependence on the input-deviation term (i.e., the penalty on $u - u_{sp}$). In particular, one may adopt an alternative offset-free tracking cost that penalizes only state/output deviations (or other suitable terms) while retaining offset-free performance (Wallace et al., 2016).

Remark 5. In this study, we assume that the process states are continuously measurable and available to the controller. Therefore, the controller uses the measured state directly, and the offset-free observer is used only to estimate the mismatch term $\hat{\theta}$. In practice, state measurements may be sampled at a finite rate or may not be available for all states. In such cases, the offset-free observer in Eq. (13) can be used as a state estimator, and the offset-free LMPC optimization in Eq. (16) and (18) can be initialized using the estimated state, i.e., by replacing the initialization condition with $\tilde{x}(t_k) = \hat{x}(t_k)$.

Remark 6. The Lyapunov constraint Eq. (18f) is evaluated using $\dot{V}(x,u)$ computed from the nominal model Eq. (1), rather than from the true process dynamics Eq. (3). Under unknown disturbance/model-plant mismatch, this nominal $\dot{V}$ can be an imperfect surrogate of the true Lyapunov derivative. If enforced everywhere, the resulting constraint biases the optimizer toward decisions that are misleading for the true closed-loop behavior, especially near the origin where the mismatch can dominate the nominal decrease prediction. For this reason, we activate the Lyapunov constraint only in the outer region $\Omega_\rho \backslash \Omega_{\rho_{sw}}$, where its role is primarily to provide a stabilizing mechanism that drives the state into the inner set $\Omega_{\rho_{sw}}$. Once the state enters $\Omega_{\rho_{sw}}$, the constraint is removed to avoid relying on a potentially inaccurate nominal decrease certificate. In this inner region, the offset-free structure and the MPC objective are used to achieve improved tracking performance while reducing steady-state offsets.

Remark 7. We set $\dot{\tilde{\theta}} = 0$ over the prediction horizon, i.e., $\tilde{\theta}$ is held constant at its current estimate. This choice is primarily practical: future real-time process measurements are unavailable beyond t_k, so the controller cannot reliably predict the evolution of the mismatch within the horizon. This constant-$\tilde{\theta}$ assumption is widely adopted in offset-free MPC implementations (see Wallace et al., 2016; Pannocchia, 2015; Pannocchia et al., 2015; Hassanpour et al., 2024a).

Remark 8. The gain matrix of the augmented term G_θ is a tuning parameter. In practice, it is selected (offline) to reduce the overall estimation error and to ensure convergence of the observer error to a bounded neighborhood by satisfying the conditions discussed in Section 4.2.

2.6. Reinforcement learning

Reinforcement learning (RL) studies how an *agent* selects actions using a policy π to interact with an *environment* so as to maximize cumulative reward. At each sampling instant, the agent observes the state s, applies an action $a = \pi(s)$, receives a reward $r(s,a)$, and transitions to a successor state s'. Collected transitions are commonly stored in a replay buffer

$$\mathcal{D} := \left\{\left(s_i, a_i, r_i, s_i'\right)\right\}_{i=1}^{N} \tag{19}$$

which is used to train function approximators in an off-policy manner.

A key concept is the value function, which quantifies long-term performance under a policy. The state-value function satisfies the Bellman equation

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma V^{\pi}(s') \tag{20}$$

where $0 < \gamma < 1$. Equivalently, many RL algorithms learn the action-value function $Q(s,a)$, which evaluates state–action pairs and supports policy improvement. In discrete settings, classical updates include the on-policy SARSA and off-policy Q-learning equations,

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma Q(s', a') - Q(s,a) \right] \tag{21}$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s,a) \right] \tag{22}$$

with learning rate $\alpha > 0$. For continuous-state and continuous-action control problems, modern actor-critic methods parameterize both a policy (actor) and a value function (critic), enabling direct policy optimization while leveraging replay data for sample-efficient learning.

3. Robust and stable reinforcement learning design and implementation

This section presents the proposed reinforcement learning (RL) design and implementation framework for nonlinear process control under unknown disturbances and model-plant mismatch. The key idea is to integrate an offset-free augmentation into the RL formulation: the extended Luenberger observer Eq. (13) estimates a structured mismatch state $\hat{\theta}$ online, and the controller is conditioned on the corresponding augmented state $\hat{\chi} := \left[x^{\top}, \hat{\theta}^{\top}\right]^{\top}$. Although the RL policy uses the measured state $x(t_k)$, $\hat{\theta}(t_k)$ estimates disturbance/mismatch from the observer and helps the policy adjust its action under mismatch. This enables the learned policy to adapt its action to the inferred disturbance realization and mitigates steady-state offsets that commonly arise when learning and deploying based on nominal models.

Building on this offset-free augmentation, we develop two complementary RL realizations: (i) a robust Hamilton–Jacobi-Bellman (HJB)-based value-critic approach that learns a differentiable value function over $\hat{\chi}$ and induces a feedback policy via the HJB stationarity condition, and (ii) a robust TD3-based actor-critic approach that is trained and deployed entirely in closed loop, using the augmented state $\hat{\chi}$ (through the observer estimate $\hat{\theta}$) to account for disturbance realizations during online interaction. To ensure closed-loop stability and safe deployment, the resulting RL actions are further protected by a stabilizing shield composed of provably stabilizing controllers, which overrides the RL decision whenever a predefined stability/safety condition is violated. These components yield a robust and stable RL framework suitable for real-time implementation.

3.1. Robust hamilton–Jacobi-Bellman-based RL design

Hamilton–Jacobi-Bellman (HJB)-based reinforcement learning formulates the control problem as solving a stationary HJB optimality equation (Zhu et al., 2025; Wang et al., 2025). To handle unknown disturbances/model-plant mismatch in a manner consistent with the offset-free framework in Section 2.5, we perform the value-based design on the observer-conditioned augmented state $\hat{\chi} := \left[x^{\top}, \hat{\theta}^{\top}\right]^{\top}$, where $\hat{\theta}(t_k)$ is the disturbance estimate provided by the extended Luenberger observer Eq. (13).

In this framework, the optimal value function is defined using the augmented state as follows:

$$V^{\star}(\hat{\chi}) = \min_{u(\cdot)} \left\{ \int_t^{\infty} r(\hat{\chi}(\tau), u(\tau))\, d\tau \right\} \tag{23}$$

and it satisfies the stationary HJB condition

$$\min_u H\left(\hat{\chi}, u, V^{\star}\right) = \min_u \left\{ r(\hat{\chi}, u) + \frac{\partial V^{\star}}{\partial \hat{\chi}} \bar{F}(\hat{\chi}, u) \right\} = 0 \tag{24}$$

Consequently, the optimal policy is the action that minimizes this Hamiltonian at each augmented state:

$$\pi^{\star}(\hat{\chi}) = \arg\min_u H\left(\hat{\chi}, u, V^{\star}\right) \tag{25}$$

In closed-loop implementation, the policy is evaluated using $\hat{\chi}(t_k) := \left[x(t_k)^{\top}, \hat{\theta}(t_k)^{\top}\right]^{\top}$, where $\hat{\theta}(t_k)$ is obtained from Eq. (13).

To ensure a fair comparison between LMPC and RL, the instantaneous cost r in the RL formulation is chosen to match the quadratic stage cost used in LMPC,

$$r(\hat{\chi}, u) = L(x, u) = x^{\top} W_x x + u^{\top} W_u u \tag{26}$$

so that the Hamiltonian can be written as

$$H\left(\hat{\chi}, u, V^{\star}\right) = x^{\top} W_x x + u^{\top} W_u u + \frac{\partial V^{\star}}{\partial \hat{\chi}} \bar{F}(\hat{\chi}, u) \tag{27}$$

When the augmented dynamics are input-affine,

$$\bar{F}(\hat{\chi}, u) = \bar{f}(\hat{\chi}) + \bar{g}(\hat{\chi})\, u \tag{28}$$

the HJB condition Eq. (24) can be solved under the stationarity conditions (Lewis et al., 2012) to yield the feedback law:

$$\pi^{\star}(\hat{\chi}) = -\frac{1}{2} W_u^{-1} \bar{g}(\hat{\chi})^{\top} \frac{\partial V^{\star}(\hat{\chi})}{\partial \hat{\chi}} \tag{29}$$

This expression replaces the nominal state with the observer-conditioned augmented state and uses the corresponding input matrix $\bar{g}(\hat{\chi})$ induced by the offset-free augmentation.

Since the exact optimal value function $V^{\star}(\hat{\chi})$ is not available in closed form, we introduce a differentiable critic network $V_w(\hat{\chi})$ parameterized by neural network weights w. The critic is trained to minimize the mean-squared residual of the augmented HJB equation evaluated at sampled augmented states $\{\hat{\chi}(t_i)\}_{i=1}^{N}$. Specifically, the training objective and the weight update are given by

$$\mathcal{L}(w) = \frac{1}{N} \sum_{i=1}^{N} \left(r\left(\hat{\chi}(t_i), \pi_w(\hat{\chi}(t_i))\right) + \left(\frac{\partial V_w}{\partial \hat{\chi}}\right)(\hat{\chi}(t_i))\, \bar{F}\left(\hat{\chi}(t_i), \pi_w(\hat{\chi}(t_i))\right) \right)^2 \tag{30a}$$

$$w \leftarrow w - \alpha_w \nabla_w \mathcal{L}(w) \tag{30b}$$

where $\mathcal{L}(w)$ denotes the loss function and α_w is the learning rate that determines the step size of gradient descent.

Finally, by embedding the disturbance/mismatch information into the augmented state $\hat{\chi}$ through the online estimate $\hat{\theta}(t_k)$ provided by Eq. (13), the resulting HJB-RL policy becomes explicitly conditioned on the disturbance realization inferred from measurements. As a result, the learned critic $V_w(\hat{\chi})$ and the induced policy $\pi(\hat{\chi})$ adapt the control action according to $\hat{\theta}$, providing offset-free correction and mitigating steady-state offsets under bounded model-plant mismatch. In this sense, the observer-conditioned augmented-state formulation upgrades a nominal HJB-RL design into a *robust HJB-RL* methodology that systematically accounts for structured disturbances via online disturbance estimation and feedback on $\hat{\chi}$. Algorithm 1 summarizes this proposed training procedure, where ϵ_k denotes an exploration-noise term (e.g., Gaussian noise) whose magnitude can be scheduled/decayed and bounded, and clip$(\cdot)$ enforces the admissible input set U by saturating the tentative action to the componentwise input limits. The replay buffer D stores transition data (s_k, a_k, r_k, s_{k+1}). Finally, `update_every` specifies the number of environment steps between two consecutive critic-parameter updates, and N is the mini-batch size used when sampling from D.

Remark 9. The optimal control policy obtained from the HJB condition does not explicitly account for stability constraints. To enhance the accuracy of the approximated optimal value function, the neural network performance could be further improved by incorporating stability criteria in the training process; however, this aspect is not within the scope of the present study.

Algorithm 1 Robust HJB-RL training with offset-free augmented state

Initialize critic $V_w(\hat{\chi})$ with random parameters w
Initialize replay buffer $\mathcal{D}$
Initialize process and observer states, set iteration counter $k \leftarrow 0$
while not converged **do**
  Update observer Eq. (13) using measurements and obtain $\hat{\theta}(t_k)$
  Form augmented estimate $\hat{\chi}(t_k) \leftarrow \left[x^\top(t_k),\ \hat{\theta}^\top(t_k)\right]^\top$
  Select action with exploration: $u_k \leftarrow \text{clip}\left(\pi_w(\hat{\chi}(t_k)) + \epsilon_k\right)$
  Apply u_k to the process and obtain next measurement at t_{k+1}
  Update observer and form $\hat{\chi}(t_{k+1})$
  Store $(\hat{\chi}(t_k), u_k, \hat{\chi}(t_{k+1}))$ in $\mathcal{D}$
  if $k \mod \texttt{update\_every} = 0$ **then**
    Sample mini-batch from $\mathcal{D}$
    Compute HJB residuals: $\mathcal{R}_j(w) = r(\hat{\chi}_j, \pi_w(\hat{\chi}_j)) + \nabla_{\hat{\chi}} V_w(\hat{\chi}_j)^\top \bar{F}\left(\hat{\chi}_j, \pi_w(\hat{\chi}_j)\right)$
    Update w by minimizing $\frac{1}{N}\sum_{j=1}^{N} \mathcal{R}_j(w)^2$
  end if
  $k \leftarrow k+1$
end while

3.2. Robust TD3-RL design

TD3-RL, combined with an offset-free structure, has been explored as a candidate robust RL design. This approach was first proposed by Hassanpour et al. (2024a,b) and was reported to outperform offset-free MPC in the studied cases; however, the method does not provide an explicit closed-loop stability guarantee, and the potential reduction in online computational time relative to solving MPC optimizations is not explicitly highlighted or quantified. Therefore, in this study, we design the robust TD3-RL based on the offset-free TD3-RL proposed by Hassanpour et al. (2024a,b) and further modify it with the shield layer to guarantee the closed-loop stability.

TD3-RL (Twin Delayed Deep Deterministic Policy Gradient) is an off-policy actor-critic reinforcement learning algorithm designed for continuous-action control problems, and it is widely used in applications such as robotics and process control where the manipulated inputs are continuous-valued. Compared with DDPG, TD3 improves training stability and performance through three key modifications: (i) twin critic networks are used and the minimum of their estimates is taken to mitigate Q-value overestimation, (ii) the policy (actor) is updated less frequently than the critics to reduce the impact of policy updates on volatile value function estimates, and (iii) target policy smoothing is employed by adding clipped noise to the target action to suppress exploitation of function-approximation errors. By leveraging replay buffers and target networks, TD3 typically achieves improved sample efficiency in continuous control tasks.

We first generate an offline dataset by running an offset-free LMPC in closed loop under unknown disturbances, where an offset-free observer is executed online to estimate the mismatch/disturbance state $\hat{\theta}$ and the LMPC incorporates this estimate in its prediction model to mitigate steady-state offsets; at each sampling instant t_k, the LMPC optimization is solved and the first control move is implemented in a sample-and-hold fashion over $[t_k, t_{k+1})$. From these trajectories, we store transition tuples

$$\mathcal{D}_{\text{off}} := \left\{(s_k, a_k, r_k, s_{k+1})\right\}_{k=0}^{N_{\text{off}}-1}, \qquad N_{\text{off}} := |\mathcal{D}_{\text{off}}| \tag{31}$$

where the learning state includes both the process deviation state and the observer-based disturbance estimate

$$s_k := \left[x(t_k)^\top,\ \hat{\theta}(t_k)^\top\right]^\top \tag{32}$$

a_k is the applied LMPC control input, r_k is the instantaneous reward consistent with the control objective, and s_{k+1} is the future state after one sampling interval. Using $\mathcal{D}_{\text{off}}$, we train a feedforward neural network (FNN) policy $\pi_{\text{teach}}(s)$ to approximate the LMPC feedback law via supervised imitation and use this behavior-cloned policy to initialize the actor for the subsequent RL stage; we then pretrain the twin critic networks using only the offline buffer while keeping the actor fixed at (or close to) its cloned initialization, and finally perform an additional offline refinement stage where the actor is updated using a TD3-BC objective on the offline buffer. After critic-only pretraining, the actor is refined on $\mathcal{D}_{\text{off}}$ using a weighted combination of a TD3 policy objective and a behavior-cloning regularizer (TD3-BC), where the TD3 term encourages actions that maximize the estimated value under the learned twin critics $\{Q_{\psi_1}, Q_{\psi_2}\}$

$$\mathcal{L}_{\text{TD3}}(\phi) = -\frac{1}{N_{\text{off}}}\sum_{k=0}^{N_{\text{off}}-1} \min\Big(Q_{\psi_1}\left(s_k, \pi_\phi(s_k)\right), Q_{\psi_2}\left(s_k, \pi_\phi(s_k)\right)\Big) \tag{33}$$

while the cloning term penalizes deviation from the LMPC-mimicking teacher policy π_{teach}

$$\mathcal{L}_{\text{BC}}(\phi) = \frac{1}{N_{\text{off}}}\sum_{k=0}^{N_{\text{off}}-1} \left\|\pi_\phi(s_k) - \pi_{\text{teach}}(s_k)\right\|_2^2 \tag{34}$$

The final actor objective is the convex combination

$$\mathcal{L}_{\text{actor}}(\phi) = (1-\alpha)\,\mathcal{L}_{\text{TD3}}(\phi) + \alpha\,\mathcal{L}_{\text{BC}}(\phi) \qquad \alpha \in [0,1] \tag{35}$$

where α controls the strength of imitation regularization (typically reduced over training so that the policy can depart from the teacher when doing so improves the critic-evaluated return); since all updates in this section rely exclusively on $\mathcal{D}_{\text{off}}$ prior to deployment, the procedure is offline from the standpoint of closed-loop operation and yields an actor-critic initialization that closely follows the stabilizing LMPC behavior while substantially reducing the exploration burden of the subsequent online stage

3.3. Shielded online implementation

Gerold and Lucia (2025) proposes an adaptive robust *model predictive shielding* framework, where a predictive safety check filters the RL action and replaces unsafe proposals with an offline-trained approximate robust NMPC backup policy. In contrast, we employ a Lyapunov-based supervision layer that certifies the learned candidate via the contractive condition $\dot{V}(x,u) \le S(x)$, and switch to LMPC and a stabilizing reference controller when the condition is violated. Fig. 1 illustrates the proposed shielded online implementation, which combines a learned controller with a supervision layer. In the closed loop, the process is affected by unknown disturbances, W in Eq. (3), while sensors provide real-time feedback. An observer runs in parallel to estimate not only the process state but also a slowly varying disturbance/mismatch term, denoted by $\hat{\theta}$. This estimate provides a compact representation of uncertainty that is directly used by the control layer.

At each sampling instant, the control layer generates a candidate input using the available online information (measured/estimated state and $\hat{\theta}$). The primary controller is an online-updated RL policy Φ_{RL}, trained to approximate the LMPC input–output mapping using the same conditioning variables and a cost-aligned objective, with the goal of retaining LMPC-like performance while avoiding online optimization; however, online learning may occasionally produce degraded policies due to limited data, nonstationarity, or distribution shift, especially during early stages of deployment.

To improve practical robustness, we introduce a computationally efficient fallback learned policy implemented as a feedforward neural network (FNN) Φ_{FNN}, trained offline to imitate the offset-free LMPC. Since Φ_{FNN} can be evaluated with negligible computational cost, it serves as a reliable baseline when online updates temporarily degrade the RL policy. In operation, the controller first compares the predicted Lyapunov time-derivative of the RL proposal and the FNN proposal

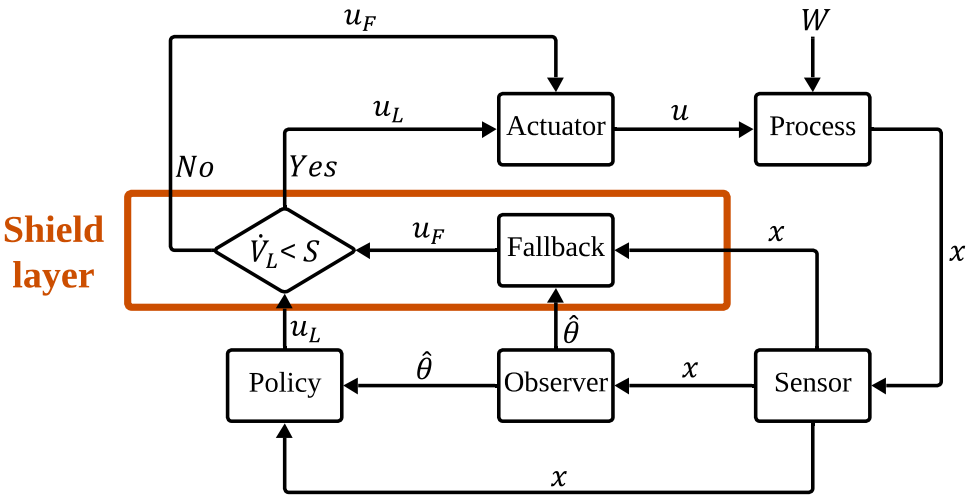


Fig. 1. Closed-loop structure with a Lyapunov-based shield layer. A learning-based policy block selects a candidate input u_L (from either the RL policy or the fast-learned FNN policy) using the measured state x and the disturbance/mismatch estimate $\hat{\theta}$ provided by the observer. A safety check evaluates the Lyapunov condition $\dot{V}(x,u_L) \le S(x)$; if satisfied, u_L is applied to the actuator, otherwise the controller switches to a model-based fallback input u_F. The actuator drives the process subject to an external disturbance W, while the sensor and observer close the feedback loop.

under the current information, and selects the more contractive one as the learned candidate.

Instead of directly applying the learned candidate, a Lyapunov-based shield (gate) is inserted between the learned policy and the actuator. Consistent with the stability condition used in the LMPC design, the gate checks whether the candidate input satisfies the chosen Lyapunov decrease requirement, e.g., of the form in Eq. (9e) or Eq. (10). If the check is satisfied, the candidate is accepted; otherwise, the gate rejects the learned candidate and switches to a model-based fallback input computed by a short-horizon LMPC, which enforces the same Lyapunov condition but calculated by the nominal model while still optimizing the online cost.

To further guarantee that a valid control signal can always be issued, the fallback pathway is complemented by a final failsafe layer. If the LMPC fallback is unavailable or fails to return an input that satisfies the Lyapunov condition, a stabilizing reference controller $\Phi(\cdot)$ (as in Section 2.3) is applied. Overall, the architecture prioritizes learned performance when the candidate is certified as safe, while preserving stability-oriented behavior via systematic switching to the LMPC fallback and the reference controller when needed. The complete shield-layer decision logic is summarized in Algorithm 2.

Beyond online deployment, the same structure supports online training. During operation, closed-loop data are stored and periodically used to refine the RL policy. Importantly, the shield remains active throughout data collection and policy updates, so performance can improve over time while the closed-loop system continues to operate under the same Lyapunov-based supervision. The presence of the offline FNN surrogate further reduces the risk that online training temporarily degrades the closed-loop behavior.

Remark 10. When comparing the learned policies in the first stage of the shield (i.e., Φ_{RL} versus the offline FNN surrogate Φ_{FNN}), we evaluate the Lyapunov time-derivative using the offset-free prediction model Eq. (11) and denote it by $\dot{V}_{\mathrm{of}}(x,u)$. In contrast, when assessing the Lyapunov condition for the LMPC fallback (and the stabilizing reference controller Φ), we compute $\dot{V}(x,u)$ using the nominal model Eq. (1). This choice is consistent with the stabilizability assumption and the stability guarantees established in Section 4, which are derived with respect to the nominal dynamics. The use of $\dot{V}_{\mathrm{of}}$ is solely for selecting between the two learned candidates to improve practical setpoint-tracking performance. Since the FNN is trained to imitate the *offset-free* LMPC feedback map, evaluating its contraction using the offset-free model typically provides a more faithful prediction of its tracking behavior than using the nominal model, whose mismatch-induced bias can lead to overly conservative or misleading comparisons.

Algorithm 2 Shielded online implementation (constraint-enforced switching)

for $k = 0, 1, 2, \ldots$ **do**
  Measure/estimate $x(t_k)$ (and $\hat{\theta}(t_k)$ if used); compute $V(x(t_k))$
  Compute proposals: $u_{\mathrm{RL}} \leftarrow \Phi_{\mathrm{RL}}(x(t_k)) \in U$, $u_{\mathrm{FNN}} \leftarrow \Phi_{\mathrm{FNN}}(x(t_k)) \in U$
  Form $u_{\mathrm{L}} \leftarrow \Phi_{\mathrm{L}}(x(t_k))$, where $\Phi_{\mathrm{L}}(x) \in \arg\min_{\Phi_C \in \{\Phi_{\mathrm{RL}}, \Phi_{\mathrm{FNN}}\}} \dot{V}(x, \Phi_C(x))$
  if $V(x(t_k)) \le \rho_{\mathrm{sw}}$ **or** $\dot{V}(x(t_k), u_{\mathrm{L}}) \le S(x(t_k))$ **then**
    Apply $u(t_k) \leftarrow u_{\mathrm{L}}$
  else
    Compute $u_{\mathrm{LMPC}}(t_k)$
    if $\dot{V}(x(t_k), u_{\mathrm{LMPC}}(t_k)) \le S(x(t_k))$ **then**
      Apply $u(t_k) \leftarrow u_{\mathrm{LMPC}}(t_k)$
    else
      Apply failsafe $u(t_k) \leftarrow \Phi(x(t_k))$
    end if
  end if
  Hold $u(t) = u(t_k)$ for all $t \in [t_k, t_k + \Delta)$
end for

Remark 11. The above two RL realizations are selected as complementary representatives for implementing the proposed shielded, offset-free framework. For the model-free branch, we start from DDPG since it is a classic deterministic actor–critic method for continuous-action control and has been widely adopted as a baseline in continuous-time/continuous-input process systems; we then adopt TD3 as a generally more reliable and more stable improvement over DDPG in practice. This choice is further supported by prior studies that successfully combine TD3-RL with offset-free augmentation by including the mismatch/disturbance estimate in the RL state (e.g., (Hassanpour et al., 2024a,b)). For the model-based branch, we include the HJB-based value–critic approach as a representative realization because the offset-free mechanism continuously estimates and compensates for model–plant mismatch, so the effective prediction model improves over time; consequently, the HJB-residual training and the induced feedback policy can become progressively better aligned with the true closed-loop

behavior. Moreover, HJB/value-function based RL has been studied and applied in chemical process control (e.g., (Wang et al., 2025; Zhu et al., 2025)), motivating it as a practical model-based counterpart to the model-free TD3 branch.

Remark 12. In Algorithm 2, an LMPC fallback failure refers to situations where the LMPC action cannot be computed reliably within the current control interval. Typical causes include: (i) solver failure (e.g., infeasibility, non-convergence, or numerical issues) and (ii) exceeding the prescribed computational time limit, i.e., the solve time is longer than the sampling period Δ. In either case, the LMPC solution is not safely deployable at the current sampling time, and the shield triggers the final failsafe by applying the stabilizing reference controller.

Remark 13. The proposed architecture does not assume that the learned policies (the online RL policy and the offline FNN surrogate) are stabilizing by themselves, nor does it require that the RL training explicitly enforces the Lyapunov decrease condition used in the supervisory layer. Instead, the learned policies are treated as performance-oriented candidate controllers that may occasionally violate the Lyapunov gate. In our application studies, the learned policies nevertheless generate a substantial fraction of actions that satisfy the Lyapunov condition; however, even if one were to incorporate this condition directly into training (e.g., via reward penalties or critic regularization), there is still no guarantee that a learned policy will satisfy it at every time step. Therefore, rigorous closed-loop stability/ultimate-boundedness guarantees stem from the supervisory shield: a learned candidate is applied only if it passes the Lyapunov-based gate, and otherwise the controller switches to the model-based LMPC fallback and, if needed, the stabilizing reference controller. Consequently, the closed-loop inherits the stability properties established for the LMPC/reference controller under sample-and-hold implementation and bounded mismatch, while learning primarily serves to improve performance and reduce the online optimization burden.

4. Closed-loop stability guarantees of the proposed reinforcement learning-based controller

In this section, the closed-loop stability guarantees of the proposed reinforcement learning-based controller are demonstrated under the shielded online implementation.

4.1. Closed-loop stability

For a continuous-time implementation of the reference feedback, the stabilizability assumption in Section 2.3 implies exponential stability of the origin for the *nominal* model. In practice, however, controllers are implemented digitally (sample-and-hold) and the plant may be subject to bounded model-plant mismatch/disturbances. Consequently, the closed-loop behavior is characterized in terms of *practical stability* and *ultimate boundedness*, i.e., convergence to (and invariance of) a neighborhood of the origin. Let $D \subset \mathbb{R}^n$ be an open neighborhood of the origin on which Eq. (7b) and (7c) hold, and pick $\rho > 0$ such that the sublevel set $\Omega_\rho := \{x \in D : V(x) \le \rho\}$ satisfies $\Omega_\rho \subset D$. The analysis is restricted to trajectories with $x(t_0) \in \Omega_\rho$ and to inequalities enforced on Ω_ρ.

Before addressing the uniformly bounded disturbance case, we first consider the case where the mismatch/disturbance is *vanishing*. Consider the true process Eq. (3) under the continuous-time reference feedback $u = \Phi(x)$, i.e., $\dot{x} = F(x, \Phi(x)) + W(x,t)$. Along its trajectories,

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x}\Big(F(x,\Phi(x)) + W(x,t)\Big) \le -c_3 |x|^2 + c_4 |x|\ |W(x,t)| \tag{36}$$

where Eq. (7b) and (7c) were used. If, in addition, the mismatch is vanishing in the sense that there exists a nonnegative function $\delta(t)$ with $\delta(t) \to 0$ as $t \to \infty$ such that

$$|W(x,t)| \le \delta(t)\ |x|\,, \qquad \forall (x,t) \in \Omega_\rho \times \mathbb{R} \tag{37}$$

then

$$\dot{V}(x) \le -\big(c_3 - c_4\delta(t)\big)\,|x|^2 \tag{38}$$

Therefore, there exists a finite time $T \ge t_0$ such that $\delta(t) \le \bar{\delta} < \frac{c_3}{c_4}$ for all $t \ge T$, which implies $\dot{V}(x) \le -\big(c_3 - c_4\bar{\delta}\big)\,|x|^2 < 0$ for all $x \in \Omega_\rho \backslash \{0\}$ and all $t \ge T$. In other words, once the vanishing mismatch becomes sufficiently small, $\dot{V}$ is negative definite on Ω_ρ, and the closed-loop inherits the same exponential decay mechanism as in the nominal case. This case reflects a common practical situation where the model we construct is relatively accurate near the setpoint steady state. Hence, as the state converges toward the setpoint, the model-plant mismatch diminishes, and the closed-loop behavior approaches that of the nominal exponentially stable system.

We now turn to the more general and practically relevant case in which the mismatch/disturbance is only known to be uniformly bounded (not necessarily vanishing). In this case, a strict decrease of V cannot be guaranteed globally, but one can establish practical stability and ultimate boundedness with an explicit ultimate bound, as stated next.

Theorem 1. *Consider the true process Eq. (3) under the reference feedback $u = \Phi(x)$:*

$$\dot{x} = F(x, \Phi(x)) + W(x,t) \tag{39}$$

Let $\Omega_\rho \subset D$ be the sublevel set defined above. Suppose the stabilizability assumption for the nominal model Eq. (1) holds on Ω_ρ with a continuously differentiable Lyapunov function $V(x)$ and constants c_1, c_2, c_3, c_4 satisfying Eq. (7). Assume the disturbance is uniformly bounded as in Eq. (4), i.e., $|W(x,t)| \le W_{\max}$ for all $(x,t) \in \Omega_\rho \times \mathbb{R}$. Let $r > 0$ be such that the ball $\{x \mid |x| < r\}$ is contained in Ω_ρ.

Fix any $\theta \in (0,1)$ and define

$$\mu := \frac{c_4}{c_3}\frac{W_{\max}}{\theta}, \quad k := \sqrt{\frac{c_2}{c_1}}, \quad \gamma := \frac{(1-\theta)c_3}{2c_2}, \quad b := k\,\mu = \frac{c_4}{c_3}\sqrt{\frac{c_2}{c_1}}\frac{W_{\max}}{\theta} \tag{40}$$

If $\mu < k^{-1}r$ (equivalently $W_{\max} < \frac{c_3}{c_4}\sqrt{\frac{c_1}{c_2}}\,\theta\, r$), then for every initial condition satisfying $|x(t_0)| < k^{-1}r$, the solution of Eq. (39) exists for all $t \ge t_0$, remains in Ω_ρ, and there exists a finite time $t_1 \ge t_0$ such that

$$|x(t)| \le k\exp\big(-\gamma(t-t_0)\big)\ |x(t_0)|\,, \qquad t_0 \le t < t_1 \tag{41a}$$

$$|x(t)| \le b, \qquad t \ge t_1 \tag{41b}$$

In particular, the origin of the nominal model is practically stable for the perturbed closed-loop system and the trajectories are ultimately bounded with ultimate bound b. (see Khalil and Grizzle (2002, Thm. 4.10) and Khalil and Grizzle (2002, Lem. 4.8))

Proof. Along trajectories of Eq. (39),

$$\begin{aligned}\dot{V}(x) &= \frac{\partial V(x)}{\partial x}\left(F(x,\Phi(x)) + W(x,t)\right)\\ &\le \frac{\partial V(x)}{\partial x}F(x,\Phi(x)) + \left|\frac{\partial V(x)}{\partial x}\right||W(x,t)|\\ &\le -c_3|x|^2 + c_4|x|\,W_{\max}\end{aligned} \tag{42}$$

where Eq. (7b) and (7c) and $|W(x,t)| \le W_{\max}$ were used (on Ω_ρ). For any fixed $\theta \in (0,1)$,

$$\begin{aligned}-c_3|x|^2 + c_4|x|\,W_{\max} &= -(1-\theta)c_3|x|^2 - \big(\theta c_3|x|^2 - c_4|x|\,W_{\max}\big)\\ &\le -(1-\theta)c_3|x|^2, \qquad \forall\ |x| \ge \mu\end{aligned} \tag{43}$$

with μ defined in Eq. (40). Define $\alpha_1(s) = c_1 s^2$, $\alpha_2(s) = c_2 s^2$, and $\alpha_3(s) = (1-\theta)c_3 s^2$, which are class-$\mathcal{K}$ functions on $[0, r]$. Then Eq. (43) gives $\dot{V} \le -\alpha_3(\|x\|)$ for all $\|x\| \ge \mu$. Applying (Khalil and Grizzle, 2002, Thm. 4.10) on the domain $\{x : \|x\| < r\} \subset \Omega_\rho$ yields practical stability/ultimate boundedness.

Moreover, the explicit bounds Eq. (41) follow from (Khalil and Grizzle, 2002, Lem. 4.8) by viewing W as an additive perturbation with $\delta = W_{\max}$, which yields k, γ, b as in Eq. (40). The condition $\mu < k^{-1}r$ ensures the relevant balls are contained in $\{x : \|x\| < r\} \subset \Omega_\rho$, so the corresponding trajectories remain in Ω_ρ. □

The demonstration characterizes the effect of bounded (or vanishing) mismatch under a *continuous-time* reference feedback $u = \Phi(x)$. In our implementation, however, the applied input is piecewise constant and is taken as the first decision returned by the LMPC at each sampling instant. We therefore next provide a sampled-data stability characterization for the true process under the sample-and-hold application of the LMPC input, showing forward invariance of Ω_ρ and ultimate boundedness to a (possibly inflated) inner level set.

Theorem 2. *Consider the true process Eq. (3) under the sample-and-hold implementation of the first LMPC input:*

$$\dot{x}(t) = F\left(x(t), u(t_k)\right) + W\left(x(t), t\right), \qquad t \in [t_k, t_k + \Delta) \tag{44}$$

where $u(t_k)$ is the first control input vector returned by the LMPC problem Eq. (9), and the LMPC prediction model is the nominal model Eq. (1). Assume Section 2.3 holds for the nominal model on $\Omega_\rho := \{x \in \mathbb{R}^n \mid V(x) \le \rho\} \subset D$, and the disturbance satisfies $W \in \mathcal{W}$ in Eq. (4) on Ω_ρ. Assume $W(x,t)$ is measurable in t and locally Lipschitz in x on D, so that Eq. (44) admits a unique Carathéodory solution. Assume the bounds $\|F(x,u)\| \le M_F$ on $\Omega_\rho \times U$ and the Lipschitz bound Eq. (8c) hold on $\Omega_\rho \times U$ (with constant L'_x). Suppose the LMPC enforces at each sampling instant t_k either Eq. (9e) or Eq. (10), and choose $\alpha = \frac{c_3}{c_2}$.

Let $0 < \rho_s < \rho$ and define the one-step inflated level

$$\rho_{\min} := \sup\left\{V\left(x(t_k + \Delta)\right) \;||\; x(t_k) \in \Omega_{\rho_s},\ W \in \mathcal{W},\ \text{Eq. (44) holds on } [t_k, t_k + \Delta)\right\} \tag{45}$$

Assume $\rho_s < \rho_{\min} < \rho$. If there exists a constant $\epsilon_w > 0$ such that

$$-\alpha\rho_s + L'_x\left(M_F + W_{\max}\right)\Delta + c_4\left(\sqrt{\frac{\rho}{c_1}} + \left(M_F + W_{\max}\right)\Delta\right) W_{\max} \le -\epsilon_w \tag{46}$$

then the closed-loop trajectory starting from any $x(t_0) \in \Omega_\rho$ satisfies:

(1) For any sampling instant t_k with $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_s}$,

$$\dot{V}\left(x(t), u(t_k)\right) \le -\epsilon_w, \qquad \forall\, t \in [t_k, t_k + \Delta) \tag{47}$$

(2) There exists a finite time $t_1 \ge t_0$ such that

$$x(t) \in \Omega_{\rho_{\min}} \subset \Omega_\rho, \qquad \forall\, t \ge t_1 \tag{48}$$

In particular, Ω_ρ is forward invariant and the closed-loop is ultimately bounded.

Proof. Fix any sampling instant t_k and any $t \in [t_k, t_k + \Delta)$. Along Eq. (44),

$$\dot{V}\left(x(t), u(t_k)\right) = \frac{\partial V(x(t))}{\partial x}\Big(F\left(x(t), u(t_k)\right) + W\left(x(t), t\right)\Big) \tag{49}$$

Add and subtract $\frac{\partial V(x(t_k))}{\partial x} F\left(x(t_k), u(t_k)\right)$ to split the derivative into three terms. Using the Lipschitz bound Eq. (8c), the gradient bound Eq. (7c), and the disturbance bound $\|W(x,t)\| \le W_{\max}$, we obtain

$$\dot{V}\left(x(t), u(t_k)\right) \le L'_x\|x(t) - x(t_k)\| + \frac{\partial V(x(t_k))}{\partial x} F\left(x(t_k), u(t_k)\right) + c_4\|x(t)\| W_{\max} \tag{50}$$

Next, by Eq. (44) and the bounds $\|F(x,u)\| \le M_F$ on $\Omega_\rho \times U$ and $\|W(x,t)\| \le W_{\max}$ on Ω_ρ, the sample-and-hold state increment satisfies

$$\|x(t) - x(t_k)\| \le \int_{t_k}^{t} \|\dot{x}(\tau)\|\, d\tau \le \int_{t_k}^{t} \Big(\|F(x(\tau), u(t_k))\| + \|W(x(\tau), \tau)\|\Big)\, d\tau$$

$$\le \left(M_F + W_{\max}\right)\Delta \tag{51}$$

and therefore $\|x(t)\| \le \|x(t_k)\| + (M_F + W_{\max})\Delta$. If $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_s}$, then $\|x(t_k)\| \le \sqrt{\rho/c_1}$ and $V(x(t_k)) \ge \rho_s$. Moreover, depending on which Lyapunov constraint is enforced at t_k: (i) under Eq. (9e), the reference-controller decrease Eq. (7b) gives $\frac{\partial V(x(t_k))}{\partial x} F(x(t_k), u(t_k)) \le -c_3\|x(t_k)\|^2 \le -\frac{c_3}{c_2} V(x(t_k)) \le -\alpha\rho_s$; (ii) under Eq. (10), we have directly $\frac{\partial V(x(t_k))}{\partial x} F(x(t_k), u(t_k)) \le -\alpha V(x(t_k)) \le -\alpha\rho_s$. Substituting these bounds yields, for all $t \in [t_k, t_k + \Delta)$,

$$\dot{V}\left(x(t), u(t_k)\right) \le -\alpha\rho_s + L'_x(M_F + W_{\max})\Delta + c_4\left(\sqrt{\frac{\rho}{c_1}} + (M_F + W_{\max})\Delta\right) W_{\max} \tag{52}$$

so Eq. (46) implies Eq. (47).

For Item 2, when $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_s}$, Item 1 implies $\dot{V} \le -\epsilon_w$ on $[t_k, t_k + \Delta)$, so V strictly decreases on that interval. When $x(t_k) \in \Omega_{\rho_s}$, the definition Eq. (45) together with $\rho_s < \rho_{\min} < \rho$ implies $x(t_{k+1}) \in \Omega_{\rho_{\min}} \subset \Omega_\rho$. Therefore, Ω_ρ is forward invariant. Finally, since $\dot{V} \le -\epsilon_w$ whenever $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_s}$, the Lyapunov level at sampling instants decreases until the trajectory reaches Ω_{ρ_s} in finite time at some sampling instant $t_{k^\star}$. Once $x(t_{k^\star}) \in \Omega_{\rho_s}$, the definition Eq. (45) implies $x(t_{k^\star+1}) \in \Omega_{\rho_{\min}}$, and inductively the trajectory remains in $\Omega_{\rho_{\min}}$ thereafter, establishing Eq. (48). □

As seen in the proposed constraint-enforced switching framework, the control law operates by conditionally selecting which control signal is applied to the actuators. In addition to the online-updated RL policy, we include an offline-trained feedforward neural network (FNN) as a computationally efficient surrogate of the Lyapunov-contractive MPC. The FNN provides a reliable baseline when online learning temporarily degrades performance. At each sampling instant, the controller first selects a *learned candidate* by comparing the predicted Lyapunov decrease of the online RL policy and the FNN surrogate, and then applies the Lyapunov-based constraint enforcer. The closed-loop stability/ultimate-boundedness guarantees are inherited from the Lyapunov gate and the LMPC/reference-controller fallback, rather than from intrinsic stability properties of the learned policies.

Despite neither the online RL policy nor the FNN surrogate having an a priori stability guarantee, the stability guarantees for the LMPC shown in Theorem 2 also guarantee stability for the modified framework via the constraint enforcer

Theorem 3. *Consider the true process Eq. (3) under sample-and-hold control with sampling period Δ:*

$$\dot{x}(t) = F\left(x(t), u(t_k)\right) + W\left(x(t), t\right), \qquad t \in [t_k, t_k + \Delta) \tag{53}$$

Assume Section 2.3 holds on $\Omega_\rho := \{x \mid V(x) \le \rho\} \subset D$ and the disturbance satisfies $W \in \mathcal{W}$ in Eq. (4) on Ω_ρ. Assume $W(x,t)$ is measurable in t and locally Lipschitz in x on D, so that Eq. (53) admits a unique Carathéodory solution.

Let $\Phi_{\mathrm{RL}} : \mathbb{R}^n \to U$ be an RL controller satisfying $\Phi_{\mathrm{RL}}(x) \in U$ for all x, and let $\Phi_{\mathrm{FNN}} : \mathbb{R}^n \to U$ be an offline FNN surrogate policy satisfying $\Phi_{\mathrm{FNN}}(x) \in U$ for all x.

Fix $0 < \rho_{\mathrm{sw}} < \rho$ and define

$$S(x) \in \left\{\ \dot{V}\left(x, \Phi(x)\right),\ -\alpha V(x)\ \right\} \tag{54}$$

consistent with whether Eq. (9e) or Eq. (10) is used.

Define the learned candidate policy Φ_{L} by selecting between Φ_{RL} and Φ_{FNN} based on the predicted Lyapunov decrease:

$$\Phi_{\mathrm{L}}(x) \in \underset{\Phi_C \in \{\Phi_{\mathrm{RL}}, \Phi_{\mathrm{FNN}}\}}{\arg\min}\ \dot{V}\left(x, \Phi_C(x)\right) \tag{55}$$

Use the following constraint enforcer:

$$u\left(t_k\right) = \begin{cases} \Phi_{\mathrm{L}}\left(x\left(t_k\right)\right), & \text{if } \begin{array}{l} V\left(x\left(t_k\right)\right) \le \rho_{\mathrm{sw}} \ \text{ or} \\ \dot{V}\left(x\left(t_k\right), \Phi_{\mathrm{L}}\left(x\left(t_k\right)\right)\right) \le S\left(x\left(t_k\right)\right) \end{array} \\ \Phi_{\mathrm{LMPC}}\left(t_k\right), & \text{if } \begin{array}{l} V\left(x\left(t_k\right)\right) > \rho_{\mathrm{sw}} \ \text{ and} \\ \dot{V}\left(x\left(t_k\right), \Phi_{\mathrm{L}}\left(x\left(t_k\right)\right)\right) > S\left(x\left(t_k\right)\right) \end{array} \end{cases} \tag{56}$$

where Φ_{LMPC} is defined by

$$\Phi_{\mathrm{LMPC}}\left(t_k\right)=\begin{cases}u_{\mathrm{LMPC}}\left(t_k\right), & \text{if } \dot{V}\left(x\left(t_k\right),u_{\mathrm{LMPC}}\left(t_k\right)\right)\le S\left(x\left(t_k\right)\right)\\ \Phi\left(x\left(t_k\right)\right), & \text{if } \dot{V}\left(x\left(t_k\right),u_{\mathrm{LMPC}}\left(t_k\right)\right)> S\left(x\left(t_k\right)\right)\end{cases} \tag{57}$$

and always returns an input in U.

Assume the following outer-region decay property holds: there exist constants $0<\rho_s<\rho$ and $\epsilon_w>0$ such that for any sampling instant t_k with $x(t_k)\in\Omega_\rho\backslash\Omega_{\rho_s}$, whenever the applied input satisfies

$$\dot{V}\left(x(t_k),u(t_k)\right)\le S\left(x(t_k)\right) \tag{58}$$

the sample-and-hold trajectory satisfies

$$\dot{V}\left(x(t),u(t_k)\right)\le-\epsilon_w,\qquad \forall\, t\in[t_k,t_k+\Delta) \tag{59}$$

In particular, when $u(t_k)=u_{\mathrm{LMPC}}(t_k)$ is the first input returned by the LMPC and the LMPC enforces Eq. (58) at t_k, Theorem 2 (Item 1) provides sufficient conditions for Eq. (59).

Define the inner level

$$\rho_{\mathrm{in}}:=\max\{\rho_s,\rho_{\mathrm{sw}}\} \tag{60}$$

Define the one-step worst-case Lyapunov level:

$$\rho^\star_{\mathrm{min}}:=\sup\left\{V\left(x(t_k+\Delta)\right)\;||\;x(t_k)\in\Omega_{\rho_{\mathrm{in}}},\;W\in\mathcal{W},\;Eq.(53)\text{holds on }[t_k,t_k+\Delta)\right\} \tag{61}$$

Assume the invariance-feasibility condition

$$\rho_{\mathrm{in}}<\rho^\star_{\mathrm{min}}<\rho \tag{62}$$

Then for any $x(t_0)\in\Omega_\rho$, the closed-loop trajectory under Eq. (56) satisfies:

1. ***Forward invariance (hence boundedness):** for all $t\ge t_0$, $x(t)\in\Omega_\rho$, and thus by Eq. (7a), $\|x(t)\|\le\sqrt{\rho/c_1}$ for all $t\ge t_0$.*
2. ***Ultimate boundedness with explicit bound:** there exists a finite time $t_1\ge t_0$ such that*

$$x(t)\in\Omega_{\rho^\star_{\mathrm{min}}}\subset\Omega_\rho,\qquad\forall\, t\ge t_1 \tag{63}$$

Consequently, again using Eq. (7a), $\|x(t)\|\le\sqrt{\rho^\star_{\mathrm{min}}/c_1}$ for all $t\ge t_1$. Moreover, at any sampling instant with $V(x(t_k))\le\rho_{\mathrm{sw}}$, we have $\|x(t_k)\|\le\sqrt{\rho_{\mathrm{sw}}/c_1}$.

Proof. Fix any sampling instant t_k.

(1) Forward invariance of Ω_ρ. Assume $x(t_k)\in\Omega_\rho$, i.e., $V(x(t_k))\le\rho$.

If $V(x(t_k))>\rho_{\mathrm{in}}$, then $V(x(t_k))>\rho_{\mathrm{sw}}$ and $V(x(t_k))>\rho_s$. By the constraint enforcer Eq. (56), the applied input satisfies $\dot{V}(x(t_k),u(t_k))\le S(x(t_k))$. Since $x(t_k)\in\Omega_\rho\backslash\Omega_{\rho_s}$, the outer-region decay property Eq. (59) gives

$$\dot{V}(x(t),u(t_k))\le-\epsilon_w,\qquad\forall\, t\in[t_k,t_k+\Delta)$$

so V strictly decreases on $[t_k,t_k+\Delta)$ and thus $V(x(t))\le V(x(t_k))\le\rho$ for all $t\in[t_k,t_k+\Delta)$.

If $V(x(t_k))\le\rho_{\mathrm{in}}$, then by the definition Eq. (61) and the assumption $\rho^\star_{\mathrm{min}}<\rho$, we have $V(x(t_k+\Delta))\le\rho^\star_{\mathrm{min}}<\rho$, hence $x(t_{k+1})\in\Omega_{\rho^\star_{\mathrm{min}}}\subset\Omega_\rho$. Therefore, in both cases, whenever $x(t_k)\in\Omega_\rho$ we obtain $x(t_{k+1})\in\Omega_\rho$. By induction over k, Ω_ρ is forward invariant, and the bound $\|x(t)\|\le\sqrt{\rho/c_1}$ follows from Eq. (7a).

(2) Ultimate boundedness. Whenever $V(x(t_k))>\rho_{\mathrm{in}}$, the argument above yields $\dot{V}\le-\epsilon_w$ on $[t_k,t_k+\Delta)$, so V strictly decreases on that interval. Hence, after finitely many sampling instants there exists $t_{k^\star}$ such that $V(x(t_{k^\star}))\le\rho_{\mathrm{in}}$. For this sampling instant, the definition Eq. (61) implies $V(x(t_{k^\star+1}))\le\rho^\star_{\mathrm{min}}$.

We now show that once $V(x(t_j))\le\rho^\star_{\mathrm{min}}$ for some j, then $V(x(t_{j+1}))\le\rho^\star_{\mathrm{min}}$. If $V(x(t_j))\le\rho_{\mathrm{in}}$, then $\Omega_{V(x(t_j))}\subseteq\Omega_{\rho_{\mathrm{in}}}$ and Eq. (61) implies $V(x(t_{j+1}))\le\rho^\star_{\mathrm{min}}$. If instead $\rho_{\mathrm{in}}<V(x(t_j))\le\rho^\star_{\mathrm{min}}$, then $V(x(t_j))>\rho_s$ and $V(x(t_j))>\rho_{\mathrm{sw}}$, so the enforcer gives $\dot{V}(x(t_j),u(t_j))\le S(x(t_j))$ and the outer-region decay property yields $\dot{V}(x(t),u(t_j))\le-\epsilon_w$ on $[t_j,t_j+\Delta)$, hence $V(x(t_{j+1}))<V(x(t_j))\le\rho^\star_{\mathrm{min}}$.

Therefore $V(x(t_k))\le\rho^\star_{\mathrm{min}}$ holds for all sufficiently large k, and the corresponding continuous-time statement Eq. (63) follows. The norm bound follows from Eq. (7a). □

Remark 14. The alternate form of the stability constraint from Eq. (10) does not explicitly use the reference controller, but still requires it to exist due to the Lyapunov function needing to satisfy Eq. (7). A consequence of using this form is that α must be defined by the user. An excessively small α risks being overpowered by the sample-and-hold and disturbance terms in Eq. (46) (e.g., $L'_x(M_F+W_{\max})\Delta$ and $c_4(\sqrt{\rho/c_1}+(M_F+W_{\max})\Delta)W_{\max}$), whereas an excessively large α risks the solution being infeasible due to the control bounds.

Remark 15. The proof demonstrates stability guarantees for any such interval in which the stability constraints are enforced. As presented in Eq. (9), this implies that stability guarantees do not exist beyond the first sampling interval; hence, the receding horizon approach would functionally satisfy the stability guarantees but is not guaranteed to optimize with respect to a trajectory that satisfies these guarantees for all points beyond the first sampling interval.

Remark 16. The RL-based controller has no stability guarantees alone, hence why every case eventually reforms to be in terms of the reference controller, as the reference controller's existence and use are solely for the enforcement and satisfaction of stability guarantees.

Remark 17. The ultimate bound size can be reduced if a *certified* backup controller is also enforced inside the switching region. Specifically, when $V(x(t_k))\le\rho_{\mathrm{sw}}$ (hence $\|x(t_k)\|\le\sqrt{\rho_{\mathrm{sw}}/c_1}$), one may apply an alternative backup law that satisfies a stronger Lyapunov-decrease condition under sample-and-hold, e.g., an offset-free constraint of the form $\dot{V}_{\mathrm{of}}(x(t_k),u(t_k))\le-\alpha V(x(t_k))$. This replaces the generic one-step inflation bound used in Theorem 3 by a tighter inner-region bound (see Appendix), yielding a smaller ultimate invariant set; under constant matched mismatch, the residual term vanishes and the resulting ultimate bound coincides with the nominal sample-and-hold bound.

4.2. Offset-free observer error bound

We analyze the extended Luenberger observer Eq. (13) under a matched structured mismatch assumption consistent with Eq. (11). Specifically, assume that the true process Eq. (3) satisfies

$$W(x,t)=G_\theta\,\theta^\star(t) \tag{64}$$

Here, $G_\theta\in\mathbb{R}^{n\times p}$ is known and constant and $\theta^\star(t)\in\mathbb{R}^p$ is possibly time-varying.

Suppose $F(\cdot,\cdot)$ is Lipschitz in x on D uniformly in u, namely there exists $L_F>0$ such that for all $x_1,x_2\in D$

$$\left|F\left(x_1,u\right)-F\left(x_2,u\right)\right|\le L_F\,\left|x_1-x_2\right| \tag{65}$$

Also suppose there exists $d_{\max}\ge0$ such that

$$\left|\dot{\theta}^\star(t)\right|\le d_{\max},\qquad\forall t\ge0 \tag{66}$$

Assume full-state measurement is available to the observer. Under sample-and-hold control, the applied input satisfies $u(t)=u(t_k)$ for all $t\in[t_k,t_{k+1})$. Accordingly, on each interval $[t_k,t_{k+1})$ the observer dynamics are given by Eq. (13) with $u(t)$ replaced by $u(t_k)$.

Define the estimation errors

$$e_x(t):=x(t)-\hat{x}(t),\qquad e_\theta(t):=\theta^\star(t)-\hat{\theta}(t),\qquad e(t):=\left[e_x(t)^\top,\;e_\theta(t)^\top\right]^\top \tag{67}$$

Theorem 4. *Consider the true process Eq. (3) with Eq. (64) and the observer Eq. (13) with full-state measurement implemented under sample-and-hold input $u(t) = u(t_k)$ on $[t_k, t_{k+1})$. Define*

$$A := \begin{bmatrix} -K_x & G_\theta \\ -K_\theta & 0 \end{bmatrix}, \qquad B := \begin{bmatrix} 0 \\ I_p \end{bmatrix} \tag{68}$$

Assume the gains K_x, K_θ are chosen such that A is Hurwitz. Let $Q = Q^\top \succ 0$ be arbitrary and let $P = P^\top \succ 0$ solve

$$A^\top P + PA = -Q \tag{69}$$

Define $V_e(e) := \frac{1}{2} e^\top P e$, $\underline{\lambda} := \lambda_{\min}(P)$, and $\overline{\lambda} := \lambda_{\max}(P)$. If

$$\alpha := \tfrac{1}{2}\lambda_{\min}(Q) - \|P\| L_F > 0 \tag{70}$$

then the following statements hold as long as $x(t), \hat{x}(t) \in D$.

(i) Eventually constant mismatch (zero steady-state error). *If there exists a finite time $t_c \geq 0$ such that $\dot{\theta}^\star(t) \equiv 0$ for all $t \geq t_c$, then for all $t \geq t_c$*

$$V_e(e(t)) \leq \exp\left[-\kappa(t - t_c)\right] V_e\left(e(t_c)\right), \qquad \kappa := \frac{\alpha}{\overline{\lambda}} \tag{71}$$

Consequently

$$\|e(t)\| \leq \sqrt{\frac{\overline{\lambda}}{\underline{\lambda}}} \exp\left[-\frac{\kappa}{2}(t - t_c)\right] \|e(t_c)\|, \qquad \forall t \geq t_c \tag{72}$$

In particular, $e(t) \to 0$, namely $\hat{x}(t) \to x(t)$ and $\hat{\theta}(t) \to \theta^\star$ as $t \to \infty$.

(ii) Time-varying bounded mismatch (bounded estimation error). *If $\|\dot{\theta}^\star(t)\| \leq d_{\max}$ for all $t \geq 0$, then for all $t \geq 0$*

$$V_e(e(t)) \leq \exp[-\kappa t]\, V_e(e(0)) + \rho^\star \left[1 - \exp[-\kappa t]\right] \tag{73}$$

where $\kappa = \alpha/\overline{\lambda}$ and

$$\beta := \|PB\|\, d_{\max}, \qquad \rho^\star := \frac{\overline{\lambda}}{2\alpha^2}\beta^2 = \frac{\overline{\lambda}}{2\alpha^2}\|PB\|^2 d_{\max}^2 \tag{74}$$

Consequently

$$\limsup_{t\to\infty} V_e(e(t)) \leq \rho^\star \qquad \limsup_{t\to\infty} \|e(t)\| \leq \sqrt{\frac{2\rho^\star}{\underline{\lambda}}} \tag{75}$$

Moreover, the mismatch reconstruction error $\tilde{w}(t) := G_\theta e_\theta(t)$ is ultimately bounded

$$\limsup_{t\to\infty} \|\tilde{w}(t)\| \leq \|G_\theta\| \sqrt{\frac{2\rho^\star}{\underline{\lambda}}} \tag{76}$$

Proof. Using the true process Eq. (3) with Eq. (64) and the observer Eq. (13) implemented under sample-and-hold input $u(t) = u(t_k)$ on $[t_k, t_{k+1})$, the estimation error e satisfies the compact dynamics

$$\dot{e} = Ae + \underbrace{\begin{bmatrix} F(x,u) - F(\hat{x},u) \\ 0 \end{bmatrix}}_{\Delta(e_x)} + B\,\dot{\theta}^\star(t) \tag{77}$$

where A, B are given in Eq. (68) and $u = u(t_k)$ on $[t_k, t_{k+1})$. By Eq. (65), $\|\Delta(e_x)\| \leq L_F \|e_x\| \leq L_F \|e\|$.

Let $V_e(e) = \frac{1}{2} e^\top P e$ with P solving Eq. (69). Along trajectories of Eq. (77)

$$\begin{aligned} \dot{V}_e &= e^\top P \dot{e} \\ &= e^\top PAe + e^\top P\Delta(e_x) + e^\top PB\,\dot{\theta}^\star(t) \\ &= -\tfrac{1}{2} e^\top Q e + e^\top P\Delta(e_x) + e^\top PB\,\dot{\theta}^\star(t) \qquad (78) \\ &\leq \left(-\tfrac{1}{2}\lambda_{\min}(Q) + \|P\| L_F\right)\|e\|^2 + \|PB\|\,\|\dot{\theta}^\star(t)\|\,\|e\| \\ &\leq -\alpha\|e\|^2 + \beta\|e\| \qquad (79) \end{aligned}$$

where α is defined in Eq. (70). Under $\|\dot{\theta}^\star(t)\| \leq d_{\max}$, the term β is given in Eq. (74). Using Young's inequality

$$\beta\|e\| \leq \frac{\alpha}{2}\|e\|^2 + \frac{\beta^2}{2\alpha} \tag{80}$$

Substituting into Eq. (79) gives

$$\dot{V}_e \leq -\frac{\alpha}{\overline{\lambda}} V_e + \frac{\beta^2}{2\alpha} = -\kappa V_e + c \tag{81}$$

with $\kappa = \alpha/\overline{\lambda}$ and $c = \beta^2/(2\alpha)$.

Assume there exists $t_c \geq 0$ such that $\dot{\theta}^\star(t) \equiv 0$ for all $t \geq t_c$. Then for all $t \geq t_c$ we have $\beta = 0$ and Eq. (81) reduces to $\dot{V}_e \leq -\kappa V_e$ on $[t_c, \infty)$, which implies Eq. (71). Using $\frac{1}{2}\underline{\lambda}\|e\|^2 \leq V_e(e) \leq \frac{1}{2}\overline{\lambda}\|e\|^2$ yields Eq. (72).

With $\|\dot{\theta}^\star(t)\| \leq d_{\max}$, Eq. (81) implies $\dot{V}_e \leq -\kappa V_e + c$. Solving the comparison system $\dot{z} = -\kappa z + c$ with $z(0) = V_e(e(0))$ yields Eq. (73) with $\rho^\star = c/\kappa$ as in Eq. (74). The bounds in Eq. (75) follow from $\frac{1}{2}\underline{\lambda}\|e\|^2 \leq V_e(e) \leq \frac{1}{2}\overline{\lambda}\|e\|^2$. Finally, Eq. (76) follows from $\tilde{w} = G_\theta e_\theta$ and $\|e_\theta\| \leq \|e\|$ □

Corollary 1. *Under the matched mismatch assumption Eq. (64), let $u_{\text{sp}}^\star(t)$ denote the (ideal) equilibrium input obtained from Eq. (14) by replacing $\hat{\theta}(t_k)$ with the true mismatch parameter $\theta^\star(t)$, and recall that the implemented update computes $u_{\text{sp}}(t_k)$ via Eq. (14). Suppose the Jacobian condition Eq. (15) holds on a neighborhood of interest. Moreover, assume the equilibrium Jacobian is uniformly nonsingular on a set Θ containing $\theta^\star(t)$ and $\hat{\theta}(t)$, i.e.,*

$$\sup_{\theta\in\Theta} \left\| \left(\frac{\partial}{\partial u}\left[F(0,u) + G_\theta \theta\right] \right)^{-1} \right\| \leq M_u < \infty \tag{82}$$

Then the induced equilibrium mapping $u_{\text{sp}} = \Psi(\theta)$ is locally Lipschitz on Θ, and for all t with $\theta^\star(t), \hat{\theta}(t) \in \Theta$ the steady-state input error satisfies

$$\|u_{\text{sp}}(t) - u_{\text{sp}}^\star(t)\| \leq L_\Psi \|\hat{\theta}(t) - \theta^\star(t)\| = L_\Psi \|e_\theta(t)\| \leq L_\Psi \|e(t)\| \tag{83}$$

*where one may take $L_\Psi := M_u \|G_\theta\|$. Consequently, in case **(i)** of Theorem 4, we have $u_{\text{sp}}(t) \to u_{\text{sp}}^\star(t)$; in case **(ii)**, the computed steady-state input is ultimately bounded as*

$$\limsup_{t\to\infty} \|u_{\text{sp}}(t) - u_{\text{sp}}^\star(t)\| \leq L_\Psi \sqrt{\frac{2\rho^\star}{\underline{\lambda}}} \tag{84}$$

5. Application to a chemical process example

In this section, we apply the proposed stable and robust RL-based controller with the shielded layer to a representative chemical process. In particular, we implement the framework in a closed loop and evaluate how the Lyapunov-based safety shield (fallback controller) improves reliability under unknown model-process mismatch and disturbances. Additionally, to highlight the benefit of the proposed design, we compare it against a conventional RL controller and an RL implementation without any backup. Finally, we benchmark the resulting controller against various other controllers to quantify both closed-loop performance and computational efficiency.

5.1. Process description

The model chemical process of choice for this study is a simulated continuous stirred-tank reactor (CSTR). The CSTR is assumed to be perfectly mixed and insulated. We consider a singular irreversible elementary reaction that is exothermic, making the CSTR non-isothermal. The reaction is treated as an arbitrary liquid-phase reaction ($A \to B$) with second-order dynamics. Heat is added or removed from the system through a controllable heating rate $\dot{Q}$. These assumptions yield the following dynamic model:

$$\frac{\mathrm{d}C_A}{\mathrm{d}t} = \frac{F}{V_L}\left(C_{A0} - C_A\right) - kC_A^2 \tag{85a}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{F}{V_L}\left(T_0 - T\right) - \frac{\Delta H}{\rho_L C_p} kC_A^2 + \frac{\dot{Q}}{\rho_L C_p V_L} \tag{85b}$$

$$k = k_0 \exp\left[-\frac{E}{RT}\right] \tag{85c}$$

Table 1
Parameter values of the CSTR model.

Variable	Value	Variable	Value
C_{As}	$1.954\,\mathrm{kmol\,m^{-3}}$	C_{A0s}	$4.0\,\mathrm{kmol\,m^{-3}}$
C_p	$0.231\,\mathrm{kJ\,kg^{-1}\,K^{-1}}$	ΔH	$-1.15\times10^{4}\,\mathrm{kJ\,kmol^{-1}}$
E	$5.0\times10^{4}\,\mathrm{kJ\,kmol^{-1}}$	F	$5\,\mathrm{m^3\,h^{-1}}$
k_0	$8.46\times10^{6}\,\mathrm{m^3\,kmol^{-1}\,h^{-1}}$	$\dot{Q}_s$	$0.0\,\mathrm{kJ\,h^{-1}}$
R	$8.314\,\mathrm{kJ\,kmol^{-1}\,K^{-1}}$	ρ_L	$1.0\times10^{3}\,\mathrm{kg\,m^{-3}}$
T_0	300.0 K	T_{0s}	300.0 K
T_s	401.9 K	V_L	$1\,\mathrm{m^3}$

Here—with the exception of k_0, which denotes the isothermal rate-constant—the "0" subscript denotes feed values, C_A denotes concentration of A, and T denotes the temperature of the solution within the CSTR. $\rho_L, C_p, \Delta H, E$ and V_L denote the solution density, specific heat, heat of reaction, activation energy and liquid volume, respectively.

5.2. Control problem

The inlet concentration C_{A0} and the heating rate $\dot{Q}$ are selected as the manipulated inputs, and the state variables are chosen to be C_A and T. In order to utilize the origin as the steady state (denoted by the s subscript) without loss of generality, the state and control variables are expressed as deviation variables. Accordingly, the vectors used in the problem formulation are defined as $x^\top = \begin{bmatrix} C_A - C_{As}, & T - T_s \end{bmatrix}$ and $u^\top = \begin{bmatrix} C_{A0} - C_{A0s}, & \dot{Q} - \dot{Q}_s \end{bmatrix}$ for the state and control vectors, respectively. The control inputs are subject to bounds, specifically $-3.5 \le C_{A0} - C_{A0s} \le 3.5\,\mathrm{kmol\,m^{-3}}$ and $-5\times10^{5} \le \dot{Q} - \dot{Q}_s \le 5\times10^{5}\,\mathrm{kJ\,h^{-1}}$. Specifics on the various constants used in the CSTR dynamic model are provided in Table 1.

The design goal is to create a controller that drives the closed-loop system from any given initial state bounded by $-0.6 \le C_A - C_{As} \le 0.6\,\mathrm{kmol\,m^{-3}}$ and $-10 \le T - T_s \le 10\,\mathrm{K}$ to the origin. Because of the deviation-variable notation, this origin represents the desired operating (unstable) steady state. To achieve this, a reference controller satisfying Section 2.3 is chosen as a proportional (P) controller acting on both deviation states with gains $k_{c,1} = 2$ and $k_{c,2} = 5{,}000$ and saturations consistent with the input bounds, i.e., $C_{A0} - C_{A0s} = \mathrm{clip}\big(-k_{c,1}(C_A - C_{As}), [-3.5, 3.5]\big)$ and $\dot{Q} - \dot{Q}_s = \mathrm{clip}\big(-k_{c,2}(T - T_s), [-5\times10^{5}, 5\times10^{5}]\big)$. Similarly, a Lyapunov function of the form $V = x^\top P x$ with

$$P = \begin{bmatrix} 1{,}060 & 22 \\ 22 & 0.52 \end{bmatrix}$$

is used.

The immediate cost of the LMPC and RL at $t = t_k$ is designed as the quadratic form:

$$L\big(x(t_k), u(t_k)\big) = x(t_k)^\top W_x\, x(t_k) + u(t_k)^\top W_u\, u(t_k) \tag{86}$$

with weighting matrices $W_x = \mathrm{diag}(1000,\, 1)$ and $W_u = \mathrm{diag}(10,\, 10^{-8})$. The consistent cost (negative reward) function is used to ensure a fair comparison between all controllers.

In this study, we evaluate robustness under bounded model mismatch and time variation by considering additive disturbances ($W(x,t)$ in Eq. (3)) in both Eqs. (85a) and (85b). Over the operating region induced by the initial-state set, we assume that the disturbance terms are bounded as $|W_1(x,t)| \le 10$ and $|W_2(x,t)| \le 500$ for all t, where $W_1(x,t)$ and $W_2(x,t)$ denote the additive disturbances in the concentration and temperature dynamics, respectively. To generate parametric mismatch scenarios that are consistent with these bounds, we introduce time-varying perturbations through the feed temperature $T_0(t)$ and activation energy $E(t)$ and select their admissible ranges such that the resulting induced mismatch remains within the assumed bounds on W_1 and W_2 over the operating region. Physically, perturbing $T_0(t)$ models variations in the inlet/feed thermal condition (e.g., upstream heat-exchanger performance changes or ambient-induced fluctuations). Perturbing $E(t)$ models uncertainty and slow drift in reaction kinetics (e.g., catalyst aging/deactivation, impurities, or unmodeled chemistry), effectively changing the temperature sensitivity of the reaction rate and thereby impacting both the concentration consumption rate and the heat-release rate. Specifically, we constrain the perturbations to remain within $T_0(t) \in [290\,\mathrm{K}, 300\,\mathrm{K}]$ and $E(t) \in [5.0\times10^{4}\,\mathrm{kJ\,kmol^{-1}}, 5.15\times10^{4}\,\mathrm{kJ\,kmol^{-1}}]$ for all t.

We test robustness under two types of mismatch/nonstationarity: (i) a transient drift that settles after an initial period and (ii) a persistent drift that continues throughout the evaluation horizon. In our setting, these two cases are emulated by applying time-varying perturbations to the feed temperature $T_0(t)$ and activation energy $E(t)$ on the plant side while keeping the controller model nominal. In both cases, the perturbation is composed of a constant offset (offline model-plant mismatch) and a time-varying component:

$$T_0(t) = 300\,\mathrm{K} + \Delta T_0 + \delta T_0(t), \qquad E(t) = 5.0\times10^{4}\,\mathrm{kJ\,kmol^{-1}} + \Delta E + \delta E(t)$$

where ΔT_0 and ΔE are constant offsets, and $\delta T_0(t)$ and $\delta E(t)$ describe the drift profile. For the transient-drift case, we ramp $\delta T_0(t)$ and $\delta E(t)$ linearly over an initial duration T_d and then hold them constant (equivalently, once the drift reaches its prescribed terminal value or a bound it remains fixed thereafter). For the persistent-drift case, we ramp $\delta T_0(t)$ and $\delta E(t)$ continuously over the entire horizon $t \in [0, T_{\mathrm{end}}]$ without settling. In all simulations, the drift magnitudes and offsets are selected (and saturated when needed) so that $T_0(t)$ remains between 290 K and 300 K, and $E(t)$ remains between $5.0\times10^{4}\,\mathrm{kJ\,kmol^{-1}}$ and $5.15\times10^{4}\,\mathrm{kJ\,kmol^{-1}}$, for all t.

Remark 18. All closed-loop simulations are implemented in a sample-and-hold fashion with sampling period $\Delta = 5\,\mathrm{s}$. Within each sampling interval $[t_k, t_k + \Delta)$, the control input is held constant as $u(t) = u(t_k)$, and the CSTR dynamics are numerically integrated using the forward Euler method with integration step size $\mathrm{d}t = 0.1\,\mathrm{s}$. This integration step size was validated by repeating representative closed-loop simulations with smaller $\mathrm{d}t$, which yielded essentially identical state and observer trajectories.

Remark 19. When selecting the Lyapunov function $V(x) = x^\top P x$ and the reference stabilizing controller $\Phi(x)$ (the saturated P controller), we proceed as follows. First, we tune $\Phi(x)$ so that it stabilizes the deviation dynamics and drives the state to the origin while respecting the input bounds over the operating region D. Next, with $\Phi(x)$ fixed, we construct a quadratic Lyapunov candidate by choosing a symmetric positive definite matrix $P \succ 0$ and verifying (numerically) that there exists a level set $\Omega_\rho := \{x : V(x) \le \rho\}$ such that $\Omega_\rho \subset D$, $\Phi(x) \in U$ for all $x \in \Omega_\rho$, and the assumptions in Section 2.3 hold on Ω_ρ. If a sampled P fails these checks, it is discarded and resampled until a valid P is obtained.

Remark 20. In the numerical studies, the admissible operating region for the stability analysis and switching logic is chosen as $\Omega_\rho = \{x \mid V(x) \le \rho\}$ with $\rho = 200$, and the switching level in Eq. (56) is set to $\rho_{\mathrm{sw}} = 20$. The choice $\rho = 200$ is made such that, over Ω_ρ, the saturated reference controller $\Phi(\cdot)$ together with the quadratic Lyapunov function $V(x) = x^\top P x$ satisfies the conditions stated in Section 2.3. The switching level $\rho_{\mathrm{sw}} = 20$ is selected as an inner level set that the reference controller can robustly drive trajectories into from the outer region $\Omega_\rho \setminus \Omega_{\rho_{\mathrm{sw}}}$, so that states starting in Ω_ρ are brought into $\Omega_{\rho_{\mathrm{sw}}}$ in finite time under the reference controller.

5.3. Offset-free observer and LMPC

The online implementation of the offset-free LMPC follows the same closed-loop structure shown in Fig. 1. In this case, the control policy is the proposed offset-free LMPC described in Section 2.5. Specifically, the prediction model is the offset-free model Eqs. (11) and (12), the

disturbance states are estimated online using the extended Luenberger observer Eq. (13), and the steady-state tracking input $u_{sp}(t_k)$ is computed from the equilibrium condition Eq. (14). At each sampling instant t_k, the optimizer is initialized with the current measured state and uses the most recent disturbance estimate $\hat{\theta}(t_k)$ as a constant parameter over the horizon to solve the offset-free Lyapunov-based MPC problem Eq. (18).

In this work, the LMPC optimization problem is solved using the sequential least squares quadratic programming (SLSQP) algorithm, which is a gradient-based method for constrained nonlinear programs. At each iteration, SLSQP solves a quadratic approximation of the original problem and updates the decision variables until convergence. Based on test simulations, the convergence tolerance and the finite-difference step size are set as 1×10^{-12} and 1×10^{-6}. The resulting first control move $u(t_k)$ is applied to the plant in a sample-and-hold fashion over $[t_k, t_{k+1})$, while the observer Eq. (13) is integrated using the available state measurement to update $\hat{x}(t)$ and $\hat{\theta}(t)$ online.

To demonstrate the proposed offset-free LMPC, Fig. 2 compares its setpoint-tracking performance with that of the nominal LMPC under the time-varying parametric mismatch introduced in this study. Specifically, the true plant evolves with perturbed feed temperature and activation energy while the controller model remains nominal: $T_0(t)$ is ramped from 300 K to 290 K and $E(t)$ is ramped from 5.0×10^4 kJ kmol^{-1} to 5.15×10^4 kJ kmol^{-1} over the first 10 min and then held constant (transient-drift case). Under this mismatch, the nominal LMPC uses the fixed nominal prediction model Eq. (1), so its horizon predictions become biased relative to the disturbed plant Eq. (3); consequently, the closed-loop trajectories converge to a neighborhood with a visible steady-state offset, even though the Lyapunov constraint still enforces convergence to a Lyapunov level set. In contrast, the offset-free LMPC incorporates the disturbance estimate $\hat{\theta}(t_k)$ from the extended observer Eq. (13) into the prediction model via Eq. (18b) and updates the steady-state tracking input $u_{sp}(t_k)$ online via Eq. (14). As a result, both the horizon predictions and the tracking reference are corrected toward the disturbed equilibrium induced by the ramped $T_0(t)$ and $E(t)$, which improves steady-state accuracy and removes the offset, consistent with the trajectories in Fig. 2. This mechanism is further supported by Fig. 3, which shows that after an initial transient (during which the mismatch is being identified), the observer estimates converge and remain aligned with the actual disturbed process states.

As shown in Fig. 3, the real disturbed process states (red) and the offset-free observer estimates (blue) do not perfectly match at the beginning because the observer starts with limited information about the disturbance and the associated mismatch. During this transient period, the observer must correct the state estimate while also identifying the disturbance states, so a small estimation gap is expected. As the disturbance estimates settle, the observer compensates the mismatch more accurately. Consequently, the estimated trajectories align with the actual process states and remain overlapped afterward, indicating negligible steady-state estimation error.

5.4. Fast learned policy for approximating offset-free LMPC via FNN

To enable a computationally efficient surrogate of the constrained offset-free LMPC (OFLMPC), we first generate an offline imitation dataset by running closed-loop simulations of the OFLMPC under bounded disturbances and model mismatch, and then train a feedforward neural network (FNN) to approximate the resulting feedback law. The data-generation procedure follows the same OFLMPC closed-loop implementation described in Section 5.3: the disturbed plant evolves under the time-varying mismatch profile, the disturbance estimates are produced online by the extended Luenberger observer Eq. (13), and the OFLMPC control action is obtained by solving Eq. (18) at each sampling instant. Each closed-loop rollout is simulated for 30 min with sampling period $\Delta = 5$ s (sample-and-hold control over each interval) and numerical integration step $dt = 0.1$ s. The OFLMPC steady-state reference input $u_{sp}(t_k)$ required by the tracking objective is updated online at each sampling time using the current disturbance estimate via the equilibrium condition Eq. (14).

To cover the operating region, the initial deviation state $x(t_0)$ is sampled from randomized Lyapunov level-set rings of $V(x) = x^\top P x$ (with the same P used in the safety design). Specifically, we construct $N_{\text{ring}} = 200$ rings over $V \in [0.01, 200]$, select a ring index uniformly at random, and sample an initial point on the selected ring, yielding diverse initial conditions ranging from near-setpoint to outer-region states. For robustness, each rollout is generated using the true-process model Eq. (3) with additive disturbances $W(x,t) = \left[W_1(x,t),\ W_2(x,t)\right]^\top$ entering both Eq. (85a) and (85b). Over the operating region induced by the above initial-state set, we assume the disturbance bounds $|W_1(x,t)| \le 10, |W_2(x,t)| \le 500$, where $W_1(x,t)$ and $W_2(x,t)$ denote the additive disturbances in the concentration and temperature dynamics, respectively. In the data-generation simulations, a disturbance realization is sampled once per trajectory (uniformly within the above bounds) and then held constant over the entire 30 min trajectory simulation.

With this dataset, we train an FNN to approximate the OFLMPC feedback map. The measured/estimated learning state is $s = \left[C_A,\ T,\ \hat{\theta}_{C_A},\ \hat{\theta}_T\right]^\top$, and the corresponding OFLMPC control action is $u_{\text{LMPC}} = \left[C_{A0} - C_{A0s},\ \dot{Q} - \dot{Q}_s\right]^\top$ with bounds $C_{A0} - C_{A0s} \in [-3.5, 3.5]$ and $\dot{Q} - \dot{Q}_s \in [-3 \times 10^5, 3 \times 10^5]$. The dataset is formed by stacking all recorded trajectories and removing missing entries. Each state component is min–max normalized using dataset extrema (saved for later deployment), and each input is mapped to $[-1, 1]$ through an affine transform consistent with the actuator bounds. The FNN then learns a two-output map $\pi_{\text{FNN}} : s \mapsto u$ by minimizing the mean-squared error (MSE) between the predicted and OFLMPC actions in the normalized action space.

To obtain a compact yet accurate approximation, the network hyperparameters are selected via Bayesian optimization over the learning rate, batch size, network width, network depth, and dropout rate. After 30 Bayesian-optimization iterations (including 10 random initial trials), the best hyperparameter set is summarized in Table 2, achieving the lowest validation objective $\text{MSE}_{\text{mean,norm}} = 3.1681 \times 10^{-4}$ during the search. Using these hyperparameters, the final FNN is trained with early stopping (best epoch 22; training terminated at epoch 34).

Table 3 reports the resulting errors on the train, validation, and held-out test sets in both physical units and normalized units. The normalized mean MSE is $\text{MSE}_{\text{mean,n}} = 2.20 \times 10^{-4}$ on the test set, indicating that the learned policy closely matches the OFLMPC actions over the sampled operating region. In addition, Fig. 4 shows scatter plots comparing the FNN outputs against the OFLMPC targets on the test set for both inputs, where points concentrate near the 45 ° line, confirming accurate action replication.

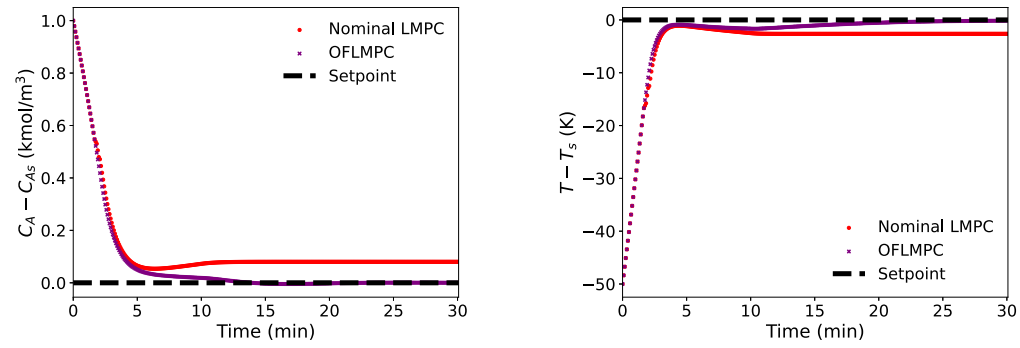
Table 2
Best hyperparameters used in offline FNN training.

Learning rate	Batch size	Network width	Network depth	Dropout rate
3.5948×10^{-3}	64	256	2	0.0236

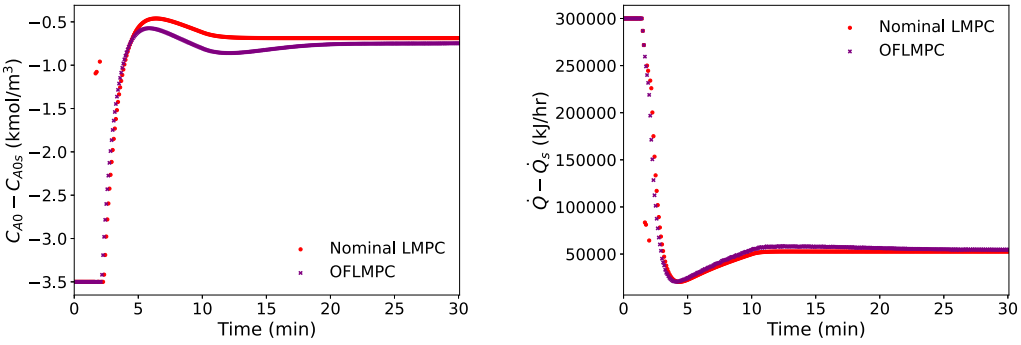
Table 3
FNN imitation accuracy for the offset-free LMPC policy (train/validation/test).

Split	$\text{MSE}_{C_{A0}}$	$\text{MSE}_{\dot{Q}}$	$\text{MSE}_{C_{A0},n}$	$\text{MSE}_{\dot{Q},n}$	$\text{MSE}_{\text{mean},n}$
Train	0.010748	7.0982×10^7	2.19×10^{-4}	1.97×10^{-4}	2.08×10^{-4}
Val	0.011731	7.2057×10^7	2.39×10^{-4}	2.00×10^{-4}	2.20×10^{-4}
Test	0.011375	7.4675×10^7	2.32×10^{-4}	2.07×10^{-4}	2.20×10^{-4}

Remark 21. At the start of data-generation, the disturbance/mismatch parameters are sampled randomly and then held constant over the entire trajectory, i.e., we train with a constant disturbance realization $W(t) \equiv W$ within each episode. This choice is intentional. Since the learning state includes the disturbance estimate $\hat{\theta}$, it is sufficient to

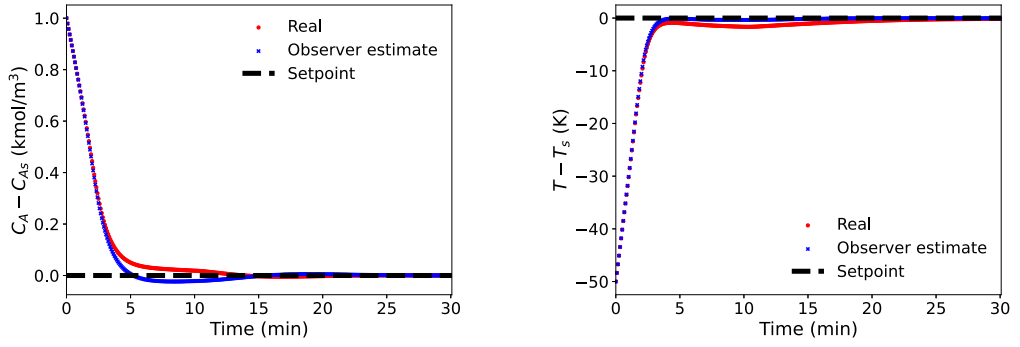


(a) Concentration deviation. (b) Temperature deviation.



(c) Feed concentration input deviation. (d) Heat-input deviation.

Fig. 2. Closed-loop trajectories comparing Nominal LMPC and offset-free LMPC (OFLMPC). The dashed line indicates the setpoint ($C_A = C_{As}$ and $T = T_s$).



(a) Concentration (b) Temperature

Fig. 3. Comparison between the states of the disturbed process and the offset-free observer estimates.

sample different constant disturbance realizations across trajectories to expose the observer to diverse mismatch patterns and train the network as a function of $\hat{\theta}$. If $W(t)$ were allowed to vary rapidly within a trajectory, the resulting time-varying targets could mislead the supervised imitation objective by mixing controller responses to exogenous fluctuations with the underlying $\hat{\theta}$-dependent feedback law. This is especially undesirable for subsequent RL training, where we rely on $\hat{\theta}$ to provide a consistent representation of mismatch and want the learned models/policies to reflect a stable mapping rather than chase nonstationary disturbances.

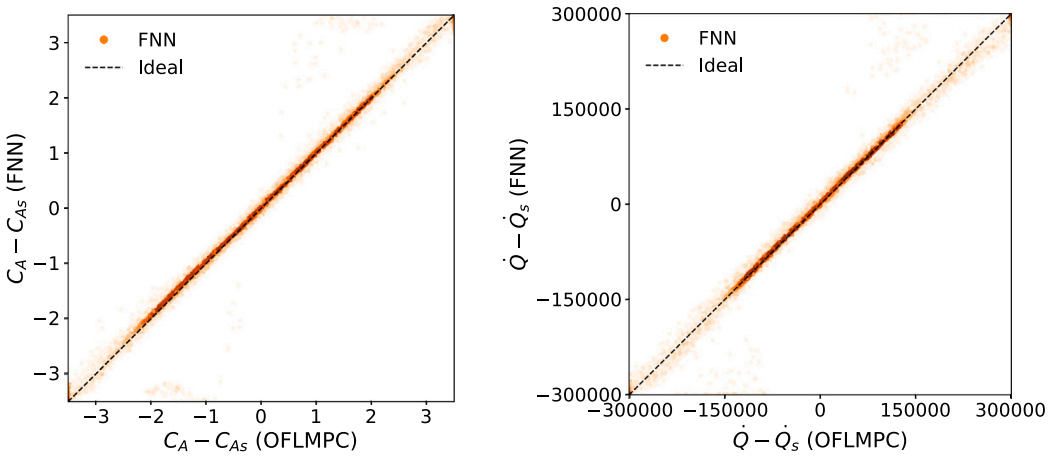
5.5. Robust HJB-RL policy

Following the robust HJB-RL formulation in Section 3.1 and the offset-free augmented model and observer in Section 2.5, we implement an offset-free HJB-RL controller for the unknown disturbed CSTR example. In particular, the value-based design is carried out on the augmented state $\hat{\chi} := \left[x^\top,\ \hat{\theta}^\top\right]^\top$.

The offset-free HJB-RL policy is trained in simulation using closed-loop rollouts of the disturbed CSTR together with the extended Luenberger observer. At each sampling time t_k, the RL observation is taken as the augmented estimate

$$s(t_k) = \hat{\chi}(t_k) := \begin{bmatrix} x(t_k) \\ \hat{\theta}(t_k) \end{bmatrix} = \begin{bmatrix} C_A(t_k) - C_{As} \\ T(t_k) - T_s \\ \hat{\theta}_{C_A}(t_k) \\ \hat{\theta}_T(t_k) \end{bmatrix} \tag{87}$$

where $x(t_k)$ denotes the measured plant deviation state and $\hat{\theta}(t_k)$ is the disturbance estimate generated by the observer in Eq. (13). The



(a) Comparison of $C_A - C_{As}$: OFLMPC targets vs. FNN outputs (test set).

(b) Comparison of $\dot{Q} - \dot{Q}_s$: OFLMPC targets vs. FNN outputs (test set).

Fig. 4. Test-set action replication performance of the FNN policy. Each point compares the OFLMPC action (data source) with the corresponding FNN prediction for the same input state.

control action is implemented under sample-and-hold with sampling period $\Delta = 5\,\mathrm{s}$, and both the plant and observer are integrated using explicit Euler with step size 0.1 s. Each training episode simulates a 30 min closed-loop trajectory.

To ensure coverage of both near-origin and far-from-origin regions, the initial deviation state $x(0)$ is sampled from Lyapunov level-set rings defined by $V(x) = x^\top P x$ using the matrix P specified in Section 5.2. The ring sampler spans $V \in [0.01, 200]$ using 200 uniform-width rings. To expose the learner to bounded mismatch consistent with the disturbance set used throughout this work, the real chemical process is perturbed by additive disturbances $W(x,t) = \left[W_1(x,t),\, W_2(x,t)\right]^\top$ satisfying $|W_1(x,t)| \le 10$ and $|W_2(x,t)| \le 500$, sampled once per episode and held constant during the episode. In contrast, consistent with Section 3.1, the HJB residual used for learning is evaluated using the offset-free prediction model $\bar{F}(\hat{\chi}, u)$, so the learner does not explicitly incorporate $W(x,t)$ in the model used for the critic update.

The critic $V_w(\hat{\chi})$ is parameterized by a fully-connected neural network with two hidden layers of width 256 and tanh activations. Its raw output is passed through a $\mathrm{softplus}(\cdot)$ nonlinearity and augmented with a small quadratic term in the input to improve numerical robustness

$$V_w(\hat{\chi}) = \mathrm{softplus}\big(f_w(\hat{\chi})\big) + \varepsilon_q \|\hat{\chi}\|_2^2, \qquad \varepsilon_q > 0 \tag{88}$$

Given a differentiable critic, the control action is computed from the stationarity condition of the Hamiltonian, consistent with Eq. (29), as a closed-form function of the gradient $\nabla_{\hat{\chi}} V_w(\hat{\chi})$ and then clipped to the admissible input set. Moreover, in accordance with the offset-free tracking construction in Section 2.5, the offset-free steady-state input $u_{sp}\big(\hat{\theta}(t_k)\big)$ is computed online from the latest disturbance estimate and used as the reference in both the policy computation and the stage cost. This yields an offset-free regulation objective consistent with the quadratic benchmarking objective in Eq. (86) and the cost-matching choice in Eq. (26)

The critic parameters are updated by minimizing the mean-squared stationary augmented HJB residual as introduced in Eq. (30). For a mini-batch $\{\hat{\chi}_i\}_{i=1}^N$, the residual is formed as

$$\mathcal{R}(\hat{\chi}_i, w) = L\big(x_i, u^\star(\hat{\chi}_i)\big) + \big(\nabla_{\hat{\chi}} V_w(\hat{\chi}_i)\big)^\top \bar{F}\big(\hat{\chi}_i, u^\star(\hat{\chi}_i)\big) \tag{89}$$

where $\bar{F}(\cdot)$ denotes the offset-free model used for residual evaluation and $u^\star(\hat{\chi}_i)$ is the HJB-induced action computed from $\nabla_{\hat{\chi}} V_w(\hat{\chi}_i)$ and clipped to satisfy the input bounds. The critic is optimized using the `Adam` optimizer with gradient clipping, together with a small anchoring regularization at the origin to stabilize learning near $\hat{\chi} = 0$

$$\mathcal{L}(w) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{R}(\hat{\chi}_i, w)^2 + \lambda_V\, V_w(0)^2 + \lambda_\nabla \|\nabla_{\hat{\chi}} V_w(0)\|_2^2, \qquad \lambda_V, \lambda_\nabla > 0 \tag{90}$$

A replay buffer is used to maintain a representative distribution of visited augmented states. At each interaction step, the transition data are stored and mini-batches are sampled uniformly from the buffer for stochastic gradient updates. Training begins with 2,000 random-action steps for warm-up, after which Gaussian exploration noise is added to the HJB-induced action and clipped to satisfy the input bounds. Unless otherwise stated, training uses a batch size of 128 and performs one critic update per environment step

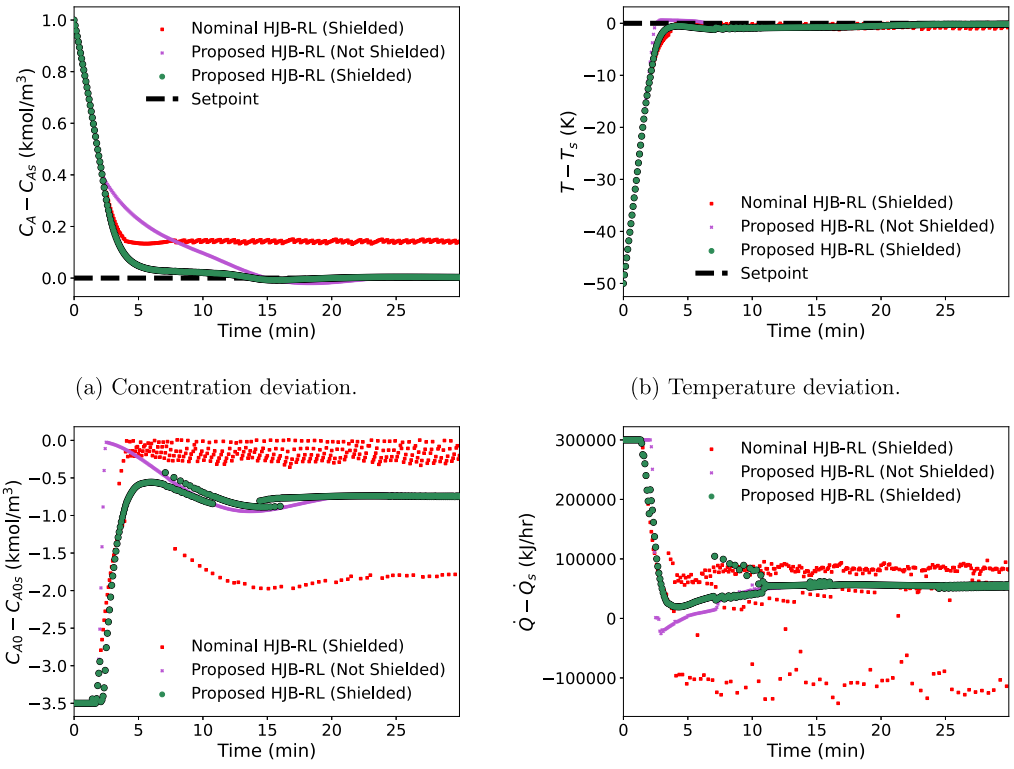
Policy evaluations are performed every 6,000 interaction steps on a fixed set of initial conditions sampled from rings with indices $\{10, 30, \ldots, 190\}$, where the evaluation disturbances are sampled within the same bounds as in training. The best-performing critic is selected based on the evaluation return and saved for subsequent closed-loop studies, and all training and evaluation statistics are logged continuously. The key hyperparameters used in the robust HJB-RL implementation for training the critic network $V_w(\hat{\chi})$ via the augmented HJB-residual minimization in Eq. (30) are summarized in Table 4.

To demonstrate the proposed offset-free HJB-RL (OFHJB-RL) design under the same parametric mismatch mechanism introduced in Section 5.2, Fig. 5 compares its closed-loop performance with two baselines: a nominal HJB-RL controller (trained/derived without offset-free compensation) and a deployment without the Lyapunov-based shield. In this comparison, the true plant evolves with perturbed feed temperature and activation energy while the controller model remains nominal; specifically, $T_0(t)$ is ramped from 300 K to 290 K and $E(t)$ is ramped from $5.0 \times 10^4\,\mathrm{kJ\,kmol^{-1}}$ to $5.15 \times 10^4\,\mathrm{kJ\,kmol^{-1}}$ over the first 10 min and then held constant, which corresponds to the transient-drift (Set 1) mismatch pattern described in Section 5.2.

As shown in Figs. 5(a) and 5(b), the proposed OFHJB-RL achieves the most reliable setpoint regulation: $C_A - C_{As}$ decays rapidly and remains tightly clustered near zero, and $T - T_s$ approaches the setpoint with limited dispersion across trajectories. The nominal HJB-RL exhibits a visible steady-state offset and a wider spread, indicating sensitivity to the biased plant dynamics induced by the thermal/kinetic mismatch. The input trajectories in Figs. 5(c) and 5(d) further reflect this difference: the nominal controller produces more scattered actions with larger excursions, whereas the proposed method yields

Table 4
Robust HJB-RL (value-critic) training hyperparameters.

Parameter	Value	Parameter	Value
Optimizer	`Adam`	Learning rate α_w	1×10^{-3}
Mini-batch size N	128	Critic hidden width ×depth	256×2
Critic activation	`tanh`	Output activation	`softplus`
ℓ_2 regularization weight	1×10^{-5}	Quadratic positivity term in V_w	$1 \times 10^{-4}\|\hat{x}\|_2^2$
Update frequency	1	Replay buffer capacity	5×10^5
Random exploration steps N_{rand}	2000	Exploration noise std σ_ϵ	0.10
Evaluation interval	6000	Total training steps	300 000



(a) Concentration deviation.

(b) Temperature deviation.

(c) Feed concentration input deviation.

(d) Heat-input deviation.

Fig. 5. Closed-loop trajectories comparing the proposed offset-free HJB-RL (OFHJB-RL), a nominal HJB-RL baseline, and the proposed controller deployed without the Lyapunov-based shield under the transient-drift mismatch profile (Set 1) generated by ramping $T_0(t)$ and $E(t)$ over the first 10 min and then holding them constant. The dashed line indicates the setpoint ($C_A = C_{As}$ and $T = T_s$).

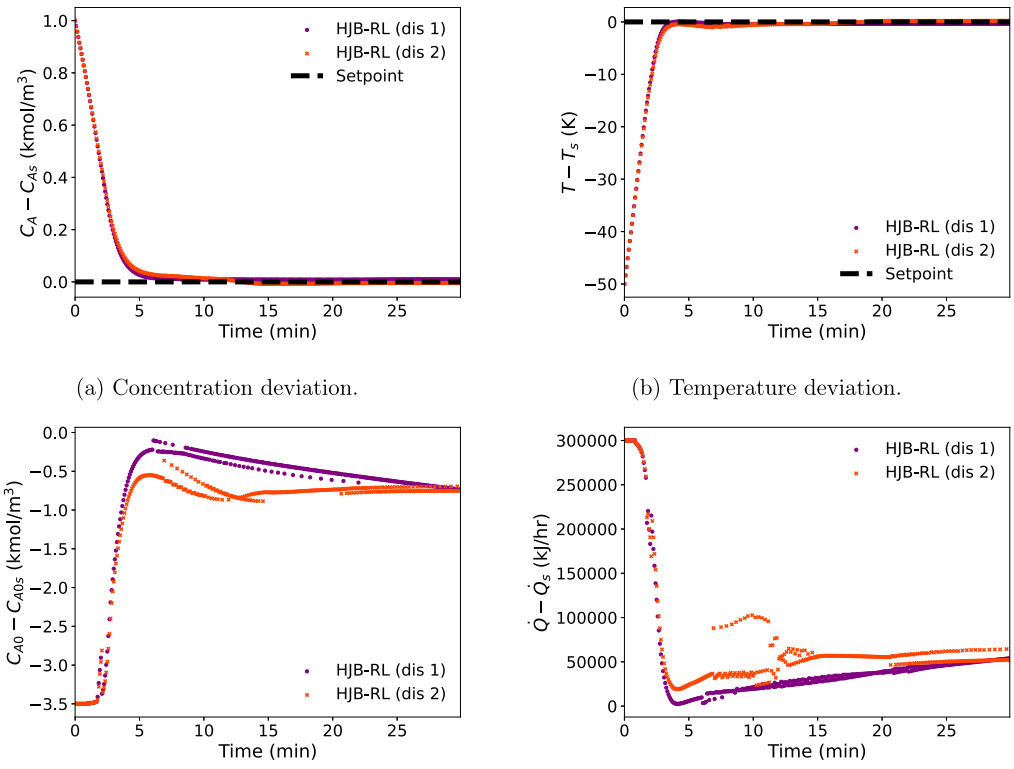
smoother and better-clustered inputs. Comparing the proposed method with the unshielded deployment highlights the role of the Lyapunov-based shield: removing the shield leads to less consistent transients and larger input excursions, even when convergence is still achieved for some trajectories.

Next, Fig. 6 evaluates the proposed OFHJB-RL under the two bounded mismatch profiles defined in Section 5.2. In dis 1 (Set 1), $T_0(t)$ and $E(t)$ are ramped over an initial period and then held constant, emulating a transient drift that settles. In dis 2 (Set 2), the perturbations start at $t = 0$ and vary continuously until the end of the horizon, reaching the prescribed bounds at the final time, which emulates a persistent drift that does not settle.

Despite the stronger time variation in dis 2, the proposed OFHJB-RL maintains reliable regulation in both channels. In Figs. 6(a) and 6(b), $C_A - C_{As}$ and $T - T_s$ converge rapidly and remain close to the setpoint for both mismatch profiles, without a visible steady-state offset even when the mismatch continues to evolve. The input trajectories in Figs. 6(c) and 6(d) are consistent with this behavior: in dis 1 the inputs settle after the drift ends, whereas in dis 2 they continue adjusting over time to track the changing plant conditions while remaining within the admissible bounds.

To complement the time-series comparisons, Fig. 7 visualizes the phase-plane behavior under the two mismatch sets. In both subfigures, the initial states are selected on the outer Lyapunov level set $V = 200$ by sampling the polar angle every 36° (10 starting points), and the dashed ellipses indicate the reference level sets $V = 20$ and $V = 200$.

The quantitative comparison in Table 5 supports the qualitative trends observed in the figures and indicates that the proposed design (offset-free augmentation with Lyapunov-based shielding) provides the most robust closed-loop tracking under mismatch. The proposed offset-free HJB-RL with the Lyapunov-based shield (Proposed (S)) achieves the best average reward in both sets (R = -1.277×10^3 for Set 1 and R = -1.266×10^3 for Set 2), outperforming the shielded nominal HJB-RL (Nominal (S)) and the unshielded deployment (Proposed (NS)). A comparison between Proposed (S) and Nominal (S) highlights the role of the offset-free augmentation: while both methods employ the same shield, Proposed (S) reduces the terminal offset O from 1.022×10^{-1} to 1.340×10^{-3} in Set 1 and from 8.990×10^{-2} to 7.557×10^{-3} in Set 2, lowering O from $\mathcal{O}(10^{-1})$ to $\mathcal{O}(10^{-3})$–$\mathcal{O}(10^{-2})$, which is consistent with offset-free compensation (via the augmented state with $\hat{\theta}$) removing steady-state bias under plant–model mismatch and disturbance drift.



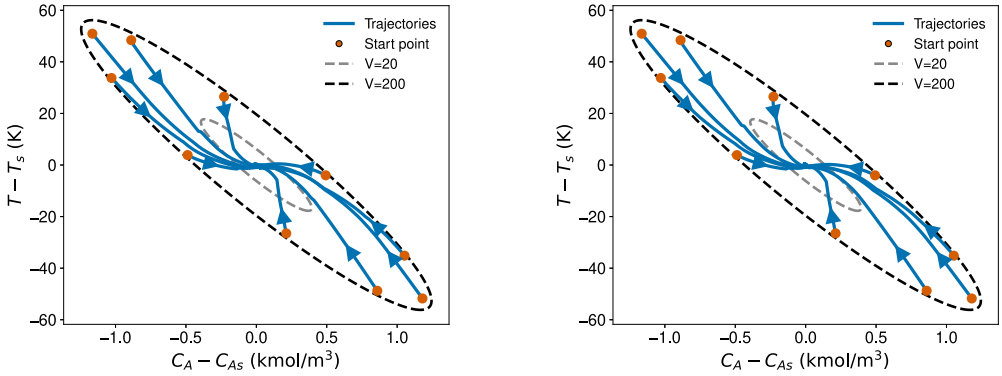
(a) Concentration deviation.

(b) Temperature deviation.

(c) Feed concentration input deviation.

(d) Heat-input deviation.

Fig. 6. Closed-loop trajectories of the proposed OFHJB-RL under two bounded time-varying parametric mismatch profiles implemented on the true plant via perturbed feed temperature $T_0(t)$ and activation energy $E(t)$ while keeping the controller model nominal. In dis 1 (Set 1), $T_0(t)$ and $E(t)$ are ramped over the first 10 min and then held constant (transient drift). In dis 2 (Set 2), the perturbations start at $t = 0$ and vary continuously until the end of the horizon, reaching the prescribed bounds at the final time (persistent drift). The dashed line indicates the setpoint ($C_A = C_{As}$ and $T = T_s$).



(a) Set 1: transient drift that settles after the initial period.

(b) Set 2: persistent drift that continues throughout the horizon.

Fig. 7. Phase-plane trajectories ($C_A - C_{As}$, $T - T_s$) of the closed-loop CSTR under the two bounded mismatch profiles. Initial conditions are chosen on $V(x) = 200$ by sampling the polar angle every 36° (10 starting points). The dashed ellipses show the Lyapunov level sets $V = 20$ and $V = 200$, and arrows indicate the trajectory direction under sample-and-hold control.

A comparison between Proposed (S) and Proposed (NS) highlights the role of the Lyapunov-based shield: enabling the shield improves the average reward (from -1.430×10^3 to -1.277×10^3 in Set 1 and from -1.422×10^3 to -1.266×10^3 in Set 2) and further reduces the terminal offset, especially in Set 2 (from 3.670×10^{-2} to 7.557×10^{-3}), consistent with rejecting nonconforming actions and invoking the LMPC fallback/failsafe to limit adverse transients and input excursions when the learned policy temporarily degrades.

5.6. Robust TD3-RL policy

Following the robust TD3-RL design described in Section 3.2, we implement a two-stage training procedure that first exploits the offline dataset $\mathcal{D}_{\text{off}}$ generated by the offset-free Lyapunov-contractive LMPC, and then continues with online interaction and off-policy updates under disturbances.

Table 5
Summary performance metrics for Set 1 and Set 2. Here, R denotes the average reward per trajectory, and O denotes the average terminal offset measured by the final Lyapunov value. The three HJB variants are: Proposed (S) = proposed offset-free HJB-RL with the Lyapunov-based shield; Nominal (S) = nominal HJB-RL with the shield; Proposed (NS) = proposed HJB-RL without the shield (no shield).

Set	Metric	Proposed (S)	Nominal (S)	Proposed (NS)
Set 1	R	-1.277×10^{3}	-1.329×10^{3}	-1.430×10^{3}
	O	1.340×10^{-3}	1.022×10^{-1}	8.400×10^{-3}
Set 2	R	-1.266×10^{3}	-1.319×10^{3}	-1.422×10^{3}
	O	7.557×10^{-3}	8.990×10^{-2}	3.670×10^{-2}

In the offline stage, we construct a large replay buffer from the stored transition tuples $\{(s_{t_k}, a_{t_k}, r_{t_k}, s_{t_{k+1}})\}$, where $s_{t_k} = [x(t_k)^\top, \hat{\theta}(t_k)^\top]^\top$ matches the augmented learning state used throughout this work. The action stored in the buffer is the applied LMPC input a_{t_k} (equivalently the deviation input $u(t_k)$ in Section 5.2), and the reward is computed consistently with Eq. (86) using the current disturbance estimate through $u_{sp}(\hat{\theta}(t_k))$. To improve numerical conditioning of neural network training, the augmented state is normalized componentwise to $s_{\mathrm{sc}} \in [0,1]^4$ using fixed min–max bounds, and the bounded physical deviation input $u(t_k) \in [U_{\min}, U_{\max}]$ is mapped to a normalized action $a(t_k) \in [-1,1]^2$ via an affine transform.

We first train a behavior-cloned teacher policy $\pi_{\mathrm{teach}}(s)$ from $\mathcal{D}_{\mathrm{off}}$ to approximate the LMPC feedback law, and we use this teacher to initialize the TD3 actor network. With the actor initialized, we pretrain the twin critic networks using only the offline buffer while keeping the actor fixed. This critic-only pretraining follows the standard TD3 target-network update with clipped target-action noise and uses the minimum of the two target critics to reduce Q-value overestimation. After critic-only pretraining, the actor can be optionally refined on $\mathcal{D}_{\mathrm{off}}$ using the TD3-BC objective described previously, i.e., a convex combination of the TD3 policy loss and the behavior-cloning regularizer with an α schedule. During online rollout, we additionally impose a trust-region clamp in the normalized action space to keep $\pi_\phi(s)$ within a fixed neighborhood of $\pi_{\mathrm{teach}}(s)$, which reduces the risk of abrupt policy deviation from the stabilizing LMPC-like behavior.

In the online pretraining stage, we create a new replay buffer and prefill it with a small number of transitions sampled from the offline reservoir buffer to avoid an empty-buffer transient. The controller then interacts with the disturbed plant in closed loop, where the plant evolves under the unknown disturbance realization while the offset-free observer runs online to update $\hat{\theta}(t_k)$ and thus the augmented state s_{t_k}. At each sampling instant, the actor receives the scaled observation $s_{\mathrm{sc}}(t_k)$, outputs $a(t_k) \in [-1,1]^2$, and applies the corresponding bounded physical deviation input $u(t_k)$ in a sample-and-hold fashion over the control interval. The collected online transitions are appended to the replay buffer, and the TD3-BC updates are performed off-policy using minibatches from this buffer, with critic updates at every iteration. We use a delayed actor update with period d_π, and apply Polyak averaging to update the actor target network only when the actor is updated. In contrast, the critic target networks are updated by Polyak averaging after each critic update.

To monitor training progress and select a deployable policy, we perform periodic fixed evaluations using a fixed set of initial conditions sampled from Lyapunov rings and disturbance realizations within the prescribed bounds. Each evaluation episode is simulated over the full horizon, and the model achieving the best average evaluation return is retained for the subsequent closed-loop comparisons with and without the proposed shield layer. The TD3-RL (TD3-BC) training hyperparameters are summarized in Table 6.

To demonstrate the proposed offset-free TD3-RL (OFTD3-RL) design under the same parametric mismatch mechanism introduced in Section 5.2, Fig. 8 compares its closed-loop performance with two baselines: a nominal TD3-RL controller (trained without offset-free compensation) and a deployment without the Lyapunov-based shield. In this comparison, the true plant evolves with perturbed feed temperature and activation energy while the controller model remains nominal; specifically, $T_0(t)$ is ramped from 300 K to 290 K and $E(t)$ is ramped from $5.0\times10^4\,\mathrm{kJ\,kmol^{-1}}$ to $5.15\times10^4\,\mathrm{kJ\,kmol^{-1}}$ over the first 10 min and then held constant, which corresponds to the transient-drift (Set 1) mismatch pattern described in Section 5.2.

As shown in Figs. 8(a) and 8(b), the proposed OFTD3-RL achieves the most reliable setpoint regulation: $C_A - C_{As}$ decays rapidly and remains tightly clustered near zero, and $T - T_s$ approaches the setpoint with limited dispersion across trajectories. In contrast, the nominal TD3-RL exhibits a larger spread and a visible steady-state bias under the biased plant dynamics induced by the thermal/kinetic mismatch. The input trajectories in Figs. 8(c) and 8(d)(c)–(d) further reflect this difference: the nominal policy produces more scattered actions with larger excursions, whereas the proposed method yields smoother and better-clustered inputs. Comparing the proposed method with the unshielded deployment highlights the role of the Lyapunov-based shield: removing the shield leads to substantially degraded transients and much larger terminal deviations, consistent with loss of reliability under the same mismatch.

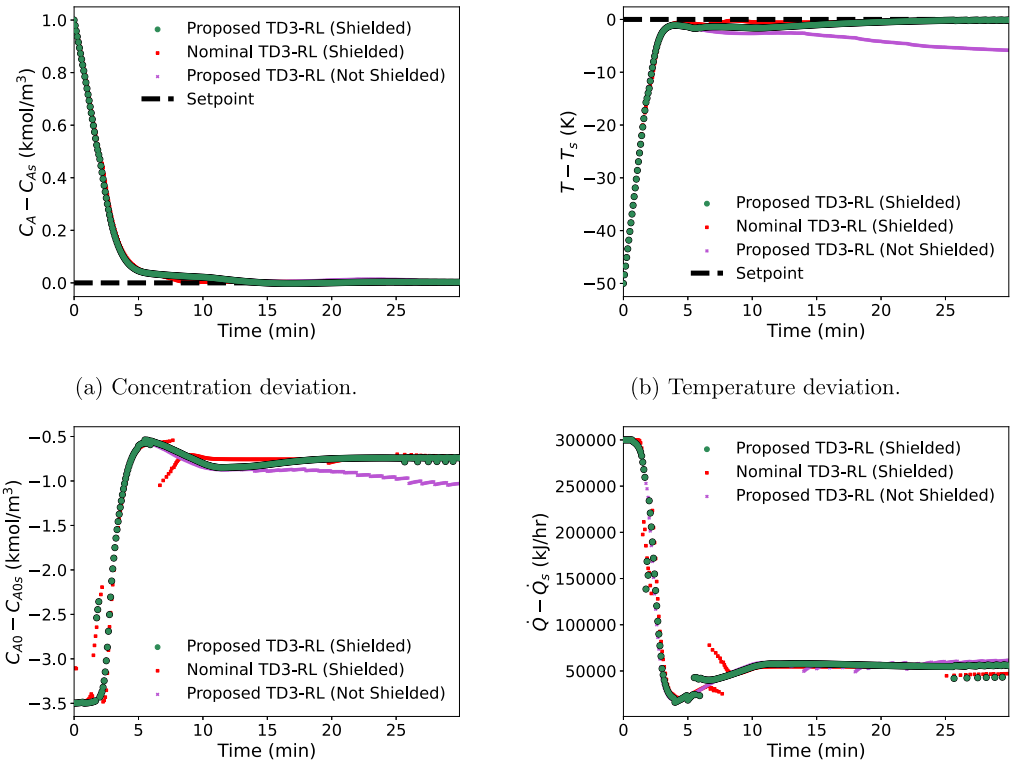
Next, Fig. 9 evaluates the proposed OFTD3-RL under the two bounded mismatch profiles defined in Section 5.2. In dis 1 (Set 1), $T_0(t)$ and $E(t)$ are ramped over an initial period and then held constant, emulating a transient drift that settles. In dis 2 (Set 2), the perturbations start at $t = 0$ and vary continuously until the end of the horizon, reaching the prescribed bounds at the final time, which emulates a persistent drift that does not settle. Despite the stronger time variation in dis 2, the proposed OFTD3-RL maintains reliable regulation in both channels: $C_A - C_{As}$ and $T - T_s$ converge rapidly and remain close to the setpoint without a visible steady-state offset. The input trajectories are consistent with this behavior: in dis 1 the inputs settle after the drift ends, whereas in dis 2 they continue adjusting over time to accommodate the evolving plant conditions while remaining within admissible bounds.

To complement the time-series comparisons, Fig. 10 visualizes the phase-plane behavior under the two mismatch sets. In both subfigures, the initial states are selected on the outer Lyapunov level set $V = 200$ by sampling the polar angle every 36° (10 starting points), and the dashed ellipses indicate the reference level sets $V = 20$ and $V = 200$. The trajectories move consistently toward the inner level set under sample-and-hold control, illustrating contraction toward the origin region in both mismatch profiles.

The quantitative comparison in Table 7 supports the qualitative trends observed in the figures and indicates that the proposed design (offset-free augmentation with Lyapunov-based shielding) provides the most robust closed-loop tracking under mismatch. The proposed offset-free TD3-RL with the Lyapunov-based shield (Proposed (S)) achieves the best average reward in both sets ($\mathrm{R} = -1.262\times10^3$ for Set 1 and $\mathrm{R} = -1.254\times10^3$ for Set 2), outperforming the shielded nominal TD3-RL (Nominal (S)) and the unshielded deployment (Proposed (NS)). A comparison between Proposed (S) and Nominal (S) highlights the role of the offset-free augmentation: while both methods employ the same shield, Proposed (S) reduces the terminal offset O from 6.785×10^{-2} to 3.918×10^{-3} in Set 1 and from 3.061×10^{-2} to 2.844×10^{-3} in Set 2, lowering O from $\mathcal{O}(10^{-2})$ to $\mathcal{O}(10^{-3})$, which is consistent with offset-free compensation (via the augmented state with $\hat{\theta}$) removing steady-state bias under plant–model mismatch and disturbance drift. A comparison between Proposed (S) and Proposed (NS) highlights the role of the Lyapunov-based shield: enabling the shield improves the average reward (from -1.623×10^3 to -1.262×10^3 in Set 1 and from -1.641×10^3 to -1.254×10^3 in Set 2) and, more importantly, prevents the severe loss of reliability observed without shielding by reducing the terminal offset from $\mathcal{O}(10^1)$ (i.e., 1.606×10^1 in Set 1 and 1.754×10^1 in Set 2) to $\mathcal{O}(10^{-3})$, consistent with rejecting nonconforming actions and invoking the LMPC fallback/failsafe to limit adverse transients and input excursions.

Table 6
TD3-RL (TD3-BC) training hyperparameters.

Parameter	Value	Parameter	Value
Discount factor γ	0.99	Target update rate τ	0.005
Policy update delay d	2	Target smoothing noise std σ_{policy}	0.03
Target noise clip c_{noise}	0.08	Minibatch size	256
Actor learning rate	1×10^{-4}	Critic learning rate	1×10^{-4}
Critic network size (width, depth)	(256, 2)	Exploration noise std σ_{expl}	0.03
TD3-BC weight α (start → end)	$1.0 \rightarrow 0.70$	α schedule length (actor updates)	60 000
Actor warmup steps	12 000	Trust-region radius Δ_a	0.20



(a) Concentration deviation.

(b) Temperature deviation.

(c) Feed concentration input deviation.

(d) Heat-input deviation.

Fig. 8. Closed-loop trajectories comparing the proposed offset-free TD3-RL (OFTD3-RL), a nominal TD3-RL baseline, and the proposed controller deployed without the Lyapunov-based shield under the transient-drift mismatch profile (Set 1) generated by ramping $T_0(t)$ and $E(t)$ over the first 10 min and then holding them constant. The dashed line indicates the setpoint ($C_A = C_{As}$ and $T = T_s$).

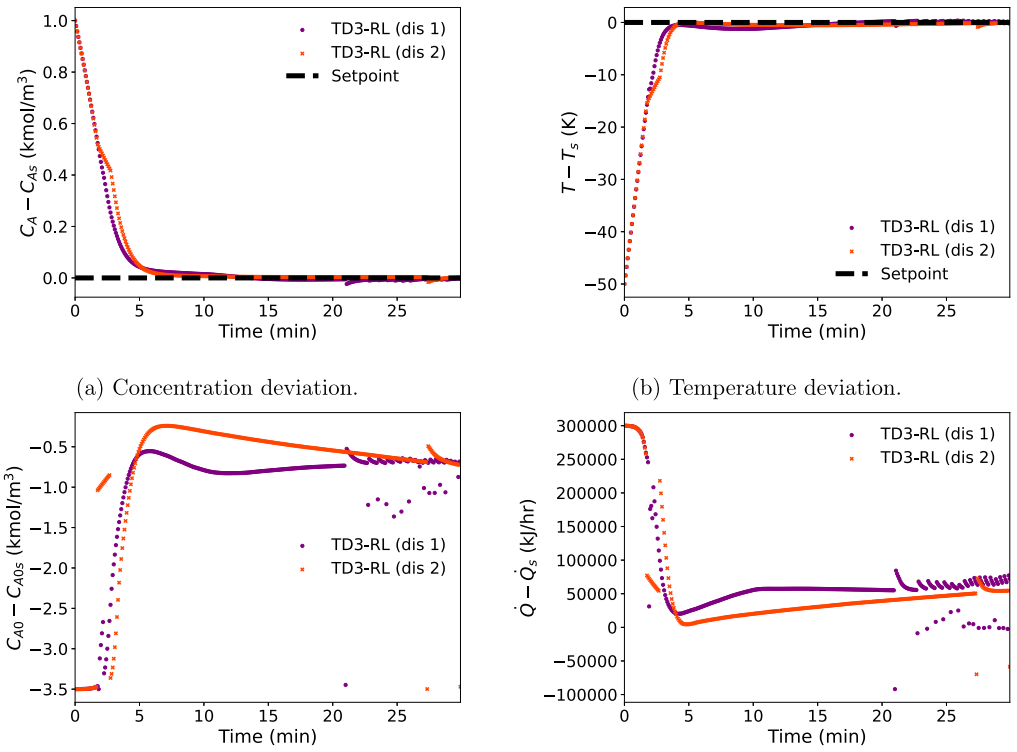
Table 7
Summary performance metrics for Set 1 and Set 2. Here, R denotes the average reward per trajectory, and O denotes the average terminal offset measured by the final Lyapunov value. The three TD3 variants are: Proposed (S) = proposed offset-free TD3-RL with the Lyapunov-based shield; Nominal (S) = nominal TD3-RL with the shield; Proposed (NS) = proposed TD3-RL without the shield (no shield).

Set	Metric	Proposed (S)	Nominal (S)	Proposed (NS)
Set 1	R	-1.262×10^3	-1.289×10^3	-1.623×10^3
	O	3.918×10^{-3}	6.785×10^{-2}	1.606×10^1
Set 2	R	-1.254×10^3	-1.273×10^3	-1.641×10^3
	O	2.844×10^{-3}	3.061×10^{-2}	1.754×10^1

5.7. Comparison of different controllers

In this part, we compare the closed-loop performance of the proposed RL-based controllers with two MPC-based baselines: the offset-free LMPC (discussed in Section 5.3) implemented with different prediction horizons and an FNN controller that approximates the offset-free LMPC policy (Section 5.4).

First, all controllers are tested under the same plant-side mismatch profile as in Section 5.3. In the true plant, the feed temperature is changed from 300 K to 290 K and the activation energy is increased from 5.0×10^4 kJ kmol^{-1} to 5.15×10^4 kJ kmol^{-1} by a linear ramp over the first 10 min, and then both parameters are kept constant for the rest of the simulation, while the controller model uses the nominal parameters. The long-horizon offset-free LMPC (LH-OFLMPC) is used as the reference because it solves the offset-free Lyapunov-based MPC problem with a longer prediction horizon at every sampling instant. The short-horizon offset-free LMPC (SH-OFLMPC) uses the same offset-free formulation (the same observer, the same online update of $u_{\text{sp}}(t_k)$, and the same Lyapunov-based constraint), but it solves the optimization with a shorter horizon at every sampling instant throughout the closed loop. The FNN controller removes online optimization by directly outputting the control action from a learned approximation of the offset-free LMPC policy. As shown in Fig. 11, both proposed RL controllers (TD3-RL and HJB-RL) produce closed-loop trajectories that closely match the LH-OFLMPC reference under the same disturbance, suggesting that the learned policies achieve performance comparable to the optimization-based baseline while keeping the online computation small.



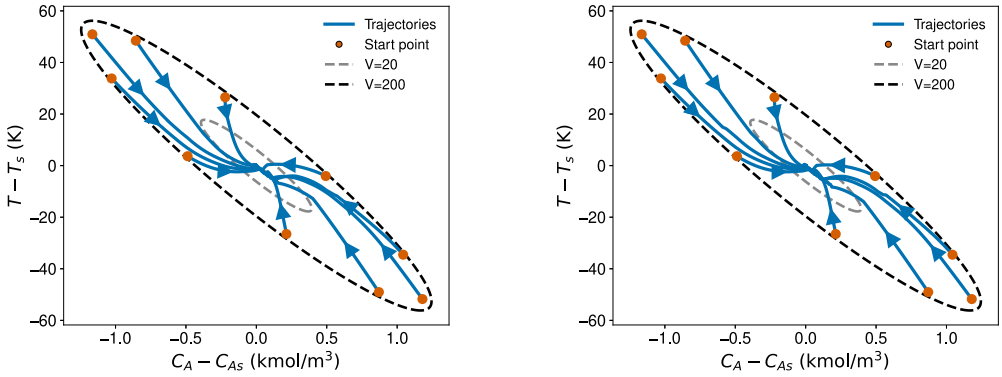
(a) Concentration deviation.

(b) Temperature deviation.

(c) Feed concentration input deviation.

(d) Heat-input deviation.

Fig. 9. Closed-loop trajectories of the proposed OFTD3-RL under two bounded time-varying parametric mismatch profiles implemented on the true plant via perturbed feed temperature $T_0(t)$ and activation energy $E(t)$ while keeping the controller model nominal. In dis 1 (Set 1), $T_0(t)$ and $E(t)$ are ramped over the first 10 min and then held constant (transient drift). In dis 2 (Set 2), the perturbations start at $t = 0$ and vary continuously until the end of the horizon, reaching the prescribed bounds at the final time (persistent drift). The dashed line indicates the setpoint ($C_A = C_{As}$ and $T = T_s$).



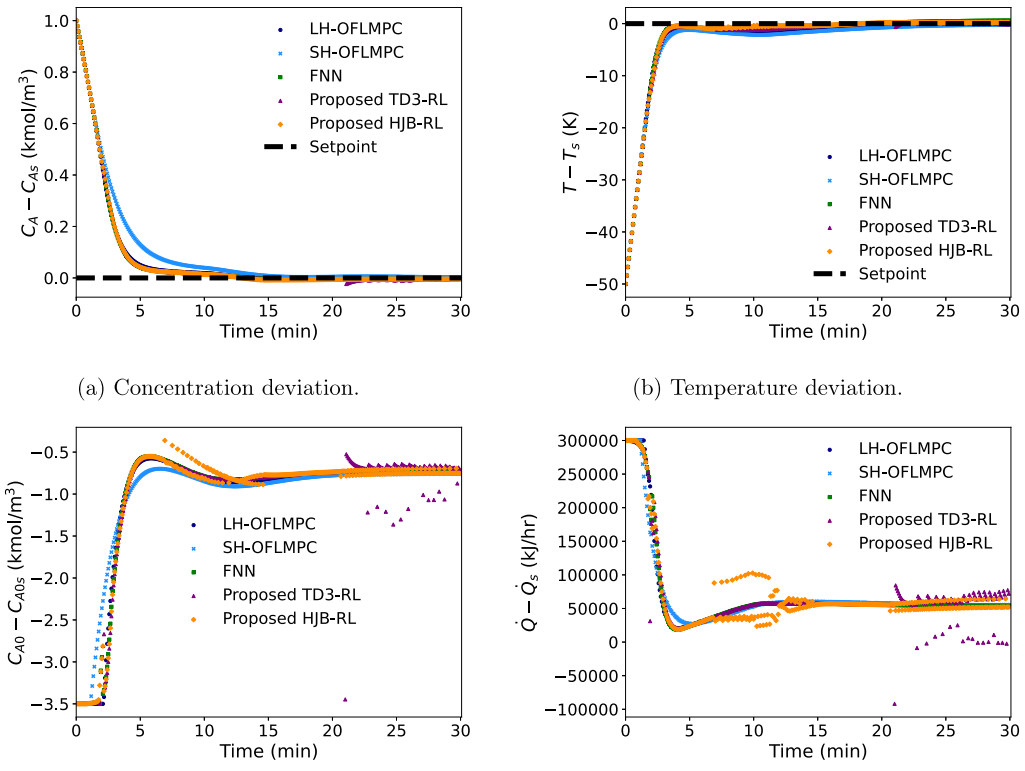
(a) Set 1: transient drift that settles after the initial period.

(b) Set 2: persistent drift that continues throughout the horizon.

Fig. 10. Phase-plane trajectories ($C_A - C_{As}$, $T - T_s$) of the closed-loop CSTR under the two bounded mismatch profiles. Initial conditions are chosen on $V(x) = 200$ by sampling the polar angle every 36° (10 starting points). The dashed ellipses show the Lyapunov level sets $V = 20$ and $V = 200$, and arrows indicate the trajectory direction under sample-and-hold control.

We then evaluate robustness and runtime cost under *two* disturbance sets that remain within the admissible ranges defined in Section 5.2. Set 1 represents a drift that ramps during an initial period and then becomes constant, and Set 2 represents a drift that evolves throughout the full horizon and reaches its bounds at the final time. These two sets are used for the computation-time and the setpoint-tracking performance comparison reported below.

Fig. 12 and Table 8 together show that the proposed RL controllers (TD3-RL and HJB-RL) provide the strongest overall trade-off between real-time execution and setpoint tracking under both disturbance sets: TD3-RL achieves the best reward in Set 1 and Set 2 ($R = -1.262 \times 10^3$ and $R = -1.254 \times 10^3$), while both TD3-RL and HJB-RL keep the terminal offset small at the 10^{-3} level ($O = 3.918 \times 10^{-3}$ and $O = 1.340 \times 10^{-3}$ in Set 1, and $O = 2.844 \times 10^{-3}$ and $O = 7.557 \times 10^{-3}$ in Set 2), and their computation times remain well below the 5 s limit with millisecond-level mean times (about 7.6–17 ms in Set 1 and 8.4–15.4 ms in Set 2) and sub-second to ~1.2 s worst-case times across both sets; in contrast, SH-LMPC stays within the time budget (maximum time below 5 s) but suffers a clear performance penalty with the lowest rewards and a much larger offset in Set 2 ($O = 4.125 \times 10^{-2}$), supporting its use mainly as



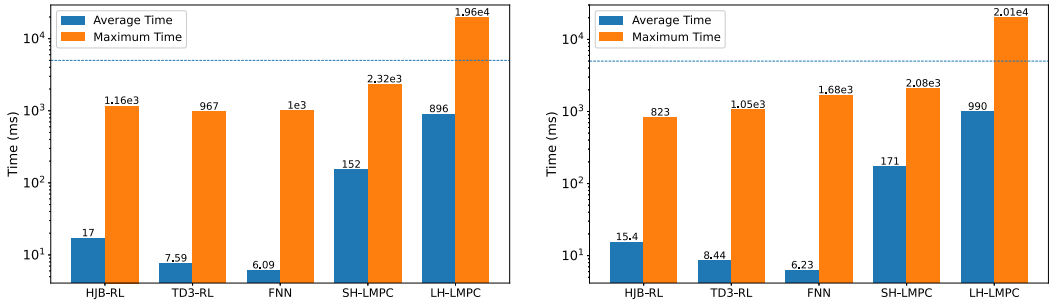
(a) Concentration deviation.

(b) Temperature deviation.

(c) Feed concentration input deviation.

(d) Heat-input deviation.

Fig. 11. Closed-loop trajectories of different controllers under the same transient-drift mismatch profile (plant-side $T_0(t)$ ramped from 300 K to 290 K and $E(t)$ ramped from 5.0×10^4 kJ kmol^{-1} to 5.15×10^4 kJ kmol^{-1} over the first 10 min, then held constant; controller model remains nominal). The dashed line indicates the setpoint ($C_A = C_{As}$ and $T = T_s$).



(a) Set 1.

(b) Set 2.

Fig. 12. Per-step computation time of each controller for Set 1 and Set 2. Blue/orange bars show the mean and the maximum computation time, respectively, for HJB-RL, TD3-RL, FNN, SH-LMPC, and LH-LMPC. The dashed horizontal line indicates the sampling-time limit (5 s = 5,000 ms).

Table 8
Summary performance metrics for Set 1 and Set 2. Here, R denotes the average reward per trajectory, and O denotes the average terminal offset measured by the final Lyapunov value.

Set	Metric	HJB-RL	TD3-RL	FNN	SH-LMPC	LH-LMPC
Set 1	R	-1.277×10^3	-1.262×10^3	-1.281×10^3	-1.395×10^3	-1.286×10^3
	O	1.340×10^{-3}	3.918×10^{-3}	5.686×10^{-3}	1.001×10^{-2}	1.792×10^{-3}
Set 2	R	-1.266×10^3	-1.254×10^3	-1.278×10^3	-1.381×10^3	-1.285×10^3
	O	7.557×10^{-3}	2.844×10^{-3}	1.137×10^{-2}	4.125×10^{-2}	1.423×10^{-2}

a backup, whereas LH-LMPC can deliver competitive tracking in some cases (e.g., Set 1 O $= 1.792 \times 10^{-3}$) but exhibits large computation-time spikes that exceed the 5 s limit and also a larger Set 2 offset ($O = 1.423 \times 10^{-2}$), making it less suitable when strict per-step real-time constraints must be enforced.

Remark 22. At each sampling instant t_k, the reported computational time corresponds to the wall-clock time required to compute and return the control input that is actually applied to the plant (i.e., policy evaluation and, when activated, the back-up controller computation). The online RL training routine is executed asynchronously in parallel

and does not block the computation of the applied control action; therefore, its runtime is not included in the per-step computational time statistics. Hence, the computational-time metric is intended to reflect real-time implementability under a fixed sampling period, rather than the total CPU/GPU usage associated with background learning.

6. Conclusion

In this study, we proposed a stability- and robustness-oriented RL framework for nonlinear constrained process control by combining a Lyapunov-based shield with an offset-free design inspired by MPC. The RL policy is treated as a candidate controller and is applied only when a Lyapunov condition is satisfied; otherwise, the control action is replaced by a Lyapunov-designed fallback controller, so the implemented input follows the stability requirement at every sampling instant while still allowing RL to improve performance whenever it is safe. To address steady-state offsets and model-plant mismatch, the learning state is augmented with online-estimated disturbance/mismatch variables, enabling the RL policy/value function to adapt its decisions to the current uncertainty level. We demonstrated the framework using two representative RL methods, namely an HJB-based value-critic approach and a TD3-based actor-critic approach, and showed that the resulting RL-based controllers handle different disturbance scenarios more effectively than conventional RL designs while maintaining competitive setpoint tracking and online computational cost relative to advanced baseline controllers. Overall, this work provides a practical path toward deploying RL in nonlinear process control with explicit stability guarantees and improved robustness to uncertainty.

CRediT authorship contribution statement

Xiaodong Cui: Writing – original draft, Methodology, Investigation, Conceptualization. **Arthur Khodaverdian:** Writing – original draft, Methodology, Investigation, Conceptualization. **Panagiotis D. Christofides:** Writing – original draft, Methodology, Investigation, Conceptualization.

Declaration of competing interest

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

Acknowledgments

Financial support from the National Science Foundation, United States, CBET-2140506, is gratefully acknowledged.

Appendix. Alternative offset-free Lyapunov constraint

We offer an offset-free Lyapunov constraint that is enforced not only in the outer region (where an LMPC backup is invoked), but also inside the inner region where the backup would otherwise be inactive. The key idea is that the offset-free model replaces the unknown mismatch by an online estimate, so that the residual mismatch entering $\dot{V}$ can be made small (and even vanishing for constant mismatch), leading to a strictly smaller sample-and-hold ultimate bound on $\|x(t)\|$.

Let the true process evolve under sample-and-hold control with sampling period $\Delta > 0$:

$$\dot{x}(t) = F\left(x(t), u(t_k)\right) + W\left(x(t), t\right), \qquad t \in [t_k, t_k + \Delta), \qquad u(t_k) \in U \tag{A.1}$$

where $U \subset \mathbb{R}^m$ is the admissible input set in Eq. (2). Fix $\rho > 0$ and define the Lyapunov sublevel set $\Omega_\rho := \{x \in \mathbb{R}^n \mid V(x) \le \rho\} \subset D$. Throughout, assume Section 2.3 holds on Ω_ρ with constants c_1, c_2, c_3, c_4, and the disturbance satisfies $\|W(x,t)\| \le W_{\max}$ on $\Omega_\rho \times \mathbb{R}$.

Assume the matched structured mismatch model

$$W(x,t) = G_\theta\, \theta^\star(t), \qquad G_\theta \in \mathbb{R}^{n\times p}, \quad \theta^\star(t) \in \mathbb{R}^p \tag{A.2}$$

and let the extended observer Eq. (13) provide $\hat{\theta}(t)$. At each sampling instant t_k, define the frozen offset-free injection and the corresponding residual mismatch on $[t_k, t_k + \Delta)$ as

$$\hat{w}_k := G_\theta \hat{\theta}(t_k) \in \mathbb{R}^n, \qquad \tilde{w}_k(t) := W\left(x(t), t\right) - \hat{w}_k = G_\theta \left[\theta^\star(t) - \hat{\theta}(t_k)\right] \tag{A.3}$$

Denote the residual magnitude on the interval by

$$\bar{W}_k := \sup_{t\in[t_k, t_k+\Delta)} \|\tilde{w}_k(t)\| \tag{A.4}$$

Then, along Eq. (A.1), the Lyapunov derivative admits the decomposition

$$\begin{aligned} \dot{V}\left(x(t), u(t_k)\right) &= \frac{\partial V(x(t))}{\partial x}\left[F\left(x(t), u(t_k)\right) + \hat{w}_k\right] + \frac{\partial V(x(t))}{\partial x}\tilde{w}_k(t) \\ &=: \dot{V}_{\text{of}}\left(x(t), u(t_k)\right) + \frac{\partial V(x(t))}{\partial x}\tilde{w}_k(t) \end{aligned} \tag{A.5}$$

Using Eqs. (7c) and (A.4), for all $t \in [t_k, t_k + \Delta)$,

$$\dot{V}\left(x(t), u(t_k)\right) \le \dot{V}_{\text{of}}\left(x(t), u(t_k)\right) + c_4 \|x(t)\| \bar{W}_k \tag{A.6}$$

Let ρ_s and ρ_{sw} be the outer-region threshold and the switching level used in the main sample-and-hold analysis, and define the inner level

$$\rho_{\text{in}} := \max\{\rho_s, \rho_{\text{sw}}\} \tag{A.7}$$

For any $\eta \in (0, \rho)$, define the following one-step worst-case Lyapunov *inflation map*

$$\mathcal{R}_{\min}(\eta) := \sup\left\{ V(x(t)) \;\middle|\; \begin{array}{l} x(t_k) \in \Omega_\eta, \quad u(t_k) \in U, \quad W \in \mathcal{W} \\ \text{Eq. (A.1) holds on } [t_k, t_k + \Delta) \end{array} \right\} \tag{A.8}$$

Fix $\eta = \rho_{\text{in}}$ and define the constant

$$\rho^\star_{\min} := \mathcal{R}_{\min}(\rho_{\text{in}}) \tag{A.9}$$

In the earlier analysis (without an inner-region Lyapunov constraint), once the trajectory reaches $\Omega_{\rho_{\text{in}}}$, the invariance argument yields an ultimate bound of the form

$$V(x(t)) \le \rho^\star_{\min}, \qquad \|x(t)\| \le \sqrt{\rho^\star_{\min}/c_1}, \qquad \forall t \ge t_1 \tag{A.10}$$

for some t_1.

We now impose an additional offset-free Lyapunov constraint in the inner region. Specifically, choose $\alpha > 0$ (e.g., $\alpha = c_3/c_2$), and require that for every sampling instant t_k with $x(t_k) \in \Omega_{\rho_{\text{in}}}$, the applied input $u(t_k) \in U$ satisfies

$$\dot{V}_{\text{of}}\left(x(t_k), u(t_k)\right) \le -\alpha V\left(x(t_k)\right) \tag{A.11}$$

To relate Eq. (A.11) to $\dot{V}(x(t), u(t_k))$ for $t \in [t_k, t_k + \Delta)$, boundedness of F on $\Omega_\rho \times U$ and $\|W\| \le W_{\max}$ imply the increment bound

$$\|x(t) - x(t_k)\| \le \int_{t_k}^{t} \|\dot{x}(\tau)\|\, d\tau \le \left(M_F + W_{\max}\right)\Delta, \qquad t \in [t_k, t_k + \Delta) \tag{A.12}$$

Using Eq. (8c) together with Eq. (A.12) yields

$$\dot{V}_{\text{of}}\left(x(t), u(t_k)\right) \le \dot{V}_{\text{of}}\left(x(t_k), u(t_k)\right) + L'_x\left(M_F + W_{\max}\right)\Delta, \qquad t \in [t_k, t_k + \Delta) \tag{A.13}$$

Moreover, if $x(t_k) \in \Omega_{\rho_{\text{in}}}$, then $\|x(t_k)\| \le \sqrt{\rho_{\text{in}}/c_1}$ and Eq. (A.12) implies

$$\|x(t)\| \le \sqrt{\frac{\rho_{\text{in}}}{c_1}} + \left(M_F + W_{\max}\right)\Delta, \qquad t \in [t_k, t_k + \Delta) \tag{A.14}$$

Combining Eqs. (A.6), (A.11), (A.13) and (A.14), for any t_k with $x(t_k) \in \Omega_{\rho_{\text{in}}}$ and all $t \in [t_k, t_k + \Delta)$,

$$\dot{V}\left(x(t), u(t_k)\right) \le -\alpha V\left(x(t_k)\right) + L'_x\left(M_F + W_{\max}\right)\Delta$$

$$+\ c_4\left(\sqrt{\frac{\rho_{\text{in}}}{c_1}}+\left(M_F+W_{\max}\right)\Delta\right)\bar{W}_k \tag{A.15}$$

For readability, define the two (deterministic) sampling-error coefficients

$$B_0\ :=\ L_x'\left(M_F+W_{\max}\right)\Delta,\qquad B_1\ :=\ c_4\left(\sqrt{\frac{\rho_{\text{in}}}{c_1}}+\left(M_F+W_{\max}\right)\Delta\right) \tag{A.16}$$

Then Eq. (A.15) is equivalently

$$\dot{V}\left(x(t),u(t_k)\right)\le -\alpha V\left(x(t_k)\right)+B_0+B_1\,\bar{W}_k,\qquad t\in[t_k,t_k+\Delta) \tag{A.17}$$

Define the residual-dependent inner-level *map*

$$\mathcal{I}_{\text{in}}^{\text{of}}(w)\ :=\ \max\left\{\rho_{\text{in}},\ \frac{1}{\alpha}\left[B_0+B_1\,w\right]\right\},\qquad w\ge 0 \tag{A.18}$$

Fix a constant $\bar{W}\ge 0$ and define the (constant) offset-free inner level

$$\bar{\rho}_{\text{in}}^{\text{of}}\ :=\ \mathcal{I}_{\text{in}}^{\text{of}}(\bar{W}) \tag{A.19}$$

If $\bar{W}_k\le\bar{W}$ holds for all sufficiently large k (i.e., after some transient), then for those k and all $t\in[t_k,t_k+\Delta)$,

$$\dot{V}\left(x(t),u(t_k)\right)\le -\alpha\left[V\left(x(t_k)\right)-\bar{\rho}_{\text{in}}^{\text{of}}\right] \tag{A.20}$$

In particular, if $V(x(t_k))\ge\bar{\rho}_{\text{in}}^{\text{of}}+\eta$ for some $\eta>0$, then

$$\dot{V}\left(x(t),u(t_k)\right)\le -\alpha\eta\ =:\ -\varepsilon\quad\text{for all } t\in[t_k,t_k+\Delta) \tag{A.21}$$

Consequently, there exists k_2 such that $V\left(x(t_k)\right)\le\bar{\rho}_{\text{in}}^{\text{of}}$ holds for all $k\ge k_2$.

To translate this sampling-time bound into a continuous-time ultimate bound without introducing an explicit inflation map, define the offset-free one-step inflation *map*

$$\mathcal{R}_{\min}^{\text{of}}(w)\ :=\ \sup\left\{V(x(t))\ \middle|\ \begin{array}{l}x(t_k)\in\Omega_{\mathcal{I}_{\text{in}}^{\text{of}}(w)},\quad u(t_k)\in U\ \text{satisfies Eq. (A.11)},\\ \bar{W}_k\le w,\quad \text{Eq. (A.1) holds on } [t_k,t_k+\Delta)\end{array}\right\} \tag{A.22}$$

Fix the same constant $\bar{W}$ and define

$$\rho_{\min}^{\text{of}}\ :=\ \mathcal{R}_{\min}^{\text{of}}(\bar{W}) \tag{A.23}$$

Hence there exists $t_2\ge t_0$ such that

$$V(x(t))\le\rho_{\min}^{\text{of}},\qquad\forall t\ge t_2 \tag{A.24}$$

and by Eq. (7a) the corresponding sample-and-hold ultimate state ball is

$$\|x(t)\|\le\sqrt{\rho_{\min}^{\text{of}}/c_1},\qquad\forall t\ge t_2 \tag{A.25}$$

Moreover, the new ultimate level is no larger than the raw-disturbance inflation level. Indeed, $\rho_{\min}^{\star}$ in Eq. (A.9) is generated by the full disturbance class $W\in\mathcal{W}$ (with $\|W\|\le W_{\max}$), whereas $\rho_{\min}^{\text{of}}$ in Eq. (A.23) restricts admissible closed-loop trajectories by enforcing Eq. (A.11) and by replacing the unknown mismatch contribution in $\dot{V}$ with the residual bound $\bar{W}_k\le\bar{W}$. Therefore the optimization set underlying $\mathcal{R}_{\min}^{\text{of}}(\bar{W})$ is a subset of that underlying $\mathcal{R}_{\min}(\rho_{\text{in}})$, and

$$\rho_{\min}^{\text{of}}\ \le\ \rho_{\min}^{\star} \tag{A.26}$$

When $\bar{W}\ll W_{\max}$ (and Δ is sufficiently small so that the residual term dominates the inflation mechanism), The inequality in Eq. (A.26) is typically strict, yielding a strictly smaller ultimate set.

In the vanishing-residual limit $\bar{W}\to 0$, recalling $\bar{\rho}_{\text{in}}^{\text{of}}=\mathcal{I}_{\text{in}}^{\text{of}}(\bar{W})$, we obtain

$$\bar{\rho}_{\text{in}}^{\text{of}}\to\max\left\{\rho_{\text{in}},\ \frac{B_0}{\alpha}\right\}=\max\left\{\rho_{\text{in}},\ \frac{L_x'(M_F+W_{\max})\Delta}{\alpha}\right\} \tag{A.27}$$

and $\rho_{\min}^{\text{of}}$ approaches the one-step inflated level associated with the compensated (offset-free) model rather than that associated with the raw disturbance bound $W_{\max}$.

Two disturbance cases are of interest.

If $\dot{\theta}^{\star}(t)\equiv 0$ (constant mismatch), then Theorem 4(i) implies $\hat{\theta}(t)\to\theta^{\star}$ and thus $\bar{W}_k\to 0$. Therefore, for any $\varepsilon>0$ there exists k_ε such that $\bar{W}_k\le\varepsilon$ for all $k\ge k_\varepsilon$; picking $\bar{W}=\varepsilon$ yields the ultimate state ball Eq. (A.25) with $\rho_{\min}^{\text{of}}=\mathcal{R}_{\min}^{\text{of}}(\varepsilon)$.

If $\|\dot{\theta}^{\star}(t)\|\le d_{\max}$ (time-varying bounded mismatch), then Theorem 4(ii) yields an ultimate bound on the estimation error, and Eqs. (A.3) and (A.4) give

$$\limsup_{k\to\infty}\bar{W}_k\le\bar{W}_{\text{tv}}\ :=\ \|G_\theta\|\left[\sqrt{\frac{2\rho^{\star}}{\underline{\lambda}}}+d_{\max}\Delta\right] \tag{A.28}$$

with $\underline{\lambda}=\lambda_{\min}(P)$ and $\rho^{\star}$ defined in Eq. (74). Picking $\bar{W}=\bar{W}_{\text{tv}}$ yields $\bar{\rho}_{\text{in}}^{\text{of}}=\mathcal{I}_{\text{in}}^{\text{of}}(\bar{W}_{\text{tv}})$ and $\rho_{\min}^{\text{of}}=\mathcal{R}_{\min}^{\text{of}}(\bar{W}_{\text{tv}})$ (up to an arbitrarily small slack), which is smaller than the raw-disturbance inflation ball whenever $\bar{W}_{\text{tv}}\ll W_{\max}$.

References

Berkenkamp, F., Turchetta, M., Schoellig, A., Krause, A., 2017. Safe model-based reinforcement learning with stability guarantees. Adv. Neural Inf. Process. Syst. 30.

Chow, Y., Nachum, O., Duenez-Guzman, E., Ghavamzadeh, M., 2018. A lyapunov-based approach to safe reinforcement learning. Adv. Neural Inf. Process. Syst. 31.

Christofides, P.D., Scattolini, R., Muñoz de la Peña, D., Liu, J., 2013. Distributed model predictive control: A tutorial review and future research directions. Comput. Chem. Eng. 51, 21–41.

Dulac-Arnold, G., Mankowitz, D., Hester, T., 2019. Challenges of real-world reinforcement learning. arXiv preprint arXiv:1904.12901.

Ellis, M., Durand, H., Christofides, P.D., 2014. A tutorial review of economic model predictive control methods. J. Process Control 24, 1156–1178.

Faria, R.d., Capron, B.D.O., Secchi, A.R., de Souza Jr., M.B., 2022. Where reinforcement learning meets process control: Review and guidelines. Processes 10 (11), 2311.

Garcıa, J., Fernández, F., 2015. A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. 16, 1437–1480.

Gerold, H., Lucia, S., 2025. Safe reinforcement learning via adaptive robust model predictive shielding. Comput. Chem. Eng. 109521.

Hassanpour, H., Mhaskar, P., Corbett, B., 2024a. A practically implementable reinforcement learning control approach by leveraging offset-free model predictive control. Comput. Chem. Eng. 181, 108511.

Hassanpour, H., Wang, X., Corbett, B., Mhaskar, P., 2024b. A practically implementable reinforcement learning-based process controller design. AIChE J. 70, e18245.

Iyengar, G.N., 2005. Robust dynamic programming. Math. Oper. Res. 30, 257–280.

Khalil, H.K., Grizzle, J.W., 2002. Nonlinear Systems, vol. 3, Prentice hall Upper Saddle River, NJ.

Khodaverdian, A., Cui, X., Christofides, P.D., 2025a. Utilizing reinforcement learning in feedback control of nonlinear processes with stability guarantees. Digit. Chem. Eng. 17, 100277.

Khodaverdian, A., Gohil, D., Christofides, P.D., 2025b. Enhancing cybersecurity of nonlinear processes via a two-layer control architecture. Digit. Chem. Eng. 15, 100233.

Levine, S., Kumar, A., Tucker, G., Fu, J., 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643.

Lewis, F.L., Vrabie, D., Syrmos, V.L., 2012. Optimal Control. John Wiley & Sons.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Liu, J., Muñoz de la Peña, D., Christofides, P.D., 2010. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. Automatica 46, 52–61.

Mhaskar, P., El-Farra, N.H., Christofides, P.D., 2006. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. Syst. Contr. Lett. 55, 650–659.

Morimoto, J., Doya, K., 2005. Robust reinforcement learning. Neural Comput. 17, 335–359.

Nian, R., Liu, J., Huang, B., 2020. A review on reinforcement learning: Introduction and applications in industrial process control. Comput. Chem. Eng. 139, 106886.

Nilim, A., El Ghaoui, L., 2005. Robust control of Markov decision processes with uncertain transition matrices. Oper. Res. 53, 780–798.

Pannocchia, G., 2015. Offset-free tracking MPC: A tutorial review and comparison of different formulations. In: Proceedings of the European Control Conference. Linz, Austria, pp. 527–532.

Pannocchia, G., Gabiccini, M., Artoni, A., 2015. Offset-free MPC explained: novelties, subtleties, and applications. IFAC-PapersOnLine 48, 342–351.

Muñoz de la Peña, D., Christofides, P.D., 2008. Lyapunov-based model predictive control of nonlinear systems subject to data losses. IEEE Trans. Autom. Control 53, 2076–2089.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11 (7), 733–764.
Rawlings, J.B., Mayne, D.Q., Diehl, M., et al., 2020. Model Predictive Control: Theory, Computation, and Design, vol. 2, Nob Hill Publishing Madison, WI.
Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction, second ed. MIT Press.
Wallace, M., Pon Kumar, S.S., Mhaskar, P., 2016. Offset-free model predictive control with explicit performance specification. Ind. Eng. Chem. Res. 55, 995–1003.
Wang, Y., Zhu, X., Wu, Z., 2025. A tutorial review of policy iteration methods in reinforcement learning for nonlinear optimal control. Digit. Chem. Eng. 100231.
Zhu, X., Wang, Y., Wu, Z., 2025. Reinforcement learning for optimal control of stochastic nonlinear systems. AIChE J. 71, e18840.