

Article

Value-Regularized Reinforcement Learning for Model Predictive Control of Autonomous Mobile Robots Under Stochastic Disturbances

Changyuan Yu , Weiguo Zhang *, Qi Li, Yongdong Cheng and Zhanming Li

Xi'an Institute of Applied Optics, Xi'an 710065, China; yuchangyuan0108@163.com (C.Y.)

* Correspondence: zzwwgg1975@163.com

Abstract

Autonomous mobile robots depend on sensing and state estimation to provide feedback, while their controllers must remain effective under process disturbances and model mismatch. Reinforcement learning-based model predictive control (RL-MPC) learns the terminal cost online, while the deterministic RLMPc baseline uses a nominal stage cost. This study considers the control layer and assumes that the robot state is available; measurement noise and state-estimation errors are outside the modeled disturbance channel. Isotropic state regularization is introduced into the RL-MPC stage cost, yielding value-regularized RLMPc (VR-RLMPc). The same β term changes both the physical state penalty and the N -step terminal-value learning target without adding online optimization variables, constraints, or sampling operations. Under explicit value-function, feasibility, domain-containment, and bounded-disturbance assumptions, a Lyapunov-drift analysis yields a conditional one-step expected-drift bound outside an explicit radius. Simulations on linear and nonlinear nonholonomic vehicle systems show empirical VFA weight-update settling-step indices (CR) that are approximately 25% lower than those of RLMPc across the nominal and five model-mismatch comparisons. Across the tested mismatch range, VR-RLMPc has a worst-case performance degradation rate of 2.8%, compared with 25–49% for conventional MPC baselines; its cost improvement also increases with task difficulty. In the tested nominal setting, VR-RLMPc also outperforms an explicit stochastic temporal-difference baseline without adding online optimization variables or sampling operations.

Keywords: model predictive control; reinforcement learning; robust cost design; online learning; expected Lyapunov drift; autonomous mobile robots; stochastic disturbances

1. Introduction

Autonomous mobile robots operating under stochastic disturbances require controllers that adapt online to uncertain dynamics while satisfying state and input constraints. Model predictive control (MPC) addresses this need by solving a constrained finite-horizon optimization problem online at each time step [1]. Closed-loop stability typically relies on the design of a terminal cost $F(\cdot)$ and a terminal constraint set $\mathbb{X}_f$ [2]: $F(\cdot)$ must approximate the infinite-horizon optimal value function, and $\mathbb{X}_f$ must be a positively invariant set. For linear systems, $F(\cdot)$ can be obtained analytically from the discrete algebraic Riccati equation (DARE); for nonlinear systems, an offline computation based on local linearization around the equilibrium is required. When the plant is subject to external disturbances or model parameter deviations, a terminal design based on the nominal model may fail to



Received: 14 June 2026

Revised: 20 July 2026

Accepted: 20 July 2026

Published: 21 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

capture the true long-horizon cost, leading to closed-loop performance degradation or even constraint violation.

In a physical mobile robot, the control state is typically supplied by onboard sensors and a state-estimation pipeline. This work isolates the control-layer problem and assumes that the state is available at each control step. Measurement noise, state-estimation error, and multi-sensor fusion are not included in the disturbance model or evaluated in the simulations. Recent Sensors studies have used deep reinforcement learning (DRL) for mobile-robot path following and sensor-fusion-based navigation [3,4], illustrating the broader sensing–navigation context in which autonomous mobile robots operate. The uVGS-2 framework is a recent example that combines vision-based 6-DoF pose measurements with factor-graph localization [5]. Its treatment of camera calibration, illumination, line-of-sight occlusion, and noise rejection illustrates practical sensing issues upstream of the control layer.

Robust and stochastic MPC address such uncertainty within the online optimization. Tube MPC [6] forms a feasible tube by augmenting the nominal trajectory with the Minkowski sum of a robust positively invariant set, guaranteeing recursive feasibility under bounded disturbances at the expense of conservatism. Stochastic MPC [7,8] relaxes hard constraints to chance constraints, reducing conservatism but incurring higher online computation. Recent extensions expand these frameworks: Paulson et al. [9] address joint chance-constraint feasibility, Arcari et al. [10] incorporate parametric uncertainty, Bujarbaruah et al. [11] unify adaptive robust and stochastic MPC, and Köhler et al. [12] provide a computationally efficient framework for uncertain nonlinear systems. On the theoretical side, the input-to-state stability (ISS) Lyapunov approach [13,14] establishes robust stability of MPC through value-function decrease conditions, and Allan et al. [15] further prove inherent robustness of MPC without terminal constraints.

These methods explicitly model uncertainty in the online optimization and are suitable when the uncertainty set, chance-constraint level, or invariant tube is part of the controller specification. The present study addresses a narrower question within RL-MPC: how to couple the constrained control objective and the learned terminal-value target to a disturbance-sensitive state penalty without enlarging the online MPC problem. The proposed regularization modifies the stage-cost signal that drives value-function learning while preserving the RL MPC decision variables, constraints, and sampling structure. It is a low-complexity learning-side intervention for additive process disturbances rather than a substitute for tube or stochastic MPC.

Online terminal-cost learning provides an alternative to a fixed offline design. Gaussian process (GP) modeling [16,17] learns dynamics residuals online and quantifies uncertainty, but its inference cost limits real-time applicability to large-scale systems. Related safety-oriented work imposes Lyapunov stability constraints in model-based RL [18], designs predictive safety filters [19], and surveys safety mechanisms for learning-based MPC [20]. RL and MPC have also been combined to learn terminal costs online. Gros and Zanon [21] treat MPC as a parameterized policy approximator and tune its parameters via temporal-difference (TD) errors, with subsequent work [22,23] introducing robustness and stability guarantees. Rosolia and Borrelli [24] iteratively construct safe sets and terminal costs from historical trajectories, while Soloperto et al. [25] provide closed-loop performance guarantees. These RL-MPC methods build on approximate dynamic programming (ADP) [26]; Lewis et al. [27] and Kiumarsi et al. [28] survey RL for optimal control, and Al-Tamimi et al. [29] prove convergence of the discrete-time Hamilton–Jacobi–Bellman (HJB) equation in this framework. Recent DRL-MPC applications address cloud-assisted path tracking and distributed cooperative control [30,31]; their different learning objects

and evaluation protocols make them contextual references rather than direct numerical baselines for online terminal-cost learning.

Regularized RL provides a separate point of comparison. General regularized Markov decision process formulations modify the Bellman operator through a convex policy regularizer, commonly based on entropy or Kullback–Leibler divergence [32]. VR-RLMPC does not regularize the policy distribution, action entropy, or policy-update divergence. Instead, βI changes the physical quadratic state penalty inside the constrained RL-MPC problem, and this modified stage cost enters the TD target used to learn the terminal value function. The term “value-regularized” therefore denotes stage-cost shaping of online terminal-value learning rather than standard entropy- or policy-regularized RL.

Lin et al. [33] propose an RL-MPC scheme that bridges RL and MPC through policy iteration (PI): MPC serves as the policy generator, while VFA estimates the terminal cost online via TD learning; the two alternate until convergence. This scheme removes the need for an offline terminal set and terminal controller, and provides convergence and closed-loop stability guarantees. The cost function in this formulation, however, assumes a deterministic system. Under additive process disturbances, the nominal terminal-value target may not represent the disturbed finite-horizon behavior. This work examines disturbance robustness in RL-MPC through a direct modification of the stage cost. Specifically, VR-RLMPC places the same β term in the physical state penalty and the N -step TD target while retaining the online MPC structure. The analysis uses deterministic PI monotonicity as its baseline and treats disturbances through a conditional expected-drift bound and model-mismatch evaluation.

This work proposes value-regularized RLMPC (VR-RLMPC), which modifies the stage cost within the RL-MPC framework to couple constrained control with online terminal-value learning under process disturbances. The main contributions are as follows.

1. Isotropic state regularization is embedded in the stage cost so that the value function learning target penalizes state deviations more strongly. The same β term changes both the physical state penalty and the N -step TD target used for terminal-value learning, without changing the online MPC dimension, constraints, or sampling structure. This coupled control–learning modification distinguishes VR-RLMPC from changing the MPC weight alone and from policy-distribution regularization in standard regularized RL. Two effects provide a mechanism-level explanation for the observed empirical weight-update settling changes: amplification of the TD stage-cost signal and reduction of the condition number of the stage-cost matrix. Direction-dependent priorities remain encoded by Q and the constraint sets.
2. Under explicit VFA, Bellman-residual, feasibility, domain-containment, and bounded-disturbance assumptions, a Lyapunov-drift analysis gives a conditional one-step expected-drift bound. For fixed disturbance covariance, Hessian bound, and Bellman residual, the radius outside which the expected drift is negative decreases as β increases; this is not a recursive-feasibility or trajectory-level probability guarantee.
3. The proposed regularization is evaluated on a nonlinear nonholonomic vehicle system through ablation and model-mismatch simulations. The evaluation also includes an MPC_Q ablation, a stochastic-TD comparison, a matched training-budget study, and a β -sensitivity sweep. At $\beta = 0.5$, empirical CR is lower in the nominal and all five model-mismatch comparisons, the cost improvement grows with task difficulty, and the observed performance degradation rate is lower than that of the conventional MPC baselines.

The remainder of this paper is organized as follows. Section 2 introduces the system model with disturbances, MPC fundamentals, and the theoretical background of value function approximation. Section 3 details the robust RL-MPC algorithm design, including the robust cost function, MPC-RL interaction procedure, convergence analysis, and the conditional expected-drift result. Section 4 presents and discusses simulation results on both linear and nonlinear systems. Section 5 concludes the paper and outlines future validation directions.

2. Background and System Modeling

2.1. System Model with Process Disturbances

Consider a discrete-time nonlinear system with additive disturbances:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (1)$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control input, $f(\cdot)$ is the nominal state transition function, and w_k denotes an external process disturbance. The nominal MPC predictions impose $x_{i|k} \in \mathbb{X}$ and $u_{i|k} \in \mathbb{U}$, where both sets are bounded, closed, and contain the origin. The applied input is constrained by $\mathbb{U}$, but recursive satisfaction of the state constraints by the disturbed closed-loop state is neither assumed nor proved for unbounded disturbances.

Assumption 1 (Smoothness and state availability). $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is continuously differentiable with $f(0, 0) = 0$. The state x_k is available at each time step through an idealized control interface. Measurement noise, state-estimation error, and the sensing architecture are not modeled separately. Sensor latency and estimator–controller coupling are likewise excluded. All analytical statements therefore concern the state-feedback interconnection and do not extend to an estimator–controller interconnection.

Assumption 2 (Simulation and analysis disturbance models). The simulations use independent Gaussian process disturbances $w_k^G \sim \mathcal{N}(0, \Sigma_G)$. The expected-drift analysis uses a distinct analytical disturbance $\tilde{w}_k$ that is independent of the closed-loop history, satisfies $\mathbb{E}[\tilde{w}_k | x_k] = 0$ and $\mathbb{E}[\tilde{w}_k \tilde{w}_k^T | x_k] = \Sigma_{\tilde{w}}$, and is supported on the compact set

$$\mathbb{W} = \{\tilde{w} : \|\tilde{w}\| \leq \bar{w}\}. \quad (2)$$

A symmetrically truncated Gaussian is one admissible analytical model, but its covariance is $\Sigma_{\tilde{w}}$, not automatically the covariance of the untruncated simulation noise. The theoretical result is conditional on this bounded zero-mean model and makes no recursive-feasibility, tail-risk, or trajectory-level claim for untruncated Gaussian disturbances.

Following the deterministic RLMPC baseline, terminal-value learning is defined for the nominal disturbance-free dynamics $\bar{x}_{i+1} = f(\bar{x}_i, \pi(\bar{x}_i))$, with $\bar{x}_k = x_k$. The nominal infinite-horizon objective is

$$J_0(x_k, \pi) = \sum_{i=k}^{\infty} l(\bar{x}_i, \pi(\bar{x}_i)) \quad (3)$$

where $l(x, u)$ is the stage cost. This nominal value-function objective is distinct from the finite-horizon accumulated cost used to evaluate the persistently disturbed simulations in Section 4.1; no undiscounted infinite-horizon expected-cost claim is made for the untruncated Gaussian process.

Assumption 3 (Positive definiteness and quadratic weights). *The nominal stage cost used in the MPC formulations is $l(x, u) = x^T Q x + u^T R u$, where $Q = Q^T \succ 0$ and $R = R^T \succeq 0$. Consequently, l is positive definite in the state, and there exist $\mathcal{K}_\infty$ -class functions α_1, α_2 [13] such that*

$$\alpha_1(\|x\|) \leq \min_u l(x, u) \leq \alpha_2(\|x\|) \quad (4)$$

The optimal value function is defined as

$$V^*(x_k) = \min_{\pi} J_0(x_k, \pi) \quad (5)$$

with the corresponding optimal policy

$$\pi^*(x_k) = \arg \min_{u_k} \{l(x_k, u_k) + V^*(f(x_k, u_k))\}. \quad (6)$$

Assumptions 1 and 3 and the nominal objective follow the deterministic RL-MPC setting [33]; Assumption 2 separately defines the simulation disturbance and the bounded analytical disturbance used below.

2.2. MPC and Terminal Cost Design

MPC solves the following finite-horizon optimization at each time step k :

$$\min_{u_{0|k}, \dots, u_{N-1|k}} \sum_{i=0}^{N-1} l(x_{i|k}, u_{i|k}) + F(x_{N|k}) \quad (7a)$$

$$\text{s.t. } x_{0|k} = x_k, \quad x_{i+1|k} = f(x_{i|k}, u_{i|k}) \quad (7b)$$

$$x_{i|k} \in \mathbb{X}, \quad u_{i|k} \in \mathbb{U}, \quad x_{N|k} \in \mathbb{X}_f \quad (7c)$$

where N is the prediction horizon, $F(\cdot)$ is the terminal cost, and $\mathbb{X}_f$ is the terminal constraint set. Only the first input $u_{0|k}^*$ is applied, and the problem is re-solved at the next time step.

The terminal cost $F(\cdot)$ compensates the cost truncation introduced by the finite horizon, making the finite-horizon cost approximate J_0 . Its design relies on linearization of the nominal model near the equilibrium. For linear-quadratic problems, $F(x) = x^T P x$, where P solves the discrete algebraic Riccati equation (DARE). For nonlinear systems, P is obtained from a local linearization at the equilibrium. Both approaches require an accurate model, and the terminal set $\mathbb{X}_f$ must satisfy a positive invariance condition.

When the system is subject to Gaussian disturbances or model parameter deviations, the offline-computed $F(\cdot)$ and $\mathbb{X}_f$ may fail to represent the disturbed finite-horizon closed-loop behavior. An online-learned value function approximation $\hat{V}(x)$ can directly replace $F(\cdot)$, bypassing the offline design.

2.3. Value Function Approximation and Online Policy Iteration

Given a policy π , the infinite-horizon value function under deterministic dynamics is

$$V^\pi(x_k) = \sum_{i=k}^{\infty} l(x_i, \pi(x_i)) \quad (8)$$

The optimal value function V^* satisfies the Bellman equation (deterministic form):

$$V^*(x_k) = \min_{u_k} \{l(x_k, u_k) + V^*(f(x_k, u_k))\} \quad (9)$$

Solving (9) directly is intractable for nonlinear systems. A parameterized value function approximation (VFA) is introduced:

$$\hat{V}(x; W) = W^T \phi(x) \quad (10)$$

where $\phi(x) = [\phi_1(x), \dots, \phi_L(x)]^T \in \mathbb{R}^L$ is a polynomial basis function vector and $W \in \mathbb{R}^L$ is the weight vector. In the standard one-step TD form, a transition tuple (x_k, u_k, x_{k+1}) defines the residual

$$\delta_k = l(x_k, u_k) + \hat{V}(x_{k+1}; W) - \hat{V}(x_k; W) \quad (11)$$

Semi-gradient updates use δ_k to reduce the empirical TD residual. If the learning process converges, $\hat{V}(\cdot; W)$ approximates V^* and serves as an online replacement for the terminal cost $F(\cdot)$ in (7). This one-step expression provides background notation; the reported implementation evaluates uniformly sampled states with the nominal N -step MPC rollouts defined in Section 3.3.

3. Robust RL-MPC Algorithm Design and Analysis

3.1. Problem Analysis

Section 2.3 introduces the value-function approximation used as the MPC terminal cost. In the reported RLMPC implementation, uniformly sampled states $x^{(s)}$ are propagated through the nominal MPC model, and the finite-horizon stage-cost sum plus the terminal VFA value forms an N -step policy-evaluation target. Gaussian process disturbances enter the outer closed-loop training and evaluation trajectories, but they are not sampled inside the ordinary RLMPC or VR-RLMPC terminal target. Consequently, the learner remains driven by a nominal stage-cost objective while the controller is evaluated on disturbed trajectories. VR-RLMPC modifies this nominal learning signal and the constrained control objective through the same state-weight increment βI .

3.2. Robust Cost Function Design

Define the modified stage cost

$$l_r(x_k, u_k) = x_k^T (Q + \beta I) x_k + u_k^T R u_k \quad (12)$$

where $\beta \geq 0$ is the robust compensation coefficient. When $\beta = 0$, this reduces to the nominal cost $l(x, u)$.

Remark 1 (Design motivation). The term βI superimposes an isotropic penalty on the state weight Q , tightening the tolerance for state deviation. It also enters the terminal-value TD target through the modified stage cost, and the resulting increase in $\lambda_{\min}(Q + \beta I)$ appears in the conditional one-step expected-drift bound of Theorem 2. Under $w_k^G \sim \mathcal{N}(0, \sigma^2 I)$, the single-step state deviation due to the disturbance is of order $\mathcal{O}(\sigma)$; the value of β should modify the learning signal without dominating the stage cost. In practice, β is chosen to be on the same order as the eigenvalues of Q . Since $Q + \beta I \succeq Q \succ 0$, the modified cost l_r preserves the positive-definiteness condition used in the subsequent convergence and expected-drift analyses and increases the minimum state-cost curvature.

Remark 2 (Scope of isotropic weighting). The implemented state-weight matrix is $Q_r = Q + \beta I$, so the original directional priorities remain in Q . For the nonlinear configuration, $Q = \text{diag}(1, 1, 0.2)$ becomes $Q_r = \text{diag}(1 + \beta, 1 + \beta, 0.2 + \beta)$, retaining a lower heading weight with reduced contrast. The model f preserves the nonholonomic coupling, while R , $\mathbb{X}$, and $\mathbb{U}$ retain the input priorities and admissible sets. Direction-dependent regularization could encode additional actuator priorities but would require more tuning parameters; this study uses the reproducible single-parameter form.

3.3. Algorithm Procedure

3.3.1. MPC Stage

At each time step k , MPC replaces l with l_r and solves the finite-horizon optimization:

$$\min_{u_{0|k}, \dots, u_{N-1|k}} \sum_{i=0}^{N-1} l_r(x_{i|k}, u_{i|k}) + \hat{V}(x_{N|k}; W_j) \quad (13a)$$

$$\text{s.t. } x_{0|k} = x_k, \quad x_{i+1|k} = f(x_{i|k}, u_{i|k}) \quad (13b)$$

$$x_{i|k} \in \mathbb{X}, \quad u_{i|k} \in \mathbb{U} \quad (13c)$$

where W_j is the VFA weight after the j -th policy iteration. Compared with (7), problem (13) differs in two ways: (i) the stage cost changes from l to l_r ; (ii) the terminal set constraint $\mathbb{X}_f$ is removed, with the VFA terminal cost implicitly assuming the stabilization role. The MPC solver still uses the nominal dynamics $f(x, u)$ for prediction; disturbance compensation is achieved indirectly through the βI term in l_r and the online VFA learning. Only the first input $u_{0|k}^*$ is applied to the controlled plant.

3.3.2. RL Stage

For each sampled state $x^{(s)}$, the nominal MPC rollout supplies an N -step semi-gradient target. With predicted state–input pairs $\{x_{i|j}^{(s)}, u_{i|j}^{*(s)}\}_{i=0}^{N-1}$ and terminal state $x_{N|j}^{(s)}$, the robust TD residual is

$$\delta_j^{(s)} = \sum_{i=0}^{N-1} l_r(x_{i|j}^{(s)}, u_{i|j}^{*(s)}) + W_j^T \phi(x_{N|j}^{(s)}) - W_j^T \phi(x^{(s)}). \quad (14)$$

The corresponding unclipped batch semi-gradient update is

$$W_{j+1} = W_j + \alpha \sum_{s=1}^q \delta_j^{(s)} \phi(x^{(s)}), \quad (15)$$

where $\alpha > 0$ is the learning rate and q is the number of samples per policy-iteration step. The code applies samples sequentially with ℓ_2 shrinkage λ_W , update-norm clipping at $G_{\max}$, and elementwise projection to $[-W_{\max}, W_{\max}]$. The nonlinear simulations use $\alpha = 10^{-5}$, $\lambda_W = 10^{-3}$, $G_{\max} = 1$, and $W_{\max} = 100$; the linear simulations use $\alpha = 10^{-3}$ with the same safeguards. Learning driven by l_r targets its nominal deterministic infinite-horizon value. A finite basis, sampled states, and finite updates provide an approximation of this target rather than exact finite-sample recovery.

Figure 1 is a system-level signal-flow schematic. The “Robust MPC Optimizer” label denotes the value-regularized constrained optimizer in (13), rather than a tube-, set-, or chance-constrained robust MPC formulation. The critic-side single-step notation is conceptual; the reported update uses the sampled-state nominal N -step residual in (14) and the safeguards in (15). The disturbed plant closes the state-feedback loop, whereas the sampled policy-evaluation rollouts remain nominal. The reported implementation uses the polynomial basis $\hat{V}(x; W_j) = W_j^T \phi(x)$, and Algorithm 1 specifies the complete procedure.

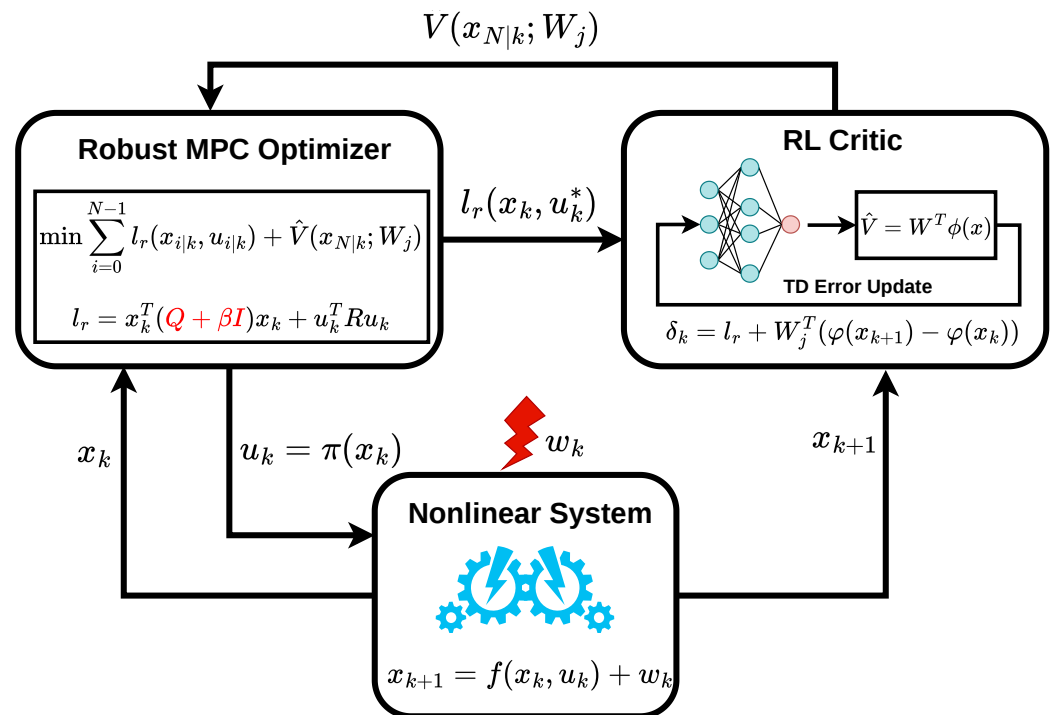


Figure 1. Conceptual system-level closed-loop architecture of VR-RLMPC; the implemented sampled-state nominal N -step VFA update is given in (14) and Algorithm 1.

Algorithm 1 Value-Regularized RL-MPC (VR-RLMPC)

Require: System f , parameters $Q, R, \beta, N, \alpha, \epsilon, q, \lambda_W, G_{\max}, W_{\max}$, initial weights W_0

Ensure: Closed-loop control policy $\mu_{W^*}(x) = u_{0|k}^*$

- 1: $Q_r \leftarrow Q + \beta I$
 - 2: **for** policy iteration $j = 0, 1, 2, \dots$ **do**
 - 3: Sample q states $\{x^{(s)}\}_{s=1}^q$ uniformly from $\mathbb{X}$
 - 4: **for** $s = 1, \dots, q$ **do**
 - 5: Solve (13) with terminal weights W_j and obtain the N -step nominal rollout
 - 6: Compute $\delta_j^{(s)}$ from (14)
 - 7: Apply the ℓ_2 -regularized update, clip its norm at $G_{\max}$, and project W to $[-W_{\max}, W_{\max}]$
 - 8: **end for**
 - 9: $W_{j+1} \leftarrow$ the final projected weight after the q sequential updates
 - 10: **if** $\|W_{j+1} - W_j\| \leq \epsilon$ **then**
 - 11: break
 - 12: **end if**
 - 13: **end for**
 - 14: **Deployment:** At each time step k , solve (13) with W^* and apply $u_{0|k}^*$
-

3.4. Convergence Analysis

The analysis below is carried out under deterministic training ($w_k = 0$), consistent with the analysis framework of deterministic RL-MPC. The deterministic result inherited from RL MPC concerns an idealized policy-iteration (PI) mapping in which each policy-evaluation step is completed before the policy is updated [33]. It must be distinguished from the finite-sample SGD iterates used in the implementation. In this subsection, V_r^* denotes the deterministic optimal value function associated with l_r .

Assumption 4 (Idealized policy evaluation). *For the analytical PI sequence, $w_k = 0$, $V_0(x) = 0$, and each outer iteration evaluates the finite-horizon Bellman operator exactly on $\mathbb{X}$,*

$$V_{j+1}(x) = (\mathcal{T}_N V_j)(x) \triangleq \min_{\{u_i\}_{i=0}^{N-1}} \left\{ \sum_{i=0}^{N-1} l_r(x_i, u_i) + V_j(x_N) \right\}, \quad (16)$$

subject to the nominal dynamics and constraints in (13). The minimum is assumed to exist for every $x \in \mathbb{X}$ included in the analysis.

Theorem 1 (Monotonicity of the idealized PI sequence). *Under Assumptions 1, 3, and 4, suppose that the nonnegative robust optimal value V_r^* is a fixed point of $\mathcal{T}_N$ on $\mathbb{X}$. Then*

$$0 = V_0(x) \leq V_j(x) \leq V_{j+1}(x) \leq V_r^*(x), \quad x \in \mathbb{X}. \quad (17)$$

Proof. Because $l_r \geq 0$, $\mathcal{T}_N V_0 \geq V_0$. The Bellman operator is order preserving: $V_a \leq V_b$ implies $\mathcal{T}_N V_a \leq \mathcal{T}_N V_b$. Induction therefore gives $V_j \leq V_{j+1}$. Likewise, $V_0 \leq V_r^*$ and $\mathcal{T}_N V_r^* = V_r^*$ imply $V_j \leq V_r^*$ for every j . $\square$

Theorem 1 applies only to the exact PI sequence in Assumption 4. Algorithm 1 uses a finite basis, finitely many sampled states, and finite semi-gradient updates; pointwise monotonicity and a convergence rate for its weight sequence $\{W_j\}$ do not follow from this theorem. The simulations therefore report CR only as an empirical weight-update settling-step index rather than as verification of a finite-sample convergence theorem.

Remark 3 (Interpretation and limit of β in the learning dynamics). *The observed weight-update settling changes can be interpreted through two effects. (i) TD stage-cost signal: the stage-cost component of the TD error (14) increases by $\beta \|x\|^2$. Before empirical weight settling, this changes the scale of the effective semi-gradient signal in the SGD update (15). (ii) Stage-cost condition number: the condition number of $Q_r = Q + \beta I$ satisfies*

$$\kappa(Q_r) = \frac{\lambda_{\max}(Q) + \beta}{\lambda_{\min}(Q) + \beta} \leq \kappa(Q) \quad (18)$$

with strict inequality when $\beta > 0$ and Q is not a scalar matrix. A better-conditioned stage-cost matrix reduces differences in the stage-cost weighting across state dimensions and can lower the approximation difficulty for a finite basis set ϕ . Together, these two effects provide a mechanism-level interpretation of the empirical settling-step changes observed in the nominal and mismatch comparisons. They do not constitute a convergence-rate proof. Because $\delta_j^{(s)}$ enters the update together with α and $\phi(x^{(s)})$, finite-sample updates can grow with larger β , learning rates, state excursions, or basis magnitudes despite the reported clipping safeguards. The present analysis does not prove bounded unclipped weights or exclude divergence for untested β - α combinations or more dynamic regimes. The sensitivity sweep instead shows that larger β can increase CR under the high- Q_θ configuration.

3.5. Conditional Expected Lyapunov-Drift Analysis

This subsection uses a Taylor/Lyapunov decomposition [13] to derive a conditional one-step expected-drift bound for the bounded analytical disturbance in Assumption 2. The result applies to the state-feedback interconnection of Assumption 1; recursive feasibility and trajectory-level behavior require separate analysis.

Assumption 5 (VFA Hessian boundedness). *For fixed deployed weights W^* , $V(x) \triangleq \hat{V}(x; W^*)$ is twice continuously differentiable on a domain $\mathbb{D} \supseteq \mathbb{X}$, and there exists a constant $M_V > 0$ such that*

$$\|\nabla^2 V(z)\| \leq M_V, \quad z \in \mathbb{D}. \quad (19)$$

For a quadratic polynomial basis, $\nabla^2 V$ is a constant matrix and the Hessian bound holds globally; for higher-order bases, M_V remains finite on the bounded domain used by the analysis.

Assumption 6 (Lyapunov upper and lower bounds). *There exist $\mathcal{K}_\infty$ -class functions $\underline{\alpha}, \bar{\alpha}$ such that*

$$\underline{\alpha}(\|x\|) \leq V(x) \leq \bar{\alpha}(\|x\|), \quad \forall x \in \mathbb{X}. \quad (20)$$

These are the standard Lyapunov upper and lower bound conditions used in the ISS framework [13].

Assumption 7 (Nominal Bellman residual). *For the deployed MPC policy μ_{W^*} , there is a uniform constant $\epsilon_B \geq 0$ satisfying*

$$V(x) - V(f(x, \mu_{W^*}(x))) \geq l_r(x, \mu_{W^*}(x)) - \epsilon_B, \quad x \in \mathbb{X}. \quad (21)$$

This residual bound is an explicit analytical assumption; it is not implied by Theorem 1 or by a one-step concentration inequality. The present simulations do not certify a uniform numerical value of ϵ_B .

Assumption 8 (Feasibility and domain containment). *At every state to which the result is applied, problem (13) is feasible and returns $\mu_{W^*}(x)$. Moreover, for every $\tilde{w} \in \mathbb{W}$ and $t \in [0, 1]$,*

$$f(x, \mu_{W^*}(x)) + t\tilde{w} \in \mathbb{D}. \quad (22)$$

Theorem 2 (Conditional expected Lyapunov drift). *Under Assumptions 2, 3, and 5–8, consider*

$$x_{k+1} = f(x_k, \mu_{W^*}(x_k)) + \tilde{w}_k. \quad (23)$$

For every $x_k \in \mathbb{X}$ at which Assumption 8 holds,

$$\begin{aligned} \mathbb{E}[V(x_{k+1}) \mid x_k] - V(x_k) \\ \leq -\lambda_{\min}(Q + \beta I)\|x_k\|^2 + \epsilon_B + \frac{M_V}{2} \text{tr}(\Sigma_{\tilde{w}}). \end{aligned} \quad (24)$$

Consequently, the conditional expected drift is negative whenever

$$\|x_k\| > r_\beta \triangleq \sqrt{\frac{\epsilon_B + M_V \text{tr}(\Sigma_{\tilde{w}})/2}{\lambda_{\min}(Q + \beta I)}}. \quad (25)$$

Proof. Let $\bar{x}_{k+1} = f(x_k, \mu_{W^*}(x_k))$. All expectations below are conditional on x_k .

Step 1 (Nominal decrease). By Assumption 7 and $l_r(x, u) \geq \lambda_{\min}(Q + \beta I)\|x\|^2$,

$$\begin{aligned} V(x_k) - V(\bar{x}_{k+1}) &\geq l_r(x_k, \mu_{W^*}(x_k)) - \epsilon_B \\ &\geq \lambda_{\min}(Q + \beta I)\|x_k\|^2 - \epsilon_B. \end{aligned} \quad (26)$$

The second inequality follows from the quadratic lower bound of l_r in (12), because $l_r(x, u) \geq x^T(Q + \beta I)x$.

Step 2 (Disturbance increment). The disturbed successor is $x_{k+1} = \bar{x}_{k+1} + \tilde{w}_k$. Given x_k , $\bar{x}_{k+1}$ is deterministic and randomness comes solely from $\tilde{w}_k$. A second-order Taylor expansion of V at $\bar{x}_{k+1}$ (Lagrange remainder form), along the segment specified in Assumption 8, gives

$$V(\bar{x}_{k+1} + \tilde{w}_k) \leq V(\bar{x}_{k+1}) + \nabla V(\bar{x}_{k+1})^T \tilde{w}_k + \frac{M_V}{2} \|\tilde{w}_k\|^2. \quad (27)$$

Equivalently, the Lagrange remainder is evaluated at an intermediate point $\xi_k = \bar{x}_{k+1} + t_k \tilde{w}_k$ with $t_k \in [0, 1]$. The first-order term vanishes after conditioning because $\nabla V(\bar{x}_{k+1})$ is deterministic given x_k and $\mathbb{E}[\tilde{w}_k | x_k] = 0$. For the second-order term, ξ_k depends on $\tilde{w}_k$, so $\nabla^2 V(\xi_k)$ cannot be pulled outside the expectation. Assumptions 5 and 8 instead provide the pathwise bound

$$\tilde{w}_k^T \nabla^2 V(\xi_k) \tilde{w}_k \leq M_V \|\tilde{w}_k\|^2. \quad (28)$$

This is a pathwise inequality that does not rely on statistical independence between $\nabla^2 V(\xi_k)$ and $\tilde{w}_k$. Taking the conditional expectation and using $\mathbb{E}[\|\tilde{w}_k\|^2 | x_k] = \text{tr}(\Sigma_{\tilde{w}})$ yields

$$\mathbb{E}[V(x_{k+1}) | x_k] - V(\bar{x}_{k+1}) \leq \frac{M_V}{2} \text{tr}(\Sigma_{\tilde{w}}). \quad (29)$$

The use of the Hessian bound M_V requires $\xi_k \in \mathbb{D}$. For a quadratic VFA, the Hessian bound holds globally and this requirement is automatically satisfied. For a higher-order VFA, the segment-containment condition in Assumption 8 ensures that every intermediate point remains in the stated domain.

Step 3 (Combination). From the nominal decrease and the disturbance increment,

$$\begin{aligned} & \mathbb{E}[V(x_{k+1}) | x_k] - V(x_k) \\ &= [\mathbb{E}[V(x_{k+1}) | x_k] - V(\bar{x}_{k+1})] \\ & \quad - [V(x_k) - V(\bar{x}_{k+1})] \\ & \leq -\lambda_{\min}(Q + \beta I) \|x_k\|^2 + \epsilon_B + \frac{M_V}{2} \text{tr}(\Sigma_{\tilde{w}}), \end{aligned} \quad (30)$$

which proves (24). When $\|x_k\| > r_\beta$, the right-hand side is strictly negative by (25). This establishes the stated conditional one-step result without asserting recursive feasibility, set invariance, or trajectory-level ultimate boundedness. $\square$

Remark 4 (Interpretation of β). If M_V , $\Sigma_{\tilde{w}}$, and ϵ_B are held fixed, r_β decreases as $\lambda_{\min}(Q + \beta I)$ increases. In the learned controller, however, V , M_V , and ϵ_B can themselves depend on β . Equation (25) is therefore a conditional sensitivity relation, not a proof that increasing β universally improves a closed-loop stability margin. When $\Sigma_{\tilde{w}} = 0$ and $\epsilon_B = 0$, (24) recovers a nominal decrease inequality; an asymptotic-stability conclusion still requires the usual feasibility and invariance conditions.

Remark 5 (Scope). Assumption 8 is not a recursive-feasibility proof. The β term changes the cost but not the constraint sets, and the absence of infeasibility in the reported simulations is empirical rather than a guarantee. The theorem also does not apply to the untruncated Gaussian simulation disturbance without a separate tail and trajectory analysis.

4. Results and Discussion

4.1. Simulation Setup

Simulations are conducted on a second-order linear system and a nonlinear nonholonomic vehicle system; the latter captures the kinematics of a differential-drive mobile robot. The linear system admits an analytical linear-quadratic regulator (LQR) solution and serves to verify RL-MPC convergence behavior; the nonlinear system introduces Gaussian disturbances and model mismatch to examine robustness. Baseline parameters for both systems follow existing RL-MPC settings for direct comparability with prior benchmarks [33]. The simulator directly supplies the state vector to the controller. No sensor model, measurement noise, state estimator, or sensor-fusion algorithm is included, so the results evaluate the control layer rather than a complete sensing-to-control stack.

Consider the discrete-time linear system $x_{k+1} = Ax_k + Bu_k$ with $x_k \in \mathbb{R}^2$, $u_k \in \mathbb{R}$, and

$$A = \begin{bmatrix} 1 & 0.5 \\ 0.1 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (31)$$

The cost weights are $Q = I_2$, $R = 0.5$, yielding an LQR optimal cost of $J^* \approx 3.49$. The RL-MPC prediction horizon is $N = 3$ with a polynomial basis of order $p = 2$. Four methods are compared on the linear system: RLMPC, LQR, MPC without terminal cost ($N = 3$), and MPC with terminal cost ($N = 3$), where LQR serves as the global optimum benchmark.

Consider the nonholonomic kinematic vehicle model with additive disturbances:

$$x_{k+1} = x_k + g(x_k)u_k + w_k^G \quad (32)$$

The state $x_k = [\chi_k, y_k, \theta_k]^T$ contains the planar coordinates and heading angle, and the control input $u_k = [v_k, \omega_k]^T$ consists of the linear and angular velocities. The system matrix

$$g(x_k) = \delta \begin{bmatrix} \cos \theta_k & 0 \\ \sin \theta_k & 0 \\ 0 & 1 \end{bmatrix} \quad (33)$$

encodes the nonholonomic coupling between displacement and heading, with sampling interval $\delta = 0.2$ s. The disturbance $w_k^G \sim \mathcal{N}(0, \sigma^2 I)$ is additive Gaussian white noise whose intensity σ varies by scenario. Input constraints are $|v_k| \leq 1$ m/s and $|\omega_k| \leq 4$ rad/s; the initial state is $x_0 = [1.5, 1.5, \pi/4]^T$. The initial planar displacement is $\sqrt{1.5^2 + 1.5^2} \approx 2.12$ m, so the maneuver is explicitly meter-scale. For comparison, Cao et al. use a 0.4 m/s limit for nonholonomic differential-steering path following, whereas Ou et al. use a 1 m/s limit for indoor mobile-robot navigation [3,4]. The setup is therefore best interpreted as a simulation-level, low-to-moderate-speed indoor mobile-robot maneuver with differential-drive kinematics, rather than as a parameterization of a specific hardware platform. The control objective is to drive the vehicle to the neighborhood of the origin.

Five methods are compared on the nonlinear system: (i) VR-RLMPC (proposed), using the robust cost $l_r = x^T(Q + \beta I)x + u^T Ru$ with $\beta = 0.5$ and $N = 5$; (ii) RLMPC, the original scheme with $\beta = 0$; (iii) long-horizon MPC with $N = 20$ and no terminal cost; (iv) MPC without terminal cost, $N = 5$; (v) MPC with terminal cost, $N = 5$, where the terminal cost is obtained from DARE at the linearized equilibrium. The value $\beta = 0.5$ is chosen on the same order as $\lambda_{\min}(Q)$ so that the regularization intensity is comparable to the nominal state weight without excessively distorting the learning target.

The nominal nonlinear configuration uses $Q = \text{diag}(1, 1, 0.2)$, $R = \text{diag}(0.05, 0.01)$, and Gaussian disturbance level $\sigma = 0.05$ when enabled; activation follows the batch-specific training and evaluation protocols stated below. The lower heading weight prioritizes position tracking, while the lower angular-velocity penalty permits flexible heading adjustment. The value function uses a polynomial basis of order $p = 4$. Simulations run in MATLAB R2025a. The fmincon solver uses SQP first, with interior-point or active-set methods as fallbacks after a failed attempt. RLMPC and VR-RLMPC use the same learning safeguards from Section 3.3. The proposed β term changes the cost matrix without changing the online decision variables, constraints, or sampling loops. The archived ACT values report average solver time per MPC step.

Closed-loop performance is measured by the accumulated cost (ACC):

$$\text{ACC} = \sum_{k=0}^{T-1} l(x_k, u_k) \quad (34)$$

For cross-method comparability, ACC is computed using the nominal cost l (not l_r). The reported values are means $\overline{ACC}$ over multiple independent runs. VFA weight-update settling is summarized by an empirical convergence-step index, denoted CR. For each run, let $d_j = \|W_{j+1} - W_j\|_2$ over the learning window and define the run-specific relative threshold $\epsilon_{\text{rel}} = 0.15 \max_j d_j$. Then

$$\text{CR} = \inf\{j \in \mathbb{N} \mid d_{j+i} < \epsilon_{\text{rel}}, i = 0, \dots, 4\}. \quad (35)$$

CR is undefined if no five-update window meets this criterion. Because the threshold is normalized by each run's peak update norm, uniform rescaling of all update magnitudes alone does not automatically reduce CR. Nevertheless, CR remains a relative settling indicator; a finite CR does not prove finite-sample convergence, weight boundedness without safeguards, or resolution of the exploration–exploitation trade-off.

The following derived metrics are also used. The performance degradation rate (PDR) quantifies the effect of model mismatch on a single method:

$$\text{PDR}(\kappa) = \frac{\text{ACC}(\kappa) - \text{ACC}(1)}{\text{ACC}(1)} \times 100\% \quad (36)$$

The ACC improvement rate measures the cost reduction of VR-RLMPC relative to RLMPC:

$$\text{ACC imp.} = \frac{\text{ACC}_{\text{RLMPC}} - \text{ACC}_{\text{VR-RLMPC}}}{\text{ACC}_{\text{RLMPC}}} \times 100\% \quad (37)$$

The VFA terminal cost ratio (VTCR) measures the cost reduction of VR-RLMPC relative to MPC with terminal cost:

$$\text{VTCR} = \frac{\text{ACC}_{\text{MPC+TC}} - \text{ACC}_{\text{VR-RLMPC}}}{\text{ACC}_{\text{MPC+TC}}} \times 100\% \quad (38)$$

The ACC variance reduction (AVR) quantifies the compression of ACC standard deviation by β :

$$\text{AVR} = \frac{\text{std}_{\text{RLMPC}} - \text{std}_{\text{VR-RLMPC}}}{\text{std}_{\text{RLMPC}}} \times 100\% \quad (39)$$

4.2. Linear System Verification

The LQR analytical solution serves as a quantitative benchmark. Simulations are conducted under both noise-free ($\sigma = 0$) and noisy ($\sigma = 0.05$) conditions. The noise-free case is deterministic and each method is run once from identical initial conditions; the noisy case is repeated 10 times independently, reporting means and standard deviations.

Figure 2 shows the state trajectories and ACC curves for the four methods under the noise-free condition. The four trajectories nearly coincide, and all ACC values converge to the analytical optimum: RLMPC yields 3.493, LQR 3.491, MPC without terminal cost 3.495, and MPC with terminal cost 3.493, with a maximum relative deviation below 0.2%. After introducing noise, the ACC of all four methods increases synchronously to approximately 3.86 ± 0.11 , and the difference between RLMPC and LQR does not exceed 0.001%. The near-identical performance indicates that the learned terminal cost reproduces LQR-level performance on the tested linear system.

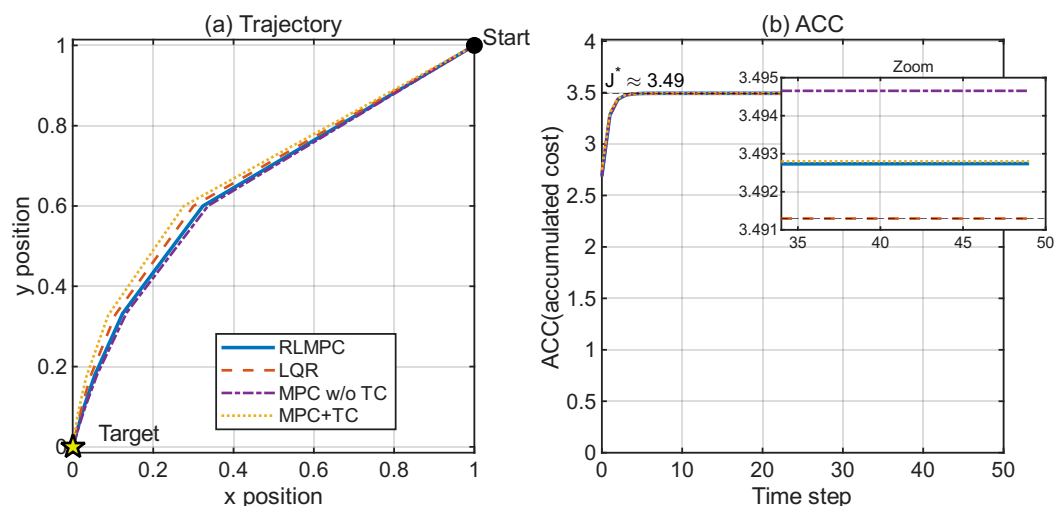


Figure 2. Linear system verification: phase-plane trajectories and ACC curves.

4.3. Nonlinear Vehicle System: Nominal Configuration

This subsection examines the practical effect of β compensation on the nonlinear vehicle system under Gaussian disturbances. Under the nominal parameter configuration ($\sigma = 0.05$, $\kappa = 1$, no model mismatch), five methods are compared using 7 independent runs per method.

Figure 3 shows the XY trajectories of the five methods. VR-RLMPC and RLMPC follow similar paths, both tracing a compact arc from the initial point toward the origin; MPC with terminal cost produces a comparable trajectory. MPC without terminal cost, lacking long-horizon cost information, shows a clear deviation with a lag in the y direction. Long-horizon MPC reaches the target vicinity but exhibits a larger path envelope.

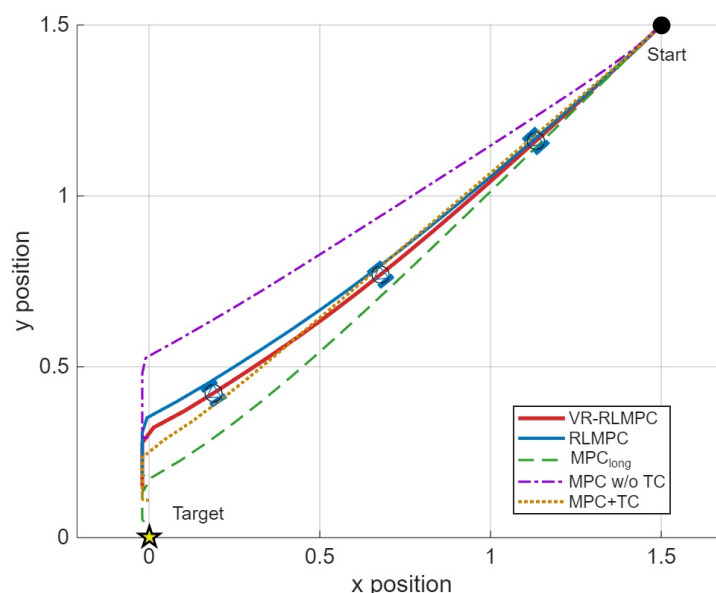


Figure 3. XY trajectory comparison on the nonlinear vehicle system.

Figure 4 further shows the time evolution of state components and control inputs. Differences are most pronounced in the y position and heading angle θ : the y component of MPC without terminal cost converges most slowly, and θ remains elevated for an extended period. Regarding control inputs, both VR-RLMPC and RLMPC reach the lower linear velocity constraint ($v = -1$ m/s) during the first 20 steps, then relax smoothly to zero.

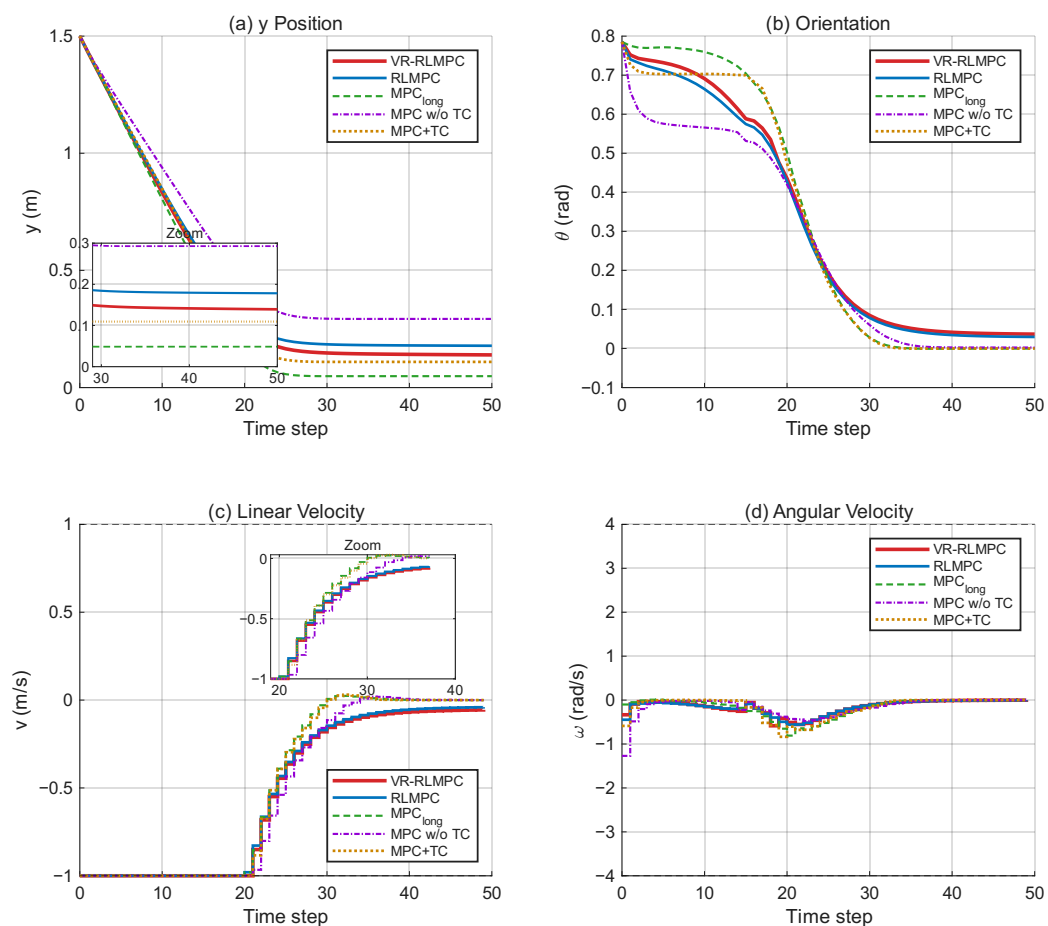


Figure 4. State and control responses under the nominal configuration (gray dashed lines: numerical input bounds).

Table 1 lists the ACC statistics under the nominal configuration. The five primary methods use 7 independent runs, whereas the separately evaluated MPC_{Q_r} baseline uses 3 independent runs.

Table 1. Accumulated cost comparison on the nonlinear vehicle system under the nominal configuration.

Method	ACC
VR-RLMPC	42.53 ± 3.37
RLMPC	42.84 ± 3.46
Long-horizon MPC	42.31 ± 3.23
MPC w/o terminal cost	45.34 ± 4.20
MPC w/ terminal cost	42.52 ± 3.37
MPC_{Q_r} (no VFA)	50.18 ± 1.26

VR-RLMPC reduces ACC by approximately 0.7% in the 7-run nominal comparison and by 1.36% in the separate 3-run stochastic-TD comparison. These changes are modest relative to run-to-run dispersion; the clearer nominal effect is the approximately 27% lower empirical CR settling-step index. The mismatch sweep and high-demand configuration produce larger, single-digit gains. Long-horizon MPC with $N = 20$ achieves slightly lower ACC than the $N = 5$ RLMPC methods. MPC without terminal cost exceeds VR-RLMPC by 6.6%, consistent with its trajectory lag. MPC with terminal cost is nearly tied with VR-RLMPC (42.52 vs 42.53), indicating comparable nominal performance from the learned VFA and DARE terminal costs.

To disentangle the source of β compensation (whether from the stage cost modification itself or from VFA online learning), an MPC_{Q_r} baseline is introduced: MPC with the modified cost matrix $Q + \beta I$ but without value function learning ($N = 5$, no terminal cost). As shown in Table 1, MPC_{Q_r} achieves an ACC of 50.18 ± 1.26 , exceeding VR-RLMPC by 18.0% (50.18 vs 42.53) and performing worse than MPC without terminal cost (45.34). In this tested setting, modifying the stage cost matrix alone does not substitute for the VFA terminal cost; the increased stage cost magnitude from βI degrades short-horizon MPC performance. This baseline supports the interpretation that the observed advantage depends on the interaction between the modified stage cost and online VFA learning, rather than on the state-weight change alone. The mechanism is therefore framed as learning-side regularization rather than simple MPC weight scaling.

Figure 5 compares the weight update processes of RLMPC ($\beta = 0$) and VR-RLMPC ($\beta = 0.5$). The left panel shows $\|\Delta W_k\|$ versus learning step: VR-RLMPC drops below the relative five-update settling threshold at approximately step 22 (CR = 22), while RLMPC requires about 30 steps (CR = 30), giving an observed CR reduction of approximately 27%. The right panel displays the evolution of individual VR-RLMPC weight components over the 100-step learning window; it is descriptive and does not establish finite-sample convergence outside this run.

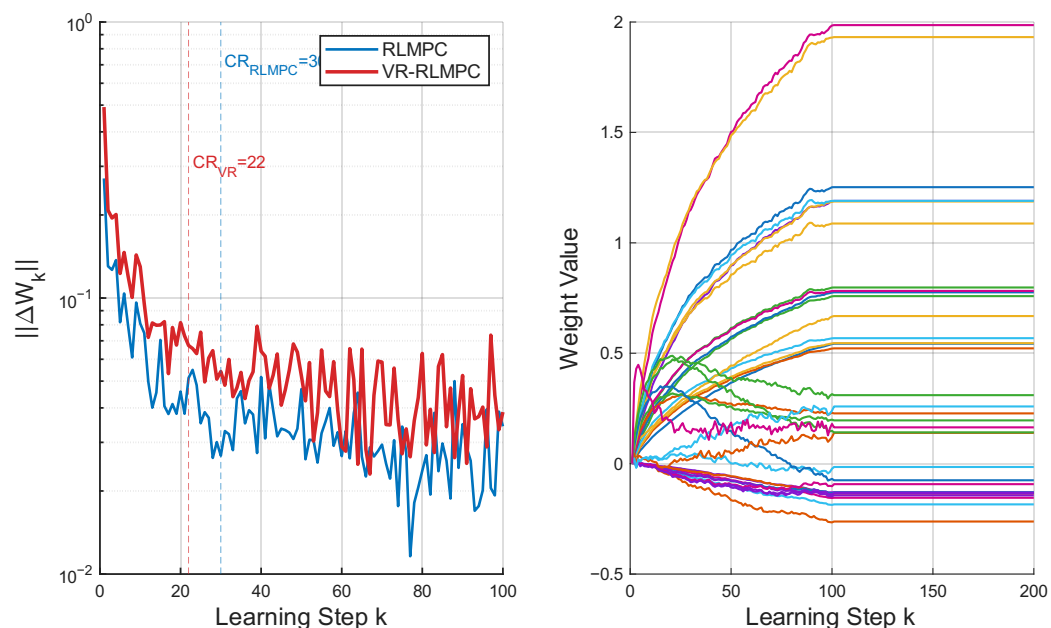


Figure 5. VFA weight-update histories under the nominal configuration. In the left panel, blue denotes RLMPC and red denotes VR-RLMPC; in the right panel, the colored lines distinguish the individual VR-RLMPC weight components.

An alternative to β regularization is to perform Monte Carlo averaging of the terminal value during TD learning, explicitly accounting for the disturbance effect on value function estimation. Specifically, the TD target terminal value $W^T \phi(x_{N|k})$ is replaced by

$$\hat{V}_{MC}(x_{N|k}; W) = \frac{1}{M} \sum_{m=1}^M W^T \phi(x_{N|k} + w^{(m)}),$$

$$w^{(m)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 I) \quad (40)$$

This method approximates $\mathbb{E}_w[\hat{V}(x_{N|k} + w)]$ via sampling, with computational overhead growing linearly in M (M additional $\phi(\cdot)$ evaluations per step). For a fair comparison with the implicit regularization of VR-RLMPC, all Stochastic TD variants use the nominal

cost l ($\beta = 0$) and handle disturbances solely through terminal value averaging. With $M \in \{5, 10, 20\}$, alongside RLMPCC ($\beta = 0$) and VR-RLMPCC ($\beta = 0.5$), 3 independent runs are performed under the nominal configuration; results are listed in Table 2.

Table 2. Explicit stochastic temporal-difference (TD) baselines under the nominal configuration. ACC: accumulated cost; CR: empirical VFA settling-step index; Δ ACC: percentage ACC improvement relative to RLMPCC ($\beta = 0$); positive values indicate lower ACC).

Method	ACC (Mean $\pm$ Std)	CR	Δ ACC
RLMPCC ($\beta = 0$)	35.54 $\pm$ 4.88	38.3	—
VR-RLMPCC ($\beta = 0.5$)	35.06 $\pm$ 5.35	27.7	+1.36%
Stochastic TD ($M = 5$)	35.43 $\pm$ 5.06	39.3	+0.32%
Stochastic TD ($M = 10$)	35.40 $\pm$ 5.10	36.7	+0.41%
Stochastic TD ($M = 20$)	35.42 $\pm$ 5.07	41.7	+0.35%

Δ ACC: improvement over RLMPCC ($\beta = 0$); positive is better.

The three Stochastic TD variants improve ACC by no more than 0.5%, with no monotone improvement as M increases from 5 to 20. VR-RLMPCC achieves an improvement of +1.36%, compared with +0.41% for the best Stochastic TD variant ($M = 10$), and obtains the lowest ACC across all 3 random seeds (paired win rate 3/3). These absolute ACC differences remain modest. Regarding empirical VFA weight-update settling, VR-RLMPCC achieves CR = 27.7, a reduction of approximately 27% relative to RLMPCC, while the Stochastic TD variants have CR = 36.7–41.7 in this nominal comparison. At $\sigma = 0.05$, $\mathbb{E}[\phi(x+w)] \approx \phi(x) + \mathcal{O}(\sigma^2)$, so Monte Carlo averaging corrects only a second-order perturbation of the terminal value estimate. In contrast, β regularization directly changes the stage-cost matrix through $Q \rightarrow Q + \beta I$. This distinction provides a mechanism-level interpretation of the different CR values observed in the tested nominal configuration, not a convergence-rate proof.

Under the nominal configuration, the primary observed benefit of β is the approximately 27% lower empirical CR settling-step index; the absolute ACC gain remains modest. In this regime, RLMPCC already operates near optimality, constraining the ACC improvement from β . The next subsection increases task difficulty to examine whether the ACC improvement changes in the tested configurations.

4.4. Parameter Sensitivity and Model-Mismatch Analysis

The nominal-configuration simulations show that β improves ACC by approximately 0.7% in the 7-run comparison, with the larger observed difference appearing in CR (27%). The following analysis tests whether the ACC improvement changes when the simulation places greater weight on heading accuracy (large Q_θ) or control effort (large R). Single-factor ablation identifies the tested settings in which the observed β benefit is larger, after which the two dominant factors are combined into a high-demand simulation configuration.

4.4.1. Single-Factor Ablation

Four factors that may affect terminal-cost accuracy requirements are selected. Factor A (tightened linear velocity, $|v_k| \leq 0.5$) compresses the constraint feasible region; B (increased Q_θ , with $Q_\theta = 1, 2, 3$) raises the state-cost weight along the heading dimension; C (increased control cost, $R \times 2$) strengthens the penalty on control effort; and D (increased initial offset) extends the transient process. Only one factor is varied at a time, with $\sigma = 0.05$ and $\kappa = 1$. Increasing Q_θ is the dominant tested factor: B2 ($Q_\theta = 2$) achieves an improvement of +3.72%, compared with nominal (+1.12%), A (+0.75%), and C (+1.16%). The full ablation also includes B1 ($Q_\theta = 1$, +2.35%), B3 ($Q_\theta = 3$, +3.96%), and D (+0.57%); the Q_θ results increase monotonically over the three tested values.

Increasing Q_θ requires the value function $\hat{V}$ to represent a more strongly weighted heading dimension. The β compensation via $Q + \beta I$ changes the learning signal across all state dimensions, with the largest observed difference under high- Q_θ settings. VR-RLMPC also achieves lower ACC than MPC with terminal cost in both B2 and B3: 57.18 vs 59.03 in B2 (3.2% lower), and 64.08 vs 68.10 in B3 (6.3% lower). These results indicate better observed performance than the DARE linearized terminal-cost baseline in the tested high- Q_θ tasks.

4.4.2. High-Demand Engineering Configuration (B2+C): Statistical Verification

Figure 6 provides a coarse cross-configuration overview. Mismatch denotes the $\kappa = 0.5$ model-mismatch result, Precision denotes B2, and High-demand denotes B2+C. The exact configuration-specific percentages and the detailed A/B/C/D single-factor results are reported in the text; the bars are descriptive rather than a matched statistical comparison.

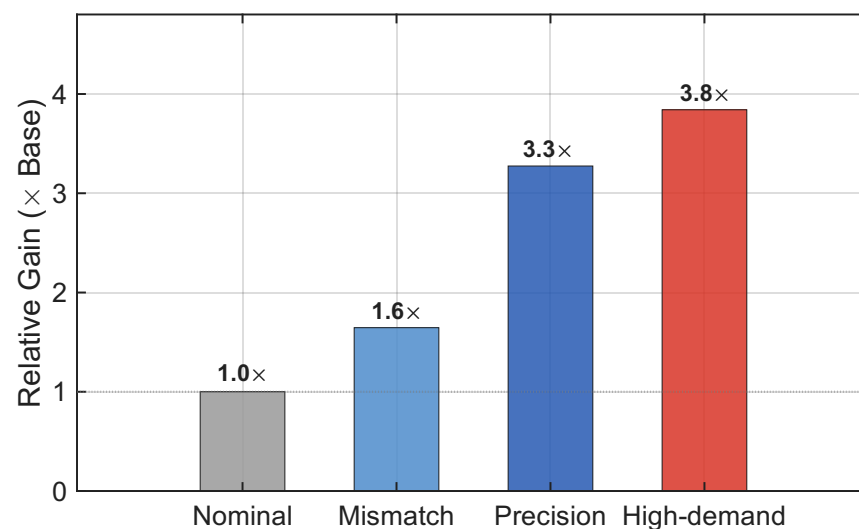


Figure 6. Relative ACC improvement with β regularization across four representative configurations. The bar colors distinguish the four labeled configurations and do not encode an additional quantitative variable.

The ablation results show that A and D have smaller effects on β improvement (+0.75%/+0.57%), while the tested B settings produce larger changes. B2+C combines a larger heading weight with a larger control-effort weight. Specifically, B2 ($Q_\theta = 2$) is selected as the primary configuration: the improvement rate is near saturation (Q_θ from 2 to 3 adds only 0.24 percentage points). On this basis, C ($R \times 2$) is added. The final high-demand configuration (denoted B2+C) uses $Q = \text{diag}(1, 1, 2)$, $R = \text{diag}(0.1, 0.05)$, with all other parameters unchanged. Across 5 independent repeats, RLMPC ($\beta = 0$) achieves an ACC of 63.03 ± 0.07 , whereas VR-RLMPC achieves an ACC of 60.28 ± 0.05 , an improvement of +4.36% (one-sided paired t -test, $p = 1.31 \times 10^{-7}$). Compared with MPC with terminal cost (ACC = 62.40), VR-RLMPC is lower by 3.4% under this high- Q_θ task.

The B2+C improvement is statistically significant in the one-sided paired test ($p = 1.31 \times 10^{-7}$). This p -value reflects the high reproducibility of the mean difference between the two methods under low-variance conditions, not the engineering magnitude of the absolute improvement. The ACC standard deviation in B2+C (0.05–0.07) is far smaller than that of the nominal configuration (≈ 3.4). The exact 4.36% high-demand and 1.12% nominal single-factor values, rather than the coarse overview in Figure 6, are used for quantitative comparison. This scenario-specific result combines a single-digit effect with unusually low within-batch variance, whose source was not isolated.

4.4.3. Model-Mismatch Robustness

In practical systems, the prediction model typically deviates from the true dynamics. A mismatch factor κ is introduced: the MPC prediction model uses sampling interval $\kappa\delta$ (the true system uses $\delta = 0.2\text{ s}$); $\kappa \neq 1$ indicates model overestimation ($\kappa > 1$) or underestimation ($\kappa < 1$) of the time step. The sweep uses $\kappa \in \{0.5, 0.7, 1.0, 1.5, 2.0\}$ with 7 independent runs per group under the nominal parameter configuration.

Figure 7 and Table 3 summarize the ACC of each method across different κ values.

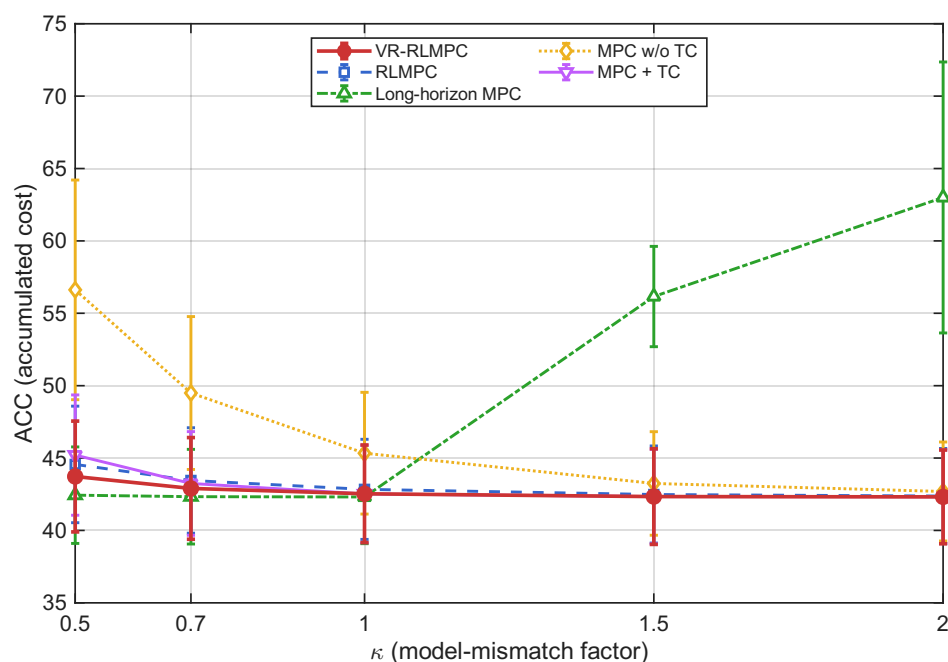


Figure 7. Mean ACC under model-mismatch factor κ (error bars: ± 1 standard deviation).

Table 3. Model-mismatch comparison. ACC is reported as mean $\pm$ standard deviation, whereas CR is the mean empirical VFA settling-step index.

ACC (Accumulated Cost)					
κ	VR-RLMPC	RLMPC	Long-horizon MPC	MPC w/o TC	MPC w/ TC
0.5	43.73 $\pm$ 3.83	44.56 $\pm$ 4.03	42.44 $\pm$ 3.33	56.62 $\pm$ 7.59	45.21 $\pm$ 4.16
0.7	42.90 $\pm$ 3.51	43.45 $\pm$ 3.64	42.33 $\pm$ 3.27	49.49 $\pm$ 5.28	43.24 $\pm$ 3.58
1.0	42.53 $\pm$ 3.37	42.84 $\pm$ 3.46	42.31 $\pm$ 3.23	45.34 $\pm$ 4.20	42.52 $\pm$ 3.37
1.5	42.34 $\pm$ 3.30	42.48 $\pm$ 3.35	56.16 $\pm$ 3.47	43.25 $\pm$ 3.58	42.34 $\pm$ 3.27
2.0	42.32 $\pm$ 3.24	42.37 $\pm$ 3.29	63.00 $\pm$ 9.36	42.69 $\pm$ 3.42	42.33 $\pm$ 3.23
CR (Empirical VFA Settling-Step Index)					
κ	VR-RLMPC	RLMPC	—		
0.5	32.7	46.4			
0.7	29.3	38.9			
1.0	22.4	29.4			
1.5	20.9	27.7			
2.0	20.7	27.0			

The data reveal two opposing degradation patterns. When $\kappa < 1$, the model underestimates the sampling interval; MPC plans aggressively, and the true system response overshoots the prediction. MPC without terminal cost degrades first: at $\kappa = 0.5$, ACC rises to 56.62 (+25% over baseline), and the standard deviation increases to 7.59. When $\kappa > 1$, the model overestimates the sampling interval and prediction errors accumulate along the longer horizon. Long-horizon MPC then degrades first: at $\kappa = 2.0$, ACC increases to 63.00

(+49%), with a standard deviation of 9.36. The two degradation directions are complementary: the mismatch direction determines which prediction structure is impaired first.

The RLMPC family shows little ACC variation across the tested κ values. VR-RLMPC ACC ranges from 42.32 to 43.73, with a maximum deviation of 2.8%; the standard deviation stays within 3.24–3.83. The short prediction horizon ($N = 5$) limits prediction error accumulation, and the VFA terminal cost provides a consistent long-horizon cost estimate in this sweep.

Regarding empirical VFA weight-update settling (lower half of Table 3), VR-RLMPC has lower CR than RLMPC at all five κ levels, with reductions of 29.5% ($\kappa = 0.5$), 24.7% ($\kappa = 0.7$), 23.8% ($\kappa = 1.0$), 24.5% ($\kappa = 1.5$), and 23.3% ($\kappa = 2.0$), averaging approximately 25%. At $\kappa = 0.5$ the absolute CR reduction is largest ($46.4 \rightarrow 32.7$, -13.7 steps), coinciding with the largest ACC improvement from β in this sweep; this empirical association is not a finite-sample convergence-rate result.

From the β compensation perspective, the improvement magnitude follows a monotone trend with mismatch direction. VR-RLMPC improves ACC over RLMPC by +1.9% at $\kappa = 0.5$, +1.3% at $\kappa = 0.7$, +0.7% at $\kappa = 1.0$, and +0.1% at $\kappa = 2.0$. When $\kappa < 1$, model underestimation amplifies the effective disturbance, and the largest observed benefit of β occurs in this regime. For $\kappa > 1$, the smaller improvement is consistent with prediction-error accumulation becoming more important relative to VFA accuracy.

Figure 8 shows the PDR (defined in Section 4.1) as a function of κ . The maximum |PDR| of VR-RLMPC is 2.8% ($\kappa = 0.5$); RLMPC reaches 4.0%, and MPC with terminal cost reaches 6.3%. Long-horizon MPC and MPC without terminal cost have maximum PDR values of 48.9% and 24.9%, respectively.

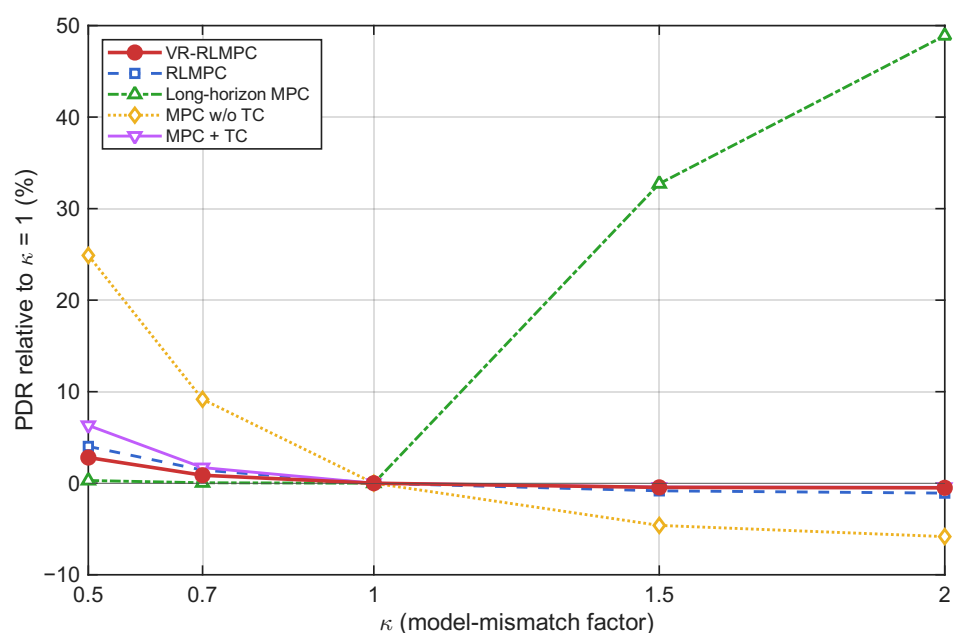


Figure 8. Performance degradation rate (PDR) as a function of κ .

At $\kappa = 0.5$, VFA learning contributes most to final performance. To assess performance under a limited training budget, the learning phase length T is reduced from 100 to 20. Both $\beta = 0$ and $\beta = 0.5$ are run 3 times at each budget. Figure 9 shows ACC versus T . At $T = 50$, VR-RLMPC achieves an ACC of 37.94 ± 3.32 , close to the RLMPC value of 37.60 ± 3.71 at $T = 100$ (a difference of 0.9%). In this simulated setting, β compensation therefore reduces the number of learning steps needed to reach a comparable ACC level. The paired comparison shows that $\beta = 0.5$ outperforms $\beta = 0$ at all 4 training-budget levels (a 100% win rate across 12 pairs). The advantage increases monotonically as the training

budget decreases, from $\Delta\text{ACC} = 2.5\%$ at $T = 100$ to 3.5% at $T = 20$. This result should be interpreted as a simulated training-budget effect rather than a general claim of RL sample efficiency, and it requires future validation with a physical sensing and estimation chain.

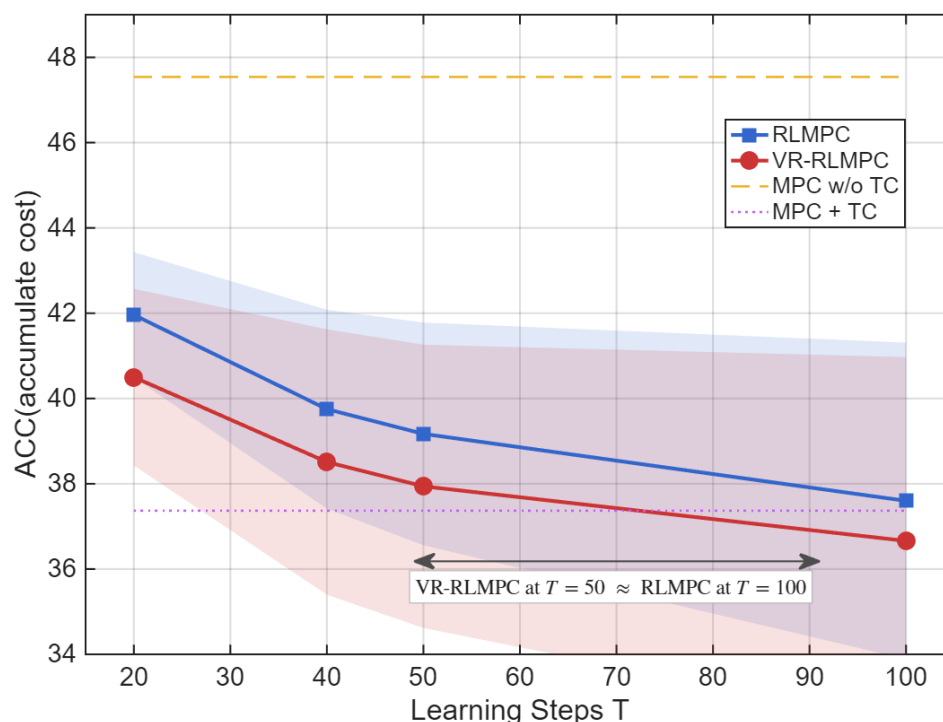


Figure 9. Training-budget comparison at $\kappa = 0.5$ (shaded region: $\pm 1\sigma$). Colors distinguish the methods identified in the legend.

Figure 10 provides a descriptive comparison of dispersion within the same 7-run model-mismatch sweep. The ACC standard deviation of VR-RLMP is lower than that of RLMP at all five tested κ values; the corresponding AVR descriptor decreases from 4.9% at $\kappa = 0.5$ to 1.4% at $\kappa = 2.0$. These values summarize dispersion in the tested runs and are not a probabilistic safety, stability, or tail-risk guarantee.

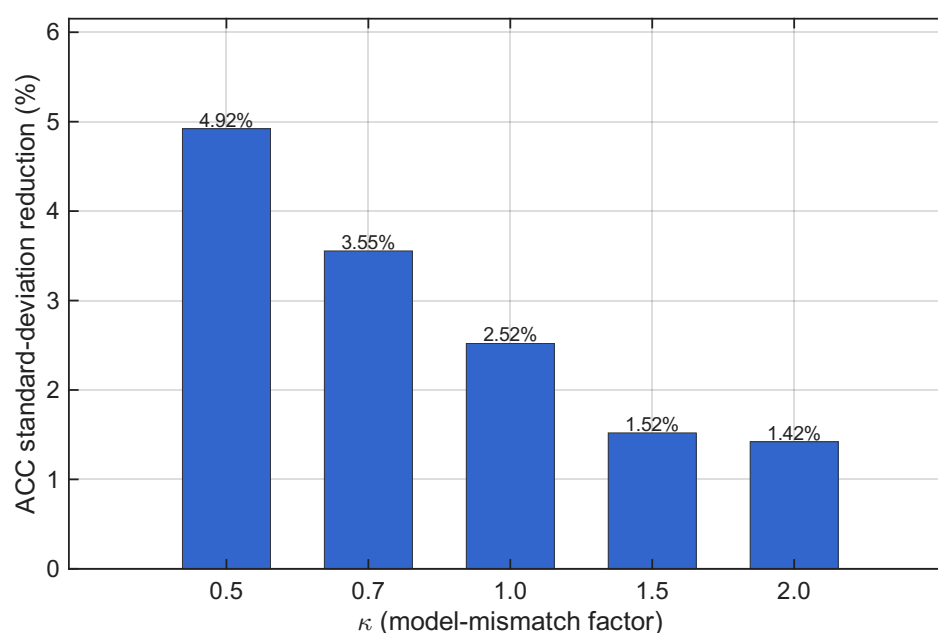


Figure 10. ACC standard-deviation reduction (AVR) under model mismatch.

4.5. β Sensitivity Analysis

All preceding simulations use $\beta = 0.5$. This subsection sweeps $\beta \in \{0, 0.2, 0.5, 1.0, 2.0\}$ under the high-demand configuration (B2+C) with 3 independent repeats (paired design) to examine the effect of β on performance and empirical VFA weight-update settling. Gaussian disturbances ($\sigma = 0.05$) are applied during both training and evaluation for every β value. Figure 11 shows the results.

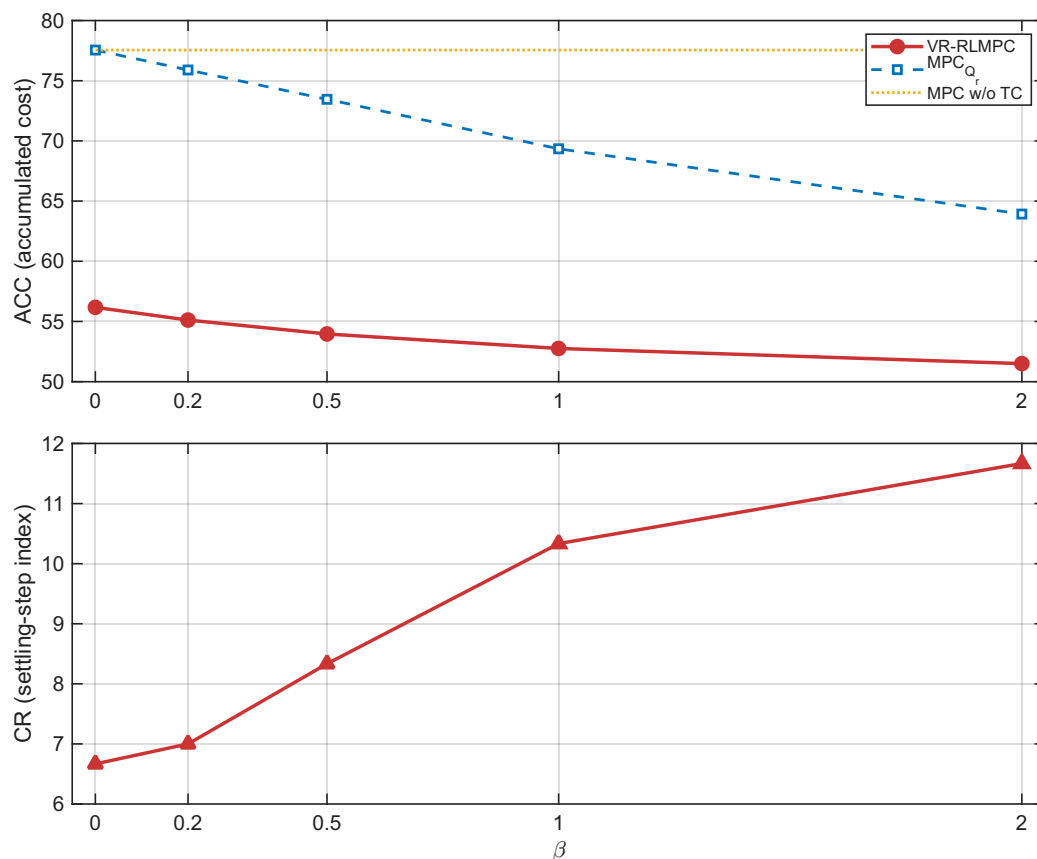


Figure 11. Mean ACC and CR as functions of β .

ACC decreases monotonically with β (i.e., performance improves monotonically), from 56.17 ± 1.30 at $\beta = 0$ to 51.50 ± 2.37 at $\beta = 2.0$ (improvement of 8.3%), with no degradation inflection point within the tested range. Simultaneously, CR increases monotonically from 6.7 ± 1.2 to 11.7 ± 4.7 . An ACC–CR trade-off therefore exists: $\beta = 0.5$ captures approximately 47% of the maximum ACC improvement (+3.9%) while increasing CR by 24%; $\beta = 2.0$ achieves the best ACC but increases CR by 75%. Within the tested range, $\beta = 0.5$ provides an engineering compromise between ACC and CR. All 15 archived runs yield finite ACC and CR records under the reported safeguards. Untested β – α combinations and more dynamic regimes remain outside this single-configuration sweep.

Equation (25) provides a complementary assumption-conditioned interpretation: if the Bellman-residual bound, Hessian bound, and disturbance covariance are held fixed, increasing β raises $\lambda_{\min}(Q + \beta I)$ and reduces the radius r_{β} outside which the conditional one-step drift upper bound is negative. This coefficient-level relation is not a formal stability margin, and neither it nor the single-configuration sweep constitutes a multi-configuration transient-response analysis.

At the same β , VR-RLMPC consistently outperforms the pure cost-modification baseline MPC ($Q + \beta I$, no VFA) in the tested sweep: at $\beta = 0.5$ the gap reaches 26.6% (53.96 vs 73.46). This comparison supports the interpretation that the observed improve-

ment depends on the interaction between regularization and online VFA learning, rather than on increasing the stage-cost weight alone.

To provide an implementation-level timing reference, the archived simulation outputs report ACT, defined as the average solver time per MPC step. Table 4 summarizes representative ACT values.

Table 4. Solver timing across representative evaluation scenarios (ACT, seconds per MPC step).

Scenario	RLMPC ACT (s)	VR-RLMPC ACT (s)
Model mismatch	0.0173–0.0211	0.0241–0.0256
High-demand scenario	0.03997 ± 0.00081	0.04515 ± 0.00184
Limited training budget	0.0459–0.0519	0.0524–0.0605

Mismatch and training-budget entries are ranges across tested κ and T settings.

VR-RLMPC has slightly higher average solver time than RLMPC in these runs. All representative mean ACT values remain below the 0.2 s simulation sampling interval, and β adds no online decision variables, constraints, or sampling loops. This is encouraging for transfer, but ACT is an average solver-timing measurement rather than a worst-case end-to-end latency. Embedded real-time feasibility requires implementation on a target processor, integration with the sensing and state-estimation stack, and platform-specific worst-case timing tests.

4.6. Scope and Limitations

The results establish the behavior of the proposed controller in simulation, not the performance of a complete autonomous mobile robot. The simulated controller receives the state directly, whereas a physical platform would obtain feedback through onboard sensors, calibration, filtering, and state estimation. Measurement noise, estimator delay, state-estimation error, sensor faults, and multi-sensor fusion can alter the effective closed-loop disturbance and are not represented by the additive process disturbance used here. The theoretical result is also conditional on the stated VFA, feasibility, domain-containment, and bounded-support assumptions; recursive feasibility and a trajectory-level guarantee for untruncated Gaussian disturbances remain open. Accordingly, the conclusions are limited to the state-feedback simulations and the conditional analytical model; sensor-integrated validation is required before extending them to a complete autonomous mobile robot.

The evaluation does not cover heavy-tailed or biased disturbances, hardware-in-the-loop operation, or physical robot tests. Actuator saturation, backlash, wheel slip, communication delay, and estimator–controller coupling may require explicit robust constraints or safety filters. The isotropic regularization term adds no direction-specific priority beyond Q , R , f , $\mathbb{X}$, and $\mathbb{U}$; anisotropic regularization is a natural extension. The finite β sweep also leaves high-gain updates and untested β – α combinations for future study.

The numerical comparison covers the implemented RLMPC, MPC, LQR, MPC_{Q_r}, and stochastic-TD baselines. Robust MPC and DRL–MPC studies with different uncertainty models, learned objects, and evaluation protocols are used for literature positioning; matched numerical comparison requires separate implementations.

5. Conclusions

This work introduces isotropic state regularization into the reinforcement learning-based model predictive control (RL-MPC) framework [33], yielding value-regularized RLMPC (VR-RLMPC). By inserting the same β term into the physical state penalty and the N -step TD target, VR-RLMPC couples the control and learning objectives without adding online MPC variables, constraints, or sampling operations. The MPC_{Q_r} ablation indicates

that changing the stage-cost matrix alone does not reproduce the learned-terminal-value result. Under explicit Bellman-residual, feasibility, domain-containment, and bounded-disturbance assumptions, the analysis gives a conditional one-step expected Lyapunov-drift bound. Simulations on a nonlinear nonholonomic vehicle system show lower empirical CR in the nominal and model-mismatch comparisons at $\beta = 0.5$, while the sensitivity sweep identifies an ACC–CR trade-off. Nominal ACC reductions remain modest, and the larger high-demand improvement is scenario-specific and single-digit.

These findings are limited to the stated analytical assumptions and state-feedback simulations. They do not establish recursive feasibility, trajectory-level guarantees for untruncated Gaussian disturbances, embedded worst-case timing, or sensor-integrated and physical-robot performance. Future work will investigate trajectory-level probabilistic bounds, finite-sample VFA behavior, joint β –learning-rate design, and adaptive or anisotropic regularization. It will also integrate onboard sensing and state estimation, evaluate platform-specific embedded timing, and validate the controller on a physical unmanned mobile-robot testbed.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s26144620/s1>, Supplementary File S1: Minimal dataset supporting the reported simulation results, including raw and processed data, metadata, and MATLAB scripts for regenerating the processed summaries and verifying the central numerical claims.

Author Contributions: Conceptualization, C.Y.; methodology, C.Y.; software, C.Y.; validation, C.Y.; formal analysis, C.Y.; data curation, Q.L.; writing—original draft preparation, C.Y.; writing—review and editing, W.Z., Y.C., Z.L. and Q.L.; supervision, W.Z., Y.C., Z.L. and Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The minimal dataset supporting the central findings of this study is provided as Supplementary Material. It contains raw and processed simulation results, metadata, and a MATLAB verification script. The complete source code is not publicly available.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rawlings, J.B.; Mayne, D.Q.; Diehl, M.M. *Model Predictive Control: Theory, Computation, and Design*, 2nd ed.; Nob Hill Publishing: Madison, WI, USA, 2017.
2. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Sokaert, P.O. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814.
3. Cao, Y.; Ni, K.; Kawaguchi, T.; Hashimoto, S. Path following for autonomous mobile robots with deep reinforcement learning. *Sensors* **2024**, *24*, 561. <https://doi.org/10.3390/s24020561>.
4. Ou, Y.; Cai, Y.; Sun, Y.; Qin, T. Autonomous navigation by mobile robot with sensor fusion based on deep reinforcement learning. *Sensors* **2024**, *24*, 3895. <https://doi.org/10.3390/s24123895>.
5. Gutierrez, H.; Cornejo, J.; Bertaska, I. uVGS-2: The Micro Video Guidance Sensor: A 6-DoF Robust Pose Estimator for Autonomous Proximity Maneuvers in Drones, Spacecraft and Mobile Robot Navigation. *Drones* **2026**, *10*, 535. <https://doi.org/10.3390/drones10070535>.
6. Langson, W.; Chrysoschoos, I.; Raković, S.V.; Mayne, D.Q. Robust model predictive control using tubes. *Automatica* **2004**, *40*, 125–133.
7. Cannon, M.; Kouvaritakis, B.; Raković, S.V.; Cheng, Q. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Trans. Autom. Control* **2011**, *56*, 194–200.
8. Mesbah, A. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Syst. Mag.* **2016**, *36*, 30–44.

9. Paulson, J.A.; Lucia, S.; Findeisen, R. Stochastic model predictive control with joint chance constraints. *Int. J. Robust Nonlinear Control* **2020**, *30*, 4793–4817.
10. Arcari, E.; Iannelli, A.; Carron, A.; Zeilinger, M.N. Stochastic MPC with robustness to bounded parametric uncertainty. *IEEE Trans. Autom. Control* **2023**, *68*, 7601–7615.
11. Bujarbaruah, M.; Zhang, X.; Tanaskovic, M.; Borrelli, F. Adaptive MPC under time varying uncertainty: Robust and stochastic. *IEEE Trans. Autom. Control* **2021**, *66*, 2572–2587.
12. Köhler, J.; Soloperto, R.; Müller, M.A.; Allgöwer, F. A computationally efficient robust model predictive control framework for uncertain nonlinear systems. *IEEE Trans. Autom. Control* **2021**, *66*, 794–801.
13. Jiang, Z.P.; Wang, Y. Input-to-state stability for discrete-time nonlinear systems. *Automatica* **2001**, *37*, 857–869.
14. Limon, D.; Alamo, T.; Salas, F.; Camacho, E. Input-to-state stability: A unifying framework for robust model predictive control. *Automatica* **2006**, *42*, 797–803.
15. Allan, D.A.; Bates, C.N.; Risbeck, M.J.; Rawlings, J.B. On the inherent robustness of optimal and suboptimal nonlinear MPC. *Syst. Control Lett.* **2017**, *106*, 68–78.
16. Koller, T.; Berkenkamp, F.; Turchetta, M.; Krause, A. Learning-based model predictive control for safe exploration. In *Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018*; IEEE: New York, NY, USA, 2018; pp. 6059–6066.
17. Hewing, L.; Kabzan, J.; Zeilinger, M.N. Cautious model predictive control using Gaussian process regression. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 2736–2743.
18. Berkenkamp, F.; Turchetta, M.; Schoellig, A.P.; Krause, A. Safe model-based reinforcement learning with stability guarantees. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30.
19. Wabersich, K.P.; Zeilinger, M.N. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica* **2021**, *129*, 109597.
20. Hewing, L.; Wabersich, K.P.; Menner, M.; Zeilinger, M.N. Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 269–296.
21. Gros, S.; Zanon, M. Data-driven economic NMPC using reinforcement learning. *IEEE Trans. Autom. Control* **2020**, *65*, 636–648.
22. Zanon, M.; Gros, S. Safe reinforcement learning using robust MPC. *IEEE Trans. Autom. Control* **2021**, *66*, 3638–3652.
23. Gros, S.; Zanon, M. Learning for MPC with stability & safety guarantees. *Automatica* **2022**, *146*, 110598.
24. Rosolia, U.; Borrelli, F. Learning how to autonomously race a car: A predictive control approach. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 2713–2719.
25. Soloperto, R.; Müller, M.A.; Allgöwer, F. Guaranteed closed-loop learning in model predictive control. *IEEE Trans. Autom. Control* **2023**, *68*, 991–1006.
26. Bertsekas, D.P. *Dynamic Programming and Optimal Control*, 4th ed.; Athena Scientific: Nashua, NH, USA, 2017; Volume 1.
27. Lewis, F.L.; Vrabie, D.; Vamvoudakis, K.G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Syst. Mag.* **2012**, *32*, 76–105.
28. Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2042–2062.
29. Al-Tamimi, A.; Lewis, F.L.; Abu-Khalaf, M. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2008**, *38*, 943–949.
30. Zhang, Y.; Chen, B.; Wang, Y.; Li, N. Cloud-assisted nonlinear model predictive control with deep reinforcement learning for autonomous vehicle path tracking. *Actuators* **2025**, *14*, 609. <https://doi.org/10.3390/act14120609>.
31. Li, M.; Li, H.; Yao, Y.; Zhu, Y.; Weng, H.; Jin, H.; Yang, T. Real-time cooperative path planning and collision avoidance for autonomous logistics vehicles using reinforcement learning and distributed model predictive control. *Machines* **2026**, *14*, 27. <https://doi.org/10.3390/machines14010027>.
32. Geist, M.; Scherrer, B.; Pietquin, O. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2019; Volume 97, pp. 2160–2169.
33. Lin, M.; Sun, Z.; Xia, Y.; Zhang, J. Reinforcement learning-based model predictive control for discrete-time systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 3312–3324.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.