ACCSCIENCE PUBLISHING
An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.16, No.4, pp.1273-1291 (2026)
https://doi.org/10.36922/IJOCTA026190075
IJOCTA

RESEARCH ARTICLE

Q-learning as a feedback control mechanism in deterministic systems

Utti M. Rifanti[1,2], Lina Aryati[1], Nanang Susyanto[1*], and Hadi Susanto[3]

[1]Department of Mathematics, Universitas Gadjah Mada, Yogyakarta, Indonesia
[2]Telecommunications Engineering Study Program, Universitas Telkom, Purwokerto Campus, Purwokerto, Indonesia
[3]Department of Mathematics, Khalifa University, Abu Dhabi, United Arab Emirates
utti.marina.r@mail.ugm.ac.id, lina@ugm.ac.id, nanang_susyanto@ugm.ac.id, hadi.susanto@ku.ac.ae

ARTICLE INFO

Article History:
Received: May 4, 2026
Revised: May 27, 2026
Accepted: May 29, 2026
Published Online: June 9, 2026

Keywords:
Q-learning
Feedback control
Deterministic discrete-time systems
Dynamical systems
Infinite visitation
State recurrence

AMS Classification 2010:
68T05; 90C40; 93C55
93E20; 93E35; 37B20

ABSTRACT

This paper studies tabular Q-learning when implemented as a feedback controller in deterministic discrete-time systems, where the learning process interacts with the system dynamics to generate closed-loop trajectories. Unlike classical convergence results, which assume infinite visitation as an external sampling condition, this property arises from the interaction between recurrence of the induced state dynamics and persistent exploration under GLIE-compliant ϵ-greedy policies. In particular, we prove that exploration remains sufficiently persistent along recurrent states, implying infinite visitation of all state-action pairs within the recurrent region. The results provide a dynamical-systems characterization of the sampling condition underlying classical Q-learning convergence and establish a formal connection between reinforcement learning and recurrence analysis in discrete-time systems. Numerical experiments on nonlinear epidemic dynamics and linear quadratic control benchmarks illustrate the resulting closed-loop behavior.

1. Introduction

Reinforcement learning (RL) is increasingly used for data-driven feedback control of discrete-time dynamical systems.[1–5] In such settings, the learning algorithm operates without explicit knowledge of the system model, while the system dynamics generate the state transitions used for learning. In this context, Q-learning can be viewed as a feedback controller that interacts with the system to generate closed-loop trajectories.

Q-learning plays a central role due to its simplicity and well-established convergence guarantees.[6–8] In particular, Watkins and Dayan[9] establish convergence under the assumption that all state-action pairs are visited infinitely often, but do not specify how this condition is realized by an explicit behavior policy. Similarly, Tsitsiklis[10] analyzes asynchronous Q-learning within a stochastic approximation framework and assumes sufficient updates of all state-action pairs without modeling the policy that generates these samples. From a dynamic programming perspective, Bertsekas[11] studies convergence under general sampling assumptions, but does not address how exploration policies operate when learning is embedded in a closed-loop system. As a result, the dynamical origin of the infinite-visitation condition remains unclear.

In reinforcement learning, the exploration-exploitation trade-off has been studied from multiple perspectives. Classical approaches rely on simple randomized policies, such as ϵ-greedy strategies, to ensure sufficient exploration.[12–16]

*Corresponding Author

However, these frameworks primarily focus on policy design and do not address how the resulting state-action visitation patterns emerge from the interaction between the policy and the system dynamics in closed-loop operation.

Recent developments in RL-based control have extended beyond tabular methods to include deep Q-networks (DQN), actor-critic architectures, safe reinforcement learning, constrained RL, and Lyapunov-based learning frameworks. Deep RL approaches have been applied to nonlinear and robotic control problems, including humanoid robot push-recovery and stabilization tasks using DQN-based control strategies.[17,18] Actor-critic and safe RL frameworks further address continuous control, stability, and safety requirements through policy improvement mechanisms, constraint handling, and Lyapunov- or barrier-function-based formulations.[19,20] These developments highlight the growing role of modern reinforcement learning in control-oriented applications. Nevertheless, the present work focuses on the tabular setting, where the infinite-visitation condition underlying the classical Watkins-Dayan convergence framework can be formulated and analyzed rigorously.

From a feedback control perspective, when Q-learning is applied to deterministic discrete-time systems, the data used for learning are generated by the induced state trajectories, whose structure depends on both the system dynamics and the exploration policy. In this setting, infinite visitation must arise from their interaction rather than being imposed externally. This leads to a fundamental question: how can the infinite-visitation condition be characterized in terms of the dynamical properties of the resulting closed-loop trajectories?

This paper provides a dynamical-systems characterization of the infinite-visitation requirement underlying classical Q-learning convergence theory in deterministic settings. In particular, we interpret Q-learning as a feedback-driven process whose behavior is governed by the induced closed-loop dynamics. While existing results assume that all state-action pairs are visited infinitely often, they do not explain how this condition arises when learning operates in a closed-loop dynamical system. We show that infinite visitation can be understood through two complementary mechanisms: recurrence of the induced state dynamics and persistent exploration ensured by GLIE-compliant ϵ-greedy policies. Under a mild condition on how frequently states are revisited, this interaction guarantees that exploration persists along recurrent trajectories. As a result, the classical infinite-visitation assumption emerges naturally from the system dynamics, without requiring additional ad hoc conditions on the exploration schedule.

The main contributions of this paper are as follows:

(1) We derive the classical infinite-visitation requirement from recurrence of the induced closed-loop dynamics and GLIE exploration.
(2) We establish a formal bridge between recurrence analysis in dynamical systems and convergence theory for tabular Q-learning.
(3) We provide a control-oriented interpretation of Q-learning as a feedback mechanism in deterministic discrete-time systems.

The scope of this work is restricted to the classical tabular setting with finite state and action spaces. This restriction is deliberate, as the objective is to explain the dynamical origin of the infinite-visitation condition that underlies the Watkins-Dayan convergence theorem.[9] Extensions to function approximation and deep reinforcement learning involve additional analytical challenges and are outside the scope of the present study.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation and recalls the classical convergence conditions for tabular Q-learning. Section 3 develops the main theoretical results. Section 4 presents numerical experiments and comparisons with MPC and LQR benchmarks. Section 5 discusses the theoretical and practical implications of the results, together with their limitations. Section 6 concludes the paper.

2. Methods

In this work, we study tabular Q-learning with ϵ-greedy exploration when implemented as a feedback controller in deterministic discrete-time systems. The objective is to analyze how the interaction between closed-loop system dynamics and persistent exploration influences the infinite-visitation property underlying classical Q-learning convergence theory. To this end, we formulate the learning process within a deterministic control framework and introduce the recurrent-state structure used in the subsequent analysis.

2.1. Optimal control problem

We consider a nonlinear discrete-time dynamical system of the form

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

where $x_k \in \mathbb{R}^n$ denotes the system state and $u_k \in \mathbb{R}^m$ denotes the control input at time $k \in \mathbb{N}$. The function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is assumed to be continuous, ensuring that closed-loop trajectories are well defined. We assume that **System (1)** is subject to state and input constraints

$$x_k \in \mathcal{X}, \qquad u_k \in \mathcal{U}, \tag{2}$$

where $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ are compact sets containing the origin as an interior point. These constraints define the admissible state and control domains.

To evaluate the performance of a control policy applied to the discrete-time **System (1)**, we consider a discounted infinite-horizon cost functional of the form

$$J(x_0, \pi) = \sum_{k=0}^{\infty} \gamma^k \, \ell(x_k, u_k), \tag{3}$$

where $\gamma \in (0, 1)$ is a discount factor ensuring the convergence of the series for any bounded trajectory, $\ell : \mathcal{X} \times \mathcal{U} \to [0, \infty)$ is the stage cost, $u_k = \pi(x_k)$ is the control action dictated by policy π, and $x_{k+1} = f(x_k, u_k)$ evolves according to **(1)**. The stage cost $\ell(x, u)$ is assumed to be continuous on the compact domain $\mathcal{X} \times \mathcal{U}$ and bounded, i.e., $0 \le \ell(x, u) \le \ell_{\max}$ for some $\ell_{\max} < \infty$. These conditions guarantee that the discounted cost **(3)** is well defined and finite. The goal is to determine a control policy that minimizes the cost **(3)**.

2.2. State recurrence on the recurrent grid region

In deterministic control systems, state recurrence must therefore be addressed explicitly. In this section, we interpret state recurrence in terms of deterministic discrete-time control. In particular, we clarify how policy-level exploration interacts with state recurrence in deterministic discrete-time systems. Throughout this paper, the state and action spaces are assumed to be finite, and the action-value function is represented explicitly as a lookup table.

Let $\mathcal{X}$ and $\mathcal{U}$ denote finite discretized grids embedded in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Let us also consider the deterministic dynamics **(1)**. Classical convergence results require that every state-action pair is visited infinitely often.[9] Therefore, we first clarify how state recurrence should be interpreted in deterministic discrete-time systems with discretized state grids.

Definition 1. *The reachable set is defined as*

$$\mathcal{X}_{\text{reach}} := \left\{ x \in \mathcal{X} : \begin{array}{l} \exists\, x_0 \in \mathcal{X}_0, \\ \exists\, (u_0, \ldots, u_{T-1}) \\ \textit{such that } x_T = x \end{array} \right\}$$

where $\mathcal{X}_0$ denotes the admissible set of initial states.

Under the ϵ_k-greedy behavior policy, the deterministic dynamics combined with random exploration induces a stochastic process on $\mathcal{X}_{\text{reach}}$. State visitation properties must therefore be understood with respect to this induced process.

Definition 2. *The recurrent grid region is defined as*

$$\mathcal{X}_{\text{rec}} := \{ x \in \mathcal{X}_{\text{reach}} : \ \mathbb{P}(x_k = x \ \textit{i.o.}) = 1 \}.$$

The subsequent convergence analysis is restricted to $\mathcal{X}_{\text{rec}}$, since infinite state-action visitation can only be expected at recurrent states of the induced process. The recurrence property depends on the structure of the directed graph induced by the deterministic transition map $x_{k+1} = f(x_k, u_k)$ on the discretized state grid. In particular, recurrence on $\mathcal{X}_{\text{reach}}$ requires that the induced graph admits sufficient connectivity under admissible controls, together with persistent excitation from the exploration policy.

2.3. Reinforcement learning framework

The control problem described in **(1)**–**(3)** can be naturally formulated within the framework of a Markov Decision Process (MDP).[11,21] To align the optimal control formulation with the reinforcement learning framework, we introduce the reward function

$$r(x, u) := -\,\ell(x, u).$$

Under this definition, minimizing the discounted cost **(3)** is equivalent to maximizing the discounted cumulative reward. An MDP is specified by the tuple $(\mathcal{X}, \mathcal{U}, P, r, \gamma)$, where $\mathcal{X}$ denotes the state space, $\mathcal{U}$ denotes the action space, $P(\cdot \mid x, u)$ denotes the state transition kernel, $r(x, u)$ denotes the reward, and $\gamma \in (0, 1)$ denotes the discount factor.[22,23] In the deterministic setting considered in this work, the transition kernel P is induced by the system dynamics **(1)**, so that the next state is uniquely determined by the current state-action pair.

Let Π denote the set of admissible feedback control policies. The optimal value function is defined as

$$V^*(x) = \inf_{\pi \in \Pi} V^{\pi}(x),$$

where the infimum is taken over all admissible feedback policies $\pi : \mathcal{X} \to \mathcal{U}$ that generate well-defined closed-loop trajectories and finite discounted cost.[24] The corresponding optimal cost-action value function (the optimal Q-function)

satisfies the Bellman optimality equation

$$Q^*(x,u) = \ell(x,u) + \gamma \min_{u' \in \mathcal{U}} Q^*\big(f(x,u),u'\big), \quad (4)$$

and an optimal policy π^* can be obtained as

$$\pi^*(x) \in \arg\min_{u \in \mathcal{U}} Q^*(x,u), \forall x \in \mathcal{X}.$$

Equivalently, one may work with the reward action-value function $\widetilde{Q}^*(x,u) = -Q^*(x,u)$, in which case the Bellman operator and greedy policy are written using maximization.

However, when Q-learning is implemented as a feedback controller, the exploration strategy plays a critical role. In closed-loop operation, how the data used for learning is generated by the controller itself. Consequently, whether all state-action pairs are sufficiently sampled depends on the behavior policy. In control-oriented applications, the controller must improve performance while maintaining adequate exploration to learn how the system responds to different actions.[25–29] Q-learning updates the table of cost-action values using sample transitions $(x_k, u_k, \ell_k, x_{k+1})$, where $\ell_k = \ell(x_k, u_k)$ denotes the instantaneous stage cost observed at time k.[22,30]

Definition 3. ***(Tabular Q-learning update).*** *Given a transition sample $(x_k, u_k, \ell_k, x_{k+1})$ at time k, the Q-learning iteration is updated according to*

$$Q_{k+1}(x_k,u_k) = Q_k(x_k,u_k) + \alpha_{N_k(x_k,u_k)}\Big[\ell_k + \gamma \min_{u' \in \mathcal{U}} Q_k(x_{k+1},u') - Q_k(x_k,u_k)\Big]. \quad (5)$$

where $\alpha_{N_k(x_k,u_k)}$ denotes the learning rate associated with the number of visits $N_k(x_k,u_k)$ of the state-action pair (x_k,u_k) up to iteration k, and satisfies $0 < \alpha_{N_k(x_k,u_k)} \leq 1$.

The update rule **(5)** follows the standard off-policy Q-learning recursion written in cost-minimization form, where the action u_k can be generated by a behavior policy different from the greedy target policy $\min_{u'} Q_k(\cdot)$.

2.4. Classical convergence background

To establish the theoretical basis for the subsequent results, we first briefly recall the classical convergence conditions for tabular Q-learning. These conditions are based on stochastic approximation theory. They form the foundation of the Watkins-Dayan convergence theorem.[9] In particular, we summarize the Robbins-Monro step-size assumptions, the GLIE exploration condition, and the associated infinite-update requirement that underlies almost-sure convergence.

First, for convergence, the learning rates must satisfy the Robbins-Monro conditions, which ensure continued learning while the step sizes gradually decrease.[31]

Definition 4. *A sequence of learning rates $\{\alpha_k\}$ satisfies the Robbins-Monro conditions if*

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

In addition to appropriate step sizes, sufficient exploration is required to guarantee convergence of off-policy methods. In particular, Q-learning requires that every state-action pair be visited infinitely often.[9] This requirement is not a property of the update rule itself, but rather a condition imposed on the behavior policy that generates the data. A purely greedy policy always exploits the current Q-value estimates. As a result, some actions may never be explored again, so the infinite-visitation requirement is not satisfied.

To explicitly address the infinite-visitation requirement, we consider an ϵ-greedy behavior policy.[12–16] At each decision time k, the control input is selected using an ϵ_k-greedy policy with respect to the current Q-values Q_k.[32,33] The resulting policy is a state-feedback law that is updated online based on observed state transitions and incurred costs. Given a state x_k, the action is chosen as

$$u_k = \begin{cases} \arg\min_u Q_k(x_k,u), & \text{w.p. } 1-\epsilon_k, \\ \text{Unif}(\mathcal{U}), & \text{w.p. } \epsilon_k. \end{cases} \quad (6)$$

The parameter $\epsilon_k \in [0,1]$ controls the exploration level: with probability $1-\epsilon_k$ the agent selects the greedy action, while with probability ϵ_k it explores by choosing a random action from $\mathcal{U}$.

Remark 1 (Importance of uniform random exploration)**.** *The use of a* uniform *random action during the exploratory phase of the ϵ-greedy policy **(6)** is essential for ensuring persistent action exploration. Uniform exploration guarantees that, for any action $u \in \mathcal{U}$,*

$$\Pr(u_k = u \mid x_k = x) \geq \frac{\epsilon_k}{|\mathcal{U}|} > 0.$$

Consequently, at any state that is visited infinitely often, each action is selected infinitely many times with probability 1, provided that the sequence $\{\epsilon_k\}$ satisfies the GLIE conditions. This ensures that the stochastic approximation updates for each (x,u) pair do not cease prematurely.

If exploration were non-uniform or biased toward a subset of actions, certain actions could receive vanishing selection probability. In such

cases, even at recurrent states, some state-action pairs may be updated only a finite number of times. This situation violates the infinite-update requirement in classical convergence theory. Therefore, uniform random exploration is not only a design choice. It also acts as a structural mechanism that guarantees persistent excitation at recurrent states.

The sequence ϵ_k in **(6)** is designed to satisfy the Greedy in the Limit with Infinite Exploration (GLIE) condition. In particular, Singh[34] studies reinforcement learning under policies that satisfy the GLIE condition. The GLIE condition provides a concrete way to interpret persistent exploration at the policy level. GLIE policies ensure that, at states visited infinitely often, each action is selected with nonzero probability while the policy becomes asymptotically greedy. This condition ensures that actions are explored infinitely often while the policy gradually becomes greedy. In this way, it provides an explicit and implementable mechanism for persistent action exploration at recurrent states, as required by classical convergence theory. We now formally state the definition of the GLIE condition.

Definition 5. *An ϵ-greedy policy is said to satisfy Greedy in the Limit with Infinite Exploration (GLIE) if*

$$\epsilon_k \in [0,1], \qquad \epsilon_k \to 0, \qquad \sum_{k=0}^{\infty} \epsilon_k = \infty.$$

The GLIE condition guarantees asymptotic greediness together with persistent action exploration. At states that are visited infinitely often, each action continues to be selected infinitely many times, while the policy gradually converges toward a greedy policy with respect to the learned Q-function. The convergence result itself does not depend on a particular exploration mechanism, but on the satisfaction of the underlying infinite-visitation requirement. Importantly, the GLIE condition does not by itself imply that all states are visited infinitely often. Rather, it guarantees persistent exploration only at states that are revisited infinitely often. Additional recurrence properties of the induced closed-loop dynamics are therefore required to ensure infinite visitation of state-action pairs.

We adopt an ϵ-greedy policy as a simple and explicit policy that satisfies the GLIE conditions required by the theorem. Together with suitable state-recurrence conditions, this policy provides a concrete and implementable way to realize the classical infinite-update assumption in feedback control settings. The action-selection rule in **(6)** is controlled by the exploration parameter ϵ_k, which sets the probability of choosing a random action at time k. Accordingly, to support the convergence conditions of Q-learning, it is essential to specify how the sequence $\{\epsilon_k\}$ evolves. The GLIE condition specifies how ϵ_k must decay to ensure persistent exploration alongside eventual greedy behavior. In this sense, the GLIE condition governs the admissible evolution of ϵ_k in the ϵ-greedy behavior policy **(6)**.

Consider a finite MDP with state space $\mathcal{X}$, action space $\mathcal{U}$, discount factor $0 < \gamma < 1$, and bounded stage costs $0 \le \ell_k \le \ell_{\max}$. Let $Q_k(x,a)$ denote the tabular Q-learning iterates updated according to **(5)**. In this update scheme, each state-action pair (x,a) has its own learning-rate sequence $\alpha_t(x,a)$. This sequence corresponds to the times when that particular state-action pair is updated. Watkins and Dayan[9] established that tabular Q-learning converges almost surely to the optimal action-value function under the following classical conditions:

(1) uniformly bounded one-step costs,
$$0 \le \ell_k \le \ell_{\max} < \infty,$$
(2) learning rates satisfying the Robbins-Monro conditions
$$\alpha_t(x,a) \in [0,1), \ \sum_t \alpha_t = \infty, \ \sum_t \alpha_t^2 < \infty, \ \forall x, a.$$

The second condition implicitly requires that each state-action pair (x,a) be updated infinitely often. Since the learning rate $\alpha_t(x,a)$ is nonzero only when the pair (x,a) is updated, the condition $\sum_{t=1}^{\infty} \alpha_t(x,a) = \infty$ can hold only if (x,a) is updated infinitely many times. Thus, infinite state-action visitation is implicit in the stochastic approximation requirement.

3. Results

We now formalize the preceding discussion by analyzing how the infinite-visitation requirement arises in deterministic discrete-time systems. When Q-learning is implemented as a feedback policy, the sequence of state-action updates is no longer generated exogenously, but is induced by the closed-loop interaction between the system dynamics and the exploration mechanism. In this setting, the classical assumption of infinite visitation must be interpreted in terms of structural properties of the resulting trajectories rather than imposed independently. To this end, we study how two key components—recurrence of the induced state dynamics and persistent exploration under GLIE policies—jointly determine whether the infinite-visitation condition is satisfied.

3.1. Problem setting and assumptions

We first isolate the role of the exploration policy at visited states, and then combine it with a structural recurrence condition to obtain a dynamical characterization of infinite visitation.

(A1) Bounded one-step costs.
The instantaneous stage cost is uniformly bounded, i.e.,

$$0 \le \ell_k \le \ell_{\max} < \infty \qquad \forall k.$$

(A2) Robbins-Monro stepsizes.
The learning rates satisfy the Robbins-Monro conditions as defined in Definition 4.

(A3) GLIE ϵ-greedy exploration.
The behavior policy is an ϵ-greedy policy with respect to Q_k that satisfies the GLIE conditions defined in Definition 5.

Remark 2 (Dynamical interpretation)**.** *The recurrence property on $\mathcal{X}_{\text{rec}}$ can be interpreted as a structural condition on the discretized deterministic dynamics under closed-loop operation. In particular, recurrence requires sufficient connectivity in the directed graph induced by the transition map $f(x, u)$ under admissible control actions. This connectivity ensures that, under persistent exploration, the system can repeatedly revisit states and excite all outgoing transitions within the recurrent region. Importantly, this structural property depends on the interaction between the system dynamics and the exploration policy, and is not implied by the GLIE condition alone.*

Assumptions (A1)–(A3) are standard in the convergence analysis of tabular Q-learning and serve complementary roles. The bounded one-step cost condition in Assumption (A1) ensures that the temporal-difference updates have finite variance. The Robbins-Monro conditions in Assumption (A2) ensure that learning continues over time while the step sizes gradually decrease. Assumption (A3) governs the exploration mechanism, but does not impose any recurrence property on the state process. Recurrence is treated separately through the recurrent region $\mathcal{X}_{\text{rec}}$.

Assumptions (A1))–((A3) correspond to the standard conditions in classical Q-learning convergence theory. However, when learning is embedded in a deterministic closed-loop system, the infinite-visitation requirement must be understood in terms of the interaction between the exploration policy and the induced state dynamics. In particular, we show that the infinite-visitation requirement can be derived from the interplay between GLIE exploration and recurrence of the induced dynamics, under an explicit regularity condition on the recurrence frequency. Instead, this property follows directly from the GLIE condition when combined with infinite recurrence of the state.

3.2. Infinite visitation: Exploration and recurrence

We first analyze the behavior of the GLIE exploration sequence along the visit times of a recurrent state. This isolates the role of the exploration policy independently of the mechanism that generates state recurrence.

Lemma 1 (Divergence of exploration along recurrent visits)**.** *Let $\{\epsilon_k\}_{k\ge 0}$ be a nonincreasing sequence satisfying the GLIE conditions*

$$\epsilon_k \to 0, \qquad \sum_{k=0}^{\infty} \epsilon_k = \infty.$$

Let $x \in \mathcal{X}$ be a state that is visited infinitely often, and let $\{\tau_n(x)\}_{n\ge 1}$ denote the corresponding visit times. Assume that the visits to x are not asymptotically too sparse in the sense that there exists a constant $C_x > 0$ such that

$$\tau_n(x) \le C_x n \qquad \text{for all sufficiently large } n.$$

Then

$$\sum_{n=1}^{\infty} \epsilon_{\tau_n(x)} = \infty.$$

Proof. Since $\{\epsilon_k\}_{k\ge 0}$ is nonincreasing and $\tau_n(x) \le C_x n$ for all sufficiently large n, we have

$$\epsilon_{\tau_n(x)} \ge \epsilon_{\lceil C_x n \rceil}$$

for all sufficiently large n.

Therefore,

$$\sum_{n=1}^{\infty} \epsilon_{\tau_n(x)} \ge \sum_{n=N}^{\infty} \epsilon_{\lceil C_x n \rceil}$$

for some sufficiently large N.

Since $\sum_{k=0}^{\infty} \epsilon_k = \infty$ and $\{\epsilon_k\}$ is nonincreasing, the subseries

$$\sum_{n=N}^{\infty} \epsilon_{\lceil C_x n \rceil}$$

also diverges. Hence,

$$\sum_{n=1}^{\infty} \epsilon_{\tau_n(x)} = \infty.$$

□

The following result shows that GLIE exploration ensures persistent action excitation at states that are visited infinitely often.

Lemma 2 (GLIE ensures infinite action exploration conditional on infinite state visits)**.** *Suppose the behavior policy is ϵ_k-greedy and satisfies the GLIE condition. Fix any state $x \in \mathcal{X}$ and let*

$\{\tau_n(x)\}_{n\geq 1}$ denote its visit times. If x is visited infinitely often, then for every action $u \in U$, the pair (x, u) is selected infinitely often with probability one.

Proof. At each visit time $\tau_n(x)$, the $\epsilon_{\tau_n(x)}$-greedy policy selects action u with probability at least $\epsilon_{\tau_n(x)}/|U|$.

By Lemma 1, we have

$$\sum_{n=1}^{\infty} \epsilon_{\tau_n(x)} = \infty.$$

Hence,

$$\sum_{n=1}^{\infty} \mathbb{P}(u_{\tau_n(x)} = u \mid \mathcal{F}_{\tau_n(x)}) = \infty.$$

A conditional Borel-Cantelli argument implies that u is selected infinitely often almost surely. □

Lemma 2 shows that, under the divergence condition established in Lemma 1, GLIE ensures persistent exploration at visited states. However, infinite visitation of state-action pairs also requires that the underlying dynamics revisit states infinitely often. We now combine the recurrence property of the induced dynamics with the exploration properties ensured by GLIE policies.

Theorem 1 (Dynamical characterization of infinite visitation)**.** *Consider tabular Q-learning applied to the deterministic discrete-time system*

$$x_{k+1} = F(x_k, u_k),$$

on a finite state space $\mathcal{X}$ and action set $\mathcal{U}$.

Suppose Assumptions (A1)–(A3) hold, and let X_{rec} denote the recurrent grid region. Then, for every $(x, u) \in \mathcal{X}_{\text{rec}} \times \mathcal{U}$, the state-action pair (x, u) is visited infinitely often almost surely.

Proof. Fix any $x \in \mathcal{X}_{\text{rec}}$ and $u \in \mathcal{U}$. By definition of X_{rec}, the state x is visited infinitely often almost surely.

By Lemma 1, we have

$$\sum_{n=1}^{\infty} \epsilon_{\tau_n(x)} = \infty.$$

Then, by Lemma 2, the action u is selected infinitely often at state x almost surely. Since x and u are arbitrary, the result follows. □

As a direct consequence of the above characterization, the classical convergence result of Watkins and Dayan applies.

Corollary 1 (Almost-sure convergence under GLIE exploration)**.** *Under Assumptions (A1)–(A3), every $(x, u) \in \mathcal{X}_{\text{rec}} \times \mathcal{U}$ is updated infinitely often almost surely. Consequently, the classical Watkins-Dayan convergence theorem applies.*

Corollary 1 shows that the classical Watkins-Dayan convergence result holds under GLIE-compliant exploration. To better understand this result, we next explain the role of the exploration conditions and discuss what can go wrong when exploration decays too quickly.

3.3. Implications of the theoretical results

The sequence of results developed in Section 3.2 provides a step-by-step characterization of how the classical infinite-visitation requirement arises in closed-loop Q-learning. Lemma 1 establishes a key analytical property of the exploration schedule along the visit times of a recurrent state. Specifically, under a mild regularity condition on the growth of visit times, the GLIE exploration sequence remains sufficiently persistent when restricted to those times, in the sense that the accumulated exploration probability along visits diverges. Building on this result, Lemma 2 translates this persistence into an action-level statement: if a state is visited infinitely often, then every action continues to be selected infinitely often with probability one under a GLIE ϵ-greedy policy. These two lemmas isolate the role of the exploration mechanism at recurrent states, independent of how such states are generated.

Theorem 1 then combines the exploration property established above with the recurrence structure of the induced closed-loop dynamics. By restricting attention to the recurrent grid region, where states are revisited infinitely often by definition, the theorem shows that infinite visitation of all state-action pairs follows from the interaction between recurrence and GLIE exploration. In particular, neither mechanism alone is sufficient: recurrence ensures repeated state visits, whereas GLIE guarantees continued action exploration at those recurrent states. Finally, Corollary 1 connects this characterization to the classical convergence theory by showing that, once infinite visitation is established on the recurrent region, the Watkins-Dayan convergence result applies directly.

While the preceding results establish sufficient conditions for infinite visitation, it is equally important to understand how these conditions may fail in practice. Proposition 1 illustrates a simple failure case in which the infinite-exploration requirement is violated when the exploration probability decays too rapidly.

Proposition 1 (Failure of infinite exploration under fast decay)**.** *Consider an ϵ_k-greedy behavior policy. If the exploration sequence satisfies $\sum_{k=0}^{\infty} \epsilon_k < \infty$, then with probability one, only finitely many exploratory actions are taken. In this case, there is no mechanism that forces the agent to continue sampling non-greedy actions. As a result, some state-action pairs may be updated only a finite number of times, so the infinite-visitation requirement in the classical convergence theorem may not be satisfied.*

Proof. Let E_k denote the event that exploration occurs at time k. Since $\Pr(E_k) = \epsilon_k$ and $\sum_k \epsilon_k < \infty$, the first Borel-Cantelli lemma[35] implies that $\Pr(E_k \text{ i.o.}) = 0$. Hence, exploration occurs only finitely many times almost surely, which may prevent infinite updates of some state-action pairs. □

For clarity and reproducibility, we summarize the standard tabular off-policy Q-learning algorithm with an explicit GLIE exploration policy. **Algorithm 1** shows how the abstract exploration assumptions in classical convergence analysis can be implemented using a practical behavior policy for feedback control. With bounded one-step costs and Robbins-Monro step sizes, the algorithm satisfies the conditions of the classical convergence theorem when the exploration sequence ϵ_t follows the GLIE condition.

The independent random restart distribution μ_0 is used to sample initial states at the beginning of each episode. This mechanism improves coverage of the discretized state space and enlarges the portion of the grid that is visited with positive limiting frequency.

In Section 4, we illustrate this mechanism numerically by comparing GLIE-compliant exploration schedules with faster-decaying non-GLIE schedules.

4. Numerical experiments

The numerical experiments examine how Q-learning performance changes during training. They also assess whether the learned policy approaches the performance of MPC under stationary epidemic dynamics. We report learning curves, gap-to-baseline metrics, and closed-loop state trajectories to illustrate convergence behavior.

For reproducibility, all numerical parameters used in the SIR and LQR experiments are summarized in **Table 1**.

Algorithm 1 Episodic tabular off-policy Q-learning with i.i.d. random restarts and GLIE ϵ-greedy exploration

Require: Finite state grid $\mathcal{X}$, finite action set $\mathcal{U}$, discount factor $\gamma \in (0,1)$, bounded initial table Q_0, episode length $T \in \mathbb{N}$, restart distribution μ_0 on $\mathcal{X}_0 \subseteq \mathcal{X}$ (i.i.d. across episodes), GLIE exploration schedule $\{\epsilon_t\}_{t\geq 0}$, bounded cost sequence $0 \leq \ell_t \leq \ell_{\max}$

Ensure: Sequence of Q-tables $\{Q_t\}$

Initialize $Q(x,u) \leftarrow Q_0(x,u)$ for all $(x,u) \in \mathcal{X} \times \mathcal{U}$
Initialize visit counters $N(x,u) \leftarrow 0$ for all $(x,u) \in \mathcal{X} \times \mathcal{U}$
Set global time $t \leftarrow 0$
for $e = 1, 2, \ldots$ **do** ▷ Episodes
  Sample initial state $x \sim \mu_0$
  for $k = 0, 1, \ldots, T-1$ **do**
    Set exploration probability ϵ_t
    Select action u_t using ϵ_t-greedy policy
    Apply u_t, observe cost ℓ_t and next state $x' = F(x, u_t)$
    Update visit counter: $N(x,u_t) \leftarrow N(x,u_t) + 1$
    Set step size α_t (Robbins-Monro)
    Update Q-value **(5)**
    Set $x \leftarrow x'$
    Increment global time: $t \leftarrow t + 1$
  end for
end for

4.1. Q-learning setup with GLIE exploration

The numerical experiments are conducted using a controlled susceptible-infected-recovered (SIR) epidemic model formulated in discrete time. Let $x_k = (S_k, I_k, R_k)$ denote the state at time step k, representing the susceptible, infected, and recovered populations, respectively. The system dynamics are given by[36–38]

$$S_{k+1} = S_k - \Delta t\, \beta(1-u_k) S_k I_k, \tag{7}$$

$$I_{k+1} = I_k + \Delta t\, \beta(1-u_k) S_k I_k - \Delta t\, \gamma I_k, \tag{8}$$

$$R_{k+1} = R_k + \Delta t\, \gamma I_k, \tag{9}$$

where $\beta > 0$ is the transmission rate, $\gamma > 0$ is the recovery rate, and $u_k \in [0,1]$ represents the control input, interpreted as the intensity of intervention measures. The time step Δt is selected to ensure an accurate discrete-time representation of the epidemic dynamics. For tabular Q-learning, the (S, I) state space is discretized into a finite grid, with R_k determined by population conservation. The control input set $\mathcal{U}$ is discretized

into a finite number of admissible intervention levels. This discretization yields a finite-state, finite-action Markov decision process consistent with the assumptions required for Q-learning convergence.

In the present SIR training setup, each training episode begins from an initial state sampled independently from a prescribed restart distribution within a compact region of the discretized state grid. This episodic design prevents the learning process from repeatedly following a single deterministic trajectory and promotes broader exploration of the state space. Consequently, the set $\mathcal{X}_{\text{rec}}$ corresponds to the portion of the discretized state space that is visited with positive limiting frequency under the exploration policy. The subsequent convergence statements are therefore understood to hold on $\mathcal{X}_{\text{rec}}$.

The controller performance is evaluated using the following stage cost:

$$\ell(x_k, u_k) = w_I I_k^2 + w_u u_k^2, \tag{10}$$

where $w_I > 0$ penalizes the infected population and $w_u > 0$ penalizes the control effort. This quadratic cost captures the trade-off between epidemic mitigation and intervention intensity. No terminal cost is imposed. The performance of a policy is evaluated using the cumulative cost induced by the closed-loop trajectory generated by that policy. Specifically, given a stationary policy π, we define the finite-horizon cost

$$J_T^\pi = \sum_{k=0}^{T-1} \ell(x_k, u_k), \qquad u_k = \pi(x_k), \tag{11}$$

where T denotes the simulation horizon and $\ell(x_k, u_k)$ is the stage cost defined in **(10)**.

Table 1. Summary of numerical parameters used in the SIR and LQR experiments

Parameter	**SIR experiment**	**LQR benchmark**
System parameters	$\beta = 0.3$, $\gamma = 0.1$, $\Delta t = 1.0$, $N = 1.0$	$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}, \Delta t = 0.1$
Action space	10 uniformly spaced actions in $[0, 1]$	$\{-1, -0.5, 0, 0.5, 1\}$
Number of actions	10	5
State grid	21×21 grid over (S, I)	21×21 grid over (x_1, x_2)
State ranges	$S \in [0, 1]$, $I \in [0, 1]$	$x_1, x_2 \in [-1.5, 1.5]$
Total tabular states	441	441
Cost weights	$w_I = 1.0$, $w_u = 0.05$, $w_{\Delta u} = 4.65$	$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = [0.1]$
Discount factor Q-learning	$\gamma_{\text{RL}} = 0.95$	$\gamma_{\text{RL}} = 0.95$
Exploration schedule	$\epsilon_t = 1/(1+t)^{0.6}$	$\epsilon_t = 1/(1+t)^{0.6}$
Learning rate	$\alpha_t = 1/(1 + N_t(x, u))$	$\alpha_t = 1/(1 + N_t(x, u))$
Training episodes	5000	5000
Episode length	100	100
Random seeds	1 (seed = 123)	10 independent seeds
Evaluation interval	–	Every 50 episodes
Initial condition	$(0.99, 0.01, 0)$	$(1.5, -1.0)$
Evaluation horizon	160	100
Prediction horizon MPC	$H = 1, 3, 5$	$H = 5$

We employ tabular Q-learning with the update rule **(5)**. Each state-action pair (x, u) is associated with its own learning-rate sequence $\{\alpha_t(x, u)\}$ satisfying the Robbins-Monro conditions in Definition 4. Suppose that visit counts index the learning rates

$$\alpha_k(x, u) = \frac{1}{1 + N_k(x, u)}, \tag{12}$$

with $N_k(x, u)$ denoting the number of visits to the state-action pair (x, u) up to time k. Episodes start from an initial condition $(S_0, I_0) = (0.99, 0.01)$ and

$$R_0 = 1 - S_0 - I_0.$$

We use an ϵ_k-greedy behavior policy **(6)** based on Q_k to implement the GLIE condition and sat-

isfy the exploration assumptions required for convergence. In particular, we consider a polynomially decaying exploration schedule

$$\epsilon_k = \frac{\epsilon_0}{(1+k)^\beta}, \qquad 0 < \beta \leq 1, \tag{13}$$

where $\epsilon_0 > 0$ is the initial exploration rate. For $0 < \beta \leq 1$, this schedule satisfies $\epsilon_k \to 0$ as $k \to \infty$ while $\sum_{k=0}^{\infty} \epsilon_k = \infty$, and therefore fulfills the GLIE conditions and the hypotheses of Lemma 2.

Although our convergence analysis assumes GLIE-compliant exploration, it is useful to compare this setting with exploration schedules that do not satisfy this condition. In practice, fast-decaying ϵ-greedy policies are commonly used, even though they do not guarantee infinite exploration and therefore do not meet classical convergence assumptions. From a theoretical perspective, the difference between GLIE and non-GLIE exploration is whether exploration persists indefinitely. GLIE schedules maintain exploration long enough to support the classical convergence guarantees under the stated assumptions, whereas faster-decay schedules may terminate exploration too early and therefore fall outside the scope of the classical theory. To highlight the role of the decay rate, we contrast the GLIE-compliant schedule **(13)** with faster decaying exploration rules that do not satisfy GLIE. For example, an exponentially decaying schedule

$$\epsilon_k = \epsilon_0 \lambda^k, \qquad 0 < \lambda < 1, \tag{14}$$

satisfies $\epsilon_k \to 0$ but yields $\sum_{k=0}^{\infty} \epsilon_k < \infty$. In this case, exploration vanishes too rapidly and the infinite-visitation assumption required for convergence may be violated. This highlights that not all decaying ϵ-greedy policies are GLIE, and that the decay rate plays a critical role in guaranteeing convergence.

The experiments compare these two exploration schedules. The GLIE-compliant polynomial decay satisfies the theoretical assumptions of Proposition 1. In contrast, the exponential decay violates the infinite-exploration component of the GLIE condition and therefore does not satisfy the sufficient exploration requirement underlying the classical convergence theory. In all experiments, ϵ_0 is initialized close to 1 to encourage early exploration. Figure 1 illustrates how the choice of exploration schedule affects both learning dynamics and closed-loop performance. To monitor convergence of the Q-learning iterates, we report the sup-norm difference

$$\begin{aligned} \Delta Q_k &\triangleq \|Q_{k+1} - Q_k\|_\infty \\ &= \max_{(x,u)} |Q_{k+1}(x,u) - Q_k(x,u)|. \end{aligned} \tag{15}$$

which is the natural metric in classical convergence analyses based on Bellman contraction arguments.

Figure 1a compares a GLIE-compliant exploration schedule with a faster non-GLIE decay that rapidly drives ϵ_k toward zero. Under the GLIE schedule, exploration remains active throughout training, whereas the non-GLIE schedule becomes nearly greedy after relatively few episodes. The effect of these exploration strategies is reflected in **Figure 1b**, which reports the sup-norm difference between successive Q-tables defined in **(15)**. This quantity provides an empirical indicator of the evolution of the learned action-value function. Under GLIE exploration, the Q-value updates remain moderate over time, indicating continued refinement of the value estimates as additional state-action pairs continue to be sampled. In contrast, the non-GLIE schedule causes ΔQ_k to decrease rapidly toward zero, reflecting early stabilization of the Q-table due to diminished exploration. The learning curves in **Figure 1c** exhibit a similar trend. With GLIE exploration, the accumulated episode cost decreases gradually and stabilizes at a lower level. By comparison, the faster non-GLIE decay converges more rapidly but stabilizes at a slightly higher cost, suggesting premature exploitation and reduced policy quality. These observations are consistent with the theoretical role of GLIE exploration in sustaining continued sampling of state-action pairs during learning.

4.2. Learning behavior and performance

In this section, we introduce Model Predictive Control (MPC) as a conceptual and numerical benchmark.[39–41] Because MPC has explicit access to the system model while Q-learning learns solely from sampled transitions, the comparison is intended to assess whether the learned policy exhibits behavior consistent with an established model-based control strategy rather than establish algorithmic superiority. Since the underlying SIR dynamics are nonlinear, the benchmark can be interpreted as a nonlinear MPC (NMPC) formulation solved in a receding-horizon manner.[42]

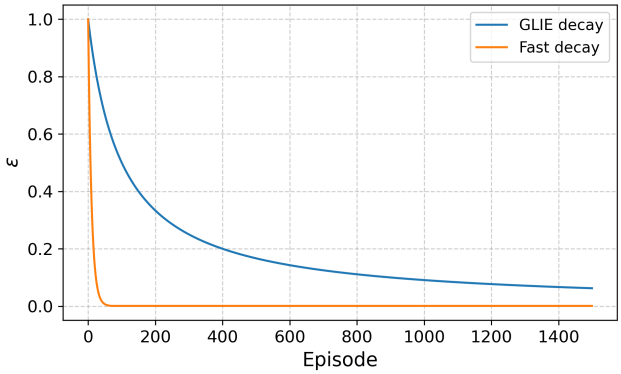
As a model-based reference, we consider a receding-horizon MPC scheme constructed using the same discrete-time SIR dynamics as in **(7)**. At each time step k, MPC assumes full state information and predicts future trajectories starting from the current state $x_k = (S_k, I_k, R_k)$. The MPC controller minimizes the finite-horizon cost

$$J_T^{\pi_{\mathrm{MPC},H}} = \sum_{k=0}^{T-1} \ell(x_k, u_k), \quad u_k = \pi_{\mathrm{MPC},H}(x_k), \tag{16}$$

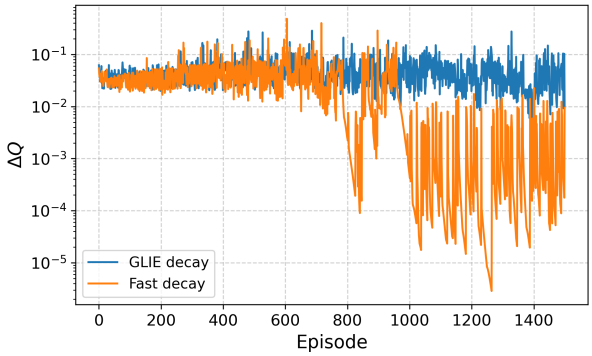
where $H \in \{1, 3, 5\}$ denotes the prediction horizon. The stage cost $\ell(\cdot)$ is chosen consistently with the reinforcement learning formulation and is given by **(10)**. At each time step, the optimization problem **(16)** is solved over the horizon H, and only the first control action is applied in a receding-horizon manner.[43]

All experiments use the same model parameters, costs, constraints, and initial conditions, with only the MPC prediction horizon varied ($H = 1, 3, 5$). Different horizon lengths are included to illustrate the effect of look-ahead depth on MPC performance and to provide multiple model-based reference levels for comparison with Q-learning.

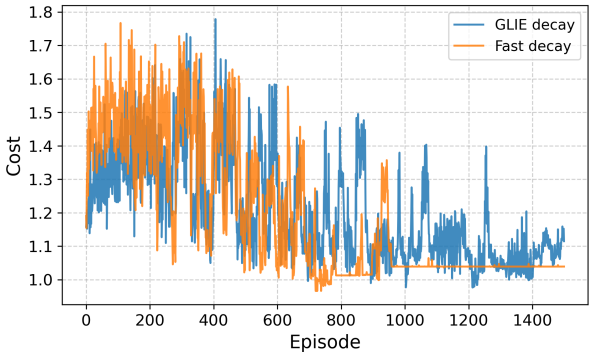
Figure 2 shows the evolution of the finite-horizon cost J_T of the Q-learning controller as a function of the number of temporal-difference (TD) updates. The solid blue curve represents the mean performance across training runs, while the shaded region indicates one standard deviation due to exploration variability. For comparison, the horizontal dashed lines correspond to MPC benchmarks with prediction horizons $H = 1, 3, 5$.



(a) Epsilon decay schedules.



(b) Sup-norm difference of successive Q-tables.



(c) Accumulated cost per episode.

Figure 1. Comparison of ϵ-greedy exploration strategies: (a) exploration schedules, (b) convergence behavior measured by successive Q-table differences, and (c) learning curves measured by accumulated episode cost.

During the early stages of training, the Q-learning controller exhibits relatively high cost values and substantial variability, reflecting the exploratory nature of the learning process. As training progresses, the average cost decreases significantly and the variability across runs becomes smaller, indicating progressive stabilization of the learned policy. After the transient learning phase, the resulting performance becomes comparable to the MPC benchmarks with horizons $H = 3$ and $H = 5$, while achieving lower cost than the short-horizon benchmark $H = 1$.

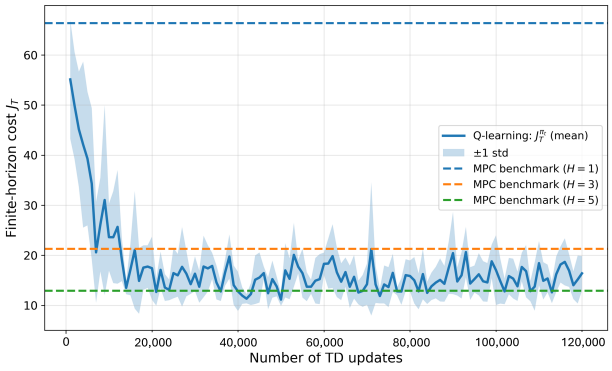


Figure 2. Learning curve of Q-learning showing the finite-horizon cost versus the number of TD updates.

Figure 3 shows the performance gap

$$\Delta_H(t) = J_T^{\pi_t} - J_T^{\pi_{\mathrm{MPC},H}},$$

between the learned Q-learning policy and MPC benchmarks with different prediction horizons. The horizontal dashed line at $\Delta_H(t) = 0$ indicates equal performance relative to the corresponding MPC controller, with negative values representing lower cost achieved by Q-learning. During the early stages of training, the performance gap is relatively large and exhibits substantial variability across random seeds. As the number of TD updates increases, the gap decreases significantly for all MPC horizons, indicating progressive improvement of the learned policy relative to the model-based benchmarks.

Figure 3. Gap-to-benchmark plot showing $\Delta_H(t) = J_T^{\pi_t} - J_T^{\pi_{\mathrm{MPC},H}}$ versus the number of TD updates for different MPC horizons. The shaded regions indicate mean ± one standard deviation across repeated runs.

For the short-horizon benchmark ($H = 1$), the gap becomes consistently negative after the transient learning phase, showing that the learned policy achieves lower cost than the corresponding MPC controller. For moderate and longer horizons ($H = 3, 5$), the gap progressively approaches zero, indicating that the learned policy attains performance comparable to MPC with longer look-ahead capability. Although the longest-horizon benchmark ($H = 5$) generally maintains the lowest cost, the remaining gap becomes relatively small after sufficient training. Overall, **Figure 3** indicates that Q-learning with GLIE ϵ-greedy exploration progressively reduces the performance gap relative to MPC and eventually achieves closed-loop behavior comparable to model-based control benchmarks across multiple prediction horizons.

While cost-based metrics provide a compact summary of performance, it is equally important to assess the behavior of the learned Q-learning policy. Now, we compare its closed-loop performance against an MPC benchmark. The purpose of this comparison is not to establish the theoretical dominance of reinforcement learning over MPC, but to provide a well-understood control-theoretic reference for interpreting the learned policy.

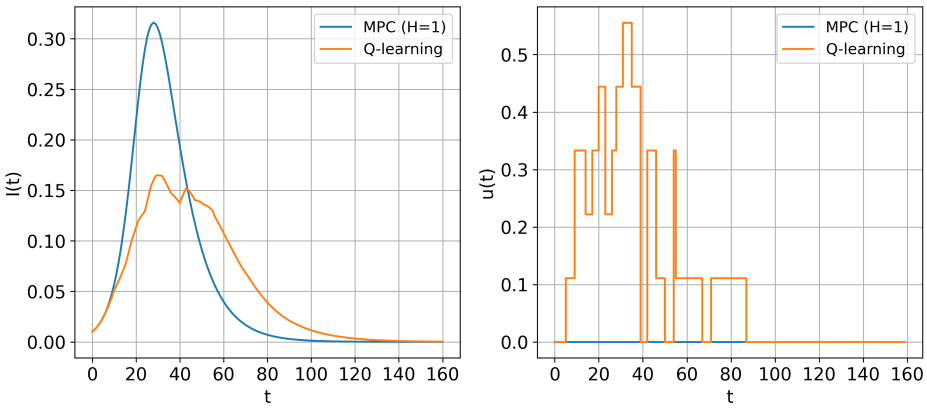
Figure 4 illustrates representative closed-loop responses generated by the learned Q-learning (RL) controller and MPC controllers with different prediction horizons ($H = 1, 3, 5$). Each row corresponds to a specific MPC horizon. The left panel shows the evolution of the infected population $I(t)$, while the right panel depicts the corresponding control input $u(t)$.

Figure 4a corresponds to the short-horizon MPC controller ($H = 1$). In this case, the closed-loop responses exhibit distinct transient behaviors: MPC produces a sharper initial infection peak and smoother control actions, while the RL policy yields a more gradual response and more discrete control inputs during the early stages.

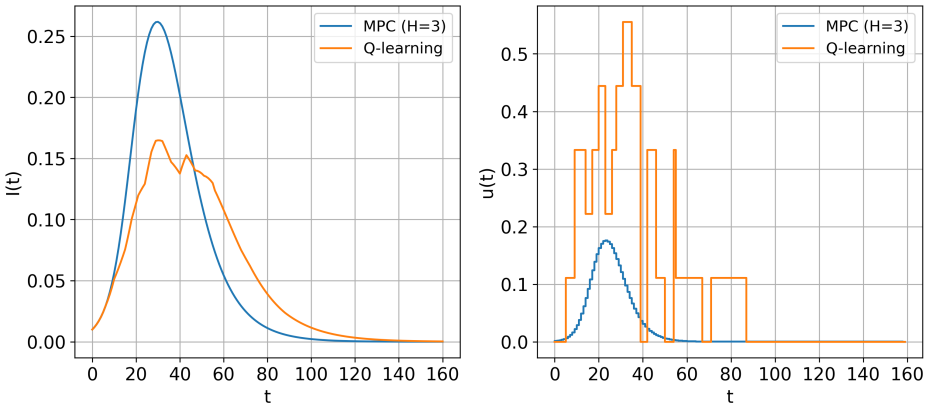
Figure 4b presents the comparison with a moderate MPC prediction horizon ($H = 3$). Here, the infection trajectories generated by RL and MPC are qualitatively similar, and both controllers produce well-behaved closed-loop dynamics. The control profiles differ in structure, reflecting the contrast between model-based optimization in MPC and sample-based decision making in Q-learning.

Figure 4c corresponds to the longer-horizon MPC controller ($H = 5$). As expected, the longer prediction horizon leads to smoother control actions and more anticipatory behavior in the MPC policy. The RL controller continues to exhibit a discrete control pattern while maintaining stable closed-loop dynamics.

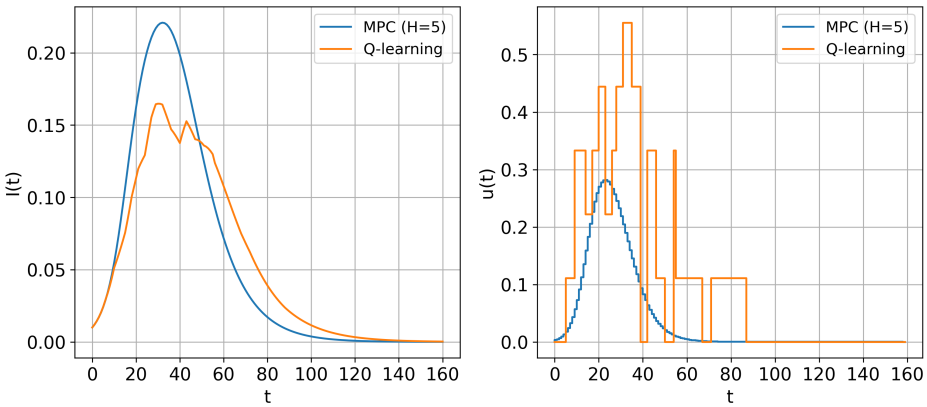
Overall, the figure demonstrates that the learned Q-learning policy yields stable and well-posed closed-loop behavior across different operating regimes. The qualitative similarities with MPC for short and moderate prediction horizons support using MPC as a meaningful reference for illustrating the learned controller's behavior, rather than as a basis for performance ranking.



(a) Q-learning and MPC trajectories ($H = 1$).



(b) Q-learning and MPC trajectories ($H = 3$).



(c) Q-learning and MPC trajectories ($H = 5$).

Figure 4. Comparison of closed-loop state and control trajectories generated by the learned Q-learning policy and MPC controllers with prediction horizons (a) $H = 1$, (b) $H = 3$, and (c) $H = 5$.

4.3. Linear quadratic regulator benchmark

To demonstrate that the proposed recurrence-exploration interpretation is not restricted to the epidemiological setting, we include an additional benchmark based on a discrete-time Linear Quadratic Regulator (LQR) problem. The LQR system is a classical deterministic control benchmark with a known optimal feedback structure and is widely used in control theory.[44–46] This example provides a substantially different setting from the nonlinear SIR model while preserving the finite-state and finite-action assumptions required by the tabular Q-learning analysis.

We consider the discrete-time double-integrator system

$$x_{k+1} = Ax_k + Bu_k, \tag{17}$$

where

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \Delta t \end{bmatrix}, \tag{18}$$

with sampling time $\Delta t = 0.1$. The state vector is defined as

$$x_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \tag{19}$$

where $x_{1,k}$ represents the position and $x_{2,k}$ represents the velocity at time step k. The control input u_k corresponds to the acceleration applied to the system. This model is widely used as a benchmark in optimal control because it captures the essential dynamics of many mechanical systems while remaining sufficiently simple to allow analytical solutions.

The objective is to minimize the infinite-horizon quadratic cost

$$J = \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right), \tag{20}$$

where $Q \succeq 0$ penalizes deviations from the origin and $R \succ 0$ penalizes control effort.

In the experiments, we use

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = [0.1]. \tag{21}$$

The larger weight on the position state reflects the priority of driving the system to the desired equilibrium, while the smaller weight on the velocity state and control input allows smooth convergence without excessive actuation.

The optimal LQR control law is given by

$$u_k = -K x_k, \tag{22}$$

where the feedback gain K is obtained by solving the discrete-time algebraic Riccati equation.

The same quadratic stage cost in **(20)** is used in the cost-based tabular Q-learning update; equivalently, the reward formulation is obtained by setting $r(x, u) = -\ell(x, u)$. The continuous state space is discretized into a finite grid, and the control input is selected from a finite set of admissible actions. Training is performed using a GLIE ϵ-greedy policy with decaying exploration. For comparison with MPC, we use the representative prediction horizon $H = 5$, which, among the tested horizons in the preceding SIR experiments, produced the closest qualitative correspondence with the learned Q-learning policy.

Figure 5 illustrates the evolution of the finite-horizon cost obtained by the Q-learning algorithm and compares its final performance with two model-based benchmarks, LQR and MPC ($H = 5$). At the beginning of training, the Q-learning policy exhibits a high cost, indicating that the agent is still exploring and has not yet identified effective control actions. The large shaded region around the learning curve reflects substantial variability across independent runs during this exploratory phase. As the number of temporal-difference (TD) updates increases, the cost decreases rapidly, and the variability is progressively reduced, demonstrating that the learned policy becomes both more effective and more consistent. After approximately 2×10^5 TD updates, the learning curve stabilizes and converges to a cost level that is nearly indistinguishable from those achieved by LQR and MPC. This result indicates that the model-free Q-learning algorithm is able to recover a control policy whose performance closely matches that of established optimal control methods, despite relying solely on sampled interactions with the system and without requiring explicit knowledge of the state-space model.

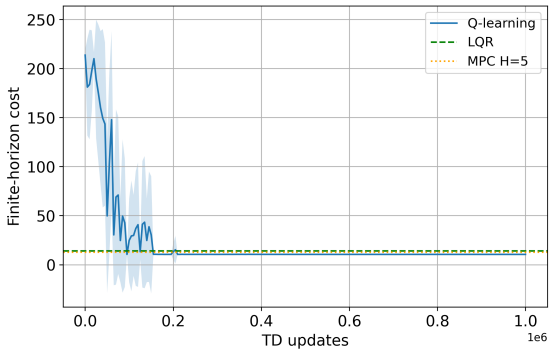
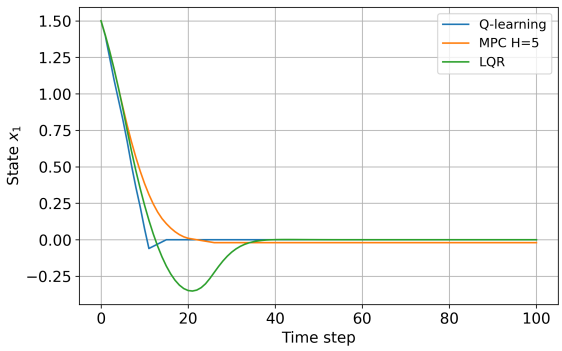
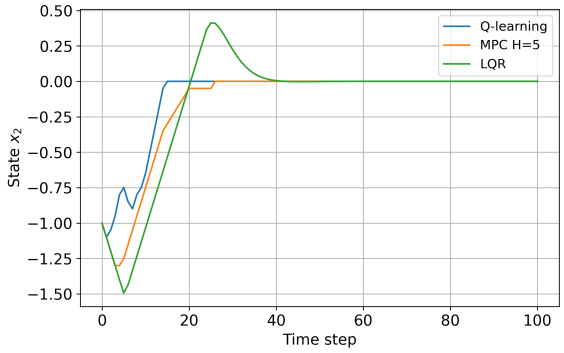


Figure 5. Learning curve of Q-learning compared with the LQR and MPC benchmark costs.

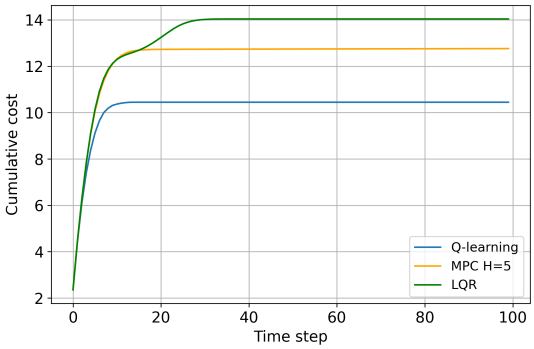
Figure 6 compares the closed-loop trajectories obtained with Q-learning, LQR, and MPC. All three controllers drive both the position and velocity states toward zero. The LQR solution provides the exact optimal linear feedback, MPC produces nearly identical trajectories with a sufficiently long prediction horizon, and Q-learning yields a similar response after convergence. **Figure 5** shows the learning curve of Q-learning. The cumulative cost decreases steadily as the number of episodes increases, indicating that the learned policy converges toward a stabilizing control law.



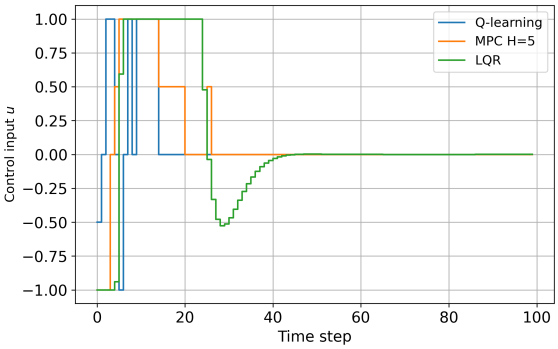
(a) Closed-loop trajectory of the position state x_1.



(b) Closed-loop trajectory of the velocity state x_2.



(c) Cumulative cost over time.



(d) Closed-loop control input generated by Q-learning, MPC, and LQR.

Figure 6. Comparison of Q-learning, Model Predictive Control (MPC), and Linear Quadratic Regulator (LQR) on the double-integrator benchmark: (a) position, (b) velocity, (c) cumulative cost, and (d) control input.

The LQR benchmark provides an analytical reference for the linear-quadratic setting. Because the model is linear and the cost is quadratic, the LQR controller is globally optimal within this problem class. MPC approaches the same solution when the prediction horizon is sufficiently large and no active constraints alter the control law. The results show that the learned Q-learning policy reproduces the qualitative behavior of both LQR and MPC. The learning curve demonstrates progressive improvement during training, while the closed-loop trajectories confirm that the resulting policy stabilizes the system and achieves a performance close to that of classical optimal controllers.

5. Discussion

The theoretical and numerical results together indicate that the convergence-relevant behavior of Q-learning in deterministic feedback systems cannot be understood from the exploration schedule alone. GLIE exploration ensures persistent action sampling, but this mechanism becomes effective only on regions of the discretized state space that are revisited sufficiently often by the induced closed-loop dynamics. Accordingly, recurrence of the state process and persistent exploration play complementary roles in realizing the classical infinite-visitation condition.

The numerical experiments support this interpretation. The GLIE-compliant schedule maintains continued refinement of the Q-table and leads to stable closed-loop behavior, whereas the faster-decaying non-GLIE schedule stabilizes earlier and may limit further exploration. The comparison with MPC further indicates that the learned policy produces interpretable control trajectories and achieves performance comparable to short- and moderate-horizon model-based baselines.

These findings suggest that, in deterministic control applications, exploration design should be considered jointly with the recurrence structure induced by the dynamics and the admissible controls. The present analysis has several important limitations. First, the theoretical results are restricted to tabular Q-learning with finite state and action spaces, where the action-value function is represented explicitly as a lookup table.

Second, the framework does not address function approximation, deep reinforcement learning, or continuous state representations, which introduce additional approximation and stability challenges beyond the classical Watkins-Dayan setting. Third, the GLIE ϵ-greedy exploration mechanism considered here does not incorporate explicit safety constraints. In practical control applications, exploratory actions may violate operational limits or produce undesirable transient responses. Extending the present recurrence-based characterization to safe and constrained reinforcement learning settings remains an important direction for future research.

6. Conclusion

This paper has studied tabular Q-learning from the perspective of feedback control in deterministic discrete-time systems. By treating the learning algorithm and the controlled system as a single closed-loop dynamical process, the analysis provides a dynamical-systems interpretation of the infinite-visitation condition underlying the classical Watkins-Dayan convergence theory. The principal theoretical contribution is the characterization that infinite visitation emerges from the interaction between recurrence of the induced state dynamics and persistent exploration generated by GLIE-compliant ϵ-greedy policies. Under a mild regularity condition on the recurrence structure, this interaction guarantees infinite visitation of all state-action pairs in the recurrent region.

From a practical perspective, the results clarify the complementary roles of system dynamics and exploration policy when Q-learning is deployed as a feedback controller. The numerical experiments support this interpretation and show that, under standard Robbins-Monro and GLIE conditions, the learned policy produces stable closed-loop behavior with performance qualitatively consistent with established model-based controllers, including MPC and LQR benchmarks.

The present study is restricted to tabular Q-learning with finite state and action spaces and does not address function approximation, deep reinforcement learning, continuous state representations, or explicit safety constraints during exploration. Extending the present recurrence-based characterization to broader classes of learning-based controllers, including safe and constrained reinforcement learning in stochastic and partially observed systems, remains an important direction for future research.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the Indonesian Education Scholarship (BPI), administered by the Center for Higher Education Funding and Assessment (PPAPT), Ministry of Education, Culture, Research, and Technology, and the Indonesian Endowment Fund for Education (LPDP) (see the Funding section for details).

Funding

This research was supported by the Indonesian Education Scholarship (BPI) through the Center for Higher Education Funding and Assessment (PPAPT), Ministry of Education, Culture, Research, and Technology, and the Indonesian Endowment Fund for Education (LPDP), under decree number 00827/BPPT/BPI.06/9/2023.

Conflict of interest

The authors declare that they have no conflict of interest.

Author contributions

Conceptualization: Nanang Susyanto, Utti M. Rifanti
Methodology: Nanang Susyanto, Utti M. Rifanti
Software: Utti M. Rifanti
Data curation: Utti M. Rifanti
Formal analysis: Utti M. Rifanti, Nanang Susyanto
Funding acquisition: Utti M. Rifanti
Investigation: Utti M. Rifanti
Project administration: Utti M. Rifanti
Resources: Utti M. Rifanti
Supervision: Lina Aryati, Nanang Susyanto, Hadi Susanto
Validation: Lina Aryati, Nanang Susyanto, Hadi Susanto
Visualization: Utti M. Rifanti
Writing – original draft: Utti M. Rifanti
Writing – review & editing: Lina Aryati, Nanang Susyanto, Hadi Susanto

Availability of data

Not applicable.

AI tools statement

During the preparation of this work, the authors used Grammarly and ChatGPT in order to improve language and readability. After using these tools/services, the authors reviewed and edited

the content as needed and take full responsibility for the content of the publication.

References

1. Huang M, Liu C, He X, Ma L, Lu Z, Su, H. Reinforcement Learning-Based Control for Nonlinear Discrete-Time Systems. *Neurocomputing*. 2020;402:50–65. https://doi.org/10.1016/j.neucom.2020.03.061
2. Kiumarsi B, Lewis FL, Modares H, Karimpour A, Naghibi-Sistani MB. Reinforcement Q-learning for Optimal Tracking Control. *Automatica*. 2014;50(4):1167–1175. https://doi.org/10.1016/j.automatica.2014.02.015
3. Mukhopadhyay R, Sutradhar A, Chattopadhyay P. A Novel Investigation on The Effects of State and Reward Structure in Designing Deep Reinforcement Learning-Based Controller for Nonlinear Dynamical Systems. *Int J Dyn Control*. 2024;12(8):3017–3032. https://doi.org/10.1007/s40435-024-01407-6
4. Yadav KP, Narayan J, Kushwaha P. Learning to Balance: Reinforcement Learning Control for Single-Leg Balance of An Underactuated Biped Robot. *Int J Dyn Control*. 2025;13(7):268. https://doi.org/10.1007/s40435-025-01782-8
5. Semenov SS, Tsurkov VI. Reinforcement Learning for Optimal Control Problems. *J Comput Syst Sci Int*. 2023;62(3):508–521. https://doi.org/10.31857/S0002338823030125
6. Chen S, Zheng J. A Q-learning Grey Wolf Optimizer for A Distributed Hybrid Flowshop Rescheduling Problem with Urgent Job Insertion. *J Appl Math Comput*. 2025;71(3):3645–3670. https://doi.org/10.1007/s12190-024-02364-1
7. Kankashvar M, Rafiee S, Bolandi H. Fault-Tolerant Q-Learning for Discrete-Time Linear Systems with Actuator and Sensor Faults Using Input-Output Measured Data. *Frankl Open*. 2025;11:100259. https://doi.org/10.1016/j.fraope.2025.100259
8. Rifanti UM, Aryati L, Susyanto N, Susanto H. A Reinforcement Learning Based Decision-Support System for Mitigate Strategies During COVID-19: A Systematic Review. *Jambura J Biomath*. 2025;6(1):60–70. https://doi.org/10.37905/jjbm.v6i1.30513
9. Watkins CJCH, Dayan P. Q-learning. *Mach Learn*. 1992;8:279–292. https://doi.org/10.1007/BF00992698
10. Tsitsiklis JN. Asynchronous Stochastic Approximation and Q-Learning. *Mach Learn*. 1994;16(3):185–202. https://doi.org/10.1023/A:1022689125041
11. Bertsekas DP. *Reinforcement Learning and Optimal Control*. Athena Scientific; 2019. Available from: https://web.mit.edu/dimitrib/www/RL_OC_Short_View.pdf [Last accessed on June 4, 2026].
12. Hariharan N, Anand GP. A Brief Study of Deep Reinforcement Learning with Epsilon-Greedy Exploration. *Int J Comput Digit Syst*. 2022;11(1):541–551. https://doi.org/10.12785/ijcds/110144
13. Aghanim A, Chekenbah H, Oulhaj O, Lasri, R. Q-learning Empowered Cavity Filter Tuning with Epsilon Decay Strategy. *Prog Electromagn Res C*. 2024;140:31–40. https://doi.org/10.2528/PIERC23111903
14. Tokic M, Palm G. Value-Difference Based Exploration: Adaptive Control between Epsilon-Greedy and Softmax. In: *KI 2011: Advances in Artificial Intelligence*. Springer; 2011:335–346. https://doi.org/10.1007/978-3-642-24455-1_33
15. Mignon A, Rocha RLA. An Adaptive Implementation of Epsilon-Greedy. *Procedia Comput Sci*. 2017;109:1146–1151. https://doi.org/10.1016/j.procs.2017.05.431
16. Kumar A, Singh D. Adaptive Epsilon-Greedy Reinforcement Learning for IoT Security. *Discov Internet Things*. 2024;4(1):27. https://doi.org/10.1007/s43926-024-00080-7
17. Aslan E, Arserim MA, Uçar A. Development of Push-Recovery Control System for Humanoid Robots Using Deep Reinforcement Learning. *Ain Shams Eng J*. 2023;14(10):102167. https://doi.org/10.1016/j.asej.2023.102167
18. Ben-Akka M, Tanougast C, Diou C. Novel Design of Reward and Epsilon-Greedy Decay Strategy Tailored for Q-Learning in Optimizing Local Mobile Robot Path Planning. *Knowl-Based Syst*. 2025;324:113836. https://doi.org/10.1016/j.knosys.2025.113836
19. Du D, Han S, Qi N, Ammar HB, Wang J, Pan W. Reinforcement Learning for Safe Robot Control Using Control Lyapunov Barrier Functions. In: Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA). 2023:9442–9448. https://doi.org/10.1109/ICRA48891.2023.10160991
20. Zhao L, Gatsis K, Papachristodoulou A. Stable and Safe Reinforcement Learning via A Barrier-Lyapunov Actor-Critic Approach. In: Proceedings of the 2023 62nd IEEE Conference on Decision and Control (CDC). 2023:1320–1325. https://doi.org/10.1109/CDC49753.2023.10383742

21. Lewis FL, Vrabie D, Vamvoudakis KG. Reinforcement Learning and Feedback Control. *IEEE Control Syst Mag*. 2012;32(6):76–105. https://doi.org/10.1109/MCS.2012.2214134

22. Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press; 2018. Available from: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf [Last accessed on June 4, 2026].

23. Liu S, Pu T, Zeng L, Wang Y, Cheng H, Liu Z. Reinforcement Learning-Based Network Dismantling by Targeting Maximum-Degree Nodes in The Giant Connected Component. *Mathematics*. 2024;12(17):2766. https://doi.org/10.3390/math12172766

24. Vamvoudakis KG, Wan Y, Lewis FL, & Cansever D, eds. *Handbook of Reinforcement Learning and Control*. Springer; 2021. https://doi.org/10.1007/978-3-030-60990-0

25. Ladosz P, Weng L, Kim M, Oh H. Exploration in Deep Reinforcement Learning: A Survey. *Inf Fusion*. 2022;85:1–22. https://doi.org/10.1016/j.inffus.2022.03.003

26. Wang D, Wei W, Li L, Wang X, Liang J. Rethinking Exploration-Exploitation Trade-Off. *Neural Netw*. 2025;187:107342. https://doi.org/10.1016/j.neunet.2025.107342

27. Zhang Y, Lyu Y, Zhan G, Zou W, Li SE. Boosting Exploration in Reinforcement Learning for Sparse Reward Tasks. In: Proceedings of the 2025 American Control Conference (ACC). 2025:3492–3499. https://doi.org/10.23919/ACC63710.2025.11107911

28. Sledge IJ, Principe JC. Balancing Exploration and Exploitation in Reinforcement Learning Using A Value of Information Criterion. In: Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017:2816–2820. https://doi.org/10.1109/ICASSP.2017.7952670

29. Garg P, Silvas E, Willems F. Safe and Time-Efficient Exploration in Reinforcement Learning-Based Control of A Vehicle Thermal Systems. *Control Eng Pract*. 2025;164:106458. https://doi.org/10.1016/j.conengprac.2025.106458

30. Ma Z, Liu Z. An Improved Q-learning Algorithm with Particle Swarm Optimization for Path Planning. In: Proceedings of the 2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC). 2024:1662–1667. https://doi.org/10.1109/ICFTIC64248.2024.10913234

31. Rokhlin DB. Robbins-Monro Conditions for Persistent Exploration Learning Strategies. In: *Springer Proceedings in Mathematics & Statistics*. Springer; 2019:237–247. https://doi.org/10.1007/978-3-030-26748-3_14

32. Littman ML. Value-Function Reinforcement Learning in Markov Games. *Cogn Syst Res*. 2001;2(1):55–66. https://doi.org/10.1016/S1389-0417(01)00015-8

33. Szepesvári C, Littman ML. A Unified Analysis of Value-Function-Based Reinforcement-Learning Algorithms. *Neural Comput*. 1999;11(8):2017–2060. https://doi.org/10.1162/089976699300016070

34. Singh S, Jaakkola T, Littman ML, Szepesvári C. Convergence Results for Single-Step Reinforcement Learning Algorithms. *Mach Learn*. 2000;38(3):287–308. https://doi.org/10.1023/A:1007678930559

35. Beresnevich V, Velani S. The Divergence Borel-Cantelli Lemma Revisited. *J Math Anal Appl*. 2023;519(1):126750. https://doi.org/10.1016/j.jmaa.2022.126750

36. Lemos-Silva M, Vaz S, Torres DFM. Exact Solution for A Discrete-Time SIR Model. *Appl Numer Math*. 2025;207:339–347. https://doi.org/10.1016/j.apnum.2024.09.014

37. Gairat A, Shcherbakov V. Discrete SIR Model on A Homogeneous Tree and Its Continuous Limit. *J Phys A Math Theor*. 2022;55(43):434004. https://doi.org/10.1088/1751-8121/ac9655

38. Rokaya M, Hemdan DI, Alzain MA, Atlam E-S. A Novel Fractional-Order Model with Data-Driven Validation for The Dynamics of Complex Epidemic Spreading in Networks. *Int J Optim Control Theor Appl*. 2026;16(1):111–137. https://doi.org/10.36922/IJOCTA025220107

39. Bertsekas DP. Model Predictive Control and Reinforcement Learning: A Unified Framework Based on Dynamic Programming. *IFAC-PapersOnLine*. 2024;58:363–383. https://doi.org/10.1016/j.ifacol.2024.09.056

40. Morcego B, Yin W, Boersma S, van Henten E, Puig V, Sun C. Reinforcement Learning Versus Model Predictive Control on Greenhouse Climate Control. *Comput Electron Agric*. 2023;215:108372. https://doi.org/10.1016/j.compag.2023.108372

41. Ernst D, Glavic M, Capitanescu F, Wehenkel L. Reinforcement Learning Versus Model Predictive Control: A Comparison on A Power System Problem. *IEEE Trans Syst Man Cybern B Cybern*. 2009;39(2):517–529. https://doi.org/10.1109/TSMCB.2008.2007630

42. Sajjadi SS, Pariz N, Karimpour A, Jajarmi A. An Off-Line NMPC Strategy for Continuous-Time Nonlinear Systems Using An Extended Modal

Series Method. *Nonlinear Dyn.* 2014;78(4):2651–2674. https://doi.org/10.1007/s11071-014-1616-6

43. Camacho EF, ordons C. *Model Predictive Control.* Springer; 2007. https://doi.org/10.1007/978-0-85729-398-5

44. Sassano M. Policy Algebraic Equation for The Discrete-Time Linear Quadratic Regulator Problem. *IEEE Trans Autom Control.* 2025;70(4):2106–2121. https://doi.org/10.1109/TAC.2024.3465566

45. Zhang H, Duan G, Xie L. Linear Quadratic Regulation for Linear Time-Varying Systems with Multiple Input Delays. *Automatica.* 2006;42(9):1465–1476. https://doi.org/10.1016/j.automatica.2006.04.007

46. Zhang H, Li L, Xu J, Fu M. Linear Quadratic Regulation and Stabilization of Discrete-Time Systems with Delay and Multiplicative Noise. *IEEE Trans Autom Control.* 2015;60(10):2599–2613. https://doi.org/10.1109/TAC.2015.2411911

Utti M. Rifanti *is a doctoral student in applied mathematics with a focus on reinforcement learning and control of discrete-time dynamical systems. Her research investigates the convergence properties of Q-learning under structured exploration policies and its interpretation within feedback control frameworks. She has worked on theoretical and applied aspects of reinforcement learning, including epidemic modeling, decision-support systems, and optimal control formulations. Her recent work emphasizes the interaction between learning dynamics and system behavior, particularly under deterministic settings and limited exploration.*
https://orcid.org/0000-0002-6622-9823

Lina Aryati *is an Associate Professor of Mathematics at Universitas Gadjah Mada, Indonesia. Her research interests include applied mathematics, differential equations, perturbation theory, and mathematical biology. She has contributed to the analysis of nonlinear dynamical systems and their applications in modeling real-world phenomena, particularly in biological and interdisciplinary contexts. Her work integrates analytical and computational approaches to study complex systems arising in applied sciences. She is actively involved in research, teaching, and academic collaboration in the field of applied mathematics.*
https://orcid.org/0009-0004-2783-7422

Nanang Susyanto *is an Associate Professor of Mathematics at Universitas Gadjah Mada, Indonesia. His research focuses on mathematical modeling, semiparametric methods, and copula-based approaches for analyzing complex data structures. He has contributed to both theoretical developments and applied methodologies in statistical and applied mathematics. His work addresses challenges in modeling dependence structures and data-driven systems across interdisciplinary applications. He is actively involved in academic research and collaboration.*
https://orcid.org/0000-0001-8332-3363

Hadi Susanto *is a Professor of Mathematics at Khalifa University, United Arab Emirates. His research interests include nonlinear waves, solitons, pattern formation, and applied mathematics. He has made significant contributions to the analysis of nonlinear dynamical systems and their applications across physics and interdisciplinary sciences. His work combines analytical and computational approaches to study complex phenomena arising in continuous and discrete systems. He is actively involved in international research collaboration and the advancement of applied mathematical sciences.*
https://orcid.org/0000-0003-0425-107X

An International Journal of Optimization and Control: Theories & Applications (https://accscience.com/journal/ijocta)