Online and Robust Intermittent Motion Planning in Dynamic and Changing Environments

Zirui Xu [1], *Student Member, IEEE,* George P. Kontoudis [2], *Member, IEEE,*
Kyriakos G. Vamvoudakis [3], *Senior Member, IEEE*

***Abstract*—In this paper, we propose RRT-Q_∞^X, an online and intermittent kinodynamic motion planning framework for dynamic environments with unknown robot dynamics and unknown disturbances. We leverage RRT^X for global path planning and rapid replanning to produce waypoints as a sequence of boundary value problems (BVPs). For each BVP, we formulate a finite-horizon, continuous-time zero-sum game, where the control input is the minimizer, and the worst-case disturbance is the maximizer. We propose a *robust intermittent Q-learning* controller for waypoint navigation with completely unknown system dynamics, external disturbances, and intermittent control updates. We execute a relaxed persistence of excitation technique to guarantee that the Q-learning controller converges to the optimal controller. We provide rigorous Lyapunov-based proofs to guarantee the closed-loop stability of the equilibrium point. The effectiveness of the proposed RRT-Q_∞^X is illustrated with Monte-Carlo numerical experiments in numerous dynamic and changing environments.**

I. Introduction

Safe motion planning of mobile robots in dynamic environments is a challenging task with applications in autonomous vehicles and human-crowded robotic navigation. One difficulty of safe motion planning in dynamic environments lies in the limited ability of perception and inference, which leads to inaccuracies in multiple aspects of understanding the environment. Examples include inaccurate motion prediction for obstacles or other agents, false representation map of the environment, and imprecise localization of the ego agent. Whenever the environment model is updated, a fast replanning is needed. Moreover, optimal control is desired for autonomous systems, yet it typically requires extensive, cumbersome offline computation. In addition, precomputed optimal control suffers from the uncertain nature of the system dynamics, which compromises its optimality and safety. External disturbances are usually stochastic and unknown *a priori*. In practice, rejecting external disturbances is essential not only for optimality of the control policy, but also for maintaining safety in planning. Furthermore, mobile robots are often equipped with limited on-board computation and communication capabilities—their energy resources are limited and require judicious distribution. In this work, our focus is on providing a safe, online, and model-free kinodynamic motion planning methodology in finite horizon without any offline computation and with intermittent communication.

Sampling-based algorithms, e.g., the Rapidly-exploring Random Tree (RRT) [1], are efficient in high-dimensional motion planning tasks. While RRT is probabilistically complete, RRT* provides solutions that are probabilistically complete as well as asymptotically optimal in static environments by rewiring the search tree [2]. In dynamic environments, the Execution-extended RRT (ERRT) is considered as one of the first algorithms for real-time replanning [3]. While ERRT rebuilds a new tree from the current robot configuration to goal when replanning, the Dynamic RRT (DRRT) [4] only rebuilds part of the tree from the colliding nodes to goal and thus reports higher efficiency in execution time. The authors in [5] present the RRT^X, an asymptotically optimal motion planning algorithm for both static and dynamic environments. Growing from goal to start, RRT^X keeps the main spanning tree while replanning. During replanning, while also performing the rewiring operation as RRT* to guarantee asymptotic optimality, RRT^X improves efficiency by maintaining a limited neighborhood size and having a relaxed rewiring cascade.

For kinodynamic motion planning, optimization-based techniques are used in kinodynamic RRT* [6] where the optimality of paths is ensured for controllable linear systems. Yet, this open-loop approach is vulnerable to disturbances, model inaccuracies, and requires offline computation. Another kinodynamic motion planning technique is the Linear Quadratic Regulator Trees (LQR-Trees) presented in [7]. This approach employs convex optimization tools to compute Lyapunov functions and regions of attraction for feedback motion planning in linearized systems. A similar approach [8] that combines LQR and RRT*, considers time as an additional dimension of the state space. The authors in [9] present RRT-Q*, an online, model-free kinodynamic motion planning framework that computes approximately optimal control policies for motion planning in static environments. The Stable Sparse RRT (SST) and SST* are introduced in [10] for asymptotically near-optimal and optimal sampling-based kinodynamic motion planning, respectively. They propagate the system dynamics forward-in-time using Monte-Carlo-based random control and random propagation time without solving the boundary-value problems (BVP). Similarly to RRT, SST and SST* provide only high-level paths without low-level control laws. A multi-query algorithm for asymptotically optimal kinodynamic mo-

[1]Z. Xu is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA (email: ziruixu@umich.edu).

[2]G. P. Kontoudis is with the Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA (email: kont@umd.edu).

[3]K. G. Vamvoudakis is with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA (email: kyriakos@gatech.edu).

This work was supported, in part by NSF under grant Nos. CAREER CPS-1851588, CPS-2038589, CPS-2227185, SATC-1801611, by NASA ULI under grant number 80NSSC20M0161, and by the United States Army under Contract No. W56HZV-17-C-0095.

tion planning that sample edges instead of configurations is proposed in [11]. A survey on sampling-based motion planning algorithms appears in [12] and a benchmark in [13].

The authors in [14] combine a variant of deep reinforcement learning (RL) with RRT and propose the RL-RRT for kinodynamic motion planning. They first use deep RL to learn an obstacle avoiding policy with the system dynamics, serving as a local planner and controller. Based on the RL policy, they use supervised learning to predict the time to reach a state and guide the growth of the tree. However, this method requires extensive offline computations for policy training and still needs higher accuracy for safe navigation. In [15], a geometric version of kinodynamic motion planning is proposed. The authors combine sampling-based motion planning techniques in static environments and model-based adaptive control, to develop a robust trajectory tracking controller. The authors in [16] construct an optimal motion planning algorithm for static environments by employing RL [17] and artificial harmonic potential fields [18]. A probabilistic roadmap method of kinodynamic motion planning with known trajectories of moving obstacles is presented in [19]. This method samples new nodes in the state-time space and then maps them into the physical state space, using random control inputs of known dynamics. In [20], the authors propose an online kinodynamic motion planning framework which includes a replanning scheme to reduce replanning events in dynamic environments. Also, offline machine learning techniques drastically reduce the number of BVPs and facilitate the online implementation. However, the feasibility of this technique depends solely on the offline training of the reachability sets. The work of funnel libraries [21] employs similar convex optimization tools as in the LQR-Trees, yet incorporates the influence of bounded model uncertainties and disturbances. Online kinodynamic motion planning is addressed in unknown environments after computing offline the computationally expensive funnel libraries. The authors in [22] present a sampling-based kinodynamic motion planning framework that uses supervised learning to train a neural network controller for solving BVPs offline and apply it in real-time. The authors in [23] propose an end-to-end deep-learning-based kinodynamic motion planning, where they use a deep neural network to predict waypoints and use model predictive control to drive the system to the next waypoint. A real-time feedback motion planning and replanning method for nonlinear systems in dynamic environments with a novel graph data structure is proposed in [24]. A two-player zero-sum game (TPZSG) is formulated in [25] for the worst-case disturbance rejection in controller synthesis. Bounded rationality [26]–[28] has been combined with RRT-Q^X [29] to perform multi-agent, human-like motion planning [30].

These motion planning works have reported outstanding results in various applications. Yet, almost all of them are *model-based* methods, requiring accurate knowledge of system dynamics and external disturbances for robust control synthesis. Besides, all motion planning methods are executed with constant update of the controller, leading to significant communication requirements. The constant update of control laws is not always necessary, especially when optimality can be guaranteed with an intermittent update.

Approximate dynamic programming [17] serves as a connection between adaptive control [31] and optimal control [32] by employing the principles of RL. In [33], a partially model-free algorithm is introduced to solve the optimal control problem online via policy iteration. In [34], a Q-learning technique is presented to solve the infinite-horizon optimal control problem with unknown continuous-time linear systems. The authors in [35] developed an intermediate method between policy iteration and value iteration for discrete-time systems.

Intermittent control is able to operate optimally in realistic systems with limited communication between sensors, controllers, and actuators [36]. The systems evaluate a user-defined triggering condition to determine whenever the loop should be closed. The decision is determined by the equilibrium stability of the closed-loop system. The authors in [37] propose a Q-learning approach with event-trigger conditions to solve an infinite-horizon game-theoretic problem.

The authors in [38] employ experience replay mechanisms to reduce the non-stationarities and instability of the reinforcement learning algorithm. The latter captures previous data to alleviate the update process. A similar constraint appears in adaptive control [31] that the probing signal is required to be persistently exciting for convergence of weight parameters to the ideal values. That condition is conservative and is hard to accomplish in real-world applications. In [39], [40] the authors utilize previous experiences concurrently with current data to relax the condition of persistence of excitation, which makes it easier to implement in practice. Experience replay is more recently leveraged to relax the persistence of excitation requirement in Hamiltonian-driven methods [41].

Preliminary results of this work are presented in [29], [42]. In this paper, we extend the mathematical formulation to systems with external disturbances as a two-player zero-sum game in Sections II and III and propose a robust intermittent Q-learning method in Section IV to solve it. We also provide an in-depth discussion of the algorithm along with a computational complexity analysis in Section V. In Section VI we illustrate the efficiency of the methodology in more challenging dynamic environments and present the variation of the maximum kinodynamic distances in different scenarios. Finally, in the Appendix we provide a complete Lyapunov-based stability proof by considering external disturbances.

Contribution. The contribution of this paper is threefold. First, we develop a *robust intermittent Q-learning* method for solving a finite-horizon game-theoretic control problem for continuous-time linear systems, without any knowledge of the system dynamics (Sections III to V). Compared to existing control techniques, our method can achieve robust intermittent control with unknown dynamics and a relaxed persistence of excitation for the finite-horizon optimal control problem. Then, we provide rigorous stability, robustness, and convergence guarantees (Theorems 2 and 3). Finally, we propose a decoupled, real-time motion planning framework, RRT-Q^X_∞, which combines robust intermittent Q-learning and a sampling-based motion planner. The proposed RRT-Q^X_∞ can achieve safe, online, robust kinodynamic motion planning in unpredictable dynamic environments without offline computation/training (Table I).

Structure. In Section II we formulate the problem. Section III provides an intermittent optimal solution to the TPZSG. The robust intermittent Q-learning method is shown in Section IV. In Section V we discuss the algorithmic details of the proposed technique. Section VI illustrates the efficacy of the proposed motion planning method in dynamic environments, and Section VII concludes the paper.

Notation and Nomenclature. See the following.

$\mathbb{R}$, $\mathbb{N}$	real numbers, natural numbers
$\mathbb{R}^+$, $\mathbb{R}^{n\times m}$	positive real numbers, $n \times m$ real matrices
$\underline{\lambda}(A)$, $\overline{\lambda}(A)$	minimum, maximum eigenvalues of matrix A
$\|v\|$	Euclidean norm of vector v
$\mathrm{vech}(A)$	half-vectorization of matrix A
$\mathrm{vec}(A)$	vectorization of matrix A
$\mathrm{mat}(A)$	matrization of matrix A
$A[a\colon b, c\colon d]$	submatrix of row a to b and column c to d of matrix A
$\kappa(\cdot)$	a class $\mathcal{K}$ function
$\otimes$	Kronecker product of two matrices
$\oplus$	Minkowski sum of two sets
$\mathbf{0}_p$	vector of length p with all elements 0
$\nabla_x(y)$	gradient of y with respect to x
M, R	penalizing matrices
γ	disturbance rejection constant
ρ	admissible window for completing a BVP problem
L, L_1	Lipschitz constraints
D_{rob}, $D_{\text{rob}}^{\text{kin}}$	kinodynamic distance, maximum kinodynamic distance
W_{c}, W_{a}, W_{d}	critic, control actor, disturbance actor approx. weights
α_{c}, α_{a}, α_{d}	gradient descent learning rates

II. Problem Formulation

In this section, we first present the formulation of a model-free two-player zero-sum game (TPZSG). Then, we incorporate the solution of the game into sampling-based path planning for dynamic environments.

A. System Description

Consider the following continuous time linear time-invariant system describing the motion of the system,

$$\dot{x}(t) = Ax(t) + Bu(t) + Fd(t), \;\; x(0) = x_0, \;\; t \geq 0,$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ is a measurable state vector, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, $d(t) \in \mathcal{D} \subseteq \mathbb{R}^q$ is the disturbance input, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $F \in \mathbb{R}^{n\times q}$ are the unknown plant, input, and disturbance matrices, respectively. We seek to use an intermittently updated control input to drive the system from an initial state x_0 to a desired state x_{r} in a finite horizon $T \in \mathbb{R}^+$. Thus, we define a new state of differences $\bar{x}(t) \coloneqq x(t) - x_{\text{r}}$, to obtain the new system,

$$\dot{\bar{x}}(t) = A\bar{x}(t) + Bu(t) + Fd(t), \;\; \bar{x}(0) = x_0 - x_{\text{r}}, \; t \geq 0. \quad (1)$$

To conserve computational resources and reduce communication efforts, we shall employ a sampled version of the state for intermittent learning, which is defined as,

$$\hat{\bar{x}}(t) \coloneqq \begin{cases} \bar{x}(r_j), & t \in (r_j, r_{j+1}] \\ \bar{x}(t), & t = r_j \end{cases}$$

where $\bar{x}(r_j)$ is the state of flow dynamics, $\bar{x}(t)$ is the state of jump dynamics, and r_j is a strictly monotonically increasing sequence of samples $\{r_j\}_{j=1}^{\infty}$. The trigger of the control signal update at $t = r_j$ is based on an error gap described as,

$$e(t) \coloneqq \hat{\bar{x}}(t) - \bar{x}(t). \quad (2)$$

We formulate the problem as a non-cooperative zero-sum game in a strategic form $\mathcal{G} = (\mathcal{N}, \{\mathcal{S}_i\}_{i\in\mathcal{N}}, J)$, where $\mathcal{N} = \{P_1, P_2\}$ is the set of players, P_1 is the control player selecting u_{d} that desires to minimize the cost and P_2 is the disturbance player selecting d that aims to maximize the cost, $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 = \mathcal{U} \times \mathcal{D}$ is the set of strategies, and $J : \mathcal{S} \to \mathbb{R}$ is the finite-horizon common cost function, which is formulated as,

$$J(\bar{x}; u_{\text{d}}, d; t_0, T) \quad (3)$$
$$= \phi(T) + \frac{1}{2}\int_{t_0}^{T} (\bar{x}^\intercal M\bar{x} + u_{\text{d}}^\intercal R u_{\text{d}} - \gamma^2\|d\|^2)\,\mathrm{d}\tau,$$

where $u_{\text{d}} \coloneqq \kappa(\hat{\bar{x}}(t))$ is the intermittent control input, $\phi(T) \coloneqq \frac{1}{2}\bar{x}^\intercal(T)P(T)\bar{x}(T)$ is the final cost with $x(T)$ the free final state and $P(T) \in \mathbb{R}^{n\times n} \succ 0$ a user-defined, symmetric, and positive-definite final Riccati matrix, $M \in \mathbb{R}^{n\times n} \succeq 0$ and $R \in \mathbb{R}^{m\times m} \succ 0$ are user-defined matrices penalizing the state and the control input respectively, and $\gamma \in \mathbb{R}^+$ is a disturbance rejection constant, $\gamma \geq \gamma^\star$, where $\gamma^\star$ is the smallest value that can stabilize system (1) [43]. In the finite-horizon boundary-value problem (BVP) we seek to drive the system from an initial state to a desired state in a finite time.

We seek to obtain a saddle-point equilibrium $(u_{\text{d}}^\star, d^\star)$ such that, $J(\bar{x}; u_{\text{d}}^\star, d; t_0, T) \leq J(\bar{x}; u_{\text{d}}^\star, d^\star; t_0, T) \leq J(\bar{x}; u_{\text{d}}, d^\star; t_0, T)$, for all $\bar{x}, u_{\text{d}}, d$, provided by the optimization $J(\bar{x}_0; u_{\text{d}}^\star, d^\star; t_0, T) = \min_{u_{\text{d}}}\max_d\{J(\bar{x}; u_{\text{d}}, d; t_0, T)\}$ subject to system (1). To solve the game given the cost function (3), we define a value function,

$$V^\star(\bar{x}; t_0, T) \coloneqq \quad (4)$$
$$\min_{u_{\text{c}}}\max_d \left\{ \phi(T) + \frac{1}{2}\int_{t_0}^{T} (\bar{x}^\intercal M\bar{x} + u_{\text{c}}^\intercal R u_{\text{c}} - \gamma^2\|d\|^2)\,\mathrm{d}\tau \right\}.$$

Assumption 1: The unknown pair (A, B) is controllable and the unknown pair ($\sqrt{M}$, A) is detectable. □

B. Motion Planning Foundations

We now provide basic foundations for the motion planning case. We consider the known obstacle closed space as, $\mathcal{X}_{\text{obs}} \coloneqq \bigcup_{l=1}^{N_o} \mathcal{X}_{\text{obs},l} \subset \mathcal{X}$, where $N_o \in \mathbb{N}$ is the total number of obstacles. Thus, the free space is the open space $\mathcal{X}_{\text{free}} = (\mathcal{X}_{\text{obs}})^{\complement} = \mathcal{X}\backslash\mathcal{X}_{\text{obs}}$. In dynamic environments, the obstacle space $\mathcal{X}_{\text{obs}}$ and the free space $\mathcal{X}_{\text{free}}$ are functions of time. Denote the unpredictable change of the obstacle space as $\Delta\mathcal{X}_{\text{obs}}$, where $\Delta\mathcal{X}_{\text{obs}}$ is unknown. For path planning, we employ RRT$^{\text{X}}$ that provides an optimal sub-tree which contains the planned path $\pi(x_{0,k}, x_{\text{r},k}; t) \in \mathbb{R}^{2(K\times n)}$, with $k = 1, \ldots, K$, $K \in \mathbb{N}$ the index of BVPs. Each BVP is described by its initial and desired states as a tuple $(x_{0,k}, x_{\text{r},k})$. Since the obstacle space $\mathcal{X}_{\text{obs}}$ evolves in time, π is also a function of time, and thus K also changes accordingly. RRT$^{\text{X}}$ constructs a graph $G = (V, E)$, with V the set of nodes and E the set of edges. As a slight abuse of notation, we shall refer to nodes $v \in V$ as states $x \in \mathcal{X}$.

Next, let us provide connections of the game-theoretic formulation to the motion planning problem. For each BVP provided by RRT$^{\mathrm{X}}$, we aim to drive the unknown system to the desired state considering the worst-case disturbance $d^\star$ that seeks to maximize the value function (4). For the k-th BVP, define the *initial distance* as the distance from the initial state $x_{0,k}$ to the desired state $x_{\mathrm{r},k}$ as,

$$D_0(\bar{x}_{0,k}) \coloneqq \|x_{0,k} - x_{\mathrm{r},k}\| = \|\bar{x}_{0,k}\|, \ \ \forall \bar{x}_0 \in \mathbb{R}^n, \tag{5}$$

and the agent's *relative distance* to $x_{\mathrm{r},k}$ as,

$$D(\bar{x}) \coloneqq \|x - x_{\mathrm{r},k}\| = \|\bar{x}\|, \ \ \forall \bar{x} \in \mathbb{R}^n. \tag{6}$$

Since the problem has a *free-final state*, $x(T)$ will converge to a close neighborhood around $x_{\mathrm{r},k}$ [32], [44]. Therefore, to reduce the navigation time, we consider that $x_{\mathrm{r},k}$ is reached when the state vector enters its close neighborhood within a finite horizon. That is to say, when $D(\bar{x}) \leq \rho D_0(\bar{x}_{0,k})$, where ρ is the user-defined *admissible window*, the agent has reached the desired state $x_{\mathrm{r},k}$. Subsequently, the system will continue on the $(k+1)$-th problem.

Moreover, since the system dynamics are unknown, when RRT$^{\mathrm{X}}$ generates the collision-free path π, it can only use straight-line paths as edges in E. However, the agent's actual trajectory is curved due to the dynamical constraints (1) as well as the optimal requirement (3). Thus, the actual trajectory is deviated from π, and collisions may occur when π is very close to the obstacles. To address this issue, we follow an obstacle augmentation strategy similarly to [9]. In particular, the algorithm constantly computes the *kinodynamic distance*,

$$D_{\mathrm{rob}}(\bar{x}) \coloneqq \frac{|\bar{x}_{0,k} \times \bar{x}|}{D_{0,k}}, \tag{7}$$

to capture the deviation of the agent's position from the corresponding straight path determined by $(x_{0,k},\, x_{\mathrm{r},k})$. Then, an augmented obstacle space $\mathcal{X}_{\mathrm{obs}}^{\mathrm{aug}}$ is obtained based on the *maximum kinodynamic distance* , $D_{\mathrm{rob}}^{\mathrm{kin}}$, via the Minkowski sum,

$$\mathcal{X}_{\mathrm{obs}}^{\mathrm{aug}} \coloneqq \mathcal{X}_{\mathrm{obs}} \oplus \mathcal{X}_{\mathrm{kin}}, \tag{8}$$

where $\mathcal{X}_{\mathrm{kin}}$ is the space of a compact set bounded by a circle with center at the origin and radius $D_{\mathrm{rob}}^{\mathrm{kin}}$. Every time $D_{\mathrm{rob}}^{\mathrm{kin}}$ is updated, $\mathcal{X}_{\mathrm{obs}}^{\mathrm{aug}}$ will be updated by (8), and then RRT$^{\mathrm{X}}$ will replan a path that is further from the actual obstacles with the newly-invalid nodes and their descendants pruned.

III. Intermittent Two-Player Zero-Sum Game

The Hamilton-Jacobi-Isaacs (HJI) equation [45] for the finite-horizon problem with respect to (1) and (4) yields,

$$\mathcal{H}(\bar{x}; u_{\mathrm{c}}, d; \frac{\partial V^\star}{\partial t}, \frac{\partial V^\star}{\partial \bar{x}}) = \frac{1}{2}(\bar{x}^\intercal M \bar{x} + u_{\mathrm{c}}^\intercal R u_{\mathrm{c}} - \gamma^2 \|d\|^2) + \frac{\partial V^{\star\,\intercal}}{\partial \bar{x}} (A\bar{x} + Bu_{\mathrm{c}} + Fd) + \frac{\partial V^\star}{\partial t}. \tag{9}$$

An optimal value function for (1) can be defined as,

$$V^\star(\bar{x}; t) = \frac{1}{2}\bar{x}^\intercal P(t) \bar{x}, \ \ \forall \bar{x}, t \geq 0, \tag{10}$$

where $P(t) \in \mathbb{R}^{n\times n} \succ 0$ is the solution to the following game differential Riccati equation,

$$-\dot{P}(t) = P(t)A + A^\intercal P(t) + M + \gamma^{-2} P(t) F F^\intercal P(t) - P(t) B R^{-1} B^\intercal P(t), \ t \geq 0. \tag{11}$$

Theorem 1: Suppose that there exists a positive definite $P(t), \ t \geq 0$ satisfying the game differential Riccati equation (11) with a terminal condition $P(T) \succ 0$. Then, the time-triggered state feedback optimal control takes the form of,

$$u_{\mathrm{c}}^\star(\bar{x}; t) = -R^{-1} B^\intercal P(t) \bar{x}, \ \forall \bar{x}, t, \tag{12}$$

and the worst-case disturbance yields,

$$d^\star(\bar{x}; t) = \gamma^{-2} F^\intercal P(t) \bar{x}, \ \forall \bar{x}, t, \tag{13}$$

with saddle-point equilibrium value $V^\star = \bar{x}_0^\intercal P(0) \bar{x}_0$.

Proof. The proof follows from [46, Corollary 17.1]. ∎

Next, we consider an intermittent feedback controller which leads to significant reduction in communication when closing the loop. The optimal intermittent controller results in,

$$u_{\mathrm{d}}^\star(\hat{\bar{x}}; t) \coloneqq -R^{-1} B^\intercal P(t) \hat{\bar{x}}, \ \forall \hat{\bar{x}}, t. \tag{14}$$

The main difference between the *optimal time-triggered control* $u_{\mathrm{c}}^\star$ and the *optimal intermittent control* $u_{\mathrm{d}}^\star$ is that, the former uses $\bar{x}$ while the latter uses $\hat{\bar{x}}$. Therefore, every time the control loop is closed and $u_{\mathrm{d}}^\star$ is updated, $u_{\mathrm{d}}^\star \leftarrow u_{\mathrm{c}}^\star$.

Remark 1: As x approaches the desired reference state x_{r}, then $\bar{x}$ approaches $\mathbf{0}_n$. Thus, the control laws $u_{\mathrm{c}}^\star$, $u_{\mathrm{d}}^\star$, and the disturbance $d^\star$ also approach $\mathbf{0}_m$ per (12), (13), (14), i.e., $u_{\mathrm{c}}^\star(T) \to \mathbf{0}_m$, $u_{\mathrm{d}}^\star(T) \to \mathbf{0}_m$, and $d^\star(T) \to \mathbf{0}_q$. □

Fact 1: Since the system (1) is linear and both controllers (12), (14) are linear mappings of the state, $u_{\mathrm{c}}^\star(\bar{x}; t) : \mathbb{R}^n \to \mathbb{R}^m$, $u_{\mathrm{d}}^\star(\hat{\bar{x}}; t) : \mathbb{R}^n \to \mathbb{R}^m$, the following inequality holds,

$$\|u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star\| = \|R^{-1} B^\intercal P(t)(\bar{x} - \hat{\bar{x}})\| \leq \|R^{-1} B^\intercal P(t)\| \|\bar{x} - \hat{\bar{x}}\| \leq L(t)\|e\|,$$

where $L(t) \mapsto \mathbb{R}^+$ is a strictly positive-definite function [47]. □

Lemma 1: Given the *intermittent Hamiltonian* as,

$$\mathcal{H}\Big(\bar{x}; u_{\mathrm{d}}, d; \frac{\partial V^\star}{\partial t}, \frac{\partial V^\star}{\partial \bar{x}}\Big) = \frac{1}{2}(\bar{x}^\intercal M \bar{x} + u_{\mathrm{d}}^\intercal R u_{\mathrm{d}} - \gamma^2 \|d\|^2) + \frac{\partial V^{\star\,\intercal}}{\partial \bar{x}} (A\bar{x} + Bu_{\mathrm{d}} + Fd) + \frac{\partial V^\star}{\partial t}, \tag{15}$$

the following inequality is satisfied.

$$\Big\|\mathcal{H}\Big(\bar{x}; u_{\mathrm{d}}^\star, d^\star; \frac{\partial V^\star}{\partial t}, \frac{\partial V^\star}{\partial \bar{x}}\Big)\Big\| \leq \frac{\overline{\lambda}(R)}{2} L(t)^2 \|e\|^2. \tag{16}$$

Proof. Substitute the time-triggered Hamiltonian (9) into the intermittent Hamiltonian (15). Considering the optimal control (12) and the value function (10), we obtain,

$$\mathcal{H}\Big(\bar{x}; u_{\mathrm{d}}^\star, d^\star; \frac{\partial V^\star}{\partial t}, \frac{\partial V^\star}{\partial \bar{x}}\Big) = \frac{1}{2}(u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star)^\intercal R (u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star). \tag{17}$$

Then, using Fact 1, inequality (16) holds. ∎

Theorem 2: Let a positive-definite radially unbounded function $V(\bar{x};t) = \frac{1}{2}\bar{x}^\intercal P(t)\bar{x}$ with P satisfying the game differential Riccati equation (11). Then, the squared norm of the error gap follows,

$$\|e\|^2 \leq \frac{(1-\beta^2)\underline{\lambda}(M)\|\bar{x}\|^2 + \underline{\lambda}(R)\|u_{\mathrm{d}}^\star\|^2 - \gamma^2\|d^\star\|^2}{\overline{\lambda}(R)L(t)^2},$$

where $\beta \in [0,1)$ is a user-defined bandwidth parameter, and the equilibrium point of the closed-loop system (1) is asymptotically stable as $T \to \infty$ with the intermittent control (14) for all $t \in (r_j, r_{j+1}]$.

Proof. See Appendix A. ∎

Remark 2: The selection of β depends on the available resources. By selecting β closer to 1, the condition in Theorem 2 is triggered more often, and the intermittent controller performs closer to the time-triggered controller. □

IV. Robust Intermittent Q-learning

In this section, we present a robust intermittent Q-learning approach to solve the model-free TPZSG described in Section III. This approach approximates the time-triggered optimal control policy in real-time.

Let us define the $\mathcal{Q}$-function as,

$$\begin{aligned}\mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t) \coloneqq & V^\star(\bar{x};t) + \mathcal{H}\Big(\bar{x};u_{\mathrm{d}},d;\frac{\partial V^\star}{\partial t},\frac{\partial V^\star}{\partial \bar{x}}\Big) \\ & - \mathcal{H}\Big(\bar{x};u_{\mathrm{c}}^\star,d^\star;\frac{\partial V^\star}{\partial t},\frac{\partial V^\star}{\partial \bar{x}}\Big), \end{aligned} \tag{18}$$

where the time-triggered Hamiltonian associated with the optimal control and the worst-case disturbance vanishes, i.e., $\mathcal{H}(\bar{x};u_{\mathrm{c}}^\star,d^\star;\partial V^\star/\partial t,\partial V^\star/\partial \bar{x}) = 0$, and $\mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t) : \mathbb{R}^{n+m+q} \to \mathbb{R}^+$ is an action-dependent scalar value.

Lemma 2: Given the optimal value function $V^\star(\bar{x};t)$ (4), the game-theoretic problem $\mathcal{Q}^\star(\bar{x};u_{\mathrm{d}}^\star,d^\star;t) \coloneqq \min_{u_{\mathrm{d}}}\max_{d}\mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t)$ has the following optimal value,

$$\mathcal{Q}^\star(\bar{x};u_{\mathrm{d}}^\star,d^\star;t) = V^\star(\bar{x};t) + \frac{1}{2}(u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star)^\intercal R(u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star). \tag{19}$$

Proof. Substitute the intermittent Hamiltonian (17) into the Q-function (18). Since the time-triggered Hamiltonian associated with the optimal control and the worst-case disturbance is $\mathcal{H}(\bar{x};u_{\mathrm{c}}^\star,d^\star;\partial V^\star/\partial t,\partial V^\star/\partial \bar{x}) = 0$, the result follows. ∎

We express the Q-function (18) in a compact quadratic form,

$$\begin{aligned}\mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t) &= \frac{1}{2}U^\intercal \begin{bmatrix} Q_{\mathrm{xx}}(t) & Q_{\mathrm{xu_d}}(t) & Q_{\mathrm{xd}}(t) \\ Q_{\mathrm{u_dx}}(t) & Q_{\mathrm{u_du_d}} & Q_{\mathrm{u_dd}} \\ Q_{\mathrm{dx}}(t) & Q_{\mathrm{du_d}} & Q_{\mathrm{dd}} \end{bmatrix} U \\ &\coloneqq \frac{1}{2}U^\intercal \bar{\mathcal{Q}}(t) U = \frac{1}{2}\mathrm{vech}(\bar{\mathcal{Q}}(t))^\intercal (U \otimes U), \end{aligned} \tag{20}$$

where $U \coloneqq [\bar{x}^\intercal \;\; u_{\mathrm{d}}^\intercal \;\; d^\intercal]^\intercal$ is the augmented state, $Q_{\mathrm{xx}}(t) = \dot{P}(t) + P(t) + M + P(t)A + A^\intercal P(t)$, $Q_{\mathrm{xu_d}}(t) = Q_{\mathrm{u_dx}}^\intercal(t) = P(t)B$, $Q_{\mathrm{xd}}(t) = Q_{\mathrm{dx}}^\intercal(t) = P(t)F$, $Q_{\mathrm{u_du_d}} = R$, $Q_{\mathrm{u_dd}} = Q_{\mathrm{du_d}} = 0$, and $Q_{\mathrm{dd}} = -\gamma^2$. Using the submatrices of $\bar{\mathcal{Q}}(t)$ and the stationarity condition $\partial\mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t)/\partial u_{\mathrm{d}} = 0$, we obtain a model-free formulation of the optimal intermittent control,

$$u_{\mathrm{d}}^\star(\hat{\bar{x}};t) = \arg\min_{u_{\mathrm{d}}} \mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t) = -Q_{\mathrm{u_du_d}}^{-1}Q_{\mathrm{u_dx}}(t)\hat{\bar{x}}. \tag{21}$$

Similarly, by solving $\partial\mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t)/\partial d = 0$ we formulate the worst-case disturbance as,

$$d^\star(\bar{x};t) = \arg\max_{d} \mathcal{Q}(\bar{x};u_{\mathrm{d}},d;t) = -Q_{\mathrm{dd}}^{-1}Q_{\mathrm{dx}}(t)\bar{x}. \tag{22}$$

A. Actor-Critic Framework

We use a critic approximator augmented with past data to approximate the Q-function (18). Considering the compact quadratic form of the Q-function (20), let us define $\nu(t)^\intercal W_{\mathrm{c}} \coloneqq \mathrm{vech}(\bar{\mathcal{Q}}(t))/2 \in \mathbb{R}^{(n+m+q)(n+m+q+1)/2}$, where $\nu(t) \in \mathbb{R}^{((n+m+q)(n+m+q+1)/2)\times((n+m+q)(n+m+q+1)/2)}$ is a universal basis function that depends explicitly on time, $t \geq 0$. Since the ideal weight parameters are unknown, we leverage adaptive control [31] for tuning the weights. Hence, the approximated Q-function takes the form of,

$$\hat{\mathcal{Q}}(\bar{x};u_{\mathrm{d}},d;t) = \hat{W}_{\mathrm{c}}^\intercal \nu(t)(U \otimes U). \tag{23}$$

The control actor to approximate the optimal intermittent control (21) is defined as $W_{\mathrm{a}}^\intercal \mu(t) \coloneqq -Q_{\mathrm{u_du_d}}^{-1}Q_{\mathrm{u_dx}}(t) = -R^{-1}Q_{\mathrm{u_dx}}(t) \in \mathbb{R}^{m\times n}$, where $\mu(t) \in \mathbb{R}^{n\times n}$ is also a universal basis function that depends explicitly on time, $t \geq 0$. Thus, the approximated intermittent control policy yields,

$$\hat{u}_{\mathrm{d}}(\hat{\bar{x}};t) = \hat{W}_{\mathrm{a}}^\intercal \mu(t)\hat{\bar{x}}. \tag{24}$$

Similarly, we define the disturbance actor for approximating the worst-case disturbance (22) as $W_{\mathrm{d}}^\intercal \xi(t) \coloneqq -Q_{\mathrm{dd}}^{-1}Q_{\mathrm{dx}}(t) = \gamma^{-2}Q_{\mathrm{dx}}(t) \in \mathbb{R}^{q\times n}$, and $\xi(t) \in \mathbb{R}^{n\times n}$ is a universal basis function that depends explicitly on time, $t \geq 0$. The approximated worst-case disturbance yields,

$$\hat{d}(\bar{x};t) = \hat{W}_{\mathrm{d}}^\intercal \xi(t)\bar{x}. \tag{25}$$

Remark 3: The critic and actor approximators (23), (24), (25) have no approximation errors since $\nu(t)$, $\mu(t)$, $\xi(t)$ are universal spatio-temporal approximators. Therefore, since we use the whole space instead of just a compact set, the critic and actor approximators will converge to the optimum. □

Next, we leverage the integral reinforcement learning structure [17] to formulate the integral Bellman equation as,

$$\begin{aligned} V^\star(\bar{x}(t);t) =& V^\star(\bar{x}(t-\Delta t);t-\Delta t) \\ & - \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^\intercal M\bar{x} + u_{\mathrm{c}}^{\star\intercal}Ru_{\mathrm{c}}^\star - \gamma^2\|d^\star\|^2)\,\mathrm{d}\tau, \end{aligned} \tag{26}$$

$$V^\star(\bar{x}(T);T) = \frac{1}{2}\bar{x}^\intercal(T)P(T)\bar{x}(T), \tag{27}$$

where $\Delta t \in \mathbb{R}^+$ is a small, fixed time resolution. Moreover, using Lemma 2, (26), and (27) the following equations hold,

$$\begin{aligned} &\mathcal{Q}^\star(\bar{x}(t);u_{\mathrm{d}}^\star(t),d^\star(t);t) = \\ &\quad \mathcal{Q}^\star(\bar{x}(t-\Delta t);u_{\mathrm{d}}^\star(t-\Delta t),d^\star(t-\Delta t);t-\Delta t) \\ &\quad -\frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^\intercal M\bar{x} + u_{\mathrm{d}}^{\star\intercal}Ru_{\mathrm{d}}^\star - \gamma^2\|d^\star\|^2)\,\mathrm{d}\tau, \\ &\mathcal{Q}^\star(\bar{x}(T);u_{\mathrm{d}}^\star(T),d^\star(T);T) = \frac{1}{2}\bar{x}^\intercal(T)P(T)\bar{x}(T). \end{aligned}$$

B. Learning with Past Data

The next step is to develop a learning framework to approximate $\hat{\mathcal{Q}}$, $\hat{u}_{\rm d}$, $\hat{d}$ in (23), (24), (25). First, we need to guarantee that the probing signal is persistently exciting (PE).

Definition 1: A signal vector $\Delta(t) : \mathbb{R}^+ \to \mathbb{R}^n$ is PE over the time interval $[t, t+T_{\rm PE}]$, $T_{\rm PE} \in \mathbb{R}^+$, if there exists a strictly positive constant $\zeta \in \mathbb{R}^+$ such that $\zeta \mathrm{I} \leq \int_t^{t+T_{\rm PE}} \Delta(\tau)\Delta(\tau)^\intercal \mathrm{d}\tau$, where I is an identity matrix of appropriate dimensions. □

Corollary 1: If a signal vector $\Delta(t)$ is PE, then it is guaranteed that the unknown parameter vectors of Q-learning will exponentially converge to the optimal values, i.e., $\hat{W}_{\rm c}^\intercal \nu(t) \to (W_{\rm c}^\intercal \nu)^\star$, $\hat{W}_{\rm a}^\intercal \mu(t) \to (W_{\rm a}^\intercal \mu)^\star$, and $\hat{W}_{\rm d}^\intercal \xi(t) \to (W_{\rm d}^\intercal \xi)^\star$.

Proof. The proof follows from [31, Corollary 4.3.1]. ■

In practice, ensuring a PE probing signal is challenging. To this end, we inherit the analysis in [39], [40] to relax the PE condition. This approach employs past recorded data concurrently with current data for learning adaptation. Hence, if the recorded data is sufficiently rich, then the convergence of parameters to the optimal values can be guaranteed without enforcing the PE condition. In other words, initially we apply a PE signal and concurrently we store the past data of the probing signal in a buffer. When the buffer ensures a rich enough signal, we terminate the PE condition and continue the learning with the data collected in the buffer. Note that the data recording mechanism can be aperiodic.

The unknown elements of the intermittent Hamiltonian (15) are $\dot{\bar{x}}$ and $V^\star$. Given Δt as the time resolution, we approximate $\dot{\bar{x}}$ using finite differences,

$$\dot{\bar{x}} \approx (\bar{x}(t) - \bar{x}(t-\Delta t))/\Delta t. \tag{28}$$

According to Lemma 2 and (23), we approximate the derivatives of $V^\star$ as,

$$\frac{\partial V^\star}{\partial \bar{x}} \approx \frac{\partial \hat{\mathcal{Q}}}{\partial \bar{x}} = \hat{W}_{\rm c}^\intercal \nu(t) \nabla_{\bar{x}} \Big(U \otimes U \Big), \tag{29}$$

$$\frac{\partial V^\star}{\partial t} \approx \frac{\partial \hat{\mathcal{Q}}}{\partial t} = \hat{W}_{\rm c}^\intercal \nabla_t \Big(\nu(t)(U \otimes U) \Big), \tag{30}$$

where $\nabla_t(\cdot)$ denotes the gradient with respect to t, and $\nabla_{\bar{x}}(\cdot)$ with respect to $\bar{x}$. Therefore, with eqs. (29) and (30), the intermittent Hamiltonian (15) is approximated as,

$$\begin{aligned}\hat{\mathcal{H}}(U; \hat{W}_{\rm c}; t) =& \frac{1}{2} \mathrm{vech}\Big(\begin{bmatrix} M & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & -\gamma^2 \end{bmatrix} \Big)^\intercal (U \otimes U) \\ &+ \hat{W}_{\rm c}^\intercal \Big(\nabla_t \Big(\nu(t)(U \otimes U) \Big) + \nu(t) \nabla_{\bar{x}} \Big(U \otimes U \Big) \dot{\bar{x}} \Big).\end{aligned} \tag{31}$$

Using the integral Bellman equation, let us define the critic approximation error $e_{\rm c_1} \in \mathbb{R}$ as,

$$\begin{aligned} e_{\rm c_1}(t) :=& \hat{\mathcal{Q}}(\bar{x}(t); \hat{u}_{\rm d}(t), \hat{d}(t); t) \\ &- \hat{\mathcal{Q}}(\bar{x}(t-\Delta t); \hat{u}_{\rm d}(t-\Delta t), \hat{d}(t-\Delta t); t-\Delta t) \\ &+ \frac{1}{2} \int_{t-\Delta t}^{t} (\bar{x}^\intercal M \bar{x} + \hat{u}_{\rm d}^\intercal R \hat{u}_{\rm d} - \gamma^2 \|d\|^2) \, \mathrm{d}\tau \\ =& \hat{W}_{\rm c}^\intercal \Big(\nu(t) U(t) \otimes U(t) - \nu(t-\Delta t) U(t-\Delta t) \otimes U(t-\Delta t) \Big) \\ &+ \frac{1}{2} \int_{t-\Delta t}^{t} (\bar{x}^\intercal M \bar{x} + \hat{u}_{\rm d}^\intercal R \hat{u}_{\rm d} - \gamma^2 \|d\|^2) \, \mathrm{d}\tau. \end{aligned}$$

To drive the system to the final states—penalized by $P(T)$—we define the final critic error $e_{\rm c_2} \in \mathbb{R}$ as,

$$e_{\rm c_2}(t) := \frac{1}{2} \bar{x}(t)^\intercal P(T) \bar{x}(t) - \hat{W}_{\rm c}^\intercal \nu(t) (U(t) \otimes U(t)).$$

Moreover, every time we record data, the corresponding buffer critic error $e_{{\rm buff},i} \in \mathbb{R}$ is defined as, $e_{{\rm buff},i}(t_i) := \hat{\mathcal{H}}(U(t_i); \hat{W}_{\rm c}; t_i) - (\hat{u}_{\rm c}(t_i) - \hat{u}_{\rm d}(t_i))^\intercal R (\hat{u}_{\rm c}(t_i) - \hat{u}_{\rm d}(t_i))/2$, which by using (31) yields,

$$\begin{aligned} e_{{\rm buff},i}(t_i) :=& \frac{1}{2} \mathrm{vech}\Big(\begin{bmatrix} M & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & -\gamma^2 \end{bmatrix} \Big)^\intercal (U(t_i) \otimes U(t_i)) \\ &+ \hat{W}_{\rm c}^\intercal \nu(t_i) \nabla_{\bar{x}} \Big(U(t_i) \otimes U(t_i) \Big) \dot{\bar{x}} \\ &+ \hat{W}_{\rm c}^\intercal \nabla_t \Big(\nu(t_i)(U(t_i) \otimes U(t_i)) \Big) \\ &- \frac{1}{2} (\hat{u}_{\rm c}(t_i) - \hat{u}_{\rm d}(t_i))^\intercal R (\hat{u}_{\rm c}(t_i) - \hat{u}_{\rm d}(t_i)), \end{aligned}$$

where t_i is the data recording time. Next, we define the control actor approximation error $e_{\rm a} \in \mathbb{R}^m$ at $t = r_j$ as,

$$e_{\rm a}(t) := \hat{W}_{\rm a}^\intercal \mu(t) \bar{x}(r_j) + \hat{Q}_{\rm u_d u_d}^{-1} \hat{Q}_{\rm u_d x}(t) \bar{x}(r_j),$$

and the disturbance actor approximation error $e_{\rm d} \in \mathbb{R}^q$ as,

$$e_{\rm d}(t) := \hat{W}_{\rm d}^\intercal \xi(t) \bar{x}(t) + \hat{Q}_{\rm dd}^{-1} \hat{Q}_{\rm dx}(t) \bar{x}(t).$$

We aim to drive the errors to zero by tuning the parameters $\hat{W}_{\rm c}^\intercal$, $\hat{W}_{\rm a}^\intercal$, $\hat{W}_{\rm d}^\intercal$ in (23), (24), (25). Thus, we apply gradient descent to the squared norms of the errors,

$$K_1(\hat{W}_{\rm c}) = \frac{1}{2} \|e_{\rm c_1}\|^2 + \frac{1}{2} \|e_{\rm c_2}\|^2 + \frac{1}{2} \sum_{i=1}^{N_k} \|e_{{\rm buff},i}\|^2, \tag{32}$$

$$K_2(\hat{W}_{\rm a}) = \frac{1}{2} \|e_{\rm a}\|^2, \tag{33}$$

$$K_3(\hat{W}_{\rm d}) = \frac{1}{2} \|e_{\rm d}\|^2, \tag{34}$$

where $N_k \in \mathbb{N}$ is the number of data recording at the k-th two point zero-sum game.

Remark 4: Since K_1, K_2, K_3 are the sum of squared norms, the corresponding optimization problems are convex. Thus, the gradient descent rule yields the global value. Moreover, solving the game (4) is transformed into solving convex optimization problems, and hence a real-time execution is ensured for the robust intermittent Q-learning controller. □

C. Learning Framework

The learning framework is employed to drive the approximated cost, intermittent control, and disturbance to the optimal ones, and it consists of three tuning laws. We apply a normalized gradient descent method (similar to adaptive control techniques [31]) to (32) that result in closed-form critic approximation weights,

$$\dot{\hat{W}}_{\mathrm{c}} = -\alpha_{\mathrm{c}}\frac{\partial K_1}{\partial \hat{W}_{\mathrm{c}}} = -\alpha_{\mathrm{c}}\Big(\frac{\sigma(t)e_{\mathrm{c}_1}}{(1+\sigma(t)^\intercal\sigma(t))^2} + \frac{\sigma_{\mathrm{f}}(t)e_{\mathrm{c}_2}}{(1+\sigma_{\mathrm{f}}(t)^\intercal\sigma_{\mathrm{f}}(t))^2} + \sum_{i=1}^{N_k}\frac{\omega(t_i)e_{\mathrm{buff},i}}{(1+\omega(t_i)^\intercal\omega(t_i))^2}\Big), \quad (35)$$

where $\sigma_{\mathrm{f}}(t) \coloneqq \nu(t)(U(t)\otimes U(t))$, $\sigma(t) \coloneqq \nu(t)(U(t)\otimes U(t)) - \nu(t-\Delta t)(U(t-\Delta t)\otimes U(t-\Delta t))$, $\alpha_{\mathrm{c}} \in \mathbb{R}^+$ is the gradient descent learning rate that specifies the convergence rate of the critic, and $\omega(t_i)$ is defined as,

$$\omega(t_i) \coloneqq \nabla_t\Big(\nu(t_i)(U(t_i)\otimes U(t_i))\Big) + \nu(t_i)\nabla_{\bar{x}}\Big(U(t_i)\otimes U(t_i)\Big)\dot{\bar{x}},$$

where $\sigma(t), \sigma_{\mathrm{f}}(t), \omega(t_i) \in \mathbb{R}^{(n+m+q)(n+m+q+1)/2}$. By following the critic tuning law (35), the sum of squared critic errors is regulated to zero.

For the update law of the control actor, we need to consider its intermittent behavior. More specifically, the intermittent controller will be updated only when the error gap (2) is large enough to activate the triggering condition, otherwise it will remain constant as the last updated value. That is precisely an impulsive system, hence we inherit the approaches in [48], [49] to describe the intermittent control actor behavior. We obtain a closed-form tuning law for the control actor approximation weights $\hat{W}_{\mathrm{a}}$ by applying gradient descent to (33) as,

$$\begin{cases} \dot{\hat{W}}_{\mathrm{a}} = 0, & t \in (r_j, r_{j+1}) \\ \hat{W}_{\mathrm{a}}^+ = \hat{W}_{\mathrm{a}} - \alpha_{\mathrm{a}}\frac{\partial K_2}{\partial \hat{W}_{\mathrm{a}}} \\ \qquad = \hat{W}_{\mathrm{a}} - \alpha_{\mathrm{a}}\frac{\mu(t)\bar{x}}{1+(\mu(t)\bar{x})^\intercal(\mu(t)\bar{x})}e_{\mathrm{a}}^\intercal, & t = r_j \end{cases} \quad (36)$$

where the convergence rate is determined by the gradient descent learning rate $\alpha_{\mathrm{a}} \in \mathbb{R}^+$. The control actor tuning law (36) ensures that e_{a} converges to zero. Next, the disturbance actor tuning law in closed-form yields,

$$\dot{\hat{W}}_{\mathrm{d}} = -\alpha_{\mathrm{d}}\frac{\partial K_3}{\partial \hat{W}_{\mathrm{d}}} = -\alpha_{\mathrm{d}}\frac{\xi(t)\bar{x}}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}e_{\mathrm{d}}^\intercal, \quad (37)$$

where $\alpha_{\mathrm{d}} \in \mathbb{R}^+$ is the gradient descent learning rate. Through the disturbance tuning law (37), the disturbance error e_{d} also converges to zero.

For the theoretical convergence analysis, we define the approximation weights errors of the critic, the control actor, and the disturbance actor as $\tilde{W}_{\mathrm{c}} \coloneqq W_{\mathrm{c}} - \hat{W}_{\mathrm{c}} \in \mathbb{R}^{(n+m+q)(n+m+q+1)/2}$, $\tilde{W}_{\mathrm{a}} \coloneqq W_{\mathrm{a}} - \hat{W}_{\mathrm{a}} \in \mathbb{R}^{n\times m}$, and $\tilde{W}_{\mathrm{d}} \coloneqq W_{\mathrm{d}} - \hat{W}_{\mathrm{d}} \in \mathbb{R}^{n\times q}$. Thus, using (35), (36), (37) we express their dynamics as,

$$\dot{\tilde{W}}_{\mathrm{c}} = -\alpha_{\mathrm{c}}\Big(\frac{\sigma(t)\sigma(t)^\intercal}{(1+\sigma(t)^\intercal\sigma(t))^2} + \Lambda\Big)\tilde{W}_{\mathrm{c}},$$

$$\begin{cases} \dot{\tilde{W}}_{\mathrm{a}} = 0, & t \in (r_j, r_{j+1}) \\ \tilde{W}_{\mathrm{a}}^+ = \tilde{W}_{\mathrm{a}} - \alpha_{\mathrm{a}}\frac{(\mu(t)\bar{x})(\mu(t)\bar{x})^\intercal}{1+(\mu(t)\bar{x})^\intercal(\mu(t)\bar{x})}\tilde{W}_{\mathrm{a}} \\ \qquad -\alpha_{\mathrm{a}}\frac{(\mu(t)\bar{x})\bar{x}^\intercal}{1+(\mu(t)\bar{x})^\intercal(\mu(t)\bar{x})}\tilde{Q}_{\mathrm{xu_d}}Q_{\mathrm{u_du_d}}^{-1}, & t = r_j \end{cases}$$

$$\dot{\tilde{W}}_{\mathrm{d}} = -\alpha_{\mathrm{d}}\frac{(\xi(t)\bar{x})(\xi(t)\bar{x})^\intercal}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\tilde{W}_{\mathrm{d}} - \alpha_{\mathrm{d}}\frac{(\xi(t)\bar{x})\bar{x}^\intercal}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\tilde{Q}_{\mathrm{xd}}Q_{\mathrm{dd}}^{-1},$$

where $\Lambda \coloneqq \sum_{i=1}^{N_k}\frac{\omega(t_i)\omega(t_i)^\intercal}{(1+\omega(t_i)^\intercal\omega(t_i))^2}$, $\tilde{Q}_{\mathrm{xu_d}} \coloneqq \mathrm{mat}(\tilde{W}_{\mathrm{c}})[1\colon n, n+1\colon n+m]$, and $\tilde{Q}_{\mathrm{xd}} \coloneqq \mathrm{mat}(\tilde{W}_{\mathrm{c}})[1\colon n, n+m+1\colon n+m+q]$.

Remark 5: After the data recording is rich enough to obtain $\mathtt{rank}([\omega(t_1), \omega(t_2), \ldots, \omega(t_{N_k})]) \geq n+m+q$, matrix Λ gets a full rank and $\Lambda \succ 0$. This is the relaxed PE condition which indicates that the recorded data of the PE probing signal is sufficient for convergence of $\hat{W}_{\mathrm{c}}$ to the optimum, and thus it is no longer needed to record new data afterwards [40]. Using the realxed PE condition $\hat{\mathcal{Q}}$ converges to the actual optimal cost as $\hat{W}_{\mathrm{c}}$, $\hat{W}_{\mathrm{a}}$ and $\hat{W}_{\mathrm{d}}$ converge to their optimal solutions.

D. Impulsive System

Since the controller has an intermittent behavior with discrete jumps, we consider it as an impulsive system. Therefore, the closed-loop dynamics of (1) with the intermittent controller (24) and the worst-case disturbance (25) take the form of,

$$\begin{aligned} \dot{\bar{x}} &= A\bar{x} + B\hat{u}_{\mathrm{d}} + F\hat{d} \\ &= A\bar{x} + B(-R^{-1}Q_{\mathrm{u_dx}}(t) - \tilde{W}_{\mathrm{a}}^\intercal\mu(t))\hat{\bar{x}} \\ &\quad + F(\gamma^{-2}Q_{\mathrm{dx}}(t) - \tilde{W}_{\mathrm{d}}^\intercal\xi(t))\bar{x}. \end{aligned} \quad (38)$$

Let us define the augmented state that captures the flow dynamics of the system in the time interval $t \in (r_j, r_{j+1}]$ as, $\psi \coloneqq [\bar{x}^\intercal\ \hat{\bar{x}}^\intercal\ \tilde{W}_{\mathrm{c}}^\intercal\ \mathrm{vec}(\tilde{W}_{\mathrm{a}})^\intercal\ \mathrm{vec}(\tilde{W}_{\mathrm{d}})^\intercal]^\intercal \in \mathbb{R}^{2n+nm+nq+(n+m+q)(n+m+q+1)/2}$, with time derivative,

$$\dot{\psi} = \begin{bmatrix} \dot{\hat{\bar{x}}} \\ \mathbf{0}_n \\ -\alpha_{\mathrm{c}}\Big(\frac{\sigma(t)\sigma(t)^\intercal}{(1+\sigma(t)^\intercal\sigma(t))^2} + \Lambda\Big)\tilde{W}_{\mathrm{c}} \\ \mathbf{0}_{nm} \\ \Psi(\tilde{W}_{\mathrm{d}}) \end{bmatrix}. \quad (39)$$

Then, the augmented state is defined as $\psi^+ \coloneqq [\bar{x}^{+\intercal}\ \hat{\bar{x}}^{+\intercal}\ \tilde{W}_{\mathrm{c}}^{+\intercal}\ \mathrm{vec}(\tilde{W}_{\mathrm{a}}^+)^\intercal\ \mathrm{vec}(\tilde{W}_{\mathrm{d}}^+)^\intercal]^\intercal$ for the jump dynamics at time $t = r_j$, with the time derivative yielding,

$$\psi^+ = \psi(t) + \begin{bmatrix} \mathbf{0}_n \\ \hat{\bar{x}}(t) - \bar{x}(t) \\ \mathbf{0}_{(n+m+q)(n+m+q+1)/2} \\ \Psi(\tilde{W}_{\mathrm{a}}^+) \\ \mathbf{0}_{nq} \end{bmatrix}, \quad (40)$$

where $\Psi(\tilde{W}_{\mathrm{d}}) \coloneqq \mathrm{vec}(-\alpha_{\mathrm{d}}\frac{(\xi(t)\bar{x})(\xi(t)\bar{x})^\intercal}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\tilde{W}_{\mathrm{d}} - \alpha_{\mathrm{d}}\frac{(\xi(t)\bar{x})\bar{x}^\intercal}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\tilde{Q}_{\mathrm{xd}}Q_{\mathrm{dd}}^{-1})$, $\Psi(\tilde{W}_{\mathrm{a}}^+) \coloneqq \mathrm{vec}(-\alpha_{\mathrm{a}}\frac{(\mu(t)\bar{x})(\mu(t)\bar{x})^\intercal}{1+(\mu(t)\bar{x})^\intercal(\mu(t)\bar{x})}\tilde{W}_{\mathrm{a}} - \alpha_{\mathrm{a}}\frac{(\mu(t)\bar{x})\bar{x}^\intercal}{1+(\mu(t)\bar{x})^\intercal(\mu(t)\bar{x})}\tilde{Q}_{\mathrm{xu_d}}Q_{\mathrm{u_du_d}}^{-1})$.

By analyzing the impulsive system, the main stability theorem of the robust intermittent Q-learning is then presented.

Theorem 3: Consider the impulsive closed-loop system (38), with the critic approximator (23), the actor approximator (24), and the disturbance approximator (25), which are tuned by the tuning laws (35), (36), and (37) respectively. Then, given that $T \to \infty$ the origin is a globally asymptotically stable equilibrium of the closed-loop system with state ψ for all initial conditions $\psi(0)$, if the following inequality holds,

$$\|e\|^2 \leq \frac{(1-\beta^2)\underline{\lambda}(M)\|\bar{x}\|^2 + \underline{\lambda}(R)\|\hat{u}_{\mathrm{d}}\|^2 - \gamma^2\|\hat{d}\|^2}{4\overline{\lambda}(R)\Big(L(t)^2 + L_1(t)^2\Big)}, \quad (41)$$

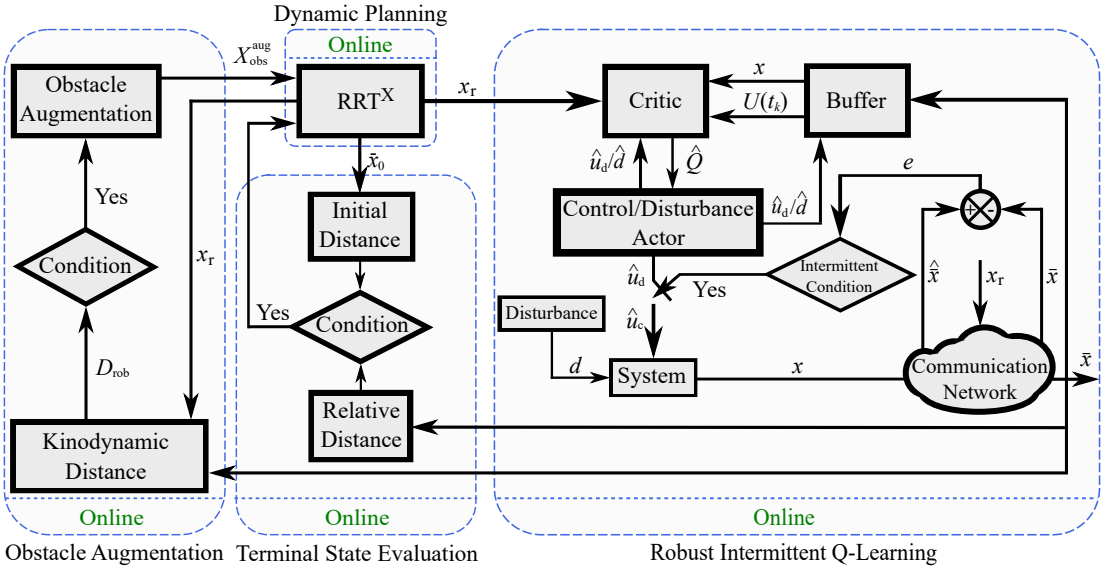

Fig. 1: **Structure of RRT-Q$^{\mathrm{X}}_{\infty}$.** RRT-Q$^{\mathrm{X}}_{\infty}$ comprises four stages: i) global dynamic path planning, ii) online actor-critic framework, iii) online terminal state evaluation, and iv) online obstacle augmentation. The framework starts from global dynamic path planning and runs clockwise.

and the following inequalities of constants hold,

$$0 < \alpha_{\mathrm{d}} \leq \frac{\beta^2 \underline{\lambda}(M) - 2L_1(t)^2 \overline{\lambda}(R)}{\eta \overline{\lambda}\left(\frac{\xi(t) Q_{\mathrm{dd}}^{-1}}{\|1+(\xi(t)\bar{x})^{\intercal}(\xi(t)\bar{x})\|^2}\right)}; \quad \frac{\underline{\lambda}(M)}{\overline{\lambda}(R)} > \frac{2L_1(t)^2}{\beta^2}; \tag{42}$$

$$0 < \alpha_{\mathrm{a}} < \frac{8\overline{\lambda}(R) - 2}{\overline{\lambda}(R) + 2}; \quad \overline{\lambda}(R) > 0.25; \quad \alpha_{\mathrm{c}} \gg \alpha_{\mathrm{a}}. \tag{43}$$

Proof. See Appendix A. ∎

V. The RRT-Q$^{\mathrm{X}}_{\infty}$ Framework

We present the RRT-Q$^{\mathrm{X}}_{\infty}$ framework (Sections V-A and V-B) and analyze its computational complexity (Section V-C), In particular, RRT-Q$^{\mathrm{X}}_{\infty}$ consists of four stages: i) global path planning/replanning by RRT$^{\mathrm{X}}$; ii) online control with an actor-critic framework equipped with a data buffer; iii) online obstacle augmentation; and iv) online terminal state evaluation (see Figure 1).

The main routine is presented in Algorithm 1. The gradient descent gains $\alpha_{\mathrm{c}}, \alpha_{\mathrm{a}}, \alpha_{\mathrm{d}}$ are designed according to Theorem 3, and the weights of approximators $\hat{W}_{\mathrm{c}}, \hat{W}_{\mathrm{a}}, \hat{W}_{\mathrm{d}}$ are randomly initialized (lines 1–2). In the beginning, the agent has not started moving and computes the global path using RRT$^{\mathrm{X}}$ (line 20). Next, the agent performs safe navigation (lines 7–18). The agent first applies the output of the proposed robust intermittent Q-learning controller (line 7). Then, it performs the obstacle augmentation strategy to ensure a collision-free trajectory (lines 8–13). The augmented obstacle space $\mathcal{X}^{\mathrm{aug}}_{\mathrm{obs}}$ is calculated by means of the Minkowski sum (8) based on the maximum kinodynamic distance $D^{\mathrm{kin}}_{\mathrm{rob}}$. Finally, the agent evaluates if it should move on to the next BVP (lines 14–18): if the agent is close enough to the desired state $x_{\mathrm{r},k}$, i.e., $D \leq \rho D_0$, $\rho \in (0,1)$, then $x_{\mathrm{r},k}$ will be considered as reached, and the agent will proceed to the $(k+1)$-th BVP. The larger the *admissible window* ρ is, the sooner the agent can proceed to the next BVP. RRT-Q$^{\mathrm{X}}_{\infty}$ terminates once the goal state x_{goal} is considered as reached.

Algorithm 1 RRT-Q$^{\mathrm{X}}_{\infty}$

Input: $\mathcal{X}$ - state space; $\mathcal{S}$ - random sample sequence; $\mathcal{X}_{\mathrm{obs}}$ - obstacle space; Δt - small time resolution; M, R - cost weight matrices; γ - disturbance rejection constant; $P(T)$ - fixed final Riccati matrix; L, L_1 - event-trigger constants; β - bandwidth parameter; ρ - admissible window; x_{start} - start state; x_{goal} - goal state; $x(t)$ - state feedback.
Output: $\hat{u}_{\mathrm{d}}(t)$ - control.

1: $\alpha_{\mathrm{c}}, \alpha_{\mathrm{a}}, \alpha_{\mathrm{d}} \leftarrow$ `stability`(M, R, γ, L, L_1);
2: $\hat{W}_{\mathrm{c}}, \hat{W}_{\mathrm{a}}, \hat{W}_{\mathrm{d}} \leftarrow$ `random`; $D^{\mathrm{kin}}_{\mathrm{rob}} \leftarrow 0$; $\mathcal{X}^{\mathrm{aug}}_{\mathrm{obs}} \leftarrow \mathcal{X}_{\mathrm{obs}}$; $k \leftarrow 1$;
3: $V \leftarrow \{x_{\mathrm{goal}}\}$;
4: $x(0) \leftarrow x_{\mathrm{start}}$;
5: **while** $x(t) \neq x_{\mathrm{goal}}$ **do**
6: **if** *agent is moving* **then**
7: $\hat{u}_{\mathrm{d}}(t) \leftarrow$ `robustIntermittentQLearning`;
8: $D_{\mathrm{rob}} \leftarrow$ `kinodynamicDistance`$(x(t))$;
9: **if** $D_{\mathrm{rob}} > D^{\mathrm{kin}}_{\mathrm{rob}}$ **then**
10: $D^{\mathrm{kin}}_{\mathrm{rob}} \leftarrow D_{\mathrm{rob}}$;
11: $\mathcal{X}_{\mathrm{kin}} \leftarrow$ `circle`$(D^{\mathrm{kin}}_{\mathrm{rob}})$;
12: $\mathcal{X}^{\mathrm{aug}}_{\mathrm{obs}} \leftarrow \mathcal{X}_{\mathrm{obs}} \oplus \mathcal{X}_{\mathrm{kin}}$;
13: **end if**
14: $D \leftarrow \|x(t) - x_{\mathrm{r},k}\|$; $D_0 \leftarrow \|x_{0,k} - x_{\mathrm{r},k}\|$;
15: **if** $D \leq \rho D_0$ **then**
16: $x_{0,k+1} \leftarrow x(t)$;
17: $k \leftarrow k+1$;
18: **end if**
19: **end if**
20: $G, \pi \leftarrow$ RRT$^{\mathrm{X}}(\mathcal{X}^{\mathrm{aug}}_{\mathrm{obs}}, \mathcal{S})$;
21: **end while**

A. Global Path Planning/Replanning by RRT$^{\mathrm{X}}$

RRT-Q$^{\mathrm{X}}_{\infty}$ achieves global path planning and online replanning by integrating RRT$^{\mathrm{X}}$ [5]. The inputs of RRT$^{\mathrm{X}}$ are the agent's state space $\mathcal{X}$ and a uniformly distributed random sample sequence $\mathcal{S}$ of which the samples are from $\mathcal{X}$, and

the output is a collision-free path π. The waypoints of π will then induce a series of BVPs for the robust intermittent Q-learning control problem.

Each iteration of RRT$^{\mathrm{X}}$ (Algorithm 2) begins with updating the neighborhood radius (line 4). Then, the algorithm updates the change of the obstacle space and the agent's state if any (lines 5–10). Next, the "standard" sampling-based path planner grows and refines the graph by obtaining new samples and connecting them to the existing graph if possible (lines 11–18). The rewiring operation guarantees asymptotic optimality while maintaining a limited neighborhood size (line 20). ϵ-consistency is performed for rewiring cascade (line 21).

B. Robust Intermittent Q-Learning

We present the robust intermittent Q-learning method (Algorithm 3). For the k-th BVP, i.e., $(x_{0,k},\ x_{\mathrm{r},k})$ in the path π, a zero-sum game between the optimal intermittent control and the worst-case disturbance is formulated (Section III), and the proposed method performs waypoint navigation, i.e., drives the system from $x_{0,k}$ to $x_{\mathrm{r},k}$, by solving the game. Particularly, in Algorithm 3, the critic parameters $\hat{W}_{\mathrm{c}}$ are regulated online by (35), and the critic approximates the Q-function $\hat{\mathcal{Q}}$ by (23) (lines 1–3). The gradient of $\hat{W}_{\mathrm{c}}$ is also influenced by `recordData` such that all weight parameters converge to the optimal values with a relaxed PE requirement (line 2). Given the approximated Q-function $\hat{\mathcal{Q}}$, every time the triggering condition (41) is activated, the control actor (24) updates the approximated optimal intermittent control $\hat{u}_{\mathrm{d}}$ and the control actor parameters $\hat{W}_{\mathrm{a}}$ are tuned (36) (lines 4–8). Then, the disturbance actor parameters $\hat{W}_{\mathrm{d}}$ are tuned by (37) to approximate the worst-case disturbance $\hat{d}$ (25) (lines 9–10). Finally, Algorithm 3 returns $\hat{u}_{\mathrm{d}}$ and applies it to the system.

With the data recording operation (Algorithm 4), the robust intermittent Q-learning method makes sure $\hat{W}_{\mathrm{c}}$, $\hat{W}_{\mathrm{a}}$, and $\hat{W}_{\mathrm{d}}$ converge to their optimal values without needing the PE requirement (Corollary 1). In particular, this operation records concurrent data in a buffer and updates the gradient of $\hat{W}_{\mathrm{c}}$ until $\mathtt{rank}([\omega(t_1),\omega(t_2),\ldots,\omega(t_{N_k})]) \geq n+m+q$, where N_k is the number of already recorded data points. This operation can be executed either periodically or aperiodically [39], [40].

C. Computational Complexity

The computational complexity of RRT-Q$^{\mathrm{X}}_{\infty}$ is determined by three components: $\mathcal{C}_1$ for RRT$^{\mathrm{X}}$, $\mathcal{C}_2$ for robust intermittent Q-learning, and $\mathcal{C}_3$ for obstacle augmentation. In particular, $\mathcal{C}_1$ for RRT$^{\mathrm{X}}$ is different for static and dynamic environments. For a static environment, $\mathcal{C}_1 = \Theta(|V|\log|V|)$ is required to build a graph $G=(V,E)$. When the environment changes, RRT$^{\mathrm{X}}$ will first find V_{obs}, the set of all nodes whose trajectories cross the variation of the obstacle space $\Delta\mathcal{X}_{\mathrm{obs}}$. Then, the expected running time will be $\mathcal{C}_1 = \mathcal{O}(|\mathcal{D}(V_{\mathrm{obs}})|\log|V|)$, where $\mathcal{D}(V_{\mathrm{obs}})$ contains all descendants of all nodes in V_{obs} [5].

As for $\mathcal{C}_2$ for robust intermittent Q-learning, three operations are constantly executed during the flow dynamics: approximation of Q-function, approximation of disturbance, and error gap checking. For the approximation of Q-function by (23) and (35), the complexity has a quadratic growth of $\mathcal{O}((n+m+q)^2)$ where $n+m+q$ is the size of the augmented state. Similarly, the approximation of disturbance by (25) and (37) requires $\mathcal{O}(q^2)$. When checking the error gap by (41), the complexity is $\mathcal{O}(n^2+m^2+q^2)$. Thus, the computational complexity during the flow dynamics is $\mathcal{O}((n+m+q)^2)$. For the jump dynamics, the additional computation is required for updating $\hat{W}_{\mathrm{a}}$ and $\hat{u}_{\mathrm{d}}$ by (36) and (24), which takes $\mathcal{O}(n^2+m^2)$. Therefore, $\mathcal{C}_2 = \mathcal{O}((n+m+q)^2)$.

The obstacle augmentation complexity $\mathcal{C}_3$ depends on the Minkowski sum operation to the whole obstacle space and the following replanning. Let us assume all obstacles are convex polygons, the regular polygon that approximates the circular state space $\mathcal{X}_{\mathrm{kin}}$ has p_{kin} vertices, and each of the N_l obstacles in $\mathcal{X}_{\mathrm{obs}}$ has p_l vertices. According to [50, Theorem 13.11], the Minkowski sum of two convex polygons with p_1 and p_2 vertices in a plane can be computed in $\mathcal{O}(p_1+p_2)$ time. Then, the Minkowski sum takes $\mathcal{O}(N_o p_{\mathrm{kin}} + \Sigma_{l=1}^{N_o} p_l)$. The following

Algorithm 2 RRT$^{\mathrm{X}}$

1: $V \leftarrow \{x_{\mathrm{goal}}\}$;
2: $x(0) \leftarrow x_{\mathrm{start}}$;
3: **while** $x(t) \neq x_{\mathrm{goal}}$ **do**
4: $\quad r \leftarrow \mathtt{shrinkingBallRadius}(|V|)$;
5: $\quad$ **if** $\Delta\mathcal{X}_{\mathrm{obs}} \neq \varnothing$ **then**
6: $\quad\quad \mathtt{updateObstacles}(\mathcal{X}_{\mathrm{obs}})$;
7: $\quad$ **end if**
8: $\quad$ **if** *agent is moving* **then**
9: $\quad\quad x(t) \leftarrow \mathtt{measureState}()$;
10: $\quad$ **end if**
11: $\quad x_{\mathrm{new}} \leftarrow \mathtt{randomNode}(\mathcal{S})$;
12: $\quad x_{\mathrm{nearest}} \leftarrow \mathtt{nearest}(x_{\mathrm{new}})$;
13: $\quad$ **if** $d(x_{\mathrm{new}}, x_{\mathrm{nearest}}) > \delta$ **then**
14: $\quad\quad x_{\mathrm{new}} \leftarrow \mathtt{saturate}(x_{\mathrm{new}}, x_{\mathrm{nearest}})$;
15: $\quad$ **end if**
16: $\quad$ **if** $x_{\mathrm{new}} \notin \mathcal{X}_{\mathrm{obs}}$ **then**
17: $\quad\quad \mathtt{extend}(x_{\mathrm{new}}, r)$;
18: $\quad$ **end if**
19: $\quad$ **if** $x_{\mathrm{new}} \in V$ **then**
20: $\quad\quad \mathtt{rewireNeighbors}(x_{\mathrm{new}})$;
21: $\quad\quad \mathtt{reduceInconsistency}()$;
22: $\quad$ **end if**
23: **end while**

Algorithm 3 `robustIntermittentQLearning`

1: $\dot{\hat{W}}_{\mathrm{c}} \leftarrow \mathtt{critic}(\bar{x}, \hat{u}_{\mathrm{d}}, \alpha_{\mathrm{c}}, \hat{W}_{\mathrm{c}})$;
2: $\hat{W}_{\mathrm{c}} \leftarrow \mathtt{recordData}(\dot{\bar{x}}, \bar{x}, \hat{u}_{\mathrm{d}}, \hat{d}, N_k)$;
3: $\hat{\mathcal{Q}} \leftarrow \mathtt{approximateQ}(\bar{x}, \hat{u}_{\mathrm{d}}, \hat{W}_{\mathrm{c}})$;
4: $\|e\|^2 \leftarrow \mathtt{errorGap}(\bar{x}, \hat{\bar{x}})$;
5: **if** $\|e\|^2 \geq \mathtt{errorThreshold}(\beta, \bar{x}, \hat{u}_{\mathrm{d}}, \hat{d})$ **then**
6: $\quad \hat{W}_{\mathrm{a}}^{+} \leftarrow \mathtt{controlActor}(\bar{x}, \alpha_{\mathrm{a}}, \hat{W}_{\mathrm{a}}, \hat{\mathcal{Q}})$;
7: $\quad \hat{u}_{\mathrm{d}} \leftarrow \mathtt{approximateControl}(\bar{x}, \hat{W}_{\mathrm{a}})$;
8: **end if**
9: $\hat{W}_{\mathrm{d}} \leftarrow \mathtt{disturbanceActor}(\bar{x}, \alpha_{\mathrm{d}}, \hat{W}_{\mathrm{d}}, \hat{\mathcal{Q}})$;
10: $\hat{d} \leftarrow \mathtt{approximateDisturbance}(\bar{x}, \hat{W}_{\mathrm{d}})$;
11: **return** $\hat{u}_{\mathrm{d}}$;

Algorithm 4 `recordData`

if $\texttt{rank}([\omega(t_1), \omega(t_2), \ldots, \omega(t_{N_k})]) < n + m + q$ **then**
  $\omega(t_{N_k+1}) \leftarrow \nabla_t(\nu(t)(U(t) \otimes U(t))) + \nu(t)\nabla_{\bar{x}}(U(t) \otimes U(t))\dot{\bar{x}}$;
  $e_{\text{buff},N_k+1} \leftarrow \texttt{buffErr}(\dot{\bar{x}}, \bar{x}, \hat{u}_\text{d}, \hat{u}_\text{c}, \hat{d}, \hat{W}_\text{c}, \omega(t_{N_k+1}))$;
  $\dot{\hat{W}}_\text{c} \leftarrow \texttt{criticRelaxedPE}(\alpha_\text{c}, \hat{W}_\text{c}, e_{\text{buff},N_k+1})$;
  $N_k \leftarrow N_k + 1$;
  return $\hat{W}_\text{c}$;
end if

replanning considering the enlargement of the whole obstacle space requires $\mathcal{O}(|\mathcal{D}(V'_\text{obs})|\log|V|)$, where V'_obs is the set of all the nodes whose trajectories cross the enlarged obstacle space, and $\mathcal{D}(V'_\text{obs})$ contains all descendants of all nodes in V'_obs. Therefore, $\mathcal{C}_3 = \mathcal{O}(N_o p_\text{kin} + \Sigma_{l=1}^{N_o} p_l + |\mathcal{D}(V'_\text{obs})|\log|V|)$.

VI. Simulations and Results

In this section, we first evaluate the robust intermittent Q-learning method in a single two-player zero-sum game (TPZSG) (Section VI-A). We next evaluate RRT-Q$^\text{X}_\infty$ and show its effectiveness and safety through 30 Monte-Carlo trials of the navigation task in a simulated dynamic environment with appearing/disappearing obstacles (Section VI-B). Then, we present the dependence of the kinodynamic distance on the orientation of BVP and event-trigger bandwidth (Section VI-C). Finally, we qualitatively compare RRT-Q$^\text{X}_\infty$ with multiple other kinodynamic motion planning works (Section VI-D).

A. Solving a Two-Player Zero-Sum Game

We present how the robust intermittent Q-learning technique solves a single TPZSG. Consider the following linear time-invariant system that is unknown,

$$\begin{bmatrix}\dot{x}_1\\ \dot{y}_1\\ \dot{x}_2\\ \dot{y}_2\end{bmatrix} = A\begin{bmatrix}x_1\\ y_1\\ x_2\\ y_2\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0.025 & 0\\ 0 & 0.025\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\\ 1\end{bmatrix} d, \tag{44}$$

$$A = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -0.5 & 0 & -1.125 & 0\\ 0 & -0.5 & 0 & -1.125\end{bmatrix},$$

where x_1, y_1 denote the translations, $\dot{x}_1$, $\dot{y}_1$ the velocities, $\ddot{x}_1$, $\ddot{y}_1$ the accelerations, u_1, u_2 the control inputs, and d denotes the disturbance.

We seek to drive the system from $x_0 = [0, 3, 1, -1]$ to $x_\text{r} = [5, 1, 0, 0]$ for $T = 15$ s with the event-trigger bandwidth $\beta = 0.1$ and no disturbance, i.e., $d = 0$. The performance penalizing matrices are $M = 5I_4$, $R = I_2$, the disturbance rejection constant is $\gamma = 0.1$, and the final Riccati matrix is $P(T) = 0.3I_4$. According to Theorem 3, we select the tuning gains as $\alpha_\text{c} = 90$, $\alpha_\text{a} = 0.8$, $\alpha_\text{d} = 3$, and the Lipschitz constants are $L = 10$ and $L_1 = 0.9\beta(\underline{\lambda}(M)/(2\overline{\lambda}(R)))^{1/2} = 0.14$.

The evolution of states is shown in Figure 2(a) and the evolution of controls in Figure 2(b). The squared norm error (2) and its threshold (41) are presented by the blue and red lines in Figure 2(c). Every time the error (2) is close to the calculated threshold (41), the intermittent control is activated and the error is reset to zero. We set the upper bar of the error as 95% of the threshold, and the intermittent control is activated 61 times. The inter-event time, i.e., the time interval of two consecutive control updates, is shown in Figure 2(d). We find that the minimum inter-event time is 0.11 s and the average inter-event time is 0.24 s, which means the Zeno behavior [51] is not observed. Finally, the evolution of the approximation weights, i.e., $\hat{W}_\text{c}$, $\hat{W}_\text{a}$, and $\hat{W}_\text{d}$, is shown in Figure 2(e)–(g), where the convergence of all approximation weights is observed.

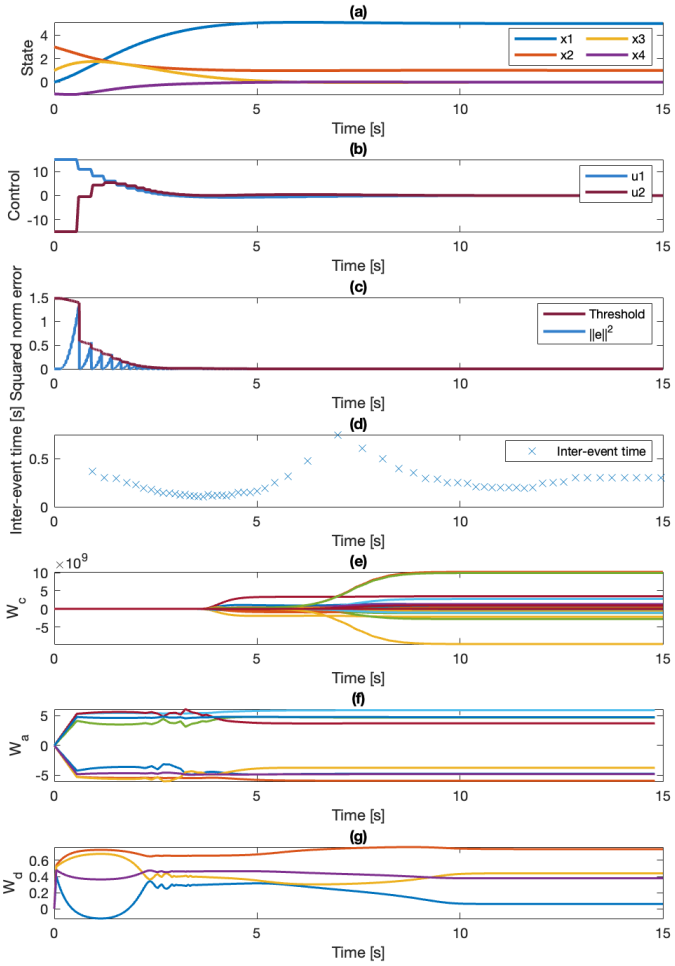

Fig. 2: **Solving a single TPZSG.** Limiting the squared norm error with a calculated threshold, the intermittent control policy guarantees the stability of the equilibrium point and conserves computational and communication efforts. (a)–(d) contain the evolution of the states, of the controls, of the error gap with the threshold, and the inter-event times of control updates for solving the game, respectively. (e)–(g) show the convergence of the approximation weights for the critic, control actor, and disturbance actor, respectively.

B. Motion Planning in a Dynamic Environment

In this section, we present the kinodynamic motion planning of an autonomous rover that follows (44) in a random forest obstacle environment. We set the finite horizon to $T = 10$ s, the event-trigger bandwidth $\beta = 0.9$, the performance penalizing matrices $M = 10I_4$, $R = 0.525I_2$, the disturbance rejection constant $\gamma = 15$, the final Riccati matrix $P(T) = 0.5I_4$, and the gradient descent gains $\alpha_\text{c} = 90$, $\alpha_\text{a} = 0.8$, $\alpha_\text{d} = 3$ per Theorem 3. The Lipschitz constants are $L = 10$,

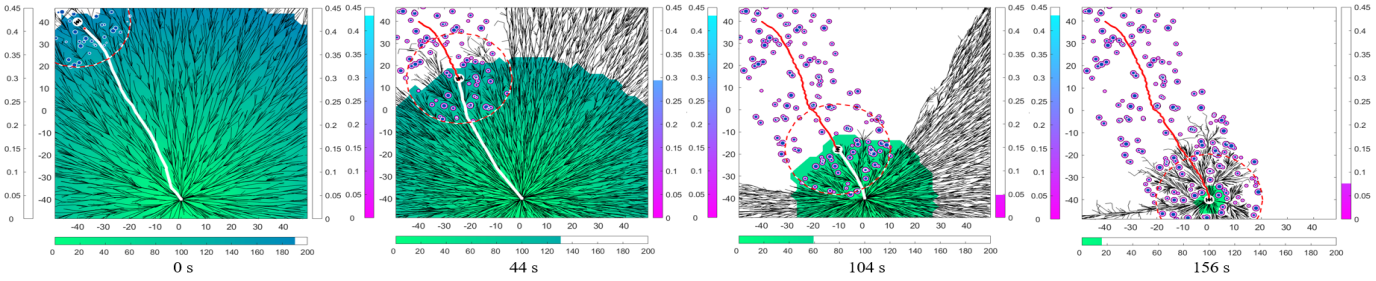

Fig. 3: **Collision-free navigation of an autonomous rover in a forest environment using RRT-Q$^{\mathrm{X}}_{\infty}$.** The rover has the start state $x_{\text{start}} = [-40, 40, 0, 0]$ and the goal state $x_{\text{goal}} = [0, -40, 0, 0]$. It can find the obstacles omnidirectionally within a radius of 20 represented by a red dashed circle. Its traversed path is shown by a red solid line and the RRT$^{\mathrm{X}}$ path by a white line. The black solid lines represent the search-tree of RRT$^{\mathrm{X}}$. The actual obstacles (trees) are denoted by blue polygons and the corresponding augmentation in magenta. The maximum kinodynamic distance $D^{\text{kin}}_{\text{rob}}$, the current kinodynamic distance D_{rob}, and the *cost-to-go* of RRT$^{\mathrm{X}}$ are shown in the left, right, and bottom colorbars, respectively.

$L_1 = 0.9\beta(\underline{\lambda}(M)/(2\overline{\lambda}(R)))^{1/2} = 2.5$. The initial values of $\hat{W}_{\text{c}}$, $\hat{W}_{\text{a}}$, $\hat{W}_{\text{d}}$ are randomly generated separately for each BVP. The admissible window for each game is $\rho = 0.2$, i.e., when $D(\bar{x}) \leq 0.2 D_0(\bar{x}_{0,k})$, the rover will proceed to the next BVP. In this scenario, we set the external disturbance as $d(t) = 0.3 \sin t$.

Remark 6: The larger α_{c}, α_{a}, and α_{d} are (as long as Theorem 3 is satisfied), the faster the proposed robust intermittent Q-learning controller will converge to the optimal controller (Section IV-C). Also, the larger the admissible window ρ is, the sooner the robot can proceed to the next BVP (Section V). Finally, the larger β is, the more frequently the control is updated (Theorem 3), and the smaller the maximum kinodynamic distance will be. □

The autonomous rover has access to state feedback and a disturbance observer. In addition, its perception range is limited, so only the obstacles within the perception range can be detected. Therefore, the obstacle space keeps enlarging while the rover navigates in the environment and perceives the newly "appearing" obstacles. RRT-Q$^{\mathrm{X}}_{\infty}$ measures the kinodynamic distance D_{rob} and computes the augmented obstacle space $\mathcal{X}^{\text{aug}}_{\text{obs}}$ at every time step. Whenever the current path intersects with $\mathcal{X}^{\text{aug}}_{\text{obs}}$, replanning is performed to compute a new collision-free and optimal path.

We implemented RRT-Q$^{\mathrm{X}}_{\infty}$ partly based on the software package from [52]. The effectiveness of RRT-Q$^{\mathrm{X}}_{\infty}$ is shown in a video demonstration[1] that contains four different navigation scenarios with random forests of obstacles. Motion snapshots of one trial are depicted in Figure 3. We conduct 30 Monte-Carlo trials with different random forest obstacle configurations and different start and goal states to remove the effect of random obstacle-free navigation. The distance between the rover and the nearest obstacle in 30 replications is presented in Figure 4. The results reveal that the rover can safely navigate with no collision through the unknown random forest environment with unpredictably appearing obstacles in all trials. Moreover, the proposed controller is robust to disturbances unlike [29]. Compared to [9], [25], the proposed controller operates in open loop most of the time and closes the loop only when the error gap threshold is achieved. Hence, the controller as well as the actuator inputs are not constantly

[1]https://youtu.be/iS_PzDmlpfs.

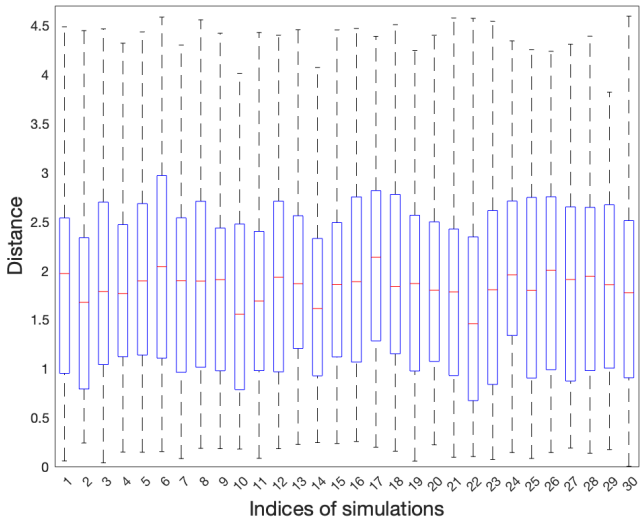

Fig. 4: **Distance to the nearest obstacle.** The distance between the robot and the nearest obstacle is positive in all 30 Monte-Carlo trials with the same navigation scenario, which shows the navigation is always collision-free.

updated. As a result, we transition from classical periodic learning to aperiodic learning that effectively balances the system performance over resources usage.

C. Maximum Kinodynamic Distance

The value of the maximum kinodynamic distance , $D^{\text{kin}}_{\text{rob}}$, of a single BVP depends on multiple parameters, including the initial velocity, orientation of BVP θ (i.e., the bearing angle of $x_{\text{r}} - x_0$), event-trigger bandwidth β, as well as disturbances. We investigate how θ and β are correlated with $D^{\text{kin}}_{\text{rob}}$ with zero initial velocity and no disturbances. We implement $2,880$ simulations of the rover (44) traversing from $[0, 0]$ to $[10\cos\theta, 10\sin\theta]$. The orientation of θ ranges from 0° to 355° with 72 values, and the event-trigger bandwidth β spans from 0 to 0.975 with 40 values. The other parameters are identical to the setting in Section VI-B. The colormap of $D^{\text{kin}}_{\text{rob}}$ is presented in Figure 5. We observe that: i) $D^{\text{kin}}_{\text{rob}}$ changes periodically with respect to θ; and ii) $D^{\text{kin}}_{\text{rob}}$ decreases for β close to 1. Particularly, as β increases, the control loop is more frequently closed, which indicates more frequent regulation to the agent's motion and subsequently smaller $D^{\text{kin}}_{\text{rob}}$.

TABLE I: Kinodynamic Motion Planning Comparison

	SST & SST* [10]	**Funnel Libraries [21]**	**Allen, Pavone [20]**	**RL-RRT [14]**	**RRT-Q* [9]**	**RRT-Q$^{\mathrm{X}}_\infty$**
Optimality	-	Energy	Appr. Time	No	Appr. Energy	Appr. Energy
Scalability	-	Yes (offline computation)	Yes (training)	Yes (training)	Yes	Yes
Feedback	-	Closed-loop	Open-loop	Open-loop	Closed-loop	Intermittent
Dynamics	Model-based	Model-based (uncertain)	Model-based	Model-based	Model-free	Model-free
Disturbance	-	Bounded	No	No	No	Bounded
Environment	Static	Dynamic	Dynamic	Static	Static	Dynamic
PE	-	-	-	-	Strong	Relaxed

D. Qualitative Comparison

In Table I, we present a qualitative comparison of the proposed framework RRT-Q$^{\mathrm{X}}_\infty$ with several other works on kinodynamic motion planning. We consider seven aspects: i) optimality in terms of control; ii) scalability in terms of real-time execution; iii) feedback loop of the controller; iv) requirement for system dynamics knowledge; v) type of bearable disturbance, vi) type of environment; and vii) probing signal requirement for PE.

Although SST and SST* [10] are near-optimal and optimal kinodynamic motion planners, the control-related specifications are not applicable due to lacking control synthesis. While the energy optimality of RRT-Q* [9] and RRT-Q$^{\mathrm{X}}_\infty$ is approximate due to the learning process with unknown system dynamics, the time optimality of Allen and Pavone [20] arises from the prediction error of the support vector machine. Real-time computation of the control input can be achieved by all methods except of SST and SST* [10]. Funnel libraries [21], Allen and Pavone [20], and RL-RRT [14] require extensive offline computations for training, but RRT-Q* and RRT-Q$^{\mathrm{X}}_\infty$ require no training *a priori*. While RRT-Q* can be used only in environments with static obstacles, RRT-Q$^{\mathrm{X}}_\infty$ can be applied in environments with appearing/disappearing obstacles. A strong PE of the probing signal is required in RRT-Q* for parameters to converge to the ideal values while a relaxed PE condition is needed for RRT-Q$^{\mathrm{X}}_\infty$.

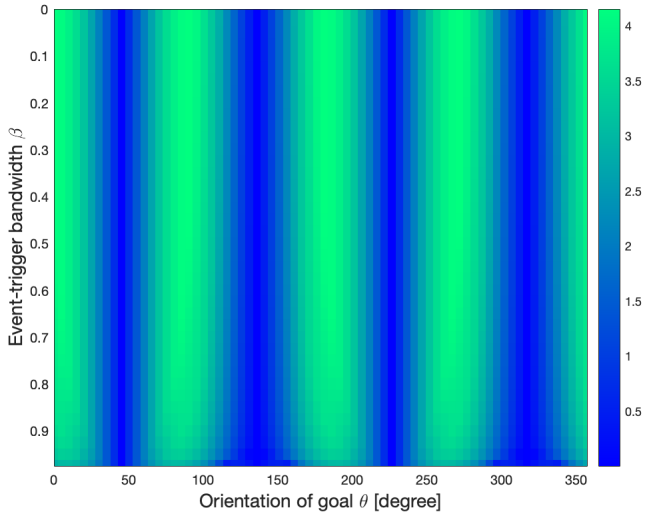

Fig. 5: **Evolution of maximum kinodynamic distance.** In $2,880$ problems over 72 values of BVP orientation θ and 40 event-trigger bandwidth values of β, $D^{\mathrm{kin}}_{\mathrm{rob}}$ evolves periodically with θ and decreases as β approaches 1.

VII. Conclusion

We propose RRT-Q$^{\mathrm{X}}_\infty$, an online kinodynamic motion planning framework for unpredictable dynamic environments with unknown robot dynamics. RRT-Q$^{\mathrm{X}}_\infty$ reduces computation and communication resources by intermittently updating the controller instead of continuously and is a fully online framework with no offline computation. In particular, we propose a robust intermittent Q-learning method for low-level model-free control. The results of kinodynamic motion planning in simulated random forest environments indicate that the proposed framework enables robust collision-free navigation for completely unknown systems in unknown dynamic environments with external disturbances.

Ongoing work is focusing on: i) adaptive obstacle augmentation strategies, ii) kinodynamic motion planning in human-crowded environments, and iii) reduced-order model-free control for large-scale unknown systems.

References

[1] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 995–1001.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[3] J. Bruce and M. M. Veloso, "Real-time randomized path planning for robot navigation," in *Robot Soccer World Cup*. Springer, 2002, pp. 288–295.

[4] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with RRTs," in *IEEE Internat. Conference on Robotics and Automation*, 2006, pp. 1243–1248.

[5] M. Otte and E. Frazzoli, "RRT$^{\mathrm{X}}$: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.

[6] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE Inter. Conference on Robotics and Automation*, 2013, pp. 5054–5061.

[7] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-Trees: Feedback motion planning via sums-of-squares verification," *The Int. J. of Rob. Res.*, vol. 29, no. 8, pp. 1038–1052, 2010.

[8] G. Goretkin, A. Perez, R. Platt, and G. Konidaris, "Optimal sampling-based planning for linear-quadratic kinodynamic systems," in *IEEE Internat. Conference on Robotics and Automation*, 2013, pp. 2429–2436.

[9] G. P. Kontoudis and K. G. Vamvoudakis, "Kinodynamic motion planning with continuous-time Q-learning: An online, model-free, and safe navigation framework," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3803–3817, 2019.

[10] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.

[11] R. Shome and L. E. Kavraki, "Asymptotically optimal kinodynamic planning using bundles of edges," in *2021 IEEE International Conference on Robotics and Automation*, 2021, pp. 9988–9994.

[12] J. D. Gammell and M. P. Strub, "Asymptotically optimal sampling-based motion planning methods," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 295–318, 2021.

[13] F. Atas, G. Cielniak, and L. Grimstad, "Benchmark of sampling-based optimizing planners for outdoor robot navigation," in *International Conference on Intelligent Autonomous Systems*, 2023, pp. 231–243.

[14] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.

[15] C. Verginis, D. V. Dimarogonas, and L. Kavraki, "Sampling-based motion planning for uncertain high-dimensional systems via adaptive control," in *Algor. Foundations of Robotics XIV*, 2021, pp. 159–175.

[16] P. Rousseas, C. Bechlioulis, and K. J. Kyriakopoulos, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2005–2011, 2021.

[17] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Tr. on Neur. Net. and Learn. Sys.*, vol. 29, no. 6, pp. 2042–2062, 2018.

[18] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *IEEE Intern. Conf. on Robotics and Automation*, 2018, pp. 1726–1731.

[19] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[20] R. E. Allen and M. Pavone, "A real-time framework for kinodynamic planning in dynamic environments with application to quadrotor obstacle avoidance," *Rob. and Aut. Sys.*, vol. 115, pp. 174–193, 2019.

[21] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[22] D. Zheng and P. Tsiotras, "Sampling-based kinodynamic motion planning using a neural network controller," in *AIAA Scitech 2021 Forum*, 2021, p. 1754.

[23] L. Li, Y. Miao, A. H. Qureshi, and M. C. Yip, "MPC-MPNet: Model-predictive motion planning networks for fast, near-optimal planning under kinodynamic constraints," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4496–4503, 2021.

[24] M. K. M. Jaffar and M. Otte, "PiP-X: Funnel-based online feedback motion planning/replanning in dynamic environments," in *Workshop on the Algorithmic Foundations of Robotics*, 2022, pp. 132–148.

[25] G. P. Kontoudis and K. G. Vamvoudakis, "Robust kinodynamic motion planning using model-free game-theoretic learning," in *American Control Conference*, 2019, pp. 273–278.

[26] A. Kanellopoulos and K. G. Vamvoudakis, "Non-equilibrium dynamic games and cyber-physical security: A cognitive hierarchy approach," *Systems & Control Letters*, vol. 125, pp. 59–66, 2019.

[27] K. G. Vamvoudakis and N.-M. T. Kokolakis, "Synchronous reinforcement learning-based control for cognitive autonomy," *Foundations and Trends® in Systems and Control*, vol. 8, no. 1–2, pp. 1–175, 2020.

[28] N.-M. T. Kokolakis, A. Kanellopoulos, and K. G. Vamvoudakis, "Bounded rationality in differential games: A reinforcement learning-based approach," in *Handbook of Reinforcement Learning and Control*. Springer, 2021, pp. 467–489.

[29] G. P. Kontoudis, Z. Xu, and K. G. Vamvoudakis, "Online, model-free motion planning in dynamic environments: An intermittent, finite horizon approach with continuous-time Q-learning," in *American Control Conference*, 2020, pp. 3873–3878.

[30] J. Netter, G. P. Kontoudis, and K. G. Vamvoudakis, "Bounded rational RRT-QX: Multi-agent motion planning in dynamic human-like environments using cognitive hierarchy and Q-learning," in *IEEE Conference on Decision and Control*, 2021, pp. 3597–3602.

[31] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Courier Corporation, 2012.

[32] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. John Wiley & Sons, 2012.

[33] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[34] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.

[35] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free λ-policy iteration for discrete-time linear quadratic regulation," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[36] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Tr. on Aut. Con.*, vol. 52, no. 9, pp. 1680–1685, 2007.

[37] K. G. Vamvoudakis and H. Ferraz, "Event-triggered H-infinity control for unknown continuous-time linear systems using Q-learning," in *IEEE Conference on Decision and Control*, 2016, pp. 1376–1381.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[39] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Inte. Journal of Adaptive Control and Signal Processing*, vol. 27, no. 4, pp. 280–301, 2013.

[40] K. G. Vamvoudakis, M. F. Miranda, and J. P. Hespanha, "Asymptotically stable adaptive–optimal control algorithm with saturating actuators and relaxed persistence of excitation," *IEEE Tr. on Neural Net. and Learning Sys.*, vol. 27, no. 11, pp. 2386–2398, 2016.

[41] Y. Yang, Y. Pan, C.-Z. Xu, and D. C. Wunsch, "Hamiltonian-driven adaptive dynamic programming with efficient experience replay," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[42] G. P. Kontoudis, K. G. Vamvoudakis, and Z. Xu, "RRT-QX real-time kinodynamic motion planning in dynamic environments with continuous-time reinforcement learning," in *Brain and Cognitive Intelligence Control in Robotics*. CRC Press, 2022, pp. 1–19.

[43] A. J. Van Der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, 1992.

[44] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control (revised edition)*. Routledge, 1975.

[45] T. Başar and P. Bernhard, *H_∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Springer, 2008.

[46] J. P. Hespanha, *Noncooperative Game Theory: An Introduction for Engineers and Computer Scientists*. Princeton University Press, 2017.

[47] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer, 1984, vol. 100.

[48] W. M. Haddad, V. Chellaboina, and S. G. Nersesov, *Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control*. Princeton University Press, 2014, vol. 49.

[49] J. P. Hespanha, D. Liberzon, and A. R. Teel, "Lyapunov conditions for input-to-state stability of impulsive systems," *Automatica*, vol. 44, no. 11, pp. 2735–2744, 2008.

[50] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*. Springer, 1997.

[51] J. Zhang, K. H. Johansson, J. Lygeros, and S. Sastry, "Zeno hybrid systems," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 5, pp. 435–451, 2001.

[52] M. Otte, "RRT-X (dynamic obstacles)," [Online]. Available: http://ottelab.com/html_stuff/code.html#RRTXcode, August 2023.

[53] K. G. Vamvoudakis, "Event-triggered optimal adaptive control algorithm for continuous-time nonlinear systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 3, pp. 282–293, 2014.

[54] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

APPENDIX

Proof of Theorem 2. The proof follows from [53, Theorem 1]. We adopt the positive definite function $\mathcal{V} := V^\star$ as a Lyapunov candidate that satisfies the time-triggered HJI equation (9) with $\mathcal{V}(0;\cdot) = 0$. Then, the orbital derivative along the solution of (1) with the intermittent controller for $t \in [r_j, r_{j+1})$ yields,

$$\dot{\mathcal{V}} = \frac{\partial V^{\star\intercal}}{\partial \bar{x}}\,\dot{\bar{x}} + \frac{\partial V^\star}{\partial t} = \frac{\partial V^{\star\intercal}}{\partial \bar{x}}\,(A\bar{x} + Bu_{\mathrm{d}}^\star + Fd^\star) + \frac{\partial V^\star}{\partial t}. \quad (45)$$

After writing the time-triggered HJI equation (9) as,

$$-\frac{1}{2}(\bar{x}^\intercal M\bar{x} + u_{\mathrm{c}}^{\star\intercal} R u_{\mathrm{c}}^\star - \gamma^2\|d^\star\|^2) \equiv \frac{\partial V^{\star\intercal}}{\partial \bar{x}}\,(A\bar{x} + Bu_{\mathrm{c}}^\star + Fd^\star) + \frac{\partial V^\star}{\partial t},$$

we rewrite the orbital derivative (45) as,

$$\begin{aligned}\dot{\mathcal{V}} &= -\frac{1}{2}\bar{x}^\intercal M\bar{x} - \frac{1}{2}u_{\mathrm{c}}^{\star\intercal} R u_{\mathrm{c}}^\star + \frac{1}{2}\gamma^2\|d^\star\|^2 + \frac{\partial V^{\star\intercal}}{\partial \bar{x}}\,B(u_{\mathrm{d}}^\star - u_{\mathrm{c}}^\star)\\ &= -\frac{1}{2}\bar{x}^\intercal M\bar{x} + \frac{1}{2}(u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star)^\intercal R(u_{\mathrm{c}}^\star - u_{\mathrm{d}}^\star) - \frac{1}{2}u_{\mathrm{d}}^{\star\intercal} R u_{\mathrm{d}}^\star + \frac{1}{2}\gamma^2\|d^\star\|^2.\end{aligned}$$

Per Fact 1, we have $(u_{\rm c}^\star - u_{\rm d}^\star)^\intercal R(u_{\rm c}^\star - u_{\rm d}^\star) \leq \overline{\lambda}(R)L(t)^2\|e\|^2$. Thus, the orbital derivative (45) is upper-bounded by,

$$\dot{\mathcal{V}} \leq -\frac{1}{2}\Big(\beta^2\underline{\lambda}(M) + (1-\beta^2)\underline{\lambda}(M)\Big)\|\bar{x}\|^2 + \frac{1}{2}\overline{\lambda}(R)L(t)^2\|e\|^2 - \frac{1}{2}\underline{\lambda}(R)\|u_{\rm d}^\star\|^2 + \frac{1}{2}\gamma^2\|d^\star\|^2. \tag{46}$$

Hence, if the following inequality is satisfied $\forall t \in [r_j, r_{j+1})$,

$$\|e\|^2 \leq \frac{(1-\beta^2)\underline{\lambda}(M)\|\bar{x}\|^2 + \underline{\lambda}(R)\|u_{\rm d}^\star\|^2 - \gamma^2\|d^\star\|^2}{\overline{\lambda}(R)L(t)^2}, \tag{47}$$

then the right-hand side of (46) will be less than or equal to 0, so $\dot{\mathcal{V}} \leq 0$, and the equilibrium point of the closed-loop system will be asymptotically stable as $T \to \infty$. ∎

Proof of Theorem 3. We start by considering the flow dynamics (39). We take the Lyapunov candidate as,

$$\mathcal{L}(\psi;t) := V^\star(\bar{x};t) + V^\star(\hat{\bar{x}};t) + \frac{1}{2}\|\tilde{W}_{\rm c}\|^2 + \frac{1}{2}{\rm tr}\{\tilde{W}_{\rm a}^\intercal\tilde{W}_{\rm a}\} + \frac{1}{2}{\rm tr}\{\tilde{W}_{\rm d}^\intercal\tilde{W}_{\rm d}\} > 0,\ \forall t \geq 0, \tag{48}$$

where ψ is the augmented state as defined in Section IV-D. The time derivative of (48) is partitioned to $\dot{\mathcal{L}} = T_1 + T_2 + T_3 + T_4 + T_5$ with,

$$T_1 = \frac{1}{2}\bar{x}^\intercal\dot{P}(t)\bar{x} + \bar{x}^\intercal P(t)(A\bar{x} + B\hat{u}_{\rm d} + F\hat{d}), \tag{49}$$

$$T_3 = \tilde{W}_{\rm c}^\intercal\dot{\tilde{W}}_{\rm c} = -\alpha_{\rm c}\tilde{W}_{\rm c}^\intercal\Big(\frac{\sigma\sigma^\intercal}{(1+\sigma^\intercal\sigma)^2} + \Lambda\Big)\tilde{W}_{\rm c}, \tag{50}$$

$$T_5 = {\rm tr}\{\tilde{W}_{\rm d}^\intercal\dot{\tilde{W}}_{\rm d}\} = -\alpha_{\rm d}{\rm tr}\{\tilde{W}_{\rm d}^\intercal\frac{(\xi(t)\bar{x})(\xi(t)\bar{x})^\intercal}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\tilde{W}_{\rm d} + \tilde{W}_{\rm d}^\intercal\frac{(\xi(t)\bar{x})\bar{x}^\intercal}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\tilde{Q}_{\rm xd}Q_{\rm dd}^{-1}\}, \tag{51}$$

and $T_2 = 0$, $T_4 = 0$. The terms T_2 and T_4 vanish because they are updated only at jumps and remain constant in the remaining time. We define $\Xi := \frac{\sigma\sigma^\intercal}{(1+\sigma^\intercal\sigma)^2} + \Lambda$ to upper bound T_3 (50), i.e., $T_3 \leq -\alpha_{\rm c}\underline{\lambda}(\Xi)\|\tilde{W}_{\rm c}\|^2$. Since $\frac{\sigma\sigma^\intercal}{(1+\sigma^\intercal\sigma)^2} \succeq 0$ and $\Lambda \succ 0$, we know $\underline{\lambda}(\Xi) > 0$. Thus, with $\alpha_{\rm c} \in \mathbb{R}^+$, the following inequality holds,

$$T_3 \leq -\alpha_{\rm c}\underline{\lambda}(\Xi)\|\tilde{W}_{\rm c}\|^2 \leq 0. \tag{52}$$

Next, the first term T_1 (49) yields,

$$T_1 \leq -\frac{1}{2}\underline{\lambda}(M)\|\bar{x}\|^2 + \frac{1}{2}\overline{\lambda}(R)\|\tilde{W}_{\rm a}^\intercal\mu(t)\bar{x} - \hat{W}_{\rm a}^\intercal\mu(t)e\|^2 - \frac{1}{2}\underline{\lambda}(R)\|\hat{u}_{\rm d}\|^2 - \frac{1}{2}\gamma^2\|\tilde{W}_{\rm d}^\intercal\xi(t)\bar{x}\|^2 + \frac{1}{2}\gamma^2\|\hat{d}\|^2, \tag{53}$$

where $\bar{x}^\intercal P(t)B = -u_{\rm c}^{\star\intercal}R$, and $\bar{x}^\intercal P(t)F = \gamma^2 d^{\star\intercal}$. Considering the actor weights remain constant during the flow dynamics, we have $L_1(t) \mapsto \mathbb{R}^+$ a strictly positive-definite function [47] such that $\|\tilde{W}_{\rm a}^\intercal\mu(t)\| \leq L_1(t)$. Thus, by adding and subtracting $\frac{1}{2}\beta^2\underline{\lambda}(M)\|\bar{x}\|^2$ to the right-hand side, and by using Fact 1, the Young's inequality, and (53), we obtain,

$$T_1 \leq -\frac{1}{2}\Big((1-\beta^2)\underline{\lambda}(M) - 2\overline{\lambda}(R)L_1(t)^2\Big)\|\bar{x}\|^2 - \frac{1}{2}\beta^2\underline{\lambda}(M)\|\bar{x}\|^2 + 2\overline{\lambda}(R)\Big(L(t)^2 + L_1(t)^2\Big)\|e\|^2 - \frac{1}{2}\underline{\lambda}(R)\|\hat{u}_{\rm d}\|^2 - \frac{1}{2}\gamma^2\|\tilde{W}_{\rm d}^\intercal\xi(t)\bar{x}\|^2 + \frac{1}{2}\gamma^2\|\hat{d}\|^2. \tag{54}$$

Provided that in the time interval $t \in [r_j, r_{j+1})$, the upper bound inequality (47) holds, then (54) yields,

$$T_1 \leq -\frac{1}{2}\Big(\beta^2\underline{\lambda}(M) - 2\overline{\lambda}(R)L_1(t)^2\Big)\|\bar{x}\|^2 - \frac{1}{2}\gamma^2\|\tilde{W}_{\rm d}^\intercal\xi(t)\bar{x}\|^2. \tag{55}$$

Now consider the last term T_5 (51). It yields,

$$T_5 \leq -\frac{\alpha_{\rm d}}{2}\frac{\|\bar{x}^\intercal\xi(t)^\intercal\tilde{W}_{\rm d}\|^2}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})} + \frac{\alpha_{\rm d}}{2}\eta\overline{\lambda}\Big(\frac{\xi(t)Q_{\rm dd}^{-1}}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\Big)\|\bar{x}\|^2, \tag{56}$$

where η is a constant of unity order. Given the upper bounds (52), (55) and (56), we have,

$$\dot{\mathcal{L}}(\psi;t) \leq -\frac{1}{2}\Big(\beta^2\underline{\lambda}(M) - 2\overline{\lambda}(R)L_1(t)^2 - \alpha_{\rm d}\eta\overline{\lambda}\Big(\frac{\xi(t)Q_{\rm dd}^{-1}}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\Big)\Big)\|\bar{x}\|^2 - \frac{1}{2}\gamma^2\|\tilde{W}_{\rm d}^\intercal\xi(t)\bar{x}\|^2 - \frac{\alpha_{\rm d}}{2}\frac{\|\tilde{W}_{\rm d}^\intercal\xi(t)\bar{x}\|^2}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})} - \alpha_{\rm c}\underline{\lambda}(\Xi)\|\tilde{W}_{\rm c}\|^2.$$

Thus, if the following inequalities are satisfied,

$$0 < \alpha_{\rm d} \leq \frac{\beta^2\underline{\lambda}(M) - 2\overline{\lambda}(R)L_1(t)^2}{\eta\overline{\lambda}\Big(\frac{\xi(t)Q_{\rm dd}^{-1}}{1+(\xi(t)\bar{x})^\intercal(\xi(t)\bar{x})}\Big)},\quad \frac{\underline{\lambda}(M)}{\overline{\lambda}(R)} > \frac{2L_1(t)^2}{\beta^2},$$

then $\dot{\mathcal{L}}(\psi;t)$ is non-positive for all ψ and $t \geq t_0$. By defining $W_1(\psi;t) = W_2(\psi;t) := V^\star(\bar{x};t) + \frac{1}{2}\|\tilde{W}_{\rm c}\|^2 + \frac{1}{2}{\rm tr}\{\tilde{W}_{\rm d}^\intercal\tilde{W}_{\rm d}\} > 0$, $W_1(\psi;t) \leq \mathcal{L}(\psi;t) \leq W_2(\psi;t)$ is satisfied. Hence, the origin $\psi_{\rm e} = 0$ is uniformly stable per the Lyapunov stability theorem. Also, given that $\mathcal{L}(\psi;t)$ is lower-bounded, non-increasing, and that its time derivative $\dot{\mathcal{L}}(\psi;t) = T_1 + T_3 + T_5$ is also bounded, the Lyapunov function in (48) is uniformly continuous, which satisfies the Barbalat's lemma, $\mathcal{L}(\psi;t) \to 0$ as $t \to \infty$. Since $\dot{\mathcal{L}}(\psi;t) < 0$, asymptotic stability holds from the Lyapunov stability theorem. Moreover, $W_1(\psi;t)$ is radially unbounded with respect to $\|\bar{x}\|$, $\|\tilde{W}_{\rm c}\|$, $\|\tilde{W}_{\rm d}^\intercal\xi(t)\bar{x}\|^2$, and thus globally properties hold. Therefore, the equilibrium point of the flow dynamics at the origin $\psi_{\rm e} = 0$ is globally uniformly asymptotically stable as $T \to \infty$ [54].

We continue with the jump dynamics (40) comprising the sampled states and the control actor weight updates. Let us define the Lyapunov function at $t = r_j$ as,

$$\Delta\mathcal{L}(\psi;t) := \Delta V_1(\bar{x}^+, \bar{x}(r_j);t) + \underbrace{\Big(V^\star(\hat{\bar{x}}^+;t) - V^\star(\hat{\bar{x}}(r_j);t)\Big)}_{\Delta V_2} + \Delta V_3(\tilde{W}_{\rm c}^+, \tilde{W}_{\rm c}(r_j)) + \Delta V_5(\tilde{W}_{\rm d}^+, \tilde{W}_{\rm d}(r_j)) + \underbrace{\frac{1}{2\alpha_{\rm a}}\Big({\rm tr}\{\tilde{W}_{\rm a}^{+\intercal}\tilde{W}_{\rm a}^+\} - {\rm tr}\{\tilde{W}_{\rm a}(r_j)^\intercal\tilde{W}_{\rm a}(r_j)\}\Big)}_{\Delta V_4}, \tag{57}$$

where $\Delta\mathcal{L}(\psi;t) > 0$. Note that $\bar{x}$, $\tilde{W}_{\rm c}$, $\tilde{W}_{\rm d}$ are continuous even at the triggering events, hence they all vanish during the jump dynamics, i.e., $\Delta V_1(\bar{x}^+, \bar{x}(r_j);t) = 0$, $\Delta V_3(\tilde{W}_{\rm c}^+, \tilde{W}_{\rm c}(r_j)) = 0$, and $\Delta V_5(\tilde{W}_{\rm d}^+, \tilde{W}_{\rm d}(r_j)) = 0$. With $\hat{\bar{x}}^+ = \bar{x}(r_j)$ and $\hat{\bar{x}}(r_j) = \bar{x}(r_{j-1})$ at the jump, the second term of (57) yields,

$$\Delta V_2 = V^\star(\hat{\bar{x}}^+) - V^\star(\hat{\bar{x}}(r_j)) = V^\star(\bar{x}(r_j)) - V^\star(\bar{x}(r_{j-1})) \leq 0,$$

because of the convergence of $\bar{x}$. Hence, as $T \to \infty$, the sampled state asymptotically converges to the origin, i.e.,

$\|\hat{\tilde{x}}(r_j)\| \to 0$.

Using $Q_{\mathrm{u_d u_d}} = R$ and Young's inequality, taking entry-wise norms for the fourth term of (57) results in,

$$\Delta V_4 \leq -\|\tilde{W}_{\mathrm{a}}^{\intercal}\mu(t)\bar{x}(r_j)\|^2\Big(1 - \frac{1+\alpha_{\mathrm{a}}}{4\overline{\lambda}(R)} - \frac{\alpha_{\mathrm{a}}}{8}\Big) + \frac{1+\alpha_{\mathrm{a}}}{4\overline{\lambda}(R)}\|\tilde{Q}_{\mathrm{xu_d}}\|^2 + \frac{\alpha_{\mathrm{a}}}{2\overline{\lambda}(R)^2}\|\tilde{Q}_{\mathrm{xu_d}}\|^2. \tag{58}$$

Therefore, the inequality $\Delta V_4 < 0$ is satisfied if we guarantee in (58) that $\tilde{W}_{\mathrm{a}}$ remains outside the compact set $\Omega = \left\{\tilde{W}_{\mathrm{a}} \in \mathbb{R}^{n\times m} \mid \|\tilde{W}_{\mathrm{a}}\| \leq \sqrt{\frac{\frac{1+\alpha_{\mathrm{a}}}{4\overline{\lambda}(R)}\|\tilde{Q}_{\mathrm{xu_d}}\|^2 + \frac{\alpha_{\mathrm{a}}}{2\overline{\lambda}(R)^2}\|\tilde{Q}_{\mathrm{xu_d}}\|^2}{1-\frac{1+\alpha_{\mathrm{a}}}{4\overline{\lambda}(R)}-\frac{\alpha_{\mathrm{a}}}{8}}}\right\}$. The last two constraints arise from the denominator,

$$0 < \alpha_{\mathrm{a}} < \frac{8\overline{\lambda}(R) - 2}{\overline{\lambda}(R) + 2}, \quad \overline{\lambda}(R) > 0.25.$$

Since the elements in Ω are asymptotically stable for $T \to \infty$, the set becomes a single point, thus $\|\tilde{W}_{\mathrm{a}}\| \to 0$. ■