

Online, Model-Free Motion Planning in Dynamic Environments: An Intermittent, Finite Horizon Approach with Continuous-Time Q-Learning

George P. Kontoudis¹, Zirui Xu², and Kyriakos G. Vamvoudakis³

Abstract—This paper presents an online kinodynamic motion planning scheme for dynamically evolving environments, by employing Q-learning. The methodology addresses the finite horizon continuous-time optimal control problem with completely unknown system dynamics. An actor-critic structure is employed along with a buffer of previous experiences, to approximate the optimal policy and alleviate the learning signal requirements. The methodology is equipped with a terminal state evaluation to achieve fast navigation. The path planning is assigned to the RRT^X. An obstacle augmentation and a local replanning strategy are responsible for collision-free navigation. Rigorous Lyapunov-based proofs are provided to guarantee closed-loop stability of the equilibrium point. We evaluate the efficacy of the methodology with simulations.

I. INTRODUCTION

Artificial intelligence has enabled tremendous opportunities on developing autonomous systems. Robotic motion planning has played a key role in autonomy and robotics. In practice, the *kinodynamic* nature of the autonomous systems imposes more constraints, especially for *real-time* implementation. Moreover, the environment may dynamically evolve. In particular, *unpredictable dynamic environments* may include moving obstacles, other autonomous systems, and/or even human motions. Although optimal motion planning is desired in autonomous systems, it requires extensive offline computations that make the problem infeasible. A mandatory element for the optimal control computation is the model of the system, which is always challenging to derive. Furthermore, the robots are often equipped with limited energy resources, while on-board data processing and communication networks consume a large amount of energy, without being always necessary. Our focus in this work is on providing an *online* kinodynamic motion planning strategy for continuous-time linear systems that operate in unpredictable dynamic environments, by using *deep intermittent Q-learning*.

Related Work: A sampling-based path planning algorithm, namely rapidly-exploring random tree (RRT), is presented in [1]. This methodology is proved to be probabilistically complete, yet not optimal. A variation of this methodology with rewiring of the search tree, the RRT*, is proposed in [2]. RRT* is a probabilistically complete and asymptotically

optimal path planning algorithm for static environments. Recently, another sampling-based algorithm is introduced in [3], namely RRT^X. The latter achieves asymptotically optimal, sampling-based motion planning and replanning in dynamic environments, yet requires the dynamics of the system. The problem of kinodynamic motion planning is introduced in [4]. The authors approximated a near-optimal solution in static environments, but this algorithm requires the system dynamics and excessive offline computations. Optimal control techniques are also employed in [5] and the optimality of paths between pairs of states is guaranteed for controllable linear systems. This approach operates in an open-loop fashion and incorporates the dynamics of the system. Randomized kinodynamic motion planning in dynamic environments is presented in [6], where a space of admissible control functions is obtained under kinodynamic constraints. In [7], the authors proposed an online, machine learning-based, kinodynamic motion planning algorithm which was experimentally validated in dynamic indoor environments. The technique requires perfect knowledge of the system, continuous communication network, and its feasibility depends on the offline training of reachability sets.

In realistic systems with limited bandwidth communication, *intermittent control* is proved to operate optimally [8]. A user-defined, intermittent triggering condition is responsible for closing the loop whenever is necessary. This is justified by an equilibrium point stability criterion. A *model-free*, intermittent control algorithm for continuous-time linear systems with infinite horizon performance is discussed in [9].

Experience replay mechanisms are employed in [10] to reduce non-stationarities and stabilize the reinforcement learning algorithm. This approach stores data from previous experiences to alleviate the update process. A similar constraint appears in adaptive control techniques [11], where the learning signal must be *persistently exciting* (PE) to converge to ideal parameters. That is a conservative condition, which is usually difficult to accomplish in realistic systems. In [12], [13] the authors employ concurrently previous experiences and current data to relax the PE condition.

Connection between adaptive control [11] and optimal control [14] can be established by employing the principles of reinforcement learning [15], [16]. In [17], a Q-learning approach for solving the model-free, infinite horizon optimal control problem for continuous-time linear systems is presented. In [18], we proposed the RRT-Q*, a model-free kinodynamic motion planning algorithm which guarantees optimal and online navigation in static environments with a *finite horizon* performance. In this work, the latter methodol-

¹G. P. Kontoudis is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA (email: gpkont@vt.edu).

²Z. Xu is with the School of Electrical and Computer Engineering, Georgia Tech, USA (email: zirui.xu@gatech.edu).

³K. G. Vamvoudakis is with the Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, USA (email: kyriakos@gatech.edu).

This work was supported, in part, by ONR Minerva under grant No. N00014-18-1-2160 in part by an NSF CAREER under grant Nos. CPS-1750789 and S&AS-1849198, and NATO under grant No. SPS G5176.

ogy is extended to unpredictable dynamic environments with a controller that operates intermittently.

Contributions: The contribution of this paper is threefold. First, we formulate an intermittent Q-learning methodology to solve the finite horizon optimal control problem for continuous-time linear systems, without any information of the system dynamics. Next, a stability proof is provided, based on rigorous Lyapunov-based analysis. Finally, we incorporate the Q-learning scheme into a path planning algorithm to perform online navigation in dynamic environments.

II. PROBLEM FORMULATION

Consider a continuous-time linear time-invariant system,

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

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, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are the unknown/uncertain plant and input matrices respectively. We seek to drive the robot from an initial state x_0 to a final state x_r . Thus, we define the difference between the state $x(t)$ and the desired state x_r , which yields,

$$\dot{\bar{x}}(t) = A\bar{x}(t) + Bu(t), \quad \bar{x}_0, \quad t \geq 0. \quad (1)$$

To conserve resources and sensor computational efforts, a sampled state is provided by,

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

where $\bar{x}(r_j)$ is the state of the flow dynamics, $\bar{x}(r_{j+1})$ is the state of the jump dynamics, and r_j a monotonically increasing sequence of samples $\{r_j\}_{j=1}^{N_j}$ with $T = r_{N_j}$ the finite horizon. The decision of intermittent control at time $t = r_{j+1}$ is based on the error gap system that is given by,

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

A finite horizon cost functional is selected to drive the system from an initial state to a desired state,

$$J(\bar{x}_0; u_d; t_0, T) = \phi(T) + \frac{1}{2} \int_{t_0}^T \bar{x}^\top M \bar{x} + u_d^\top R u_d \, d\tau, \quad (3)$$

where $u_d(\hat{\bar{x}}; t)$ is the control input with reduced updates based on the intermittent rule, $\phi(T) := 1/2 \bar{x}^\top(T) P(T) \bar{x}(T)$ is the terminal cost with a symmetric, positive-definite final Riccati matrix $P(T) \in \mathbb{R}^{n \times n} \succ 0$, and $M \in \mathbb{R}^{n \times n} \succeq 0$, $R \in \mathbb{R}^{m \times m} \succ 0$ are user-defined matrices that penalize the state and the control input respectively. Our objective is to obtain the optimal control $u_d^*(\hat{\bar{x}}; t)$ such that $J(\bar{x}_0; u_d^*; t_0, T) \leq J(\bar{x}_0; u_d; t_0, T)$ is satisfied for all \bar{x}, u . Hence, we formulate the minimization problem $J(\bar{x}_0; u_d^*; t_0, T) = \min_{u_d} J(\bar{x}_0; u_d; t_0, T)$ subject to (1). To this end, a time-triggered control input $u_c(\bar{x}; t)$ is used to approximate the optimal, time-triggered value function $V^*(\bar{x}; t_0, T)$ considering intermittent updates of the control $u_d(\hat{\bar{x}}; t)$, yet without any information of the system dynamics. That is defined by,

$$V^*(\bar{x}; t_0, T) := \min_{u_c} \left\{ \phi(T) + \frac{1}{2} \int_{t_0}^T \bar{x}^\top M \bar{x} + u_c^\top R u_c \, d\tau \right\}. \quad (4)$$

Assumption 1: The unknown pair (A, B) is controllable and the unknown pair $(M^{1/2}, A)$ is detectable. \square

Let the known closed obstacle space be $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$. For multiple obstacles, the obstacle space is defined by $\mathcal{X}_{\text{obs}} := \bigcup_{l=1}^{N_l} \mathcal{X}_{\text{obs},l}$, where $N_l \in \mathbb{N}$ is the total number of obstacles. Then, the free space is computed by $\mathcal{X}_{\text{free}} = (\mathcal{X}_{\text{obs}})^c = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$. In dynamic environments, the obstacle space and the free space may propagate in time. Thus, the unpredictable variation of the obstacle space is defined by $\Delta \mathcal{X}_{\text{obs}} := f(\mathcal{X}_{\text{obs}}; t)$ where $f(\cdot)$ is unknown.

The output of the RRT^X provides a time-varying path $\pi(x_{0,i}, x_{r,i}; t) \in \mathbb{R}^{2(k \times 2)}$, for $i = 1, \dots, N_i$ and $N_i \in \mathbb{N}$. In particular, the path π is a collection of sets of initial states x_0 and final states x_r , which may vary in time t , as the obstacle space dynamically evolves in time. The algorithm constructs a graph $\mathcal{G} = (V, E)$, where V is the set of nodes and E the set of edges. Furthermore, the augmented obstacle space is defined by $\mathcal{X}_{\text{obs}}^{\text{aug}} := g(\mathcal{X}_{\text{obs}}; D_{\text{rob}}; t)$, where D_{rob} is the kinodynamic distance [18]. Since we address the finite horizon optimal control problem with free final state, the system will approximate the final state, i.e. converge to the final state [14]. To reduce the motion planning time, the final state is assumed to be reached if the system is around a neighborhood of the final state. Let us define the distance between the initial state x_0 and the desired state x_r as,

$$D_0(\bar{x}_0) := \|\bar{x}_0\|_n, \quad \forall \bar{x}_0 \in \mathbb{R}^n. \quad (5)$$

During the navigation, we measure the relative distance as,

$$D(\bar{x}) := \|\bar{x}\|_n, \quad \forall \bar{x} \in \mathbb{R}^n. \quad (6)$$

The distance error is defined $e_d(\bar{x}_0, \bar{x}) := |D_0(\bar{x}_0) - D(\bar{x})|$.

III. INTERMITTENT OPTIMAL CONTROL PROBLEM

The time-triggered Hamiltonian for the finite horizon optimal control problem with respect to (1) and (4) is given by,

$$\mathcal{H}(\bar{x}; u_c; \frac{\partial V^*}{\partial t}, \frac{\partial V^*}{\partial \bar{x}}) = \frac{\partial V^{*\top}}{\partial \bar{x}} (A\bar{x} + Bu_c) + \frac{\partial V^*}{\partial t} + \frac{1}{2} (\bar{x}^\top M \bar{x} + u_c^\top R u_c), \quad (7)$$

Since system (1) is linear, the optimal value function results in $V^*(\bar{x}; t) = 1/2 \bar{x}^\top P(t) \bar{x}$, where $P(t) \in \mathbb{R}^{n \times n} \succ 0$ is the symmetric positive-definite Riccati matrix computed by,

$$-\dot{P}(t) = P(t)A + A^\top P(t) + M - P(t)BR^{-1}B^\top P(t). \quad (8)$$

Therefore, the optimal control yields,

$$u_c^*(\bar{x}; t) = -R^{-1}B^\top P(t)\bar{x}, \quad \forall \bar{x}, t. \quad (9)$$

Next, a sampled version of the controller (9) is utilized to reduce the communication between system and controller. The optimal intermittent controller is defined by,

$$u_d^*(\hat{\bar{x}}; t) := -R^{-1}B^\top P(t)\hat{\bar{x}}, \quad \forall \hat{\bar{x}}, t. \quad (10)$$

Corollary 1: Since the system (1) is linear and both controllers in (9), (10) are linear mappings of their state $u_c^*(\bar{x}; t) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $u_d^*(\hat{\bar{x}}; t) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, the following

inequality is derived,

$$\begin{aligned} \|u_c^* - u_d^*\| &\leq \|R^{-1}B^\top P(t)\| \|(\bar{x} - \hat{x})\| \\ &\leq L(t)\|e\|, \end{aligned}$$

where $L(t) \mapsto \mathbb{R}^+$ is a strictly positive function. \square

Lemma 1: The intermittent Hamiltonian is defined by,

$$\begin{aligned} \mathcal{H}(\bar{x}; u_d^*, \frac{\partial V^*}{\partial t}, \frac{\partial V^*}{\partial \bar{x}}) &=: \frac{\partial V^*}{\partial \bar{x}} (A\bar{x} - BR^{-1}B^\top P(t)\hat{x}) + \frac{\partial V^*}{\partial t} \\ &\quad + \frac{1}{2}(\bar{x}^\top M\bar{x} + \hat{x}^\top P(t)BR^{-1}B^\top P(t)\hat{x}), \end{aligned} \quad (11)$$

and satisfies the inequality,

$$\left\| \mathcal{H}(\bar{x}; u_d^*, \frac{\partial V^*}{\partial t}, \frac{\partial V^*}{\partial \bar{x}}) \right\| \leq \frac{\bar{\lambda}(R)}{2} L(t)^2 \|e\|^2. \quad (12)$$

Proof. Consider the difference between (11) and (7). Substitute the optimal control (9) and the optimal intermittent control (10). By using the optimal value function and the differential Riccati equation (8) we obtain,

$$\mathcal{H}(\bar{x}; u_d^*, \frac{\partial V^*}{\partial t}, \frac{\partial V^*}{\partial \bar{x}}) = \frac{1}{2}(u_c^* - u_d^*)^\top R(u_c^* - u_d^*). \quad (13)$$

Then use Corollary 1 to result in inequality (12). \blacksquare

Theorem 1: Let a positive-definite radially unbounded function be $V(\bar{x}; t) = 1/2 \bar{x}^\top P(t)\bar{x}$ with P provided by (8). If the following error inequality is satisfied,

$$\|e\|^2 \leq \frac{(1 - \beta^2)\lambda(M)}{L(t)^2\bar{\lambda}(R)} \|\bar{x}\|^2 + \frac{\lambda(R)}{L(t)^2\bar{\lambda}(R)} \|u_d^*\|^2,$$

where $\beta \in (0, 1)$ is a user-defined scalar characterizing the available bandwidth, then the equilibrium point of the closed-loop system (1) with intermittent control (10) is asymptotically stable for all $t \in [r_j, r_{j+1})$.

Proof. The proof follows from [19]-(Theorem 1). \blacksquare

IV. DEEP INTERMITTENT Q-LEARNING

Let us define the advantage Q-function as,

$$\begin{aligned} \mathcal{Q}(\bar{x}; u_d, u_c; t) &:= V^*(\bar{x}) + \mathcal{H}(\bar{x}; u_d; \frac{\partial V^*}{\partial t}, \frac{\partial V^*}{\partial \bar{x}}) \\ &\quad - \mathcal{H}(\bar{x}; u_c; \frac{\partial V^*}{\partial t}, \frac{\partial V^*}{\partial \bar{x}}), \end{aligned} \quad (14)$$

where the Hamiltonian associated with the optimal control vanishes $\mathcal{H}(\bar{x}; u_c^*; \partial V^*/\partial t, \partial V^*/\partial \bar{x}) = 0$ and $\mathcal{Q}(\bar{x}; u_d; t) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^+$ is an action-dependent value.

Lemma 2: The minimization problem $\mathcal{Q}^*(\bar{x}; u_d^*; t) := \min_{u_d} \mathcal{Q}(\bar{x}; u_d; t)$ given $V^*(\bar{x}; t)$ yields,

$$\mathcal{Q}^*(\bar{x}; u_d^*; t) = V^*(\bar{x}; t) + \frac{1}{2}(u_c^* - u_d^*)^\top R(u_c^* - u_d^*). \quad (15)$$

Proof. Substitute (13) to the Q-function (14). Since $\mathcal{H}(\bar{x}; u_c^*; \partial V^*/\partial t, \partial V^*/\partial \bar{x}) = 0$, the result follows. \blacksquare

The augmented state is defined by $U := [\bar{x}^\top \ u_d^\top]^\top$. Thus, the Q-function (14) results in a compact quadratic form,

$$\begin{aligned} \mathcal{Q}(\bar{x}; u_d; t) &= \frac{1}{2}U^\top \begin{bmatrix} Q_{xx}(t) & Q_{xu_d}(t) \\ Q_{u_d x}(t) & Q_{u_d u_d}(t) \end{bmatrix} U =: \frac{1}{2}U^\top \bar{Q}(t)U \\ &= \frac{1}{2} \text{vech}(\bar{Q}(t))^\top (U \otimes U), \end{aligned}$$

where $Q_{xx}(t) = \dot{P}(t) + P(t) + M + P(t)A + A^\top P(t)$, $Q_{xu_d}(t) = Q_{u_d x}^\top(t) = P(t)B$, and $Q_{u_d u_d} = R$. By

solving the stationarity condition $\partial \mathcal{Q}(\bar{x}; u_d; t)/\partial u_d = 0$ and employing the elements of the compact \bar{Q} , we formulate a model-free optimal sampled controller, $u_d^*(\bar{x}; t) = \arg \min_{u_d} \mathcal{Q}(\bar{x}; u_d; t) = -Q_{u_d u_d}^{-1} Q_{u_d x}(t)\bar{x}$.

A. Actor/Critic Structure

We shall use a critic augmented with previous experiences to approximate the Q-function (14), and an actor to approximate the intermittent control policy (10). Let us define $\nu(t)^\top W_c := 1/2 \text{vech}(\bar{Q}(t))$, where $\nu(t)$ is a bounded basis function of proper dimensions that depends explicitly on time $t \geq 0$. Since the weight parameters are unknown, adaptive control techniques are used [11] to find tuning laws of the current weight values. Hence, the estimated Q-function yields,

$$\hat{Q}(\bar{x}; u_d; t) = \hat{W}_c^\top \nu(t)(U \otimes U). \quad (16)$$

Let us also define the actor that approximates the intermittent control policy by $W_a^\top \mu(t) := -Q_{u_d u_d}^{-1} Q_{u_d x} \in \mathbb{R}^{n \times m}$, where $\mu(t)$ is a bounded basis function, depending explicitly on time $t \geq 0$. Thus, the estimated intermittent control policy takes the form of,

$$\hat{u}_d(\bar{x}; t) = \hat{W}_a^\top \mu(t)\bar{x}. \quad (17)$$

Using the integral Bellman equation [16] and Lemma 2,

$$\begin{aligned} \mathcal{Q}^*(\bar{x}(t); \hat{u}_d^*(t); t) &= \mathcal{Q}^*(\bar{x}(t - \Delta t); \hat{u}_d^*(t - \Delta t); t - \Delta t) \\ &\quad - \frac{1}{2} \int_{t - \Delta t}^t (\bar{x}^\top M\bar{x} + \hat{u}_d^{*\top} R \hat{u}_d^*) d\tau, \\ \mathcal{Q}^*(\bar{x}(T), T) &= \frac{1}{2} \bar{x}^\top(T)P(T)\bar{x}(T). \end{aligned}$$

B. Relaxed Persistence of Excitation

The next step is to develop a learning framework for the estimation of \hat{Q} and \hat{u}_d . To this end, one needs to ensure that the signal is persistently exciting (PE).

Definition 1: A signal vector $\Delta(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ is said to be persistently exciting (PE) over the interval $[t, t + T_{PE})$, where $T_{PE} \in \mathbb{R}^+$, if there exists a strictly positive constant $\gamma \in \mathbb{R}^+$ such that $\gamma I \leq \int_t^{t+T_{PE}} \Delta(\tau)\Delta(\tau)^\top d\tau$, where I is an identity matrix of appropriate dimensions. \square

Corollary 2: If the signal vector $\Delta(t)$ is PE, then the unknown vectors converge exponentially fast $(\hat{W}_c^\top \nu(t)) \rightarrow (W_c^\top \nu)^*$ and $(\hat{W}_a^\top \mu(t)) \rightarrow (W_a^\top \mu)^*$.

Proof. The proof follows from [11]-(Corollary 4.3.1). \blacksquare

Unambiguously, the PE condition is demanding and usually is not satisfied in real systems. Thus, we employ [12] to relax the PE condition. The latter utilizes past recorded data concurrently with current data for learning adaptation. Equivalently, deep learning employs an experience replay mechanism [10], thus we name the proposed technique: *deep intermittent Q-learning*. In the intermittent Hamiltonian (7) the unknown elements are the value function and the dynamics of the system. Past and current data are used to

obtain, $\dot{\hat{x}} \approx (\hat{x}(t) - \hat{x}(t - \Delta t))/\Delta t$, where $\Delta t \in \mathbb{R}^+$ is a small time resolution. Using Lemma 2 and (16) yields,

$$\frac{\partial V^*}{\partial \bar{x}} \approx \frac{\partial \hat{Q}}{\partial \bar{x}} = \hat{W}_c^\top \nu(t) \nabla_{\bar{x}}(U \otimes U), \quad (18)$$

$$\frac{\partial V^*}{\partial t} \approx \frac{\partial \hat{Q}}{\partial t} = \hat{W}_c^\top \nabla_t (\nu(t)(U \otimes U)). \quad (19)$$

Hence, the intermittent Hamiltonian approximation (7), (11) by using (18), (19) yields,

$$\begin{aligned} \hat{\mathcal{H}}(U; \hat{W}_c; t) &= \frac{1}{2} \text{vech} \left(\begin{bmatrix} M & 0 \\ 0 & R \end{bmatrix} \right)^\top (U \otimes U) \\ &+ \hat{W}_c^\top \left(\nabla_t (\nu(t)(U \otimes U)) + \nu(t) \nabla_{\bar{x}}(U \otimes U) \right) \hat{x}. \end{aligned} \quad (20)$$

C. Learning Framework

Let us define the critic estimation error $e_c \in \mathbb{R}$ as,

$$\begin{aligned} e_{c1}(t) &:= \hat{Q}(\bar{x}(t); \hat{u}_d(t); t) \\ &- \hat{Q}(\bar{x}(t - \Delta t); \hat{u}_d(t - \Delta t); t - \Delta t) \\ &+ \frac{1}{2} \int_{t-\Delta t}^t (\bar{x}(\tau)^\top M \bar{x}(\tau) + \hat{u}_d(\tau)^\top R \hat{u}_d(\tau)) d\tau \\ &= \hat{W}_c \nu(t)^\top (U(t) \otimes U(t) - U(t - \Delta t) \otimes U(t - \Delta t)) \\ &+ \frac{1}{2} \int_{t-\Delta t}^t (\bar{x}(\tau)^\top M \bar{x}(\tau) + \hat{u}_d(\tau)^\top R \hat{u}_d(\tau)) d\tau, \end{aligned}$$

and the second critic error as $e_{c2}(t, T) := 1/2 \bar{x}^\top(T) P(T) \bar{x}(T) - \hat{W}_c(t)^\top \nu(t)(U(T) \otimes U(T))$. Next, the buffer critic error is defined by,

$$\begin{aligned} e_{\text{buff},k}(t, t_k) &:= \hat{\mathcal{H}}(U(t_k); \hat{W}_c; t_k) \\ &- \frac{1}{2} (\hat{u}_c(t_k) - \hat{u}_d(t_k))^\top R (\hat{u}_c(t_k) - \hat{u}_d(t_k)), \end{aligned}$$

which by using (20) results in,

$$\begin{aligned} e_{\text{buff},k}(t, t_k) &= \frac{1}{2} \text{vech} \left(\begin{bmatrix} M & 0 \\ 0 & R \end{bmatrix} \right)^\top (U(t_k) \otimes U(t_k)) \\ &+ \hat{W}_c^\top \left(\nu(t_k) \nabla_{\bar{x}}(U(t_k) \otimes U(t_k)) \hat{x} + \nabla_t (\nu(t_k)(U(t_k) \otimes U(t_k))) \right) \\ &- \frac{1}{2} (\hat{u}_c(t_k) - \hat{u}_d(t_k))^\top R (\hat{u}_c(t_k) - \hat{u}_d(t_k)). \end{aligned}$$

The actor approximator error is provided by $e_a(t, r_{j+1}) := \hat{W}_a^\top \mu(t) \bar{x}(r_{j+1}) + \hat{Q}_{\text{u}_d \text{u}_d}^{-1} \hat{Q}_{\text{u}_d \text{x}}(t) \bar{x}(r_{j+1})$, where $e_a \in \mathbb{R}^m$. Our objective is to drive the errors to zero by tuning the parameters of the critic (16) and the actor (17). Thus, the squared-norm of errors are constructed as,

$$K_1(\hat{W}_c) = \frac{1}{2} \|e_{c1}\|^2 + \frac{1}{2} \|e_{c2}\|^2 + \frac{1}{2} \sum_{k=1}^{N_k} \|e_{\text{buff},k}\|^2, \quad (21)$$

$$K_2(\hat{W}_a) = \frac{1}{2} \|e_a\|^2. \quad (22)$$

where $k = 1, \dots, N_k$, $N_k \in \mathbb{N}$ is number of data recordings for the relaxed PE. The learning framework consists of two tuning laws. A normalized gradient descent technique [11] is applied in (21) for the critic estimation weights,

$$\begin{aligned} \dot{\hat{W}}_c &= -\alpha_c \frac{\partial K_1}{\partial \hat{W}_c} = -\alpha_c \left(\frac{\sigma(t) e_{c1}}{(1 + \sigma(t)^\top \sigma(t))^2} \right. \\ &\left. + \frac{\sigma_f e_{c2}}{(1 + \sigma_f^\top \sigma_f)^2} + \sum_{k=1}^{N_k} \frac{\omega(t_k) e_{\text{buff},k}}{(1 + \omega(t_k)^\top \omega(t_k))^2} \right), \end{aligned} \quad (23)$$

where $\sigma(t) := \nu(t)(U(t) \otimes U(t) - U(t - \Delta t) \otimes U(t - \Delta t))$, $\sigma_f := \nu(T)(U(T) \otimes U(T))$, and $\alpha_c \in \mathbb{R}^+$ is the critic gain of the gradient descent. The accumulated term is given by,

$$\omega(t_k) := \nu(t_k) \nabla_{\bar{x}}(U(t_k) \otimes U(t_k)) \hat{x} + \nabla_t (\nu(t_k)(U(t_k) \otimes U(t_k))).$$

By following (23) the error regulates to zero, $e_c \rightarrow 0$.

The intermittent controller will be updated whenever an event occurs, otherwise it will remain constant, keeping its latest value. By applying gradient descent, the actor weights' tuning law results in,

$$\begin{cases} \dot{\hat{W}}_a = 0 & , \text{ if } t \in [r_j, r_{j+1}) \\ \hat{W}_a^+ = \hat{W}_a - \alpha_a \frac{\bar{x} \bar{x}^\top}{(1 + \bar{x}^\top \bar{x})} e_a^\top & , \text{ if } t = r_{j+1} \end{cases} \quad (24)$$

where the convergence rate is specified by the gain $\alpha_a \in \mathbb{R}^+$. The actor tuning law (24) guarantees that $e_a \rightarrow 0$. The weighted estimation errors of the critic and the actor are defined by $\tilde{W}_c := \hat{W}_c - \hat{W}_c$ and $\tilde{W}_a := \hat{W}_a - \hat{W}_a$ respectively. Thus, by using (23) and (24) their dynamics yield,

$$\dot{\tilde{W}}_c = -\alpha_c \left(\frac{\sigma(t) \sigma(t)^\top}{(1 + \sigma(t)^\top \sigma(t))^2} + \Lambda \right) \tilde{W}_c,$$

$$\begin{cases} \dot{\tilde{W}}_a = 0 & , \text{ if } t \in [r_j, r_{j+1}) \\ \tilde{W}_a^+ = \tilde{W}_a - \alpha_a \frac{\bar{x} \bar{x}^\top}{(1 + \bar{x}^\top \bar{x})} (\tilde{W}_a + \tilde{Q}_{\text{xu}_d} R^{-1}) & , \text{ if } t = r_{j+1}, \end{cases}$$

where $\Lambda := \sum_{k=1}^{N_k} \frac{\omega(t_k) \omega(t_k)^\top}{(1 + \omega(t_k)^\top \omega(t_k))^2} \succ 0$.

D. Impulsive System Structure

Since the controller behaves as an impulsive system with discrete jumps, the closed-loop dynamics of the system (1) with the intermittent controllers (10), (17) takes the form of,

$$\begin{aligned} \dot{\hat{x}} &= A \bar{x} + B(u_d^* - \hat{u}_d) \\ &= A \bar{x} - B \left(Q_{\text{u}_d \text{u}_d}^{-1} Q_{\text{u}_d \text{x}} + \tilde{W}_a^\top \mu(t) \right) \hat{x}. \end{aligned} \quad (25)$$

The augmented state that captures the flow dynamics of the system at time $t \in [r_j, r_{j+1})$ is defined by $\psi := [\bar{x}^\top \hat{x}^\top \text{vec}(\tilde{W}_c^\top) \text{vec}(\tilde{W}_a^\top)]^\top \in \mathbb{R}^{((n+m)(n+m+1)/2) + 2n + nm}$, with time derivative,

$$\dot{\psi} = \begin{bmatrix} A \bar{x} + B(-Q_{\text{u}_d \text{u}_d}^{-1} Q_{\text{u}_d \text{x}} - \tilde{W}_a^\top) \hat{x} \\ \mathbf{0}_n \\ -\alpha_c \left(\frac{\sigma(t) \sigma(t)^\top}{(1 + \sigma(t)^\top \sigma(t))^2} + \Lambda \right) \tilde{W}_c \\ \mathbf{0}_{nm} \end{bmatrix}. \quad (26)$$

Similarly, the augmented state for the jump dynamics at time $t = r_{j+1}$ is defined by $\psi^+ := [\bar{x}^{+\top} \hat{x}^{+\top} \text{vec}(\tilde{W}_c^{+\top}) \text{vec}(\tilde{W}_a^{+\top})]^\top$ and its time derivative,

$$\psi^+ = \psi(t) + \begin{bmatrix} \mathbf{0}_n \\ \hat{x}(t) - \bar{x}(t) \\ \mathbf{0}_{(n+m)(n+m+1)/2} \\ \Psi(\tilde{W}_a^+) = \text{vec}(\Psi_1 + \Psi_2) \end{bmatrix}, \quad (27)$$

with partitioned vector,

$$\Psi(\tilde{W}_a^+) := \text{vec} \left(-\alpha_a \frac{\bar{x} \bar{x}^\top}{(1 + \bar{x}^\top \bar{x})} (\tilde{W}_a + \tilde{Q}_{\text{xu}_d} R^{-1}) \right). \quad (28)$$

Theorem 2: Consider the closed-loop system (25) with critic (16) and actor (17) approximators, tuned by (23) and (24) respectively. The origin is a globally asymptotically stable equilibrium point of the closed-loop system with state

ψ for all initial conditions $\psi(0)$, if the following,

$$\|e\|^2 \leq \frac{(1 - \beta^2)\underline{\lambda}(M)\|\bar{x}\|^2 + \underline{\lambda}(R)\|\hat{u}_d\|^2}{4(L(t)^2 + L_1(t)^2)\bar{\lambda}(R)}; \quad \frac{\underline{\lambda}(M)}{\bar{\lambda}(R)} > \frac{2L_1(t)^2}{\beta^2},$$

and the actor and critic gain inequalities hold,

$$0 < \alpha_a < \frac{2(4\bar{\lambda}(R) - 1)}{\bar{\lambda}(R) + 2}; \quad \alpha_c \gg \alpha_a; \quad \alpha_c \underline{\lambda}(\Xi) > 0. \quad (29)$$

Proof. See the Appendix. ■

V. MOTION PLANNING FRAMEWORK

The motion planning structure comprises of four stages: i) dynamic planning; ii) Q-learning; iii) terminal state evaluation; and iv) obstacle augmentation, as shown in Fig. 1.

Dynamic Planning: The RRT^X contains not only the subtree, but also the search-graph of the initial planning process. In this way, the algorithm can reuse the search-graph for a rewiring cascade whenever the environment changes. Consequently, information is transferred rapidly throughout the tree in the modified environment. Moreover, RRT^X maintains an ϵ -consistent graph, which guarantees the quality of existing paths and allows for quick replanning. The neighborhood size at each node remains constant by selecting neighbors to maintain the runtime at each iteration.

Obstacle Augmentation: Since the model of the system is unknown, it is assumed that the robot traverses straight paths for the RRT^X algorithm. In addition, optimality in terms of path planning usually indicates narrow distance between the obstacles and the path. In our case, kinodynamic constraints (1) as well as the optimal performance (3) result in traversing curved paths. Thus, there exists a deviation from the assumed straight-line path and the actual motion of the robot. These two factors may result in unsafe navigation and even collisions. To address this problem, we introduce the concept of kinodynamic distance and follow an obstacle augmentation strategy [18], [20]. Therefore, instead of considering the original shape of the obstacles, their augmented shape is taken into account. Whenever new obstacles are detected, the obstacle augmentation precedes the replanning process to avoid collision.

Q-Learning: At every pair of waypoints (x_0, x_r) of the planned path π , the proposed control law (17) is implemented to drive the system. The control scheme is realized by constantly evaluating the error gap (2). Whenever the bound of Theorem 2 is exceeded, the intermittent condition is activated to close the loop and update the controller. For the new controller, the critic is used to assess the policy, and the actor to perform the policy update. The critic approximates the Q-function according to (16), where \hat{W}_c are the critic parameters that can be computed online by (23). The buffer stores past experiences and assists the excitation of the learning signal. The process terminates as soon as the matrix $[\omega(t_1), \dots, \omega(t_{N_k})]$ becomes full rank [13]. The actor approximates the control policy according to (17), where \hat{W}_a are the actor parameters following the tuning law (24).

Terminal State Evaluation: A distance metric is employed to evaluate the terminal condition. At every Δt we compute the initial distance D_0 (5) and the relative distance D (6).

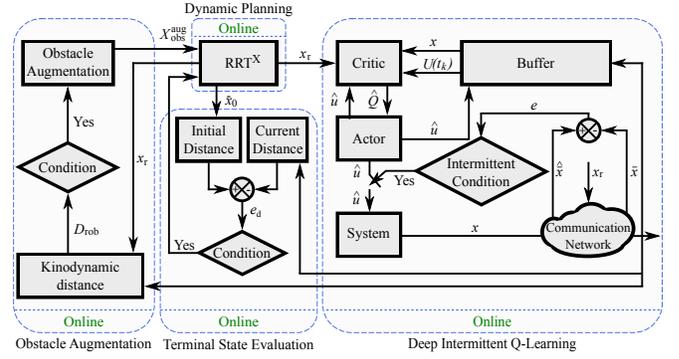


Fig. 1. The motion planning structure consists of four stages.

When the distance error drops below an admissible portion of the initial distance $e_d \geq \rho D_0$, where $\rho \in (0, 1)$, then the algorithm assigns the current state as the new initial state $x_{0,i+1} = x(t)$ and proceeds to the next pair of waypoints.

VI. SIMULATIONS AND RESULTS

We consider the system in [18] with plant and input matrices,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.5 & 0 & -1.125 & 0 \\ 0 & -0.5 & 0 & -1.125 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.025 & 0 \\ 0 & 0.025 \end{bmatrix}$$

with $x = [x_1 \ y_1 \ \dot{x}_1 \ \dot{y}_1]$ the state. The translations denoted \dot{x}_1, \dot{y}_1 , the velocities \dot{x}_1, \dot{y}_1 , the accelerations \ddot{x}_1, \ddot{y}_1 . The inputs are forces denoted f_1, f_2 . We set the finite horizon $T = 10$ s and the admissible window $\rho = 0.9$. The user-defined matrices are $M = 15I_4$ and $R = 0.55I_2$. To ensure positive values for the actor gain $\alpha_a > 0$, the matrix R is lower bounded by $\underline{\lambda}(R) > 0.25$ (29). Next, we set $P(T) = 0.5I_4$, $\beta = 0.6$, and $L = 30$. We select the rest parameters based on Theorem 2, $L_1 = 0.9\{(\beta\sqrt{\underline{\lambda}(M)/\bar{\lambda}(R)})/2\} = 2.82$, $\alpha_c = 90$, and $\alpha_a = 0.25\{(8\bar{\lambda}(R) - 4)/(\bar{\lambda}(R) + 2)\} = 0.01$.

The simulation results of the proposed method are presented in Fig. 2 and a demonstrating video in the URL:

<https://youtu.be/Sxu04gSdsEA>

The unpredictable dynamic environment consists of obstacles that appear, vanish, or stay fixed. Moreover, the obstacles are augmented based on the kinodynamic distance to guarantee collision-free planning. The initial shape of the obstacles is illustrated by blue polygons and their augmentation by the magenta around them. The traversed path of the robot is illustrated with a red solid line and the RRT^X path with a white line. The start state is located at $(-40, 40)$ and the goal state at $(0, -40)$. The robot moves toward the red location, which indicates the current terminal state. The colored background represent the *cost-to-go* from every location to the final goal. The methodology provides a safe motion planning framework that operates in real-time with reduced computations and limited communication. The robot successfully avoids the obstacles throughout the navigation for all 41 pairs of waypoints.

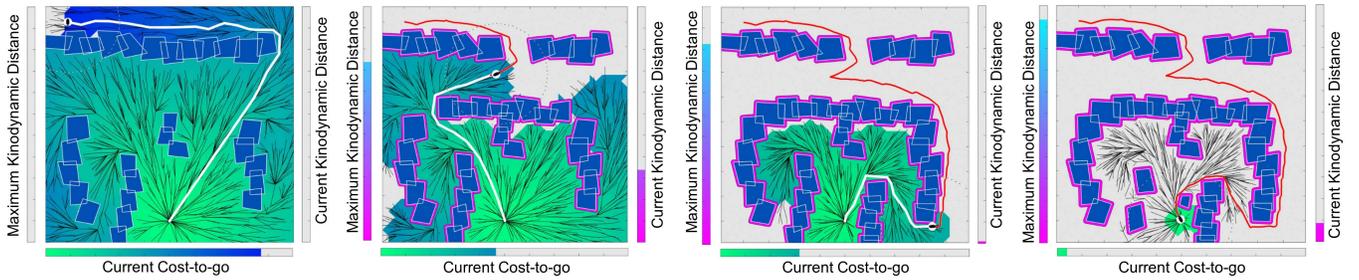


Fig. 2. Various time frames of the robot's collision-free navigation in an unpredictable dynamic environment.

VII. CONCLUSION

This paper proposed a real-time motion planning framework in unpredictable dynamic environments with intermittent Q-learning. The robot performed safe navigation with no collision. The methodology does not require the conservative PE condition and operates intermittently, which conserves computational and communication efforts.

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] M. Otte and E. Frazzoli, "RRT^X: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The Int. J. of Rob. Res.*, vol. 35, no. 7, pp. 797–822, 2016.
- [4] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM*, vol. 40, no. 5, pp. 1048–1066, 1993.
- [5] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE Intern. Conference on Robotics and Automation*, 2013, pp. 5054–5061.
- [6] 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.
- [7] 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.
- [8] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Tr. on Aut. Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
- [9] K. G. Vamvoudakis and H. Ferraz, "Model-free event-triggered control algorithm for continuous-time linear systems with optimal performance," *Automatica*, vol. 87, pp. 412–420, 2018.
- [10] 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.
- [11] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Courier Corporation, 2012.
- [12] G. Chowdhary, T. Yucelen, M. Muehlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 4, pp. 280–301, 2013.
- [13] 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 Networks and Learning Systems*, vol. 27, no. 11, pp. 2386–2398, 2016.
- [14] F. L. Lewis, D. Vrabie, and V. L. Szymos, *Optimal Control*, 3rd ed. John Wiley & Sons., 2012.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
- [16] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] G. P. Kontoudis and K. G. Vamvoudakis, "Robust kinodynamic motion planning using model-free game-theoretic learning," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 273–278.
- [21] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

APPENDIX

Proof of Theorem 2. First we consider the flow dynamics (26) and we define the Lyapunov function,

$$\mathcal{L}(\psi; t) := V^*(\bar{x}; t) + V^*(\hat{x}; t) + \frac{1}{2} \|\tilde{W}_c\|^2 + \frac{1}{2} \text{tr}\{\tilde{W}_a^T \tilde{W}_a\}, \quad (30)$$

where $\mathcal{L} > 0$ for all $t \geq 0$, and ψ is the augmented state. The time derivative for the closed-loop system of (30) is partitioned to, $\dot{\mathcal{L}} = T_1 + T_2 + T_3 + T_4$ with,

$$T_1 = \frac{1}{2} \bar{x}^T \dot{P}(t) \bar{x} + \bar{x}^T P(t) (A\bar{x} + B\hat{u}_d), \quad (31)$$

$$T_3 = -\alpha_c \tilde{W}_c^T \Xi \tilde{W}_c, \quad (32)$$

where $\Xi := \frac{\sigma\sigma^T}{(1+\sigma^T\sigma)^2} + \Lambda$. The parts $T_2 = 0$ and $T_4 = 0$ vanish, as they are updated only at every jump and remain constant the rest time. Then, from (32) the upper bound results in,

$$T_3 \leq -\alpha_c \lambda(\Xi) \|\tilde{W}_c\|^2. \quad (33)$$

Next, by substituting (8) to (31),

$$\begin{aligned} T_1 &= -\frac{1}{2} \bar{x}^T \left(P(t)A + A^T P(t) + M - P(t)BR^{-1}B^T P(t) \right) \bar{x} \\ &\quad + \bar{x}^T P(t) A\bar{x} + \bar{x}^T P(t) B\hat{u}_d \\ &= -\frac{1}{2} \bar{x}^T M \bar{x} - u_c^* R \hat{u}_d + \frac{1}{2} u_c^* R u_c^* \\ &\leq -\frac{1}{2} (\underline{\lambda}(M) \|\bar{x}\|^2 + \underline{\lambda}(R) \|\hat{u}_d\|^2) - \bar{\lambda}(R) \|\tilde{W}_a^T \mu(t) \bar{x} + \hat{W}_a^T \mu(t) e\|^2 \end{aligned} \quad (34)$$

Since the actor weights remain constant, then $\|\tilde{W}_a^T \mu(t)\| \leq L_1(t)$. By adding in both sides $1/2 \beta^2 \underline{\lambda}(M) \|\bar{x}\|^2$, using Young's inequality, and Assumption 1, the (34) takes the form of,

$$\begin{aligned} T_1 &\leq -\frac{1}{2} \left(\beta^2 \underline{\lambda}(M) - 2L_1(t)^2 \bar{\lambda}(R) \right) \|\bar{x}\|^2 - \frac{1}{2} (1 - \beta^2) \underline{\lambda}(M) \|\bar{x}\|^2 \\ &\quad + 2 \left(L(t)^2 + L_1(t)^2 \right) \bar{\lambda}(R) \|e\|^2 - \frac{1}{2} \underline{\lambda}(R) \|\hat{u}_d\|^2. \end{aligned} \quad (35)$$

If the following inequality is satisfied,

$$\|e\|^2 \leq \frac{(1 - \beta^2)\underline{\lambda}(M)\|\bar{x}\|^2 + \underline{\lambda}(R)\|\hat{u}_d\|^2}{4(L(t)^2 + L_1(t)^2)\bar{\lambda}(R)},$$

and $\underline{\lambda}(M)/\bar{\lambda}(R) > 2L_1(t)^2/\beta^2$ hold, then (35) yields,

$$T_1 \leq -\frac{1}{2} \left(\beta^2 \underline{\lambda}(M) - 2L_1(t)^2 \bar{\lambda}(R) \right) \|\bar{x}\|^2. \quad (36)$$

Given both upper bounds (33), (36), then $\dot{\mathcal{L}}(\psi; t)$ is non-positive for all ψ and $t \geq t_0$. Define $W_1(\psi; t) = W_2(\psi; t) := V^*(\bar{x}; t) + 1/2 \|\tilde{W}_c\|^2 > 0$ to get $W_1(\psi; t) \leq \mathcal{L}(\psi; t) \leq W_2(\psi; t)$. Hence, according to the Lyapunov stability theorem, the origin $\psi_e = 0$ is uniformly stable. Provided $\mathcal{L}(\psi; t)$ is lower-bounded, non-increasing, and its time derivative, comprising of (33) and (36), $\dot{\mathcal{L}}(\psi; t) = T_1 + T_3$ is bounded, then the $\mathcal{L}(\psi; t)$ (30) is uniformly continuous. Thus, Barbalat's lemma is satisfied, $\mathcal{L}(\psi; t) \rightarrow 0$ as $t \rightarrow \infty$. Since $\dot{\mathcal{L}}(\psi; t) > 0$ is positive definite, asymptotic stability holds from the Lyapunov stability theorem. Next, $W_1(\psi; t)$ is radially unbounded wrt $\|\bar{x}\|$, $\|\tilde{W}_c\|$, and hence globally properties hold. Thus, the equilibrium point at the origin $\psi_e = 0$ is globally uniformly asymptotically stable [21].

We continue with the jump dynamics (27) comprising of the sampled states and the actor policy updates. Define the Lyapunov function for the jump dynamics as,

$$\begin{aligned} \Delta \mathcal{L}(\psi; t) &:= \Delta V_1(\bar{x}^+, \bar{x}(r_{j+1}); t) + \Delta V_2 \\ &\quad + \Delta V_3(\tilde{W}_c^+, \tilde{W}_c(r_{j+1})) + \Delta V_4 \end{aligned} \quad (37)$$

where $\Delta \mathcal{L} > 0$. Note that both \bar{x} and \tilde{W}_c are defined continuously without jumps at the intermittent events. Thus, they both $\Delta V_1 = \Delta V_3 = 0$ vanish from the jump dynamics. Since during the jump $\hat{x}^+ = \hat{x}(r_{j+1})^+$, then ΔV_2 yields,

$$\begin{aligned} \Delta V_2 &= V^*(\hat{x}^+) - V^*(\hat{x}(r_{j+1})) = V^*(\hat{x}(r_{j+1})^+) - V^*(\hat{x}(r_{j+1})) \\ &\leq -\kappa(\hat{x}(r_{j+1}); t) \|\hat{x}(r_{j+1})\|, \end{aligned}$$

where $\kappa(\hat{x}(r_{j+1}); t)$ is of a class \mathcal{K} function wrt $\hat{x}(r_{j+1})$ [21]. Hence, the sampled state converges asymptotically to the origin, $\|\hat{x}(r_{j+1})\| \rightarrow 0$. By employing the partitioned vector (28), the last term of the Lyapunov function $\Delta \mathcal{L}$ (37) becomes,

$$\begin{aligned} \Delta V_4 &= \frac{1}{2\alpha_a} \left(\text{tr}\{\tilde{W}_a^T \Psi_1\} + \text{tr}\{\tilde{W}_a^T \Psi_2\} + \text{tr}\{\Psi_1^T \tilde{W}_a\} + \text{tr}\{\Psi_1^T \Psi_1\} \right. \\ &\quad \left. + \text{tr}\{\Psi_1^T \Psi_2\} + \text{tr}\{\Psi_2^T \tilde{W}_a\} + \text{tr}\{\Psi_2^T \Psi_1\} + \text{tr}\{\Psi_2^T \Psi_2\} \right). \end{aligned} \quad (38)$$

Next, by using Young's inequality, (38) takes the form of,

$$\begin{aligned} \Delta V_4 &\leq \|\tilde{W}_a^T x(r_{j+1})\|^2 \left(1 - \frac{1 + \alpha_a}{4\bar{\lambda}(R)} - \frac{\alpha_a}{8} \right) + \frac{1 + \alpha_a}{4\bar{\lambda}(R)} \|\tilde{Q}_{x_{ud}}\|^2 \\ &\quad + \frac{\alpha_a}{2\bar{\lambda}(R)^2} \|\tilde{Q}_{x_{ud}}\|^2. \end{aligned} \quad (39)$$

Then, the inequality $\Delta V_4 < 0$ is satisfied if \tilde{W}_a in (39) remains into the compact set $\Omega = \left\{ \tilde{W}_a \in \mathbb{R}^{n \times m} \mid \|\tilde{W}_a\| \leq \sqrt{\frac{\frac{1 + \alpha_a}{4\bar{\lambda}(R)} \|\tilde{Q}_{x_{ud}}\|^2 + \frac{\alpha_a}{2\bar{\lambda}(R)^2} \|\tilde{Q}_{x_{ud}}\|^2}{1 - \frac{1 + \alpha_a}{4\bar{\lambda}(R)} - \frac{\alpha_a}{8}}} \right\}$.

Note that another constraint arises from the denominator of the inequality which defines the compact set Ω as,

$$0 < \alpha_a < \frac{2(4\bar{\lambda}(R) - 1)}{\bar{\lambda}(R) + 2}.$$

Since the signals of the compact set Ω are asymptotically stable, then the set becomes a single point (vanishes) and thus $\|\tilde{W}_a\| \rightarrow 0$. ■