2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Assured Learning-Based Optimal Control subject to Timed Temporal Logic Constraints

Filippos Fotiadis[1], Christos K. Verginis[2], Kyriakos G. Vamvoudakis[1], and Ufuk Topcu[2]

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683417

Abstract— **We develop an algorithm for the optimal control of systems governed by unknown, nonlinear dynamics, to deliver tasks expressed as timed temporal logic constraints. The algorithm first computes a sequence of points in the operating environment, along with associated time stamps, so that the system completes its task if it follows the sequence. For the algorithm's second step, we develop a data-driven, on-the-fly control mechanism that learns how to transition from a point in the sequence to the next within a pre-specified time horizon. This algorithm accounts for the unknown dynamics, any unsafe zones in the environment and additional optimality criteria. We show that, after a finite period of data gathering, the resulting controller guarantees that the system indeed follows the sequence of points, leading to the satisfaction of the task.**

I. Introduction

Autonomous systems are prone to failures and abrupt changes that might render the underlying dynamics unknown; the control of such systems necessitates data-driven, learning-based techniques, which rely on data obtained on the fly. At the same time, resource limitations require that the exerted control effort of the underlying system is minimized, thus also generating an optimal control problem (OCP) [1]. Such a combination of objectives can be a complex task, which needs to be expressed in a comprehensive manner.

One method of conveying an autonomous system's objectives is via temporal logic languages, which can describe tasks more complex than the well-studied point-to-point navigation [2]. In particular, a special form of temporal logic, namely *timed temporal logic*, offers the incorporation of time constraints in planning objectives, hence providing for a rich variety of tasks [3]. However, resource limitations and optimality is often neglected in these objectives.

This paper addresses the OCP of an *unknown* control-affine system, which has to deliver tasks expressed as timed temporal logic constraints. The system is assumed to be continuous in state and time, and operates in an environment with unsafe zones. Our contribution lies in the integration of timed temporal tasks with control optimality, while accounting for the unknown, nonlinear system dynamics.

We develop a two-step algorithm to solve the aforementioned problem. The first step is the computation of a discrete timed path, i.e., a sequence of points to be visited at specific time stamps, that yields the execution of the task if followed by the system. The second step is the design of a control algorithm that exhibits the following properties: (i) it achieves the sequential navigation of the system to the points dictated by the computed path in the given time stamps; (ii) it minimizes the exerted control effort; and (iii) it guarantees the avoidance of the unsafe zones. In particular, we transform the problem to a finite-horizon OCP with safety constraints, and we use data obtained online from the current trajectory to accommodate the unknown dynamics. We prove that, after obtaining a sufficient amount of data, the system learns to navigate among the predefined points within the time intervals dictated by the derived path. Additionally, the control effort is minimized and the unsafe zones are avoided, hence leading to the successful execution of the task.

There exist numerous related works that consider planning and control under timed temporal logic specifications [4]–[15]. Most of the aforementioned works, however, consider simplistic single integrator models [5], [7], [11], finite-state systems [15] or neglect entirely the underlying dynamics [4]. The works [8]–[10], [12]–[14] consider more complex models that are either fully [10], [12]–[14] or partially [8], [9] known; [3] assumes unknown dynamics, restricted, however, to Lagrangian models with positive-definite input matrices.

Another issue with the related works on timed temporal logic-based planning is the lack of optimality characteristics; [8], [11], [13] are a few exemptions which, however, fail to guarantee optimality of the resulting controller, and use the underlying dynamics. On the other hand, in this paper we do ensure optimality through the use of a neural-network-based learning scheme. In particular, we extend previous works on actor-critic learning [16]–[18] by solving a series of OCPs over finite time horizons, for the safe timed transitions among predefined points related to the timed temporal task.

II. Preliminaries

Notation: We denote $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$, where $\mathbb{N}$ is the set of natural numbers. The sets of n-dimensional nonnegative and positive reals, with $n \in \mathbb{N}$, are denoted by $\mathbb{R}^n_{\geqslant 0}$ and $\mathbb{R}^n_{>0}$, respectively. $Z_1 \otimes Z_2$ is the Kronecker product of matrices Z_1 and Z_2. The operator $\mathrm{vec}(\cdot)$ denotes the vectorization of a matrix. Given an infinite sequence $\mathsf{s} = s_0 s_1 s_2 \dots$, we denote its j-suffix by $\mathsf{suff}(\mathsf{s}, j) = s_j s_{j+1} \dots$, respectively; $I_q \in \mathbb{R}^{q \times q}$ denotes the identity matrix. The closed ball centered at c_k with radius r_k is denoted by $\bar{\mathcal{B}}(c_k, r_k)$.

[1]F. Fotiadis and K. G. Vamvoudakis are with the Georgia Institute of Technology, Atlanta, GA, USA. Email: {ffotiadis, kyriakos}@gatech.edu. [2]C. K. Verginis and U. Topcu are with the University of Texas at Austin, Austin, TX, USA. Email: {cverginis,utopcu}@utexas.edu. This work was supported in part, by ARO under grant No. W911NF-$19-1-0270$, by ONR Minerva under grant No. N00014 $-18-1-2160$, and by NSF under grant Nos. CAREER CPS-1851588, SATC-1801611, and 1646522, by NASA ULI under grant number 80NSSC20M0161, by ARL ACC-APG-RTP W911NF1920333, by AFOSR FA9550-19-1-0005, and by the Onassis Foundation-Scholarship ID: F ZQ $064-1/2020-2021$.

Definition 1 ([19])**.** A time sequence $t_1, t_2, \ldots$ is a (infinite unless otherwise stated) sequence of time values $t_j \in \mathbb{R}_{\geqslant 0}$, for all $j \in \mathbb{N}$, satisfying (i) $t_j < t_{j+1}$, for all $j \in \mathbb{N}$ and (ii) for all $t' \in \mathbb{R}_{\geqslant 0}$ there exists $j \geqslant 1$ such that $t_j \geqslant t'$. □

An *atomic proposition* is a statement over the variables or parameters of a problem that is either True ($\top$) or False ($\bot$). Assume now that $\mathcal{AP}$ is a finite set of such propositions.

Definition 2. Let $\mathcal{AP}$ be a finite set of atomic propositions. A *timed word* w over $\mathcal{AP}$ is an infinite sequence $w := (w_1, t_1)(w_2, t_2) \ldots$ where $w_1, w_2, \ldots$ is an infinite word over $2^{\mathcal{AP}}$ and $t_1, t_2, \ldots$ is a time sequence. □

Definition 3. A *Weighted Transition System (WTS)* is a tuple $(\Pi, \Pi_0, \longrightarrow, \mathcal{AP}, \mathcal{L}, \gamma)$, where Π is a finite set of states, $\Pi_0 \subseteq \Pi$ is a set of initial states, $\longrightarrow \subseteq \Pi \times \Pi$ is a transition relation, $\mathcal{AP}$ is a finite set of atomic propositions, $\mathcal{L} : \Pi \to 2^{\mathcal{AP}}$ is a labeling function, and $\gamma : \longrightarrow \to \mathbb{R}_{>0}$ is a map that assigns a positive weight to each transition. □

Definition 4. A *timed run* of a WTS is an infinite sequence $r = (r_1, t_1)(r_2, t_2) \ldots$, such that $r_1 \in \Pi_0$ and $r_j \in \Pi$, $(r_j, r_{j+1}) \in \longrightarrow$, for all $j \in \mathbb{N}$. The time stamps t_j are inductively defined with $t_1 = 0$ and $t_{j+1} = t_j + \gamma(r_j, r_{j+1})$, for all $j \in \mathbb{N}$. The timed run r generates a timed word $w(r) = w_1(r_1), w_2(r_2), \ldots = (\mathcal{L}(r_1), t_1), (\mathcal{L}(r_2), t_2), \ldots$ over the set $2^{\mathcal{AP}}$, where $\mathcal{L}(r_j)$ is the subset of atomic propositions $\mathcal{AP}$ that are true at state r_j at time t_j, $j \in \mathbb{N}$. □

The syntax of a timed temporal logic formula over $\mathcal{AP}$ is defined by a grammar that has the form

$$\varphi := \mathsf{p} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc_I \varphi \mid \Diamond_I \varphi \mid \Box_I \varphi \mid \varphi_1 \mathcal{U}_I \varphi_2, \quad (1)$$

where $\varphi \in \mathcal{AP}$, and $\bigcirc$, $\Diamond$, $\Box$, and $\mathcal{U}$ are the next, future, always, and until operators, respectively; I is a nonempty time interval in one of the followings forms: $[i_1, i_2], [i_1, i_2), (i_1, i_2], (i_1, i_2), [i_1, \infty), (i_1, \infty)$, with i_1, $i_2 \in \mathbb{Q}$. Several languages are subsets of the form (1), such as Metric Temporal Logic (MTL), Metric Interval Temporal Logic (MITL), Bounded MTL, coFlat MTL, or Time Window Temporal Logic (TWTL) [20], [21]. Here we define the generalized semantics of (1) over discrete observations (point-wise semantics) [22]. The next definition considers the satisfaction of a formula by a timed run.

Definition 5. [22], [23] Given a sequence $\mathsf{R} = (\pi_0, t_0)(\pi_1, t_1) \ldots$ and a timed formula φ, we define $(\mathsf{R}, i) \models \varphi$, $i \in \mathbb{N}_0$ (R satisfies φ at i) as follows:

$$(\mathsf{R}, i) \models p \Leftrightarrow p \in \mathcal{L}(\pi_i), \qquad (\mathsf{R}, i) \models \neg\varphi \Leftrightarrow (\mathsf{R}, i) \not\models \varphi,$$
$$(\mathsf{R}, i) \models \varphi_1 \wedge \varphi_2 \Leftrightarrow (\mathsf{R}, i) \models \varphi_1 \text{ and } (\mathsf{R}, i) \models \varphi_2,$$
$$(\mathsf{R}, i) \models \bigcirc_I \varphi \Leftrightarrow (\mathsf{R}, i+1) \models \varphi \text{ and } t_{i+1} - t_i \in I,$$
$$(\mathsf{R}, i) \models \varphi_1 \mathcal{U}_I \varphi_2 \Leftrightarrow \exists k \geqslant i \text{ such that } (\mathsf{R}, k) \models \varphi_2,$$
$$t_k - t_i \in I \text{ and } (\mathsf{R}, m) \models \varphi_1, \forall m \in \{i, \ldots, k\}.$$

Also, $\Diamond_I \varphi = \top \mathcal{U}_I \varphi$ and $\Box_I \varphi = \neg \Diamond_I \neg \varphi$. Finally, R satisfies φ, denoted by $\mathsf{R} \models \varphi$, if and only if $(\mathsf{R}, 0) \models \varphi$. □

III. Problem Formulation

Consider, $\forall t \geqslant t_0 \geqslant 0$, a nonlinear system with dynamics

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x(t), t), \; x(t_0) = x_0, \quad (2)$$

where $x : [t_0, \infty) \to \mathbb{R}^n$ denotes the system's states with initial condition $x_0 \in \mathbb{R}^n$ at $t = t_0$, $u : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^m$ is a control input, and $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^m \to \mathbb{R}^{n \times m}$ are unknown, locally Lipschitz functions.

Consider also $K \in \mathbb{N}$ points of interest in the state space, denoted by $c_k \in \mathbb{R}^n$, for $k \in \mathcal{K} := \{1, \ldots, K\}$. Each point c_k, $k \in \mathcal{K}$, corresponds to certain properties of interest, which are expressed as Boolean variables via the finite set of atomic propositions $\mathcal{AP}$. These properties shared by a point of interest are naturally inherited to some neighborhood of that point. Hence, we also define for each $k \in \mathcal{K}$ the region of interest π_k, corresponding to the point of interest c_k, as the set $\pi_k := \bar{\mathcal{B}}(c_k, \rho_k)$, with $\rho_k > 0$ chosen such that $\pi_k \cap \pi_{k'} = \emptyset$, for all $k, k' \in \mathcal{K}$ with $k \neq k'$.

Denote $\Pi := \{\pi_1, \ldots, \pi_K\}$. Then, the properties satisfied at each region of interest are provided by the labeling function $\mathcal{L} : \Pi \to 2^{\mathcal{AP}}$. Informally, $\mathcal{L}$ assigns to each region π_k, $k \in \mathcal{K}$, the subset of the atomic propositions that hold true in that region. The system is assumed to be in a π_k simply when $x \in \pi_k$. We further need the following assumption.

Assumption 1. It holds that $f(c_k) = 0$ for all $k \in \mathcal{K}$. □

Along with Π, we further consider a set of K_o unsafe pairwise disjoint spherical zones $\mathcal{O} := \{o_1, \ldots, o_{K_o}\}$, with $o_k := \bar{\mathcal{B}}(c_{o_k}, \rho_{o_k})$, $k \in \mathcal{K}_o := \{1, \ldots, K_o\}$, satisfying $o_k \cap \pi_{k'} = \emptyset$, for all $(k, k') \in \mathcal{K}_o \times \mathcal{K}$, which defines the free space $\mathcal{F} := \mathbb{R}^n \backslash \mathcal{O}$. We are interested in achieving timed temporal specifications over the atomic propositions $\mathcal{AP}$ while avoiding the unsafe zones. We achieve that by guaranteeing safe timed transitions between the regions of interest in Π. We first need the following definition regarding the *behavior* of the system.

Definition 6. Consider an agent trajectory $x : [t_0, \infty) \to \mathbb{R}^n$ of (2). Then, a *timed behavior* of x is the infinite sequence $\mathsf{b}(t_0) := (x(t_0), \sigma_0, t_0)(x(t_1), \sigma_1, t_1) \ldots$, where $t_0, t_1, \ldots$ is a time sequence according to Definition 1, $x(t_i) \in \pi_{j_i}$, $j_i \in \mathcal{K}$ for all $i \in \mathbb{N}_0$, and $\sigma_i = \mathcal{L}(\pi_{j_i}) \subseteq 2^{\mathcal{AP}}$, i.e., the subset of atomic propositions that are true when $x(t_j) \in \pi_{j_i}$, for $i \in \mathbb{N}_0$. The timed behavior b satisfies a timed formula φ *safely* if $\mathsf{b}_\sigma(t_0) := (\sigma_0, t_0)(\sigma_1, t_1) \ldots \models \varphi$ and $x(t) \in \mathcal{F}$, for all $t \geqslant t_0$. It *eventually* satisfies φ safely if there exists $j \in \mathbb{N}$ such that $\mathsf{suff}(b_\sigma(t_0), j) = \mathsf{suff}(a_\sigma(t_0), j)$, for some $a_\sigma(t_0) \models \varphi$ and $x(t) \in \mathcal{F}$, for all $t \geqslant t_j$. □

We develop a learning-based control strategy such that the system learns how to safely execute transitions in Π, resulting in eventual satisfaction of φ, while also achieving optimality with respect to some user-defined cost. Note that eventual satisfaction implies that φ dictates repetitive tasks and/or tasks over long time horizons that the system is able to learn to execute. The latter, however, is not a restrictive assumption, since such tasks encompass the full potential of timed temporal logic languages.

Define now, for each point of interest c_i, the error $e_i := x - c_i$, evolving according to the dynamics

$$\dot{e}_i = F_i(e_i) + G_i(e_i)u := f(e_i + c_i) + g(e_i + c_i)u, \quad (3)$$

for all $i \in \mathcal{K}$, and the performance criteria:

$$J(e_i(t_0), t_0, t_f, u) := \int_{t_0}^{t_f} r(e_i(\tau), u(e_i(\tau), \tau))\mathrm{d}\tau, \quad (4)$$

with $t_0 \geqslant 0$, $t_f > t_0$. Here, $r(e, u) := Q(e) + S(u)$ is a metric of performance, with $S(u) := u^\top Ru$, $R > 0$, and $Q : \mathbb{R}^n \to \mathbb{R}_{\geqslant 0}$ being a positive-definite function. This gives rise to the *timed behavior cost* of a timed behavior b.

Definition 7. Consider a system closed-loop trajectory $x : [t_0, \infty) \to \mathbb{R}^n$ along the control input u and the associated timed behavior $\mathsf{b} = (x(t_0), \sigma_0, t_0)(x(t_1), \sigma_1, t_1) \ldots$, with $x(t_i) \in \pi_{j_i}$, $j_i \in \mathcal{K}$ for all $i \in \mathbb{N}_0$. The *timed behavior cost* J is the infinite sequence of functions $\mathsf{J} := J_0 J_1 \cdots$, where $J_i := J(e_{j_{i+1}}(t_i), t_i, t_{i+1}, u)$, for all $i \in \mathbb{N}_0$. □

The cost of the timed behavior naturally leads to the *ϵ-optimal timed behavior* defined next, where $\mathcal{A}(t_a, t_b)$ is the set of all functions from $\mathbb{R}^n \times [t_a, t_b]$ to $\mathbb{R}^m$, $t_b > t_a \geqslant t_0$:

Definition 8. Consider a system trajectory $x : [t_0, \infty) \to \mathbb{R}^n$. Given $\epsilon > 0$, its timed behavior $\mathsf{b}(t_0) = (x(t_0), \sigma_0, t_0)(x(t_1), \sigma_1, t_1) \ldots$ is said to be *ϵ-optimal*, if the associated timed behavior cost $\mathsf{J} = J_0 J_1 \ldots$ satisfies $\|J_i - J_i^\star\| \leqslant \epsilon$, for all $i \in \mathbb{N}$, where $J_i^\star := \min\limits_{\alpha \in \mathcal{A}(t_i, t_{i+1})} J(e_{j_{i+1}}(t_i), t_i, t_{i+1}, \alpha)$. □

We can now state the problem considered in this work.

Problem 1. Let a system evolve according to the unknown dynamics (2), and with initial position $x(t_0) \in \pi_{j_0}$, $j_0 \in \mathcal{K}$. Given a timed formula φ over $\mathcal{AP}$ and a labeling function $\mathcal{L}$, design a control law $u : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^m$ that results in a solution $x : [t_0, \infty) \to \mathbb{R}^n$, which achieves an *$\epsilon$-optimal* timed behavior that eventually satisfies φ safely. □

The next sections describe our two-layered solution to Problem 1. We first synthesize a high-level timed path over Π that satisfies φ, by neglecting the unknown dynamics (2). Then, we design a novel learning-based control algorithm that learns how to execute safe timed transitions over Π based on data obtained online from the current trajectory, which leads to the eventual safe satisfaction of φ.

IV. High-level Plan Generation

The first ingredient of the proposed solution is the derivation of a high-level plan that satisfies the given formula φ. To this end, we abstract the motion of the system as a finite weighted transition system [2]

$$\mathcal{T} := (\Pi, \Pi_0, \longrightarrow, \mathcal{AP}, \mathcal{L}, \gamma), \quad (5)$$

where Π is the discretized state space, $\Pi_0 \subseteq \Pi$ is the initial region, computed as $\Pi_0 := \pi_{k_0}$, $k_0 := \arg\min_{k \in \mathcal{K}}\{\|x(t_0) - c_k\|\}$, $\longrightarrow \subseteq \Pi \times \Pi$ is a transition relation, $\mathcal{AP}$ and $\mathcal{L}$ are the set of atomic propositions and labeling function, respectively, defined in the previous section, and $\gamma : \longrightarrow \to \mathbb{R}_{>0}$ is a map that assigns a positive weight to each transition. For now, we assume that the system can execute the transitions among the regions in Π within the time interval dictated by γ; the latter can be chosen according to several criteria, such as input capabilities of the system, Euclidean distance among points of interest, etc[1]. In the next section we will consider the control design for the execution of these timed transitions.

Given the transition system $\mathcal{T}$ and the formula φ, we can apply standard formal verification methodologies in order to compute a timed path over Π that satisfies φ. The most common practice to achieve this is the following: Firstly, φ is algorithmically translated to a Timed Büchi Automaton (TBA) $\mathcal{A}_B$, a system consisting of a discrete set of states associated with $\mathcal{AP}$, whose accepting runs satisfy φ [2]. Secondly, we compute the product between the two discrete systems $\widetilde{\mathcal{T}} := \mathcal{T} \otimes \mathcal{A}_B$. Finally, $\widetilde{\mathcal{T}}$ is viewed as a graph and standard graph-based algorithms are used to derive a *timed* path that satisfies φ. This path has the prefix-suffix form

$$\mathsf{p} = (\pi_{k_0}, t_0) \ldots (\pi_{k_{\mu-1}}, t_{k_{\mu-1}}) \Big[(\pi_{k_\mu}, t_{k_{\mu_1}}) \ldots (\pi_{k_{\mu+\nu}}, t_{k_{\mu_2}}) \Big]^\omega,$$

where $\mu_1 := \mu + \iota\nu$, $\mu_2 := \mu + (\iota + 1)\nu$, for positive μ, ν, where the superscript ω denotes infinite repetition and $\iota = 0, 1, \ldots$ denotes the repetition index. The execution of p produces a trajectory $x(t)$, $t \geqslant t_0$, with timed behavior $\mathsf{b}(t_0)$ that satisfies φ, i.e., $\mathsf{b}_\sigma(t_0) \models \varphi$ (see Definition 6). One can also obtain a timed path p satisfying φ using optimization methodologies. In particular, it has been shown that the satisfaction of a timed temporal formula can be formulated as a Mixed Integer Linear Programming (MILP) problem [4], where binary variables are introduced to represent the several atomic propositions and the time constraints involved in φ.

We assume that the timed path p is feasible. Hence, in the next section, we design a data-based learning control protocol that learns over time how to successfully execute the timed transitions in p, while avoiding the unsafe zones. This leads to the eventual satisfaction of φ, as per Def. 6.

V. Optimal transition

This section describes the data-based optimal control design for the *optimal* timed transition among two regions π_k and π_ℓ, which is defined as follows.

Definition 9. Assume that $x(t_k) \in \pi_k$, for $t_k \in \mathbb{R}_{\geqslant 0}$. Then, the system performs an *optimal* timed transition to π_ℓ, $\ell \in \mathcal{K} \backslash \{k\}$, denoted by $\pi_k \longrightarrow \pi_\ell$, if it applies a time-varying feedback control law $u : \mathbb{R}^n \times [t_k, t_\ell] \to \mathbb{R}^m$ such that, for some $\delta \in \mathbb{R}_{>0}$, the solution of the closed loop system (2) satisfies the following:

1) $x(t) \in \pi_\ell$ for all $t \in [t_k + \delta, t_\ell)$,
2) $x(t) \in \mathcal{F} \backslash \bigcup_{m \in \mathcal{K} \backslash \{\ell\}} \pi_m$ for all $t \in [t_k, t_\ell]$,
3) $u(x(t), t) = \arg J_i^\star$, where we define $J_i^\star = \min_{\alpha \in \mathcal{A}(t_k, t_\ell)} J(e_\ell(t_k), t_k, t_\ell, \alpha)$.

The timed transition is ϵ-optimal, if 3) is replaced by $\|J(e_\ell(t_k), t_k, t_\ell, u(x(t), t)) - J_i^\star\| \leqslant \epsilon$. □

[1] One can also consider online reconfiguration algorithms that give an optimal time duration based on exerted control effort [11].

A. Optimal Control with Soft Constraints

Evidently, it is not necessary that a control law u that minimizes (4) can always achieve the timed behavior described in Problem 1. Hence, the minimization of (4) has to be subject to some hard constraints imposing the desired timed behavior, or the desired behavior can be incorporated as a soft constraint in the cost (4). To achieve the latter, let us consider two regions of interest $\pi_k,\ \pi_\ell$, corresponding to two subsequent time instances t_k and t_ℓ, with $k,\ \ell \in \mathcal{K}$. Then, we redefine the performance criterion (4) into:

$$J_\ell(e_\ell(t_k), t_k, u) := \int_{t_k}^{t_\ell} \Big(\gamma r(e_\ell(\tau), u(e_\ell(\tau), \tau)) + L_{k,\ell}(e_\ell(\tau))\Big) \mathrm{d}\tau + \phi(e_\ell(t_\ell)), \quad (6)$$

$$\text{subject to:} \qquad \dot{e}_\ell = F_\ell(e_\ell) + G_\ell(e_\ell)u, \quad (7)$$

where we also drop the final time t_ℓ argument for brevity. In (6), $\phi : \mathbb{R}^n \to \mathbb{R}_{\geqslant 0}$ is a positive-definite term that penalizes the deviation of the terminal state from the point of interest c_ℓ. In addition, $L_{k,\ell} : \mathbb{R}^n \to \mathbb{R}_{\geqslant 0}$ is another penalty term satisfying $L_{k,\ell}(0) = 0$, and designed so that the system under the controller that minimizes (6) avoids all unsafe zones $\mathcal{O}$ and regions $\pi_i,\ i \in \mathcal{K}\backslash\{\ell\}$. It is strictly positive in these regions, monotonically decays to and remains equal to zero after a small distance outside them, and is also continuously differentiable. Finally, $\gamma > 0$ dictates a trade-off between: a) ensuring avoidance of the unsafe zones and the regions $\pi_i,\ i \in \mathcal{K}\backslash\{\ell\}$ and satisfaction of the terminal region specification; b) achieving good performance according to the metric $r(\cdot,\cdot)$.

Following [1], it can be shown that an infinitesimal expression for a continuously differentiable value function $V_\ell^u := J_\ell(\cdot,\cdot,u)$, which is equivalent to (6), is given by

$$\mathrm{LE}(V_\ell^u, u) = \nabla_t V_\ell^u(e_\ell, t) + \nabla_\ell V_\ell^u(e_\ell, t)^{\mathrm{T}}(F_\ell(e_\ell) + G_\ell(e_\ell)u) + \gamma r(e_\ell, u) + L_{k,\ell}(e_\ell) = 0, \quad (8)$$

which is a partial differential equation. If we let $t_k = t_\ell$, then owing to (6) the boundary condition of (8) is

$$V_\ell^u(e_\ell(t_\ell), t_\ell) = \phi(e(t_\ell)). \quad (9)$$

Define the optimal value function as $V_\ell^\star(e_\ell, t) = \min_u \ J_\ell(e_\ell, t, u)$, for all $e_\ell \in \mathbb{R}^n,\ t \in [t_k,\ t_\ell]$. If $V_\ell^\star$ is continuously differentiable, by following [1] we can derive the corresponding minimizing controller $u_\ell^\star$ as:

$$u_\ell^\star(e_\ell, t) = -\frac{1}{2\gamma} R^{-1} G_\ell(e_\ell)^{\mathrm{T}} \nabla_{e_\ell} V_\ell^\star(e_\ell, t). \quad (10)$$

Combining (10) with $\mathrm{LE}(V_\ell^\star, u_\ell^\star) = 0$, we obtain the Hamilton-Jacobi-Bellman (HJB) equation:

$$\nabla_t V_\ell^\star(e_\ell, t) + \gamma Q(e_\ell) + L_{k,\ell}(e_\ell) + \nabla_{e_\ell} V_\ell^\star(e_\ell, t)^{\mathrm{T}} F_\ell(e_\ell) - \frac{1}{4\gamma} \nabla_{e_\ell} V_\ell^\star(e_\ell, t)^{\mathrm{T}} G_\ell(e_\ell) R^{-1} G_\ell(e_\ell)^{\mathrm{T}} \nabla_{e_\ell} V_\ell^\star(e_\ell, t) = 0,$$
$$V_\ell^\star(e_\ell, t_\ell) = \phi(e_\ell). \quad (11)$$

Equation (11) needs to be solved in order to compute (10).

The following theorem shows that if γ is picked sufficiently small and a controllability condition holds, then the optimal policy $u_\ell^\star$ can achieve an optimal timed transition.

Theorem 1. *Assume that there exists a control law $u_\ell^c : \mathbb{R}^n \times [t_k,\ t_\ell]$ such that the closed-loop system $e_\ell(t)$ under $u = u_\ell^c$ satisfies: a) $e_\ell(t_\ell) = 0$; b) $L_{k,\ell}(e_\ell(t)) = 0$, for all $t \in [t_k,\ t_\ell]$. Then, there exists $\gamma^\star > 0$, such that if $\gamma < \gamma^\star$, the closed-loop system $e_\ell(t)$ under $u = u_\ell^\star$ executes an optimal timed transition, according to Def. 9.*

Proof. Denote by e_ℓ^c the trajectories of e_ℓ under $u = u_\ell^c$, and by $e_\ell^\star$ the trajectories of e_ℓ under $u = u_\ell^\star$. By definition, it holds that $L_{k,\ell}(e_\ell^c(t)) = 0$ for all $t \in [t_k,\ t_\ell]$, and $\phi(e_\ell^c(t_\ell)) = 0$. Hence,

$$J_\ell(e_\ell(t_k), t_k, u_\ell^c) = \int_{t_k}^{t_\ell} \gamma r(e_\ell^c(\tau), u_\ell^c(e_\ell^c(\tau)) \mathrm{d}\tau. \quad (12)$$

By optimality, it follows that

$$0 \leqslant J_\ell(e_\ell(t_k), t_k, u_\ell^\star) \leqslant J_\ell(e_\ell(t_k), t_k, u_\ell^c). \quad (13)$$

Due to (12), $\lim_{\gamma \to 0^+} J_\ell(e_\ell(t_k), t_k, u_\ell^c) = 0$. As a result, it follows from (13) that $\lim_{\gamma \to 0^+} J_\ell(e_\ell(t_k), t_k, u_\ell^\star) = 0$, hence also $\lim_{\gamma \to 0^+} L_{k,\ell}(e_\ell^\star(t)) = 0$ for all $t \in [t_k,\ t_\ell]$, and $\lim_{\gamma \to 0^+} \phi(e_\ell^\star(t_\ell)) = 0$. Hence, for all $\epsilon^\star > 0$ there exists $\gamma^\star > 0$, such that if $\gamma < \gamma^\star$, then $L_{k,\ell}(e_\ell^\star(t)) < \epsilon^\star$ for all $t \in [t_k,\ t_\ell]$ and $\phi(e_\ell^\star(t_\ell)) < \epsilon^\star$. The result follows by the design properties of $L_{k,\ell}$ and ϕ. ■

Next, we drop the subscript ℓ for ease of exposition.

B. Policy Iteration

The analytic solution to (11) is hard to obtain, hence we have to resort to approximate solution methods. To do so, we will require a few definitions and assumptions. First, for the cost (6) to be properly defined and for the corresponding value function to be continuously differentiable, we only consider control laws that are *admissible*.

Definition 10. A control law $u : \mathbb{R}^n \times [t_k,\ t_\ell] \to \mathbb{R}^m$ is *admissible* with respect to the cost (6), denoted by $u \in \mathcal{U}$, if

- u is continuous over $\mathbb{R}^n \times [t_k,\ t_\ell]$ with $u(0,t) = 0$ for all $t \in [t_k,\ t_\ell]$; and
- the origin of system (7) is uniformly Lyapunov stable under u, the trajectories of (7) are bounded for all $t \in [t_k,\ t_\ell]$, and the cost (6) is finite for all e_ℓ, t_k. □

Next, let $\mathcal{P}_+$ denote the set of continuously differentiable functions $\mathbb{R}^n \times [t_k,\ t_\ell] \to \mathbb{R}$. For any function $V \in \mathcal{P}_+$, assume that $V(\cdot, \bar{t})$ is positive-definite for all $\bar{t} \in [t_k,\ t_\ell]$. We now need the following assumption for $V^\star$ [16], [18].

Assumption 2. The optimal value function $V^\star$, which solves the HJB equation (11), belongs to $\mathcal{P}_+$, i.e., $V^\star \in \mathcal{P}_+$. □

Next, we present the Policy Iteration (PI) algorithm for solving the finite-horizon, time-varying HJB equation.

Policy Iteration

Let $u_0 \in \mathcal{U}$. Then, for all $i \in \mathbb{N}$, perform the iteration:

1) Evaluate the value function V^{u_i} by solving (8):

$$\nabla_t V^{u_i}(e,t) + \nabla_e V^{u_i}(e,t)^{\mathrm{T}}\left(F(e)+G(e)u_i\right) + \gamma r(e,u_i) + L(e) = 0,\ \forall t \in [t_k,\ t_\ell], \tag{14}$$

with $V^{u_i}(e(t_\ell),t_\ell) = \phi(e(t_\ell))$.

2) Choose the next control law u_{i+1} as

$$u_{i+1}(e,t) = -\frac{1}{2\gamma}R^{-1}G(e)^{\mathrm{T}}\nabla_e V^{u_i}(e,t). \tag{15}$$

The following lemma is needed to prove convergence of PI.

Lemma 1. *Consider the sequence of control laws* $\{u_i\}_{i\in\mathbb{N}}$ *and continuously differentiable value functions* $\{V^{u_i}\}_{i\in\mathbb{N}}$ *generated by the PI algorithm through equations* (14)-(15). *Let* u_i *be admissible, for some* $i \in \mathbb{N}$. *Then:*

1) u_{i+1} *is admissible.*

2) $V^\star(e,t) \leqslant V^{u_{i+1}}(e,t) \leqslant V^{u_i}(e,t),\ \forall (e,t) \in \mathbb{R}^n \times [t_k,\ t_\ell]$.

Proof. We will provide only a sketch of the proof. For the first part, over the trajectories of (3) under $u = u_{i+1}$:

$$\begin{aligned}\dot{V}^{u_i} &= \nabla_t V^{u_i} + (\nabla_e V_i^u)^{\mathrm{T}}(F+Gu_i) + (\nabla_e V_i^u)^{\mathrm{T}}G(u_{i+1}-u_i)\\ &= -\gamma Q(e) - \gamma S(u_i) - L(e) - 2\gamma u_{i+1}^{\mathrm{T}}R(u_{i+1}-u_i)\\ &= -\gamma Q(e) - L(e) - \gamma S(u_{i+1}) - \gamma S(u_{i+1}-u_i) \leqslant 0, \end{aligned} \tag{16}$$

where we used (14) and (15). Hence, exploiting also Assumption 2, and using arguments similar to [24], one can prove the admissibility of u_{i+1}. For item 2), the integration of (16) over $t \in [t_k,\ t_\ell]$ yields:

$$\begin{aligned}V^{u_i}(e(t_\ell),t_\ell) - V^{u_i}(e_k,t_k) = -\int_{t_k}^{t_\ell}\big(\gamma Q(e)\\ + \gamma S(u_{i+1}) + L(e)\big)\mathrm{d}\tau - \int_{t_k}^{t_\ell}\gamma S(u_{i+1}-u_i)\mathrm{d}\tau.\end{aligned} \tag{17}$$

Owing to (9), we have $V^{u_i}(e(t_\ell),t_\ell) = V^{u_{i+1}}(e(t_\ell),t_\ell) = \phi(e(t_\ell))$. Therefore, (17) is equivalent to:

$$V^{u_i}(e_k,t_k) = V^{u_{i+1}}(e_k,t_k) + \int_{t_k}^{t_\ell}\gamma S(u_{i+1}-u_i)\mathrm{d}\tau.$$

Hence, $V^{u_{i+1}}(e,t) \leqslant V^{u_i}(e,t), \forall (e,t) \in \mathbb{R}^n \times [t_k,\ t_\ell]$, while the inequality $V^\star(e,t) \leqslant V^{u_i}(e,t)$ holds by optimality. ■

Theorem 2. *Let* $u_0 \in \mathcal{U}$. *Then, the PI algorithm described through equations* (14)-(15) *guarantees that* $\lim\limits_{i\to\infty} V^{u_i} = V^\star$ *and* $\lim\limits_{i\to\infty} u_i = u^\star$. *The convergence is uniform on any compact subset of* $\mathbb{R}^n \times [t_k,\ t_\ell]$.

Proof. Given the monotonicity results of Lemma 1, the proof follows similar steps with [24] and is thus omitted. ■

C. Approximate Solution to the Time-Varying HJB Equation

The PI algorithm requires knowledge of the system's dynamics functions $F,\ G$. Towards implementing a model-free version of PI, we rewrite the system error dynamics as

$$\dot{e} = F(e) + G(e)u_i(e,t) + G(e)v_i(e,t),\ t \geqslant 0, \tag{18}$$

where $v_i = u - u_i$, $i \in \mathbb{N}$, and u_i is as defined in (15). Taking the total time derivative of the value function V^{u_i}, $i \in \mathbb{N}$, along the trajectories of (18), and using (14)-(15), we obtain

$$\begin{aligned}\dot{V}^{u_i} &= \nabla_t V^{u_i} + (\nabla_e V^{u_i})^{\mathrm{T}}\left(F(e) + G(e)u_i(e,t) + G(e)v_i(e,t)\right)\\ &= -\gamma Q(e) - \gamma S(u) - L(e) - 2\gamma u_{i+1}(e,t)^{\mathrm{T}}Rv_i(e,t).\end{aligned} \tag{19}$$

Integrating (19) over any time interval $[t,\ t+T] \subseteq [t_k,\ t_\ell]$, with $T > 0$ and for all $t \in [t_k,\ t_l - T]$, we derive

$$\begin{aligned}V^{u_i}(e(t+T),t+T) - V^{u_i}(e(t),t) = -\int_t^{t+T}\Big(\gamma Q(e)\\ + L(e) + \gamma S(u_i(e,\tau)) + 2\gamma u_{i+1}(e,\tau)^{\mathrm{T}}Rv_i(e,\tau)\Big)\mathrm{d}\tau,\end{aligned} \tag{20}$$

$$V^{u_i}(e(t_\ell),t_\ell) = \phi(e(t_\ell)). \tag{21}$$

Notice that (20)-(21) is a model-free version of (14), as it is independent of the functions $F,\ G$. However, we need to resort to approximation theory to solve it with respect to u_{i+1} and V^{u_i}. Particularly, we can use the Weierstrass approximation theorem [1] and deduce that $V^{u_i},\ u_i$ can be uniformly approximated on a compact set $\Omega \times [t_k,\ t_\ell] =: D$, with $\Omega \subset \mathbb{R}^n$. Then, we can express $V^{u_i},\ u_{i+1},\ \forall i \in \mathbb{N}$, as

$$V^{u_i}(e,t) = (w_i^v)^{\mathrm{T}}\psi^v(e,t) + \phi(e) + \epsilon_i^v(e,t), \tag{22a}$$

$$u_{i+1}(e,t) = (w_i^u)^{\mathrm{T}}\psi^u(e,t) + \epsilon_i^u(e,t), \tag{22b}$$

where $w_i^v \in \mathbb{R}^{N_v}$, $w_i^u \in \mathbb{R}^{N_u \times m}$ are weights, $\psi^v : \mathbb{R}^n \times [t_k,\ t_\ell] \to \mathbb{R}^{N_v}$, $\psi^u : \mathbb{R}^n \times [t_k,\ t_\ell] \to \mathbb{R}^{N_u}$ are basis functions and $\epsilon_i^v : \mathbb{R}^n \times [t_k,\ t_\ell] \to \mathbb{R}$, $\epsilon_i^u : \mathbb{R}^n \times [t_k,\ t_\ell] \to \mathbb{R}^m$ are the approximation errors. The approximation errors $\epsilon_i^v,\ \epsilon_i^u$ converge to zero, uniformly on D, as $N_v, N_u \to \infty$.

As w_i^v and w_i^u in (22) are unknown, we construct a critic and an actor neural network to approximate $V^{u_i},\ u_{i+1}$ as

$$\hat{V}^{u_i}(e,t) := (\hat{w}_i^v)^{\mathrm{T}}\psi^v(e,t) + \phi(e), \tag{23a}$$

$$\hat{u}_{i+1}(e,t) := (\hat{w}_i^u)^{\mathrm{T}}\psi^u(e,t), \tag{23b}$$

where $\hat{w}_i^v \in \mathbb{R}^{N_v}$, $\hat{w}_i^u \in \mathbb{R}^{N_u \times m}$ are the critic and the actor weights respectively, and $i \in \mathbb{N}$. Notice that a bias term has been introduced for the approximation of the value function in (22)-(23). Its purpose is to impose the boundary condition (21) to hold irrespectively of how the critic weights $\hat{w}_i^u$ are chosen, as long as the basis functions are appropriate.

Corollary 1. *Let* $\psi^v(0,t) = 0,\ \forall t \in [t_k,\ t_\ell]$, *and* $\psi^v(e,t_\ell) = 0,\ \forall e \in \mathbb{R}^n$. *Then,* $\forall i \in \mathbb{N}$, *it holds that:* $\hat{V}^{u_i}(e,t_\ell) = \phi(e)$, $\forall e \in \mathbb{R}^n$, *and* $\hat{V}^{u_i}(0,t) = 0,\ \forall t \in [t_k,\ t_\ell]$.

Consider now a number of time instances $\tau_j,\ j \in \{0,\ldots,N\} =: \mathcal{N}$ such that $t_k = \tau_0 < \tau_1 < \ldots < \tau_N = t_\ell$. Using the approximation (23), the left hand side of (20) for $t = \tau_j$ and $t + T = \tau_{j+1},\ j \in \mathcal{N}\backslash\{N\}$, is approximated as:

$$\begin{aligned}\hat{V}^{u_i}(e(\tau_{j+1}),\tau_{j+1}) - \hat{V}^{u_i}(e(\tau_j),\tau_j) = \phi(e(\tau_{j+1})) - \phi(e(\tau_j))\\ + (\hat{w}_i^v)^{\mathrm{T}}(\psi^v(e(\tau_{j+1}),\tau_{j+1}) - \psi^v(e(\tau_j),\tau_j)).\end{aligned} \tag{24}$$

In addition, the term $2\gamma u_{i+1}(e,\tau)^{\mathrm{T}}Rv_i(e,\tau)$ at the right hand side of (20) can be approximated using the actor as:

$$\begin{aligned}2\gamma\hat{u}_{i+1}(e,\tau)^{\mathrm{T}}R\hat{v}_i(e,\tau) &= 2\gamma\psi^u(e,\tau)^{\mathrm{T}}\hat{w}_i^u R\hat{v}_i(e,\tau)\\ &= 2\gamma\Big(\big(\hat{v}_i(e,\tau)^{\mathrm{T}}R\big) \otimes \psi^u(e,\tau)^{\mathrm{T}}\Big)\mathrm{vec}(\hat{w}_i^u),\end{aligned} \tag{25}$$

where $\hat{v}_i = u - \hat{u}_i$. Hence, the residual error created by approximating equation (20) through (24)-(25) is

$$e_{j,i} := \hat{V}^{u_i}(e(\tau_{j+1}),\tau_{j+1}) - \hat{V}^{u_i}(e(\tau_j),\tau_j) + \int_{\tau_j}^{\tau_{j+1}}\Big(\gamma Q(e)$$

Algorithm 1 Model-Free PI

```
Employ an arbitrary behavioral policy $u_b$ to the system (3), and collect input-state data online.
Let $u_0 \in \mathcal{U}$ be admissible, select $\epsilon > 0$ and set $i = 0$.
repeat
    Solve for $\hat{w}_i^v$ and $\hat{w}_i^u$ from equation (27) and $i = i+1$.
until $\left\|\hat{w}_i^v - \hat{w}_{i-1}^v\right\| < \epsilon$
Switch from $u_b$ to the learnt control policy $\hat{u}_i$.
```

$$+L(e)+\gamma S(\hat{u}_i(e,\tau))+2\gamma\hat{u}_{i+1}(e,\tau)^{\mathrm{T}}R\hat{v}_i(e,\tau)\Big)\mathrm{d}\tau,$$

which can be written as:

$$e_{j,i} = \Theta_{j,i}\hat{W}_i + \Psi_{j,i}, \tag{26}$$

where $\Theta_{j,i} := [\Theta_{j,i}^v \ \Theta_{j,i}^u]$, $\hat{W}_i := [(\hat{w}_i^v)^{\mathrm{T}} \ \mathrm{vec}(\hat{w}_i^u)^{\mathrm{T}}]^{\mathrm{T}}$, and

$$\Theta_{j,i}^v := \Big(\psi^v(e(\tau_{j+1}),\tau_{j+1}) - \psi^v(e(\tau_j),\tau_j)\Big)^{\mathrm{T}},$$

$$\Theta_{j,i}^u := \int_{\tau_j}^{\tau_{j+1}} 2\gamma\Big(\big(\hat{v}_i(e,\tau)^{\mathrm{T}}R\big)\otimes\psi^u(e,\tau)^{\mathrm{T}}\Big)\mathrm{d}\tau,$$

$$\Psi_{j,i} := \phi(e(\tau_{j+1})) - \phi(e(\tau_j)) + \int_{\tau_j}^{\tau_{j+1}} \Big(\gamma Q(e) + L(e) + \gamma S(\hat{u}_i(e,\tau))\Big)\mathrm{d}\tau.$$

If enough data is obtained along the system's trajectories, we can find $\hat{W}_i$ by least squares to minimize the error (26). To that end, we impose a standard assumption [17], [18].

Assumption 3. There exist $\delta > 0$ and $l_0 \in \mathcal{N}$, so that for all $l \geqslant l_0$ it holds that $\sum_{j=0}^{l}\Theta_{j,i}^{\mathrm{T}}\Theta_{j,i} \succ l\delta I_{N_v+mN_u}$. □

Given Assumption 3, the least squares solution to (26) is:

$$\hat{W}_i = -\Big(\sum_{j=0}^{l}\Theta_{j,i}^{\mathrm{T}}\Theta_{j,i}\Big)^{-1}\Big(\sum_{j=0}^{l}\Theta_{j,i}^{\mathrm{T}}\Psi_{j,i}\Big). \tag{27}$$

As a result, we can obtain the model-free version of PI, as shown in Algorithm 1. Its convergence is shown next.

Theorem 3. *Let Assumption 3 hold. Then, for all $\epsilon > 0$ there exist constants N_v^m, N_u^m, $i^\star \in \mathbb{N}$, such that if $N_v \geqslant N_v^m$ and $N_u \geqslant N_u^m$, then for all $(e,t) \in D$, $i \geqslant i^\star$, it holds that*

$$\left\|\hat{V}^{u_i}(e,t) - V^\star(e,t)\right\| \leqslant \epsilon, \ \ \|\hat{u}_{i+1}(e,t) - u^\star(e,t)\| \leqslant \epsilon.$$

Proof. We will provide only a sketch of the proof. For $i \in \mathbb{N}$, let $\tilde{V}^{u_i}$ be the value function of $\hat{u}_i$, where $\hat{u}_0 = u_0$, so that $\mathrm{LE}(\tilde{V}^{u_i},\hat{u}_i) = 0$, $\tilde{V}^{u_i}(0,t) = 0$, for all $t \in [t_k, \ t_\ell]$, and $\tilde{V}^{u_i}(e,t_\ell) = \phi(e)$, for all $e \in \Omega$. Let also $\tilde{u}_{i+1}(e,t) = -\frac{1}{2\gamma}R^{-1}G(e)^{\mathrm{T}}\nabla_e\tilde{V}^{u_i}(e,t)$, $\forall(e,t) \in D$. Then:

$$\tilde{V}^{u_i}(e(\tau_{j+1}),\tau_{j+1}) - \tilde{V}^{u_i}(e(\tau_j),\tau_j) = -\int_{\tau_j}^{\tau_{j+1}}\Big(\gamma Q(e) + L(e)+\gamma S(\hat{u}_i(e,\tau)) + 2\gamma\tilde{u}_{i+1}^{\mathrm{T}}(e,\tau)R\hat{v}_i(e,\tau)\Big)\mathrm{d}\tau \tag{28}$$

There exist $\tilde{w}_i^v \in \mathbb{R}^{N_v}$, $\tilde{w}_i^u \in \mathbb{R}^{N_u\times m}$ such that $\tilde{V}^{u_i}(e,t) = (\tilde{w}_i^v)^{\mathrm{T}}\psi^v(e,t)+\phi(e)+\tilde{\epsilon}_i^v(e,t)$ and $\tilde{u}_{i+1}(e,t) = (\tilde{w}_i^u)^{\mathrm{T}}\psi^u(e,t)+\tilde{\epsilon}_i^u(e,t)$. The approximation errors $\tilde{\epsilon}_i^v : \mathbb{R}^n \times [t_k, \ t_\ell] \to \mathbb{R}$, $\tilde{\epsilon}_i^u : \mathbb{R}^n \times [t_k, \ t_\ell] \to \mathbb{R}^m$ vanish uniformly on D as $N_v, N_u \to \infty$. Substituting these expressions in (28), we have:

$$0 = \Theta_{j,i}\tilde{W}_i + \Psi_{j,i} + E_{j,i}, \ \forall j \in \mathcal{N}, \ i \in \mathbb{N}, \tag{29}$$

where $\tilde{W}_i$=$[\tilde{w}_i^{v\mathrm{T}} \ \mathrm{vec}(\tilde{w}_i^u)^{\mathrm{T}}]^{\mathrm{T}}$ and $E_{j,i}$=$\tilde{\epsilon}_i^v(e(\tau_{j+1}),\tau_{j+1}) - \tilde{\epsilon}_i^v(e(\tau_j),\tau_j) + \int_{\tau_j}^{\tau_{j+1}} 2\gamma\tilde{\epsilon}_i^u(e,\tau)^{\mathrm{T}}R\hat{v}_i(e,\tau)\mathrm{d}\tau$. Hence, using the least-squares law (27), (29), Assumption 3 and the Weierstrass approximation theorem, one can show that for all $\epsilon > 0$ there exist $N_v^\star$, $N_u^\star > 0$, such that if $N_v \geqslant N_v^\star$, $N_u \geqslant N_u^\star$ then $\forall(e,t) \in D$ it holds that

$$|\hat{V}^{u_i}(e,t) - \tilde{V}^{u_i}(e,t)| \leqslant \|(\hat{w}_i^v - \tilde{w}_i^v)\|\,\|\psi_i^v(e,t)\| + |\tilde{\epsilon}_i^v(e,t)| \leqslant \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon, \tag{30}$$

$$\|\hat{u}_{i+1}(e,t) - \tilde{u}_{i+1}(e,t)\| \leqslant \|\hat{w}_i^u - \tilde{w}_i^u\|\,\|\psi_i^u(e,t)\| + \|\tilde{\epsilon}_i^u(e,t)\| \leqslant \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \tag{31}$$

Finally, an induction is used to derive the final result.

1) For $i = 0$, we have $\tilde{V}^{u_0} = V^{u_0}$ and $\tilde{u}_1 = u_1$. Hence, due to the uniform convergence (30)-(31), it follows that $\lim_{N_v,N_u\to\infty}\hat{V}^{u_0}(e,t) = V^{u_0}(e,t)$ and $\lim_{N_v,N_u\to\infty}\hat{u}_1(e,t) = u_1(e,t)$, uniformly on D.

2) Suppose that $\lim_{N_v,N_u\to\infty}\hat{V}^{u_{i-1}}(e,t) = V^{u_{i-1}}(e,t)$ and $\lim_{N_v,N_u\to\infty}\hat{u}_i(e,t) = u_i(e,t)$, uniformly on D, for some $i \in \mathbb{N}_+$. Then, using these assumptions along with Assumption 3 and (20)-(21), one can prove that $\lim_{N_u,N_v\to\infty}\tilde{V}^{u_i}(e,t) = V^{u_i}(e,t)$. Hence, since $|V^{u_i}(e,t)-\hat{V}^{u_i}(e,t)| \leqslant |V^{u_i}(e,t)-\tilde{V}^{u_i}(e,t)|+|\tilde{V}^{u_i}(e,t)-\hat{V}^{u_i}(e,t)|$, we can use the inductive assumption to conclude that, $\forall\epsilon > 0$, there exist $N_v^{\star\star}$, $N_u^{\star\star} > 0$ such that if $N_v \geqslant N_v^{\star\star}$, $N_u \geqslant N_u^{\star\star}$ then $\forall(e,t) \in D$, one has $|V^{u_i}(e,t)-\hat{V}^{u_i}(e,t)| \leqslant \epsilon$, which concludes the induction. The result follows using the triangular inequality and Theorem 2. ■

Remark 1. Due to Theorems 1 and 3, if N_v, N_u are large enough and γ is small, the closed-loop system eventually guarantees an ϵ-optimal safe timed transition between π_k and π_ℓ as per Def. 9. Since the aforementioned results apply for the transition among any pair of regions, we conclude that the closed-loop system eventually satisfies φ safely. □

VI. Simulations

We consider a two-link manipulator as in [25], with $q = [q_1 \ q_2]^{\mathrm{T}}$ being the angular positions (in rad) and $\dot{q} = [\dot{q}_1 \ \dot{q}_2]^{\mathrm{T}}$ the angular velocities (in rad/s), respectively. We also consider three regions of interest $\Pi = \{\pi_1,\ldots,\pi_3\}$ centered at $c_1 = [-0.2,-0.5,0,0]^{\mathrm{T}}$, $c_2 = [0.5,-0.4,0,0]^{\mathrm{T}}$, $c_3 = [0,0.2,0,0]^{\mathrm{T}}$, and a joint-state obstacle centered at $o_4 = [0.2,-0.46]^{\mathrm{T}}$, all with radius 0.05. Further, we consider $\mathcal{AP}$:=$\{$'1', '2', '3'$\}$ and $\mathcal{L}(\pi_i)$=$\{$'i'$\}$, $i \in \{1,\ldots,3\}$.

We impose a timed temporal logic task dictated by the formula $\varphi = \Box\Diamond_{[0,5]}$'i', $i \in \{1,2,3\}$, implying periodic visit to regions π_1, π_2, π_3 every 5 seconds; we also require avoidance of the obstacle, for which we compute L using o_4. By setting $\gamma(\cdot) = 5$ for all transitions in $\Pi\times\Pi$ in (5), and following the methodology of Section IV, we obtain the repetitive timed path $\mathsf{p} = \big[(\pi_1,5k+5)(\pi_2,5k+10)(\pi_3,5k+15)\big]^\omega$ for $k \in \{0,1,\ldots,\}$. We perform Alg. 1 by employing a sinusoidal behavioral policy u_b for 150 seconds, and then

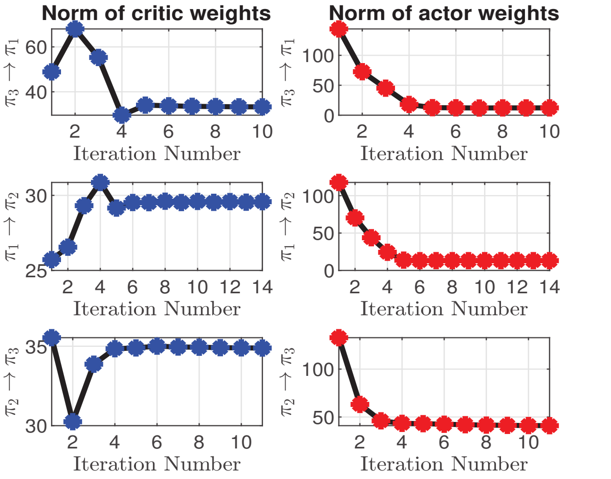

Fig. 1. Evolution of the Frobenius norms of the actor and the critic weights, as derived by Alg. 1.

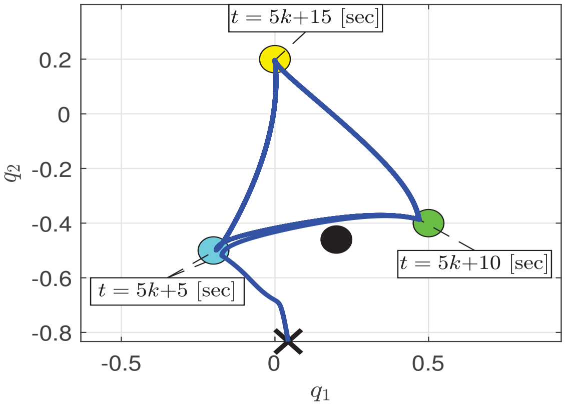

Fig. 2. Evolution of q after employing the policy given by Alg. 1.

executing the model-free PI by solving Eq. (27) iteratively. The evolution of the critic-actor weight norms during the execution of Alg. 1 are illustrated in Fig. 1, showing their convergence. After the passage of the 150 seconds, the policy derived by Alg. 1 substitutes the behavioral policy, and the resulting closed-loop trajectories for $t \geqslant 150$ [sec] can be seen in Fig. 2. It can be verified that the closed-loop system executes successfully the timed path, leading to the eventual satisfaction of φ. For all three repetitive OCPs, we chose $R{=}0.5I_2$, $\phi(e){=}Q(e){=}e^{\mathrm{T}}\left(\mathrm{diag}[20\ 20\ 10\ 10]\right)e$, $\gamma{=}0.1$.

VII. Conclusion

We develop a two-layered algorithm for the planning and control of unknown systems with timed temporal logic tasks. We design a novel data-driven control protocol that learns how to execute optimal timed transition between regions of the state-space, which guarantees the eventual satisfaction of the task. Future efforts will be devoted towards addressing continuous-time temporal tasks under the same framework.

References

[1] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[2] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[3] C. K. Verginis and D. V. Dimarogonas, “Timed abstractions for distributed cooperative manipulation,” *Autonomous Robots*, vol. 42, no. 4, pp. 781–799, 2018.

[4] S. Karaman and E. Frazzoli, “Vehicle routing problem with metric temporal logic specifications,” in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 3953–3958.

[5] C. C. Constantinou and S. G. Loizou, “Automatic controller synthesis of motion-tasks with real-time objectives,” in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 403–408.

[6] Z. Lin and J. S. Baras, “Metric interval temporal logic based reinforcement learning with runtime monitoring and self-correction,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 5400–5406.

[7] C. N. Mavridis, C. Vrohidis, J. S. Baras, and K. J. Kyriakopoulos, “Robot navigation under mitl constraints using time-dependent vector field based control,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 232–237.

[8] P. Varnai and D. V. Dimarogonas, “Prescribed performance control guided policy improvement for satisfying signal temporal logic tasks,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 286–291.

[9] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, “On the timed temporal logic planning of coupled multi-agent systems,” *Automatica*, vol. 97, pp. 339–345, 2018.

[10] A. Nikou, S. Heshmati-Alamdari, C. K. Verginis, and D. V. Dimarogonas, “Decentralized abstractions and timed constrained planning of a general class of coupled multi-agent systems,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 990–995.

[11] C. K. Verginis, C. Vrohidis, C. P. Bechlioulis, K. J. Kyriakopoulos, and D. V. Dimarogonas, “Reconfigurable motion planning and control in obstacle cluttered environments under timed temporal tasks,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 951–957.

[12] L. Lindemann and D. V. Dimarogonas, “Control barrier functions for signal temporal logic tasks,” *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2018.

[13] W. Xiao, C. A. Belta, and C. G. Cassandras, “High order control lyapunov-barrier functions for temporal logic specifications,” *arXiv preprint arXiv:2102.06787*, 2021.

[14] C. Sun and K. G. Vamvoudakis, “Continuous-time safe learning with temporal logic constraints in adversarial environments,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4786–4791.

[15] D. Muniraj, K. G. Vamvoudakis, and M. Farhood, “Enforcing signal temporal logic specifications in multi-agent adversarial environments: A deep q-learning approach,” in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4141–4146.

[16] K. G. Vamvoudakis and F. L. Lewis, “Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[17] Y. Jiang and Z.-P. Jiang, “Robust adaptive dynamic programming and feedback stabilization of nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 882–893, 2014.

[18] W. Gao and Z.-P. Jiang, “Learning-based adaptive optimal tracking control of strict-feedback nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2614–2624, 2017.

[19] R. Alur and D. L. Dill, “A theory of timed automata,” *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[20] P. Bouyer, N. Markey, J. Ouaknine, and J. Worrell, “The cost of punctuality,” in *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*. IEEE, 2007, pp. 109–120.

[21] C.-I. Vasile, D. Aksaray, and C. Belta, “Time window temporal logic,” *Theoretical Computer Science*, vol. 691, pp. 27–54, 2017.

[22] D. D’Souza and P. Prabhakar, “On the expressiveness of mtl in the pointwise and continuous semantics,” *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 1, pp. 1–4, 2007.

[23] J. Ouaknine and J. Worrell, “On the decidability of metric temporal logic,” in *20th Annual IEEE Symposium on Logic in Computer Science (LICS’05)*. IEEE, 2005, pp. 188–197.

[24] M. Abu-Khalaf and F. L. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach,” *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[25] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, “Approximate optimal trajectory tracking for continuous-time nonlinear systems,” *Automatica*, vol. 51, pp. 40–48, 2015.