Intermittent Learning Through Operant Conditioning for Cyber-Physical Systems

Prachi Pratyusha Sahoo, *Member, IEEE*, Aris Kanellopoulos, *Member, IEEE*, and Kyriakos G. Vamvoudakis, *Senior Member, IEEE*

***Abstract*— This article presents a novel scheme, namely, an intermittent learning scheme based on Skinner's operant conditioning techniques that approximates the optimal policy while decreasing the usage of the communication buses transferring information. While traditional reinforcement learning schemes continuously evaluate and subsequently improve, every action taken by a specific learning agent based on received reinforcement signals, this form of continuous transmission of reinforcement signals and policy improvement signals can cause overutilization of the system's inherently limited resources. Moreover, the highly complex nature of the operating environment for cyber-physical systems (CPSs) creates a gap for malicious individuals to corrupt the signal transmissions between various components. The proposed schemes will increase uncertainty in the learning rate and the extinction rate of the acquired behavior of the learning agents. In this article, we investigate the use of fixed/variable interval and fixed/variable ratio schedules in CPSs along with their rate of success and loss in their optimal behavior incurred during intermittent learning. Simulation results show the efficacy of the proposed approach.**

***Index Terms*— Cyber-physical systems (CPSs), intermittent learning, operant conditioning, Q-learning.**

I. Introduction

INTERCONNECTED computational subsystems that control physical devices interacting with the operating environment make up a class of platforms called cyber-physical systems (CPSs) [1]. Implementing decision-making mechanisms aided by artificial intelligence (AI) constitutes a significant research endeavor that aims to render CPS autonomous. Various military and civilian domains, such as aerospace [2], healthcare [3], [4], transportation systems [5], network security [6], human/robot interaction [7], and the Internet [8], find applications for CPS. Subsequently, safety and efficiency concerns in the realm of CPS arise due to their exposure to the human environment in all its complexities. A major challenge that the AI industry faces today deals with the integration and adaptation of the "closed-world" laboratory solutions into a more volatile and unpredictable "open-world."

It is evident that incorporating reinforcement learning (RL) algorithms into the decision-making procedures shall allow the CPS to unlock higher levels of smart autonomy, adaptivity, and self-governance. Such algorithms in combination with neuroscience concepts attempt, at a higher level of intelligence, to aid data sharing and decision making [9]. Data sharing and decentralized deployment of actions introduce "modularity," a double-edged sword, in CPS [10]. While decentralized learning and deployment remain desirable, they also increase the number of sensors and actuators interacting with each other, and the communication network topology modeling this exchange of information, thus fueling the added complexity of such CPS platforms. Therefore, existing learning algorithms create potential gaps for overutilization of resources, loss in optimal behavior and learning strategies, and increased vulnerability to malicious agents [11].

Continuous communication and data sharing between the sensors, actuators, and learning subsystems, a typical assumption in learning algorithms, leads to degradation of communication effectiveness, and increased data-stream exposure for adversarial agents to exploit and perform a successful attack. This, along with a discrete action space and state space limits smart, efficient, and secure autonomy due to an infinite bandwidth requirement, lack of robustness in learning subsystems, incomplete knowledge of the operating environment, and inability to detect network compromises.

Behavioral scientists have validated the need for intermittent data sharing in learning tasks [12]. They have shown that the central nervous system in human beings minimizes effort and sorts through impulses and stimuli by maintaining intermittent signaling. Specifically, the spinal cord transmits a channel of information and effectively exploits its neural resources via intermittent strategies to produce a sequence of muscle-bone interactions that induce movement [13]–[15]. Learning of tracking movements works similarly via "step-hold" strategies employing threshold-based rules [12], [16], and leveraging of analogous mechanisms to explain the learning of rhythmic movements [17], [18]. Some of those intermittency principles have been utilized in the context of control systems as well. Vamvoudakis *et al.* [19] employ an event-triggered controller,

Manuscript received November 7, 2020; revised April 21, 2021 and July 20, 2021; accepted September 3, 2021. This work was supported in part by the National Science Foundation under Grant S&AS-1849264 and Grant CPS-1851588, in part by the Office of Naval Research (ONR) Minerva under Grant N00014-18-1-2874, and in part by the Army Research Office (ARO) under Grant W911NF-19−1 − 0270. *(Corresponding author: Prachi Pratyusha Sahoo.)*

Prachi Pratyusha Sahoo is with the Woodruff School for Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30318 USA (e-mail: prachi@gatech.edu).

Aris Kanellopoulos and Kyriakos G. Vamvoudakis are with the Daniel H. Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30318 USA (e-mail: ariskan@gatech.edu; kyriakos@gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3111826.

Digital Object Identifier 10.1109/TNNLS.2021.3111826

informed by an RL algorithm, to achieve target tracking. Xie *et al.* [20] bring those ideas to a fuzzy system via the design of an event-triggered scheduler. Similarly, Zhang and Yang [21] propose an intermittent updating scheme for an adaptive controller with the goal of alleviating the computational burden to the system. Finally, those event-triggered ideas have been used with H_∞ controllers in [22] alongside an experience replay-inspired concurrent learning framework.

This article suggests a novel scheme in learning theory and methodology that exploits the principles of operant conditioning [23] to alleviate the load on communication channels of a CPS. Scheduling methods in the context of CPS discussed in this article evolve with continuous dynamics in an adversarial environment. The introduction of interval-based reinforcement signals over predetermined "fixed intervals" and stochastic "variable intervals" pave the path for the introduction of ratio-based methods, as discussed in operant conditioning. Ratio-based methods receive reinforcement signals from the environment upon performing a predetermined or stochastic number of positive actions, namely, the "fixed ratio" and the "variable ratio" schedules, respectively.

II. Background on Classical Reinforcement Learning

Consider a CPS evolving as follows:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad t \geq 0, \ x(0) = x_0 \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector of the system, $u \in \mathbb{R}^m$ is the control input or decision vector of the agent, and $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the plant and input matrices, respectively.

To quantify the agent's objectives, we shall define an appropriate performance index that will serve as the reward, given by

$$\mathcal{J}(x(0); u) \equiv \frac{1}{2}\int_0^\infty \left(x^{\mathrm{T}}Mx + u^{\mathrm{T}}Ru\right)d\tau \tag{2}$$

where $M \succeq 0$ and $R \succ 0$ are user defined matrices of appropriate dimensions. The autonomous agent wishes to find the policy $u^\star(t)$ that optimizes the CPS operation in the sense that $\mathcal{J}(x(0); u^\star(t)) \leq \mathcal{J}(x(0); u(t))$, $\forall u$.

In order to derive a feedback rule that enables the agent to extract decision policies autonomously, our ultimate goal is to find the value function

$$\mathcal{V}^\star(x) = \min_u \frac{1}{2}\int_t^\infty \left(x^{\mathrm{T}}Mx + u^{\mathrm{T}}Ru\right)d\tau \quad \forall x, \ t \geq 0 \tag{3}$$

that evaluates the cost-to-go from any given initial state x of the CPS.

Furthermore, we can express the Hamiltonian function associated with (3) and (1) as

$$\mathcal{H}\left(x, u, \frac{\partial \mathcal{V}}{\partial x}\right) = \frac{1}{2}x^{\mathrm{T}}Mx + \frac{1}{2}u^{\mathrm{T}}Ru + \frac{\partial \mathcal{V}^{\mathrm{T}}}{\partial x}(Ax + Bu) \ \forall x, u.$$

According to [24], the optimal value function $\mathcal{V}^\star(x)$ satisfies the following Hamilton–Jacobi–Bellman (HJB) equation:

$$\min_u \mathcal{H}\left(x, u, \frac{\partial \mathcal{V}^\star}{\partial x}\right) = 0. \tag{4}$$

Solving for the optimal policy $u^\star$ yields

$$u^\star(x) = -R^{-1}B^{\mathrm{T}}\frac{\partial \mathcal{V}^\star}{\partial x} \quad \forall x. \tag{5}$$

Finally, it is known that for a functional (2) and system given by (1), the optimal value function $\mathcal{V}^\star(x)$ takes a quadratic in the states form, i.e., $\mathcal{V}^\star(x) = x^{\mathrm{T}}Px$ and $P \succ 0, \quad \forall x$. Direct substitution into (4) yields the following Riccati equation:

$$A^{\mathrm{T}}P + PA + M - \mathrm{PBR}^{-1}B^{\mathrm{T}}P = 0. \tag{6}$$

Even though the problem of deriving the solution of (6) has been investigated before [25], the solution of the nonlinear optimal decision making problem still suffers from various facets of the *curse of dimensionality* introduced by Bellman [24]. In the continuous-time state and action case, this suffers from the difficulties arising in analytically solving (4) without any knowledge of the system (1).

To solve the optimal feedback problem without the need for explicit knowledge of the system matrices, we will employ approximation-based RL methods. In order to derive approximate solutions to (4) and (5), we employ two distinct approximation structures, thus expressing the unknown value function and optimal policy as linear in the parameters inside a simply connected, compact subset $\Omega \subseteq \mathbb{R}^n$, $\mathcal{V}^\star(x) = \theta_c^{\star\mathrm{T}}\phi(x) + \epsilon$, and $u^\star = \theta_u^{\star\mathrm{T}}\phi_u(x) + \epsilon_u, \ \forall x \in \Omega$, respectively, where $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^l$ is the critic basis, $\phi_u(\cdot) : \mathbb{R}^n \to \mathbb{R}^{l_u}$ the actor basis, $\theta_c^\star \in \mathbb{R}^l$ the optimal critic weights, $\theta_u^\star \in \mathbb{R}^{l_a}$ the optimal actor weights, and ϵ and ϵ_u the approximation errors of the critic and the actor, respectively. However, since our system (1) is linear and the cost (2) is quadratic, we can employ approximation in the whole space $\Omega \equiv \mathbb{R}^n$ and the approximation errors will be zero.

In order to assess the current approximation of the critic and the actor weights, we introduce the critic error signal

$$e(t) = \mathcal{H}\left(x, \hat{u}, \frac{\partial \hat{\mathcal{V}}}{\partial x}\right) \ \forall x$$

and the actor error signal

$$e_u(t) = \hat{\theta}_u^{\mathrm{T}}\phi_u - \arg\min_u \mathcal{H}\left(x, \hat{u}, \frac{\partial \hat{\mathcal{V}}}{\partial x}\right) \ \forall x$$

respectively.

Once the appropriate reinforcement signals for the approximation schemes have been transmitted from the environment to the agent, a number of different parameter estimation algorithms can be employed to update the weights $\hat{\theta}_c$ and $\hat{\theta}_u$.

Throughout this work, we will employ a modified and normalized gradient descent algorithm, which minimizes the squared normed errors, $E_c(t) = \|e(t)\|^2$ and $E_u(t) = e_u^{\mathrm{T}}(t)e_u(t)$. Thus, the update laws are computed as follows for the critic weight estimates $\forall t \geq 0$:

$$\dot{\hat{\theta}}_c(t) = -\alpha_c \frac{1}{\left(1 + \sigma^{\mathrm{T}}\sigma\right)^2}\frac{\partial E_c}{\partial \hat{\theta}_c} = -\alpha_c \frac{\sigma}{\left(1 + \sigma^{\mathrm{T}}\sigma\right)^2}e_c \tag{7}$$

where $\sigma(t) = \phi(t) - \phi(t - T)$ and $\alpha_c \in \mathbb{R}^+$ is a tuning parameter that determines the speed of convergence.

Similarly, the update law for the actor weight estimate will be $\forall t \geq 0$

$$\begin{aligned}\dot{\hat{\theta}}_u &= -\alpha_a \frac{1}{\left(1+\phi_u^{\mathrm{T}}(t)\phi_u(t)\right)^2}\frac{\partial E_u}{\partial \hat{\theta}_u} \\ &= -\alpha_u \frac{\phi_u(t)}{\left(1+\phi_u^{\mathrm{T}}(t)\phi_u(t)\right)^2} e_u^{\mathrm{T}} \end{aligned} \tag{8}$$

where $\alpha_u \in \mathbb{R}^+$ is a tuning gain that determines the speed of convergence.

To obfuscate the need for partial knowledge of the system dynamics which arises in formulations involving linear and control-affine systems, Vamvoudakis [26] introduced a Q-learning algorithm for continuous-time, continuous-state, and action problems. A state-action dependent function, i.e., Q-function, which maps both the current state and the decision vector to the cost-to-go, is constructed as follows:

$$\begin{aligned} Q(x,u) &:= \mathcal{V}^\star(x) + \mathcal{H}\left(x, u, \frac{\partial \mathcal{V}^\star}{\partial x}\right) \\ &= \mathcal{V}^\star(x) + \frac{1}{2}x^{\mathrm{T}}P(Ax+Bu) + \frac{1}{2}(Ax+Bu)^{\mathrm{T}}Px \\ &\quad + \frac{1}{2}x^T Mx + \frac{1}{2}u^{\mathrm{T}}Ru \quad \forall x,u. \end{aligned}$$

The value function and the Q-function achieve the same minimum value. In compact form, the Q-function becomes

$$\begin{aligned} Q(x,u) &= \frac{1}{2}U^{\mathrm{T}}\begin{bmatrix} P+M+PA+A^{\mathrm{T}}P & PB \\ B^{\mathrm{T}}P & R \end{bmatrix}U \\ &:= \frac{1}{2}U^{\mathrm{T}}\begin{bmatrix} Q_{\mathrm{xx}} & Q_{\mathrm{xu}} \\ Q_{\mathrm{ux}} & Q_{\mathrm{uu}} \end{bmatrix}U := \frac{1}{2}U^{\mathrm{T}}\bar{Q}U \quad \forall x,u \end{aligned}$$

where $U = [x^{\mathrm{T}}\ u^{\mathrm{T}}]^{\mathrm{T}}$ is the augmented state-action regression vector. The compact Q-function, specifically the matrix $\bar{Q}$, comprises of data-driven basis weights, activated with states and policies, encoded in the critic weight vector.

The construction of the basis, which is shown in Fig. 1(a), allows complete approximation of the Q-function over finite time intervals of duration $T \in \mathbb{R}^+$. Each arrow corresponds to a critic weight; these critic weights have dynamics that capture the integral Bellman error [26] in appropriately approximating the value function. We ensure this by setting the following:

$$\begin{aligned} \hat{Q}(x,u) &= \hat{\theta}_c^{\mathrm{T}}(U \otimes U) \\ e_c &:= \hat{Q}(x(t),u(t)) - \hat{Q}(x(t-T),u(t-T)) \\ &\quad + \frac{1}{2}\int_{t-T}^{t}\left(x^{\mathrm{T}}Mx + u^{\mathrm{T}}Ru\right)dt \\ e_c &= \hat{\theta}_c^{\mathrm{T}}(U(t)\otimes U(t)) - \hat{\theta}_c^{\mathrm{T}}(U(t-T)\otimes U(t-T)) \\ &\quad + \frac{1}{2}\int_{t-T}^{t}\left(x^{\mathrm{T}}Mx + u^{\mathrm{T}}Ru\right)dt \qquad (9) \\ e_u &:= \hat{\theta}_u^{\mathrm{T}}x + \hat{Q}_{\mathrm{uu}}^{-1}\hat{Q}_{\mathrm{ux}}x. \qquad (10) \end{aligned}$$

Algorithm 1 presented next describes the Q-learning framework with a "classical" RL structure.

Static Event-Triggering Q-Learning: The learning method presented earlier, depends on a continuous stream of measurements and updates from the environment that inform the evolution of the critic and the actor approximators. The problems arising from applying such continuous-time update laws in complex systems, where the communication channels have finite resources, have been extensively investigated [27]–[29]. In the context of Q-learning, there exist event-triggered mechanisms to inform actor updates, instead of a real-time update schedule. To this end, [30] suggests that the actor subsystem samples states in an event-triggered manner as:

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

where $\{r_j\}$ is a monotonically increasing sequence of time stamps to sample state measurements. In the static event-triggering framework, those instances are derived as

$$r_j = \min\{t \mid 0 \leq t \leq r_{j-1},\ p(t) \leq 0\}$$

where the triggering signal $p(t)$ is

$$p(t) := \left(1-\beta^2\right)\underline{\lambda}(M)\|x\|^2 + \underline{\lambda}(R)\|\hat{W}_a^{\mathrm{T}}x\|^2 - L^2\bar{\lambda}(R)\|e\|^2$$

with $\beta \in (0,1)$ and L a Lipschitz of the feedback control mapping. The reader is referred to [31] for details on the framework.

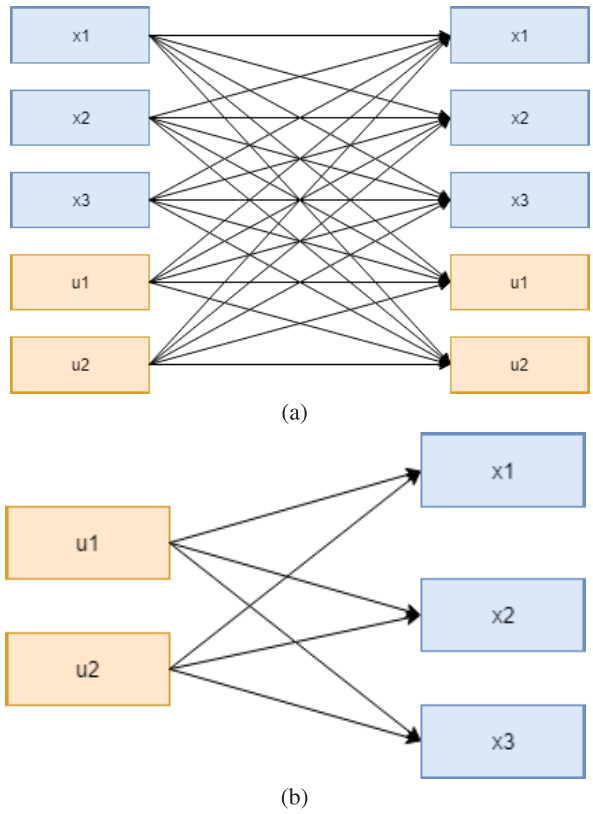


Fig. 1. Blue boxes in (a) stand for the states in the system while the orange one shows the control inputs. The arrows summarize the basis described by all combinations of the three states and two control inputs. Whereas, actor weights indicated by the arrows in (b) establish connections between each state (in blue) and each control input (in orange). The relationship captured by the arrows stands for a single actor weight. There are mn actor weights if there are n states and m control inputs. (a) Construction of the polynomial basis employed in continuous state Q-learning. (b) Activation functions for Actor Weights.

Algorithm 1 Continuous Update in the Actor-Critic Framework

Given initial state x_0, initial critic weights $\hat{\theta}_c$, initial actor weights $\hat{\theta}_u$,
procedure
  Propagate t, $x(t)$ using (1).
  Compute $u(t) = -\hat{K}x(t)$ where $\hat{K} = -\hat{\theta}_u^{\mathrm{T}}$.
  if $t < T_{\text{exp}}$
    Add probing noise $u(t) \leftarrow u(t) + u_{\text{PE}}(t)$
  end if
  Propagate $\hat{\theta}_c$ and $\hat{\theta}_u$ ($\dot{\hat{\theta}}_c$ and $\dot{\hat{\theta}}_u$ according to update laws as in (7) and (8) respectively).
  Estimate error in the Critic weights, e_c, and Actor weights, e_u, as in (9) and (10) respectively.
  if $e_u \neq 0$ and $e_c \neq 0$
    Go to step 8
  end if ▷ $e_u \approx 0$ and $e_c \approx 0$
end procedure

In this case, the actor updates follow:

$$\hat{\theta}_u^+ = \begin{cases} 0, & \forall t \in (r_j, r_{j+1}] \\ \hat{\theta}_u(r_{j-1}) - \alpha_u \dfrac{\phi_u}{\left(1+\phi_u^{\mathrm{T}}\phi_u\right)^2} e_u^{\mathrm{T}}(r_j), & t = r_j. \end{cases} \tag{11}$$

This procedure is described in Algorithm 2

Algorithm 2 Event-Triggered Updates in the Actor-Critic Framework

Given initial state x_0, initial critic weights $\hat{\theta}_c$, initial actor weights $\hat{\theta}_u$,
procedure
  Propagate t, $x(t)$ using (1) $\forall t \in (r_j, r_j + 1]$.
  if If $t = r_j$, then set $\hat{x}(t) := x(t)$, $\forall t \in (r_j, r_j + 1]$.
  else $\hat{x}(t) = x(r_j)$, $\forall t \in (r_j, r_j + 1]$ and use the previous control input with zero-order-hold.
  end if
  Compute $u(t) = -\hat{K}x(t)$ where $\hat{K} = -\hat{\theta}_u^{\mathrm{T}}$.
  if $t < T_{\text{exp}}$
    Add probing noise $u(t) \leftarrow u(t) + u_{\text{PE}}(t)$
  end if
  Propagate $\hat{\theta}_c$ and $\hat{\theta}_u$ according to update laws as in (7) and (11) respectively.
  Estimate error in the Critic weights, e_c, and Actor weights, e_u, as in (9) and (10) respectively.
  if $e_u \neq 0$ and $e_c \neq 0$
    Go to step 8
  end if ▷ $e_u \approx 0$ and $e_c \approx 0$
end procedure

III. Intermittent Reinforcement Learning

In this section, we aim to construct intermittent learning frameworks for critic networks, unlike the algorithms presented in Section II, wherein agents will remain connected through the cloud. The existence of a centralized processing unit within the cloud will enable improved utilization of the network resources. Acting as a supervising mechanism, we design intermittent rules that are implemented in the cloud and determine the transmission time, the transmission content, or both, on the data from which the agent will learn.

Remark 1: While previous work on reduction of transmission loads for learning-enabled CPS has focused on event-triggered updates for the actor and the controller in an effort to stabilize the system [19], [30], [31], the proposed intermittent learning framework aims to optimally use environmental measurements to facilitate expedited learning and, therefore, introducing discontinuities in the critic's tuning mechanism, instead of the actor's. □

Toward constructing the intermittent RL framework, we turn to the theory of operant conditioning which originated in Skinner's [23], [32], [33] work. During his experiments, Skinner [34] reported the emergence of intermittent learning, which, furthermore, could be scheduled via different methods. Such learning is used instead of continuous reinforcement once the desired response is conditioned by continuous reinforcement and the reinforce-er seeks to reduce or eliminate the number of reinforcements necessary to encourage the intended response and to slow extinction. Skinner found that continuous reinforcement in the early stages of training seems to increase the rate of learning.

A. Schedules of Intermittent Learning

In this section, we will introduce the four different schedules proposed in the framework of operant conditioning.

1) Fixed Interval Schedule: This update schedule focuses on rewarding the training agent every fixed interval of time. This allows for the agent to expect a reward every fixed interval of time despite the occurrence of negative behaviors, given at least one occurrence of desirable behavior occurs in the time interval. The training agent learns the optimal behavior slower than the continuous reinforcement schedule but retains established optimal behavior for longer, thus making it resistant to temporary changes in the operating environment.

Through this schedule, the supervising cloud transmits the reinforcement signal to the agent after a set amount of time. In describing this approach, we define the transmission time vector, given by $\{t_j\}_{j=0}^{\infty}$, where t_j is the j-th consecutive sampling instant that satisfy $0 \leq t_0 < t_1 < \cdots < t_j < \cdots$ and $\lim_{j\to\infty} t_j = \infty$.

This way, the continuously evolving agent, utilizes a zero-order hold (ZOH) structure on the weight estimates, resulting in the overall hybrid dynamics combining the continuous-time interaction of the agent with the environment (1), and the intermittent update laws

$$\begin{cases} \dot{\hat{\theta}}_c = 0, & t \neq t_j \\ \hat{\theta}_c^+ = \hat{\theta}_c(t_{j-1}) - \alpha_c \dfrac{\sigma}{\left(1+\sigma^{\mathrm{T}}\sigma\right)^2} e_c^{\mathrm{T}}(t_j), & t = t_j \end{cases} \tag{12}$$

for the critic, and similar for the actor, the continuous update law given by

$$\dot{\hat{\theta}}_u = -\alpha_u \frac{\phi_u}{\left(1+\phi_u^{\mathrm{T}}\phi_u\right)^2} e_u^{\mathrm{T}} = -\alpha_u x e_u^{\mathrm{T}}.$$

Remark 2: It can be seen that while we consider intermittent updates in the critic network, the actor operates in a continuous fashion. It is possible to have both networks update with specific triggering schedules; however, for ease of exposition, we retain a simple actor update law, so that we are able to focus on the different critic schedules; which constitute the main contribution of our work. There is a large body of literature in event-triggered RL algorithms where the actor-network is updated intermittently [19], [22], [31], [35], whereas this is not the case for the critic. □

Note that in a fixed interval schedule the reinforcement signal is transmitted—and, by extension, the weight estimates are updated—over predetermined intervals, such that the update instances $\{t_j\}$ satisfy $\Delta t = t_{j+1} - t_j$, $\forall j \in \{1, 2, \ldots\}$, where Δt is the value of the fixed interval. Algorithm 3 describes the learning method of an autonomous agent under fixed interval operant conditioning.

In practice, the algorithm used is outlined in Fig. 2(a) wherein the critic updates every $t_j = 2$ s, and in the meantime, we employ a ZOH mechanism to mitigate the tuning action of the critic weights.

2) Variable Interval Schedule: According to this scheduling approach, the supervising cloud transmits the reinforcement signal to the agent according to a time-based stochastic triggering mechanism. Rewarding on the variable interval reinforcement schedule occurs at specific timestamps that follow a uniform, a normal, and an exponential distribution. The use of different distributions allows for varied learning times and better retention of optimal behavior over time. The triggering instances $\{t_j\}$, $j \in \{1, 2, \ldots\}$, are such that the algorithm selects a schedule with variable update time stamps $t_j \in [0, T_{\text{exp}}]$. At these specific times, the critic weights update according to steepest gradient descent tuning laws in (7), meanwhile not updating at other times. These timestamps are subject to change based on the type of distribution used. For example, the exponential distribution has a higher likely hood to pick values near T_{exp}, while the normal distribution will likely concentrate update timestamps in the middle of the exploration window. Algorithm 3 describes this procedure that is visualized in Fig. 2.

3) Fixed Ratio Schedule: A fixed ratio update schedule reinforces every time a desirable or optimal action takes place a fixed number of times. These updates encourage the training agent to perform the desired task many times in order to earn a reward or update the critic. This reinforcement method ensures a very **fast response rate**. Updates occur more often, and upon exploring portions of the state space that reduce the cost. To formulate ratio-based schedules in control systems, it is imperative to define an appropriate metric that differentiates desirable behaviors from undesirable ones is integral to intermittent reward generation. The metric proposed in this work involves the use of a buffer that collects information on the number of desired behaviors. Once the buffer has stored a predetermined number of behaviors, the supervising cloud transmits the appropriate reinforcement signal. The criteria were chosen to be met to qualify as "**desirable action**" is

Algorithm 3 Fixed/Variable Interval Update Schedule for Operant Conditioning in Actor-Critic Framework

1: Given initial state x_0, initial critic weights $\hat{\theta}_c$, initial actor weights $\hat{\theta}_u$, and
   1) t_{up} = updateInterval for fixed interval update schedule, and
   2) sequence of t_{up} = [updateIntervals] for variable interval update schedule
2: **procedure**
3: Propagate t, $x(t)$ using (1).
4: Compute $u(t) = -\hat{K}x(t)$ where $\hat{K} = -\hat{\theta}_u^{\text{T}}$.
5: **if** $t < T_{\text{exp}}$
6: Add probing noise $u(t) \leftarrow u(t) + u_{\text{PE}}(t)$
7: **end if**
8: **while** mod $(t, t_{\text{up}}) \approx 0$
9: Propagate $\hat{\theta}_c$ using the update law for $\dot{\hat{\theta}}_c$ as in (12) for 0.05 s.
10: **end while**
11: Propagate $\hat{\theta}_u$ using the update law for $\dot{\hat{\theta}}_u$ as in (16) for all time t.
12: Estimate error in the Critic weights, e_c, and Actor weights, e_u, as in (9) and (10) respectively.
13: **if** $e_u \neq 0$ and $e_c \neq 0$
14: Go to step 8
15: **end if** ▷ $e_u \approx 0$ and $e_c \approx 0$
16: **end procedure**

given by

$$\Sigma = \hat{\theta}_c^{\text{T}}\sigma \tag{13}$$

estimates how well the current estimate of the critic weights evaluates the value function between two points of time in the exploration window.

The resulting hybrid system has the following form:

$$\begin{cases} \dot{\hat{\theta}}_c = 0 \\ \hat{\theta}_c^{+} = \hat{\theta}_c(t_{j-1}) - \alpha_c \dfrac{\sigma}{\left(1 + \sigma^{\text{T}}\sigma\right)^2} e_c^{\text{T}}(t_j), \quad \text{when } \Sigma \\ \text{is negative for a fixed number of actor updates} \end{cases}$$

$$\dot{\hat{\theta}}_u = -\alpha_u \frac{\phi_u}{\left(1 + \phi_u^{\text{T}}\phi_u\right)^2} e_u^{\text{T}} = -\alpha_u x e_u^{\text{T}} \tag{14}$$

where $\alpha_u \in \mathbb{R}^+$ is a tuning gain. In the fixed ratio update schedule, the reinforcement signal enables critic weight updates when Σ remains negative for a fixed number of times, e.g., 75. After updating the critic weights, we go back to employing the ZOH until Σ criteria is met again 75 times, as outlined in Fig. 3 and descriptively showcased in Algorithm 4.

Remark 3: For computational purposes, in Algorithm 4, the critic update occurs for $\delta t = 0.05$ s. Theoretically, this jump in critic tuning law occurs "instantaneously;" however, to realize this using Runge–Kutta methods for numerical integration, we must provide some buffer time δt. □

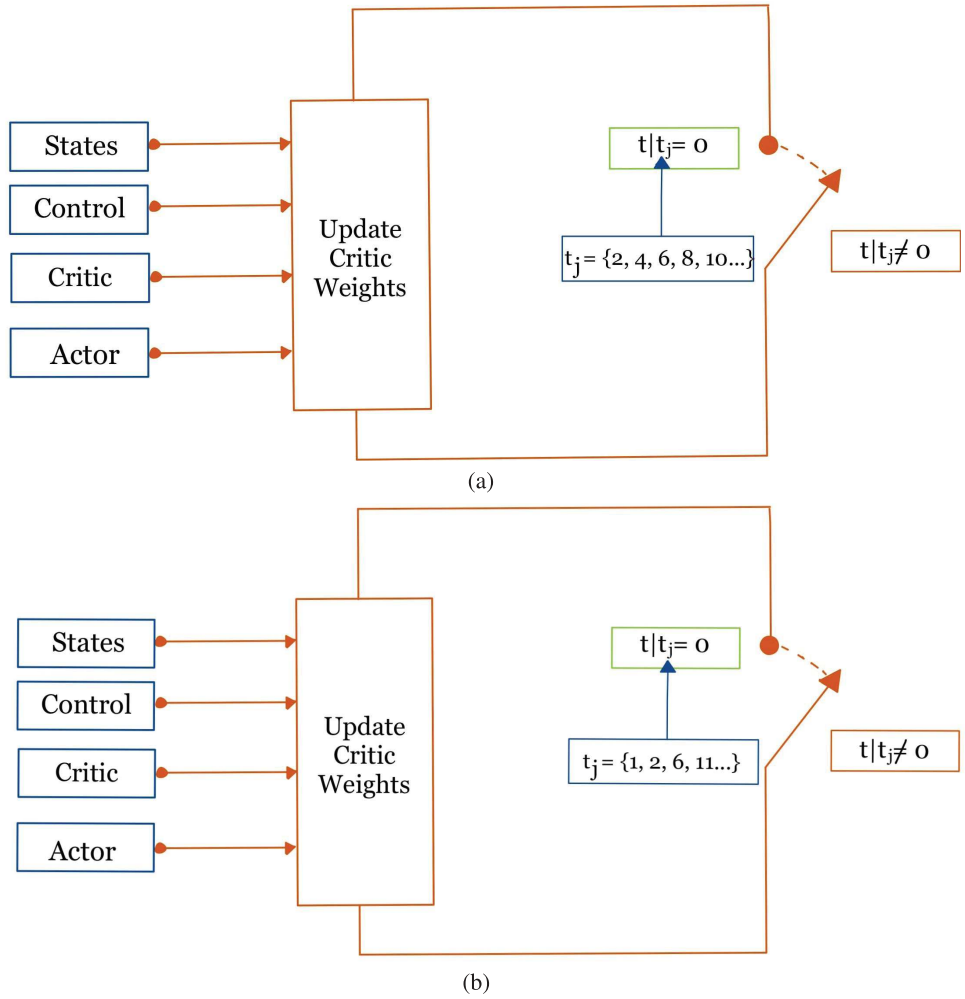


Fig. 2. Flowchart visualizing the interval-based reinforcement schedules. (a) Fixed interval. (b) Variable interval.

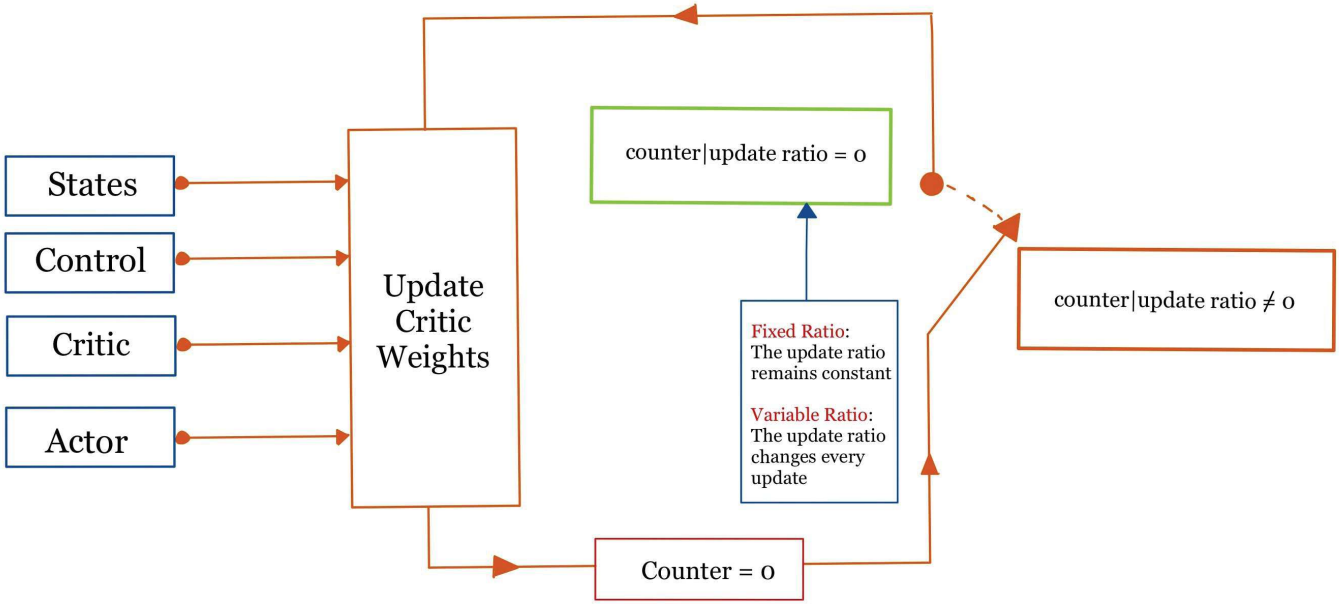


Fig. 3. Flowchart visualizing the fixed ratio and variable ratio update schedule implementations.

4) Variable Ratio Schedule: Variable Ratio, the most unpredictable rewarding policy, generates the **fastest response rate** while also ensuring **slowest extinction rate** by embedding a stochastic process to the transmission algorithm. Once again, the criterion for a **desirable action**, given by (13), estimates quality of value-function evaluation by critic weights to trigger

Algorithm 4 Fixed/Variable Ratio Update Schedule for Operant Conditioning in Actor-Critic Framework

Given initial state x_0, initial critic weights $\hat{\theta}_c$, initial actor weights $\hat{\theta}_u$, and
  1) update ratio R_{up} for fixed ratio update schedule, and
  2) vector of randomized update ratios $R_{\text{rand}}(t)$ for variable ratio update schedule.
procedure
  Propagate t, $x(t)$ using (1).
  Compute $u(t) = -\hat{K}x(t)$ where $\hat{K} = -\hat{\theta}_u^{\text{T}}$.
  if $t < T_{\text{exp}}$
    Add probing noise $u(t) \leftarrow u(t) + u_{\text{PE}}(t)$
  end if
  while $\text{sgn}(\hat{\theta}_c^{\text{T}}(U(t) \otimes U(t) - U(t-T) \otimes U(t-T))) < 0$
    counter $\leftarrow$ counter $+ 1$
    if counter $= R_{\text{rand}}(t)$ for variable ratio and counter $= R_{\text{up}}$ for fixed ratio updates
      Propagate $\hat{\theta}_c$ using the update law for $\dot{\hat{\theta}}_c$ as in (14) for 0.05 s.
      counter $\leftarrow 0$
    end if
  end while
  Propagate $\hat{\theta}_u$ using the update law for $\dot{\hat{\theta}}_u$ as in (16) for all time t.
  Estimate error in the Critic weights, e_c, and Actor weights, e_u, as in (9) and (10) respectively.
  if $e_u \neq 0$ and $e_c \neq 0$
    Go to step 8
  end if ▷ $e_u \approx 0$ and $e_c \approx 0$
end procedure

future updates. The critic network weights, which update every time Σ, remain negative for a variable number of times, dictated by a uniform, a normal, and an exponential randomizer.

The algorithm, visualized in Fig. 3 and outlined in Algorithm 4, represents the critic update procedure undertaken to implement this reinforcement schedule, with an optimal policy given as

$$u^{\star}(x) = \arg\min_{u} Q(x,u) = -Q_{\text{uu}}^{-1} Q_{\text{ux}} x. \tag{15}$$

Actor weights populate the feedback gain matrix, derived from the Q-function upon employing the stationary condition, in (15). These weights correspond to each arrow shown in Fig. 1. The relationship between the states and the actor weight matrix, given by $\hat{u}(x) = \hat{\theta}_u^{\text{T}}(x)$ induced the actor-error $e_u := \hat{\theta}_u^{\text{T}} x + \hat{Q}_{\text{uu}}^{-1} \hat{Q}_{\text{ux}} x$.

The tuning law employed to mitigate the error in (10) between the actor weights approximating the feedback gain matrix and that obtained from the stationary condition applied to the Q-function, ensures that the actor weights tune appropriately to produce the optimal policy. This tuning law, given by

$$\dot{\hat{\theta}}_u = -\alpha_u \frac{\partial\left(\frac{1}{2}\|e_u\|^2\right)}{\partial \hat{\theta}_u} = -\alpha_u x e_u^{\text{T}} \tag{16}$$

uses the error in actor weights and provides dynamics for these weights. Finally, appending the states, critics, actor, and Bellman error along with their dynamics, numerically compute trajectories which completes the cycle of synchronous learning.

IV. Asymptotic Stability of States, Critic Weights, and Actor Weights

We define estimation error for the critic and actor weights as $\tilde{\theta}_c := \theta_c - \hat{\theta}_c$, and $\tilde{\theta}_u := -Q_{\text{xu}} Q_{\text{uu}}^{-1} - \hat{\theta}_u$. The critic estimation error dynamics can be summarized as follows:

$$\begin{cases} \dot{\tilde{\theta}}_c = 0, & t \neq t_j \\ \tilde{\theta}_c^{+} = \tilde{\theta}_c(t_j)^{-} - \alpha_c \dfrac{\sigma\sigma^{\text{T}}}{\left(1+\sigma^{\text{T}}\sigma\right)^2} \tilde{\theta}_c(t_j)^{-}, & t = t_j \end{cases} \tag{17}$$

while the actor estimation error dynamics can be shown by

$$\dot{\tilde{\theta}}_u = -\alpha_u x x^{\text{T}} \tilde{\theta}_u - \alpha_u x x^{\text{T}} \tilde{Q}_{\text{xu}} R^{-1}. \tag{18}$$

Assumption 1: For each operant conditioning update schedule, we assume that the sequence $\{t_j\}_j$ of updating time instances is such that $\lim_{j\to\infty} t_j = \infty$. □

Remark 4: Assumption 1 implies that the update process of the critic network does not terminate during the system run, even under probabilistic schedules. □

Initially, the following lemma and fact are needed.

Lemma 1: Consider the critic error dynamics given in (17), which can be rewritten as

$$\begin{cases} \dot{\tilde{\theta}}_c = 0, & t \neq t_j \\ \tilde{\theta}_c^{+} = \left(I - \alpha_c \Delta_{\text{M}} \Delta_{\text{M}}^{\text{T}}\right) \tilde{\theta}_c(t_j)^{-}, & t = t_j \end{cases} \tag{19}$$

wherein $\Delta_{\text{M}} := (\sigma/(1+\sigma^{\text{T}}\sigma))$. The critic error dynamics evolve by means of an infinite sequence due to Assumption 1. This sequence converges to zero, i.e., $\|\tilde{\theta}_c\| \to 0$ as the length of the sequence tends toward infinity ($t \to \infty$).

Proof: According to the *conditions for exponential convergence of LMS* in [36]; and when the signal Δ_{M} is sufficiently and persistently excited, this result follows. ■

Fact 1: The entrywise norm of a submatrix is always at most the entrywise norm of the parent matrix itself. For example, $\|\tilde{Q}_{(.)}\| \leq \|\tilde{\theta}_c\|$. This is adopted from [37]. □

Now, we are able to show stability and convergence of the system via the following theorem.

Theorem 1: Consider the system dynamics given by (1), the critic approximator error dynamics given by (17), actor error dynamics given by (18), and the optimal control given by $K = -\hat{\theta}_u^{\text{T}}$. The critic dynamics are given by (14) and the actor dynamics are given by (16). Then, the equilibrium point of the closed-loop system with state $\gamma := [x^{\text{T}}, \tilde{\theta}_c^{\text{T}}, \tilde{\theta}_u^{\text{T}}]^{\text{T}}$ for initial conditions $\gamma(0)$ is asymptotically stable given a tuning gain for the hybrid critic tuning α_c is picked according to

$$1 < \alpha_u < \frac{2\left[\underline{\lambda}\left(M + Q_{\text{xu}} R^{-1} Q_{\text{ux}}\right) - \bar{\lambda}(Q_{\text{xu}} Q_{\text{ux}})\right]}{\delta \bar{\lambda}\left(R^{-1}\right)} \tag{20}$$

where δ is of unity order.

Proof: To prove stability of the hybrid, intermittent learning model, we analyze the dynamics first. We write the following Lyapunov function:

$$\begin{aligned}\mathcal{V} &= V_1 + V_2 + V_3 \\ &= \frac{1}{2}x^{\mathrm{T}}Px + \frac{1}{2}\|\tilde{\theta}_c\|^2 + \frac{1}{2}\mathrm{tr}\{\tilde{\theta}_u^{\mathrm{T}}\tilde{\theta}_u\}. \end{aligned} \tag{21}$$

Taking the time derivative of $\mathcal{V}$ during flow trajectories, when $t \neq t_j$, and keeping in mind that the critic weights, $\hat{\theta}_c$, are constant, we get

$$\dot{\mathcal{V}} = x^{\mathrm{T}}P(Ax + B\hat{u}) + \mathrm{tr}\left\{\tilde{\theta}_u^{\mathrm{T}}\dot{\tilde{\theta}}_u\right\}. \tag{22}$$

This can be further reduced to

$$\begin{aligned}\dot{\mathcal{V}} &= x^{\mathrm{T}}P(Ax + B\hat{u}) + \mathrm{tr}\left\{\tilde{\theta}_u^{\mathrm{T}}\dot{\tilde{\theta}}_u\right\} \\ &= x^{\mathrm{T}}P\big(Ax + Bu^* - B\tilde{\theta}_u^{\mathrm{T}}x\big) \\ &\quad - \alpha_u\mathrm{tr}\big\{\tilde{\theta}_u^{\mathrm{T}}xx^{\mathrm{T}}\tilde{\theta}_u + \alpha_u xx^{\mathrm{T}}\tilde{Q}_{\mathrm{xu}}R^{-1}\big\}.\end{aligned}$$

Defining T_1 as follows and using (5) and (6) and using Young's Inequalities, we get the reduction:

$$\begin{aligned}T_1 &= x^{\mathrm{T}}P(Ax + B\hat{u}) \\ &= \frac{1}{2}x^{\mathrm{T}}\big(M + Q_{\mathrm{xu}}R^{-1}Q_{\mathrm{ux}}\big)x - x^{\mathrm{T}}Q_{\mathrm{xu}}\tilde{\theta}_a^{\mathrm{T}}x \\ &\leq -\left[\underline{\lambda}\big(M + Q_{\mathrm{xu}}R^{-1}Q_{\mathrm{ux}}\big) - \frac{1}{2}\bar{\lambda}(Q_{\mathrm{xu}}Q_{\mathrm{ux}})\right]\|x\|^2 \\ &\quad + \frac{1}{2}\|x^{\mathrm{T}}\tilde{\theta}_u\|^2. \end{aligned} \tag{23}$$

Defining T_3 as follows provides the bounds:

$$\begin{aligned}T_3 &= -\alpha_u\mathrm{tr}\big\{\tilde{\theta}_u^{\mathrm{T}}xx^{\mathrm{T}}\tilde{\theta}_u\big\} - \alpha_u\mathrm{tr}\big\{\alpha_u xx^{\mathrm{T}}\tilde{Q}_{\mathrm{xu}}R^{-1}\big\} \\ &\leq -\frac{\alpha_u}{2}\|x^{\mathrm{T}}\tilde{\theta}_u\|^2 - \alpha_u\mathrm{tr}\big\{\tilde{\theta}_u^{\mathrm{T}}xx^{\mathrm{T}}\tilde{Q}_{\mathrm{xu}}R^{-1}\big\}. \end{aligned} \tag{24}$$

Following result in Lemma 1, where we establish that the discrete infinite sequence governing $\tilde{\theta}_c$—which is independent of the rest of the system states—is asymptotically stable under appropriate persistence of excitation conditions, according to [36] and [38]. Thus, we can conclude that $\|\tilde{\theta}_c\| \to 0$ is piecewise constant, discrete jumps. Thus, we can conclude that $T_2 = (1/2)\|\tilde{\theta}_c\|^2 \to 0$.

Putting (22), (23), and (24) for flow dynamics, we get

$$\begin{aligned}\dot{\mathcal{V}} \leq &-\left[\underline{\lambda}\big(M + Q_{\mathrm{xu}}R^{-1}Q_{\mathrm{ux}}\big) - \frac{1}{2}\bar{\lambda}(Q_{\mathrm{xu}}Q_{\mathrm{ux}})\right]\|x\|^2 \\ &- \frac{1}{2}(\alpha_u - 1)\|x^{\mathrm{T}}\tilde{\theta}_u\|^2 - \alpha_u\mathrm{tr}\big\{\tilde{\theta}_u^{\mathrm{T}}xx^{\mathrm{T}}\tilde{Q}_{\mathrm{xu}}R^{-1}\big\}.\end{aligned}$$

Now, taking Lemma 1 along with Fact 1 to rewrite $\alpha_u\mathrm{tr}\{\tilde{\theta}_u^{\mathrm{T}}xx^{\mathrm{T}}\tilde{Q}_{\mathrm{xu}}R^{-1}\}$, we have

$$\begin{aligned}\dot{\mathcal{V}} \leq &-\left[\underline{\lambda}\big(M + Q_{\mathrm{xu}}R^{-1}Q_{\mathrm{ux}}\big) - \frac{1}{2}\bar{\lambda}(Q_{\mathrm{xu}}Q_{\mathrm{ux}})\right]\|x\|^2 \\ &- \frac{1}{2}(\alpha_u - 1)\|x^{\mathrm{T}}\tilde{\theta}_u\|^2 + \frac{\alpha_u\delta}{2}\bar{\lambda}\big(R^{-1}\big)\|x\|^2. \end{aligned} \tag{25}$$

From (25), as long as (20) holds, the equilibrium point (origin) is asymptotically stable. ∎

This completes our stability analysis for the proposed operant conditioning framework.

V. Simulation and Results

Simulations serve to validate our results from an operant conditioning standpoint. This article uses a linear model of the F-16 fighter jet [39] with three states and two actuators, with the dynamics and parameters given by

$$\begin{aligned}\dot{x} &= \begin{bmatrix}-1.0189 & -0.9051 & -0.0022 \\ 0.8223 & -1.0774 & -0.1756 \\ 0 & 0 & -1.0000\end{bmatrix}x + \begin{bmatrix}0 & 1 \\ 0 & 1 \\ 1 & 1\end{bmatrix}u \\ M &= \mathrm{diag}([1\ 1\ 1]), \quad R = \mathrm{diag}([5\ 5]) \\ \alpha_c &= 50, \quad \alpha_u = 2, \quad T_{\mathrm{exp}} = 40\ \mathrm{s}, \quad T = 0.001\ \mathrm{s}.\end{aligned}$$

Running the framework multiple times drew up error bars and the region of likelihood to update for critic weights. Figs. 4 and 5 visualize this.

The distribution functions, namely the normal, uniform, and exponential distribution functions, used to trigger updates in the variable interval and variable-ratio update schedules introduce unpredictability to the updated law. This increases the learning rate and produces a "behavior" that lasts "longer." The error bars in Fig. 4 capture this unpredictability, and the degree of change in the adaptive tuning laws. For example, the variable interval reinforcement schedule using an exponential distribution, in Fig. 4(c), shows the most uncertainty in update pattern and tuning behavior, while the variable ratio reinforcement schedule using a uniform distribution, in Fig. 4(b), shows the least divergence. The confidence interval lines, shown in red, corresponding to the upper confidence interval, and blue, corresponding to the lower confidence interval, represent 68.3% confidence that the critic weights will lie inside this interval.

The mean and standard deviation of an exponential distribution, given by the ratio $(1/\lambda)$, where $\lambda = 0.01$, chosen for inherent similarities to Poisson process models updates as a random arrival pattern [Fig. 4(c) and (d)]. The uniform distribution function has compact, finite support and selects any number in the allowable range, given by 60–100, by the same probability.

When update ratios are selected from a uniform distribution, Fig. 4(b), we see very tight bounds in the confidence interval, while variable interval update schedule, Fig. 4(a), shows looser bounds in confidence. This likely arises because the learning agent updates its critic weights whenever it explores positively reinforcing state–space domain repeatedly for a number of times, Σ that lies in a small range of values. Ratios strictly between 60 and 100 are very similar, compared with those generated by the normal and exponential distributions.

Normal distributions have infinite support and can generate update ratios that are very large or very small to diversify the tuning law and creating looser confidence intervals, as shown in Fig. 4(e) and (f). In general, the variable-ratio update laws have smaller error bars than the variable interval update laws. This implies that the variable interval update schedule remains less predictable than the variable-ratio update schedule, thus revealing that learned behavior is better reinforced with a variable ratio update schedule.

Fig. 5(a) shows the difference in tuning history between update schedules for the same critic weight to highlight

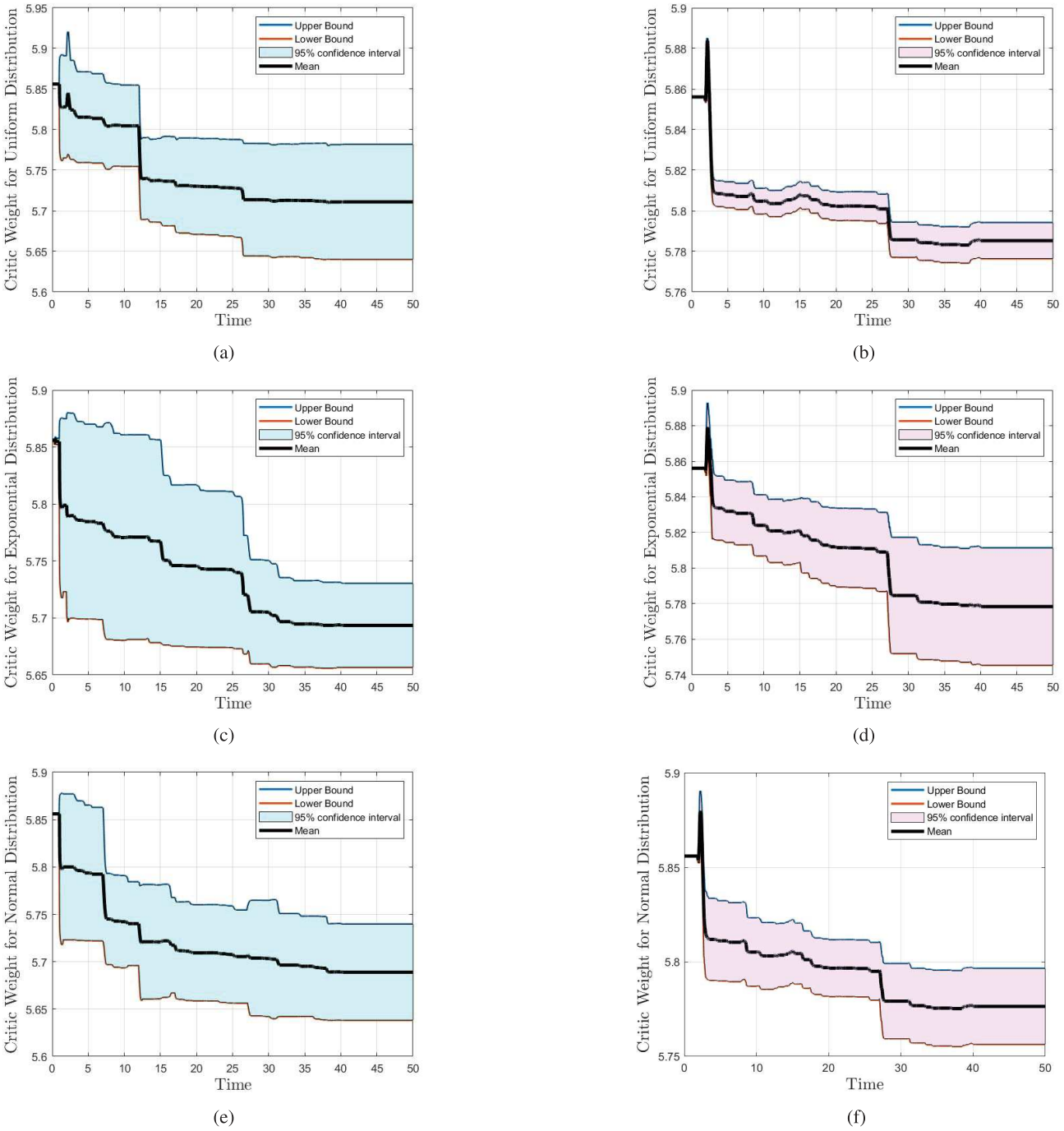


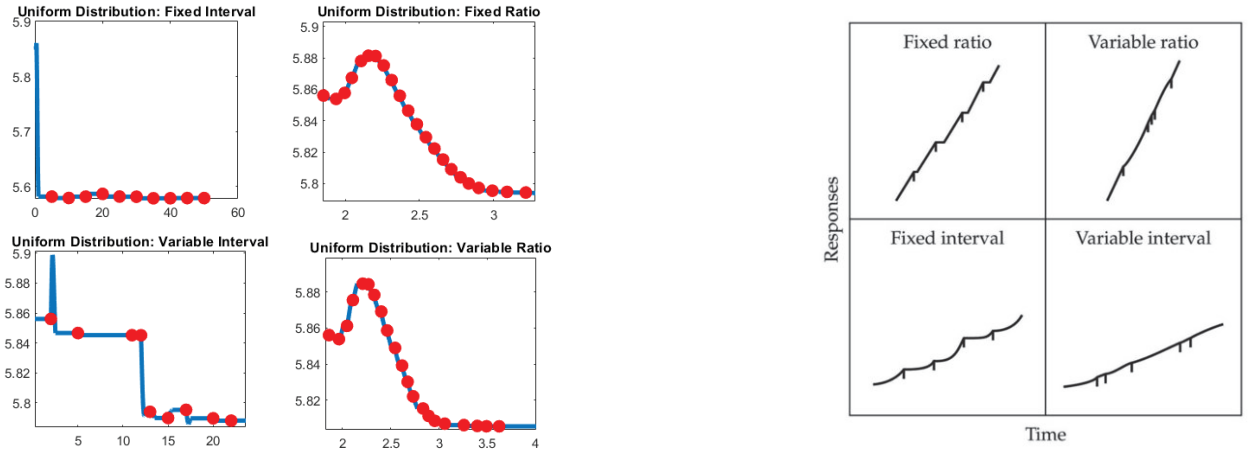
Fig. 4. Plot shows a critic weight's dynamic average learning pattern, in black, which represents the mean trajectory of the learned behavior over ten trials, while the blue and red lines represent the 68.3% confidence intervals. Fig. 4(b), (d), and (f) shows a smaller deviation (pink) from the mean learning pattern, when compared with their variable interval counterparts in Fig. 4(a), (c), and (e) (cyan). (a) Variable interval with uniform distribution. (b) Variable ratio with uniform distribution. (c) Variable interval with exponential distribution. (d) Variable ratio with exponential distribution. (e) Variable interval with normal distribution. (f) Variable ratio with normal distribution.

features of each update schedule and contrast it from classical RL by Skinner, shown in Fig. 5(b). This visualization of triggering critic updates draws a parallel between our technique to "learn" the optimal "behavior" and Skinner's operant conditioning schedules. The error, defined by

$$E = \frac{\|\hat{\theta}_c - \theta_{c,\text{optimal}}\|}{\|\theta_{c,\text{optimal}}\|}$$

compares the different update schedules used for operant conditioning in the actor-critic framework described in Section IV. The run-time for each experiment is set to 30 s, with added exploration noise for the first 20 s.

The fixed interval update schedule produces an error, given by $E_{\text{fix}}^{\text{Int}} = 0.5041$. Compared with the constant update regime that produces $E_{\text{constant}} = 0.5899$, the fixed interval update schedule provides a better estimate of optimal critic weights. The variable interval update regime approximates



(a) (b)

Fig. 5. Comparison of the different schedules: Each of the four subplots in Fig. 5(a) shows a different update schedule in the context of a CPS. The blue line represents a single critic weight's dynamics, while the red dots indicate the instant of time updates were triggered for the various schedules. In contrast, Fig. 5(b) shows the performance of fixed and variable schedules of reinforcements based on Skinner's hypothesis as seen in [40]. (a) Comparison between different reinforcement schedules. (b) Skinner's [40] response for variable/fixed reinforcements.

the optimal critic weights with an error $E_{\text{var}}^{\text{Int}} = 0.6208$. Compared with the constant update schedule and the fixed interval schedule, the variable interval update performs poorly. However, the variable interval update schedule depends on the random triggering instances of the update law. The number of updates may be constrained by the user, but the time of each update remains random. The fixed ratio update schedule yields an error of $E_{\text{fix}}^{\text{Ratio}} = 0.5834$, making it a weaker approximation of the optimal values of the critic weights as opposed to the variable-ratio update schedule which has an error of $E_{\text{var}}^{\text{Ratio}} = 0.5679$. The numbers picked from a uniform distribution trigger the critic updates for these values in the cases of variable interval and variable-ratio update regimens.

VI. Conclusion

We present a framework that incorporates Skinner's operant conditioning reinforcement schedules to induce an RL algorithm that enables autonomous agents to find optimal policies through direct interaction with their environment via intermittent tuning of critic weights. Intermittent tuning of critic weights prevents overutilization of the limited resources possessed by the CPS. These novel intermittent reinforcement schemes increase uncertainty in the learning rate of the desired behavior and the extinction rate of the acquired behavior of the learning agents. Thus, they create a buffer of uncertainty that prevents malicious agents from disrupting the smooth operation of the CPS. Simulation results compare each reinforcement schedule, fixed/variable interval, and fixed/variable ratio, based on the standardized error between tuned critic weights, and their optimal counterparts.

Future research endeavors will focus on extending the ideas presented in this work on intermittent learning, by considering the use of compressed sensing algorithms for communication transmission decrease. Furthermore, we will seek to increase the autonomy of the CPS agents by designing learning schemes that develop reward representations online based on high-level control objectives. Bounded rationality concepts will be leveraged, allowing the CPS to successfully predict the evolution of human-centric environments based on experimental results from behavioral economics and cognitive sciences. Intelligent autonomous vehicles will be employed as experimental platforms for our proposed algorithms. Finally, we will investigate the effect of our approach to issues of security and privacy of CPS, by building on our previous results, both on the co-design of moving target defense algorithms for unpredictability and Q-learning [41] and on issues of privacy in learning under attacks [42].

References

[1] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. 11th IEEE Int. Symp. Object Compon.-Oriented Real-Time Distrib. Comput. (ISORC)*, May 2008, pp. 363–369.
[2] K. Sampigethaya and R. Poovendran, "Aviation cyber–physical systems: Foundations for future aircraft and air transport," *Proc. IEEE*, vol. 101, no. 8, pp. 1834–1855, Aug. 2013.
[3] Z. Jiang, M. Pajic, and R. Mangharam, "Cyber–physical modeling of implantable cardiac medical devices," *Proc. IEEE*, vol. 100, no. 1, pp. 122–137, Jan. 2011.
[4] D. F. Sittig and H. Singh, "A new socio-technical model for studying health information technology in complex adaptive healthcare systems," in *Cognitive Informatics for Biomedicine*. Springer, 2015, pp. 59–80.
[5] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.
[6] T. Alpcan and T. Basar, "An intrusion detection game with limited observations," in *Proc. 12th Int. Symp. Dyn. Games Appl.*, vol. 26. Sophia Antipolis, France: Citeseer, 2006.
[7] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The future of human-in-the-loop cyber-physical systems," *Computer*, vol. 46, no. 1, pp. 36–45, Jan. 2013.
[8] S. N. Dorogovtsev and J. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford, U.K.: OUP, 2013.
[9] P. J. Werbos, "Neural networks for control and system identification," in *Proc. 28th IEEE Conf. Decis. Control*, Dec. 1989, pp. 260–265.
[10] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.
[11] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Proc. Int. Workshop Hybrid Syst., Comput. Control*. San Francisco, CA, USA: Springer, 2009, pp. 31–45.
[12] P. D. Neilson, M. D. Neilson, and N. J. O'Dwyer, "Internal models and intermittency: A theoretical account of human tracking behavior," *Biol. Cybern.*, vol. 58, no. 2, pp. 101–112, Jan. 1988.
[13] J. A. Doeringer and N. Hogan, "Intermittency in preplanned elbow movements persists in the absence of visual feedback," *J. Neurophysiol.*, vol. 80, no. 4, pp. 1787–1799, Oct. 1998.

[14] A. Fishbach, S. A. Roy, C. Bastianen, L. E. Miller, and J. C. Houk, "Kinematic properties of on-line error corrections in the monkey," *Exp. Brain Res.*, vol. 164, no. 4, pp. 442–457, Aug. 2005.

[15] S. Hanneton, A. Berthoz, J. Droulez, and J. J. E. Slotine, "Does the brain use sliding variables for the control of movements?" *Biol. Cybern.*, vol. 77, no. 6, pp. 381–393, Dec. 1997.

[16] R. C. Miall, D. J. Weir, and J. F. Stein, "Visuomotor tracking with delayed visual feedback," *Neuroscience*, vol. 16, no. 3, pp. 511–520, Nov. 1985.

[17] N. Hogan and D. Sternad, "On rhythmic and discrete movements: Reflections, definitions and implications for motor control," *Exp. Brain Res.*, vol. 181, no. 1, pp. 13–30, Jun. 2007.

[18] R. M. C. Spencer, H. N. Zelaznik, J. Diedrichsen, and R. B. Ivry, "Disrupted timing of discontinuous but not continuous movements by cerebellar lesions," *Science*, vol. 300, no. 5624, pp. 1437–1439, May 2003.

[19] K. G. Vamvoudakis, A. Mojoodi, and H. Ferraz, "Event-triggered optimal tracking control of nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 27, no. 4, pp. 598–619, Mar. 2017.

[20] X. Xie, Q. Zhou, D. Yue, and H. Li, "Relaxed control design of discrete-time Takagi–Sugeno fuzzy systems: An event-triggered real-time scheduling approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2251–2262, Dec. 2018.

[21] L. Zhang and G.-H. Yang, "Low-computation adaptive fuzzy tracking control for nonlinear systems via switching-type adaptive laws," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 10, pp. 1931–1942, Oct. 2019.

[22] Q. Zhang, D. Zhao, and Y. Zhu, "Event-triggered H_∞ control for continuous-time nonlinear system via concurrent learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1071–1081, Jul. 2017.

[23] B. F. Skinner, "Operant behavior," *Amer. Psychol.*, vol. 18, no. 8, p. 503, 1963.

[24] R. Bellman, "Dynamic programming," *Science*, vol. 153, nos. 37–31, pp. 34–37, 1966.

[25] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.

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

[27] W. S. Wong and R. W. Brockett, "Systems with finite communication bandwidth constraints. II. Stabilization with limited information feedback," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 1049–1053, May 1999.

[28] D. Hristu and K. Morgansen, "Limited communication control," *Syst. Control Lett.*, vol. 37, no. 4, pp. 193–205, 1999.

[29] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 3270–3285.

[30] 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, Jan. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000510981730136X

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

[32] B. Skinner, "The experimental analysis of behavior," *Amer. Scientist*, vol. 45, no. 4, pp. 343–371, 1957.

[33] B. F. Skinner, "'Superstition' in the pigeon," *J. Exp. Psychol.*, vol. 38, no. 2, p. 168, 1948.

[34] B. F. Skinner, *Contingencies of Reinforcement: A Theoretical Analysis*, vol. 3. Cambridge, MA, USA: BF Skinner Foundation, 2014.

[35] X. Zhong and H. He, "An event-triggered ADP control approach for continuous-time system with unknown internal states," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 683–694, Mar. 2017.

[36] R. Bitmead, "Persistence of excitation conditions and the convergence of adaptive schemes," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 2, pp. 183–191, Mar. 1984.

[37] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[38] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Chelmsford, MA, USA: Courier Corporation, 2012.

[39] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[40] O. Perez, "A cooperative dual-system model of instrumental conditioning," Ph.D. dissertation, Darwin College, Cambridge, U.K., Apr. 2017.

[41] P. P. Sahoo and K. G. Vamvoudakis, "On-off adversarially robust Q-learning," *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 749–754, Jul. 2020.

[42] L. Zhai and K. G. Vamvoudakis, "A data-based private learning framework for enhanced security against replay attacks in cyber-physical systems," *Int. J. Robust Nonlinear Control*, vol. 31, no. 6, pp. 1817–1833, Apr. 2021.

Prachi Pratyusha Sahoo (Member, IEEE) was born in Odisha, India. She received the B.Sc. degree in mechanical engineering and the M.Sc. degree in aerospace engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2018 and 2020, respectively, where she is currently pursuing the Ph.D. degree with the Daniel Guggenheim School of Aerospace Engineering, under the supervision of Dr. Kyriakos G. Vamvoudakis.

She is currently a Research Assistant with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology. Her current research interests include safe autonomy, optimal control, and reinforcement learning-based control.

Aris Kanellopoulos (Member, IEEE) was born in Athens, Greece. He received the Diploma degree in mechanical engineering from the National Technical University of Athens, Athens, in 2016. He is currently pursuing the Ph.D. degree with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

Until 2018, he was a Research Assistant with the Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, USA. He is currently a Research Assistant with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology. His current research interests include cyber-physical security, game theory, and optimal and learning-based control.

Kyriakos G. Vamvoudakis (Senior Member, IEEE) was born in Athens, Greece. He received the Diploma degree (Hons.) (M.Sc., a five-year degree) in electronic and computer engineering from the Technical University of Crete, Chania, Greece, in 2006, and the M.Sc. and Ph.D. degrees in electrical engineering from The University of Texas at Arlington, Arlington, TX, USA, in 2008 and 2011, respectively.

From 2011 to 2012, he was an Adjunct Professor and a Faculty Research Associate with The University of Texas at Arlington and the Automation and Robotics Research Institute. From 2012 to 2016, he was a Project Research Scientist with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA, USA. He was an Assistant Professor with the Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, USA, until 2018. He currently serves as an Assistant Professor with Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA, where he also holds a secondary appointment with the School of Electrical and Computer Engineering. His research interests include reinforcement learning, game theory, cyber-physical security, networked control, and safe autonomy.

Dr. Vamvoudakis currently is a member of the IEEE Control Systems Society Conference Editorial Board, and a member of the Technical Chamber of Greece. He was a recipient of the 2018 NSF CAREER Award, the 2019 Army Research Office (ARO) Young Investigator Program (YIP) Award, the 2021 GT Chapter Sigma Xi Young Faculty Award, and several international awards, including the Best Paper Award for Autonomous/Unmanned Vehicles at the 27th Army Science Conference in 2010 and the 2016 International Neural Network Society Young Investigator Award. He has also served on various international program committees and has organized special sessions, workshops, and tutorials for several international conferences. He is an Associate Editor of *Automatica*, *IEEE Computational Intelligence Magazine*, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, *Neurocomputing*, the *Journal of Optimization Theory and Applications*, the IEEE CONTROL SYSTEMS LETTERS, and the *Frontiers in Control Engineering-Adaptive, Robust and Fault-Tolerant Control*. He is a registered Electrical/Computer Engineer (PE) of Greece.