2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Continuous-Time Safe Learning with Temporal Logic Constraints in Adversarial Environments

Chuangchuang Sun[1], *Member, IEEE,* Kyriakos G. Vamvoudakis[2], *Senior Member, IEEE*

***Abstract*— This paper investigates a safe learning problem that satisfies linear temporal logic (LTL) constraints with persistent adversarial inputs, and quantified performance and robustness. Via a finite state automaton, the LTL specification is first decomposed to a sequence of several two point boundary value problems (TPBVP), each of which has an invariant safety zone. Then we employ a system transformation that guarantees state, and control safety with logarithmic barrier and hyperbolic-type functions as well as a worst-case adversarial input that wants to push the system outside the safety set. A safe learning method is used to solve the sub-problem, where the actors (approximators of the optimal control and the worst-case adversarial inputs) and the critic (approximator of the cost) are tuned to learn the optimal policies without violating any safety. Finally, by following a Lyapunov stability analysis we prove boundedness of the closed-loop system while simulation results are used to validate the effectiveness.**

***Index Terms*— Linear temporal logic, barrier functions, reinforcement learning, formal methods.**

I. Introduction

Reinforcement Learning (RL), which is essentially a trial and error learning, finds the optimal policy via interactions with the environment [1]–[6]. Over time, the learning agent modifies her policy to optimize a long-term reward. Such approaches have been applied to various applications, including systems with discrete and continuous state-action spaces [7]. However, before the optimal policy is learned, the agent is highly likely to explore some unsafe regions as she aims to optimize a given reward. This shortcoming significantly limits such methods to be applicable to real-world applications, since this might lead to hardware failures of physical systems or harm human operators. Consequently, safe learning focuses on guaranteeing the satisfaction of safety constraints. Additionally, due to the expressiveness of Linear Temporal Logic (LTL) in control systems, the LTL specifications can provide a tool to model complex systems as mathematical entities. By building a mathematically rigorous model of a complex system, it is possible to verify the system's properties in a more thorough fashion than empirical testing.

[1]C. Sun is with Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139. Email: ccsun1@mit.edu.

[2]K. G. Vamvoudakis is with the Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, Atlanta, GA 30332, USA, Email: kyriakos@gatech.edu.

This work was supported in part, by ONR Minerva under grant No. N00014 − 18 − 1 − 2160, and by an NSF CAREER under grant No. CPS-1851588.

Related Work: Existing safe learning algorithms greatly depend on the design of barrier function methods [8]. Given the critical role of Lyapunov-based analysis in the controller design, a barrier Lyapunov function (BLF) method has been proposed in [9] to potentially satisfies output constraints. Moreover, the work of [10] presents a neuro-inspired output feedback BLF, for nonlinear tracking problems. While the BLF method is further extended to full state constraints, the authors did not consider actuator constraints. The authors in [11]–[13] exploited control barrier function (CBF) where neural networks (NN) and learning are used. But in such works, the data to train the NN are CBF-certified trajectories with random priors which allows the CBF controller to project the raw control input onto the CBF condition. Furthermore, if the initial condition is not contained in the feature range of the training data, robustness is fragile. The work of [14] leverages barrier-certified RL and considers recovering safety after violations due to non-stationary dynamics. While it is true that CBF can guarantee safety during the learning process given a safe initial state (due to the forward invariance), the CBF certificate can become infeasible. The authors of [15], [16] utilize safe learning with worst-case disturbances for linear systems but they require a backup of safety controllers that do not guide the learning process at the cost of exploration efficiency.

Contributions: The contribution of this paper is twofold. We first decompose the LTL specification to a finite state automaton (FSA), where each sub-problem is solved as a TPBVP. The composition of the sub-problems will accomplish the whole LTL specification. Moreover, by following the work of reinforcement learning-based control [17]–[20] we propose a novel structure to optimally solve the sub-problems that might have different safety constraints, given their current state in the FSA.

Structure: The remainder of the paper is structured as follows. Section II formulates the problem wile also including backgrounds on: LTL decomposition, systems transformation, and game theory. Section III provides a learning mechanism to solve the problem. Finally Section IV provides a simulation example to validate the efficacy of the framework while Section V concludes and talks about future work.

II. Problem Formulation

Consider a nonlinear system of the form,

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, \forall i = 1, \dots, n-1,$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u + h_n(\bar{x}_n)v$$

978-1-5386-8266-1/$31.00 ©2020 AACC

$$x(t_0) = x_0,\ t \geq 0 \tag{1}$$

where $x = [x_1, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n, u \in \mathbb{R}$, and $v \in \mathbb{R}$ are the states, the control inputs and the adversarial inputs, respectively, and $f_i(x) : \mathbb{R}^n \to \mathbb{R}, g_i(x) : \mathbb{R}^n \to \mathbb{R}$ and $d(x) : \mathbb{R}^n \to \mathbb{R}, \forall i = 1, \ldots, n$. Moreover, $\bar{x}_i = [x_1, \ldots, x_i]^{\mathrm{T}}$ is a subset of the states.

A. Temporal Logic Syntax and Semantics

A TL formula defines predicates of the form $lb \leq p(x) = a^T x + \bar{r} \leq ub$, where $p(x) = \mathbb{R}^n \to \mathbb{R}$ is a linear combination of the states with $a \in \mathbb{R}^n, \bar{r} \in \mathbb{R}$ and $lb \in \mathbb{R}, ub \in \mathbb{R}$. For predicates with only lower bound constraints, we can always trivially add large enough upper bounds; and vice-versa for the absence of lower bounds. With predicates p_1 and p_2, a TL specification has the following syntax,

$$\begin{aligned} p \models \top | \bigcirc p_1 | \Box p_1 | \neg p_1 | \Diamond p_1 | p_1 \wedge p_2 | p_1 \vee p_2 | \\ p_1 \Rightarrow p_2 | p_1 \mathcal{U} p_2 | p_1 \mathcal{T} p_2, \end{aligned} \tag{2}$$

where $\top$ is the true Boolean constant, $\bigcirc$(next), $\Box$ (always), $\Diamond$ (eventually), $\mathcal{U}$ (until), $\mathcal{T}$ (then) are the temporal operators, and $\neg$(negation/not), $\wedge$ (conjunction/and), and $\vee$ (disjunction/or) are the Boolean connectives. Consider an example formula of the form $p \models \Diamond p_1 \wedge \Diamond p_2 \wedge \Box p_3$ which denotes that the trajectory of the state will eventually satisfy p_1 and p_2 while always satisfying p_3.

Problem 1: For the system given in (1), find a control policy such that the closed-loop system has a stable equilibrium point, while the control input satisfies $\|u\| \leq \lambda_1$, the adversarial input satisfies $\|v\| \leq \lambda_2$, and the states satisfy LTL safety constraints given by p.

In the following, we will discuss the satisfaction of a TL specification.

B. Decomposition of LTL

Given that the TL specification is related to the time evolution, the concrete safety constraint in a given time interval varies. As a result, via the use of FSA, we manually divided Problem 1 into a series of sequential sub-problems, that each satisfies a certain safety constraint. Consequently, the original TL safety constraint is eventually satisfied. Consider now the LTL $p \models \Diamond p_2 \wedge (\neg p_2 \mathcal{U} p_1) \wedge \Box \neg p_3$, whose FSA is shown in Figure 1. Given that p_1 and p_2 are disjoint, the only path from the initial FSA state T0_init to the final state accept_S0 is: T0_init $\to$ T0_S2 $\to$ accept_S0, where the state accept_S0 means the LTL p is satisfied. Thus, there are two TPBVP sub-problems, with different safety constraints. For the first one (T0_init $\to$ T0_S2), the boundary conditions are the initial states and p_1 while the safety zone is outside of $p_2 \cup p_3$. Similarly, for the second sub-problem (T0_S2 $\to$ accept_S0), the boundary conditions are p_1 and p_2 while the safety zone is outside of p_3. However, systematically for an LTL formula with a complex FSA, it is not straightforward to get the path from an initial FSA state to a final FSA state. Hence, we can view the FSA as a directed graph and approximately estimate the edge weights, i.e., distance, and then cast it as a shortest path problem. So far, we have finished decomposing complex LTL specifications into sequential sub-problems, that as a whole will eventually satisfy the original LTL constraint.

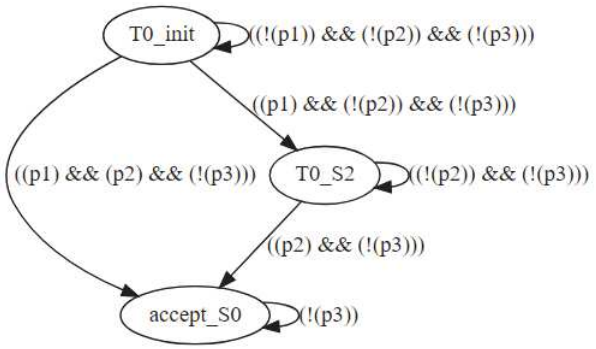

Fig. 1. Finite state automaton generated by formula $p \models \Diamond p_2 \wedge (\neg p_2 \mathcal{U} p_1) \wedge \Box \neg p_3$.

C. Sub-Problems as TPBVPs

In a TPBVP sub-problem, with the predicates as $c_i \leq p_i = a_i x + r_i \leq C_i$ with $i = 1, \ldots, m$, the safety constraints are defined as,

$$c \leq Ax + r \leq C$$

with $A = [a_1^{\mathrm{T}}; a_2^{\mathrm{T}}; \ldots; a_m^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{m \times n}$, $r = [r_1, \ldots, r_m]$, $c = [c_1, \ldots, c_m]$, and $C = [C_1, \ldots, C_m] \in \mathbb{R}^m$.

Problem 2: (sub-problem) For the system given by (1), find a control policy such that the system reaches a certain terminal condition x_T given an initial condition x_0, while the control input satisfies $\|u\| \leq \lambda_1$, the adversarial input satisfies $\|v\| \leq \lambda_2$, and the states satisfy $c \leq Ax + r \leq C$.

As a sub-problem of Problem 1, the Problem 2 translates a temporal logic constraint p to a more detailed and time-invariant constraint in the current time interval.

In order to deal with the safety constraint, a barrier function is defined by,

$$b(p, c_0, C_0) = \log\Big(\frac{C_0}{c_0}\frac{c_0 - p}{C_0 - p}\Big), \forall p \in (c_0, C_0), \tag{3}$$

where $c_0 < 0 < C_0$. This does not lose any generality since we can always satisfy this via adjusting r_i. Moreover, $b(p, c_0, C_0)$ is invertible in the interval (c_0, C_0) as follows

$$b^{-1}(y, c_0, C_0) = c_0 C_0 \frac{e^{\frac{y}{2}} - e^{-\frac{y}{2}}}{c_0 e^{\frac{y}{2}} - C_0 e^{-\frac{y}{2}}}, \forall y \in \mathbb{R}, \tag{4}$$

with dynamics,

$$\frac{\mathrm{d}b^{-1}(y, c_0, C_0)}{\mathrm{d}y} = \frac{C_0 c_0^2 - c_0 C_0^2}{c_0^2 e^y - 2c_0 C_0 + C_0^2 e^{-y}}. \tag{5}$$

A system transformation of (1) that accounts for safety can be written as,

$$\begin{aligned} s_i &= b(p_i(x), c_i, C_i) \\ p_i(x) &= b^{-1}(s_i, c_i, C_i) \\ p_i(x) &= a_i x + r_i, \forall i = 1, \ldots, m. \end{aligned} \tag{6}$$

Through the use of the chain rule, we have that,

$$\frac{\mathrm{d}p_i(x)}{\mathrm{d}t} = a_i \dot{x} = \frac{\mathrm{d}b^{-1}(s_i, c_i, C_i)}{\mathrm{d}s_i}\frac{\mathrm{d}s_i}{\mathrm{d}t},$$

which yields,

$$\frac{\mathrm{d}s_i}{\mathrm{d}t} = \frac{1}{\frac{\mathrm{d}b^{-1}(s_i,c_i,C_i)}{\mathrm{d}s_i}} a_i^T \dot{x}. \tag{7}$$

We can thus write in a compact form,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + d(x)v \\ x(t_0) &= x_0, \end{aligned} \tag{8}$$

with $f(x) = [f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, \ldots, f_{n-1}(\bar{x}_{n-1}) + g_{n-1}(\bar{x}_{n-1})x_n, f_n(\bar{x}_n)]^{\mathrm{T}}$, $g(x) = [0, \ldots, 0, g_n(\bar{x}_n)]$ and $d(x) = [0, \ldots, 0, d_n(\bar{x}_n)]$. Additionally, from equation (6), we have that,

$$Ax + r = b^{-1}(s, c, C), \tag{9}$$

with $b^{-1}(s,c,C) = [b^{-1}(s_1,c_1,C_1), \ldots, b^{-1}(s_m,c_m,C_m)]^T \in \mathbb{R}^m$.

It also follows that,

$$x = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}(b^{-1}(s,c,C) - r). \tag{10}$$

Remark 3: It is worth noting that in order to make $A^{\mathrm{T}}A \in \mathcal{S}^n$ invertible, it will be necessary for A to be full column rank, i.e., $m \geq n$. That is not restricted since we can always add sufficiently large trivial bounds on the states, such as $-M \leq x_1 \leq M$ with $M > 0$.

Now combining (7), (8), and (10), one has,

$$\begin{aligned} \frac{\mathrm{d}s_i}{\mathrm{d}t} &= \frac{1}{\frac{\mathrm{d}b^{-1}(s_i,c_i,C_i)}{\mathrm{d}s_i}} a_i \dot{x} \\ &= \frac{1}{\frac{\mathrm{d}b^{-1}(s_i,c_i,C_i)}{\mathrm{d}s_i}} a_i \big(f(x) + g(x)u + h(x)v\big), \end{aligned} \tag{11}$$

with $x = (A^TA)^{-1}A^T(b^{-1}(s,c,C) - r)$. For notational convenience, the system (11) will be rewritten in a compact form as

$$\dot{s} = F(s) + G(s)u + H(s)v. \tag{12}$$

In Problem 2, the terminal condition is required to be x_T. Correspondingly, in 12 the terminal condition is $b(p(x_T), c, C) = [b(s_1,c_1,C_1), \ldots, b(s_m,c_m,C_m)]^T$. Through some algebra, the TPBVP is transformed now, from $b(p(x_0), c, C)$ to $b(p(x_T), c, C)$. Then a new equivalent problem is defined as follows.

Problem 4: For the system (12), find a policy u such that the system reaches the terminal condition $b(p(x_T), c, C)$ from the initial condition $b(p(x_0), c, C)$ such that the performance

$$J(s_0) = \int_{t_0}^{\infty} \big(Q(s) + \Theta(u) - \Phi(v)\big)\mathrm{d}t$$

is minimized by u and maximized by v subject to the dynamics given by (12), $\|u\| \leq \lambda_1, \|v\| \leq \lambda_2$, where $\lambda_1 > 0, \lambda_2 > 0$, $Q(s)$ is positive definite and monotonically increasing with regards to $\|s\|$, and $\Theta(u)$, is a positive definite integrand function. For notational simplicity, we shall define $U(s,u,v) := Q(s) + \Theta(u) - \Phi(v)$. Note now that, v is treated as the worst-case perturbation rather than a random one.

Lemma 5: (sub-problem) Suppose that $u^\star$, $v^\star$ solve the optimization Problem 4. Then $u^\star$, $v^\star$ also solve Problem 2 with x_0 satisfying $c \leq Ax_0 + b \leq C$.

Proof: The proof is given in [9]. ■

In order to satisfy the safety constraint on u, v, i.e., $\|u\| \leq \lambda_1, \|v\| \leq \lambda_2$, $\Theta(u)$ and $\Phi(v)$ has the following form adopted from [21],

$$\begin{aligned} \Theta(u) &= 2\int_0^u \lambda_1 \tanh^{-1}\Big(\frac{z}{\lambda_1}\Big)\gamma_1 \mathrm{d}z \\ \Phi(v) &= 2\int_0^v \lambda_2 \tanh^{-1}\Big(\frac{z}{\lambda_2}\Big)\gamma_2 \mathrm{d}z \end{aligned} \tag{13}$$

where $\tanh^{-1}(\cdot)$ denotes the inverse of the hyperbolic tangent function. One also needs to note that the states s and control v are not coupled in the safety constraints.

We are interested to find a saddle-point solution, which is equivalent to,

$$J(s_0, u^\star, v) \leq J(s_0, u^\star, v^\star) \leq J(s_0, u, v^\star). \tag{14}$$

As a result, the ultimate goal is to find the following optimal value function,

$$V^\star(s_t) = \min_u \max_v \int_t^\infty (Q(s) + \Theta(u) - \Phi(v))\mathrm{d}t \;\forall s_t,\; t \geq 0.$$

Given an admissible pair of policies u, v, the Hamiltonian function is,

$$\begin{aligned} 0 = \mathcal{H}(s,u,v,\frac{\partial V^\star}{\partial s}) = \\ \left(\frac{\partial V^\star}{\partial s}\right)^{\mathrm{T}}\Big[F(s) + G(s)u + H(s)v\Big] + U(s,u,v). \end{aligned} \tag{15}$$

Consequently, we can apply the stationary condition to (15) i.e.,, $\frac{\partial H}{\partial u} = 0$ and $\frac{\partial H}{\partial v} = 0$, and for the minimizer (control input), we can get

$$\frac{\partial H}{\partial u} = G^{\mathrm{T}}(s)\frac{\partial V^\star}{\partial s} + \frac{\partial \Theta(u^\star)}{\partial u} = 0. \tag{16}$$

Combining $\frac{\partial \Theta(u)}{\partial u} = 2\lambda \tanh^{-1}\Big(\frac{u}{\lambda}\Big)\gamma_1$, (16) leads to

$$u^\star = -\lambda_1 \tanh\Big(\frac{1}{2\lambda_1\gamma_1}G^{\mathrm{T}}(s)\frac{\partial V^\star}{\partial s}\Big). \tag{17}$$

In a similar way, the worst-case adversarial input (maximizer) $v^\star$ is

$$v^\star = \lambda_2 \tanh\Big(\frac{1}{2\lambda_2\gamma_2}H^{\mathrm{T}}(s)\frac{\partial V^\star}{\partial s}\Big). \tag{18}$$

Given (17) and (18), and plugging $u^\star$ and $v^\star$ into $\mathcal{H}(s, u^\star, v^\star, \frac{\partial V^\star}{\partial s})$ one has,

$$\begin{aligned} 0 = \quad & \left(\frac{\partial V^\star}{\partial s}\right)^{\mathrm{T}} F(s) + Q(s) + \lambda_1^2\gamma_1 \ln\Big(1 - \tanh^2(D_2^\star)\Big) \\ & -\lambda_2^2\gamma_2 \ln\Big(1 - \tanh^2(D_1^\star)\Big), \end{aligned} \tag{19}$$

with $D_1^\star = \frac{1}{2\lambda_1\gamma_1}G^{\mathrm{T}}(s)\frac{\partial V^\star}{\partial s}$ and $D_2^\star = \frac{1}{2\lambda_2\gamma_2}H^{\mathrm{T}}(s)\frac{\partial V^\star}{\partial s}$.

III. Safe Learning

Based on the transformed system, we will use two actors. One will approximate the control input (17) and one will approximate the worst-case adversarial input (18). Finally we will use a critic network to approximate the value (14). All approximators will be updated synchronously. A schematic representation of the framework is shown in Figure 2.

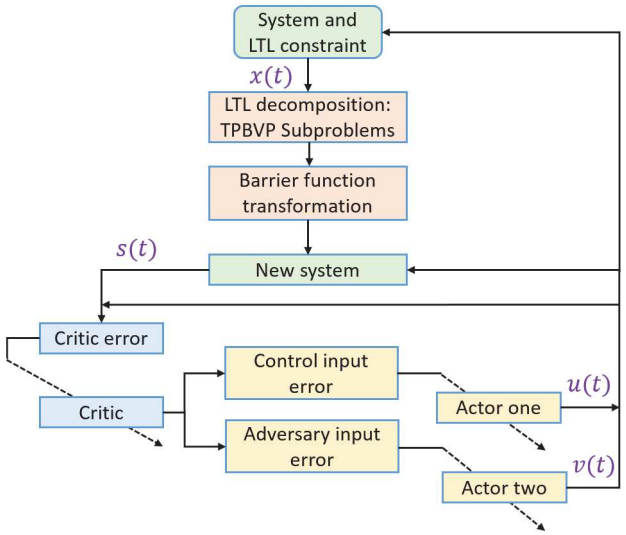

Fig. 2. The flowchart of the framework. The critic learning, control input actor and adversarial input learning are driven by the errors (25), (35) and (37), respectively.

Assumption 6: There exists a positive definite and differentiable $V(x)$ to (15). Also, there exists an approximator such that the value function $V(s)$ and its gradient $\nabla V(s) := \frac{\partial V(s)}{\partial s}$ can be uniformly approximated in $\Omega \in \mathbb{R}^n$ as

$$\begin{aligned} V^\star(s) &= W^{\mathrm{T}}\phi(s) + \epsilon(s) \\ \nabla V^\star(s) &= [\nabla\phi(s)]^{\mathrm{T}} W + \nabla\epsilon(s) \end{aligned} \tag{20}$$

where $W \in \mathbb{R}^N$ is the critic weight, $\phi(s) : \mathbb{R}^n \to \mathbb{R}^N$ is the critic basis, $\epsilon(s)$ and $\nabla\epsilon(s)$ are approximation errors bounded as $\|\epsilon(s)\| \le b_\epsilon$ and $\|\nabla\epsilon(s)\| \le b_{d\epsilon}$. Moreover, it is assumed that $\|\phi(s)\| \le b_\phi$ and $\|\nabla\phi(s)\|$ for all $s \in \Omega$.

Given the control policy $u^\star$ and the worst-case adversarial input $v^\star$, the approximation error of the Bellman equation (15) is

$$\epsilon_B = U(s, u^\star, v^\star) + W^{\mathrm{T}}\sigma, \tag{21}$$

with $\sigma = \nabla\phi(s)(F(s) + G(s)u^\star + H(s)v^\star)$. Moreover, multiplying $[F(s) + G(s)u^\star + H(s)v^\star]$ at both sides of the $\nabla V(s)$ approximation in (20), and combining the Bellman equation (15), then (21) can be written as

$$\epsilon_B = -[\nabla\epsilon(s)]^{\mathrm{T}}[F(s) + G(s)u^\star + H(s)v^\star],$$

with $\|\epsilon_B\| \le b_B$.

Furthermore, the approximation error of the HJB equation (19) is

$$\epsilon_{\mathrm{HJB}}(s) = W^{\mathrm{T}}\nabla\phi(s)F(s) + Q(s)$$
$$\lambda_1^2\gamma_1 \ln\Big(1 - \tanh^2(D_1)\Big) - \lambda_2^2\gamma_2 \ln\Big(1 - \tanh^2(D_2)\Big),$$

where $D_1^\star := \frac{1}{2\lambda_1\gamma_1} G^{\mathrm{T}}(s)[\nabla\phi(s)]^T W$ and $D_2^\star := \frac{1}{2\lambda_2\gamma_2} H^{\mathrm{T}}(s)[\nabla\phi(s)]^T W$. It is further assumed that approximator (20) can guarantee that the HJB approximation error is also bounded as $\|\epsilon_{\mathrm{HJB}}(s)\| \le b_{\mathrm{HJB}}$.

A. Value Function Approximation

The ideal weights W, which best approximate the value function $V^\star(s)$ in (20) are unknown. Hence, we will use an estimation of W namely, W_c to write,

$$\begin{aligned} \hat{V}(s) &= \hat{W}_c^{\mathrm{T}}\phi(s) \\ \nabla\hat{V}(s) &= [\nabla\phi(s)]^{\mathrm{T}}\hat{W}_c. \end{aligned} \tag{22}$$

Then the residual of the Bellman equation (15) expressed via $\hat{W}_c$ is

$$e_c(t) = U(s(t), u(t), v(t)) + \hat{W}_c^{\mathrm{T}}\sigma(t). \tag{23}$$

Then the critic weight approximation error is

$$\tilde{W}_c = W - \hat{W}_c. \tag{24}$$

After combining equations (21), (23), and (24), one has,

$$e_c = \epsilon_B - \tilde{W}_c^{\mathrm{T}}\sigma. \tag{25}$$

Subject to the worst-case adversarial input, the policy evaluation of the admissible control u can be formulated to continuously adapt W_c such as that the following error index is minimized [19]

$$E_c = \frac{1}{2}\frac{[e_c(t)]^2}{(1 + \sigma^T\sigma)^2}. \tag{26}$$

Then via chain rule and the definition of e_c in (23), a gradient descent algorithm is used to minimize E_c as,

$$\begin{aligned} \dot{\hat{W}}_c &= -\alpha_c\frac{\partial E_c}{\partial\hat{W}_c} \\ &= -\alpha_c\frac{\sigma(t)}{(1 + \sigma^T(t)\sigma(t))^2}\Big(U(s(t), u(t), v(t)) + \hat{W}_c^{\mathrm{T}}\sigma(t)\Big). \end{aligned} \tag{27}$$

Definition 7: (Persistency of Excitation (PE)) A vector signal $y(t) \in \mathbb{R}^p$ is exciting in the interval $[t, t+T], \forall T > 0$, if there exists $\beta_1, \beta_2 \in \mathbb{R}^+$ such that

$$\beta_1 I_{p\times p} \le \int_t^{t+T} y(\tau)y^T(\tau)\mathrm{d}\tau \le \beta_2 I_{p\times p}, \forall t. \tag{28}$$

Theorem 8: For any admissible policy, let the critic network (22) be updated according to (27). Suppose that the signal $\sigma(t)/(1 + \sigma^T(t)\sigma(t))$ satisfies the PE condition, then $\tilde{W}_c$ is uniformly ultimately bounded.

Proof: Combining (23), (24), (25) together with (27), it yields that

$$\begin{aligned} \dot{\tilde{W}}_c = &-\alpha_c\left[\frac{\sigma(t)\sigma^{\mathrm{T}}(t)}{(1 + \sigma^{\mathrm{T}}(t)\sigma(t))^2}\right]\tilde{W}_c \\ &+\alpha_c\left[\frac{\sigma(t)}{(1 + \sigma^{\mathrm{T}}(t)\sigma(t))^2}\right]\epsilon_B(t). \end{aligned} \tag{29}$$

The dynamics of $\tilde{W}_c$ can be viewed as a linear time-varying system with $\epsilon_B(t)$ as the control input and write $\tilde{W}_c$ as

$$\tilde{W}_c(t) = \varphi(t,t_0)\tilde{W}_c(t_0) + \int_{t_0}^{t} \varphi(t,\tau)\frac{\alpha_c \sigma(\tau)\epsilon_B(\tau)}{[1+\sigma^{\mathrm{T}}(\tau)\sigma(\tau)]^2}\mathrm{d}\tau, \tag{30}$$

where the state transition matrix is defined as

$$\frac{\partial \varphi(t,t_0)}{\partial t} = -\alpha_c \frac{\sigma(t)\sigma^{\mathrm{T}}(t)}{(1+\sigma^{\mathrm{T}}(t)\sigma(t))^2}\varphi(t,t_0), \varphi(t,t) = I.$$

Moreover, as it is assumed that the signal $\sigma(t)/(1+\sigma^T(t)\sigma(t))$ satisfies the PE condition, there exist $\rho_1,\ \rho_2 \in \mathbb{R}^+$ such that

$$\|\tilde{W}_c(t)\| = \rho_1 e^{-\rho_2(t-t_0)}\|\tilde{W}_c(t_0)\| + \frac{\alpha_c}{\rho_2[1+\sigma^{\mathrm{T}}(t)\sigma(t)]}\|\epsilon_B(t)\|. \tag{31}$$

It is thus easy to conclude that $\|\tilde{W}_c(t)\|$ is uniformly ultimately bounded given $\epsilon_B(t)$. ■

B. Actor Learning

The control policy is,

$$u(s) = -\lambda_1 \tanh(D_1), \tag{32}$$

with $D_1 = \frac{1}{2\lambda_1\gamma_1}G^{\mathrm{T}}(s)[\nabla\phi]^{\mathrm{T}}\hat{W}_c$.

However, this policy improvement will not guarantee the stability of the equilibrium point of the closed-loop system. As a result, the policy that is going to be implement in a form of an actor network as follows,

$$u_a(s) = -\lambda_1 \tanh(D_a) \tag{33}$$

with $D_a = \frac{1}{2\lambda_1\gamma_1}G^{\mathrm{T}}(s)[\nabla\phi]^{\mathrm{T}}\hat{W}_a$. Similarly, for the adversarial input, we have

$$v_d(s) = \lambda_2 \tanh(D_d) \tag{34}$$

with $D_d = \frac{1}{2\lambda_2\gamma_2}H^{\mathrm{T}}(s)[\nabla\phi]^{\mathrm{T}}\hat{W}_d$.

To minimize the error function,

$$E_u = \frac{\gamma_1}{2}\|e_u\|_2^2. \tag{35}$$

where $e_u = u_c - u_a = \lambda_1[\tanh(D_a) - \tanh(D_1)]$. In order to minimize (35) we will use a gradient-based rule to write,

$$\dot{\hat{W}}_a = -\alpha_a[\nabla\phi G e_u - \nabla\phi G \tanh^2(D_a)e_u + Y_a\hat{W}_a]. \tag{36}$$

Similarly, for the adversarial actor, we have

$$\dot{\hat{W}}_d = -\alpha_d[\nabla\phi H e_v - \nabla\phi H \tanh^2(D_d)e_v + Y_d\hat{W}_d]. \tag{37}$$

with $e_v = v_d - v_c = \lambda_2[\tanh(D_d) - \tanh(D_2)]$ with $D_2 = \frac{1}{2\lambda_2\gamma_2}H^{\mathrm{T}}(s)[\nabla\phi]^{\mathrm{T}}W_c$.

Theorem 9: Consider the system given in (12). Let the control input and the worst-case adversarial input be given by (33) and (34), respectively. Moreover, the critic learning is given by (27) and the tuning laws for the control input and adversarial input are (36) and (37), respectively. Suppose that the signal $\sigma/(1+\sigma^T\sigma)$ satisfies the PE condition, then the closed-loop system is uniformly ultimately bounded for a sufficient large basis with, $Y_a \geq \frac{M_a M_a^{\mathrm{T}}}{2}$, and $Y_d \geq \frac{M_d M_d^{\mathrm{T}}}{2}$ where $M_a = \nabla\phi G\lambda_1[\tanh(\kappa D_a) - \tanh(D_a)]$ and $M_d = \nabla\phi H\lambda_2[\tanh(\kappa D_d) - \tanh(D_d)]$.

Proof: Omitted here due to space limit. Interested readers are referred to our long version preprint. ■

IV. Simulation

To validate that our algorithm solves Problem 1, we apply our learning algorithm on the controlled Van-der-Pol oscillator given as

$$\dot{x} = \begin{bmatrix} x_2 \\ -x_1 + 0.5(1-x_2^2)x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ x_1 \end{bmatrix} u + \begin{bmatrix} 0 \\ x_1 \end{bmatrix} v. \tag{38}$$

We set $g(x) = h(x)$, and v is considered to be the perturbation of the control u. The LTL specification is given as $p \models \Diamond p_1 \wedge \Diamond p_2 \wedge \Box p_3$, with p_1 and p_2 and p_3 denotes the regions Ω_1, Ω_2 and Ω_3, respectively. Specifically, Ω_3 is a polygon defined by $c \leq Ax + r \leq C$ with $A = [0,1;4,1]$, $c = [-0.2;-1.9]$ and $C = [0.5;0.2]$. The FSA of p is shown in Figure 3. We use polynomials up to the 6th order as our basis functions ϕ. The path we choose to satisfy the LTL is T0_init → T0_S3 → accept_S0. The converse HJB method [22] cannot guarantee the safety and the presence of the adversarial input worsens it. Then we apply our algorithm described in Theorem 9 and the phase plot is demonstrated in Figures 4 and 5, respectively. It can be seen that there are multiple times that the adversarial input tries to get the trajectory out of the safety zone but eventually the control input prevails to keep safety while the LTL specification is satisfied.

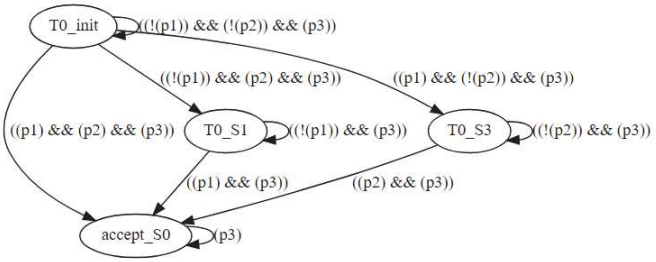

Fig. 3. Finite state automaton generated from formula $p \models \Diamond p_1 \wedge \Diamond p_2 \wedge \Box p_3$.

V. Conclusion

We propose a learning method to address the safe learning problem satisfying an LTL specification in adversarial environments. Via a finite state automaton, we first decompose the LTL specification into a sequence of TPBVPs. Then a system transformation is employed to guarantee safety. Then we use a learning method to solve the sub-problem safely. Two actors are used to approximate the control input and the adversarial input while a critic is used to approximate cost and proper tuning laws are.

Future work will focus on extending the work to automatic satisfaction of the state constraint, with a more general structure that works also in stochastic settings.

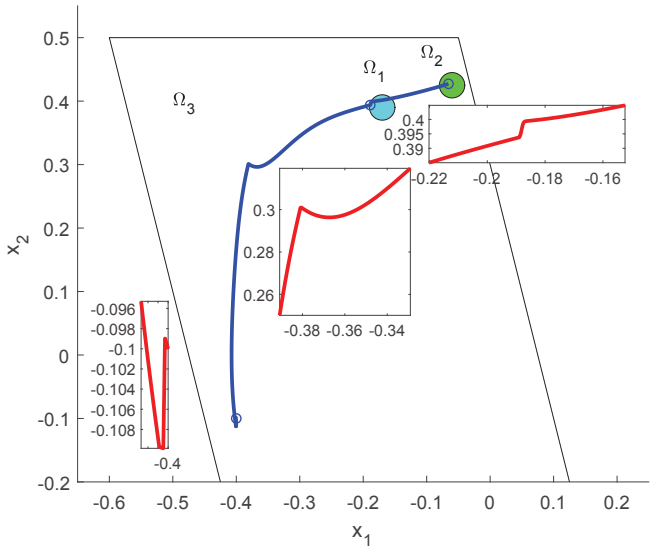

Fig. 4. Evolution of the phase plane. Ω_1 and Ω_2 are the cyan and green disks respectively and Ω_2 is the polygon. Blue circles denote the initial/terminal states of a sub-problem and the blue solid line is the overall trajectory.

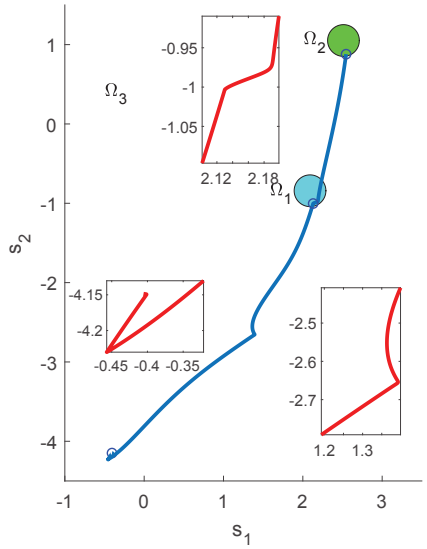

Fig. 5. Evolution of the phase plane in s space.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[2] 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 Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[3] R. Kamalapurkar, P. Walters, J. Rosenfeld, and W. Dixon, *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach*, ser. Communications and Control Engineering. Springer International Publishing, 2018.

[4] P. J. Werbos *et al.*, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*, vol. 15, pp. 493–525, 1992.

[5] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. IET Press, 2012.

[6] K. G. Vamvoudakis, P. J. Antsaklis, W. E. Dixon, J. P. Hespanha, F. L. Lewis, H. Modares, and B. Kiumarsi, "Autonomy and machine intelligence in complex systems: A tutorial," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 5062–5079.

[7] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[8] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[9] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[10] B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, "Adaptive neural control for output feedback nonlinear systems using a barrier lyapunov function," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1339–1345, 2010.

[11] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[12] Y. Chen, A. Hereid, H. Peng, and J. Grizzle, "Enhancing the performance of a safe controller via supervised learning for truck lateral control," 2018.

[13] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," 2018.

[14] M. Ohnishi, L. Wang, G. Notomista, and M. Egerstedt, "Safety-aware adaptive reinforcement learning with applications to brushbot navigation," *arXiv preprint arXiv:1801.09627*, 2018.

[15] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, 2018.

[16] A. K. Akametalu, S. Kaynama, J. F. Fisac, M. N. Zeilinger, J. H. Gillula, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes." in *CDC*. Citeseer, 2014, pp. 1424–1431.

[17] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[18] K. G. Vamvoudakis, M. F. Miranda, and J. P. Hespanha, "Asymptotically stable adaptive–optimal control algorithm with saturating actuators and relaxed persistence of excitation," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 11, pp. 2386–2398, 2016.

[19] 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.

[20] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1513–1525, 2013.

[21] 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.

[22] V. Nevistić and J. A. Primbs, "Constrained nonlinear optimal control: a converse HJB approach," 1996.