



# Safe Model-Based Reinforcement Learning for Systems With Parametric Uncertainties

S. M. Nahid Mahmud<sup>1</sup>, Scott A. Nivison<sup>2</sup>, Zachary I. Bell<sup>2</sup> and Rushikesh Kamalapurkar<sup>1\*</sup>

<sup>1</sup>School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK, United States, <sup>2</sup>Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL, United States

## OPEN ACCESS

### Edited by:

Yongping Pan,  
National University of Singapore,  
Singapore

### Reviewed by:

Mohamed Hamdy,  
University of Menoufia, Egypt  
Sayan Basu Roy,  
Indraprastha Institute of Information  
Technology Delhi, India

### \*Correspondence:

Rushikesh Kamalapurkar  
rushikesh.kamalapurkar@  
okstate.edu

### Specialty section:

This article was submitted to  
Robotic Control Systems,  
a section of the journal  
Frontiers in Robotics and AI

**Received:** 29 June 2021

**Accepted:** 27 September 2021

**Published:** 16 December 2021

### Citation:

Mahmud SMN, Nivison SA, Bell ZI and  
Kamalapurkar R (2021) Safe Model-  
Based Reinforcement Learning for  
Systems With  
Parametric Uncertainties.  
*Front. Robot. AI* 8:733104.  
doi: 10.3389/frobt.2021.733104

Reinforcement learning has been established over the past decade as an effective tool to find optimal control policies for dynamical systems, with recent focus on approaches that guarantee safety during the learning and/or execution phases. In general, safety guarantees are critical in reinforcement learning when the system is safety-critical and/or task restarts are not practically feasible. In optimal control theory, safety requirements are often expressed in terms of state and/or control constraints. In recent years, reinforcement learning approaches that rely on persistent excitation have been combined with a barrier transformation to learn the optimal control policies under state constraints. To soften the excitation requirements, model-based reinforcement learning methods that rely on exact model knowledge have also been integrated with the barrier transformation framework. The objective of this paper is to develop safe reinforcement learning method for deterministic nonlinear systems, with parametric uncertainties in the model, to learn approximate constrained optimal policies without relying on stringent excitation conditions. To that end, a model-based reinforcement learning technique that utilizes a novel filtered concurrent learning method, along with a barrier transformation, is developed in this paper to realize simultaneous learning of unknown model parameters and approximate optimal state-constrained control policies for safety-critical systems.

**Keywords:** safe learning, barrier transformation, model-based reinforcement learning, control theory, parameter estimation, nonlinear systems

## 1 INTRODUCTION

Due to advantages such as repeatability, accuracy, and lack of physical fatigue, autonomous systems have been increasingly utilized to perform tasks that are dull, dirty, or dangerous. Autonomy in safety-critical applications such as autonomous driving and unmanned flight relies on the ability to synthesize safe controllers. To improve robustness to parametric uncertainties and changing objectives and models, autonomous systems also need the ability to simultaneously synthesize and execute control policies online and in real time. This paper concerns reinforcement learning (RL), which has been established as an effective tool for safe policy synthesis for both known and uncertain dynamical systems with finite state and action spaces [see, e.g., Sutton and Barto (1998); Doya (2000)].

RL typically requires a large number of iterations due to sample inefficiency [see, e.g., Sutton and Barto (1998)]. Sample efficiency in RL can be improved via model-based reinforcement learning (MBRL); however, MBRL methods are prone to failure due to inaccurate models [see, e.g.,

Kamalapurkar et al. (2016a,b, 2018)]. Online MBRL methods that handle modeling uncertainties are motivated by complex tasks that require systems to operate in dynamic environments with changing objectives and system models. Accurate models of the system and the environment are generally not available due to sparsity of data. In the past, MBRL techniques under the umbrella of approximate dynamic programming (ADP) have been successfully utilized to solve reinforcement learning problems online with model uncertainty [see, e.g., Modares et al. (2013); Kiumarsi et al. (2014); Qin et al. (2014)]. ADP utilizes parametric methods such as neural networks (NNs) to approximate the value function and the system model online. By obtaining an approximation of both the value function and the system model, a stable closed-loop adaptive control policy can be developed [see, e.g., Vamvoudakis et al. (2009); Lewis and Vrabie (2009); Bertsekas (2011); Bhasin et al. (2012); Liu and Wei (2014)].

Real-world optimal control applications typically include constraints on states and/or inputs that are critical for safety [see, e.g., He et al. (2017)]. ADP was successfully extended to address input constrained control problems in Modares et al. (2013) and Vamvoudakis et al. (2016). The state-constrained ADP problem was studied in the context of obstacle avoidance in Walters et al. (2015) and Deptula et al. (2020), where an additional term that penalizes proximity to obstacles was added to the cost function. Since the added proximity penalty in Walters et al. (2015) was finite, the ADP feedback could not guarantee obstacle avoidance, and an auxiliary controller was needed. In Deptula et al. (2020), a barrier-like function was used to ensure unbounded growth of the proximity penalty near the obstacle boundary. While this approach results in avoidance guarantees, it relies on the relatively strong assumption that the value function is continuously differentiable over a compact set that contains the obstacles in spite of penalty-induced discontinuities in the cost function.

Control Barrier Function (CBF) is another approach to guarantee safety in safety-critical systems [see e.g., Ames et al. (2017)], with recent applications to the safe reinforcement learning problems [see e.g., Choi et al. (2020); Cohen and Belta (2020); Marvi and Kiumarsi (2021)]. Choi et al. (2020) have addressed the issue of model uncertainty in safety-critical control with an RL-based data-driven approach. A drawback of this approach is that it requires a nominal controller that keeps the system stable during the learning phase, which may not be always possible to design. In Marvi and Kiumarsi (2021), the authors develop a safe off-policy RL scheme which trades-off between safety and performance. In Cohen and Belta (2020), the authors develop a safe RL scheme in which the proximity penalty approach from Deptula et al. (2020) is cast into the framework of CBFs. While the control barrier function results in safety guarantees, the existence of a smooth value function, in spite of a nonsmooth cost function, needs to be assumed. Furthermore, to facilitate parametric approximation of the value function, the existence of a forward invariant compact set in the interior of the safe set needs to be established. Since the invariant set needs to be in the interior of the safe set, the penalty becomes superfluous,

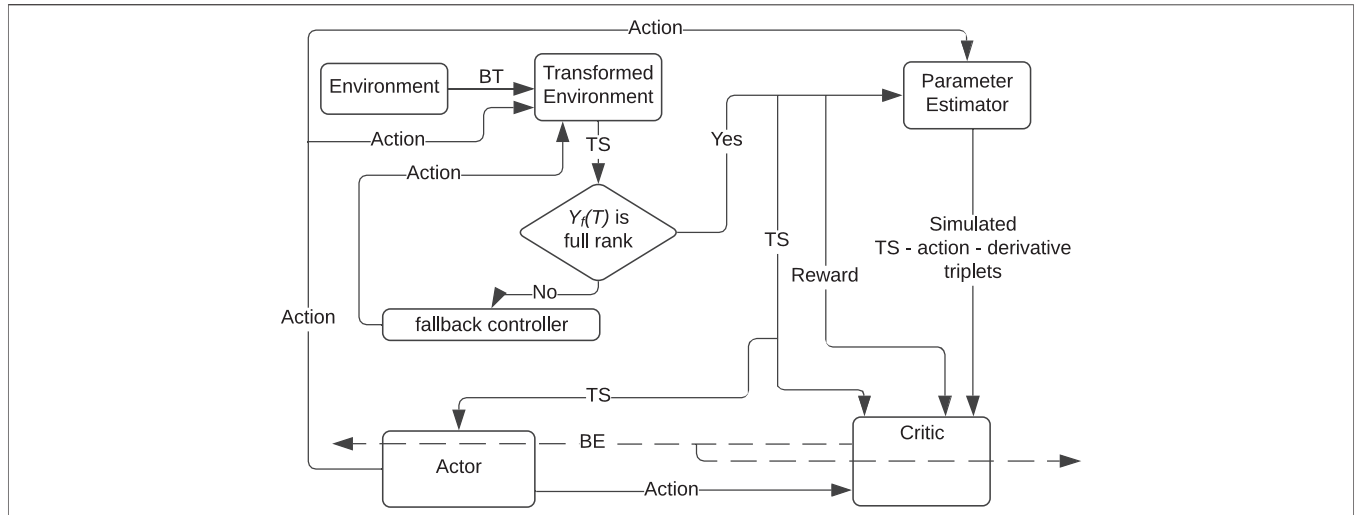
and safety can be achieved through conventional Lyapunov methods.

This paper is inspired by a safe reinforcement learning technique, recently developed in Yang et al. (2019), based on the idea of transforming a state and input constrained nonlinear optimal control problem into an unconstrained one with a type of saturation function, introduced in Graichen and Petit (2009), and Bechlioulis and Rovithakis (2009). In Yang et al. (2019), the state constrained optimal control problem is transformed using a barrier transformation (BT), into an equivalent, unconstrained optimal control problem. A learning technique is then used to synthesize the feedback control policy for the unconstrained optimal control problem. The controller for the original system is then derived from the unconstrained approximate optimal policy by inverting the barrier transformation. In Greene et al. (2020), the restrictive persistence of excitation requirement in Yang et al. (2019) is softened using model-based reinforcement learning (MBRL), where exact knowledge of the system dynamics is utilized in the barrier transformation.

One of the primary contributions of this paper is a detailed analysis of the connection between the transformed dynamics and the original dynamics, which is missing from results such as Yang et al. (2019), Greene et al. (2020), and Yang et al. (2020). While the stability of the transformed dynamics under the designed controllers is established in results such as Yang et al. (2019), Greene et al. (2020), and Yang et al. (2020), the implications of the behavior of the transformed system on the original system are not examined. In this paper, it is shown that the trajectories of the original system are related to the trajectories of the transformed system *via* the barrier transformation as long as the trajectories of the transformed system are complete.

While the transformation in Yang et al. (2019) and Greene et al. (2020) results in verifiable safe controllers, it requires exact knowledge of the system model, which is often difficult to obtain. Another primary contribution of this paper is the development of a novel filtered concurrent learning technique for online model learning, and its integration with the barrier transformation method, to yield a novel MBRL solution to the online state-constrained optimal control problem under parametric uncertainty. The developed MBRL method learns an approximate optimal control policy in the presence of parametric uncertainties for safety critical systems while maintaining stability and safety during the learning phase. The inclusion of filtered concurrent learning makes the controller robust to modeling errors and guarantees local stability under a *finite* (as opposed to *persistent*) excitation condition.

In the following, the problem is formulated in **Section 2** and the BT is described and analyzed in **Section 3**. A novel parameter estimation technique is detailed in **Section 4** and a model-based reinforcement learning technique for synthesizing feedback control policy in the transformed coordinates is developed in **Section 5**. In **Section 6**, a Lyapunov-based analysis is utilized to establish practical stability of the closed-loop system resulting from the



**FIGURE 1 |** The developed BT MBRL framework. The control system consists of a model-based barrier-actor-critic-estimator architecture. In addition to the transformed state-action measurements, the critic also utilizes states, actions, and the corresponding state derivatives, evaluated at arbitrarily selected points in the state space, to learn the value function. In the figure, BT: Barrier Transformation; TS: Transformed State; BE: Bellman Error.

developed MBRL technique in the transformed coordinates, which guarantees that the safety requirements are satisfied in the original coordinates. Simulation results in **Section 7** demonstrate the performance of the developed method and analyze its sensitivity to various design parameters, followed by a comparison of the performance of the developed MBRL approach to an offline pseudospectral optimal control method. Strengths and limitations of the developed method are discussed in **Section 8**, along with possible extensions.

## 2 PROBLEM FORMULATION

### 2.1 Control Objective

Consider a continuous-time affine nonlinear dynamical system

$$\dot{x} = f(x)\theta + g(x)u, \quad (1)$$

where  $x = [x_1; \dots; x_n] \in \mathbb{R}^n$  is the system state,  $\theta \in \mathbb{R}^p$  are the unknown parameters,  $u \in \mathbb{R}^q$  is the control input, and the functions  $f: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times p}$  and  $g: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times q}$  are known, locally Lipschitz functions. In the following,  $[a; b]$  denotes the vector  $[a \ b]^T$  and  $(v)_i$  denotes the  $i$ th component of the vector  $v$ .

The objective is to design a controller  $u$  for the system in (1) such that starting from a given feasible initial condition  $x^0$ , the trajectories  $x(\cdot)$  decay to the origin and satisfy  $x_i(t) \in (a_i, A_i), \forall t \geq 0$ , where  $i = 1, 2, \dots, n$  and  $a_i < 0 < A_i$ . While MBRL methods such as those detailed in Kamalapurkar et al. (2018) guarantee stability of the closed loop, state constraints are typically difficult to establish without extensive trial and error. To achieve the stated objective, an MBRL framework (see **Figure 1**) is developed in this paper, where a BT is used to guarantee satisfaction of state constraints.

## 3 BARRIER TRANSFORMATION

### 3.1 Design

Let the function  $b: \mathbb{R} \rightarrow \mathbb{R}$ , referred to as the barrier function (BF), be defined as

$$b_{(a_i, A_i)}(y) := \log \frac{A_i(a_i - y)}{a_i(A_i - y)}, \quad \forall i = 1, 2, \dots, n. \quad (2)$$

Define  $b_{(a, A)}: \mathbb{R}^n \rightarrow \mathbb{R}^n$  as  $b_{(a, A)}(x) := [b_{(a_1, A_1)}((x)_1); \dots; b_{(a_n, A_n)}((x)_n)]$  with  $a = [a_1; \dots; a_n]$  and  $A = [A_1; \dots; A_n]$ . The inverse of (2) on the interval  $(a_i, A_i)$ , is given by

$$b_{(a_i, A_i)}^{-1}(y) = a_i A_i \frac{e^y - 1}{a_i e^y - A_i}. \quad (3)$$

Define  $b_{(a, A)}^{-1}: \mathbb{R}^n \rightarrow \mathbb{R}^n$  as  $b_{(a, A)}^{-1}(s) := [b_{(a_1, A_1)}^{-1}((s)_1); \dots; b_{(a_n, A_n)}^{-1}((s)_n)]$ . Taking the derivative of (3) with respect to  $y$  yields

$$\frac{db_{(a_i, A_i)}^{-1}(y)}{dy} = \frac{1}{B_i(y)}, \quad \text{where } B_i(y) := \frac{a_i^2 e^y - 2a_i A_i + A_i^2 e^{-y}}{A_i a_i^2 - a_i A_i^2}. \quad (4)$$

Consider the BF based state transformation

$$s_i := b_{(a_i, A_i)}(x_i), \quad x_i = b_{(a_i, A_i)}^{-1}(s_i), \quad (5)$$

where  $s := [s_1; \dots; s_n]$  denotes the transformed state. In the following derivation, whenever clear from the context, the subscripts  $a_i$  and  $A_i$  of the BF and its inverse are suppressed for brevity. The time derivative of the transformed state can be computed using the chain rule as  $\dot{s}_i = B_i(s_i)\dot{x}_i$  which yields the transformed dynamics

$$\dot{s}_i = B_i(s_i)(f(x)\theta + g(x)u)_i. \quad (6)$$

The dynamics of the transformed state can then be expressed as

$$\dot{s} = F(s) + G(s)u, \quad (7)$$

where  $F(s) := y(s)\theta$ ,  $(y(s))_i := B_i(s_i)(f(b^{-1}(s)))_i \in \mathbb{R}^{1 \times p}$ , and  $(G(s))_i := B_i(s_i)(g(b^{-1}(s)))_i \in \mathbb{R}^{1 \times q}$ .

Continuous differentiability of  $b^{-1}$  implies that  $F$  and  $G$  are locally Lipschitz continuous. Furthermore,  $f(0) = 0$  along with the fact that  $b^{-1}(0) = 0$  implies that  $F(0) = 0$ . As a result, for all compact sets  $\Omega \subset \mathbb{R}^n$  containing the origin,  $G$  is bounded on  $\Omega$  and there exists a positive constant  $L_y$  such that  $\forall s \in \Omega$ ,  $\|y(s)\| \leq L_y \|s\|$ . The following section relates the solutions of the original system to the solutions of the transformed system.

### 3.2 Analysis

In the following lemma, the trajectories of the original system and the transformed system are shown to be related by the barrier transformation provided the trajectories of the transformed system are *complete* [see, e.g., Page 33 of Sanfelice (2021)]. The completeness condition is not vacuous, it is not difficult to construct a system where the transformed trajectories escape to infinity in finite time, while the original trajectories are complete. For example, consider the system  $\dot{x} = x + x^2u$  with  $x \in \mathbb{R}$  and  $u \in \mathbb{R}$ . All nonzero solutions of the corresponding transformed system  $\dot{s} = B_1(s)b_{(-0.5,0.5)}^{-1}(s) + B_1(s)(b_{(-0.5,0.5)}^{-1}(s))^2u$  under the feedback  $\zeta(s, t) = -b_{(-0.5,0.5)}^{-1}(s)$  escape in finite time. However, all nonzero solutions of the original system under the feedback  $\xi(x, t) = \zeta(b_{(-0.5,0.5)}(x), t) = -x$  converge to either  $-1$  or  $1$ .

**Lemma 1.** *If  $t \mapsto \Phi(t, b(x^0), \xi)$  is a complete Carathéodory solution to (7), starting from the initial condition  $b(x^0)$ , under the feedback policy  $(s, t) \mapsto \zeta(s, t)$  and  $t \mapsto \Lambda(t, x^0, \xi)$  is a Carathéodory solution to (1), starting from the initial condition  $x^0$ , under the feedback policy  $(x, t) \mapsto \xi(x, t)$ , defined as  $\xi(x, t) = \zeta(b(x), t)$ , then  $\Lambda(\cdot, x^0, \xi)$  is complete and  $\Lambda(t, x^0, \xi) = b^{-1}(\Phi(t, b(x^0), \zeta))$  for all  $t \in \mathbb{R}_{\geq 0}$ .*

*Proof.* See Lemma 1 in the **Supplementary Material**.

Note that the feedback  $\xi$  is well-defined at  $x$  only if  $b(x)$  is well-defined, which is the case whenever  $x$  is inside the barrier. As such, the main conclusion of Lemma 1 also implies that  $\Lambda(\cdot, x^0, \xi)$  remains inside the barrier. It is thus inferred from Lemma 1 that if the trajectories of (7) are bounded and decay to a neighborhood of the origin under a feedback policy  $(s, t) \mapsto \zeta(s, t)$ , then the feedback policy  $(x, t) \mapsto \zeta(b(x), t)$ , when applied to the original system in (1), achieves the control objective stated in **Section 2.1**.

To achieve BT MBRL in the presense of parametric uncertainties, the following section develops a novel parameter estimator.

## 4 PARAMETER ESTIMATION

The following parameter estimator design is motivated by the subsequent Lyapunov analysis, and is inspired by the finite-time estimator in Adetola and Guay (2008) and the filtered

concurrent learning (FCL) method in Roy et al. (2016). Estimates of the unknown parameters,  $\hat{\theta} \in \mathbb{R}^p$ , are generated using the filter

$$\dot{Y} = \begin{cases} y(s), & \|Y_f\| \leq \overline{Y}_f, \\ 0, & \text{otherwise,} \end{cases} \quad Y(0) = 0, \quad (8)$$

$$\dot{Y}_f = \begin{cases} Y^T Y, & \|Y_f\| \leq \overline{Y}_f, \\ 0, & \text{otherwise,} \end{cases} \quad Y_f(0) = 0, \quad (9)$$

$$\dot{G}_f = \begin{cases} G(s)u, & \|Y_f\| \leq \overline{Y}_f, \\ 0, & \text{otherwise,} \end{cases} \quad G_f(0) = 0, \quad (10)$$

$$\dot{X}_f = \begin{cases} Y^T(s - s^0 - G_f), & \|Y_f\| \leq \overline{Y}_f, \\ 0, & \text{otherwise,} \end{cases} \quad X_f(0) = 0, \quad (11)$$

where  $s^0 = [b(x_1^0); \dots; b(x_n^0)]$ , and the update law

$$\dot{\hat{\theta}} = \beta_1 Y_f^T (X_f - Y_f \hat{\theta}), \quad \hat{\theta}(0) = \theta^0, \quad (12)$$

where  $\beta_1$  is a symmetric positive definite gain matrix and  $\overline{Y}_f$  is a tunable upper bound on the filtered regressor  $Y_f$ .

**Equations 7–12** constitute a nonsmooth system of differential equations

$$\dot{z} = h(z, u) = \begin{cases} h_1(z, u), & \|Y_f\| \leq \overline{Y}_f, \\ h_2(z, u), & \text{otherwise,} \end{cases} \quad (13)$$

where  $z = [s; \text{vec}(Y); \text{vec}(Y_f); G_f; X_f; \hat{\theta}]$ ,  $h_1(z, u) = [F(s) + G(s)u; \text{vec}(y(s)); \text{vec}(Y^T Y); G(s)u; Y^T(s - s^0 - G_f); \beta_1 Y_f^T (X_f - Y_f \hat{\theta})]$ , and  $h_2(z, u) = [F(s) + G(s)u; 0; 0; 0; \beta_1 Y_f^T (X_f - Y_f \hat{\theta})]$ . Since  $\|Y_f\|$  is non-decreasing in time, it can be shown that (13) admits Carathéodory solutions.

**Lemma 2.** *If  $\|Y_f\|$  is non-decreasing in time then (13) admits Carathéodory solutions.*

*Proof.* see Lemma 2 in **Supplementary Material**.

Note that (9), expressed in the integral form

$$Y_f(t) = \int_0^{t_3} Y^T(\tau) Y(\tau) d\tau, \quad (14)$$

where  $t_3 := \inf\{t \geq 0 \mid \|Y_f(t)\| \leq \overline{Y}_f\}$ , along with (11), expressed in the integral form

$$X_f(t) = \int_0^{t_3} Y^T(\tau) (s(\tau) - s^0 - G_f(\tau)) d\tau, \quad (15)$$

and the fact that  $s(\tau) - s^0 - G_f(\tau) = Y(\tau)\theta$ , can be used to conclude that  $X_f(t) = Y_f(t)\theta$ , for all  $t \geq 0$ . As a result, a measure for the parameter estimation error can be obtained using known quantities as  $Y_f \tilde{\theta} = X_f - Y_f \hat{\theta}$ , where  $\tilde{\theta} := \theta - \hat{\theta}$ . The dynamics of the parameter estimation error can then be expressed as

$$\dot{\tilde{\theta}} = -\beta_1 Y_f^T Y_f \tilde{\theta}. \quad (16)$$

The filter design is thus motivated by the fact that if the matrix  $Y_f^T Y_f$  is positive definite, uniformly in  $t$ , then the Lyapunov function  $V_1(\tilde{\theta}) = \frac{1}{2} \tilde{\theta}^T \beta_1^{-1} \tilde{\theta}$  can be used to establish convergence of the parameter estimation error to the origin. Initially,  $Y_f^T Y_f$  is a matrix of zeros. To ensure that there exists some finite time  $T$  such that  $Y_f^T(t) Y_f(t)$  is positive definite,

uniformly in  $t$  for all  $t \geq T$ , the following *finite* excitation condition is imposed.

**Assumption 3.** There exists a time instance  $T > 0$  such that  $Y_f(T)$  is full rank.

Note that the minimum eigenvalue of  $Y_f$  is trivially non-decreasing for  $t \geq t_3$  since  $Y_f(t)$  is constant  $\forall t \geq t_3$ . Indeed, for  $t_4 \leq t_5 \leq t_3$ ,  $Y_f(t_5) = Y_f(t_4) + \int_{t_4}^{t_5} Y^T(\tau)Y(\tau)d\tau$ . Since  $Y_f(t_4)$  is positive semidefinite, and so is the integral  $\int_{t_4}^{t_5} Y^T(\tau)Y(\tau)d\tau$ , we conclude that  $\lambda_{\min}(Y_f(t_5)) \geq \lambda_{\min}(Y_f(t_4))$ . As a result,  $t \mapsto \lambda_{\min}(Y_f(t))$  is non-decreasing. Therefore, if **Assumption 3** is satisfied at  $t = T$ , then  $Y_f(t)$  is also full rank for all  $t \geq T$ . Similar to other MBRL methods that rely on system identification [see e.g., Chapter 4 of Kamalapurkar et al. (2018)], the following assumption is needed to ensure boundedness of the state trajectories over the interval  $[0, T]$ .

**Assumption 4.** A fallback controller  $\psi: \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^q$  that keeps the trajectories of (7) inside a known bounded set over the interval  $[0, T]$ , without requiring the knowledge of  $\theta$ , is available.

If a fallback controller that satisfies **Assumption 4** is not available, then, under the additional assumption that the trajectories of (7) are exciting over the interval  $[0, T]$ , such a controller can be learned online, while maintaining system stability, using model-free reinforcement learning techniques such as Bhasin et al. (2013); Vrabie and Lewis (2010), and Modares et al. (2014).

**Remark 5.** While the analysis of the developed technique dictates that a different stabilizing controller should be used over the time interval  $[0, T]$ , typically, similar to the examples in **Sections 7.1** and **7.2**, the transient response of the developed controller provides sufficient excitation so that  $T$  is small (in the examples provided in **Sections 7.1** and **7.2**,  $T$  is the order of  $10^{-5}$  and  $10^{-6}$ , respectively), and a different stabilizing controller is not needed in practice.

## 5 MODEL-BASED REINFORCEMENT LEARNING

Lemma 1 implies that if a feedback controller that practically stabilizes the transformed system in (7) is designed, then the same feedback controller, applied to the original system by inverting the BT, also achieves the control objective stated in **Section 2.1**. In the following, a controller that practically stabilizes (7) is designed as an estimate of the controller that minimizes the infinite horizon cost.

$$J(u(\cdot)) := \int_0^\infty r(\phi(\tau, s^0, u(\cdot)), u(\tau))d\tau, \quad (17)$$

over the set  $\mathcal{U}$  of piecewise continuous functions  $t \mapsto u(t)$ , subject to (7), where  $\phi(\tau, s^0, u(\cdot))$  denotes the trajectory of (7), evaluated at time  $\tau$ , starting from the state  $s^0$  and under the controller  $u(\cdot)$ ,  $r(s, u) := s^T Qs + u^T R u$ , and  $Q \in \mathbb{R}^{n \times n}$  and  $R \in \mathbb{R}^{q \times q}$  are symmetric positive definite (PD) matrices<sup>1</sup>.

<sup>1</sup>For applications with bounded control inputs, a non-quadratic penalty function similar to **Eq. 17** of Yang et al. (2020) can be incorporated in (17).

Assuming that an optimal controller exists, let the optimal value function, denoted by  $V^*: \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}$ , be defined as

$$V^*(s) := \min_{u(\cdot) \in \mathcal{U}_{[t, \infty)}} \int_t^\infty r(\phi(\tau, s, u_{[t, \tau]}(\cdot)), u(\cdot))d\tau, \quad (18)$$

where  $u_I$  and  $\mathcal{U}_I$  are obtained by restricting the domains of  $u$  and functions in  $\mathcal{U}_I$  to the interval  $I \subseteq \mathbb{R}$ , respectively. Assuming that the optimal value function is continuously differentiable, it can be shown to be the unique positive definite solution of the Hamilton-Jacobi-Bellman (HJB) equation [see, e.g., Theorem 1.5 of Kamalapurkar et al. (2018)].

$$\min_{u \in \mathbb{R}^q} (\nabla_s V(s)(F(s) + G(s)u) + s^T Qs + u^T R u) = 0, \quad (19)$$

where  $\nabla_{(\cdot)} := \frac{\partial}{\partial(\cdot)}$ . Furthermore, the optimal controller is given by the feedback policy  $u(t) = u^*(\phi(t, s, u_{[0, t]}))$  where  $u^*: \mathbb{R}^n \rightarrow \mathbb{R}^q$  defined as

$$u^*(s) := -\frac{1}{2}R^{-1}G(s)^T(\nabla_s V^*(s))^T. \quad (20)$$

**Remark 6.** In the developed method, the cost function is selected to be quadratic in the transformed coordinates. However, a physically meaningful cost function is more likely to be available in the original coordinates. If such a cost function is available, it can be transformed from the original coordinates to the barrier coordinates using the inverse barrier function, to yield a cost function that is not quadratic in the state. While the analysis in this paper addresses the quadratic case, it can be extended to address the non-quadratic case with minimal modifications as long as  $s \mapsto r(s, u)$  is positive definite for all  $u \in \mathbb{R}^q$ .

### 5.1 Value Function Approximation

Since computation of analytical solutions of the HJB equation is generally infeasible, especially for systems with uncertainty, parametric approximation methods are used to approximate the value function  $V^*$  and the optimal policy  $u^*$ . The optimal value function is expressed as

$$V^*(s) = W^T \sigma(s) + \epsilon(s), \quad (21)$$

where  $W \in \mathbb{R}^L$  is an unknown vector of bounded weights,  $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}^L$  is a vector of continuously differentiable nonlinear activation functions such that  $\sigma(0) = 0$  and  $\nabla_s \sigma(0) = 0$ ,  $L \in \mathbb{N}$  is the number of basis functions, and  $\epsilon: \mathbb{R}^n \rightarrow \mathbb{R}$  is the reconstruction error.

The basis functions are selected such that the approximation of the functions and their derivatives is uniform over the compact set  $\chi \subset \mathbb{R}^n$  so that given a positive constant  $\bar{\epsilon} \in \mathbb{R}$ , there exists  $L \in \mathbb{N}$  and known positive constants  $\bar{W}$  and  $\bar{\sigma}$  such that  $\|W\| \leq \bar{W}$ ,  $\sup_{s \in \chi} \|\epsilon(s)\| \leq \bar{\epsilon}$ ,  $\sup_{s \in \chi} \|\nabla_s \epsilon(s)\| \leq \bar{\epsilon}$ ,  $\sup_{s \in \chi} \|\sigma(s)\| \leq \bar{\sigma}$ , and  $\sup_{s \in \chi} \|\nabla_s \sigma(s)\| \leq \bar{\sigma}$  [see, e.g., Sauvigny (2012)]. Using (19), a representation of the optimal controller, that uses the same basis as the optimal value function, is derived as

$$u^*(s) = -\frac{1}{2}R^{-1}G^T(s)(\nabla_s \sigma^T(s)W + \nabla_s \epsilon^T(s)). \quad (22)$$



Since the ideal weights,  $W$ , are unknown, an actor-critic approach is used in the following to estimate  $W$ . To that end, let the NN estimates  $\hat{V}: \mathbb{R}^n \times \mathbb{R}^L \rightarrow \mathbb{R}$  and  $\hat{u}: \mathbb{R}^n \times \mathbb{R}^L \rightarrow \mathbb{R}^q$  be defined as

$$\hat{V}(s, \hat{W}_c) := \hat{W}_c^T \sigma(s), \quad (23)$$

$$\hat{u}(s, \hat{W}_a) := -\frac{1}{2} R^{-1} G^T(s) \nabla_s \sigma^T(s) \hat{W}_a, \quad (24)$$

where the critic weights,  $\hat{W}_c \in \mathbb{R}^L$  and actor weights,  $\hat{W}_a \in \mathbb{R}^L$  are estimates of the ideal weights,  $W$ .

## 5.2 Bellman Error

Substituting (23) and (24) into (19) results in a residual term,  $\hat{\delta}: \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^p \rightarrow \mathbb{R}$ , referred to as Bellman Error (BE), defined as

$$\begin{aligned} \hat{\delta}(s, \hat{W}_c, \hat{W}_a, \hat{\theta}) := & \nabla_s \hat{V}(s, \hat{W}_c) (y(s) \hat{\theta} + G(s) \hat{u}(s, \hat{W}_a)) \\ & + \hat{u}(s, \hat{W}_a)^T R \hat{u}(s, \hat{W}_a) + s^T Q s. \end{aligned} \quad (25)$$

Traditionally, online RL methods require a persistence of excitation (PE) condition to be able learn the approximate control policy [see, e.g., Modares et al. (2013); Kamalapurkar et al. (2016a); Kiumarsi et al. (2014)]. Guaranteeing PE a priori and verifying PE online are both typically impossible. However, using virtual excitation facilitated by model-based BE extrapolation, stability and convergence of online RL can be established under a PE-like condition that, while impossible to guarantee a priori, can be verified online by monitoring the minimum eigenvalue of a matrix in the subsequent **Assumption 7** [see, e.g., Kamalapurkar et al. (2016b)]. Using the system model, the BE can be evaluated at any arbitrary point in the state space. Virtual excitation can then be implemented by selecting a set of states  $\{s_k \mid k = 1, \dots, N\}$  and evaluating the BE at this set of states to yield

$$\begin{aligned} \hat{\delta}_k(s_k, \hat{W}_c, \hat{W}_a, \hat{\theta}) := & \nabla_{s_k} \hat{V}(s_k, \hat{W}_c) (y_k \hat{\theta} + G_k \hat{u}(s_k, \hat{W}_a)) \\ & + \hat{u}(s_k, \hat{W}_a)^T R \hat{u}(s_k, \hat{W}_a) + s_k^T Q s_k, \end{aligned} \quad (26)$$

where,  $\nabla_{s_k} := \frac{\partial}{\partial s_k}$ ,  $y_k := y(s_k)$  and  $G_k := G(s_k)$ . Defining the actor and critic weight estimation errors as  $\tilde{W}_c := W - \hat{W}_c$  and  $\tilde{W}_a := W - \hat{W}_a$  and substituting the estimates (21) and (22) into (19), and subtracting from (25) yields the analytical BE that can be expressed in terms of the weight estimation errors as

$$\hat{\delta} = -\omega^T \tilde{W}_c + \frac{1}{4} \tilde{W}_a^T G_\sigma \tilde{W}_a - W^T \nabla_s \sigma y \tilde{\theta} + \Delta, \quad (27)$$

where  $\Delta := \frac{1}{2} W^T \nabla_s \sigma G_R \nabla_s \epsilon^T + \frac{1}{4} G_\epsilon - \nabla_s \epsilon F$ ,  $G_R := G R^{-1} G^T \in \mathbb{R}^{n \times n}$ ,  $G_\epsilon := \nabla_s \epsilon G_R \nabla_s \epsilon^T \in \mathbb{R}$ ,  $G_\sigma := \nabla_s \sigma G R^{-1} G^T \nabla_s \sigma^T \in \mathbb{R}^{L \times L}$ , and  $\omega := \nabla_s \sigma (y \tilde{\theta} + G \hat{u}(s, \hat{W}_a)) \in \mathbb{R}^L$ . In (27) and the rest of the manuscript, the dependence of various functions on the state,  $s$ , is omitted for brevity whenever it is clear from the context. Similarly, (26) implies that

$$\hat{\delta}_k = -\omega_k^T \tilde{W}_c + \frac{1}{4} \tilde{W}_a^T G_{\sigma_k} \tilde{W}_a - W^T \nabla_{s_k} \sigma_k y_k \tilde{\theta} + \Delta_k, \quad (28)$$

where,  $F_k := F(s_k)$ ,  $\epsilon_k := \epsilon(s_k)$ ,  $\sigma_k := \sigma(s_k)$ ,  $\Delta_k := \frac{1}{2} W^T \nabla_{s_k} \sigma_k G_{R_k} \nabla_{s_k} \epsilon_k^T + \frac{1}{4} G_{\epsilon_k} - \nabla_{s_k} \epsilon_k F_k G_{\epsilon_k} := \nabla_{s_k} \epsilon_k G_{R_k} \nabla_{s_k} \epsilon_k^T$ ,  $\omega_k := \nabla_{s_k} \sigma_k (y_k \tilde{\theta} + G_k \hat{u}(s_k, \hat{W}_a)) \in \mathbb{R}^L$ ,  $G_{R_k} := G_k R^{-1} G_k^T \in \mathbb{R}^{n \times n}$  and  $G_{\sigma_k} := \nabla_{s_k} \sigma_k G_k R^{-1} G_k^T \nabla_{s_k} \sigma_k^T \in \mathbb{R}^{L \times L}$ . Note that  $\sup_{s \in \chi} \|\Delta\| \leq d \bar{\epsilon}$  and if  $s_k \in \chi$  then  $\|\Delta_k\| \leq d \bar{\epsilon}_k$ , for some constant  $d > 0$ . While the extrapolation states  $s_k$  are assumed to be constant in this analysis for ease of exposition, the analysis extends in a straightforward manner to time-varying extrapolation states that are confined to a compact neighborhood of the origin.

## 5.3 Update Laws for Actor and Critic Weights

The actor and the critic weights are held at their initial values over the interval  $[0, T]$  and starting at  $t = T$ , using the instantaneous BE  $\hat{\delta}$  from (25) and extrapolated BEs  $\hat{\delta}_k$  from (26), the weights are updated according to.

$$\dot{\hat{W}}_c = -k_{c1} \Gamma \frac{\omega}{\rho} \hat{\delta} - \frac{k_{c2}}{N} \Gamma \sum_{k=1}^N \frac{\omega_k}{\rho_k} \hat{\delta}_k, \quad (29)$$

$$\dot{\Gamma} = \beta \Gamma - k_{c1} \Gamma \frac{\omega \omega^T}{\rho^2} \Gamma - \frac{k_{c2}}{N} \Gamma \sum_{k=1}^N \frac{\omega_k \omega_k^T}{\rho_k^2} \Gamma, \quad (30)$$

$$\dot{\hat{W}}_a = -k_{a1} (\hat{W}_a - \hat{W}_c) - k_{a2} \hat{W}_a + \frac{k_{c1} G_\sigma^T \hat{W}_a \omega^T}{4\rho} \hat{W}_c + \sum_{k=1}^N \frac{k_{c2} G_{\sigma_k}^T \hat{W}_a \omega_k^T}{4N \rho_k} \hat{W}_c, \quad (31)$$

with  $\Gamma(t_0) = \Gamma_0$ , where  $\Gamma: \mathbb{R}_{\geq t_0} \rightarrow \mathbb{R}^{L \times L}$  is a time-varying least-squares gain matrix,  $\rho(t) := 1 + \gamma_1 \omega^T(t) \omega(t)$ ,  $\rho_k(t) := 1 + \gamma_1 \omega_k^T(t) \omega_k(t)$ ,  $\beta > 0 \in \mathbb{R}$  is a constant forgetting factor, and  $k_{c1}, k_{c2}, k_{a1}, k_{a2} > 0 \in \mathbb{R}$  are constant adaptation gains. The control commands sent to the system are then computed using the actor weights as

$$u(t) = \begin{cases} \psi(s(t), t), & 0 < t < T, \\ \hat{u}(s(t), \hat{W}_a(t)), & t \geq T, \end{cases} \quad (32)$$

where the controller  $\psi$  was introduced in **Assumption 4**. The following verifiable PE-like rank condition is then utilized in the stability analysis.

**Assumption 7.** There exists a constant  $\underline{c}_3 > 0$  such that the set of points  $\{s_k \in \mathbb{R}^n \mid k = 1, \dots, N\}$  satisfies

$$\underline{c}_3 I_L \leq \inf_{t \in \mathbb{R}_{\geq T}} \left( \frac{1}{N} \sum_{k=1}^N \frac{\omega_k(t) \omega_k^T(t)}{\rho_k^2(t)} \right). \quad (33)$$

Since  $\omega_k$  is a function of the weight estimates  $\hat{\theta}$  and  $\hat{W}_a$ , **Assumption 7** cannot be guaranteed a priori. However, unlike the PE condition, **Assumption 7** does not impose excitation requirements on the system trajectory, the excitation requirements are imposed on a user-selected set of points in the state space. Furthermore, **Assumption 7** can be verified online. Since  $\lambda_{\min} \left( \sum_{k=1}^N \frac{\omega_k(t) \omega_k^T(t)}{\rho_k^2(t)} \right)$  is non-decreasing in the number of samples,  $N$ , **Assumption 7** can be met, heuristically, by increasing the number of samples.

## 6 STABILITY ANALYSIS

In the following theorem, the stability of the trajectories of the transformed system, and the estimation errors  $\tilde{W}_c$ ,  $\tilde{W}_a$ , and  $\tilde{\theta}$  are shown.

**Theorem 8.** Provided **Assumptions 3, 4, and 7** hold, the gains are selected large enough based on (46) - (49) in **Supplementary Material**, and the weights  $\hat{\theta}$ ,  $\hat{W}_c$ ,  $\Gamma$ , and  $\hat{W}_a$  are updated according to (12), (29), (30), and (31), respectively, then the estimation errors  $\tilde{W}_c$ ,  $\tilde{W}_a$ , and  $\tilde{\theta}$  and the trajectories of the transformed system in (7) under the controller in (32) are locally uniformly ultimately bounded.

Proof. See Theorem 8 in **Supplementary Material**.

Using Lemma 1, it can then be concluded that the feedback control law

$$u(t) = \begin{cases} \psi(b_{(a,A)}(x(t)), t), & 0 < t < T, \\ \hat{u}(b_{(a,A)}(x(t)), \hat{W}_a(t)), & t \geq T, \end{cases} \quad (34)$$

applied to the original system in (1), achieves the control objective stated in **Section 2.1**.

## 7 SIMULATION

To demonstrate the performance of the developed method, two simulation results, one for a two-state dynamical system, and one for a four-state dynamical system corresponding to a two-link planar robot manipulator, are provided.<sup>2</sup>

### 7.1 Two State Dynamical System

The dynamical system is given by

$$\dot{x} = f(x)\theta + g(x)u \quad (35)$$

where

$$f(x) = \begin{bmatrix} x_2 & 0 & 0 & 0 \\ 0 & x_1 & x_2 & x_2(\cos(2x_1) + 2)^2 \end{bmatrix}, \quad (36)$$

$\theta = [\theta_1; \theta_2; \theta_3; \theta_4]$ , and  $g(x) = [0; \cos(2x_1) + 2]$ . The BT version of the system can be expressed in the form (7) with  $G(s) =$

$$[0; G_{21}] \text{ and } y(s) = \begin{bmatrix} F_{11} & 0 & 0 & 0 \\ 0 & F_{22} & F_{23} & F_{24} \end{bmatrix}, \text{ where}$$

$$F_{11} = B_1(s_1)x_2, \quad F_{22} = B_2(s_2)x_1, F_{23} = B_2(s_2)x_2, \\ F_{24} = B_2(s_2)x_2(\cos(2x_1) + 2)^2, \text{ and } G_{21} = B_2(s_2)\cos(2x_1) + 2.$$

The state  $x = [x_1 \ x_2]^T$  needs to satisfy the constraints  $x_1 \in (-7, 5)$  and  $x_2 \in (-5, 7)$ . The objective for the controller is to minimize the infinite horizon cost function in (17), with  $Q = \text{diag}(10, 10)$  and  $R = 0.1$ . The basis functions for value function approximation are selected as  $\sigma(s) = [s_1^2; s_1s_2; s_2^2]$ . The initial conditions for the system and the initial guesses for the weights and parameters are selected as  $x(0) = [-6.5; 6.5]$ ,  $\hat{\theta}(0) = [0; 0; 0; 0]$ ,

$\Gamma(0) = \text{diag}(1, 1, 1)$ , and  $\hat{W}_a(0) = \hat{W}_c(0) = [1/2; 1/2; 1/2]$ . The ideal values of the unknown parameters in the system model are  $\theta_1 = 1, \theta_2 = -1, \theta_3 = -0.5, \theta_4 = 0.5$ , and the ideal values of the actor and the critic weights are unknown. The simulation uses 100 fixed Bellman error extrapolation points in a  $4 \times 4$  square around the origin of the  $s$  - coordinate system.

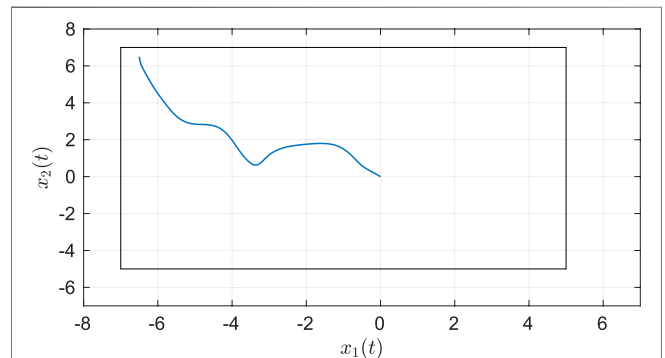
#### 7.1.1 Results for the Two State System

As seen from **Figure 2**, the system state stays within the user-specified safe set while converging to the origin. The results in **Figure 3** indicate that the unknown weights for both the actor and critic NNs converge to similar values. As demonstrated in **Figure 4** the parameter estimation errors also converge to a small neighborhood of the origin.

Since the ideal actor and critic weights are unknown, the estimates cannot be directly compared against the ideal weights. To gauge the quality of the estimates, the trajectory generated by the controller  $u(t) = \hat{u}(s(t), \hat{W}_c^*)$ , where  $\hat{W}_c^*$  is the final value of the critic weights obtained in **Figure 3**, starting from a specific initial condition, is compared against the trajectory obtained using an *offline* numerical solution computed using the GPOPS II optimization software [see, e.g., Patterson and Rao (2014)]. The total cost, generated by numerically integrating (17), is used as the metric for comparison. The costs are computed over a finite horizon, selected to be roughly 5 times the time constant of the optimal trajectories. The results in **Table 1** indicate that while the two solution techniques generate slightly different trajectories in the phase space (see **Figure 5**) the total cost of the trajectories is similar.

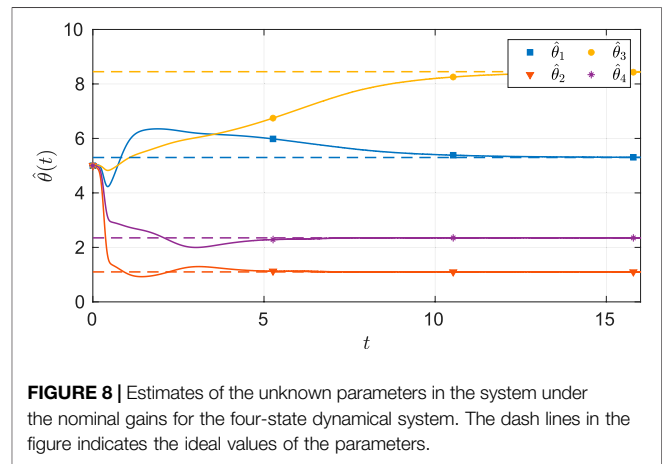
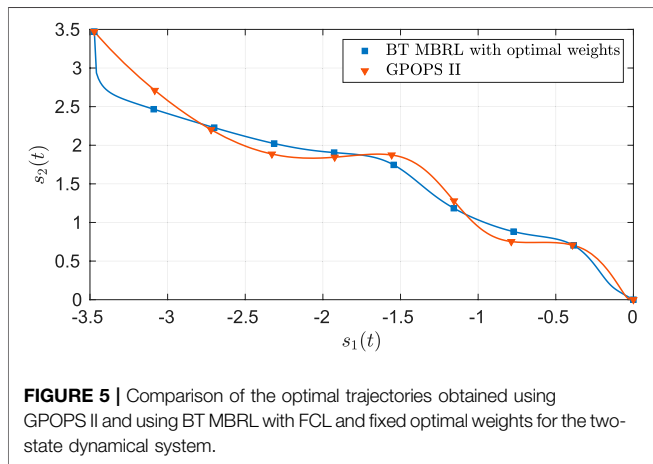
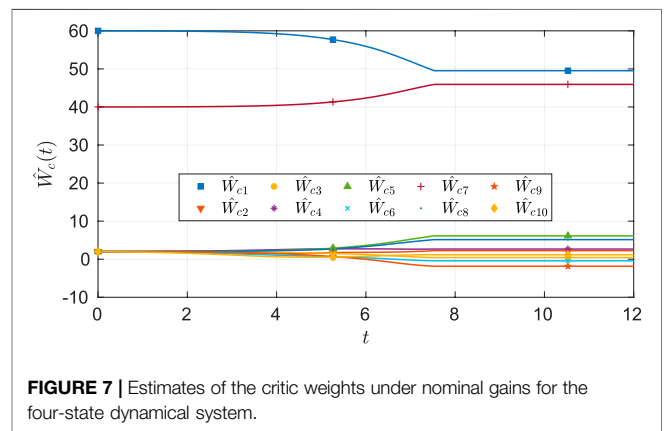
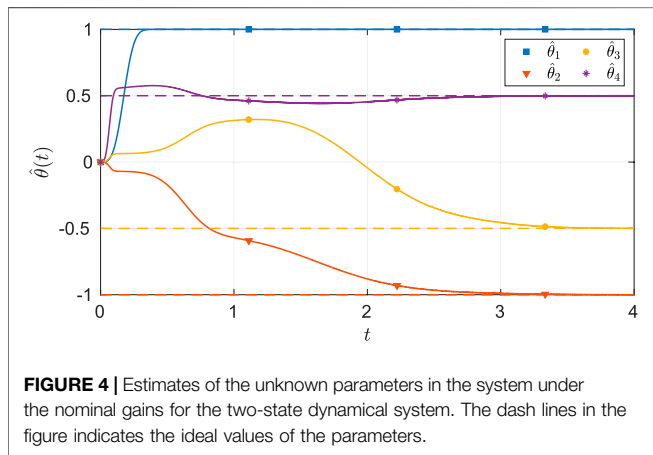
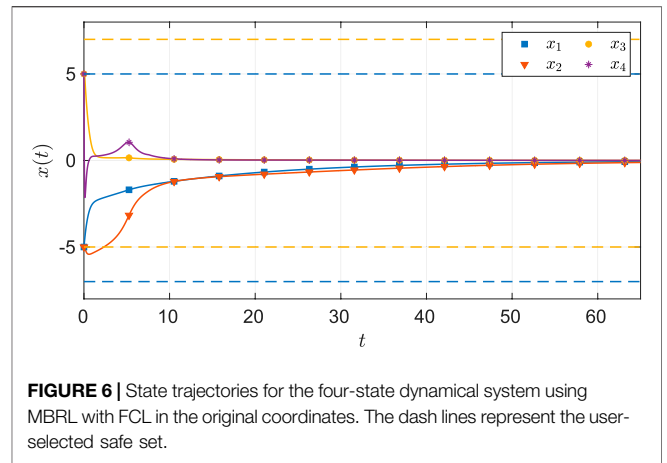
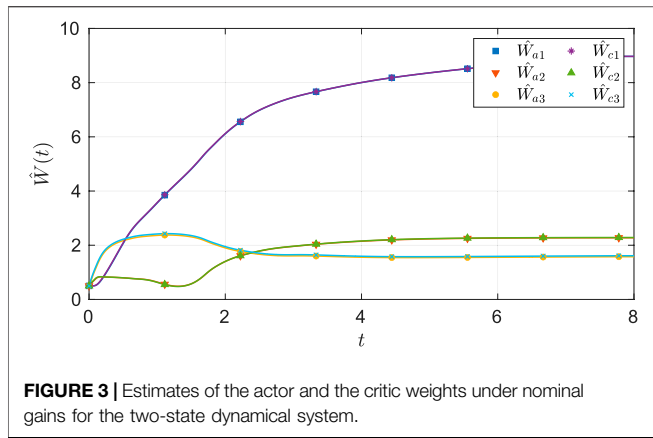
#### 7.1.2 Sensitivity Analysis for the Two State System

To study the sensitivity of the developed technique to changes in various tuning parameters, a one-at-a-time sensitivity analysis is performed. The parameters  $k_{c1}, k_{c2}, k_{a1}, k_{a2}, \beta$ , and  $\nu$  are selected for the sensitivity analysis. The costs of the trajectories, under the optimal feedback controller obtained using the developed method, are presented in **Table 2** for five different values of each parameter. The parameters are varied in a neighborhood of the nominal values (selected through trial and error)  $k_{c1} = 0.3, k_{c2}$



**FIGURE 2 |** Phase portrait for the two-state dynamical system using MBRL with FCL in the original coordinates. The boxed area represents the user-selected safe set.

<sup>2</sup>The source code for these simulations is available at <https://github.com/scc-lab/publications-code/tree/master/2021-Frontiers-BT-MBRL>.

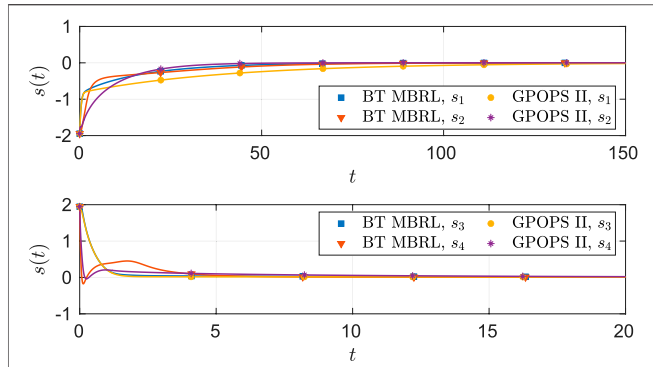


= 5,  $k_{a1} = 180$ ,  $k_{a2} = 0.0001$ ,  $\beta = 0.03$ , and  $\nu = 0.5$ . The value of  $\beta_1$  is set to be  $\text{diag}(50, 50, 50, 50)$ . The results in **Table 2** indicate that the developed method is robust to small changes in the learning gains.

### 7.2 Four State Dynamical System

The four-state dynamical system corresponding to a two-link planar robot manipulator is given by





**FIGURE 9 |** Comparison of the optimal angular position (**top**) and angular velocity (**bottom**) trajectories obtained using GPOPS II and BT MBRL with fixed optimal weights for the four-state dynamical system.

**TABLE 1 |** Comparison of costs for a single barrier transformed trajectory of (35), obtained using the optimal feedback controller generated via the developed method, and obtained using pseudospectral numerical optimal control software.

Method	Cost
BT MBRL with FCL	71.8422
GPOPS II (Patterson and Rao (2014))	72.9005

**TABLE 2 |** Sensitivity Analysis for the two state system.

$k_{c_1} =$	0.01	0.05	0.1	0.2	0.3
Cost	72.7174	72.6919	72.5378	72.3019	72.1559
$k_{c_2} =$	2	3	5	10	15
Cost	71.7476	72.3198	72.1559	71.8344	71.7293
$k_{a_1} =$	175	180	250	500	1,000
Cost	72.1568	72.1559	72.1384	72.1085	72.0901
$k_{a_2} =$	0.0001	0.0009	0.001	0.005	0.01
Cost	72.1559	72.1559	72.1559	72.1559	72.1559
$\beta =$	0.001	0.005	0.01	0.03	0.04
Cost	72.2141	72.1559	72.1958	72.1559	72.1352
$v =$	0.5	1	10	50	100
Cost	72.1559	72.4054	72.6582	79.1540	81.32

$$\dot{x} = f_1(x) + f_2(x)\theta + g(x)u \quad (37)$$

where

$$f_1(x) = \begin{bmatrix} x_3 \\ x_4 \\ -M^{-1}V_m \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \end{bmatrix}, f_2(x) = \begin{bmatrix} 0, 0, 0, 0 \\ 0, 0, 0, 0 \\ -[M^{-1}, M^{-1}]D \end{bmatrix}, \quad (38)$$

$$g(x) = \begin{bmatrix} 0, 0 \\ 0, 0 \\ (M^{-1})^T \end{bmatrix}, \quad D := \text{diag}[x_3, x_4, \tanh(x_3), \tanh(x_4)], \quad (39)$$

**TABLE 3 |** Costs for a single barrier transformed trajectory of (37), obtained using the developed method, and using pseudospectral numerical optimal control software.

Method	Cost
BT MBRL with FCL	95.1490
GPOPS II	57.8740

**TABLE 4 |** Sensitivity Analysis for the four state system.

$k_{c_1} =$	0.01	0.05	0.1	0.5	1
Cost	95.91	95.4185	95.1490	94.1607	93.5487
$k_{c_2} =$	1	5	10	20	30
Cost	304.4	101.0786	95.1490	92.7148	93.729
$k_{a_1} =$	5	10	20	30	50
Cost	94.9464	95.1224	95.1490	95.1736	95.1974
$k_{a_2} =$	0.05	0.1	0.2	0.5	1
Cost	95.2750	95.2480	95.1490	94.9580	94.6756
$\beta =$	0.1	0.5	0.8	0.9	0.95
Cost	125.33	109.7721	95.1490	92.91	93.7231
$v =$	50	70	100	125	150
Cost	92.2836	93.34	95.1490	96.1926	97.9870

$$M := \begin{bmatrix} p_1 + 2p_3c_2 & p_2 + p_3c_2 \\ p_2 + p_3c_2 & p_2 \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \text{ and} \quad (40)$$

$$V_M := \begin{bmatrix} -p_3s_2x_4 & -p_3s_2(x_3 + x_4) \\ p_3s_2x_3 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 2},$$

with  $s_2 = \sin(x_2)$ ,  $c_2 = \cos(x_2)$ ,  $p_1 = 3.473$ ,  $p_2 = 0.196$ , and  $p_3 = 0.242$ . The state  $x = [x_1 \ x_2 \ x_3 \ x_4]^T$ , that corresponds to angular positions and the angular velocities of the two links, needs to satisfy the constraints  $x_1 \in (-7, 5)$ ,  $x_2 \in (-7, 5)$ ,  $x_3 \in (-5, 7)$ , and  $x_4 \in (-5, 7)$ . The objective for the controller is to minimize the infinite horizon cost function in (17), with  $Q = \text{diag}(1, 1, 1, 1)$  and  $R = \text{diag}(1, 1)$  while identifying the unknown parameters  $\theta \in \mathbb{R}^4$  that correspond to static and dynamic friction coefficients in the two links. The ideal values of the unknown parameters are  $\theta_1 = 5.3$ ,  $\theta_2 = 1.1$ ,  $\theta_3 = 8.45$ , and  $\theta_4 = 2.35$ . The basis functions for value function approximation are selected as  $\sigma(s) = [s_1s_3; s_2s_4; s_3s_2; s_4s_1; s_1s_2; s_4s_3; s_1^2; s_2^2; s_3^2; s_4^2]$ . The initial conditions for the system and the initial guesses for the weights and parameters are selected as  $x(0) = [-5; -5; 5; 5]$ ,  $\hat{\theta}(0) = [5; 5; 5; 5]$ ,  $\Gamma(0) = \text{diag}(10, 10, 10, 10, 10, 10, 10, 10, 10, 10)$ , and  $\hat{W}_a(0) = \hat{W}_c(0) = [60; 2; 2; 2; 2; 2; 40; 2; 2; 2]$ . The ideal values of the actor and the critic weights are unknown. The simulation uses 100 fixed Bellman error extrapolation points in a  $4 \times 4$  square around the origin of the  $s$  - coordinate system.

### 7.2.1 Results for the Four State System

As seen from Figure 6, the system state stays within the user specified safe set while converging to the origin. As demonstrated in Figures 7 and 8, the actor and the critic weights converge, and estimates of the unknown parameters in the system model converge to their true values. Since the true actor and critic weights are unknown, the learned optimal controller is compared with an offline numerical optimal controller. The results of the comparison (see

**Table 3**) indicate that the two solution techniques generate slightly different trajectories in the state space (see **Figure 9**) and that the total cost resulting from the learned controller is higher. We hypothesize that the difference in costs can be attributed to the exact basis for value function approximation being unknown.

In summary, the newly developed method can achieve online optimal control through a BT MBRL approach while estimating the value of the unknown parameters in the system dynamics and ensuring safety guarantees in the original coordinates during the learning phase.

The following section details a one-at-a-time sensitivity analysis and study the sensitivity of the developed technique to changes in various tuning parameters.

### 7.2.2 Sensitivity Analysis for the Four State System

The parameters  $k_{c1}$ ,  $k_{c2}$ ,  $k_{a1}$ ,  $k_{a2}$ ,  $\beta$ , and  $\nu$  are selected for the sensitivity analysis. The costs of the trajectories under the optimal feedback controller, obtained using the developed method, are presented in **Table 4** for five different values of each parameter.

The parameters are varied in a neighborhood of the nominal values (selected through trial and error)  $k_{c1} = 0.1$ ,  $k_{c2} = 10$ ,  $k_{a1} = 20$ ,  $k_{a2} = 0.2$ ,  $\beta = 0.8$ , and  $\nu = 100$ . The value of  $\beta_1$  is set to be  $\text{diag}(100, 100, 100, 100)$ .

The results in **Tables 2** and **4**, indicate that the developed method is not sensitive to small changes in the learning gains. While reduced sensitivity to gains simplifies gain selection, as indicated by the local stability result, the developed method is sensitive to selection of basis functions and initial guesses of the unknown weights. Due to high dimensionality of the vector of unknown weights, a complete characterization of the region of attraction is computationally difficult. As such, the basis functions and the initial guess were selected *via* trial and error.

## 8 CONCLUSION

This paper develops a novel online safe control synthesis technique which relies on a nonlinear coordinate transformation that transforms a constrained optimal control problem into an unconstrained optimal control problem. A model of the system in the transformed coordinates is simultaneously learned and utilized to simulate experience. Simulated experience is used to realize convergent RL under relaxed excitation requirements. Safety of the closed-loop system, expressed in terms of box constraint, regulation of the system states to a neighborhood of the origin, and convergence of the estimated policy to a neighborhood of the optimal policy in transformed coordinates is established using a Lyapunov-based stability analysis.

While the main result of the paper states that the state is uniformly ultimately bounded, the simulation results hint towards asymptotic convergence of the part of the state that corresponds to the system trajectories,  $x(\cdot)$ . Proving such a result is a part of future research.

Limitations and possible extensions of the ideas presented in this paper revolve around two key issues: (a) safety, and (b) online learning and optimization. The barrier function used in the BT to

address safety can only ensure a fixed box constraint. A more generic and adaptive barrier function, constructed, perhaps, using sensor data is a subject for future research.

For optimal learning, parametric approximation techniques are used to approximate the value functions in this paper. Parametric approximation of the value function requires selection of appropriate basis functions which may be hard to find for the barrier-transformed dynamics. Developing techniques to systematically determine a set of basis functions for real-world systems is a subject for future research.

The barrier transformation method to ensure safety relies on knowledge of the dynamics of the system. While this paper addresses parametric uncertainties, the BE method could potentially result in a safety violation due to unmodeled dynamics. In particular, the safety guarantees developed in this paper rely on the relationship (**Lemma 1**) between trajectories of the original dynamics and the transformed system, which holds in the presence of parametric uncertainty, but fails if a part of the dynamics is not included in the original model. Further research is needed to establish safety guarantees that are robust to unmodeled dynamics (for a differential games approach to robust safety, see Yang et al. (2020)).

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

## AUTHOR CONTRIBUTIONS

NM: Conceptualization, Methodology, Validation, Resources, Investigation, Numerical experiment, Formal analysis, Writing—Original Draft; SN: Writing—Reviewing; ZB: Writing—Reviewing; RK: Conceptualization, Formal analysis—Reviewing and Editing; Writing—Reviewing and Editing, Supervision.

## FUNDING

This research was supported, in part, by the Air Force Research Laboratories under award number FA8651-19-2-0009. Any opinions, findings, or recommendations in this article are those of the author(s), and do not necessarily reflect the views of the sponsoring agencies.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2021.733104/full#supplementary-material>

## REFERENCES

- Adetola, V., and Guay, M. (2008). Finite-time Parameter Estimation in Adaptive Control of Nonlinear Systems. *IEEE Trans. Automat. Contr.* 53, 807–811. doi:10.1109/tac.2008.919568
- Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. (2017). Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Automat. Contr.* 62, 3861–3876. doi:10.1109/TAC.2016.2638961
- Bechlioulis, C. P., and Rovithakis, G. A. (2009). Adaptive Control with Guaranteed Transient and Steady State Tracking Error Bounds for Strict Feedback Systems. *Automatica* 45, 532–538. doi:10.1016/j.automatica.2008.08.012
- Bertsekas, D. P. (2011). *Dynamic Programming and Optimal Control*. 3rd edition, II. Belmont, MA: Athena Scientific.
- Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K. G., Lewis, F. L., and Dixon, W. E. (2012). “An Actor-Critic-Identifier Architecture for Adaptive Approximate Optimal Control,” in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Editors F. L. Lewis and D. Liu (Piscataway, NJ, USA: Wiley and IEEE Press), 258–278. IEEE Press Series on Computational Intelligence. doi:10.1002/9781118453988.ch12
- Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K. G., Lewis, F. L., and Dixon, W. E. (2013). A Novel Actor-Critic-Identifier Architecture for Approximate Optimal Control of Uncertain Nonlinear Systems. *Automatica* 49, 89–92. doi:10.1016/j.automatica.2012.09.019
- Choi, J., Castañeda, F., Tomlin, C., and Sreenath, K. (2020). “Reinforcement Learning for Safety-Critical Control under Model Uncertainty, Using Control Lyapunov Functions and Control Barrier Functions,” in *Proceedings of Robotics: Science and Systems*, Corvallis, Oregon, USA, July 12–16, 2020. doi:10.15607/RSS.2020.XVI.088
- Cohen, M. H., and Belta, C. (2020). “Approximate Optimal Control for Safety-Critical Systems with Control Barrier Functions,” in *Proceedings of the IEEE Conference on Decision and Control*, Jeju, Korea (South), 14–18 Dec. 2020, 2062–2067. doi:10.1109/CDC42340.2020.9303896
- Deptula, P., Chen, H.-Y., Licitra, R. A., Rosenfeld, J. A., and Dixon, W. E. (2020). Approximate Optimal Motion Planning to Avoid Unknown Moving Avoidance Regions. *IEEE Trans. Robot.* 36, 414–430. doi:10.1109/TRO.2019.2955321
- Doya, K. (2000). Reinforcement Learning in Continuous Time and Space. *Neural Comput.* 12, 219–245. doi:10.1162/089976600300015961
- Graichen, K., and Petit, N. (2009). Incorporating a Class of Constraints into the Dynamics of Optimal Control Problems. *Optim. Control. Appl. Meth.* 30, 537–561. doi:10.1002/oca.880
- Greene, M. L., Deptula, P., Nivison, S., and Dixon, W. E. (2020). Sparse Learning-Based Approximate Dynamic Programming with Barrier Constraints. *IEEE Control. Syst. Lett.* 4, 743–748. doi:10.1109/LCSYS.2020.2977927
- Greene, M. L., Deptula, P., Kamalapurkar, R., and Dixon, W. E. (2021). “Mixed Density Methods for Approximate Dynamic Programming,” in *Handbook of Reinforcement Learning and Control. Studies in Systems, Decision and Control*. Editors K. G. Vamvoudakis, Y. Wan, F. Lewis, and D. Cansever (Cham, Switzerland: Springer International Publishing), 325, 139–172. chap. 5. doi:10.1007/978-3-030-60990-0\_5
- He, W., Li, Z., and Chen, C. L. P. (2017). A Survey of Human-Centered Intelligent Robots: Issues and Challenges. *Ieee/caa J. Autom. Sin.* 4, 602–609. doi:10.1109/jas.2017.7510604
- Kamalapurkar, R., Rosenfeld, J. A., and Dixon, W. E. (2016a). Efficient Model-Based Reinforcement Learning for Approximate Online Optimal Control. *Automatica* 74, 247–258. doi:10.1016/j.automatica.2016.08.004
- Kamalapurkar, R., Walters, P., and Dixon, W. E. (2016b). Model-based Reinforcement Learning for Approximate Optimal Regulation. *Automatica* 64, 94–104. doi:10.1016/j.automatica.2015.10.039
- Kamalapurkar, R., Walters, P., Rosenfeld, J., and Dixon, W. (2018). “Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach,” in *Communications and Control Engineering*. Cham, Switzerland: Springer International Publishing. doi:10.1007/978-3-319-78384-0
- Khalil, H. K. (2002). *Nonlinear Systems*. third edn. Upper Saddle River, NJ: Prentice-Hall.
- Kiumarsi, B., Lewis, F. L., Modares, H., Karimpour, A., and Naghibi-Sistani, M.-B. (2014). Reinforcement -Learning for Optimal Tracking Control of Linear Discrete-Time Systems with Unknown Dynamics. *Automatica* 50, 1167–1175. doi:10.1016/j.automatica.2014.02.015
- Lewis, F. L., and Vrabie, D. (2009). Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control. *IEEE Circuits Syst. Mag.* 9, 32–50. doi:10.1109/mcas.2009.933854
- Liu, D., and Wei, Q. (2014). Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 621–634. doi:10.1109/TNNLS.2013.2281663
- Marvi, Z., and Kiumarsi, B. (2021). Safe Reinforcement Learning: A Control Barrier Function Optimization Approach. *Int. J. Robust Nonlinear Control*. 31, 1923–1940. doi:10.1002/rnc.5132
- Modares, H., Lewis, F. L., and Naghibi-Sistani, M.-B. (2013). Adaptive Optimal Control of Unknown Constrained-Input Systems Using Policy Iteration and Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 24, 1513–1525. doi:10.1109/tnnls.2013.2276571
- Modares, H., Lewis, F. L., and Naghibi-Sistani, M.-B. (2014). Integral Reinforcement Learning and Experience Replay for Adaptive Optimal Control of Partially-Unknown Constrained-Input Continuous-Time Systems. *Automatica* 50, 193–202. doi:10.1016/j.automatica.2013.09.043
- Patterson, M. A., and Rao, A. V. (2014). GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. *ACM Trans. Math. Softw.* 41, 1–37. doi:10.1145/2558904
- Qin, C., Zhang, H., and Luo, Y. (2014). Online Optimal Tracking Control of Continuous-Time Linear Systems with Unknown Dynamics by Using Adaptive Dynamic Programming. *Int. J. Control*. 87, 1000–1009. doi:10.1080/00207179.2013.863432
- Roy, S. B., Bhasin, S., and Kar, I. N. (2016). “Parameter Convergence via a Novel PI-like Composite Adaptive Controller for Uncertain Euler-Lagrange Systems,” in *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 12–14 Dec. 2016, 1261–1266. doi:10.1109/cdc.2016.7798439
- Sanfelice, R. G. (2021). *Hybrid Feedback Control*. Princeton, NJ, USA: Princeton University Press.
- Sauvigny, F. (2012). *Partial Differential Equations 1*. London: Springer-Verlag.
- Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press.
- Vamvoudakis, K., Vrabie, D., and Lewis, F. L. (2009). “Online Policy Iteration Based Algorithms to Solve the Continuous-Time Infinite Horizon Optimal Control Problem,” in *IEEE Symp. Adapt. Dyn. Program. Reinf. Learn.*, Nashville, TN, USA, 30 March–2 April 2009, 36–41. doi:10.1109/adprl.2009.4927523
- Vamvoudakis, K. G., Miranda, M. F., and Hespanha, J. P. (2016). Asymptotically Stable Adaptive-Optimal Control Algorithm with Saturating Actuators and Relaxed Persistence of Excitation. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 2386–2398. doi:10.1109/tnnls.2015.2487972
- Vrabie, D., and Lewis, F. (2010). “Online Adaptive Optimal Control Based on Reinforcement Learning,” in *Optimization and Optimal Control: Theory and Applications*. Editors A. Chinchuluun, P. M. Pardalos, R. Enkhbat, and I. Tseveendorj (New York, NY: Springer New York), 309–323. doi:10.1007/978-0-387-89496-6\_16
- Walters, P., Kamalapurkar, R., and Dixon, W. E. (2015). “Approximate Optimal Online Continuous-Time Path-Planner with Static Obstacle Avoidance,” in *Proceedings of the IEEE Conference on Decision and Control*, Osaka, Japan, 15–18 Dec. 2015 (Osaka, Japan, 650–655. doi:10.1109/cdc.2015.7402303

- Yang, Y., Yin, Y., He, W., Vamvoudakis, K. G., Modares, H., and Wunsch, D. C. (2019). "Safety-aware Reinforcement Learning Framework with an Actor-Critic-Barrier Structure," in Proceedings of the American Control Conference, Philadelphia, PA, USA, 10–12 July 2019, 2352–2358. doi:10.23919/acc.2019.8815335
- Yang, Y., Ding, D.-W., Xiong, H., Yin, Y., and Wunsch, D. C. (2020). Online Barrier-Actor-Critic Learning for H<sub>∞</sub> Control with Full-State Constraints and Input Saturation. *J. Franklin Inst.* 357, 3316–3344. doi:10.1016/j.jfranklin.2019.12.017

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

*Copyright © 2021 Mahmud, Nivison, Bell and Kamalapurkar. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.*

## NOMENCLATURE

**RL** Reinforcement Learning

**MBRL** Model-based Reinforcement Learning

**ADP** Approximate Dynamic Programming

**NN** Neural Network

**BF** Barrier Function

**BT** Barrier Transformation

**CL** Concurrent Learning

**FCL** Filtered Concurrent Learning

**PE** Persistence of Excitation

**FE** Finite Excitation

**TS** Transformed State

**BE** Bellman Error

**HJB** Hamilton-Jacobi-Bellman

**VI** Value Iteration

**PI** Policy Iteration.