Sparse Learning-Based Approximate Dynamic Programming With Barrier Constraints

Max L. Greene, Patryk Deptula, Scott Nivison, and Warren E. Dixon

***Abstract*—This letter provides an approximate online adaptive solution to the infinite-horizon optimal control problem for control-affine continuous-time nonlinear systems while formalizing system safety using barrier certificates. The use of a barrier function transform provides safety certificates to formalize system behavior. Specifically, using a barrier function, the system is transformed to aid in developing a controller which maintains the system in a pre-defined constrained region. To aid in online learning of the value function, the state-space is segmented into a number of user-defined segments. Off-policy trajectories are selected in each segment, and sparse Bellman error extrapolation is performed within each respective segment to generate an optimal policy within each segment. A Lyapunov-like stability analysis is included which proves uniformly ultimately bounded regulation in the presence of the barrier function transform and discontinuities. Simulation results are provided for a two-state dynamical system to compare the performance of the developed method to existing methods.**

***Index Terms*—Data-based control, dynamic programming, nonlinear control, optimal control, reinforcement learning.**

I. Introduction

WHEN formulating optimal control problems, the Hamilton-Jacobi-Bellman Equation (HJB) provides an optimality condition. The designed optimal control policy depends on the value function [1], [2]. However, it is generally difficult to solve the HJB due to system uncertainties and nonlinearities. Approximate dynamic programming (ADP) [3]–[8] is a strategy to learn the value function. By obtaining an approximate value function, a stabilizing and approximate control policy can be developed.

A challenge for ADP methods, unlike traditional adaptive methods, is the need to simultaneously identify uncertain parameters. Traditional adaptive control methods require a persistence of excitation (PE) condition to exactly learn the approximate optimal policy [9]–[12]. The difficulty associated with the PE condition, which, generally, cannot be verified for nonlinear systems, motivates ad hoc methods, which can potentially affect performance or destabilize the system. To relax the PE condition, concurrent learning-based system identifiers that use recorded data for learning the value function are used [13]–[15].

In ADP, the Bellman Error (BE) is used as a performance metric, which is an indirect measure of the estimation of the value function along the system trajectory. Previous work (see, [5], [16], [17]) showed that if the system dynamics are known, then the dynamic model can be used to evaluate the BE at any number of arbitrary points in a system's state-space. This process is called BE extrapolation. Works such as [13], [16] explore the case of uncertain, linear parameterizable dynamics in which the system identification and value function approximation are simultaneously performed using BE extrapolation. Results such as [5], [18] use NNs that provide sufficient value function approximation in a neighborhood of the current state. BE extrapolation is typically performed by selecting off-trajectory points around the neighborhood of the current state. Since a global value function approximation is desired, BE extrapolation is sometimes performed over the entire state-space, which is computationally expensive. Increasing the number of basis functions may better estimate the value function, however, this is not computationally efficient.

Sparse neural networks (SNNs), like conventional NNs, are a tool to facilitate learning in uncertain systems (see, [19]–[23]). SNNs have been used to reduce computational complexity in NNs by decreasing the number of active neurons; hence, reducing the number of computations overall. Sparse adaptive controllers have been developed to update a small number of neurons at certain points in the state-space in works such as [21]. Sparsification encourages local learning through intelligent segmentation [20]. A SNNs architecture can facilitate learning without relying on a high adaptive learning rate [23]. In practice, high learning rates can cause oscillations or instability due to unmodeled dynamics

Manuscript received November 18, 2019; revised January 21, 2020; accepted February 16, 2020. Date of publication March 3, 2020; date of current version May 25, 2020. This work was supported in part by the Office of Naval Research under Grant N00014-13-1-0151, in part by the Naval Engineering Education Consortium under Award N00174-18-1-0003, in part by the Air Force Office of Scientific Research (AFOSR) under Award FA9550-18-1-0109 and Award FA9550-19-1-0169, and in part by the National Science Foundation under Award 1762829. Recommended by Senior Editor M. Guay. *(Corresponding author: Max L. Greene.)*

Max L. Greene and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: maxgreene12@ufl.edu; wdixon@ufl.edu).

Patryk Deptula is with the Perception and Autonomy Group Charles Stark Draper Laboratory, Inc., Cambridge, MA 02139 USA (e-mail: pdeptula@draper.com).

Scott Nivison is with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542 USA (e-mail: scott.nivison@us.af.mil).

Digital Object Identifier 10.1109/LCSYS.2020.2977927

in the control bandwidth [23]. SNNs create a framework for switching and segmentation as well as computational benefits due to the small number of active neurons. Sparsification techniques enable local approximation across the segments, which characterizes regions with significantly varying dynamics or unknown uncertainties.

Barrier functions (BFs) have been used to generate safety certificates for control systems [24], [25]. Ensuring safety of dynamical systems is desirable for the implementation of control systems. BFs have a natural relationship with Lyapunov-like functions, set invariance, and multi-objective control. A balance exists between satisfying both the control objectives of a system and the safety constraints of the environment [26]. BFs have been previously used with ADP [27], but not always in the context of safety certificates. The results in [28] and [29] provide a united framework for solving the optimal control problem online while still providing formal guarantees of performance and correctness.

This letter leverages the BF transformation developed in the model-free ADP controller in [29] for the development of a model-based ADP controller. The model-free result in [29] evaluates the BE only along the system trajectory, resulting in the typical exploration versus exploitation tradeoff. A contribution of this letter is the development of a framework for sparse BE extrapolation (motivated by [5], [16], [17]) for off-trajectory learning of the value function, while also adhering to state-space constraints (unlike [5], [16], [17]). Specifically, this letter provides a first investigation of BE extrapolation using a state-constraint BF transform. The unique combination of BE extrapolation with the use of a BF raises new questions such as which states should be transformed (e.g., there is a computational penalty for each state transformation), should the BF transformation be applied before or after the BE extrapolation (i.e., in what space should the extrapolation be performed), and what are the implications of history stack updates in the transformed state-space. The subsequent design and Lyapunov-based stability analysis provides the first exploration of such questions in a manner that yields uniformly ultimately bounded (UUB) convergence of the transformed states. Simulation results are presented for a two-state dynamical system to compare the developed method to existing methods. Specifically, the developed model-based RL approach with BFs, segmentation, and sparse BE extrapolation can be applied to systems to achieve online approximate optimal control with additional safety guarantees.

II. Barrier Functions

Consider the continuous-time control-affine nonlinear dynamical system

$$\dot{x} = f(x(t)) + g(x(t))u(t) \tag{1}$$

with initial condition $x(0) = x_0 \in \mathbb{R}^n$, where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes the system state, $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ denotes the control input, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is the control effectiveness. The goal is to design a control policy u for the system in (1) while regulating the system state, $x = [x_1, \ldots, x_n]^T$, to the origin while also ensuring the states lie within distinct user-specified sets (i.e., within the user-defined barriers) such that

$$x_i \in (a_i, A_i)\, \forall\, i = 1, \ldots, N, \tag{2}$$

where $a \in \mathbb{R}^n$ and $A \in \mathbb{R}^n$ represent the vectors of all lower and upper bounds of the sets, respectively, with $a_i \in \mathbb{R}$ and $A_i \in \mathbb{R}$ being the i^{th} row of a and A, respectively, where $i \in \mathbb{Z} \bigcap [1, N]$. Let a logarithmic BF, $b : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, be defined as

$$b(z_i, a_i, A_i) \triangleq \ln\left(\frac{A_i}{a_i}\frac{a_i - z_i}{A_i - z_i}\right), \forall z_i \in (a_i, A_i), \tag{3}$$

such that the constants a_i and A_i satisfy $a_i < 0 < A_i$, $z_i \in \mathbb{R}$, and the inverse of the BF in (3), is

$$b^{-1}(z_i, a_i, A_i) = a_i A_i \frac{e^{z_i} - 1}{e^{z_i} a_i - A_i}. \tag{4}$$

The logarithmic BF in (4) is as a tool to ensure the system remains within the user-defined barriers. To this end, the derivative of (4) is taken with respect to z_i to yield

$$\frac{db^{-1}(z_i, a_i, A_i)}{dz_i} = \frac{a_i^2 A_i - a_i A_i^2}{a_i^2 e^{z_i} - 2A_i a_i + A_i^2 e^{-z_i}}. \tag{5}$$

Let $s_i \in \mathbb{R}$ be the state-space to barrier-space coordinate transformed state such that

$$s_i = b\big(x_i, a_i, A_{,i}\big). \tag{6}$$

Using (4), the transformation from the barrier-space to the state-space is

$$x_i = b^{-1}\big(s_i, a_i, A_{,i}\big), \tag{7}$$

Taking the time-derivative of (7) and rearranging yields

$$\dot{s}_i = \frac{\left(a_i^2 e^{s_i} - 2A_i a_i + A_i^2 e^{-s_i}\right)}{a_i^2 A_i - a_i A_i^2}\dot{x}_i. \tag{8}$$

Using (1) in (8) results in the transformed state

$$\dot{s}_i = F_i(s_i, a_i, A_i) + G_i(s_i, a_i, A_i)u(t), \tag{9}$$

where

$$F_i(s_i, a_i, A_i) \triangleq \left(\frac{a_i^2 e^{s_i} - 2A_i a_i + A_i^2 e^{-s_i}}{a_n^2 A_n - a_n A_n^2}\right) \times \left(f_i\Big(b^{-1}\big(s_i, a_i, A_{,i}\big)\Big)\right),$$

$$G_i(s_i, a_i, A_i) \triangleq \left(\frac{a_i^2 e^{s_i} - 2A_i a_i + A_i^2 e^{-s_i}}{a_n^2 A_n - a_n A_n^2}\right) \times g_i\Big(b^{-1}\big(s_i, a_i, A_{,i}\big)\Big),$$

and $b^{-1}(s_i, a_i, A_{,i})$ with $f_i : \mathbb{R} \to \mathbb{R}$ and $g_i : \mathbb{R} \to \mathbb{R}^{1\times m}$ being the i^{th} row of the functions f and g in (1), respectively. The transformed states $s \triangleq [s_1, \ldots, s_n]^T \in \mathbb{R}^n$ can be written using (8) in a compact form as

$$\dot{s} = F(s) + G(s)u(t), \tag{10}$$

where $F(s) \triangleq [F_1(s_1, a_1, A_1) \quad \cdots \quad F_n(s_n, a_n, A_n)]^T$ and $G(s) \triangleq [G_1(s_1, a_1, A_1) \quad \cdots \quad G_n(s_n, a_n, A_n)]^T$.

The drift dynamic, F, is assumed to be a locally Lipschitz function with $F(0) = 0$, where $F' : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is continuous.[1] There exists a constant b_f, such that for $x \subset \Phi$, $\|F(s)\| \leq b_f\|s\|$, where $\Phi \subset \mathbb{R}^n$ is a compact set containing the origin. The system is assumed to be controllable over the compact set Φ, and the control effectiveness, G, is assumed to be a locally Lipschitz function and bounded such that $0 < \|G(x)\| \leq \bar{G}$, where $\bar{G} \in \mathbb{R}_{\geq 0}$.

[1]The notation $(\cdot)'$ denotes the partial derivative with respect to the first argument of the function.

III. Approximate Optimal Controller Development

The control objective is to solve the infinite-horizon optimal regulation problem, i.e., determine a control policy, u, that minimizes the infinite horizon cost function, $J : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$, defined as

$$J(s,u) \triangleq \int_{t_0}^{\infty} r(s(\tau),u(\tau))d\tau, \tag{11}$$

subject to (1) while regulating the system states to the origin (i.e., $s=0$), where $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is the instantaneous cost defined as $r(s,u) \triangleq s^T Q s + u^T R u$, $Q \in \mathbb{R}^{n\times n}$ is a constant user-defined symmetric positive definite (PD) matrix, and $R \in \mathbb{R}^{m\times m}$ is a constant positive definite symmetric matrix.

Remark 1: The state cost matrix, Q, satisfies $\underline{q} I_n \leq Q \leq \overline{q} I_n$ where $\underline{q}, \overline{q} \in \mathbb{R}_{>0}$, and I_n represents the $n \times n$ identify matrix.

The infinite horizon value function (i.e., the cost to go) for the optimal solution is denoted by $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and given by

$$V^*(s) = \min_{u(\tau)\in U,\, \tau\in\mathbb{R}_{\geq t}} \int_t^{\infty} r(s(\tau),u(\tau))d\tau, \tag{12}$$

where $U \subseteq \mathbb{R}^m$ denotes the action space. Provided and optimal control policy exists, the value function is characterized by the corresponding HJB

$$0 = \min_{u(\tau)\in U} \left(V^{*\prime}(s)(F(s)+G(s)u) + s^T Q s + u^T R u\right), \tag{13}$$

with the boundary condition $V^*(0) = 0$. Provided the HJB in (13) admits a continuously differentiable PD solution, then the optimal closed-loop control policy $u^* : \mathbb{R}^n \to \mathbb{R}^m$ is $u^*(s) = -\frac{1}{2} R^{-1} G(s)^T (V^{*\prime}(s))^T$.

A. Value Function Approximation

The HJB in (13) requires knowledge of the optimal value function, which, generally, is an unknown function for nonlinear systems. Parametric methods can be used to approximate the value function over a compact domain. To facilitate the solution of (13), let $\Omega \subset \mathbb{R}^n$ be a compact set containing the origin with $s \in \Omega$. The universal function approximation property of single-layer NNs is used to represent the value function, V^*, as

$$V^*(s) = W^T \sigma(s) + \epsilon(s), \tag{14}$$

where $W \in \mathbb{R}^L$ is an unknown bounded weight, $\sigma : \mathbb{R}^n \to \mathbb{R}^L$ is a user-defined vector of basis functions, and $\epsilon : \mathbb{R}^n \to \mathbb{R}$ is the bounded function approximation error. Solving (13) for u and using (14), the approximate optimal control policy, u^*, can be expressed in terms of the gradient of the value function, V^*, as

$$u^*(s) = -\frac{1}{2} R^{-1} G(s) \left(\sigma'(s)^T W + \epsilon'(s)^T\right). \tag{15}$$

There exists a set of constants that upper bound the unknown weight vector, W, the user-defined basis vector, σ, and approximation error, ϵ, such that $\|W\| \leq \bar{W}$, $\sup_{s\in\Omega} \|\sigma\| \leq \bar{\sigma}$, $\sup_{s\in\Omega} \|\sigma'\| \leq \bar{\sigma'}$, $\sup_{s\in\Omega} \|\epsilon\| \leq \bar{\epsilon}$, $\sup_{s\in\Omega} \|\epsilon'\| \leq \bar{\epsilon'}$, where $\bar{W}, \bar{\sigma}, \bar{\sigma'}, \bar{\epsilon}, \bar{\epsilon'} \in \mathbb{R}_{>0}$ (see, [3], [30], [31]).

Since the ideal weights are unknown, a parametric estimate, called a critic weight, $\hat{W}_c \in \mathbb{R}^L$, is substituted to estimate the optimal value function, $\hat{V} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$ where

$$\hat{V}\left(s, \hat{W}_c\right) = \hat{W}_c^T \sigma(s). \tag{16}$$

An actor weight estimate, $\hat{W}_a \in \mathbb{R}^L$, is used to provide an estimated version of (15), $\hat{u} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^m$, given by

$$\hat{u}\left(s, \hat{W}_a\right) = -\tfrac{1}{2} R^{-1} G(s)^T \left(\sigma'(s)^T \hat{W}_a\right). \tag{17}$$

B. Bellman Error

The HJB in (13) is equal to zero under optimal conditions; however, substituting (16) and (17) into (13) results in a residual term, $\delta : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$, which is referred to as the BE, defined as

$$\hat{\delta}\left(s, \hat{W}_c, \hat{W}_a\right) \triangleq \hat{V}'\left(s, \hat{W}_c\right)\left(F(s) + G(s)\hat{u}\left(s, \hat{W}_a\right)\right) + \hat{u}\left(s, \hat{W}_a\right)^T R \hat{u}\left(s, \hat{W}_a\right) + s^T Q s \tag{18}$$

where $\hat{V}'(s, \hat{W}_c) = \hat{W}_c^T \sigma'(s)$ denotes the gradient of the value function estimate. The BE is indicative of how close the actor and critic weight estimates are to the ideal weights. By defining the mismatch between the estimates and the ideal values as $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$, substituting (14) and (17) in (13), and subtracting from (18) yields

$$\hat{\delta} = \frac{1}{4} \tilde{W}_a^T G_\sigma \tilde{W}_a - \omega^T \tilde{W}_c + O(\varepsilon), \tag{19}$$

where $\omega : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^n$ is defined as

$$\omega\left(s, \hat{W}_a\right) \triangleq \sigma'(s)\left(F(s) + G(s)\hat{u}\left(s, \hat{W}_a\right)\right),$$

and $O(\varepsilon) \triangleq \frac{1}{2} W^T \sigma' G_R \varepsilon'^T + \frac{1}{4} G_\varepsilon - \varepsilon' F$.[2]

C. Sparse Bellman Error Extrapolation

At each time instant, the BE in (18) is calculated using the control policy given by (17) evaluated using the current system state, critic weight estimates, and actor weight estimates to obtain the instantaneous BE denoted by $\hat{\delta}(t) \triangleq \hat{\delta}(s(t), \hat{W}_c(t), \hat{W}_a(t))$ and control policy denoted by $u(t) \triangleq \hat{u}(s(t), \hat{W}_a(t))$. Our previous work in [22] explored using the computational efficiency of SNNs and segmentation to extrapolate the BE, so that the BE can be active across the active state-space, thereby relaxing the traditional PE condition. The benefit to performing BE extrapolation across multiple segments is that the process can be performed in parallel. Since SNNs are more efficient than traditional full-weight neurons in this application, BE extrapolation within multiple segments can be computed simultaneously.

To relax the strictness of the PE condition, virtual excitation using BE extrapolation is performed. The state-space is divided into a user-specified number of segments. Let the operating domain Ω be a partition into $S \in \mathbb{N}$ segments such that $\mathbb{S} \triangleq \{j \in \mathbb{N} | j \leq S\}$ defines the set of segments in the operating domain as $\Omega = \bigcup_{k=1}^{S} \Omega_j$.

[2]The notation G_R, G_σ, and G_ε is defined as $G_R = G_R(s) \triangleq G(s) R^{-1} G(s)^T$, $G_\sigma = G_\sigma \triangleq \sigma'(s) G_R(s) \sigma'(s)^T$, and $G_\varepsilon = G_\varepsilon(s) \triangleq \varepsilon'(s) G(s) \varepsilon'(s)^T$, respectively.

Each segment is assigned a user-specified number and location of off-trajectory points, $\{s_{i,j} : s_{i,j} \in \Omega_j\}_{i=1}^{N_j}$, where $N_j \in \mathbb{N}$ denotes the user-specified number of points in the segment Ω_j, and $x_{i,j} = b^{-1}(s_{i,j}, a_i, A_i)$. Using the extrapolated barrier-space trajectories, $s_{i,j}$ for a given $j \in S$, the tuple $(\Sigma_c^j, \Sigma_a^j, \Sigma_\Gamma^j)$ is defined as the history stack corresponding to Ω_j where $\Sigma_c^j \triangleq \frac{1}{N_j}\sum_{i=1}^{N_j} \frac{\omega_{i,j}(t)}{\rho_{i,j}(t)}\hat{\delta}_{i,j}(t)$, $\Sigma_a^j \triangleq \frac{1}{N_j}\sum_{i=1}^{N_j} \frac{G_{\sigma i,j}^T \hat{W}_a(t)\omega_{i,j}^T(t)}{4\rho_{i,j}(t)}$, $\Sigma_\Gamma^j \triangleq \frac{1}{N_j}\sum_{i=1}^{N_j} \frac{\omega_{i,j}(t)\omega_{i,j}^T(t)}{\rho_{i,j}(t)}$, $\omega_{i,j}(t) \triangleq \omega(s_{i,j}, \hat{W}_a(t)) = \sigma'(s_{i,j})(F(s_{i,j}) + G(s_{i,j})\hat{u}(s_{i,j}, \hat{W}_a(t)))$, $\rho_{i,j}(t) = 1 + \nu\omega_{i,j}^T(t)\Gamma(t)\omega_{i,j}(t)$, and $\nu \in \mathbb{R}_{>0}$ are user-defined gains. Furthermore, the notation $G_R = G_R(s) \triangleq G(s)R^{-1}G^T(s)$, $G_\sigma = G_\sigma(s) \triangleq \sigma'(s)G_R(s)\sigma'^T(s)$, and $G_\epsilon = G_\epsilon(s) \triangleq \epsilon'(s)G_R(s)\epsilon'^T(s)$.

Remark 2: BE extrapolation is performed in the barrier-space since function approximation is taken in a compact set over the barrier-space.

Assumption 1: Over each segment $j \in S$, there exists a finite set of trajectories $\{s_{i,j} : s_{i,j} \in \Omega_j\}_{i=1}^{N_j}$ such that

$$0 < \underline{c} \triangleq \inf_{t\in\mathbb{R}_{\geq 0}, j\in S} \lambda_{\min}\left\{\Sigma_\Gamma^j(t)\right\}, \tag{20}$$

for all $t \in \mathbb{R}_{\geq 0}$, where $\lambda_{\min}\{\cdot\}$ is the minimum eigenvalue.

Remark 3: The constant $\underline{c}$ is the lower bound of the value of each input-output data pair's minimum eigenvalues.

D. Update Laws for Actor and Critic Weights

Using the instantaneous BE $\hat{\delta}(t)$, policy $u(t)$, and extrapolated BEs $\hat{\delta}_{i,j}(t)$, the critic and actor weights are updated according to

$$\dot{\hat{W}}_c(t) = -\eta_{c1}\Gamma\frac{\omega(t)}{\rho(t)}\hat{\delta}(t) - \eta_{c2}\Sigma_c^j(t), \tag{21}$$

$$\dot{\Gamma}(t) = \left(\lambda\Gamma(t) - \eta_{c1}\frac{\Gamma(t)\omega(t)\omega(t)^T\Gamma(t)}{\rho^2(t)} - \Gamma(t)\eta_{c2}\left(\Sigma_\Gamma^j(t)\right)\Gamma(t)\right)\mathbf{1}_{\{\underline{\Gamma}\leq\|\Gamma\|\leq\overline{\Gamma}\}}, \tag{22}$$

$$\dot{\hat{W}}_a(t) = -\eta_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - \eta_{a2}\hat{W}_a(t) + \frac{\eta_{c1}G_\sigma^T(t)\hat{W}_a(t)\omega^T(t)}{4\rho(t)}\hat{W}_c(t) + \left(\eta_{c2}\Sigma_a^j(t)\right)\hat{W}_c(t), \tag{23}$$

where $\eta_{c1}, \eta_{c2}, \eta_{a1}, \eta_{a2}, \lambda$ are positive constant learning gains, $\overline{\Gamma}, \underline{\Gamma} \in \mathbb{R}_{>0}$ are upper and lower bound saturation constants,[3] and $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function.[4]

IV. Stability Analysis

To facilitate the analysis, the notation $\overline{(\cdot)}$ is defined as $\overline{(\cdot)} \triangleq \sup_{x\in\Omega}(\cdot)$. Let $r \triangleq [s^T, \tilde{W}_c^T, \tilde{W}_a^T]^T$ denote a concatenated state, and let $V_L : \mathbb{R}^{n+2L} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a candidate Lyapunov function defined as

$$V_L(r,t) = V^*(s) + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}(t)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a, \tag{24}$$

which, using the positive definiteness of V^* and [32, Lemma 4.3], can be bounded as $\underline{v}_l(\|r\|) \leq V_L(r,t) \leq \overline{v}_l(\|r\|)$ for class $\mathcal{K}_\infty$ functions $\underline{v}_l, \overline{v}_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Using (22), the normalized regressors $\frac{\omega}{\rho}$ and $\frac{\omega_{i,j}}{\rho_{i,j}}$ can be bounded as $\sup_{t\in\mathbb{R}_{\geq 0}}\|\frac{\omega}{\rho}\| \leq \frac{1}{2\sqrt{\nu\underline{\Gamma}}}$ for all $s \in \Omega$ and $\sup_{t\in\mathbb{R}_{\geq 0}}\|\frac{\omega_{i,j}}{\rho_{i,j}}\| \leq \frac{1}{2\sqrt{\nu\underline{\Gamma}}}$ for all $s_i \in \Omega_j$ for all $j \in \mathbb{S}$. The matrices G_R and G_σ can be bounded as $\sup_{s\in\Omega}\|G\| \leq \lambda_{\max}\{R^{-1}\}\overline{g}^2$ and $\sup_{s\in\Omega}\|G_\sigma\| \leq (\overline{\sigma' g})^2\lambda_{\max}\{R^{-1}\}$, respectively, where $\lambda_{\max}\{\cdot\}$ denotes the maximum eigenvalue.

Theorem 1: Provided the class of dynamics in (10), Assumption 1, and the sufficient gain conditions

$$\eta_{a1} + \eta_{a2} > \frac{1}{\sqrt{\nu\underline{\Gamma}}}(\eta_{c1} + \eta_{c2})\|W\|\overline{\|G_\sigma\|}, \tag{25}$$

$$\underline{c} > \frac{3\eta_{a1}}{\eta_{c2}} + \frac{3(\eta_{c1} + \eta_{c2})^2\|W\|^2\overline{\|G_\sigma\|}^2}{16\nu\underline{\Gamma}\eta_{c2}(\eta_{a1} + \eta_{a2})}, \tag{26}$$

$$v_l^{-1}(l) < \overline{v}_l^{-1}\left(\underline{v}_l(r)\right), \tag{27}$$

hold, then the system state $s(t)$ in (10), weight estimation errors $\tilde{W}_c(t)$ and $\tilde{W}_a(t)$, and policy $u(t)$ are UUB.

Proof: Let $z(t)$ for $t \in \mathbb{R}_{\geq 0}$ be a Filippov solution to the differential inclusion $\dot{z} \in K[h](z)$, where $K[\,\cdot\,]$ is defined in [33] and $h : \mathbb{R}^{n+2L+L^2} \to \mathbb{R}^{n+2L+L^2}$ is defined as $h \triangleq [\dot{s}^T, \dot{\tilde{W}}_c^T, \dot{\tilde{W}}_a^T, \mathrm{vec}(\dot{\Gamma}^{-1})^T]^T$. Due to the discontinuity in the update laws in (21)-(23), the time derivative of (24) exists almost everywhere (a.e., i.e., for almost all $t \in \mathbb{R}_{\geq 0}$) and $\dot{V}_L(z) \overset{a.e.}{\in} \dot{\tilde{V}}_L(z)$, where $\dot{\tilde{V}}_L$ is the generalized time-derivative of (24) along the Filippov trajectories of $\dot{z} = h(z)$ [34]. Using the calculus of $K[\,\cdot\,]$ from [34], and $\dot{V}^*(s) = V^{*\prime}(s)(F(s) + G(s)u(t))$, then substituting (21)-(23) yields

$$\begin{aligned}\dot{\tilde{V}}_L \subseteq\ & V^{*\prime}(F + Gu) - \tilde{W}_c^T\Gamma^{-1}\left(-\eta_{c1}\Gamma\frac{\omega}{\rho}\hat{\delta} - \eta_{c2}K\left[\Sigma_c^j\right]\right) \\ & - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\left[\lambda\Gamma - \eta_{c1}\frac{\Gamma\omega\omega^T\Gamma}{\rho} - \Gamma\eta_{c2}K\left[\Sigma_\Gamma^j\right]\Gamma\right]\Gamma^{-1}\tilde{W}_c \\ & - \tilde{W}_a^T\left(-\eta_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - \eta_{a2}\hat{W}_a(t)\right) \\ & - \tilde{W}_a^T\left[\frac{\eta_{c1}G_\sigma^T(t)\hat{W}_a(t)\omega^T(t)}{4\rho(t)}\hat{W}_c(t) + \eta_{c2}K\left[\Sigma_a^j\right]\hat{W}_c\right].\end{aligned} \tag{28}$$

Using the class of dynamics in (10), Assumption 1, and substituting the sufficient conditions in (25) and (26) yields

$$\dot{\tilde{V}}_L \overset{a.e.}{\leq} -v_l(\|Z\|),\ \forall\|Z\| \geq v_l^{-1}(l), \forall t \in \mathbb{R}_{\geq 0}, \tag{29}$$

where

$$v_l(\|Z\|) \leq \frac{\underline{q}\|x\|^2}{2} + \frac{(\eta_{a1} + \eta_{a2})}{16}\left\|\tilde{W}_a\right\|^2 + \frac{\eta_{c2}\underline{c}}{12}\left\|\tilde{W}_c\right\|^2, \tag{30}$$

where l is a known positive constant.

Since (24) is a common Lyapunov function across each segment $j \in \mathbb{S}$, [32, Th. 4.18] can be invoked to conclude that

[3] $\|\Gamma(t)\|$ is upper and lower bounded by some user-defined saturation gains, $\overline{\Gamma}$ and $\underline{\Gamma}$, respectively. Using (22) ensures that $\underline{\Gamma} \leq \|\Gamma(t)\| \leq \overline{\Gamma}$ for all $t \in \mathbb{R}_{>0}$, where $\underline{\Gamma} \in \mathbb{R}_{>0}$.

[4] The indicator function in (22) can be removed provided ρ and ρ_i are changed to $\rho = 1 + \nu\omega^T\omega$ and $\rho_i = 1 + \nu\omega_i^T\omega_i$, and additional assumptions are included for the regressors $\frac{\omega(t)}{\rho(t)}$ and $\Sigma_\Gamma^j(t)$ in order for $\Gamma(t)$ to be bounded (see, [5], [18]).

r is UUB such that $\limsup_{t\to\infty}\|r\| \le \underline{v}_l^{-1}(\overline{v}_l(\underline{v}_l^{-1}(l)))$. Since $r \in \mathcal{L}_\infty$, it follows that $s, \tilde{W}_c, \tilde{W}_a \in \mathcal{L}_\infty, \hat{W}_c, \hat{W}_a \in \mathcal{L}_\infty$, and $u \in \mathcal{L}_\infty$. Moreover, if $s \in \mathcal{L}_\infty$, by (4) $x \in \mathcal{L}_\infty$ and satisfies $x_i \in (a_i, A_i)$ for each $i \in \{1, \ldots, n\}$. ■

Remark 4: The sufficient condition in (25) can be satisfied by increasing the gains η_{a2} and ν, and selecting a penalty weight matrix R such that $\lambda_{\max}\{R^{-1}\}$ is small.[5] Selecting a R with a large minimum eigenvalue and a large gain ν will also help satisfy the gain condition in (26) by decreasing the right-hand-side. The sufficient condition in (26) can be satisfied by selecting off-policy trajectories for sparse BE extrapolation in each Ω_j such that the minimum eigenvalue $\underline{c} \le \underline{c}_j \triangleq \inf_{t\in\mathbb{R}_\ge}\{\Sigma_\Gamma^j(t)\}$ is large enough for each $j \in \mathbb{S}$.[6] Provided the basis functions used for approximation are selected such that $\overline{\sigma'}$, $\overline{\varepsilon}$, and $\overline{\varepsilon'}$ are small, and η_{a2}, $\lambda_{\max}\{R\}$, ν, and $\underline{c}$ are selected sufficiently large, then the sufficient condition in (27) can be satisfied.

V. Simulation

To demonstrate the performance of the developed method for a nonlinear system, simulation results are performed for the two-state dynamical system described in [35]. The simulation is performed with the system given in (1) where $x = [x_1, x_2]^T$,

$$f(x) = \begin{bmatrix} -x_1 + x_2 \\ -\frac{1}{2}x_1 - \frac{1}{2}x_2\left(1 - (\cos(2x_1) + 2)^2\right) \end{bmatrix},$$

and

$$g(x) = \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix}. \tag{31}$$

The control objective is to minimize (11), where $Q = \text{diag}\{0.01, 0.1\}$ and $R = 0.1$. To approximate the value function, a polynomial basis is selected as $\sigma(s(t)) = [s_1^2(t), s_1(t)s_2(t), s_2^2(t)]^T$. The barrier sets as defined in (2) are $x_1 \in (-5.25, 0.25)$ and $x_2 \in (-0.25, 5.25)$. To facilitate the sparse BE extrapolation, two segments are selected as $\Omega_1 \subset \mathbb{R}^2$ and $\Omega_2 \subset \mathbb{R}^2$ where $\Omega_1 \triangleq \{s \in \mathbb{R}^2 : b(-5.25, -5.25, -3.5) < s_1 < b(-3.5, -5.25, -3.5),\ b(3.5, 3.5, 5.25) < s_2 < b(5.25, 3.5, 5.25)\}$ and $\Omega_2 \triangleq \{s \in \mathbb{R}^2 : b(-3.5, -3.5, 0.25) \le s_1 < b(0.25, -3.5, 0.25),\ b(-0.25, -0.25, 3.5) \le s_2 < b(3.5, -0.25, 3.5)\}$. The basis used over segment 1 is $\sigma_i(s_i) = [s_{1,i}^2, s_{1,i}s_{2,i}, s_{2,i}^2]^T$ for all $s_i \in \Omega_1$ with $N_1 = 16$ extrapolated trajectories, while in segment 2 the second element is turned off such that $\sigma_i(s_i) = [s_{1,i}^2, 0, s_{2,i}^2]^T$ for all $s_i \in \Omega_2$ with $N_2 = 144$ extrapolated trajectories used for BE extrapolation.

The initial conditions for the system (i.e., $t = 0$) are $x(0) = [-5, 5]^T$, $\hat{W}_a(0) = \hat{W}_c(0) = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]^T$, $\Gamma(0) = 250 \times I_3$. The gains were selected as $\eta_{c1} = 0.001$, $\eta_{c2} = 5$, $\lambda = 0.5$, $\eta_{a1} = 25$, $\eta_{a2} = 0.1$, $\nu = 0.005$.

Figure 1 shows that the state converge to the origin while staying within the user-specified barriers. System parameters were selected to show the impact of the barriers on the system. As shown, as the state nears $x_1(t) = 0$, but does not cross over the boundary at $A_1 = 0.25$. By design, the state trajectory comes close to the barrier without intersecting it. As $x_2(t)$ approaches a_2, the controller forces the system trajectory away from the boundary, and toward the origin. The proposed method and the method from [29] obey the barrier constraints, whereas the methods in [5], [22] do not.

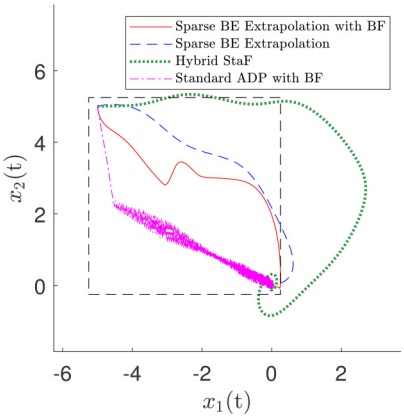

Fig. 1. State-space portrait for the system in (31). The black dashed lines represent the barriers. The solid red line denotes the trajectory of the proposed method, the dashed blue line denotes the trajectory using the method in [22], the dotted green line denotes the trajectory using the method in [5], the dash-dotted magenta line denotes the trajectory using the method in [29] without input saturation. Each method is simulated with the same Q and R values.

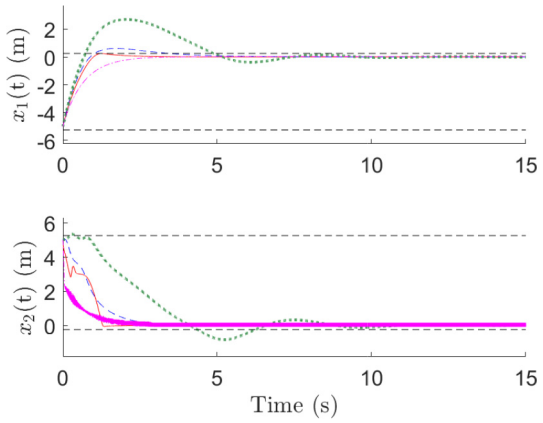

Fig. 2. State trajectory for the system in (31). The black horizontal dashed lines represent the barriers of each state. The proposed method is shown to converge to the origin first.

Figure 2 illustrates the convergence of the systems' states to the origin with respect to the barriers. The colors and line styles correspond to those in Figure 1. All of the simulated methods converge to the origin. The proposed method converges first. The noisy behavior of the simulation of [29] is partially due to added noise in the controller to satisfy the PE condition. The initial control input in [29] is high since the state is close to the barriers and due to the structure of (31).

Tab. I compares the proposed method to [5], [22], [29] in terms of execution time of a 15 second simulation. Each method was simulated in MATLAB.

Based on this simulation, BFs constrain an ADP algorithm that uses sparse BE extrapolation, which enables the system to converge to the origin while switching history stacks between active segments, thus formalizing system safety constraints. The developed method converges faster than [5], [22], [29] but is more computationally expensive.

[5]For $\lambda_{\max}\{R^{-1}\}$ to be small, $\lambda_{\min}\{R\}$ needs to be large, such that $\frac{1}{\lambda_{\max}\{R\}}I_m \le R^{-1} \le \frac{1}{\lambda_{\min}\{R\}}I_m = \lambda_{\max}\{R^{-1}\}I_m$.

[6]The minimum eigenvalue of each $\Sigma_\Gamma^j(t)$ can be increased by collecting redundant data, i.e., selecting $N_j \gg L$ for each segmented neighborhood $\Omega_j \subset \Omega$ and $j \in \mathbb{S}$.

TABLE I
SIMULATION EXECUTION TIME

Method	Computation Time (s)
Sparse BE Extrapolation with BF	3.39
[22]	2.58
[5]	2.71
[29]	1.97

VI. CONCLUSION

This letter presents a framework that combines the use of sparse BE extrapolation with BFs. Using this framework, a dynamic model can be used to evaluate the BE over unexplored areas of the state-space when the states have been transformed using BFs. An online approximate optimal controller is developed using sparse, segmented BE extrapolation and BFs to optimally regulate a dynamical system while providing formal safety guarantees. A BF transform is applied to a fully-constrained dynamical system to generate an unconstrained optimization problem. RL is used to solve the optimization problem online, leading to the development of an approximate optimal controller. The value function is approximated via sparse BE extrapolation over segments of the state-space. A Lyapunov-like stability analysis in the presence of discontinuities shows UUB regulation of the system states to the neighborhood of the origin and convergence of the control policy to the neighborhood of the optimal policy. A simulation of a two-state dynamical system compares the proposed method to related existing methods.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of sponsoring agencies.

REFERENCES

[1] D. Kirk, *Optimal Control Theory: An Introduction*. Mineola, NY, USA: Dover, 2004.
[2] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2012.
[3] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, Jan. 2013.
[4] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. E. Dixon, "Robust identification-based state derivative estimation for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 187–192, Jan. 2013.
[5] P. Deptula, J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "Approximate dynamic programming: Combining regional and local state following approximations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2154–2166, Jun. 2018.
[6] M. Johnson, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Approximate N-player nonzero-sum game solution for an uncertain continuous nonlinear system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1645–1658, Aug. 2015.
[7] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online policy iteration based algorithms to solve the continuous-time infinite horizon optimal control problem," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn.*, 2009, pp. 36–41.
[8] 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.
[9] 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.
[10] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, Jan. 2015.
[11] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, Apr. 2014.
[12] C. Qin, H. Zhang, and Y. Luo, "Online optimal tracking control of continuous-time linear systems with unknown dynamics by using adaptive dynamic programming," *Int. J. Control*, vol. 87, no. 5, pp. 1000–1009, 2014.
[13] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for infinite-horizon approximate optimal tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 753–758, Mar. 2017.
[14] G. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, Daniel Guggenheim School Aerosp. Eng., Georgia Inst. Technol., Dec. 2010.
[15] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.
[16] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, Feb. 2016.
[17] P. Walters, R. Kamalapurkar, F. Voight, E. Schwartz, and W. E. Dixon, "Online approximate optimal station keeping of a marine craft in the presence of an irrotational current," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 486–496, Apr. 2018.
[18] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.
[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2011, pp. 315–323.
[20] S. A. Nivison and P. Khargonekar, "Improving long-term learning of model reference adaptive controllers for flight applications: A sparse neural network approach," in *Proc. AIAA Guid. Navig. Control Conf.*, Jan. 2017, pp. 1–17.
[21] S. A. Nivison and P. Khargonekar, "A sparse neural network approach to model reference adaptive control with hypersonic flight applications," in *Proc. AIAA Guid. Navig. Control Conf.*, 2018, p. 0842.
[22] M. L. Greene, P. Deptula, S. Nivison, and W. E. Dixon, "Reinforcement learning with sparse Bellman error extrapolation for infinite-horizon approximate optimal regulation," in *Proc. IEEE Conf. Decis. Control*, Nice, France, Dec. 2019, pp. 1959–1964.
[23] S. A. Nivison, "Sparse and deep learning-based nonlinear control design with hypersonic flight applications," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Florida, 2017.
[24] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proc. Vol.*, vol. 40, no. 12, pp. 462–467, 2007.
[25] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Proc. Int. Workshop Hybrid Syst. Comput. Control*, 2004, pp. 477–492.
[26] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *Proc. Amer. Control Conf.*, 2015, pp. 4542–4548.
[27] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *Proc. IEEE Int. Symp. Intell. Control*, 2011, pp. 613–619.
[28] Y. Yang, D.-W. Ding, H. Xiong, Y. Yin, and D. C. Wunsch, "Online barrier-actor-critic learning for H_∞ control with full-state constraints and input saturation," *J. Franklin Inst.*, to be published.
[29] Y. Yang, Y. Yin, W. He, K. G. Vamvoudakis, H. Modares, and D. C. Wunsch, "Safety-aware reinforcement learning framework with an actor-critic-barrier structure," in *Proc. Amer. Control Conf.*, 2019, pp. 2352–2358.
[30] R. Kamalapurkar, P. S. Walters, J. A. Rosenfeld, and W. E. Dixon, *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach*. Cham, Switzerland: Springer, 2018.
[31] D. Wang and C. Mu, *Adaptive Critic Control With Robust Stabilization for Uncertain Nonlinear Systems*. Singapore: Springer, 2019.
[32] H. K. Khalil, *Nonlinear Systems*, 3 ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
[33] A. F. Filippov, "Differential equations with discontinuous right-hand side," in *Fifteen Papers on Differential Equations* (American Mathematical Society Translations-Series 2), vol. 42. Providence, RI, USA: Amer. Math. Soc., 1964, pp. 199–231.
[34] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.
[35] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, "Adaptive optimal controllers based on generalized policy iteration in a continuous-time framework," in *Proc. Mediterranean Conf. Control Autom.*, 2009, pp. 1402–1409.