Approximate Dynamic Programming: Combining Regional and Local State Following Approximations

Patryk Deptula, Joel A. Rosenfeld, Rushikesh Kamalapurkar, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*— An infinite-horizon optimal regulation problem for a control-affine deterministic system is solved online using a local state following (StaF) kernel and a regional model-based reinforcement learning (R-MBRL) method to approximate the value function. Unlike traditional methods such as R-MBRL that aim to approximate the value function over a large compact set, the StaF kernel approach aims to approximate the value function in a local neighborhood of the state that travels within a compact set. In this paper, the value function is approximated using a state-dependent convex combination of the StaF-based and the R-MBRL-based approximations. As the state enters a neighborhood containing the origin, the value function transitions from being approximated by the StaF approach to the R-MBRL approach. Semiglobal uniformly ultimately bounded (SGUUB) convergence of the system states to the origin is established using a Lyapunov-based analysis. Simulation results are provided for two, three, six, and ten-state dynamical systems to demonstrate the scalability and performance of the developed method.**

***Index Terms*— Data-driven control, local estimation, nonlinear control, optimal control, reinforcement learning.**

I. Introduction

SOLVING the Hamilton Jacobi Bellman (HJB) equation yields the value function, which is used to determine an optimal controller. Because the HJB is a nonlinear partial differential equation that is generally infeasible to solve analytically or in real time, an approximate solution is often used. For example, by using parametric approximation methods, such as neural-networks (NNs), the optimal value function can be estimated and used to compute an approximate optimal policy. To establish closed-loop stability, the error between the optimal and estimated value function needs to decay to a small bound sufficiently fast [1].

Manuscript received April 17, 2017; revised October 6, 2017 and January 19, 2018; accepted February 10, 2018. Date of publication March 15, 2018; date of current version May 15, 2018. This work was supported in part by NSF under Grant 1509516, in part by the Office of Naval Research under Grant N00014-13-1-0151, and in part by the OSD Sponsored Autonomy Research Pilot Initiative project entitled A Privileged Sensing Framework. *(Corresponding author: Patryk Deptula.)*

P. Deptula and W. E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: pdeptula@ufl.edu; wdixon@ufl.edu).

J. A. Rosenfeld is with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA (e-mail: joelar@ufl.edu).

R. Kamalapurkar is with the School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74074 USA (e-mail: rushikesh.kamalapurkar@okstate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2808102

The rate at which the value function approximation error decays is determined by the richness of the data utilized for learning. In traditional adaptive dynamic programming (ADP) methods such as [2] and [3] richness of the data correlates to the amount of excitation in the system. Typically excitation is introduced by adding an exploration signal to the control input (see [4]–[9]). Because the addition of the exploration signal causes undesirable oscillations and noise, hardware implementation of traditional ADP techniques such as [2], [10]–[14], and [15] is challenging. In data-driven experience replay-based techniques such as [16]–[20], data richness is quantified by the eigenvalues of the recorded history stack. However, the required amount of data storage grows exponentially as the demand for richer data increases, making hardware implementation challenging.

Approximating the value function over a large region typically requires a large number of basis functions. For general nonlinear systems, generic basis functions, such as Gaussian radial basis functions, sigmoid functions, polynomials, or universal kernel functions are used to approximate the value function (see [1], [14], [19], [21]–[29]). One limitation of these generic approximation methods is that they only ensure approximation over a compact neighborhood of the origin. Once outside the compact set, the approximation tends to either grow or decay depending on the selected functions. Consequently, in the absence of domain knowledge, a large number of basis functions, and hence, a large number of unknown parameters, is required for value function approximation. Reduction in the number of unknown parameters motivates the use of StaF basis functions such as [21] which travel with the state to maintain an accurate local approximation. However, the StaF approximation method trades global optimality for computational efficiency since it lacks memory. Since accurate estimation of the value function results in a better closed-loop response and lower operating costs, it is desirable to accurately estimate the value function near the origin in optimal regulation problems.

In this paper, a novel framework is developed to merge local and regional value function approximation methods to yield an online optimal control method that is computationally efficient and simultaneously accurate over a specified critical region of the state space. Ability to R-MBRL such as [19] to approximate the value function over a predefined region and the computational efficiency of the StaF method [1] in approximating the value function locally along the state trajectory motivates the following development. Instead of generating an

approximation of the value function over the entire operating region, which would be computationally expensive, the operating domain is separated into two regions: a closed set A, containing the origin, where a regional approximation method is used to approximate the value function, and the complement of A, where the StaF method is used to approximate the value function. Using a switching-based approach to combine regional and local approximations would inject discontinuities to the system and result in a nonsmooth value function which would introduce discontinuities in the control signal. To overcome this challenge, a state varying convex combination of the two approximation methods is used to ensure a smooth transition from the StaF to the R-MBRL approximation as the state enters the closed convex set containing the origin. Once the state enters the this region, R-MBRL regulates the state to the origin. The developed result is generalized to allow for the use of any R-MBRL method.

While the StaF method is computationally efficient, it lacks memory, i.e., the information about the value function in a region is lost once the system state leaves that region. To maintain an accurate approximation of the value function near the goal state (i.e., the origin), the developed method uses R-MBRL in A; the weights are learned based on selected points in that set and the value function does not have to be relearned once the state leaves this neighborhood. The developed architecture is motivated by the observation that in many applications such as station keeping of marine craft, like in [30], accurate approximation of the value function in a neighborhood of the goal state can improve the performance of the closed-loop system near the goal state.

Since the StaF method uses state-dependent centers, the unknown optimal weight are themselves also state dependent, which makes analyzing stability difficult. To add to the technical challenge, using a convex combination of R-MBRL and StaF results in a complex representation of the value function and resulting Bellman error (BE). To provide insights into how to combine StaF and R-MBRL while also preserving stability, the estimates are designed using a Lyapunov-based stability analysis. The analysis of the closed-loop systems with the smoothly switching approximation guarantees semi-global uniformly ultimately bounded (SGUUB) convergence. The performance of the developed method is illustrated through numerical simulations. Simulations are provided for a two-state system with a known value function as well as three, six, and ten-state systems with unknown value functions to illustrate the scalability of the method in terms of computational time, cost, and final root-mean-square (RMS) error. Comparisons with [1] and [19] illustrate the advantage of the developed method.

This paper is organized as follows. Section II introduces the optimal control problem. The motivation for using a combination of the StaF and R-MBRL methods is discussed in Section III. The proposed value function approximation scheme along with the derived BE are presented in Section IV, and BE extrapolation for online learning along with the actor and critic weight update laws is discussed in Section V. Section VI presents a Lyapunov stability analysis. Simulations are discussed in Section VII, while conclusions are drawn in Section VIII.

Notation: In the following development, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the sets of real n-vectors and $n \times m$ matrices, and $\mathbb{R}_{\geq a}$ and $\mathbb{R}_{>a}$ denote the sets of real numbers greater than or equal to a and strictly greater than a, respectively, where $a \in \mathbb{R}$. The $n \times n$ identity matrix and column vector of ones of dimension j are denoted by I_n and 1_j, respectively. The partial derivative of h with respect to the state x is denoted by $\nabla h(x, y, \ldots)$. The notation $(\cdot)^o$ denotes an arbitrary variable of the set which the variable belongs to, and $(\cdot)^T$ denotes the transpose of a matrix or vector. The notation $G_{\nabla F}$, $G_{\nabla F \nabla K}$, G_F, G_{FK}, and $G_{\nabla FK}$ is defined as $G_{\nabla F} \triangleq \nabla F g R^{-1} g^T \nabla F^T$, $G_{\nabla F \nabla K} \triangleq \nabla F g R^{-1} g^T \nabla K^T$, $G_F \triangleq F g R^{-1} g^T F^T$, $G_{FK} \triangleq F g R^{-1} g^T K^T$, and $G_{\nabla FK} \triangleq \nabla F g R^{-1} g^T K^T$, respectively, where F and K denote arbitrary functions.

II. Problem Formulation

Consider a control affine nonlinear dynamical system

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

where $x : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denotes the system state, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes the control effectiveness, and $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$ denotes the control input.

Assumption 1: Both f and g are assumed to be locally Lipschitz continuous. Furthermore, $f(0) = 0$, and $\nabla f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is continuous.

In the following, the notation $\phi^u(t; t_0, x_0)$ denotes the trajectory of the system in (1) under the controller u with initial condition $x_0 \in \mathbb{R}^n$ and initial time $t_0 \in \mathbb{R}_{\geq 0}$. The objective is to solve the infinite-horizon optimal regulation problem, i.e., find a control policy u online to minimize the cost functional

$$J(x, u) \triangleq \int_{t_0}^{\infty} r(x(\tau), u(\tau)) d\tau \tag{2}$$

while regulating the system states to the origin under the dynamic constraint (1). In (2), $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ denotes the instantaneous cost defined as

$$r(x^o, u^o) \triangleq x^{oT} Q x^o + u^{oT} R u^o \tag{3}$$

for all $x^o \in \mathbb{R}^n$ and $u^o \in \mathbb{R}^m$, where $R \in \mathbb{R}^{m\times m}$ and $Q \in \mathbb{R}^{n\times n}$ are constant positive definite matrices and the matrix Q can be bounded as $\underline{q}\|x^o\|^2 \leq x^{oT} Q x^o \leq \overline{q}\|x^o\|^2$.

The infinite-horizon scalar value function for the optimal solution, i.e., the function which maps each state to the total cost-to-go, denoted by $V^\star : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, can be expressed as

$$V^\star(x^o) = \inf_{u(\tau)\in U \,|\, \tau \in \mathbb{R}_{\geq t}} \int_t^{\infty} r(\phi^u(\tau; t, x^o), u(\tau)) d\tau \tag{4}$$

where $U \subset \mathbb{R}^m$ is the action space. The optimal value function is characterized by the corresponding HJB equation

$$\nabla V^\star(x^o)(f(x^o) + g(x^o)u^\star(x^o)) + r(x^o, u^\star(x^o)) = 0 \tag{5}$$

with the boundary condition $V(0) = 0$, where $u^\star : \mathbb{R}^n \to \mathbb{R}^m$ is the optimal control policy which can be determined from (5) as

$$u^\star(x^o) \triangleq -\frac{1}{2} R^{-1} g^T(x^o)(\nabla V^\star(x^o))^T. \tag{6}$$

Using (6), the open-loop HJB in (5) can be expressed in a closed-loop form as

$$-\frac{1}{4}\nabla V^\star(x^o) g(x^o) R^{-1} g^T(x^o)(\nabla V^*(x^o))^T + \nabla V^\star(x^o) f(x^o) + x^{oT} Q x^o = 0. \tag{7}$$

The analytical expression in (6) requires knowledge of the optimal value function which is the solution to the HJB in (5), but since the analytical solution for the HJB is generally infeasible to compute, an approximation of the solution is sought.

III. Combining Regional and Local State Following Approximations

Traditional approaches to approximating the value function establish the approximation over the entire state-space. When implementing the approximation online, traditional methods spend computational resources approximating the value function in regions where the state may not enter. The StaF method reduces the computational efforts of the approximation problem by approximating the value function in a moving neighborhood of the state.

A drawback of the StaF method is that it does not establish an approximation of the value function in regions where the state will travel in the future; the StaF method only approximates the value function at the *current* position of the state. In general, it is difficult to provide a perfect prediction of the future state of an uncertain nonlinear system. However, since convergence to the origin is the goal of regulation problems, approximating the function in a neighborhood around the origin is well motivated.

The operating domain χ of the state is segregated into two sets, the set A, which is a closed compact set containing the the origin, and the set $B = \chi \setminus A$. Two different approximation strategies will be used over A and B. Various R-MBRL methods can be used to approximate the value function inside A. For the set B, the StaF method is employed since there are large regions of B that the state does not visit for the regulation problem. Thus, the value function is approximated by the StaF method when the state is in B and some R-MBRL method is used when the state is in A. A regional approximation method is also used to approximate the value function in the set $A' = \{x \in \chi : d(x, A) \leq \ell\}$ (also known as an *inflation* of A), where $d(x, A) = \inf\{d(x, y) : y \in A\}$ and $\ell \in \mathbb{R}_{>0}$ is a constant, and approximation of the value function over the transition region $A' \setminus A$ will be a state-dependent convex combination of the two controllers.

Let $\hat{V}_1(x)$ denotes the approximation of the value function over A' using the R-MBRL method, and denote $\hat{V}_2(x)$ as the StaF approximation of the value function over B. The resulting approximation of the value function over χ will then be $\hat{V}(x) = \lambda(x)\hat{V}_1(x) + (1-\lambda(x))\hat{V}_2(x)$, where $\lambda : \chi \to [0, 1]$ such that $\lambda(x) = 1$ when $x \in A$ and $\lambda(x) = 0$ when $x \in \chi \setminus A' \subset B$. If $\epsilon > 0$ and $|\hat{V}_1(x) - V^\star(x)| < \epsilon$ over A' and $|\hat{V}_2(x) - V^\star(x)| < \epsilon$ over B, then $|\hat{V}(x) - V^\star(x)| < \epsilon$ for all $x \in \chi$, since $\hat{V}$ is a convex combination of $\hat{V}_1$ and $\hat{V}_2$ over the transition region $A' \setminus A \subset B$.

The following analysis is agnostic with respect to the compact set A and the transition function λ. However, the transition function λ should be a continuously differentiable compactly supported function such that $\|\nabla\lambda(x^o)\| \leq \overline{\nabla\lambda}$, where $\overline{\nabla\lambda} \in \mathbb{R}_{>0}$. An example of such a function is

$$\lambda(x) = \begin{cases} 1, & x \in A \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{d(x, A)}{\ell}\right)\right], & x \in A' \setminus A \\ 0, & x \notin A'. \end{cases} \tag{8}$$

Examples of A for which λ is continuously differentiable include $[-1, 1]^n$ as well as $\overline{B_1(0)} = \{y \in \mathbb{R}^n : \|y\| \leq 1\}$.

IV. Value Function Approximation

The value function $V^\star$ evaluated at x^o using StaF kernels centered at $y^o \in \overline{B_r(x^o)}$ can be represented using a convex combination as[1]

$$V^\star(x^o) = \lambda(x^o) W_1^T \sigma(x^o) + (1 - \lambda(x^o)) W_2^T(y^o)\phi(x^o, c(y^o)) + \epsilon(x^o, y^o). \tag{9}$$

In (9), $\sigma : \chi \to \mathbb{R}^P$ is a bounded vector of continuously differentiable nonlinear basis functions such that $\sigma(0) = 0$ and $\nabla\sigma(0) = 0$, $\phi\big(x^o, c(y^o)\big) = \big[k(x^o, c_1(y^o), k(x^o, c_2(y^o), \ldots, k(x^o, c_L(y^o)\big]^T$ where $k : \chi \times \chi^L \to \mathbb{R}^L$ is a strictly positive definite continuously differentiable kernel, $W_1 \in \mathbb{R}^P$ is a constant ideal R-MBRL weight vector which is upper-bounded by a known positive constant $\overline{W}_1$ such that $\|W_1\| \leq \overline{W}_1$ (see [19], [20], [31]–[33]). Furthermore, $W_2 : \chi \to \mathbb{R}^L$ is the continuously differentiable ideal local StaF weight function which changes with the state-dependent centers, and $\epsilon : \chi \to \mathbb{R}$ is the continuously differentiable function reconstruction error such that $\sup_{x^o \in \chi,\, y^o \in \overline{B_r(x^o)}} |\epsilon(x^o, y^o)| \leq \overline{\epsilon}$ and $\sup_{x^o \in \chi,\, y^o \in \overline{B_r(x^o)}} |\nabla\epsilon(x^o, y^o)| \leq \overline{\nabla\epsilon}$.

The subsequent analysis is based on an approximation of the value function and optimal policy, evaluated at x^o using StaF kernels centered at $y^o \in \overline{B_r(x^o)}$, expressed as

$$\hat{V}(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}) = \lambda(x^o)\hat{W}_{1c}^T\sigma(x^o) + (1 - \lambda(x^o))\hat{W}_{2c}^T\phi(x^o, c(y^o)) \tag{10}$$

[1]The value function can be expressed as a single NN by stacking the StaF and R-MBRL basis functions. However, the single NN representation would function exactly like the two separate NNs because the StaF and R-MBRL weights are trained using different update laws, and the StaF weights are not connected to the R-MBRL basis functions and vice-versa. Hence, to make the structure of the NN apparent, the value function is expressed via two different NNs.

and

$$\hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a}) = -\frac{1}{2}R^{-1}g^T(x^o) \times (\lambda(x^o)\nabla\sigma^T(x^o)\hat{W}_{1a} + (1-\lambda(x^o)) \times \nabla\phi^T(x^o, c(y^o))\hat{W}_{2a} + \nabla\lambda^T(x^o) \times (\sigma^T(x^o)\hat{W}_{1a} - \phi^T(x^o, c(y^o))\hat{W}_{2a})). \quad (11)$$

In (10) and (11), $\hat{W}_{1a}$, $\hat{W}_{1c} \in \mathbb{R}^P$ and $\hat{W}_{2a}$, $\hat{W}_{2c} \in \mathbb{R}^L$ are weight estimates for the ideal weight vectors W_1 and $W_2(y^o)$, respectively, and λ denotes the transition function introduced in Section III. In an approximate actor-critic-based solution, the optimal value function $V^\star$ and control policy $u^\star$ in (5) are replaced by their respective estimates $\hat{V} : \chi \times \mathbb{R}^L \times \mathbb{R}^P \to \mathbb{R}$ and $\hat{u} : \chi \times \mathbb{R}^L \times \mathbb{R}^P \to \mathbb{R}^m$. This results in a residual error $\delta : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}$ called the BE which is defined as

$$\delta(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}, \hat{W}_{1a}, \hat{W}_{2a}) \triangleq \nabla\hat{V}(x^o, y^o, \hat{W}_{1c}, \hat{W}_{2c}) \times (f(x^o) + g(x^o)\hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a})) + r(x^o, \hat{u}(x^o, y^o, \hat{W}_{1a}, \hat{W}_{2a})). \quad (12)$$

Motivated by classical ADP solutions which aim to find a set of weights so that the BE is zero $\forall x^o \in \mathbb{R}^n$, to solve the optimal control problem, the critics and actors aim to find a set of weights that minimize the BE $\forall x^o \in \mathbb{R}^n$.

V. Online Learning

At a given time instant t, the BE $\delta_t : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is evaluated as

$$\delta_t(t) \triangleq \delta(x(t), x(t), \hat{W}_{1c}(t), \hat{W}_{2c}(t), \hat{W}_{1a}(t), \hat{W}_{2a}(t)) \quad (13)$$

where $\hat{W}_{1c}$, $\hat{W}_{1a}$, and $\hat{W}_{2c}$, $\hat{W}_{2a}$, denote estimates of the critic and actor weights for the R-MBRL approximation method and StaF approximation method, respectively, at time t. Furthermore, $x(t)$ denotes the state of the system in (1) when starting from initial time t_0 and initial state x_0 under the influence of the state feedback controller

$$u(t) = \hat{u}(x(t), x(t), \hat{W}_{1a}(t), \hat{W}_{2a}(t)). \quad (14)$$

The BE is extrapolated to unexplored areas of the state space to learn via simulation of experience (see [1], [19]). The critic $\hat{W}_{1c}$ selects sample points $\{x_i \in A' | i = 1, \ldots, N\}$ based on prior information about the desired behavior of the system, i.e., selected about the origin, and evaluates a form of the BE, $\delta_{1t,i} : \mathbb{R}_{\geq t_0} \to \mathbb{R}$. Similarly, sample trajectories $\{x_j(x(t), t) \in B_r(x(t)) | j = 1, 2, \ldots, M\}$ that follow the current state $x(t)$ are selected so that the StaF critic $\hat{W}_{2c}$ evaluates another extrapolated form of the BE $\delta_{2t,j} : \mathbb{R}_{\geq t_0} \to \mathbb{R}$. The extrapolated BEs are expressed as

$$\delta_{1t,i}(t) = \hat{W}_{1c}^T(t)\omega_{\nabla\sigma i}(t) + r(x_i, \hat{u}_i(t)) \quad (15)$$

$$\delta_{2t,j}(t) = \hat{W}_{2c}^T(t)\omega_{\nabla\phi j}(t) + r(x_j(x(t), t), \hat{u}_j(t)) \quad (16)$$

where

$$\omega_{\nabla\sigma i}(t) \triangleq \nabla\sigma(x_i)(f(x_i) + g(x_i)\hat{u}_i(t))$$

$$\omega_{\nabla\phi j}(t) \triangleq \nabla\phi(x_j(x(t), t), c(x(t)))(f(x_j(x(t), t)) + g(x_j(x(t), t))\hat{u}_j(t))$$

and

$$\hat{u}_i(t) = -\frac{1}{2}R^{-1}g^T(x_i)\nabla\sigma(x_i)^T\hat{W}_{1a}(t)$$

$$\hat{u}_j(t) = -\frac{1}{2}R^{-1}g^T(x_j(x(t), t)) \times \nabla\phi(x_j(x(t), t), c(x(t)))^T\hat{W}_{2a}(t).$$

A. Regional Update Laws

The BE and extrapolated BE in (13) and (15), respectively, contain R-MBRL actor and critic estimates, $\hat{W}_{1a}$ and $\hat{W}_{1c}$. Various approximation methods could be used to evaluate the BE in A. See [1], [19], [31], [34], [35] for examples of R-MBRL actor and critic update laws.

B. Local Update Laws

While the state is not in the local domain of A', the StaF critic uses the BEs in (13) and (16) to improve the estimate of $\hat{W}_{2c}$. Specifically, the StaF critic can be designed using the recursive least-squares update law

$$\dot{\hat{W}}_{2c}(t) = -k_{c1}\Gamma_2(t)\frac{\omega_{\nabla\phi}(t)}{\rho_2(t)}\delta_t(t) - \frac{k_{c2}}{M}\Gamma_2(t)\sum_{j=1}^{M}\frac{\omega_{\nabla\phi j}(t)}{\rho_{2j}(t)}\delta_{2t,j}(t) \quad (17)$$

$$\dot{\Gamma}_2(t) = \beta_2\Gamma_2(t) - k_{c1}\Gamma_2(t)\frac{\omega_{\nabla\phi}(t)\omega_{\nabla\phi}^T(t)}{\rho_2^2(t)}\Gamma_2(t) - \frac{k_{c2}}{M}\Gamma_2(t)\sum_{j=1}^{M}\frac{\omega_{\nabla\phi j}(t)\omega_{\nabla\phi j}^T(t)}{\rho_{2j}^2(t)}\Gamma_2(t) \quad (18)$$

where $\Gamma_2(t_0) = \Gamma_{2o}$ and $\Gamma_2(t)$ is the least-squares learning gain matrix, k_{c1}, k_{c2}, $\in \mathbb{R}_{\geq 0}$ are constant adaptation gains, $\beta_2 \in \mathbb{R}_{\geq 0}$ is a constant forgetting factor, $\rho_2(t) \triangleq 1 + \gamma_2\omega_{\nabla\phi}^T(t)\omega_{\nabla\phi}(t)$, $\rho_{2j}(t) \triangleq 1 + \gamma_2\omega_{\nabla\phi j}^T(t)\omega_{\nabla\phi j}(t)$, and $\gamma_2 \in \mathbb{R}_{\geq 0}$ is a constant positive gain. In (18)

$$\omega_{\nabla\phi}(t) \triangleq ((1-\lambda(x(t)))\nabla\phi(x(t), c(x(t))) - \phi(x(t), c(x(t)))\nabla\lambda(x(t))) \times (f(x(t)) + g(x(t))\hat{u}(x(t), x(t), \hat{W}_{1a}(t), \hat{W}_{2a}(t))) \quad (19)$$

is an instantaneous regressor matrix. The StaF actor update law is given by

$$\dot{\hat{W}}_{2a}(t) = -k_{a1}(\hat{W}_{2a}(t) - \hat{W}_{2c}(t)) - k_{a2}\hat{W}_{2a}(t) + \frac{k_{c1}G_{\nabla\phi}^T(t)\hat{W}_{2a}(t)\omega_{\nabla\phi}^T(t)}{4\rho_2(t)}\hat{W}_{2c}(t) + \frac{k_{c2}}{4M}\sum_{j=1}^{M}\frac{G_{\nabla\phi j}^T(t)\hat{W}_{2a}(t)\omega_{\nabla\phi j}^T(t)}{\rho_{2j}(t)}\hat{W}_{2c}(t) \quad (20)$$

where k_{a1}, $k_{a2} \in \mathbb{R}$ are positive constant adaptation gains and $G_{\nabla\phi}(t) \triangleq \nabla\phi(x(t), c(x(t)))g(x(t))R^{-1}g^T(x(t))\nabla\phi^T(x(t), c(x(t)))$.

Remark 1: In typical BE extrapolation approaches, the extrapolated BEs $\delta_{1t,i}$, $\delta_{2t,j}$ and controls $\hat{u}_i(t)$, $\hat{u}_j(t)$ take similar forms to the actual BE δ_t and control $u(t)$, respectively, with the exception of using extrapolated states. However, the extrapolated BEs and extrapolated inputs in this paper take a different form compared to the true BE and control. The goal is to approximate the ideal weight W_1 irrespective of the system state and W_2 in a region around the state, therefore the extrapolated BEs do not rely on a convex combination in the transition region $A' \setminus A$. Furthermore, when the state is in $B = \chi \setminus A$, only BE extrapolation is used in A' to approximate the weight W_1. Hence, the developed method is fundamentally different from the approach in [1] and [19].

VI. Stability Analysis

For notational brevity, time dependence of all signals is suppressed hereafter. The approach in this paper was generalized to allow the use of any model-based approximation method in A. However, to facilitate the following analysis a certain structure is given to the R-MBRL update laws. Without a loss of generality, let the R-MBRL update laws take a similar form to the StaF update laws in (17), (18), and (20). The R-MBRL update laws contain the extrapolated regressor $\omega_{\nabla\sigma i}$ defined in Section V, where the regressor $\omega_{\nabla\sigma}$ is defined as

$$\omega_{\nabla\sigma} \triangleq (\lambda(x)\nabla\sigma(x) + \sigma(x)\nabla\lambda(x))(f(x) + g(x)u) \quad (21)$$

where the BE δ_t is defined in (22), and the extrapolated BE $\delta_{1t,i}$ is defined in (23). The constant gains for the R-MBRL update laws are $\eta_{c1}, \eta_{c2} \in \mathbb{R}_{\geq 0}$. The R-MBRL least-squares learning gain matrix is $\Gamma_1(t)$ with a forgetting factor $\beta_1 \in \mathbb{R}_{\geq 0}$, and with normalizing factors $\rho_1(t) \triangleq 1 + \gamma_1\omega_{\nabla\sigma}^T(t)\omega_{\nabla\sigma}(t)$, $\rho_{1i}(t) \triangleq 1 + \gamma_1\omega_{\nabla\sigma i}^T(t)\omega_{\nabla\sigma i}(t)$ where $\gamma_1 \in \mathbb{R}_{\geq 0}$ is a constant positive gain.

To facilitate the analysis, let $\tilde{W}_{1a} \triangleq W_1 - \hat{W}_{1a}$, $\tilde{W}_{1c} \triangleq W_1 - \hat{W}_{1c}$, $\tilde{W}_{2a} \triangleq W_2 - \hat{W}_{2a}$, and $\tilde{W}_{2c} \triangleq W_2 - \hat{W}_{2c}$ denote the weight estimation errors. Unmeasurable forms of the BEs in (13), (15), and (16) can be written as

$$\delta_t = \delta_{t1} + \delta_{t2} + \delta_{t3} \quad (22)$$

where

$$\begin{aligned}
\delta_{t1} &= -\omega_{\nabla\sigma}^T\tilde{W}_{1c} + \frac{1}{4}\lambda^2\tilde{W}_{1a}^T G_{\nabla\sigma}\tilde{W}_{1a} + \Delta_1 \\
\delta_{t2} &= -\omega_{\nabla\phi}^T\tilde{W}_{2c} + \frac{1}{4}(1-\lambda)^2\tilde{W}_{2a}^T G_{\nabla\phi}\tilde{W}_{2a} + \Delta_2 \\
\delta_{t3} &= \frac{1}{2}(1-\lambda)\big(\lambda\tilde{W}_{2a}^T G_{\nabla\phi\nabla\sigma}\tilde{W}_{1a} + \tilde{W}_{1a}^T\sigma G_{\nabla\lambda\nabla\phi}\tilde{W}_{2a} \\
&\qquad - \frac{1}{2}\tilde{W}_{2a}^T\phi G_{\nabla\lambda\nabla\phi}\tilde{W}_{2a}\big) \\
&\quad + \frac{1}{4}\big(\tilde{W}_{1a}^T\sigma G_{\nabla\lambda}\sigma^T\tilde{W}_{1a} \\
&\qquad - 2\tilde{W}_{2a}^T\phi G_{\nabla\lambda}\sigma^T\tilde{W}_{1a} + \tilde{W}_{2a}^T\phi G_{\nabla\lambda}\phi^T\tilde{W}_{2a}\big) \\
&\quad + \frac{1}{2}\lambda\big(\tilde{W}_{1a}^T\sigma G_{\nabla\lambda\nabla\sigma}\tilde{W}_{1a} - \tilde{W}_{2a}^T\phi G_{\nabla\lambda\nabla\sigma}\tilde{W}_{1a}\big) + \Delta_3
\end{aligned}$$

and

$$\begin{aligned}
\delta_{1t,i} &= -\omega_{\nabla\sigma i}^T\tilde{W}_{1c} + \frac{1}{4}\tilde{W}_{1a}^T G_{\nabla\sigma i}\tilde{W}_{1a} + \Delta_{1i} \\
\delta_{2t,j} &= -\omega_{\nabla\phi j}^T\tilde{W}_{2c} + \frac{1}{4}\tilde{W}_{2a}^T G_{\nabla\phi j}\tilde{W}_{2a} + \Delta_{2j} \quad (23)
\end{aligned}$$

where the functions $\Delta_1, \Delta_2, \Delta_3, \Delta_{1i}, \Delta_{2j} : \mathbb{R}^n \to \mathbb{R}$ are uniformly bounded over χ such that the bounds $\{\overline{\|\Delta_k\|} | k = 1, 2, 3\}$, $\overline{\|\Delta_{1i}\|}$, and $\overline{\|\Delta_{2j}\|}$ decrease with decreasing $\overline{\|\nabla\epsilon\|}$ and $\overline{\|\nabla W\|}$.

Using the R-MBRL and StaF update laws, the system states x and selected states x_i and x_j are assumed to satisfy the following inequalities.

Assumption 2: There exists a positive constant $T \in \mathbb{R}_{\geq 0}$ such that

$$\begin{aligned}
\underline{c}_1 I_P &\leq \int_t^{t+T}\left(\frac{\omega_{\nabla\sigma}(\tau)\omega_{\nabla\sigma}^T(\tau)}{\rho_1^2(\tau)}\right)d\tau \quad \forall t \in \mathbb{R}_{\geq t_0} \\
\underline{c}_2 I_P &\leq \inf_{t\in\mathbb{R}_{\geq t_0}}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_{\nabla\sigma i}(t)\omega_{\nabla\sigma i}^T(t)}{\rho_{1i}^2(t)}\right) \\
\underline{c}_3 I_P &\leq \frac{1}{N}\int_t^{t+T}\left(\sum_{i=1}^{N}\frac{\omega_{\nabla\sigma i}(\tau)\omega_{\nabla\sigma i}^T(\tau)}{\rho_{1i}^2(\tau)}\right)d\tau \quad \forall t \in \mathbb{R}_{\geq t_0} \\
\underline{b}_1 I_L &\leq \int_t^{t+T}\left(\frac{\omega_{\nabla\phi}(\tau)\omega_{\nabla\phi}^T(\tau)}{\rho_1^2(\tau)}\right)d\tau \quad \forall t \in \mathbb{R}_{\geq t_0} \\
\underline{b}_2 I_L &\leq \inf_{t\in\mathbb{R}_{\geq t_0}}\left(\frac{1}{M}\sum_{j=1}^{M}\frac{\omega_{\nabla\phi j}(t)\omega_{\nabla\phi j}^T(t)}{\rho_{2j}^2(t)}\right) \\
\underline{b}_3 I_L &\leq \frac{1}{M}\int_t^{t+T}\left(\sum_{j=1}^{M}\frac{\omega_{\nabla\phi j}(\tau)\omega_{\nabla\phi j}^T(\tau)}{\rho_{2j}^2(\tau)}\right)d\tau \quad \forall t \in \mathbb{R}_{\geq t_0}
\end{aligned}$$

where $\{\underline{c}_k | k = 1, 2, 3\}$, $\{\underline{b}_k | k = 1, 2, 3\} \in \mathbb{R}_{\geq 0}$ are nonnegative constants, and at least one of the constants from each set is strictly positive.

Remark 2: Assumption 2 requires the regressors $\omega_{\nabla\sigma}$, $\omega_{\nabla\phi}$ or $\omega_{\nabla\sigma i}$, $\omega_{\nabla\phi j}$ to be persistently exciting. The regressors $\omega_{\nabla\sigma}$ and $\omega_{\nabla\phi}$ are completely determined by the state x and weights $\hat{W}_{1a}$ and $\hat{W}_{2a}$. Typically, to ensure that $\underline{c}_1, \underline{b}_1 > 0$, meaning $\omega_{\nabla\sigma}$ and $\omega_{\nabla\phi}$ are persistently excited, a probing signal is added to the control input. However, this introduces undesired oscillations in the system and produces noisy signals in the response. In addition, as the system and state converge to the origin, excitation will usually vanish. Hence, it is difficult to ensure that $\underline{c}_1, \underline{b}_1 > 0$. On the other hand, $\omega_{\nabla\sigma i}$ and $\omega_{\nabla\phi j}$ are dependent on x_i and x_j, which are designed independent of the system state x. In fact, $\omega_{\nabla\sigma i}$ is designed based on the desired behavior of the system, i.e., regulate the states to the origin. Therefore, without the need of a probing signal, $\underline{c}_2$ and $\underline{b}_2$ can be made strictly positive by selecting a sufficient number of extrapolated sample states in both regions of the state space, or if

x_i and x_j contain enough frequencies then $\underline{c}_3$, $\underline{b}_3$ become strictly positive.[2]

Let a candidate Lyapunov function $V_L : \mathbb{R}^{n+2L+2P} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ be defined as

$$V_L(Z,t) = V^\star(x) + \frac{1}{2}\tilde{W}_{1c}^T\Gamma_1^{-1}(t)\tilde{W}_{1c} + \frac{1}{2}\tilde{W}_{2c}^T\Gamma_2^{-1}(t)\tilde{W}_{2c} + \frac{1}{2}\tilde{W}_{1a}^T\tilde{W}_{1a} + \frac{1}{2}\tilde{W}_{2a}^T\tilde{W}_{2a}$$

where $V^\star$ is the unknown, positive, continuously differentiable optimal value function, and

$$Z = \left[x^T, \tilde{W}_{1a}^T, \tilde{W}_{1c}^T, \tilde{W}_{2a}^T, \tilde{W}_{2c}^T\right]^T.$$

The least-squares update laws which take the form of (18) ensure that the least-squares gain matrices satisfy [37, Corollary 4.3.2]

$$\underline{\Gamma_1} I_P \leq \Gamma_1(t) \leq \overline{\Gamma_1} I_P \tag{24}$$

$$\underline{\Gamma_2} I_L \leq \Gamma_2(t) \leq \overline{\Gamma_2} I_L \tag{25}$$

provided the minimum eigenvalues $\lambda_{\min}\left\{\Gamma_{1o}^{-1}\right\}$, $\lambda_{\min}\left\{\Gamma_{2o}^{-1}\right\} > 0$ and Assumption 2 holds (see [1]).

Since the optimal value function $V^\star$ is positive definite, using [38, Lemma 4.3], the candidate Lyapunov function V_L can be bounded as

$$\underline{v_l}(\|Z^o\|) \leq V_L(Z^o,t) \leq \overline{v_l}(\|Z^o\|) \tag{26}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Z^o \in \mathbb{R}^{n+2L+2P}$, where $\underline{v_l}, \overline{v_l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ in (26) are class $\mathcal{K}$ functions. To facilitate the analysis, let $\underline{c}, \underline{b} \in \mathbb{R}_{>0}$ be constants defined as

$$\underline{c} \triangleq \frac{\beta_1}{2\overline{\Gamma}_1\eta_{c2}} + \frac{\underline{c}_2}{2}, \quad \underline{b} \triangleq \frac{\beta_2}{2\overline{\Gamma}_2 k_{c2}} + \frac{\underline{b}_2}{2}. \tag{27}$$

Let $v_l : \mathbb{R}_{\geq 0}$ be a class $\mathcal{K}$ function such that

$$v_l(||Z||) \leq \frac{q}{2}\|x\|^2 + \frac{(k_{a1}+k_{a2})}{8}\|\tilde{W}_{2a}\|^2 + \frac{k_{c2}\underline{b}}{8}\|\tilde{W}_{2c}\|^2 + \frac{(\eta_{a1}+\eta_{a2})}{8}\|\tilde{W}_{1a}\|^2 + \frac{\eta_{c2}\underline{c}}{8}\|\tilde{W}_{1c}\|^2. \tag{28}$$

Theorem 1: Provided assumption 2 is satisfied and the control gains are selected sufficiently large (see the Appendix), then the controller in (11) along with the R-MBRL and StaF update laws taking the form of (17)-(20) ensure that the state x and weight estimation errors $\tilde{W}_{1a}$, $\tilde{W}_{1c}$, $\tilde{W}_{2a}$, and $\tilde{W}_{2c}$ are semiglobally uniformly ultimately bounded.[3]

Proof: The time-derivative of the Lyapunov function is

$$\dot{V}_L = \dot{V}^\star - \tilde{W}_{1c}^T\Gamma_1^{-1}\dot{\hat{W}}_{1c} + \tilde{W}_{2c}^T\Gamma_2^{-1}(\dot{W}_2 - \dot{\hat{W}}_{2c}) - \tilde{W}_{1a}^T\dot{\hat{W}}_{1a} + \tilde{W}_{2a}(\dot{W}_2 - \dot{\hat{W}}_{2a}) + \frac{1}{2}\tilde{W}_{1c}^T\dot{\Gamma}_1^{-1}\tilde{W}_{1c} + \frac{1}{2}\tilde{W}_{2c}^T\dot{\Gamma}_2^{-1}\tilde{W}_{2c}. \tag{29}$$

[2]Typical results in ADP require excitation along the system trajectory (see [3], [7]–[9], [11], [31], [36]), which may potentially cause the system to go unstable. However, in this result, virtual excitation can be used without injecting destabilizing dither signals into the system. The sample trajectories x_i and x_j can be designed to contain enough frequencies if they are selected to follow a highly oscillatory trajectory or they are chosen from a sampling distribution such as a normal or uniform distribution.

[3]Results such as [39] could potentially be used to achieve an asymptotic convergence to the origin, but the additional feedback to eliminate the residual error would deviate from the optimal policy.

Using the chain rule, the time derivative of the ideal weights $\dot{W}_2$ can be expressed as

$$\dot{W}_2 = \nabla W_2(f(x) + g(x)u). \tag{30}$$

Provided the sufficient conditions in the Appendix are met, substituting for (13), (15)-(20), and (30), using the bounds in (24), (25), and (28), completing the squares, and using Young's inequality, the time derivative in (29) can be upper bounded as

$$\dot{V}_L \leq -v_l(\|Z\|) \quad \forall \|Z\| > v_l^{-1}(\iota). \tag{31}$$

After using (26), (31), and (38), [38, Th. 4.18] can be invoked to conclude that Z is uniformly ultimately bounded such that $\limsup_{t\to\infty}\|Z(t)\| \leq \underline{v_l}^{-1}(\overline{v_l}(v_l^{-1}(\iota)))$. Since $Z \in L_\infty$, it follows that $x, \tilde{W}_{1a}, \tilde{W}_{1c}, \tilde{W}_{2a}, \tilde{W}_{2c} \in L_\infty$. The function W_2 is a continuous function of x and $x \in L_\infty$ which implies $W_2(x) \in L_\infty$. Hence, $\hat{W}_{1a}, \hat{W}_{1c}, \hat{W}_{2a}, \hat{W}_{2c} \in L_\infty$, and $u \in L_\infty$. ∎

VII. Simulation

A. Two-State Dynamical System

To demonstrate the performance of the developed ADP method for a nonlinear system with a known value function, simulation results for a two-state dynamical system are provided. The simulation is performed for the control affine system given in (1) where $x^o = [x_1^o, x_2^o]^T$

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

and

$$g(x^o) = \begin{bmatrix} 0 \\ \cos\left(2x_1^o\right)+2 \end{bmatrix}. \tag{32}$$

The control objective is to minimize the cost functional in (2) with the instantaneous cost in (3) and the weighting matrices being $Q = I_2$ and $R = 1$. The optimal value function, $V^\star(x^o)$, and optimal control policy, $u^\star(x^o)$, for these particular dynamics and cost function are known to be $V^\star(x^o) = (1/2)x_1^{o2} + x_2^{o2}$ and $u^\star(x^o) = -(\cos(2x_1^o)+2)x_2^o$, respectively (see [3]). The regions A and A' are selected as circles around the origin such that $A = \overline{B_{1.5}(0)} \triangleq \{x^o : \|x^o\| \leq 1.5\}$ and $A' = \overline{B_{2.5}(0)} \triangleq \{x^o : \|x^o\| \leq 2.5\}$, respectively. The transition function $\lambda(x^o)$ is selected to be (8) with $\ell = 1.0$ as discussed in Section III.

To simulate the developed technique, the MBRL approach from [19] is used to learn the value function in A. The MBRL basis function vector for value function approximation in the set $A?$ is selected as $\sigma(x^o) = [x_1^{o2}, x_1^o x_2^o, x_2^{o2}]^T$, with thirteen uniformly distributed points selected in A' for BE extrapolation. To approximate the value function in $B = \chi \setminus A$, the StaF basis function vector is selected as $\phi(x^o, c(x^o)) = [x^{oT}c_1(x^o), x^{oT}c_2(x^o), x^{oT}c_3(x^o)]^T$ where $c_i(x^o) = x^o + d_i$ for $i = 1, 2, 3$. The centers of the StaF kernels are selected as $d_1 = 0.25 \times [0, 1]^T$, $d_2 = 0.25 \times [-0.886, -0.5]^T$, and $d_2 = 0.25 \times [-0.886, 0.5]^T$. To ensure sufficient excitation in B, a single trajectory $x_j^o : \mathbb{R}_{t\geq t_0} \to \mathbb{R}^n$ is selected

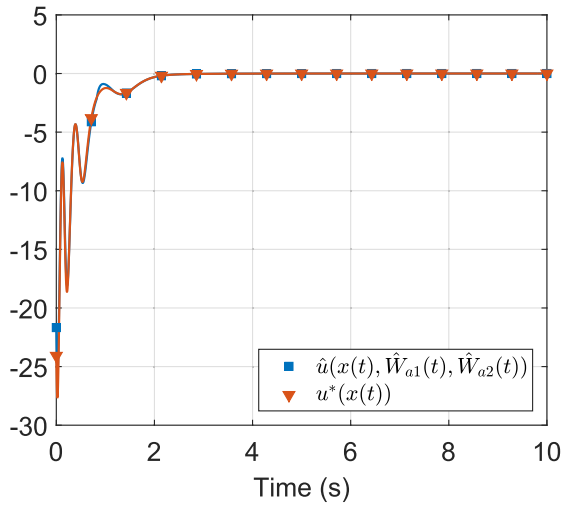

Fig. 1. Optimal control policy and estimate for the two-state system in (32).

for BE extrapolation such that at each time instant t, $x_j^o(t)$ is selected at random from a uniform distribution over a $\nu(x^o(t)) \times \nu(x^o(t))$ square centered at the current state $x^o(t)$ where $\nu(x^o(t)) = ((x^{oT}x^o + 0.01)/(1 + x^{oT}x^o))$. The initial conditions for the system at $t_0 = 0$ are

$$\begin{aligned} x(0) &= [-10, 10]^T \\ \hat{W}_{1c}(0) &= \hat{W}_{1a}(0) = 2 \times 1_3 \\ \hat{W}_{2c}(0) &= \hat{W}_{2a}(0) = 0.3 \times 1_3 \\ \Gamma_1(0) &= 350 \times I_3, \quad \Gamma_2(0) = 50 \times I_3. \end{aligned}$$

The gains for the MBRL update laws are selected as

$$\begin{aligned} \eta_{c1} &= 0.001, \quad \eta_{c2} = 2, \quad \eta_{a1} = 25 \\ \eta_{a2} &= 0.1, \quad \beta_1 = 0.5, \quad \gamma_1 = 2 \end{aligned}$$

and the gains for the StaF update laws (17), (18), and (20) are selected as

$$\begin{aligned} k_{c1} &= 0.001, \quad k_{c2} = 0.09, \quad ck_{a1} = 1.5 \\ k_{a2} &= 0.01, \quad \beta_1 = 0.003, \quad \text{and } \gamma_2 = 0.05. \end{aligned}$$

Results: Fig. 1 indicates that the control policy estimate converges to the optimal controller, while regulating the states to the origin, as seen from Fig. 2. Fig. 3 shows the value function approximation error, from which it is clear that the value function estimate $\hat{V}$ converges to the optimal value function. Fig. 4 shows that the estimated value function and policy weights for both the StaF [Fig. 4(b) and (d)] and MBRL [Fig. 4(a) and (c)] methods converge to steady-state values and remain bounded. The MBRL weights converge close to their optimal weights $W_1 = [0.5, 0, 1]^T$; however, the approximate StaF weights cannot be compared to their ideal weights because the optimal StaF weight are unknown.

B. Ten-State Dynamical System

To demonstrate the performance of the developed ADP method on a higher dimensional system, consider a centralized controller computing the control policies for a network of ten

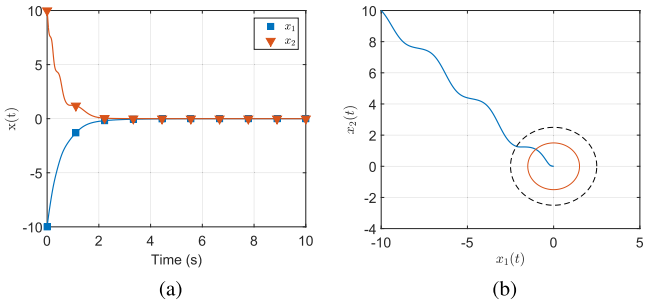

Fig. 2. State regulation and state-space portrait for the two-state dynamical system. In (b) the region A' is represented by the larger dashed circle while A is represented via the smaller circle. (a) State trajectory for the two-state system in (32). (b) Phase-space portrait for the two-state system in (32).

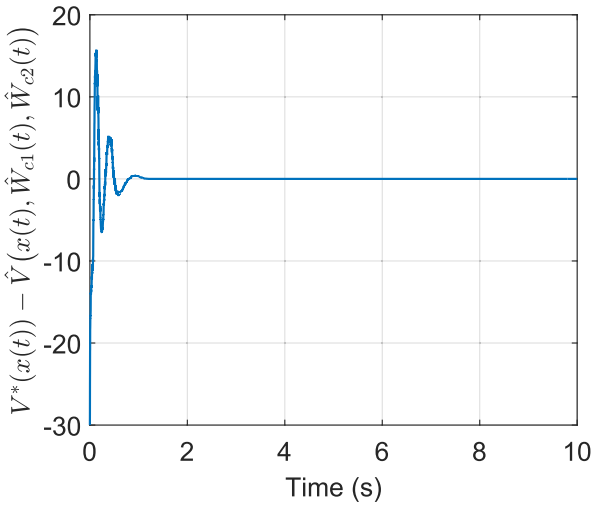

Fig. 3. Value function estimation error for the two-state system in (32).

one-state dynamical systems, where each system is in control affine form with dynamics represented as

$$\begin{aligned} f_i(x_i^o) &= (\theta_{a,i}x_i^o + \theta_{b,i}(x_i^o)^2) \\ g_i(x_i^o) &= (\cos(2x_i) + 2) \quad \forall i = 1, \ldots, 10 \end{aligned}$$

where $\theta_{a,i} = 2, 5, 0.1, 0.5, 2.5, 0.3, 0.5, 0.15, 3.5, 2$ and $\theta_{b,i} = 1, 0.5, , 1, 1, 1, 0.3, 1.1, 0.7, 0.9, 0.8$ for $i = 1, \ldots, 10$, respectively. The agent dynamics are combined to form one large dynamical system given by

$$\begin{aligned} f(x) &= \begin{bmatrix} \theta_{a,1}x_1^o + \theta_{b,1}(x_1^o)^2 \\ \vdots \\ \theta_{a,10}x_6^o + \theta_{b,10}(x_6^o)^2 \end{bmatrix} \\ g(x) &= \text{diag}[(\cos(2x_1) + 2), \ldots, (\cos(2x_{10}) + 2)]. \end{aligned} \quad (33)$$

The transition function is selected to be the same as in (8) with $A = \overline{B_1(0)} \triangleq \{x^o : \|x^o\| \leq 1\}$ and $A' = \overline{B_2(0)} \triangleq \{x^o : \|x^o\| \leq 2\}$ and $\ell = 1.0$. The control objective is to minimize the cost functional in (2) with the instantaneous cost in (3) using the weighting matrices $Q = I_{10}$ and $R = I_{10}$.

The MBRL basis is selected to be a vector of twenty polynomials, and for BE extrapolation, twenty-one equally distributed points are selected in A'. The StaF basis is selected to be $\phi(x^o, c(x^o)) = [x^{oT}c_1(x^o), \ldots, x^{oT}c_{11}(x^o)]$, where

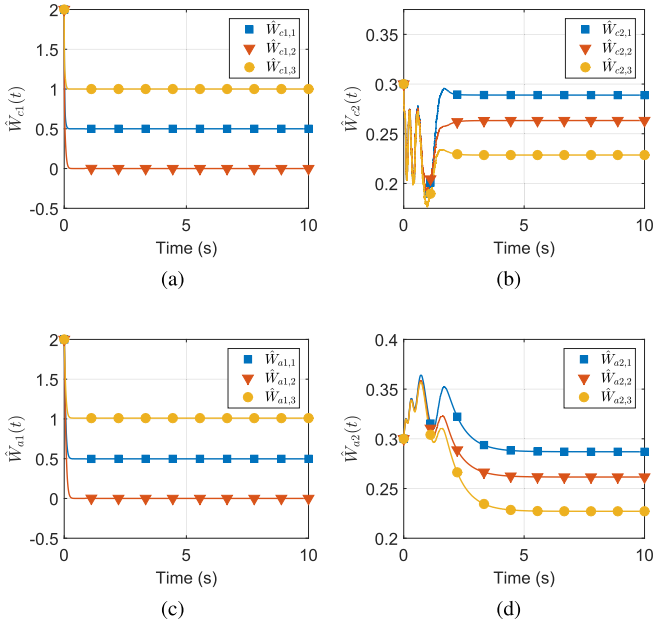

Fig. 4. Value function and policy weight approximations for the two-state system in (32). The StaF actor and critic weights are updated using (17), (18), and (20). The R-MBRL actor and critic weights are updated using adaptation schemes which take a similar form to the StaF update laws, as discussed in Section VI. (a) R-MBRL critic approximations. (b) StaF critic approximations. (c) R-MBRL actor approximations. (d) StaF actor approximations.

$c_i(x^o) = x^o + d_i$ for $i = 1, \ldots, 11$. The centers d_i are selected to be the vertices of a 10-simplex. For BE extrapolation in B, a single point is selected at random from a uniform distribution over a $[2\nu(x^o(t))]^{10}$ hypercube centered at the current state, where $\nu(x^o(t)) = (0.0003x^{oT}x^o/(1 + 0.5x^{oT}x^o))$. When the states converge to A, the StaF update laws are turned OFF to reduce computational burden. The initial conditions for the system at $t_0 = 0$ are selected as

$$\begin{aligned} x(0) &= [1.2, -0.3, 3, -2.4, -2.1, -2.7, -1.2, 1.2, \\ &\quad 0.3, -1.8]^T \\ \hat{W}_{1c}(0) &= \hat{W}_{1a}(0) = 5 \times 1_{20} \\ \hat{W}_{2c}(0) &= \hat{W}_{2a}(0) = 0.25 \times 1_{11} \\ \Gamma_1(0) &= 350 \times I_{20}, \quad \Gamma_2(0) = 100 \times I_{11}. \end{aligned}$$

The gains for the MBRL update laws are selected as

$$\begin{aligned} \eta_{c1} &= 0.0005, \quad \eta_{c2} = 30, \quad \eta_{a1} = 25 \\ \eta_{a2} &= 0.01, \quad \beta_1 = 0.06, \quad \gamma_1 = 3 \end{aligned}$$

and for the StaF update laws in (17), (18), and (20) the gains are selected as

$$\begin{aligned} k_{c1} &= 0.001, \quad k_{c2} = 0.8, \quad k_{a1} = 0.4 \\ k_{a2} &= 0.001, \quad \beta_1 = 0.0001, \quad \text{and } \gamma_2 = 0.9. \end{aligned}$$

Results: Figs. 5 and 6 show that the control policy and the system states converge to the origin. The oscillation-like effect between 0 and 1 s in Fig. 5 comes from StaF approximation in B. Fig. 7 indicates that the BE converges to zero. The transition of the BE between 0 and 1 second in Fig. 7 is attributed to the transition of the value function approximation

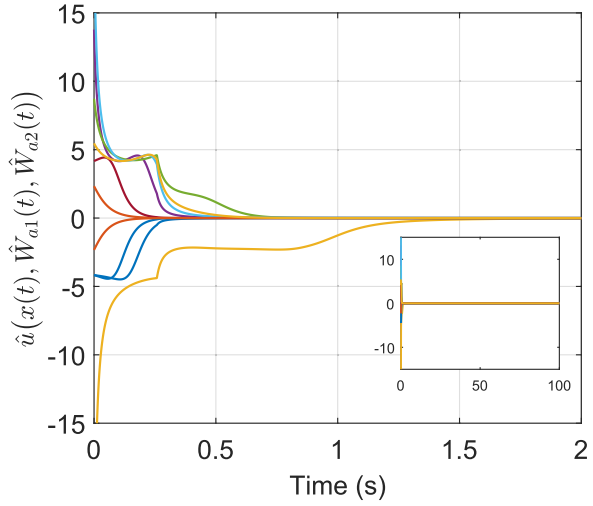

Fig. 5. Optimal control policy estimate for the ten-state dynamical system.

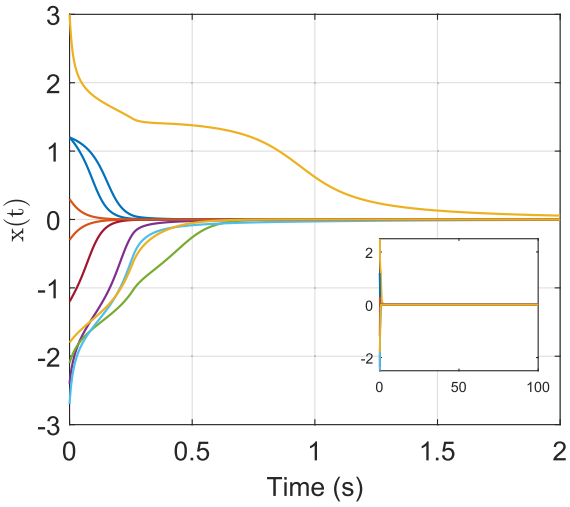

Fig. 6. States for the ten-state dynamical system converge to the origin.

weight approximation as the state enters A'. Fig. 8 shows that the approximate MBRL weights converge to steady-state values, and the StaF weights remain bounded.

C. Comparison

The developed technique is compared to the R-MBRL approximation technique in [19] and the StaF approximation technique in [1] via MATLAB Simulink running at 1000 Hz on an Intel Core i5-2500K CPU at 3.30 GHz. All systems are simulated for 100 s and the total cost, steady-state RMS error, and running time are compared, with the results displayed in Tables I–IV. The approximation method from [1] is implemented using polynomial StaF basis functions with centers at the vertices of an n-simplex for each n-dimensional problem.[4] The approximation method from [19] is implemented using polynomial basis functions selected via trial-and-error.

[4] At a minimum $n+1$ kernels need to be used with an n-dimensional system. The choice of kernel is only governed a few rules imposed by the StaF method, which can be found in [1], [21], [35]. Dot product kernels work well for the StaF application; examples include polynomial kernels and exponential kernels.

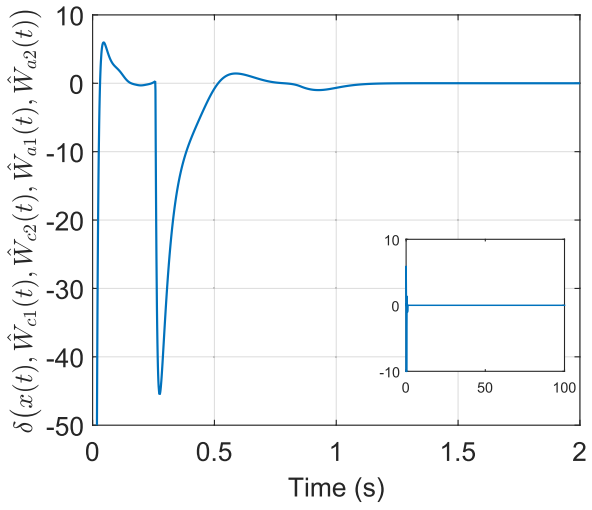

Fig. 7. BE using the developed method for a ten-state dynamical system.

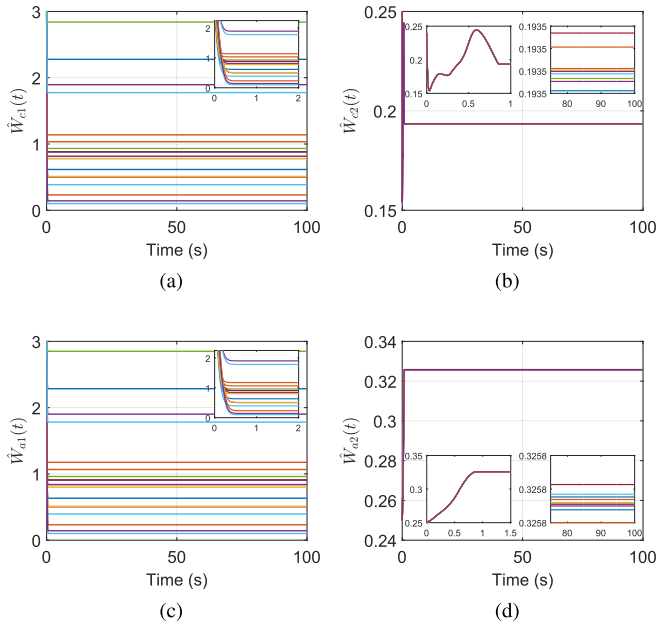

Fig. 8. Value function and policy weight approximations using the R-MBRL and StaF critic and actor update laws for the ten-state dynamical system in (33). (a) R-MBRL critic approximations. (b) StaF critic approximations. (c) R-MBRL actor approximations. (d) StaF actor approximations.

Furthermore, the sets A and A' are selected via trial-and-error to demonstrate the effect of selecting different regions.[5] It is seen that the developed technique converges similar to the R-MBRL technique in [19] but at a smaller cost and running time as the dimension of the system increases. In theory, the R-MBRL method should be closest to optimal because it provides an approximation over the entire operating domain. However, the choice of basis functions and the number of basis functions used for approximation has a major influence on the approximation. Hence, when the exact parameterization is known such as in the case of the two-state system, R-MBRL provides the smallest cost, but this is not necessarily true when the basis is not known a priori. The basis function used is directly correlated to the cost through the input; hence, basis functions with larger gradients will exhibit higher control efforts which can increase cost. An examination of the correlation between the type of basis function used and total cost for the R-MBRL method is out of the scope of this paper. The increase in running time for the R-MBRL method in [19] for the six and ten-state systems occurs because the value function is approximated over the entire domain of operation instead of just a local region around the origin, requiring a large number of basis functions. The RMS error is practically zero since all of the methods provide a sufficiently accurate approximation of the value function, resulting in a stabilizing feedback.

The StaF-only approximation and the developed approximation technique results in a similar cost for the two, three, and six-state simulation when using difference gains. But for the ten-state simulation, the cost is smaller for the developed approximation technique. The StaF method in [1] also results in a slightly higher steady-state RMS error compared to the developed method. When increasing to a higher dimensional system such as the six and ten-state systems, the StaF method in [1] results in a much shorter running time when compared to the developed method because the developed method still requires stationary basis functions around the origin, which increases the running time.

In many applications such as station keeping of marine craft, the local cost or the cost which starts being calculated once the marine craft reaches a goal region is more important than the total cost for regulating to that region and staying there. Table III displays the local cost once the system enters the set A for the developed and StaF-based methods. The developed method results in a smaller local cost compared to the StaF method in [1]. Since the R-MBRL method contains a larger number of basis functions over A' compared to the StaF method, a better approximation over A is learned, resulting in a reduced local cost.

Table IV provides a comparison of the developed method compared to the StaF method in [1] when the same gains are used and a large region A is selected with respect to the initial conditions. The results show that the StaF method has a smaller running time compared to the developed method; however, the developed method yields a lower cost compared to the StaF-only method. The developed method is capable of quickly learning the value function via BE extrapolation in the neighborhood A while the state still has not A.

Table II provides a comparison of the developed method with StaF and R-MBRL when the sets A and A' and the transition region $A' \setminus A$ are increased for the three-state dynamical system using different gains. As the sets get larger, a smaller total cost results. The lower cost is because the R-MBRL method is approximating the value function over a larger area, and hence, provides a more accurate approximation compared to the local approximations of the StaF method. As the sets A and A' are increased, the developed method produces a smaller total cost compared to the R-MBRL method in [19],

[5]The performance of the proposed method depends on the choice of A and A'. Hence, if the initial conditions are far from the origin then larger sets may be used, otherwise the sets A and A' should be smaller to provide enough time for the R-MBRL weights to be learned.

TABLE I

SIMULATION RESULTS. STEADY-STATE RMS ERRORS BELOW 1×10^{-16} ARE CONSIDERED TO BE ZERO. (a) TWO AND THREE-STATE SIMULATION RESULTS. (b) SIX AND TEN-STATE SIMULATION RESULTS

	Two-State Dynamical System			Three-State Dynamical System		
Controller	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [19]	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [19]
Total cost	150.55	150.62	150.50	24.85	25.57	24.55
RMS steady-state error	0	1.66×10^{-2}	0	0	1.19×10^{-4}	0
Running time (sec)	4.91	2.95	4.11	11.49	3.40	15.57

(a)

	Six-State Dynamical System			Ten-State Dynamical System		
Controller	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [19]	R-MBRL + StaF	StaF controller in [1]	R-MBRL controller in [19]
Total cost	37.72	39.56	59.12	60.22	65.43	88.30
RMS steady-state error	0	1.81×10^{-8}	1.7×10^{-3}	0	2.76×10^{-10}	0
Running time (sec)	28.12	5.57	73.6	84.34	9.77	217.98

(b)

TABLE II

THREE-STATE SIMULATION RESULTS WITH DIFFERENT SETS A AND A'. STEADY-STATE RMS ERRORS BELOW 1×10^{-16} WERE CONSIDERED TO BE ZERO

	Three-State Dynamical System $x(0) = [-3, 3, -2.5]^T$				
Controller	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 0.25\}$ $A' = \{x^o : \|x^o\| \leq 1.25\}$	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 1.5\}$ $A' = \{x^o : \|x^o\| \leq 2.75\}$	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 3.0\}$ $A' = \{x^o : \|x^o\| \leq 4.5\}$	StaF controller in [1]	R-MBRL controller in [19]
Total cost	24.85	23.98	23.27	25.57	24.55
RMS steady-state error	0	0	0	1.19×10^{-4}	0
Running time (sec)	11.49	11.49	11.59	3.40	9.82

TABLE III

LOCAL COST WHEN THE SYSTEM ENTERS THE SET A FOR THE DEVELOPED METHOD AND THE STAF-BASED METHOD IN [1]

	Two-State Dynamical System		Three-State Dynamical System		Six-State Dynamical System		Ten-State Dynamical System	
Controller	R-MBRL + StaF	StaF in [1]	R-MBRL + StaF	StaF in [1]	R-MBRL + StaF	StaF in [1]	R-MBRL + StaF	StaF in [1]
Total cost	150.55	150.62	24.85	25.57	37.72	39.56	60.22	65.43
Local cost	1.71	1.76	0.11	0.27	0.46	1.97	0.75	1.58

TABLE IV

SIX-STATE SIMULATION RESULTS WITH DIFFERENT SETS A AND A' UNDER DIFFERENT INITIAL CONDITIONS USING THE SAME GAINS FOR THE UPDATE LAWS (17), (18), AND (20). STEADY-STATE RMS ERRORS BELOW 1×10^{-16} WERE CONSIDERED TO BE ZERO

	Six-State Dynamical System			
	$\|x(0)\| = 6.61$		$\|x(0)\| = 13.23$	
Controller	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 4\}$ $A' = \{x^o : \|x^o\| \leq 5\}$	StaF controller in [1]	R-MBRL + StaF $A = \{x^o : \|x^o\| \leq 10\}$ $A' = \{x^o : \|x^o\| \leq 10.5\}$	StaF controller in [1]
Total cost	21.59	27.78	83.55	111.57
Local cost	6.95	11.98	39.75	70.14
RMS steady-state error	0	1.61×10^{-9}	0	1.90×10^{-9}
Running time (sec)	26.46	5.35	26.86	5.16

this is partially attributed to the fact that implementation of R-MBRL over a large region is challenging when an exact basis for value function approximation is not available. As the transition region $A' \setminus A$ increases, the gradient of λ decreases, possibly contributing to the smaller cost. Also in [19], the least-squares learning gain matrix $\Gamma_1(t)$ was updated without using recorded data, while the developed R-MBRL update law similar to (18) includes recorded data to improve the selection of $\Gamma_1(t)$.

The results in Tables I–IV indicate that the optimal choice of the approximation method depends on the circumstance, and several advantages and disadvantages need to be taken into consideration when selecting which method to use. The StaF method is best suited for a high-dimensional application

requiring real-time performance where global optimality is not required. However, Table IV shows that there are circumstances in which the developed method outperforms the StaF method in [1] in terms of total and local cost. Moreover, since the StaF method in [1] lacks memory, the weights need to be relearned every time the system passes through the predefined area of interest in the operating domain, whereas the developed uses the R-MBRL method to learn to static weights in that region and does not need to relearn the weights when the system leaves the neighborhood. The R-MBRL method in [19] is the best suited for lower dimensional applications where global optimality is a premium. However, approximating the value function over the entire state space requires a large number of basis functions, and hence, a large computational burden. Since the developed method reduces the area of interest, which reduces the number of basis functions required, it is computationally efficient when compared to R-MBRL in [19]. Applications with large operating domains may benefit from the developed method since the value function can be learned in desired areas of the state space, e.g., around the origin, independent of where the state is, using R-MBRL, while StaF keeps the system stable by approximating the value function around the state trajectory. Although the developed method shows a slight improvement over [1] in terms of cost and RMS error, more tuning parameters and an overall larger number of unknown parameters are introduced. Having more tuning parameters provides freedom for the designer to select different parameters in the tuning process and also allows for a better approximation as shown by the RMS error in Tables I–IV. However, having multiple update laws also increases computational complexity. Although the overall computational complexity depends on the total number of basis functions used for approximation and the number of off-policy trajectories used for BE extrapolation, the parameters are design choices. Hence, as shown in Tables I–IV, depending on the circumstance, such as the size of the regional approximation area or the system complexity, it is beneficial to have both local and regional update laws to approximate the value function.

VIII. Conclusion

An infinite horizon optimal control problem was solved using a novel approximation methodology utilizing the StaF kernel method and an R-MBRL method. The operating domain χ of the system was segregated into two parts; a neighborhood, $A \subset \chi$, containing the origin where R-MBRL was employed, and the set $B = \chi \setminus A$ where the StaF method was employed. For a state initialized in B, the StaF method ensured stable and computationally efficient operation while an R-MBRL method achieved a sufficiently accurate estimate of the value function over the set A. When the state entered A, the R-MBRL technique was used to regulate the state to the origin.

Under specific conditions, Theorem 1 established that the developed control strategy results in semiglobal uniform ultimate boundedness of the state trajectory. Simulation examples for two, three, six, and ten-state dynamical systems showed that the developed approximation method outperforms previous methods. As the dimension of the system increases, the developed method is able to estimate the value function sufficiently to reduce the local cost and the RMS error.

To ensure smooth transition between the two approximate optimal controllers as the state transitioned from B to A, a state varying convex combination of the two controllers was used based on the distance from set A. However, the convex combination in the approximation approach resulted in the need for large gains. A possible subject for future research would be a switched-systems-based modification to the developed method which employs a buffer to allow for a sufficient dwell time in the transition region B to A.

In the developed method, the sets A and A' were selected by trial and error to demonstrate the difference in performance. The rate at which the optimal value function is learned in A depends on the size and location of A' in the state space. A possible subject of future research would be to investigate designing time-varying sets A and A'. Moreover, investigations into finding optimal sets A and A' remain as topics for future research.

Appendix A
Sufficient Conditions

In the following, the notation $\overline{\|(\cdot)\|}$ is defined as $\overline{\|h\|} \triangleq \sup_{\xi \in B_\zeta} \|h(\xi)\|$, for some continuous function $h : \mathbb{R}^n \to \mathbb{R}^k$, where $B_\zeta \subset \mathbb{R}^{n+2L+2P}$ denotes a closed ball with radius ζ centered at the origin. The sufficient conditions that facilitate the stability analysis are given by

$$\frac{(k_{a1}+k_{a2})}{2} \geq \left(\left(\frac{k_{c1}}{\sqrt{\gamma_2}}+\frac{\eta_{c1}}{\sqrt{\gamma_1}}\right)\vartheta_5+\frac{k_{c1}}{\sqrt{\gamma_2}}\vartheta_6\right)\overline{v}_l\left(\|Z(t_0)\|\right) + 2\vartheta_1+\vartheta_2+\frac{\vartheta_4\overline{\|W_2\|}}{\sqrt{\gamma_2}} \tag{34}$$

$$\frac{(\eta_{a1}+\eta_{a2})}{2} \geq \left(\frac{\eta_{c1}}{\sqrt{\gamma_1}}\vartheta_6+\left(\frac{\eta_{c1}}{\sqrt{\gamma_1}}+\frac{k_{c1}}{\sqrt{\gamma_2}}\right)\vartheta_7\right)\overline{v}_l(\|Z(t_0)\|) + \left(\frac{1}{\underline{\Gamma_2}}+1\right)\vartheta_2+\frac{\vartheta_3\overline{W_1}}{\sqrt{\gamma_1}} \tag{35}$$

$$\frac{k_{c2}\underline{b}}{4} \geq \max\left\{\frac{\vartheta_2}{2\underline{\Gamma_2}}+\frac{k_{c1}(\vartheta_5+\vartheta_6+\vartheta_7)}{4\sqrt{\gamma_2}}+\frac{\vartheta_{10}}{2}, \frac{(k_{a1}+\vartheta_9)^2}{(k_{a1}+k_{a2})}\right\} \tag{36}$$

$$\frac{\eta_{c2}\underline{c}}{4} \geq \max\left\{\frac{\eta_{c1}(\vartheta_5+\vartheta_6+\vartheta_7)}{4\sqrt{\gamma_1}}+\frac{\vartheta_{10}}{2}, \frac{\left(\eta_{a1}+\frac{\vartheta_3}{2\sqrt{\gamma_1}}\overline{W_1}\right)^2}{(\eta_{a1}+\eta_{a2})}\right\} \tag{37}$$

and

$$v_l^{-1}(\iota) < \overline{v_l}^{-1}(\underline{v_l}(\zeta)). \tag{38}$$

In (34)–(38), the constants ι, $\{\vartheta_i | i = 1, \ldots 12\} \in \mathbb{R}_{>0}$ are defined as

$$\vartheta_1 = \frac{\overline{\|G\nabla W_2 \nabla\phi\|}}{2}+\frac{\overline{\|G\nabla W_2\nabla\lambda\phi^T\|}}{2}$$

$$\vartheta_2 = \frac{\overline{\|G\nabla W_2\nabla\sigma\|}}{2}+\frac{\overline{\|G\nabla W_2\nabla\lambda\sigma^T\|}}{2}$$

$$\vartheta_3 = \frac{\eta_{c1}+\eta_{c2}}{4}\overline{\|G_{\nabla\sigma}\|}, \quad \vartheta_4 = \frac{k_{c1}+k_{c2}}{4}\overline{\|G_{\nabla\phi}\|}$$

$$\vartheta_5 = \frac{\overline{\|G_{\nabla\phi}\|}}{2} + \frac{\overline{\|\phi G_{\nabla\lambda\nabla\phi}\|}}{4} + \frac{\overline{\|\phi G_{\nabla\lambda\nabla\lambda}\phi^T\|}}{4}$$

$$\vartheta_6 = \frac{\overline{\|G_{\nabla\phi\nabla\sigma}\|}}{2} + \frac{\overline{\|\phi G_{\nabla\lambda\nabla\lambda}\sigma^T\|}}{2} + \frac{\overline{\|\phi G_{\nabla\lambda\nabla\sigma}\|}}{2} + \frac{\overline{\|\sigma G_{\nabla\lambda\nabla\phi}\|}}{2}$$

$$\vartheta_7 = \frac{\overline{\|G_{\nabla\sigma}\|}}{2} + \frac{\overline{\|\sigma G_{\nabla\lambda\nabla\lambda}\sigma^T\|}}{4} + \frac{\overline{\|\sigma G_{\nabla\lambda\nabla\sigma}\|}}{2}$$

$$\vartheta_8 = \overline{\|\nabla W_2 f\|} + \frac{1}{2}\overline{\|G_{\nabla V^\star\nabla\phi}\|} + \frac{1}{2}\overline{\|G_{\nabla V^\star\nabla\lambda}\phi^T\|} + \vartheta_2\overline{W_1} + \vartheta_1\overline{\|W_2\|} + \frac{\vartheta_4}{2\sqrt{\gamma_2}}\overline{\|W_2\|}^2$$

$$\vartheta_9 = \frac{\vartheta_1}{\underline{\Gamma_2}} + \frac{\vartheta_4}{2\sqrt{\gamma_2}}\overline{\|W_2\|}$$

$$\vartheta_{10} = \frac{k_{c1}}{2\sqrt{\gamma_2}}\overline{\|\omega_{\nabla\sigma}\|} + \frac{\eta_{c1}}{2\sqrt{\gamma_1}}\overline{\|\omega_{\nabla\phi}\|}$$

$$\vartheta_{11} = \frac{\vartheta_2}{\underline{\Gamma_2}}\overline{W_1} + \frac{\vartheta_1}{\underline{\Gamma_2}}\overline{\|W_2\|} + \frac{\overline{\|\nabla W_2 f\|}}{\underline{\Gamma_2}}$$

$$\vartheta_{12} = \frac{\overline{\|G_{\nabla V^\star\nabla\sigma}\|}}{2} + \frac{\overline{\|G_{\nabla V^\star\nabla\lambda}\sigma^T\|}}{2} + \eta_{a2}\overline{W_1} + \frac{\vartheta_3}{2\sqrt{\gamma_1}}\overline{W_1}^2$$

and

$$\iota = \frac{\overline{\|G_{\nabla V^\star\nabla W_2\phi}\|}}{2} + \frac{\overline{\|G_{\nabla V^\star\nabla\epsilon}\|}}{2} + \frac{(\vartheta_{12})^2}{(\eta_{a1}+\eta_{a2})} + \frac{(k_{a2}+\vartheta_8)^2}{(k_{a1}+k_{a2})} + \frac{\left(\frac{\eta_{c1}}{2\sqrt{\gamma_1}}(2\overline{\|\Delta_1\|}+\overline{\|\Delta_2\|}+\overline{\|\Delta_3\|})\right)^2}{\eta_{c2}\underline{c}} + \frac{\left(\frac{k_{c1}}{2\sqrt{\gamma_2}}(\overline{\|\Delta_1\|}+2\overline{\|\Delta_2\|}+\overline{\|\Delta_3\|})+\vartheta_{11}\right)^2}{k_{c2}\underline{b}}.$$

The sufficient condition in (34) can be satisfied by increasing the gain k_{a2}. This will not affect the sufficient conditions in (35) and (37) and it may decrease the sufficient condition in (36). The sufficient condition in (35) can be satisfied without affecting the sufficient conditions (34) and (36) by increasing the gain η_{a2}. The sufficient condition in (36) can be satisfied by selecting points for BE extrapolation in $B \subset \chi \setminus A$ so that the minimum eigenvalue $\underline{b}$ in (27) is large enough and by increasing the gain k_{a2}. By selecting points for BE extrapolation in $A \subset \chi$ such that the minimum eigenvalue, $\underline{c}$, is large enough, and a large η_{a2}, the sufficient condition in (37) can be satisfied. Provided the transition function λ is selected such that $\overline{\nabla\lambda}$ is small, the basis functions used for approximation are selected such that $\overline{\|\epsilon\|}$, $\overline{\|\nabla\epsilon\|}$, and $\overline{\|\nabla W_2\|}$ are small, and k_{a2}, η_{a2}, $\underline{c}$, and $\underline{b}$ are selected to be sufficiently large, then the sufficient condition in (38) can be satisfied.[6]

[6]The minimum eigenvalue of $(1/N)\sum_{i=1}^{N}(\omega_{\nabla\sigma i}(t)\omega_{\nabla\sigma i}^T(t)/\rho_{1i}^2(t))$ can be increased by collecting redundant data, i.e., selecting $N \gg P$ in the area of interest, where the extrapolated off-policy trajectories x_i can be selected a priori based on the desired behavior of the system. The bound on the gradient of λ, i.e., $\overline{\nabla\lambda}$, can be decreased by selecting larger transition regions $A' \setminus A$. The size of the steady-state error can be decreased by increasing the number of basis functions used for approximation (i.e., increasing L and P). However, increasing the number of basis function to gain a better approximation comes at a cost of an increase in computational complexity.

ACKNOWLEDGMENT

Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

REFERENCES

[1] 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.

[2] K. G. Vamvoudakis and F. L. Lewis, "Online synchronous policy iteration method for optimal control," in *Recent Advances in Intelligent Control Systems*, W. Yu, Ed. London, UK: Springer, 2009, pp. 357–374.

[3] 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, May 2010.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[5] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.*, vol. 22, no. 3, pp. 237–246, 2009.

[6] P. Mehta and S. Meyn, "Q-learning and Pontryagin's minimum principle," in *Proc. IEEE Conf. Decision Control*, Dec. 2009, pp. 3598–3605.

[7] Q.-Y. Fan and G.-H. Yang, "Nearly optimal sliding mode fault-tolerant control for affine nonlinear systems with state constraints," *Neurocomputing*, vol. 216, pp. 78–88, Dec. 2016.

[8] Q. Fan and G. Yang, "Adaptive actor–critic design-based integral sliding-mode control for partially unknown nonlinear systems with input disturbances," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 165–177, Jan. 2016.

[9] H. Modares, F. L. Lewis, and M.-B. N. Sistani, "Online solution of nonquadratic two-player zero-sum games arising in the H_∞ control of constrained input systems," *Int. J. Adapt. Control Signal Process.*, vol. 28, nos. 3–5, pp. 232–254, 2014.

[10] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control Algorithms and Stability* (Communications and Control Engineering). London, U.K.: Springer-Verlag, 2013.

[11] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.

[12] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[13] T. Dierks, B. T. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Netw.*, vol. 22, nos. 5–6, pp. 851–860, 2009.

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

[15] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[16] G. Chowdhary, M. Mühlegg, and E. Johnson, "Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation," *Int. J. Control*, vol. 87, no. 8, pp. 1583–1603, 2014.

[17] G. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, School Aerosp. Eng., Georgia Inst. Technol., Atlanta, AZ, USA, Dec. 2010.

[18] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[19] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, Feb. 2016.

[20] 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.

[21] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "State following (StaF) kernel functions for function approximation Part I: Theory and motivation," in *Proc. Amer. Control Conf.*, 2015, pp. 1217–1222.

[22] L. Dong, X. Zhong, C. Sun, and H. He, "Event-triggered adaptive dynamic programming for continuous-time systems with control constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1941–1952, Aug. 2016.
[23] X. Zhong, H. He, D. Wang, and Z. Ni, "Model-free adaptive control for unknown nonlinear zero-sum differential game," *IEEE Trans. Cybern.*, to be published.
[24] G. Xiao, H. Zhang, Y. Luo, and Q. Qu, "General value iteration based reinforcement learning for solving optimal tracking control problem of continuous–time affine nonlinear systems," *Neurocomputing*, vol. 245, pp. 114–123, Jul. 2017.
[25] J. Wang, T. Yang, G. Staskevich, and B. Abbe, "Approximately adaptive neural cooperative control for nonlinear multiagent systems with performance guarantee," *Int. J. Syst. Sci.*, vol. 48, no. 5, pp. 909–920, 2017.
[26] C. Mu, Z. Ni, C. Sun, and H. He, "Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 584–598, Mar. 2017.
[27] I. Steinwart and A. Christmann, *Support Vector Machines*. New York, NY, USA: Springer, 2008.
[28] H. Wendland, "Scattered data approximation," in *Cambridge Monographs on Applied and Computational Mathematics*, vol. 17. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[29] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, vol. 12. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[30] P. S. Walters, "Guidance and control of marine craft: An adaptive dynamic programming approach," Ph.D. dissertation, Dept. Mech. Aerosp. Eng., Univ. Florida, Gainesville, FL, USA, 2015.
[31] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and E. W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, 2013.
[32] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA, USA: SIAM, 2002.
[33] V. Stepanyan and N. Hovakimyan, "Robust adaptive observer design for uncertain systems with bounded disturbances," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1392–1403, Sep. 2007.
[34] P. Walters, R. Kamalapurkar, L. Andrews, and W. E. Dixon, "Online approximate optimal path-following for a mobile robot," in *Proc. IEEE Conf. Decision Control*, Dec. 2014, pp. 4536–4541.
[35] P. Walters, R. Kamalapurkar, and W. E. Dixon, "Approximate optimal online continuous-time path-planner with static obstacle avoidance," in *Proc. IEEE Conf. Decision Control*, Dec. 2015, pp. 650–655.
[36] Y. Lv, J. Na, Q. Yang, X. Wu, and Y. Guo, "Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics," *Int. J. Control*, vol. 89, no. 1, pp. 99–112, 2016.
[37] P. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
[38] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[39] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, Oct. 2008.

Patryk Deptula received the B.Sc. degree (Hons.) in mechanical engineering (major) and mathematics (minor) from Central Connecticut State University, New Britain, CT, USA, in 2014. He is currently pursuing the Ph.D. degree under the supervision of Dr. Dixon from the University of Florida, Gainesville, FL, USA.

He performed research related to hybrid propellant rocket engines and completed internships at Norgren-KIP Fluid Controls and Belcan Corporation, Farmington, CT, USA, as an undergraduate. He was an Engineer with the Control and Diagnostic Systems Group, Belcan Corporation, Cincinnati, OH, USA, where he analyzed aircraft engine software. His current research interests include, but are not limited to, nonlinear controls and robotics applied to a variety of fields.

Joel A. Rosenfeld received the Ph.D. degree in mathematics from the University of Florida, Gainesville, FL, USA, in 2013, under the supervision of Dr. M. T. Jury.

He joined as a Post-Doctoral Researcher with the Nonlinear Controls and Robotics Group, Department of Mechanical and Aerospace Engineering, University of Florida, in 2013, with a focus on approximation problems in control theory. In 2017, he joined the VeriVital Laboratory, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA, as a Post-Doctoral Researcher.

Rushikesh Kamalapurkar received the M.S. and Ph.D. degrees from the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA, in 2011 and 2014, respectively.

After working for a year as a Post-Doctoral Researcher with Dr. W. E. Dixon, he was appointed as the MAE PostDoctoral Teaching Fellow for 2015–2016. In 2016, he joined the School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK, USA, as an Assistant Professor. He has published three book chapters, over 20 peer-reviewed journal papers, and over 20 peer-reviewed conference papers. His current research interests include the intelligent, learning-based optimal control of uncertain nonlinear dynamical systems.

His work has been recognized by the 2015 University of Florida Department of Mechanical and Aerospace Engineering Best Dissertation Award and the 2014 University of Florida Department of Mechanical and Aerospace Engineering Outstanding Graduate Research Award.

Warren E. Dixon (M'94–F'16) received the Ph.D. degree from the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA, in 2000.

He was a Research Staff Member and a Eugene P. Wigner Fellow with the Oak Ridge National Laboratory, Oak Ridge, TN, USA, until 2004. In 2004, he joined the Mechanical and Aerospace Engineering Department, University of Florida. His current research interests include the development and application of Lyapunov-based control methods for uncertain nonlinear systems.

Dr. Dixon is an American Society of Mechanical Engineers (ASME) Fellow, an IEEE Control Systems Society Distinguished Lecturer, and served as the Director of Operations for the Executive Committee of the IEEE CSS Board of Governors from 2012 to 2015. Some key research recognitions include the the 2006 IEEE Robotics and Automation Society Early Academic Career Award, the 2009 and 2015 American Automatic Control Council O. Hugo Schuck (Best Paper) Award, the 2011 ASME Dynamics Systems and Control Division Outstanding Young Investigator Award, the 2013 Fred Ellersick Award for Best Overall MILCOM Paper, and the NSF CAREER Award (2006–2011). He also received the Air Force Commander's Public Service Award in 2016 for his contributions to the U.S. Air Force Science Advisory Board.